Recommendations On Configuration of IBM LinuxONE Systems for Large-Scale Linux Deployment

IBM

Executive Summary/Abstract

This whitepaper is intended to provide guidance to IBM clients, and the IBM employees who support them, in the deployment of large-scale Linux® workloads on IBM® LinuxONE systems. The recommendations here are designed to balance efficient scaling, consistent performance, and ease of operational management and to cover a range of enterprise workloads.

These recommendations generally apply to Linux on both IBM® LinuxONE and IBM Z® systems but are focused only on Linux workloads and do not address configurations in which Linux and other operating systems (e.g. z/OS®, TPF) are running on the same IBM Z system. For that reason, IBM® LinuxONE is assumed here as the platform.

These recommendations encompass Linux images deployed directly in logical partitions under PR/SM™ and those running in virtual machines under either the z/VM® or Linux/KVM hypervisor.

This whitepaper is the result of collaboration among IBM developers spanning Linux, KVM, z/VM, PR/SM, performance analysis, architecture, and hardware development.

Distinguishing Values of IBM® LinuxONE

Efficient Virtualization

IBM® LinuxONE system offers benefits of both scale-up and scale-out by using high-performance logical partitioning via Processor Resource/Systems Manager (PR/SM). PR/SM is implemented in firmware and is a standard component in all IBM® LinuxONE systems. PR/SM allocates physical system resources, including processors, memory, and I/O, among isolated logical partitions (LPARs). Depending on the LPAR configuration, those resources can be either dedicated to a partition or shared among the LPARs (memory is always dedicated), and it is possible to add resources to and remove resources from an LPAR while the LPAR is active.

Each logical partition (LPAR) functions as an independent system running its own operating environment. Within an LPAR, it is possible to run the industry-leading virtualization hypervisor z/VM and the open-source hypervisor kernel-based virtual machine (KVM); this allows a single IBM® LinuxONE system to efficiently host of thousands of virtual machines (VMs).

Reliability, Resiliency and Security

IBM® LinuxONE provides the highest levels of availability (near 100 percent uptime with no single point of failure), performance, throughput, and security. IBM® LinuxONE Systems facilitate transparent use of redundant processor execution steps and integrity checking, which is necessary in financial services industries. IBM® LinuxONE servers typically enable hot-swapping of hardware, such as processors and memory. This swapping is typically transparent to the operating system, enabling routine repairs to be performed without shutting down the system.

Predictable Performance

The combination of PR/SM and either z/VM or KVM allows for a wide range of configurations, including ones with a high ratio of virtual processors to physical cores. To achieve the predictable and balanced performance required in an enterprise IT environment, however, the configuration of LPARs and VMs needs to be carefully considered. This includes balancing the virtualization capabilities of PR/SM and z/VM or KVM, properly leveraging PR/SM capabilities such as partition weights, and matching the total workload with the physical resources of the system. Done right, this allows effective consolidation of Linux workloads onto a IBM® LinuxONE system with predictably high performance. The recommendations in this paper are intended to help clients achieve this result.

Definitions

System Topolozgy

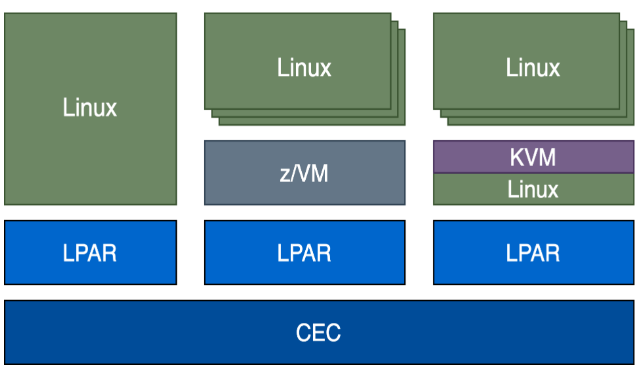
The operating system topology on IBM Z and IBM® LinuxONE can be divided into the following layers:

1. **CEC (Central Electronics Complex) or CPC (Central Processor Complex)**
   This is the physical machine that consists of the CPUs, memory, PCIe cards and everything else physical that is necessary to build a working mainframe.
   This can be called the physical layer, managing and controlling physical devices.

1. **LPAR (Logical Partition)**
   This is a partition created by PR/SM (Processor Resource/ Systems Manager), a partitioning engine that allows the creation of up to 85 partitions on one CEC.
   This can be called the logical layer, providing logical resources to an operating system running in it.

