TPC Benchmark™ H
Full Disclosure Report

VMware® ESX™ Using
ParAccel Analytic Database™

Submitted for Review
Report Date: April 11, 2010
TPC Benchmark H Full Disclosure Report
Pricing revision: August 24, 2010

First Edition – April 2010

VMware, Inc., the Sponsor of this benchmark test, believes that the information in this document is accurate as of the publication date.

The information in this document is subject to change without notice. The Sponsor assumes no responsibility for any errors that may appear in this document.

The pricing information in this document is believed to accurately reflect the current prices as of the publication date. However, the Sponsor provides no warranty of the pricing information included in this document.

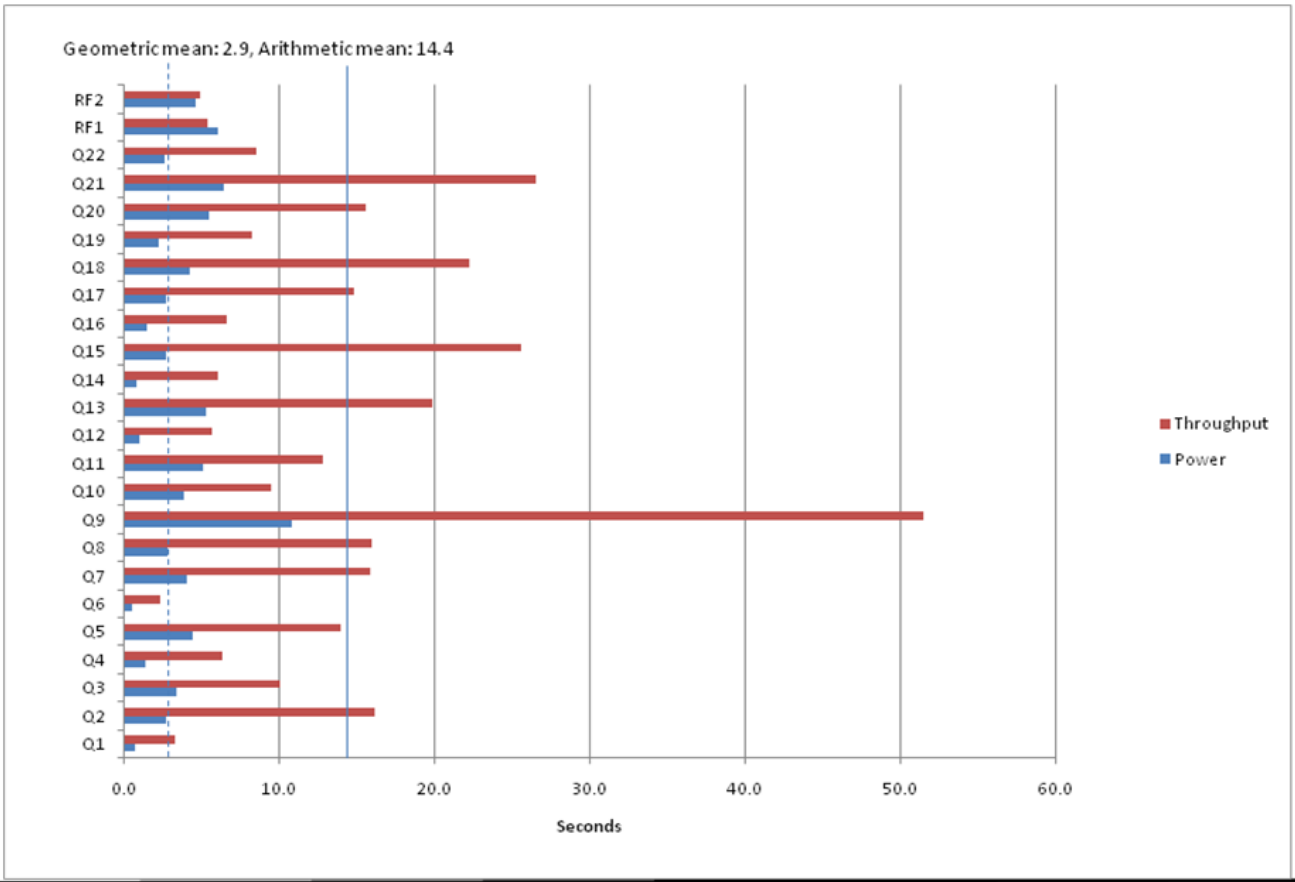
Benchmark results are highly dependent upon workload, specific application requirements, and system design and implementation. Relative system performance will vary as a result of these and other factors. Therefore, the TPC Benchmark H should not be used as a substitute for a specific customer application benchmark when critical capacity planning and/or product evaluation decisions are contemplated.

All performance data contained in this report was obtained in a rigorously controlled environment. Results obtained in other operating environments may vary significantly. No warranty of system performance or price/performance is expressed or implied in this report.

VMware ESXi™ Platform HP DL380 ParAccel Analytic Database	TPC-H Rev. 2.9.0 TPC Pricing Specification 1.5.0 Report Date: April 11, 2010

Total System Cost	Composite Query per Hour Metric	Price/Performance
$ 1,168,311.44 USD	1,316,882.1 QphH@1000GB	$ 0.89 USD $/QphH@1000GB

Database Size	Database Manager	Operating System	Other Software	Availability Date
1000GB	ParAccel Analytic Database 2.5	ParAccel Standard Linux	VMware ESXi 4.0	June 30, 2010



Database Load Time = 0:16:23
Load Includes Backup: N
Total Data Storage/Database Size = 89.4
Memory/Database Size (%) = 288.0

Storage Redundancy Levels:
Base Tables: Level One
Auxiliary Data Structures: N/A
DBMS Temporary Space: Level One
OS and DBMS Software: Level One

System Configuration: 40 x HP DL380 G6 Server, each system with
- 2 VMware ESXi VMs
- 2 Xeon x5560 2880 Mhz processors, 8 cores, 8 threads
- 72GB memory
- 8 x 300GB disks

Total processors: 80 CPUs, 320 cores, 320 threads
Total memory: 2,880GB
Total Storage: 89,406GB (in this calculation 1GB is defined as 1024*1024*1024 bytes)

Description	Part Number	Source	Unit Price	Qty	Ext. Price	3 Yr Maint
HP DL380 G6 X5560 PERF	CPR-491315-001	2	$ 4,889.00	40	$ 195,560.00	
HP 4GB 2RX4 PC3-10600R-9 KIT	CPR-500658-B21	2	$ 196.40	720	$ 141,408.00	
HP 300GB 3G PLUG SAS 10K DP SFF HD	CPR-492620-B21	2	$ 408.25	320	$ 130,640.00	
HP NC382T PCIE DP GBIT SVR ADPTR	CPR-458492-B21	2	$ 176.00	40	$ 7,040.00	
HP DL380 G6 3 SLOT PCI-E RISER KIT	CPR-500579-B21	2	$ 85.00	40	$ 3,400.00	
HPE 3YR 24X7X4 DL380	HPE-U4545E	2	$ 738.00	40		$ 29,520.00
CDW HARDWARE INSTALL SERVER	CTS-HWINSTALLSERVER	2	$ 25.00	40	$ 1,000.00	
PRIORITY SERVICE FOR INSTALL	CTS-PRIORITY-INSTALL	2	$ 12.50	40	$ 500.00	
HP ILO ADV 1 SRV TSU SW	CPR-512485-B21	2	$ 279.00	40	$ 11,160.00	
BELKIN 10'CAT5E RJ45M PATCH CAB BLUE	BEL-A3L781-10BL-CDW	2	$ 2.35	480	$ 1,128.00	
CISCO CATALYST 3750 48PT 10/100/1000	CIS-WS-C3750G-48TS-S	2	$ 8,579.00	4	$ 34,316.00	
CISCO SMARTNET OS 24X7X4	CD1-CON-OSP-3750G48T	2	$ 1,665.00	12		$ 19,980.00
BELKIN ENTERPRISE 4 POST RACK 42U 7'	BKN-RK4000	2	$ 425.00	3	$ 1,275.00	
APC RACK PDU METERED 2U 30A 208V	AME-AP7811	2	$ 535.00	5	$ 2,675.01	
ACER V173 BB 17"	ET.BV3RP.B01	2	$ 115.00	1	$ 115.00	
LOGITECH DELUXE 250 DT KB&MOU BLK	967973-0403	2	$ 19.00	1	$ 19.00	
RECYCLING FEE	(for Acer)	2	$ 19.00	1	$ 19.00	
				SubTotal	$ 579,755.01	
				Sales tax	$ 46,264.43	
				Total	$ 626,019.44	
ParAccel Analytic Database	PADB-V250	3	$ 400,000.00	1	$ 400,000.00	
ParAccel Analytic Database 1yr 24/7 Support	PADB-V250-S	3	$ 96,000.00	3		$ 288,000.00
				Total	$ 400,000.00	
		3	System Discount(40%)		($ 275,200.00)	
VMware vSphere 4 Standard Edition	VS4-STD-C	1	$ 795.00	80	$ 63,600.00	
VMware vSphere 3yr 24/7 support	VS4-STD-3P-SSS-C	1	$ 823.65	80	$ 65,892.00	
				Total	$ 129,492.00	
				Grand Total	$ 1,168,311.44	
Vendor	**Source key**					
VMware	1					
CDW	2					
ParAccel	3					

Audited by: Francois Raab, InfoSizing, Inc. (www.sizing.com)

Prices used in TPC benchmarks reflect the actual prices a customer would pay for a one -time purchase of the stated components. Individually negotiated discounts are not permitted. Special prices based on assumptions about past or future purchases are not permitted. All discounts reflect standard pricing policies for the listed components. For complete details, see the pricing sections of the TPC benchmark specifications. If you find that the stated prices are not available according to these terms, please inform at pri cing@tpc.org. Thank you.

Measurement Results	
Database Scale Factor	1000GB
Total Data Storage / Database Size	89.4
Memory/Database Size (%)	288.0
Start of database load time	21:40:34
End of database load time	21:56:57
Database Load Time	16Min 23sec
Query Streams for Throughput Test	7
TPC-H Power	1,260,595.1
TPC-H Throughput	1,375,682.4
TPC-H Composite Query-per-Hour Rating (QphH@1000GB)	1,316,882.1
Total System Price Over 3 Years	$1,168,311.44
TPC-H Price/Performance Metric ($/QphH@1000GB)	$0.89
Measurement Intervals	
Measurement Interval in Throughput Test	403 sec

Duration of Stream Execution

Stream ID	Seed	Start Date	Start Time	End Date	End Time	Duration
Stream 0	322215657	3/22/2010	21:59:36	3/22/2010	22:01:02	0:01:26
Stream 1	322215658	3/22/2010	22:01:02	3/22/2010	22:06:19	0:05:17
Stream 2	322215659	3/22/2010	22:01:02	3/22/2010	22:06:32	0:05:30
Stream 3	322215660	3/22/2010	22:01:02	3/22/2010	22:05:48	0:04:46
Stream 4	322215661	3/22/2010	22:01:02	3/22/2010	22:06:28	0:05:26
Stream 5	322215662	3/22/2010	22:01:02	3/22/2010	22:06:25	0:05:23
Stream 6	322215663	3/22/2010	22:01:02	3/22/2010	22:06:16	0:05:14
Stream 7	322215664	3/22/2010	22:01:02	3/22/2010	22:06:28	0:05:26
Refresh		3/22/2010	22:06:32	3/22/2010	22:07:45	0:01:13

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
Stream0	0.8	2.7	3.4	1.4	4.5	0.5	4.1	2.8	10.8	3.8	5.1	1.1
Stream1	3.3	17.1	13.3	3.5	12.2	3.5	21.5	13.9	56.2	6.3	17.9	4.3
Stream2	3.8	20.1	11.8	7.0	9.2	1.5	17.3	21.9	48.4	14.3	13.9	4.7
Stream3	1.8	11.2	9.3	7.1	12.8	1.5	10.0	16.3	44.8	5.2	11.7	6.3
Stream4	4.8	17.4	6.1	5.0	16.6	1.6	20.9	12.8	58.9	9.8	14.9	6.4
Stream5	3.7	10.6	7.1	7.2	17.1	3.5	16.0	20.4	34.6	7.4	10.0	11.6
Stream6	3.2	20.7	8.7	5.4	13.2	1.9	12.6	15.8	61.1	13.9	11.8	4.6
Stream7	2.7	15.8	14.0	9.2	16.7	2.9	12.9	10.3	56.1	9.6	9.8	1.7
Avg	3.3	16.1	10.0	6.3	14.0	2.4	15.9	15.9	51.4	9.5	12.9	5.6
Min	1.8	10.6	6.1	3.5	9.2	1.5	10.0	10.3	34.6	5.2	9.8	1.7
Max	4.8	20.7	14.0	9.2	17.1	3.5	21.5	21.9	61.1	14.3	17.9	11.6

	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20	Q21	Q22	RF1	RF2
Stream0	5.3	0.9	2.7	1.5	2.8	4.3	2.2	5.5	6.5	2.7	6.1	4.7
Stream1	15.9	11.4	12.0	5.2	14.7	26.4	8.6	12.7	29.9	7.3	5.5	5.0
Stream2	18.0	6.8	45.5	5.1	14.8	27.2	7.7	15.2	9.0	7.2	5.2	5.1
Stream3	15.4	3.6	19.6	9.6	19.2	9.7	7.6	16.5	34.8	11.9	5.5	4.9
Stream4	24.3	9.6	18.8	7.3	15.0	30.5	7.9	9.0	23.1	5.8	5.4	5.2
Stream5	19.8	3.6	46.5	6.7	11.9	22.1	10.2	16.1	31.0	6.2	5.6	5.1
Stream6	14.0	3.5	20.4	5.6	15.0	15.0	9.2	16.6	27.5	14.3	5.7	5.3
Stream7	31.6	3.8	15.9	6.7	13.3	25.0	6.9	23.4	30.5	7.2	5.0	4.1
Avg	19.8	6.0	25.6	6.6	14.8	22.3	8.3	15.6	26.5	8.6	5.4	5.0
Min	14.0	3.5	12.0	5.1	11.9	9.7	6.9	9.0	9.0	5.8	5.0	4.1
Max	31.6	11.4	46.5	9.6	19.2	30.5	10.2	23.4	34.8	14.3	5.7	5.3

TPC Benchmark H Overview

The TPC BenchmarkTM H (TPC-H) is a Decision Support benchmark. It is a suite of business-oriented ad-hoc queries and concurrent modifications. The queries and the data populating the database have been chosen to have broad industry-wide relevance while maintaining a sufficient degree of ease of implementation. This benchmark illustrates Decision Support systems that:

- Examine large volumes of data
- Execute queries with a high degree of complexity
- Give answers to critical business questions

TPC-H evaluates the performance of various Decision Support systems by the execution of sets of queries against a standard database under controlled conditions. The TPC-H queries:

- Give answers to real-world business questions
- Simulate generated ad-hoc queries
- Are far more complex than most OLTP transactions
- Include a rich breadth of operators and selectivity constraints
- Generate intensive activity on the part of the database server component of the system under test
- Are executed against a database complying to specific population and scaling requirements
- Are implemented with constraints derived from staying closely synchronized with an on-line production database

Contents

INFO SIZING

TPC TRANSACTION PROCESSING PERFORMANCE COUNCIL CERTIFIED AUDITOR

Benchmark Sponsor: Bob Goldsand
VMware Inc.
3401 Hillview Avenue
Palo Alto, CA 94304

April 7, 2010

I verified the TPC Benchmark™ H performance of the following configuration:

Platform: **HP ProlLiant DL 380 G6 (40-node cluster)**

Database Manager: **Paraccel PADB Analytic Database V250**

Operating System: **VMware ESXi 4.0 / Paraccel Standard Linux**

The results were:

CPU (Speed)	Memory	Disks	**QphH@30000GB**
40 nodes, each with: HP ProLiant DL785 G6 (each node with)			
2 x Intel Xeon 5560 Quad Core (2.8 GHz)	72 GB Main	8 x 300 GB SAS 10K RPM int.	**1,316,882.1**

In my opinion, this performance result was produced in compliance with the TPC's requirements for the benchmark. The following verification items were given special attention:

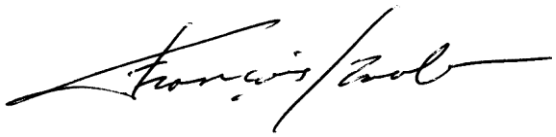
- The database records were defined with the proper layout and size
- The database population was generated using DBGEN
- The database was properly scaled to 1,000GB and populated accordingly
- The compliance of the database auxiliary data structures was verified

- The database load time was correctly measured and reported
- The required ACID properties were verified and met
- The query input variables were generated by QGEN
- The query text was produced using minor modifications and no query variant
- The execution of the queries against the SF1 database produced compliant answers
- A compliant implementation specific layer was used to drive the tests
- The throughput tests involved 7 query streams
- The ratio between the longest and the shortest query was such that no query timing was adjusted
- The execution times for queries and refresh functions were correctly measured and reported
- The repeatability of the measured results was verified
- The system pricing was verified for major components and maintenance
- The major pages from the FDR were verified for accuracy

Additional Audit Notes:

None.

Respectfully Yours,



François Raab
President

1. General Items

1.1. Benchmark Sponsor

A statement identifying the benchmark sponsor(s) and other participating companies must be provided.

VMware, Inc. is the sponsor of this TPC-H benchmark.

1.2. Parameter Settings

Settings must be provided for all customer-tunable parameters and options that have been changed from the defaults found in actual products, including but not limited to:

- *Database Tuning Options*
- *Optimizer/Query execution options*
- *Query processing tool/language configuration parameters*
- *Recovery/commit options*
- *Consistency/locking options*
- *Operating system and configuration parameters*
- *Configuration parameters and options for any other software component incorporated into the pricing structure*
- *Compiler optimization options*

Appendix A contains the ESXi, Paraccel Standard Linux and ParAccel Analytic Database parameters used in this benchmark.

1.3. Configuration Settings

Provide diagrams of both the measured and priced configurations, accompanied by a description of the differences.

The priced configuration is as follows:

40 HP DL 380 G6 Servers (each with):

2 X 2.8 GHz Intel Xeon x5560 Processors (4 cores, 4 threads per CPU)

72 GB Memory

8 X 300 GB Internal disks (1GB = 1,000,000,000 bytes)

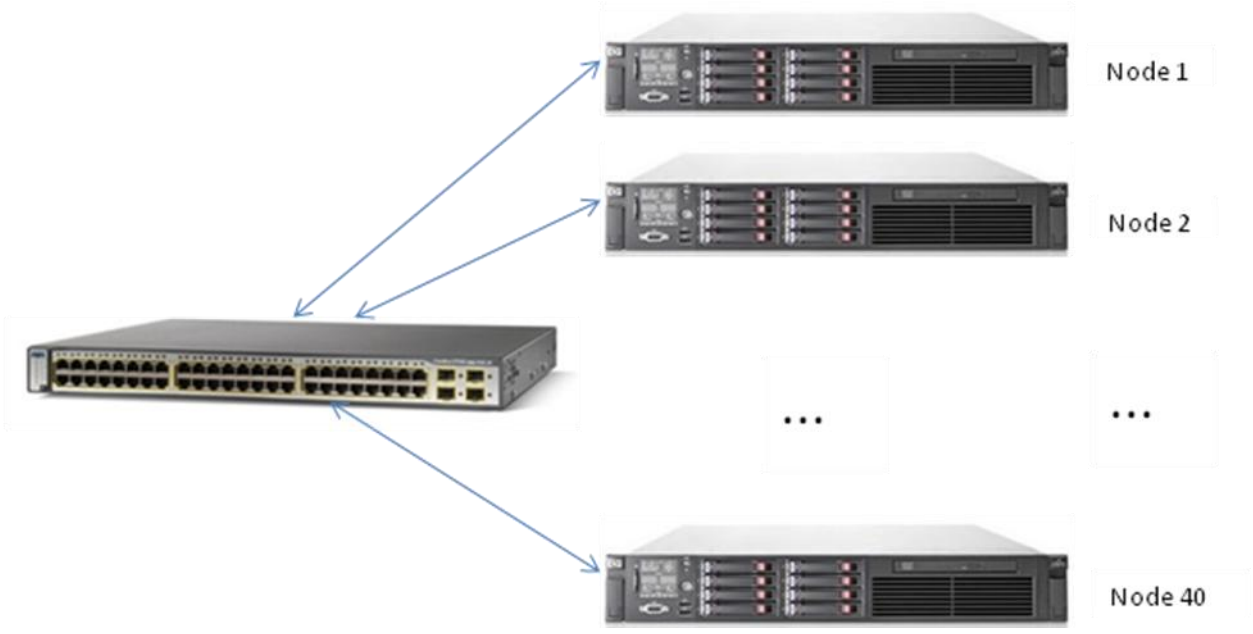
Plus:

4 Cisco 3750 48 port switches

5 APC Rack PDUs

4 x 1 Gb ethernet connections per node

System Component Configuration



The measured configuration is the same as the priced configuration.

VMware ESXi Configuration

VMware Software: Using VMware ESXi™ 4.0 as the hypervisor, an 80 node ParAccel cluster was created using the VMware vSphere Client interface. The ParAccel nodes were consolidated 2 per ESXi 4.0 server and distributed evenly across 40 physical servers. Hyperthreading was disabled at the BIOS level.

ESXi Server with Leader Node and Compute Node

Name	Runtime Name	LUN	Capacity	Identification[1]	Type
HP Serial Attached SCSI Disk	vmhba0:C0:TO:LO	0	279.37 GB	162_datastore0	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L1	1	279.37 GB	162_datastore1	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L2	2	279.37 GB	162_datastore2	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L3	3	279.37 GB	162_datastore3	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L4	4	279.37 GB	162_datastore4	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L5	5	279.37 GB	162_datastore5	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L6	6	279.37 GB	162_datastore6	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L7	7	279.37 GB	162_datastore7	vmfs

ESXi Server with Compute Nodes only

Name	Runtime Name	LUN	Capacity	Identification[2]	Type
HP Serial Attached SCSI Disk	vmhba0:C0:TO:LO	0	279.37 GB	163_datastore0	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L1	1	279.37 GB	163_datastore1	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L2	2	279.37 GB	163_datastore2	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L3	3	279.37 GB	163_datastore3	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L4	4	279.37 GB	163_datastore4	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L5	5	279.37 GB	163_datastore5	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L6	6	279.37 GB	163_datastore6	vmfs
HP Serial Attached SCSI Disk	vmhba0:C0:TO:L7	7	279.37 GB	163_datastore7	vmfs

Networking : Each physical server contained a quad port GigE interface. 4 virtual switches were created, one assigned to each port. The 4 GigE interfaces, vmnic0 thru vmnic3, and the 4 virtual switches, vSwitch1 thru vSwitch4, were used to support the ParAccel private grid network. vSwitch0 was used to support the ParAccel leader node which communicates with the corporate LAN. There is only one leader node in a ParAccel cluster, therefore it is only necessary to utilized vSwitch0 on a single ESXi 4.0 Server where the leader node is resident.

[1] Datastores are named uniquely by using the last octet of the ESXi server IP address; in the above example the last octet of the IP address is 162

[2] Datastores are named uniquely by using the last octet of the ESXi server IP address; in the above example the last octet of the IP address is 163

ESX Server with Leader Node and Compute Node

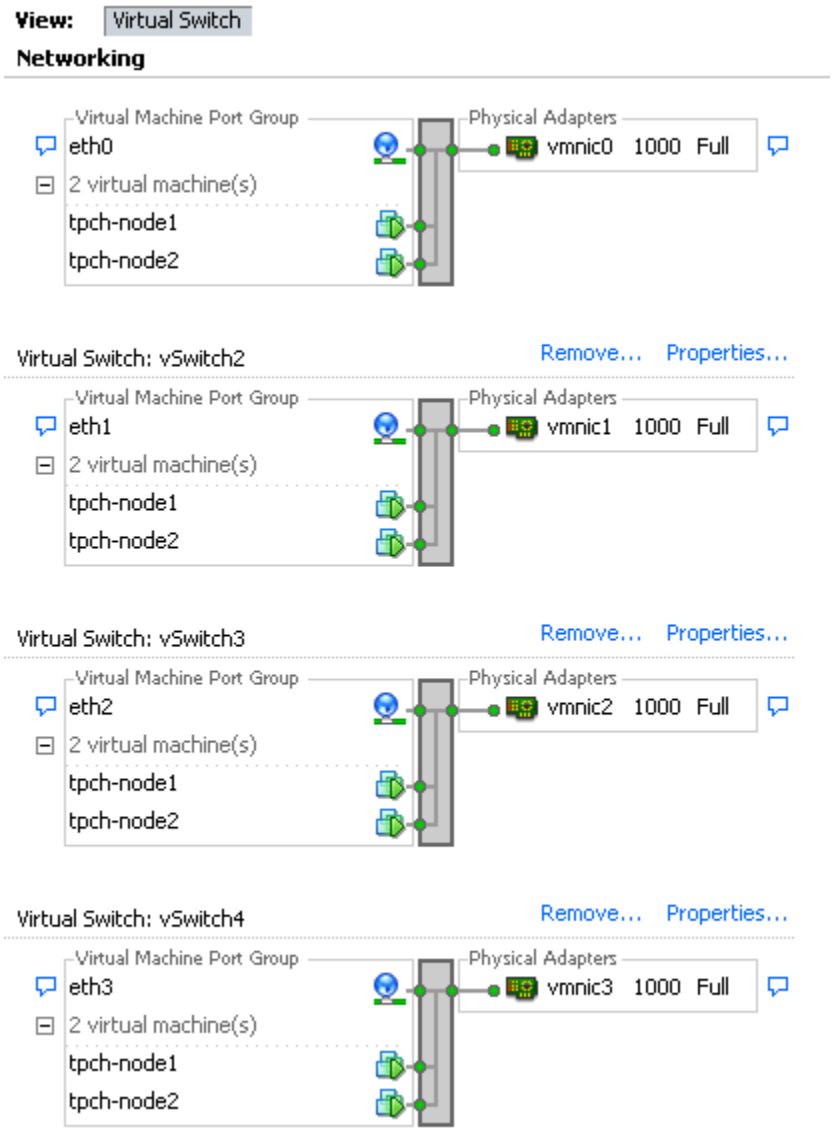
Virtual Switch	Network Label	Network Adapter	Speed	Virtual [3]Machines	MTU	Driver
vSwitch1	eth0	vmnic0	1000 Full	tpch-node1 & tpch-node2	9000	bnx2
vSwitch2	eth1	vmnic1	1000 Full	tpch-node1 & Tpch-node2	9000	bnx2
vSwitch3	eth2	vmnic2	1000 Full	tpch-node1 & tpch-node2	9000	bnx2
vSwitch4	eth3	vmnic3	1000 Full	tpch-node1 & tpch-node2	9000	bnx2
vSwitch0	VMNetwork	vmnic4	1000 Full	tpch-node1[4]	9000	bnx2

ESX Server with Compute Nodes only

Virtual Switch	Network Label	Network Adapter	Speed	Virtual [5]Machines	MTU	Driver
vSwitch1	eth0	vmnic0	1000 Full	tpch-node3 & tpch-node4	9000	bnx2
vSwitch2	eth1	vmnic1	1000 Full	tpch-node3 & tpch-node4	9000	bnx2
vSwitch3	eth2	vmnic2	1000 Full	tpch-node3 & tTpch-node4	9000	bnx2
vSwitch4	eth3	vmnic3	1000 Full	tpch-node3 & tpch-node4	9000	bnx2

[3] The naming convention for the virtual machines corresponds to the total number of nodes in the cluster (80). The node named tpch-node1 was reserved for the leader node.
[4] This interface is only necessary for the ParAccel leader node; no compute nodes are connect to this virtual switch
[5] The naming convention for the virtual machines corresponds to the total number of nodes in the cluster (80). The node named tpch-node1 was reserved for the leader node.

Networking Configuration



Virtual Machines:

Leader Node (One per ParAccel Cluster)

- Guest OS: Redhat Enterprise Linux 5, 64-bit[6]
- VM Version: 7
- CPU: 4 vCPU
- Memory: 32768 MB
- Memory Overhead: 535.31 MB
- Virtual Disks:

[6] The actual OS used by ParAccel is ParAccel Standard Linux , 64-bit. When creating a virtual machine select the OS which most closely matches the deployed OS.

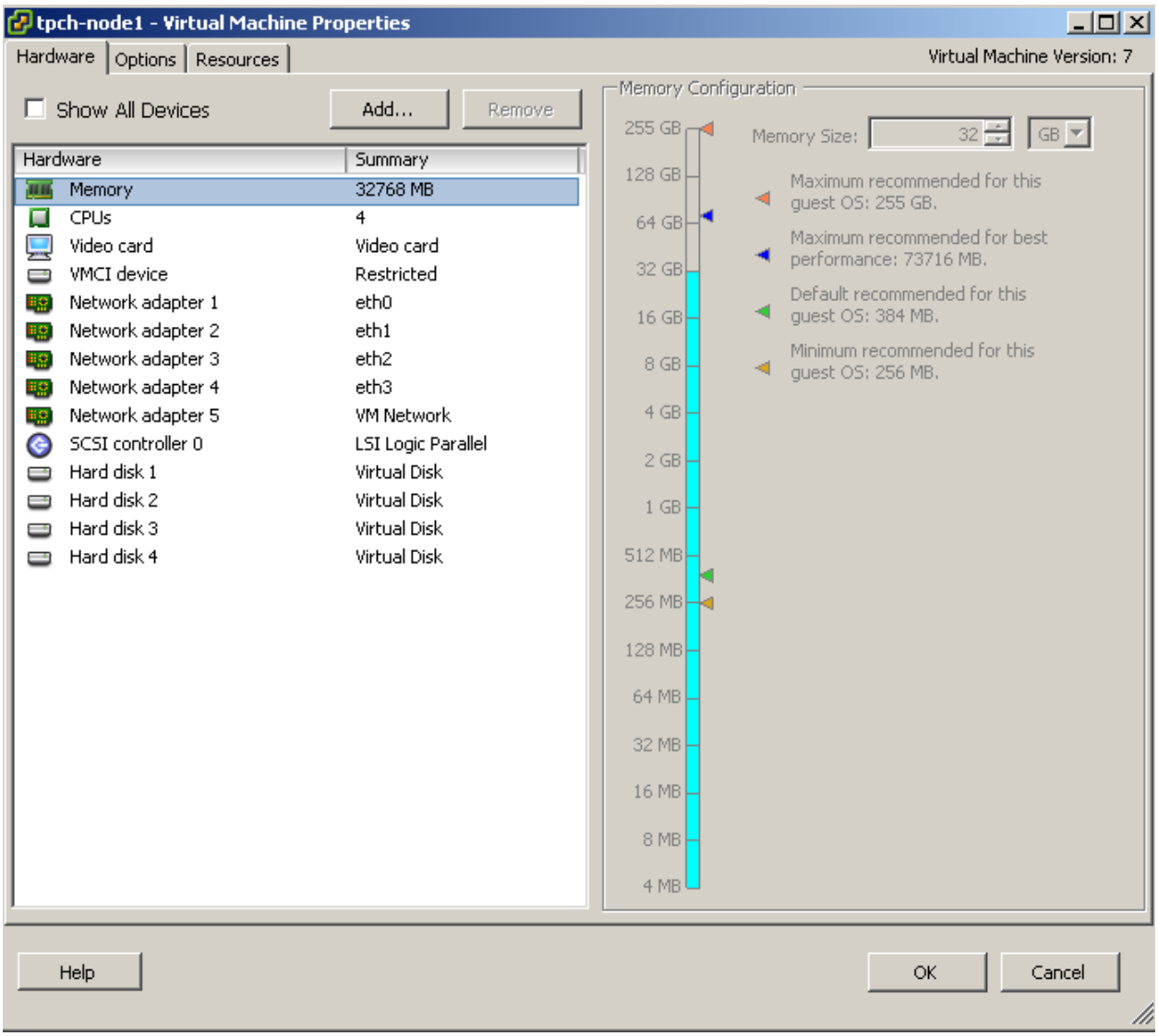
    - Hard Disk 1(Independent Persistent) :
        - 162_datastore0
    - Hard Disk 2 (Independent Persistent):
        - 162_datastore1
    - Hard Disk 3 (Independent Persistent):
        - 162_datastore2
    - Hard Disk 4 (Independent Persistent):
        - 162_datastore3
- Network:
    - VMNetwork (Leader Node to external LAN)
    - Eth0 (ParAccel Private Network)
    - Eth1 (ParAccel Private Network)
    - Eth2 (ParAccel Private Network)
    - Eth3 (ParAccel Private Network)
- Other Devices:
    - Video Card: 800x600
    - SCSI Controller: LSI Logic Parallel

Compute Node (79 ParAccel Compute Nodes in TPC-H Cluster)

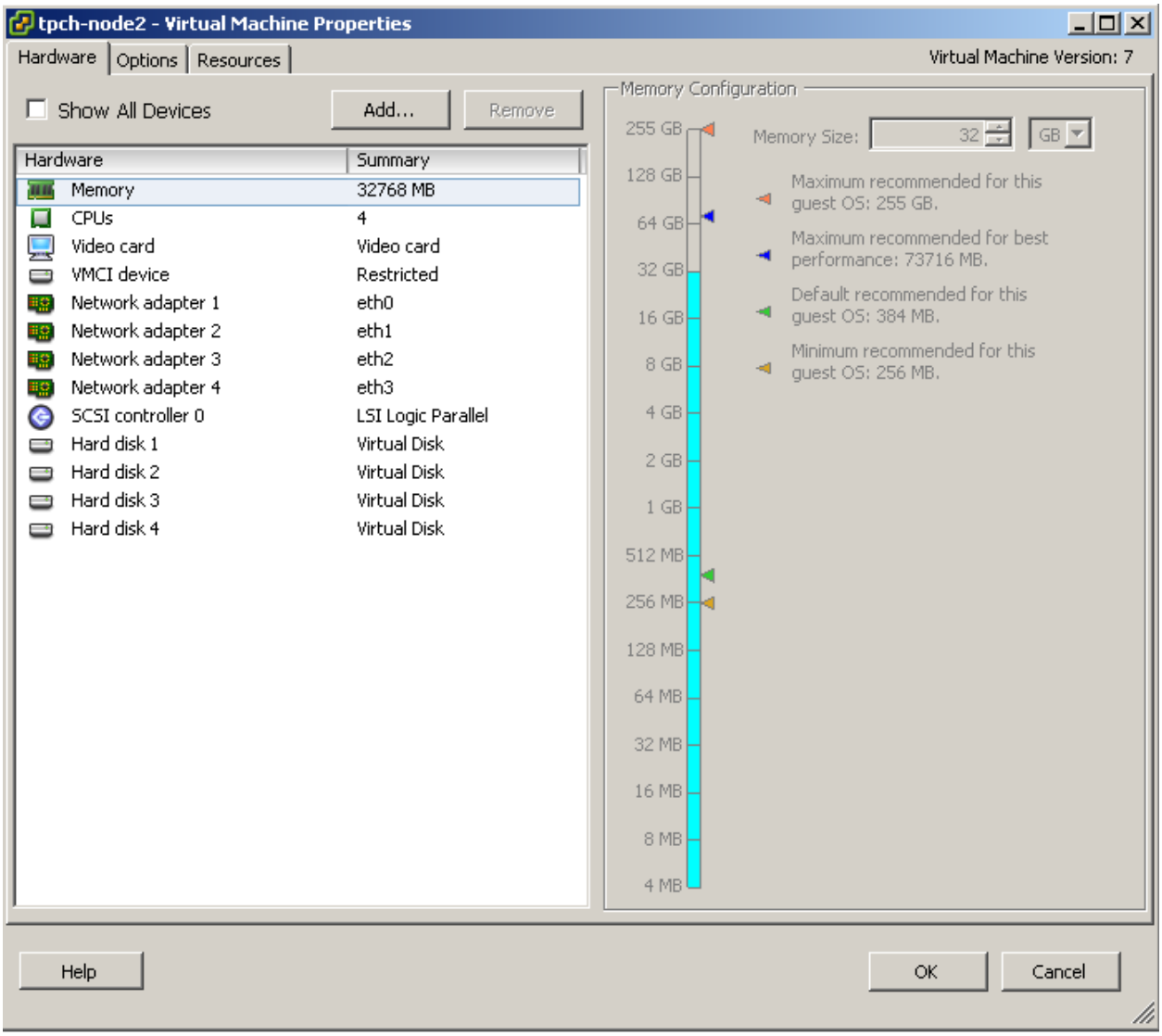
- Guest OS: Redhat Enterprise Linux 5, 64-bit[7]
- VM Version: 7
- CPU: 4 vCPU
- Memory: 32768 MB
- Memory Overhead: 535.31 MB
- Virtual Disks:
    - Hard Disk 1(Independent Persistent) :
        - 162_datastore4
    - Hard Disk 2 (Independent Persistent):
        - 162_datastore5
    - Hard Disk 3 (Independent Persistent):
        - 162_datastore6
    - Hard Disk 4 (Independent Persistent):
        - 162_datastore7
- Network:
    - Eth0 (ParAccel Private Network)
    - Eth1 (ParAccel Private Network)
    - Eth2 (ParAccel Private Network)
    - Eth3 (ParAccel Private Network)
- Other Devices:
    - Video Card: 800x600
    - SCSI Controller: LSI Logic Parallel

[7] The actual OS used by ParAccel is ParAccel Standard Linux , 64-bit. When creating a virtual machine select the OS which most closely matches the deployed OS.

