Information Center for Linux

Linux tuning considerations for Sybase IQ

Information Center for Linux

Linux tuning considerations for Sybase IQ

Note

Before using this information and the product it supports, read the information in "Notices" on page 13.

First Edition (March 2011)

Contents

Configuring Linux servers for Sybase IQ

Linux configuration considerations that affect performance of Sybase IQ on a Linux system include kernel parameters, volume management, file systems, and I/O tuning.

Introduction

The task of configuring a large Linux data server can be daunting for database and system administrators. Many factors, including hardware used and software configuration, can impact system performance. It is often unclear what the best configuration choices are. Questions that must be addressed when configuring a database system include: How should storage be laid out? Is overall bandwidth more important than latency? Often these two aspects of configuration are not complementary.

This article presents experimental data to show the advantages of one configuration compared to another. Each section describes a set of experiments and compares their results.

The closing section provides a set of best practices based on the results of the configuration experiments. These best practices are starting points that database and system administrators can use when configuring their data servers. You should do rigorous performance testing and validation to determine the best configuration for your workload.

Scope

This article examines some configuration alternatives on Linux that influence performance of Sybase IQ 15.2 running a decision support workload. Various configurations are compared to illustrate the performance benefit of one configuration compared to another. This article will focus on the following configuration areas:

- The kernel parameters that are best suited for Sybase IQ.
- The use of the Linux Logical Volume Manager (LVM) to manage the disks.
- Various file systems that can be use for the database containers.
- The use of direct I/O to bypass the file system page cache.
- Advanced configuration techniques, such as IRQ Management and I/O scheduling algorithms.

Author Names

Tom Lendacky
Steven Pratt

Test environment

These are details of the software, and hardware components used in the configuration experiments.

All experiments were conducted with SUSE Linux Enterprise Server 11 SP1 and Sybase IQ 15.2.

Workloads

The following four decision support workloads were used to drive the transactions against the data server:

Workload 1
This workload has a high I/O throughput rate and many concurrent users in a nearly read-only usage.

Workload 2
This workload consists of adding a write (update), component to workload 1, resulting in a mixed I/O usage.

Workload 3
This workload is similar to Workload 1, but working against a much larger database size.

Workload 4
This workload consists of adding a write (update), component to Workload 3, resulting in a mixed I/O usage.

Hardware

System A
Workloads 1 and 2 were run on System A, an 8-socket, 64-core IBM® Power 780 server with 256 GB of memory and 16 146 GB 15K RPM SFF SAS hard disks. One hard disk was used to hold the SUSE Linux Enterprise Server 11 SP1 operating system. The remaining 15 hard disks were used to run the tests.

Note: In the evaluation of the storage system, SSD storage devices replace the 16 spinning hard disks to compare the performance.

System B

Workloads 3 and 4 were run on System B, an 8-socket, 64-core Power 780 server with 1,024 GB of memory and 24 146 GB 15K RPM SFF SAS hard disks. One hard disk was used to hold the SUSE Linux Enterprise Server 11 SP1 operating system and the remaining 23 were used to run the tests.

Note: In the tests of Simultaneous multi-threading (SMT), I/O schedulers, and device interrupt coalescing, 23 of the spinning hard disks are replaced by 52 73 GB SAS SSD devices for main database storage. Fiber-attached storage was used to hold the data required in order to load the database and to hold the temporary database space.

Experiment results

Test results shown in the examples may include three different measurements of the activity produced by the workloads:
- I/O Operations Per Second
- Throughput
- Performance metric – a measure of relative performance of the workload

Linux settings

The Linux kernel has several parameters that can influence the behavior of a database server. These parameters and their suggested values are listed below. You can define the values of these parameters in the `/etc/sysctl.conf` configuration file to ensure that they are set appropriately when the operating system starts up.

Swappiness

The **`vm.swappiness`** parameter, is a run time tunable option that controls how aggressively the Linux VM should swap memory to disk. The value range is between 0 and 100 where the higher the number, the more aggressively the Linux VM will attempt to swap application data to disk. A high swappiness value can result in the VM swapping application data in favor of discarding file backed page cache. This can lead to excessive delays as the user waits for the application to be swapped back in from disk. A low swappiness value makes the system less aggressive about deciding to swap, making the system less likely to swap out used application memory. This may improve latency when switching between applications. However, a low swappiness value may also result in delays when new applications are invoked or new memory allocation requests are made.

Note: Setting `vm.swappiness=0` does not prevent the VM from swapping. The **`vm.swappiness`** parameter simply controls the aggressiveness of the VM swapping behavior; it does not disable swapping for the VM.

