

# *Linux on z/VM Configuration Guidelines*

Barton@VelocitySoftware.com  
HTTP://VelocitySoftware.com

June, 2009  
Condensed

“If you can’t Measure it,  
I am Just Not Interested <sup>TM</sup>”



## Configuring z/VM for Linux on zSeries

- Must configure z/VM – many defaults incorrect
- Linux must be configured for shared resource environment
- Many actions not intuitive
- There are “Best Practices” that have been tested and work

## Infrastructure unknowns for “new” installations

- How to manage performance / capacity planning?
- What are the limits of a configuration and how to measure
- How to share resources to reduce ROI

## Measurement and Tuning for z/VM IS Required

- Start with Proper Configurations



# Configuration Options

## General Storage Options

### Linux Options

- Storage Sizes
- Swapping for Linux
- Linux virtual processors

### z/VM Configuration

- Network, Virtual Switch, I/O, FTP Topics
- MDC
- Paging for z/VM
- DASD/Cache/Channels
- z/VM System parameters
- Expanded Storage

### Infrastructure

- Linux infrastructure – monitoring availability and performance



# General Storage Requirements

## Configuration requirements changing and different :

- Small Infrastructure Servers – “Small” Systems (2000)
  - DNS, Apache, Samba
  - Low I/O rate
  - Real storage less than 2gb
  - Virtual servers sized 64mb to 256mb
- Medium (31bit) Application Servers - Small to Large Systems (2003)
  - Websphere, Domino, Oracle
  - z/VM Real storage greater than 2GB
  - High I/O rate potential
  - Typical 512MB to 2GB
- Large (64bit) Application Servers - Large Systems (2007)
  - Oracle, SAP
  - z/VM Real storage greater than 10GB
  - High I/O rate potential
  - Virtual Servers Typical 512MB to 16GB



# Technology Update and Comparison

From a “performance perspective”

2000 (compare the 1 engine Cessna to a ferry)

- (Ferry carries lots of people reliably, on schedule, but slowly and for short distances)
- G5 processors, 600-800 Mhz
- Escon channels (100 mbit)
- TCPIP required software interrupt
- z/VM VERY storage constrained (2GB really)

2009 (compare the “single purpose” Cessna to a 747)

(747 carries lots of people reliably, VERY long distance, on schedule, and has lots of redundancy)

- Z10 (4+ Ghz, 64 “cores”)
- Ficon Express (4 Gbit)
- TCPIP/Network QDIO (hardware interrupt)
- z/VM 5,4 with no storage limitations



# Storage and Paging Considerations

## z/VM Paging

- Over-committing storage improves costs per server
- Over commitment of storage causes paging
- **Over commitment of storage reduces cost**
- Paging is common **(manageable)** performance problem

## Storage requirements of Linux very high

- Linux designed for dedicated storage, references all storage
- **Linux is LRU, competing with VM's reference pattern**
- High percent of referenced pages – what can z/VM page out?

## Linux Swapping

- Swapping result of over commitment of Linux storage
- Swapping to vdisk very fast, uses storage when it happens
- Swapping to dasd very slow, always noticeable



## Linux Cache

- Linux avoids I/O by using cache, cache gigabytes of data if allowed
- Oracle SGA MUST fit in page cache
- Swap historically was slow SCSI device

## Reduce size of Linux Virtual Machine MAJOR Knob.

- Reducing virtual machine size reduces caching of old data
- Define virtual disk for swap
- Virtual Disk paged out when not in use - Unlike “Real” memory
- Experiment with Linux server swapped 40,000 per second.

## Linux qdrop is biggest problem today

- 100 timer pops per second was 1<sup>st</sup> problem, fixed (hertz timer)
- **Current release of IBM JDK (WAS) polls every 10 ms**



# Tailoring Linux Storage

Linux data shows  
Real storage  
Swap storage  
“cache”

Some Swapping is “good”

