Guidelines for Configuring Virtual Partitions on Cellular Platforms

A Technical White Paper

Abstract

The HP-UX virtual partitions (vPars) solution is one of the popular choices available to customers who are considering workload consolidation on nPartitions (hard partitions on cellular HP platforms). The vPars solution provides users with configuration flexibility and negligible performance overhead. It is however important from a performance perspective to understand and account for the localities of the resources assigned to virtual partitions hosted on cellular platforms. The paper provides some guidelines for configuring virtual partitions to achieve good workload performance on cellular platforms where an nPartition's resources span multiple cells. This paper also illustrates the influence of resource localities on workload performance by using results from a focused study.

This paper assumes that the reader is familiar with the concepts of vPars and has a good knowledge of the HP-UX operating system and its NUMA features (HP-UX support for the non-uniform memory architecture on cellular platforms [2]).

Introduction

vPars allows a system administrator to create multiple soft partitions within an nPartition. Each virtual partition runs its own instance of an HP-UX Operating Environment which can be independently tuned. Each virtual partition is assigned a set of physical resources to allow the HP-UX Operating Environment to boot and be operational. The vPars command line interfaces offer a wide range of choices for assigning resources to the individual virtual partitions. A thin trusted layer of software called the vPars Monitor resides between the firmware layer and the HP-UX Operating Environments. The vPars Monitor controls the actual assignment of resources to the individual virtual partitions based on a user specified partition plan (vPars database). The vPars Monitor, along with the virtual-partition-aware firmware on Integrity platforms, provides platform emulation services for the individual virtual partitions hosted on the nPartition.

Customers who use various solutions for consolidating servers in their datacenters also have expectations about the degree of performance impact these solutions may have on the individual workloads. One of the key advantages of the vPars solution is that it allows direct, non-virtualized access to the assigned subset of the nPartition's physical resources. The performance overhead due to the vPars software stack is therefore negligible. In fact, vPars can help improve overall resource utilization.

On cellular platforms, memory access latencies vary depending on the type of access: local access, remote access, or access to interleaved memory. This variation has an impact on the performance of the workloads. The vPars software offers a high degree of flexibility for configuring virtual partitions. Virtual partition resource assignments done without attention to locality considerations can in some cases have a negative influence on the workload performance.

Configuration Flexibility Versus Resource Locality

Any valid virtual partition needs a set of resources assigned to it to enable it to host an HP-UX Operating Environment and workload(s). The following is a brief overview of the types of choices available for picking resources for assignments to virtual partitions or setting attributes at the nPartition and the virtual partition level.

The granularity of processor assignment to a virtual partition is a CPU core. CPU cores can be specified for assignment using one or more of the following methods:

- a count of CPU cores or
- a count of CPU cores from a given cell (CLPs: cell local processors) or
- Hardware path of the CPU.

CPU cores can also be dynamically migrated in or out of a virtual partition OS instance. For more information on guidelines for CPU assignments please refer to [3].

The unit of memory assignment to a virtual partition is a granule, the size of which can be specified as an attribute when the first virtual partition in an nPartition is created [5]. Memory, either cell local memory (CLM) from a specific cell and/or interleaved memory (ILM) can be assigned to a virtual partition by specifying either:

- an amount of memory (in MB) and/or
- an explicit memory address and the size of the range specification

With HP-UX 11i v3 and beginning with the vPars A.05.01 release, the vPars software supports dynamic addition of memory granules to an active virtual partition as either base or floating memory and deletion of memory granules that were marked as floating from an active virtual partition. Attributes like base and floating memory can be specified at the individual virtual partition level to help the HP-UX kernel decide how much of the assigned CLM or the ILM could be treated as ejectable and non-ejectable from the virtual partition. For more information on guidelines for memory assignments and memory migrations please refer to [4].

The minimum granularity of I/O assignment is a Local Bus Adapter (LBA) which translates to a single I/O slot on the cellular platforms. I/O assignments are specified by their hardware paths. Any available I/O slot (or LBA) in an nPartition can be assigned to any virtual partition.

Given the wide range of options available for specifying resource assignments to virtual partitions and the varying needs of each workload, a typical user is often faced with a difficult task of planning a vPars environment that has:

(a) Enough resources to satisfy the needs of the workloads hosted in individual virtual partitions, and

(b) A resource layout that ensures good workload performance.

To address the (a), tools like HP Integrity Essentials Capacity Advisor can be used to gather resource usage profile data for the individual workloads in a non-consolidated environment and to analyze them to help predict the amount of resources needed for each workload on a target consolidated environment.

To address (b), the user should consider the impact of memory access latencies in a NUMA based cellular platform which vary [9] depending on whether the accesses are:

- local to a cell (CLM: cell local memory)
- remote between cells connected to the same crossbar (1 hop)
- remote between cells connected to different crossbars (2 hops)
- to interleaved memory (ILM)

In large cellular platforms (that is, with more than four cells), as the size of the nPartition increases there is a higher chance that the cells assigned to the nPartition are connected to different crossbars. These variations in memory access latencies do impact the workload performance.

Virtual partition configurations that have resource localities may have lower workload performance. The extent of performance impact depends on the size of the nPartition and also the configuration of the virtual partition and the nature of the workload.

Workload Performance Study

In order to help illustrate the influence of resource localities on workload performance the following study was done. The scope of this study is focused to provide basic insights. Hence, customers (or users) should not treat results as comprehensive or applicable to all workloads and configurations. Two sample workloads that are considered representative of typical enterprise customer applications—namely: SAP-SD 2-tier and SPEC-jbb—were chosen to be run in a vPars environment hosted on a small hardware configuration.

Overview of Workloads

SAP SD 2-tier

SAP SD (Sales and Distribution) is a standard application benchmark developed by SAP and is one of the most published and visible SAP benchmarks [6]. It measures the system performance for processing sales and distribution transactions. A transaction consists of a full business process that captures activities from the full order processing cycle -- from order creation to delivery to invoicing.

