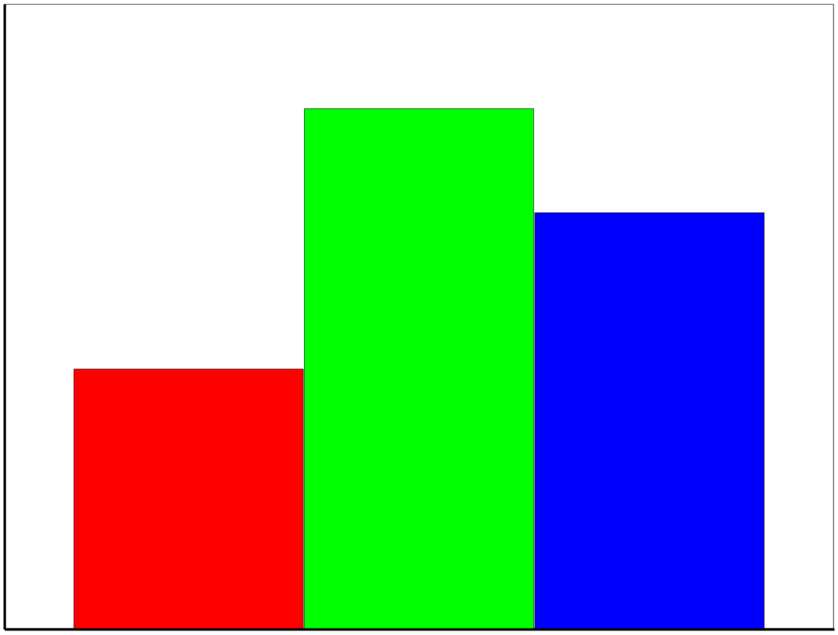
Large Systems Performance Reference



Large Systems Performance Reference

Document Number SC28-1187-25

Note

Before using this information and the product it supports, be sure to read the general information under “Notices”.

Twenty five (April 2023) this edition contains performance information for the new IBM Z16 models.

Information contained in this publication is subject to change from time to time. Any such changes will be reported in subsequent revisions or Technical Newsletters.

This publication will only be available in PDF format at the following web address.

https://www.ibm.com/support/pages/ibm-z-large-systems-performance-reference

Contents

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM’s product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM’s intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the user’s responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Commercial Relations, IBM Corporation, Purchase, NY 10577.

Trademarks and Service Marks

The following terms, denoted by the symbol (™) in this publication, are trademarks or service marks of the IBM Corporation in the United States or other countries or both:

z900	zSeries	IBM eServer
z/Architecture	z/VM	z/OS
ACF/VTAM	AIX	AS/400
CICS	CICS/ESA	CICS/MVS
CICS/VSE	DB/2	DB2
DFSMS	ES/3090	ES/4381
IBM S/390 Parallel Enterprise Server-Generation 6	ES/9000	ES/9370
IBM S/390 Parallel Enterprise Server-Generation 5	ESA/370	ESA/390
IBM S/390 Parallel Enterprise Server-Generation 3	IBM	3090
IBM S/390 Parallel Enterprise Server-Generation 4	IMS/ESA	GDDM
IBM S/390 Multiprise 2000	OS/390	VSE/ESA
Large Systems Performance Reference	LPAR	VTAM
LSPR	LSPR/PC	MVS/DFP
MVS/ESA	MVS/SP	MVS/XA
Processor Resource/Systems Manager	PR/SM	OS/400
System/370	System/390	RACF
S/390 Multiprise	VM/ESA	VM/XA
WebSphere	VisualAge	
IBM System z9 EC formerly IBM System z9 109		
IBM System z9 BC		
IBM System z10 EC		
IBM System z10 BC		
IBM System zEnterprise 196 or z196		
IBM System zEnterprise 114 or z114		
IBM System zEnterprise EC12 or zEC12		
IBM System zEnterprise BC12 or zBC12		
IBM z13		
IBM z13s		
IBM z14		
IBM z14 MODEL ZR1		
IBM z15		
IBM z15 T02		
IBM z16		
IBM z16 A02		

The following terms, also denoted the symbol (™) in this publication, are trademarks or service marks of other companies or individuals listed below:

MARK	OWNER
EXPLORE	Goal Systems International Inc
MSC/NASTRAN	MacNeal-Schwendler Corp.
UNIX	The Open Group in the United States and other countries
Linux	Linus Torvalds

Preface

The IBM™ Large System Performance Reference™ (LSPR™) ratios represent IBM's assessment of relative processor capacity in an unconstrained environment for the specific benchmark workloads and system control programs specified in the tables. Ratios are based on measurements and analysis. The amount of analysis as compared to measurement varies with each processor.

Many factors, including but not limited to the following, may result in the variances between the ratios provided herein and actual operating environments:

- Differences between the specified workload characteristics and your operating environment
- Differences between the specified system control program and your actual system control program
- I/O constraints in your environment
- Incorrect assumptions in the analysis
- Unknown hardware defects in processors used for measurement
- Inaccurate vendor claims.

IBM does not guarantee that your results will correspond to the ratios provided herein. This information is provided "as is", without warranty, express or implied.

Abstract

IBM’s Large Systems Performance Reference (LSPR) method is designed to provide relative processor capacity data for IBM System/370™, System/390™, and z/Architecture™ processors. All LSPR data is based on a set of measured benchmarks and analysis, covering a variety of system control program (SCP) and workload environments. LSPR data is intended to be used to estimate the capacity expectation for a production workload when considering a move to a new processor.

IBM considers LSPR data to be a reliable set of relative processor capacity data. This is the only reference of its type that is based primarily on actual processor measurements. Because it is based on measurements, LSPR data takes into account individual SCP and workload sensitivities to the underlying design of each processor represented.

Chapter 1. Background

IBM's Large Systems Performance Reference™ (LSPR)™ method is intended to provide IBM System/370™ and System/390™ architecture, and z/Architecture™ processor capacity data across a wide variety of system control programs (SCP's) and workload environments.

The LSPR's focus is solely on processor capacity, without regard to external resources such as storage size, or number of channels, control units, or I/O devices. To assure that the processor is the primary focus, the processor capacity data reported assume sufficient external resources so as to prevent any significant external resource constraints. With this approach, ***the LSPR is designed to represent each processor in its best light;   that is, the processor itself is the only limiting factor to doing work***. Resulting LSPR capacity data is therefore meaningful for establishing a realistic view of relative capacity between specific processors for SCP's and workload environments that have characteristics similar to those measured.

Rationale for Reliable Processor Capacity Data

When considering the acquisition of a large central processor, one needs to understand its capacity potential as precisely as possible. This capacity potential is generally expressed in terms relative to a currently installed processor. If expected capacity is understated or overstated, the cost of the error can be significant. That cost can be misspent dollars, or lost ability to accommodate work.

Any processor's ability to support work, either in terms of jobs, transactions, or end-users, is a function of the nature of the work to be performed. A processor's absolute capacity can be determined for any specific workload, but such absolute capacity information is not particularly useful to a capacity planner unless the information represents his exact production workload. It is difficult to produce tables that meaningfully represent processor capacity in absolute terms, except when they relate to a specific workload.

However, given that a table of absolute processor capacity values can be built for a given workload environment, we then have a basis for determining the relative capacity between those processors. These relative capacity values are meaningful, not only for the exact workload represented, but also for workloads of a similar nature.

Many processor acquisitions today are replacements for existing machines, made for the purpose of adding (or consolidating) capacity. Since the capacity of the current processor is normally well understood, the capacity of a potential new processor, relative to the current one, can be assessed by using known capacity relationships between those machines for the appropriate workload type.

Sources for Processor Capacity Data

There are several ways to establish a capacity expectation for a new processor, each with its advantages and disadvantages.

Customized Benchmark

A customized individual benchmark, which could be run on all processors of interest, will produce the most accurate capacity data on which to base a processor acquisition decision. However, ***for results to be meaningful, it is essential that a benchmark consist of representative work run in a representative way***.

> **Insight**
>
> *If done properly, a customized representative benchmark will provide the most accurate view of relative processor capacity. However, customized benchmarks are expensive to create, maintain, and run.*

A measured benchmark that is not representative of production work has little value in trying to understand processor capacity relationships.

There are many considerations to preparing a benchmark, including:

- SCP and related products (JES, RACF™, VTAM, RMF ...)
- Application subsystems (CICS™, DB2™, IMS, TSO, CMS, WAS ...)
- Other program offerings
- Application programs
- Performance monitors
- Data files (datasets), and databases
- Scripts (end-user commands), or jobs
- Working set sizes
- Terminal simulation
- Size of end-user population
- Average think time, and think time distribution
- Transaction rates
- Response time criteria
- Operational methodology
- Metrics to be used
- Repeatability and consistency
- Portability

The creation, maintenance, and measurement activity associated with a benchmark is likely to become an extremely resource intensive proposition given the complexities of the typical DP environment today. For that reason, individual customized benchmarks are not as common as they once were.

Benchmarks are all too often assembled without concern for representativeness. For example, “kernels” are used because they require minimal effort. Or a batch workload is created to represent an on-line environment, simply because batch is easier to construct and run. There is no short-cut approach to benchmarking that can assure reliable results. If the results of these types of benchmarks should happen to match those of a production workload, it would be purely coincidental, with no assurance that they would continue to match when a different set of processors is measured.

MIPS Tables

There are many published sources of processor capacity data available in the industry today. Most of these sources provide data in the form of MIPS tables. MIPS tables available from consultants and industry watchers are not based on independent measurements. Rather, they typically are developed using manufacturer’s announced performance claims. Over time, some of these MIPS tables may include a subjective analysis of feedback from various clients of these systems.

Insight

While MIPS tables may be useful for rough processor positioning, they should not be used for capacity planning purposes. Single-number processor capacity tables are inherently prone to error because they are not sensitive to the type of work being processed or to the LPAR configuration of the processor.

Most published MIPS tables carry a single-number connotation; that is, the capacity of each processor can be represented with one number. Such tables are insensitive to the workload environment run on those processors.

A perceived advantage in the use of MIPS is that the implied scale is easily recognized; that is, a MIPS rating for a processor provides an easy visual positioning for that machine. This type of rating may be useful for “rough” processor positioning, but does not offer the accuracy necessary for doing capacity sizing. Studies of LSPR data for the various workload environments measured show that the potential for error, when using MIPS tables to assess relative capacity, can sometimes be significant. This is true even if the MIPS table is built from LSPR data. The problem relates to the fact that workload sensitivity and specific LPAR configurations are simply not considered in a single-number table.

Other Sources

Modeling Tools

Several modeling tools are available to aid in capacity planning for DP systems. Models are designed to provide a “big picture” view of performance, rather than to expose the capacity of any single component. In other words, models are designed for the purpose of analyzing overall system performance, given any number of resource tradeoff scenarios.

All performance models must carry some form of processor capacity data, since that element is a part of the total resource picture. Processor capacity data within a model is usually either carried as a single-number per processor table, or modeled in some way based on a set of processor and workload related characteristics.

Sophisticated modeling tools allow the use of alternative processor capacity data, instead of the built-in data or algorithm. Providers of such tools generally recommend that user-provided processor data be used unless better data is available. IBM considers LSPR data as a reliable alternative, since it is workload sensitive, and based primarily on measurements (see Figure 1).

Often a model will be used for other than its intended purpose, such as to extract processor capacity information. Modeling of this nature does nothing more than expose the underlying processor capacity data contained in the model.

System Resources Manager (SRM) Constants

One of the features of MVS, OS/390, and z/OS is that they attempt to offer somewhat consistent service units for the processor resource to do work, no matter what specific processor is being used. These service units are computed by applying the MVS, OS390, or z/OS built-in SRM constant to the CPU time consumed by each unit of work. There is a unique SRM constant defined for every processor model. IBM assigns the SRM constants for IBM processors; LSPR data is used as an input when developing the IBM assigned SRM constants. SRM constants for IBM compatible processors are supplied by each vendor.

Although SRM constants are sometimes used as a source of processor capacity data, this practice is highly discouraged. By design, SRM constants are single-number metrics. Therefore, SRM constants have the same problems as MIPS when it comes to providing a precise view of processor capacity relationships, because there is no consideration for workload type. Furthermore, SRM constants at the LPAR level can deviate substantially from the actual capacity of the LPAR due to their sensitivity to the number of logical engines.

The LSPR Alternative

Each of the above sources for processor capacity data is based on “single-number per processor” data. The interrelationships between workloads, LPAR configurations and processor design, today, are extremely complex. To be accurate, the performance of each processor must be assessed in an environment sensitive way. For this reason, ***no “single-number per processor” capacity table can necessarily provide an accurate view of relative processor capacity***.

As an alternative to the above sources, IBM offers the Large Systems Performance Reference (LSPR). LSPR data consists of a variety of workloads, each representing a type of production environment. LSPR results are based on measurements and analysis across the majority of contemporary System 370/390 and z/Architecture processors, both IBM and IBM compatible. Workloads include various batch and on-line environments that provide a reasonable representation of major types of data processing activity, such as WebSphere Application Serving, traditional on-line transactions processing, and commercial batch. These benchmarks are run using the same software that would be installed in a production environment.

The goal of the LSPR is to offer reliable relative capacity information, which takes into account processor design sensitivities to workload type.

zPCR – LSPR taken to the next level

The LSPR shows relative capacity ratios that are sensitive to workload type. However, LPAR configuration is also a very sensitive factor in capacity relationships.   IBM offers a tool for client use, zPCR, that takes the LSPR to the next level by estimating capacity relationships that are sensitive to workload type and LPAR configuration, processor configuration, as well as specialty engine configuration.   All these factors may be customized to match a client's configuration. The LSPR data is contained in the tool. For the most accurate capacity sizings, zPCR should be used.

Chapter 2. Metrics

Over time, various approaches to characterizing processor capacity have evolved. Early metrics tended to concentrate on the rate at which a processor executes instructions. One metric of this type that has survived is MIPS (millions of instructions per processor second).

As processors have grown larger and more complex in their design, the ability to characterize processor capacity relationships accurately with MIPS has diminished drastically. Processor sensitivity to different workload types must be considered as a factor in establishing capacity relationships. Therefore, metrics that relate directly to work done have been defined. These terms are **external throughput rate (ETR)** and **internal throughput rate (ITR).**

MIPS (IER or Instruction Execution Rate)

Processors are designed to execute instructions (OPCODES) that are in its inventory of functional activities. These instructions are processed at some average rate. That rate is quoted as MIPS (millions of instructions per processor second).

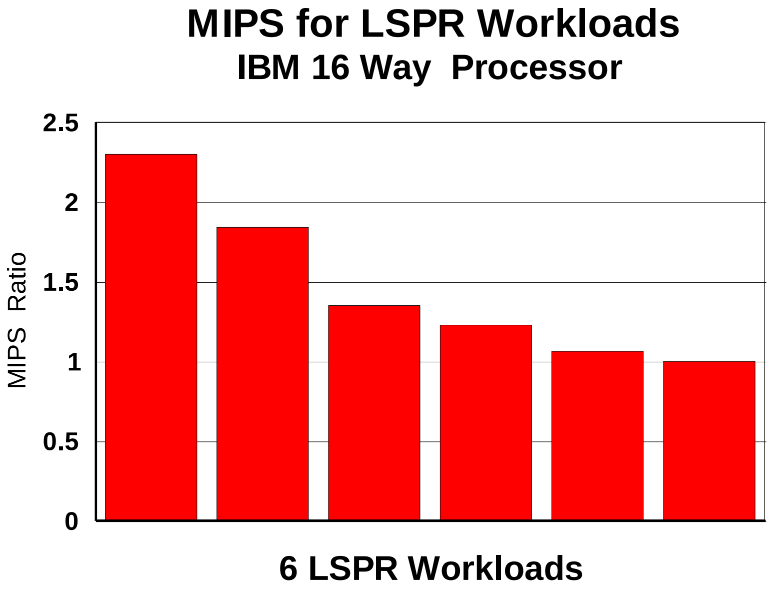
To express MIPS rates, IBM traditionally has used the term **instruction execution rate (IER)** instead. The capacity for IBM processors announced through the early 1980’s was generally made in terms of relative IER. That is, the IER of the new processor was compared to the IER of a previously announced model. Then, the capacity of the new processor was expressed as an IER ratio relative to the older model.