If not swapping,  
reduce vm size  
Use CMM to reduce

Report: ESAUCD2		LINUX UCD Memory Analysis Report								TEST MAP	
Node/ Time/ Date	<-----Storage Sizes (in MegaBytes)----->										
	<--Real Storage-->			<-----SWAP Storage----->			Total	<--Storage in Use-->			
	Total	Avail	Used	Total	Avail	Used	MIN	Avail	Shared	Buffer	Cache
20:58:35											
LNXdap	122.4	4.6	117.8	511.4	501.2	10.2	15.6	505.8	0	17.1	49.6
LNXnfs	193.1	4.6	188.5	511.4	511.0	0.4	15.6	515.6	0	29.6	55.7
LNXzero	122.8	3.4	119.3	444.2	436.1	8.1	15.6	439.5	0	19.6	43.2
LNXdna2	499.6	182.9	316.8	317.3	317.3	0	15.6	500.1	0	25.7	164.5
LNXdna3	499.6	25.0	474.6	511.4	511.4	0	15.6	536.4	0	38.7	315.0
LN Xtux	502.2	6.7	495.5	571.1	571.1	0	15.6	577.8	0	108.9	180.8
LN XPRbt0	499.6	22.9	476.7	511.4	511.4	0	15.6	534.3	0	94.6	241.5
LN XPRbt1	499.6	27.6	472.0	511.4	511.4	0	15.6	539.0	0	25.2	299.9
LN XPRbt2	287.4	18.5	268.9	511.4	511.4	0	15.6	529.9	0	30.7	106.3
LN XPRci1	499.6	10.1	489.5	511.4	358.6	152.9	15.6	368.7	0	20.6	269.4
LN XPRci2	499.6	21.3	478.4	511.4	449.8	61.7	15.6	471.0	0	17.7	164.5
LN XPRot1	499.6	8.5	491.1	511.4	394.6	116.8	15.6	403.1	0	39.0	164.5
LN XPRot3	704.0	12.1	691.8	511.4	511.4	0	15.6	523.6	0	28.9	239.9
LN XPRot5	499.6	4.0	495.6	511.4	451.3	60.1	15.6	455.3	0	4.4	426.5
LN XPRrg1	499.6	15.1	484.5	511.4	431.8	79.6	15.6	446.9	0	22.1	104.1
LN XPRrg2	499.6	24.6	475.0	511.4	465.3	46.1	15.6	489.9	0	23.1	127.1
LN XPRmk1	499.6	24.0	475.6	511.4	453.2	58.2	15.6	477.3	0	8.5	156.1
LN XPRmk2	499.6	27.2	472.4	511.4	465.2	46.3	15.6	492.4	0	13.6	136.3
LN XPRmx1	499.6	36.0	463.6	511.4	465.4	46.0	15.6	501.4	0	14.2	141.6
LN XPRic1	499.6	31.6	468.0	511.4	462.5	48.9	15.6	494.1	0	20.6	184.6
LN XPRic5	248.1	5.6	242.5	511.4	437.8	73.6	15.6	443.4	0	2.4	201.0
LN XPRic6	248.1	5.7	242.4	511.4	467.7	43.7	15.6	473.5	0	2.3	194.8
LN XPRic2	499.6	27.6	472.0	511.4	511.4	0	15.6	539.0	0	38.7	213.9
LN XPRiv1	499.6	16.0	483.6	511.4	316.7	194.7	15.6	332.7	0	2.8	281.7
LN XPRmx1	499.6	29.7	470.0	511.4	511.4	0	15.6	541.1	0	15.3	151.6
LN XPRmx2	499.6	27.8	471.8	511.4	459.2	52.3	15.6	487.0	0	14.6	143.1
LN XPRbq1	499.6	11.6	488.1	511.4	453.2	58.2	15.6	464.8	0	16.3	92.5
LN XPRsd1	499.6	23.7	475.9	1023	1023	0	15.6	1047	0	3.9	411.0
LN XPRkf1	499.6	161.8	337.8	511.4	511.4	0	15.6	673.2	0	22.7	178.9
LN XPRot2	751.1	13.5	737.6	511.4	511.4	0	15.6	524.9	0	47.3	235.8
LN XPRa8	502.3	21.5	480.8	507.7	507.7	0	15.6	529.2	0	18.9	92.3



# Linux Swapping

## Reducing virtual storage size may cause swap

- Linux does not swap until out of storage

## Swapping to disk

- VERY VERY SLOW
- Other platforms increase storage size because disk is slow
- **Swap to disk if you want to penalize a server**
- Max swap rate maybe 200 on a very good day

