z/VM Guest Performance

Bill Bitner
IBM Endicott
bitnerb@us.ibm.com
Last Updated: May 21, 2003

Legal Stuff

The information contained in this document has not been submitted to any formal IBM test and is distributed on an "as is" basis without any warranty either express or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the 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. Customers attempting to adapt these techniques to their own environment do so at their own risk.

In this document, any references made to an IBM licensed program are not intended to state or imply that only IBM's licensed program may be used; any functionally equivalent program may be used instead.

Any performance data contained in this document was determined in a controlled environment and, therefore, the results which may be obtained in other operating environments may vary significantly.

Users of this document should verify the applicable data for their specific environments.

It is possible that this material may contain references to, or information about, IBM products (machines and programs), programming, or services that are not announced in your country or not yet announced by IBM. Such references or information should not be construed to mean that IBM intends to announce such IBM products, programming, or services.

Should the speaker start getting too silly, IBM will deny any knowledge of his association with the corporation.

The following are **Trademarks** of the IBM Corporation:
VM/ESA, e-business logo*, HiperSockets, IBM*, IBM logo*,
IBM eServer, RAMAC*, TotalStorage, z/OS, z/VM, zSeries
LINUX is a registered trademark of Linus Torvalds
Penguin (Tux) compliments of Larry Ewing

- I will show various examples of reports and data in this presentation. Many of the reports have been slightly edited to allow them to fit on the page and to highlight the important information.
- The speaker notes you are reading are meant as a supplement to the presentation. I can not guarantee that they will have the same impact or accuracy as seeing the presentation first hand. Please excuse grammar and typos. However, any suggestions or corrections are appreciated.

Overview

- General management of resources
- Processor
- I/O
- Storage and Paging
- Linux® guidelines
- Performance Monitoring

- This presentation will give an overview of areas of VM performance that pertain to managing Linux guests. The presentation will first look at the key resources of a system and discuss how VM allows for the configuration and control of these resources. There will be some discussion on guidelines for Linux guests. Network performance will be discussed as well. Finally, we will end with a look at the various performance and accounting tools available from IBM.

What do you mean by "Performance?"

- ITR = Internal Throughput Rate = a measure of work per CPU second
- ETR = External Throughput Rate = a measure of work per wallclock second
- CPU Utilization = how busy processor is; tied to ITR
- Response Time (Elapsed Time) = how long jobs take; tied to ETR
- Interactive Users vs. Batch Work
- How many phone calls you get

It is critical to be clear about the meaning of "Performance". When I hear people criticize the performance of VM, I am amazed that they consider CPU utilization to be the only performance indicator. In general there are two schools of thought, one that looks at CPU utilization and one that looks at response time. The internal throughput rate, or ITR, is a measure of commands per CPU second. Another way of thinking of this is how many commands could be completed if the processor was running at 100%. ITRs can be used to compare processor performance. When done properly, there should be an implied response time limit as well. External transaction rate (ETR) is a measure of commands complete per wallclock time. You will probably also need to determine the priority of batch versus interactive users.

