Paper 320-2010

Virtualization in a UNIX Environment and SAS®

Leigh Ihnen, SAS Institute Inc., Cary, NC

ABSTRACT

As organizations seek to efficiently utilize resources and reduce total cost of ownership of their SAS environments, virtualization of the environment is one possible approach. Depending upon the operation system (AIX, Solaris, or HP-UX) and OS release there are a wide range of virtualization options available for the SAS system architect to exploit. The intent of this paper is to provide the SAS system architect with a working knowledge of pros and cons of running SAS in a virtualized environment. The goal is to facilitate efficient communication with IT system architects and administrators. Logical partitions (LPARs), micro-partitions, domains, zones, nPars, and vPars as well as virtualization of input/output (I/O) and resource management in a SAS environment will be discussed in relation to architecting a SAS environment.

Why Virtualization?

A virtualized environment allows resources such as central processing unit (CPU), Memory, I/O, and Network from a single host system to be partitioned into multiple environments. With multi-tiered architectures and increasing core count per system, configuring tiers with low resource utilization on their own host system would cause over provisioning of CPU resources. In addition, use of virtualization may allow dynamic sharing or movement of resources between operating system OS instances within a host system and provide flexibility when adding additional resources to tiers, in an architecture, which cannot be distributed across multiple host system such as the SAS® Metadata Server.

SAS 9.2 Tiered Architecture

With SAS 9.2, the EBI platform and other solutions are deployed based on a logical architecture featuring three or four tiers. Some of the tiers can be implemented in a distributed environment such as the compute tier if a clustered file system is used to present a global storage. Other tiers, such as the metadata tier, must be implemented within a single OS instance.

These are the SAS 9.2 architecture logical tiers:

- SAS Metadata Server
- Middle-tier Server
- Compute Server
- Data/Transaction Server

Depending upon the size of the deployment, the following may be a typical suggested core allocation by tier.

Tier	Cores for Small Configuration	Cores for Medium Configuration
Metadata	**1**	**2**
Middle-tier	**1**	**2**
Compute	**6**	**12**

To guarantee solution responsiveness, the best practice is to implement the tiers in separate OS instances with minimum guaranteed resources. In the above examples, all of the tiers could be accommodated on an 8 or 16 core system using virtualization.

Consider implementing the above configurations on individual host systems based on dual cores and 1, 2, or 4 sockets architectures. The two configurations demonstrate different issues. The small configuration is initially over provisioned, and the medium configuration will suffer over provisioning when additional capacity is needed. For the small configuration, an extra 4 cores would be required, which would over provision CPU resources by 50%. Even more problematic is that 50% of the excess capacity is likely to be underutilized in the metadata and middle-tier host systems and unavailable to support increased CPU requirements of the compute tier.

Tier	Cores for Small Configuration	Cores for Host Systems
Metadata	1	2
Middle-tier	1	2
Compute	6	8

In the medium configuration, the issue occurs when additional capacity was needed on all three tiers. The provision of the metadata and middle-tier server is a 2X increase, and the computational tier a 1.25X increase.

Tier	Cores for Medium Configuration	Cores for Host Systems for Additional Capacity
Metadata	2	4
Middle-tier	2	4
Compute	12	16

In the future, as the core count per socket will continue to increase, the issues of over provision in systems that do not employee some form of virtualization will be exacerbated.

Just as virtualization can be used to provide CPU resources at a granularity appropriate for a tiered architecture, it may provide for creation of multiple instances of the compute tier, which can provide isolation of resources to meet service level agreements (SLAs) across different workloads, business units, or cost centers.

Can Virtualized Resources Be Shared between Environments?

All three major UNIX systems allow for partitioning of CPUs, memory, storage, and network into disjoint resource sets that can host an OS instance. In addition, depending upon the technology, CPUs, storage, and network may be virtualized and shared between multiple environments. The degree of virtualization and resource sharing varies between the three UNIX vendors and a summary of the various capabilities, architecture, and performance will be presented later in this paper.

The technical approaches by the three vendors vary widely but roughly provide the ability to share CPUs, storage, and network resources between multiple OS environments with each OS environment having its own identity. Depending upon the degree of virtualization and sharing of resources, there may be performance trade-offs that impact the SAS system performance.

