



Linux on the System i Platform Performance Considerations: Hints, Tips, and Strategies

Erwin Earley
System i Technology Center
Linux Center of Competency

October, 2006

Contents

2 Overview
3 Virtual I/O Support
7 Virtual Network Support
10 Processor Considerations
14 Summary

Overview

The System i platform provides a robust, flexible, platform on which to implement Linux-based solutions. Features such as virtual I/O, virtual network, and processor sharing provide the ability to roll-out solutions with little (if any) additional hardware requirements. However, the very features that allow for flexible solutions can also turn the typical resource questions (how much memory does the solution require, how much processor, etc) on it's head. Where traditional implementations on stand-alone servers may require X memory dedicated to the operating system, we may find that the majority of that memory needs to actually be applied to the operating system that is hosting the resources. The classic "that depends" answer in the performance world becomes ever more prevalent in the virtualized world.

This paper presents a number of considerations with regards to implementing Linux-based solutions on the System i platform. This paper is not intended to be the definitive word on performance and will not present specific performance characteristics of a given workload. The classic "your mileage may differ" message certainly holds true in the performance arena. Instead, this paper will pull from the varied customer experiences encountered by the Linux Center of Competency to provide a sense of best practices that we have encountered that you may find useful. As with anything involving performance, it is strongly recommended that you plan on prototyping your solutions and be prepared to make changes as you go forward

The information presented in this paper can be broken down into three broad categories:

- **Virtual I/O support:** One of the key strengths to implementing Linux-based solutions on the System i platform is the ability to have the I/O (disk) hosted by an i5/OS partition and thereby extend the benefits of single-level store to the Linux operating system. We will discuss considerations with regards to performance as well as changes that have occurred in recent versions of i5/OS that you need to be aware of.

Highlights

Virtual I/O Support
Multi-Path I/O

Virtual Network support: Another advantage of implementing Linux-based solutions on the System i platform is the ability to build virtual Ethernet networks (LANs) inside of the managed system (i.e., no physical hardware). We will discuss the importance of the frame sizes of the network adapters as well as when it may be advantageous to implement multiple virtual LANs.

- **Processor considerations:** The System i platform supports the ability to share processors between multiple logical partitions and to have the firmware (hypervisor) balance workload across the available processors through the uncapped partition support. This section will discuss performance considerations of fractional processor support as well as virtual processors.

Virtual I/O Support

With Virtual I/O support, a Linux partition can be implemented with no physical I/O resources – all of the I/O resources are owned by a host partition (i5/OS) and the Linux partition is provided access to those resources through a Virtual I/O driver running in i5/OS.

Virtual I/O can, given the appropriate workload, provide significant performance advantages over the same workload implemented on a desperate server. It all depends on the workload being implemented as well as proper configuration of both the Linux and host (i5/OS) partitions.

Multi-Path I/O

One of the advantages of a hosted Linux solution is that multi-path I/O support can be provided to the underlying storage architecture (i.e., single-level store) without any driver considerations within Linux.

So, what do we mean by multi-path I/O. Put simply, multi-path I/O provides multiple software threads from the virtual I/O driver in i5/OS to the underlying single-level store architecture. The more threads that are reading/writing data from/to the storage architecture the faster I/O throughput is conceptually possible.

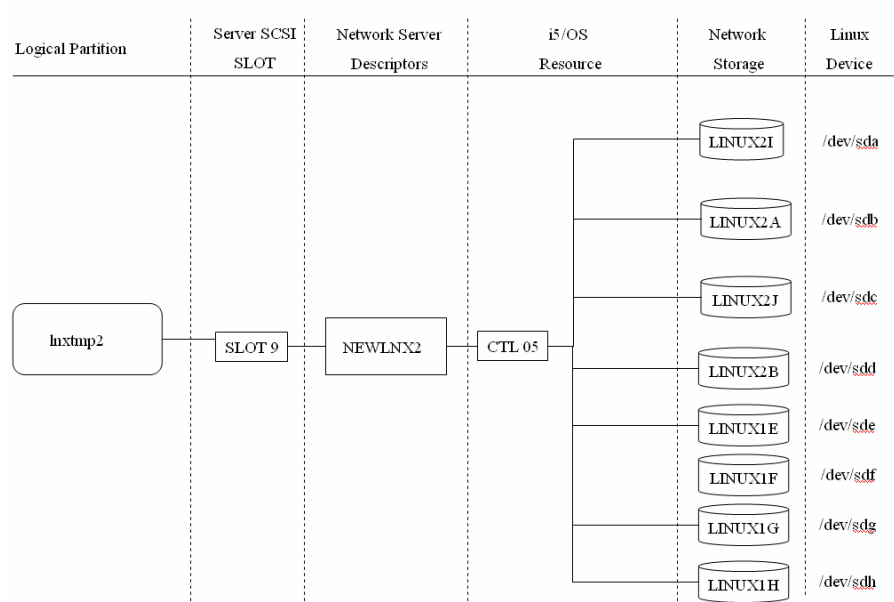
At this point, it is important to make a distinction between the System i5 platform and previous PowerPC iSeries models as well as different versions of i5/OS with regards to multi-path I/O before recommendations can be made:

	Pre-i5 Models	System i5
V5R3M0 – Multi-Path I/O	Yes	No
V5R3M5 – Multi-Path I/O	Yes	Partial
V5R4M0 – Multi Path I/O	Yes	Yes

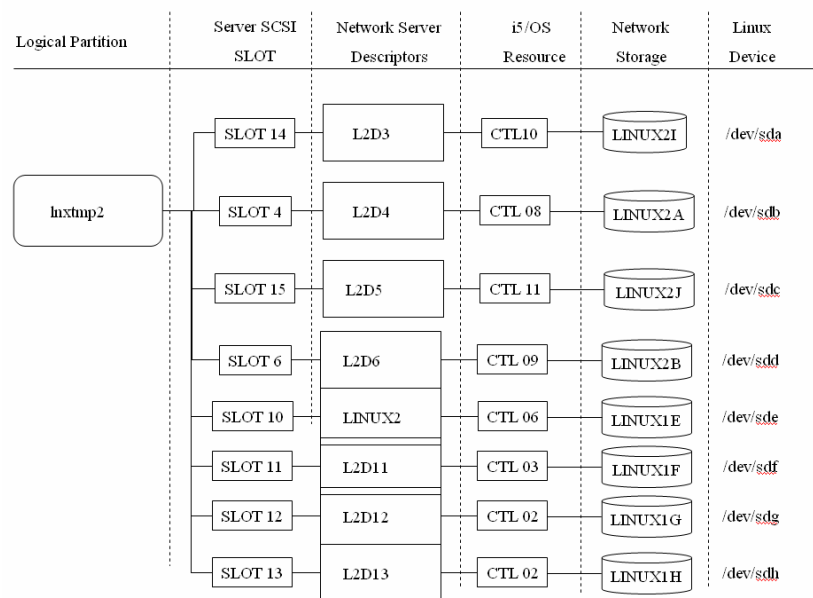
If your Linux solution is implemented on an iSeries model that is a predecessor to the Power5 and Power5+ based i5 models then there is nothing you need to do to obtain multi-path I/O support. However, if your Linux solution is implemented on the System i5 platform then you need to be aware of the support for multi-path I/O. Basically, if you have a heavy I/O workload (e.g., file serving) implemented on a System i5 model and you are currently running at V5R3M0 then you should consider upgrading to V5R4M0 to provide multi-path I/O support. If such an upgrade is not currently possible then you may want to consider changing the configuration of the Linux partition to have multiple Network Server Descriptors with multiple virtual SCSI adapters. This type of configuration will simulate, to a degree, the multi-path I/O provided in V5R4.

NOTE: It is recommended that for systems implementing Linux with heavy I/O requirements the i5/OS partition that is hosting the I/O for Linux should be upgraded to V5R4M0 to achieve the best possible virtual I/O performance.

As an example, imagine the following configuration:



This configuration provides multiple disk drives (storage spaces) to Linux; however, they are going through a single SCSI adapter being hosted by a single Network Server. On a System i5 model if the I/O for this partition was hosted by a V5R3M0 i5/OS partition then the I/O to all of the storage spaces would be single-threaded. The configuration could be changed to:



Highlights

Memory Pool

which provides multiple client/server SCSI pairs with multiple network servers and associated network storage spaces to provide multiple virtual I/O connections.