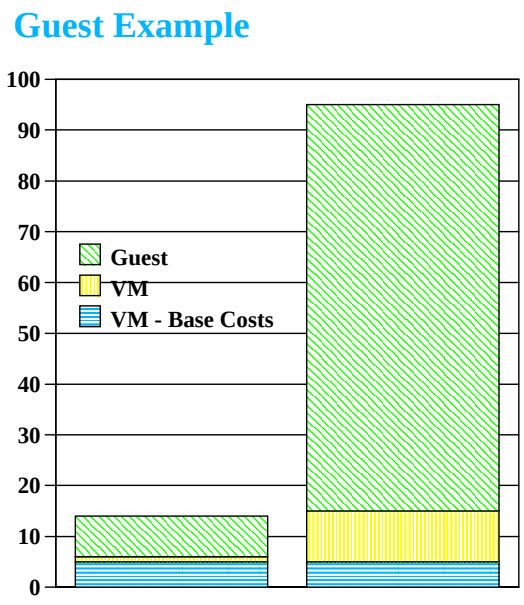
Processor Resources

- Configuration
  - Virtual 1- to 64-way, defined in user directory or via CP command
  - A real processor can be dedicated to a virtual machine
  - Do not recommend use of more virtual processors than there are real
  - Do not recommend mixing shared and dedicated processors
- Control and Limits
  - "Share" setting
  - Absolute or Relative
  - Target minimum and maximum values
  - Maximum values (limit shares) either hard or soft
  - "Share" for virtual machine, divided amongst its virtual processors

- VM has long supported virtual multiprocessor configurations. In fact it supports more virtual processors than the zSeries in 2002 has real processors. You can dedicate a real processor to a virtual machine's processor. In which case, that virtual processor will be the only one dispatched on that real processor. While VM allows mixing dedicated and shared processors, it is not recommended.
- The Share setting is the primary control for processor resources.

Processor Usage by VM

- Base costs and background work
  - Scheduling and dispatching
  - Accounting
  - Monitor
- Costs proportional to Guest requests or requirements of VM



One mistake people make is by trying to determine the overhead of running a guest with VM with a trivial test case. There are some base costs to running VM such as infrastructure, scheduling, dispatching, accounting, and monitor. Many of these are a constant cost. That is, the CPU they require stays the same no matter how busy the system. So if you run a trivial guest workload as a test, you'll see VM being a larger percentage of the total CPU usage. The chart on the right shows this for a trivial VSE workload.

Processor Usage - SIE

- SIE = Start Interpretive Execution
- Used by z/VM™ to run a guest
- Exits from SIE indicate work for VM
- Rate of SIE executions available from most performance monitor products (e.g. VMPRF, RTM, etc.)
- Hardware assists can help avoid SIE exits
- Most common reasons for exiting SIE
  - I/O processing
  - Page fault resolution
  - Instruction simulation
  - Minor time slice expires
  - Loaded wait state

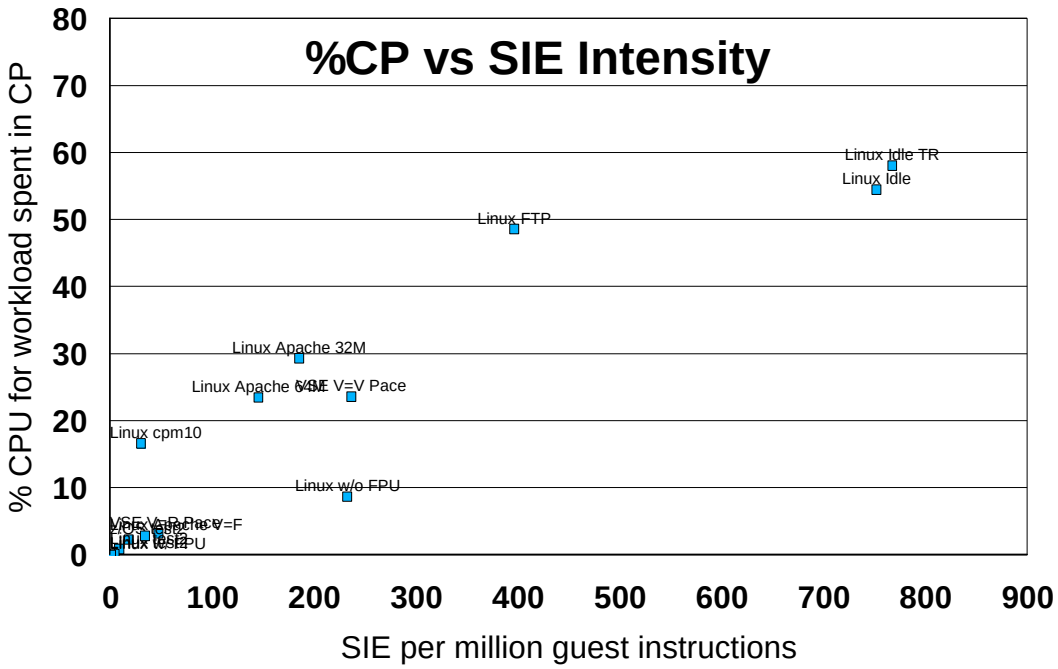
VM uses the SIE (start interpreted execution) instruction to run virtual processors. The overhead and function processing costs in the VM control program are tied to how often we exit from SIE (via intercept or interrupt). This is a case where the hardware assists play a significant role. The rate of SIE executions is available on from most performance monitor products. For example, the VMPRF product reports this on the PRF113 report. Listed are the four most common reasons for exiting SIE. I/O Processing tends to be the most significant. VM gets involved with all V=V I/O and some V=R/F I/O. VM also gets involved for page fault processing for V=V guests. SIE is also exited for certain instruction simulation such as unassisted SIGPs and IUCV. VM will also get control when the minor time slice expires.
The cost of entry and exit from SIE is also processor dependent as some machines have implemented the SIE instruction more efficiently.

