z13 Capacity Planning
(Part 2)

Fabio Massimo Ottaviani – EPV Technologies

February 2015

6 IBM zPCR

A new version (V8.7a) of the IBM zPCR free tool, supporting the z13 machines, is already available on the web. You can download it at:
https://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1381.

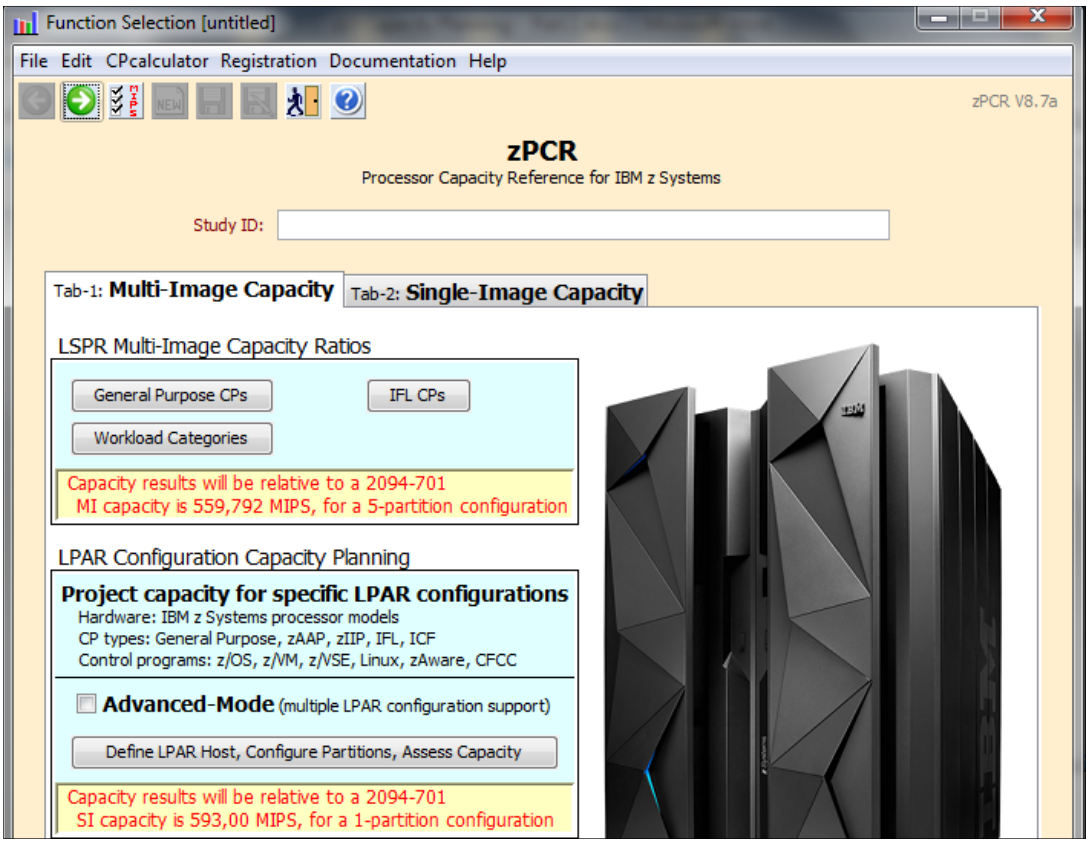

Figure 7

zPCR is a "must have" tool for capacity planners. With it you can estimate the capacity of a machine taking into consideration LPAR configuration, operative system level and workload characteristics.

As you can see in Figure 7 the reference CPU is still a 2094-701 estimated at 593,00 MIPS. This is the base of zPCR capacity studies and also the base for the ratio or MIPS estimates provided in the LSPR Multi-Image Capacity table.

It's worth noting that the zPCR MIPS values are a bit more precise than the ones you can calculate starting from LSPR benchmarks and provided in the Appendix to the first part of this paper.

A snapshot of the table is provided in Figure 8.