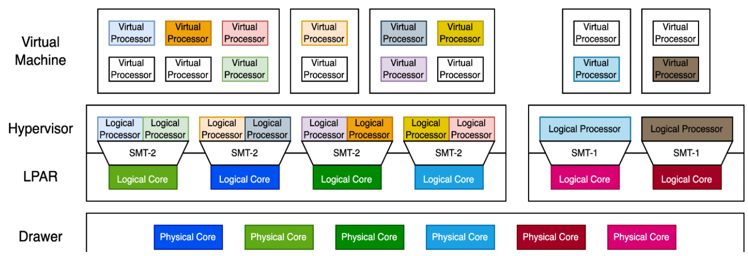
1. **Hypervisor**
   This optional layer allows the creation and management of virtual machines within one partition, sharing processors, memory and I/O.
   This can be called the virtualization layer, providing virtual devices to virtual machines.

1. **Operating System**
   The operating system is run within a virtual machine created by a hypervisor or directly within a logical partition created by PR/SM. Note that while IBM Z supports several operating systems, in the context of this paper it is assumed that some form of Linux is running in each virtual machine, and that either Linux, Linux/KVM, or z/VM is running in each logical partition.



Processor & Hypervisor Topology

The following definitions will be used in this whitepaper for various types of "core" and "processor." Note that these have not always been used consistently in other documentation, including some from IBM. The figure below shows the relationships among these entities, with matching colors illustrating the dispatching (by PR/SM) of logical cores onto physical cores and the dispatching (by the hypervisor) of virtual processors onto logical processors.



Processor Drawer: A physical unit in the CEC which includes some number of physical cores and some amount of physical memory.

Physical Cores: A distinct portion of hardware that executes the z/Architecture® instruction set. Note that each physical core in a IBM® LinuxONE system is designed to support two independent instruction streams ("hardware threads") when operating in SMT-2 mode. Note also that a single "processor chip" contains multiple physical cores, but this does not directly impact the considerations addressed in this whitepaper.

Dedicated Processor/Core: A core that is dedicated to one LPAR.

Shared Processor/Core: A core that can be shared among multiple LPARs.

Logical Cores: A logical partition (LPAR) is equipped with a certain number of logical cores. Based on the LPAR definition, these are either dedicated (physical cores assigned for the exclusive use of this LPAR) or shared (dynamically mapped onto a pool of shared cores). Logical cores are dispatched on physical cores by PR/SM.

Logical Processors: When Simultaneous Multithreading (SMT-2) is active for an LPAR, the operating system or hypervisor running in that LPAR sees each logical core as two logical processors, each of which can execute a z/Architecture program. When SMT-2 is not active, each logical core in the LPAR acts as a single logical processor.

Virtual Machine: Within a hypervisor environment, an arbitrary number of virtual machines (VMs) may be defined, each of which runs an operating system (in this case Linux). Each virtual machine is equipped with some number of virtual processors and some amount of memory. The memory for a virtual machine appears to the operating system running in that virtual machine as "real" memory ("real storage" in z/Architecture terminology) but is in fact a virtual address space mapped into the real memory managed by the hypervisor.

Virtual Processors: A virtual machine (VM) is equipped with virtual processors (vCPUs) that are dispatched on logical processors by the hypervisor.

LPAR Weights & Entitlement: The definition of an LPAR using shared processors includes a relative weight for that LPAR. This determines the compute capacity (processing power) which is guaranteed to be available to that LPAR; this is referred to as entitled processing power. In general, an LPAR's entitlement will correspond to a number of physical cores which may be less than the number of logical cores defined for that LPAR.

When the collective demand of all shared-processor LPARs equals or exceeds the total available processing power, the weights determine the apportionment of that processing power among the LPARs. The entitlement is also used by PR/SM to position the memory and processors used by the various LPARs such that each will be contained within a single processor drawer when the LPARs are running within their entitled processing power.

When an LPAR needs more processing power than its entitlement, it may use additional physical cores (up to its number of logical cores) provided they are not in use by other LPARs. There is no guarantee that this unentitled processing power is provided by physical cores in the same processor drawer as the entitled cores for that LPAR. As an example, an LPAR with 8 logical cores and an entitlement of 4 physical cores might use up to 4 additional (unentitled) cores when other LPARs are not using what they are entitled to.

HiperDispatch & Polarization

Each LPAR is configured by the operating system to be dispatched by PR/SM either in horizontal or vertical polarization. This configuration is controlled by the operating system running in the LPAR, comparable to the usage of SMT.