Leader Node Virtual Machine



Compute Node Virtual Machine



2. Clause 1 Logical Database Design

2.1. Database Definition Statements

Listings must be provided for all table definition statements and all other statements used to set up the test and qualification databases.

Appendix B contains the programs and scripts that create and analyze the tables for the TPC-H database.

2.2. Physical Organization

The physical organization of tables and indices within the test and qualification databases must be disclosed. If the column ordering of any table is different from that specified in Clause 1.4, it must be noted.

No record clustering or index clustering was used.

2.3. Horizontal Partitioning

Horizontal partitioning of tables and rows in the test and qualification databases (see Clause 1.5.4) must be disclosed.

Data is horizontally distributed across all nodes based on hashing the first column for each table. No additional data partitioning was used.

2.4. Replication

Any replication of physical objects must be disclosed and must conform to the requirements of Clause 1.5.6.

No form of data replication was used.

3. Clause 2 Queries and Refresh Functions

3.1. Query Language

The query language used to implement the queries must be identified.

SQL was the query language used to implement all queries.

3.2. Verifying Method for Random Number Generation

The method of verification for the random number generation must be described unless the supplied DBGEN and QGEN were used.

TPC supplied versions 2.9.0 of DBGEN and QGEN were used for this TPC-H benchmark.

3.3. Generating Values for Substitution Parameters

The method used to generate values for substitution parameters must be disclosed. If QGEN is not used for this purpose, then the source code of any non-commercial tool used must be disclosed. If QGEN is used, the version number, release number, modification number, and patch level of QGEN must be disclosed.

The supplied QGEN version 2.9.0 was used to generate the substitution parameters.

3.4. Query Text and Output Data from Qualification Database

The executable query text used for query validation must be disclosed along with the corresponding output data generated during the execution of the query text against the qualification database. If minor modifications (see Clause 2.2.3) have been applied to any functional query definitions or approved variants in order to obtain executable query text, these modifications must be disclosed and justified. The justification for a particular minor query modification can apply collectively to all queries for which it has been used. The output data for the power and throughput tests must be made available electronically upon request.

Appendix C contains the query text and query output. The standard queries were used throughout with the following modifications:

- In Q1 the select list expression avg(l_discount) was modified to avg(cast(l_discount as numeric(14,4)) in order to provide the required accuracy.
- In Q2, Q3, Q10, Q18 and Q21, the "limit" function is used to restrict the number of output rows.
- The semicolon (;) is used as a command delimiter.

3.5. Query Substitution Parameters and Seeds Used

The query substitution parameters used for all performance tests must be disclosed in tabular format, along with the seeds used to generate these parameters.

Appendix D contains the seed and query substitution parameters.

3.6. Query Isolation Level

The isolation level used to run the queries must be disclosed. If the isolation level does not map closely to the levels defined in Clause 3.4, additional descriptive detail must be provided.

The queries and transactions were run with isolation level 3.

3.7. Source Code of Refresh Functions

The details of how the refresh functions were implemented must be disclosed (including source code of any non-commercial program used).

Appendix B contains the scripts for the refresh functions.

4. Clause 3 Database System Properties

4.1. ACID Properties

The ACID (Atomicity, Consistency, Isolation and Durability) properties of transaction processing systems must be supported by the system under test during the timed portion of this benchmark. Since TPC-H is not a transaction processing benchmark, the ACID properties must be evaluated outside the timed portion of the test.

Scripts for the ACID tests is included in Appendix B. A three node system was used for the ACID test.

4.2. Atomicity

The system under test must guarantee that transactions are atomic; the system will either perform all individual operations on the data, or will assure that no partially-completed operations leave any effects on the data.

4.2.1 Completed Transaction

Perform the ACID Transaction for a randomly selected set of input data and verify that the appropriate rows have been changed in the ORDERS, LINEITEM, and HISTORY tables

1. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.
2. The ACID Transaction was performed using the order key from step 1.
3. The ACID Transaction committed.
4. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for the same order key. It was verified that the appropriate rows had been changed.

4.2.2 Aborted Transaction

Perform the ACID Transaction for a randomly selected set of input data, substituting a ROLLBACK of the transaction for the COMMIT of the transaction. Verify that the appropriate rows have not been changed in the ORDERS, LINEITEM, and HISTORY tables.

1. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for a randomly selected order key.
2. The ACID Transaction was performed using the order key from step 1. The transaction was stopped prior to the commit.
3. The ACID Transaction was ROLLED BACK.
4. The total price from the ORDERS table and the extended price from the LINEITEM table were retrieved for the same order key. It was verified that the appropriate rows had not been changed.

4.3. Consistency

Consistency is the property of the application that requires any execution of transactions to take the database from one consistent state to another.

4.3.1 Consistency Test

Verify that ORDERS and LINEITEM tables are initially consistent, submit the prescribed number of ACID Transactions with randomly selected input parameters, and re-verify the consistency of the ORDERS and LINEITEM.

1. The consistency of the ORDERS and LINEITEM tables was verified based on a sample of order keys.
2. 200 ACID Transactions were submitted by each of eleven execution streams.
3. The consistency of the ORDERS and LINEITEM tables was re-verified.

4.4. Isolation

Operations of concurrent transactions must yield results which are indistinguishable from the results which would be obtained by forcing each transaction to be serially executed to completion in the proper order.

4.4.1 Read-Write Conflict with Commit

Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the read-write transaction is committed.

1. An ACID Transaction was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID Transaction was suspended prior to COMMIT.
2. An ACID Query was started for the same O_KEY used in step 1.
3. The ACID Transaction was resumed and COMMITTED.
4. The ACID Query completed. It returned the data committed previous to the ACID Transaction.

4.4.2 Read-Write Conflict with Rollback

Demonstrate isolation for the read-write conflict of a read-write transaction and a read-only transaction when the read-write transaction is rolled back.

1. An ACID Transaction was started for a randomly selected O_KEY, L_KEY, and DELTA. The ACID Transaction was suspended prior to ROLLBACK.
2. An ACID Query was started for the same O_KEY used in step 1. The ACID Query did not see the uncommitted changes made by the ACID Transaction.
3. The ACID Transaction was ROLLED BACK.
4. The ACID Query completed.

4.4.3 Write-Write Conflict with Commit

Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is committed.

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. T1 was suspended prior to COMMIT.
2. Another ACID Transaction, T2, was started using the same O_KEY and L_KEY and a randomly selected DELTA.
3. T2 waited.
4. T1 was allowed to COMMIT and T2 completed.
5. It was verified that T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE + (DELTA1*(T1.L_EXTENDEDPRICE/T1.L_QUANTITY))

4.4.4 Write-Write Conflict with Rollback

Demonstrate isolation for the write-write conflict of two update transactions when the first transaction is rolled back.

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. T1 was suspended prior to COMMIT.
2. Another ACID Transaction, T2, was started using the same O_KEY and L_KEY and a randomly selected DELTA.
3. T2 waited.
4. T1 was allowed to ROLLBACK and T2 completed.
5. It was verified that T2.L_EXTENDEDPRICE = T1.L_EXTENDEDPRICE.

4.4.5 Concurrent Progress of Read and Write Transactions

Demonstrate the ability of read and write transactions affecting different database tables to make progress concurrently.

1. An ACID Transaction, T1, was started for a randomly selected O_KEY, L_KEY, and DELTA. T1 was suspended prior to ROLLBACK.
2. Another Transaction, T2, was started which did the following:

   For random values of PS_PARTKEY and PS_SUPPKEY, all columns of the PARTSUPP table for which PS_PARTKEY and PS_SUPPKEY are equal, are returned.
3. T2 completed.
4. T1 was allowed to COMMIT.
5. It was verified that appropriate rows in ORDERS, LINEITEM and HISTORY tables were changed.

4.4.6 Read-Only Query Conflict with Update Transaction

Demonstrate that the continuous submission of arbitrary (read-only) queries against one or more tables of the database does not indefinitely delay update transactions affecting those tables from making progress.

1. A Transaction, T1, executing iso6-Query against the qualification database, was started using a randomly selected O_KEY.
2. An ACID Transaction T2, was started for a randomly selected O_KEY, L_KEY and DELTA.
3. T2 completed and appropriate rows in the ORDERS, LINEITEM and HISTORY tables had been changed.
4. Transaction T1 completed executing iso6-Query.

4.5. Durability

The SUT must guarantee durability: the ability to preserve the effects of committed transactions and insure database consistency after recovery from any one of the failures listed in Clause 3.5.3

The durability tests were executed on a subset of the SUT comprising 3 nodes (1 leader + 2 compute nodes).

4.5.1 Failure of a Durable Medium

Guarantee the database and committed updates are preserved across a permanent irrecoverable failure of any single durable medium containing TPC-H database tables or recovery log tables.

All disks containing TPC-H tables are housed on mirrored (RAID1) volumes. After 8 streams had processed at least 100 transactions, a permanent irrecoverable failure of a single durable medium was simulated by removing one side of a mirror from one of the DL380 servers. The system continued to process transactions without any noticeable impact, as is to be expected when only half a mirror is no longer functioning. All transactions from the success file were verified to be present in the History table. The consistency of the database was verified.

4.5.2 System Crash

Guarantee the database and committed updates are preserved across an instantaneous interruption (system crash/system hang) in processing which requires the system to reboot to recover.

Power was cut off to all of the DL380 Servers in the cluster (i.e., leader node + compute nodes) to simulate a system crash. When power was restored all nodes automatically rebooted. The database was manually restarted. All transactions from the success file were verified to be present in the History table. The consistency of the database was verified.

3.5.3 Memory Failure

Guarantee the database and committed updates are preserved across failure of all or part of memory (loss of contents).

The failure was tested as described in 4.5.2.

4. Clause 4 Scaling and Database Population

4.1. Ending Cardinality of Tables

The cardinality (i.e., the number of rows) of each table of the test database, as it existed at the completion of the database load (see clause 4.2.5) must be disclosed.

Table	Rows
lineitem	5999989709
orders	1500000000
partsupp	800000000
part	200000000
customer	150000000
supplier	10000000
nation	25
region	5

4.2. Distribution of Tables and Logs Across Media

The distribution of tables and logs across all media must be explicitly described.

All Base Tables, DBMS Temporary Space and OS/DBMS software was stored using Storage Redundancy Level One (RAID1) in four volumes per DL380 server. Each server houses two virtual machines. All sizes below are post-RAID1 sizes on the compute virtual nodes, as per the "df -h" command, performed within a single virtual machine. Therefore, the physical space used by each partition is twice the figure listed below due to mirroring, and twice again because there are two virtual machines per server.

Partition	Use	Size
/dev/md0	Ext3 file system – landing zone for DBGEN	182.9GB
/dev/md20	ParAccel Table Space (LUN0)	46.5GB
/dev/md21	ParAccel Table Space (LUN1)	46.5GB
/dev/sda3	/boot	94.6MB
tmpfs	Swap	16.5GB

The file system on the leader virtual machine is organized as follows:

Partition	Use	Size
/dev/md0	Ext3 file system – general file system	182.9GB
/dev/sda3	/boot	94.6MB
tmpfs	Swap	16.5GB
/dev/ram0	in-memory file system	122MB

5.3. Database partition/replication mapping

The mapping of database partitions/replications must be explicitly described.

The rows of each table are horizontally distributed across all nodes based on hashing the first field in each table. No additional data partitioning was used. In addition, no form of data replication was used.

5.4. RAID Feature

Implementations may use some form of RAID to ensure high availability. If used for data, auxiliary storage (e.g. indexes) or temporary space, the level of RAID must be disclosed for each device.

RAID 1 was used for all base tables and auxiliary data structures.

5.5. Modifications to the DBGEN

Any modifications to the DBGEN (see Clause 4.2.1) source code must be disclosed. In the event that a program other than DBGEN was used to populate the database, it must be disclosed in its entirety.

The supplied DBGEN version 2.9.0 was used to generate the database population for this benchmark.

5.6. Database Load Time

The database load time for the test database (see clause 4.3) must be disclosed.

The database load time was = 16Min 23Sec

5.7. Data Storage Ratio

The data storage ratio must be disclosed. It is computed as the ratio between the total amount of priced disk space, and the chosen test database size as defined in Clause 4.1.3.

The data storage ratio is computed from the following information:

# Disks	GB per Disk (1GB=1000^3)	Total Disk Space in GB (1GB=1024^3)
320	300	89406
	Total Space	89406
	Data Storage Ratio	89.4:1

5.8. Database Load Mechanism Details and Illustration

The details of the database load must be described, including a block diagram illustrating the overall process.

The test database was loaded using flat files. All load scripts are included in Appendix B.

Qualification Database Configuration

Any differences between the configuration of the qualification database and the test database must be disclosed.

The qualification database used identical scripts to create and load the data with only the necessary adjustments for size differences.

6. Clause 5 Performance Metrics and Execution Rules

6.1. System Activity Between Load and Performance Tests

Any system activity on the SUT that takes place between the conclusion of the load test and the beginning of the performance test must be fully disclosed.

1. The DBMS on the SUT was restarted between the conclusion of the load test and the beginning of the performance test
2. No other activity on the SUT occurred between the conclusion of the load test and the beginning of the performance test

All scripts and queries used are included in Appendix B

6.2. Steps in the Power Test

The details of the steps followed to implement the power test (.e.g., system boot, database restart, etc.) must be disclosed.

The following steps were used to implement the power test:

1. RF1 Refresh Transaction in update stream
2. Stream 00 Execution in query stream
3. RF2 Refresh Transaction in update stream

6.3. Timing Intervals for Each Query and Refresh Functions

The timing intervals for each query and for both refresh functions must be reported for the power test.

The timing intervals for each query and both update functions are reported on the page titled “Numerical Quantities”,

contained in the beginning of this document and replicated in the Executive Summary document.

6.4. Number of Streams for the Throughput Test

The number of execution streams used for the throughput test must be disclosed.

7 query streams and one refresh stream were used for the throughput test.

6.5. Start and End Date/Times for Each Query Stream

The start time and finish time for each query stream must be reported for the throughput test.

The start times and finish times for each query stream in the throughput test are reported on the page titled "Numerical Quantities", contained in the beginning of this document and replicated in the Executive Summary document.

6.6. Total Elapsed Time of the Measurement Interval

The total elapsed time of the measurement interval must be reported for the throughput test.

The total elapsed time of the throughput test is reported on the page titled "Numerical Quantities", contained in the beginning of this document and replicated in the Executive Summary document.

6.7. Refresh Function Start Date/Time and Finish Date/Time

Start and finish time for each refresh function in the refresh stream must be reported for the throughput test.

The start and finish times for each refresh function:

Stream ID	Refresh Function	Start Date	Start Time	End Date	End Time
Stream01	RF1	3/22/2010	22:06:32	3/22/2010	22:06:37
Stream01	RF2	3/22/2010	22:06:38	3/22/2010	22:06:43
Stream02	RF1	3/22/2010	22:06:43	3/22/2010	22:06:48
Stream02	RF2	3/22/2010	22:06:48	3/22/2010	22:06:53
Stream03	RF1	3/22/2010	22:06:53	3/22/2010	22:06:58
Stream03	RF2	3/22/2010	22:06:59	3/22/2010	22:07:03
Stream04	RF1	3/22/2010	22:07:04	3/22/2010	22:07:09
Stream04	RF2	3/22/2010	22:07:09	3/22/2010	22:07:14
Stream05	RF1	3/22/2010	22:07:14	3/22/2010	22:07:19
Stream05	RF2	3/22/2010	22:07:20	3/22/2010	22:07:25
Stream06	RF1	3/22/2010	22:07:25	3/22/2010	22:07:30
Stream06	RF2	3/22/2010	22:07:31	3/22/2010	22:07:36
Stream07	RF1	3/22/2010	22:07:36	3/22/2010	22:07:41
Stream07	RF2	3/22/2010	22:07:41	3/22/2010	22:07:45

6.8. Timing Intervals for Each Query and Each Refresh Function for Each Stream

The timing intervals for each query of each stream and each refresh function must be reported for the throughput test.

The timing intervals for each query and each refresh function for the throughput test are reported on the page titled "Numerical Quantities", contained in the beginning of this document and replicated in the Executive Summary document.

6.9. Performance Metrics

The computed performance metric, related numerical quantities and price performance metric must be reported.

The performance metrics, and the numbers on which they are based, are reported on the page titled "Numerical Quantities", contained in the beginning of this document and replicated in the Executive Summary document.

6.10. The Performance Metric and Numerical Quantities from Both Runs

The performance metric and numerical quantities from both runs must be disclosed.

Performance results from the first two executions of the TPC-H benchmark indicated the following percent difference for the three metrics:

Run ID	Qpp	Qth	Qph
Run 1	1,260,595.1	1,375,682.4	1,316,882.1
Run 2	1,381,240.5	1,400,000.0	1,390,588.6
% Difference	9.6%	1.8%	5.6%

6.11. System Activity Between Performance Tests

Any activity on the SUT that takes place between the conclusion of Run1 and the beginning of Run2 must be disclosed.

No activity occurred between Run 1 and Run 2.

7. Clause 6 SUT and Driver Implementation

7.1. Driver

A detailed description of how the driver performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the driver.

The entire test is run by executing the do1tb.sh shell script. The text of do1tb.sh is reproduced in Appendix B.

7.2. Implementation-Specific Layer

If an implementation-specific layer is used, then a detailed description of how it performs its functions must be supplied, including any related source code or scripts. This description should allow an independent reconstruction of the implementation-specific layer.

All database configuration is disclosed in Appendix A.

The performance tests are performed using psql. It is a ParAccel-provided utility that allows SQL statements to be executed against the ParAccel Analytic Database™. The psql utility is invoked from the command-line on the SUT. It reads

input from files containing SQL statements and sends results to stdout. The performance test scripts utilizing psql can be found in Appendix B.

The ACID tests are performed using psql. All the ACID test scripts are reproduced in Appendix B.

7.3. Profile-Directed Optimization

If profile-directed optimization as described in Clause 5.2.9 is used, such use must be disclosed.

Profile-directed optimization was not used.

8. Clause 7 Pricing

8.1. Hardware and Software Used

A detailed list of hardware and software used in the priced system must be reported. Each item must have vendor part number, description, and release/revision level, and either general availability status or committed delivery date. If package-pricing is used, contents of the package must be disclosed. Pricing source(s) and effective date(s) of price(s) must also be reported.

Refer to the Executive Summary for the pricing spreadsheet and Appendix E for the actual price quotes used to create the spreadsheet.

8.2. Total Three Year Price

The total 3-year price of the entire configuration must be reported, including hardware, software, and maintenance charges. Separate component pricing is recommended. The basis of all discounts used must be disclosed.

The total 3-year price of the configuration is $915,511.44. For details of pricing, see the second page of the Executive Summary.

Discounts were taken from actual price quotes, available to any buyer with like conditions, provided by CDW, VMware, Inc. and ParAccel, Inc. The respective price quotes are included in Appendix E of this document.

8.3. Availability Date

The committed delivery date for general availability of products used in the price calculations must be reported. When the priced system includes products with different availability dates, the reported availability date for the priced system must be the date at which all components are committed to be available.

All hardware and software components used in the measured configuration will be generally available as of June 30, 2010.

9. Auditor's Information and Attestation Letter

The auditor's agency name, address, phone number, and Attestation letter with a brief audit summary report indicating compliance must be included in the full disclosure report. A statement should be included specifying who to contact in order to obtain further information regarding the audit process.

The auditor's attestation letter follows the table of contents.