LSPR Capacity Ratio Table

Workload Graph Help

zPCR V8.7a

z/OS-2.1 LSPR Data (01/14/2015)

LSPR Multi-Image Capacity Ratios

General Purpose CPs

Values are applicable for z/OS; representative of z/VM and Linux

Capacity basis: 2094-701 @ 559,792 MIPS for a typical multi-partition configuration

Capacity for z/OS on z10 and later processors is represented with HiperDispatch turned ON

Processor	Features	Flag	MSU	LSPR Workload Category				
				Low	Low-Avg	Average	Avg-High	High
z13/700								
2964-701	1W	=	210	1.779	1.736	1.695	1.614	1.540
2964-702	2W	=	394	3.452	3.319	3.196	3.003	2.833
2964-703	3W	=	571	5.085	4.854	4.644	4.340	4.073
2964-704	4W	=	740	6.678	6.344	6.041	5.625	5.262
2964-705	5W	=	905	8.238	7.792	7.392	6.866	6.410
2964-706	6W	=	1.062	9.765	9.202	8.700	8.066	7.518
2964-707	7W	=	1.212	11.260	10.573	9.964	9.224	8.587
2964-708	8W	=	1.356	12.724	11.906	11.188	10.344	9.618
2964-709	9W	=	1.496	14.157	13.204	12.371	11.425	10.613
2964-710	10W	=	1.632	15.560	14.466	13.515	12.469	11.574
2964-711	11W	=	1.764	16.933	15.693	14.622	13.479	12.501
2964-712	12W	=	1.891	18.278	16.887	15.693	14.453	13.395
2964-713	13W	=	2.011	19.594	18.049	16.729	15.395	14.258
2964-714	14W	=	2.129	20.883	19.178	17.731	16.305	15.091
2964-715	15W	=	2.244	22.144	20.277	18.700	17.184	15.895
2964-716	16W	=	2.358	23.400	21.371	19.665	18.058	16.695
2964-717	17W	=	2.472	24.650	22.458	20.624	18.929	17.490
2964-718	18W	=	2.584	25.895	23.541	21.579	19.794	18.282
2964-719	19W	=	2.695	27.134	24.618	22.529	20.656	19.070

Figure 8

As you can see zPCR also provides Low-Avg and Avg-High values which are calculated as an harmonic mean of the Low, Average and High RNI benchmarks. They should be used when workload characteristics are on the border between Low and Average RNI or between Average and High RNI.

To understand which benchmark best represents your system workload you need to collect the hardware measurement facility counters (recorded in SMF 113) and pass them as input to zPCR which will automatically select the appropriate LSPR benchmark[1].

A better solution is collecting SMF 113 in a tool, such as EPV for z/OS, and analysing system workload behaviour in multiple days and at different times of the day.

Whatever method you choose, the benchmark to use depends on the number of misses in the Level 1 cache and on the RNI values of the system workload. Starting from these two values you can classify it by using the rules in Figure 9.

[1] The z13 processor cache architecture and the hardware measurement facility counters will be discussed in the third part of this paper.

%L1 Miss	RNI	Benchmark
< 3%	>= 0,75	AVG
< 3%	< 0,75	Low
3% to 6%	> 1,00	High
3% to 6%	0,60 to 1,00	AVG
3% to 6%	< 0,60	Low
> 6 %	>= 0,75	High
> 6 %	< 0,75	AVG

Figure 9

Beside z13 support, the biggest change introduced in this zPCR version is the possibility to take into account the effect of Simultaneous Multi-Threading (SMT) on zIIP engines.

