IBM® DB2® for Linux®, UNIX®, and Windows®

Best Practices

Improving Data Server Utilization and Management through Virtualization

Sunil Kamath
Senior Technical Manager
DB2 Performance Benchmarks and Solution Development

Punit Shah
Information Management

Basker Shanmugam
Information Management Performance Benchmarks and Solution Development

Rimas P. Kalesnykas
DB2 Information Development

Executive summary

This document describes the best practices for deploying the IBM DB2 Version 9 product with the IBM System p™ virtualization technology. When you run the DB2 product on the System p platform, selecting the right blend of virtualization features and their configurations to achieve desired business goals, while improving the utilization of IT resources, is a challenge. Achievable business goals are reducing administration, power, cooling, or floor space costs by consolidating data servers. Examples of ways to improve resource utilization are optimizing the performance of the DB2 product, improving processor utilization, sharing system resources, using dynamic resource allocation without rebooting, and using workload management.

This paper describes the primary System p virtualization technologies that concern the selection of the types of logical partition, disk I/O, and network interface, as well as workload management considerations. The following brief summaries describe the major considerations discussed in this paper and how they can benefit your business.

Logical partition type

Increasingly, with most hardware systems being heavily under-utilized due to sizings based on forecasted peak server activity, businesses today continually face the challenge of driving the average level of system processor utilization even higher in order to maximize their return on investment (ROI). By using shared processor partitions, businesses can efficiently consolidate multiple databases that are housed on different physical servers or dedicated partitions onto shared processor partitions on a single physical server. This sharing of processor resources, while balancing the processor requirements for both peak and average operations, reduces the total cost of ownership (TCO). One can then assign a quality of service for each of the shared processor partitions to ensure more important workloads will always get the processor resources they need while lower priority workloads will get resources on a best-effort basis. Hosting test and production applications together, on different shared processor partitions, can also help improve the quality of the test results as the test environment faithfully mimics the production environment.

Disk I/O type

With the capability to create multiple shared partitions, each with a fractional entitlement capacity, it is possible to exhaust all the physical I/O slots of the machine if a dedicated I/O slot was assigned to each logical partition. Also, in many production environments with multiple databases consolidated into multiple logical partitions, the I/O performance requirements vary quite significantly between many applications. In these cases, the virtual I/O server (VIOS) enables sharing of the disk adapter and I/O resources across multiple applications to better utilize the overall storage infrastructure and meet varied performance needs while maximizing the ROI. The VIOS feature also provides additional value-added capabilities, such as Live Partition Mobility – a feature on the POWER6™ processor family – that allows for the movement of a running partition from one POWER6 server to another with no application downtime, resulting in better system utilization, improved application availability, and energy savings.

Network type

Similar to the reasons already mentioned for the sharing of storage resources, the VIOS also handles the sharing of network adapters and, therefore, the sharing of network bandwidth across various partitions on a system. This maximizes both system resource utilization and ROI.

Workload management considerations

The types of workload management capabilities available with System p virtualization technologies are most important to businesses running customer relationship management (CRM) or transactional workloads with CPU intensive batch jobs during off-peak hours and less intensive CPU activity on the transactional system during peak business hours. These capabilities also have applicability in industries similar to retail in which typically the demand on the data server system is much higher on particular days of the year, such as Black Friday after Thanksgiving or Boxing Day after Christmas, than on other days. Efficient workload management maximizes both system resource utilization and ROI, while reducing TCO.

Test these best-practice guidelines in your test environment before implementing them in your production environment.

Introduction

Virtualization is a broad term encompassing a set of server deployment and management features. According to one definition, virtualization is a technique used to abstract the physical characteristics of the resources of a system from other systems, applications, or users interacting with those resources.

Virtualization is extremely useful because you can use it to make a single physical resource appear to be multiple logical resources, or multiple physical resources appear to be a single logical resource; for example, a processor core can appear to function as multiple virtual processors, or multiple storage devices can be consolidated into a single logical pool in order to increase utilization of the available storage space. Therefore, virtualization makes server deployment and utilization more flexible. You can reduce the costs of administration, power, and floor space by consolidating data servers through virtualization. As an added benefit, you can use virtualization to significantly increase server utilization and improve overall performance efficiency.

This DB2 best-practices document describes how to select the right blend of System p virtualization features and configurations to help you to achieve your desired business goals. (Unless otherwise noted, DB2 Version 9 refers to both DB2 Version 9.1 and DB2 Version 9.5). This document provides guidance in the following areas:

- Understanding DB2 performance and scalability in a virtualized environment
- Using Advanced Power Virtualization on IBM System p
- Choosing the right virtualization method for your DB2 environment by considering the following primary considerations:
    - logical partition type
    - disk I/O type
    - network type
    - workload management
- Planning and sizing

As other virtualization technologies, such as VMware® ESX and Windows hypervisor, become available in future, this document will be extended to incorporate best practice guidelines applicable to them. It should be noted that the concepts and techniques presented in this paper are primarily applicable to the System p platform.

DB2 Version 9 and System p virtualization overview

The System p virtualization technology offers a rich set of virtualization features implemented in the hardware and firmware. These features range from simple resource

isolation to an array of the most advanced and powerful functions, including server resource partitioning, autonomic dynamic resource reallocation, and workload management. The IBM System p family of servers has incrementally delivered virtualization features with several processors in the IBM POWER™ processor family and multiple supported operating systems. The System p platform now boasts mature, well-proven server virtualization features.

Unlike traditional hosting environments where an operating system instance controls all of the hardware resources of a server (for example, the processors, memory, and I/O devices), virtualization, in its rudimentary form, allows partitioning of server resources (logical partitioning). The virtualization is made possible by a layer called IBM POWER Hypervisor™ (PHYP), which provides an abstract view of system hardware resources to the operating system of the shared processor partition.

Virtualization features, such as IBM Micro-Partitioning™, virtual I/O (VIO), and virtual ethernet, deliver value through methods such as resource sharing, workload management, and dynamic resource allocation without operating system instance reboots (dynamic logical partitioning). VIO enables you to perform storage partitioning and sharing with just a few commands. Running in a partitioned environment, VIO can also alleviate storage administration overhead by providing a centralized focal point. The DB2 product works with VIO without any special package or driver installation.

