IBM Systems & Technology Group

z/VM Performance Case Studies

Please consider sitting near the front.

Bill Bitner
VM Performance Evaluation
bitnerb@us.ibm.com

Trademarks

Trademarks

NOTES:

Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput 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 improvements equivalent to the performance ratios stated here.

IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

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

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the 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.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

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

Any proposed use of claims in this presentation outside of the United States must be reviewed by local IBM country counsel prior to such use.

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

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

Agenda

- **Brief review of collecting performance data.**
- **Brief review of IBM Performance Support**
- **Critical Situations**
- **Case Studies**

Collecting Raw Monitor Data

- **What is raw monitor data?**
- **How do I set up to collect it?**
- **When do I collect it?**
- **How do I package it for transmission?**
- **Summary**

What is Raw Monitor Data?

- **It is unformatted binary data describing system configuration or activity**
- **Logically, it is a sequence of *monitor records***
  - Each record comments on some specific aspect of system activity or performance
  - In aggregate they constitute a comprehensive, time-indexed record of system activity
- **There are three large classes of monitor records**
  - *Configuration records:* emitted when monitor starts, these describe system configuration
  - *Sample records:* emitted every so often, these comment on the accumulated activity of an entity (device, user, …)
  - *Event records:* emitted as needed, these comment on some specific phenomenon that just now occurred
- **Some records come from the Control Program and comment on its experience in running the system**
- **Other records come from guests and comment on their experiences in doing whatever it is they do**
- **We collect this data using an IBM-supplied utility program called MONWRITE**
- **During the rest of this presentation, we will call this data *MONWRITE data***

How Do I Collect MONWRITE Data?

- **By Default the z/VM system is set up with DCSS and user ID named MONWRITE**
- **If somehow skipped, then:**
  - You set up a DCSS where CP will buffer the monitor records it emits
    - CP DEFSEG and SAVESEG commands
  - You tell CP which kinds of records to emit, and how often to emit them, and in fact to begin emitting them
    - CP MONITOR command
  - You set up a guest that drains the DCSS to a disk or a tape via the MONWRITE utility
- **On some occasions, the default DCSS (named MONDCSS) is too small.**
  - See http://www.vm.ibm.com/perf/tips/mondcss.html
- **You run the guest**
- **You archive the resultant files or tapes, so that you have a long-term historical record of system activity and performance**

When Do I Collect MONWRITE Data?

- **Periodically, collect and archive some data during your peak periods, so that you have a historical record**
  - Peak loads
  - Month-end processing
  - Significant Changes (e.g. moving from z10 to z196, refreshing level of application code)
- **When directed by IBM**
  - Health check, PMR, crit sit, ESP, whatever

Tool: Running MONWRITE By Hand

- **The default install of z/VM will create a MONWRITE userid. If you did not do this, it is pretty straight forward. See http://www.vm.ibm.com/perf/tips/collect.html**
- **Basic Steps**
  - Create the DCSS to hold the buffered records
  - Set up a guest to run our MONWRITE MODULE (collector)
  - Issue some CP MONITOR commands to start CP emitting records
    - Enable all samples
    - Enable all events except seeks and scheduler
    - Use a 1-minute sample interval and a 1-second HFS rate
  - In your guest, start MONWRITE to collect the z/VM Monitor data
  - To stop collecting, type this: MONWSTOP
- **You will end up with one MONWRITE file that you can:**
  - Archive for the historical record
  - Analyze yourself with z/VM Performance Toolkit
  - Send to IBM so we can look at it
- **There is an option for MONWRITE to close the file at regular times of day and a user exit to process the just-closed file.**
- **Good references:**
  - http://www.vm.ibm.com/perf/tips/collect.html - a good cheat sheet
  - z/VM Performance, chapter 9, “Monitoring Performance Using CP Monitor” – an excellent writeup of every last detail

Packaging MONWRITE Data For Transmission

- **MONWRITE files are binary CMS files, F 4096.**
- **Just attaching them to an e-mail is NOT recommended.**
- **The standard z/VM Level 2 process for FTPing files calls for COPYFILE (PACK**
  - This is unnecessary for MONWRITE files.
- **Most important thing:**
  - Move the files in binary (ASCII is a sure showstopper)
- **You will probably FTP your data to IBM's receiving server in Boulder, CO**
  - Testcase.boulder.ibm.com, cd /toibm/vm
  - Name your file according to the service process.
  - See http://techsupport.services.ibm.com/390/tcprocs.html for additional info on the Testcase process
  - In PMR and/or note be clear as to what is sent and how packaged
- **We suggest you use the VMARC file archiver that runs on CMS when several files are being sent or if network is very slow.**
  - Kind of like "zipping" on a PC (compresses, combines)
    - MONWRITE data is very compressible
    - Sometimes you also want to send us a console spool, or some QUERY outputs, or whatever
    - You can package everything into one VMARC archive and just send us that

Studying MONWRITE Data

- **z/VM Performance Toolkit**
- **Interactively – possible, but not so useful**
- **PERFKIT BATCH command – pretty useful**
  - Control files tell Perfkit which reports to produce
  - You can then inspect the reports by hand or programmatically
- **See z/VM Performance Toolkit Reference for information on how to use PERFKIT BATCH**

Other Types of Data Confused with MONWRITE Data

- **Asking for “raw VM monitor” data can be confusing.**
  - Velocity has their own form of raw monitor data and history files, and even a form that matches MONWRITE.
  - “VM Monitor” sounds like the “VM:product” often associated with CA products.
  - Performance Toolkit’s history, trend, and summary files do not have the same detail.
- **Be specific when asking for data.**

Monwrite Summary

- **MONWRITE data is a comprehensive record of system activity**
- **It is invaluable in diagnosing performance concerns**
- **If you ask IBM for performance help, IBM will very likely ask you for MONWRITE data**
- **Practice collecting and transmitting MONWRITE data when you are not under duress**
- **Archive your MONWRITE data routinely so that you have a good record of your system's usual behavior**
- **Learn to use PERFKIT BATCH to generate reports, and get familiar with a few of the basic reports**

Performance Support

The typical lines of support:

1. Your FTSS (Field Technical Sales Support)
2. If FTSS needs help, they will contact Region Designated Support (RDS)
3. If RDS needs help, they will contact Advanced Technical Support (ATS)
4. If ATS needs help, they will contact z/VM Development Lab

– **You may also have contract for Q&A help**

Crit Sit Survival: Agenda

- **How are problems in a virtualized environment different?**
  - A bigger picture view is required
  - More SMEs should be involved
  - Manifestation of the problem is frequently different than source of the problem
- **How are they the same?**
  - Communication is key
  - Project (IBM & Customer) owner is key
  - May involve non-IBM/customer parties

