z/VM 6.4 Performance Update

Version 2018-02-18.1

Brian K. Wade, Ph.D.
IBM z/VM Performance
bkw@us.ibm.com

Credits

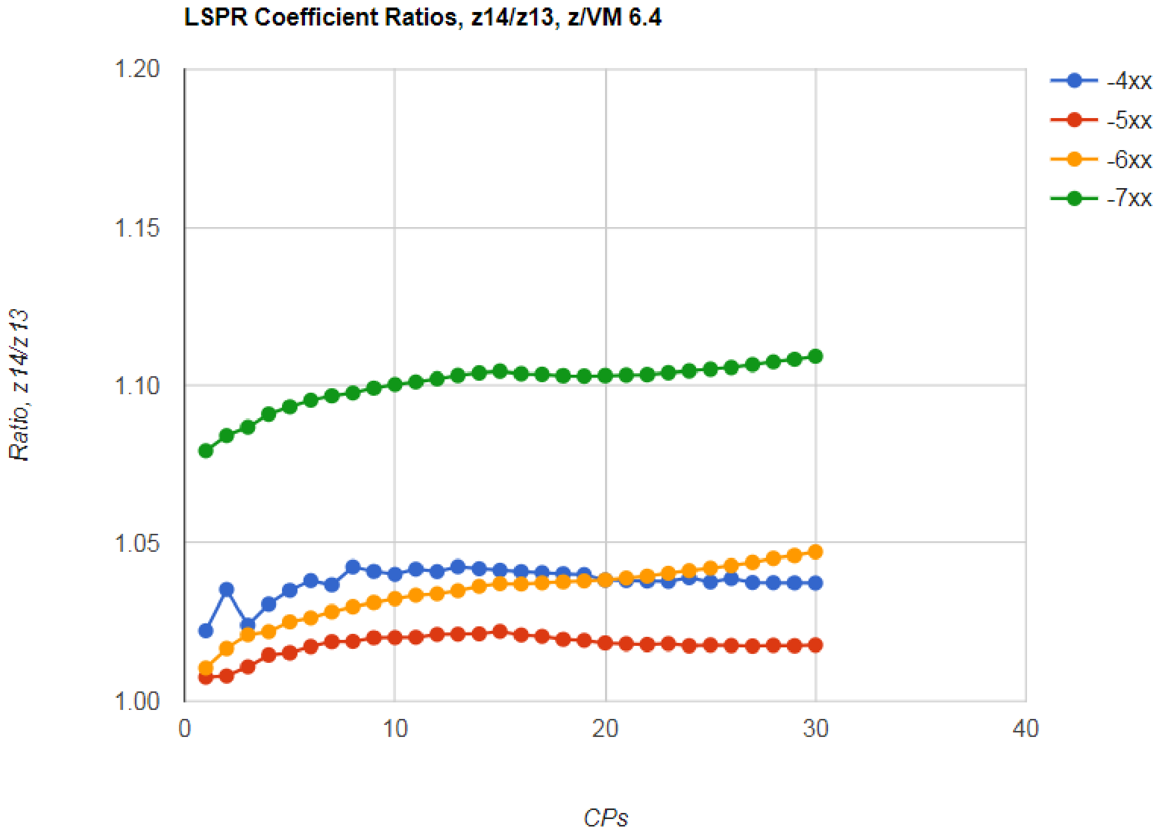
- Your z/VM 6.4 Performance team:
  - Dean DiTommaso
  - Bill Guzior
  - Steve Jones
  - Virg Meredith
  - Bernd Nitsch
  - Dave Spencer
  - Marek Szermutzky
  - Eric Thornton
  - Xenia Tkatschow
  - Brian Wade
- The z/VM 6.4 Performance Report:
  - Dean DiTommaso
  - Steve Jones
  - Bob Neill
  - Patty Rando
  - Ann Shepherd
  - Dave Spencer
  - Eric Thornton
  - Xenia Tkatschow
  - Brian Wade
- Your z/VM 6.4 Performance Toolkit team:
  - Maxim Bochagov
  - Victor Boyarintsev
  - Rob van der Heij
  - Don McGlynn
  - Bob Neill
  - Mikhail Podmarkov
  - Ann Shepherd
- This chart deck:
  - Steve Jones
  - Xenia Tkatschow
  - Brian Wade
  - Eric Thornton
- Thanks also to anyone we inadvertently failed to mention