System p hardware supports the IBM AIX, IBM i, SuSE® Linux Enterprise Server (SLES), and Red Hat® Enterprise Linux® (RHEL) operating systems. This document focuses primarily on the AIX operating system, but you can extend the same guidelines to all other supported operating systems running on POWER-based processors with little or no modification.

The DB2 Version 9 data server is IBM's fastest-growing, flagship database offering. It comes equipped with host dynamic resource awareness, automatic features such as self-tuning memory management (STMM), and enhanced automatic storage, which greatly reduce administrative overhead for tuning and maintenance. These functions make the DB2 product well suited for the virtualization environment and enable it to leverage the System p virtualization technology.

The DB2 product works seamlessly in the System p virtualization environment, straight out of the box. DB2 recognizes and reacts to any dynamic LPAR event, such as runtime changes to the computing and physical memory resources of a host partition. The STMM feature automatically adjusts and redistributes DB2 heap memory in response to dynamic changes in partition memory and workload conditions.

You can obtain additional information from the various Web sites listed in the "Further reading" section.

Virtualization terminology and concepts

This section briefly explains virtualization components and features and provides quick references to the virtualization environment. Readers familiar with virtualization terminology and concepts can skim through this section.

Term	Description
Logical partition (LPAR) or partition	A logical partition is an isolated computing domain with its own resources (processor, memory, and I/O interfaces) and operating system instance. Supported operating systems include the AIX, Linux (RedHat, SLES), and IBM i operating systems. Each LPAR can run a different type, version or level of an operating system. For example, one LPAR can run AIX 5L™ v5.2, a second LPAR can run AIX 5L v5.3 TL06, a third LPAR can run AIX 6, and a fourth can run the Linux operating system. In addition to processor and memory resources, each LPAR needs its own root disk, network interface, and storage. There are ways to simplify and share network and storage adapters using virtual I/O, as explained in more detail later. There are two types of LPARs: • Dedicated processor • Shared processor (which uses the IBM Micro-Partitioning feature).
Dynamic logical partition (DLPAR) or dynamic reconfiguration	The DLPAR facility lets you alter the resources of a partition at run time, without restarting the operating system. Examples of items that you can alter are the number of processors for dedicated partitions; the number of virtual processors and entitled capacity for shared processor partitions; and the number of virtual I/O adapter slots and the amount of physical memory for either type of partition. The DLPAR facility enhances resource utilization by allocating resources where they are needed most. You can access the facility manually through the Hardware Management Console [1] (HMC), or you can automate the access by using a workload management tool. The DLPAR facility is essential for workload management tools such as

[1] HMC provides administration functions to manage a system or group of systems, including the ability to create and alter a partition.

	IBM Enterprise Workload Manager™ (EWLM).
POWER Hypervisor (PHYP)	PHYP acts as the abstraction layer between the system hardware and the LPARs, enabling multiple operating systems to run on POWER processor-based systems. PHYP is a key component of the IBM virtualization technology that enables Micro-Partitioning, shared processor pools, dynamic LPAR, virtual I/O, and virtual LANs. Among the many tasks that PHYP performs is saving and restoring all processor-state information during LPAR context switching.
Dedicated processor partition	A dedicated processor partition has one or more assigned processors exclusively reserved for it by PHYP. (You can assign processors in increments of one.) While this partition is active, other processors cannot use the idle processor capacity. PHYP uses the same physical cores to schedule partitions to benefit from a warm cache. Dedicated processor partitions can improve LPAR throughput through processor and memory affinity and help ensure maximum processor cache hierarchy performance. This feature applies to the POWER5™ [2] family of processors.
Shared processor partition	You can assign processor capacity to a shared processor partition in increments of 1/100th or 1% of a physical processor. However, each partition requires a minimum of 1/10th or 10% of physical processor capacity. As a result, there is a maximum of 10 partitions per physical processor. Shared processor partitions require the IBM Advanced Power Virtualization (APV) [3] feature and use the IBM Micro-Partitioning feature.
Entitled capacity	Entitled capacity refers to processor capacity that is assigned to a shared processor partition.

[2] A new feature for POWER6, called Shared Dedicated Capacity, enables spare CPU cycles that are "donated" by dedicated processor partitions to be utilized by the shared pool, thus increasing performance and overall system utilization. The dedicated partition maintains first priority for using the dedicated CPU cycles, and sharing occurs only when the dedicated partition has not consumed all of its resources.

[3] APV is a paid, separately licensed feature. It is also required for VIOS.

	This capacity is a guaranteed amount of processing capacity. PHYP slices up physical processor time to manage fractional processor allocation.
Capped shared processor partition	Entitled capacity is the hard limit for a capped shared processor partition (that is, it cannot exceed its entitled capacity when demand is high, even if idle processing capacity is available in a shared pool). Capped shared processor partitions enable you to better manage the shared-pool processor resources.
Uncapped shared processor partition	An uncapped shared processor partition can use not only its entitled capacity but also available idle processing capacity in the shared pool. Idle shared-pool processor capacity is immediately yielded to meet workload demand at peak times when a partition needs more than its entitled capacity. Uncapped partitions are very flexible and useful for unpredictable workloads. When there is more than one uncapped shared processor partition, shared-pool processor capacity is assigned to a shared processor partition based on its assigned uncapped weight.
Uncapped weight	The uncapped weight is a number that specifies the portion (weighted average) of shared-pool idle processor capacity to be assigned to a shared processor partition when there is more than one overburdened shared processor partition. The uncapped weight is a number between 0 and 255; the default uncapped weight is 128. The higher the uncapped weight, the more idle shared-pool processor resources are granted. You can set the uncapped weights for various shared processor partitions according to partition workload priority, setting the highest uncapped weight for the highest-priority workload.
Virtual processor	This concept is relevant to shared processor partitioning only. A virtual processor is the entity to which the operating system dispatches application threads (or processes). You can view the virtual processor as a processor in a traditional

nonpartitioned environment.