Exits from SIE

- Data in memory techniques avoid I/O
- I/O Assist avoids SIE exit to handle:
  - I/O interrupt processing
  - CCW translation from virtual to real addresses
- CCW translation bypass for V=R guest
- Minor time slice: SET SRM DSPSLICE
- Avoid Paging
  - V=R/F
  - Reserved pages for V=V
  - Sufficient storage

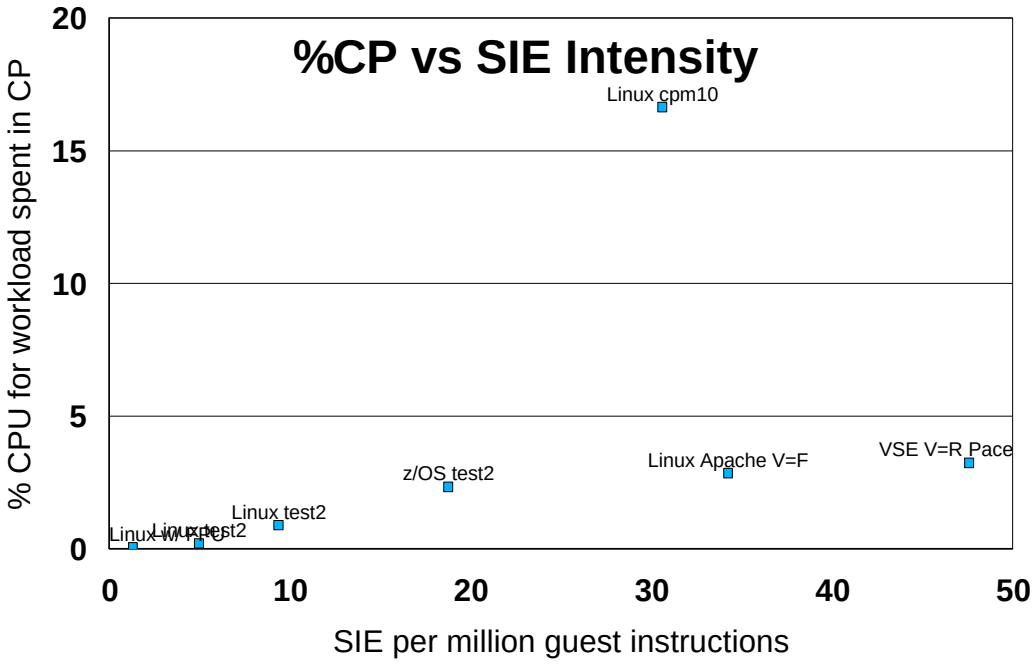
Minimizing I/O in the guest, by using larger buffers or data-in-memory techniques, can lower VM/ESA overhead. Page fault overhead can be minimized by adding storage or reserving pages as appropriate. The scheduling overhead can be adjusted with the SET SRM DSPSLICE command. However, caution should be used when adjusting the minor time slice. While increasing it may lower the VM/ESA overhead, it also lowers the ability of VM scheduling to adjust to system changes. We have seen scenarios where the ITR improves, but ETR gets worse when increasing the minor time slice. Note also that dedicated virtual machines get a 500 millisecond dispatch time slice.