## Linux Swapping to Vdisk

- Not a performance degradation
- 40,000 / second is FAST (on 1Ghz processor)

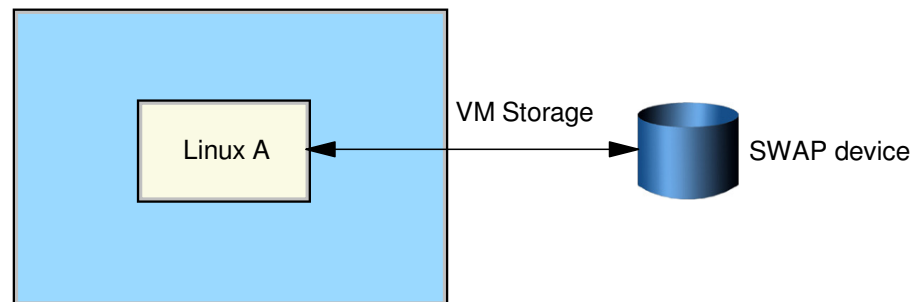


# VM Storage Overview, Paging Hierarchy

## Swap Guideline:

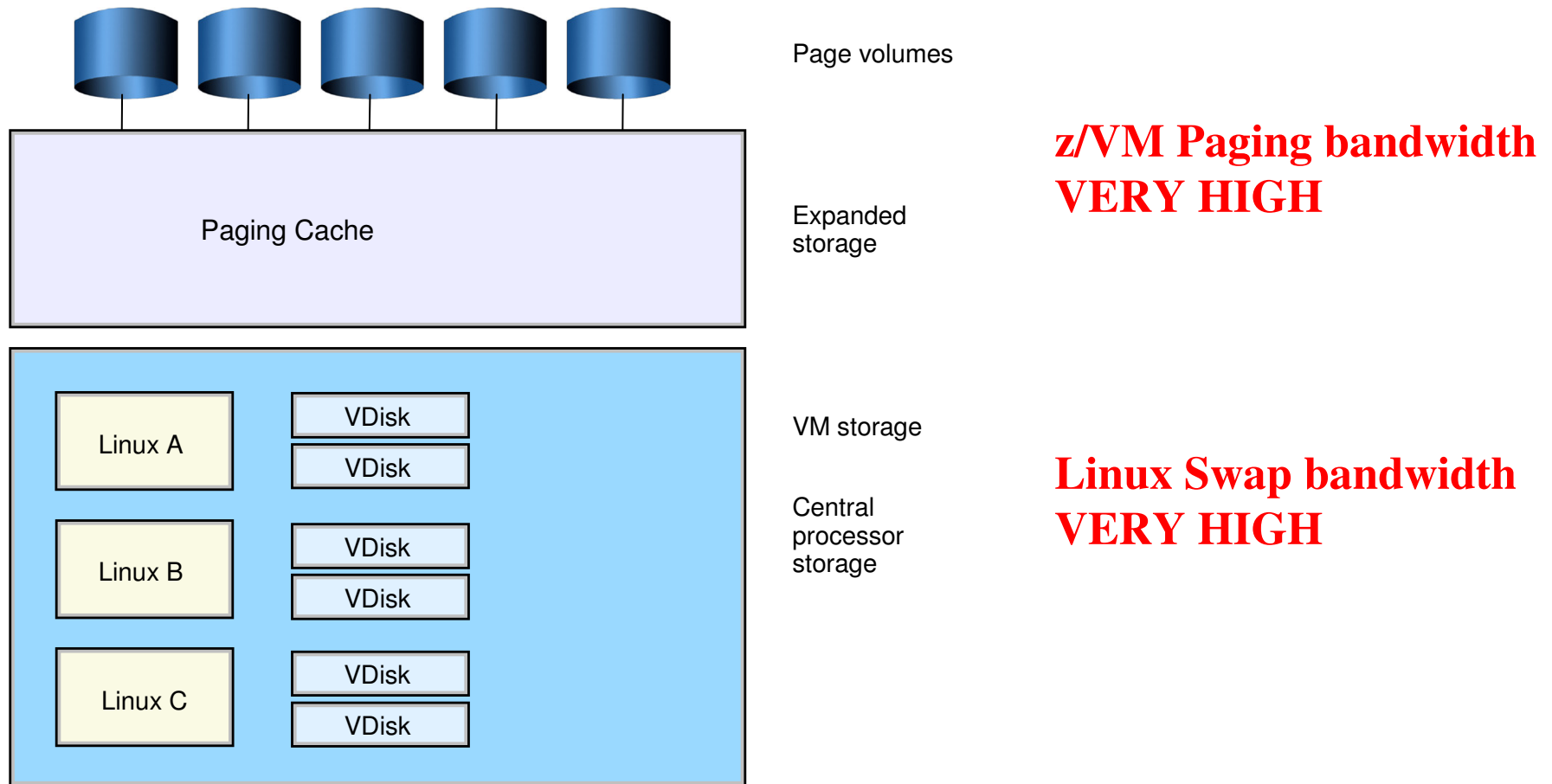
- Define 2 virtual disks, prioritized swap
- Use DIAG driver instead of FBA - Reduces I/O by factor of 8