For example, assume that a shared processor partition has 1.50 entitled capacity with two virtual processors running AIX 5L. In this case, the AIX kernel scheduler acts as if it has two real processors. Assuming simultaneous multi-threading (SMT) is turned OFF, the operating system schedules a maximum of two threads at a time.

You can consider the virtual processor to be virtual core. The number of virtual processors is unaffected by SMT.

Virtual I/O (VIO)

Virtual I/O is a broad term that refers to a set of storage and network virtualization features:

- virtual Ethernet
- shared Ethernet adapter (SEA)
- virtual storage.

Virtual Ethernet. Without requiring additional hardware or external cables, a virtual LAN (VLAN) facilitates high-speed virtual Ethernet communication paths among multiple partitions within a physical system that run AIX, Linux, and other operating systems. You can dynamically create virtual Ethernet segments and restrict access to a VLAN segment to meet security or traffic segregation requirements. A virtual Ethernet has the same characteristics as a high-bandwidth, physical Ethernet network and supports multiple networking protocols, such as IPv4, IPv6, and ICMP. PHYP provides this feature.

Virtual I/O server (VIOS). This is a special-purpose partition that provides virtual I/O resources to client partitions. The VIOS owns physical adapters. You can share a physical adapter by assigning it to multiple client partitions, which minimizes the number of physical adapters that you require for individual clients. Thus, the VIOS can reduce costs by eliminating the need for dedicated network and disk adapters. The VIOS provides two major functions:

- **A virtual Ethernet bridge.** The shared Ethernet adapter (SEA) hosted in the VIOS acts as a layer-2 bridge between the internal virtual and external physical networks. The SEA enables partitions to

communicate outside the system without having to dedicate a physical I/O slot and a physical network adapter to a client partition.

- **A virtual SCSI adapter.** With this virtual interface, physical storage (in cases of device backing) and logical volumes (in cases of logical volume backing) that you create and export from a virtual I/O partition are shown at the client partition as a SCSI disk.

VIOS requires the APV [4] feature.

Free-pool capacity

Free-pool capacity refers to the amount of free processing capacity available for use by uncapped shared processor partitions. You can calculate this amount by adding the unassigned processing capacity (that is, it is not guaranteed to any LPAR using processor entitlement or by virtue of a dedicated LPAR) to the sum of unused entitlements for all the shared processor partitions active in a system.

Live Partition Mobility (LPM)

Live Partition Mobility is available starting with POWER6 processor-based systems. LPM is designed to enable migration of an LPAR from one system to another compatible system. LPM uses a simple and automated procedure that transfers state and configuration information from the source server to the destination server without disrupting the hosted operating system or applications. This feature requires a VIOS.

Workload partition (WPAR)

A workload partition is a new feature in AIX 6.1 that creates multiple AIX virtual partitions based on a single global AIX install. One can create a System WPAR or an Application WPAR. DB2 product installation is currently supported only on the system WPAR. WPAR mobility enables partitions to be moved from one physical system to another. Refer to the AIX and DB2 documentation for minimum AIX and DB2 levels to install the DB2 product within a WPAR.

[4] APV is a paid, separately licensed feature. It is also required for shared processor partitions.

Choosing virtualization features

Now that you are familiar with the System p virtualization concepts, consider how you can apply virtualization to the DB2 environment. There are four primary decisions that you must make about the virtualization environment:

- Logical partition type: dedicated or shared processor partition
- Disk I/O type: locally-attached I/O or virtual I/O
- Network type: physical or virtual
- Workload management settings

In some cases, it is easy to change a decision later. For example, changing a partition type from a dedicated to a shared processor partition is relatively easy. However, switching the disk I/O type between virtual I/O and locally-attached I/O requires a data backup and restore. Therefore, careful planning is necessary before putting a system into production.

Note that the shared processor partition and VIOS features are not part of the standard virtualization set. These features are part of the APV feature.

The following sections show how you can best apply the primary System p virtualization technologies to your DB2 environment to build a comprehensive IT infrastructure that continues to meet your desired business goals. The basic assumptions are that there is a single instance of the DB2 product running on each logical partition, and a single database per instance. The test results, best practices, and other information in these sections should make your decision-making process easier. For graphical representation of the test results, refer to the "DB2 and System p virtualization" white paper listed in the "Further reading" section.

In the following discussions, unless otherwise noted, the term partition refers to a System p logical partition (LPAR). Do not confuse this term with a database partition created by using the DB2 Database Partitioning Feature (DPF).

Logical partition type

Frequently, questions arise regarding whether dedicated or shared processor is the correct logical partition type for a particular workload and system environment. There are a number of factors to consider, such as performance, overall system utilization, and consolidation of multiple workloads onto a single server to improve power usage and cooling efficiencies. The following subsections address these factors.

Partition type: dedicated compared to shared processor

Use a shared processor partition with the DB2 product. This partition type provides many benefits, including flexibility and better processor utilization.

Theoretically, due to underlying processor and memory affinity, a dedicated partition should offer optimal performance under a number of conditions. However, it has been recently determined that this is incorrect in practice. In the tests described in this subsection, there was no significant performance difference between dedicated and shared processor partitions for the system under test here (see the Appendix for the test computer system specifications). Table 1 summarizes the test environment.

Table 1 - DB2 and System p virtualization test environment

	Number of partitions					
	1		2		4	
	Partition type		Partition type		Partition type	
	Dedicated	Shared	Dedicated	Shared	Dedicated	Shared
Processing capacity	8	8.00	4	4.00	2	2.00
Virtual processors	NA	8	NA	4	NA	2
Database size	~200 GB		~100 GB		~50 GB	
Memory	128 GB		64 GB		32 GB	
Number of data disks	192		96		48	
I/O type	Locally Attached		Locally Attached		Locally Attached	

We performed six performance runs: two performance runs each for a single partition, for two partitions running simultaneously, and for four partitions running simultaneously. We performed one run for each partition type for each number of partitions. The partition type change was a non-destructive one (there was no data loss); the only requirement was to restart the partition or partitions. All tests used locally-attached I/O with data that was correctly scaled.