Appendix A: Configuration Settings

VMWare Configuration Settings

```
========================
ESXi Configuration File (esx.conf)
========================
/adv/Misc/HostIPAddr = "192.168.6.162"
/adv/Misc/HostName = "tpch-
esxi237.paraccel.com"
/adv/Misc/HostdStatsVMCount = "512"
/adv/Net/ManagementAddr = "192.168.6.162"
/adv/Net/ManagementIface = "vmk0"
/adv/UserMem/UserMemASRandomSeed =
"413398228"
/advUserOptions/options[0000]/default =
"1"
/advUserOptions/options[0000]/description
= "Enable or Disable the CIM service"
/advUserOptions/options[0000]/hidden =
"false"
/advUserOptions/options[0000]/max = "1"
/advUserOptions/options[0000]/min = "0"
/advUserOptions/options[0000]/name =
"CIMEnabled"
/advUserOptions/options[0000]/type =
"int"
/advUserOptions/options[0001]/default =
"1"
/advUserOptions/options[0001]/description
= "Enable or Disable the CIM OEM
Providers"
/advUserOptions/options[0001]/hidden =
"false"
/advUserOptions/options[0001]/max = "1"
/advUserOptions/options[0001]/min = "0"
/advUserOptions/options[0001]/name =
"CIMOEMProvidersEnabled"
/advUserOptions/options[0001]/type =
"int"
/advUserOptions/options[0002]/default =
"1"
/advUserOptions/options[0002]/description
= "Enable or Disable the CIM Custom
Providers"
/advUserOptions/options[0002]/hidden =
"false"
/advUserOptions/options[0002]/max = "1"
/advUserOptions/options[0002]/min = "0"
/advUserOptions/options[0002]/name =
"CIMCustomProvidersEnabled"
/advUserOptions/options[0002]/type =
"int"
/advUserOptions/options[0003]/default =
"60"
/advUserOptions/options[0003]/description
= "Set the watchdog polling interval for
the CIM service"
/advUserOptions/options[0003]/hidden =
"false"
/advUserOptions/options[0003]/max = "300"
/advUserOptions/options[0003]/min = "0"
/advUserOptions/options[0003]/name =
"CIMWatchdogInterval"
/advUserOptions/options[0003]/type =
"int"
/device/000:29.0/owner = "vmkernel"
/device/000:29.1/owner = "vmkernel"
/device/000:29.2/owner = "vmkernel"
/device/000:29.3/owner = "vmkernel"
/device/000:29.7/owner = "vmkernel"
/device/001:04.4/owner = "vmkernel"
/device/002:00.0/vmkname = "vmnic0"
/device/002:00.1/vmkname = "vmnic1"
/device/003:00.0/vmkname = "vmnic2"
/device/003:00.1/vmkname = "vmnic3"
/device/004:00.0/vmkname = "vmhba0"
/device/007:00.0/vmkname = "vmnic4"
/device/007:00.1/vmkname = "vmnic5"
/net/pnic/child[0000]/mac =
"00:26:55:35:58:d4"
/net/pnic/child[0000]/name = "vmnic0"
/net/pnic/child[0000]/virtualMac =
"00:50:56:55:58:d4"
/net/pnic/child[0001]/mac =
"00:26:55:35:58:d6"
/net/pnic/child[0001]/name = "vmnic1"
/net/pnic/child[0001]/virtualMac =
"00:50:56:55:58:d6"
/net/pnic/child[0002]/mac =
"00:26:55:35:58:d8"
/net/pnic/child[0002]/name = "vmnic2"
/net/pnic/child[0002]/virtualMac =
"00:50:56:55:58:d8"
/net/pnic/child[0003]/mac =
"00:26:55:35:58:da"
/net/pnic/child[0003]/name = "vmnic3"
/net/pnic/child[0003]/virtualMac =
"00:50:56:55:58:da"
/net/pnic/child[0004]/mac =
"d8:d3:85:5e:55:6c"
/net/pnic/child[0004]/name = "vmnic4"
/net/pnic/child[0004]/virtualMac =
"00:50:56:5e:55:6c"
/net/pnic/child[0005]/mac =
"d8:d3:85:5e:55:6e"
/net/pnic/child[0005]/name = "vmnic5"
/net/pnic/child[0005]/virtualMac =
"00:50:56:5e:55:6e"
```

```
/net/routes/kernel/gateway =
"192.168.6.1"
/net/vmkernelnic/child[0000]/dhcp =
"false"
/net/vmkernelnic/child[0000]/dhcpDns =
"false"
/net/vmkernelnic/child[0000]/dhcpv6 =
"false"
/net/vmkernelnic/child[0000]/enable =
"true"
/net/vmkernelnic/child[0000]/ipv4address
= "192.168.6.162"
/net/vmkernelnic/child[0000]/ipv4broadcas
t = "192.168.6.255"
/net/vmkernelnic/child[0000]/ipv4netmask
= "255.255.255.0"
/net/vmkernelnic/child[0000]/mac =
"d8:d3:85:5e:55:6c"
/net/vmkernelnic/child[0000]/macFromPnic
= "vmnic4"
/net/vmkernelnic/child[0000]/name =
"vmk0"
/net/vmkernelnic/child[0000]/portgroup =
"Management Network"
/net/vmkernelnic/child[0000]/routAdv =
"false"
/net/vmkernelnic/child[0000]/tsoMss = "0"
/net/vswitch/child[0000]/beacon/enable =
"false"
/net/vswitch/child[0000]/cdp/status =
"listen"
/net/vswitch/child[0000]/name =
"vSwitch0"
/net/vswitch/child[0000]/portgroup/child[
0000]/name = "Management Network"
/net/vswitch/child[0000]/portgroup/child[
0000]/teamPolicy/hasUplinkOrder = "true"
/net/vswitch/child[0000]/portgroup/child[
0000]/teamPolicy/linkCriteria/beacon =
"ignore"
/net/vswitch/child[0000]/portgroup/child[
0000]/teamPolicy/linkCriteria/fullDuplex
= "ignore"
/net/vswitch/child[0000]/portgroup/child[
0000]/teamPolicy/linkCriteria/minSpeed =
"10"
/net/vswitch/child[0000]/portgroup/child[
0000]/teamPolicy/linkCriteria/pctError =
"ignore"
/net/vswitch/child[0000]/portgroup/child[
0000]/teamPolicy/maxActive = "2"
/net/vswitch/child[0000]/portgroup/child[
0000]/teamPolicy/notifySwitch = "true"
/net/vswitch/child[0000]/portgroup/child[
0000]/teamPolicy/reversePolicy = "true"
/net/vswitch/child[0000]/portgroup/child[
0000]/teamPolicy/rollingRestoration =
"false"
/net/vswitch/child[0000]/portgroup/child[
0000]/teamPolicy/team = "lb_srcid"
/net/vswitch/child[0000]/portgroup/child[
0000]/teamPolicy/uplinks[0000]/pnic =
"vmnic4"
/net/vswitch/child[0000]/portgroup/child[
0001]/name = "VM Network"
/net/vswitch/child[0000]/teamPolicy/hasUp
linkOrder = "true"
/net/vswitch/child[0000]/teamPolicy/linkC
riteria/beacon = "ignore"
/net/vswitch/child[0000]/teamPolicy/linkC
riteria/fullDuplex = "ignore"
/net/vswitch/child[0000]/teamPolicy/linkC
riteria/minSpeed = "10"
/net/vswitch/child[0000]/teamPolicy/linkC
riteria/pctError = "ignore"
/net/vswitch/child[0000]/teamPolicy/maxAc
tive = "1"
/net/vswitch/child[0000]/teamPolicy/notif
ySwitch = "true"
/net/vswitch/child[0000]/teamPolicy/rever
sePolicy = "true"
/net/vswitch/child[0000]/teamPolicy/rolli
ngRestoration = "false"
/net/vswitch/child[0000]/teamPolicy/team
= "lb_srcid"
/net/vswitch/child[0000]/teamPolicy/uplin
ks[0000]/pnic = "vmnic4"
/net/vswitch/child[0000]/uplinks/child[00
00]/pnic = "vmnic4"
/net/vswitch/child[0001]/beacon/enable =
"false"
/net/vswitch/child[0001]/capabilities/Che
cksumOffload = "true"
/net/vswitch/child[0001]/capabilities/Che
cksumOffloadIPv6 = "true"
/net/vswitch/child[0001]/capabilities/Hig
hDMA = "true"
/net/vswitch/child[0001]/capabilities/Sca
tterGatherSpanPagesTx = "true"
/net/vswitch/child[0001]/capabilities/Sca
tterGatherTx = "true"
/net/vswitch/child[0001]/capabilities/Tcp
SegmentationOffload = "true"
/net/vswitch/child[0001]/capabilities/Tcp
SegmentationOffload256k = "true"
/net/vswitch/child[0001]/capabilities/Tcp
SegmentationOffloadIPv6 = "true"
/net/vswitch/child[0001]/capabilities/Vla
nTag = "true"
/net/vswitch/child[0001]/capabilities/Vla
nUntag = "true"
/net/vswitch/child[0001]/cdp/status =
"listen"
/net/vswitch/child[0001]/mtu = "9000"
/net/vswitch/child[0001]/name =
"vSwitch1"
```

```
/net/vswitch/child[0001]/portgroup/child[
0000]/name = "eth0"
/net/vswitch/child[0001]/portgroup/child[
0000]/vlanId = "0"
/net/vswitch/child[0001]/securityPolicy/f
orgedTx = "true"
/net/vswitch/child[0001]/securityPolicy/m
acChange = "true"
/net/vswitch/child[0001]/securityPolicy/p
romiscuous = "false"
/net/vswitch/child[0001]/shapingPolicy/en
abled = "false"
/net/vswitch/child[0001]/teamPolicy/hasUp
linkOrder = "true"
/net/vswitch/child[0001]/teamPolicy/linkC
riteria/beacon = "ignore"
/net/vswitch/child[0001]/teamPolicy/linkC
riteria/fullDuplex = "ignore"
/net/vswitch/child[0001]/teamPolicy/linkC
riteria/minSpeed = "10"
/net/vswitch/child[0001]/teamPolicy/linkC
riteria/pctError = "ignore"
/net/vswitch/child[0001]/teamPolicy/maxAc
tive = "1"
/net/vswitch/child[0001]/teamPolicy/notif
ySwitch = "true"
/net/vswitch/child[0001]/teamPolicy/rever
sePolicy = "true"
/net/vswitch/child[0001]/teamPolicy/rolli
ngRestoration = "false"
/net/vswitch/child[0001]/teamPolicy/team
= "lb_srcid"
/net/vswitch/child[0001]/teamPolicy/uplin
ks[0000]/pnic = "vmnic0"
/net/vswitch/child[0001]/uplinks/child[00
00]/pnic = "vmnic0"
/net/vswitch/child[0002]/beacon/enable =
"false"
/net/vswitch/child[0002]/capabilities/Che
cksumOffload = "true"
/net/vswitch/child[0002]/capabilities/Che
cksumOffloadIPv6 = "true"
/net/vswitch/child[0002]/capabilities/Hig
hDMA = "true"
/net/vswitch/child[0002]/capabilities/Sca
tterGatherSpanPagesTx = "true"
/net/vswitch/child[0002]/capabilities/Sca
tterGatherTx = "true"
/net/vswitch/child[0002]/capabilities/Tcp
SegmentationOffload = "true"
/net/vswitch/child[0002]/capabilities/Tcp
SegmentationOffload256k = "true"
/net/vswitch/child[0002]/capabilities/Tcp
SegmentationOffloadIPv6 = "true"
/net/vswitch/child[0002]/capabilities/Vla
nTag = "true"
/net/vswitch/child[0002]/capabilities/Vla
nUntag = "true"
/net/vswitch/child[0002]/cdp/status =
"listen"
/net/vswitch/child[0002]/mtu = "9000"
/net/vswitch/child[0002]/name =
"vSwitch2"
/net/vswitch/child[0002]/portgroup/child[
0000]/name = "eth1"
/net/vswitch/child[0002]/portgroup/child[
0000]/vlanId = "0"
/net/vswitch/child[0002]/securityPolicy/f
orgedTx = "true"
/net/vswitch/child[0002]/securityPolicy/m
acChange = "true"
/net/vswitch/child[0002]/securityPolicy/p
romiscuous = "false"
/net/vswitch/child[0002]/shapingPolicy/en
abled = "false"
/net/vswitch/child[0002]/teamPolicy/hasUp
linkOrder = "true"
/net/vswitch/child[0002]/teamPolicy/linkC
riteria/beacon = "ignore"
/net/vswitch/child[0002]/teamPolicy/linkC
riteria/fullDuplex = "ignore"
/net/vswitch/child[0002]/teamPolicy/linkC
riteria/minSpeed = "10"
/net/vswitch/child[0002]/teamPolicy/linkC
riteria/pctError = "ignore"
/net/vswitch/child[0002]/teamPolicy/maxAc
tive = "1"
/net/vswitch/child[0002]/teamPolicy/notif
ySwitch = "true"
/net/vswitch/child[0002]/teamPolicy/rever
sePolicy = "true"
/net/vswitch/child[0002]/teamPolicy/rolli
ngRestoration = "false"
/net/vswitch/child[0002]/teamPolicy/team
= "lb_srcid"
/net/vswitch/child[0002]/teamPolicy/uplin
ks[0000]/pnic = "vmnic1"
/net/vswitch/child[0002]/uplinks/child[00
00]/pnic = "vmnic1"
/net/vswitch/child[0003]/beacon/enable =
"false"
/net/vswitch/child[0003]/capabilities/Che
cksumOffload = "true"
/net/vswitch/child[0003]/capabilities/Che
cksumOffloadIPv6 = "true"
/net/vswitch/child[0003]/capabilities/Hig
hDMA = "true"
/net/vswitch/child[0003]/capabilities/Sca
tterGatherSpanPagesTx = "true"
/net/vswitch/child[0003]/capabilities/Sca
tterGatherTx = "true"
/net/vswitch/child[0003]/capabilities/Tcp
SegmentationOffload = "true"
/net/vswitch/child[0003]/capabilities/Tcp
SegmentationOffload256k = "true"
/net/vswitch/child[0003]/capabilities/Tcp
SegmentationOffloadIPv6 = "true"
```

```
/net/vswitch/child[0003]/capabilities/Vla
nTag = "true"
/net/vswitch/child[0003]/capabilities/Vla
nUntag = "true"
/net/vswitch/child[0003]/cdp/status =
"listen"
/net/vswitch/child[0003]/mtu = "9000"
/net/vswitch/child[0003]/name =
"vSwitch3"
/net/vswitch/child[0003]/portgroup/child[
0000]/name = "eth2"
/net/vswitch/child[0003]/portgroup/child[
0000]/vlanId = "0"
/net/vswitch/child[0003]/securityPolicy/f
orgedTx = "true"
/net/vswitch/child[0003]/securityPolicy/m
acChange = "true"
/net/vswitch/child[0003]/securityPolicy/p
romiscuous = "false"
/net/vswitch/child[0003]/shapingPolicy/en
abled = "false"
/net/vswitch/child[0003]/teamPolicy/hasUp
linkOrder = "true"
/net/vswitch/child[0003]/teamPolicy/linkC
riteria/beacon = "ignore"
/net/vswitch/child[0003]/teamPolicy/linkC
riteria/fullDuplex = "ignore"
/net/vswitch/child[0003]/teamPolicy/linkC
riteria/minSpeed = "10"
/net/vswitch/child[0003]/teamPolicy/linkC
riteria/pctError = "ignore"
/net/vswitch/child[0003]/teamPolicy/maxAc
tive = "1"
/net/vswitch/child[0003]/teamPolicy/notif
ySwitch = "true"
/net/vswitch/child[0003]/teamPolicy/rever
sePolicy = "true"
/net/vswitch/child[0003]/teamPolicy/rolli
ngRestoration = "false"
/net/vswitch/child[0003]/teamPolicy/team
= "lb_srcid"
/net/vswitch/child[0003]/teamPolicy/uplin
ks[0000]/pnic = "vmnic2"
/net/vswitch/child[0003]/uplinks/child[00
00]/pnic = "vmnic2"
/net/vswitch/child[0004]/beacon/enable =
"false"
/net/vswitch/child[0004]/capabilities/Che
cksumOffload = "true"
/net/vswitch/child[0004]/capabilities/Che
cksumOffloadIPv6 = "true"
/net/vswitch/child[0004]/capabilities/Hig
hDMA = "true"
/net/vswitch/child[0004]/capabilities/Sca
tterGatherSpanPagesTx = "true"
/net/vswitch/child[0004]/capabilities/Sca
tterGatherTx = "true"
/net/vswitch/child[0004]/capabilities/Tcp
SegmentationOffload = "true"
/net/vswitch/child[0004]/capabilities/Tcp
SegmentationOffload256k = "true"
/net/vswitch/child[0004]/capabilities/Tcp
SegmentationOffloadIPv6 = "true"
/net/vswitch/child[0004]/capabilities/Vla
nTag = "true"
/net/vswitch/child[0004]/capabilities/Vla
nUntag = "true"
/net/vswitch/child[0004]/cdp/status =
"listen"
/net/vswitch/child[0004]/mtu = "9000"
/net/vswitch/child[0004]/name =
"vSwitch4"
/net/vswitch/child[0004]/portgroup/child[
0000]/name = "eth3"
/net/vswitch/child[0004]/portgroup/child[
0000]/vlanId = "0"
/net/vswitch/child[0004]/securityPolicy/f
orgedTx = "true"
/net/vswitch/child[0004]/securityPolicy/m
acChange = "true"
/net/vswitch/child[0004]/securityPolicy/p
romiscuous = "false"
/net/vswitch/child[0004]/shapingPolicy/en
abled = "false"
/net/vswitch/child[0004]/teamPolicy/hasUp
linkOrder = "true"
/net/vswitch/child[0004]/teamPolicy/linkC
riteria/beacon = "ignore"
/net/vswitch/child[0004]/teamPolicy/linkC
riteria/fullDuplex = "ignore"
/net/vswitch/child[0004]/teamPolicy/linkC
riteria/minSpeed = "10"
/net/vswitch/child[0004]/teamPolicy/linkC
riteria/pctError = "ignore"
/net/vswitch/child[0004]/teamPolicy/maxAc
tive = "1"
/net/vswitch/child[0004]/teamPolicy/notif
ySwitch = "true"
/net/vswitch/child[0004]/teamPolicy/rever
sePolicy = "true"
/net/vswitch/child[0004]/teamPolicy/rolli
ngRestoration = "false"
/net/vswitch/child[0004]/teamPolicy/team
= "lb_srcid"
/net/vswitch/child[0004]/teamPolicy/uplin
ks[0000]/pnic = "vmnic3"
/net/vswitch/child[0004]/uplinks/child[00
00]/pnic = "vmnic3"
/storage/claimrule[0101]/match/model =
"Universal Xport"
/storage/claimrule[0101]/match/vendor =
"DELL"
/storage/claimrule[0101]/plugin =
"MASK_PATH"
/storage/claimrule[0101]/type = "vendor"
/storage/lun[naa.600508b10010373839414243
44450400]/displayName = "HP Serial
```

```
Attached SCSI Disk
(naa.600508b1001037383941424344450400)"
/storage/lun[naa.600508b10010373839414243
44450400]/fromUser = "false"
/storage/lun[naa.600508b10010373839414243
44450c00]/displayName = "HP Serial
Attached SCSI Disk
(naa.600508b1001037383941424344450c00)"
/storage/lun[naa.600508b10010373839414243
44450c00]/fromUser = "false"
/storage/lun[naa.600508b10010373839414243
44450d00]/displayName = "HP Serial
Attached SCSI Disk
(naa.600508b1001037383941424344450d00)"
/storage/lun[naa.600508b10010373839414243
44450d00]/fromUser = "false"
/storage/lun[naa.600508b10010373839414243
44450e00]/displayName = "HP Serial
Attached SCSI Disk
(naa.600508b1001037383941424344450e00)"
/storage/lun[naa.600508b10010373839414243
44450e00]/fromUser = "false"
/storage/lun[naa.600508b10010373839414243
44450f00]/displayName = "HP Serial
Attached SCSI Disk
(naa.600508b1001037383941424344450f00)"
/storage/lun[naa.600508b10010373839414243
44450f00]/fromUser = "false"
/storage/lun[naa.600508b10010373839414243
44451000]/displayName = "HP Serial
Attached SCSI Disk
(naa.600508b1001037383941424344451000)"
/storage/lun[naa.600508b10010373839414243
44451000]/fromUser = "false"
/storage/lun[naa.600508b10010373839414243
44451100]/displayName = "HP Serial
Attached SCSI Disk
(naa.600508b1001037383941424344451100)"
/storage/lun[naa.600508b10010373839414243
44451100]/fromUser = "false"
/storage/lun[naa.600508b10010373839414243
44451200]/displayName = "HP Serial
Attached SCSI Disk
(naa.600508b1001037383941424344451200)"
/storage/lun[naa.600508b10010373839414243
44451200]/fromUser = "false"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/defaultpsp = "VMW_PSP_MRU"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0000]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0000]/description = "EVA 3000 GL
with ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0000]/model = "HSV101"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0001]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0001]/description = "EVA 5000 GL
with ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0001]/model = "HSV111"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0002]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0002]/description = "EVA 4000/6000
XL with ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0002]/model = "HSV200"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0003]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0003]/description = "EVA 8000/8100
XL with ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0003]/model = "HSV210"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0004]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0004]/description = "EVA 4100/4400
with ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0004]/model = "HSV300"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0005]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0005]/description = "EVA 6400 with
ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0005]/model = "HSV400"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0006]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0006]/description = "EVA 8400 with
ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0006]/model = "HSV450"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0007]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0007]/description = "NetApp with
ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0007]/vendor = "NETAPP"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0008]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0008]/description = "HP MSA A/A
with ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0008]/model = "^MSA2*"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0008]/vendor = "HP"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0009]/claimOptions = "tpgs_on"
```

```
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0009]/description = "HP MSA A/A
with ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0009]/model = "MSA 3G SAS Cntlr"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0009]/vendor = "HP"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0010]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0010]/description =
"active/passive HP StorageWorks SVSP with
ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0010]/model = "HSVX700"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0011]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0011]/description = "active/active
HP StorageWorks SVSP with ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0011]/model = "HSVX740"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0012]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0012]/description = "Intel
Promise"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0012]/model = "Multi-Flex"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0012]/vendor = "Intel"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0013]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0013]/description = "ETERNUS with
ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0013]/vendor = "FUJITSU"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0014]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0014]/description = "EVA 4600 with
ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0014]/model = "HSV340"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0015]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0015]/description = "EVA 6600 with
ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0015]/model = "HSV360"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0016]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0016]/description = "ETERNUS with
ALUA"
/storage/plugin/NMP/config[VMW_SATP_ALUA]
/rules[0016]/vendor = "FUJITSU"
/storage/plugin/NMP/config[VMW_SATP_ALUA_
CX]/defaultpsp = "VMW_PSP_FIXED"
/storage/plugin/NMP/config[VMW_SATP_ALUA_
CX]/rules[0000]/claimOptions = "tpgs_on"
/storage/plugin/NMP/config[VMW_SATP_ALUA_
CX]/rules[0000]/description = "CLARiiON
array in ALUA mode"
/storage/plugin/NMP/config[VMW_SATP_ALUA_
CX]/rules[0000]/vendor = "DGC"
/storage/plugin/NMP/config[VMW_SATP_CX]/d
efaultpsp = "VMW_PSP_MRU"
/storage/plugin/NMP/config[VMW_SATP_CX]/r
ules[0000]/claimOptions = "tpgs_off"
/storage/plugin/NMP/config[VMW_SATP_CX]/r
ules[0000]/description = "All non-ALUA
Clariion Arrays"
/storage/plugin/NMP/config[VMW_SATP_CX]/r
ules[0000]/vendor = "DGC"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/defaultpsp = "VMW_PSP_FIXED"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/rules[0000]/description = "Default
for Fibre Channel"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/rules[0000]/transport = "fc"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/rules[0001]/description = "Default
for iSCSI"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/rules[0001]/transport = "iscsi"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/rules[0002]/description = "IBM SAS
SES-2"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/rules[0002]/model = "SAS SES-2
DEVICE"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/rules[0002]/vendor = "IBM"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/rules[0003]/description = "IBM BCS
RSSM"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/rules[0003]/model = "1820N00"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/rules[0003]/vendor = "IBM"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AA]/rules[0004]/vendor = "HITACHI"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AP]/defaultpsp = "VMW_PSP_MRU"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AP]/rules[0000]/model = "HSG80"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AP]/rules[0000]/vendor = "DEC"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AP]/rules[0001]/description =
"active/passive EVA 3000 GL"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AP]/rules[0001]/model = "HSV100"
```

```
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AP]/rules[0002]/description =
"active/passive EVA 5000 GL"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AP]/rules[0002]/model = "HSV110"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AP]/rules[0003]/claimOptions =
"tpgs_off"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AP]/rules[0003]/description =
"active/passive HP StorageWorks SVSP"
/storage/plugin/NMP/config[VMW_SATP_DEFAU
LT_AP]/rules[0003]/model = "HSVX700"
/storage/plugin/NMP/config[VMW_SATP_EQL]/
defaultpsp = "VMW_PSP_FIXED"
/storage/plugin/NMP/config[VMW_SATP_EQL]/
rules[0000]/description = "All EqualLogic
Arrays"
/storage/plugin/NMP/config[VMW_SATP_EQL]/
rules[0000]/vendor = "EQLOGIC"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
defaultpsp = "VMW_PSP_FIXED"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0000]/claimOptions = "tpgs_off"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0000]/description = "active/active
EVA 3000 GL"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0000]/model = "HSV101"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0001]/claimOptions = "tpgs_off"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0001]/description = "active/active
EVA 5000 GL"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0001]/model = "HSV111"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0002]/claimOptions = "tpgs_off"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0002]/description = "active/active
EVA 4000/6000 XL"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0002]/model = "HSV200"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0003]/claimOptions = "tpgs_off"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0003]/description = "active/active
EVA 8000/8100 XL"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0003]/model = "HSV210"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0004]/claimOptions = "tpgs_off"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0004]/description = "active/active
HP StorageWorks SVSP"
/storage/plugin/NMP/config[VMW_SATP_EVA]/
rules[0004]/model = "HSVX740"
/storage/plugin/NMP/config[VMW_SATP_INV]/
defaultpsp = "VMW_PSP_FIXED"
/storage/plugin/NMP/config[VMW_SATP_INV]/
rules[0000]/model = "Invista"
/storage/plugin/NMP/config[VMW_SATP_INV]/
rules[0000]/vendor = "EMC"
/storage/plugin/NMP/config[VMW_SATP_INV]/
rules[0001]/description = "Invista LUNZ"
/storage/plugin/NMP/config[VMW_SATP_INV]/
rules[0001]/model = "LUNZ"
/storage/plugin/NMP/config[VMW_SATP_INV]/
rules[0001]/vendor = "EMC"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/defaultpsp = "VMW_PSP_FIXED"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0000]/description = "USB Devices"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0000]/transport = "usb"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0001]/description = "IDE Devices"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0001]/transport = "ide"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0002]/description = "RAID Block
Devices"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0002]/transport = "block"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0003]/description = "Parallel
SCSI Devices"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0003]/transport = "parallel"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0004]/description = "Serial
Attached SCSI Devices"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0004]/transport = "sas"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0005]/description = "Serial ATA
Devices"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0005]/transport = "sata"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0006]/description = "Unknown
Devices"
/storage/plugin/NMP/config[VMW_SATP_LOCAL
]/rules[0006]/transport = "unknown"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
defaultpsp = "VMW_PSP_MRU"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0000]/description = "FAStT 700/900"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0000]/model = "^1742*"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0000]/vendor = "IBM"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0001]/description = "FAStT 200"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0001]/model = "^3542*"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0001]/vendor = "IBM"
```

```
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0002]/description = "FAStT 500"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0002]/model = "^3552*"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0002]/vendor = "IBM"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0003]/description = "FAStT
600/DS4300"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0003]/model = "^1722*"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0003]/vendor = "IBM"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0004]/description = "FAStT DS4800"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0004]/model = "^1815*"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0004]/vendor = "IBM"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0005]/description = "FAStT 100"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0005]/model = "^1724*"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0005]/vendor = "IBM"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0006]/description = "DS3X00"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0006]/model = "^1726-*"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0006]/vendor = "IBM"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0007]/description = "DS4000"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0007]/model = "^1814*"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0007]/vendor = "IBM"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0008]/description = "DS5100/DS5300"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0008]/model = "^1818*"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0008]/vendor = "IBM"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0009]/description = "FAStT"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0009]/model = "Universal Xport"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0010]/description = "Dell MD3000"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0010]/model = "MD3000"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0010]/vendor = "DELL"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0011]/description = "Dell MD3000i"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0011]/model = "MD3000i"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0011]/vendor = "DELL"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0012]/model = "OPENstorage 9176"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0012]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0013]/model = "OPENstorage D173"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0013]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0014]/model = "OPENstorage D178"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0014]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0015]/model = "OPENstorage D210"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0015]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0016]/model = "OPENstorage D220"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0016]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0017]/model = "OPENstorage D240"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0017]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0018]/model = "OPENstorage D280"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0018]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0019]/model = "BladeCtlr BC82"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0019]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0020]/model = "BladeCtlr BC84"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0020]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0021]/model = "BladeCtlr BC88"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0021]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0022]/model = "BladeCtlr B210"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0022]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0023]/model = "BladeCtlr B220"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0023]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0024]/model = "BladeCtlr B240"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0024]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0025]/model = "BladeCtlr B280"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0025]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0026]/model = "INF-01-00"
```

```
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0026]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0027]/model = "FLEXLINE 380"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0027]/vendor = "STK"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0028]/model = "OPENstorage 9176"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0028]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0029]/model = "OPENstorage D173"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0029]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0030]/model = "OPENstorage D178"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0030]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0031]/model = "OPENstorage D210"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0031]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0032]/model = "OPENstorage D220"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0032]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0033]/model = "OPENstorage D240"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0033]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0034]/model = "OPENstorage D280"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0034]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0035]/model = "BladeCtlr BC82"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0035]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0036]/model = "BladeCtlr BC84"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0036]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0037]/model = "BladeCtlr BC88"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0037]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0038]/model = "BladeCtlr B210"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0038]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0039]/model = "BladeCtlr B220"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0039]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0040]/model = "BladeCtlr B240"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0040]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0041]/model = "BladeCtlr B280"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0041]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0042]/model = "INF-01-00"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0042]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0043]/model = "FLEXLINE 380"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0043]/vendor = "LSI"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0044]/model = "CSM100_R_FC"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0044]/vendor = "SUN"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0045]/model = "FLEXLINE 380"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0045]/vendor = "SUN"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0046]/model = "CSM200_R"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0046]/vendor = "SUN"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0047]/model = "LCSM100_F"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0047]/vendor = "SUN"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0048]/model = "LCSM100_I"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0048]/vendor = "SUN"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0049]/model = "LCSM100_S"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0049]/vendor = "SUN"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0050]/description = "Sun StorageTek
6580/6780"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0050]/model = "STK6580_6780"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0050]/vendor = "SUN"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0051]/model = "INF-01-00"
/storage/plugin/NMP/config[VMW_SATP_LSI]/
rules[0051]/vendor = "ENGENIO"
/storage/plugin/NMP/config[VMW_SATP_MSA]/
defaultpsp = "VMW_PSP_MRU"
/storage/plugin/NMP/config[VMW_SATP_MSA]/
rules[0000]/model = "MSA1000 VOLUME"
/storage/plugin/NMP/config[VMW_SATP_SVC]/
defaultpsp = "VMW_PSP_FIXED"
/storage/plugin/NMP/config[VMW_SATP_SVC]/
rules[0000]/model = "2145"
/storage/plugin/NMP/config[VMW_SATP_SVC]/
rules[0000]/vendor = "IBM"
/storage/plugin/NMP/config[VMW_SATP_SYMM]
/defaultpsp = "VMW_PSP_FIXED"
```

```
/storage/plugin/NMP/config[VMW_SATP_SYMM]
/rules[0000]/description = "EMC
Symmetrix"
/storage/plugin/NMP/config[VMW_SATP_SYMM]
/rules[0000]/model = "SYMMETRIX"
/storage/plugin/NMP/config[VMW_SATP_SYMM]
/rules[0000]/vendor = "EMC"
/system/uservars/nmp/SATP/DefaultVMwareRu
lesLoaded = "1"
/system/uservars/nmp/SATP/DefaultVMwareSA
TPLocalRulesLoaded = "1"
/system/uservars/nmp/SATP/VMWEsx40U1SatpR
ulesLoaded = "1"
/system/uservars/psa/defaultLunMasksInsta
lled = "1"
/system/uuid = "4b9df9f2-13e7-21a7-fdec-
0026553558d4"
/upgrades/complete[0000]/name =
"depricatePrettyName"
/upgrades/complete[0001]/name =
"moduleLineReformat"
/upgrades/complete[0002]/name =
"enableTSO310"
/upgrades/complete[0003]/name =
"persistVmkNicName"
/vmkernel/module/bnx2.o/options =
"disable_msi=1"
/vmkernel/module/tcpip2v6/enabled =
"false"

========================
```

Leader Node (tpch-node1.vmx)

```
========================

.encoding = "UTF-8"
config.version = "8"
virtualHW.version = "7"
pciBridge0.present = "TRUE"
pciBridge4.present = "TRUE"
pciBridge4.virtualDev = "pcieRootPort"
pciBridge4.functions = "8"
pciBridge5.present = "TRUE"
pciBridge5.virtualDev = "pcieRootPort"
pciBridge5.functions = "8"
pciBridge6.present = "TRUE"
pciBridge6.virtualDev = "pcieRootPort"
pciBridge6.functions = "8"
pciBridge7.present = "TRUE"
pciBridge7.virtualDev = "pcieRootPort"
pciBridge7.functions = "8"
vmci0.present = "TRUE"
nvram = "tpch-node1.nvram"
deploymentPlatform = "windows"
virtualHW.productCompatibility = "hosted"
unity.customColor = "|23C0C0C0"
tools.upgrade.policy = "useGlobal"
powerType.powerOff = "soft"
powerType.powerOn = "default"
powerType.suspend = "hard"
powerType.reset = "soft"

displayName = "tpch-node1"
extendedConfigFile = "tpch-node1.vmxf"

numvcpus = "4"
scsi0.present = "TRUE"
scsi0.sharedBus = "none"
scsi0.virtualDev = "lsilogic"
memsize = "32768"
scsi0:0.present = "TRUE"
scsi0:0.fileName = "tpch-node1.vmdk"
scsi0:0.deviceType = "scsi-hardDisk"
guestOS = "rhel5-64"
uuid.bios = "42 00 4f c0 18 f9 38 82-60
ae e3 80 93 1d 03 65"
vc.uuid = "52 4d 4d 97 db be 12 3c-49 58
c2 63 5e 77 d5 29"
scsi0:0.mode = "independent-persistent"
scsi0:1.present = "TRUE"
scsi0:1.fileName =
"/vmfs/volumes/4b9094e1-906bfbc7-c3f7-
0026553558d6/tpch-node1/tpch-node1.vmdk"
scsi0:1.mode = "independent-persistent"
scsi0:1.deviceType = "scsi-hardDisk"

scsi0:2.present = "TRUE"
scsi0:2.fileName =
"/vmfs/volumes/4b9094f6-51f70a44-4cea-
0026553558d6/tpch-node1/tpch-node1.vmdk"
scsi0:2.mode = "independent-persistent"
scsi0:2.deviceType = "scsi-hardDisk"

scsi0:3.present = "TRUE"
scsi0:3.fileName =
"/vmfs/volumes/4b90950c-adc24fc1-4cf9-
0026553558d6/tpch-node1/tpch-node1.vmdk"
scsi0:3.mode = "independent-persistent"
scsi0:3.deviceType = "scsi-hardDisk"

tools.syncTime = "FALSE"
uuid.location = "56 4d 53 6d fe 1d 3f 9b-
27 80 02 cf cc 14 e1 45"
cleanShutdown = "FALSE"
replay.supported = "FALSE"
sched.swap.derivedName =
"/vmfs/volumes/4b9094cb-34162b35-49fd-
0026553558d6/tpch-node1/tpch-node1-
51b9dfb5.vswp"
scsi0:0.redo = ""
scsi0:1.redo = ""
scsi0:2.redo = ""
scsi0:3.redo = ""
vmotion.checkpointFBSize = "4194304"
pciBridge0.pciSlotNumber = "17"
pciBridge4.pciSlotNumber = "21"
pciBridge5.pciSlotNumber = "22"
pciBridge6.pciSlotNumber = "23"
```

```
pciBridge7.pciSlotNumber = "24"
scsi0.pciSlotNumber = "16"
vmci0.pciSlotNumber = "37"
vmci0.id = "-1826815129"
hostCPUID.0 =
"0000000b756e65476c65746e49656e69"
hostCPUID.1 =
"000106a500100800009ce3bdbfebfbff"
hostCPUID.80000001 =
"00000000000000000000000128100800"
guestCPUID.0 =
"0000000b756e65476c65746e49656e69"
guestCPUID.1 =
"000106a500010800809822010febfbff"
guestCPUID.80000001 =
"00000000000000000000000128100800"
userCPUID.0 =
"0000000b756e65476c65746e49656e69"
userCPUID.1 =
"000106a500100800009822010febfbff"
userCPUID.80000001 =
"00000000000000000000000128100800"
evcCompatibilityMode = "FALSE"

guestOSAltName = "Red Hat Enterprise
Linux 5 (64-bit)"
bios.bootDelay = "10000"

debugStub.linuxOffsets =
"0xfc052158,0xffffffff,0x0,0x0,0xb8e83d8,
0x0,0xb8e83eb,0x0,0xfc0521ac,0xffffffff,0
x0,0x0,0xb8e83f8,0x0"
migrate.hostlog = "./tpch-node1-
51b9dfb5.hlog"
ethernet0.present = "TRUE"
ethernet0.allowGuestConnectionControl =
"FALSE"
ethernet0.virtualDev = "vmxnet"
ethernet0.features = "15"
ethernet0.wakeOnPcktRcv = "FALSE"
ethernet0.networkName = "eth0"
ethernet0.addressType = "vpx"
ethernet0.generatedAddress =
"00:50:56:80:6f:ab"
ethernet1.present = "TRUE"
ethernet1.allowGuestConnectionControl =
"FALSE"
ethernet1.virtualDev = "vmxnet"
ethernet1.features = "15"
ethernet1.wakeOnPcktRcv = "FALSE"
ethernet1.networkName = "eth1"
ethernet1.addressType = "vpx"
ethernet1.generatedAddress =
"00:50:56:80:13:89"
ethernet2.present = "TRUE"

ethernet2.allowGuestConnectionControl =
"FALSE"
ethernet2.virtualDev = "vmxnet"
ethernet2.features = "15"
ethernet2.wakeOnPcktRcv = "FALSE"
ethernet2.networkName = "eth2"
ethernet2.addressType = "vpx"
ethernet2.generatedAddress =
"00:50:56:80:28:ce"
ethernet3.present = "TRUE"

ethernet3.allowGuestConnectionControl =
"FALSE"
ethernet3.virtualDev = "vmxnet"
ethernet3.features = "15"
ethernet3.wakeOnPcktRcv = "FALSE"
ethernet3.networkName = "eth3"
ethernet3.addressType = "vpx"
ethernet3.generatedAddress =
"00:50:56:80:4c:97"
ethernet4.present = "TRUE"

ethernet4.allowGuestConnectionControl =
"FALSE"
ethernet4.virtualDev = "vmxnet"
ethernet4.features = "15"
ethernet4.wakeOnPcktRcv = "FALSE"
ethernet4.networkName = "VM Network"
ethernet4.addressType = "vpx"
ethernet4.generatedAddress =
"00:50:56:80:62:90"
ethernet5.present = "FALSE"
ethernet6.present = "FALSE"
floppy0.present = "FALSE"

ethernet0.pciSlotNumber = "32"
ethernet1.pciSlotNumber = "33"
ethernet2.pciSlotNumber = "34"
ethernet3.pciSlotNumber = "35"
ethernet4.pciSlotNumber = "36"

monitor.virtual_mmu = "hardware"
```

==========

Compute Nodes (tpch-node2.vmx)

==========

```
.encoding = "UTF-8"
config.version = "8"
virtualHW.version = "7"
pciBridge0.present = "true"
pciBridge4.present = "true"
pciBridge4.virtualDev = "pcieRootPort"
pciBridge4.functions = "8"
pciBridge5.present = "true"
pciBridge5.virtualDev = "pcieRootPort"
pciBridge5.functions = "8"
pciBridge6.present = "true"
pciBridge6.virtualDev = "pcieRootPort"
pciBridge6.functions = "8"
pciBridge7.present = "true"
```

```
pciBridge7.virtualDev = "pcieRootPort"
pciBridge7.functions = "8"
vmci0.present = "true"
nvram = "tpch-node2.nvram"
deploymentPlatform = "windows"
virtualHW.productCompatibility = "hosted"
unity.customColor = "|23C0C0C0"
tools.upgrade.policy = "useGlobal"
powerType.powerOff = "soft"
powerType.powerOn = "default"
powerType.suspend = "hard"
powerType.reset = "soft"

displayName = "tpch-node2"
extendedConfigFile = "tpch-node2.vmxf"

numvcpus = "4"
memsize = "32768"
guestOS = "rhel5-64"
uuid.bios = "42 00 78 05 cf 6f ab 89-fc
9c f7 86 1b df 33 61"
vc.uuid = "52 e8 54 67 de b5 93 f1-9c 0d
91 25 7e 46 6d d8"

uuid.location = "56 4d 94 38 09 80 74 d2-
3f 50 67 81 ee 69 36 94"
cleanShutdown = "FALSE"
replay.supported = "FALSE"
sched.swap.derivedName =
"/vmfs/volumes/4b909522-0c3f9f42-d754-
0026553558d6/tpch-node2/tpch-node2-
f56c430a.vswp"
vmotion.checkpointFBSize = "4194304"
pciBridge0.pciSlotNumber = "17"
pciBridge4.pciSlotNumber = "21"
pciBridge5.pciSlotNumber = "22"
pciBridge6.pciSlotNumber = "23"
pciBridge7.pciSlotNumber = "24"
vmci0.pciSlotNumber = "37"
vmci0.id = "467612513"
hostCPUID.0 =
"0000000b756e65476c65746e49656e69"
hostCPUID.1 =
"000106a500100800009ce3bdbfebfbff"
hostCPUID.80000001 =
"00000000000000000000000128100800"
guestCPUID.0 =
"0000000b756e65476c65746e49656e69"
guestCPUID.1 =
"000106a500010800809822010febfbff"
guestCPUID.80000001 =
"00000000000000000000000128100800"
userCPUID.0 =
"0000000b756e65476c65746e49656e69"
userCPUID.1 =
"000106a500100800009822010febfbff"
userCPUID.80000001 =
"00000000000000000000000128100800"
evcCompatibilityMode = "FALSE"
tools.syncTime = "FALSE"

scsi0.present = "true"
scsi0:0.present = "true"

scsi0.sharedBus = "none"
scsi0.virtualDev = "lsilogic"
scsi0:0.fileName = "tpch-node2.vmdk"
scsi0:0.mode = "independent-persistent"
scsi0:0.deviceType = "scsi-hardDisk"
scsi0:1.present = "true"

scsi0:1.fileName =
"/vmfs/volumes/4b909537-c178d13c-2653-
0026553558d6/tpch-node2/tpch-node2.vmdk"
scsi0:1.mode = "independent-persistent"
scsi0:1.deviceType = "scsi-hardDisk"
scsi0:2.present = "true"

scsi0:2.fileName =
"/vmfs/volumes/4b90954d-223b2c16-08c0-
0026553558d6/tpch-node2/tpch-node2.vmdk"
scsi0:2.mode = "independent-persistent"
scsi0:2.deviceType = "scsi-hardDisk"
scsi0:3.present = "true"

scsi0:3.fileName =
"/vmfs/volumes/4b909563-821c8988-c81f-
0026553558d6/tpch-node2/tpch-node2.vmdk"
scsi0:3.mode = "independent-persistent"
scsi0:3.deviceType = "scsi-hardDisk"

scsi0:0.redo = ""
scsi0:1.redo = ""
scsi0:2.redo = ""
scsi0:3.redo = ""
scsi0.pciSlotNumber = "16"

bios.bootDeviceClasses = "allow:hd,net"

guestOSAltName = "Red Hat Enterprise
Linux 5 (64-bit)"
bios.bootDelay = "0"

ethernet0.present = "true"

ethernet0.allowGuestConnectionControl =
"false"
ethernet0.virtualDev = "vmxnet"
ethernet0.features = "15"
ethernet0.wakeOnPcktRcv = "false"
ethernet0.networkName = "eth0"
ethernet0.addressType = "vpx"
ethernet0.generatedAddress =
"00:50:56:80:2b:69"
ethernet1.present = "true"
```

```
ethernet1.allowGuestConnectionControl =
"false"
ethernet1.virtualDev = "vmxnet"
ethernet1.features = "15"
ethernet1.wakeOnPcktRcv = "false"
ethernet1.networkName = "eth1"
ethernet1.addressType = "vpx"
ethernet1.generatedAddress =
"00:50:56:80:1c:71"
ethernet2.present = "true"

ethernet2.allowGuestConnectionControl =
"false"
ethernet2.virtualDev = "vmxnet"
ethernet2.features = "15"
ethernet2.wakeOnPcktRcv = "false"
ethernet2.networkName = "eth2"
ethernet2.addressType = "vpx"
ethernet2.generatedAddress =
"00:50:56:80:05:f0"
ethernet3.present = "true"

ethernet3.allowGuestConnectionControl =
"false"
ethernet3.virtualDev = "vmxnet"
ethernet3.features = "15"
ethernet3.wakeOnPcktRcv = "false"
ethernet3.networkName = "eth3"
ethernet3.addressType = "vpx"
ethernet3.generatedAddress =
"00:50:56:80:7d:25"

cpuid.80000001.eax = "-------------------
-------------"
cpuid.80000001.eax.amd = "---------------
-----------------"
cpuid.80000001.ebx = "-------------------
-------------"
cpuid.80000001.ebx.amd = "---------------
-----------------"
cpuid.80000001.ecx = "-------------------
-------------"
cpuid.80000001.ecx.amd = "---------------
-----------------"
cpuid.80000001.edx = "-----------H-------
-------------"
cpuid.80000001.edx.amd = "-----------H---
-----------------"

ethernet4.present = "FALSE"
ethernet5.present = "FALSE"
ethernet6.present = "FALSE"
floppy0.present = "FALSE"

ethernet0.pciSlotNumber = "32"
ethernet1.pciSlotNumber = "33"
ethernet2.pciSlotNumber = "34"
ethernet3.pciSlotNumber = "35"
monitor.virtual_mmu = "hardware"
```