Agenda

- z/VM 6.4 on z14, performance
- z/VM 6.4, regression performance
- Performance of new functions
  - Dynamic SMT
  - Memory scaling and the 2 TB support limit
  - HyperPAV and zHPF paging
  - CP scheduler improvements
  - Impact of the changed default for the SSL cipher
  - RSCS TCPNJE encryption
- z/VM 6.3 performance APARs that are in z/VM 6.4
- z/VM 6.4 performance APARs
- Small performance fixes in z/VM 6.4
- z/VM Performance Toolkit changes
- z/VM 6.4 1Q17 performance update
  - Concurrent I/O support for XIV EDEVs
  - Dump channel program improvements
  - CRYPTO APVIRT support
  - Encrypted paging
  - Pause-reduced garbage collection
- z/VM 6.4 3Q17 performance update
- z/VM 6.4 4Q17 performance update
- z/VM 6.4 1Q18 performance update
- Summary
- Appendix: Monitor record changes

z/VM 6.4 on z14, Performance

z/VM 6.4 on z14



Per-core capacity statements:

1. From z13 non-SMT to z14 non-SMT is in the range of 6% to 17% (high end did better) with an average of 10%

2. From z13 SMT-2 to z14 SMT-2 is expected to be in the neighborhood of 15%

3. From z14 non-SMT to z14 SMT-2 is expected to be in the range of 10% to 40%, with an average of 25%

Sources:
1. https://www-304.ibm.com/servers/resourcelink/lib03060.nsf/pages/lsprITRzVMv6r4?OpenDocument
2. https://www.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=ZSD03046USEN&

z/VM Support for the z14

- Refer to http://www.vm.ibm.com/service/vmreqz14.html

Regression Performance

z/VM 6.4 Regression Performance

- We ran about 120 scenarios:
  - Some non-SMT, some SMT-2
  - Some using Apache static file web serving in various ways
  - Some using our VIRSTOR load generator
  - Some using DayTrader (a WAS and DB/2 workload)
  - Some storage-rich, some storage-constrained
  - Some 1-core, some 3-core, some mid-sized, and some as large as 64-core
  - Most on z13, but some on zEC12

- z/VM levels we used:
  - Base runs were done on z/VM 6.3 plus all closed PTFs as of March 31, 2016
  - Comparison runs were done on the z/VM 6.4 *code freeze* driver of August 15, 2016

- Typical measures of accomplishment:
  - ETR (external transaction rate): units of application work per second
  - ITR (internal transaction rate): what ETR would scale to if the LPAR could run this workload completely busy

- Our findings:
  - ETR ratios, comparison/base: mean (μ) = 1.10, standard deviation (σ) = 0.24
  - ITR ratios, comparison/base: μ = 1.15, σ = 0.37

- Notes:
  - Storage-constrained workloads got the benefits of the storage management and paging line items
  - VSwitch-intensive workloads got the benefits of some things we fixed along the way
  - Most other workloads had ratios close to 1

New Function

Dynamic SMT

- In z/VM 6.3 1Q15, the SMT level – non-SMT, SMT-1, or SMT-2 – was chosen in the system configuration file
- In z/VM 6.4, if in the system configuration file you chose SMT-x, you can then switch between SMT-1 and SMT-2 without an IPL
- We measured z/VM 6.4 SMT-1 compared to z/VM 6.4 non-SMT and found no difference
  - So you can feel confident about IPLing z/VM 6.4 in SMT-1 and then using z/VM 6.4’s new command CP SET MULTITHREADING to try SMT-2
- Mixed-engines note: it’s still only the logical IFL cores that run in SMT-2
- Remember the z/VM limits on logical *cores*:
  - Logical *cores* are what you define in the LPAR’s activation profile
  - Non-SMT: up to 64 logical cores are permitted in the LPAR
  - SMT-x: only the first 32 logical cores of the LPAR will be used (the rest are ignored)
- Remember: to switch between non-SMT and one of the SMT modes, you must change the system configuration file and re-IPL
- Let us know how this works for you
- Remember to collect application performance data and MONWRITE data

Non-SMT, SMT-1, and SMT-2

```
Assuming the LPAR is entirely IFL:

Non-SMT:
Core IDs ->     0    1    2    3    4    5    6    7
CPU IDs  ->     0    1    2    3    4    5    6    7

SMT-1:
Core IDs ->     0    1    2    3    4    5    6    7
CPU IDs  ->     0    2    4    6    8    A    C    E

SMT-2:
Core IDs ->     0    1    2    3    4    5    6    7
CPU IDs  ->     0 1  2 3  4 5  6 7  8 9  A B  C D  E F
```

When you use CP SET MULTITHREADING to change the SMT level, you will see logical CPUs come and go.