The results showed that there was no significant difference in performance between the types of partitions. Performance of the shared processor partition (that is, the cumulative throughput) was identical to that of the dedicated processor partition for one, two, and four partitions. This behavior validated the fact that PHYP uses a very efficient scheduling algorithm for shared processor partitions with reasonably large processor entitlements. For such shared processor partitions, PHYP attempts to schedule the virtual processor on the same physical core each time to preserve the processor cache affinity, similar to dedicated processor partition functionality.

Linear scalability

Use a shared processor partition so that you can take advantage of its ability to increase processor utilization.

When we considered the average throughput per logical partition in our tests, we made an interesting observation: Partition scalability, for either type of partition, does not have any overhead. That is, the amount of work done by an eight-processor dedicated processor LPAR (or shared processor LPAR with 8.00 entitled capacities and eight virtual processors) is twice that of a four-processor LPAR (or shared processor LPAR with 4.00 entitled capacities and four virtual processors), which, in turn, is twice that of a two-processor LPAR (or shared processor LPAR with 2.00 entitled capacities and two virtual processors). In addition, we found that there was no significant difference in the normalized average throughput per LPAR between dedicated and shared processor partitions.

A general guideline is not to waste processing capacity within a partition. It is cost effective to reduce the unused processing capacity within a partition by appropriately sizing a partition with respect to the number of virtual processors implemented, which results in improved physical processor utilization at a system level.

Therefore, with dedicated and shared processor partitions being equal in both scalability and average throughput, use a shared processor partition to take advantage of its ability to increase processor utilization.

Capped or uncapped processor partition

Use uncapped shared processor partitions, which are very effective in dealing with unpredictable workloads. However, use them carefully: to manage workloads, set partition priorities by appropriately setting their uncapped weights.

One of the primary distinguishing features of a shared processor partition is its ability to use available idle processing capacity, in addition to its entitled capacity. A dedicated processor partition is capped by definition: that is, it cannot use more than its entitled capacity on allocated processors, even if idle processing capacity is available on the system. When used appropriately, an uncapped shared processor partition is very effective in dealing with highly dynamic or unpredictable workloads. The uncapped weight setting is used to allocate free-pool capacity among competing uncapped shared processor partitions.

For example, consider a multi-tier infrastructure that consists of a DB2 server, as a back-end tier, and IBM WebSphere®, as an application server. Each of these is hosted in its own uncapped shared processor partition of a System p server. The DB2 partition is designated to have a higher priority because its uncapped weight is set higher than that of the WebSphere partition. During a period of peak demand, both servers exceed their entitled capacities and are allowed to use free-pool processing capacity (if available). However, due to the higher uncapped weight setting of the DB2 partition, it gets more free-pool processing capacity than the WebSphere partition gets.

Using an uncapped shared processor partition is a useful technique for dynamically, granularly, and automatically managing idle processor capacity to accommodate unpredictable workloads in a virtualized environment. For details about a similar workload management solution that serves as a proof of concept, refer to the "Workload Management" paper listed in the "Further reading" section of this document.

Number of virtual processors in a shared processor partition

For optimal balanced performance, set the number of virtual processors no higher than two virtual processors per 1.00 entitled capacity.

To maximize total system processor utilization, set the number of virtual processors to the rounded-up value of usable entitled capacity.

Using shared processor partitions requires setting the number of virtual processors to an appropriate value. Underconfiguring the number of virtual processors can lead to wasted processing capacity or might not allow the use of free-pool processing cycles (the maximum entitled capacity of an uncapped partition is limited to the number of virtual processors). Overconfiguring the number of virtual processors is not a problem because by default, the AIX operating system does not schedule unused virtual processors (refer to information about the AIX **schedo** command and the **vpm_xvcpus** configuration parameter). We performed a test to measure virtual processor folding effectiveness, which is a mechanism responsible for scheduling the minimum number of virtual processors required for the partition, by varying the number of virtual processors for one of the shared processor partitions contained in the four-partition scenario summarized in Table 1. The partition had an entitled processor capacity of 1.85. We performed three runs: in the first test, we set the number of virtual processors to the rounded-up entitled capacity of 2, and in subsequent tests, we configured much larger numbers of virtual processors. Indeed, as expected, a relatively higher number of virtual processors did not have any negative performance impact.

For capped partitions in the DB2 environment, set the number of virtual processors equal to the rounded-up entitled capacity. For uncapped partitions, set the number of virtual processors to a value in the range derived by using the following formula [5]:

```
RVP = {round up (CEC + [UW/TUW] * FPC) , round up (CEC + FPC)}
```

where:

- **RVP** represents the rounded-up range of virtual processors.
- **CEC** is the current entitled capacity.
- **UW** is the uncapped weight for the shared processor partition. The default value is 128, and the range is 0 - 255.
- **TUW** is the total uncapped weight of all uncapped shared processor partitions.
- **FPC** is the free-pool capacity (free unassigned processor capacity).

For example, assume that you have a 16-core IBM POWER 570 system. You create one dedicated processor partition with 4 processors and you intend to create 6 uncapped shared processor partitions across the remaining 12 processors. You leave the uncapped weight as 128 (the default value).

To run a DB2 instance, suppose that you want to create an uncapped shared processor partition with an entitled capacity of 2.5. Use the formula described previously to determine the range of the number of virtual processors that you would need to set for the partition, as shown in the following calculations. For simplicity, assume that all partitions are uncapped and have the same uncapped weight.

```
RVP = { round up (2.5 + [128 / (128 *6)] * (12-2.5)) , round up
(2.5 + (12-2.5))}

RVP = { round up (4.08) , round up (12)}

RVP = { 5 , 12 }
```

This uncapped shared processor partition formula calculates a range of values to select from when assigning the appropriate number of virtual processors to a partition. This range spans the extremes of workload objectives: optimal balanced performance and maximum total system processor utilization.

