1 Introduction

Experienced capacity planners know that every new generation of machines provides a major challenge to their skills. However, by using LSPR benchmarks and the recent zPCR tool, they have always been able to overcome this issue.

Unfortunately the latest and only benchmarks for z/OS 1.11 are Low, Average and High RNI (forget about LoIO-Mix, Di-Mix, etc) and these are the only benchmarks available for the z196 mainframes. Any upgrade from z9 and z10 to these new machines will therefore require special care.

This paper discusses these issues providing information and suggestions on how to perform accurate machine capacity evaluations and comparisons.

This paper will also give a preliminary description of the LSPR benchmarks, z10 and z196 processor hardware architecture and the new metrics available through the CPU Measurement Facility which allows you to measure each processor cache level and determine the real capacity of a machine.
2 Old LSPR benchmarks

IBM’s Large Systems Performance Reference (LSPR) is a method designed to provide relative processor capacity data for IBM mainframes. It is based on a set of measured benchmarks representing specific workload environments. These base benchmarks are also called workload primitives; in the past their names directly recalled application characteristics (e.g. CICS, IMS, etc) while in recent years their names has become less specific.

LSPR workload primitives published for z/OS 1.9 are:

- ODE-B, On Demand Environment Batch;
- CB-L, Commercial Batch Long Job Steps;
- WASDB, WebSphere Application Server and Data Base;
- OLTP-T, Traditional On-line Workload;

Being very unlikely for a z/OS system to run only one type of workload IBM decided to create workload mixes by combining the above primitives.

LSPR workload mixes published for z/OS 1.9 are:

- LoIO-Mix, low DASD I/O rate;
- CB-Mix, commercial batch;
- TM-Mix, transaction moderate;
- TD-Mix, transaction dominant;
- TI-Mix transaction intensive;
- DI-Mix, data intensive;
- LSPR-Mix, a kind of average mix normally used in contracts\(^1\); however IBM recommends against using it for Capacity Planning.

Mainframe capacity has always been closely associated with how a workload uses and interacts with a particular processor hardware design. However this information was not available in the past so “external” metrics, such as DASD I/O rate or % of online workloads have always been used, assuming they generally corresponded to the “internal” workload/processor hardware interaction.

According to IBM (and our experience) the workload that best represents most of z/OS production systems is the LoIO-Mix. The good news is that it could be easily identified by calculating the ratio between the DASD I/O rate and the MSU used. If this ratio is less than 30 the workload can be considered as LoIO.

This is a typical algorithm used to determine the workload characteristics of a z/OS system:

1) Check if the ratio between DASD I/O rate and MSU used is less than 30; if yes choose LoIO Mix;

---

\(^1\) IBM now uses PCI (Processor Capacity Index) values which essentially are LSPR-Mix rounded values.
2) If the ratio is greater than 30, analyze the system workload trying to estimate the portion of CPU used by online and other workloads; if online is less than 25% choose CB-Mix;
3) if online is between 25% and 35% choose TM-Mix;
4) if online is between 35% and 55% choose TD-Mix;
5) if online is more than 55% choose TI-Mix.

In the graph in Figure 1 the IBM 2097-730 capacity estimates based on both workload primitives and mixes are presented. It’s easy to understand the importance of using the right workload: the difference in capacity between LoIO-Mix and DI-Mix is in fact more than 4,000 MIPS.

The bad news is that a DI-Mix workloads may look like LoIO because of the low DASD I/O rate to MSU ratio but it stresses the processor cache architecture causing a big reduction of the usable capacity. Unfortunately there is no way to identify a DI-Mix workload by using “external” metrics and rules. A new metric category is required. We will discuss that in the following chapters.

---

2 We do that by aggregating SMF 30 interval using the program name; this is automatically calculated in EPV for zOS.
3  z10 and z196 processor hardware design

Figure 2 shows a simplified view of the z10 processor cache architecture.

If data and instructions to be processed are found in the Level 1 cache (L1) dedicated to each processor\(^3\), this is called a “cache hit”. In this case the speed of the clock can be exploited well.

If data and instructions cannot be found in L1 the hardware tries to load them from the Level 1.5 (L1.5) cache, which is still a cache dedicated to each processor, from the Level 2 cache (L2) of the same book\(^4\), from the Level 2 cache (L2) of another book, from local memory or remote memory in this order.