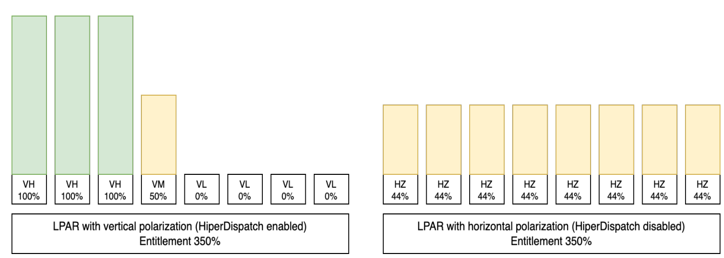
This setting has a significant impact on which physical cores PR/SM selects for dispatching the logical cores and on the time slice and priority a logical core is given. The technology used to implement vertical polarization is known as HiperDispatch [HD].

With Horizontal Polarization (HD=OFF), every logical core of an LPAR has the same priority and entitlement, requiring sharing of physical cores if the entitlement is less than the number of configured logical cores.

With Vertical Polarization (HD=ON), the entitlement is split between logical cores to maximize cache usage and minimize CPU sharing between LPARs. The logical cores for an LPAR are separated into three categories:

- Vertical High(VH) cores have entitlement to 100% of a physical core
- Vertical Medium(VM) cores have some entitlement, but less than 100% of a physical core
- Vertical Low(VL) cores have no entitlement and purely run on unentitled processing power

The example below highlights the difference in core entitlement between a LPAR with 8 cores and a total entitlement of 3.5 physical cores with HiperDispatch enabled/disabled.



HiperDispatch/Vertical Polarization requires cooperation between PR/SM and the operating system running in the LPAR. This is currently supported for z/OS and z/VM. Full support on Linux is provided starting with RHEL 9.6, SLES 15.7, and Ubuntu 24.10.

Utilizing vertical polarization is recommended, especially when the total number of logical cores across all shared-processor LPARs significantly exceeds the number of physical cores in the pool of shared cores in the CEC.

Overcommitment

IBM Z and IBM® LinuxONE supports overcommitment of cores, processors, and many I/O devices at the logical level supported by PR/SM and the firmware or virtual level supported by a hypervisor. The virtual/hypervisor level also supports memory overcommitment.

Physical Core Overcommitment

Physical Core Overcommitment happens when the CEC has fewer physical cores than the sum of all logical cores of all activated LPARs. The general concept is sometimes called LPAR CPU or Processor Overcommitment. Note that the use of SMT-2 creates an additional factor of physical core overcommitment in that the resources of one physical core are shared by both logical processors for that logical core; in this instance the allocation of those shared resources is managed entirely by the processor hardware.

Logical Processor Overcommitment

Logical Processor Overcommitment happens when the Hypervisor has fewer logical processors than the sum of all virtual processors of all running virtual machines.

Total CPU Overcommitment

Total CPU overcommit is the combination of the overcommit of the physical cores and the logical processors. For example, a system with a 1:1.5 physical core overcommitment and a 1:5 logical processor overcommitment would have a total CPU overcommitment of 1:7.5 running SMT-1, it would have a total CPU overcommitment of 1:15 with SMT-2 enabled.

Memory Overcommitment

Memory Overcommitment is provided on the virtualization layer by the hypervisor, z/VM or KVM, by utilizing paging devices to store less recently used memory pages and by paging them in should the virtual machine access the paged-out memory.

At the PR/SM layer, each partition is exclusively assigned a portion of physical memory, according to the partition definition. Thus, there is no memory overcommitment at this layer, no sharing of memory between partitions, and a fixed 1:1 ratio between physical and logical memory.

Steal Time

Steal Time is one of the most important performance metrics to track in any virtualized environment. In simple terms, steal time is the mismatch between CPU time requested by an operating system and the CPU time provided by the environment it is running in. Stated differently, it represents the time that an operating system has a task to run on a processor but is prevented from running that task for reasons outside the control of the operating system. Steal time is calculated by the operating system and expressed as a percentage of time “stolen” of the total system CPU resources. E.g. 20% steal time in a virtual machine with 20 vCPUs would represent the capacity of 4 logical processors unavailable due to outside constraints. This could result in 8 vCPUs being only available 50% of the time or a uniform loss of 20% over all vCPUs.

Note that a high level of Steal Time is a symptom of a performance constraint, not the reason for a performance problem.

In each case, steal time can result from CPU overcommitment in a resource constrained environment, since the total amount of work to be done exceeds the aggregate CPU capacity available during some period of time.