There are two configurations for the SAP SD benchmark: 2-tier and 3-tier. In a SAP SD 2-tier environment, both the database software and the SAP application run on the same system. Thus, it provides a way to demonstrate the ability to handle both the application and the database software with a single system. In a 3-tier environment, the database and SAP application run on separate systems.

The benchmark produces two metrics, the maximum number of SD users a system can handle (while keeping average response time below two seconds), and SAPS. The SAPS (SAP Application Performance Standard) is a metric used to measure the throughput of a SAP application server. For the SAP SD benchmark, 100 SAPS equal 2000 fully processed order line items per hour. A higher SAPS rating indicates better system performance and thus an ability to accommodate more users.

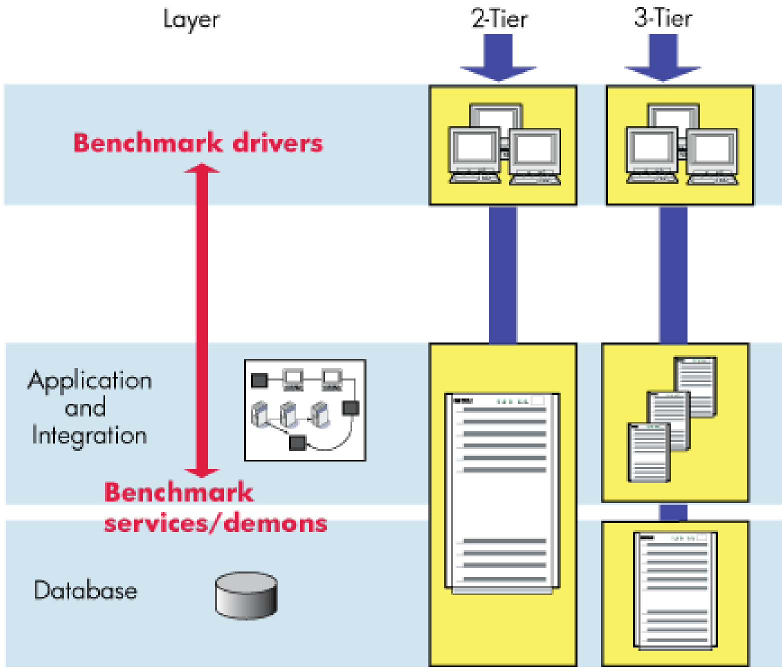


Figure 1 SAP SD Benchmark Two-Tier Setup

For this experiment, rather than measuring the maximum users for a particular configuration, we choose to keep the number of users constant for all configurations while measuring the SAPS. The faster the transactions complete, the higher the SAPS are. So, for the same number of users the SAPS number is higher for an optimal configuration than for less optimal configurations. We choose the number of users for this experiment in such a way that the system is under-utilized while in a similarly sized nPartition configuration[1]. This leaves enough room for less optimal vPars configurations. We use SAPS as a comparison measurement.

SPECjbb

SPECjbb is a standard benchmark developed by SPEC [7]. It evaluates the performance of servers running typical Java-based business applications.

SPECjbb models a wholesale supplier with a variable number of warehouses and represents an order processing application for the supplier. It emulates a three-tier client/server system, with middle-tier business logic connecting front-end thin clients to a back-end data store. The emphasis of the measurement is on the middle tier, the business logic, and object manipulation. The benchmark combines all three tiers into a single Java application for portability and ease of use.

[1] The study in this paper compares results from a single cell nPartition configuration with results form different vPars configurations all having the same amount of resources as a single cell nPartition

During a SPECjbb run, the benchmark is run repeatedly with an increasing number of warehouses starting with a specified number and ending with twice that number. The average throughput of all the runs is used as the performance measure. The unit of measure, BOPS, stands for business operations per second.

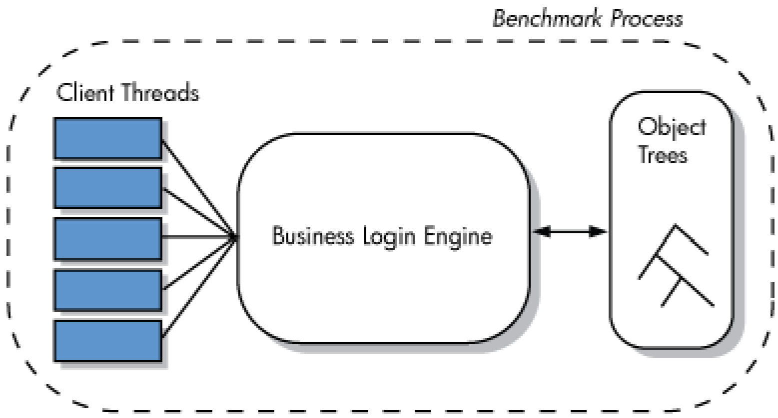


Figure 2 SPECjbb Benchmark Setup

For this focused study, the benchmark was configured with 16 warehouses. Given a set of resources a virtual partition configuration having the best resource locality should result in highest number of BOPS.

Hardware and Software Environment

An HP Integrity rx7640 Server with two cells was used for this experiment. Each cell was populated with 32 GB memory and four dual-core Intel Itanium 2 CPUs (8 processing cores). The rx7640 is a single crossbar system. To enable for workload performance comparisons with a single cell nPartition configuration this system was configured to be capable of hosting either:

- Two virtual partitions (HP-UX 11i v3 with vPars A.05.01) in a two-cell nPartition or
- Two single-cell nPartitions each capable of hosting one virtual partition.

The I/O associated with each workload was attached to each cell: the I/O for SAP SD 2-tier was connected to cell 0, and the I/O for SPECjbb was connected to cell 1.