Crit Sit: Avoid One in First Place

- **System Review Process**
- **POC as if it will be production**
- **Change management**
- **Stay current**
  - Red Alerts: http://www.vm.ibm.com/service/redalert/index.html

Crit Sit: Checklist

- **Communication**
  - Determine who needs to be in loop from a management & technical perspective
  - Point people on IBM & Customer
  - Distribution list
  - What will be communicated via which vehicle
- **Create a "Who's Who" list**
- **Gather data on problem and systems**
  - Configuration & Network Diagrams
  - Determine how many problems there might be and who owns which
  - Timezone for the different data
  - Naming conventions for the data to aid in matching z/VM to Linux to logs etc.
- **Determine "Go Home" Criteria (i.e. what constitutes success?)**

Case Study: Logical to Physical

Logical to Physical Processor Ratios

- **As the number of partitions and their size increases, questions continue to arise as to how to configure z/VM systems**
- **This case study illustrates some of the factors and information that can be examined**
- **More complex scenarios would include mixed engine environments**

Configuration

- **2097-401**
- **18 Physical Processors**
  - 1 CP
  - 17 IFLs
    - 3 Dedicated
- **11 Partitions**
  - 5 Active Shared: 3+3+13+13+2 = 34 logicals IFLs
- **Ratio of Non-dedicated Logical to Physical CPUs: 2.4**

FCX202 LPARLOG - Partition Configs

<Partition->						
Name	Nr.	Upid	#Proc	Weight	Wait-C	Cap
DRLPAR	1	..	0	0	NO	NO
A5Q1	2	..	0	0	NO	NO
A5Q2	3	..	0	0	NO	NO
A5Q3	4	..	0	0	NO	NO
A5Q4	5	..	0	0	NO	NO
A5T	6	15	2	4	NO	NO
A5X	7	14	3	DED	YES	NO
LPAR1	8	01	3	2	NO	NO
LPAR2	9	02	13	46	NO	NO
LPAR3	10	03	3	2	NO	NO
LPAR4	11	04	13	46	NO	NO

- 14 undedicated IFLs
- LPAR2 weight equates to 6.44 IFLs
  - 6.44 << 13

FCX100 CPU - Looking at Processor Time

PROC	TYPE	%CPU	%CP	%EMU	%WT	%SYS	%SP	%SIC	%LOGLD
P00	IFL	46	6	40	54	4	2	77	65
P12	IFL	46	5	41	54	3	3	76	65
P11	IFL	46	5	41	54	3	3	76	65
P01	IFL	46	5	41	54	3	3	76	65
P02	IFL	46	5	41	54	3	2	77	65
P03	IFL	46	5	40	54	4	2	76	65
P04	IFL	46	5	41	54	3	3	76	65
P05	IFL	46	5	41	54	3	2	76	65
P06	IFL	46	5	40	54	4	2	76	65
P07	IFL	46	5	41	54	3	3	77	65
P08	IFL	46	5	41	54	3	3	76	65
P09	IFL	46	5	41	54	3	3	76	65
P10	IFL	46	5	41	54	3	3	77	65

%CPU: total cycles consumed in z/VM.

%CP: total cycles in z/VM control program

%EMU: total cycles inside z/VM guests

%SYS: total cycles in CP not associated with a guest (subset of %CP)

%SP: wall clock time in formal spin locks

%LOGLD: pct busy time of time z/VM timers are running

FCX126 LPAR - Looking at Processor Time

%Load	CPU	%Busy	%Ovhd	%Susp	%VMld	%Logld	Type
34.0	0	47.2	1.4	29.8	45.6	64.9	IFL
	1	47.1	1.3	29.8	45.6	64.9	IFL
	2	47.1	1.3	29.8	45.5	64.8	IFL
	3	47.2	1.4	29.9	45.6	64.9	IFL
	4	47.1	1.3	29.8	45.5	64.9	IFL
	5	47.1	1.4	29.9	45.5	64.8	IFL
	6	47.1	1.3	29.8	45.5	64.8	IFL
	7	47.1	1.2	29.8	45.6	64.9	IFL
	8	47.1	1.2	29.8	45.7	65.0	IFL
	9	47.1	1.3	29.8	45.6	65.0	IFL
	10	47.1	1.2	29.8	45.7	65.0	IFL
	11	47.0	1.2	29.7	45.6	64.8	IFL
	12	47.1	1.2	29.8	45.6	65.0	IFL

%Busy: Total CPU

%Ovhd: LPAR Mgmt Time for LCPU

%VMld: %CPU from FCX100

%Susp: 100% - total of z/VM Timers

%Logld: %Logld from FCX100

LPAR Mgmt Time (Overhead)

- **%Ovhd on FCX126 is LPAR management time associated with a given partition's LCPU**
- **General LPAR overhead also reported, not associated with a given partition.**
- **Mgmt time can be influenced by activity and requests from within the partitions**

LPAR Suspend Time

- **An approximation of when z/VM partition is removed from running for either:**
  - Being capped
  - Running other partitions
  - z/VM giving up time via diagnoses while waiting on locks
- **Another side effect of high suspend time**
  - z/VM User State Sampling could be skewed

Reconfigure the Logical Processor Counts

Phys	Ded.	LCPUs	Log:Phy	%LPBUSY	%LPOVHD	%NCOVHD	%BUSY	%SUSP
17	3	34	2.4	1249	41	26	1316	29.8%
17	3	24	1.7	851	18	18	887	3.5%

Before and After above shows:

- Both flavors of overhead dropped.
- Suspend time back to a reasonable number.