==========

ParAccel Parameters

==========

```
  name      | setting
 -----------+-----------------------------
-------------------------------
 4byte_hash_pointers | off
 add_missing_from | off
 aggr_hash_table_size | 1048576
 archive_command | unset
 auto_compress | off
 auto_update_compression | off
 backend_concurrency | 1
 block_size | 8192
 Broadcast | xxx.xxx.xxx.xxx
 broadcast_megabytes_ceiling | 2147483647
 check_function_bodies | on
 checkpoint_segments | 3
 checkpoint_timeout | 3600
 checkpoint_warning | 0
 client_encoding | UNICODE
 client_min_messages | ERROR
 col_elim_enabled | on
 comm_compute_port_base | 5460
 comm_ip_leader | 1
 comm_ip_node_first | 2
 comm_ip_node_last | 80
 comm_ip_subnet0 | 10.0.0.%d
 comm_ip_subnet1 | 10.0.1.%d
 comm_ip_subnet2 | 10.0.2.%d
 comm_ip_subnet3 | 10.0.3.%d
 comm_ip_subnet4 | 10.0.1.%d
 comm_ip_subnet5 | 10.0.1.%d
 comm_ip_subnet6 | 10.0.1.%d
 comm_ip_subnet7 | 10.0.1.%d
 comm_leader_port_base | 5450
 comm_num_links | 4
 comm_num_processes | 8
 comm_packet_timeout | 30
 comm_packets_per_notify | 1
 commit_delay | 0
 commit_siblings | 5
 compile_max_memory | 300
 compile_max_seconds | 30
 concurrency_model | 2
 config_file |
/home/paraccel/padb/rel/etc/xenpostgresql
.conf
 consistency_action | EMAIL
```

```
 consistency_aggregate | 0
 consistency_count | 1
 consistency_count_tolerance | 0
 consistency_email_subj | subject
 consistency_mode | 0
 consistency_period | 10
 consistency_slack_period | 180
 constant_primary_sort_key_check | off
 contains_selectivity | 0.01
 cpu_index_tuple_cost | 0.001
 cpu_operator_cost | 0.0025
 cpu_tuple_cost | 0.01
 cpus_per_node | 0
 CSQ_hint_strategy | 2
 ctas_auto_analyze | on
 ctas_phys_props | on
 custom_variable_classes | unset
 data_directory |
/home/paraccel/padb/data/pg
 DateStyle | ISO, MDY
 db_user_namespace | off
 deadlock_timeout | 1000
 debug_concurrent_writes | off
 debug_pretty_print | off
 debug_print_parse | off
 debug_print_plan | off
 debug_print_rewritten | off
 debug_shared_buffers | 0
 default_compressdir | /tmp
 default_statistics_target | 10
 default_tablespace | unset
 default_transaction_isolation | read
committed
 default_transaction_read_only | off
 default_with_oids | on
 delayed_scan_limit | 0
 destroy_order_after_hj | off
 dflt_precision | 18
 dflt_scale | 0
 disk_list | md10, md11
 disk_load_sorted | on
 disk_mem_pct | 75
 disk_mirror_count | 0
 disk_prefix | /dev/
 disk_work_partitions | unset
 diskhash_pessimism | 50
 display_dist_info_in_explain | on
 display_missing_stats_in_explain | on
 dist_both_on_max_nd | on
 dist_both_on_max_nd | on
 dist_first_temp_col | off
 distinct_hash_table_size | 1048576
 Domain | mycompany.com
 downcase_delimited_identifier | on
 DSN0 |
blankDSN,192.168.1.999,fooBar,fooBar,1234
,off,1234,fooBar,userName, password
 DSN1 |
sampleDSN1,192.168.1.51,FedCivilSample,MS
2005,4494,off,1434,nop,sa,elfsql
 DSN10 | unset
 DSN11 | unset
 DSN12 | unset
 DSN13 | unset
 DSN14 | unset
 DSN15 | unset
 DSN16 | unset
 DSN17 | unset
 DSN18 | unset
 DSN19 | unset
 DSN2 |
sampleDSN2,192.168.1.51,FedCookSample,MS
2005,4494,off,1435,SQL51,sa,elfsql
 DSN20 | unset
 DSN3 |
sql51_sync,192.168.1.51,sync1_51,MS
2005,4494,off,1436,SQL51,uid,pwd,t1
 DSN4 | unset
 DSN5 | unset
 DSN6 | unset
 DSN7 | unset
 DSN8 | unset
 DSN9 | unset
 dump_dbgen | off
 dynamic_library_path | $libdir
 effective_cache_size | 1000
 email_from | cluster@paraccel.com
 email_pwd | unset
 email_recipient | dev.null@localhost
 email_server | entry.paraccel.com
 email_user | unset
 enable_add_drop_column | off
 enable_assert_in_dtor | off
 enable_auto_disk_cache | off
 enable_binary_rrscan_search | off
 enable_column_check | on
 enable_concurrent_writes | on
 enable_ddl_nowait | on
 enable_disk_by_slice | off
 enable_disk_pool | on
 enable_drop_nonexistent_table_error | on
 enable_fast_commit | on
```

```
 enable_hash_distinct | on
 enable_hashagg | on
 enable_hashjoin | on
 enable_hashjoin_not_in | on
 enable_incremental_vacuum | off
 enable_indexscan | on
 enable_intersect_dist_workaround | on
 enable_intersect_network_node | on
 enable_larger_inner_for_hashjoin | off
 enable_larger_inner_for_mergejoin | off
 enable_like_inline | off
 enable_logging | on
 enable_memory_integrity_check | off
 enable_numeric_inline | off
 enable_oracle_outer_join_syntax | on
 enable_query_stats | off
 enable_read_caching | on
 enable_rrscan | on
 enable_seqscan | on
 enable_single_node | off
 enable_sort | on
 enable_tidscan | on

enable_update_cardinality_violation_check
| on
 enable_varchar_fast_lookup | 1
 eth_list | eth0, eth1, eth2, eth3
 explain_pretty_print | on
 explain_print_max_items_in_list |
2147483647
 external_pid_file | unset
 extra_float_digits | 0
 extra_scan_tuple_cost | 0.1
 float_decimal_display | -1
 force_explain | off
 force_fromexpr_str | unset
 force_joinexpr_str | unset
 force_unsupported_CSQ_check | on
 from_collapse_limit | 8
 fromexpr_max_size | 1
 fsync | on
 Gateway | xxx.xxx.xxx.xxx
 gconf_add_path_strategy | 0
 gconf_clamp_row_estimate_floor | 1
 gconf_contingent_num_compute_nodes | 48
 gconf_DecorrDbgFlags | -1
 gconf_decorrelated_explain_path | unset
 gconf_decorrelated_path | unset
 gconf_decorrelated_temp_name_base |
unset
 gconf_defrag_cache_size | 2048
 gconf_disable_substr_freq | off
 gconf_disk_cache_size | 2000
 gconf_disk_read_o_direct | on
 gconf_disk_write_o_direct | on
 gconf_dump_decorrelated_queries | off
 gconf_enable_projections | off
 gconf_explain_decorrelated_queries | off
 gconf_group_having_floor | 1000
 gconf_keep_decorrelated_temp | off
 gconf_pad_sensitive | off
 gconf_plan_for_rrscan | on
 gconf_test_abort | off
 gconf_use_keys_in_join_selectivity | off
 gconf_user_query | off
 generate_graph_file | off
 generate_sql_file | off
 generate_sql_stderr | off
 geqo | off
 geqo_effort | 5
 geqo_generations | 0
 geqo_pool_size | 0
 geqo_selection_bias | 2
 geqo_stable_plan | on
 geqo_threshold | 12
 group_cols_for_analyze | 0
 gui_isDbClean | 1
 hash_keys_limit | 100
 hba_file |
/home/paraccel/padb/data/pg/pg_hba.conf
 Hostname | paraccel
 hot_standby_list | unset
 ident_file |
/home/paraccel/padb/data/pg/pg_ident.conf
 in_to_scalar_array_threshold | 100000000
 integer_datetimes | on
 IP_Address | xxx.xxx.xxx.xxx
 join_collapse_limit | 8
 join_rrscan_limit | 1000
 joinsel_floor | on
 krb_server_keyfile | unset
 lc_collate | en_US.UTF-8
 lc_ctype | en_US.UTF-8
 lc_messages | en_US.UTF-8
 lc_monetary | en_US.UTF-8
 lc_numeric | en_US.UTF-8
 lc_time | en_US.UTF-8
 leader_bcast_sendwin | 1
 leader_distribute_sendwin | 1
 leader_shared_mem_size | 24000
 leader_unload | off
 limit_varchar_prec | 65535
```

```
 linkedin_demo | off
 listen_addresses | *
 load_directio | 1
 lock_memory | off
 log_connections | off
 log_destination | stderr
 log_directory | pg_log
 log_disconnections | off
 log_duration | off
 log_error_verbosity | default
 log_executor_stats | off
 log_explain_plans | off
 log_filename | postgresql-%Y-%m-
%d_%H%M%S.log
 log_hostname | off
 log_line_prefix | unset
 log_min_duration_statement | -1
 log_min_error_statement | PANIC
 log_min_messages | ERROR
 log_parser_stats | off
 log_planner_stats | off
 log_protocol_level | 0
 log_protocol_user | unset
 log_rotation_age | 1440
 log_rotation_size | 10240
 log_statement | none
 log_statement_stats | off
 log_truncate_on_rotation | off
 maintenance_work_mem | 16384
 max_connections | 100
 max_distincts | 0
 max_files_per_process | 1000
 max_fsm_pages | 20000
 max_fsm_relations | 1000
 max_function_args | 32
 max_hash_table_size | 8388608
 max_identifier_length | 127
 max_index_keys | 32
 max_locks_per_transaction | 64
 max_rules_passes | 2
 max_scale | 100
 max_stack_depth | 204800
 mcv_threshold | 10
 mem_low_pct | 97
 memory_dump_size | 1024
 min_hash_table_size | 65536
 min_step_mem | 5
 mirror_temp_data | off
 mssql_numeric_rules | off
 mtu0_size | 9000
 mtu1_size | 9000
 mtu2_size | 9000
 mtu3_size | 9000
 mtu4_size | unset
 mtu5_size | unset
 mtu6_size | unset
 mtu7_size | unset
 Netmask | 255.255.255.0
 Network | xxx.xxx.xxx.xxx
 network_byte_cost | 80
 new_copy_dist_pathkeys | on
 notin_null_check | off
 numa_memory | 0
 odbc_errors | 1
 odbcini | /usr/local/etc/odbc.ini
 odbcsysini | /usr/local/etc
 optimize_qualifier_order | on
 pack_hash_table | off
 padb_analyze | on
 parallel_load_linenums | on
 parser_language | pg
 parser_override | off
 password_encryption | on
 pgdata | /home/paraccel/padb/data/pg
 pgport | 5439
 pgschema_directory |
/home/paraccel/padb/data/pg
 plan_as_if_num_compute_nodes | 0
 plan_as_if_slices_per_node | 0
 plan_for_rrjoin | off
 plan_perf_stats | off
 plan_querytext | off
 plan_rule_stats | off
 plan_state_log | on
 planner_invocation_info | off
 planner_mode | volt_partial_join_order
 planner_mode_log | off
 port | 5439
 pre_auth_delay | 0
 prefetch_limit | 200
 preload_libraries | unset
 preload_memory | off
 preserve_timestamp_trailing_zeros | off
 Primary_DNS_Server | xxx.xxx.xxx.xxx
 print_join_relids | off
 print_plan_recursive_for_compare_plan |
on
 print_qual_cleaning | off
 print_sample_set | off
 qrtr_advanced | 0
```

```
 qrtr_basic | inmemory, localhost, dev,
MS 2005, 61433, on, 61450, SQL00, uid,
pwd
 qrtr_enable | on
 qrtr_graceful_type | 0
 qrtr_passthrough | 0
 qrtr_qlist | 1
 qrtr_tdsver | 9
 qrtr_temptable | 0
 qrtr_xen_user | paraccel
 qrtr_xenerror | 0
 query_group | unset
 query_working_mem | 10000
 raidp_standby_node_count | 0
 ramdisk_directory | /mnt/ramdisk
 random_page_cost | 4
 random_stat_sample | on
 read_ahead | 4
 recursive_rpp | off
 redirect_stderr | off
 regex_flavor | advanced
 rendezvous_name | unset
 require_maint_log_messages | off
 rrscan_bool_expr_limit | 0
 san_list | unset
 scan_pristine | on
 scan_table_for_count | on
 search_path | $user, public
 Secondary_DNS_Server | xxx.xxx.xxx.xxx
 sem_pool_size | 10000
 server_encoding | UNICODE
 server_version | 8.0.2
 shared_buffers | 1000
 shared_mem_size | 27000
 show_explain_entity_names | on
 silent_mode | off
 simulate_bcast | on
 skip_decorrelated_analyze | off
 skip_pg_decorrelation | off
 slices_per_node | 4
 sort_limit_linear_max | 10
 sort_limit_max | 50000
 spike | off
 sql_database_pwd | elfsql
 sql_database_uid | sa
 sql_inheritance | on
 ssb_dim_mult | 1
 ssl | off
 star_planner_broadcast_threshold | 1e+06
 star_planner_threshold | 12
 start_command | amake -f run
 startup_aborted_transactions_percent |
20
 stat_sample_threshold | 12000
 statement_timeout | 0
 stats_block_level | off
 stats_command_string | off
 stats_reset_on_server_start | on
 stats_row_level | off
 stats_start_collector | on
 stop_command | killall -9 padb psql
 superuser_reserved_connections | 2
 sync_action_email | true
 sync_email_subject | Synchronization
Error
 sync_period | 10
 sync_rows_uncommitted | 1000000
 synchronization_active | false
 synchronization_enabled | false
 syslog_facility | LOCAL0
 syslog_ident | postgres
 sysmgr_default_log_level | 4
 sysmgr_graceful | 1800
 sysmgr_health_check_period | 30
 sysmgr_mc_addr | 239.255.79.151
 sysmgr_mc_port | 4579
 sysmgr_res_disabled | unset
 sysmgr_res_limits | restartdb, 3, 300
 sysmgr_svc_directory |
/home/paraccel/padb_svc
 sysmgr_xen_supp_group | unset
 sysmgr_xen_user_name | paraccel
 temp_table_distribution | leave local
 TimeZone | PST8PDT
 trace_notify | off
 transaction_isolation | read committed
 transaction_read_only | off
 transform_null_equals | off
 treat_nulls_as_empty_strings | off

truncate_on_implicit_cast_to_shorter_char
| off
 tsql_statement_autocommit | off
 tuple_header_size | 8
 unix_socket_directory | unset
 unix_socket_group | unset
 unix_socket_permissions | 511
 unload_buf_size | 4
 unload_io_mode | stream
 unload_permissions_mask | 600
 use_avg_count_for_sample | off
 use_build_keys_methods | on
```

```
 use_column_alias_everywhere | on
 use_partial_order_for_aggr | off
 use_tname_for_sample_seed | off
 use_version2_string_hashfn | off
 user_queues | 10
 vacuum_cost_delay | 0
 vacuum_cost_limit | 200
 vacuum_cost_page_dirty | 20
 vacuum_cost_page_hit | 1
 vacuum_cost_page_miss | 10
 vacuum_max_buffer | 10000
 virident_demo | off
 volt_all_check_error | off
 volt_canonical_sql | on
 volt_csq_check | on
 volt_csq_check_error | on
 volt_debug_manager | off
 volt_debug_translation | off
 volt_decorr_csq | on
 volt_dump_stack_trace | off
 volt_enable | on
 volt_fudge_factor | 0.5
 volt_good | off
 volt_group_size | 10
 volt_ignore_count_bug | off
 volt_log_dir |
/home/paraccel/padb/rel/bin/../../data/tr
ace/planner

volt_null_fraction_floor_to_add_is_not_nu
ll | 0.1
 volt_planner_threshold | 12
 volt_push_outer_BEJPred | on
 volt_rule_settings | unset
 volt_sanity_check_level | 1
 volt_sequential_temp_table_names | off
 volt_shape | ()
 volt_vms_dist_sq | on
 volt_vms_skiplevel_only | on
 volt_vms_td_materialize | on
 wal_buffers | 8
 wal_sync_method | fdatasync
 window_frame_size_max | 50000
 WLQ0 | stv_queries, 0, 1, .*stv_.*
 WLQ1 | stl_queries, 0, 1, .*stl_.*
 WLQ10 | unset
 WLQ11 | unset
 WLQ12 | unset
 WLQ13 | unset
 WLQ14 | unset
 WLQ15 | unset
 WLQ16 | unset
 WLQ17 | unset
 WLQ18 | unset
 WLQ19 | unset
 WLQ2 | unset
 WLQ20 | unset
 WLQ21 | unset
 WLQ22 | unset
 WLQ23 | unset
 WLQ24 | unset
 WLQ25 | unset
 WLQ26 | unset
 WLQ27 | unset
 WLQ28 | unset
 WLQ29 | unset
 WLQ3 | unset
 WLQ30 | unset
 WLQ31 | unset
 WLQ32 | unset
 WLQ33 | unset
 WLQ34 | unset
 WLQ35 | unset
 WLQ36 | unset
 WLQ37 | unset
 WLQ38 | unset
 WLQ39 | unset
 WLQ4 | unset
 WLQ5 | unset
 WLQ6 | unset
 WLQ7 | unset
 WLQ8 | unset
 WLQ9 | unset
 work_mem | 1048576
 xen_memory_segment_key | 20848
 xnode_bcast_sendwin | 2
 xnode_data_to_leader_sendwin | 1
 xnode_distribute_sendwin | 1
 zero_damaged_pages | off
```

Appendix B: Programs and Scripts

```
====================
calculate_load_time.bash
====================
#!/bin/bash

# calculates load time from the start and
end times in the files
#   start_load.out
#   end_load.out

# called from tpch_report.bash

# final version

if (( $# < 2 ))
then
  echo "usage:  $0  end_load_time
start_load_time"
  echo ""
  echo "           - returns load_time"
  echo "             calls;"
  echo "
sec_from_epoch.bash"
  echo "                my_calc.pl"
  exit
fi

elt=$1     # of the form: yyyy-mm-dd
hh:mm:ss
# echo "end_load_time=$elt   [debug
calculate_load_time.bash]"

end_secs_from_epoch=`secs_from_epoch.pl
"$elt"`
# echo
"end_secs_from_epoch=$end_secs_from_epoch
[debug   calculate_load_time.bash]"

slt=$2     # of the form: yyyy-mm-dd
hh:mm:ss
# echo "start_load_time=$slt   [debug
calculate_load_time.bash]"

start_secs_from_epoch=`secs_from_epoch.pl
"$slt"`
# echo
"start_secs_from_epoch=$start_secs_from_e
poch   [debug
tpch_throughput_interval.bash]"

((lt=end_secs_from_epoch-
start_secs_from_epoch))

lt_dhms=`s2dhms.bash $lt`

echo $lt_dhms

====================
clean.sh
====================
:
# delete output files

rm -f *.out
rm -f *.trunc
rm -f *.ooo
rm -f *.params
rm -f stream*.sql
rm -f *.log
rm -f *.sqlx
rm -f *.params
rm -f update*.sql

====================
create_tables.sql
====================
BEGIN;

create table part (
  p_partkey int8 not null  ,
  p_name varchar(55) not null,
  p_mfgr char(25) not null ,
  p_brand char(10) not null ,
  p_type varchar(25) not null,
  p_size int4 not null ,
  p_container char(10) not null ,
  p_retailprice numeric(12,2) not null,
  p_comment varchar(23) not null
);

create table region (
  r_regionkey int4 not null ,
  r_name char(25) not null,
  r_comment varchar(152) not null
);

create table nation (
  n_nationkey int4 not null ,
  n_name char(25) not null,
```

```
  n_regionkey int4 not null,
  n_comment varchar(152) not null
);

create table supplier (
  s_suppkey int4 not null ,
  s_name char(25) not null,
  s_address varchar(40) not null,
  s_nationkey int4 not null ,
  s_phone char(15) not null,
  s_acctbal numeric(12,2) not null,
  s_comment varchar(101) not null
);


create table partsupp (
  ps_partkey int8 not null  ,
  ps_suppkey int4 not null,
  ps_availqty int4 not null ,
  ps_supplycost numeric(12,2) not null,
  ps_comment varchar(199) not null
);

create table customer (
  c_custkey int8 not null  ,
  c_name varchar(25) not null,
  c_address varchar(40) not null,
  c_nationkey int4 not null ,
  c_phone char(15) not null,
  c_acctbal numeric(12,2) not null ,
  c_mktsegment char(10) not null,
  c_comment varchar(117) not null
);

create table orders (
  o_orderkey int8 not null  ,
  o_custkey int8 not null,
  o_orderstatus char(1) not null,
  o_totalprice numeric(12,2) not null ,
  o_orderdate date not null ,
  o_orderpriority char(15) not null ,
  o_clerk char(15) not null,
  o_shippriority int4 not null ,
  o_comment varchar(79) not null
);

create table lineitem (
  l_orderkey int8 not null  ,
  l_partkey int8 not null,
  l_suppkey int4 not null,
  l_linenumber int4 not null ,
  l_quantity numeric(12,2) not null ,
  l_extendedprice numeric(12,2) not null
,
  l_discount numeric(12,2) not null ,
  l_tax numeric(12,2) not null ,
  l_returnflag char(1) not null,
  l_linestatus char(1) not null,
  l_shipdate date not null ,
  l_commitdate date not null ,
  l_receiptdate date not null ,
  l_shipinstruct char(25) not null ,
  l_shipmode char(10) not null ,
  l_comment varchar(44) not null
);

create table delete_ids (
  d_orderkey int8 not null
);

COMMIT;
```

==========

dbgen.sh

==========

```
#! /bin/bash

cd $HOME/run/scripts
amake -o all
cqi allx rm -rf
/home/paraccel/padb/data/disks/\*/tpch/\*
cqi allx rm -rf
/home/paraccel/padb/data/disks/\*/dev
cqi initdb -f -c
sleep 15
cqi xstart
psql dev -c "create database tpch"
psql tpch -f create_tables.sql
psql tpch -c "xpx 'dbgen 1000'"
rm -f /tpch_data/*
cd /tpch_data
cp $HOME/padb/rel/share/tpc/dists.dss .
$HOME/padb/rel/share/tpc/dbgen -v -C 30 -
U 16 -s 1000
cd $HOME/run/scripts

if [ -z $1 ] ; then
    sh localize_updates.sh
fi

sleep 15
cqi xstop
cqi initdb -f -c
sleep 15
cqi xstart
```

==========

dbtables.sh

==========

```
#! /bin/bash

psql tpch -At -c "select 'lineitem',
count(*) from lineitem"
psql tpch -At -c "select 'orders',
count(*) from orders"
psql tpch -At -c "select 'customer',
count(*) from customer"
```

```
psql tpch -At -c "select 'part', count(*)
from part"
psql tpch -At -c "select 'partsupp',
count(*) from partsupp"
psql tpch -At -c "select 'supplier',
count(*) from supplier"
psql tpch -At -c "select 'nation',
count(*) from nation"
psql tpch -At -c "select 'region',
count(*) from region"
```

==========

dbtables-xen.sql

==========

```
\d lineitem;
\d region;
\d nation;
\d customer;
\d part;
\d supplier;
\d partsupp;
\d orders;
```

==========

do1tb.sh

==========

```
#/usr/bin/bash

# Create results directory

timestamp=`date "+%m_%d_%H:%M"`
dir=$HOME/results/$timestamp
mkdir -p $dir
echo $dir > /tmp/this_tpch

# Load

cqi xstop
echo -n sleep 15
for i in $(seq 1 15);do echo -n .;sleep
1;done
echo done

cqi allx rm ~/padb/data/log/\*

cqi initdb -f -c
cqi xstart
cd $HOME/run/scripts
pwd >> $dir/tpch.log
sh clean.sh
tpch.sh load 1000 7 < go >> $dir/tpch.log

sh dbtables.sh >> $dir/tpch.log
sh get10rows.sh >> $dir/tpch.log

echo "Executing RI" >> $dir/tpch.log
psql tpch -At -f ri.sql >> $dir/tpch.log
cqi xstop
echo -n sleep 10
for i in $(seq 1 10);do echo -n .;sleep
1;done
echo done
cqi xstart
#
# # Run
#
psql dev -c "select
sum(count),sum(time),name from
stv_sem_usage group by 3 order by 2 desc"
tpch.sh all 1000 7 48 132 50000 < go >>
$dir/tpch.log
psql dev -c "select
sum(count),sum(time),name from
stv_sem_usage group by 3 order by 2 desc"

#
# Grab files, cleanup
cp create_tables.sql $dir
cp stream*.sql $dir
echo Moving result files to $dir
rm s*q*-?.out
mv *.out $dir
```

==========

end_throughput_interval.bash

==========

```
#!/bin/bash

if (( $# < 1 ))
then
  echo
  echo "usage:  $0  num_streams  "
  echo
  echo "           - returns ending time
for the last throughput stream to finish"
  echo
  exit
fi

ns=$1

# set string to "stream1.out stream2.out
... stream$ns.out"

for i in stream[1-9]*.out;   # gets every
streamN.out except stream0.out
do
    num_plus_out=${i#stream}
    num=${num_plus_out%.out}
    #echo $num
    if ((num<=ns))
    then
         string=$string" "$i
    fi
done
```

```
# create a file of completion times of
the query streams
grep 'Stream completion' $string  | cut -
d'|' -f 2 | sed "s/^\s//" >
/tmp/completion_times

# append the largest completion time of
the refresh streams to the file
max_pre_refresh=$(grep STOP
update_throughput.out |   \
                                    cut -
d'|' -f 2 | sed "s/^\s//")
max_refresh=${max_pre_refresh%.*}
echo $max_refresh >>
/tmp/completion_times

# return the latest time
sort /tmp/completion_times >
/tmp/ascending_times

max=`tail -1 /tmp/ascending_times`     #
get the highest time
echo $max
```

==========

gen_streams.sh

==========

```
#$SHELL
echo " Gen_Streams got called"

if (( $# < 2 ))
then
     echo "usage:  $0   seed1 seed2
scale_factor  num_streams"
     exit
fi

DSS_QUERY=$HOME/padb/rel/share/tpc
SCRIPTS_DIR=$HOME/run/scripts

seed1=$1
seed2=$2
sf=$3
ns=$4

i=0

# cd $DSS_QUERY
cd $SCRIPTS_DIR

while ((i<=ns))
do

   #Create params file for stream $i
   echo "Creating param files for stream
$i"
   echo
   qn=1
   $DSS_QUERY/qgen -p $i -l
$SCRIPTS_DIR/stream$i.params -r
$seed1$seed2 -s $sf > /dev/null
   echo "Params files created"
   echo

   $DSS_QUERY/qgen -a -p $i -r
$seed1$seed2 -i $SCRIPTS_DIR/begin.sql -t
$SCRIPTS_DIR/end.sql -s $sf \
   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 >
$SCRIPTS_DIR/stream${i}.sql
   ((seed2=seed2+1))

if (($i == 0))
then
      echo "Stream $i (for power test)
created."
fi
if (($i > 0))
then
      echo "Stream $i created."
fi
((i=i+1))
done

((seed2=seed2-1))
echo $seed1$seed2
```

==========

get10rows.sh

==========

```
#/usr/bin/bash

echo "select * from lineitem order by 1
limit 10:"
psql tpch -c "select * from lineitem
order by 1 limit 10"
echo
echo

echo "select * from orders order by 1
limit 10:"
psql tpch -c "select * from orders order
by 1 limit 10"
echo
echo

echo "select * from part order by 1 limit
10:"
psql tpch -c "select * from part order by
1 limit 10"
echo
echo

echo "select * from partsupp order by 1
limit 10:"
```

```
psql tpch -c "select * from partsupp
order by 1 limit 10"
echo
echo

echo "select * from supplier order by 1
limit 10:"
psql tpch -c "select * from supplier
order by 1 limit 10"
echo
echo

echo "select * from nation order by 1
limit 10:"
psql tpch -c "select * from nation order
by 1 limit 10"
echo
echo

echo "select * from region order by 1
limit 10:"
psql tpch -c "select * from region order
by 1 limit 10"
echo
echo

echo "select * from customer order by 1
limit 10:"
psql tpch -c "select * from customer
order by 1 limit 10"
echo
echo
```

==========

get_cardinal_info.sh

==========

```
#!/usr/bin/bash
date > cardinal_info.out
echo "Cardinal info for the current data
set" >> cardinal_info.out
echo "------------------" >>
cardinal_info.out
for table in lineitem customer orders
part partsupp supplier nation region; do
echo $table count: >> cardinal_info.out;
echo "-----------" >> cardinal_info.out;
psql tpch -c "select count(*) from
$table" >> cardinal_info.out; echo "-----
------" >> cardinal_info.out;done
```

==========

go

==========

```
y
y
y
y
y
```

==========

localize_updates.sh

==========

```
psql tpch -c "xpx 'localize delete1
/tpch_data/delete.1'"
psql tpch -c "xpx 'localize delete2
/tpch_data/delete.2'"
psql tpch -c "xpx 'localize delete3
/tpch_data/delete.3'"
psql tpch -c "xpx 'localize delete4
/tpch_data/delete.4'"
psql tpch -c "xpx 'localize delete5
/tpch_data/delete.5'"
psql tpch -c "xpx 'localize delete6
/tpch_data/delete.6'"
psql tpch -c "xpx 'localize delete7
/tpch_data/delete.7'"
psql tpch -c "xpx 'localize delete8
/tpch_data/delete.8'"
psql tpch -c "xpx 'localize delete9
/tpch_data/delete.9'"
psql tpch -c "xpx 'localize delete10
/tpch_data/delete.10'"
psql tpch -c "xpx 'localize delete11
/tpch_data/delete.11'"
psql tpch -c "xpx 'localize delete12
/tpch_data/delete.12'"
psql tpch -c "xpx 'localize delete13
/tpch_data/delete.13'"
psql tpch -c "xpx 'localize delete14
/tpch_data/delete.14'"
psql tpch -c "xpx 'localize delete15
/tpch_data/delete.15'"
psql tpch -c "xpx 'localize delete16
/tpch_data/delete.16'"
psql tpch -c "xpx 'localize lineitem_u1
/tpch_data/lineitem.tbl.u1'"
psql tpch -c "xpx 'localize lineitem_u2
/tpch_data/lineitem.tbl.u2'"
psql tpch -c "xpx 'localize lineitem_u3
/tpch_data/lineitem.tbl.u3'"
psql tpch -c "xpx 'localize lineitem_u4
/tpch_data/lineitem.tbl.u4'"
psql tpch -c "xpx 'localize lineitem_u5
/tpch_data/lineitem.tbl.u5'"
psql tpch -c "xpx 'localize lineitem_u6
/tpch_data/lineitem.tbl.u6'"
psql tpch -c "xpx 'localize lineitem_u7
/tpch_data/lineitem.tbl.u7'"
psql tpch -c "xpx 'localize lineitem_u8
/tpch_data/lineitem.tbl.u8'"
psql tpch -c "xpx 'localize lineitem_u9
/tpch_data/lineitem.tbl.u9'"
psql tpch -c "xpx 'localize lineitem_u10
/tpch_data/lineitem.tbl.u10'"
```

```
psql tpch -c "xpx 'localize lineitem_u11
/tpch_data/lineitem.tbl.u11'"
psql tpch -c "xpx 'localize lineitem_u12
/tpch_data/lineitem.tbl.u12'"
psql tpch -c "xpx 'localize lineitem_u13
/tpch_data/lineitem.tbl.u13'"
psql tpch -c "xpx 'localize lineitem_u14
/tpch_data/lineitem.tbl.u14'"
psql tpch -c "xpx 'localize lineitem_u15
/tpch_data/lineitem.tbl.u15'"
psql tpch -c "xpx 'localize lineitem_u16
/tpch_data/lineitem.tbl.u16'"
psql tpch -c "xpx 'localize orders_u1
/tpch_data/orders.tbl.u1'"
psql tpch -c "xpx 'localize orders_u2
/tpch_data/orders.tbl.u2'"
psql tpch -c "xpx 'localize orders_u3
/tpch_data/orders.tbl.u3'"
psql tpch -c "xpx 'localize orders_u4
/tpch_data/orders.tbl.u4'"
psql tpch -c "xpx 'localize orders_u5
/tpch_data/orders.tbl.u5'"
psql tpch -c "xpx 'localize orders_u6
/tpch_data/orders.tbl.u6'"
psql tpch -c "xpx 'localize orders_u7
/tpch_data/orders.tbl.u7'"
psql tpch -c "xpx 'localize orders_u8
/tpch_data/orders.tbl.u8'"
psql tpch -c "xpx 'localize orders_u9
/tpch_data/orders.tbl.u9'"
psql tpch -c "xpx 'localize orders_u10
/tpch_data/orders.tbl.u10'"
psql tpch -c "xpx 'localize orders_u11
/tpch_data/orders.tbl.u11'"
psql tpch -c "xpx 'localize orders_u12
/tpch_data/orders.tbl.u12'"
psql tpch -c "xpx 'localize orders_u13
/tpch_data/orders.tbl.u13'"
psql tpch -c "xpx 'localize orders_u14
/tpch_data/orders.tbl.u14'"
psql tpch -c "xpx 'localize orders_u15
/tpch_data/orders.tbl.u15'"
psql tpch -c "xpx 'localize orders_u16
/tpch_data/orders.tbl.u16'"
```

==========

looper

==========

```
#! /bin/csh

set i = 0
while 1
    echo "Running [$i] $* ..."
    $*

    @ i = ($i + 1)
end
```

==========

my_calc.pl

==========

```
#!/usr/bin/perl

# -w removed from above to surpress
runtime warnings on garbadge expressions
i.e.
#                       function keys,
escape sequence, etc. as well a certain
#                       string
expressions eval doesn't understand

# final version:  dec 21 2002

# evaluates all kinds of expressions,
except as noted below
#
#   we can do everything except exp! when
exp contains parentheses
#   but  factorial(exp) works for
arbitrary exp
#
#    todo: maybe lift this restriction

our $ARGC = @ARGV ;

my $show_commas;      # initially turned
off
my $show_trace;       # initially turned
off

if ( $ARGC >= 1 )   # these are the batch
cases
  {
     if ($ARGC == 2)
       {
         $show_commas = $ARGV[1];
       }
     else
       {
         $show_commas = '';
       }

     if ($ARGV[0] eq "help")
       {
          print "\nusage:  $0 \n";
          print "
interactive case \n";
          print "   or\n\n";
          print "usage:  $0  help\n";
          print "              displays
usage info (i.e. this output)\n";
          print "   or\n\n";
          print "usage:  $0  expression
[any 2nd arg means commify]\n";
          print "              - displays
evaluated expression and exits\n";
```

```
          print "               - place
expression in quotes if it contains
spaces\n\n";
          exit;
       }

     # $ARGV[0] is the first actual
argument, not the script name as in C or
the shell

     my $result =
evaluate_perl($ARGV[0]);

     # print $ARGV[0], " = ", $result,
"\n";
     # just show the result in the batch
case, so it can be used in a pipeline

     print  $result, "\n";

     exit;
  }

# interactive case
#    need to commify this

my $input;

my $mode = 'perl';

for (;;)
  {
    print "enter arithmetic or string
expression (q=exit): ";
    $input = <STDIN>;
    chomp ($input);
    if ( $input =~ /quit|exit|^q$|^e$/i )
      {
        exit;
      }
    elsif ($input eq 'help')
      {
        print_help();
      }
    else  # evaluate expression
      {
        if ( $input =~
/commas\s*$|c\s*$/i )  # turn show_commas
on
          {
            $show_commas = 'yes';
            $input =~
s/\s*commas\s*$|\s*c\s*$//i;  # erase "c"
or "commas" and

# surrounding whitespace
          }

        if ($input eq '')  # do next
iteration if the rest of $input is empty
          {
            next;
          }

        my $result =
evaluate_perl($input);
        print $input, " = ", $result,
"\n";
      }
  }

sub evaluate_perl
{
   $_ = shift;  # $_ is a string, or
arithmetic expression

#print "----- $_ -----\n";

   # note: n! is not understood by Perl.
We use regular expressions to
   #       translate many cases of exp!
to factorial(exp).
   #       We can thus handle n!, (n-r)!
and  ( (n+r)! + m! ), but not ((n)!)!
   #       or ((n+m)!)!

   # replace n! with factorial(n);
   s/
      (\d+)!            # match and
capture integer!
   /factorial($1)/gx;

   # replace (exp)! with factorial(exp),
where exp is like "num1 arith_op num2"
   s/
      (\([\d\+\-\*\/\s]+\))!   # match
and capture (integer arith_operator
integer)!
   /factorial($1)/x;           # replace
$1 with factorial($1)

   s/ln/log/g;      # allow the use ln
for the natural log

# questions: why does eval return NULL on
"(2 * (3+2))!" instead of returning
#            itself as a string?

#print "----- $_ -----\n";

   my $result = eval;

   if ($show_commas and length($result) >
4)   # put commas in $result
     {
        $result = commify($result);
     }
```

```
   $show_commas = '';    # turn off commas

   return $result;  # note: $result is
local to this sunroutine
}

sub factorial
{
#print "in factorial\n";
   my $n = shift;
   if ($n <= 0)
     {
        return 1;
     }
   my $ans = 1;
   for (my $i = 1; $i <= $n; $i++)
     {
       $ans = $ans*$i;
     }
   return $ans;
}

#binomial coefficient   c(n,r)
sub c
{
#print "in c\n";
   my $n = shift;
   my $r = shift;
   if ($r > $n or $r < 0)
     {
       return undefined;
     }
   else
     {
       return factorial($n) / (
factorial($r) * factorial($n-$r) );
     }
}

# base 2 log
sub lg
{
my $n = shift;
return log($n)/log(2);    #  log = base e
log
}

# base 10 log
sub Log
{
my $n = shift;
return log($n)/log(10);    #  log = base e
log
}

sub commify
{
my $input = shift;
$input = reverse $input;
$input =~
s/(\d\d\d)(?=\d)(?!\d*\.)/$1,/g;
$input = reverse $input;
return $input;
}


sub print_help
{
print <<DONE

----------------------------------------
----------------------------------------
interactively evaluate perl expressions

   quit, exit, q or e will all exit

arithemetic expressions can contain
   perl's usual arithmetic operators:   +
-  *  /  %  and  **
   perl's builtin arithmetic functions:
exp()  sin()  cos()  log()  abs()  and
sqrt()
   as well as the following 9 functions:
      the factorial functions:  !,
factorial
      the log functions:  lg (base 2),
Log (base 10) and ln (base e -- synonym
for log)
      the binomial cooefficient function:
c(n,i)

notes:
  1. We don't completely parse
expressions, so that some expressions
cannot be
     corectly evaluated: e.g.

         (exp)!, when exp is an
expression that contains parenthesis,
        however factorial(exp) works for
all arithmetic expressions.

  2. Both arithmetic and numeric string
results are displayed with separating
commas,
     when either "c" or "commas" is
appended to any input expression (with or
without
     separating whitespace).
     "c" or "commas" should not be
appended to string expressions as
meaningless output
     may be produced.
```

```
  3. Given any arithmetic or string
function f(), this script will evaluate
expressions
     containing f whenever a subroutine,
returning the value of f() is included in
the
     script. This is the way factorial(n)
and c(n,m) are implemented.
----------------------------------------
----------------------------------------

DONE
}
```