VM Overhead Cloud Chart



- This chart shows how the overhead of VM 'depends' on various factors, such as the frequency of SIE exits and the type of SIE exits.
- The x-axis shows the rate of SIE instructions for given work (million instructions executed in guest/application code), while the y-axis shows the percent of the total processor time used by the workload that was consumed in VM control program code. Here we plot various points representing different workloads.
- If all SIE exits were the same cost, then the points would form a straight line. The bunch of points clustered near the origin is expanded on the next page. The key points are that the overhead various, though the absolute values may be very small (e.g. for idle Linux without timer patch, maybe 0.3% of a processor). Some of these points need to be re-plotted, as

VM Overhead Cloud Chart



- Linux cpm10 is a copy file operation on a 10MB file done with an early Linux using diagnose x'250'.
- Test2 is a kernel like benchmark.

I/O Resources

- Configuration
  - Dedicated devices (Tape Drives, DASD, Network devices)
  - Virtualized devices (minidisks, crypto)
  - Simulated devices (Guest LAN, v-disks)
  - Define or attach dynamically
- Control and Limits
  - Indirect control through "share" setting
  - Real devices can be throttled at device level
  - Priority can be set for virtual machine
    - CP uses to effect queue placement for DASD devices
    - HW uses to effect priority in channel usage
  - Minidisk Cache fair share limits can be turned off for virtual machine

I/O Considerations

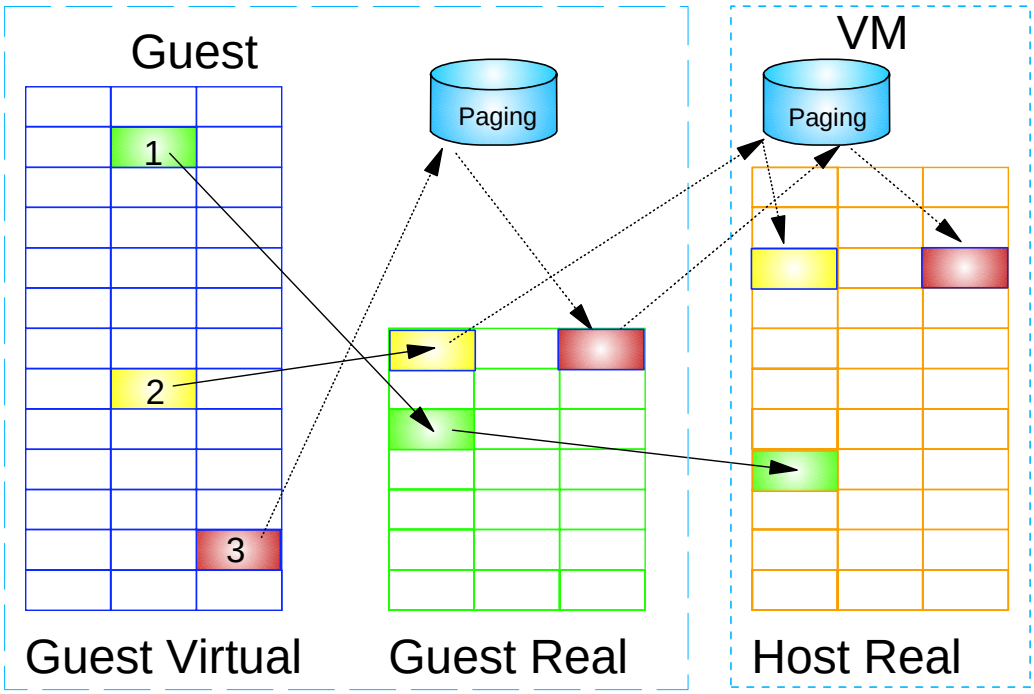
- Traditional benefit of V=R/F guests and I/O Assist usually does not apply to Linux guests
  - Integrated Facility for Linux (IFL) processors most often used for Linux
  - IFL requires LPAR which results in loss of I/O Assist
- Dedicated I/O is not eligible for Minidisk Cache (MDC)
- Fullpack minidisks defined with DEVNO instead of VOLSER are not eligible for MDC
- MDC read performance is as good as VM vdisk performance
- Both VM vdisks and MDC require sufficient storage