In the early days of data processing, when processor design was simple, instruction execution rates correlated reasonably well with the ability of the processor to do work. Expressing capacity with instruction execution rates could then provide an adequate view of how one processor might perform relative to another. However, as processors got more complex in their internal design, and as user interactions got more varied and specialized, the usefulness of instruction execution rates as an expression of capacity began to diminish.

With today’s high-performance processors, the actual MIPS rate achieved is extremely sensitive to the workload type being run, and its relationship to underlying processor design. LSPR workloads can be used to demonstrate this fact, since actual MIPS rates are frequently captured in the LSPR measurement process for IBM processors.

Figure 1 shows the ratio of MIP rates measured for several individual LSPR workloads on the same physical processor. Although the actual MIPS rates are not identified in this figure, you can see that, depending on workload, the ratio of MIPS for the same machine could exceed 2.

Figure 1. Relative MIP rates for LSPR workloads on a single processor



Generally, the workloads that generate the highest MIPS rates are batch, and the workloads that generate the lowest MIPS rates are on-line. This is true for all System/370 and System/390 architecture, and z/Architecture processors including IBM-compatible processors. The actual range from lowest to highest MIPS rate for LSPR workloads (or production workloads) on any given processor depends on that processor's design.

There are several high-level design factors on contemporary processors that prevent IER (or MIPS) from being meaningful as a capacity indicator.

- Overlapped function

  One of the techniques used to enhance the level of performance for large processors is to design high levels of overlap for the functions that a processor must perform to execute instructions. Various degrees of overlap are achieved in the instruction decode and execution process. The degree to which overlap is achieved is extremely dependent on the processor's specific design, and on the type of workload being processed. The greater the degree of overlap achieved the higher the effective instruction execution rate.

  One of the design factors usually known about a processor is its basic cycle time. Two processors with the same cycle time are not necessarily equal in capacity because one may be able to accomplish more within a cycle than the other. Instruction efficiency relates the number of cycles an average instruction requires to execute. Instruction efficiency is a function of all the overlapped capability that can be realized by the processor.

Each different production workload environment tends to have its dominant instruction sets. Various processor designs can be sensitive to dominant instruction sets and instruction sequences in different ways. This relationship of design to dominant instruction sets has a significant influence on individual processor capacity, and therefore, on processor capacity relationships.

- High-speed buffer (HSB) or Cache

  All System/370, System/390, and z/Architecture high-end processors today have one or more high-speed buffers (HSBs) implemented in their design to enhance overall performance. By keeping data and instructions that are being referenced (or likely to be referenced) in the HSB, the effective time to access these items is reduced dramatically. The movement between the slower central storage and the HSB is managed automatically by the processor, being overlapped with normal instruction processing activity.

  The size and design of the HSB plays an important role in the ability of a machine to process a workload. Some workload types benefit more from certain HSB design implementations than do others. Workloads with the best buffer hit ratios will cause the processor to have higher MIPS rates. Workloads with lower buffer hit ratios will cause that processor to have lower MIPS rates. Processor capacity and processor capacity relationships, therefore, become very sensitive to the HSB implementations of the processors being considered.

- N-way processing

  Early data processing systems were primarily uniprocessors. Over time multiprocessors have evolved to the point where today there are as many as one hundred and one tightly-coupled processors in a complex. System control program software is especially designed to manage multi-engine processors. Some portion of the instructions executed must go toward this N-way management function, as opposed to doing application work. Any use of instruction execution rates as a capacity indicator on these systems would include processing time that did not represent application work.

  There are also hardware performance considerations related to N-way processing. For most SCP operating environments, any work may be dispatched on any of the “N” engines, at any time. For most N-way designs, each engine has its own high speed buffer (HSB). The hardware must assure that any particular memory location that is changed is represented in only one engine’s HSB at a time. Therefore, as work tends to move around to different engines, so must any HSB-associated storage. When SCP dispatching decisions are frequent, as is typical with on-line workloads, this hardware N-way overhead will be the greatest.

- Micro code

  In the beginning, data processing systems were primarily hardware - based. Over time processor technology has evolved into extensive use of micro code to provide function. The advantage of micro code is that modifications to a design could be made without expensive hardware rework. As micro code flexibility and use grew, some functions normally performed by software were implemented more effectively in micro code. Usually, each micro coded software function becomes a new or extended instruction, more complex than typical OPCODEs, but doing the work that would normally be done with an entire routine in software. The use of these micro coded instructions has the tendency to lower the actual MIPS rate, while improving the processor's ability to do work.

Every workload (production or benchmark) has its own characteristics relative to how each of these hardware design features is exploited. For example, batch workloads tend to exploit N-way processors more efficiently than do on-line workloads, simply because of the difference in the rate of dispatching decisions that the operating system must make. On-line workloads tend to realize a greater performance benefit from larger and more sophisticated high speed buffer designs than do batch workloads. This is because the storage reference patterns of on-line workloads tend to be much more random.

Insight

Hardware design defines how a processor can perform. The software being exercised determines how a processor actually does perform.

The reason any particular processor performs as it does, lies in the interrelationship of the particular workload being run, to the underlying processor design.

Contemporary Use of MIPS

Various MIPS tables are used by organizations for the purpose of calculating relative capacity between processors. The individual MIPS values supplied for each processor are only loosely tied to the traditional meaning of the term. One perceived advantage in the use of MIPS tables is that the implied scale is easily recognized, that is, a MIPS rating for a processor provides an easy visual positioning of that machine on some grand scale of processor capability.

The use of MIPS tables produces a major problem when trying to understand relative processor capacity. The problem relates to the fact that different workload environments can have a significant effect on the way any particular processor design behaves. Therefore, the relative capacity of one processor to another will be very dependent on the type of work being run. As a result, ***it is***

often difficult to accurately position the processing capability of today's high-end processors with single-number tables.

The perception is that MIPS tables aid in understanding relative processor capacity across vendors, since these processor ratings are all tied to the same scale. However, just as IBM processor designs are extremely sensitive to the workload environment being run, so are those of the IBM compatible vendors. In fact, one can often see greater workload sensitivity when comparing processors from two different vendors, than when comparing two processors of the same vendor.

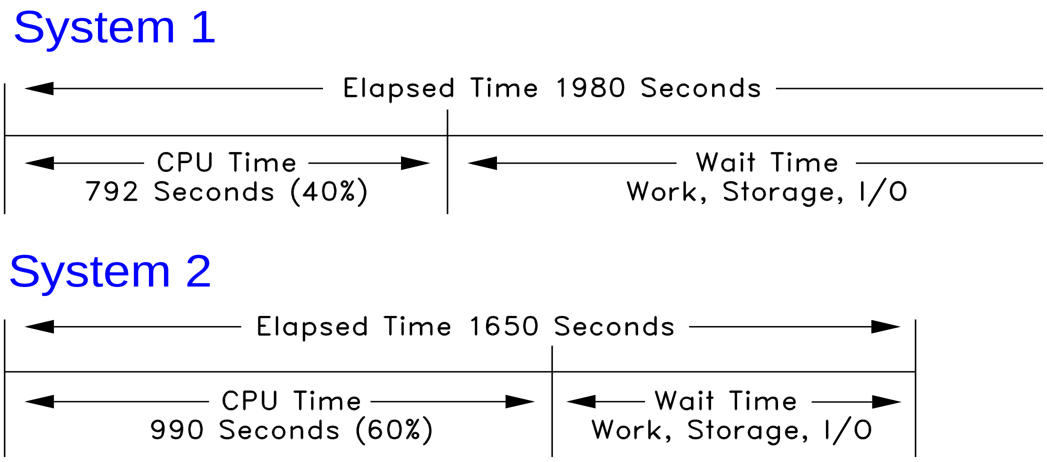
In today's world, it would be unwise to ignore the effects of different SCP and workload environments when making processor capacity comparisons. For this reason, IBM has chosen to provide capacity data in terms of work accomplished for a variety of workloads and SCP's, rather than in MIPS or instruction execution rates.

Workload Throughput Rates

Processors are purchased to do work rather than to do "MIPS". Given that the rate at which work is processed can be easily determined, it would seem natural that the best way to rate a processor is in terms of work units that it can do over time. To measure work done, two metrics have been defined. These metrics are external throughput rate (ETR) and internal throughput rate (ITR).

Assume that a benchmark workload is measured on each of two systems, with the results noted in Figure 2.

Figure 2. System capacity versus processor capacity



If the question being asked is:

"Which of the two systems is the better one for this workload?"

the correct answer is system number 2, because it processed the work in less elapsed time. ***External throughput rate (ETR), an elapsed time measure, focuses on system capacity.***

If the question being asked is:

"Which of the two systems has the better processor for this workload?"

the correct answer is system number 1, because it used less processor time to accomplish the same work. Because of the way that the question is posed here the focus must be changed to the processor itself. ***Internal throughput rate (ITR), a processor time measure, focuses on processor capacity.***

External Throughput Rate (ETR)

External throughput rate is computed as:

ETR = Units of Work / Elapsed Time

"Units of work" are normally expressed as jobs (or job-steps) for batch workloads, and as transactions or commands for on-line workloads (SCPs and most major software products have facilities to provide this information). To be useful, the "units of work" measured must represent a large and repeatable sample of the total workload, in order to best represent the average. Elapsed time is normally expressed in seconds.

ETR characterizes system capacity because it is an elapsed time measurement (system capacity encompasses the performance of the processor and all of its external resources, considered together). As such, ***ETR lends itself to the "system comparison methodology"***. This methodology requires the data processing system to be configured with all intended resources, including the processor, with appropriate amounts of central storage, expanded storage, channels, control units, I/O devices, TP network, and so on.

Once configured, the goal is to determine how much work the system, as a whole, can process over time. To do this, the system is loaded with the appropriate workload, until it cannot absorb work at any greater rate. The highest ETR achieved is the processing capability of the system.

When you make a system measurement of this type, all resources on the system are potential capacity inhibitors. If a resource other than the processor itself is, in fact, a capacity inhibitor, then it is likely that the processor will be running at something less than optimal utilization.

This **system comparison methodology** is a legitimate way to measure when the intent is to assess the capacity of the system as a whole. For on-line systems, response time also becomes an important system related metric, as poor response times will inhibit a user's ability to do work. Therefore, system measurements for on-line work usually involve some type of response time criteria. If the response time criteria is not met, then it does not matter what ETR can be realized.

Internal Throughput Rate (ITR)

Internal throughput rate is computed as:

ITR = Units of Work / Processor Busy

As with ETR, “units of work” are normally expressed as jobs (or job-steps) for batch workloads, and as transactions or commands for on-line workloads (SCPs and most major software products have facilities to provide this information). To be useful, the “units of work” measured must represent a large and repeatable sample of the total workload, in order to best represent the average. Processor busy time is normally expressed in seconds.

For the purpose of computing an ITR, processor busy time should include all processing time to do work, including the operating system related overhead. On an N-way processor, processor busy time must represent the entire complex of engines as if it were a single resource. Therefore, processor busy time is the sum of the busy times for each individual engine, divided by the total number of engines. Since all processor time is included, “captured” and “UN-captured” time considerations are unnecessary.

ITR characterizes processor capacity, since it is a CPU busy time measurement. As such, ***ITR lends itself to the “processor comparison methodology”***. Because the LSPR’s focus is on a single resource (the processor), you must modify the measurement approach from that used for a ***system*** comparison methodology.

To ensure that the processor is the primary point of focus, you must configure it with all necessary external resources (including central storage, expanded storage, channels, control units, I/O devices) in adequate quantities so that they do not become constraints. You need to avoid using processor cycles to manage external resource constraints in order to assure consistent and comparable measurement data across the spectrum of processors being tested.

There are many acceptance criteria for LSPR measurements that help assure that external resources are adequate. For example, internal response times should be sub second; if they are not, then there is some type of resource constraint that needs to be resolved. For various DASD device types, expected nominal service times are known. If the measured service times are high, then some type of queuing is occurring, indicating a constrained resource. When unexpected resource constraints are detected, they are fixed, and the measurement is redone.

Because the processor itself is also a resource which must be managed by the SCP, steps must be taken to ensure that excess queuing on it does not occur. The way to avoid this type of constraint is to make the measurements at preselected utilization levels that are less than 100%. Because the LSPR is designed to relate processor capacity, measurements must be made at reasonably high utilization, but without causing uncontrolled levels of processor queuing. Typically, LSPR measurements for on-line workloads are made at a utilization level of approximately 90%. Batch workloads are always measured with steady-state utilization’s above 90%. Mixed workloads containing both an on-line and batch component are measured at utilizations near 99%.

One additional point needs to be made about processor utilization. Whenever two processors are to be compared for capacity purposes, they should both be viewed at the same loading point, or, in other words, at equal utilization. It is imprecise to assess relative capacity when one processor is running at low utilization and the other is running at high utilization. The LSPR methodology mandates that processor comparisons be made at equivalent utilization levels.

ITR/ETR Relationship

An ITR can be viewed as a special case ETR, that is, ***an ITR is the measured ETR normalized to full processor utilization***. Therefore, an alternate way to compute an ITR is:

ITR = ETR / Processor Utilization

To show that the arithmetic above works out to be the same as the previous ITR formula, consider the following. The formula for processor utilization is:

Processor Utilization = Processor Busy Time / Elapsed Time

Substituting in the above for *ETR* and for *Processor Utilization*, gives:

ITR = (Units of Work / Elapsed Time)/(Processor Busy Time / Elapsed Time)

You will see that the two **Elapsed Time** values factor out, giving the same formula as originally stated for ITR.

Insight

The sole purpose of computing an ITR is to normalize out the slightly unequal utilization that may be represented by the ETR, since measurement techniques cannot assure exactly equal utilization levels.

There is a reason to normalize ETRs when comparing processor capacity. If every benchmark measurement could be made at the **exact** same utilization, you could simply compare the two ETR values to determine relative processor capacity. However, measurement techniques seldom allow identical utilization levels to be achieved between runs.

Table 1 shows results from two online workload measurements made for the LSPR. The target utilization for these measurements was 90%. To achieve this target, the number of logged-on users is adjusted as necessary, so that we are within three percentage points of the target. As you can see, the measurements resulted in utilizations close to the target, but slightly different.

Table 1. **LSPR Measurement example for Online workload**.

	Processor A	**Processor B**	**Ratio**
Measured Data:			
Elapsed Seconds	720.54	720.32	
Processor Seconds	630.11	627.72	
Transaction count	727,736	1,273,150	
Calculated Data:			
Utilization (%)	87.45	87.14	
ETR	1001.0	1767.5	(1.77)
ITR	1154.9	2028.3	1.76

With the measured data, you can compute an ETR and an ITR for each processor. If you were to simply compare the two ETR values to determine relative capacity, the ratio would be flawed, because of this slightly unequal utilization. To make the results comparable to each other, we must normalize out the slightly unequal utilization values measured. It does not matter what we normalize to; for LSPR purposes, we chose to normalize the ETR to 100% utilization, and call it an ITR.

Throughput Rates Are Workload Unique

Both ITR values and ETR values are unique to the specific SCP and workload that was measured. ITRs are useful for determining the relative capacity between two processors running the exact same workload environment. ETRs are useful for determining the relative capacity between two appropriately configured processing systems running the exact same workload environment. ***The absolute ITR and ETR values from one workload and SCP cannot be meaningfully compared to those of a different workload or SCP.*** Nor should absolute ITR and ETR values from a specific benchmark workload be compared to those of a production workload.