==========

my_transform.pl

==========

```
#!/usr/bin/perl -w

# displays the result of applying the
substitution operation of the 2nd
# argument on the 1st argument


our $ARGC = @ARGV ;

if ( $ARGC < 2)
  {
     print "\n",'usage:  $0  "string"
"substitution_exp"',"\n\n";
     print "               where
substitution_exp is of the form
\n";
     print "
s/regex1/regex2/modifier
\n";
     print "               place quotes
around the two arguments          \n";
     print "                 - double
quotes always work for the 2nd arg  \n";
     print "                 - double
quotes usually work for the 1st arg \n";
     print "                   (unless it
contains a \$ sign -- in which  \n";
     print "                   case
single quotes must be used and the
\n";
     print "                   \$ sign
must be escaped                 \n\n";
     exit;
  }

$_ = $ARGV[0];          # string
$sub = $ARGV[1];        # substition
expression

# construct the following:
$_=string;substition_expression;$_
# and then pass it to eval

$exp='$_="'.$ARGV[0].'"';$sub;'$_';
#
#print "$exp    [debug]\n";

$n=eval $exp;

print "$n\n";
```

==========

refresh1.sql

==========

```
select 'Stream 0 Query RF1 begin - ',
LOCALTIMESTAMP(3);
select 'Stream 0 Query RF1 START - ',
date_part('epoch',LOCALTIMESTAMP(3)),
localtimestamp;

begin;
copy orders from 'local' with parallel
subdir 'orders_u1' delimiter '|';
copy lineitem from 'local' with parallel
subdir 'lineitem_u1' delimiter '|';
commit;



select 'Stream 0 Query RF1 STOP - ',
date_part('epoch',LOCALTIMESTAMP(3)),
localtimestamp;
```

==========

refresh2.sql

==========

```
select 'Stream 0 Query RF2 START - ',
date_part('epoch',LOCALTIMESTAMP(3)),
localtimestamp;

begin;
copy delete_ids from 'local' with
parallel subdir 'delete1' delimiter '|';
delete from lineitem where l_orderkey in
(select d_orderkey from delete_ids);
delete from orders where o_orderkey in
(select d_orderkey from delete_ids);
truncate table delete_ids;
commit;
select 'Stream 0 Query RF2 STOP - ',
date_part('epoch',LOCALTIMESTAMP(3)),
localtimestamp;
select 'Stream 0 Query RF2 complete - ',
LOCALTIMESTAMP(3);
```

==========

round_to_tenths.pl

==========

```
#!/usr/bin/perl -w

# used for tpch; spec requires most
numbers to be rounded to the nearest
tenth

our $ARGC = @ARGV ;

if ( $ARGC < 1)
  {
     print "\nusage:  $0  number  \n\n";
     print "               displays
number rounded to the nearest tenth
\n\n";
     exit
  }

$_=$ARGV[0];
$_ =~ s/,//g;
printf "%.1f\n", $_;
```

==========

s2dhms.bash

==========

```
#!/bin/bash


if (( $# < 1 ))
then
  echo ""
  echo "usage:  $0  time_in_secs"
  echo ""
  echo '           - converts "secs" to
"days:hrs:min:secs'
  echo " "
  exit
fi

ts=$1  # time in secs

td=`my_calc.pl "int ($ts/86400)"`
remsecs=`my_calc.pl $ts%86400`

th=`my_calc.pl "int $remsecs/3600"`
remsecs=`my_calc.pl $remsecs%3600`

tm=`my_calc.pl "int ($remsecs/60)"`
remsecs=`my_calc.pl "$remsecs%60"`

ts=$remsecs

if [[ $td > 1 ]]
then
  echo -e "${td}days\c"
  colon=:
elif [[ $td = 1 ]]
then
  echo -e "${td}day\c"
  colon=:
else
  colon=
fi

if [[ $th > 1 ]]
then
  echo -e "$colon${th}hrs\c"
  colon=:
elif [[ $th = 1 ]]
then
  echo -e "$colon${th}hr\c"
  colon=:
else
  colon=
fi

if [[ $tm > 1 ]]
then
  echo -e "$colon${tm}mins\c"
  colon=:
elif [[ $tm = 1 ]]
then
  echo -e "$colon${tm}min\c"
  colon=:
else
  colon=
fi

if [[ $ts > 1 ]]
then
  echo -e "$colon${ts}secs"
elif [[ $lts = 1 ]]
then
  echo -e "$colon${ts}sec"
else
  echo ""    # writes out newline
fi
```

==========

secs_from_epoch.pl

==========

```
#!/usr/bin/perl -w

# computes the number of seconds from
epoch for a given timestamp in the form
#    yyyy-mm-dd hh:mm:ss
#
# the SybaseIQ TPC-H kit uses this
timestamp format

use Time::Local;

our $ARGC = @ARGV ;

if ( $ARGC < 1 )
  {
    print "\nusage:  $0  time_stamp \n";
```

```
    print " \n";
    print "            time_stamp is in
the form:  yyyy-mm-dd hh:mm:ss  \n";
    print "                put time_stamp
in single or double quotes  \n\n";
    exit;
  }

# $ARGV[0] is the first actual argument,
not the script name as in C or the shell
$_=$ARGV[0];

my $arg_length = length($_);
if ($arg_length == 0)
{
   print "Input invalid\n";
   exit;
}

# print "$_     [debug1
secs_from_epoch.pl]\n";
s/-/:/g;
s/ /:/g;
# print "$_     [debug2
secs_from_epoch.pl]\n";

@fields=split(/:/,$_);   # separates $_
using : as delimiter

# print "0 = $fields[0]   [debug
secs_from_epoch.pl\n";
# print "1 = $fields[1]   [debug
secs_from_epoch.pl\n";
# print "2 = $fields[2]   [debug
secs_from_epoch.pl\n";
# print "3 = $fields[3]   [debug
secs_from_epoch.pl\n";
# print "4 = $fields[4]   [debug
secs_from_epoch.pl\n";
# print "5 = $fields[5]   [debug
secs_from_epoch.pl\n";

my $yr   = $fields[0]-1900;    # epoch
begins at 1900
my $mon  = $fields[1]-1;       # months
go from 0-11
my $day  = $fields[2];         # days go
from 1-31
my $hr   = $fields[3];         # hours go
from 0-23
my $min  = $fields[4];         # mins  go
from 0-59
my $sec  = $fields[5];         # mins  go
from 0-59

# print "$yr,$mon,$day,$hr,$min,$sec
[debug3   secs_from_epoch.pl\n";
$time =
timelocal($sec,$min,$hr,$day,$mon,$yr);
print "$time\n";
```

==========

show_machine

==========

```
#!/bin/bash

# March 2006: used psrinfo and prtconf,
when prtdiag is not properly implemented
# April 2006: added online versus total
processors

# shows hostname, ipaddress, num
processors, processor type and memory
size
# for Solaris, Linux and CYGWIN systems

# won't run on Solaris unless picld is
running

os=$(uname)
host=$(hostname)

if [[ $os = SunOS ]]
then

   hipa=$(arp `uname -n` | cut -d ' ' -
f1,2)    # hostname (ip_address)

   op=`psrinfo | grep on-line | wc -l`
# processor name begins with US
   tp=`psrinfo | wc -l`

   ((online_processors=op))
   ((total_processors=tp))

   speed=$(psrinfo -v | sort | uniq |
grep MHz | tr -s ' ' | cut -d ' ' -f 7,8
| tr -d ',')

   Ultra=no
   prtdiag > /dev/null 2>&1
   rc=$?
   if (( rc==0 ))    # prtdiag displays
processor type
   then
     nup=`prtdiag | grep Ultra | wc -l`
# processor name begins with Ultra
     if ((nup>0))
     then
        Ultra=yes
     fi

      if [[ $Ultra = no ]]   # look for
US
      then
          for i in $(prtdiag | grep US |
head -1)
```

```
          do
              if [[ $i = US* ]]
              then
                 ptype=$i
              fi
          done
      else
          for i in $(prtdiag | grep Ultra
| head -1 | tr ','   ' ')
          do
              if [[ $i = Ultra* ]]
              then
                 ptype=$i
              fi
          done
      fi
   else           # get generic processor
type from psrinfo
      ptype=$(psrinfo -v | egrep
"sparc|386" | head -1 | awk '{ print $2
}')
   fi


   speed_ptype="$speed $ptype"

   mem=$(prtconf | grep 'Mem' | cut -d' '
-f 3,4)


elif [[ $os = Linux ]]
then

   hipa=$(arp `uname -n` | cut -d ' ' -
f1,2)

   online_processors=$(cat /proc/cpuinfo
| grep 'model name' | wc -l)
   total_processors=$online_processors

   st=$(cat /proc/cpuinfo | grep 'model
name' | uniq | cut -d ':' -f 2)
   st2=${st//(R)/}      # remove all
occurances of (R) in $st
   st3=${st2//(TM)/}    # remove all
occurances of (TM) in $st2
   st4=${st3# Intel }   # remove " Intel
" from beginning of $st3
   speed_ptype=${st4/ CPU /, }    #
change " CPU " in $st4 to ", "


   #if [[ $speed_ptype = *Xeon* ||
$speed_ptype = XEON* ]] # if Xeon or XEON
divide by 2
   #then
# since all Xeons are hyperthreaded
   #
((online_processors=online_processors/2))
# and hyperthreaded processors are
   #fi
# regarded as 2 processors

   m1=$(cat /proc/meminfo | grep
MemTotal|tr -s ' ' | cut -d' ' -f 2)
   ((m2=m1/1024))
   mem="$m2 Megabytes"

elif [[ $os = CYGWIN* ]]
then

   ipa=$(ipconfig | grep 'IP Address' |
grep -v Auto | \
       my_transform.pl "232T . . . . . .
: 123.34.56.43" "s/\. //g" | cut -d ' ' -
f3)

   hipa="$host ($ipa)"

   online_processors=$(cat /proc/cpuinfo
| grep 'cpu count' | wc -l)    # change
linux to this
   total_processors=$online_processors

   st=$(cat /proc/cpuinfo | grep 'model
name' | uniq | tr -s ' ' | cut -d ' ' -f
4,5,6,7,8,9)
   st2=${st//(R)/}      # remove all
occurances of (R) in $st
   st3=${st2#Intel }    # remove "Intel "
from beginning of $st
   speed_ptype=${st3/ CPU /,   }    #
change " CPU   " in $st3 to ", "

   m1=$(cat /proc/meminfo | grep
MemTotal|tr -s ' ' | cut -d' ' -f 2)
   ((m2=m1/1024))
   mem="$m2 Megabytes"

else
   echo $host is a $os system, no further
information at this time
   exit
fi

result="$hipa:   $online_processors X
$speed_ptype,   $mem"
((res_len=${#result}))

echo " "
((i=0))    # print a line consisting of
res_len dashes
echo "  "
while ((i<res_len + 15))
do
   printf "%s" "-"
```

```
   ((i=i+1))
done
# now print a newline
echo "  "

#printf "%s" 1 $result
echo "   $hipa:   $online_processors(of
$total_processors) X $speed_ptype,
$mem"

((i=0))    # print a line consisting of
res_len dashes
while ((i<res_len + 15))
do
   printf "%s" "-"
   ((i=i+1))
done
# now print a newline
echo " "
#end with a blank line
echo " "
```

show_refresh_start_stop_times

```
#!/bin/bash

# displays list of refresh start and stop
times from timestamps in
update_throughput.out
# this works as long as no refresh takes
more than 24 hours

if (( $# < 1 ))
then
     echo
     echo  "usage:  num_streams"
     exit
     echo
     exit
fi

ns=$1   # num streams

start_date=$(egrep "updates b"
update_throughput.out |
             cut -d '|' -f2 | sed
"s/\(..\:..\:..\....\)//")

end_date=$(egrep "updates c"
update_throughput.out |
             cut -d '|' -f2 | sed
"s/\(..\:..\:..\....\)//")

echo "Refresh Streams - Start and End
Times"
echo "-----------------------------------
--"
echo

echo "Stream ID   Refresh ID  Duration"
echo "-----------------------------------
----------"
((i=1))
while ((i<=ns))
do
   for j in 1 2
   do
     start_time=$(egrep "START"
update_throughput.out |
     grep RF$j |
     grep "Stream $i" |
     cut -d '|' -f2 |
     sed "s/ //g")
     end_time=$(egrep "STOP"
update_throughput.out |
     grep RF$j |
     grep "Stream $i" |
     cut -d '|' -f2 |
     sed "s/ //g")
     calc_args="$end_time-$start_time"
     elapsed_time=$(my_calc.pl
$calc_args)
     elapsed_time=$(round_to_tenths.pl
$elapsed_time)
     echo "stream0$i    RF$j
$elapsed_time secs"
   done
((i=i+1))
done
echo "-----------------------------------
----------"
last_date=$start_date

echo
```

start_throughput_interval.bash

```
#!/bin/bash

if (( $# < 1 ))
then
  echo
  echo "usage:  $0  num_streams  "
  echo
  echo "           - returns starting
time of the first started throughput
stream"
  echo
  exit
fi

ns=$1
```

```
for i in stream[1-9]*.out;   # gets every
streamN.out except stream0.out
do
    num_plus_out=${i#stream}
    num=${num_plus_out%.out}
    #echo $num
    if ((num<=ns))
    then
         string=$string" "$i
    fi
done
# $string is "stream1.out stream2.out ...
stream$ns.out"


grep 'Stream begin' $string          |
cut -d'|' -f 2 | sed "s/^ //" | \

sort > /tmp/ascnd_times

pre_max=`head -1 /tmp/ascnd_times`     #
get the highest time

pre_max=${pre_max%%.*}   # remove the
trailing .ddd from $pre_max
max=${pre_max#\'}     # remove the
leading ' from $pre_max

echo $max
```

==========

storage_ratio_new.bash

==========

```
#!/bin/bash

if (( $# < 2 ))
then
  echo "usage:  $0   scale_factor
num_disks  disk_size(GB)   "
  echo " "
  echo "       - assumes all disks are
the same size"
  echo " "
  exit
fi

sf=$1
ds=$2
nd=$3

((ts=nd*ds))

ratio=`my_calc.pl
"($ts*1000000000)/(${sf}*1024**3)"`

echo $ratio
```

==========

time_wait.sql

==========

```
create or replace function sleep
(integer) returns time as '
declare
  seconds alias for $1;
  later time;
  thetime time;
begin
  thetime := timeofday()::timestamp;
  later := thetime + (seconds::text || ''
seconds'')::interval;
  loop
  if thetime >= later then
    exit;
  else
    thetime := timeofday()::timestamp;
  end if;
end loop;
return later;
end;
' language plpgsql;
```

==========

timing_intervals.bash

==========

```
#!/bin/bash

#   to do:  RF1 and RF2

if (( $# < 1 ))
then
  echo "usage:  $0  num_streams  "
  echo ""
  echo "           - shows timing
intervals as required by the FDR   "
  exit
fi

ns=$1

for qnum in 1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20 21 22
do
   echo " "
   echo "Query $qnum Response Times"
   echo "-----------------------"

   grep LENGTH stream[0-5].out|grep
"Query $qnum "| grep -v cast|cut -d' ' -
f1,2,3,4,7,8| \

cut -d' ' -f5 | tr  "'" " "|tr , " "
>/tmp/full

   echo  "stream 0  `head -1 /tmp/full`"
   echo "-----------------------"
```

```
   # we will now discard the first
(stream0) row of /tmp/full
   # and print the remaining rows
together with the min, avg and max
   tail -$ns /tmp/full  >/tmp/qrt

   sum=
   ((k=1))
   for j in `cat /tmp/qrt`
   do
      echo  "stream $k   $j"
      sum="$sum+$j"
      ((k=k+1))
   done

   sort -n /tmp/qrt >  /tmp/sorted
   echo "------------------------"
   echo "minimum  `head -1
/tmp/sorted`"
   echo "------------------------"

   s="($sum)/$ns"
   #echo s=$s   [trace
throughput_response_times.bash]
   avg=`my_calc.pl "$s"`
   echo "average    $avg"
   echo "------------------------"

   echo "maximum  `tail -1
/tmp/sorted`"
   echo "------------------------"
   echo " "
   echo " "

done
```

==========

tpch.sh

==========

```
#$SHELL

source $HOME/.bashrc

# Set env var's needed for ntest
export xenv="xen-TPCH9-1"
nq=22

cur_dir=$(pwd)
if [[ $cur_dir != $HOME/run/scripts ]]
then
      echo
      echo ERROR: do_test must be run
from $HOME/run/scripts
      echo
      echo "       the current dir is
$cur_dir "
      echo
      exit
fi

audit_file_dir=`cat /tmp/this_tpch`/

if (( $# < 1 ))
then
   echo
   echo "usage: $0    scope   (plus
additional args depending upon scope)
"
   echo
   echo "   scope values:
"
   echo
   echo "   - load            create and
load a tpch database from scratch
"
   echo "   - qi        do query i
(i=1,2,...,22)
"
   echo "   - stream0   do all 22 queries
without any refreshes
"
   echo "   - refresh1  do refresh 1
"
   echo "   - refresh2  do refresh 2"
   echo "   - thrustreams     do a single
thrurefresh run without any refreshes
"
   echo "   - thrurefresh     do all the
throughput refreshes without query
streams"
   echo "   - all       do full audit
run: 2 power-thrurefresh cycles:"
   echo "   run1 <onerefresh 1 + stream0
+ onerefresh 2 + thrustreans +
thrurefresh> + run2<>"
   echo
   exit
fi

scope=$1

case $scope in
          load)  if (( $# < 3  || $# > 3
))
                 then
                       echo
                       echo "usage: $0
load     scale_factor
num_query_streams"
                       echo
                       exit
                 fi
             seed=1
             sf=$2
             nqs=$3;;
```

```
            q*)  if (( $# < 2 || $# > 2
))
                 then
                       echo
                       echo "usage: $0
qi   scale_factor
"
                       echo
                       echo "     note:
database must be running; if not restart
it first"
                       echo
                       exit
                 fi

                 query_num=${scope:1}
                 case $query_num in
                    [1-9])   ;;  # q1  to
q9
                   1[0-9])   ;;  # q10 to
q19
                   2[0-2])   ;;  # q20 to
q22
                        *)  echo
                            echo "ERROR:
query number ($query_num) must be between
1 and 22"
                            echo
                            exit
                 esac

                 sf=$2  ;;

       stream0)  if (( $# < 3 || $# > 3
))
                 then
                       echo
                       echo "usage: $0
stream0    scale_factor  input_seed"
                       echo
                       exit
                 fi

                 sf=$2
                 let input_seed=$3   ;;

       refresh1)  if (( $# < 2 || $# > 2
))
                 then
                       echo
                       echo "usage: $0
refresh1    scale_factor  "
                       echo
                       exit
                 fi

                 sf=$2 ;;

       refresh2)  if (( $# < 2 || $# > 2
))
                 then
                       echo
                       echo "usage: $0
refresh2    scale_factor  "
                       echo
                       exit
                 fi

                 sf=$2 ;;


    thrustreams)  if (( $# < 4 || $# > 4
))
                 then
                       echo
                       echo "usage: $0
thrustreams    scale_factor  "
                       echo "
num_query_streams  input_seed "
                       echo
                       exit
                 fi

                 sf=$2
                 nqs=$3
                 let input_seed=$4   ;;

    thrurefresh)  if (( $# < 4 || $# > 4
))
                 then
                       echo
                       echo "usage: $0
thrurefresh    scale_factor  "
                       echo "
num_query_streams  input_seed "
                       echo
                       exit
                 fi

                 sf=$2
                 nqs=$3
                 let input_seed=$4   ;;


           all)  if (( $# < 6 || $# > 6
))
                 then
                       echo
                       echo "usage: $0
all    scale_factor  "
                       echo "
num_query_streams
"
                       echo "
num_disks  disk_size(in GB)  system_cost
"
```

```
                        echo
                        exit
                        echo
                fi

                sf=$2
                nqs=$3
                let nd=$4
                let ds=$5
                let sc=$6
                input_seed=$(grep "as a
seed" stream0.sql | cut -d 'g' -f2 | cut
-d 'a' -f1) ;;
             *)  echo
                 echo "ERROR: scope
(=$scope) must be one of:
"
                 echo
                 echo "
load,q1,q2,...,q22,stream0,refresh1,refre
sh2,"
                 echo "
thrustreams,thrurefresh,lthrurefresh,all
"
                 echo
                 echo
                 exit
esac

# we impose the minimum stream
requirement only in the "all" case
# for all other scopes the min number of
query streams is 1
# aminqs is thus the "allowed minimum"
for the current run

aminqs=1
#
# For testing purposes, we aren't
enforcing this, though.
#
# if [[ ($scope = all ) && -z $real_scope
&& -z $real_scope2 ]]
# then
#    case $sf in
#          .1)  aminqs=1
#               ;;
#           1)  aminqs=2
#               ;;
#          10)  aminqs=3
#               ;;
#          30)  aminqs=4
#               ;;
#         100)  aminqs=5
#               ;;
#         300)  aminqs=6
#               ;;
#        1000)  aminqs=7
#               ;;
#        3000)  aminqs=8
#               ;;
#       10000)  aminqs=9
#               ;;
#       30000)  aminqs=10
#               ;;
#          * )
#               echo
#               echo "ERROR:  scale
factor (=$sf) must be one of
(.1,1,10,30,100,300,1000,3000,10000,30000
)"
#               echo
#               exit
#               echo;;
#    esac
# fi

# These are the minimum number of streams
for a compliant run

case $sf in
       1)  cmin=2
           ;;
      10)  cmin=3
           ;;
      30)  cmin=4
           ;;
     100)  cmin=5
           ;;
     300)  cmin=6
           ;;
    1000)  cmin=7
           ;;
    3000)  cmin=8
           ;;
   10000)  cmin=9
           ;;
   30000)  cmin=10
           ;;
       *)  cmin=1
           ;;
esac


PLATFORM=`uname -m`

# make sure input_seed value is legal
(i.e. >= 0 ) for the scopes where it is
been
# supplied as a commandline arg ("lrun1"
and "lthrurefresh" are covered by "all"

if [[ $scope = stream0    || $scope =
power || $scope = thrustreams ||  \
      $scope = thrurefresh || $scope =
run1  || $scope = all           ]]
then
      if [[ $input_seed -lt 0 ]]
```

```
      then
           echo
           echo "ERROR:
input_seed($input_seed) must be >= 0"
           echo
           exit
      fi
fi


# check that the scale factor is legal
where it has been supplied as
# a command line arg

      if [[ $sf != .1 && $sf != 1 && $sf
!= 10 && $sf != 30 && \
            $sf != 100 && $sf != 300 &&
$sf != 1000 && $sf != 3000 && $sf !=
10000 && $sf != 30000 ]]
      then
           echo
           echo "ERROR: scale factor
(=$sf) must be one of
[.1,1,10,30,100,300,1000,3000,10000,30000
]"
           echo
           exit
      fi


if [[ $scope = thrurefresh || $scope =
all  ]]
then
   # generate update_throughputX.sql
   i=2
   run=1
   q="'"
   ((max_streams=nqs + 1))
   while((run<=2))
   do

upd_tput_fname="update_throughput_run_${r
un}_${nqs}.sqlx"
      cat update_throughput_header >
$upd_tput_fname
      stream_number=1
      while ((i<=max_streams))
      do
         echo "select 'Stream
${stream_number} RF1 START - ',
date_part('epoch',LOCALTIMESTAMP(3)),
localtimestamp;" >> $upd_tput_fname
       echo "begin;" >> $upd_tput_fname
       echo "copy orders from 'local'
with parallel subdir ${q}orders_u${i}${q}
delimiter '|';" >> $upd_tput_fname
       echo "copy lineitem from 'local'
with parallel subdir
${q}lineitem_u${i}${q} delimiter '|';" >>
$upd_tput_fname
       echo "commit;" >> $upd_tput_fname
         echo "select 'Stream
${stream_number} RF1 STOP - ',
date_part('epoch',LOCALTIMESTAMP(3)),
localtimestamp;" >> $upd_tput_fname

         echo "select 'Stream
${stream_number} RF2 START - ',
date_part('epoch',LOCALTIMESTAMP(3)),
localtimestamp;" >> $upd_tput_fname
       echo "begin;" >> $upd_tput_fname
       echo "copy delete_ids from 'local'
with parallel subdir ${q}delete${i}${q}
delimiter '|';" >> $upd_tput_fname
       echo "delete from lineitem where
l_orderkey in (select d_orderkey from
delete_ids);" >> $upd_tput_fname
       echo "delete from orders where
o_orderkey in (select d_orderkey from
delete_ids);" >> $upd_tput_fname
       echo "truncate table delete_ids;"
>> $upd_tput_fname
       echo "commit;" >> $upd_tput_fname
         echo "select 'Stream
${stream_number} RF2 STOP - ',
date_part('epoch',LOCALTIMESTAMP(3)),
localtimestamp;" >> $upd_tput_fname
        ((i=i+1))

((stream_number=stream_number+1))
      done
      cat update_throughput_footer >>
$upd_tput_fname
      ((i=nqs + 2))
      ((max_streams=max_streams + nqs))
      ((run=run + 1))
   done
fi

if [[ $scope = thrurefresh || $scope =
run1 || $scope = all || $scope = mall ]]
then
      run=1
      while((run<=2))
      do
         if [[ -f
update_throughput_run_${run}_${nqs}.sqlx
]]
         then
              echo
         else
              echo
              echo "ERROR:
update_throughput_run_${run}_${nqs}.sqlx
must exist for a run of $nqs streams"
              echo
              exit
```

```
      fi
      ((run=run + 1))
      done
fi

# make sure that the number of query
streams >= minimum allowed num streams
# aminqs has been previously set to 1
except when $scope is "all" in which
# case aminqs is the complient minimun

if [[ $scope = all ]]
then
      if ((nqs < aminqs))    # nqs:
requested number of query streams
      then
            echo
            echo "ERROR: requested query
streams (=$nqs) must be >= $aminqs for
sf=$sf & scope=$scope"
            echo
            exit
      fi
fi

# we have checked the commandline args
for reasonableness, now
# describe the scope of the run and its
configuration

echo " "
case $scope in
           q* )
                 echo "DOING QUERY
$query_num  with: " ;;
          all )
                 if [[ $real_scope =
lrun1 ]]
                 then
                        echo "DOING LOAD,
FOLLOWED BY POWER, FOLLOWED BY THROUGHPUT
with:"
                 else
                        echo "DOING FULL
AUDIT TEST (load, plus 2 full runs) with:
"
                 fi ;;
         load )
                 echo "ONLY DOING LOAD
(and create database) with: " ;;
      refresh1 )
                 echo "ONLY DOING ONE
REFRESH 1 with: " ;;
      refresh2 )
                 echo "ONLY DOING ONE
REFRESH 2 with: " ;;
   thrustreams )
                 echo "ONLY DOING ONE
THROUGHPUT TEST WITHOUT REFRESHES with:"
;;
   thrurefresh )
                 echo "ONLY DOING ONE
THROUGHPUT TEST with:" ;;
      stream0 )
                 echo "ONLY DOING 22
SINGLE-USER QUERIES (NO REFRESHES):" ;;
           * )
                 echo "ERROR:  scope
(=$scope) must be one of"
                 echo "
(all,load,power,refresh,stream0,thrurefre
sh,run1)"
                 echo
                 exit ;;
esac

echo " "



      echo "      scale factor = $sf "

if [[ $scope = all || $scope =
thrurefresh || $scope = thrustreams ]]
then
   echo "      num thrurefresh streams =
$nqs    (compliant minimum for this scale
factor: $cmin)"
fi


if [[ $scope != load && $scope !=
refresh1 && $scope != refresh2 ]]
then
   if [[ $input_seed == 1 ]]
   then
         seed=$input_seed
         echo "      using newly
gererated seed"
   elif  [[ $input_seed == 0 ]]
   then
         if [[ -f stream0.sql ]]
         then
              existing_seed=$(grep "as a
seed" stream0.sql | cut -d 'g' -f2 | cut
-d 'a' -f1)
              seed=$existing_seed
              echo  "      using existing
seed = $seed"
         else
              echo "ERROR: Cannot use
existing seed when stream0.sql does not
exist"
              exit
         fi
   else
```

```
      echo "      using $input_seed as
seed"
      seed=$input_seed
   fi
fi


echo "      Using the following nodes"
echo "      --------------------------"
cqi show | tee node_config.out

# echo -e "Are these OK? (type y or n)
\c"
# read ans
#
#
# if [[ $ans = n || $ans = no ]]
# then
#      echo
#      echo change one or more of
#      echo "  options.sql, tpch.cfg,
create_database.sql, create_tables or
/etc/system"
#      echo and try again
#      echo
#      exit
# fi

# -------------  start real execution ---
-------------------------------

power=0           # used in run1 to
determine if Composite should be
calculated
throughput=0      # this might not be
necessary

if [[ $scope = q* ]]
then
     # assumes database is running
     one_query $query_num $sf
     echo
     exit
fi

echo
echo

if [[ $scope = load  ]]
then

     echo " "
     echo " "
     echo " "

     echo "Creating tpch database:
`date`..."
     psql dev -c "drop database tpch"
     $HOME/padb/rel/bin/createdb tpch
     echo "  "

     echo "Installing sleep stored
procedure"
     psql -d tpch -f time_wait.sql
     echo "  "

     echo "Performing Pre-Test clean-up."
     sh clean.sh
     echo "  "

     echo "Creating TPCH schema"
     psql -d tpch -f create_tables.sql
     echo "  "

     #
     # Load the database
     #

     starttime=`date "+%Y-%m-%d
%H:%M:%S"`
     echo "    Load started  $starttime "
> start_load.out
     echo "Load region `date`"
     psql tpch -c "copy region from
'copied' with parallel delimiter '|'"
     echo "Load nation `date`"
     psql tpch -c "copy nation from
'copied' with parallel delimiter '|'"
     echo "Load customer `date`"
     psql tpch -c "copy customer from
'copied' with parallel delimiter '|'"
     echo "Load lineitem `date`"
     psql tpch -c "copy lineitem from
'copied' with parallel delimiter '|'"
     echo "Load orders `date`"
     psql tpch -c "copy orders from
'copied' with parallel delimiter '|'"
     echo "Load partsupp `date`"
     psql tpch -c "copy partsupp from
'copied' with parallel delimiter '|'"
     echo "Load part `date`"
     psql tpch -c "copy part from
'copied' with parallel delimiter '|'"
     echo "Load supplier `date`"
     psql tpch -c "copy supplier from
'copied' with parallel delimiter '|'"
#      echo "Defrag storage `date`"
#     psql tpch -c "xpx 'defrag'"
     echo "Build statistics `date`"
     psql -d tpch -a -c 'analyze' >>
analyze.out
     psql -d tpch -a -c 'set
random_stat_sample to on; analyze
partsupp(ps_partkey,ps_suppkey); set
random_stat_sample to off;'
     echo "Load completed `date`"
     endyear=`date '+%Y'`
     enddate=`date '+%m%d'`
```

```
     endtime=`date '+%H%M%S'`
     endstamp=`date "+%Y-%m-%d %H:%M:%S"`
     echo "    Load Finished  $endstamp"
> end_load.out
     echo "    Seed time is:
$endyeaar$enddate$endtime"
     echo
     slt=`tr -s ' ' <start_load.out | cut
-d' ' -f 4,5 `
     elt=`tr -s ' ' <end_load.out | cut -
d' ' -f 4,5 `
     lt=`calculate_load_time.bash "$elt"
"$slt" `  # calls secs_from_epoch.pl
     echo "Database Load Time:  $lt"
     echo " "
     if ((seed==1))
     then
           echo "Generating $((nqs+1))
Query Streams "
           echo "Using the appropriate
timestamp seed = $enddate$endtime"
           gen_streams.sh $enddate
$endtime $sf $nqs
     else
        if [[ $input_seed == 0 ]]
        then
           echo  "Using existing seed =
$seed ###################"
        else
           echo  "Using provided seed =
$seed ###################"
           rm -f stream*.sql
           gen_streams.sh 0 $seed $sf
$nqs
        fi
     fi
     echo
     if [[ $scope = load ]]
     then
          echo exiting $0
          echo
          exit
     fi
fi

#Load completed

if [[ $scope = all ]]
then
     echo "Starting Audit Verification
Scripts   `date` "
     echo

     # After the load Completes run the
Audit SQL

     psql -d tpch -f dbtables-xen.sql >
rdbtablest_start.out
     echo
     echo "Finished Audit Start
Verification Scripts   `date` "
     echo
fi


if [[ ( $scope = refresh1 || $scope =
refresh2 || $scope = stream0 || $scope =
all ) ]]
then
     if [[  $scope = all ]]
     then
#          psql tpch -c "xpx
'optimize_reads y'" >/dev/null
          echo
          echo "Start Run 1 Power Test
`date` "
          echo "----------------------
"
     elif [[ $scope = stream0 ]]
     then
          echo
          echo "Start Stream0   `date` "
          echo "-------------          "
     else
          echo
          echo "Start Refresh   `date` "
          echo "-------------          "
     fi


     if [[ $scope = refresh1 || $scope =
all ]]
     then
          echo
          echo "     Start refresh 1
`date`  "
          psql -d tpch -f refresh1.sql >
rf1.out
          echo
          echo "     End refresh 1
`date`  "
        #create the refresh file
        cat rf1.out  > update_power.out
     fi


     if [[ $scope = stream0  || $scope =
all ]]
     then
           echo
           echo "     Start Query Stream
0   `date` "
           psql -d tpch -f stream0.sql >
stream0.out &

         wait
```

```
          cat s0q01-1.out s0q01-4.out
s0q02-1.out s0q02-4.out s0q03-1.out
s0q03-4.out s0q04-1.out s0q04-4.out
s0q05-1.out s0q05-4.out s0q06-1.out
s0q06-4.out s0q07-1.out s0q07-4.out
s0q08-1.out s0q08-4.out s0q09-1.out
s0q09-4.out s0q10-1.out s0q10-4.out
s0q11-1.out s0q11-4.out s0q12-1.out
s0q12-4.out s0q13-1.out s0q13-4.out
s0q14-1.out s0q14-4.out s0q15-1.out
s0q15-4.out s0q16-1.out s0q16-4.out
s0q17-1.out s0q17-4.out s0q18-1.out
s0q18-4.out s0q19-1.out s0q19-4.out
s0q20-1.out s0q20-4.out s0q21-1.out
s0q21-4.out s0q22-1.out s0q22-4.out >>
stream0.out
          echo
          echo "     Finish Query Stream
0  `date` "
     fi


     if [[ $scope = refresh2 || $scope =
all ]]  # do refresh2
     then
          echo
          echo "     Start refresh 2
`date` "
         psql -d tpch -f refresh2.sql >
rf2.out
          echo
          echo "     End refresh 2
`date`  "
          # append the refresh file
          cat rf2.out >>
update_power.out
     else
          echo
          echo "End Stream0   `date`"
          echo "-----------         "
          echo will now make report
          echo
     fi


     if [[ $scope = refresh2 ]]
     then
          echo
          echo "     End Refresh 2
      `date`    "
          echo "-------------------------
---"
          echo
          tpch_stats.pl power $sf $nqs
none
          echo
          if [[ $scope = refresh2 ]]
          then
           exit   # on refresh
          fi
     else
          echo
          echo "End Run 1 Power Test
       `date` "
          echo "---------------------
"
          echo "   Computing response
times.."
          echo " "
          echo " "
          tpch_stats.pl power $sf "1"
"none"
          power=$(tpch_stats.pl power
$sf 1 power)
          echo " "
          echo " "
          echo "   Done computing
response times"
          echo " "
          echo "       Power = $power"
          echo
          echo
          ps -eaf | grep asiq| grep -v
grep | tr -s ' ' | cut -d ' ' -f8,9
          echo
     fi
fi

if [[ ($scope = thrurefresh || $scope =
all || $scope = thrustreams) ]]
then
     echo
     echo "Start Run 1 Throughput `date`
"
     echo "----------------------
"
     echo "  "
     echo "     Start Query Streams
`date` "
     echo "  "

     ((i=1))
     while ((i<=nqs))  # run all query
streams concurrently
     do
        echo "        Starting stream
${i}"
        psql -d tpch -f stream${i}.sql >
stream${i}.out &
       echo "        Stream ${i} started
        `date`"
        ((i=i+1))
     done

     if [[ $scope != thrustreams ]]  #
thrustreams does a thrurefresh without
refreshes
     then
```

```
          echo "     Start Refresh Stream
for set $nqs  `date` "
          psql -d tpch -f
update_throughput_run_1_${nqs}.sqlx >>
update_throughput.out&
     fi

     wait   # for everything

     ((i=1))
     while ((i<=nqs))
     do
      cat s${i}q01-1.out s${i}q01-4.out
s${i}q02-1.out s${i}q02-4.out s${i}q03-
1.out s${i}q03-4.out s${i}q04-1.out
s${i}q04-4.out s${i}q05-1.out s${i}q05-
4.out s${i}q06-1.out s${i}q06-4.out
s${i}q07-1.out s${i}q07-4.out s${i}q08-
1.out s${i}q08-4.out s${i}q09-1.out
s${i}q09-4.out s${i}q10-1.out s${i}q10-
4.out s${i}q11-1.out s${i}q11-4.out
s${i}q12-1.out s${i}q12-4.out s${i}q13-
1.out s${i}q13-4.out s${i}q14-1.out
s${i}q14-4.out s${i}q15-1.out s${i}q15-
4.out s${i}q16-1.out s${i}q16-4.out
s${i}q17-1.out s${i}q17-4.out s${i}q18-
1.out s${i}q18-4.out s${i}q19-1.out
s${i}q19-4.out s${i}q20-1.out s${i}q20-
4.out s${i}q21-1.out s${i}q21-4.out
s${i}q22-1.out s${i}q22-4.out >>
stream${i}.out
        ((i=i+1))
     done

     echo " "
     echo "     All Query and Refresh
Streams have completed `date` "
     echo " "



throughput_interval=$(tpch_throughput_int
erval.bash  $nqs)
     tpch_throughput_interval.bash $nqs >
throughput_interval.out
     throughput=$(tpch_stats.pl
"thrurefresh"  $sf  $nqs "throughput")
     echo
     echo "     Throughput Interval =
$throughput_interval secs "
     echo
     echo "     Throughput = $throughput"
     echo
     if [[ $power != 0 ]]
     then
          composite=$(my_calc.pl
"sqrt($power*$throughput)")
          echo "     Composite =
$composite"
          echo
     fi
     echo
     #ps -eaf | grep asiq | grep -v grep
| tr -s ' ' |cut -d ' ' -f8,9
     echo
fi

if [[ $scope = stream0 || $scope = power
|| $scope = thrustreams ||
      $scope = thrurefresh || $scope =
all  ]]
then
     dayHr=`date '+%m%d%H'`
     echo
     echo
     if [[ -z $real_scope2 ]]     #
non_null means lthrurefresh
     then
        if [[ -z $real_scope ]]   #
non_null means lrun1
        then
           if [[ $scope = stream0 ]]
           then
              composite=0
           fi
           if [[ $scope = power ]]
           then
              composite=$power
           fi
           if [[ $scope = thrurefresh ||
$scope = all ]]
           then
              composite=$throughput
           fi
           rpt_file_name="run1.log"

#rpt_file_name="mrun1_${scope}_${sf}g_${n
qs}s_${dayHr}_${xenv}_1n_$(round_to_tenth
s.pl ${composite}).r"
           echo "Producing
${rpt_file_name}"
           echo "Running tpch_report
########################################
#####"
           tpch_report.new $scope $sf
\

run1_${scope}_${sf}g_${nqs}s_${dayHr}_${x
env}_1n.r \
                              $seed $nqs
$nd $ds $sc
\
                           >
${rpt_file_name}
           mv ${rpt_file_name}
$audit_file_dir
        else
        echo" "
```

```
        fi
     else
        # lthrurefresh
        composite=$throughput
        rpt_file_name="run1.log"
        echo "Producing ${rpt_file_name}"
        tpch_report.new $scope $sf
\

run1_${scope}_${sf}g_${nqs}s_${dayHr}_${x
env}_1n.r \
                          $seed $nqs $nd
$ds $sc
\
                         >
${rpt_file_name}
           mv ${rpt_file_name}
$audit_file_dir
     fi
     echo
fi


if [[ $scope = all ]]
then
     echo
     ((i=0))
     while ((i<=nqs))  # move the
streamX.out files to audit naming
standard
     do
        ((q=1))
        while ((q<=22))
        do
           if [ $q -lt 10 ]
           then
              qn="0$q"
           else
              qn="$q"
           fi
           cat s${i}q${qn}-?.out >
${audit_file_dir}m1s${i}q${qn}.out
           lc=`wc -l < s${i}q${qn}-3.out`
           if [ $lc -gt 200 ]
           then
              head -102 s${i}q${qn}-3.out
>  s${i}q${qn}-3.trunc
              echo
"**************************************
******" >>  s${i}q${qn}-3.trunc
              tail -102 s${i}q${qn}-3.out
>> s${i}q${qn}-3.trunc
              cat s${i}q${qn}-1.out
s${i}q${qn}-2.out s${i}q${qn}-3.trunc
s${i}q${qn}-4.out >
${audit_file_dir}m1s${i}q${qn}.trunc.out
           fi
        ((q=q+1))
        done
        ((i=i+1))
     done

     mv update_power.out
$audit_file_dir"m1s00rf.out" #change to
move
     mv update_throughput.out
$audit_file_dir"m1s01rf.out" #change to
move
     echo cp
$HOME/padb/rel/etc/xenpostgresql.conf
$audit_file_dir
     cp
$HOME/padb/rel/etc/xenpostgresql.conf
$audit_file_dir

     echo "Moved  *.out files to "
$audit_file_dir
     echo
     echo
     echo "FINISHED Run1      `date`"
     echo
fi
####################
# Run 2
####################
if [[ ( $scope = refresh1 || $scope =
refresh2 || $scope = stream0 || $scope =
all ) ]]
then
     if [[  $scope = all ]]
     then
#       echo
#       echo "Restart Database   `date` "
#       cqi xstop
#       sleep 20
#       cqi xstart
          echo
          echo "Start Run 2 Power Test
`date` "
          echo "----------------------
"
     elif [[ $scope = stream0 ]]
     then
          echo
          echo "Start Stream0   `date` "
          echo "-------------           "
     else
          echo
          echo "Start Refresh   `date` "
          echo "-------------           "
     fi


     if [[ $scope = refresh1 || $scope =
all ]]
     then
          echo
```

```
          echo "     Start refresh 1
`date`  "
          psql -d tpch -f
run_2_refresh1.sql > rf1.out
          echo
          echo "     End refresh 1
`date`  "
        #create the refresh file
        cat rf1.out  > update_power.out
     fi


     if [[ $scope = stream0  || $scope =
all ]]
     then
           echo
           echo "     Start Query Stream
0   `date` "
           psql -d tpch -f stream0.sql >
stream0.out &
         wait

           cat s0q01-1.out s0q01-4.out
s0q02-1.out s0q02-4.out s0q03-1.out
s0q03-4.out s0q04-1.out s0q04-4.out
s0q05-1.out s0q05-4.out s0q06-1.out
s0q06-4.out s0q07-1.out s0q07-4.out
s0q08-1.out s0q08-4.out s0q09-1.out
s0q09-4.out s0q10-1.out s0q10-4.out
s0q11-1.out s0q11-4.out s0q12-1.out
s0q12-4.out s0q13-1.out s0q13-4.out
s0q14-1.out s0q14-4.out s0q15-1.out
s0q15-4.out s0q16-1.out s0q16-4.out
s0q17-1.out s0q17-4.out s0q18-1.out
s0q18-4.out s0q19-1.out s0q19-4.out
s0q20-1.out s0q20-4.out s0q21-1.out
s0q21-4.out s0q22-1.out s0q22-4.out >>
stream0.out
           echo
           echo "     Finish Query Stream
0  `date` "
     fi


     if [[ $scope = refresh2 || $scope =
all ]]  # do refresh2
     then
           echo
           echo "     Start refresh 2
`date` "
         psql -d tpch -f
run_2_refresh2.sql > rf2.out
           echo
           echo "     End refresh 2
`date`  "
           # append the refresh file
           cat rf2.out >>
update_power.out
     else
           echo
           echo "End Stream0   `date`"
           echo "-----------         "
           echo will now make report
           echo
     fi


     if [[ $scope = refresh2 ]]
     then
           echo
           echo "     End Refresh 2
      `date`    "
           echo "-----------------------
---"
           echo
           tpch_stats.pl power $sf $nqs
none
           echo
           if [[ $scope = refresh2 ]]
           then
            exit   # on refresh
           fi
     else
           echo
           echo "End Run 2 Power Test
      `date` "
           echo "---------------------
"
           echo "   Computing response
times.."
           echo " "
           echo " "
           tpch_stats.pl power $sf "1"
"none"
           power=$(tpch_stats.pl power
$sf 1 power)
           echo " "
           echo " "
           echo "   Done computing
response times"
           echo " "
           echo "       Power = $power"
           echo
           echo
           ps -eaf | grep asiq| grep -v
grep | tr -s ' ' | cut -d ' ' -f8,9
           echo
     fi
fi

if [[ ($scope = thrurefresh || $scope =
all || $scope = thrustreams) ]]
then
     echo
     echo "Start Run 2 Throughput `date`
"
     echo "---------------------
"
```

```
     echo "  "
     echo "     Start Query Streams
`date` "
     echo "  "

     ((i=1))
     while ((i<=nqs))  # run all query
streams concurrently
     do
        echo "        Starting stream
${i}"
        psql -d tpch -f stream${i}.sql >
stream${i}.out &
        echo "        Stream ${i} started
      `date`"
        ((i=i+1))
     done

     if [[ $scope != thrustreams ]]  #
thrustreams does a thrurefresh without
refreshes
     then
          echo "     Start Refresh Stream
for set $nqs  `date` "
          psql -d tpch -f
update_throughput_run_2_${nqs}.sqlx >>
update_throughput.out&
     fi

     wait   # for everything

     ((i=1))
     while ((i<=nqs))  # get timings for
each stream
     do
      cat s${i}q01-1.out s${i}q01-4.out
s${i}q02-1.out s${i}q02-4.out s${i}q03-
1.out s${i}q03-4.out s${i}q04-1.out
s${i}q04-4.out s${i}q05-1.out s${i}q05-
4.out s${i}q06-1.out s${i}q06-4.out
s${i}q07-1.out s${i}q07-4.out s${i}q08-
1.out s${i}q08-4.out s${i}q09-1.out
s${i}q09-4.out s${i}q10-1.out s${i}q10-
4.out s${i}q11-1.out s${i}q11-4.out
s${i}q12-1.out s${i}q12-4.out s${i}q13-
1.out s${i}q13-4.out s${i}q14-1.out
s${i}q14-4.out s${i}q15-1.out s${i}q15-
4.out s${i}q16-1.out s${i}q16-4.out
s${i}q17-1.out s${i}q17-4.out s${i}q18-
1.out s${i}q18-4.out s${i}q19-1.out
s${i}q19-4.out s${i}q20-1.out s${i}q20-
4.out s${i}q21-1.out s${i}q21-4.out
s${i}q22-1.out s${i}q22-4.out >>
stream${i}.out
        ((i=i+1))
     done

     echo " "
     echo "     All Query and Refresh
Streams have completed `date` "
     echo " "


throughput_interval=$(tpch_throughput_int
erval.bash  $nqs)
     tpch_throughput_interval.bash $nqs >
throughput_interval.out
     throughput=$(tpch_stats.pl
"thrurefresh"  $sf  $nqs "throughput")
     echo
     echo "     Throughput Interval =
$throughput_interval secs "
     echo
     echo "     Throughput = $throughput"
     echo
     if [[ $power != 0 ]]
     then
          composite=$(my_calc.pl
"sqrt($power*$throughput)")
          echo "     Composite =
$composite"
          echo
     fi
     echo
     #ps -eaf | grep asiq | grep -v grep
| tr -s ' ' |cut -d ' ' -f8,9
     echo
fi

if [[ $scope = stream0 || $scope = power
|| $scope = thrustreams ||
      $scope = thrurefresh || $scope =
all  ]]
then
     dayHr=`date '+%m%d%H'`
     echo
     echo
     if [[ -z $real_scope2 ]]     #
non_null means lthrurefresh
     then
        if [[ -z $real_scope ]]   #
non_null means lrun1
        then
           if [[ $scope = stream0 ]]
           then
              composite=0
           fi
           if [[ $scope = power ]]
           then
              composite=$power
           fi
           if [[ $scope = thrurefresh ||
$scope = all ]]
           then
              composite=$throughput
           fi
           rpt_file_name="run2.log"
```

```
           echo "Producing
${rpt_file_name}"
           echo "Running tpch_report
########################################
#####"
           tpch_report.new $scope $sf
\

run1_${scope}_${sf}g_${nqs}s_${dayHr}_${x
env}_1n.r \
                              $seed $nqs
$nd $ds $sc
\
                          >
${rpt_file_name}
           mv ${rpt_file_name}
$audit_file_dir
        else
          echo" "
        fi
     else
        # lthrurefresh
        composite=$throughput
        rpt_file_name="run2.log"
        echo "Producing ${rpt_file_name}"
        tpch_report.new $scope $sf
\

run1_${scope}_${sf}g_${nqs}s_${dayHr}_${x
env}_1n.r \
                           $seed $nqs $nd
$ds $sc
\
                          >
${rpt_file_name}
           mv ${rpt_file_name}
$audit_file_dir
     fi
     echo
fi


if [[ $scope = all ]]
then
     echo
     ((i=0))
     while ((i<=nqs))  # move the
streamX.out files to audit naming
standard
     do
        ((q=1))
        while ((q<=22))
        do
           if [ $q -lt 10 ]
           then
              qn="0$q"
           else
              qn="$q"
           fi
           cat s${i}q${qn}-?.out >
${audit_file_dir}m2s${i}q${qn}.out
           lc=`wc -l < s${i}q${qn}-3.out`
           if [ $lc -gt 200 ]
           then
              head -102 s${i}q${qn}-3.out
>  s${i}q${qn}-3.trunc
              echo
"**************************************
******" >> s${i}q${qn}-3.trunc
              tail -102 s${i}q${qn}-3.out
>> s${i}q${qn}-3.trunc
              cat s${i}q${qn}-1.out
s${i}q${qn}-2.out s${i}q${qn}-3.trunc
s${i}q${qn}-4.out >
${audit_file_dir}m2s${i}q${qn}.trunc.out
           fi
        ((q=q+1))
        done
        ((i=i+1))
        echo
     done

     mv update_power.out
$audit_file_dir"m2s00rf.out" #change to
move
     mv update_throughput.out
$audit_file_dir"m2s01rf.out" #change to
move
     echo cp
$HOME/padb/rel/etc/xenpostgresql.conf
$audit_file_dir
     cp
$HOME/padb/rel/etc/xenpostgresql.conf
$audit_file_dir

     echo "Moved  *.out files to "
$audit_file_dir
     echo
     echo "FINISHED Run2      `date`"
     echo
fi
exit
```