In addition, depending upon the virtualization technologies used, it may be possible to move CPU and/or memory resources dynamically between OS instances without taking an outage. This is especially useful when business cycles allow stacking of applications in the same host system with non overlapping peak usage periods.

The ability to partition a host system into multiple virtual environments is dependent upon the virtualization approach and the system resources available, such as the following:

- Boot disk
- Paging space
- I/O adaptor
- Network adaptor
- System architecture – cell or board based resources

Covering all possible requirements and partitioning options is beyond the scope of this paper and should be undertaken as a planning exercise with your IT system architect and vendor. In addition, partitioning the resources needed to preserve the necessary system performance to meet SLAs may also constrain the number of virtual environments that are suitable for deploying SAS workloads.

Virtualization in an HP-UX Environment

HP-UX provides several technologies that can be used to create virtual environments and manage resource allocations:

- nPars – Hard OS partition based on hardware isolation with dedicated memory, CPUs, I/O, and network based on cells within the system. Cells based resources cannot be split across nPars.
- vPars – OS partition based on software isolation with dedicated memory, CPUs, I/O, and network based on resources within an nPar. The smallest CPU resource allocation unit in a vPar is a single CPU.
- HP Integrity Virtual Machines (VMs) – virtual machines with shared CPUs and I/O and software isolation.
- HP Secure Resource Partitions – partitions within a single OS image with guaranteed compute resources and security Isolation.
- Workload Manager – provides resource management based upon policies.

vPars and SAS

In the examples of small and medium EBI deployment, vPars would be the appropriate technology to use. In addition, vPars allow the following advantages:

- dynamic CPU migration across vPars
- dynamic memory migration across vPars for HP-UX 11i v3
- mixed HP-UX 11i v2 & v3
- vPars in same nPar

When used in conjunction with the Workload Manager (WLM), product resources can be moved between vPars based on utilization and policies allowing for dynamically adapting the system to match SLAs.

HP Integrity VMs and SAS

HP Integrity VMs allow for sub CPU allocations, and may be appropriate for environments with less demanding performance requirements. If the environment's performance is heavily dependent upon I/O, then the use of HP's VM Accelerated Virtual I/O may need to be considered. HP VMs may be suitable for hosting multiple SAS functional testing environments within a single partition.

For more information about HP virtualization technology and a SAS EBI implementation using HP technology, see the following documentation:

http://docs.hp.com/en/14241/VSEStartUpdate18July2008.pdf

http://docs.hp.com/en/vse

http://www.sas.com/partners/directory/hp/virtualbi.pdf

Virtualization in a Solaris 10 SPARC Environment

Partitioning

Large Sun enterprise SPARC based systems can be partitioned in multiple domains (separate OS instances) each with dedicated CPU, Memory, I/O, and Network resources. The granularity of the resource partitioning is dependent upon the specific model and configuration. In addition, there is the capability to dynamically migrate resources between domains based on SUN's dynamic domain reconfiguration capabilities. Each domain may contain multiple multi-core sockets, which may then be future virtualized using zones and resource pools.

Zones

Zones provide the means to create one or more virtual environments within a single Solaris instance. Each zone has its own namespace for users, file systems, network ports, and naming services providing what appears to the application its own instance of Solaris. Zones provide the following qualities:

- security
- isolation
- virtualization
- additional granularity of resource allocation
- no dedicated resources necessary
- capability for CPU shares when more than one zone is assigned to a resource pool
- can only be assigned to a single resource pool

Zones can have the following memory requirements associated with them:

- physical memory
- maximum swap space
- locked memory
- shared memory

Resource Pools

Resource pools provide a framework for managing processor sets and thread scheduling classes. Using resource pools enables you to separate workloads so that workload consumption of certain resources does not overlap. The resource pool framework also allows the definition of dynamic resource pools with a maximum and minimum CPU count requirement. Depending upon defined objectives, CPU resources may dynamically vary in the resource pool. It is also possible to dedicate resources in a resource pool.

Solaris Virtualization and SAS

Using zones and resource pools allows flexible configuration of CPU resources to support multiple SLAs in a mix workload environment. In the example of a SAS EBI deployment, each tier would be deployed in their own zone. For the metadata tier, it would be recommended to use a resource pool with dedicated CPUs and a zone with defined memory requirements.