In this case a “cache miss” occurs and clock cycles are lost while waiting for data and instructions to be loaded in the L1 cache. The number of lost cycles really depends on the cache level accessed, ranging from a few cycles for L1.5 to hundreds of cycles for memory.

---

\(^3\) There are two L1 caches, one for data the other for instructions, dedicated to each processor but for simplicity only one cache is depicted in the figure.

\(^4\) L2 serves all the processors in a book.
The same concepts apply to the z196 processor cache architecture presented in Figure 3. The main difference compared to the z10 is the addition of the Level 3 (L3) cache which is a cache serving all the processors on the same chip.\textsuperscript{5}

The bottom line is: “Workload capacity performance will be quite sensitive to how deep into the memory hierarchy the processor must go to retrieve the workload’s instructions and data for execution. Best performance occurs when the instructions and data are found in the cache(s) nearest the processor so that little time is spent waiting prior to execution; as instructions and data must be retrieved from farther out in the hierarchy, the processor spends more time waiting for their arrival.”\textsuperscript{6}

The two main factors determining workload performance are:

- Percentage of L1 misses over total searches;
- Percentage of L1 misses satisfied by each cache level (including memory).

In the next chapter we will show how to calculate them.

\textsuperscript{5} L1.5 has been renamed to L2 and the former L2 (the book cache) has been renamed to L4.
\textsuperscript{6} From IBM Large Systems Performance Reference
4 CPU Measurement Facility

A new hardware facility called “CPU Measurement Facility” (CPU MF) has been introduced with z10 machines. The CPU MF (Measurement Facility) together with the new z/OS Hardware Instrumentation Services (HIS) provides the ability to gain measurements on processor cache effectiveness.

To collect these measurements you need to perform the following steps:

- authorize the sampling facilities and counter set types you want to use through the support element (SE) console (only counters need to be activated);
- define a user ID for the HIS started task (provided in SYS1.PROCLIB);
- define the user id with a segment that specifies a default UID and a default HOME directory;
- create the HOME directory of the user ID in a local file system;
- enable SMF record type 113 writing in the SMFPRMxx member of the SYS1.PARMLIB;
- start the HIS by executing the S HIS command;
- activate data collection by issuing the following command:

  \[ F \text{ HIS}, B, TT='COUNTERS', PATH='/var/his', CTRONLY, CTR=ALL \]

Collected information are recorded in SMF 113 and in a USS file written in the HOME directory.

**BASIC COUNTERS**

Six metrics are provided in the Basic Counters section:

- B0, CYCLE COUNT
- B1, INSTRUCTION COUNT
- B2, L1 I-CACHE DIRECTORY-WRITE COUNT
- B3, L1 I-CACHE PENALTY CYCLE COUNT
- B4, L1 D-CACHE DIRECTORY-WRITE COUNT
- B5, L1 D-CACHE PENALTY CYCLE COUNT

Starting from these measurements the percentage of L1 misses over total searches can be calculated by using the following formula:

\[
\%L1 \text{ Miss} = \left( \frac{B2 + B4}{B1} \right) \times 100
\]

In the following figure a portion of the output USS file, produced when HIS data collection ends, is presented. Needed measurements are identified, translated to decimal and used to calculate the %L1 Miss value.

The output refers to a z10 machine (see CPU SPEED value) but the %L1 Miss formula works for both z10 and z196 machines.

---

7 Only CPU 00 counters are presented in the example; a counter section per each processor is provided.
z10 EXTENDED COUNTERS

Eight metrics are provided in the Extended Counters section on z10 machines:

- E128, L1 MISS SOURCED FROM L1.5 (instructions);
- E129, L1 MISS SOURCED FROM L1.5 (data);
- E130, L1 MISS SOURCED FROM LOCAL L2 (instructions);
- E131, L1 MISS SOURCED FROM LOCAL L2 (data);
- E132, L1 MISS SOURCED FROM REMOTE L2 (instructions);
- E133, L1 MISS SOURCED FROM REMOTE L2 (data);
- E134, L1 MISS SOURCED FROM LOCAL MEMORY (data);
- E135, L1 MISS SOURCED FROM LOCAL MEMORY (instructions).