Traditionally, we would talk about I/O Assist for guests to provide the best performance from an ITR view. However, I/O Assist rarely applies in a Linux environment. Most customers use IFL processors for Linux. IFL requires LPAR and I/O Assist is not possible in an LPAR environment. Also, even if running VM in basic mode, the constraint of the number and size of the V=R/F guests usually makes it unfeasible for Linux workoads.
There are several other considerations to keep in mind. Dedicated DASD devices are not eligible for minidisk cache. They must be minidisks, either fullpack or partial pack. If fullpack, then they should be defined via VOLSER if you desire MDC. Both virtual disk in storage and minidisk cache require sufficient storage to provide good performance.

Storage Resources

- Configuration
  - Defined in user directory or via CP command
  - Can define storage with gaps (useful for testing)
  - Can attach expanded storage to virtual machine
  - Machine can be V=V, V=F, or V=R
- Control and Limits
  - Scheduler helps control over committing storage and paging resources
  - Virtual Machines that do not "fit" criteria placed in eligible list
  - Virtual Machine can be made exempt from eligible list via QUICKDSP
  - Can "reserve" or "lock" pages for V=V guests
    - Reserve a number of pages to influence storage management page steal algorithms (recommended approach)
    - Lock specific pages (less flexible and forces page below 2GB)

Paging Considerations



- It can be confusing to discuss paging of guests that support virtual storage when there is confusion over terms and without pictures. This picture shows storage as it relates to the VM control program point of view. Physical storage or central storage, is labeled host real in the picture above. Guest real is the storage that Linux believes is real even though it is virtual to VM. Guest virtual would be virtual storage from Linux's view point. A Linux application referencing data or instructions might be in one of the three numbered pages in Guest Virtual storage. In Page 1, the page happens to be in guest real, and that guest real page also is resident in Host Real. Therefore, no paging is required at all. For Guest Virtual Page 2, we see it is in Guest Real, but not host real. Therefore, a page fault would occur which VM would need to process. In the case of Guest Virtual Page 3, we see it is not in Guest Real storage. This

Paging Considerations

- For V=V guests the potential exists for "Double Paging"
- No VM paging for V=R/F
- The closer the amount of virtual storage used by Linux is to the defined storage for the virtual machine, the lower the Linux swapping.
  - However, oversizing the virtual machine size for Linux guests has other negative effects
- PAGEX and Asynchronous Page Fault used where appropriate
- VM can use expanded storage for high speed paging device
- There can be an advantage to defining some processor memory as expanded storage
  - See www.vm.**ibm.com**/perf/tips/storconf.html

One should try to avoid scenarios of double paging. This would happen if a guest page is not in the guest space and is paged in to a guest real address that VM in turn needs to page in. VM paging is avoided completely for V=R or V=F machines. The closer the virtual machine size is to the total virtual storage required by the guest virtual machine, the less guest paging (or swapping in the case of Linux) will occur. Linux at the 2.4 kernel level will use the PAGEX ON or asynchronous page fault where appropriate.