For example, an operating system running directly in an LPAR that is configured to have two VH cores, 1 VM core and 2 VL cores could have a total entitlement of 250% while having access to 5 logical cores or 10 logical processors. If it had 10 threads or processes to run and tried to fully utilize these 10 logical processors, it could observe up to 50% steal time if every other LPAR on the system is currently using its full entitlement.

Steal time can also result when some function must be executed by PR/SM on behalf of a partition or by the hypervisor on behalf of a virtual machine. A common example of this occurs with memory overcommitment at the hypervisor layer: if a virtual machine accesses memory that has been paged out, the hypervisor will take control of that processor and initiate the process to page that memory back in; the virtual machine will be dispatched again when the page is accessible. In this case, the time the hypervisor spends handling the page fault and paging in the memory would be counted as steal time.

A third source of steal time is work performed by PR/SM or the hypervisor that is not attributable to any one partition or virtual machine. This is usually not a significant source of steal time.

General Principles and Best Practices

IBM Z and IBM® LinuxONE systems are designed to provide robust performance across a wide range of workloads, making it impractical to prescribe specific universal configurations. There are however a number of principles which, when thoughtfully applied, will yield strong results and help to improve performance and scalability.

Workload Characteristics

Any virtualized system entails tradeoffs between the performance benefits to specific workloads of providing dedicated resources and the efficiency benefits to the whole system of leveraging resource sharing among workloads. Higher levels of resource (processor, memory, and I/O) sharing may enable more efficient consolidation of workloads onto one system, but at the cost of greater variability in the performance of each workload.

– For workloads that are business critical and sensitive to performance variation, or that require consistent and short response times, overcommitment should be avoided or kept to a very low level.

– For workloads that are more tolerant of fine-grained performance variance, such as most batch processing, higher levels of overcommitment are acceptable.

Another fundamental characteristic of virtualized systems is that they are more efficient overall when the workloads sharing a set of resources vary among themselves in what resources they need when. Thus, combining several workloads which are doing the same things and are driven by the same external demands is likely to yield periodic performance constraints as the workloads all need peak resources (e.g. processing capacity) at the same time. Conversely, combining heterogeneous workloads – with different roles, resource demands, and usage patterns – allows for a smoothing of resource utilization and efficient sharing of those resources.

LPAR Sizing & Configuration

For any virtualization / workload consolidation project, it is important to understand the total compute capacity required for the workloads involved and to ensure that the capacity of the system is sufficient to provide that. Per the “Workload Characteristics” section above, to the extent that resources need to be dedicated to some workloads to support performance requirements, or that homogeneous workloads are virtualized together, the ability to maintain high utilization of the entire system may be constrained, limiting the effective total capacity of the virtualized system. This should not be seen as a one-time exercise; because workloads often change over time, regular capacity monitoring and planning of all resources is highly recommended.

In particular, there must be enough physical cores enabled on the system to support the aggregate compute requirements of the workloads. Overcommitment of processors is allowable, but should in general be kept low, especially for latency-sensitive production workloads. In that case, the difference between logical cores and entitled physical cores should be 3 or less. Since PR/SM does not support overcommitment of memory, there must be enough physical memory available on the system to satisfy the sum of the memory requirements of all LPARs.

The best performance is realized when each LPAR is able to run within one processor drawer, i.e. utilizing physical processors that are in the same drawer where its memory is assigned. PR/SM will automatically arrange the LPARs to achieve this where possible, but it is important that the LPAR definitions allow this. One way to do this is to plan the LPARs in groups such that the total of entitled physical cores for the LPARs in the group sum to no more than the number of IFLs available in each drawer, and that the total memory defined for the LPARs in each group sums to no more than the memory available in each drawer. Where practical, it is generally beneficial to define the LPARs such that the processor and memory requirements are roughly in the same proportion as the physical resources in each drawer.

When supported by the operating system in use (see above under “HiperDispatch and Polarization”), it is highly recommended that vertical polarization be enabled for all partitions, as this enables more effective management of system resources by PR/SM.

Hypervisor Configuration

Software hypervisors are more flexible than PR/SM in the redistribution of resources. Nevertheless, it is important to define Virtual Machines appropriately for their respective workloads. The key dimensions to consider are:

– The number of virtual processors, which sets the maximum consumable compute capacity for the VM

– The entitled compute capacity, determined by the share of the VM relative to other VMs in that LPAR, which is the minimum compute power provided to the Virtual Machine

– The amount of memory defined for the VM

It is critical that these configuration parameters be set to match the real requirements of the workload and not merely copied from the values used on other platforms when consolidating workloads onto IBM® LinuxONE. Over-sizing VMs may result in inefficient operation at all levels of the system and may lead to significant performance problems.