Table 2. **Example showing ITR values for 4 different processors**

Processor	**Workload 1**	**Workload 2**	**Workload 3**
Processor A	0.02506	0.901	41.69
Processor B	0.04835	2.091	102.16
Processor C	0.05305	2.334	118.23
Processor D	0.06296	2.713	137.36

Table 2 shows ITR values for four different processors for several LSPR workloads labeled workload 1, 2 and 3. It should be stated that the average unit-of-work is completely different between each of these workloads, and therefore the ITR scale is also unique for each workload.

Expressing Relative Capacity with ITRs

ITRs are useful for determining capacity relationships between processors for a given SCP and workload environment. This is done by dividing the ITR of one processor by the ITR of another to produce an ITR ratio (ITRR). For example, to determine the capacity of processor “B” relative to that of processor “A”, use the formula:

ITR Ratio (or ITRR) = ITR for CPU-B / ITR for CPU-A

Note: ITR values used in this calculation must be for identical SCP and workload environments.

ITR values are intended to be used for calculating relative processor capacity. The benefit of using LSPR ITR data is that you are working with workload sensitive data. As such you will have a more reliable and accurate view of relative capacity than can be provided by any MIPS table, or any other single-number per processor source.

Table 3. **Example showing ITRR values relative to Processor A**.

Processor	Workload 1	Workload 2	Workload 3
Processor A	1.00	1.00	1.00
Processor B	1.93	2.32	2.45
Processor C	2.12	2.59	2.84
Processor D	2.47	3.01	3.30

Table 3 shows ITR ratios developed using the absolute ITR values in Table 2. By representing the capacity of a set of processors relative to Processor A, we can see how relative capacity varies with the different workload types.

Chapter 3. Workload Environments

Data processing systems are designed to provide a range of services, using a wide variety of software products and application programs. On-line systems provide services directly to the end-user, while batch systems offer deferred services. In most cases, production systems offer a combination of many different types of services.

Assuring Representativeness for a Benchmark

In order for any benchmark workload to be useful, it is essential that its instruction paths and the storage reference patterns be representative of actual production work. Because of the complex interrelationships between software and processor design, it is impossible to ascertain the instruction paths and the storage reference patterns for a production workload. Therefore, the only way to assure that a benchmark is truly representative of production work is to use actual production software and activities.

For this reason, non-representative workloads (including “kernels”) are not considered useful as benchmarks. It would only be by sheer chance that capacity relationships derived from a non-representative benchmark would match up with the capacity realized when the actual production workload is moved to another processor.

Insight

There is no reasonable way to construct a benchmark that simulates instruction paths and storage reference patterns typical of a production workload without using actual production software and activities.

Software

Many types of software are used to take advantage of System/370, System/390 architecture, and z/Architecture processors.

System Control Program (SCP)

Basic processor support software is known as the system control program, or SCP. The SCP provides the routines to manage the processor and external resources such as storage and I/O devices. The SCP controls the dispatching of work and the allocation of resources on the processor complex.

Various software products are also associated with the SCP, including JES, RACF, VTAM™, TSO, and CMS. Performance monitors, which may also be associated with the SCP, are discussed below.

Each LSPR benchmark workload includes the use of the appropriate SCP, and associated components as applicable. SCPs used for the various LSPR benchmarks measured include z/OS, OS/390, MVS, z/VM, VM, VSE, and Linux on IBM Z.

Subsystems

Most production systems include the use of one or more major application subsystems that are available for System/370, System/390 architecture, and z/Architecture processors. Examples of such subsystems include CICS, DB2, and IMS. Each of these subsystems is represented in one or more of the LSPR benchmark workloads.

Application Servers

To facilitate the rapid deployment of e-business applications, many production systems are increasing their exploitation of application server software. To reflect this trend, several LSPR benchmarks utilize the WebSphere Application Server (WAS).

Other Program Offerings

Products such as language compilers, linkage editors, and commercial or engineering/scientific programs are typically used by installations in either on-line or batch mode. LSPR measurements include the use of such program products where appropriate.

Application Software

Application software is the custom programming that must be done to make system software perform the specific functions necessary for a business enterprise. LSPR workloads include typical installation-written database application programs for use under CICS, DB2, and IMS (usually written by professional programmers), and typical end-user programs for use in batch, TSO and CMS (which may or may not be professionally written).

Performance Monitors

Software performance monitors are available, both at the SCP level, and at the application subsystem level. It is felt that the LSPR benchmark measurements should use the same performance monitor software as is commonly used in production environments. Doing so not only helps to assure that LSPR workload instruction paths are representative, but also provides a common basis for reporting detailed measurement results.

From the SCP standpoint, both RMF and SMF are used with MVS, OS/390, and z/OS, and the VM Monitor is used with z/VM. Subsystem-specific monitors are also used where applicable, such as the CICS Monitor, IMS/VS Performance Analysis Reporting System (IMS PARS), or DB2 Performance Monitor, for the MVS, OS/390 and z/OS environments.

Workload Content

Another aspect of representativeness is how the actual work is presented to the system. These are the activities (jobstreams or end-user commands) that cause the various forms of software discussed above to be exploited. The two basic ways that work enters a system is via batch submission of jobs, and online entries by an end-user at a terminal or client at a workstation.

Every individual unit of work in a production workload, whether it be a specific job or application, has its own characteristics, and therefore its own unique relationship to the hardware design of the processors being measured. Production workloads normally consist of a large and diverse cross-section of individual jobs and applications, all being managed by the operating system and related software products.

In order for any benchmark to serve a useful purpose, it must represent a rich cross-section of production workload activity. Benchmarks that focus on only one, or just a few individual types of work are very unlikely to provide the proper capacity perspective for an entire production workload.

Batch

Work associated with batch is presented to the system as jobs, read in through a job queue. Initiators select these jobs on a priority and class basis, and guide their progress through the system, obtaining all the resources required to complete the work. Typical batch work includes compile, link-edit, execution of batch oriented production applications, and utility programs (usually involving some form of data manipulation).

Batch benchmarks are generally measured as start-to-finish workloads. The job queue is loaded with a predetermined number of copies of the jobstream to assure a reasonable measurement window. Enough initiators are activated to allow steady-state processor utilization (the period when all initiators are active) to be as close to 100% as possible. The measurement starts when the job queue is released, and finishes when the last job is completed.

Job (or job step) count and processor busy time are combined to compute an ITR value (see formula 2 given earlier).

On-Line

Work associated with on-line systems is generated by end-users sitting at terminals or clients sitting at workstations, entering transactions or commands. Two different types of activity are represented by on-line workloads:

1. Structured Work

End-user transactions are directed toward the manipulation of one or more databases, or some other form of organized data. This type of on-line workload generally consists of a limited set of fixed transaction types that can be requested by end-users, each relating to the purpose of manipulating the relevant data.

2. Unstructured (Ad hoc) Work

This type of on-line work is the effect of providing a wide array of data processing capabilities to end-users, such as program entry and/or testing, file input and editing, use of decision support products, office support and management, on-line queries, and so on.

End-user on-line interactions have attributes relating to their arrival, such as average think time and think time distribution. It is essential that, not only the content of the commands be representative, but that inter-arrival times be representative also.

To benchmark on-line systems, a terminal or client simulator must be used to generate end-user activity. User transactions that comprise the workload are organized into scripts, with each script representing a set of coordinated activities. For each active terminal or client, the simulator assigns a script. From that script, it selects each end-user input in sequence, applies a think time (using representative think time distribution tables), sends the input to the system, and waits for the response before starting on the next input. Terminal and client simulators normally continue to submit commands as a never-ending process.

On-line systems are generally measured as “steady-state” systems, with the processor running at some predetermined target utilization. To reach the desired state, an adequate number of users (terminals or clients) are connected, each immediately starting to execute a script. After the final terminal or client has logged on, and the system has stabilized, a measurement is taken over an elapsed period that is considered a repeatable window of work.

Transaction count and processor busy time are combined to compute an ITR value for on-line workloads (see formula 2 given earlier). Alternatively, external transaction rate (ETR) and processor utilization can be used (see formula 3 given earlier).

Data Considerations

Data processing systems, as the name implies, are designed to manage data. A benchmark workload cannot afford to ignore this aspect of production work. Data exists in many forms and formats. Data files (data sets) and databases are used by the LSPR workloads, as appropriate.

There are two special considerations about data, when performing benchmark measurements, if repeatability is to be assured:

- Data files (datasets) and databases used by a benchmark workload must be restored to their pristine state before each measurement.
- Data files (datasets) and databases must be used in such a way that changes made by the benchmark scripts will not cause the performance of the processor to change significantly over time.

Obviously, data files and databases on a production system do not remain constant. As they get updated and extended, processor (and system) performance can be affected. However, over time, a steady-state data condition is normally achieved.

A benchmark does not have the luxury of waiting for this steady-state data condition to occur. By assuring that the benchmark data is in the same state for each measurement, we know that the processor performance data obtained will be comparable to other measurements made the same way.

LSPR Workload Categories

Introduction

Historically, LSPR workload capacity curves (primitives and mixes) have had application names or been identified by a *software* characteristic. For example, past workload names have included CICS, IMS, OLTP-T, CB-L, LoIO-mix and TI-mix. However, capacity performance has always been more closely associated with how a workload uses and interacts with a particular processor *hardware* design. With the availability of CPU MF (SMF 113) data on z10, the ability to gain insight into the interaction of workload and hardware design in production workloads has arrived. The knowledge gained is still evolving, but the first step in the process is to produce LSPR workload capacity curves based on the underlying hardware sensitivities. Thus the LSPR introduces three new workload capacity categories which replace all prior primitives and mixes.

Fundamental Components of Workload Capacity Performance

Workload capacity performance is sensitive to three major factors: instruction path length, instruction complexity, and memory hierarchy. Let us examine each of these three.

Instruction Path Length

A transaction or job will need to execute a set of instructions to complete its task. These instructions are composed of various paths through the operating system, subsystems and application. The total count of instructions executed across these software components is referred to as the transaction or job path length. Clearly, the path length will be different for each transaction or job depending on the complexity of the task(s) that must be performed. For a particular transaction or job, the application path length tends to stay the same presuming the transaction or job is asked to perform the same task each time. However, the path length associated with the operating system or subsystem may vary based on a number of factors including: a) competition with other tasks in the system for shared resources – as the total number of tasks grows, more instructions are needed to manage the resources; b) the Nway (number of logical processors) of the image or LPAR – as the number of logical processors grows, more instructions are needed to manage resources serialized by latches and locks.

Instruction Complexity

The type of instructions and the sequence in which they are executed will interact with the design of a micro-processor to affect a performance component we can define as “instruction complexity.” There are many design alternatives that affect this component such as: cycle time (GHz), instruction architecture, pipeline, superscalar, out-of-order execution, branch prediction and others. As workloads are moved between micro-processors with different designs, performance will likely vary. However, once on a processor this component tends to be quite similar across all models of that processor.

Memory Hierarchy and "Nest"

The memory hierarchy of a processor generally refers to the caches (previously referred to as HSB or High Speed Buffer), data buses, and memory arrays that stage the instructions and data needed to be executed on the micro-processor to complete a transaction or job. There are many design alternatives that affect this component such as: cache size, latencies (sensitive to distance from the micro-processor), number of levels, MESI (management) protocol, controllers, switches, number and bandwidth of data buses and others. Some of the cache(s) are "private" to the micro-processor which means only that micro-processor may access them. Other cache(s) are shared by multiple micro-processors. We will define the term memory "nest" for a System z processor to refer to the shared caches and memory along with the data buses that interconnect them.

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.

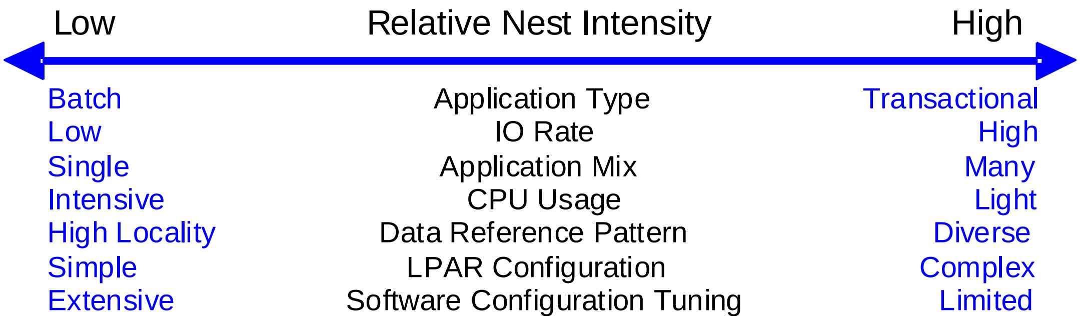
As workloads are moved between processors with different memory hierarchy designs, performance will vary as the average time to retrieve instructions and data from within the memory hierarchy will vary. Additionally, once on a processor this component will continue to vary significantly as the location of a workload's instructions and data within the memory hierarchy is affected by many factors including: locality of reference, IO rate, competition from other applications and/or LPARs, and more.

Relative Nest Intensity

The most performance sensitive area of the memory hierarchy is the activity to the memory nest, namely, the distribution of activity to the shared caches and memory. We introduce a new term, "Relative Nest Intensity (RNI)" to indicate the level of activity to this part of the memory hierarchy. Using data from CPU MF, the RNI of the workload running in an LPAR may be calculated. The higher the RNI, the deeper into the memory hierarchy the processor must go to retrieve the instructions and data for that workload.

Many factors influence the performance of a workload. However, for the most part what these factors are influencing is the RNI of the workload. It is the interaction of all these factors that result in a net RNI for the workload which in turn directly relates to the performance of the workload.

The traditional factors that have been used to categorize workloads in the past are listed along with their RNI tendency in figure 3.



Low	Relative Nest Intensity	High
Batch	Application Type	Transactional
Low	IO Rate	High
Single	Application Mix	Many
Intensive	CPU Usage	Light
High Locality	Data Reference Pattern	Diverse
Simple	LPAR Configuration	Complex
Extensive	Software Configuration Tuning	Limited

Figure 3. Relative Nest Intensity Tendency

It should be emphasized that these are simply tendencies and not absolutes. For example, a workload may have a low IO rate, intensive CPU use, and a high locality of reference – all factors that suggest a low RNI. But, what if it is competing with many other applications within the same LPAR and many other LPARs on the processor which tend to push it toward a higher RNI? It is the net effect of the interaction of all these factors that determines the RNI of the workload which in turn greatly influences its performance.

Note that there is little one can do to affect most of these factors. An application type is whatever is necessary to do the job. Data reference pattern and CPU usage tend to be inherent in the nature of the application. LPAR configuration and application mix are mostly a function of what needs to be supported on a system. IO rate can be influenced somewhat through buffer pool tuning.

However, one factor that can be affected, **software configuration tuning,** is often overlooked but can have a direct impact on RNI. Here we refer to the number of address spaces (such as CICS AORs or batch initiators) that are needed to support a workload. This factor has always existed but its sensitivity is higher with today's high frequency microprocessors. Spreading the same workload over a larger number of address spaces than necessary can raise a workload's RNI as the working set of instructions and data from each address space increases the competition for the processor caches. Tuning to reduce the number of simultaneously active address spaces to the proper number needed to support a workload can reduce RNI and improve performance. In the LSPR, we tune the number of address spaces for each processor type and Nway configuration to be consistent with what is needed to support the workload. Thus, the LSPR workload capacity ratios reflect a presumed level of software configuration tuning. This suggests that re-tuning the software configuration of a production workload as it moves to a bigger or faster processor may be needed to achieve the published LSPR ratios.

Calculating Relative Nest Intensity

Tools available from IBM (zPCR) and several vendors can extract factors from CPU MF data to calculate the RNI.

For z10, the CPU MF factors needed are:

- L2LP: percentage of L1 misses sourced from the local book L2 cache
- L2RP: percentage of L1 misses sourced from a remote book L2 cache
- MEMP: percentage of L1 misses sourced from memory.

These percentages are multiplied by weighting factors and the result divided by 100.

The formula for z10 is:

z10 RNI=(1.0xL2LP+2.4xL2RP+7.5xMEMP)/100.