Design of Experiments

The main objective behind this study was to empirically measure the performance impacts on the above-mentioned sample workloads due to:

a) The vPars software stack.
b) The different resource layouts in a virtual partition.
c) The influence of workloads running simultaneously in co-hosted virtual partitions.

It is worth pointing out that the objective of this focused study was not to determine the upper bounds on the performance impacts for all classes of workloads in all possible virtual partition and nPartition configurations.

We ran these two workloads and gathered metrics on different virtual partition configurations each with the same amount of resources (one cell's worth of: processors, memory and associated I/O for the workload being run).

For (b) and (c) some artificially non-optimal configurations were intentionally chosen to illustrate the influence of less than optimal resource localities on workload performance. With those non-optimal configurations, negative performance impact on the workload was, as expected, higher than normal.

While it may not be very obvious from the results of this focused study, it should be emphasized that the performance penalties for non-optimal configurations could be even higher on larger hardware configurations with member cells connected to different crossbars

The details of the experiments that follow in this paper provide the guidance to make sound configuration choices that avoid non-optimal configurations and achieve good performance results.

Details of Resource Layouts

The following section details the virtual partition configurations used for running the SAP SD 2-tier and SPECjbb benchmark workloads.

Layout 1: Baseline nPartition and Virtual Partition Runs

We started off the test runs by first collecting baseline data (to be used for comparisons later on) by running the two benchmark workloads one after another on a single-cell nPartition (configuration 1A). We then configured a single virtual partition on this single-cell nPartition and assigned all the physical resources on the cell to this virtual partition (configuration 1B). Each of these workloads was run in this virtual partition and results were used to compare with the nPartition run to study the overheads introduced due to the vPars software stack.

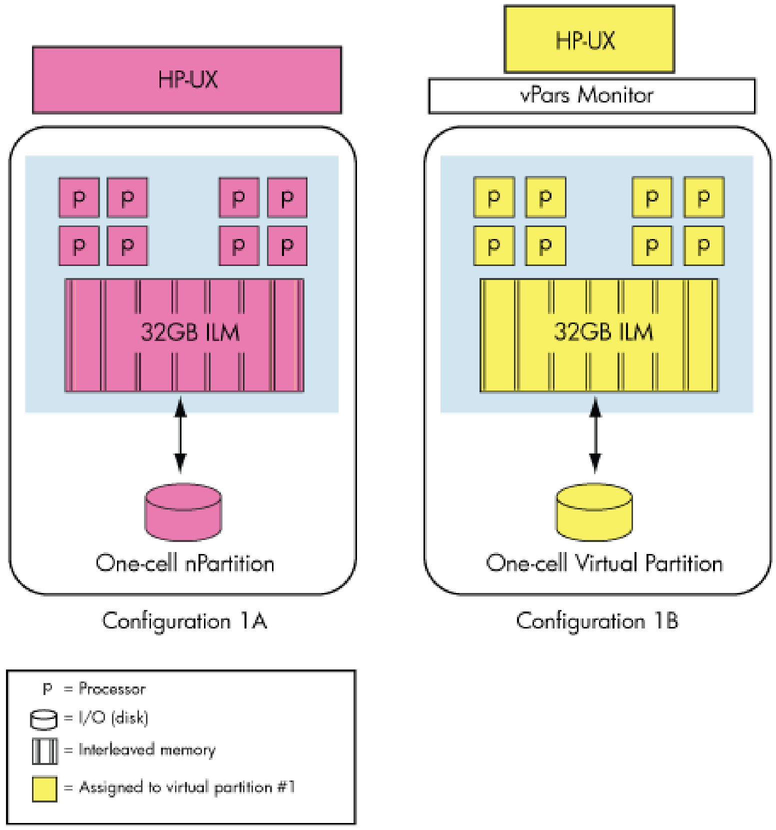


Figure 3 Baseline nPartition and Virtual Partition Configuration

- 1A (nPartition): One-cell nPartition (8 cores, 32GB ILM)
- 1B (virtual partition): One virtual partition/one-cell nPartition (8 cores, 32GB ILM[2], 100% Base memory, 1024MB granule size)

We also did additional runs to study the impact of the memory granule size and the impact of base versus floating memory:

- 100% Base with128MB granule size
- 100% Base with 4096MB granule size
- 1024MB granule size with 25% Base and 75% Float
- 1024MB granule size with 50% Base and 50% Float

[2] Some amount of memory in the nPartition will be used up by the vPars monitor and will not be available to the hosted OS.

Layout 2: Single Virtual Partition in a Two-Cell nPartition with Different Resource Layouts (ILM Only)