Ulimits

For large Sybase database instances, it may be necessary to increase the amount of memory that a single process is allowed to allocate. You can do this by using the **`ulimit`** command to increase both the maximum memory and maximum virtual memory for the Sybase process. You can either set a specific value, or set these to unlimited, as shown in the following example:

```
ulimit -m unlimited
ulimit -v unlimited
```

If you need to increase the limits allowed by the Sybase user, you can make this change in the `/etc/security/limits.conf` file.

Linux Logical Volume Manager (LVM)

The Linux Logical Volume Manger (LVM) is a valuable tool for the administration of storage subsystems on Linux.

With the LVM, you can combine partitions and disks into high-level groups that provide system administrators simple and powerful ways to allocate and manage storage for applications. Systems connected to large arrays of disks may get significant benefits from the capabilities provided by LVM, such as:

- Combining large numbers of Linux disks/partitions into single higher-level volumes.
- Assignment of sensible names to LVM volumes.
- Ability to move, resize, and assign storage easily when needed.

For these tests, only LVM striped volumes were used. It is assumed that any redundant RAID functions should be performed at the adapter or controller level.

Tested configurations

The LVM tests examined the effects of using different stripe sizes across the devices that were used to back the database for the test workloads. The stripe sizes tested were 128 KB, 256 KB, and 512 KB. The ext3 file system was the file system used for these tests.

LVM test results and conclusions

The following table shows the results of this test for Workload 1. In this test, increasing the stripe size of the logical volume improves the performance of the workload.

Table 1. Comparing LVM Stripe Size for Workload 1

Stripe size	Performance metric	IOPS	Throughput (MP/sec)
128 KB	100	3622	501
256 KB	105	3094	603
512 KB	111	2646	754

The effects of changing the LVM stripe size for the other workloads were similar to the effects on Workload 1. The 512K stripe size provided similar performance benefits compared to the smaller stripe sizes for each case. Similar results were observed when using the ext4 and xfs file systems, and when accessing the logical volume as a raw device.

Choice of file system

The choice of which file system to use for the database is independent of the Linux Logical Volume Manager (LVM) layer considerations.

The choice of which file system to use for the database is independent of the LVM layer considerations. To examine the effects that different file systems have on overall performance, we conducted a series of experiments with some file systems commonly deployed on Linux platforms: ext3, ext4 and xfs. For comparison, we also tested using raw device access to the database.

The ext3, ext4 and xfs file systems are all journaling file systems available in current Linux distributions.

The ext4 file system is a successor to the ext3 file system. Compared to the ext3 file system, the ext4 file system increases some of the size limits and provides some improved performance characteristics.

The xfs file system is renowned for its scalability and effectiveness at handling large files.

Comparison with raw device access allows you to see the performance effects of each file system tested to an environment which doesn't use a file system. Raw access to the database is enabled by creating one or more raw devices for each block device to be used for the database. The block device can be a physical disk (such as `/dev/sdb`) or a logical volume (such as `/dev/mapper/vg_db-lv_db`). To create a raw device, load the raw module and use the **`raw`** command to create and bind a raw device with a block device.

Tested file system configurations

The file system tests each consisted of two test scenarios. One test scenario used a file system that was created using default parameters. The other test scenario used a file system created with a block size of 64 KB. The “largefile” usage type was also specified the ext3 and ext4 file systems. The file systems were mounted without any additional mount options. Raw device access was also tested as a comparison to using a file system. The following tables show the results of using the various file systems to back the database.

Note: Using of a 64 KB block size with the ext4 file system on SUSE Linux Enterprise Server 11 SP1 resulted in a file system error and we were unable to obtain results for that scenario.

File system test results and conclusions

Table 2. Comparing file systems (Workload 1)

Access method	Performance metric	IOPS	Throughput (MB/sec)
Ext3 (defaults)	100	2646	754
Ext3 (large)	104	2238	750
Ext4 (defaults)	102	2218	745
Ext4 (large)	No result	No result	No result
Xfs (defaults)	104	2798	774
Xfs (large)	100	2310	784
Raw	110	1964	388

Table 3. Comparing file systems (Workload 2)

Access method	Performance metric	IOPS	Throughput (MB/sec)
Ext3 (defaults)	100	2744	685
Ext3 (large)	83	1359	467
Ext4 (defaults)	104	2846	782
Ext4 (large)	No result	No result	No result
Xfs (defaults)	100	1782	643
Xfs (large)	76	1155	402
Raw	136	3238	656

Using raw device access to the database provides the best performance for both workloads and is the recommended method of access. Using a large block size with the ext3 and ext4 file system provides some benefit with read-only Workload 1 but all the file systems tested showed a negative performance result using a large blocksize with Workload 2. Should you decide to use a file system, use the default values when creating it.