A general guideline is to set the number of virtual processors based on your workload objectives. If performance is of primary importance and the workload is relatively stable, use the lower value of the range (5, in the previous example). If maximizing the system processor utilization is of most importance, use the higher value of the range (12, in the

[5] The current amount of entitled capacity or the partition profile settings might limit the number of virtual processors. At least 0.10 (10%) of entitled processor capacity is required per virtual processor; that is, a maximum of 10 virtual processors is allowed per one whole physical processor.

previous example). The larger number in the range ensures that there is a sufficient number of virtual processors available in order to make use of free-pool processing capacity. Typically, the best compromise might be to set the number of virtual processors to a value somewhere between the lowest and highest values of the calculated range based upon your requirement of some degree of weighted balance between performance and processor utilization. This within-range setting would be particularly beneficial if you experience relatively frequent occurrences of unpredictable workloads that could take advantage of the unutilized processors in the share pool.

Additional processor-type considerations

For a dedicated partition, processor assignment is in increments of one whole processor. Therefore, idle processor capacity often occurs within a dedicated partition. For example, a dedicated partition with 3 processors that runs the DB2 product and consumes the equivalent of 2.10 processors (that is, vmstat shows 70% partition utilization) results in a waste of 0.90 (30%) of processor capacity.

The only way to alter a dedicated partition processor allocation, without having to restart the AIX operating system or the DB2 product, is by using a DLPAR operation. A DLPAR operation is relatively slower (with a latency of a few seconds, depending on the workload) than the near-instant entitled capacity change that can be performed on an uncapped shared processor partition.

The previous information is relevant mostly to servers based on processors prior to POWER6. Beginning with the POWER6 processor, there is an option to donate unused processing capacity from dedicated processor partitions. This feature eliminates unused processing cycles in a dedicated processor partition, further solidifying the System p virtualization offering.

Logical partition type decision-tree flowchart

The following decision-tree flowchart guides you through the process of determining the logical partition type for your DB2 environment, based on both functional and non-functional requirements.

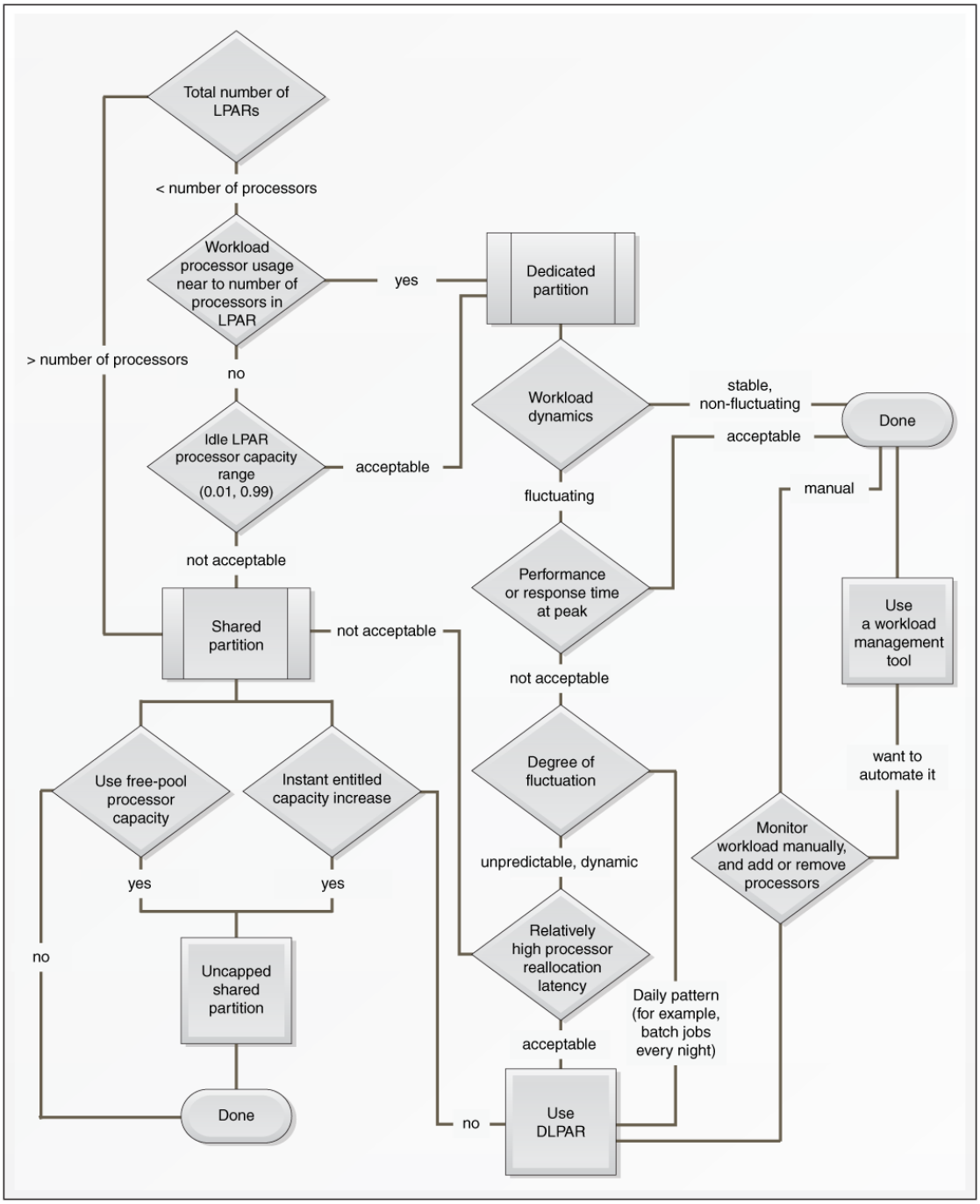

Figure 1. Logical partition type decision tree for the DB2 environment

Disk I/O type

DB2 performance is heavily dependent on the performance of the I/O subsystem. To attain the best possible I/O throughput, pay special attention to the data layouts of database tables. The I/O type that you choose greatly impacts the manageability and extensibility of the storage system. Thus, it is critical to consider workload priorities and to examine tradeoffs between disk I/O types. You can choose between locally-attached I/O, virtual I/O, or both within a partition.