FCX265 LOCKLOG

		<----- Before----->			<----- After ----->		
Interval		Locks	Average	Pct	Locks	Average	Pct
End Time	LockName	/sec	usec	Spin	/sec	usec	Spin
>>Mean>>	SRMATDLK	563.8	71.78	.311	474.8	29.76	.157
>>Mean>>	RSAAVCLK	.0	458.6	.000	.0	1.306	.000
>>Mean>>	RSA2GCLK	.0	187.3	.000	.1	6.128	.000
>>Mean>>	BUTDLKEY	.0	145.0	.000	.0	.243	.000
>>Mean>>	HCPTMFLK	.0	.000	.000	.0	.000	.000
>>Mean>>	RSA2GLCK	6.6	63.55	.003	16.8	8.880	.002
>>Mean>>	HCPRCCSL	.0	.000	.000	.0	.000	.000
>>Mean>>	RSASXQLK	2.9	61.99	.001	3.1	11.17	.000
>>Mean>>	HCPPGDML	.5	174.9	.001	.7	26.71	.000
>>Mean>>	NSUIMGLK	.0	.000	.000	.0	.000	.000
>>Mean>>	FSDVMLK	4.3	39.73	.001	6.8	14.62	.001
>>Mean>>	HCPPGDPL	1.5	190.9	.002	1.7	81.73	.002
>>Mean>>	SRMALOCK	.0	.000	.000	.0	.000	.000
>>Mean>>	HCPTRQLK	434.5	51.29	.171	306.0	3.439	.012
>>Mean>>	SRMSLOCK	3062	89.98	2.119	2193	20.15	.491

Summary

- **Various rules of thumbs for Logical to Physical**
- **Starting points**
- **Look at data**
- **Suspend time is helpful but has multiple causes**

Case Study: Long Back Ups

Customer Situation

- **Customer is a holding company for several businesses that have been acquired over time.**
- **VSE and z/OS systems run the businesses**
- **Need to back up data on a regular basis**
- **Number of VSE systems and devices increased as companies acquired.**
- **Using z/VM to address device address limits in VSE**
- **Back up Strategy**
  - Quiesce VSE and use FLASHCOPY to duplicate volumes
  - Use FASTCOPY in a utility VSE guest to backup the data

Configuration

- **2096-S03**
- **18 Physical Processors**
  - 3 CPs
- **2 Partitions**
  - Dummy: no resources associated with it
  - PROD: 3 Dedicated CPs 14GB cstore
- **Approximately 7 VSE virtual machines**
- **~11 TB of data**

Results

- **Initial**
  - ~14 hours for backups
- **Storage specialists recommendations**
  - ~13 hours for backups
- **New recommendations increase number of virtual processors for utility VSE**
  - ~14 to 16 hours
- **At this point we really needed to look closer at the data**

FCX100 CPU

CPU Load

PROC	TYPE	%CPU	%CP	%EMU	%WT	%SYS	%SP	%SIC	%LOGLD
P00	CP	72	25	47	28	1	0	82	72
P01	CP	73	24	48	27	1	0	81	73
P02	CP	73	24	48	27	1	0	81	73

- Not maxed out on CPU
- Higher amount of %CP CPU time
- Low amount of %SYS CPU time

FCX108 DEVICE for Tape Drives

```
<-- Device Descr. -->  Mdisk Pa- <-Rate/s-> <------- Time (msec) ------->
Addr Type   Label/ID   Links ths  I/O Avoid Pend Disc Conn Serv Resp CUWt
0179 3590   >VSEB        ...   3 71.9   ...   .2   .1  1.8  2.1  2.1   .0
0173 3590   >VSEB        ...   3 71.8   ...   .2   .1  1.8  2.1  2.1   .0
0178 3590   >VSEB        ...   3 69.8   ...   .2   .1  1.7  2.0  2.0   .0
0174 3590   >VSEB        ...   3 68.2   ...   .2   .1  1.6  1.9  1.9   .0
0177 3590   >VSEB        ...   3 68.0   ...   .2   .2  1.7  2.1  2.1   .0
0171 3590   >VSEB        ...   3 66.2   ...   .2   .1  1.8  2.1  2.1   .0
0176 3590   >VSEB        ...   3 51.1   ...   .2   .1  1.8  2.1  2.1   .0
0172 3590   >VSEB        ...   3 36.5   ...   .2   .1  1.8  2.1  2.1   .0
```

- Above shows subset of Tape devices used in backup
- Adding up shows about 500 I/Os per Second
- Low Pending and Disconnect time
- Looks reasonable

FCX108 DEVICE for DASD being backed up

```
<-- Device Descr. -->  Mdisk Pa- <-Rate/s-> <------- Time (msec) ------->
Addr Type   Label/ID   Links ths  I/O Avoid Pend Disc Conn Serv Resp CUWt
7153 3390   >VSEB          0   8 11.0    .0   .2  2.2  1.8  4.2  4.2   .0
718F 3390   >VSEB          0   8 11.0    .0   .2  4.4  3.7  8.3  8.3   .0
7192 3390   >VSEB          0   8 11.0    .0   .2  1.2  1.6  3.0  3.0   .0
7040 3390   >VSEB          0   8 10.9    .0   .2  6.7  2.8  9.7  9.7   .0
70DF 3390   >VSEB          0   8 10.9    .0   .2  2.3  1.9  4.4  4.4   .0
707A 3390   >VSEB          0   8 10.9    .0   .2  5.5  1.7  7.4  7.4   .0
707D 3390   >VSEB          0   8 10.7    .0   .2  6.3  1.7  8.2  8.2   .0
7190 3390   >VSEB          0   8 10.7    .0   .2  3.4  1.7  5.3  5.3   .0
7150 3390   >VSEB          0   8 10.7    .0   .2  8.7  5.0 13.9 13.9   .0
70DE 3390   >VSEB          0   8 10.6    .0   .2  8.1  2.5 10.8 10.8   .0
```

- Above shows subset of DASD devices used in backup
- Adding up shows about 500 I/Os per Second
- Most service times look reasonable for larger I/Os

FCX112 User Report

	<----- CPU Load ----->				<------ Virtual IO/s ------>					
		<-Seconds->		T/V						
Userid	%CPU	TCPU	VCPU	Ratio	Total	DASD	Avoid	Diag98	UR	Pg/s
>>Mean>>	6.49	525.3	353.3	1.49	92.3	42.8	4.2	.0	.0	.0
VSEE	85.3	6908	6622	1.04	332	321	83.5	.0	.0	.0
VSEB	85.1	6893	2656	2.59	1031	527	1.3	.0	.0	.0
VSEA	33.4	2703	1631	1.66	1558	451	31.9	.0	.0	.0
VSEC	4.45	360.7	342.3	1.05	41.3	31.2	16.1	.0	.0	.0
VSED	3.20	259.5	223.0	1.16	72.0	71.8	4.8	.0	.0	.0
VSEPROD	1.39	112.5	107.2	1.05	5.0	4.1	.0	.0	.0	.0
VSETEST	.46	36.94	34.80	1.06	2.1	1.9	.0	.0	.0	.0
PERFSVM	.27	21.99	18.52	1.19	.0	.0	.0	.0	.0	.0
HOBBITVM	.04	3.123	2.686	1.16	1.2	1.2	.0	.0	.0	.0
TCPIP	.03	2.215	1.524	1.45	.0	.0	.0	.0	.0	.0