ACID Test Execution Code

```
====================
acid_atomicity.sh
====================
#!/bin/bash
#
# TPC-H ACID Atomicity Test
#
# Copyright 2007 ParAccel, Inc
#

echo Atomicity Test -- Started at: `date`
echo
```

```
#
########################################
#################
# Run ROLLBACK case

# Clean history table
echo Atomicity Test -- Clearing History
Table
psql -q -t tp_acid <<EOF11
     delete from HISTORY;
     \q
EOF11
echo

# Testing ROLLBACK case
echo Atomicity Test -- Starting ROLLBACK
loop at: `date`
echo
i="0"
while read O_KEY L_KEY DELTA
do
if [ ${L_KEY}x != "x" ]; then
  echo ------------------------------
-------------------------
  echo Atom - Rollback -- For O_KEY =
$O_KEY L_KEY = $L_KEY DELTA = $DELTA
  echo Atom - Rollback -- Before Values
  i=$[$i+1]
  psql -q -t tp_acid <<EOF12
     select
            L_EXTENDEDPRICE as
L_EXTENDED,
            L_QUANTITY as L_QUANTITY,
            L_DISCOUNT as L_DISCOUNT,
            L_TAX as L_TAX,
            O_TOTALPRICE as O_TOTAL
       from ORDERS,
            LINEITEM
      where O_ORDERKEY = $O_KEY
        and L_ORDERKEY = O_ORDERKEY
        and L_LINENUMBER = $L_KEY;
     \q
EOF12

./acid_trans.sh $RANDOM $O_KEY $L_KEY
$DELTA ROLLBACK

  echo
  echo Atom - Rollback -- After Values
  psql -q -t tp_acid <<EOF13
     select
            L_EXTENDEDPRICE as
L_EXTENDED,
            L_QUANTITY as L_QUANTITY,
            L_DISCOUNT as L_DISCOUNT,
            L_TAX as L_TAX,
            O_TOTALPRICE as O_TOTAL
       from ORDERS,
            LINEITEM
      where O_ORDERKEY = $O_KEY
        and L_ORDERKEY = O_ORDERKEY
        and L_LINENUMBER = $L_KEY;
     \q
EOF13

fi
done < orderkeys.txt

echo
echo Atom - Rollback -- Records in
History Table
psql -q -t tp_acid <<EOF14
     select *
       from HISTORY;
     \q
EOF14

#
########################################
#################
# Run COMMIT case

echo
echo Atomicity Test -- Starting COMMIT
loop at: `date`
echo
i="0"
while read O_KEY L_KEY DELTA
do
if [ ${L_KEY}x != "x" ]; then
  echo ------------------------------
-------------------------
  echo Atom - Commit -- For O_KEY =
$O_KEY L_KEY = $L_KEY DELTA = $DELTA
  echo Atom - Commit -- Before Values
  i=$[$i+1]
  psql -q -t tp_acid <<EOF22
     select
            L_EXTENDEDPRICE as
L_EXTENDED,
            L_QUANTITY as L_QUANTITY,
            L_DISCOUNT as L_DISCOUNT,
            L_TAX as L_TAX,
            O_TOTALPRICE as O_TOTAL
       from ORDERS,
            LINEITEM
      where O_ORDERKEY = $O_KEY
        and L_ORDERKEY = O_ORDERKEY
        and L_LINENUMBER = $L_KEY;
     \q
EOF22

./acid_trans.sh $RANDOM $O_KEY $L_KEY
$DELTA COMMIT

  echo
  echo Atom - Commit -- After Values
```

```
  psql -q -t tp_acid <<EOF23
     select
            L_EXTENDEDPRICE as
L_EXTENDED,
            L_QUANTITY as L_QUANTITY,
            L_DISCOUNT as L_DISCOUNT,
            L_TAX as L_TAX,
            O_TOTALPRICE as O_TOTAL
       from ORDERS,
            LINEITEM
      where O_ORDERKEY = $O_KEY
        and L_ORDERKEY = O_ORDERKEY
        and L_LINENUMBER = $L_KEY;
     \q
EOF23

fi
done < orderkeys.txt

echo
echo Atom - Commit -- Records in History
Table
psql -q -t tp_acid <<EOF24
     select *
       from HISTORY;
     \q
EOF24

echo
echo Atomicity Test -- Completed at:
`date`
exit
```

==========

acid_consistency_loop.sh

==========

```
#!/bin/bash
#
# TPC-H ACID Consistency Transaction Loop
#
# Copyright 2007 ParAccel, Inc
#

echo Transaction Loop -- Starts `date`
echo
KEYFILE=$1
echo Transaction Loop -- Reading from
$KEYFILE

while read O_KEY L_KEY DELTA
do
   ./acid_trans.sh $RANDOM $O_KEY $L_KEY
$DELTA COMMIT
   sleep 1
done < $KEYFILE

echo Transaction Loop -- Completed `date`
```

==========

acid_consistency_main_v2.sh

==========

```
#!/bin/bash
#
# TPC-H ACID Consistency Main
#
# Copyright 2007 ParAccel, Inc
#
NUMOFKEYS=10
echo Consistency Test -- Starts `date`
echo
./acid_setup.sh $NUMOFKEYS
con_orderkeys1.txt
./acid_setup.sh $NUMOFKEYS
con_orderkeys2.txt
./acid_setup.sh $NUMOFKEYS
con_orderkeys3.txt
./acid_setup.sh $NUMOFKEYS
con_orderkeys4.txt
./acid_setup.sh $NUMOFKEYS
con_orderkeys5.txt
./acid_setup.sh $NUMOFKEYS
con_orderkeys6.txt
./acid_setup.sh $NUMOFKEYS
con_orderkeys7.txt
./acid_setup.sh $NUMOFKEYS
con_orderkeys8.txt
./acid_setup.sh $NUMOFKEYS
con_orderkeys9.txt
./acid_setup.sh $NUMOFKEYS
con_orderkeys10.txt
./acid_setup.sh $NUMOFKEYS
con_orderkeys11.txt

echo Pre verify output...
# loop through smaple keys for
verification
while read O_KEY LKEY DELTA
do
    psql -q -t tp_acid <<EOF10
    select O_ORDERKEY,
           O_TOTALPRICE,

sum(trunc(trunc(L_EXTENDEDPRICE*
           (1-
L_DISCOUNT),2)*(1+L_TAX),2)),
           trunc(O_TOTALPRICE -
sum(trunc(trunc(L_EXTENDEDPRICE*
           (1-
L_DISCOUNT),2)*(1+L_TAX),2)),2)
    from ORDERS join LINEITEM
      on O_ORDERKEY = L_ORDERKEY
     and O_ORDERKEY = $O_KEY
   group by 1, 2;
\q
EOF10
```

```
./acid_consistency_loop.sh
con_orderkeys1.txt > con_log1.out &
sleep 3
./acid_consistency_loop.sh
con_orderkeys2.txt > con_log2.out &
sleep 3
./acid_consistency_loop.sh
con_orderkeys3.txt > con_log3.out &
sleep 3
./acid_consistency_loop.sh
con_orderkeys4.txt > con_log4.out &
sleep 3
./acid_consistency_loop.sh
con_orderkeys5.txt > con_log5.out &
sleep 3
./acid_consistency_loop.sh
con_orderkeys6.txt > con_log6.out &
sleep 3
./acid_consistency_loop.sh
con_orderkeys7.txt > con_log7.out &
sleep 3
./acid_consistency_loop.sh
con_orderkeys8.txt > con_log8.out &
sleep 3
./acid_consistency_loop.sh
con_orderkeys9.txt > con_log9.out &
sleep 3
./acid_consistency_loop.sh
con_orderkeys10.txt > con_log10.out &
sleep 3
./acid_consistency_loop.sh
con_orderkeys11.txt > con_log11.out &
wait
echo Consistency Test -- Transactions
Completed
echo
echo Consistency Test -- Verify Starts
`date`
echo

# Get sample number of keys to verify
tail -3 con_orderkeys1.txt >
con_samplekeys.txt
tail -3 con_orderkeys2.txt >>
con_samplekeys.txt
tail -3 con_orderkeys3.txt >>
con_samplekeys.txt
tail -3 con_orderkeys4.txt >>
con_samplekeys.txt
tail -3 con_orderkeys5.txt >>
con_samplekeys.txt
tail -3 con_orderkeys6.txt >>
con_samplekeys.txt
tail -3 con_orderkeys7.txt >>
con_samplekeys.txt
tail -3 con_orderkeys8.txt >>
con_samplekeys.txt
tail -3 con_orderkeys9.txt >>
con_samplekeys.txt
tail -3 con_orderkeys10.txt >>
con_samplekeys.txt
tail -3 con_orderkeys11.txt >>
con_samplekeys.txt

echo Post verify...

# loop through sample keys for
verification
while read O_KEY LKEY DELTA
do
    psql -q -t tp_acid <<EOF10
    select O_ORDERKEY,
           O_TOTALPRICE,

sum(trunc(trunc(L_EXTENDEDPRICE*
           (1-
L_DISCOUNT),2)*(1+L_TAX),2)),
           trunc(O_TOTALPRICE -
sum(trunc(trunc(L_EXTENDEDPRICE*
           (1-
L_DISCOUNT),2)*(1+L_TAX),2)),2)
    from ORDERS join LINEITEM
      on O_ORDERKEY = L_ORDERKEY
     and O_ORDERKEY = $O_KEY
   group by 1, 2;
\q
EOF10

done < con_samplekeys.txt

echo Consistency Test -- Verify Complete
`date`
echo
echo Consistency Test -- Complete `date`
```

==========

acid_durability_after.sh

==========

```
#!/bin/bash

./acid_durability_verify.sh >
acid_durability_after.out

for arg in 1 2 3 4 5 6 7 8 9 10 11; do
if [ -f dur_log${arg}.log ]
then
count=`(cat dur_log${arg}.out | grep -i
Completed | wc -l)`
echo "Current count for dur_log${arg}.out
= $count"
else
echo "dur${arg}.out does not exist yet.
It is probably too early in the test
sequence."
fi
.sh
```

==========

acid_durability_loop.sh

==========

```
#!/bin/bash
#
# TPC-H ACID Durability Transaction Loop
#
# Copyright 2007 ParAccel, Inc
#

echo Transaction Loop -- Starts `date`
echo
KEYFILE=$1
echo Transaction Loop -- Reading from
$KEYFILE

while read O_KEY L_KEY DELTA
do
   ./acid_trans.sh $RANDOM $O_KEY $L_KEY
$DELTA COMMIT
   if [ $? -eq 0 ]
      then
      echo $O_KEY "|" $L_KEY >>
dur_keys.tbl
      sudo sync;sudo sync;sudo sync
   else
      exit $2
   fi
   sleep 1
done < $KEYFILE

#echo
#echo Durability Test -- First loop
completed for `wc -l $KEYFILE`
transactions
#echo
#echo Durability Test -- ----------------
-----------------------------------
#echo

#while read O_KEY L_KEY DELTA
#do
#   ./acid_trans.sh $RANDOM $O_KEY $L_KEY
$DELTA COMMIT
#   if [ $? -eq 0 ]
#      then
#      echo $O_KEY "|" $L_KEY >>
dur_keys.tbl
#      sudo sync;sudo sync;sudo sync
#   else
#      exit $2
#   fi
#   sleep 1
#done < $KEYFILE

echo
echo Transaction Loop -- Complete `date`
```

==========

acid_durability_main.sh

==========

```
#!/bin/bash
#
# TPC-H ACID Durability Main
#
# Copyright 2007 ParAccel, Inc
#
NUMOFKEYS=150
echo Durability Test -- Starts `date`
echo
echo XXXXXXXX ./acid_setup.sh $NUMOFKEYS
dur_orderkeys1.txt
psql tp_acid -At -c "select max(query)
from stl_query"
./acid_setup.sh $NUMOFKEYS
dur_orderkeys1.txt
echo XXXXXXXX ./acid_setup.sh $NUMOFKEYS
dur_orderkeys2.txt
psql tp_acid -At -c "select max(query)
from stl_query"
./acid_setup.sh $NUMOFKEYS
dur_orderkeys2.txt
echo XXXXXXXX ./acid_setup.sh $NUMOFKEYS
dur_orderkeys3.txt
psql tp_acid -At -c "select max(query)
from stl_query"
./acid_setup.sh $NUMOFKEYS
dur_orderkeys3.txt
echo XXXXXXXX ./acid_setup.sh $NUMOFKEYS
dur_orderkeys4.txt
psql tp_acid -At -c "select max(query)
from stl_query"
./acid_setup.sh $NUMOFKEYS
dur_orderkeys4.txt
echo XXXXXXXX ./acid_setup.sh $NUMOFKEYS
dur_orderkeys5.txt
psql tp_acid -At -c "select max(query)
from stl_query"
./acid_setup.sh $NUMOFKEYS
dur_orderkeys5.txt
echo XXXXXXXX ./acid_setup.sh $NUMOFKEYS
dur_orderkeys6.txt
psql tp_acid -At -c "select max(query)
from stl_query"
./acid_setup.sh $NUMOFKEYS
dur_orderkeys6.txt
echo XXXXXXXX ./acid_setup.sh $NUMOFKEYS
dur_orderkeys7.txt
psql tp_acid -At -c "select max(query)
from stl_query"
./acid_setup.sh $NUMOFKEYS
dur_orderkeys7.txt
echo XXXXXXXX ./acid_setup.sh $NUMOFKEYS
dur_orderkeys8.txt
psql tp_acid -At -c "select max(query)
from stl_query"
```

```
./acid_setup.sh $NUMOFKEYS
dur_orderkeys8.txt
echo XXXXXXXX ./acid_setup.sh $NUMOFKEYS
dur_orderkeys9.txt
psql tp_acid -At -c "select max(query)
from stl_query"
./acid_setup.sh $NUMOFKEYS
dur_orderkeys9.txt
echo XXXXXXXX ./acid_setup.sh $NUMOFKEYS
dur_orderkeys10.txt
psql tp_acid -At -c "select max(query)
from stl_query"
./acid_setup.sh $NUMOFKEYS
dur_orderkeys10.txt
echo XXXXXXXX ./acid_setup.sh $NUMOFKEYS
dur_orderkeys11.txt
psql tp_acid -At -c "select max(query)
from stl_query"
./acid_setup.sh $NUMOFKEYS
dur_orderkeys11.txt

if [ -f dur_keys.tbl ]
then
  rm dur_keys.tbl
fi
# Pre Verify Step
./acid_durability_verify.sh
# Start test
echo Durability Test - Start Transaction
Loop
./acid_durability_loop.sh
dur_orderkeys1.txt > dur_log1.out &
sleep 3
./acid_durability_loop.sh
dur_orderkeys2.txt > dur_log2.out &
sleep 3
./acid_durability_loop.sh
dur_orderkeys3.txt > dur_log3.out &
sleep 3
./acid_durability_loop.sh
dur_orderkeys4.txt > dur_log4.out &
sleep 3
./acid_durability_loop.sh
dur_orderkeys5.txt > dur_log5.out &
sleep 3
./acid_durability_loop.sh
dur_orderkeys6.txt > dur_log6.out &
sleep 3
./acid_durability_loop.sh
dur_orderkeys7.txt > dur_log7.out &
sleep 3
./acid_durability_loop.sh
dur_orderkeys8.txt > dur_log8.out &
sleep 3
./acid_durability_loop.sh
dur_orderkeys9.txt > dur_log9.out &
sleep 3
./acid_durability_loop.sh
dur_orderkeys10.txt > dur_log10.out &
sleep 3
./acid_durability_loop.sh
dur_orderkeys11.txt > dur_log11.out &
wait
echo Durability Test -- Transactions
Completed
echo
# Get count from the history table
HIST_COUNT=`psql -q -t tp_acid <<EOF10
    select count(*) as count_history
      from HISTORY;
\q
EOF10`

echo
echo Durability Test -- Complete `date`
```

acid_durability_verify.sh

```
#!/bin/bash
#
# TPC-H ACID Durability Main
#
# Copyright 2007 ParAccel, Inc
#
# Get count from the history table
HIST_COUNT=`psql -q -t tp_acid <<EOF10
    select count(*)
      from HISTORY;
\q
EOF10`
if [ $? -ne 0 ]
   then
      echo
      echo Durability Test -- Database is
DOWN
      echo
      exit $2
   fi

echo Durability Test -- History count
$HIST_COUNT
echo
echo Durability Test -- Verify Starts
`date`
echo

# Get sample number of keys to verify
tail -3 dur_orderkeys1.txt >
dur_samplekeys.txt
tail -3 dur_orderkeys2.txt >>
dur_samplekeys.txt
tail -3 dur_orderkeys3.txt >>
dur_samplekeys.txt
tail -3 dur_orderkeys4.txt >>
dur_samplekeys.txt
```

```
tail -3 dur_orderkeys5.txt >>
dur_samplekeys.txt
tail -3 dur_orderkeys6.txt >>
dur_samplekeys.txt
tail -3 dur_orderkeys7.txt >>
dur_samplekeys.txt
tail -3 dur_orderkeys8.txt >>
dur_samplekeys.txt
tail -3 dur_orderkeys9.txt >>
dur_samplekeys.txt
tail -3 dur_orderkeys10.txt >>
dur_samplekeys.txt
tail -3 dur_orderkeys11.txt >>
dur_samplekeys.txt

# loop through smaple keys for
verification
while read O_KEY LKEY DELTA
do
    psql -q -t tp_acid <<EOF10
    select O_ORDERKEY,
           O_TOTALPRICE,

sum(trunc(trunc(L_EXTENDEDPRICE*
           (1-
L_DISCOUNT),2)*(1+L_TAX),2)),
           trunc(O_TOTALPRICE -
sum(trunc(trunc(L_EXTENDEDPRICE*
           (1-
L_DISCOUNT),2)*(1+L_TAX),2)),1)
    from ORDERS join LINEITEM
      on O_ORDERKEY = L_ORDERKEY
     and O_ORDERKEY = $O_KEY
   group by 1, 2;
\q
EOF10

done < dur_samplekeys.txt

echo Durability Test -- Verify Complete
`date`
echo
```

==========

acid_isolation_main1.sh

==========

```
#!/bin/bash

echo ISO Test 1 -- Test Starting at
`date`
read O_KEY L_KEY DELTA < orderkeys.txt
echo ISO Test 1 -- Before Query
./acid_query.sh $O_KEY
echo ISO Test 1 -- Starting Transaction
# Run Transaction
./acid_trans.sh $O_KEY $L_KEY $DELTA
COMMIT 15 &
sleep 5
echo ISO Test 1 -- Starting Query
./acid_query.sh $O_KEY
echo ISO Test 1 -- Query Completed
wait
echo ISO Test 1 -- All Completed
./acid_query.sh $O_KEY
echo ISO Test 1 -- End of Test at `date`
exit 0
```