Virtual I/O in practice

You must install and run a VIOS to use virtual disk I/O.

Virtual I/O enables consolidation, serving many partitions by using fewer physical resources and significantly reducing hardware and administrative costs.

Virtual I/O can be valuable for many DB2 environments. For example, each partition requires at least one root disk. Without virtual I/O, the number of partitions is limited to the number of available PCI bus slots or physical disk adapters for the locally-attached I/O devices because each partition would need a physical disk adapter to access the disk. The only way to share a physical disk adapter and disk is to use virtual I/O with Logical Volume Manager (LVM), which enables multiple logical partitions to share a physical disk, even if saving adapter costs is not a priority.

As disk technologies evolve to increase storage capacity, the ratio of disk size to disk bandwidth gets larger as time goes on, leading to extra free disk space in a DB2 environment, typically sized by the number of disk spindles rather than by disk space.

Disk sharing can be very effective in making use of the entire disk storage capacity or in sharing disk I/O bandwidth for staggered workloads. For example, daytime online transaction processing (OLTP) workloads and nightly batch jobs can share storage subsystem bandwidth.

All virtual I/O requests are passed down to the VIOS, which performs the disk I/O operations and returns data directly to a client partition (no double buffering in the case of virtual SCSI).

What follows is a short list of additional virtual I/O features. More information about virtual I/O and ways to implement it are available in the *Advanced POWER Virtualization on IBM System p5: Introduction and Configuration* reference.

- Low-cost, LVM-level storage redundancy with a mixture of locally-attached and virtual I/O devices
- Support for dual VIOSs to eliminate single points of failure
- SEA, or trunk adapter

- Fast interpartition connectivity, using a VLAN

Virtual disk I/O scalability

The resource requirement for the VIOS itself is very modest. The memory requirement is stationary. The major processor requirement for the VIOS comes from the number of disk I/O operations.

Recall that virtual I/O is enabled by a special partition called the VIOS, which runs a specialized operating system. Similar to a regular operating system, the VIOS uses processor and memory resources. However, processor and memory requirements are modest, even for a very high number of I/O requests.

The VIOS does not use any more memory than it requires for running the VIOS operating system, nor does it need memory for client partition disk I/O requests. This is because virtual SCSI is a zero-memory copy operation: that is, data is copied directly into the LPAR-hosted application or file system space of the client.

We performed tests using the same environment described in Table 1 except that we changed the I/O type from locally-attached to virtual I/O. Table 2 summarizes the virtual I/O test environment. All of the tests used device backing for the virtual storage. For DB2 storage, we used database-managed space (DMS) table spaces, created with the NO FILESYSTEM CACHING option. The VIOS client partitions running the DB2 product have the processing capacity shown in the table. The test used eight processors: the client partitions used 7.40 of the processors, and the remaining 0.60 processor was allocated to the VIOS partition.

Table 2. DB2 and virtual I/O test environment

	Number of Partitions			
	1	2	4	VIOS
	Partition Type Shared	Partition Type Shared	Partition Type Shared	Partition Type Shared
Entitled capacity	7.40	3.70	1.85	0.60
Capped or uncapped	Uncapped	Uncapped	Uncapped	Uncapped
Virtual processors	8	4	2	1
Database size	~200 GB	~100 GB	~50 GB	N/A

Memory	128 GB	64 GB	32 GB	512 MB
Number of data disks	192	96	48	N/A
I/O type	Virtual I/O	Virtual I/O	Virtual I/O	N/A

This test was designed to demonstrate the scalability of virtual I/O and the total scalability of the DB2 product and System p virtualization solution. In this scenario, both processor and disk I/O usage was based on the virtualization technology.

The results showed that the virtual I/O scaled linearly with the increasing VIOS client partition processing capacity and with the number of increasing concurrent VIOS client partitions. The use of virtual I/O is independent of the client partition type: that is, virtual I/O does not depend on whether the client partition is a dedicated or shared processor partition.

Virtual I/O and VIOS tuning

Set the VIOS partition type to an uncapped shared processor partition with the highest uncapped weight so that the VIOS will get any additional processing capacity that it requires.

You must perform only a small amount of tuning to use virtual I/O with the DB2 product. The first and most common question is, "How much memory and processor is required for the VIOS?"

VIOS memory requirements are insignificant. Because the VIOS does not need memory for an I/O request from a client partition, the VIOS memory requirement does not depend on factors such as the number of I/O operations or the number of client partitions. We performed all of the tests described in this paper by using a constant VIOS memory of 512 MB.

VIOS processor requirements are also insignificant. In our test, the results showed that VIOS processor usage was approximately 0.55 of a processor, even for a very high number of disk operations. VIOS processor requirements increase as the number of I/O operations increases.

When you are determining processor requirements, consider the peaks and valleys of the workload, not just the average.

To help ensure proper VIOS sizing, with the ability to meet unexpected workload volumes, use the following approach:

1. To derive the initial amount of processing capacity, set the VIOS partition type as an uncapped shared processor partition.

2. Perform a trial run with a peak workload, and record the VIOS entitled capacity utilization (using the VIOS **viostat** command).

3. Add 10% extra capacity to the peak entitled capacity utilization that you recorded, and use the total capacity as the entitled capacity for the VIOS.

4. Set the VIOS partition type to an uncapped shared processor partition with the highest uncapped weight [6].

On POWER6 processor-based servers, use a dedicated processor partition with Shared Dedicated Capacity for the VIOS. A new feature on POWER6-based servers, Shared Dedicated Capacity enables a dedicated processor partition to donate spare processor cycles to the shared pool of dedicated processor partitions, thus increasing overall system performance. The dedicated partition that donated the cycles has priority for using those processor cycles; however, sharing occurs only when the dedicated partition has not consumed all of its resources.

Monitor the VIOS processor requirement frequently, especially after a change in DB2 I/O or storage characteristics, such as bufferpool size, and table space page size. Different changes in the DB2 I/O or storage characteristics stress the storage subsystems differently and might result in the VIOS needing more or less processing capacity.