Note that there is a difference in I/O throughput when measured at the disk level when using a file system compared to using a raw device. Since Sybase by default uses buffered I/O, reads by Sybase will be staged in the Linux Page Cache. This has the effect that all I/O will be done in page size increments. So, for example, when a read I/O crosses a page boundary two pages must be read. The Linux kernel is built with a page size of 64 KB on Power compared to the more typical page size of 4 KB for other architectures. Although this larger page size helps in terms of time to do memory access, it can result in more data being read when an I/O crosses a page boundary. Since access though the raw device is not buffered in the Page Cache, this reading (and possibly writing) of extra data does not occur, resulting in fewer physical I/Os being performed to do the same work when raw devices are used.

File system caching

Database workloads that are characterized by random data access patterns generally do not generally benefit from any additional caching that the file system provides. The reason is that the database buffer pools would have already cached the data. In fact the cost of using the file system cache can significantly degrade performance.

To measure the performance difference, an experiment was conducted to run Workload 2 with and without the file system cache on the ext4 file system. The file system cache can be disabled by setting the IQ_USE_DIRECTIO environment variable equal to 1 or by setting options in the database. The following database options show how the file system is disabled and enabled..

Disabling file system caching

```
OS_FILE_CACHE_BUFFERING=OFF
OS_FILE_CACHE_BUFFERING_TEMPDB=OFF
```

Enabling file system caching

```
OS_FILE_CACHE_BUFFERING=ON
OS_FILE_CACHE_BUFFERING_TEMPDB=ON
```

File system caching results

When the file system cache is disabled, Sybase IQ opens the database files using the direct I/O mode (with O_DIRECT). Opening a file with O_DIRECT will ensure that I/O to the file will by-pass the file system layer. This means that there is no CPU and memory spent on copying the data between the file system cache and the database cache.

Table 4. File system caching comparison

File system cache	Performance metric	IOPS	Throughput (MB/sec)
Enabled	100	2846	782
Disabled	54	748	138

Note: Currently, Sybase IQ has an issue that needs to be addressed when using direct I/O. Until the issue is resolved, you should access the database using the raw interface.

Major Sybase tuning parameters

You can use these parameters to tune various aspects of your Sybase configuration, including use of caching, threads, and processors.

Main buffer cache
This parameter specifies the size of the memory segment that stores all user and static data. This is the location where data that resides in tables is cached for queries and loads. Main buffer cache size is set via `–iqmc <value>` in the configuration file. The value is in the number of Megabytes. For any active database, the default main buffer cache size of 16 MB is too low. To enhance performance, allocate as much memory as possible to the IQ main buffer cache. . A common guideline is to allocate 50% of total memory dedicated to IQ.

Temp buffer cache
This parameter specifies the size of the memory segment that will cache all volatile and temporary data during loads and queries. This is also a workspace for the IQ engine when it needs memory for sorting or join algorithms. This is set via `–iqtc <value>` in the configuration file. The value is in Megabytes. For any active database, the default temp buffer cache size of 8 MB is too low. To enhance performance, allocate as much memory as possible to the IQ temp buffer cache. A common guideline is to allocate 50% of the total memory dedicated to IQ.

Number of threads for sort
This parameter specifies the number of threads to be used in a sort. Use this option for performance analysis and tuning. If you change this option, experiment to find the best value to increase performance, as choosing the wrong value might decrease performance. For large IQ temporary buffer cache size, Sybase recommends setting the `SORT_PHASE1_HELPERS` option between 5 and 10.

Number of processors
This parameter specifies the number of processors that are available to the Sybase IQ engine. While the Sybase IQ engine can automatically determine the number of processors available on the system, you should provide that information using the `–iqnumbercpus` parameter. This is very helpful for cases where other applications may share the same system or where actual number of "physical processorss" is not same as the number of logical processorss visible to IQ due to Simultaneous multithreading (SMT) or Chip-Multithreading (CMT) technology.

Sybase cache allocation

We tested different approaches to dividing memory between the two Sybase cache areas for Workload 1.

Sybase cache allocation test results and conclusions

For our workload we found that the Sybase default recommendation of a 50/50 split in the caches dedicated too much memory to the temp cache, resulting in unused memory while our main Sybase cache could have benefitted from being larger. We tested reallocating various amounts of memory from the temp cache to the main cache until we started to see I/O to the temp space on disk and performance leveled out. The best mix required for each workload will vary, but as can bee seen from our results in the following table, you can improve performance by maximizing the memory usage.