For z196, zEC12, z13, z14, and z15 the CPU MF factors needed are:

- L3P: percentage of L1 misses sourced from the shared chip-level L3 cache,
- L4LP: percentage of L1 misses sourced from the local drawer L4 cache,
- L4RP: percentage of L1 misses sourced from a remote drawer L4 cache,
- MEMP (percentage of L1 misses sourced from memory).

The formula for z196 is:

z196 RNI=1.67x(0.4xL3P+1.0xL4LP+2.4xL4RP+7.5xMEMP)/100

The formula for zEC12 is:

zEC12 RNI=2.3x(0.4xL3P+1.2xL4LP+2.7xL4RP+8.2xMEMP)/100

The formula for z13 is:

z13 RNI=2.3x(0.4xL3P+1.6xL4LP+3.5xL4RP+7.5xMEMP)/100

The formula for z14 is:

z14 RNI=2.4x(0.4xL3P+1.5xL4LP+3.2xL4RP+7.0xMEMP)/100

The formula for z15 is:

z15 RNI=2.9x(0.45xL3P+1.5xL4LP+3.2xL4RP+6.5xMEMP)/100

For IBM z16 the CPU MF factors needed are:

- L3P: percentage of L1 misses sourced from the shared chip-level VL3 cache
- L4LP: percentage of L1 misses sourced from the local drawer VL4 cache
- L4RP: percentage of L1 misses sourced from a remote drawer VL4 cache
- MEMP: percentage of L1 misses sourced from memory

The formula for IBM z16 is:

IBM z16 RNI=4.3x(0.45xL3P+1.3xL4LP+5.0xL4RP+6.1xMEMP)/100

Note these formulas may change in the future.

Use of Relative Nest Intensity

The intended use of RNI is as a workload characteristic independent of a processor family. The RNI of a workload running on one processor family is intended to be similar to the RNI of that workload running on a different processor family. The weighting factors in the RNI formula are designed to make this come true. For this reason, RNI should not be used as a performance metric. Other metrics available through CPU MF, such as CPI and its components instruction complexity and finite CPI, provide a much more accurate view of hardware performance.

LSPR Workload Categories Based on Relative Nest Intensity

As discussed above, a workload's relative nest intensity is the most influential factor that determines workload performance. Other more traditional factors such as application type or IO rate have RNI tendencies, but it is the net RNI of the workload that is the underlying factor in determining the workload's capacity performance. With this in mind, the LSPR now runs various combinations of former workload primitives such as CICS, DB2, IMS, OSAM, VSAM, WebSphere, COBOL and utilities to produce capacity curves that span the typical range of RNI. The three new workload categories represented in the LSPR tables are described below.

LOW (relative nest intensity): A workload category representing light use of the memory hierarchy. This would be similar to past high scaling primitives.

AVERAGE (relative nest intensity): A workload category representing average use of the memory hierarchy. This would be similar to the past LoIO-mix workload and is expected to represent the majority of production workloads.

HIGH (relative nest intensity): A workload category representing heavy use of the memory hierarchy. This would be similar to the past DI-mix workload.

To understand how production workloads can be matched to LSPR workloads see "**Relating Production Workloads to LSPR Workloads**" in "**Chapter 4. Using LSPR Data.**"

LSPR Workload Primitives

Various combinations of LSPR workload "primitives" have been and continue to be run to create the capacity ratios given in the LSPR tables. Each individual LSPR workload is designed to focus on a major type of activity, such as interactive, on-line database, or batch. The LSPR does not focus on individual pieces of work such as a specific job or application. Instead, each LSPR workload includes a broad mix of activity related to that workload type. Focusing on a broad mix can help assure that resulting capacity comparisons are not skewed.

The LSPR workload suite is updated periodically to reflect changing production environments. High-level workload descriptions are provided below.

z/OS and OS/390

OLTP-T (formerly IMS) - Traditional On-line Workload

The OLTP-T workload consists of moderate to heavy IMS transactions from DLI applications covering diverse business functions, including order entry, stock control, inventory tracking, production specification, hotel reservations, banking, and teller system. These applications all make use of IMS functions such as logging and recovery. The workload contains sets of 12 (17 for OS/390 Version 1 Release 1 and earlier) unique transactions, each of which has different transaction names and IDs, and uses different databases. Conversational and wait-for-input transactions are included in the workload.

The number of copies of the workload and the number of Message Processing Regions (MPRs) configured is adjusted to ensure that the IMS subsystem is processing smoothly, with no unnecessary points of contention. No Batch Message Processing regions (BMPs) are run. IMS address spaces are non-swappable.

This IMS workload accesses both VSAM and OSAM databases, with VSAM indexes (primary and secondary). DLI HDAM and HIDAM access methods are used. The workload has a moderate I/O load, and data in memory is not implemented for the DLI databases.

Measurements are made with z/OS, OS/390, DFSMS, JES2, RMF, VTAM, and IMS/ESA. IMS coat-tailing (enabling reuse of a module already in storage) is not used; since this activity is so sensitive to processor utilization, it could cause distortion when comparing ITRs between faster and slower processors. Beginning with OS/390 Version 1 Release 1, measurements were done with one or more control regions. The number of data base copies, MPR's, and control regions (within the limits of granularity) are scaled with the processing power of a particular machine in-order to assure equal and normalized tuning. Performance data collected consists of IMS PARS, and the usual SMF data, including type 72 records (workload data), and RMF data.

OLTP-W -Web-enabled On-line Workload

The OLTP-W workload reflects a production environment that has web-enabled access to a traditional data base. For the LSPR, this has been accomplished by placing a WebSphere front-end to connect to the LSPR CICS/DB2 workload (described below).

The J2EE application for legacy CICS transactions was created using the CICS Transaction Gateway (CTG) external call interface (ECI) connector enabled in a J2EE server in WebSphere for z/OS Version 5.1. The application uses the J2EE architected Common Client Interface (CCI). Clients access

WebSphere services using the HTTP Transport Handlers. Then, the appropriate servlet is run through the webcontainer, which calls EJB's in the EJB Container. Using the CTG External Call Interface (ECI) CICS is called to invoke DB2 to access the database and obtain the information for the client.
For a description of the CICS and DB2 components of this workload, please see the CICS/DB2 workload description further below.

WASDB - WebSphere Application Server and Data Base

The WASDB workload reflects a new e-business production environment that uses WebSphere applications and a DB2 data base all running in z/OS.

WASDB is a collection of Java classes, Java Servlets, Java Server Pages and Enterprise Java Beans integrated into a single application. It is designed to emulate an online brokerage firm. WASDB was developed using the IBM VisualAge™ for Java and WebSphere Studio tools. Each of the components is written to open Web and Java Enterprise APIs, making the WASDB application portable across J2EE-compliant application servers.

The WASDB application allows a user, typically using a web browser, to perform the following actions:

- Register to create a user profile, user ID/password and initial account balance.
- Login to validate an already registered user.
- Browse current stock price for a ticker symbol.
- Purchase shares.
- Sell shares from holdings.
- Browse portfolio.
- Logout to terminate the user's active interval.
- Browse and update user account information.

CB-L (formerly CBW2)-Commercial Batch Long Job Steps

The CB-L workload is a commercial batch job stream reflective of large batch jobs with fairly heavy CPU processing. The job stream consists of 1 or more copies of a set of batch jobs. Each copy consists of 18 jobs, with 107 job steps. These jobs are more resource intensive than jobs in the CB-S workload (discussed below), use more current software, and exploit ESA features. The work done by these jobs includes various combinations of C, COBOL, FORTRAN, and PL/I compile, link-edit, and execute steps. Sorting, DFSMS utilities (e.g. dump/restore and IEBCOPY), VSAM and DB2 utilities, SQL processing, SLR processing, GDDM™ graphics, and FORTRAN engineering/scientific subroutine library processing are also included. Compared to CB-S, there is much greater use of JES processing, with more JCL statements processed and more lines of output spooled to the SYSOUT and HOLD queues. This workload is heavily DB2 oriented with about half of the processing time performing DB2 related functions.

Measurements are made with z/OS, OS/390, DFSMS, JES2, RMF, and RACF. C/370, COBOL II, DB2, DFSORT, FORTRAN II, GDDM, PL/I, and SLR software are also used by the job stream. Access methods include DB2, VSAM, and QSAM. SMS is used to manage all data. Performance data collected consists of the usual SMF data, including type 30 records (workload data), and RMF data.

The CB-L job stream contains sufficient copies of the job set to assure a reasonable measurement period, and the job queue is pre loaded. Enough initiators are activated to ensure a high steady-state utilization level of 90% or greater. The number of initiators is generally scaled with processing power to achieve comparable tuning across different machines. The measurement is started when the job queue is released, and ended as the last job completes. Each copy of the job set uses its own datasets, but jobs within the job set share data.

ODE-B - On Demand Environment - Batch

The ODE-B workload reflects the billing process used in the telecommunications industry. This is a multi-step approach which includes the initial processing of Call Detail Records (CDR), the calculation of the telephone fees, and the insertion of the created telephone bills in a database. The CDRs contain the details of the telephone calls such as the source and target numbers along with the time and the duration of the call. The CDRs are stored in flat files within a zFS file system. A feeder application reads the CDRs from the files, converts them into XML format and sends them to a queue. An analyzer application reads the messages from the queue and performs analysis on the data. During the analysis further information is retrieved from the relational database, and the same database is subsequently updated with the newly created telephone bill and new records for each call. The feeder and the analyzer applications are implemented as enterprise java beans (EJB) in IBM WebSphere Application Server for z/OS. Using the concept of multi-servant regions, which is unique to the z/OS implementation of WebSphere Application Server, the threads of the feeder and the analyzer applications are distributed over several java virtual machines (JVM). The WebSphere internal queuing engine is used as the queue for the message transport between the feeder and analyzer.

CB-J - JavaBatch

The JavaBatch workload reflects the batch production environment of a clearing bank that uses a collection of java classes working on a DB2 database and a set of flat files in z/OS. JavaBatch is a native, standard Java application that can be run standalone on a single JVM (Java Virtual Machine) or in parallel to itself on multiple JVMs. Each of the parallel applications instances can be tuned separately. All parallel applications are working on the same set of flat files and database tables. The JavaBatch application is based on a Java-JDBC-framework from an external banking software vendor and has been enhanced and adapted using the Websphere Application Developer tool. Various properties such as number of banks, number of accounts, and more can be adapted for the specific runtime environment. These are kept in a special properties file, keeping the java application unchanged.

The JavaBatch application allows a user to perform the following activities:

- initialize the working database
- create a set of flat files, each containing several hundreds to thousands of payments
- read the flat files, perform various syntax-checks and validation for each payment and store the payments to the working database
- read the payments from the database and route them to destination bank's flat files

CICS/DB2 - On-line Workload for pre-z/OS version 1 release 4

The CICS/DB2 workload is an LSPR workload that was designed to represent clients' daily business by simulating the placement of orders and delivery of products, as well as business function like supply and demand management, client demographics and item selling hit list information. The workload consists of ten unique transactions.

CICS is used as a transaction monitor system. It provides both an API for designing the dialogue panels and parameters to drive the interface to the DB2 database. The interface between the two subsystems is fully supported by S/390 and exploits N-Way designs. CICS functions like dynamic workload gathering and function shipping are not exploited in this workload. The CICS implementation uses an MRO model, which is managed by CP/SM. The number of AOR (Address Owning Region) and TOR (Terminal Owning Region) used, depends on the number of engines of the processor under test. The ratio between TOR and AOR is 1:3. The utilization of the TOR and the AOR regions is kept under 60%.

The application database is implemented in a DB2 subsystem. One of the major design efforts was to achieve a read-to-write ratio exhibited by OLTP clients. Several data center surveys indicate an average read-to-write ratio to be in the range of 4:1 - 6:1. The read-to-write ratio is an indication of how much of the accessed data are changed as well. For this CICS/DB2 workload implemented on a S/390 or z/Architecture system and using DB2 as database system, an approximation of the read-to-write ratio is the ratio of SQL statements performing 'read' operation, like select, fetch, open cursor to the 'write' SQL statements, like insert, update, delete.

To reduce the number of database locks and the inter system communication required for each database update and to preserve local buffer coherency in data sharing environments, DB2 type 2 indexes have been used. Additionally, row-level-locking has been introduced for some tables. Each table and index is buffered in separate buffer pools for easy sizing and control.

Linux™ on zSeries

WASDB/L - WebSphere Application Server and Data Base under Linux on IBM Z

The WASDB/L workload reflects an e-business environment where a full function application is being run under Linux on IBM Z in an LPAR partition. For LSPR this was accomplished by taking the WASDB workload (described above under z/OS), and converting it to run both application and data base server in a single Linux on zSeries image. The WASDB/L workload is basically the same as the WASDB workload on z/OS with the exception of being enabled for Linux on zSeries. See the 'WASDB - WebSphere Application Serving and Data Base' section for a detailed description.

z/VM

WASDB/LVm - many Linux on IBM Z guests under z/VM running WebSphere Application Server and Data Base

The WASDB/LVm workload reflects a server consolidation environment where each server is running a full function application. For LSPR this was accomplished by taking the WASDB workload (described above under z/OS), and then replicating the Linux- guest a number of times based on the N-way of the processor. Guest pair activity was then adjusted to achieve a constant processor utilization for each N-way. Thus the ratios between processors of equal N-way are based on the throughput per guest rather than the number of guests.

LSPR Measurement Methodology

Each of the LSPR workloads is run according to a specific set of rules, to assure that results can be compared with other measurements of the same workload. Neither changes to setup or operation, nor unique tuning activities are done to favor any processor. Some of the measurement methodology concerns include:

- Assure adequate configuration (storage, channels, DASD)
- Distribution of system data
- Distribution of program libraries
- Distribution of data files (datasets) and databases
- Files and databases restored to pristine state
- Logon end-users (terminals or clients), or load job queue
- Determine measurement period to obtain a repeatable sample of work
- Adjust activity to realize target processor utilization level
- Assure steady-state has been achieved
- Capture appropriate performance monitor data
- Capture operator console logs
- Verify that no hardware errors occurred
- Verify measurement data against acceptance criteria
- Construct detailed measurement reports

When a suitable testing environment is not available, analysis is used to address these methodology concerns.

As stated earlier, on-line workloads require some type of terminal or client simulator to generate the workload of an end-user community. There are a variety of products available that can serve this purpose, including IBM Teleprocessing Network Simulator (TPNS). Products like TPNS generally run out-board on a stand-alone processor that is connected to the system being benchmarked with normal teleprocessing hardware. Alternatively, TPNS could be run on the host processor (in-board) along with the benchmark workload itself. LSPR will choose between an in-board and out-board network simulator based on the functionality required and the processing overhead associated with the simulation. Generally, out-board network simulators are used.

Chapter 4. Using LSPR Data

The purpose of the LSPR is to provide relative capacity data for System z Architecture processors on which a reliable capacity planning exercise may be based.   As described in Chapter 3, Workload Environments, capacity ratios for a variety of workload environments will be presented. Since each SCP/workload combination may have characteristics that react differently with the design of a particular processor, the LSPR ratios offer insight into the variability that may be experienced by different production workloads. In this chapter, the background and suggested methodologies for using LSPR data are discussed. Note that the examples are drawn from the z/OS 1.4 data tables, but the underlying techniques are applicable using the other SCP tables as well.

LSPR Data Sources

To maximize its usefulness, the LSPR includes as many System z processors as possible. Understanding capacity for older processors is important because they are the ones being migrated from. Understanding capacity for newer processors is important because they are the ones being migrated to. Without accurate LSPR data on both, it would not be possible to develop reliable relative processor capacity expectations.

Insight

All LSPR data is based on measurement and analysis.

***All LSPR data is based on measurement and analysis*.**

Measured Data

LSPR data for IBM processors is generally made available at announce time. IBM's announcement claims are based on LSPR measurements.

Many IBM compatible processor models are also measured for the LSPR, whenever machine time can be obtained.

In addition to actual processor measurements, there are two types of LSPR data which IBM believes carry essentially the same degree of accuracy as measurements. These are based on the actual processor measurements made and various analysis techniques.