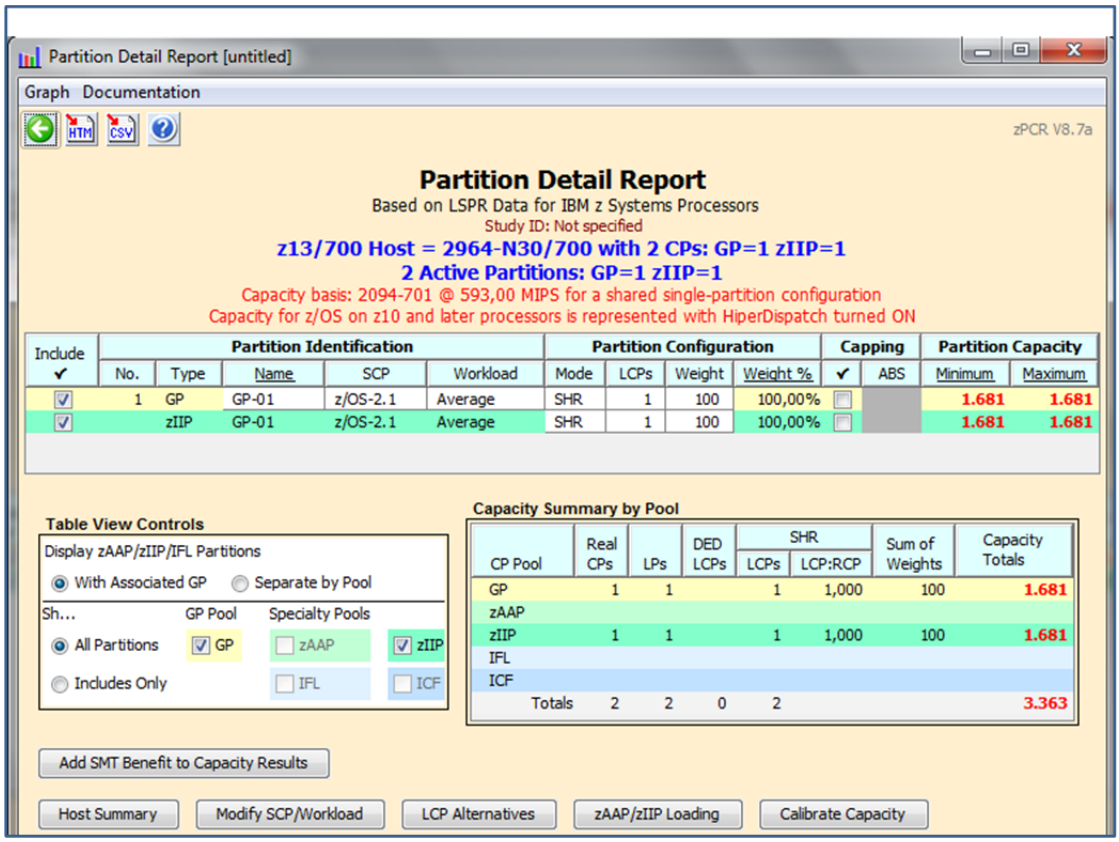

Figure 10

A new option button to“Add SMT benefit to Capacity Results” is provided in the Partition Detail Report.
When clicking it, a small box appears allowing you to choose if you want to apply a capacity increase due to SMT to zIIP, IFL or both. Default values are 25% for zIIP and 20% for IFL. Of course you can change them.