Table 5. Sybase cache split comparison

Main cache (MB)	Temp cache (MB)	Performance metric
114	114	100
125	102	104
142	85	107
154	74	109
171	57	109

Advanced tuning topics

Some advanced tuning considerations involve the storage subsystem, symmetric multithreading (SMT), I/O schedulers, and coalescing device interrupts. These topics require in-depth understanding ofLinux, Sybase IQ, or both, to determine if they are applicable to your environment.

Storage subsystem

As the size of a database grows and more users interact with it, you may need higher throughput and lower latencies from the storage subsystem in order to ensure acceptable performance.

Storage subsystem test results and conclusions

The results table below shows how changing the storage used to back the database can effect the performance of the workload. In this scenario, we ran Workload 2 on System B using System B's initial configuration. We then replaced 16 hard disks with 16 SSDs and ran Workload 2 again using the SSDs to back the database. The use of SSDs provided a significant increase in performance for the workload. The increased throughput and decreased latencies associated with the SSDs allowed the application to perform more work without having to wait as long for the data.

Table 6. Comparing storage subsystems

Disks	Performance metric	Processor utilization	IOPS	Throughput (MB/sec)
SAS HDD	100	19	3149	216
SSD	195	58	6755	443

You have seen how different storage technologies can affect the performance of a workload. You need to determine if the current performance of your workload is acceptable. If the performance of your workload is not good enough, then you may want to investigate how to improve the system. If the system has enough processing capacity, some things to consider are adding more storage to spread out the I/O load on the system, or replacing your current storage with higher performing storage.

SMT settings

Simultaneous multithreading (SMT) allows separate instruction streams, or threads, to run concurrently on the same physical processor, or core.

The IBM POWER7® processor under SUSE Linux Enterprise Server 11 SP1 supports the following SMT modes:

ST In this mode, 1 thread runs on the core

SMT2 In this mode, 2 threads run on the core concurrently

SMT4 In this mode, 4 threads run on the core concurrently

In ST mode, the thread associated with the core is known as the primary thread. In SMT2 mode, the additional thread is known as a secondary thread. In SMT4 mode, the next two additional threads are known as tertiary threads.

In this test, 23 of the spinning hard disks attached to System B are replaced by 52 73 GB SAS SSD devices for main database space.

Determining what SMT mode to run in is dependent on the threading capability of the workload and the overall utilization of the machine. SMT threads are not equivalent in their processing capabilities. The following tables of SMT mode test results show how SMT can be effective as the system becomes highly utilized.

SMT test results

The Table 7shows that enabling SMT2 helps the workload since the added processing capacity of the secondary threads provides the workload extra processing threads of which the workload takes advantage. Enabling SMT4 mode, however, causes some of the execution threads to run on the tertiary threads which are not as powerful, having the effect of reducing the performance of the workload.

Table 7. SMT comparison under low utilization (Workload 4)

SMT mode	Performance metric	Processor utilization
ST	100	68
SMT2	106	44
SMT4	94	23

The Table 8 shows that as the workload uses more of the machine capacity, SMT4 mode has a positive impact on the performance of the workload. In this situation, the workload has enough threads of execution and work to perform that the tertiary threads are utilized to provide extra processing capacity. As more work is added, SMT4 can provide even further performance improvements.

Table 8. SMT comparison under high utilization (Workload 4)

SMT mode	Performance metric	Processor utilization
ST	100	98
SMT2	116	71
SMT4	118	53

SMT conclusions

Selecting an appropriate SMT mode can provide benefits to many types of workloads. You must understand your workload when evaluating SMT. Less utilized systems or workloads that are not highly multi-threaded may not benefit from enabling SMT.

Highly utilized systems or workloads that are highly multi-threaded, as the Sybase IQ product is, may benefit substantially from using SMT.

Additionally, newer versions of the Linux kernel than the version in SUSE Linux Enterprise Server SP1 contain enhancements that enhance SMT behavior for POWER7 processors. These enhancements migrate work from tertiary threads to secondary or primary threads and from secondary threads to primary threads when possible. These enhancements can help to provide better processor utilization and decrease or eliminate, the effect shown in Table 7. These enhancements are not yet available in the SUSE Linux Enterprise 11 product. Currently, these are available in the Red Hat Enterprise Linux 6 product.

I/O schedulers

SUSE Linux Enterprise Server 11 SP1 provides a number of I/O scheduler alternatives to optimize for different I/O usage patterns. You can use the `elevator=` option at boot time to set the scheduler for I/O devices or you can assign a specific I/O scheduler to individual block devices.