- Projections

  Primary models of each processor series are always measured for the LSPR. It is not practical, however, for IBM to allocate the necessary resource to measure every individual processor model announced. For those that are not measured, ITR numbers are projected. A projected ITR is based on the known processor internal design, and on the known characteristics of the subject workload on similarly designed processor models. Projections have been shown to have accuracy comparable to that of measurements, where subsequent testing has occurred.

- Estimates

  Many older processor models that have been measured in the past, are no longer accessible for testing. It is often desirable to maintain these processors in LSPR tables, for those SCPs that are supported. Therefore, as the LSPR moves ahead to more current software levels, ITRs for these processors must be estimated. Estimates are based on known deltas between the older software and the current software on similarly designed processor models.

  As capacities of today's processors grow ever larger, it becomes more difficult to commit the time, and/or the external resources necessary for full, unconstrained LSPR measurements. Experience has shown that there are estimation techniques, based on making reduced resource measurements and analysis, that can yield reliable results.

  Estimated ITRs are considered to have an accuracy very close to that of actual measurements. Estimates are not provided in the LSPR without a high degree of confidence in the process, and in the resulting ITR values.

Relating LSPR Data to MIPS

IBM views MIPS tables as an often imprecise way to express processor capacity. The major problem is that most commonly accepted MIPS tables are insensitive to the SCP and workload environment being run. We must recognize, however, that MIPS is still a commonly used metric in the industry.

When MIPS is the processor capacity metric of choice, LSPR ITR data can still be useful to relate capacity for a potential new machine. If you want to assume that your currently installed processor represents some MIPS value to you, you can directly apply an LSPR ITR ratio for the appropriate workload to assess what a new processor's MIPS value would be. The formula is:

Expected CPU-B MIPS = CPU-A MIPS * (ITR for CPU-B / ITR for CPU-A)

When using the LSPR to compute the MIPS expectation for a new processor, the result will not necessarily line up with those in currently accepted MIPS tables. This is because our MIPS calculation is based on workload sensitive LSPR data. Published MIPS tables do not consider any such SCP or workload sensitivity.

A processor MIPS number determined by the above calculation simply states that, if you believe that your current processor is capable of “X” MIPS, then a new processor should be capable of “Y” MIPS, for the workload environment assumed. You are simply assigning a relative MIPS value for the new processor based on LSPR measurement experience. It does not suggest that IBM rates any processor with MIPS numbers.

Resource Constrained Environments

As stated previously, LSPR measurements are made on processor configurations designed to prevent any significant external constraints. The reason is that we want to represent every processor in its best light. This is the only reasonable way to assure that every processor contained in the LSPR is fairly represented. It is impossible to produce globally useful capacity tables that represent processors in various constrained situations, since there are so many different resource constraint scenarios that could exist.

Insight

LSPR data can be used to assess relative processor capacity for production workloads that are resource constrained.

Production workloads may, of course, incur some types of external constraint. Often these constraints are not terribly significant, while, in other cases they may be. Normally, an external constraint can be relieved by installing more of a resource, such as central storage, expanded storage, channels, controllers, or I/O devices. Generally, it is a price/performance tradeoff, whether to use processor cycles to manage the constraint, or to purchase more of the constrained resource.

LSPR data is useful for assessing capacity for a new processor, even if the current processor has resource constraints. If we assume that the current level of constraint will remain the same on a planned new processor, then the capacity relationship established from the LSPR will apply. If we expect to relieve the constraint when the new processor is installed, then the capacity relationship established from the LSPR is conservative, and should be elevated (this is often the case when moving to more advanced technology hardware). If we expect to incur additional constraint on the planned new processor, then the capacity relationship established from the LSPR is overstated, and should be lowered.

It is beyond the purpose of the LSPR to provide data to evaluate the many various resource constraint scenarios possible. From time to time, IBM does publish technical bulletins containing “case study” information for this purpose. To assess the cost or value of reducing any type of resource constraint, documentation outside the LSPR will need to be consulted. In the absence of such documentation, a study will be required.

New Function

Over time, new function will appear in hardware and/or software. Often a new function is directed at minimizing or eliminating resource constraints. One such example is ESA’s ability to exploit data-in-memory (DIM) in various ways. Whatever the purpose, there is always an interest in any performance or capacity benefit related to the exploitation of new function.

The LSPR's stated purpose is to compare processor capacity when running the same software. It is beyond the purpose of the LSPR to provide data to evaluate new function. As function is introduced, IBM will normally publish technical documents showing the benefits related to exploiting such function. As new function gets accepted in the data processing community, LSPR workloads may be updated to reflect such activity.

It is also beyond the purpose of the LSPR to assess the effects of a software migration. The LSPR ratios in a given table are all at the same software level. While you may be running other levels of SCP or subsystems than the LSPR, most of the software performance differences cancel out when you do a hardware only change. This will result in ratios like the LSPR. Generally, there are technical bulletins that describe software migration performance effects. A combined software and hardware migration should be approached as a two step process that multiplies the hardware ratio (i.e., LSPR) by the appropriate software ratio (from the technical bulletins).

Relating Production Workloads to LSPR Workloads

Historically, there have been a number of techniques used to match production workloads to LSPR workloads such as a) application name (a client running CICS would use the CICS LSPR workload), b) application type (create a mix of the LSPR online and batch workloads), c) IO rate (low IO rates used a mix of the low IO rate LSPR workloads). However, as discussed in the "LSPR Workload Categories" section, the underlying performance sensitive factor is how a workload interacts with the processor hardware. These past techniques were simply trying to approximate the hardware characteristics that were not available through software performance reporting tools. Beginning with the z10 processor, the hardware characteristics can now be measured using CPU MF (SMF 113) COUNTERS data. Thus, the opportunity exists to be able to match a production workload to an LSPR workload category via these hardware characteristics (see the "LSPR Workload Categories" section for a discussion about RNI – Relative Nest Intensity).

The AVERAGE RNI LSPR workload is intended to match the majority of client workloads. When no other data is available, it should be used for a capacity analysis.

DASD IO rate has been used for many years to separate workloads into two categories: those whose DASD IO per MSU (adjusted) is <30 (or DASD IO per PCI <5) and those higher than these values. The majority of production workloads fell into the "low IO" category and a LoIO-mix workload was used to represent them.   Using the same IO test, these workloads would now use the AVERAGE RNI LSPR workload. Workloads with higher IO rates may use the HIGH RNI workload or the AVG-HIGH RNI workload that is included with zPCR.

For z10 and newer processors, the CPU MF data may be used to provide a more accurate workload selection. When available, this data allows the RNI for a production workload to be calculated. Using the RNI and another value from CPU MF, the L1 cache misses per 100 instructions, a workload may be classified as LOW, AVERAGE or HIGH RNI. This classification and resulting workload selection is automated in the zPCR tool. It is highly recommended to use zPCR for capacity sizing. For those wanting to perform the workload selection by hand, the following table may be used for z10, z196, zEC12, z13, z14, z15, and IBM z16 (note L1MP stands for L1 misses per 100 instructions and is a value that may be calculated using the CPU MF counters data):

Table 4. **RNI-based Workload Selection**

L1MP	RNI	Workload Match
<3	>= 0.75	AVERAGE
	< 0.75	LOW
3 to 6	>1.0	HIGH
	0.6 to 1.0	AVERAGE
	< 0.6	LOW
>6	>=0.75	HIGH
	< 0.75	AVERAGE

Note this table may change in the future.

Estimating Utilization for a New Processor

If both the relative capacity of a planned new processor and the utilization of the current processor are known, then that information can be used to estimate the utilization of the new processor after the workload has been moved. For example, if planning to move an existing workload from CPU-A, currently running at some utilization level, to CPU-B, we can approximate the processor utilization expected on CPU-B as follows:

CPU-B Utilization = CPU-A Utilization * (ITR for CPU-A / ITR for CPU-B)

As an example, let’s assume that CPU-A has a mixed workload ITR rating of 10 and is running at 90% utilization today (a peak period average). We are considering moving this workload to CPU-B with a mixed workload ITR rating of 15.   If the workload were moved over to CPU-B without change, you could expect the utilization on the new processor to be 60% (90% × 10 ÷ 15).

Estimating Utilization with Workload Growth

A new processor model is usually chosen so that it can absorb a certain amount of workload growth before another processor change is necessary. If workload growth can be estimated, rates can be applied to the current known production workload, to determine the anticipated utilization level, for the new processor, at points in the future. In this way, one can see when the capacity of the processor will be exceeded, thereby needing to be replaced.

- Latent Demand

  Frequently the need to consider a new processor is driven by the fact that the current machine has already become a constraint to getting work done. In other words, more work would be done if more processor resource were available (making changes to other resources could also produce such an effect). Here we have pent-up demand which will result in an instantaneous workload growth when the new processor is installed. Such growth is called “latent demand”.

- Annual Growth

  Planned (and unplanned) growth over time is referred to as annual growth. As user population increases, or as new/enhanced applications are put into production, growth will occur. By historically tracking growth, one can make assumptions about future annual growth rates.

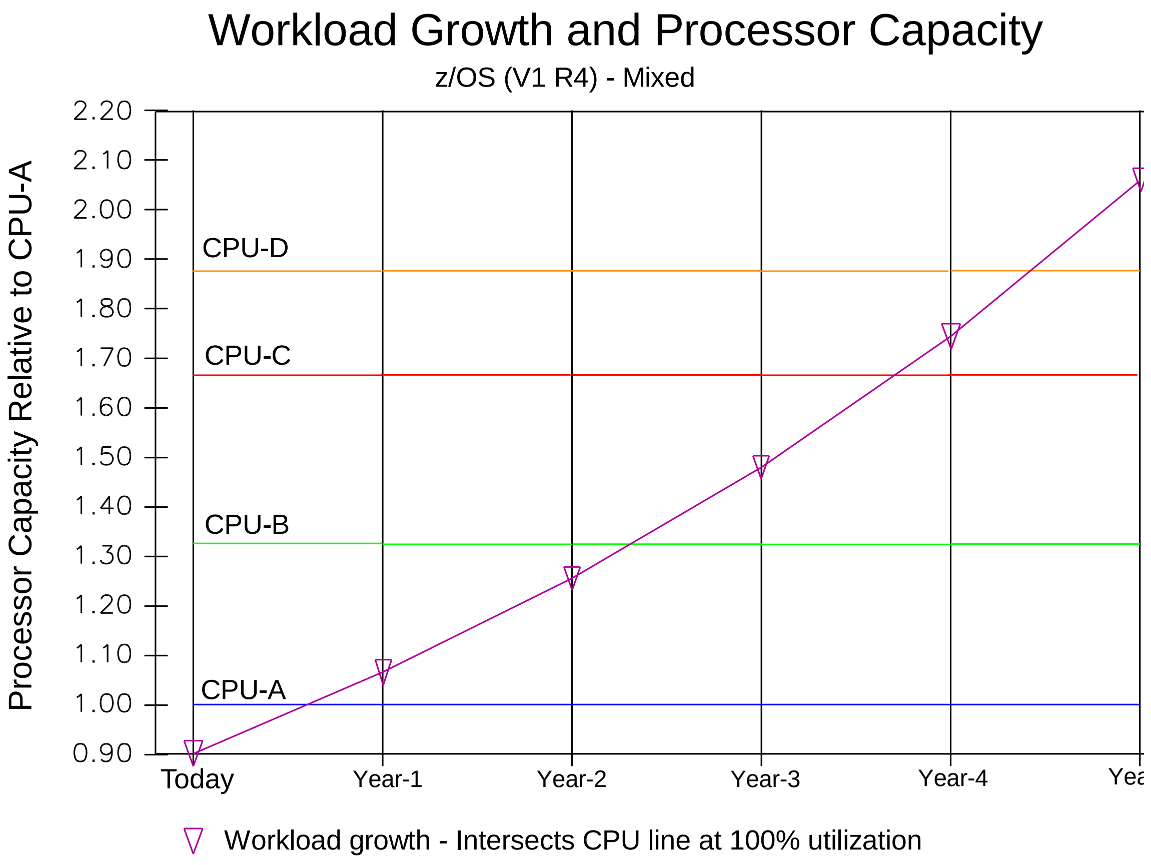
Growth rates can be applied to the average utilization of the current processor to determine what the utilization will be on a new processor as that growth occurs. The formula to adjust the current processor’s utilization for either latent or annual growth is:

CPU-A Utilization = CPU-A Utilization * (1 + Growth rate)

With workload growth

Figure 4 shows a graph, with three horizontal lines representing three potential new processors, each scaled with its capacity relative to a currently installed CPU-A, based on LSPR data for the appropriate workload mix. Workload growth is assumed to be 18%, compounded annually over a five year period.

Figure 4**. Example plotting workload growth against relative processor capacity**



Knowing that the current utilization of CPU-A is 90%, we can start the growth curve at 0.90 of the capacity of the CPU-A. With an annual growth rate of 18%, the growth curve will be at 1.062 for the first year (0.90 × 1.18), 1.253 for the second year (1.062 × 1.18), and so on.

The expected utilization can be computed for any of the processors by dividing the growth value expected at any point in time by the relative capacity of the processor in question.

If latent demand is considered to be a factor in the move to a new processor, its growth should be applied to the growth curve's starting point (today). Annual growth should then be applied to that value.

When the growth curve crosses a processor line, the estimated utilization level on that machine will be 100%, meaning that its capacity will be exceeded. Many installations would consider themselves out of capacity when the average utilization level exceeds some threshold that is less than 100%. Here the growth line can be drawn at that threshold to determine when a processor's capacity will be exceeded.

Note: The utilization estimates above do not address any generic low utilization effects (LUEs), or effects of changing the logically partitioned (LPAR) mode setup during migrations.

Chapter 5. Validating a New Processor’s Capacity Expectation

The decision about which processor to install as a replacement will be based on the relative capacity expectation for a potential new processor. No matter what source for relative processor capacity data was used, one would like to be able to show that the capacity ratios used, were, in fact, correct for the actual production workload.

As discussed in “Customized Benchmark”, processor capacity data derived from a customized benchmark, which has been designed to be representative of the production workload, should be accurate. In this case, you have the desired data before the new processor is installed, and you can have a high degree of confidence in that data.

If a representative benchmark is not possible, processor capacity data must be accepted from one or more outside sources, such as consultant MIPS tables, vendor claims, capacity planning models, or preferably IBM’s LSPR. Whatever source is used, there will be no easy means to validate the expected processor relationship until after the new processor is installed. Validating your expectation after the fact is useful, in that doing so can provide you with some level of confidence in the source of capacity data being used.

Note: Being correct once does not prove infallibility. A one-time validation of a new processor’s capacity expectation is no guarantee that the source used will always be correct. It is always possible that the source happened to have provided the correct capacity relationship for the two processors only by chance (just as it is always possible that a “kernel” might happen to produce the exact capacity ratio that a production workload would realize). That same source might well be in error when a different set of processors are compared.