Starting from these measurements the percentage of L1 misses sourced by each cache level can be calculated by using the following formulas:

\[
\%L1.5 = \frac{(E128+E129)}{(B2+B4)} \times 100 = 64.6
\]
\[
\%L2L = \frac{(E130+E131)}{(B2+B4)} \times 100 = 14.4
\]
\[
\%L2R = \frac{(E132+E133)}{(B2+B4)} \times 100 = 0.9
\]
\[
\%MEML = \frac{(E134+E135)}{(B2+B4)} \times 100 = 9.2
\]
By summing all the calculated percentages you might not get 100%. The reason is that some L1 Misses may be sourced by remote memory (memory on a different book). No counters are provided but you can easily calculate that percentage by using the following formula:

\[
\%\text{MEMR} = \frac{(B2+B4) - \text{Sum}(E128:E134))}{(B2+B4)} \times 100 = 10.9
\]

The same logic but different calculations have to be performed for z196 by using new extended counters. Unfortunately no official IBM documentation has been published at the time of writing\(^8\).
5 Relative Nest Intensity (RNI)

Only three z/OS 1.11 benchmarks are available: Low RNI, AVG RNI and High RNI. RNI means Relative Nest Intensity; it indicates the level of activity to the most performance sensitive area of the memory hierarchy: shared caches and memory. This part of the memory hierarchy is called the “Nest”.

In Figure 6 the z196 Nest has been highlighted. For this machine generation it is composed by L3 cache (shared by all the processors in the same chip), L4 cache (shared by all the processors in the same book) and memory

As already discussed, workload capacity performance is quite sensitive to how deep into the memory hierarchy the processor must go to retrieve the workload’s instructions and data to be executed. So the higher the RNI, the worse will be the workload capacity performance.

In practical terms the machine will look less powerful to a workload presenting High RNI characteristics than to a workload presenting AVG RNI or Low RNI characteristics.

The following formulas allow us to calculate the RNI of a system, depending on the machine generation, starting from the Extended Counters discussed in the previous chapter:

\[
\begin{align*}
\text{z10 } \text{RNI} &= \frac{(1,0 \times \%L2L + 2,4 \times \%L2R + 7,5 \times \%MEM)}{100} \\
\text{z196 } \text{RNI} &= \frac{(1,6 \times (0,4 \times \%L3 + 1,0 \times \%L4L + 2,4 \times \%L4R + 7,5 \times \%MEM))}{100}
\end{align*}
\]

It’s interesting to note that:

- z10 L2L corresponds to z196 L4L;
- z10 L2R corresponds to z196 L4R;
• L3 doesn’t exist in z10 machines.

The coefficient in bold are used to weight cache and memory accesses so in both formulas:
  • accessing the local book cache (L2L or L4L) is weighted 1;
  • accessing a remote book cache (L2R or L4R) is weighted 2,4;
  • accessing memory (including both local and remote book memory) is weighted 7,5.

The z196 formula looks a bit more complex because it includes L3 cache accesses (weighted 0,4) and an additional coefficient (1,6) to increase the resulting RNI value.
6 z/OS 1.11 benchmarks

Low, AVG and High RNI are the only benchmarks available for z196. To allow customers to compare current machines (mostly z9 and z10) with the new ones these benchmarks have been published for all the existing IBM machines too.

Unfortunately all the capacity planning studies for the existing machines have been performed using different benchmarks so customers will need a “bridge” to the new benchmarks when they will have to evaluate an upgrade to z196.

IBM provided some guidelines such as, for example:

- if the workload used to be LoIO-Mix then use AVG RNI;
- if the workload used to be DI-Mix then use High RNI.

The following graphs show a comparison of the expected capacity of z9 and z10 models based on LoIO-Mix and AVG RNI benchmarks.

LoIO-Mix is represented by a continuous line and AVG RNI by a dotted line. The red bars represent the increase or decrease in capacity when using AVG RNI instead of LoIO-Mix.

![Figure 7](image-url)
When using AVG RNI instead of LoIO-Mix the z9 capacity is about 2% lower. Capacity decrease for the biggest model is about 3% (560 MIPS).