- VSEB is our utility VSE; virtual 3-way
- High TVRatio and a lot of I/O

FCX162 USERLOG for VSEB

```
Resource Usage Log for User VSEB

          <----- CPU Load -----> <------ Virtual IO/s ------>
Interval       <-Seconds->   T/V
End Time  %CPU  TCPU  VCPU Ratio Total DASD Avoid Diag98   UR
>>Mean>>  85.1 255.3 98.38  2.59  1031  527   1.3     .0   .0
10:47:25   100 299.9 114.0  2.63  1219  615   1.0     .0   .0
10:52:25   101 301.4 114.8  2.63  1197  604   1.5     .0   .0
10:57:25   103 309.1 117.2  2.64  1236  625   1.6     .0   .0
11:02:25   101 303.5 116.0  2.62  1212  612   1.5     .0   .0
11:07:25   104 312.0 119.0  2.62  1238  623    .8     .0   .0
11:12:25   105 313.9 118.3  2.65  1258  635   1.3     .0   .0
```

- While average is 85.1%, often runs just over 100% (This is a virtual 3-way)

FCX164 USTATLOG for VSEB

Wait State Data Log for User TMKB

Interval										<-SVM and->						
End Time	%ACT	%RUN	%CPU	%LDG	%PGW	%IOW	%SIM	%TIW	%CFW	%TI	%EL	%DM	%IOA	%PGA	%LIM	%OTH
>>Mean>>	95	36	19	0	0	1	0	14	0	0	0	0	29	0	0	1
10:47:25	100	44	17	0	0	1	0	7	0	0	0	0	31	0	0	0
10:52:25	100	41	17	0	0	1	0	6	0	0	0	0	35	0	0	0
10:57:25	100	42	18	0	0	1	0	7	0	0	0	0	33	0	0	0
11:02:25	100	45	21	0	0	0	0	5	0	0	0	0	29	0	0	0
11:07:25	100	51	19	0	0	0	0	9	0	0	0	0	21	0	0	1
11:12:25	100	45	31	0	0	0	1	5	0	0	0	0	19	0	0	0

- At various times, there is a non-trivial %CPU wait
- Also high %IOA times
- Remember other virtual machines also used significant CPU time.

Normalized Share Values

- **VSEE 85.3% virtual 2-way Share Relative 2800**
- **VSEB 85.1% virtual 3-way Share Relative 1900**

- **VSEE each virtual processor dispatched as Relative 1400**
- **VSEB each virtual processor dispatched as Relative 633**

Additional CPU Tuning?

- **What if there is just one process/partition of work in VSE that can be run for the backup job?**
- **What if we adjusted the Shares, to give VSEB higher priority?**

- **Changing Share values resulted in:**
  - Brought ~14 to 16 hour elapsed times down to ~ 10 to 12 hour elapsed times.
  - Still not acceptable

Lets back up – How is I/O handled in z/VM?

- **Traditional SSCH I/O involves a channel program with a series of channel commands.**
- **Each virtual I/O must be translated:**
  - Virtual memory addresses must be translated to real memory addresses
  - Minidisk cylinder locations must be translated to real cylinder locations
  - The reverse translation on return (real to virtual addresses) must be performed also
- **The Backup Utility is very straight forward. It builds a channel program and just adjusts a few pointers as it ripples through the disk being backed-up. Just a few instructions**
- **Remember the high TV Ratio? A lot of overhead required in z/VM**
- **For the amount of data involved here, we estimated there were roughly 45 to 50 billion translations required!!!**

FCX112 User Report

	<----- CPU Load ----->				<------ Virtual IO/s ------>					
		<-Seconds->		T/V						
Userid	%CPU	TCPU	VCPU	Ratio	Total	DASD	Avoid	Diag98	UR	Pg/s
>>Mean>>	6.49	525.3	353.3	1.49	92.3	42.8	4.2	.0	.0	.0
VSEB	85.1	6893	2656	2.59	1031	527	1.3	.0	.0	.0

- We could reduce the elapsed time by ~20% with correcting share settings.
- Look at breakdown of CPU seconds for the interval above:
  - Total = 6893 seconds
  - Virtual = 2656 seconds
  - CP = 4237 seconds (or 60%)
- Reducing the 10 to 12 hours by 60% gets us to the goal!

Summary

- **There wasn’t a real need for z/VM for the utility VSE**
  - Could be brought up in a separate LPAR after the FlashCopy of the volumes were completed.
  - Run as shared LPARs instead of dedicated.
- **Customer decided to use z/OS on a different CEC to manage the backups**
  - One approach for both VSE & z/OS
- **Other solutions could exist**
  - Did all 11 TB need to be backed up each night? Certainly 11TB weren’t turned over each day.
- **The original solution considered the overhead of z/VM for ‘normal’ workloads, not this unique case.**

Case Study: The account team ate my sizing

Customer Situation

- **Running Domino on Linux for system z on z/VM**
- **Performance is bad**
  - High response time and applications timing out
  - Occurs almost all the time
- **Domino and Linux experts also involved**
- **Very low z/VM skills**

Customer Configuration

- **System**
  - 1 dedicated IFL on 2096-D02
  - 2GB Central Storage; 0GB Expanded Storage
- **Linux Guest**
  - Virtual 1-way
  - 2GB Memory
  - Connected to Layer 3 Network via VSwitch

FCX112 User Report – Anything else trying to run?

	<----- CPU Load ----->				<------ Virtual IO/s ------>					
		<-Seconds->		T/V						
Userid	%CPU	TCPU	VCPU	Ratio	Total	DASD	Avoid	Diag98	UR	Pg/s
>>Mean>>	2.29	26.08	24.86	1.05	21.3	21.3	11.9	.0	.0	86.7
LNXDOM1	27.4	312.9	298.3	1.05	252	252	143	.0	.0	1019
MONWRITE	.01	.067	.032	2.09	3.2	3.2	.0	.0	.0	.9
DISKACNT	0	0	0	....	0	0	0	0	0	0
DTCVSW1	.00	.002	.001	2.00	.0	.0	.0	.0	.0	1.6
DTCVSW2	.00	.002	.001	2.00	.0	.0	.0	.0	.0	1.7
MAINT	0	0	0	....	0	0	0	0	0	0
OPERATOR	0	0	0	....	0	0	0	0	0	0
OPERSYMP	0	0	0	....	0	0	0	0	0	0
TCPIP	.00	.028	.010	2.80	.0	.0	.0	.0	.0	17.4
VMSERVR	.00	.000	.000	....	.0	.0	.0	.0	.0	.1
VMSERVS	.00	.000	.000	....	.0	.0	.0	.0	.0	.1
VMSERVU	.00	.000	.000	....	.0	.0	.0	.0	.0	.1