Completely Fair Queuing (CFQ) scheduler

The Completely Fair Queuing (CFQ) scheduler is the default I/O scheduler for SUSE Linux Enterprise Server 11 SP1. The CFQ scheduler maintains a scalable per-process I/O queue and attempts to distribute the available I/O bandwidth equally among all I/O requests. The effort balancing of I/O requests has some CPU costs.

Deadline scheduler

The Deadline scheduler is one alternative to the CFQ scheduler. The deadline scheduler uses a deadline algorithm to minimize I/O latency by attempting to guarantee a start time for an I/O request. The scheduler attempts to be fair among multiple I/O requests and to avoid process starvation. This scheduler wills aggressively re-order requests to improve I/O performance.

NOOP scheduler

The NOOP scheduler is another alternative, that can help minimize the costs of CPU utilization of managing the I/O queues. The NOOP scheduler is a simple FIFO queue that uses the minimal amount of CPU/instructions per I/O operation to accomplish the basic merging and sorting functionality to complete the I/O operations.

I/O scheduler test results

In this test, twenty-three of the spinning hard disks attached to System B are replaced by fifty-two 73 GB SAS SSD devices for main database space.

Table 9 compares the database performance with the 3 different I/O schedulers using all of the workloads. We see that the in the mixed read/write workloads (2 and 4) the NOOP scheduler has a negative impact on performance and so should not be used for these workloads. The Deadline I/O scheduler shows a performance benefit when run against the smaller workloads on lesser performing storage while having no impact on the larger workloads with better performing storage.

Note: Values shown in the following table are the Performance metric result.

Table 9. Comparing I/O schedulers

Scheduler	Workload 1	Workload 2	Workload 3	Workload 4
CFQ	100	100	100	100
Deadline	105	105	100	100
NOOP	107	95	100	98

I/O scheduler conclusions

For the workloads used in this test, the Deadline scheduler was a better choice overall than the default CFQ scheduler. It yielded a higher throughput in the smaller workloads and performed equally to the CFQ scheduler in the larger workloads. While these particular workloads performed best with the deadline scheduler, not all workloads benefit equally from the same scheduler. If optimal performance is an requirement, it is worthwhile to investigate the benefits of each I/O scheduler for your workloads.

Coalescing device interrupts

Large database workload configurations typically use large amounts of storage. Using default driver settings for storage adapter device drivers can result in high interrupt rates. Because of frequent storage access by large workloads, the overhead of processing these interrupts can consume CPU resources that impact the performance of certain workloads.

Most storage adapter device drivers can report the completion of multiple I/O operations with a single interrupt. If coalescing is enabled, the driver batches CPU interrupts. If the system is heavily loaded, interrupt coalescing provides better throughput by reducing the number of interrupts and maximizing the amount of commands processed with each interrupt.

In this test, 23 of the spinning hard disks attached to System B are replaced by 52 73 GB SAS SSD devices for main database storage.

In our configuration and with the workloads used, we did not see an excessive number of I/O operations per second and so interrupt coalescing was not necessary. If your workload produces a large number of I/O operations per second you may want to investigate whether interrupt coalescing of the associated storage adapters is appropriate for you.

Best practices summary

These best practices summarize the conclusions of this examination of configurations for Linux with Sybase IQ. Specific areas of interest are the Linux Logical Volume Manager (LVM), file system selection, and Linux tunings.

Linux Logical Volume Management (LVM)

The use of LVM significantly improves the administration and management of database storage in a production environment. When using LVM, it is a best practice to use a larger stripe size than the default.

File system selection

Because of administrative ease of use, distro support, and functionality, the use of the ext4 file system is the best practice for Linux.

Since Sybase utilizes it's own data caching strategy, the use of Linux file system cache won't benefit performance. In fact, the cost of using the Linux file system cache significantly degrades performance. The best practice is to use RAW devices to optimize performance.

Linux tunings

Using the Deadline I/O scheduler is a best practice. The Deadline scheduler provides the best performance under the majority of workloads.

On systems with high I/O rates system and maximum CPU utilization, interrupt coalescing may be a good alternative for relieving system CPU load. The method for coalescing interrupts will vary by device type. Check the documentation provided by your device provider for details.

Appendix. Related information

You can find additional information about the processes and tools described in these best practices.

- Sybase IQ

  http://www.sybase.com/products/datawarehousing/sybaseiq
- SUSE Linux Enterprise 11

  http://www.novell.com/linux/

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.

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.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Dept. LRAS/Bldg. 903
11501 Burnet Road
Austin, TX 78758-3400
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

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

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.

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.

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.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

If you are viewing this information in softcopy, the photographs and color illustrations may not appear.

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 (® and ™), 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

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

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

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

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

Printed in USA