==========

acid_isolation_main2.sh

==========

```
#!/bin/bash

echo ISO Test 2 -- Test Starting at
`date`
read O_KEY L_KEY DELTA < orderkeys.txt
echo ISO Test 2 -- Before Query
./acid_query.sh $O_KEY
echo ISO Test 2 -- Starting Transaction
# Run Transaction
./acid_trans.sh $O_KEY $L_KEY $DELTA
ROLLBACK 15 &
sleep 5
echo ISO Test 2 -- Starting Query
./acid_query.sh $O_KEY
echo ISO Test 2 -- Query Completed
wait
echo ISO Test 2 -- All Completed
./acid_query.sh $O_KEY
echo ISO Test 2 -- End of Test at `date`
exit 0
```

==========

acid_isolation_main3.sh

==========

```
#!/bin/bash

echo ISO Test 3 -- Test Starting at
`date`
read O_KEY L_KEY DELTA < orderkeys.txt
echo ISO Test 3 -- Before Query
./acid_query.sh $O_KEY
echo ISO Test 3 -- Starting Transaction 1
# Run Transaction 1
./acid_trans.sh $O_KEY $L_KEY $DELTA
COMMIT 15 &
sleep 5
echo ISO Test 3 -- Starting Transaction 2
# Run Transaction 2
./acid_trans.sh $O_KEY $L_KEY $DELTA
COMMIT
echo ISO Test 3 -- Transaction 2
Completed
wait
```

```
echo ISO Test 3 -- All Completed
./acid_query.sh $O_KEY
echo ISO Test 3 -- Test Completed at
`date`
exit 0
```

==========

acid_isolation_main4.sh

==========

```
#!/bin/bash

echo ISO Test 4 -- Test Starting at
`date`
read O_KEY L_KEY DELTA < orderkeys.txt
echo ISO Test 4 -- Before Query
./acid_query.sh $O_KEY
echo ISO Test 4 -- Starting Transaction 1
# Run Transaction 1
./acid_trans.sh $O_KEY $L_KEY $DELTA
ROLLBACK 15 &
sleep 5
echo ISO Test 4 -- Starting Transaction 2
# Run Transaction 2
./acid_trans.sh $O_KEY $L_KEY $DELTA
COMMIT
echo ISO Test 4 -- Transaction 2
Completed
wait
echo ISO Test 4 -- All Completed
./acid_query.sh $O_KEY
echo ISO Test 4 -- Test Completed at
`date`
exit 0
```

==========

acid_isolation_main5.sh

==========

```
#!/bin/bash

echo ISO Test 5 -- Test Starting at
`date`

read O_KEY L_KEY DELTA < orderkeys.txt

echo ISO Test 5 -- Fetching PartKey and
SupKey
P_KEY=`psql -q -t tpch <<EOF1
     select L_PARTKEY
       from LINEITEM
      where L_ORDERKEY = $O_KEY
        and L_LINENUMBER  = $L_KEY
     \q
EOF1`

S_KEY=`psql -q -t tpch <<EOF2
     select L_SUPPKEY
       from LINEITEM
      where L_ORDERKEY = $O_KEY
        and L_LINENUMBER  = $L_KEY
     \q
EOF2`
echo ISO Test 5 -- PartKey = $P_KEY,
SuppKey = $S_KEY

echo ISO Test 5 -- Before Query
./acid_query.sh $O_KEY
echo ISO Test 5 -- Starting Transaction 1
# Run Transaction 1
./acid_trans.sh $O_KEY $L_KEY $DELTA
COMMIT 15 &
sleep 5
echo ISO Test 5 -- Starting Partsup Query
# Run Transaction 2
./iso5_query.sh $P_KEY $S_KEY
echo ISO Test 5 -- Partsup Query
Completed
wait
echo ISO Test 5 -- All Completed
./acid_query.sh $O_KEY
echo ISO Test 5 -- Test Completed at
`date`
exit 0
```

==========

acid_isolation_main6.sh

==========

```
#!/bin/bash

echo ISO Test 6 -- Test Starting at
`date`
read O_KEY1 L_KEY1 DELTA1 < orderkeys.txt
tail -1 orderkeys.txt >
/tmp/orderkeys.tail
read O_KEY2 L_KEY2 DELTA2 <
/tmp/orderkeys.tail

echo ISO Test 6 -- Before Query
./acid_query.sh $O_KEY1
echo ISO Test 6 -- Starting ISO 6 Query
./iso6_query.sh $O_KEY2 &
sleep 2
echo ISO Test 6 -- Starting Transaction
./acid_trans.sh $O_KEY1 $L_KEY1 $DELTA1
COMMIT
echo ISO Test 6 -- Transaction Completed
wait
echo ISO Test 6 -- All Completed
./acid_query.sh $O_KEY1
echo ISO Test 6 -- Test Completed at
`date`
exit 0
```

==========

acid_query.sh

==========

```
#!/bin/bash
```

```
O_KEY=$1

# Execute read-only query for isolation
tests

echo "ACID Query -- Started at `date`"
O_TOTAL=`psql -q -t tpch <<EOF13
     select round(o_totalprice,2)
     from ORDERS
      where O_ORDERKEY = $O_KEY;
     \q
EOF13`
echo "ACID Query -- o_orderkey = $O_KEY"
echo "ACID Query -- o_total = $O_TOTAL"
echo "ACID Query -- Completed at `date`"
exit 0
```

==========

acid_setup.sh

==========

```
#!/bin/bash

echo ParAccel ACID Setup Starts `date`
echo


if [ -z "$2" ]
then
  OUTFILE="orderkeys.txt"
else
  OUTFILE="$2"
fi

# Clean history table
psql -q -t tpch <<EOF10
     delete from HISTORY;
     \q
EOF10

# Create file with order keys to be used
for consistency testing

if [ -f $OUTFILE ]
then
  rm $OUTFILE
fi

# Get Max Order Key
MAX_O_KEY=`psql -q -t tpch <<EOF10
     select  max(O_ORDERKEY)
        from ORDERS;
     \q
EOF10`

# echo Maximum Order Key $MAX_O_KEY

i="0"
while [ $i -lt $1 ]
do
# Get Random Orderkey and maximum
lineitem
  DELTA=$RANDOM
  let "DELTA %= 101"
  if [ $DELTA -eq 0 ]; then
     DELTA=1
  fi
  O_KEY=$RANDOM
  let "O_KEY %= ${MAX_O_KEY}"
  L_KEY=`psql -q -t tpch <<EOF11
     select  max(L_LINENUMBER)
        from LINEITEM
       where L_ORDERKEY = $O_KEY;
  \q
EOF11`
if [ ${L_KEY}x != "x" ]; then
  echo $O_KEY $L_KEY $DELTA
  echo $O_KEY $L_KEY $DELTA >> $OUTFILE
  i=$[$i+1]

fi
done

echo
echo ParAccel ACID  Setup Complete `date`

exit
```

==========

acid_trans.sh

==========

```
#!/bin/bash

TRANS_KEY=$1
O_KEY=$2
L_KEY=$3
DELTA=$4
COMMIT=$5
SLEEP=$6

echo ACID Trans -- Started at: `date`
if [ -z $SLEEP ];   then
     let "SLEEP = 0"
     echo $TRANS_KEY ACID Trans -- No
Sleep before ${COMMIT} ;
else
     let "SLEEP = $SLEEP"
     echo $TRANS_KEY ACID Trans -- With
Sleep = $SLEEP before ${COMMIT} ;
fi

# Perform updates and insert for the ACID
test
```

```
echo $TRANS_KEY ACID Trans -- o_orderkey
= ${O_KEY}, line_number = ${L_KEY}, delta
= $DELTA

psql -q -t tp_acid <<EOFb >> /dev/null
BEGIN TRANSACTION;

-- 1. Update Orders a first time to
remove old lineitem from total price

update ORDERS
   set O_TOTALPRICE = O_TOTALPRICE - (
     select TRUNC( TRUNC(L_EXTENDEDPRICE
* (1.0 - L_DISCOUNT), 2) * (1.0 + L_TAX),
2 )
       from LINEITEM
      where L_ORDERKEY = $O_KEY
        and L_LINENUMBER = $L_KEY )
 where O_ORDERKEY = $O_KEY ;

-- 2. Update Linetime to increase
quantity

update LINEITEM
   set L_EXTENDEDPRICE = L_EXTENDEDPRICE
+ TRUNC(
TRUNC(L_EXTENDEDPRICE/L_QUANTITY, 2) *
$DELTA, 2 ),
       L_QUANTITY = L_QUANTITY +  $DELTA
 where L_ORDERKEY = $O_KEY
   and L_LINENUMBER = $L_KEY ;

-- 3. Update Orders a second time to add
updated lineitem to total price

update ORDERS
   set O_TOTALPRICE = O_TOTALPRICE + (
     select TRUNC( TRUNC(L_EXTENDEDPRICE
* (1.0 - L_DISCOUNT), 2)* (1 + L_TAX), 2
)
       from LINEITEM
      where L_ORDERKEY = $O_KEY
        and L_LINENUMBER = $L_KEY )
 where O_ORDERKEY = $O_KEY ;

-- 4. Insert History records of
transaction

insert into HISTORY
   (select L_PARTKEY, L_SUPPKEY,
L_ORDERKEY, L_LINENUMBER, $DELTA,
LOCALTIMESTAMP
      from LINEITEM
     where L_ORDERKEY = $O_KEY
       and L_LINENUMBER = $L_KEY );
select sleep($SLEEP);

$COMMIT TRANSACTION;
EOFb

RC=$?

#if [ -z $SLEEP ] then
#  echo $TRANS_KEY No sleep
#else
#  echo $TRANS_KEY going to sleep at
`date`
#select sleep($SLEEP);
#fi

echo $TRANS_KEY ACID Trans -- Completed
at: `date`
exit $RC
```

check_counts.sh

```
#!/bin/bash

while(true)
do
num=`(cat dur_keys.tbl | wc -l)`
if (( $num < 10000 ))
then
   echo "The test has not yet reached the
mid-point"
   echo "--------------------------------
--------------------"
   date
   for arg in 1 2 3 4 5 6 7 8 9 ; do cat
dur_log${arg}.out | grep -i "first loop";
done
   echo "Transaction count: $num"
   echo "--------------------------------
--------------------"
fi
sleep $1
done
```

check_progress.sh

```
#!/bin/bash

while true; do
for arg in 1 2 3 4 5 6 7 8 9 10 11; do
if [ -f dur_log${arg}.out ]
then
count=`(cat dur_log${arg}.out | grep -i
Completed | wc -l)`
echo "Current count for dur_log${arg}.out
= $count"
else
echo "dur$arg does not exist yet. It is
probably too early in the test sequence."
fi;
done
```

```
echo "-----------------------------------
---------------------------------------
"
echo "To check progress, run this script.
Also tail -f acid_durability_main.out. "
echo "Once all streams have reached 100
transactions:"
echo "     1) Pull a drive on any node"
echo "     2) Wait 60 seconds"
echo "     3) Crash the entire cluster
(26 nodes)"
echo "     4) After a couple of minutes,
power the server back on and restart xen"
echo "     5) Verify that xen is running
(if in doubt: psql tp_acid and wait)  run
acid_durability_verify.sh"
echo "     6) As a sanity check, compare
a wc -l dur_keys against psql tp_acid -c
select count(*) from history"
sleep 10;
clear;
done
```

==========

check_transactions.sh

==========

```
#!/bin/bash

count1=`(cat
$HOME/run/tpch_acid/dur_keys.tbl | wc -
l)`
count2=0
while true
  do
  sleep 10
  date
  count1=`(cat
$HOME/run/tpch_acid/dur_keys.tbl | wc -
l)`
  if (( $count2 >= $count1 ))
    then
    echo "Error: The transactions have
stopped!"
    exit
  fi
  ((count2=count1))
  echo "-----------------------------------
----------------------"
  echo "Transactions are still running.
Current count: $count1"
  echo "-----------------------------------
----------------------"
done
```

==========

clean.sh

==========

```
#!/bin/bash

rm -f *.txt
rm -f *.out
rm -f *_log*
rm -f *.tbl
```

==========

create_history.ddl

==========

```
create table HISTORY
(
H_P_KEY int,
H_S_KEY int,
H_O_KEY int,
H_L_KEY int,
H_DELTA int,
H_DATE_T datetime
);
```

==========

iso5_query.sh

==========

```
#!/bin/bash

P_KEY=$1
S_KEY=$2

echo ISO Query -- Starting at `date`

PS_DATA=`psql -q -t tp_acid <<EOF1
     select PS_AVAILQTY
       from PARTSUPP
      where PS_PARTKEY = $P_KEY
        and PS_SUPPKEY  = $S_KEY
     \q
EOF1`
echo ISO Query -- PS_AVAILQTY = $PS_DATA

PS_DATA=`psql -q -t tp_acid <<EOF2
     select PS_SUPPLYCOST
       from PARTSUPP
      where PS_PARTKEY = $P_KEY
        and PS_SUPPKEY  = $S_KEY
     \q
EOF2`
echo ISO Query -- PS_SUPPLYCOST =
$PS_DATA

PS_DATA=`psql -q -t tp_acid <<EOF3
     select PS_COMMENT
       from PARTSUPP
      where PS_PARTKEY = $P_KEY
        and PS_SUPPKEY  = $S_KEY
     \q
EOF3`
echo ISO Query -- PS_COMMENT = $PS_DATA

echo ISO Query -- Completed at `date`
```

```
====================
```
iso6_query.sh
```
====================
#!/bin/bash

O_KEY=$1

echo ISO6 Query1 -- Starting at `date`

PS_DATAgrep =`psql -q -t tp_acid <<EOF1
select L_EXTENDEDPRICE,
SLEEP(L_LINENUMBER*4)
  from LINEITEM
 where L_ORDERKEY = $O_KEY
   and L_LINENUMBER < 4
     \q
EOF1`
echo ISO6 Query1 -- L_EXTENDEDPRICE =
$PS_DATA

echo ISO6 Query1 -- Completed at `date`

echo ISO6 Query2 -- Starting at `date`

PS_DATAgrep =`psql -q -t tp_acid <<EOF1
select L_EXTENDEDPRICE,
SLEEP(L_LINENUMBER*4)
  from LINEITEM
 where L_ORDERKEY = $O_KEY
   and L_LINENUMBER < 4
     \q
EOF1`
echo ISO6 Query2 -- L_EXTENDEDPRICE =
$PS_DATA

echo ISO6 Query2 -- Completed at `date`

====================
```
tpch_acid_wrapper.sh
```
====================
#!/bin/bash

sh clean.sh

echo "----------------------------------
-------"
echo " Shutting down the database. "
echo "----------------------------------
-------"
#cqi xstop
#sleep 5
echo "----------------------------------
-------"
echo " Initializing the database. "
echo "----------------------------------
-------"
#cqi initdb -f
#sleep 5
echo "----------------------------------
-------"
echo " Starting the database. "
echo "----------------------------------
-------"
#cqi xstart
#sleep 20
echo "----------------------------------
-------"
echo " Loading the 1GB TPC-H data set. "
echo "----------------------------------
-------"
cd tpch1g/
./load_tpch.sh > aci_load_build.out
echo "----------------------------------
-------"
echo " Starting the acid tests. "
echo "----------------------------------
-------"
# go back to the home directory for the
acid tests
cd ..
echo "----------------------------------
-------"
echo " Creating the history table. "
echo "----------------------------------
-------"
psql tp_acid -f create_history.ddl >>
aci_create_history.out
echo "----------------------------------
-------"
echo " Running the verification script. "
echo "----------------------------------
-------"
psql tp_acid -f dbtables.sql >>
aci_dbtables.out
echo "Qualification complete"
echo
cd tpch1g/
run_tpch.sh > aci_run_tpch.out
cd ..
echo "----------------------------------
-------"
echo " Creating the seed file with three
records "
echo "----------------------------------
-------"
acid_setup.sh 3 >> acid_setup.out
echo "----------------------------------
-------"
echo " Running the atomicity test "
echo "----------------------------------
-------"
acid_atomicity.sh >> acid_atomicity.out
echo "----------------------------------
-------"
echo " Running the consistency test "
```

```
echo "-----------------------------------
-------"
acid_consistency_main_v2.sh >>
acid_consistency_main.out
echo "-----------------------------------
-------"
echo " Running the isolation group "
echo "-----------------------------------
-------"
for arg in 1 2 3 4 5 6; do acid_setup.sh
3 >> acid_setup.out
acid_isolation_main$arg.sh >>
acid_isolation_main$arg.out ;done
echo "-----------------------------------
-------"
echo " Running durability "
echo "-----------------------------------
-------"
acid_durability_main.sh >
acid_durability_main.out &
check_progress.sh
```