When using AVG RNI instead of LoIO-Mix the z10 capacity is about 2,5% lower. Also in this case capacity decrease for the biggest model is about 3% (1,012 MIPS).

The following graphs show a comparison of the expected capacity of z9 and z10 models based on DI-Mix and High RNI benchmarks.

DI-Mix is represented by a continuous line and High RNI by a dotted line. The red bars represents the increase or decrease in capacity when using High RNI instead of DI-Mix.
When using High RNI instead of DI-Mix the z9 capacity, starting from the 708 model, is higher. Capacity increase for the biggest model is about 12% (1.827 MIPS).

When using High RNI instead of DI-Mix the z10 capacity, also in this case starting from the 708 model, is greater. Capacity increase for the biggest model is about 11% (2.711 MIPS).
It’s also interesting to compare new and old PCI\(^9\) (Processor Capacity Index) values. These are the numbers used by managers when discussing contracts and prices with IBM so they are also very important.

\[\text{z9 MIPS} \]
\[\text{z/OS 1.11 PCI vs z/OS 1.9 PCI}\]

\[\text{z10 MIPS} \]
\[\text{z/OS 1.11 PCI vs z/OS 1.9 PCI}\]

\(^9\) PCI used to be based on LSPR-Mix (harmonic mean of LSPR workload primitives); its values now look very close to AVG RNI.
z9 new PCI numbers, starting from the 713 model, are higher than the old ones. Capacity increase for the biggest model is about 2.3% (420 MIPS).

z10 new PCI numbers, starting from the 713 model, are higher than the old one. Capacity increase for the biggest model is about 3.3% (1.015 MIPS).

From the above graphs it appears evident that moving from the old to new z/OS benchmarks using only the IBM guidelines it’s not straightforward.

A better alternative is collecting MF counters, as described in the previous chapters, and evaluate your workload %L1 Miss (percentage of L1 cache miss) and RNI. Once done that, you can classify the workload by using the following rules.

<table>
<thead>
<tr>
<th>%L1 Miss</th>
<th>RNI</th>
<th>Benchmark</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 3%</td>
<td>&gt;= 0,75</td>
<td>AVG</td>
</tr>
<tr>
<td>&lt; 3%</td>
<td>&lt; 0,75</td>
<td>Low</td>
</tr>
<tr>
<td>3% to 6%</td>
<td>&gt; 1,00</td>
<td>High</td>
</tr>
<tr>
<td>3% to 6%</td>
<td>0,60 to 1,00</td>
<td>AVG</td>
</tr>
<tr>
<td>3% to 6%</td>
<td>&lt; 0,60</td>
<td>Low</td>
</tr>
<tr>
<td>&gt; 6 %</td>
<td>&gt;= 0,75</td>
<td>High</td>
</tr>
<tr>
<td>&gt; 6 %</td>
<td>&lt; 0,75</td>
<td>AVG</td>
</tr>
</tbody>
</table>

Figure 13
7 Using zPCR

For some years IBM has made available the zPCR tool. Using zPCR, a user can easily evaluate his machine capacity, taking into account the specific machine configuration instead of using LSPR tables.

To evaluate z196 machine capacity, zPCR 7.1a is required. The reference CPU, to be used when performing zPCR studies, is still the 2094-701 but its capacity has to be reduced from 602 to 593.

When you load a study created with previous zPCR versions, and therefore based on old benchmarks, the workload characteristics are automatically changed as follows:

- CB-Mix to AVG RNI,
- LoIO-Mix to AVG RNI,
- TM-Mix to AVG-High RNI,
- TD-Mix to AVG-High RNI,
- TI-Mix to AVG-High RNI,
- DI-Mix to High RNI.

As you can see two additional workloads are provided in zPCR:

- Low-AVG,
- AVG-High.

There are no benchmarks underlying these workloads; the reported values are calculated as an harmonic mean of the base benchmarks.

---

**Figure 14**

LSPR tables are based on an average configuration not representing any specific user.
In Figure 14 a study performed using zPCR 6.3c is presented. The Reference CPU is 2094-701 quoted 602 MIPS. The machine is a 2097-719 with 7 zIIPs. All the workloads are LoIO-Mix running on z/OS 1.9. The total GCP capacity is 12.042 MIPS while zIIP capacity is 4.668 MIPS.