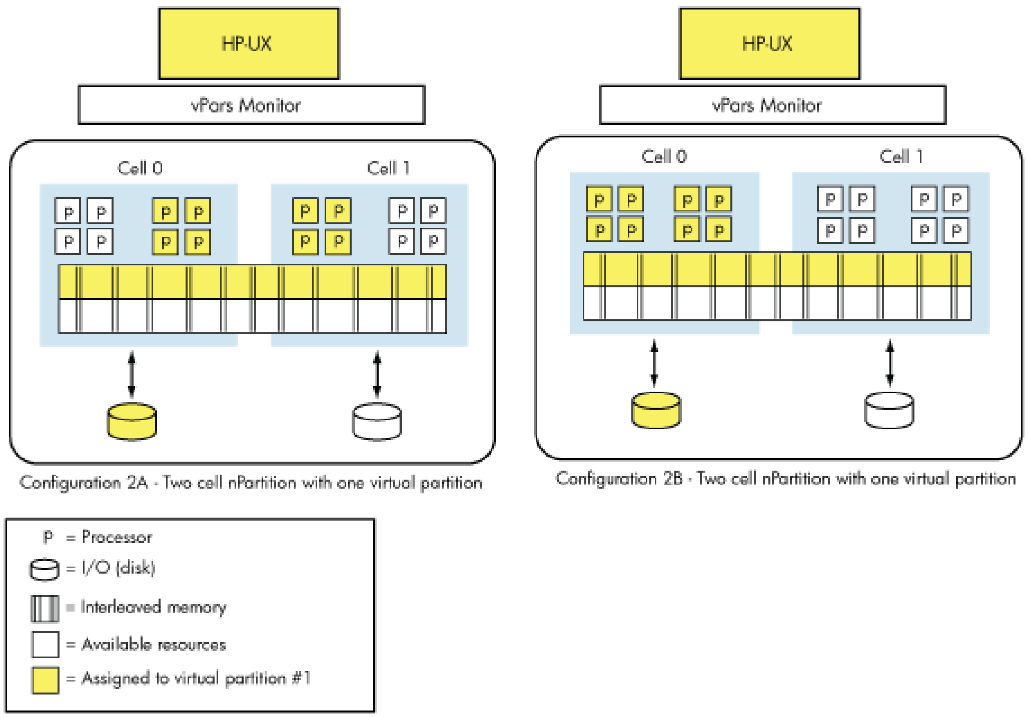


Figure 4 Single Virtual Partition Configurations 2A and 2B

- 2A – One virtual partition / two-cell nPartition (4 cores from each cell , 32GB ILM, 100% Base memory, 1024MB granule size)
- 2B – One virtual partition / two-cell nPartition (8 cores from a cell that has the workload's I/O, 32GB ILM, 100% Base memory, 1024MB granule size)

Layout 3: Single Virtual Partition in a Two-Cell nPartition with Different Resource Layouts (CLM and ILM)

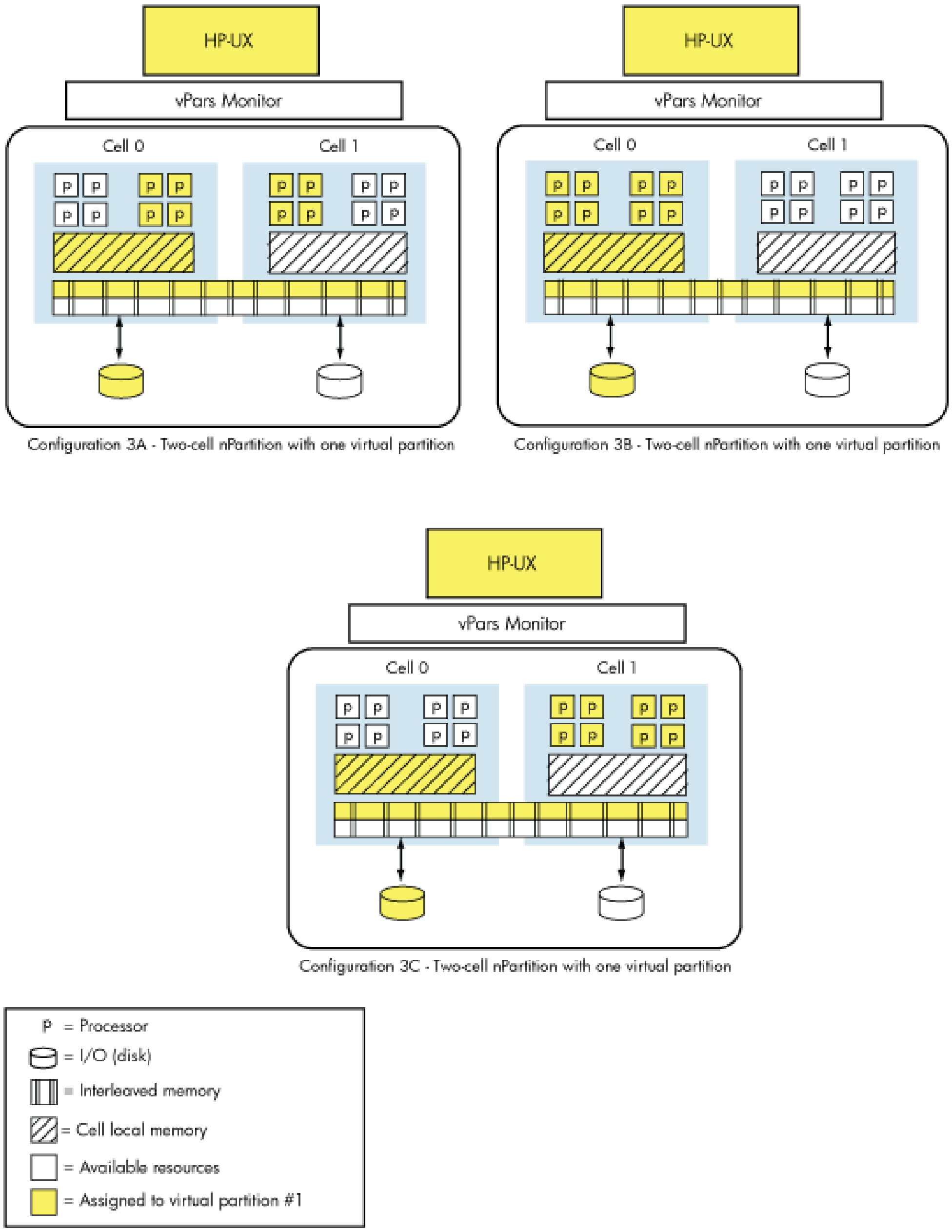


Figure 5 Single Virtual Partition Configurations 3A, 3B and 3C

- 3A – One virtual partition / two-cell nPartition (4 cores from each cell, 24GB CLM (on the cell with workload's I/O), 8GB ILM, 100% Base memory, 1024MB granule size)

- 3B – One virtual partition / two-cell nPartition (8 cores and 24GB CLM (on the cell with workload's I/O) , 8GB ILM, 100% Base memory, 1024MB granule size)

- 3C – One virtual partition / two-cell nPartition (8 cores (on the cell without workload's I/O), 24GB CLM (on the cell with workload's I/O) , 8GB ILM, 100% Base memory, 1024MB granule size)

Layout 4: Two Virtual Partitions with Workloads Running Simultaneously

Users typically configure multiple virtual partitions, each hosting its own workload, in an nPartition. Also, the resources assigned to the various virtual partitions in the nPartition can come from different cells in the nPartition. As a next step we configured two virtual partitions on this two-cell nPartition. With different virtual partition layouts for each of these virtual partitions, we ran the workloads in each of the virtual partitions simultaneously.