In general, for any partition, a virtual processor cannot span physical processors: that is, a virtual processor cannot use more than one physical processor. The same is true for the VIOS. In cases where the VIOS needs more entitled capacity than that allowed by the currently configured number of virtual processors, you can assign an additional virtual processor to the VIOS partition by using the DLPAR facility.

The virtual I/O client partition performs I/O by using the virtual SCSI adapter. As with real, physical adapters, you can tune the virtual SCSI adapter by using the **queue_depth** parameter in AIX 5L v5.3 TL05, or later. Set the appropriate **queue_depth** value for the virtual SCSI adapter of the client partition in addition to setting the appropriate **queue_depth** value for a physical adapter at the VIOS.

We performed tests with multiple virtual SCSI adapters at both the client partition and the VIOS, including using a dedicated virtual SCSI adapter for the DB2 transaction log and a dedicated virtual SCSI adapter for each table space. The results showed that there was no difference in performance between using one virtual adapter and using multiple virtual adapters.

[6] Note that processor entitlement above the entitled capacity is possible only if there is idle processing capacity in a free pool. Ensure that you set the minimum, desired, and maximum entitled capacity and virtual processors for a VIOS partition appropriately, in a partition profile at the HMC, to allow for needed increases or decreases in resources.

Network type

Virtual I/O supports two types of virtual network interfaces. The SEA, or trunk adapter, can serve many client partitions that use a single physical Ethernet adapter to connect to a public network. SEA requires a VIOS. Another type of virtual network interface, called virtual Ethernet, provides fast connections between system partitions. For example, you can use virtual Ethernet to provide fast connections between the n-tier workload, the DB2 product, and the WebSphere application server. The virtual Ethernet interface does not require physical adapters, cables, or even a VIOS. Virtual Ethernet is served by PHYP and uses PHYP memory.

Typically, a lossless, fast, within-memory connection, by way of virtual Ethernet, tends to be faster and more reliable than a physical network adapter. However, both types of virtual network interfaces use the processor cycles of the client partition to perform network communication. To determine how much additional processing capacity is required to use the virtual network interfaces, we designed the following test. Using the test environment outlined in Table 1, rather than a client workload generator running locally in the same partition, we cross-connected the partitions so that they acted as a mutual client workload generator for the two-partition and four-partition cases. We connected the clients to the DB2 server using either a physical gigabit Ethernet adapter or a virtual Ethernet interface, and we measured the transaction throughput per partition.

There was no difference in transaction throughput between the physical and virtual Ethernet interfaces. Note that the test did not involve a VIOS because virtual Ethernet does not require one.

Workload management settings

Workload management refers to a comprehensive IT strategy for addressing the dynamic nature of workload demand while reducing the costs of managing the IT infrastructure and not overburdening it. One key to workload management is balancing workloads and applying resources to high-priority applications to meet service level agreements (SLAs).

Virtualization offers many levels of workload management features. You can change the amount of memory, uncapped weight, entitled capacity, and number of virtual processors dynamically by using the DLPAR facility, without restarting the operating system or DB2 server. You can access the DLPAR facility through the HMC, either directly or by using WebSM[7] , or a remote shell invocation from anywhere on the network. When you enable the DB2 STMM feature in conjunction with the DLPAR facility, the STMM feature automatically senses changes in memory usage and can resize DB2 heaps to help maximize performance. See the "Workload Management" paper for more information about how to use STMM and its benefits.

In addition, as mentioned earlier, an uncapped shared processor partition is very effective in dealing with highly dynamic or unpredictable workloads when the uncapped weight setting for the partition running the DB2 product is higher than the uncapped weight setting for the other partitions, giving that partition top priority during the allocation of idle processor capacity.

Beyond the workload management capabilities just mentioned, IBM offers IBM Tivoli Intelligent Orchestrator workflows for provisioning and optimizing the IT infrastructure. Another offering, IBM Enterprise Workload Manager™ (EWLM), is an end-to-end IT resource optimization solution. Refer to the "Workload Management" paper for more information about using EWLM with the DB2 product.

[7] WebSM is a tool that provides a complete set of Web-based system management interfaces for the entire RS/6000 domain.

Best Practices

Logical partition type selection

- Use shared processor partitions to increase processor utilization and provide more flexibility. Uncapped shared processor partitions are very effective if workloads are unpredictable.
- For optimal balanced performance, set the number of virtual processors no higher than two virtual processors per 1.00 entitled capacity. Do not change the value of the **schedo** configuration parameter **vpm_xvcpus** from its default value of 0.
- To maximize total system processor utilization, set the number of virtual processors to the rounded-up value of usable entitled capacity. Usable entitled capacity is the sum of the current entitled capacity (CEC) of a partition and the eligible free-pool capacity (FPC).
- In order to manage unpredictable workloads, set the uncapped shared processor partition priority by setting the uncapped weight higher than any other partition.
- You can change the amount of memory, uncapped weight, entitled capacity, and number of virtual processors dynamically by using the DLPAR facility. You can automate these settings with some workload provisioning tools, including Tivoli Intelligent Orchestrator.

Disk I/O type selection

- Virtual I/O makes it possible to have more partitions than physical slots and adapters.
- One virtual SCSI client adapter is sufficient to service any number of disk I/O operations. Test results showed that there

was no difference in performance between using one virtual adapter and using multiple virtual adapters.

- Virtual I/O allows for consolidation, serving many partitions by using fewer physical resources and significantly reducing hardware and administrative costs.

- You must install and run a VIOS to use virtual disk I/O. The resource requirement for the VIOS itself is very modest. The memory requirement is stationary. The major processor requirement for the VIOS comes from the number of disk I/O operations.

- Set the VIOS partition type to an uncapped shared processor partition with the highest uncapped weight so that the VIOS will get any additional processing capacity that it requires, even in cases of unpredictable, dynamic workloads.

Virtual network selection

- You can use the SEA feature to share a single physical network adapter among many client partitions. SEA requires a VIOS and at least one physical Ethernet adapter to connect to the public network.

- You can use the virtual Ethernet interface for fast, lossless, reliable connections between system partitions. The virtual Ethernet interface does not require physical network adapters, cables, or even a VIOS.