NOTE: If a configuration is established that uses multiple Network Server Descriptors (NWSDs), you may want to configure all but the first NWSD with the Restrict Device Resources (RSTDDEVRS) value set to *ALL. This parameter is used to restrict access through the Network Server to I/O devices in i5/OS. If this value is not set to *ALL then any CD/DVD/TAPE in i5/OS will have multiple device handles created in the Linux partition since they will be presented to the partition on multiple SCSI chains.

NOTE: Even with a single storage space, V5R3M5 and V5R4M0 hosted I/O provides multi-path I/O.

Memory Pool

Keep in mind that virtual I/O requests from a Linux partition rely on a virtual I/O driver in the i5/OS partition to actually process the request. This is one of the aspects of hosting Linux workloads that can confuse the performance and resource requirements discussions. Heavy I/O workloads can increase the memory requirements on the partition that is **hosting** the I/O (i.e., the i5/OS partition).

What needs to be understood from a configuration viewpoint is that the virtual I/O driver in i5/OS runs out of the memory pools of the operating system. Reviewing the system status (WRKSYSSTS) of the hosting i5/OS partition can provide insight into the performance characteristics of the workload. As an example, the following system status represents an I/O bound Linux workload that is being hosted on an i5/OS partition:

Highlights

Virtual Network Support

```

Work with System Status
                                LINUX825
                                03/07/03 14:24:43
% CPU used . . . . . : 31.3 Auxiliary storage:
% DB capability . . . . : .0 System ASP . . . . . : 457.1 G
Elapsed time . . . . . : 00:06:31 % system ASP used . . : 40.7610
Jobs in system . . . . . : 1346 Total . . . . . : 457.1 G
% perm addresses . . . . : .007 Current unprotect used : 913 M
% temp addresses . . . . : .009 Maximum unprotect . . : 934 M

Type changes (if allowed), press Enter.

System Pool Reserved Max -----DB----- -Non-DB---
Pool Size (M) Size (M) Active Fault Pages Fault Pages
  1 400.64 218.80 +++++ .0 .0 1.6 2373
  2 2869.94 .27 220 .0 .0 .0 .0
  3 4375.91 .00 94 .0 .0 .7 .7
  4 .25 .00 1 .0 .0 .0 .0

Command
====>
F3=Exit F4=Prompt F5=Refresh F9=Retrieve F10=Restart
F11=Display transition data F12=Cancel F24=More keys
    
```

The information of interest with regards to Virtual I/O is the Non-DB pages. The number of pages shown is an indication of the number of I/O requests that are waiting to be processed by the virtual I/O driver in i5/OS. Obviously the larger the number the more requests that are waiting and the slower the performance of the partitions that have hosted I/O will be. Adding more memory to the memory pool can have an immediate and substantial impact on the processing of the I/O requests.

Two items to make note of in this regard:

- The memory allocated can be changed over time by the system if the QPFRADJ system value is set.
- The memory pool used by the virtual I/O component was changed in V5R4. Prior to V5R4 the memory pool used was *MACHINE. In V5R4 the memory pool used is *SYSTEM.

Virtual Network

Virtual Network provides the ability to build virtual Ethernet network segments (LANs) inside of the System i platform without requiring any physical hardware. Virtual Network is a commonly used feature when establishing Linux partitions and can help provide for robust configurations. There are three performance considerations with regards to Virtual Network:

Highlights
Frame Size

Frame size

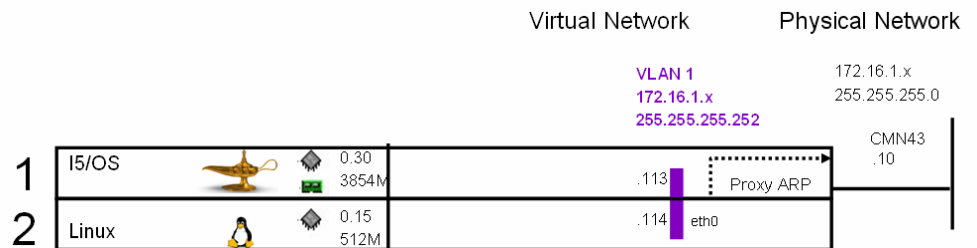
- Use of multiple network segments to split internal and external traffic
- Proxy ARP vs. Network Address Translation (NAT)