V=R/F/V Considerations

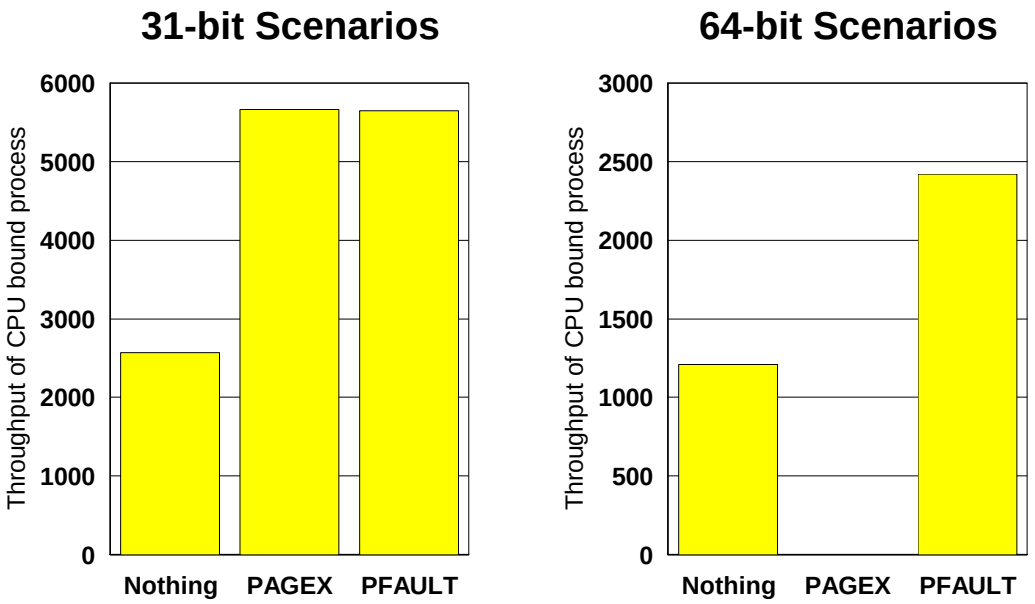
- V=R/F potential I/O assist benefit (saves CPU)
- V=F avoids overhead of recovering V=R
- 1 V=R + 5 V=F or 6 V=F
- V=V avoids dedicating storage
- V=R defaults to dedicating processors
- Running z/VM in an LPAR -
  - No V=F, only V=R, but without I/O Assist
  - Often better to use V=V and reserve pages

V=R performance can be lower than V=F performance. Extra processing is required for the recoverability part of V=R support. Preferred guests (V=R/F) on a native VM avoid VM paging and provide savings with hardware assisted I/O. The total number of preferred guests is still six, even though LPAR can provide more partitions on some processors. If you are running z/VM in an LPAR, you need to realize that it changes the characteristics. Both LPAR and VM/ESA use SIE. On older processors (3090E and older), the assists were not available to run SIE on top of SIE. Running this configuration would be very costly. All the current processors have the required interpreted SIE assist for running VM on LPAR. However, with VM on an LPAR, only the V=R machine is possible and there is no I/O Assist. In this scenario, you may be better off running the guest as a V=V machine with CP reserved pages. And as was discussed earlier, IFL environments require LPAR.

Asynchronous Page Fault Facility

- Ordinarily, page faults serialize the virtual machine. This can be a throughput and response time problem for guest systems
- Enhancements designed for Linux
- PFAULT macro
  - Accepts 64-bit inputs
  - Provides 64-bit PSW masks
- Diagnose x'258'
- Older PAGEX interface limited to 31-bit
- z/VM 4.2.0
- Linux 2.4 Kernel required

Page Fault Tests



- The graphs here show a Linux system running two applications: one that is storage-bound and one that is cpu-bound. The graphs show the throughput of the cpu-bound application. In the 'nothing' case, it is held back because of waiting on page faults for the storage-bound application. Both sets of measurements were run on z/VM 64-bit mode, with the Linux guest run in 31-bit on left and 64-bit on the right. The measurements between the two graphs were run slightly different, so you cannot compare 31-bit to 64-bit with these measurements.