A Limited View

One way that relative capacity is typically measured is to look at specific pieces of production work, such as certain jobs or applications. By comparing the CPU time to do a single piece of work on the new processor, to the same data for the old processor, you can establish a capacity relationship for that specific piece of work. There are two major problems with this approach:

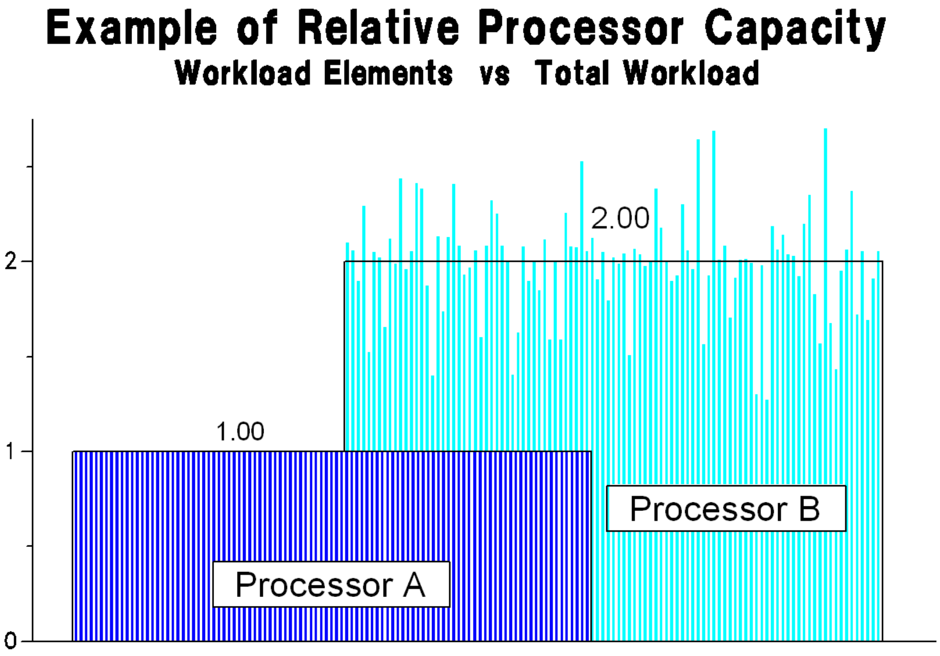
- The sample of work tested is small

  You are only looking at specific portions of the overall workload. Even though they may be running along with the normal workload mix, the numbers represent only that work. The relative capacity number that you should be looking for represents the entire production workload, taken as a whole.

Each unit of work run has its own individual sensitivity to processor design. Processor capacity ratios are even more sensitive for specific individual units of work than they are to general workload types. Capacity ratios for individual work units have a significant potential to be higher or lower than the capacity relationship for the workload as a whole. To determine the overall capacity relationship, you would have to perform our test on all (or most) of the individual work units in the production workload.

Figure 5 shows an example of two processors, where the relative capacity of the second is exactly two times the first. This relationship is determined by computing a ratio between the two measured ITR values for a specific workload environment. When you examine the relative capacity for any given unit-of-work within the overall workload, there can be a significant variation from the average.

Figure 5. Workload element capacity vs total workload capacity



- System task time is difficult to apply

Many operating system functions are necessary to support a workload, including items such as JES, VTAM, RACF, RMF, and VM monitor. These functions also contribute to the overall capacity relationship that is realized, having their own relationship with the individual processor designs. By using only the time to do a specific job or transaction, you are focusing on only a portion of the overall processor time necessary to support the work.

Processor time used by system tasks is normally captured by the operating system. However, because these system tasks support various aspects of the entire workload, it is not easy to accurately apportion that time to the specific work being tested.

- Un-captured time is ignored

  The processing time for individual work is determined from accounting data, or from self-contained timing routines. Data obtained in this way does not include all of the processing time to do the work. In other words, you are dealing only with captured processor time, and are ignoring a portion of the processing time necessary to support the work. This Un-captured processor time is the portion that the SCP cannot assign to a specific job or application.

  The amount of Un-captured time that can occur on a processor varies greatly, being dependent on the overall nature of the workload, and on the level of resource management necessary. Uncaptured time is generally SCP time used for managing resources. Just as workloads have their individual behavior characteristics, so do these SCP routines. Therefore, this Un-captured time can have an influence on the overall capacity relationship between the two machines.

The Complete View

There is a better approach toward validating a relative capacity expectation, that will yield a more realistic view of the actual capacity relationship that was realized. With appropriate planning, a validation can be made with a modest effort and minimal impact.

Because both the old and the new processors are seldom available at the same time, you must capture data when they are available. This means that the old processor must be adequately measured before the upgrade, and the new processor must be adequately measured after the upgrade. Generally, there will be no opportunity to repeat measurements on the old processor after the new one is installed.

For a validation to work, there must be a commitment that the workload run on the new processor be the same as that on the old processor. In other words, there should be no shifting of workloads until after the validation is complete.

In order to provide a realistic view of the relative capacity that was realized, you will need to develop an ITR value for the production workload, on both the old processor and the new processor. As stated in “Internal Throughput Rate (ITR)”, an ITR is computed as:

The factors used to compute the ITRs for our validation have some special considerations, since the production workload is not a controlled benchmark.

- Units of work

  Because a production workload generally consists of a mixture of different types of work (for example, on-line and batch), it becomes difficult to use traditional units, such as jobs or transactions, as the measure of work. Therefore, you need to come up with an alternate unit of work that can be used for this exercise.

  The best approach to solve this dilemma, is to take the average data I/O as the unit of work. These are logical data I/Os or EXCP counts as issued from subsystems and application software. (Physical data I/Os cannot be used, since a logical I/O does not always result in a physical I/O.) All other I/Os, such as paging, should be excluded from consideration. Using the logical data I/O is reasonable, in that the relationship of these I/Os to CPU time on any given processor should remain constant for a given production workload. It is important, however, that you capture logical data I/Os over a long enough period to get a representative view of the average workload.

  When using this approach you must ensure that the workload remains relatively constant over the measurement period (there are statistical techniques to verify this). The use of I/Os as a constant work reference is only valid if the workload characteristics do not vary significantly over the measurement interval.

  Note: The use of data-in-memory may require some special considerations because of the way I/Os are reported.

- Processor busy time

  To be meaningful, an ITR must consider all processor busy time to do the work being measured; that is, Un-captured time must not be excluded. Accounting data does not include Un-captured system overhead time. If using accounting data to determine processor time, it must be adjusted to include any amount that is Un-captured.

  Accounting data usually provides processing time in terms of a single engine. If the machine happens to be an N-way processor, this data must be adjusted to represent the complex as a single entity, rather than simply one engine. Therefore, on an N-way processor, processor busy time is computed as the sum of the individual CP busy times divided by “N”.

To compute an ITR for a production workload, formula 3 may be easier to work with, in that the units needed are more readily available. As stated in “ITR/ETR Relationship”, an alternate way to compute an ITR is:

ITR = ETR / Processor Utilization

For the ETR, substitute the logical data I/O rate (I/Os per elapsed second). Processor utilization is provided directly by most SCP related performance monitors. This utilization value includes all processing time, both captured and uncaptured, and therefore meets our requirement of representing all processor time. Processor utilization should be expressed as a fraction relative to 1.00.

No matter what approach is taken to compare the capacity realized for the two processors, care must be taken that the workload measured on the old processor is the same workload that is measured on the new processor. Therefore, the installation plan for the new processor should exclude any intentional change in the day-to-day production workload during the period of test. There should be no new applications added. Nor should there be any workload balancing attempted (shifting applications or load across different processors). Once the validation is completed, you are free to add applications, and shift workload components around as desired.

For this validity check to be fair, you must be certain that you are looking at equivalent samples of work for both measurements. The best way to ensure repeatable work is to capture a large sample, such as a full weeks worth of data, probably during the normal prime shift hours. Take care that you are not encountering business cycle differences between the two measurement periods. (There are statistical approaches for analyzing the captured data to assure that the samples are, in fact, repeatable work over the measurement period).

A major advantage of using this validation approach is that you are determining relative capacity with the ultimate benchmark, the exact workload for which the processor was purchased. The ITR computed represents all of the processing time (“captured” and “uncaptured”) to do our day-to-day production workload.

The primary disadvantage of this validation approach is that it can only be done after making a commitment to a new processor model. By doing this validation, however, you will be in a position to assess your confidence level in whatever processor capacity reference you used to make your processor decision. In fact, once the validation is completed, the results can also be used to help assess the accuracy of any other processor capacity reference data that could have been used.

Chapter 6. Summary

There is a need for reliable processor capacity planning data across high-end System z processors. Accurate capacity planning exercises and processor upgrade decisions depend on the availability of reliable data. Since workloads with different characteristics may have significantly different performance ratios when moved among various processors, the most reliable data must be workload sensitive. Additionally, LPAR configuration, specialty engines and processor configuration all should be factored into accurate capacity relationships. MIPS tables provide a single-number-metric reflecting average workloads and configurations. LSPR tables provide workload-sensitive ratios. zPCR allows customized estimates to be created that are sensitive to all the afore mentioned factors.

Appendixes

Appendix A. LSPR ITR Ratios for IBM Processors

This appendix is intended as a reference to the Large Systems Performance Reference (LSPR) processor capacity data located on the LSPR website. The website provides capacity ratios for IBM processors, running the various LSPR workload environments under z/OS, z/VM, and Linux on IBM Z.

Data contained on the LSPR website is subject to change to reflect new or additional measurements, or to provide more accurate data based on the latest information available. It is your responsibility to ensure that you are working with the latest version of the data which can be found at the following web address:

https://www.ibm.com/support/pages/ibm-z-large-systems-performance-reference

The primary purpose of the LSPR is to provide relative capacity data for IBM processors, running a variety of SCP and workload environments. The LSPR provides an extensive set of relative capacity data for, z/Architecture processors, across the entire IBM processor line. LSPR ITR data is obtained using representative benchmark workloads, run as laboratory controlled tests, with objective analysis of the results. IBM believes that, with the LSPR data, it has the most exhaustive and accurate set of relative capacity data for z/Architecture processors in the industry.

A VSE ITRR table is not included on the LSPR website. However, in those instances that a VSE performance ratio between an existing machine and a new machine is required, it is expected that the VSE performance ratio will be similar to the performance ratios established for the two processors as determined by the most current ITRR table.

Using the ITR Ratio Values in the Tables

To determine the capacity of any specific processor relative to another for any given SCP and workload, divide the ITR ratio of the 2nd processor by the ITR ratio of the 1st. For the tables contained on the LSPR website, this process is straightforward since all of the tables presented have the same “base” processor.

PR/SM LPAR Considerations / Multi-Image and Single-Image Tables

Depending on the level of the z/OS, the ratios in the LSPR tables are presented in several ways All measurements in support of the tables contained in this document were run in LPAR mode.

It has been observed that the vast majority of System z clients run fairly complex LPAR configurations on their processors, while some clients continue to grow their z/OS single-image size. To address these two environments, two tables of capacity ratios have been provided: 1) a table based on configuring multiple images of z/OS reflecting average configurations across the processor family and 2) a table based on configuring a single image of z/OS equal in size to the number of engines in the processor (subject to the z/OS 1.8 limit of 32).

Extensive profiling of client usage of System z processors has shown that over 95% of the processors have significantly exploited the virtualization capabilities of the System z platform, that is, they are configured with multiple images of z/OS. Thus, the multi-image tables are based on an average multi-image z/OS configuration. The main variables in the configuration are: 1) number of images, 2) size of each image (number of logical engines), 3) relative weight of each image, 4) overall ratio of logical engines to physical engines, 5) the number of drawers, and 6) the number of ICFs/IFLs. The configurations used for the multi-image table are based on the average values for these variables as observed across a processor family. For example, it was found that the average number of images ranged from 5 at low-end models to 8 at the high end. Most systems were configured with 2 major images (those defined with >20% relative weight). On low- to mid-range models, at least one of the major images tended to be configured with a number of logical engines close to the number of physical engines. On high-end boxes, the major images were generally configured with a number of logical engines well below the count of physical engines reflecting the more common use of these processors for consolidation. The overall ratio of logical to physical engines (often referred to as “the level of over commitment” in a virtualized environment) averaged as high as 5:1 on the smallest models, hovered around 2:1 across the majority of models, and dropped to 1.3:1 on the largest models. A majority of models were configured with an additional drawer beyond what would be required to hold the enabled processor engines, and the average model was configured with 2 ICFs/IFLs.

For high-level sizing, most users will find the multi-image table to reflect configurations closest to their own. This is simply due to the fact that most systems are run with multiple z/OS images. However, the most accurate sizings require the zPCR tool which can be customized to exactly match a specific multi-image configuration rather than the average configurations reflected in the multi-image table. The zPCR tool is publicly available.

Your Mileage May Vary

Any processor running a multitude of logical partitions is at increased risk of performance variability from minute to minute or day to day as the workload demands of each partition can affect the performance achieved by all other partitions. The likelihood of significant performance variation increases in proportion to the size (capacity) of the processor and the number of logical partitions that are active. Performance variability may manifest itself in several forms, for example, the capacity realized by an individual logical partition may be impacted as may any charge back algorithm based on CPU time or service units. While the LSPR and zPCR tool can provide good “middle-of-the-road” capacity estimates, your mileage may vary (by minute, hour or day) particularly with larger LPAR configurations.

MSU Values

The column heading MSU (Million of Service Units/hr.) contains the MSU value for each of the machines. The MSUs for z990 and later processor families include adjustments to provide increased IBM software price/performance for applicable IBM software that has MSU-based pricing. Therefore, MSU ratios among processor families do not necessarily reflect the capacity ratios among processor families. These values are provided for information only and the official source for MSU publication is at the following Web address: https://www.ibm.com/it-infrastructure/z/pricing

MSU’s of non-IBM processors are based upon vendor claims of MSU’s. They are not based upon Large Systems Performance Reference measurements.

Appendix B. IBM Capacity Planning Tools

A number of tools have been developed by IBM to assist in understanding the effects on capacity with ***IBM Z*** processors when:

- Considering an upgrade to a new processor
- Implementing logical partitioning on a processor, or changing the partition configuration on a current processor.
- Migrating to a Parallel Sysplex environment
- Upgrading to more current versions of z/OS, CICS, or IMS
- Processor sizing for new new applications
- Understanding the nature of the batch window and the effects of moving to a new processor model

All of these tools are available within IBM for use by your IBM or IBM Business Partner representative, who can assist you in assessing various aspects of capacity planning. Most of these tools must remain with the IBM representative who can work directly with you. Any output generated from these tools can be freely disseminated when capacity planning help is being provided.

The **zPCR** tool, the **zBNA** tool, and **SoftCap** tool are available directly to clients via the website noted in the abstracts on the following pages.

The tools referenced are developed by IBM's *Capacity Planning Support* (CPS) team, a part of IBM's Advanced Technical Sales Support, in Herndon, Virginia.

The following pages include an abstract for each of these tools:

1.	**zPCR**	Processor Capacity Reference for IBM Z
2.	**zCP3000**	Performance Analysis and Capacity Planning for IBM Z
3.	**zBNA**	Batch Network Analyzer Tool for IBM Z
4.	**zPSG**	Processor Selection Guide for IBM Z
5.	**zSoftCap**	Software Migration Capacity Planning Aid for IBM Z
6.	**BWATOOL**	Batch Workload Analysis Tool for IBM Z
7.	**zMCAT**	Migration Capacity Analysis Tool for IBM Z
8.	**zTPM**	Tivoli Performance Modeler for IBM Z

zPCR

Processor Capacity Reference for **IBM Z** and **IBM LinuxONE**

z PCR is a PC-based productivity tool under Windows, designed to provide capacity planning insight for **IBM Z** and **LinuxONE** processors running various z/OS, z/VM, z/VSE , KVM, Linux, SSC, and CFCC workload environments on partitioned hardware. Capacity results are based on IBM's most recently published **LSPR** data for z/OS.

Capacity is presented relative to a user-selected Reference-CPU, which may be assigned any capacity scaling-factor and metric. Function in **zPCR** includes:

1. **LSPR Processor Capacity Ratio Tables**: Displays processor capacity ratios for 5 workload environments. Processor families and workloads displayed are user controlled. Capacity tables provided are:
   - **Multi-image (IBM Z** and **LinuxONE):** Each processor assumes a partition configuration considered typical for the size and N-way of the model. Capacity for General Purpose models or IFL models may be displayed. The multi-image table assumes that every partition is running the same workload. The **LinuxONE** table is limited to IFL models.
   - **Single-image (IBM Z)**: Each processor assumes a single partition with all CPs assigned, up to a reasonable maximum of 30. Capacity for General Purpose models or IFL models may be displayed.
2. **LPAR Configuration Capacity Planning:** For the LPAR host specified, any legitimate partition configuration can be defined. The LPAR host processor can be configured with General Purpose CPs, zAAPs, zIIPs, IFLs, and ICFs where valid. Partitions are then defined, specifying type (General Purpose, IFL, or ICF), SCP/workload, and LP configuration (dedicated or shared with number of logical CPs), and weight/capping assignments. zAAP and zIIP logical CPs are can be associated with a z/OS partition; IFL logical CPs can be associated with a z/VM partition. Capacity projections are generated for each partition as well as the LPAR host as a whole. Partition configurations can be created directly from z/OS RMF data, EDF data (derived from SMF), or from previously saved zPCR studies. Absolute capping is supported for zEC12, zBC12, and later processors. SMT (Simultaneous Multi-Threading) capacity benefit for zIIPs and IFLs is supported for z13 and later processors.