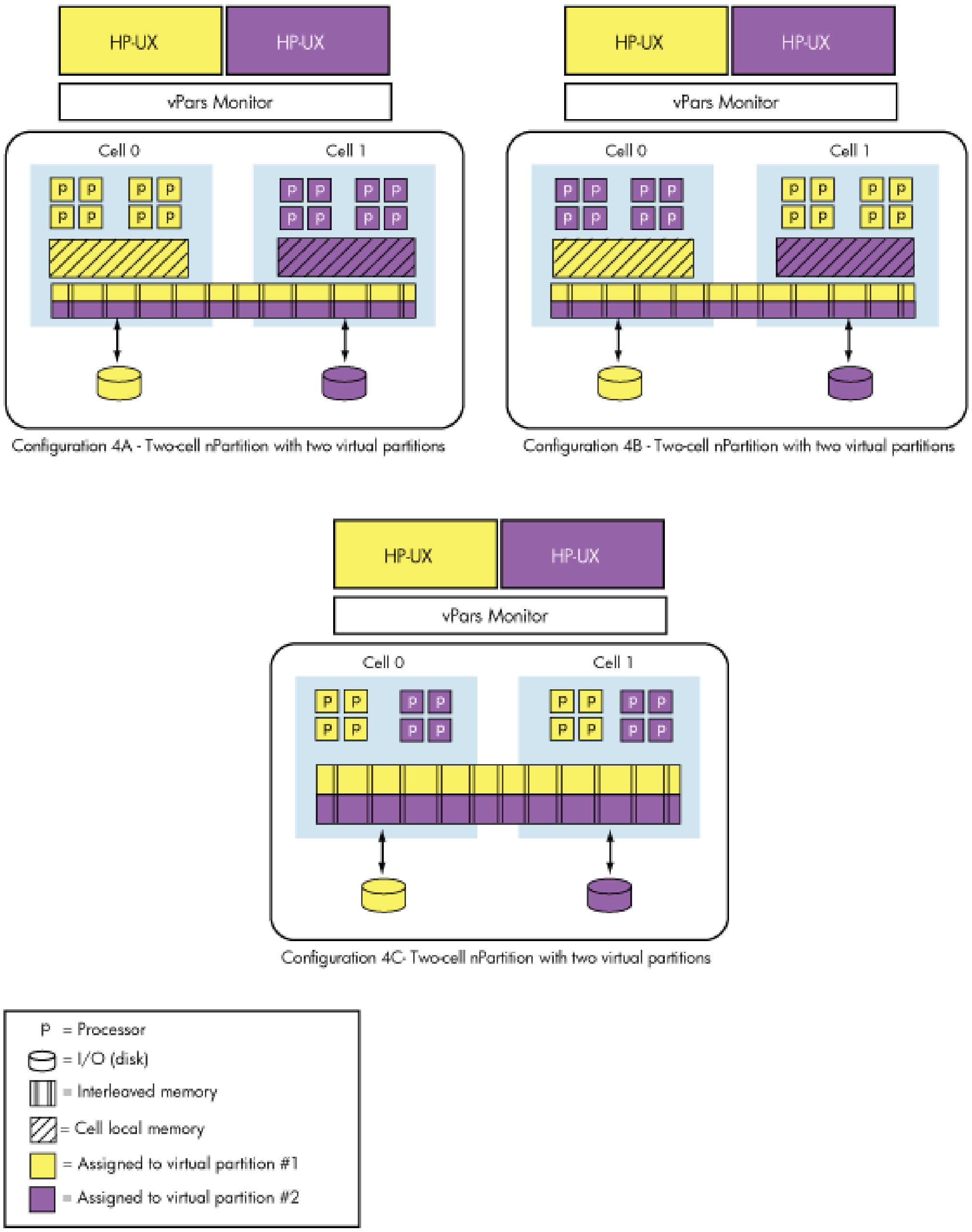


Figure 6 Virtual Partition Configurations 4A, 4B and 4C

- 4A – Two virtual partitions / two-cell nPartition (each virtual partition has 8 cores and 24GB CLM (on the cell with workload's I/O), 8GB ILM, 100% Base memory, 1024MB granule size)

- 4B – Two virtual partitions / two-cell nPartition (each virtual partition has 24GB CLM (on the cell with workload's I/O), 8 cores from remote cell, 8GB ILM, 100% Base memory, 1024MB granule size)

- 4C – Two virtual partitions / two-cell nPartition (each virtual partition has: 4 cores/cell , 32GB ILM, 100% Base memory, 1024MB granule size)