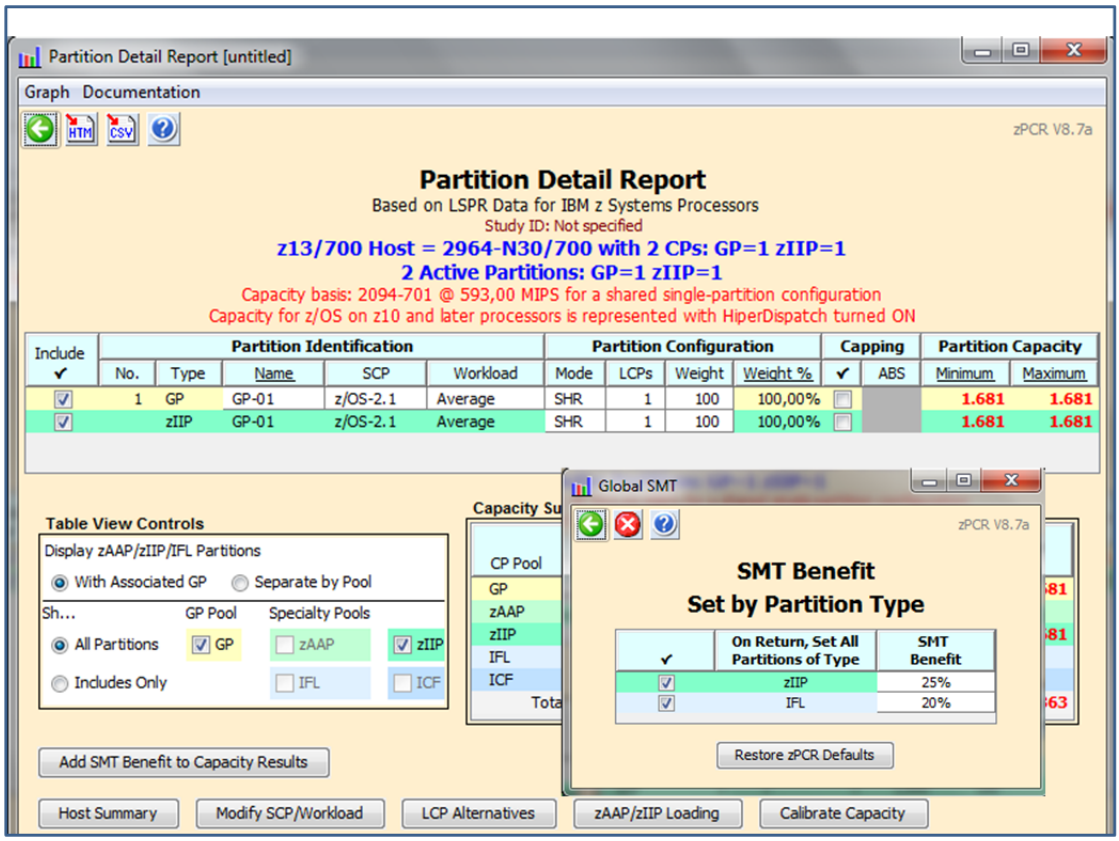

Figure 11

In Figure 12 you can see the result: zIIP capacity increased from 1.681 to 2.102 MIPS.

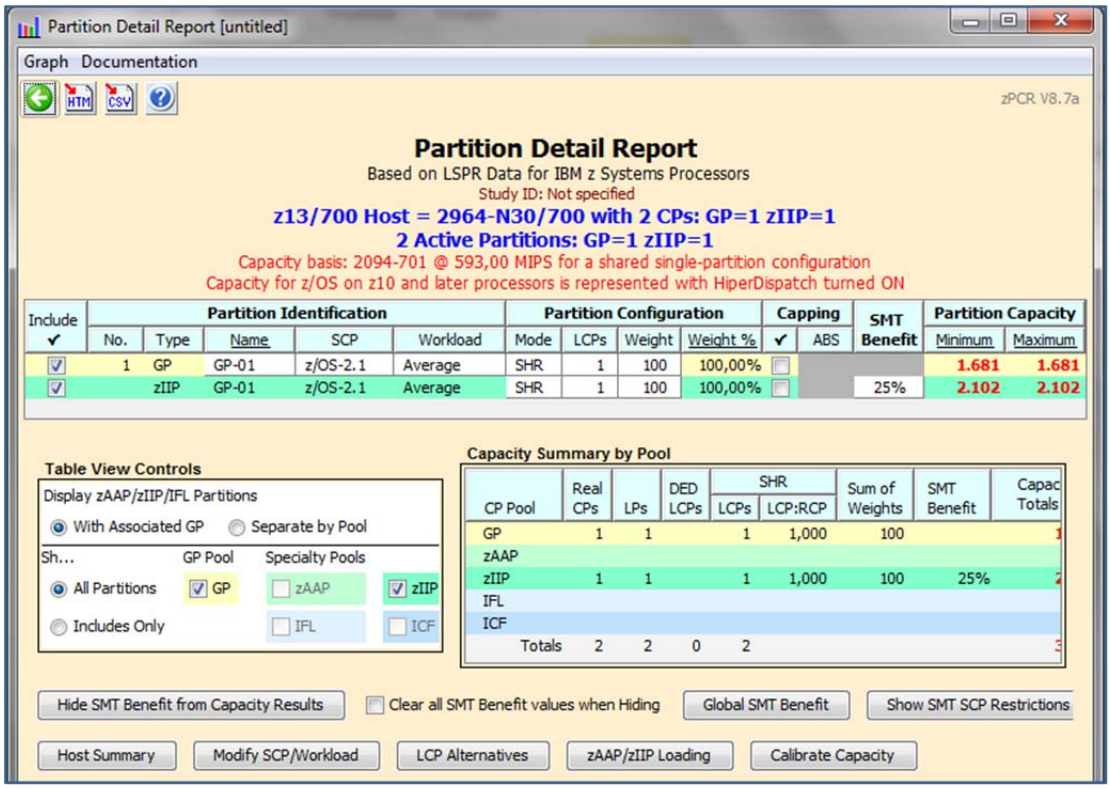

Figure 12

7 z13 Simultaneous Multi-Threading[2]

7.1 CPU or core?

Originally the CPUs (hardware chip) had a single central processing unit on it. So the term “CPU” was used to indicate both of them.