When loading the above study using zPCR 7.1a we got the following message.

Answering Yes we accepted the suggested modifications and we got the result in Figure 16.

The Reference CPU is still a 2094-701 but it is now quoted 593 MIPS. All the workloads are now Average (AVG RNI). The total GCP capacity is 11.804 MIPS while zIIP capacity is 4.566 MIPS.
A last note is related to the usable benchmarks; zPCR 7.1a always uses the z/OS 1.11 benchmarks, no matter what is reported in the SCP column.
8 Conclusions

New benchmarks are provided for z/OS 1.11 and above; they are the only ones available for z196 machines. To allow customers to compare current machines (mostly z9 and z10) with the new ones these benchmarks have been published for all the existing IBM machines.

Unfortunately all the capacity planning studies up to now have been performed using completely different benchmarks so customers will need a “bridge” to the new benchmarks when they will have to evaluate an upgrade to z196.

To understand which is the right benchmark to use to estimate your machine capacity you can follow the IBM guidelines or collect the new measurement provided in SMF 113. We showed how the first option could produce contrasting results and probably increase capacity estimate errors.

Finally the new zPCR version (7.1a) is needed to perform studies involving z196 machines. Additionally in this case special care is needed when loading studies created with older zPCR versions.
Addendum

In this addendum z196 extended counters will be discussed in detail. Some examples of using MF counters in capacity planning and performance analysis will also be presented.

A.1 CPU Measurement Facility – Counters for z196

As discussed in previous papers, a new cache level (L3) has been introduced in the z196 processor architecture to provide memory shared among all the processors on a chip.

Unfortunately this is not the only difference between z10 and z196 extended counters.

Cache levels have been renamed so:
- the second level cache dedicated to each processor, called L1.5 in z10, is now called L2;
- the cache shared among all the processors in a book, called L2 in z10, is now called L4.

Additionally most of the extended counters fields have been changed11.
These are the metrics needed to analyze the ‘sourcing’ of L1 cache from the other cache levels and from memory:

a) E128, L1 MISS SOURCED FROM L2 (data);
b) E129, L1 MISS SOURCED FROM L2 (instr.);
c) **E150**, L1 MISS SOURCED FROM L3 (data);
d) E153, L1 MISS SOURCED FROM L3 (instr.);
e) E135, L1 MISS SOURCED FROM LOC L4 (data);
f) E136, L1 MISS SOURCED FROM LOC L4 (instr.);
g) **E152**, L1 MISS SOURCED FROM L3 THROUGH LOC L4 (data);
h) **E155**, L1 MISS SOURCED FROM L3 THROUGH LOC L4 (instr.);
i) E138, L1 MISS SOURCED FROM REM L4 (data);
j) E139, L1 MISS SOURCED FROM REM L4 (instr.);
k) **E134**, L1 MISS SOURCED FROM L3 THROUGH REM L4 (data);
l) **E143**, L1 MISS SOURCED FROM L3 THROUGH REM L4 (instr.);
m) E141, L1 MISS SOURCED FROM LOC MEM (data);
n) E142, L1 MISS SOURCED FROM LOC MEM (instr.).

The counters in bold are the one related to the L3 cache activity.
As you can see E150 and E153 provide the number of times data and instructions have been found in L3 cache.

A bit more strange is the fact that E152 and E155 are included in L4 local cache counters. What happens in this case is that data and instructions are found in the L3 cache of another chip in the same book. So they have to be moved first to the upper level cache (L4) which is shared among all the processors in the same book and then to the L1 cache.

The red arrows in the following figure show the path from L3 to L1.

---

11 Only the counters for L2 cache (1.5 in z10) are the same.
Similarly E134 and E143 are included in L4 remote cache counters. What happens in this case is that data and instructions are found in the L3 cache of another chip in another book. So they have to be moved first to the upper level cache (L4) in the remote book, then to the L4 cache of the local book and finally to the L1 cache.

The red arrows in the following figure show the path from L3 to L1.
A.2 Using MF counters

MF counters can be used to create different indexes to monitor. The most important of them are:

- L1 cache misses (L1M) and Relative Nest Intensity (RNI) to understand what is the right benchmark to use in Capacity Planning to represent each system;
- Cycles per Instructions (CPI), to evaluate cache contention.