FCX164 USTATLOG for LNXDOM1

Wait State Data Log for User LNXDOM1

Interval										<-SVM and->				
End Time	%ACT	%RUN	%CPU	%LDG	%PGW	%IOW	%SIM	%TIW	%CFW	%TI	%EL	%DM	%IOA	%PGA
>>Mean>>	100	32	0	16	20	0	0	21	0	0	0	0	10	1
23:36:48	100	17	0	33	10	0	0	23	0	0	0	0	10	7
23:37:48	100	7	0	10	13	0	0	53	0	0	0	0	17	0
23:38:48	100	20	0	10	27	0	0	33	0	0	0	0	10	0
23:39:48	100	37	0	3	20	0	0	27	0	0	0	0	10	3
23:40:48	100	30	0	23	23	0	0	13	0	0	0	0	7	3
23:41:48	100	33	0	7	30	0	0	27	0	0	0	0	3	0
23:42:48	100	30	0	17	23	0	0	17	0	0	0	0	10	3
23:43:48	100	37	0	10	27	0	0	17	0	0	0	0	10	0
23:44:48	100	30	0	13	13	0	0	23	0	0	0	0	17	3
23:45:48	100	23	0	33	17	0	0	17	0	0	0	0	10	0
23:46:48	100	30	0	10	10	0	3	17	0	0	0	0	27	3
23:47:48	100	30	0	7	23	0	0	30	0	0	0	0	7	3

FCX143 PAGELOG

<-Real Stor->		<------------ Paging to DASD				------------>			<Page Table>		
	DPA	Est.					<-Single Reads-->			<Management>	
Interval	Pagable	Page	Reads	Write	Total	Shrd	Guest	Systm	Total	Reads	Writes
End Time	Frames	Life	/s	/s	/s	/s	/s	/s	/s	/s	/s
>>Mean>>	504476	1316	660.6	383.2	1044	10.6	274.4	.0	274.4	.0	.0
23:36:48	504495	1512	493.6	333.5	827.0	9.9	161.8	.0	161.8	.0	.0
23:37:48	504485	1890	495.9	266.8	762.7	10.2	195.0	.0	195.0	.0	.0
23:38:48	504479	1601	575.3	315.0	890.3	12.6	244.9	.0	245.0	.0	.0
23:39:48	504473	1834	522.2	275.0	797.2	10.1	253.6	.0	253.6	.0	.0
23:40:48	504476	1529	558.1	329.8	887.8	9.7	249.6	.0	249.7	.0	.0
23:41:48	504476	989	777.2	509.7	1287	10.8	271.0	.0	271.1	.0	.0
23:42:48	504471	942	869.6	535.2	1405	9.7	296.5	.0	296.5	.0	.0
23:43:48	504462	1436	614.6	351.3	965.9	14.3	289.5	.0	289.5	.0	.0
23:44:48	504470	1162	766.7	433.8	1201	11.7	311.5	.0	311.5	.0	.0
23:45:48	504476	1420	650.0	355.0	1005	10.7	281.8	.1	281.9	.1	.1
23:46:48	504476	1514	570.6	333.1	903.8	10.4	250.2	.1	250.2	.1	.1
23:47:48	504478	1458	623.8	345.9	969.7	9.7	254.3	.0	254.3	.0	.0

FCX109 DEVICE CPOWN

< Device Descr. ->				<-------- Rate/s ------>				User		Serv	MLOAD	Block
		Volume	Used	<--Page--->			SSCH	Inter	Queue	Time	Resp	Page
Addr	Devtyp	Serial	%	P-Rds	P-Wrt	Total	+RSCH	feres	Lngth	/Page	Time	Size
DC02	3390-9	530PAG	2	31.9	14.2	46.1	21.5	1	.05	.9	.9	9
DC05	3390-9	VM1PG1	4	128.6	74.3	202.8	69.9	1	.05	.3	.3	10
DC06	3390-9	VM1PG2	4	125.4	74.4	199.8	67.4	1	1.15	.5	.8	9
DC07	3390-9	VM1PG3	4	117.7	66.6	184.3	67.3	1	.05	.4	.4	9
DC08	3390-9	VM1PG4	4	125.7	74.0	199.7	68.3	1	0	.3	.3	9
DC09	3390-9	VM1PG5	4	131.4	79.8	211.1	69.2	1	.05	.3	.3	10

Roughly 364 I/Os / Second for Paging

FCX108 DEVICE

```
.    .                     ___     .    .    .    .    .    .    .
<-- Device Descr. --> Pa- <-Rate/s-> <------- Time (msec) ------->
Addr Type   Label/ID  ths  I/O Avoid Pend Disc Conn Serv Resp CUWt
>> All DASD <<              .1    .0   .2   .7   .8  1.7  1.7   .0
DC05 3390-9 VM1PG1 CP   4 69.8    .0   .2   .0  1.0  1.2  1.2   .0
DC09 3390-9 VM1PG5 CP   4 69.1    .0   .2   .0   .9  1.1  1.1   .0
DC08 3390-9 VM1PG4 CP   4 68.3    .0   .2   .0  1.0  1.2  1.2   .0
DC06 3390-9 VM1PG2 CP   4 67.4    .0   .2   .0  1.0  1.2  1.2   .0
```

FCX131 DEVCONF

```
 <----   Ranges   ---->   Device      <- Channel Path Ids  ->   Control
  Device-No  Subch.-ID   Type          1  2  3  4  5  6  7  8   Unit       Status
--------------------  33  line(s) not displayed --------------------
  DC00-DC37  0E21-0E58   3390-9 (E)  14 20 30 44  .  .  .  .   2107-E8    Online
  DD00-DD37  0E59-0E90   3390-9 (E)  14 20 30 44  .  .  .  .   2107-E8    Online
  DE00-DE37  0E91-0EC8   3390-9 (E)  14 20 30 44  .  .  .  .   2107-E8    Online
  DF00-DF37  0EC9-0F00   3390-9 (E)  14 20 30 44  .  .  .  .   2107-E8    Online
  E000-E08F  0F01-0F90   3390-3 (E)  15 21 31 45  .  .  .  .   2107-E8    Online
```