To increase performance, manufacturers started to increase the number of central processing units in a chip. They called them cores. A multi-core chip appears to the operating system (e.g. z/OS) as multiple processing units which can be used by different processes at the same time. This is what is relevant from a measurement and performance analysis perspective.

Mainframe machines have exploited multi-core chips for many years so we should be accustomed to the term “core”. In reality all mainframe commands, tools, manuals and people still use the term “CPU” to indicate a core.

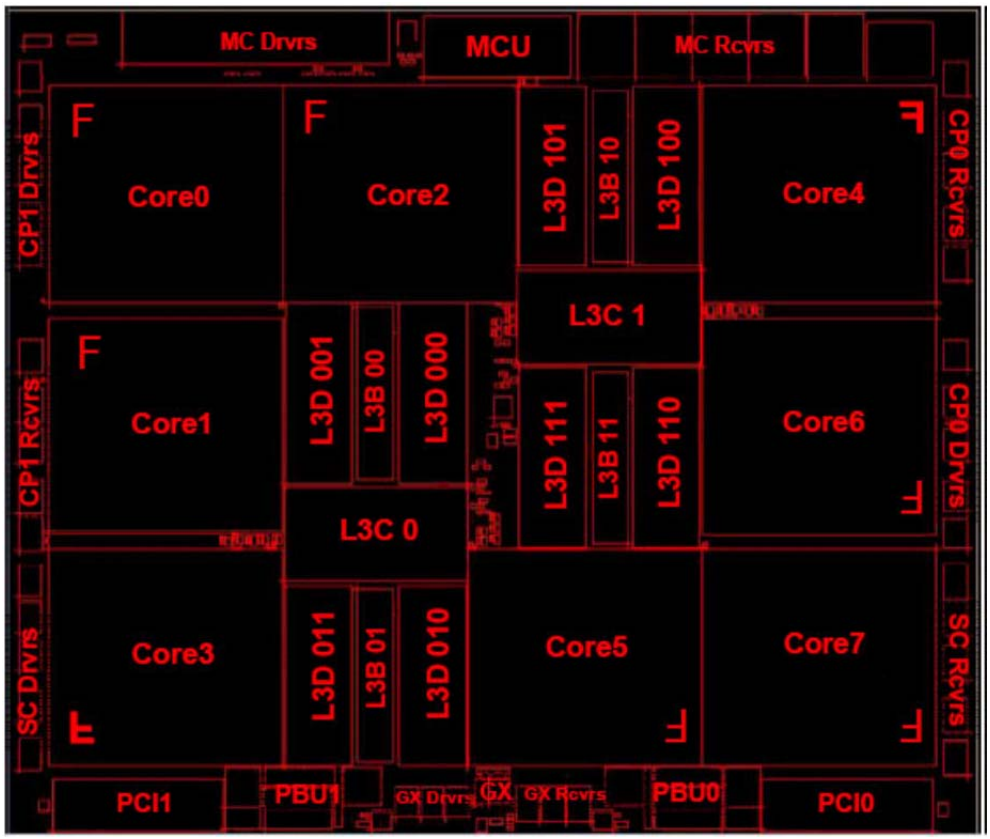

Figure 13

In the figure above you can see the structure of the z13 PU Single Chip Module (from “IBM z13 Technical Guide”). Eight cores are hosted on the SCM.

[2] Most of the content of this chapter has been inspired by “Simultaneous Multithreading and System z” written by Bob Rogers and published in number 3-2014 of Cheryl Watson’s TUNING Letter.

7.2 Advantages and issues of SMT

Mainframe cores process instructions in multiple pipes composed of a number of stages each performing one step in the processing of an instruction, similar to an assembly line. However a traditional core can operate on a single instruction stream.

A big part of the core capacity is normally wasted when an instruction stream gets stalled waiting for a cache miss to be resolved. To address this issue with z13 machines IBM decided to start exploiting Simultaneous Multi-Threading (SMT).
By using SMT multiple instruction streams can be processed simultaneously so when a thread is waiting for a cache miss the core can continue doing work on behalf of the other threads.