Linux storage/SWAP



# z/VM Paging Hierarchy

z/VM paging bandwidth very high, multi-level



# Linux Storage Case Study

## First case study:

- Process took hours, system paged significantly
- Reduced size of Linux Virtual Machine, 128mb to 24mb
- Defined 100MB Swap disk
- Linux reduces storage requirement
- Process took minutes

## Virtual Disk paged out when not in use

- This works!!! Paging greatly reduced, Linux performance greatly improved!!!

**This research critical to using Collaborative Memory Mgmt (CMM)**



# LINUX Swapping to VDISK

## Change 128MB Server to 24MB with 100MB Swap Reduction of Overall Storage Requirements of 100MB

- Unused VDISK is paged out

```
Screen: ESAVDSK  Velocity Software, Inc.          ESAMON V2.2  03/15 12:14-
<--pages-->  DASD      X-
Resi- Lock-   Page Store
dent  ed   Slots  Blks
-----
12:15:01 LINUX001 VDISK$LINUX001$0202$0009      36      0      50      0
12:16:01 LINUX001 VDISK$LINUX001$0202$0009      36      0      50      0
12:17:01 LINUX001 VDISK$LINUX001$0202$0009     173      0      50      0
12:18:01 LINUX001 VDISK$LINUX001$0202$0009     293      0      35      0
12:19:01 LINUX001 VDISK$LINUX001$0202$0009     293      0      35      0
....
12:39:01 LINUX001 VDISK$LINUX001$0202$0009     259      0      35      0
12:40:01 LINUX001 VDISK$LINUX001$0202$0009     259      0      35      0
12:41:01 LINUX001 VDISK$LINUX001$0202$0009     207      0      86      0
12:42:01 LINUX001 VDISK$LINUX001$0202$0009     207      0      86      0
12:43:01 LINUX001 VDISK$LINUX001$0202$0009      13      0     280      0
12:44:01 LINUX001 VDISK$LINUX001$0202$0009      13      0     280      0
12:45:01 LINUX001 VDISK$LINUX001$0202$0009      13      0     280      0
```



# Cost of Swap

## Cost of Swap daemon about 10%, at 1,000 per second

Report: ESAHSTA LINUX HOST Application Report Domino Redbook ESAMAP 3.4.0 08/25/03  
 Monitor initialized: on 2066 serial 71CE3 record analyzed: 08/21/03 12:00:00