   In ***Advanced-Mode***, multiple LPAR configurations can be defined, allowing direct comparison of LPAR host or individual partition capacity results to previous configurations

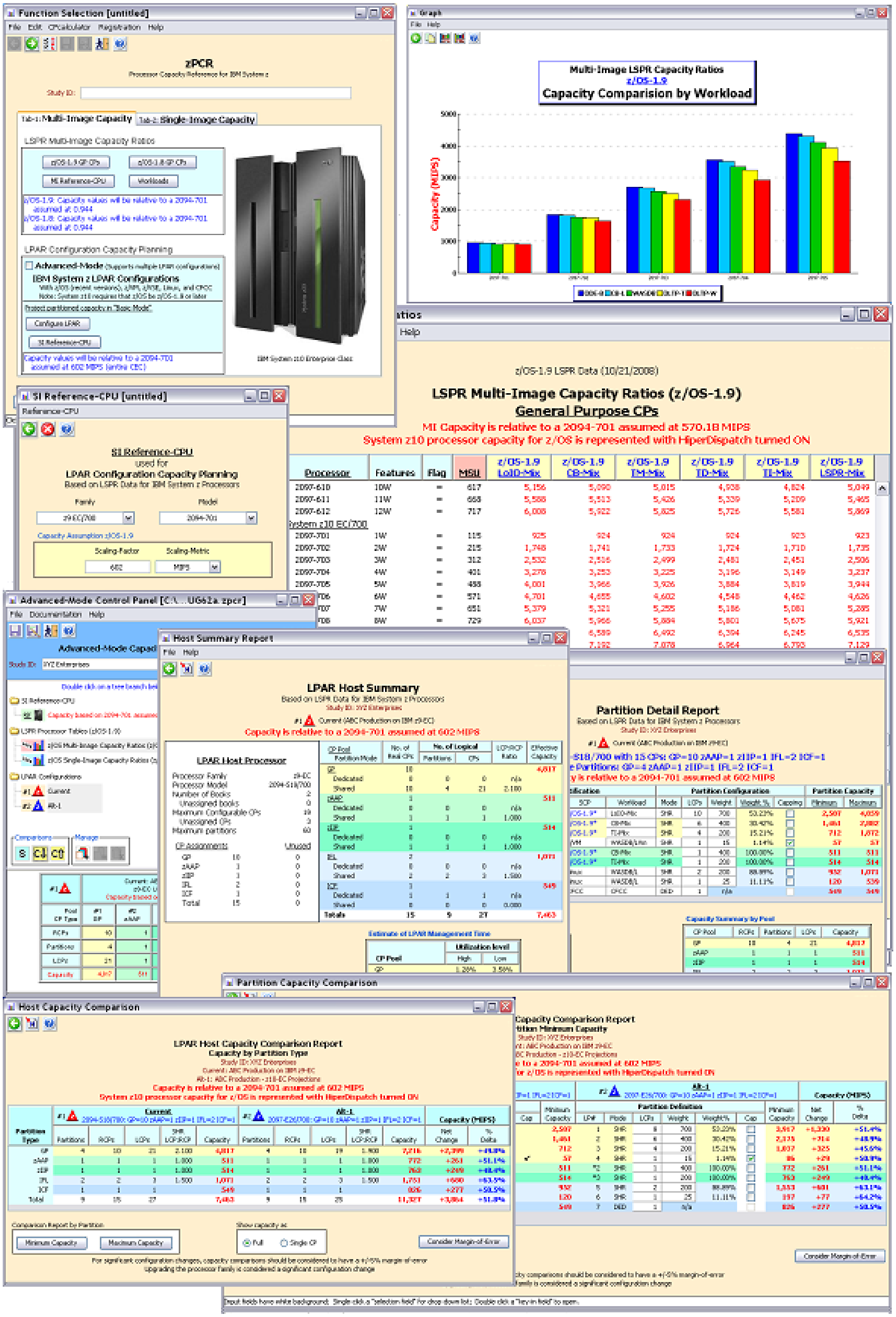
zPCR results are presented in tables and graphs that can be captured for notes, presentations, or handouts. A complete study can be saved for future reference. A User's Guide, integrated online help, and other useful documentation are included.

IBM Clients can obtain **zPCR** via the Internet at:

https://www.ibm.com/support/pages/node/6354029

For questions concerning **zPCR**, contact ***Capacity Planning Support*** via:

- E-mail: zPCR@us.ibm.com



Function Selection [untitled]

File Edit CPcalculator Registration Help

zPCR

Processor Capacity Reference for IBM System z

Study ID:

Tab-1: Multi-Image Capacity | Tab-2: Single-Image Capacity

LSPR Multi-Image Capacity Ratios

z/OS-1.9 GP CPs | z/OS-1.8 GP CPs | MI Reference-CPU | Workloads

z/OS-1.9: Capacity values will be relative to a 2094-701 assumed at 0.944

z/OS-1.8: Capacity values will be relative to a 2094-701 assumed at 0.944

LPAR Configuration Capacity Planning

☐ Advanced-Mode (supports multiple LPAR configurations)

IBM System z LPAR Configurations

With z/OS (recent versions), z/VM, z/VSE, Linux, and CFCC

Note: System z10 requires that z/OS be z/OS-1.8 or later

Protect partitioned capacity in "Basic Mode"

Configure LPAR

SI Reference-CPU

Capacity values will be relative to a 2094-701 assumed at 602 MIPS (entire CEC)

IBM System z10 Enterprise Class

Graph

File Help

Multi-Image LSPR Capacity Ratios

z/OS-1.9

Capacity Comparision by Workload

LSPR Multi-Image Capacity Ratios

Help

z/OS-1.9 LSPR Data (10/21/2008)

LSPR Multi-Image Capacity Ratios (z/OS-1.9)

General Purpose CPs

MI Capacity is relative to a 2094-701 assumed at 570.18 MIPS

System z10 processor capacity for z/OS is represented with HiperDispatch turned ON

Processor	Features	Flag	MSU	z/OS-1.9 LoIO-Mix	z/OS-1.9 CB-Mix	z/OS-1.9 TM-Mix	z/OS-1.9 TD-Mix	z/OS-1.9 TI-Mix	z/OS-1.9 LSPR-Mix
2097-610	10W	=	617	5,156	5,090	5,015	4,938	4,824	5,049
2097-611	11W	=	668	5,588	5,513	5,426	5,339	5,209	5,465
2097-612	12W	=	717	6,008	5,922	5,825	5,726	5,581	5,869
System z10 EC/700									
2097-701	1W	=	115	925	924	924	924	923	923
2097-702	2W	=	215	1,748	1,741	1,733	1,724	1,710	1,735
2097-703	3W	=	312	2,532	2,516	2,499	2,481	2,451	2,506
2097-704	4W	=	401	3,278	3,253	3,225	3,196	3,149	3,237
2097-705	5W	=	488	4,001	3,966	3,926	3,884	3,819	3,944
2097-706	6W	=	571	4,701	4,655	4,602	4,548	4,462	4,626
2097-707	7W	=	651	5,379	5,321	5,255	5,186	5,081	5,285
2097-708	8W	=	729	6,037	5,966	5,884	5,801	5,675	5,921
					6,589	6,492	6,394	6,245	6,535
					7,192	7,078	6,964	6,793	7,129

SI Reference-CPU [untitled]

Reference-CPU

SI Reference-CPU

used for

LPAR Configuration Capacity Planning

Based on LSPR Data for IBM System z Processors

Family: z9 EC/700 | Model: 2094-701

Capacity Assumption z/OS-1.9

Scaling-Factor: 602 | Scaling-Metric: MIPS

Advanced-Mode Control Panel [C:\...UG62a.zpcr]

File Documentation Help

Advanced-Mode Capacity

Study ID: XYZ Enterprises

SI Reference-CPU

LSPR Processor Tables (z/OS-1.9)

z/OS Multi-Image Capacity Ratios

z/OS Single-Image Capacity Ratios

LPAR Configurations

#1 Current

#2 Alt-1

Comparisons | Manage

Host Summary Report

File Help

LPAR Host Summary

Based on LSPR Data for IBM System z Processors

Study ID: XYZ Enterprises

#1 Current (ABC Production on IBM z9-EC)

Capacity is relative to a 2094-701 assumed at 602 MIPS

LPAR Host Processor

Processor Family	z9-EC
Processor Model	2094-S18/700
Number of Books	2
Unassigned books	0
Maximum Configurable CPs	18
Unassigned CPs	3
Maximum partitions	60

CP Assignments		Unused
GP	10	0
zAAP	1	0
zIIP	1	0
IFL	2	0
ICF	1	0
Total	15	0

CP Pool / Partition Mode	No. of Real CPs	No. of Logical Partitions	No. of Logical CPs	LCP/RCP Ratio	Effective Capacity
GP	10				4,817
Dedicated	0	0	0	n/a	
Shared	10	4	21	2.100	
zAAP	1				511
Dedicated	0	0	0	n/a	
Shared	1	1	1	1.000	
zIIP	1				514
Dedicated	0	0	0	n/a	
Shared	1	1	1	1.000	
IFL	2				1,071
Dedicated	0	0	0	n/a	
Shared	2	2	3	1.500	
ICF	1				549
Dedicated	1	1	1	n/a	
Shared	0	0	0	0.000	
Totals	15	9	27		7,463

Estimate of LPAR Management Time

CP Pool	Utilization level High	Utilization level Low
GP	1.28%	3.56%

Partition Detail Report

Based on LSPR Data for IBM System z Processors

Study ID: XYZ Enterprises

#1 Current (ABC Production on IBM z9-EC)

S18/700 with 15 CPs: GP=10 zAAP=1 zIIP=1 IFL=2 ICF=1

Partitions: GP=4 zAAP=1 zIIP=1 IFL=2 ICF=1

Capacity is relative to a 2094-701 assumed at 602 MIPS

Capacity Summary by Pool

Partition Capacity Comparison

Capacity Comparison Report

Partition Minimum Capacity

Study ID: XYZ Enterprises

Current: ABC Production on IBM z9-EC

Alt-1: ABC Production - z10-EC Projections

Capacity is relative to a 2094-701 assumed at 602 MIPS

System z10 processor capacity for z/OS is represented with HiperDispatch turned ON

Consider Margin-of-Error

Input fields have white background; Single click a "selection field" for drop down list; Double click a "key-in field" to open.

Host Capacity Comparison

LPAR Host Capacity Comparison Report

Capacity by Partition Type

Study ID: XYZ Enterprises

Current: ABC Production on IBM z9-EC

Alt-1: ABC Production - z10-EC Projections

Capacity is relative to a 2094-701 assumed at 602 MIPS

System z10 processor capacity for z/OS is represented with HiperDispatch turned ON

Partition Type	Current #1 2094-S18/700: GP=10 zAAP=1 zIIP=1 IFL=2 ICF=1 Partitions	RCPs	LCPs	SHR LCP:RCP	Capacity	Alt-1 #2 2097-E26/700: GP=10 zAAP=1 zIIP=1 IFL=2 ICF=1 Partitions	RCPs	LCPs	SHR LCP:RCP	Capacity	Capacity (MIPS) Net Change	% Delta
GP	4	10	21	2.100	4,817	4	10	19	1.900	7,216	+2,399	+49.8%
zAAP	1	1	1	1.000	511	1	1	1	1.000	772	+261	+51.1%
zIIP	1	1	1	1.000	514	1	1	1	1.000	762	+249	+48.4%
IFL	2	2	3	1.500	1,071	2	2	3	1.500	1,751	+680	+63.5%
ICF	1	1	1		549	1	1	1		826	+277	+50.5%
Total	9	15	27		7,463	9	15	25		11,327	+3,864	+51.8%

Comparison Report by Partition: Minimum Capacity | Maximum Capacity

Show capacity as: ● Full ○ Single CP

Consider Margin-of-Error

For significant configuration changes, capacity comparisons should be considered to have a +/-5% margin-of-error

Upgrading the processor family is considered a significant configuration change

zPCR Window Examples

zCP3000 Performance Analysis and Capacity Planning

zCP3000 is a PC-based productivity tool that runs under Windows. It is designed to do performance analysis and capacity planning functions for ***IBM Z*** processors, running various SCP and workload environments. It can also be used to graphically analyze logically partitioned processors and DASD configurations. Input normally comes from the client's system logs via a separate tool:

1. From ***z/OS SMF***, using **CP2KEXTR**
2. From ***VM Monitor***, using **CP3KVMXT**
3. From ***VSE CPUMON***, using **VSE2EDF**

Some of **zCP3000**'s features include:

- Enterprise Analysis
- Workload Analysis
- Coupling Facility Analysis
- Performance Index Analysis
- zIIP DRDA Analysis
- Capacity Report with graphs and "SmartText"
- zTPM (simulation) Interface both passing configuration information to and integrating data from zTPM into zCP3000 Report document.
- Consolidation/Decentralization Analysis (Quick Migration Mode)
- Performance Variable Analysis
- Metrics Report
- Enterprise DASD Analysis
- Print Processor List
- Sysplex Aggregation Report
- Optional CSV output
- Merge workloads
- Tape Analysis
- LPAR Analysis
- Sysplex Analysis
- CF Structure Analysis
- zAAP/zIIP Potential Analysis
- ZIIP IPSec Analysis
- Alternate Processor Analysis
- CF Link Type/Distance Analysis
- Growth Analysis
- Health Check Analysis
- ESCON/FICON Aggregation Analysis
- Data Set Analysis
- RIOC Adjustment
- SYSID DASD Analysis
- Capture Ratios
- SYSID Storage Analysis

zBNA IBM Z Batch Network Analysis Tool

zBNA is a PC-based productivity tool under Windows, designed to understand the batch window as follows:

- Perform “what if” analysis and estimate the CPU upgrade effect on batch window
- Identify job time sequences based on a graphical view
- Filter jobs by attributes like CPU time / intensity, job class, service class, etc.
- Review the resource consumption of all the batch jobs
- Drill down to the individual steps to see the resource usage
- Identify candidate jobs for running on different processors
- Identify jobs with speed of engine concerns (top tasks %)

Scope of Analysis

- Data Inputs
  - Provide Extractor job run on client systems to capture the data
  - SMF 70, 72,
  - SMF 30 records (subtype 4 for step info and subtype 5 for job info)
- Scope of Analysis
  - Scope is primarily single batch window of user defined length
  - What if analysis is how that specific batch window would run in a different environment on an alternate processor
  - Single system view
- Tool Filters
  - Discovered from the data
    - Service classes, job classes, account codes
  - Settable by user
    - Time Window, CPU Seconds, CPU Intensity, Task Intensity, Exclude Jobs, Key Jobs
- Output
  - Save the study (filters, and file names)
- Generate a suite of output reports

IBM clients can obtain **zBNA** via the Internet at:
http://www.ibm.com/support/techdocs/wscmastr.nsf/WebIndex/PRS5132

For questions concerning **zBNA**, contact ***Capacity Planning Support*** via:

- E-mail: **zPCR@us.ibm.com**

zPSG Processor Selection Guide

zPSG is a PC-based productivity tool under Windows. It is designed to provide sizing approximations for ***IBM Z*** processors intended to host a new application, planned to be implemented using popular, commercially available software products. Current application support includes:

- WebSphere Application Server on z/OS or Linux
- WebSphere Portal Server on z/OS or Linux
- Business Process Manager on z/OS or Linux
- WebSphere Message Broker on z/OS or Linux
- **WebSphere MQ** on z/OS or Linux
- WebSphere Enterprise Service Bus on z/OS or Linux
- ODM Decision Server Rules on z/OS or Linux
- ODM Decision Server Events on z/OS or Linux
- DB2 Transaction on z/OS
- DB2 Data Warehouse on z/OS
- Apache Webserver on Linux

Additional software products will be added to **zPSG** depending on requirements and the availability of capacity planning data.