Appendix C: Query Text and Query Output

```
====================Q01==============


select
      l_returnflag,
      l_linestatus,
      sum(l_quantity) as sum_qty,
      sum(l_extendedprice) as
sum_base_price,
      sum(l_extendedprice * (1 -
l_discount)) as sum_disc_price,
      sum(l_extendedprice * (1 -
l_discount) * (1 + l_tax)) as sum_charge,
      avg(l_quantity) as avg_qty,
      avg(l_extendedprice) as avg_price,
      avg(cast(l_discount as
numeric(14,4))) as avg_disc,
      count(*) as count_order
from
      lineitem
where
      l_shipdate <= cast(date '1998-12-
01' - interval '90 days' as date)
group by
      l_returnflag,
      l_linestatus
order by
      l_returnflag,
      l_linestatus;
 l_returnflag | l_linestatus |   sum_qty
| sum_base_price  |  sum_disc_price   |
sum_charge      | avg_qty | avg_price |
avg_disc | count_order
--------------+--------------+-----------
--+-----------------+------------------
+--------------------+---------+--------
---+----------+-------------
 A            | F            |
37734107.00 |  56586554400.73 |
53758257134.8700 |  55909065222.827692 |
25.52 |  38273.12 |   0.0499 |
1478493
 N            | F            |
991417.00 |   1487504710.38 |
1413082168.0541 |   1469649223.194375 |
25.51 |  38284.46 |   0.0500 |
38854
 N            | O            |
74476040.00 | 111701729697.74 |
106118230307.6056 | 110367043872.497010 |
25.50 |  38249.11 |   0.0499 |
2920374
 R            | F            |
37719753.00 |  56568041380.90 |
53741292684.6040 |  55889619119.831932 |
25.50 |  38250.85 |   0.0500 |
1478870
(4 rows)


====================Q02==============


select
      s_acctbal,
      s_name,
      n_name,
      p_partkey,
      p_mfgr,
      s_address,
      s_phone,
      s_comment
from
      part,
      supplier,
      partsupp,
      nation,
      region
where
      p_partkey = ps_partkey
      and s_suppkey = ps_suppkey
      and p_size = 15
      and p_type like '%BRASS'
      and s_nationkey = n_nationkey
      and n_regionkey = r_regionkey
      and r_name = 'EUROPE'
      and ps_supplycost = (
            select
                  min(ps_supplycost)
            from
                  partsupp,
                  supplier,
                  nation,
                  region
            where
                  p_partkey = ps_partkey
                  and s_suppkey =
ps_suppkey
                  and s_nationkey =
n_nationkey
                  and n_regionkey =
r_regionkey
                  and r_name = 'EUROPE'
      )
order by
      s_acctbal desc,
      n_name,
      s_name,
      p_partkey
limit 100;
```

```
 s_acctbal |          s_name           |
n_name           | p_partkey |
p_mfgr           |
s_address                |     s_phone
|
s_comment
----------+---------------------------+-
-------------------------+-----------+--
------------------------+--------------
--------------------------+-------------
----+-------------------------------------
---------------------------------------
-------------------------
   9938.53 | Supplier#000005359        |
UNITED KINGDOM           |    185358 |
Manufacturer#4           |
QKuHYh,vZGiwu2FWEJoLDx04
| 33-429-790-6131 | uriously regular
requests hag
   9937.84 | Supplier#000005969        |
ROMANIA                  |    108438 |
Manufacturer#1           |
ANDENSOSmk,miq23Xfb5RWt6dvUcvt6Qa
| 29-520-692-3537 | efully express
instructions. regular requests against
the slyly fin
   9936.22 | Supplier#000005250        |
UNITED KINGDOM           |       249 |
Manufacturer#4           |
B3rqp0xbSEim4Mpy2RH J
| 33-320-228-2957 | etect about the
furiously final accounts. slyly ironic
pinto beans sleep inside the furiously
   9923.77 | Supplier#000002324        |
GERMANY                  |     29821 |
Manufacturer#4           | y3OD9UywSTOk
| 17-779-299-1839 | ackages boost
blithely. blithely regular deposits c
   9871.22 | Supplier#000006373        |
GERMANY                  |     43868 |
Manufacturer#5           | J8fcXWsTqM
| 17-813-485-8637 | etect blithely bold
asymptotes. fluffily ironic platelets
wake furiously; blit
   9870.78 | Supplier#000001286        |
GERMANY                  |     81285 |
Manufacturer#2           |
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH
| 17-516-924-4574 |  regular accounts.
furiously unusual courts above the fi
   9870.78 | Supplier#000001286        |
GERMANY                  |    181285 |
Manufacturer#4           |
YKA,E2fjiVd7eUrzp2Ef8j1QxGo2DFnosaTEH
| 17-516-924-4574 |  regular accounts.
furiously unusual courts above the fi
   9852.52 | Supplier#000008973        |
RUSSIA                   |     18972 |
Manufacturer#2           |
t5L67YdBYYH6o,Vz24jpDyQ9
| 32-188-594-7038 | rns wake final foxes.
carefully unusual depende
…
   7980.65 | Supplier#000001288        |
FRANCE                   |     13784 |
Manufacturer#4           | zE,7HgVPrCn
| 16-646-464-8247 | ully bold courts.
escapades nag slyly. furiously fluffy
theodo
   7950.37 | Supplier#000008101        |
GERMANY                  |     33094 |
Manufacturer#5           |
kkYvL6IuvojJgTNG IKkaXQDYgx8ILohj
| 17-627-663-8014 | arefully unusual
requests x-ray above the quickly final
deposits.
   7937.93 | Supplier#000009012        |
ROMANIA                  |     83995 |
Manufacturer#2           |
iUiTziH,Ek3i4lwSgunXMgrcTzwdb
| 29-250-925-9690 | to the blithely
ironic deposits nag sly
   7914.45 | Supplier#000001013        |
RUSSIA                   |    125988 |
Manufacturer#2           |
riRcntps4KEDtYScjpMIWeYF6mNnR
| 32-194-698-3365 |  busily bold packages
are dolphi
   7912.91 | Supplier#000004211        |
GERMANY                  |    159180 |
Manufacturer#5           |
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG
| 17-266-947-7315 | ay furiously regular
platelets. cou
   7912.91 | Supplier#000004211        |
GERMANY                  |    184210 |
Manufacturer#4           |
2wQRVovHrm3,v03IKzfTd,1PYsFXQFFOG
| 17-266-947-7315 | ay furiously regular
platelets. cou
   7894.56 | Supplier#000007981        |
GERMANY                  |     85472 |
Manufacturer#4           |
NSJ96vMROAbeXP
| 17-963-404-3760 | ic platelets affix
after the furiously
   7887.08 | Supplier#000009792        |
GERMANY                  |    164759 |
Manufacturer#3           |
Y28ITVeYriT3kIGdV2K8fSZ V2UqT5H1Otz
| 17-988-938-4296 | ckly around the
carefully fluffy theodolites. slyly
ironic pack
   7871.50 | Supplier#000007206        |
RUSSIA                   |    104695 |
Manufacturer#1           | 3w
fNCnrVmvJjE95sgWZzvW                  |
32-432-452-7731 | ironic requests.
```

```
furiously final theodolites cajole.
final, express packages sleep. quickly
reg
   7852.45 | Supplier#000005864        |
RUSSIA                    |      8363 |
Manufacturer#4            |
WCNfBPZeSXh3h,c
| 32-454-883-3821 | usly unusual pinto
beans. brave ideas sleep carefully
quickly ironi
   7850.66 | Supplier#000001518        |
UNITED KINGDOM            |     86501 |
Manufacturer#1            | ONda3YJiHKJOC
| 33-730-383-3892 | ifts haggle fluffily
pending pai
   7843.52 | Supplier#000006683        |
FRANCE                    |     11680 |
Manufacturer#4            |
2Z0JGkiv01Y00oCFwUGfviIbhzCdy
| 16-464-517-8943 |  express, final pinto
beans x-ray slyly asymptotes. unusual,
unusual
(100 rows)
```

```
====================Q03==============
```

```
select
      l_orderkey,
      sum(l_extendedprice * (1 -
l_discount)) as revenue,
      o_orderdate,
      o_shippriority
from
      customer,
      orders,
      lineitem
where
      c_mktsegment = 'BUILDING'
      and c_custkey = o_custkey
      and l_orderkey = o_orderkey
      and o_orderdate < date '1995-03-15'
      and l_shipdate > date '1995-03-15'
group by
      l_orderkey,
      o_orderdate,
      o_shippriority
order by
      revenue desc,
      o_orderdate
limit 10;
 l_orderkey |   revenue   | o_orderdate |
o_shippriority
------------+-------------+-------------
+----------------
    2456423 | 406181.0111 | 1995-03-05  |
0
    3459808 | 405838.6989 | 1995-03-04  |
0
     492164 | 390324.0610 | 1995-02-19  |
0
    1188320 | 384537.9359 | 1995-03-09  |
0
    2435712 | 378673.0558 | 1995-02-26  |
0
    4878020 | 378376.7952 | 1995-03-12  |
0
    5521732 | 375153.9215 | 1995-03-13  |
0
    2628192 | 373133.3094 | 1995-02-22  |
0
     993600 | 371407.4595 | 1995-03-05  |
0
    2300070 | 367371.1452 | 1995-03-13  |
0
(10 rows)
```

```
====================Q04==============
```

```
select
      o_orderpriority,
      count(*) as order_count
from
      orders
where
      o_orderdate >= date '1993-07-01'
      and o_orderdate < cast(date '1993-
07-01' + interval '3 months' as date)
      and exists (
            select
                  *
            from
                  lineitem
            where
                  l_orderkey = o_orderkey
                  and l_commitdate <
l_receiptdate
      )
group by
      o_orderpriority
order by
      o_orderpriority;
 o_orderpriority | order_count
-----------------+-------------
 1-URGENT        |       10594
 2-HIGH          |       10476
 3-MEDIUM        |       10410
 4-NOT SPECIFIED |       10556
 5-LOW           |       10487
(5 rows)
```

```
====================Q05==============


select
      n_name,
      sum(l_extendedprice * (1 -
l_discount)) as revenue
from
      customer,
      orders,
      lineitem,
      supplier,
      nation,
      region
where
      c_custkey = o_custkey
      and l_orderkey = o_orderkey
      and l_suppkey = s_suppkey
      and c_nationkey = s_nationkey
      and s_nationkey = n_nationkey
      and n_regionkey = r_regionkey
      and r_name = 'ASIA'
      and o_orderdate >= date '1994-01-
01'
      and o_orderdate < cast(date '1994-
01-01' + interval '1 year' as date)
group by
      n_name
order by
      revenue desc;
          n_name           |    revenue
---------------------------+-------------
--
 INDONESIA                 |
55502041.1697
 VIETNAM                   |
55295086.9967
 CHINA                     |
53724494.2566
 INDIA                     |
52035512.0002
 JAPAN                     |
45410175.6954
(5 rows)



====================Q06==============


select
      sum(l_extendedprice * l_discount)
as revenue
from
      lineitem
where
      l_shipdate >= date '1994-01-01'
      and l_shipdate < cast(date '1994-
01-01' + interval '1 year' as date)
      and l_discount between .06 - 0.01
and .06 + 0.01
      and l_quantity < 24;
    revenue
---------------
 123141078.2283
(1 row)



====================Q07==============


select
      supp_nation,
      cust_nation,
      l_year,
      sum(volume) as revenue
from
      (
            select
                  n1.n_name as
supp_nation,
                  n2.n_name as
cust_nation,
                  datepart(year,
l_shipdate) as l_year,
                  l_extendedprice * (1 -
l_discount) as volume
            from
                  supplier,
                  lineitem,
                  orders,
                  customer,
                  nation n1,
                  nation n2
            where
                  s_suppkey = l_suppkey
                  and o_orderkey =
l_orderkey
                  and c_custkey =
o_custkey
                  and s_nationkey =
n1.n_nationkey
                  and c_nationkey =
n2.n_nationkey
                  and (
                        (n1.n_name =
'FRANCE' and n2.n_name = 'GERMANY')
                        or (n1.n_name =
'GERMANY' and n2.n_name = 'FRANCE')
                  )
                  and l_shipdate between
date '1995-01-01' and date '1996-12-31'
      ) as shipping
group by
      supp_nation,
      cust_nation,
      l_year
```

```
order by
      supp_nation,
      cust_nation,
      l_year;
        supp_nation        |
cust_nation        | l_year |    revenue
---------------------------+-------------
--------------+--------+---------------
 FRANCE                    | GERMANY
|   1995 | 54639732.7336
 FRANCE                    | GERMANY
|   1996 | 54633083.3076
 GERMANY                   | FRANCE
|   1995 | 52531746.6697
 GERMANY                   | FRANCE
|   1996 | 52520549.0224
(4 rows)
```

```
====================Q08==============
```

```
select
      o_year,
      sum(case
            when nation = 'BRAZIL' then
volume
            else 0
      end) / sum(volume) as mkt_share
from
      (
            select

      datepart(year,o_orderdate) as
o_year,
                  l_extendedprice * (1 -
l_discount) as volume,
                  n2.n_name as nation
            from
                  part,
                  supplier,
                  lineitem,
                  orders,
                  customer,
                  nation n1,
                  nation n2,
                  region
            where
                  p_partkey = l_partkey
                  and s_suppkey =
l_suppkey
                  and l_orderkey =
o_orderkey
                  and o_custkey =
c_custkey
                  and c_nationkey =
n1.n_nationkey
                  and n1.n_regionkey =
r_regionkey
                  and r_name = 'AMERICA'
                  and s_nationkey =
n2.n_nationkey
                  and o_orderdate between
date '1995-01-01' and date '1996-12-31'
                  and p_type = 'ECONOMY
ANODIZED STEEL'
      ) as all_nations
group by
      o_year
order by
      o_year;
 o_year | mkt_share
--------+-----------
   1995 |    0.0344
   1996 |    0.0414
(2 rows)
```

```
====================Q09==============
```

```
select
      nation,
      o_year,
      sum(amount) as sum_profit
from
      (
            select
                  n_name as nation,

      datepart(year,o_orderdate) as
o_year,
                  l_extendedprice * (1 -
l_discount) - ps_supplycost * l_quantity
as amount
            from
                  part,
                  supplier,
                  lineitem,
                  partsupp,
                  orders,
                  nation
            where
                  s_suppkey = l_suppkey
                  and ps_suppkey =
l_suppkey
                  and ps_partkey =
l_partkey
                  and p_partkey =
l_partkey
                  and o_orderkey =
l_orderkey
                  and s_nationkey =
n_nationkey
```

```
                and p_name like
'%green%'
      ) as profit
group by
      nation,
      o_year
order by
      nation,
      o_year desc;
          nation           | o_year |
sum_profit
---------------------------+--------+----
-----------
 ALGERIA                   |   1998 |
31342867.2345
 ALGERIA                   |   1997 |
57138193.0233
 ALGERIA                   |   1996 |
56140140.1330
 ALGERIA                   |   1995 |
53051469.6534
 ALGERIA                   |   1994 |
53867582.1286
 ALGERIA                   |   1993 |
54942718.1324
 ALGERIA                   |   1992 |
54628034.7127
 ARGENTINA                 |   1998 |
30211185.7081
 ARGENTINA                 |   1997 |
50805741.7523
…
 UNITED STATES             |   1994 |
49296747.1827
 UNITED STATES             |   1993 |
48029946.8014
 UNITED STATES             |   1992 |
48671944.4983
 VIETNAM                   |   1998 |
30442736.0594
 VIETNAM                   |   1997 |
50309179.7942
 VIETNAM                   |   1996 |
50488161.4100
 VIETNAM                   |   1995 |
49658284.6125
 VIETNAM                   |   1994 |
50596057.2607
 VIETNAM                   |   1993 |
50953919.1519
 VIETNAM                   |   1992 |
49613838.3151
(175 rows)



====================Q10==============
```

```
select
      c_custkey,
      c_name,
      sum(l_extendedprice * (1 -
l_discount)) as revenue,
      c_acctbal,
      n_name,
      c_address,
      c_phone,
      c_comment
from
      customer,
      orders,
      lineitem,
      nation
where
      c_custkey = o_custkey
      and l_orderkey = o_orderkey
      and o_orderdate >= date '1993-10-
01'
      and o_orderdate < cast(date '1993-
10-01' + interval '3 months' as date)
      and l_returnflag = 'R'
      and c_nationkey = n_nationkey
group by
      c_custkey,
      c_name,
      c_acctbal,
      c_phone,
      n_name,
      c_address,
      c_comment
order by
      revenue desc
limit 20;
 c_custkey |       c_name       |
revenue   | c_acctbal |          n_name
|                c_address
|     c_phone     |
c_comment
-----------+--------------------+--------
-----+-----------+----------------------
----+--------------------------------------
------+-----------------+----------------
---------------------------------------
---------------------------------------
----------------
     57040 | Customer#000057040 |
734235.2455 |    632.87 | JAPAN
| Eioyzjf4pp
| 22-895-641-3466 | sits. slyly regular
requests sleep alongside of the regular
inst
    143347 | Customer#000143347 |
721002.6948 |   2557.47 | EGYPT
| 1aReFYv,Kw4
| 14-742-935-3718 | ggle carefully
enticing requests. final deposits use
bold, bold pinto beans. ironic, idle re
```

```
     60838 | Customer#000060838 |
679127.3077 |   2454.77 | BRAZIL
| 64EaJ5vMAHWJlBOxJklpNc2RJiWE
| 12-913-494-9813 |  need to boost
against the slyly regular account
    101998 | Customer#000101998 |
637029.5667 |   3790.89 | UNITED KINGDOM
| 01c9CILnNtfOQYmZj
| 33-593-865-6378 | ress foxes wake slyly
after the bold excuses. ironic platelets
are furiously carefully bold theodolites
    125341 | Customer#000125341 |
633508.0860 |   4983.51 | GERMANY
| S29ODD6bceU8QSuuEJznkNaK
| 17-582-695-5962 | arefully even depths.
blithely even excuses sleep furiously.
foxes use except the dependencies. ca
     25501 | Customer#000025501 |
620269.7849 |   7725.04 | ETHIOPIA
|   W556MXuoiaYCCZamJI,Rn0B4ACUGdkQ8DZ
| 15-874-808-6793 | he pending
instructions wake carefully at the pinto
beans. regular, final instructions along
the slyly fina
    115831 | Customer#000115831 |
596423.8672 |   5098.10 | FRANCE
| rFeBbEEyk dl
ne7zV5fDrmiq1oK09wV7pxqCgIc | 16-715-386-
3788 | l somas sleep. furiously final
deposits wake blithely regular pinto b
     84223 | Customer#000084223 |
594998.0239 |    528.65 | UNITED KINGDOM
| nAVZCs6BaWap rrM27N 2qBnzc5WBauxbA
| 33-442-824-8191 |  slyly final deposits
haggle regular, pending dependencies.
pending escapades wake
     54289 | Customer#000054289 |
585603.3918 |   5583.02 | IRAN
| vXCxoCsU0Bad5JQI ,oobkZ
| 20-834-292-4707 | ely special foxes are
quickly finally ironic p
     39922 | Customer#000039922 |
584878.1134 |   7321.11 | GERMANY
| Zgy4s50l2GKN4pLDPBU8m342gIw6R
| 17-147-757-8036 | y final requests.
furiously final foxes cajole blithely
special platelets. f
      6226 | Customer#000006226 |
576783.7606 |   2230.09 | UNITED KINGDOM
| 8gPu8,NPGkfyQQ0hcIYUGPIBWc,ybP5g,
| 33-657-701-3391 | ending platelets
along the express deposits cajole
carefully final
       922 | Customer#000000922 |
576767.5333 |   3869.25 | GERMANY
| Az9RFaut7NkPnc5zSD2PwHgVwr4jRzq
| 17-945-916-9648 | luffily fluffy
deposits. packages c
    147946 | Customer#000147946 |
576455.1320 |   2030.13 | ALGERIA
| iANyZHjqhyy7Ajah0pTrYyhJ
| 10-886-956-3143 | ithely ironic
deposits haggle blithely ironic requests.
quickly regu
    115640 | Customer#000115640 |
569341.1933 |   6436.10 | ARGENTINA
| Vtgfia9qI 7EpHgecU1X
| 11-411-543-4901 | ost slyly along the
patterns; pinto be
     73606 | Customer#000073606 |
568656.8578 |   1785.67 | JAPAN
| xuR0Tro5yChDfOCrjkd2ol
| 22-437-653-6966 | he furiously regular
ideas. slowly
    110246 | Customer#000110246 |
566842.9815 |   7763.35 | VIETNAM
| 7KzflgX MDOq7sOkI
| 31-943-426-9837 | egular deposits serve
blithely above the fl
    142549 | Customer#000142549 |
563537.2368 |   5085.99 | INDONESIA
| ChqEoK43OysjdHbtKCp6dKqjNyvvi9
| 19-955-562-2398 | sleep pending courts.
ironic deposits against the carefully
unusual platelets cajole carefully
express accounts.
    146149 | Customer#000146149 |
557254.9865 |   1791.55 | ROMANIA
| s87fvzFQpU
| 29-744-164-6487 |  of the slyly silent
accounts. quickly final accounts across
the
     52528 | Customer#000052528 |
556397.3509 |    551.79 | ARGENTINA
| NFztyTOR10UOJ
| 11-208-192-3205 |  deposits hinder.
blithely pending asymptotes breach slyly
regular re
     23431 | Customer#000023431 |
554269.5360 |   3381.86 | ROMANIA
| HgiV0phqhaIa9aydNoIlb
| 29-915-458-2654 | nusual, even
instructions: furiously stealthy n
(20 rows)



====================Q11==============


select
      ps_partkey,
      sum(ps_supplycost * ps_availqty) as
value
from
      partsupp,
      supplier,
```

```
      nation
where
      ps_suppkey = s_suppkey
      and s_nationkey = n_nationkey
      and n_name = 'GERMANY'
group by
      ps_partkey having
            sum(ps_supplycost *
ps_availqty) > (
                  select
                        sum(ps_supplycost
* ps_availqty) * 0.0001000000
                  from
                        partsupp,
                        supplier,
                        nation
                  where
                        ps_suppkey =
s_suppkey
                        and s_nationkey =
n_nationkey
                        and n_name =
'GERMANY'
            )
order by
      value desc;
 ps_partkey |    value
------------+-------------
     129760 | 17538456.86
     166726 | 16503353.92
     191287 | 16474801.97
     161758 | 16101755.54
      34452 | 15983844.72
     139035 | 15907078.34
       9403 | 15451755.62
     154358 | 15212937.88
      38823 | 15064802.86
      85606 | 15053957.15
      33354 | 14408297.40
     154747 | 14407580.68
      82865 | 14235489.78
      76094 | 14094247.04
        222 | 13937777.74
     121271 | 13908336.00
      55221 | 13716120.47
…
     146530 |  7905336.60
      71475 |  7903367.58
      36289 |  7901946.50
      61739 |  7900794.00
      52338 |  7898638.08
     194299 |  7898421.24
     105235 |  7897829.94
      77207 |  7897752.72
      96712 |  7897575.27
      10157 |  7897046.25
     171154 |  7896814.50
      79373 |  7896186.00
     113808 |  7893353.88
      27901 |  7892952.00
     128820 |  7892882.72
      25891 |  7890511.20
     122819 |  7888881.02
     154731 |  7888301.33
     101674 |  7879324.60
      51968 |  7879102.21
      72073 |  7877736.11
       5182 |  7874521.73
(1048 rows)



====================Q12==============


select
      l_shipmode,
      sum(case
            when o_orderpriority = '1-
URGENT'
                  or o_orderpriority =
'2-HIGH'
                  then 1
            else 0
      end) as high_line_count,
      sum(case
            when o_orderpriority <> '1-
URGENT'
                  and o_orderpriority <>
'2-HIGH'
                  then 1
            else 0
      end) as low_line_count
from
      orders,
      lineitem
where
      o_orderkey = l_orderkey
      and l_shipmode in ('MAIL', 'SHIP')
      and l_commitdate < l_receiptdate
      and l_shipdate < l_commitdate
      and l_receiptdate >= date '1994-01-
01'
      and l_receiptdate < cast(date
'1994-01-01' + interval '1 year' as date)
group by
      l_shipmode
order by
      l_shipmode;
 l_shipmode | high_line_count |
low_line_count
------------+-----------------+----------
------
 MAIL       |            6202 |
9324
 SHIP       |            6200 |
9262
(2 rows)
```

```
====================Q13==============


select
      c_count,
      count(*) as custdist
from
      (
            select
                  c_custkey,
                  count(o_orderkey)
            from
                  customer left outer
join orders on
                        c_custkey =
o_custkey
                        and o_comment not
like '%special%requests%'
            group by
                  c_custkey
      ) as c_orders (c_custkey, c_count)
group by
      c_count
order by
      custdist desc,
      c_count desc;
 c_count | custdist
---------+----------
       0 |    50005
       9 |     6641
      10 |     6532
      11 |     6014
       8 |     5937
      12 |     5639
      13 |     5024
      19 |     4793
       7 |     4687
      17 |     4587
      18 |     4529
      20 |     4516
      15 |     4505
      14 |     4446
      16 |     4273
      21 |     4190
      22 |     3623
       6 |     3265
      23 |     3225
      24 |     2742
      25 |     2086
       5 |     1948
      26 |     1612
      27 |     1179
       4 |     1007
      28 |      893
      29 |      593
       3 |      415
      30 |      376
      31 |      226
      32 |      148
       2 |      134
      33 |       75
      34 |       50
      35 |       37
       1 |       17
      36 |       14
      38 |        5
      37 |        5
      40 |        4
      41 |        2
      39 |        1
(42 rows)


====================Q14==============


select
      100.00 * sum(case
            when p_type like 'PROMO%'
                  then l_extendedprice *
(1 - l_discount)
            else 0
      end) / sum(l_extendedprice * (1 -
l_discount)) as promo_revenue
from
      lineitem,
      part
where
      l_partkey = p_partkey
      and l_shipdate >= date '1995-09-01'
      and l_shipdate < cast(date '1995-
09-01' + interval '1 month' as date);
 promo_revenue
---------------
       16.3807
(1 row)



====================Q15==============


select
      s_suppkey,
      s_name,
      s_address,
      s_phone,
      total_revenue
from
      supplier,
      revenue1
where
      s_suppkey = supplier_no
      and total_revenue = (
```

```
            select
                  max(total_revenue)
            from
                  revenue1
      )
order by
      s_suppkey;
CREATE VIEW
 s_suppkey |          s_name           |
s_address     |     s_phone     |
total_revenue
-----------+---------------------------+-
-----------------+-----------------+----
-----------
      8449 | Supplier#000008449        |
Wp34zim9qYFbVctdW | 20-469-856-8873 |
1772627.2087
(1 row)

DROP VIEW


====================Q16===============



select
      p_brand,
      p_type,
      p_size,
      count(distinct ps_suppkey) as
supplier_cnt
from
      partsupp,
      part
where
      p_partkey = ps_partkey
      and p_brand <> 'Brand#45'
      and p_type not like 'MEDIUM
POLISHED%'
      and p_size in (49, 14, 23, 45, 19,
3, 36, 9)
      and ps_suppkey not in (
            select
                  s_suppkey
            from
                  supplier
            where
                  s_comment like
'%Customer%Complaints%'
      )
group by
      p_brand,
      p_type,
      p_size
order by
      supplier_cnt desc,
      p_brand,
      p_type,
      p_size;
```

```
  p_brand   |          p_type           |
p_size | supplier_cnt
------------+---------------------------
+--------+-------------
 Brand#41   | MEDIUM BRUSHED TIN        |
3 |           28
 Brand#54   | STANDARD BRUSHED COPPER   |
14 |           27
 Brand#11   | STANDARD BRUSHED TIN      |
23 |           24
 Brand#11   | STANDARD BURNISHED BRASS  |
36 |           24
 Brand#15   | MEDIUM ANODIZED NICKEL    |
3 |           24
 Brand#15   | SMALL ANODIZED BRASS      |
45 |           24
 Brand#15   | SMALL BURNISHED NICKEL    |
19 |           24
 Brand#21   | MEDIUM ANODIZED COPPER    |
3 |           24
 Brand#22   | SMALL BRUSHED NICKEL      |
3 |           24
 Brand#22   | SMALL BURNISHED BRASS     |
19 |           24
 Brand#25   | MEDIUM BURNISHED COPPER   |
36 |           24
 Brand#31   | PROMO POLISHED COPPER     |
36 |           24
 Brand#33   | LARGE POLISHED TIN        |
23 |           24
 Brand#33   | PROMO POLISHED STEEL      |
14 |           24
 Brand#35   | PROMO BRUSHED NICKEL      |
14 |           24
 Brand#41   | ECONOMY BRUSHED STEEL     |
9 |           24
 Brand#41   | ECONOMY POLISHED TIN      |
19 |           24
 Brand#41   | LARGE PLATED COPPER       |
36 |           24
 Brand#42   | ECONOMY PLATED BRASS      |
3 |           24
 Brand#42   | STANDARD POLISHED TIN     |
49 |           24
Brand#42   | MEDIUM BURNISHED BRASS    |
14 |           20
 Brand#42   | SMALL BURNISHED COPPER    |
3 |           20
 Brand#43   | ECONOMY POLISHED COPPER   |
9 |           20
 Brand#43   | SMALL PLATED STEEL        |
3 |           20
 Brand#43   | STANDARD BURNISHED TIN    |
23 |           20
 Brand#44   | LARGE ANODIZED STEEL      |
23 |           20
 Brand#44   | PROMO ANODIZED TIN        |
23 |           20
```

```
 Brand#51   | ECONOMY BRUSHED BRASS     |
49 |           20
 Brand#51   | ECONOMY POLISHED NICKEL   |
9 |           20
 Brand#51   | MEDIUM BRUSHED TIN        |
9 |           20
 Brand#51   | MEDIUM PLATED BRASS       |
9 |           20
 Brand#51   | PROMO BURNISHED BRASS     |
9 |           20
 Brand#51   | SMALL PLATED NICKEL       |
49 |           20
 Brand#51   | STANDARD ANODIZED NICKEL  |
49 |           20
…
Brand#32   | STANDARD ANODIZED COPPER  |
23 |           3
 Brand#33   | SMALL ANODIZED BRASS      |
9 |           3
 Brand#35   | MEDIUM ANODIZED TIN       |
19 |           3
 Brand#51   | SMALL PLATED BRASS        |
23 |           3
 Brand#52   | MEDIUM BRUSHED BRASS      |
45 |           3
 Brand#53   | MEDIUM BRUSHED TIN        |
45 |           3
 Brand#54   | ECONOMY POLISHED BRASS    |
9 |           3
 Brand#55   | PROMO PLATED BRASS        |
19 |           3
 Brand#55   | STANDARD PLATED TIN       |
49 |           3
(18314 rows)



====================Q17===============


select
      sum(l_extendedprice) / 7.0 as
avg_yearly
from
      lineitem,
      part
where
      p_partkey = l_partkey
      and p_brand = 'Brand#23'
      and p_container = 'MED BOX'
      and l_quantity < (
            select
                  0.2 * avg(l_quantity)
            from
                  lineitem
            where
                  l_partkey = p_partkey
      );
 avg_yearly
-------------
 348406.0542
(1 row)



====================Q18===============


select
      c_name,
      c_custkey,
      o_orderkey,
      o_orderdate,
      o_totalprice,
      sum(l_quantity)
from
      customer,
      orders,
      lineitem
where
      o_orderkey in (
            select
                  l_orderkey
            from
                  lineitem
            group by
                  l_orderkey having
                        sum(l_quantity) >
300
      )
      and c_custkey = o_custkey
      and o_orderkey = l_orderkey
group by
      c_name,
      c_custkey,
      o_orderkey,
      o_orderdate,
      o_totalprice
order by
      o_totalprice desc,
      o_orderdate
limit 100;
       c_name       | c_custkey |
o_orderkey | o_orderdate | o_totalprice |
sum
--------------------+-----------+--------
----+-------------+--------------+-------
-
 Customer#000128120 |    128120 |
4722021 | 1994-04-07  |    544089.09 |
323.00
 Customer#000144617 |    144617 |
3043270 | 1997-02-12  |    530604.44 |
317.00
 Customer#000013940 |     13940 |
2232932 | 1997-04-13  |    522720.61 |
304.00
```

```
 Customer#000066790 |     66790 |
2199712 | 1996-09-30  |    515531.82 |
327.00
 Customer#000046435 |     46435 |
4745607 | 1997-07-03  |    508047.99 |
309.00
 Customer#000015272 |     15272 |
3883783 | 1993-07-28  |    500241.33 |
302.00
 Customer#000146608 |    146608 |
3342468 | 1994-06-12  |    499794.58 |
303.00
 Customer#000096103 |     96103 |
5984582 | 1992-03-16  |    494398.79 |
312.00
 Customer#000024341 |     24341 |
1474818 | 1992-11-15  |    491348.26 |
302.00
 Customer#000137446 |    137446 |
5489475 | 1997-05-23  |    487763.25 |
311.00
…
 Customer#000093392 |     93392 |
5200102 | 1997-01-22  |    425487.51 |
304.00
 Customer#000015631 |     15631 |
1845057 | 1994-05-12  |    419879.59 |
302.00
 Customer#000112987 |    112987 |
4439686 | 1996-09-17  |    418161.49 |
305.00
 Customer#000012599 |     12599 |
4259524 | 1998-02-12  |    415200.61 |
304.00
 Customer#000105410 |    105410 |
4478371 | 1996-03-05  |    412754.51 |
302.00
 Customer#000149842 |    149842 |
5156581 | 1994-05-30  |    411329.35 |
302.00
 Customer#000010129 |     10129 |
5849444 | 1994-03-21  |    409129.85 |
309.00
 Customer#000069904 |     69904 |
1742403 | 1996-10-19  |    408513.00 |
305.00
 Customer#000017746 |     17746 |
6882 | 1997-04-09  |    408446.93 |
303.00
 Customer#000013072 |     13072 |
1481925 | 1998-03-15  |    399195.47 |
301.00
 Customer#000082441 |     82441 |
857959 | 1994-02-07  |    382579.74 |
305.00
 Customer#000088703 |     88703 |
2995076 | 1994-01-30  |    363812.12 |
302.00
(57 rows)
```

```
====================Q19==============


select
      sum(l_extendedprice* (1 -
l_discount)) as revenue
from
      lineitem,
      part
where
      (
            p_partkey = l_partkey
            and p_brand = 'Brand#12'
            and p_container in ('SM
CASE', 'SM BOX', 'SM PACK', 'SM PKG')
            and l_quantity >= 1 and
l_quantity <= 1 + 10
            and p_size between 1 and 5
            and l_shipmode in ('AIR',
'AIR REG')
            and l_shipinstruct = 'DELIVER
IN PERSON'
      )
      or
      (
            p_partkey = l_partkey
            and p_brand = 'Brand#23'
            and p_container in ('MED
BAG', 'MED BOX', 'MED PKG', 'MED PACK')
            and l_quantity >= 10 and
l_quantity <= 10 + 10
            and p_size between 1 and 10
            and l_shipmode in ('AIR',
'AIR REG')
            and l_shipinstruct = 'DELIVER
IN PERSON'
      )
      or
      (
            p_partkey = l_partkey
            and p_brand = 'Brand#34'
            and p_container in ('LG
CASE', 'LG BOX', 'LG PACK', 'LG PKG')
            and l_quantity >= 20 and
l_quantity <= 20 + 10
            and p_size between 1 and 15
            and l_shipmode in ('AIR',
'AIR REG')
            and l_shipinstruct = 'DELIVER
IN PERSON'
      );
   revenue
--------------
 3083843.0578
(1 row)
```

```
====================Q20==============


select
      s_name,
      s_address
from
      supplier,
      nation
where
      s_suppkey in (
            select
                  ps_suppkey
            from
                  partsupp
            where
                  ps_partkey in (
                        select
                              p_partkey
                        from
                              part
                        where
                              p_name like
'forest%'
                  )
                  and ps_availqty > (
                        select
                              0.5 *
sum(l_quantity)
                        from
                              lineitem
                        where
                              l_partkey =
ps_partkey
                              and
l_suppkey = ps_suppkey
                              and
l_shipdate >= date '1994-01-01'
                              and
l_shipdate < cast(date '1994-01-01' +
interval '1 year' as date)
                  )
      )
      and s_nationkey = n_nationkey
      and n_name = 'CANADA'
order by
      s_name;
          s_name           |
s_address
---------------------------+-------------
----------------------------
 Supplier#000000020        |
iybAE,RmTymrZVYaFZva2SH,j
 Supplier#000000091        |
YV45D7TkfdQanOOZ7q9QxkyGUapU1oOWU6q3
 Supplier#000000197        |
YC2Acon6kjY3zj3Fbxs2k4Vdf7X0cd2F
 Supplier#000000226        |
83qOdU2EYRdPQAQhEtn GRZEd
 Supplier#000000285        |
Br7e1nnt1yxrw6ImgpJ7YdhFDjuBf
 Supplier#000000378        |
FfbhyCxWvcPrO8ltp9
 Supplier#000000402        |
i9Sw4DoyMhzhKXCH9By,AYSgmD
 Supplier#000000530        | 0qwCMwobKY
OcmLyfRXlagA8ukENJv,
 Supplier#000000688        | D
fw5ocppmZpYBBIPI718hCihLDZ5KhKX
 Supplier#000000710        | f19YPvOyb
QoYwjKC,oPycpGfieBAcwKJo
 Supplier#000000736        |
l6i2nMwVuovfKnuVgaSGK2rDy65DlAFLegiL7
 Supplier#000000761        |
zlSLelQUj2XrvTTFnv7WAcYZGvvMTx882d4
 Supplier#000000884        | bmhEShejaS
 Supplier#000000887        |
urEaTejH5POADP2ARrf
 Supplier#000000935        | ij98czM
2KzWe7dDTOxB8sq0UfCdvrX
…
ZKyj3xmAVWC1XdP ue1p5m,i
 Supplier#000009278        |
RqYTzgxj93CLX 0mcYfCENOefD
 Supplier#000009327        |
uoqMdf7e7Gj9dbQ53
 Supplier#000009430        | igRqmneFt
 Supplier#000009567        |
r4Wfx4c3xsEAjcGj71HHZByornl D9vrztXlv4
 Supplier#000009601        |
51m637bO,Rw5DnHWFUvLacRx9
 Supplier#000009709        |
rRnCbHYgDgl9PZYnyWKVYSUW0vKg
 Supplier#000009753        |
wLhVEcRmd7PkJF4FBnGK7Z
 Supplier#000009796        |
z,y4Idmr15DOvPUqYG
 Supplier#000009799        |
4wNjXGa4OKWl
 Supplier#000009811        |
E3iuyq7UnZxU7oPZIe2Gu6
 Supplier#000009812        |
APFRMy3lCbgFga53n5t9DxzFPQPgnjrGt32
 Supplier#000009862        | rJzweWeN58
 Supplier#000009868        |
ROjGgx5gvtkmnUUoeyy7v
 Supplier#000009869        |
ucLqxzrpBTRMewGSM29t0rNTM30g1Tu3Xgg3mKag
 Supplier#000009899        |
7XdpAHrzr1t,UQFZE
 Supplier#000009974        |
7wJ,J5DKcxSU4Kp1cQLpbcAvB5AsvKT
(204 rows)
```

```
====================Q21===============


select
      s_name,
      count(*) as numwait
from
      supplier,
      lineitem l1,
      orders,
      nation
where
      s_suppkey = l1.l_suppkey
      and o_orderkey = l1.l_orderkey
      and o_orderstatus = 'F'
      and l1.l_receiptdate >
l1.l_commitdate
      and exists (
            select
                  *
            from
                  lineitem l2
            where
                  l2.l_orderkey =
l1.l_orderkey
                  and l2.l_suppkey <>
l1.l_suppkey
      )
      and not exists (
            select
                  *
            from
                  lineitem l3
            where
                  l3.l_orderkey =
l1.l_orderkey
                  and l3.l_suppkey <>
l1.l_suppkey
                  and l3.l_receiptdate >
l3.l_commitdate
      )
      and s_nationkey = n_nationkey
      and n_name = 'SAUDI ARABIA'
group by
      s_name
order by
      numwait desc,
      s_name
limit 100;
          s_name           | numwait
---------------------------+---------
 Supplier#000002829        |      20
 Supplier#000005808        |      18
 Supplier#000000262        |      17
 Supplier#000000496        |      17
 Supplier#000002160        |      17
 Supplier#000002301        |      17
 Supplier#000002540        |      17
 Supplier#000003063        |      17
 Supplier#000005178        |      17
 Supplier#000008331        |      17
 Supplier#000002005        |      16
 Supplier#000002095        |      16
 Supplier#000005799        |      16
 Supplier#000005842        |      16
 Supplier#000006450        |      16
 Supplier#000006939        |      16
 Supplier#000009200        |      16
 Supplier#000009727        |      16
 Supplier#000000486        |      15
 Supplier#000000565        |      15
 Supplier#000001046        |      15
…
 Supplier#000007602        |      13
 Supplier#000008134        |      13
 Supplier#000008234        |      13
 Supplier#000009435        |      13
 Supplier#000009436        |      13
 Supplier#000009564        |      13
 Supplier#000009896        |      13
 Supplier#000000379        |      12
 Supplier#000000673        |      12
 Supplier#000000762        |      12
 Supplier#000000811        |      12
 Supplier#000000821        |      12
 Supplier#000001337        |      12
 Supplier#000001916        |      12
 Supplier#000001925        |      12
 Supplier#000002039        |      12
 Supplier#000002357        |      12
 Supplier#000002483        |      12
(100 rows)


====================Q22===============


select
      cntrycode,
      count(*) as numcust,
      sum(c_acctbal) as totacctbal
from
      (
            select
                  substring(c_phone from
1 for 2) as cntrycode,
                  c_acctbal
            from
                  customer
            where
                  substring(c_phone from
1 for 2) in
                        ('13', '31',
'23', '29', '30', '18', '17')
                  and c_acctbal > (
                        select
```

```
      avg(c_acctbal)
                        from
                              customer
                        where
                              c_acctbal >
0.00
                              and
substring(c_phone from 1 for 2) in

      ('13', '31', '23', '29', '30',
'18', '17')
                  )
                  and not exists (
                        select
                              *
                        from
                              orders
                        where
                              o_custkey =
c_custkey
                  )
      ) as custsale
group by
      cntrycode
order by
      cntrycode;
 cntrycode | numcust | totacctbal
-----------+---------+------------
 13        |     888 | 6737713.99
 17        |     861 | 6460573.72
 18        |     964 | 7236687.40
 23        |     892 | 6701457.95
 29        |     948 | 7158866.63
 30        |     909 | 6808436.13
 31        |     922 | 6806670.18
(7 rows)
```

Appendix D: Seeds and Query Parameters

Seeds:

```
start seed = 322215657
stream00: 322215657
stream01: 322215658
stream02: 322215659
stream03: 322215660
stream04: 322215661
stream05: 322215662
stream06: 322215663
stream07: 322215664
```

Query Parameters

```
============== Stream 0 ==============
14    1997-04-01
2     17    STEEL ASIA
9     midnight
20    metallic    1996-01-01  UNITED
KINGDOM
6     1997-01-01  0.08  25
17    Brand#15    MED JAR
18    315
8     PERU  AMERICA     LARGE BURNISHED
STEEL
21    CHINA
13    pending     deposits
3     FURNITURE   1995-03-14
22    24    22    23    19    27    21
      15
16    Brand#41    ECONOMY BRUSHED   43
      28    16    11    4     30    39
      35
4     1995-02-01
11    KENYA 0.0000001000
15    1996-11-01
1     110
10    1994-10-01
19    Brand#33    Brand#45    Brand#33
      3     17    20
5     AFRICA      1997-01-01
7     CHINA PERU
12    SHIP  RAIL  1997-01-01


============== Stream 1 ==============
21    IRAN
3     AUTOMOBILE  1995-03-30
18    312
5     AMERICA     1997-01-01
11    BRAZIL      0.0000001000
7     IRAN  INDONESIA
6     1997-01-01  0.05  24
20    violet      1995-01-01  JORDAN
17    Brand#11    MED PKG
12    FOB   RAIL  1997-01-01
16    Brand#31    SMALL ANODIZED    33
      11    14    13    19    3     40
      37
15    1994-08-01
13    pending     deposits
10    1993-07-01
2     4     BRASS MIDDLE EAST
8     INDONESIA   ASIA  MEDIUM BRUSHED
COPPER
14    1997-07-01
19    Brand#35    Brand#33    Brand#23
      8     18    27
9     lime
22    23    31    20    10    15    27
      18
1     118
4     1997-09-01

============== Stream 2 ==============
6     1993-01-01  0.02  24
17    Brand#13    JUMBO CASE
14    1997-11-01
16    Brand#11    LARGE PLATED      1
      20    5     35    39    13    3
      45
19    Brand#42    Brand#11    Brand#22
      4     19    24
10    1994-05-01
9     khaki
2     42    NICKEL      ASIA
15    1997-03-01
8     ARGENTINA   AMERICA     MEDIUM
PLATED COPPER
5     ASIA  1993-01-01
22    12    13    16    11    32    25
      30
12    MAIL  RAIL  1997-01-01
7     BRAZIL      ARGENTINA
13    pending     deposits
18    314
1     65
4     1995-06-01
20    honeydew    1993-01-01  BRAZIL
3     FURNITURE   1995-03-16
11    MOROCCO     0.0000001000
21    BRAZIL

============== Stream 3 ==============
8     CHINA ASIA  MEDIUM ANODIZED COPPER
5     EUROPE      1993-01-01
```

```
4     1993-03-01
6     1993-01-01  0.08  25
17    Brand#15    JUMBO JAR
7     ROMANIA     CHINA
1     73
18    315
22    11    15    12    31    17    16
      26
14    1993-02-01
9     green
10    1993-02-01
15    1994-12-01
11    CANADA      0.0000001000
20    saddle      1996-01-01  PERU
2     30    TIN   AFRICA
21    ROMANIA
19    Brand#45    Brand#44    Brand#21
      9     20    20
13    pending     packages
16    Brand#41    PROMO POLISHED    34
      9     23    1     6     15    43
      49
12    TRUCK RAIL  1993-01-01
3     MACHINERY   1995-03-01

============== Stream 4 ==============
5     MIDDLE EAST 1993-01-01
21    IRAQ
14    1993-05-01
19    Brand#42    Brand#32    Brand#15
      4     10    27
15    1997-06-01
17    Brand#12    JUMBO PKG
12    AIR   TRUCK 1993-01-01
6     1993-01-01  0.05  24
4     1995-10-01
9     floral
8     IRAN  MIDDLE EAST SMALL POLISHED
COPPER
16    Brand#31    MEDIUM ANODIZED   45
      11    24    23    9     12    27
      10
11    MOZAMBIQUE  0.0000001000
2     18    STEEL ASIA
10    1993-11-01
18    313
1     81
13    pending     packages
7     IRAQ  IRAN
22    17    11    18    23    13    22
      25
3     FURNITURE   1995-03-18
20    cyan  1995-01-01  GERMANY

============== Stream 5 ==============
21    CANADA
15    1995-03-01
4     1993-07-01
6     1993-01-01  0.03  24
7     CANADA      BRAZIL
16    Brand#11    ECONOMY BURNISHED 41
      28    8     22    25    11    4
      15
19    Brand#54    Brand#15    Brand#14
      9     11    23
18    314
14    1993-08-01
22    15    22    31    16    10    18
      26
11    EGYPT 0.0000001000
13    pending     packages
3     MACHINERY   1995-03-03
1     89
2     6     BRASS AFRICA
5     AFRICA      1993-01-01
8     BRAZIL      AMERICA     SMALL
BURNISHED COPPER
20    orange      1993-01-01  RUSSIA
12    REG AIR     TRUCK 1993-01-01
17    Brand#14    WRAP CASE
10    1994-08-01
9     dark

============== Stream 6 ==============
10    1993-05-01
3     BUILDING    1995-03-20
15    1997-10-01
13    unusual     packages
6     1994-01-01  0.08  25
8     ROMANIA     EUROPE      SMALL
ANODIZED TIN
9     chocolate
7     SAUDI ARABIA      ROMANIA
4     1996-02-01
11    PERU  0.0000001000
22    19    21    25    30    16    26
      17
18    312
12    SHIP  TRUCK 1993-01-01
1     97
5     AMERICA     1994-01-01
16    Brand#41    STANDARD POLISHED 21
      25    42    24    3     32    11
      16
2     43    NICKEL      EUROPE
14    1993-11-01
19    Brand#51    Brand#53    Brand#14
      5     12    30
20    beige 1997-01-01  IRAQ
17    Brand#11    WRAP JAR
21    VIETNAM

============== Stream 7 ==============
18    313
8     IRAQ  MIDDLE EAST STANDARD POLISHED
TIN
20    lemon 1995-01-01  ARGENTINA
21    JORDAN
```

```
2     31    TIN   AFRICA
4     1993-11-01
22    13    33    20    23    19    26
      34
17    Brand#13    WRAP PKG
1     105
11    ETHIOPIA    0.0000001000
9     blush
19    Brand#53    Brand#31    Brand#53
      10    13    27
3     MACHINERY   1995-03-05
13    unusual     packages
5     EUROPE      1994-01-01
7     JAPAN IRAQ
10    1994-03-01
16    Brand#31    MEDIUM BRUSHED    14
      34    4     27    35    19    17
      37
6     1994-01-01  0.06  24
14    1994-03-01
15    1995-06-01
12    FOB   TRUCK 1994-01-01
```

Appendix E: Price Quotes

<page blank>

Quote

Quote Prepared By: Azevedo, Mike
Email: mike.azevedo@paraccel.com
Quote #: VMW1TBTPC-01
Date: 24-Aug-2010
Expires 23-Sep-2010

3 Results Way
Suite 450
San Diego, CA 92121

Prepared For:	**Bill To:**	**Ship To:**	**Install:**

Line #	Qty	Product ID	Description	Units	Total List Price (USD)	Discount	Total Deal Price (USD)	Extended Price
		ParAccel Analytic Database (PADB)						
1	1	PADB-V250	1TB user data	EA	$ 400,000	40%	$ 240,000	$ 240,000
			Software Sub-total					**$ 240,000**
1	0		SOFTWARE INSTALLATION (per day)	EA	$ 2,000	$ -	$ 2,000	$ -
2	0		CONSULTING SERVICES (per day)	EA	$ 2,000	$ -	$ 2,000	$ -
			Services Sub-total					**0**
1	3	PADB-V250-S	24/7 Software Support (per yr)	EA	$ 96,000	40%	$ 57,600	$ 172,800
			Maintenance and Support Sub-total				**$ 57,600**	**$ 172,800**
			Proposal Summary					**$ 412,800**
			Software Summary					**$ 240,000**
			Services Summary					**$ -**
			Support Summary					**$ 172,800**

CDW.com | 800.800.4239

OE400SPS

SALES QUOTATION

QUOTE NO.	ACCOUNT NO.	DATE
BNHP880	10000017	3/15/2010

BILL TO:
PARACCEL
9920 PACIFIC HEIGHTS BLVD

Accounts Payable
SAN DIEGO , CA 92121-4396

Customer Phone #408.896.9140

SHIP TO:
PARACCEL
Attention To: MATT DIENER
9920 PACIFIC HEIGHTS BLVD

SAN DIEGO , CA 92121-4396
Contact: MATT DIENER 858.309.4733

Customer P.O. # VMW_POC QUOTE

ACCOUNT MANAGER	SHIPPING METHOD	TERMS	EXEMPTION CERTIFICATE
HANNAH COHEE 877.299.3225	FedEx Ground - Cust Acct	NET 30-VERBAL	

QTY	ITEM NO.	DESCRIPTION	UNIT PRICE	EXTENDED PRICE
40	1723258	HP DL380 G6 X5560 PERF Mfg#: 491315-001	4,889.00	195,560.00
720	1723304	HP 4GB 2RX4 PC3-10600R-9 KIT Mfg#: 500658-B21	196.40	141,408.00
320	1640515	HP 300GB 3G PLUG SAS 10K DP SFF HD Mfg#: 492620-B21	408.25	130,640.00
40	1578335	HP NC382T PCIE DP GBIT SVR ADPTR Mfg#: 458492-B21	176.00	7,040.00
40	1723301	HP DL380 G6 3 SLOT PCI-E RISER KIT Mfg#: 500579-B21	85.00	3,400.00
40	635101	HPE 3YR 24X7X4 DL380 Mfg#: U4545E Electronic distribution - NO MEDIA	738.00	29,520.00
40	1706188	CDW HARDWARE INSTALL SERVER Mfg#: HWINSTALLSERVER	25.00	1,000.00
40	1625768	PRIORITY SERVICE FOR INSTALL Mfg#: PRIORITY-INSTALL	12.50	500.00
40	1644616	HP ILO ADV 1 SRV TSU SW Mfg#: 512485-B21	279.00	11,160.00
480	630144	BELKIN 10FTCAT5E RJ45M PATCH BLUE Mfg#: A3L781-10BL-CDW	2.35	1,128.00
4	722414	CISCO CATALYST 3750 48PT 10/100/1000 Mfg#: WS-C3750G-48TS-S	8,579.00	34,316.00
4	963921	CISCO SMARTNET OS 24X7X4 Mfg#: CON-OSP-3750G48T Electronic distribution - NO MEDIA	1,665.00	6,660.00
3	752557	BELKIN ENTERPRISE 4 POST RACK 42U 7' Mfg#: RK4000	425.00	1,275.00
5	1144299	APC RACK PDU METERED 2U 30A 208V Mfg#: AP7811	535.00	2,675.00
1	1795176	ACER V173 BB 17" Mfg#: ET.BV3RP.B01	115.00	115.00
1	1059780	LOGITECH DELUXE 250 DT KB&MOU BLK Mfg#: 967973-0403	19.00	19.00
1	654810	RECYCLING FEE 15" TO LESS THAN 35" Fee Applied to Item: 1795176	16.00	16.00

----------SPECIAL INSTRUCTIONS----------
This quote is valid until 04/30/10.

SUBTOTAL	566,432.00
FREIGHT	0.00
TAX	46,264.43

US Currency

TOTAL 612,696.43

CDW Direct
200 North Milwaukee Ave.
Vernon Hills, IL 60061
Phone: 847.465.6000
Fax: 847.968.1653

Please remit payment to:
CDW Direct
P.O. Box 75723
Chicago, IL 60675-5723