Analysis of Results

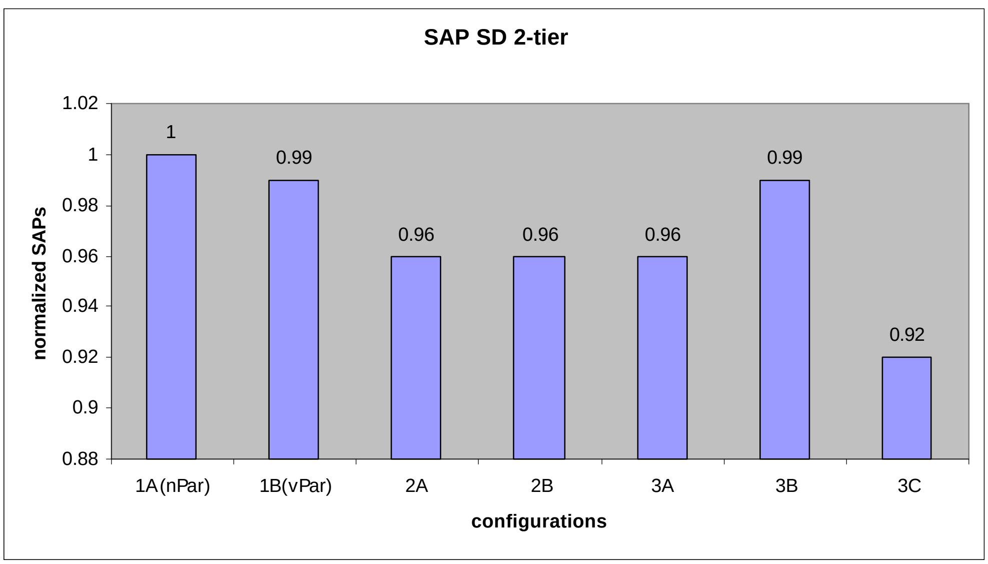
The configuration 1A (one-cell nPartition) is used as a baseline. The results from this baseline run for the two workloads (in the form of the number of SAPS and the number of BOPS respectively) are normalized to 1 in the graphs below.

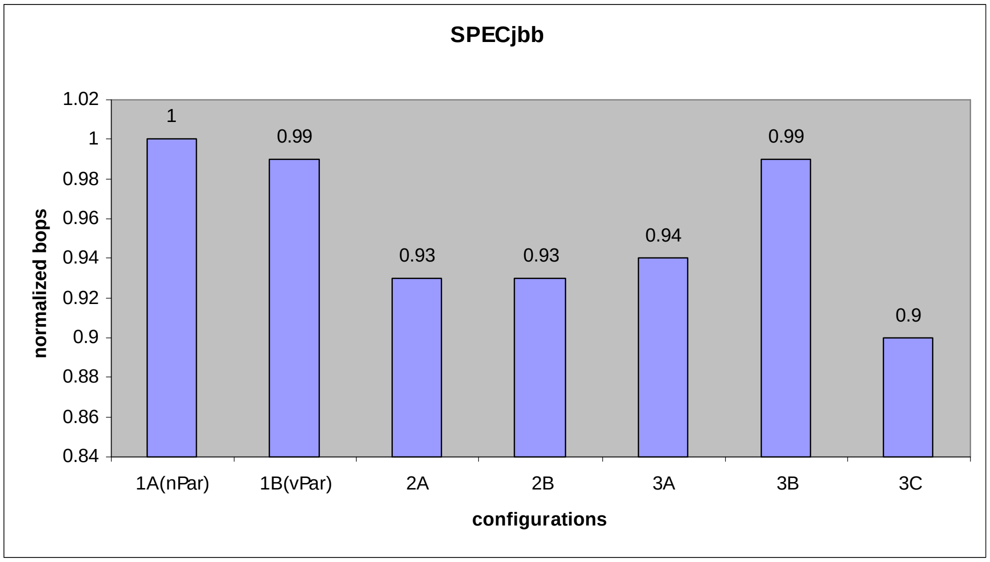
I) Single Virtual Partition Runs (Layouts 1-3) Compared with Baseline Runs

The first two graphs show results from all the runs that were done in the various single virtual partition configurations. Results were normalized relative to the baseline run and represented as fractions of 1.

Comparing the results from configuration 1B with that of 1A gives us the overhead due to the vPars software stack (in the absence of any resource locality related overheads). As shown below this overhead is 1% for both these sample workloads.

Results from configurations 2A and 2B illustrate the performance impacts due to interleaving memory across the cells in a two-cell nPartition. For the SAP SD 2-tier the impact was 4% and for SPECjbb the impact was 7%. Memory interleaving causes the average memory access latency to be higher resulting in a performance impact. The locality of the CPU cores assigned to the virtual partitions did not seem to have any noticeable impact on the performance of these two workloads.





In configurations 3A, 3B and 3C we have 75% percentage of memory configured as cell local memory and the remaining 25% as interleaved memory. The influence of the locality of the CPU core assignments with respect to the associated CLM is clearly shown here. In 3B where CPU cores and CLM are on the same cell, both the workloads exhibit just 1% degradation (as in 1B). In the case where some or all of the assigned CPU cores were from a different cell, the degradation becomes higher.

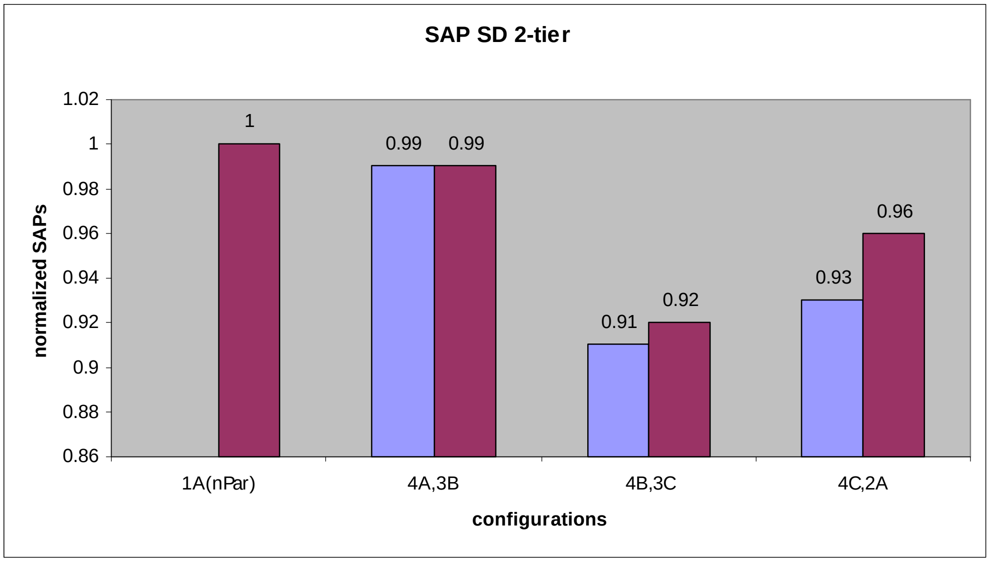
For example, in 3C where all the CPUs are from a different cell, the SAP SD 2-tier workload exhibited 8% degradation whereas in the case of SPECjbb workload exhibited 10% degradation.