FCX113 UPAGE

	Data	<---------- Paging Activity/s ---------->						
	Spaces	<Page Rate>		Page	<--Page Migration-->			
Userid	Owned	Reads	Write	Steals	>2GB>	X>MS	MS>X	X>DS
LNXDOM1	.0	648	371	651.0	.0	.0	.0	.0
MONWRITE	.0	.5	.5	.5	.0	.0	.0	.0
OPERATOR	.0	.0	.0	.0	.0	.0	.0	.0
TCPIP	.0	9.0	8.4	9.1	.0	.0	.0	.0

	<----------------- Number of Pages ----------------->							
			<-Resident->		<--Locked-->			
Userid	WSS	Resrvd	R<2GB	R>2GB	L<2GB	L>2GB	XSTOR	DASD
LNXDOM1	241010	0	238122	0	24	0	0	337367
MONWRITE	28	0	0	0	0	0	0	100
OPERATOR	1	0	0	0	0	0	0	1
TCPIP	17	0	357	0	338	0	0	2699

FCX178 MDCSTOR & FCX138 MDCACHE

<-------------- Main Storage Frames --

Interval		<--Actual--->		Min	Max	Page
End Time	Ideal	<2GB	>2GB	Set	Set	Del/s
>>Mean>>	260891	260774	0	0	524288	697
23:36:48	261683	261511	0	0	524288	449
23:37:48	263487	263240	0	0	524288	349
23:38:48	262785	262752	0	0	524288	611
23:39:48	258700	258509	0	0	524288	53
23:40:48	256061	256011	0	0	524288	58
23:41:48	256109	255981	0	0	524288	80
23:42:48	256676	256623	0	0	524288	64
23:43:48	257472	257400	0	0	524288	759
23:44:48	257160	257052	0	0	524288	878
23:45:48	257529	257397	0	0	524288	861
23:46:48	259847	259832	0	0	524288	769
23:47:48	262062	261976	0	0	524288	669

<--- MDC Requests ---->

Reads	Full	Part.	
/s	Hit%	Hit%	Miss%
216	66.3	..	..

Recommendations & Summary

- **Increase real memory and add some expanded storage.**
- **Create Vdisk for paging (Guest was also swapping significantly at Linux level)**
- **Create a plan to move to z/VM 6.1.0**
- **Put a cap on MDC of 256MB**
- **Asked about original sizing**
  - <insert sound of crickets here>
- **Original sizing recommended 6GB for the Linux guest!**
- **While any sizing will be an estimate, that isn't a reason to ignore them.**

Case Study: Virtual Machine Appears Hung

Customer Situation

- **Oracle on Linux for system z running on z/VM**
- **External application servers coming into the system z box for authentication and other processing**
- **Periodic delays of a few seconds**
  - Reported between 11:50 and 12:00

Customer Configuration

- **System**
  - 19 Logical IFLs in shared LPAR
  - 30 Physical IFLs
  - 194GB cstore / 4GB xstore
- **Virtual Machine**
  - 6 virtual processors
  - 18GB

FCX162 USERLOG Report

```
 <----- CPU Load -----> <------ Virtual IO/s ------>
Interval       <-Seconds->   T/V
End Time  %CPU  TCPU  VCPU Ratio Total DASD Avoid Diag98   UR
>>Mean>>  92.0 55.18 50.79   1.1    .0   .0    .9     .0   .0
11:50:05  47.9 28.75 27.95   1.0    .0   .0    .0     .0   .0
11:51:05  44.9 26.94 26.15   1.0    .0   .0    .0     .0   .0
11:52:05  48.1 28.87 28.05   1.0    .0   .0    .0     .0   .0
11:53:05  48.9 29.33 28.53   1.0    .0   .0    .0     .0   .0
11:54:05  51.9 31.12 30.30   1.0    .0   .0    .0     .0   .0
11:55:05  47.1 28.25 27.45   1.0    .0   .0    .0     .0   .0
11:56:05  47.7 28.63 27.83   1.0    .0   .0    .0     .0   .0
11:57:05  49.3 29.58 28.76   1.0    .0   .0    .0     .0   .0
11:58:05   244 146.3 127.9   1.1    .0   .0    .2     .0   .0
11:59:05   399 239.6 206.5   1.2    .0   .0    .2     .0   .0
12:00:05   446 267.5 232.2   1.2    .0   .0    .0     .0   .0
12:01:05   451 270.8 235.6   1.1    .0   .0    .0     .0   .0
12:02:05   240 144.0 131.0   1.1    .0   .0    .3     .0   .0
```

- See CPU time take off at 11:58:05
- I/O is not traditional SSCH – so no reflected here

FCX163 UPAGELOG Report

Data	Paging Activity/s								Number of Pages							
Interval	Spaces	<Page Rate>		Page	<--Page Migration-->						<-Resident->		<--Locked-->			
End Time	Owned	Reads	Write	Steals	>2GB>	X>MS	MS>X	X>DS	WSS	Resrvd	R<2GB	R>2GB	L<2GB	L>2GB	XSTOR	DASD
>>Mean>>	.0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	268	0	0
11:49:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	199	0	0
11:50:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	295	0	0
11:51:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	247	0	0
11:52:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	279	0	0
11:53:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	231	0	0
11:54:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	263	0	0
11:55:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	215	0	0
11:56:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	215	0	0
11:57:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	215	0	0
11:58:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	314	0	0
11:59:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	250	0	0
12:00:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	301	0	0
12:01:05	0	.0	.0	.0	.0	.0	.0	.0	3655k	0	29128	3626k	0	298	0	0

- Zero Paging activity for virtual machine
- Actually, zero paging activity for system during these times.
- The virtual machine is entirely resident

FCX162 USTATLOG

Interval									<-SVM and->							
End Time	%ACT	%RUN	%CPU	%LDG	%PGW	%IOW	%SIM	%TIW	%CFW	%TI	%EL	%DM	%IOA	%PGA	%LIM	%OTH
>>Mean>>	100	33	1	0	0	0	1	64	0	0	0	0	0	0	0	0
11:50:05	100	28	0	0	0	0	0	72	0	0	0	0	0	0	0	0
11:51:05	100	35	0	0	0	0	0	65	0	0	0	0	0	0	0	0
11:52:05	100	27	0	0	0	0	2	72	0	0	0	0	0	0	0	0
11:53:05	100	23	2	0	0	0	0	73	0	0	0	0	0	0	0	2
11:54:05	100	32	2	0	0	0	2	64	0	0	0	0	0	0	0	0
11:55:05	100	23	0	0	0	0	2	75	0	0	0	0	0	0	0	0
11:56:05	100	18	2	0	0	0	3	77	0	0	0	0	0	0	0	0
11:57:05	100	28	0	0	0	0	2	70	0	0	0	0	0	0	0	0
11:58:05	100	58	3	0	0	0	3	35	0	0	0	0	0	0	0	0
11:59:05	100	87	5	0	0	0	3	0	5	0	0	0	0	0	0	0
12:00:05	100	92	7	0	0	0	2	0	0	0	0	0	0	0	0	0
12:01:05	100	93	3	0	0	0	3	0	0	0	0	0	0	0	0	0
12:02:05	100	62	2	0	0	0	2	35	0	0	0	0	0	0	0	0