Node/ Date Time	Process/ Application name	<-Application Process Counts----->					<-----Processor----->		
		Total	active	Running	ResWait	Loaded	<---Utilization--->		
							Percent	seconds	Avg
<hr/>									
08/21/03									
12:15:00									
LINUXA	java	15.0	15.0	2.0	13.0	0	10.3	92.6	0.7
	kswapd	1.0	1.0	0	1.0	0	9.1	82.2	9.1
	router	11.0	11.0	0	11.0	0	10.6	95.4	1.0
	server	67.0	67.0	1.0	63.0	3.0	63.2	568.5	0.9
	snmpd	1.0	1.0	1.0	0	0	3.3	29.3	3.3
	update	3.0	3.0	1.0	2.0	0	10.2	91.7	3.4
<hr/>									
12:30:00									
LINUXA	java	17.0	17.0	2.0	15.0	0	9.5	85.9	0.6
	kswapd	1.0	1.0	0	1.0	0	8.8	79.5	8.8
	router	12.0	12.0	2.0	9.0	1.0	11.0	99.3	0.9
	server	61.0	61.0	4.0	55.0	2.0	62.7	563.9	1.0
	snmpd	1.0	1.0	1.0	0	0	3.2	28.8	3.2
	update	4.0	4.0	0	4.0	0	12.0	107.8	3.0
<hr/>									
12:45:00									
LINUXA	java	16.0	16.0	0	16.0	0	10.3	92.4	0.6
	kswapd	1.0	1.0	0	1.0	0	9.5	85.6	9.5
	router	10.0	10.0	0	10.0	0	11.1	99.6	1.1
	server	67.0	67.0	9.0	53.0	5.0	64.3	578.6	1.0
	snmpd	1.0	1.0	1.0	0	0	2.4	21.9	2.4
	update	5.0	5.0	0	5.0	0	13.0	116.9	2.6



# *Additional Storage Performance*

## Named Saved System

- Fast IPL, shared kernel storage
- Saves 1mb per server, difficult to implement

## DCSS with XIP File System

- Load all programs into shared DCSS,
- Saves 20-100mb/server, easy to implement

## CMM: Collaborative memory management

- Dynamically manage storage size
- Saves GB/server, requires feedback



# *How many Virtual Processors?*

- Linux is multiprocessor capable
- Global lock is very large issue
  - One processor acquires lock
  - Other processors attempt to spin
  - On 390 – spin converted to Diagnose 44
- Problem easily detected
  - High Diagnose -> Instruction Simulation -> SIE
  - High TV ratio
  - Guideline: Minimize virtual processors



# How many Virtual Processors?

Report: ESACPUA

CPU Utilization Analysis

-----										
		<CPU percents><--Internal (per second)-->						SIGP		
		Totl	Ovrhead		Diag	Inst	SIE	Fast	Page	Rate
Time	CPU	Util	Usr	Sys	nose	Sim	intrcp	path	fault	/sec
-----										
16:01:00	0	66.6	12	25	80K	82K	83275	2108	0.1	350
	1	67.6	12	25	89K	91K	91879	1051	0	332
	2	62.3	12	24	83K	85K	85768	1219	0.1	383
	3	62.7	11	25	77K	78K	79354	776	0	293
	4	63.6	12	24	84K	85K	86175	1047	0.0	329
	5	63.1	11	26	82K	84K	85064	1188	0.0	297
	6	64.1	11	22	83K	84K	84874	1079	0.0	304
	7	57.3	10	22	73K	75K	75481	1044	0.0	323
	8	62.7	10	26	53K	57K	58761	1421	0.1	267
-----										
System:		570	101	218	704K	723K	730630	11K	0.2	2879

- CPU Performance typical of many Linux Apps:
  - High Diagnose -> Instruction Simulation -> SIE
  - z/VM 5.2 modifies logic



# Mainframe I/O Expectation Issues

## Mainframe I/O expectations OFTEN wrong

- I/O traditionally tuned to operate within limitations
- Separate I/O processors
- Competition not limited by ESCON channel speeds

## Customer says “FTP on Linux under z/VM is slow”

- Benchmark was large FTP, problem NOT network
- Escon channels, 30ms “CONNECT” time
- 500K transfers

## Questions:

- How fast are ESCON channels? FICON channels? Ficon Express?
- How fast are SCSI disks on other platforms?