Frame Size

The frame size indicates the size (in bytes) of the network packet that can be supported by the network adapter. Typical frame sizes that will be encountered when implementing Linux on the System i platform include:

Network Speed	i5/OS	Linux
100mb	1496	1500
1GB	8996	9000

It is important, from a performance perspective, to ensure that the frame size of all of the adapters matches the lowest common denominator. As an example, suppose you have established a configuration that has a virtual LAN that has connections made available (or transported) across a real/physical network segment:



In this example, the network for the Linux partition (virtual LAN 1) is made available to the external network through a Proxy ARP configuration on the physical interface allocated to the i5/OS partition. Now, let's assume that the physical interface (CMN43) in i5/OS is configured with a frame size of 1496 and that the virtual LAN interfaces are configured with a frame size of 8996/9000. In this case, every packet intended for the virtual LAN that goes across the physical interface will first have to be fragmented by the i5/OS network interface to allow it to fit in the smaller network packet. Configuring the frame sizes of the virtual LAN interfaces to the

Highlights

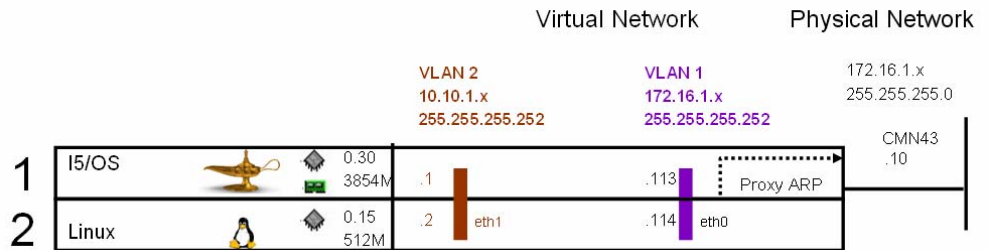
Multiple Network Segments to Separate Internal/External Traffic

Proxy ARP vs. Network Address Translation

smaller frame size (1496/1500) will negate this fragmentation of the network packets and can actually improve performance on heavy network workloads.

Multiple Network Segments to Separate Internal/External Traffic

When a workload is being implemented that has a requirement for substantial network traffic between the Linux and i5/OS partitions (ex: Linux-based Apache web application accessing DB2/400 data through ODBC) increased performance may be obtained by implementation of a separate virtual LAN just for that traffic. Take the example shown earlier – if the Linux partition has to access data from the i5/OS partition it is going to do that across virtual LAN 1 and if the adapters on that virtual LAN have been configured for a slower physical network segment then the partition to partition network communication will also be at that slower speed. By implementing an additional network segment:



The second virtual LAN (VLAN 2) could be configured at the faster 1GB frame sizes (8996/9000) and thereby provide faster data transfer between the two partitions.

Proxy ARP vs. Network Address Translation (NAT)

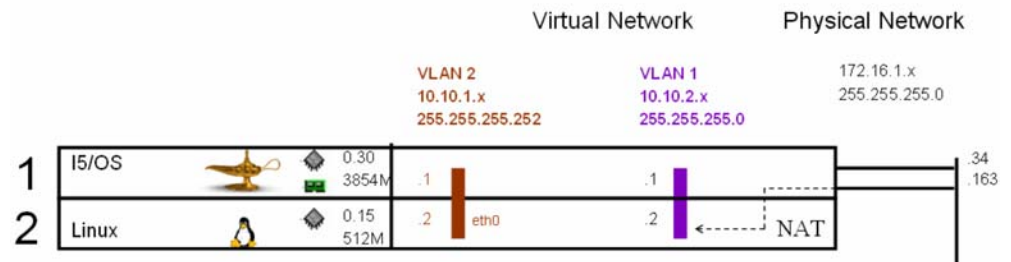
It is possible to make partitions with virtual LAN connections visible to an external network. Two of the more popular approaches are Proxy ARP and Network Address Translation. In both of these methods, an i5/OS partition is allocated connections on both the virtual LAN as well as the external network. With Proxy ARP (see examples shown above), the physical network that the i5/OS partition is on is segmented such that i5/OS essentially becomes the router for a set of addresses in the overall network. The addresses in the smaller segment are then assigned to the partitions on the virtual LAN and the i5/OS network adapter responds for those addresses and then broadcasts the resulting traffic on the virtual LAN.