- Sometimes the magnitude isn’t as interesting as the pattern.
- Look at %CFW and %OTH

Reorder Processing - Background

- ***Page reorder* is the process in z/VM of managing user frame owned lists as input to demand scan processing.**
  - It includes resetting the HW reference bit.
  - Serializes the virtual machine (all virtual processors).
  - In all releases of z/VM
- **It is done periodically on a virtual machine basis.**
- **The cost of reorder is proportional to the number of resident frames for the virtual machine.**
  - Roughly 130 ms/GB resident
  - Delays of ~1 second for guest having 8 GB resident
  - This can vary for different reasons +/- 40%

Reorder Processing - Diagnosing

- **Performance Toolkit**
  - Check resident page fields (“R<2GB” & “R>2GB”) on FCX113 UPAGE report
    - Remember, Reorder works against the resident pages, not total virtual machine size.
  - Check Console Function Mode Wait (“%CFW”) on FCX114 USTAT report
    - A virtual machine may be brought through console function mode to serialize Reorder. There are other ways to serialize for Reorder and there are other reasons that for CFW, so this is not conclusive.
- **REORDMON**
  - Available from VM Download Page http://www.vm.ibm.com/download/packages/
  - Works against raw MONWRITE data for all monitored virtual machines
  - Works in real time for a specific virtual machine
  - Provides how often Reorder processing occurs in each monitor interval

REORDMON

```
 Num. of  Average   Average
Userid   Reorders Rsdnt(MB) Ref'd(MB) Reorder Times
-------- -------- --------- --------- -------------------
LINUX01         2     18356     13090  15:59:05 16:15:05
LNXFIN01        1     14277      5207  16:29:05
LNXICR01        1      5107      2991  15:58:05
LNXIPO01        2     21409     14409  15:44:05 16:29:05
LNXNAO01        1     12238      6113  16:02:05
LNXSTM01        2      5615      1956  16:04:05
LNXTLG01        1      9686      5389  15:54:05
LNXUAM01        1      5614      4039  16:11:05
```

- LINUX01 (problem virtual machine) Had two Reorders
- Time’s are GMT so need to subtract 4 hours to match Toolkit Reports
- 15:59:05 matches 11:59:05 where we saw blip in %CFW

Reorder Processing - Mitigations

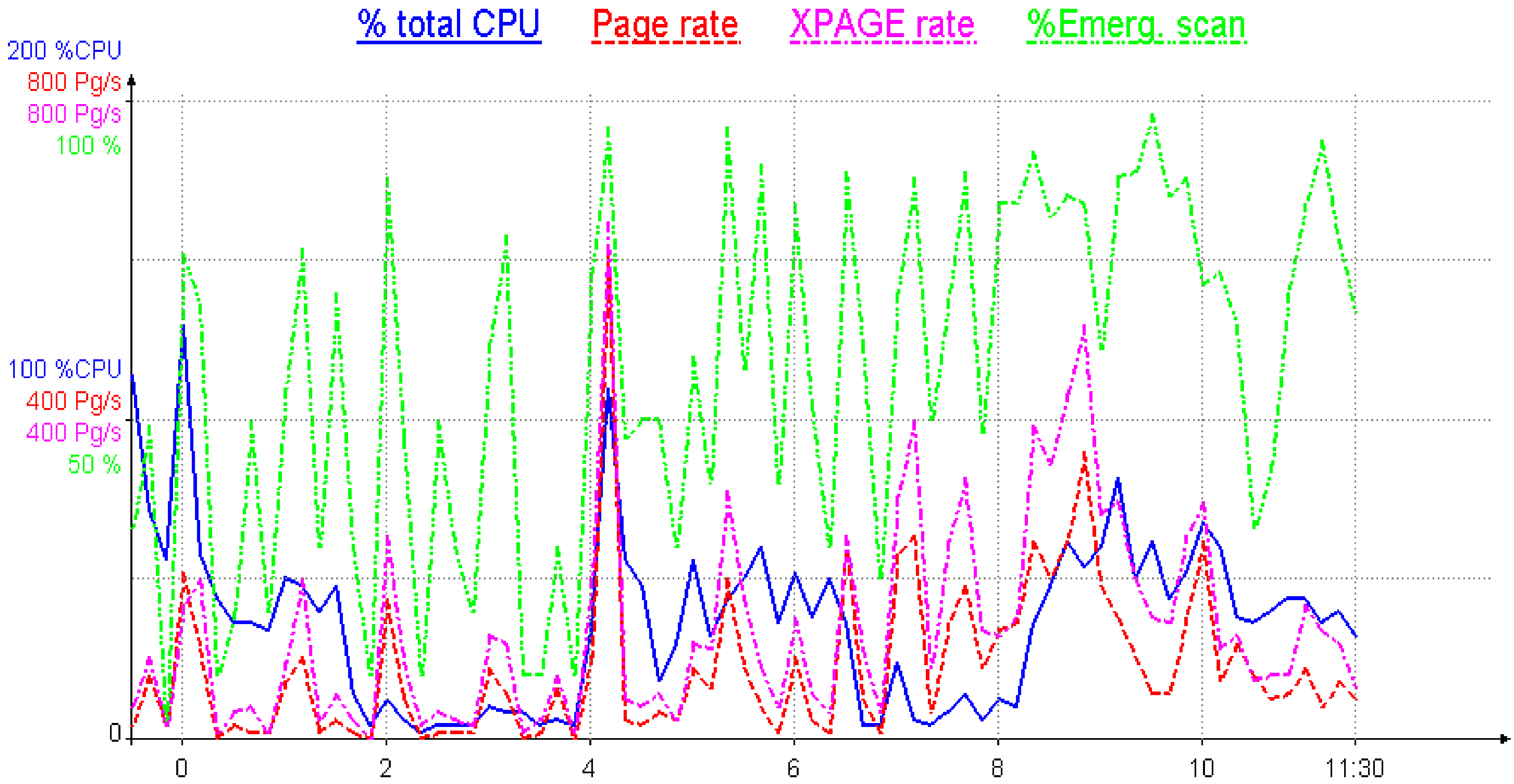
- **Try to keep the virtual machine as small as possible.**
- **Virtual machines with multiple applications may need to be split into multiple virtual machines with fewer applications.**
- **Known requirement at IBM to bring relief in this area: APAR VM64774**
- **See http://www.vm.ibm.com/perf/tips/reorder.html for more details.**