- No difference in transaction throughput was found between physical and virtual network interfaces.

Workload management settings

- Use the DLPAR facility to dynamically change the amount of memory, uncapped weight, entitled capacity, and number of virtual processors without restarting the operating system or DB2 server.

- Use an uncapped shared processor partition to effectively deal with highly dynamic or unpredictable workloads. Set the

uncapped weight setting, for the partition running the DB2 product, higher than the uncapped weight setting for the other partitions.

Software levels

- The desired minimum level of the operating system is AIX® 5.3 TL05 Service Pack 3, or above. The desired level is AIX 5.3 TL07. If using AIX 6.1, the minimum desired level is AIX 6.1 Service Pack 2.
- The desired minimum level of the DB2 product is DB2 Version 9.1 FP4, or above. If using DB2 Version 9.5, the desired level is DB2 Version 9.5 FP1.

Summary

System p virtualization offers a set of resource partitioning and management features such as LPARs, the DLPAR facility, and virtual I/O, under which you can implement SEA, virtual Ethernet, virtual SCSI, or a VLAN. Shared processor partitions enable you to create an LPAR using as little as 0.10 of a processor. The DLPAR facility enables you to change the LPAR resources (processor, memory, and I/O slots) at run time, without rebooting the operating system.

The versatility of the DB2 data server and a variety of possible combinations of System p virtualization features help DB2 applications to perform optimally in many situations. The System p virtualization technology can be configured to realize both computer system and business benefits, which include high performance, workload isolation, resource partitioning, maximum resource utilization, and high availability at low cost. This technology reduces total cost of ownership (TCO) while also enhancing expandability, scalability, reliability, availability, and serviceability.

The best practices presented in this paper are essentially lessons that have already been learned through our own testing. These best practices serve as an excellent starting point for using the DB2 product with System p virtualization. You can use them to help to avoid common mistakes and to fine-tune your infrastructure to meet your goals for both your business and IT environment. To validate the applicability of these best practices before using them in your production environment, establish a baseline and perform sufficient testing with the various virtualization features.

Appendix: Test environment

We tested DB2 9 by running a synthetic online transaction processing (OLTP) workload in a virtualized environment. The objectives of the test included the following ones:

- Ensuring compatibility between DB2 9 and virtualization features
- Measuring the performance and scalability of DB2 9 when using it with a combination of virtualization features
- Examining deployment options from performance and IT management perspectives

The test plan encompassed comparing the relative performance of the partition types and partition scalability for combinations of dedicated and shared processor partitions, locally-attached I/O, virtual I/O, shared Ethernet adapters, and virtual Ethernet. We divided a 16-core system into one, two, and four partitions with DB2 data that was adequately scaled for the partition size. For each case, we measured the DB2 throughput by using the transactions per second (TPS) metric. Theoretically, under identical conditions (an equal amount of resources and data to be processed), the total amount of work done across the partitions should be the same, whether there were one, two, or four partitions. Any TPS disparity would reveal scalability degradation that is related to the partitioning. Using the same test environment, we switched the partition types between dedicated partitions and shared partitions (that used Micro-Partitioning) to record any performance differences.

After establishing a performance baseline with the partition type, we ran tests varying the I/O types to assess virtual I/O scalability.

Table 3 summarizes the test system environment.

Table 3. System specifications

Hardware	System	System p5 570 with 16x 2.2 GHz POWER5+
	Number of cores used	Eight
	Physical memory	128 GB
	Disk characteristics	
	Number of disks	192 external and 24 internal disks
	Type, size, speed	SCSI RAID, 72 GB, 15 000 RPM

Software	Operating system	AIX 5L v5.3 TL04, 64-bit kernel with the IBM Advanced Power Virtualization (APV) feature
	Database	DB2 9.1 for Linux, UNIX, and Windows, 64-bit
Workload	Characteristics	Online transaction processing (OLTP)
	Database size	Approximately 50 GB to 200 GB (based on partition size)

Further reading

These links provide useful references to supplement the information contained in this document:

- DB2 Best Practices http://www.ibm.com/developerworks/db2/bestpractices/
- IBM System p and AIX Information Center http://publib.boulder.ibm.com/infocenter/pseries/index.jsp
- IBM Publications Center http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US
- Advanced POWER Virtualization on IBM System p5: Introduction and Configuration (SG24-7940-02) http://www.redbooks.ibm.com/abstracts/sg247940.html
- IBM PowerVM Web site http://www.ibm.com/systems/p/apv
- "System p 570 and Advanced Power Virtualization" http://www.ibm.com/servers/enable/site/peducation/wp/1311a/1311a.pdf
- Implementing System p virtualization with DB2 and WebSphere using IBM Enterprise Workload Management http://www.ibm.com/developerworks/systems/library/es-pvirtualizationewlm/index.html
- Power Systems Virtualization Wiki http://www.ibm.com/collaboration/wiki/display/virtualization/Home
- "DB2 and System p virtualization" http://www.ibm.com/developerworks/db2/library/techarticle/dm-0801shah/index.html#download
- "AIX Live application mobility demo"

  This demo illustrates how an IBM DB2 database server that is running in support of an SAP application is relocated from one IBM System p520 server to a larger p570 model. This relocation is done without shutting down the SAP application and while a simulated user load is applied to the SAP server. http://www304.ibm.com/jct09002c/partnerworld/wps/servlet/ContentHandler?cmsId=VSHA-7BHNFM

Contributors

Peter Kokosielis
DB2 OLTP Performance Benchmarks and Solution Development

Bret Olszewski
DB2 OLTP Performance Benchmarks and Solution Development

Mala Anand
AIX Performance

Tim Vincent
Chief Architect DB2 LUW

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Without limiting the above disclaimers, IBM provides no representations or warranties regarding the accuracy, reliability or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The use of this information or the implementation of any recommendations or techniques herein is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Anyone attempting to adapt these techniques to their own environment do so at their own risk.

This document and the information contained herein may be used solely in connection with the IBM products discussed in this document.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml

Windows is a trademark of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.