L1M and RNI

In the following figure the L1M and RNI hourly profiles for the SYSA development system running in a z196 machine is presented.
Based on the LSPR rules, reported in the following table, and considering the prime shift the system could be classified as AVG RNI.

<table>
<thead>
<tr>
<th>%L1 Miss</th>
<th>RNI</th>
<th>Benchmark</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 3%</td>
<td>&gt;= 0,75</td>
<td>AVG</td>
</tr>
<tr>
<td>&lt; 3%</td>
<td>&lt; 0,75</td>
<td>Low</td>
</tr>
<tr>
<td>3% to 6%</td>
<td>&gt; 1,00</td>
<td>High</td>
</tr>
<tr>
<td>3% to 6%</td>
<td>0,60 to 1,00</td>
<td>AVG</td>
</tr>
<tr>
<td>3% to 6%</td>
<td>&lt; 0,60</td>
<td>Low</td>
</tr>
<tr>
<td>&gt; 6%</td>
<td>&gt;= 0,75</td>
<td>High</td>
</tr>
<tr>
<td>&gt; 6%</td>
<td>&lt; 0,75</td>
<td>AVG</td>
</tr>
</tbody>
</table>

Figure 4

It’s also interesting to take a look at the effectiveness of each cache level. The following graph shows the effect of the L3 cache (reported in light green). Only one book is used so no activity is reported from the remote L4 cache.

Figure 5

In Figure 6 the L1M and RNI hourly profiles for two production systems running in a z10 machine are presented.
Based on the LSPR rules, reported in the following table, the system could be classified as AVG RNI. However in some hours the RNI values are above 1 so the system could also be classified as HIGH.

<table>
<thead>
<tr>
<th>%L1 Miss</th>
<th>RNI</th>
<th>Benchmark</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 3%</td>
<td>&gt;= 0,75</td>
<td>AVG</td>
</tr>
<tr>
<td>&lt; 3%</td>
<td>&lt; 0,75</td>
<td>Low</td>
</tr>
<tr>
<td>3% to 6%</td>
<td>&gt; 1,00</td>
<td>High</td>
</tr>
<tr>
<td>3% to 6%</td>
<td>0,60 to 1,00</td>
<td>AVG</td>
</tr>
<tr>
<td>3% to 6%</td>
<td>&lt; 0,60</td>
<td>Low</td>
</tr>
<tr>
<td>&gt; 6 %</td>
<td>&gt;= 0,75</td>
<td>High</td>
</tr>
<tr>
<td>&gt; 6 %</td>
<td>&lt; 0,75</td>
<td>AVG</td>
</tr>
</tbody>
</table>

Looking at the overall machine utilization we realized the machine is saturated in many hours of the day so both L1M and RNI tends to be inflated.

The effectiveness of each cache level can be analyzed in the following graph. This is a multi-book machine but thanks to HiperDispatch sourcing from remote book caches (L2R) is almost zero. About 20% of the L1 misses are sourced from the local book cache (L2L). Moving to z196 part of this activity would be satisfied by the chip cache thus improving performance.
CPI

In the following figure the CPI hourly profile for the SYSA development system is presented.

Maximum values are around 5 but after 9 am they are between 2 and 3.
Looking at the SYS1 and SYS2 production systems we get a complete different picture.

![Figure 10](image)

Values are around 6 but as soon as the machine become saturated they peaks up to 10.

As you can imagine there is not a Rules of Thumb for the ideal CPI value. However it’s intuitive that to exploit the processor power the CPI should be as low as possible.

In this case an upgrade will greatly reduce the SYS1 and SYS2 CPI in the peak hours.

Measuring this index on a regular basis will allow you to evaluate the effect of changes in:

- hardware configuration;
- microcode;
- exploitation of HiperDispatch;
- LPAR configuration such as weights, number of logical processors, number of LPARs, etc.;
- system and subsystem levels (e.g. DB2 V10 will take advantage of a new HW instruction to prefetch data and instructions);
- workload mixture.

Using this knowledge you will be able, in case of performance degradation after a change, to quickly identify the problem and solve it.