For each application, you will characterize the average transaction or the average user, selecting features of the software than will be exploited, and the frequency of their use. The application is sized to a single IBM Z processor within the tool (the specific processor model is not surfaced). Then using an LSPR workload category deemed representative of the application, capacity projections are developed for all System z processors. Capacity is given in terms of the expected utilization or in terms of the expected transaction rate that can be supported at a given utilization (SDP or Saturation Design Point). Projections are available for any/all of the processors that are included in the associated LSPR table. A summary report is also available, documenting the capacity estimate in terms of MIPS, estimated zAAP/zIIP eligibility, sizing inputs, and assumptions. Results are presented in tables and graphs which can be captured for notes, presentations, or handouts. Studies can be saved for future reference.

zPSG is installed as a stand-alone tool.

zSoftCap

Software Migration Capacity Planning Aid for IBM Z

zSoftCap is a PC-based productivity tool under Windows, designed to assess the effect on IBM Z processor capacity, when planning to upgrade to a more current operating system version and/or major subsystems versions. **zSoftCap** assumes that the hardware configuration remains constant while the software version or release changes. The capacity implication of an upgrade for the software components can be assessed independently or in any combination.

z/OS Environments Supported

- **z/OS**: **z/OS V1R5** through **z/OS V2R5**
- **CICS**: **CICS TS-1.1** through **CICS TS-5.6**
- **IMS**: **IMS V4** through **IMS V15**

Input required by **zSoftCap** includes the current z/OS version/release and the utilization represented by each of the following components: **Batch**, **CICS**, **DB2**, **IMS**, **Web**, and **System**. The planned future z/OS version/release must also be specified. The processor family, the N-way of the image, and the use of HiperDispatch (supported on z10 and later) are all considered.

For **CICS** and **IMS** software upgrades (optional), both the current and planned version/release, and a high-level description of the subsystem's implementation is required.

z/VSE Environments Supported

- **z/VSE**: **z/VSE/ESA-5.1** through **z/VSE-6.2**
- **CICS**: **CICS/TS-1.1** through **CICS/TS-2.2** (dependent on z/VSE version)

Information Provided

Results show the effective change in processor utilization and the net benefit or cost in capacity that can be expected when moving to newer software versions. For z/OS environments, if upgrading multiple components, a report is available showing the effect of each as well as their combined effect on capacity.

IBM Clients can obtain **zSoftCap** via the Internet at:

www.ibm.com/support/pages/node/6354117

For questions concerning **zSoftCap**, contact Capacity Planning Support via E-mail:

zPCR@us.ibm.com

CPU time projections are based solely on processor speed, using IBM LSPR data for z/OS batch-oriented workloads. Elapsed time projections are based on processor speed, queue time, and I/O time.

Various parameters are available to filter the SMF data, to specify the current host model and target, to define the type of batch involved (CPU intensive, I/O intensive, or somewhere between), and control the various reports.

Five types of reports are available:

1. Job report
2. Job step report
3. Job time line report
4. Critical path report

5. Job information report.

Some of the reports include graphs.

A complete User’s Guide describes how one can use the tool and interpret the results.

zMCAT Migration Capacity Analysis Tool

Capacity expectation for a new ***IBM Z*** processor model, relative to a currently installed processor, is usually determined by comparing the capacity of benchmark workloads such as those carried in LSPR data. Applications such as the ***Processor Capacity Reference for IBM Z (zPCR)*** enable the comparison of processors for a variety of workloads and workload mixes. But how close is the benchmark workload to your client's real production workload? Is there a way to get a better idea of the actual change in throughput seen on the production system?

The ***IBM Z Migration Capacity Analysis Tool*** (**zMCAT**) is intended to do just that. It can be used to compare the performance of production workloads before and after migration of the system image to a new processor, even if the number of engines on the processor has changed. Workloads for which performance is to be analyzed must be carefully chosen because the power comparison may vary considerably due to differing use of system services, I/O rate, instruction mix, storage reference patterns, etc. This is why "your mileage may vary" from an internal throughput ratio (ITRR) based on LSPR benchmark data.

zMCAT is a workstation-based tool that enables interactive filtering of workloads using a variety of parameters, for example, excluding workloads that have too small a sample, or with too much variance. The upgrade analysis can focus on either batch jobs or the intervals of the online regions. Input to zMCAT originates in the SMF type 30 records that are processed by **ZMCATX**, the ***zMCAT Extractor*** utility. Multiple days or weeks in both the before and after period can be collected all at once and then transmitted to the workstation for analysis. The data are then used to establish consistent comparisons. The primary result of running a zMCAT analysis is the calculation of the processor speedup. As user filtering progresses, this value is kept current and displayed to the user. While the estimated impact of the upgrade is derived directly from the measurements of the workloads from the client's environment, zMCAT also allows for the studied judgment of the analyst through the intervention of workload selection.

zTPM   Tivoli Performance Modeler

zTPM (Tivoli Performance Modeler) is a productivity tool designed to let you build a model of a z/OS based ***IBM Z*** processor, and then run various “what if scenarios”. **zTPM** uses simulation techniques to let you model the impact of changes on individual workload performance. **zTPM** runs on a Windows based PC. **zTPM** uses RMF or CMF reports as input. Based on these reports, **zTPM** can create summary charts showing LPAR as well as workload utilization. An automated Build function lets you build a model that represents the system for any reporting interval. Once the model is built, you can make changes to see the impact on workload performance. These can be changes to workload volumes or changes to the hardware configuration. You can model the impact of combining system images from multiple processors onto a single processor, or model the impact of workload growth over several predefined time intervals. You can also model the impact of changing LPAR definitions. **zTPM** also allows you to estimate the impact of latent demand when replacing a processor running at high utilization. A Wizard feature makes it simple to define multiple scenarios on a single screen. Once the scenarios are defined, **zTPM** will build and execute each modeling run to give you the results. **zTPM** also includes a 123 and Excel spreadsheet. These spreadsheets contain several buttons which simplify the process of consolidating the results in a single place, and charting the results in a user friendly format. Coincident with the announcement of the z10 family of processors, **zTPM** has been updated to support **HiperDispatch for z10 processors.**

An external version of **zTPM** is also available for sale to clients from the IBM Software Division as the ***Tivoli Performance Modeler for z/OS*** (5698-A18).

LSPR FAQ: z/OS V2R4 and z/VM

What are the major changes to the z/OS V2R4 LSPR?

The LSPR ratios reflect the range of performance between IBM Z servers as measured using a wide variety of application benchmarks. The latest release of LSPR continues with the methodology introduced with the z/OS V1R11 LSPR. Prior to that version, workloads had been categorized by their application type or software characteristics (for example, CICS®, OLTP-T, LoIO-mix). With the introduction of CPU MF (SMF 113) data starting with the z10 processor, insight into the underlying hardware characteristics that influence performance was made possible. The LSPR defines three workload categories, LOW, AVERAGE, HIGH, based on the metric called “Relative Nest Intensity (RNI)” which reflects a workload’s use of a processor’s memory hierarchy. For details on RNI and the workload categories, please reference the LSPR documentation or go to https://www.ibm.com/support/pages/ibm-z-large-systems-performance-reference

What is the multi-image table in the LSPR?

Typically, IBM Z processors are configured with multiple images of z/OS. Thus, the LSPR continues to include a table of performance ratios based on average multi-image z/OS configurations for each processor model as determined from the profiling data. The multi-image table is used as the basis for setting MIPS and MSUs for IBM Z processors.

What multi-image configurations are used to produce the LSPR multi-image table?

A wide variety of multi-image configurations exist. The main variables in a configuration typically are: 1) number of images, 2) size of each image (number of logical engines), 3) relative weight of each image, 4) overall ratio of logical engines to physical engines, 5) the number of drawers, and 6) the number of ICFs/IFLs. The configurations used for the LSPR multi-image table are based on the average values for these variables as observed across a processor family. It was found that the average number of images ranged from five at low-end models to nine at the high end. Most systems were configured with two major images (those defined with >20% relative weight). On low- to mid-range models, at least one of the major images tended to be configured with a number of logical engines close to the number of physical engines. On high-end boxes, the major images were generally configured with a number of logical engines well below the count of physical engines reflecting the more common use of these processors for consolidation. The overall ratio of logical to physical engines (often referred to as “the level of processor over-commitment” in a virtualized environment) averaged as high as 3:1 on the smallest models, hovered around 2:1 across the majority of models, and dropped to 1.3:1 on the largest models. The majority of models were configured with one drawer more than necessary to hold the enabled processing engines, and an average of 3 ICFs/IFLs were installed.

Can I use the LSPR multi-image table for capacity sizing?

For high-level sizing, the multi-image table may be used. However, the most accurate sizing requires using the **zPCR tool’s *LPAR Configuration Capacity Planning*** function, which can be customized to exactly match a specific multi-image configuration rather than the average configuration reflected in the multi-image LSPR table.

What model is used as the “base” or “reference” processor in the z/OS V2R4 LSPR table?

The 2094-701 processor model is used as the base in the z/OS V2R4 table. Thus, the ITRR for the 2094-701 appears as 1.00.
Note that in zPCR the reference processor may be set at the user’s discretion.

What “capacity scaling factors” are commonly used?

The LSPR provides capacity ratios among various processor families. It has become common practice to assign a capacity scaling value to processors as a high-level approximation of their capacities. The commonly used scaling factors can change based on the version of LSPR. For z/OS V2R4 studies, the capacity scaling factor commonly associated with the reference processor set to a 2094-701 is 593 which is unchanged from that used originally with z/OS V1R11. This value reflects a 2094-701 configured with a *single image* of z/OS - no complex LPAR configuration (i.e., multiple z/OS images) effects are included. For the z/OS V2R4 multi-image table the commonly used scaling factor is 0.944x593=559.792. Note the 0.944 factor reflects the fact that the multi-image table has processors configured based on the average client LPAR configuration; on a 2094-701, the cost to run this complex configuration is approximately 5.6%. The commonly used capacity scaling values associated with each model of a processor may be approximated by multiplying the AVERAGE column of ITRRs in the LSPR z/OS V2R4 multi-image table by 559.792. The PCI (Processor Capacity Index) column in the z/OS V2R4 multi-image table shows the result of this calculation. Note that the PCI column was actually calculated using zPCR, thus the full precision of each ITRR is reflected in the values. Minor differences in the resulting PCI calculation may be observed when using the rounded values from the LSPR table.
Of course, using a table of values based on a capacity scaling factor only allows for a gross approximation of the relative capacities among the processor models A more accurate analysis may be conducted by using zPCR to perform a detailed LPAR configuration assessment to develop the capacity ratio between a “before” and “after” configuration.

How much variability in performance should I expect when moving a workload to a z16 processor?

As with the introduction of any new server, workloads with differing characteristics will see variation in performance when moved to a z16. The performance ratings for a server are determined by the performance of a reference workload that represents what we understand to be the major components of our clients' production environments. While we feel the ratings provide good "middle-of-the-road" values, we also recognize some clients' workloads will differ somewhat from the reference workload we used. The z16 has improvements in its microprocessor design and in its memory hierarchy. However, workloads with different characteristics will see varying performance values from these changes. It is expected that the range of variation in performance of workloads will be similar to that seen in recent processor generations.

Once my workload is up and running on a z16, how much variability in performance will I see?

Minute-to-minute, hour-to-hour and day-to-day performance variability generally grows with the size (capacity) of the server and the complexity of the LPAR configuration. With its improved microprocessor and memory hierarchy design and support for larger numbers of engines, the z16 provides an increase in capacity over the largest previous server in each family. Continued enhancements to z/OS HiperDispatch have been made to help reduce the potential for increased performance variability. In the spirit of autonomic computing, PR/SM™ and the z/OS dispatcher cooperate to automatically place and dispatch logical partitions to help optimize the performance of the hardware and minimize the interference of one partition to another. However, while the average performance of workloads is expected to remain reasonably consistent when viewed at small increments of time or by individual jobs or transactions, some variation in performance might be seen simply due to the expected larger and more complex LPAR configurations that can be supported by the z16.

How do I get performance information for my TPF products running on a z16?

TPF provides “Workload Specifics ITRRs” separately from the LSPR tables. For more information please contact your TPF Support Representative or send a request to tpfqa@us.ibm.com.

What is z/OS HiperDispatch and how does it impact performance?

z/OS HiperDispatch is the z/OS exploitation of PR/SM’s Vertical CPU Management (VCM) capabilities and is exclusive to IBM Z processors since the IBM System z10®. Rather than dispatching tasks randomly across all logical processors in a partition, z/OS will tie tasks to small queues of logical processors and dispatch work to a “high priority” subset of the logical processors. PR/SM provides processor topology information and updates to z/OS and ties the high priority logical processors to physical processors. HiperDispatch can lead to improved efficiency in both the hardware and software in the following two manners: 1) work may be dispatched across fewer logical processors therefore reducing the “multi-processor (MP) effects” and lowering the interference among multiple partitions; 2) specific z/OS tasks may be dispatched to a small subset of logical processors which PR/SM will tie to the same physical processors thus improving the hardware cache re-use and locality of reference characteristics such as reducing the rate of cross-drawer communication. Note the value of HiperDispatch is higher on the IBM zEnterprise 196 (z196) and later processors due to their sensitivity to the chip-level shared cache topology.
A white paper is available concerning z/OS HiperDispatch at:
https://www.ibm.com/support/pages/zos-planning-considerations-hiperdispatch-mode

What is z/VM HiperDispatch and how does it impact performance?

z/VM HiperDispatch is the z/VM exploitation of PR/SM's Vertical CPU Management (VCM) capabilities. z/VM HiperDispatch improves CPU efficiency by causing the z/VM Control Program to run virtual servers in a manner that recognizes and exploits IBM Z machine topology to increase the effectiveness of physical machine memory cache. This includes: a) requesting PR/SM to handle the partition's logical processors in a manner that exploits physical machine topology, b) dispatching virtual servers in a manner that tends to reduce their movement within the partition's topology and c) dispatching multiprocessor virtual servers in a manner that tends to keep the server's virtual CPUs close to one other within the partition's topology. z/VM HiperDispatch can also improve performance by automatically tuning the LPAR's use of its logical CPUs to try to use only those logical CPUs to which it appears PR/SM will be able to deliver a full physical processor's worth of computing power. This includes: a) sensing and forecasting key indicators of workload intensity and b) autonomically configuring the z/VM system not to use underpowered logical CPUs.

An article is available concerning z/VM HiperDispatch at:
http://www.vm.ibm.com/perf/tips/zvmhd.html

What is the performance improvement a z/VM client might experience on the z16?

The performance ratios a z/VM client workload might experience when migrating to z16 from older processors will vary. For the z/VM LSPR curves, a single workload having characteristics similar to the AVERAGE relative nest intensity workload was used. However, client workloads have been shown to cover the full range from LOW to HIGH RNI workloads. Thus, it is suggested that you consider the full range of LSPR workloads.

Where can I read more about the performance of z/VM?

The z/VM Performance Resources Page, located at http://www.vm.ibm.com/perf/, contains information on z/VM performance.

What is the performance improvement a z/VSE client might experience on the z16?

The performance ratios that a z/VSE client workload might experience when migrating to a z16 are represented by the range of ratios for a comparable z/OS migration. For example, the published ratio in the LSPR between the z15 702 and the z16 702 is approximately 9% to 12%. z/VSE workloads should expect this same range of performance for this migration. Consult the LSPR for other examples of moves to z16.

Where can I get more information on the zPCR (Processor Capacity Reference for z) tool?

https://www.ibm.com/support/pages/getting-started-zpcr-ibms-processor-capacity-reference