Case Study: Emergency Scan

Question from Customer

- **My system seems to have a high percentage of emergency scan**
- **Application performance doesn't seem bothered**
- **Should I be worried?**

Graph from Customer



Source data: Storage

Finding a Memory Frame

- **Pass 1: tries to be friendly to dispatched users**
  - Unreferenced shared address space pages
  - Long-term-dormant users
  - Eligible-list users
  - Dispatch-list users' unreferenced pages down to WSS
- **Pass 2: a little more aggressive… like pass 1 except:**
  - Avoids shared address spaces
  - Will take from dispatch-list users down to their SET RESERVE
- **Emergency scan: anything we can find**
- **Bit of a misnomer**
- **Want to know more? Read the prologue of HCPALD**

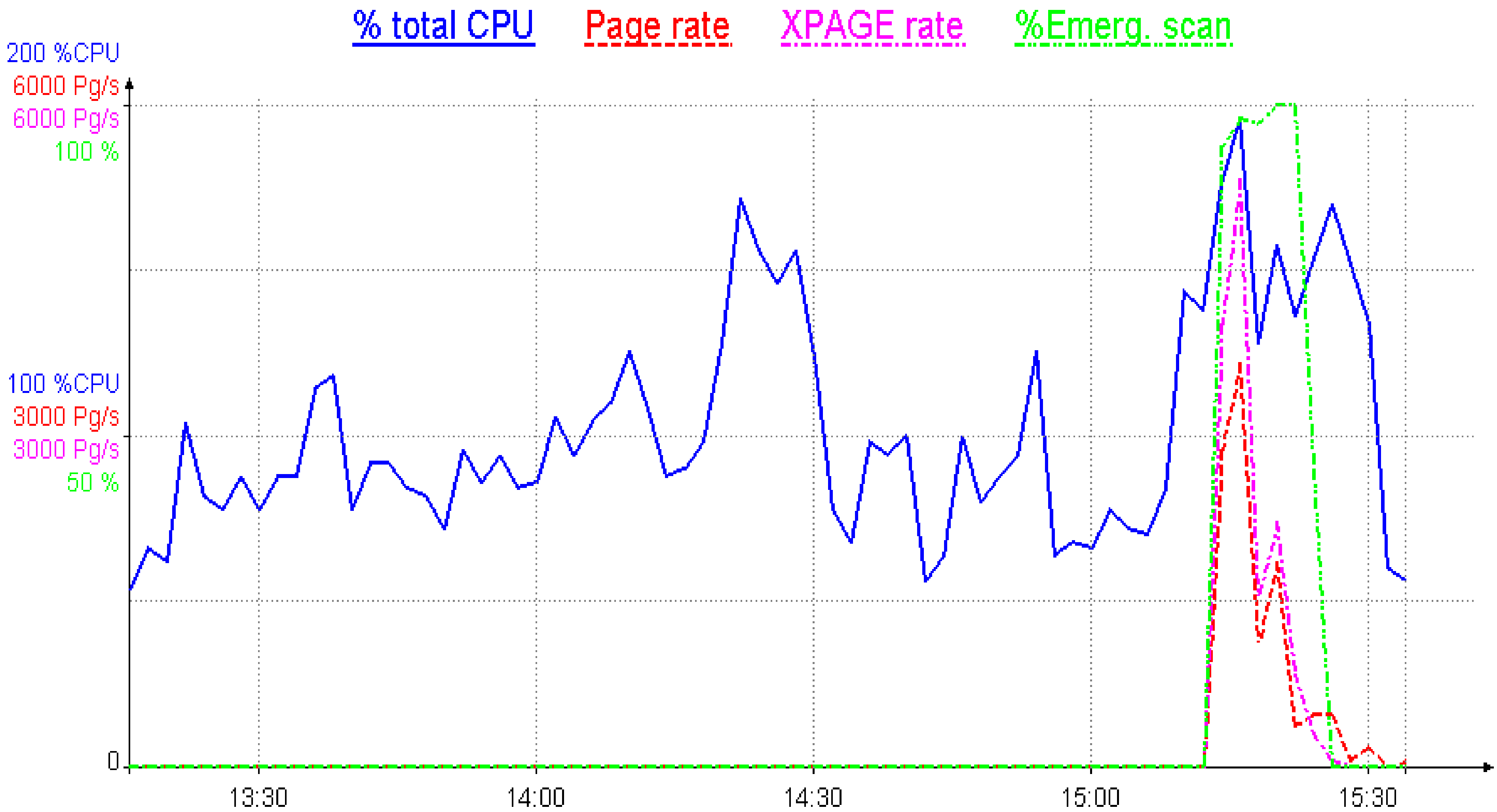
Is Emergency Scan A Sign of Duress?

- **Not alone, no.**
- **Evaluate some other things too.**
  - Are free frame lists routinely zero? (FCX254 AVAILLOG)
  - Is system T/V high? (FCX225 SYSSUMLG)
  - Are we spinning significantly on any locks? (FCX265 LOCKLOG)
  - Does USTAT show users in page wait? (FCX114 USTAT)
  - Is an eligible list forming? (FCX100 CPU)
  - Are MDC hits satisfactory? (FCX103 STORAGE, FCX108 DEVICE)
  - Do you have plenty of SXS space? (FCX264 SXSUTIL)
  - Is DASD page rate > XSTORE page rate? (FCX143 PAGELOG)
  - Are there queues at paging DASD? (FCX109 DEVICE CPOWNED)
  - Is paging MLOAD OK? (FCX109 DEVICE CPOWNED)
  - Is paging blocking factor OK? (FCX103 STORAGE)
  - Is paging space too full? (FCX109 DEVICE CPOWNED)
  - Does application performance seem OK? (you tell me)

Storage Management and VDISKs

- **Referenced VDISK pages are avoided in Pass 1**
- **This customer realized he had a lot of VDISK for Linux swap space**
- **If those VDISK pages are used often, they will tend to stick and be ejectable by only emergency scan**
- **Hmm, customer tried an experiment…**

Customer Removed His VDISKs



Source data: Storage

Summary

- **Try to look at system as a whole**
- **Whether applications seem debilitated is the best indicator of whether the system is suffering**