If the environment has a major off hours batch component and heavy middle-tier usage during the workday period, using a single resource pool to support the middle-tier zone and the compute tier zone would allow idle middle-tier CPU resource to be used during the off hours batch processing on the compute tier. Similarly, if the off hours batch processing requires more CPU resources for the computation tier than during the workday period, CPU resource can be utilized by the middle-tier server if necessary during the workday period.

Depending upon data sharing needs, it may be possible to separate compute tier workloads onto different zones with a shared resource pool. In this case, the fair share scheduler can be used with shares that guarantee minimum CPU resources to each workload but allow idle cycles in one zone to be used in another zone. If the workloads on the two zones have different SLAs, then the shares assigned to each zone can be adjusted to reflex the SLAs. For more information, see http://www.sas.com/partners/directory/sun/SASinVirtSunEnvironment.pdf.

Virtualization in an AIX 5.3 and Later Environment

Beginning with the Power5 chip architecture, CPU resources have been virtualized in AIX. Process threads are scheduled on a physical CPU by a hypervisor that dispatches threads scheduled on a virtual CPU by the OS. This virtualization of CPU resources allows multiple AIX instances to share CPU resources. When partitioning a Power5 based system, CPUs can be dedicated to an LPAR that hosts an AIX instance, or they can be placed in the micro-partition processor pool and shared with multiple micro-partitions, each running an instance of AIX. In addition, AIX allows resources to be dynamically moved between LPARs. When creating LPARs or micro-partition definition, minimum and maximum resource definitions can be set to allow for dynamically adding or removing resources without rebooting. An LPAR definition has both virtual and physical CPU definitions. For fixed LPARS, each virtual CPU will be a provision on a physical CPU. In a micro-partition, a virtual CPU can be provisioned with as little as .1 of a physical CPU. The minimum CPU entitlement of a micro-partition is a guaranteed amount of CPU resources. If the micro-partition is uncapped, then additional CPUs may be used until the maximum of the online virtual CPUs or maximum capacity entitlement is reached. With an appropriate virtual CPU definition, a micro-partition could use all available spare cycles.

In addition to virtualizing CPU resources, AIX 5.3 provides the ability to virtualize storage and network I/O. This allows devices to be shared between multiple partitions. If a partition does not need dedicated I/O, then using VIO servers will allow for a large number of micro-partitions to be defined and supported on a host system. Limited testing shows minimal overhead associated with using a VIO server. However, the use of a VIO server to deploy multiple SAS environments that are I/O intensive can lead to the workload in one micro-partition affecting the performance of another micro-partition, since I/O is shared. In addition to virtualization CPUs and I/O, AIX 5.3 supports software resource management via WLM. WLM supports placing processes into classes and allocation of processor resources between the classes. This provides another method of enforcing business priorities on your SAS workload.

AIX 6 provides the ability to create Workload partitions (WPARs). WPARs are a virtualized environment using a VIO server for I/O and WLM to manage CPU resource allocation within a partition. For an example of deploying SAS in a WPAR environment, see http://www.sas.com/partners/directory/ibm/papers.html.

In the example of an EBI deployment, micro-partitions could be used to provide the guaranteed need CPUs (minimum entitlement) and additional virtual CPUs could be defined so that idle cycles in one tier could be used in another. This approach could also be used to provide for additional capacity via dynamic LPARs if the frame has additional resources.

Virtualization and SAS® Grid Computing

SAS Grid Manager provides another method of managing workloads. When combined with virtualization techniques and a global file system, the capabilities for building a flexible SAS environment are available. Each host system could be divided into two virtualized environments. One virtual environment would support the primary workload for the host system. The second environment with a much smaller guaranteed resource set would provide the platform for using idle CPU cycles, when available, via the previously discussed dynamic resource management mechanisms. SAS Grid Manager would be responsible for not over scheduling the secondary environment to ensure reasonable performance.

CONCLUSION

This paper has presented a high-level overview of virtualization techniques for AIX, HP-UX, and Solaris on SPARC. In addition, potential performance issues have been identified and strategies have been given for deploying multi-tiered SAS solutions on virtualized systems. Finally, one potential use of SAS Grid Manager and virtualization has been described that allows for architecting a flexible environment, which is intended to maximize use of available CPU resources.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author:

Leigh Ihnen
SAS Campus Drive
SAS Institute Inc.
Cary, NC, 27511
E-mail: Leigh.ihnen@sas.com