Highlights

Processor Considerations

The important thing to make note of here is that the i5/OS partition does not process the network traffic in any way, it simply re-broadcasts the traffic on the virtual LAN.

With Network Address Translation (NAT), a private network segment is established on the virtual LAN and mappings are established that take addresses on the real network and translate them to the private addresses:



The addresses on the physical network are allocated as TCP/IP interfaces against the i5/OS adapter. When traffic is seen on one of the addresses it is routed to the i5/OS partition. At that point the network stack in i5/OS goes into the packet and changes the IP addresses in the header to the ip address of the partition on the virtual LAN and then broadcasts the traffic on the virtual LAN. Likewise as traffic is routed out of the virtual LAN, the network stack in i5/OS will again go into the packet and change the IP addresses in the packet header to the external address of the partition. For partitions that will see heavy network traffic, the use of NAT can cause performance degradation since the partition will be reliant on i5/OS to translate the addresses in the packet. The use of Proxy ARP over NAT has been shown to provide better network performance for certain workloads.

Processor Considerations

Most Linux-based workloads implemented on the System i platform leverage support of fractional processor allocation and uncapped partition processing. Fractional processor allocation allows for multiple logical partitions (and their corresponding operating system) to run on a single PowerPC processor. Uncapped partition processing allows the Hypervisor of the Managed System to monitor the performance characteristics of the logical partitions to make additional processor resources available when needed and available.

Highlights

Virtual Processors

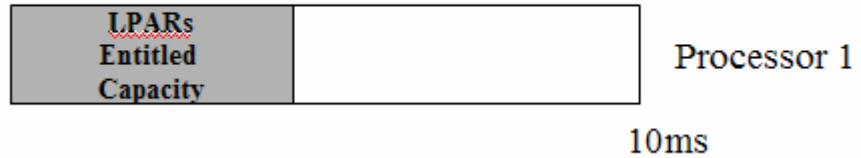
Virtual Processors

Probably the biggest concern, from a performance perspective, to be aware of with regards to processor allocation is the concept of virtual processors. Virtual processors are a representation of a processor thread as presented to the logical partition. It is possible to allocate multiple virtual processors to a logical partition even when less than a full processor has been allocated to that partition. The only rule with regards to the allocation of virtual processors is that each virtual processor has to have at least $1/10^{\text{th}}$ of a physical processor to run against. So, as an example, if 0.40 processor units are allocated to a processor, that processor allocation can be spread across from 1 to 4 virtual processors. The performance consideration here is to understand how that processor allocation is spread across the time slice of the overall system.

The virtual processor setting defines the way that a partition's processor entitlement may be spread concurrently over physical processors. The key here, as will be shown in a moment, is that the spread is concurrent. The number of virtual processors is what the operating system thinks it has for physical processors. The Hypervisor dispatches virtual processors onto physical processors.

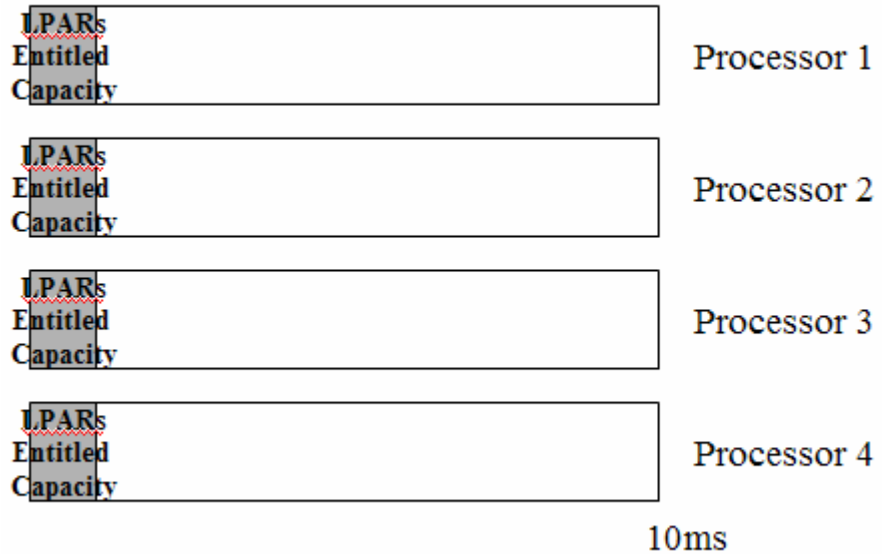
When considering the affect of the number of virtual processors on performance it is important to have at least a basic understanding of the dispatch cycle that the Hypervisor employees for allocation of processor units to the logical partitions. The dispatch cycle is 10 milliseconds (ms) and the processor capacity allocated to a partition is taken from that 10 millisecond timing cycle. As an example, if a partition is allocated .40 processor units then it is entitled 4 milliseconds in the 10 millisecond timing cycle. The number of virtual processors does not change the timing cycle (i.e., it is still 10 milliseconds); however it can have an affect on how quickly the partition can consume it's entitled capacity. If the .40 processor units is allocated across a single virtual processor then the time slice allocated to the partition will take 40% of the timing cycle:

Hypervisor Timing Cycle



The same partition with the same entitled capacity (.40 processor units) that has an allocation of 4 virtual processors would have its processor capacity allocated concurrently across the 4 processors:

Hypervisor Timing Cycle



The impact of spreading the processor allocation across multiple virtual processors can be increased gaps in the processing of the partition that have been known to result in noticeable lags in processing (such as terminal access that temporarily halts and then re-starts). As a general rule, unless there is a specific workload requirement for a large number of processor threads, limiting the number of virtual processors to the smallest number that can accommodate the processor allocation will minimize any processing lag seen by the partition.

Highlights

Uncapped Partitions

Dedicated Processors

Uncapped Partitions

One note to take in this regard is that the number of virtual processors that the processor allocation is spread across will have an impact on the maximum amount of additional processor units that can be made available to an uncapped partition. With uncapped partitions, if a partition has used its allocation of processing units during the current processor dispatch cycle and additional processor resources are required (and available) the Hypervisor can dispatch additional portions of the cycle up to the number of virtual processors allocated to the partition. The following table provides an example of the maximum amount of processor units that could be made available to an uncapped partition:

	2 Virtual Processors	4 Virtual Processors
1.5 Processor Units Allocated	2.0	4.0

Dedicated Processors

In addition to shared processors, a partition can also be configured to use dedicated processors (note: a partition cannot be configured with both shared and dedicated processors). Allocation of a dedicated processor to a partition will provide for the most efficient usage of the processor as the hypervisor will not need to perform task switching for the processor and will be able to ensure memory affinity for the processor. One trade off with use of dedicated processors is that the partition using the dedicated processor must be a capped partition – that is the partition cannot participate in the hypervisor’s load balancing of processor resources across a number of partitions. This means that if the operating system is only using a fraction of the dedicated processor allocated to it, the remaining processor allocation is wasted since the hypervisor cannot allocate it during the time slice to other partitions that may need additional processor resource.

NOTE: Most Linux partitions are configured with processor resource from the shared processor pool.

Highlights

Summary

Summary

A number of factors can affect the overall performance of Linux-based workloads hosted on the System i platform. A good understanding of both the characteristics of the workload as well as the configuration aspects will go a long way to obtaining the best performance for your workload. The following table provides a high-level summary of performance aspects that you should consider for different types of workloads:

Workload	Performance Consideration
Heavy I/O	<ul style="list-style-type: none">• Consider the amount of memory allocated to the memory pool in the i5/OS partition• Consider moving to V5R4M0 or implementing multiple network servers and multiple virtual SCSI adapter pairings
Heavy Network	<ul style="list-style-type: none">• Ensure that the frame size for all adapters is the frame size of the slowest adapter.
Heavy Intra-Partition Network Traffic	<ul style="list-style-type: none">• Consider implementing an additional virtual LAN to be used for intra-partition communication. Ensure that the adapters on this LAN are at gigabit speeds.

Disclaimers and trademarks

IBM Corporation 1994-2006. All rights reserved.

References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

AS/400	eServer	i5/OS
AS/400e	iSeries	OS/400
Blue Gene	IBM	System i5
e-business on demand	IBM (logo)	System i

Linux is a 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.

Information is provided "AS IS" without warranty of any kind.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

All statements regarding IBM future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. Contact your local IBM office or IBM authorized reseller for the full text of the specific Statement of Direction.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.