For processors, ensure that workload peaks can be handled by the number of virtual processors defined, and that the entitled processing capacity is at least sufficient for the average (steady-state) work. In no case should any VM be defined as having more virtual processors than there are logical processors defined for the containing LPAR. The entitled compute capacity for the LPAR should be bigger than the sum of the required compute power of (i.e. the total average / steady-state work) of the VMs in that LPAR. To maximize the available compute capacity, SMT should be enabled by the hypervisor. If the workload in the VMs is highly correlated, capacity planning should be based on maximum consumption and not average consumption.

Memory for a VM should be sized to hold the working set of the workload. The sum of the memory defined for the VMs in an LPAR should not exceed 130% of the physical memory defined for that LPAR (1.3x memory over-commit); for larger LPARs, a lower level of memory overcommitment is recommended. Ensure that the hypervisor has sufficient paging space available to contain at least 115% of the over-committed memory (sum of memory for all VMs minus the physical memory for the LPAR), more might be required when certain configuration options are used to increase paging performance. E.g. a z/VM LPAR with 1TB of assigned physical memory and a sum of 1.3 TB of memory defined for various virtual machines should have at least 345GB of storage dedicated to paging. This configuration is operating at the high end of the overcommit recommendations(1.3TB/1TB=1.3) and at the low end of allocated paging space; both the paging space and the physical memory should be increased before further workload is added to this LPAR.

If sufficient physical memory is available such that memory over-commitment is not required (an ideal situation, especially for latency-sensitive workload) then use of host large pages should be enabled where supported by the operating system / hypervisor to gain additional performance benefits.

Even when the configuration has been carefully researched and defined, it is vital to monitor the performance of the system in production as even small changes in workload can have wide-ranging effects. If a VM experiences 100% CPU utilization for significant periods, the virtual processors for that VM may need to be increased. If a VM experiences significant steal time, its entitlement (weight) or the available compute power of the LPAR may need to be increased. In either case, the need for change is determined by whether the system is meeting required levels of performance.

If changes in workload drive changes in VM configurations such that the VMs in an LPAR no longer fit within the capacity of that LPAR, it is in general better to move one or more VMs to other LPARs (where there is available capacity) than to change the size of the LPAR.

Periodic review of the environment, including workload requirements and configuration, is highly recommended.

Situations to Avoid

The guidelines given here are intended to be generally applicable to a broad scope of client environments. There may however be situations where specific workloads or business requirements dictate configurations which do not quite match the best practices described above. IBM® LinuxONE is designed to provide robust performance, reliability, and availability across a wide range of configurations and workloads, and these guidelines should not be taken as absolute. Wherever the desired configuration strays far from these guidelines, clients should consult with IBM (see Additional Resources).

That said, there are certain configuration situations that should be avoided in all cases, as these are likely to lead to high levels of resource contention and/or very inefficient system operation. These include:

- LPAR configuration that does not allow for LPARs to be placed without one or more LPARs spanning a drawer boundary

- Defining more virtual processors in a single virtual machine than there are logical processors in the LPAR hosting that virtual machine

The following situations should also be strongly avoided for production workloads, as they will negatively impact latency and consistent performance.

- Physical core to logical core overcommitment at a ratio greater than 2x

- Entitled processing power of an LPAR significantly less than its defined logical cores

- Memory overcommitment within an LPAR at a ratio greater than 1.3x

Additional Resources

This whitepaper is intended to provide high-level in the deployment of large-scale Linux workloads on IBM® LinuxONE systems. More detailed information on the topics addressed here and on related matters can be found in the resources listed below.

- IBM® LinuxONE product documentation:
  https://www.ibm.com/docs/en/systems-hardware/linuxone

- PR/SM Planning Guide:
  https://www.ibm.com/docs/en/module_1721331501652/pdf/SB10-7184-00.pdf

- IBM Whitepaper: Understanding and Tuning z/VM Paging
  https://www.vm.ibm.com/perf/tips/prgpage.html

- IBM LinuxONE Redbook Collection:
  https://www.redbooks.ibm.com/domains/linuxone

- IBM Redbook: Practical Migration from x86 to IBM® LinuxONE
  https://www.redbooks.ibm.com/abstracts/sg248377.html

- IBM Redbook: Leveraging IBM® LinuxONE to Maximize Your Data Serving Capabilities
  https://www.redbooks.ibm.com/abstracts/sg248518.html