Unfortunately, the additional throughput from SMT does not scale very well with the number of threads. This is because all the threads on a core share some limited resources (e.g. pipes, processor cache, TLB).
We saw in the previous chapter that the default expected increase of zIIP capacity when using SMT-2 (two threads) is only 25% in zPCR.[3]

As already mentioned IBM has been very cautious with SMT on z13: only SMT-2 can be used and only on zIIP and IFL.
The reason of this approach is that, while SMT may generally increase the overall throughput, it introduces some important issues.

a. Reduced speed; a thread in an SMT environment is slower than a thread using a dedicated core; the main reason is the fact that the Level 1 and Level 2 caches are shared among the threads; the effect on the application is similar to running on more but slower engines; the more threads the stronger the effect.

b. Throughput variability; as discussed in the first part of this paper, variability has been increasing with each new mainframe model as the processor designs get ever more complex. With SMT that variability will increase much more because the throughput will also depend on the characteristics of the threads sharing the core. If all threads need the whole Level 1 cache, throughput could be even worse than running without SMT. On the other hand if all threads have a small Level 1 cache footprint the overall throughput could be up to 100% more (with SMT-2) than running without SMT.

c. zIIP measurements; all the zIIP measurements have to be reviewed. The current CPU timer implementation accounts processor time both when using the processor and when waiting (normally for a Level 1 cache miss); using it with SMT, the time waiting for other threads will be accounted as processor time too. Even zIIP busy may become tricky: if we have only one zIIP core, only one thread is running at 100% busy and we use SMT-2 we could say that the overall zIIP busy is 50% because we have another thread to use. But if we assume that activating the second thread the maximum throughput increase we can get is about 25%, we should say that zIIP busy is 80% because by adding 20% (80%*25%) more work we will reach 100% busy[4].

[3] On P7 machines the average throughput increase with SMT-2 is about 40%; it will probably be about the same on the mainframe.
[4] A solution to this issue has been implemented on P7 machines.

7.3 Settings and commands

To activate the SMT-2 function on z/OS, you have to:

- define the PROCVIEW CORE option in LOADxx; if you do not want to use SMT-2 you can omit the PROCVIEW parameter or specify PROCVIEW CPU which is the default;
- set MT_ZIIP_MODE=2 in IEAOPTxx.

When you define PROCVIEW CORE, you cannot use the word CPU in z/OS commands. You must use CORE instead of CPU. If you want to continue to use CPU in z/OS commands, you have to define PROCVIEW CORE,CPU_OK. This parameter causes z/OS to treat CPU as an acceptable alias for CORE.

```
D M=CORE
IEE174I 14.54.21 DISPLAY M 902
CORE STATUS: HD=Y   MT=2  MT_MODE: CP=1  zIIP=2
ID    ST   ID RANGE   VP  ISCM  CPU THREAD STATUS
0000   +   0000-0001  H   0000  +N
0001   +   0002-0003  H   0000  +N
0002   +   0004-0005  H   0000  +N
0003   +   0006-0007  H   FC00  +N
0004   +I  0008-0009  H   0200  ++
0005   -   000A-000B
0006   -   000C-000D
0007   -   000E-000F
0008   -   0010-0011
0009   -I  0012-0013

CPC ND = 002964.N63.IBM.02.00000008DA87
CPC SI = 2964.735.IBM.02.000000000008DA87
         Model: N63
CPC ID = 00
CPC NAME = SCZP501
LP NAME = A01        LP ID =  1
CSS ID  = 0
MIF ID  = 1

+ ONLINE    - OFFLINE    N NOT AVAILABLE    / MIXED STATE
W WLM-MANAGED

I        INTEGRATED INFORMATION PROCESSOR (zIIP)
CPC ND  CENTRAL PROCESSING COMPLEX NODE DESCRIPTOR
CPC SI  SYSTEM INFORMATION FROM STSI INSTRUCTION
CPC ID  CENTRAL PROCESSING COMPLEX IDENTIFIER
CPC NAME CENTRAL PROCESSING COMPLEX NAME
LP NAME  LOGICAL PARTITION NAME
LP ID    LOGICAL PARTITION IDENTIFIER
CSS ID   CHANNEL SUBSYSTEM IDENTIFIER
MIF ID   MULTIPLE IMAGE FACILITY IMAGE IDENTIFIER
```

Figure 14

You can see that the output of D M=CORE is quite different from the output of D M=CPU[5].
For each CORE ID there is a range with two ids and each thread appears as a logical processor to z/OS when SMT-2 is used as you can see in the CPU column of CORE ID 0004 (online zIIP).

z13 processor cache architecture and hardware measurement facility counters will be discussed in the third part of this paper

[5] From “IBM z13 Configuration Setup”.