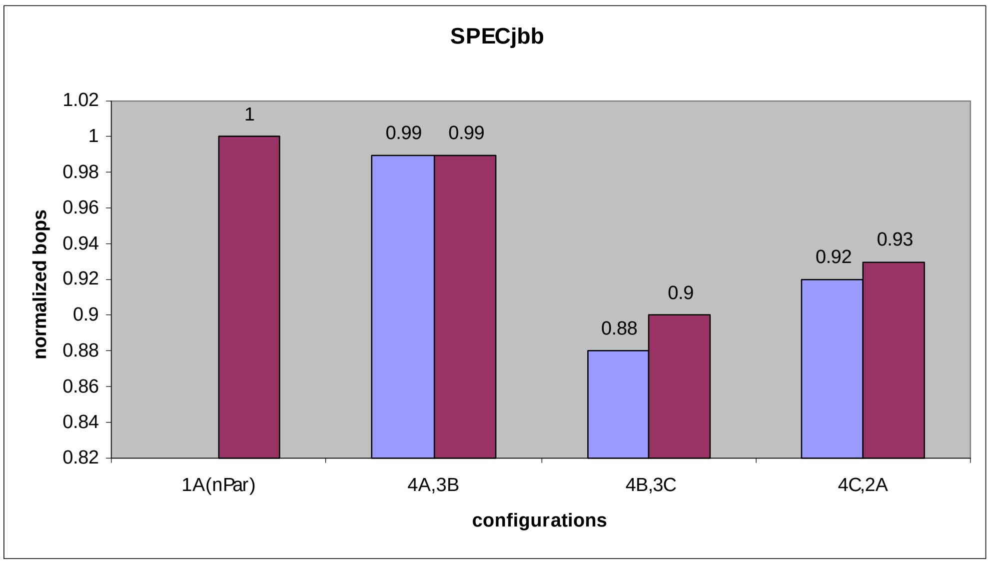
II) Two Virtual Partition Runs (Layout 4) Compared with Baseline Runs and Single Virtual Partition Runs (Layouts 1-3).

Typical customers have more than one virtual partition in a given nPartition. In some cases there may be a potential for indirect influences on the workload performance in the individual virtual partitions due to resource layouts that may arise due to the underlying shared system fabric. Following are the results of running both the workloads simultaneously in the two virtual partitions.

Results from the simultaneous runs are compared not only with the baseline (1A) but also to similar virtual partition configurations to illustrate the point. Resource configurations that are more localized do not tend to exhibit any further performance degradation. But configurations that do not have resources as local do tend to exhibit additional performance degradation.

Configuration 4A which has both the virtual partitions with localized resources performs comparably to a similarly configured virtual partition. Configuration 4B which has both virtual partitions configured with CLM and I/O in one cell but all CPU cores in another cell suffers further performance degradation when compared to the single virtual partition run (9% for SAP SD 2-tier and 12% for SPECjbb).





The results from the various experiments done as a part of this focused study are specific to the small nPartition configuration (with two cells connected to the same crossbar), and also workloads chosen to be run on them. Different workloads may have different levels of sensitivity to resource localities. The impact on the workload performance can be even higher in larger nPartition configurations (for example, with cells connected to different crossbars) that are hosting virtual partitions configured to have non-optimal resource localities.

III) Memory Granule Size and Base Versus Floating Memory

The configuration 1B was used to study the impacts of having different memory granule sizes and also different amounts of base and floating memory in the virtual partition on these two workloads. Having128MB memory granules resulted in an additional impact of about 1.5-2% degradation for the workloads. The 4096MB memory granules did not seem to exhibit any noticeable performance improvement when compared to 1024MB granules.

The percentage of base versus floating memory in the virtual partition (as long as the virtual partition had the prescribed minimum amount of base memory [4]) did not seem to make any noticeable impact on the performance of these two workloads. Other workloads may exhibit different sensitivities to the choice of the memory granule size and the percentages of base versus floating memory in a virtual partition.

Guidelines for Virtual Partition Configurations

Given the wide range of requirements for customer workloads and the various ways customers could configure their nPartitions, the vPars Monitor does not have the intelligence and higher level understanding of the intended usage of the vPars within an nPartition. It cannot make independent decisions regarding localities of resource assignments to each virtual partition. Instead the virtual partition user is given full control for assigning the virtual partition resources via a rich set of vPars command options.

The following are a few guidelines that might help a typical virtual partition customer. Please note that these guidelines are quite generic in nature, and are not meant to be interpreted as a direct outcome of the results from the focused study discussed in this paper.

1. Resource spanning. Customers should create nPartitions with the least possible number of cells and try to pick cells that are closest to one another (that is, cells that have zero or the least number of crossbar hops between them).

2. Resource locality. While configuring/modifying nPartitions with large[3] number of cells it is important to take into consideration the layout of the cells, and the number of crossbar hops. A great majority of customer workloads perform better on a cellular platform when attention is paid to resource localities. Users are highly encouraged to configure sufficient[4] amount of CLM for each cell in the nPartition.

3. Amount of resources in each virtual partition. Tools like HP Integrity Essentials Capacity Advisor can be used to collect the resource usage profiles over a period of time and analyze this data across the workloads that are being consolidated on a given target platform. Users are encouraged to use tools like these to get an estimate of the amount of resources needed for each of the workloads and, hence, each of the virtual partitions being co-hosted on an nPartition.