Virtual MP Support

- Define additional processors dynamically
  - Directory include MACHINE ESA 2
  - CP DEFINE CPU vcpu_addr
- Or put everything in the directory
  - CPU 00 NODEDICATE
  - CPU 01 NODEDICATE
- Detaching vCPU resets virtual machine
- For testing: more virtual than real processors

There are two approaches to creating a virtual MP machine. You can define the virtual processors in the directory so they are available when the virtual machine logs on. Or you can set up the directory so that you can use the DEFINE CPU command to add virtual processors dynamically. Note that detaching a virtual processor resets the virtual machine. Do not define extra virtual processors unless you are going to use them. Defined, but unused, virtual processors can cause performance problems.

Virtual MP Support

- CP commands of interest
  - QUERY VIRTUAL CPUS
  - CPU vcpu_addr cmd_line
  - DEDICATE and UNDEDICATE
- Share setting is for virtual machine, divided amongst all virtual processors
- Mixing dedicated and shared processors is not recommended
- Defined but inactive vCPU (stopped state) makes guest ineligible for I/O assist
- Dedicated processor appears 100% busy on various VM performance reports

This is a list of CP commands that can be useful when using virtual MP machines. The QUERY VIRTUAL CPUS command shows you how may virtual processors you have and their addresses. When setting traces or issuing other commands that affect a virtual processor, you will want to use the CPU command to direct the command at a particular processor or to all virtual processors with the ALL option. Output from CP commands is prefixed with the virtual processor address. The DEDICATE and UNDEDICATE commands can be used to control the dedication of real processors to virtual machines, which can be helpful in virtual UP environments as well.

VM Data in Memory Techniques

- VM Virtual disk in storage
  - Volatile FBA minidisk
  - Private or shareable
  - Can be used for the Linux swap file
- Minidisk cache
  - Undedicated 3380, 3390, 9345, RAMAC®, and Shark
  - SSCH and Diagnose I/O
  - Read-once data generally does not benefit
  - Record level MDC option applies only to diagnose I/O

On this foil, we will briefly describe two data-in-memory techniques used by z/VM. They are virtual disks in storage and minidisk cache. . The VM virtual disk in storage feature allows for volatile FBA minidisks that can be defined as shareable or private. This are backed by a VM system utility space. It is most commonly used as the swap disk for Linux. Minidisk cache was enhanced in VM/ESA 1.2.2 to be more flexible, allow more types of data, and more types of I/O. The enhancements included a series of CP commands to enable/disable the cache, flush the cache; the ability to use real storage as the cache; the eligibility of almost any type of data; and eligibility of SSCH I/Os. The minidisk cache is track oriented. One should not suppose that MDC will benefit read-once data, particularly if the reading application has been highly tuned.

Linux Guest Guidelines

- Why does my idle Linux consume Processor resources?
  - Timer pops
- Is the number and size of guests important?
  - Yes! It is virtual storage, but it isn't magic. It has to reside somewhere when Linux guest is running.
- How big should my Linux guest be?
  - Not bigger than you need
- Where should Linux swap?
  - Multiple choices: XPRAM, Mdisk, Tdisk, Vdisk
- Should I set QUICKDSP ON for my Linux Guest?
  - Production vs. Test vs. Development machines
- See the following URL for other information: www.vm.**ibm.com**/perf/tips/linuxper.html

- See the URL listed at the bottom for more details on these items. There can be a lot of discussion on each of these items and many tend to be related to one another.
- An idle Linux machine tends to never look idle to VM because of the various house keeping tasks that occur: timer pops, network polling, etc.
- VM does some remarkable things, but it is not magic. Even though guest storage is virtual, it still needs to reside in real memory when in use.
- Linux tends to use all the storage you give it, so do not give it too much.
- If you are not going to do any significant swapping, then Vdisks are very convenient. However, if you are going to do significant swapping then use minidisks or tdisks.
- QUICKDSP ON for production guests to avoid unwanted stays