## PAV?

- What are options when high utilization on shared disks?
- PAV Available z/VM 5.2 - **Use for high activity shared devices ONLY**



# *FTP Benchmarks: Results NOT intuitive*

- Benchmark 1: (G5 processor)
  - FTP through Linux router with OSA dedicated to Virtual Router
  - FTP to Linux on single (ESCON) device
  - Throughput limited to 4mb / second: why?
- Benchmark 2:
  - Eliminate router, dedicate OSA,
  - Throughput increased to 8mb / second, why?
  - **Guideline: Use dedicated OSA or Virtual Switch**
- Benchmark 3:
  - Switch to LVM striped over 2 devices
  - Throughput reduced to 7mb / second, why?
  - **Guideline: Evaluate carefully use of striped LVM**
- **Answer to all questions: CPU was limiter, get a z10**



# *FTP Benchmarks: Results NOT intuitive*

Compare Linux Asynchronous I/O vs synchronous I/O

- Asynchronous is default
- Synchronous writes data without buffering
- DASD response time
  - Asynchronous: 50ms (6 I/O / second, 512k / IO),
  - Synchronous: 1.5ms (300 I/O / second, 4k / IO)
- Which is better throughput?
- **Guideline: Use Asynchronous?**
- DASD Response time rot don't work

■ **Guideline: Fight for FICON/(express)!!!!**



Overcommitting real storage is good, reduces cost

- Back up is Paging storage

If 40GB main storage

- Overcommit factor of 2 - How much paging storage needed?
- VM installations often very underconfigured
- **Guideline: Paging storage should still be 2 times requirement**

Number of paging devices? Number of channels?

- ROT not valid

Lack of page space planning is top reason for first installation  
z/VM outage



# *Infrastructure impact on CPU*

Shared resource environment:

- Avoid unnecessary work
- Avoid “waking up Linux”

Availability Monitoring – necessary?

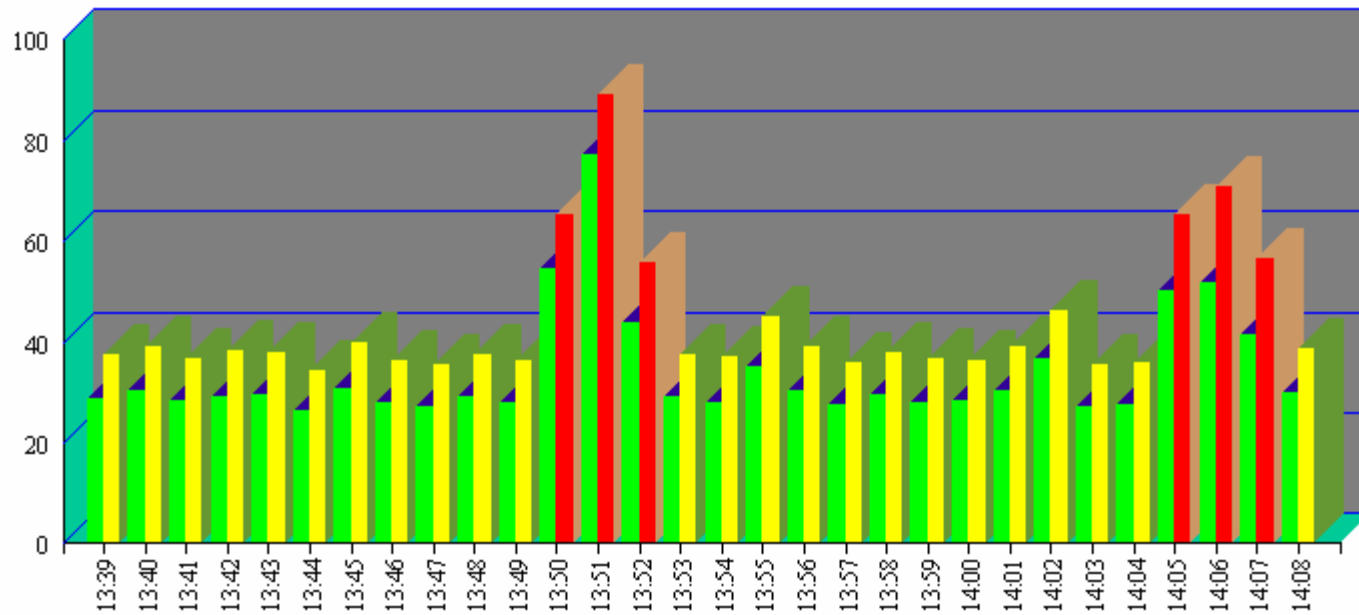
Using Encryption - necessary if on virtual lan?