4. Memory granule sizes. The memory granule size chosen determines the minimum granularity of memory assignment to the virtual partitions, and also has an impact on the amount of time it takes to do dynamic memory deletion/addition. The HP-UX 11i v3 kernel limits the large page size to the size of the granule. Hence, if the chosen memory granule size is too small it can lead to a loss in performance especially if the virtual partition contains applications that benefit from large memory page sizes.[5] Users are encouraged to do some baseline performance studies for their workloads and then configure the size of the memory granule.

---

[3] The definition of large depends on the size of the server. nPartitions with 4 or more cells are considered large.

[4] The definition of sufficient depends upon the nature of the workload and the size of the nPartition. A few workloads have shown good results in nPartition configurations having 3/4 or 7/8 of the memory in each cell configured as CLM.

[5] The sensitivity to memory granule size is low as was observed in the study discussed earlier in the paper.

5. CLM and CLPs. vPars customers are encouraged to assign CLM and CLPs for each of their virtual partitions[6]. It is better to configure some amount of base CLM from a cell if the virtual partition is going to have processors and I/O on that cell, as this will allow the kernel to allocate the I/O related data structures in the same locality. When configuring CLM and CLP for each virtual partition, refer to the HP-UX NUMA documentation to decide on how to optimize the performance [1,2].

6. Multi-core CPUs. Some multi-core CPUs have a second or a third level cache that is shared by the cores on that socket. Assigning CPU cores from the same socket to different virtual partitions may, depending on the nature of the memory usage patterns of the hosted workloads, lead to some degree of indirect performance interference. Users are encouraged to do some baseline studies and if needed try to avoid or minimize CPU core assignments to different virtual partitions from a CPU with shared caches.

7. Base versus floating memory. There is a minimum amount of base memory that needs to be configured in a virtual partition to allow the Operating Environment to boot and run. Refer to [4] for more details. Users are encouraged to do some baseline performance studies for their workloads and then tune the amount of base and floating CLM and ILM memory in the virtual partition.

8. Resource migrations. While performing dynamic resource migration operations like adding/deleting memory or adding/removing CPU cores from a virtual partition there is a need to take into consideration the resource localities. Assigning processors from one cell and cell local memory from another cell might lead to less optimal performance due to increased distance between processor and memory locality. Hence addition/deletion of CLM should also take into consideration the CLP assignments and vice versa.

9. I/O Interrupts. The kernel assigns I/O interrupts during boot time amongst the active CPU cores in an OS instance using a simple round robin algorithm. (In HP-UX 11i v3, the I/O interrupt assignment takes locality into account when assigning interrupts at boot time). An administrator can see these I/O interrupt assignments and change them using the intctl command. During CPU core deletion any assigned IO interrupts are reassigned to a different active CPU core in the OS instance. Following a CPU core addition, the new active CPU core is not automatically assigned I/O interrupts. Repeated CPU core migrations (deletions followed by additions) can lead to an imbalance in the I/O interrupt assignments amongst the active CPU cores in the OS instance. I/O intensive workloads can experience performance penalties due to this. Users are encouraged to follow the guidelines mentioned in the white paper on CPU Configuration Guidelines [3] to avoid this problem.

---

[6] When HP Integrity Essentials Global Workload Manager (gWLM) is configured/initialized in a vPars environment it discovers the CLP's and CPU cores specified by hardware paths and only decides to operate on the CPU cores specified by count and the remaining un-assigned CPU cores. Any changes to the vPars CLP's or the CPU cores specified by hardware paths will require an explicit update of the gWLM configuration. Please refer to the gWLM documentation [10] for further details on this.

Conclusion

The results from the experiments described in this paper have shown that the vPars software stack overhead is very minimal. However, due to the NUMA nature of the underlying cellular hardware platforms and the performance sensitivity of workloads to resource localities it is important to pay attention to the localities of the resources assigned to the various virtual partitions within an nPartition. Resource localities in virtual partitions become even more important as the number of cells in the nPartition increases. Dynamic resource migrations (triggered either by direct user interaction or via automated tools) done without locality considerations can over a period of time lead to locality imbalances thereby affecting the performance of the hosted workload.

Customers are advised to follow the set of general configuration guidelines provided above and tune their virtual partition configurations (using the rich set of options provided by the vPars commands) to suit their workload's performance needs.

Reference

1. See Chapter 12 in HP-UX 11i Version 2 September 2004 Release Notes: HP 9000 Servers, HP Integrity Servers, and HP Workstations located at http://docs.hp.com/en/5990-8153/index.html .

2. See http://docs.hp.com/en/4913/ccNUMA_White_Paper.pdf for the white paper titled "ccNUMA Overview".

3. See http://docs.hp.com/en/8767/cpu_config.pdf for the white paper titled "CPU Configuration Guidelines for vPars".

4. See http://docs.hp.com/en/9832/vParsMemMigration.pdf for the white paper titled "Configuring and Migrating Memory on vPars".

5. See http://docs.hp.com/en/hpux11iv3.html#Virtual%20Partitions for vPars administration documents and white papers specific to the HP-UX 11i v3 release stream.

6. SAP Standard Application Benchmarks http://www.sap.com/solutions/benchmark/index.epx

7. SPEC http://www.spec.org/jbb2005/

8. Memory System Behavior of Java-Based Middleware http://www.cs.wisc.edu/multifacet/papers/hpca03_ecperf.pdf

9. See http://www.hp.com/products1/servers/scalableservers/superdome/infolibrary/technical_wp.pdf for the white paper titled "Meet the HP9000 Superdome Servers".

10. See http://docs.hp.com/en/T2786-90209/index.html for VSE management Software Release Notes Version A.03.00.01 (Refer to the sub-sections under the Global Workload Manger Release Notes)

ED1-xxxxVP-PW, November 2007