VM and HiperSockets

- Synchronous data movement between LPARs and virtual servers within an IBM @server™ zSeries™ server
  - Provides up to 4 "internal LANs". HiperSockets™ accessible by all LPARs and virtual servers
  - Up to 1024 devices (TCP/IP stacks) across all 4 HiperSockets and up to 4000 IP addresses
  - Similar to cross-address-space memory move using memory bus
- Extends OSA-Express QDIO support
  - LAN media and IP layer functionality (internal QDIO = iQDIO)
  - Enhanced Signal Adapter (SIGA) instruction
    - New "thin interrupt" without use of System Assist Processor
  - Optional dispatcher polling mechanism
- HiperSockets Hardware I/O Configuration with new CHPID type=IQD
  - Controlled like a regular CHPID
  - Each CHPID has configurable Maximum Frame Size
- Works with both standard and IFL CPs
- Highly secure connections
- Both 31-bit and 64-bit operating systems supported
- Pre-req: IBM @server zSeries 900 LIC Update

- HiperSockets was a new hardware element added to the 2064, and later 2066, processors. HiperSockets allow for synchronous data movement between LPARs or virtual servers with a zSeries processors. The API used with HiperSockets is an extension of the OSA QDIO and is referred to as iQDIO. There is a new channel type associated with it.

VM and HiperSockets

- VM Support for real HiperSockets
  - VM TCP/IP Stack can use
  - Guests with support (z/OS™ and Linux)
- Can be used to communicate between guests on same VM system
- Guest LAN is a simulated HiperSockets within a VM system
  - Available on all machines that z/VM 4.2.0 supports
- Enabled with VM62938 and PQ51738
  - Also recommend VM63034
- Rolled into base of z/VM 4.3.0
- z/VM 4.3.0 also added support for Guest LANs through QDIO API

- VM added support for HiperSockets usage from guest machines and also the VM TCP/IP stack. This allows for communication to other LPARs or other virtual machines through HiperSockets.
- In addition to the real HiperSockets support, VM introduced Guest LANs which are simulated HiperSockets. This are supported not just on zSeries processors, but any processor that z/VM 4.2.0 are supported on.
- It is worthwhile checking for required and recommended service.
- This service was rolled into the base of z/VM 4.3.0 along with the support in that release for using Guest LANs with the QDIO API (previously only iQDIO)

Networking Choices

- Lots of variations
- Routing Virtual Machine
  - Avoid hardware limitations
  - VM Memory limitations
  - Does require additional processing resources
- Virtual Switch (new in z/VM 4.4.0)
- Both Linux and VM stacks continue to improve
- MTU size is major factor with some workloads

HiperSockets and Guest LAN - Final Thoughts

- Guest LAN and HiperSockets are improvements over QDIO GbE
- Guest LAN
  - + Configuration limits
  - + Storage Requirements
  - - Does not work between LPARs
- More efficient as load increases
- Guest LAN supports iQDIO and QDIO, slightly more processor overhead with QDIO due to asynchronous nature of QDIO
- IUCV and vCTC become less exciting with the addition of Guest LAN

- A few final thoughts are captured here to help look at the decisions you might make in network configurations. Both Guest LAN and HiperSockets are improvements over QDIO GbE.
- Since real HiperSockets are tied to special channels and there is a limit to these channels. HiperSockets also require significant storage locked. Guest LAN does not have these limitations.
- HiperSockets and Guest LAN tend to become more efficient as the load increases.
- Lastly, it is worth noting the IUCV and vCTC will probably not see many enhancements in the future. Guest LAN is more strategic for inter-virtual machine communication.

Summary

- Many features to be exploited
- The answer is "It depends. With Linux, it depends even more"
- Optimum configuration will depend on
  - What you mean by the term performance
  - What resources you have available
- See VM home page for additional information:
  - www.vm.**ibm**.com
  - www.vm.**ibm.com**/perf/
  - www.vm.**ibm.com**/perf/tips/

I hope this presentation helped generate questions as to how and where VM can be used to help with your guest performance. I'm sure not all your questions were answered here. There is a great deal of information available in manuals, listservers, IBMLINK, and the VM or VSE home pages. Check it out.

I welcome your comments and suggestions on this presentation.