Measure your infrastructure and determine scalability!



# Infrastructure: SOP Valid?

Virtual and Total Cpu Utilization



## Question:

- Why always hit every 15 minutes?

SOP: Standard Operating Procedure



# Infrastructure: Alerts

## Detect and alert looping processes

Report: ESAHST1      LINUX HOST Software Analysis Report  
Monitor initialized:      on 2066 serial 71CE3

```
-----  
Node/      <-----Software Program-----> <CPU Seconds> CPU   StgSize  
Time       Name      ID      Type   Status  Total  Intrval Pct   (Bytes)  
-----  
08:32:00  
LINUXA  
init        1      Applic ResWait  0.9      0.0   0.0    61440  
kjournal    95      Applic ResWait  2.5      0.0   0.0      0  
db2fmd      596     Applic ResWait  0.3      0.0   0.0   573440  
sshd       1081     Applic ResWait  0.4      0.0   0.0   204800  
event     10787     Applic ResWait 19.5      0.0   0.0   11188K  
snmpd     10861     Applic Running 193.4     4.2   7.1    1492K  
adminp    11452     Applic ResWait  58.5      0.0   0.1   13848K  
server    11525     Applic ResWait  1.0      0.1   0.1   35720K  
server    11533     Applic ResWait  4.3      0.0   0.0   35720K  
server    11537     Applic Running 44697     58.3  99.2  35720K  
java      13024     Applic ResWait  0.0      0.0   0.0    6632K  
java      24016     Applic ResWait  1.9      0.0   0.0    6632K  
java      24024     Applic ResWait  4.9      0.0   0.0    6632K  
server    24192     Applic ResWait 19.0      0.1   0.1   35720K  
java      26352     Applic ResWait  0.4      0.0   0.0    7320K  
sshd      26477     Applic ResWait  0.2      0.0   0.1   2028K
```

### Show process by ID

- Status
- Total CPU
- Percent CPU
- Storage

(Non-velocity mib)



# Performance Instrumentation

## Performance Instrumentation

- Cost of instrumentation often excessive
- “Native Linux” tools will not detect many problems
- Agents may take 5-10% of a processor (Per server)
- An agent that takes 1% will consume 10 IFLs for 1000 servers

**Cost of instrumentation should be < .1% (of ONE CPU) per server**

- **Performance instrumentation should not change performance**

## Active agents vs Passive agents

- Active agent wakes up at constant interval and records data
- Passive agent only responds to external request

**Dynamically turn off monitoring of idle servers!!!!**

- If z/VM data shows server is idle, should agent wake up to find out what is running?



# Linux Configuration Summary

## Virtual machine size

- Minimize until some swap

## Swapping

- Swap to virtual disk
- Define 2 virtual disks,
  - One to meet the average requirement
  - Second one for overflow
- Use DIAG driver instead of FBA
  - Reduces I/O by factor of 8

## Virtual processors

- Minimize to meet the workload/application requirement

## Infrastructure costs

- **UNDERSTAND and Minimize – shared resource architecture**



# *z/VM Subsystem Configuration*

## DASD Channels

- ESCON channels are 17MByte / second
- Ficon channels 100MB, Ficon Express 200MB
- Ficon compares to SCSI disks on other platforms

## Paging/Spooling

- How much paging is required to support 2 times over commitment of 40GB z/VM system? (80 gb of page space)

## MDC

- Caches data – read-ahead, often used data
- Default too high
- SET MDC STORAGE 0M 128M
- SET MDC XSTORE 0M 0M



# *z/VM Expanded Storage*

Expanded storage is very necessary for paging hierarchy

Expanded Storage Requirement:

- No “stealing” to disk.
- 20% should be sufficient
- Little cost if too large

If page steal to disk, convert more real storage to expanded storage

**SET MDC XSTORE 0 0**

- (MDC in expanded storage has little value)



## SET SHARE

- Use RELATIVE 100 for single virtual CPU
- Use RELATIVE 200 for two virtual CPU

## SET SRM STORBUF – allow overcommit

- SET SRM STORBUF 300 300 300
- SET SRM LDUBUF 100 80 60 (or lower)

## SET QUICKDSP

- **Use for only absolutely critical servers**