Memory Scaling and the 2 TB Limit

- z/VM 6.4 raises the supported central storage limit to 2 TB
- To get there we needed to do some things
  - We changed the frame manager to make it more open to concurrency
  - Central storage is now divided into *zones*
    - There is a spin lock associated with each zone
    - For starting point, there is an affinity of logical CPUs to zones
  - Frame returns can queue instead of waiting for the zone's lock
  - Frame manager can take reclaimable frames from the global available list without waiting for demand scan to do it

  - We changed allocations from PTRM (Page Table Resource Manager) address spaces so they are more amenable to concurrency
- These changes will be most relevant for customers trying to grow beyond 1 TB central, especially with large numbers of logical processors in the LPAR

Memory Scaling: Effect of the Changes

- Workloads we used:
  - A stress workload that was specifically crafted to be extremely difficult for the frame manager – almost no guest (SIE-2) content, high MP level, and very difficult page reference patterns
  - An Apache-web-serving workload that was much more like what a customer's paging-intensive workload might look like
- Findings:
  - The stress workload went from about 54 processors' worth of spin lock time, with 48 processors' worth in real storage locks, to about 8 processors' worth of spin lock time with almost none of it in real storage locks.
  - The Apache workload showed improved scaling beyond 1 TB of central: about 2% to 4% improvement in ETR, and about 2% to 3% improvement in ITR, compared to z/VM 6.3
- Read the article: http://www.vm.ibm.com/perf/reports/zvm/html/640mcr2t.html

Paging Improvements

- General improvements were made to the paging subsystem
  - I/O payloads increased
  - Block paging efficiency increased

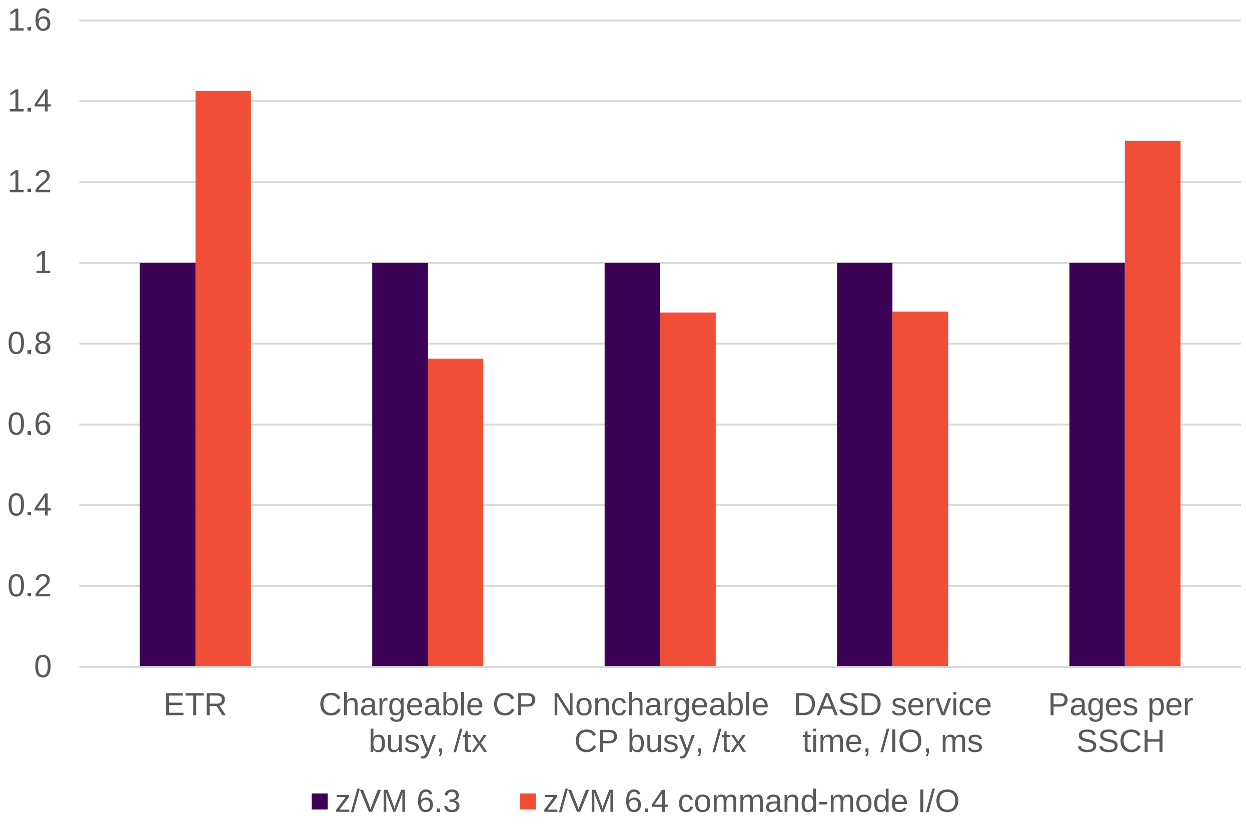
- The paging subsystem can now use zHPF (High Performance FICON, aka transport-mode I/O)
  - CP SET PAGING HPF {ON|OFF}
  - To use zHPF for paging, the FICON has to be FICON Express8 or later

- The paging subsystem can now use HyperPAV aliases
  - CP SET PAGING ALIAS {ON|OFF}

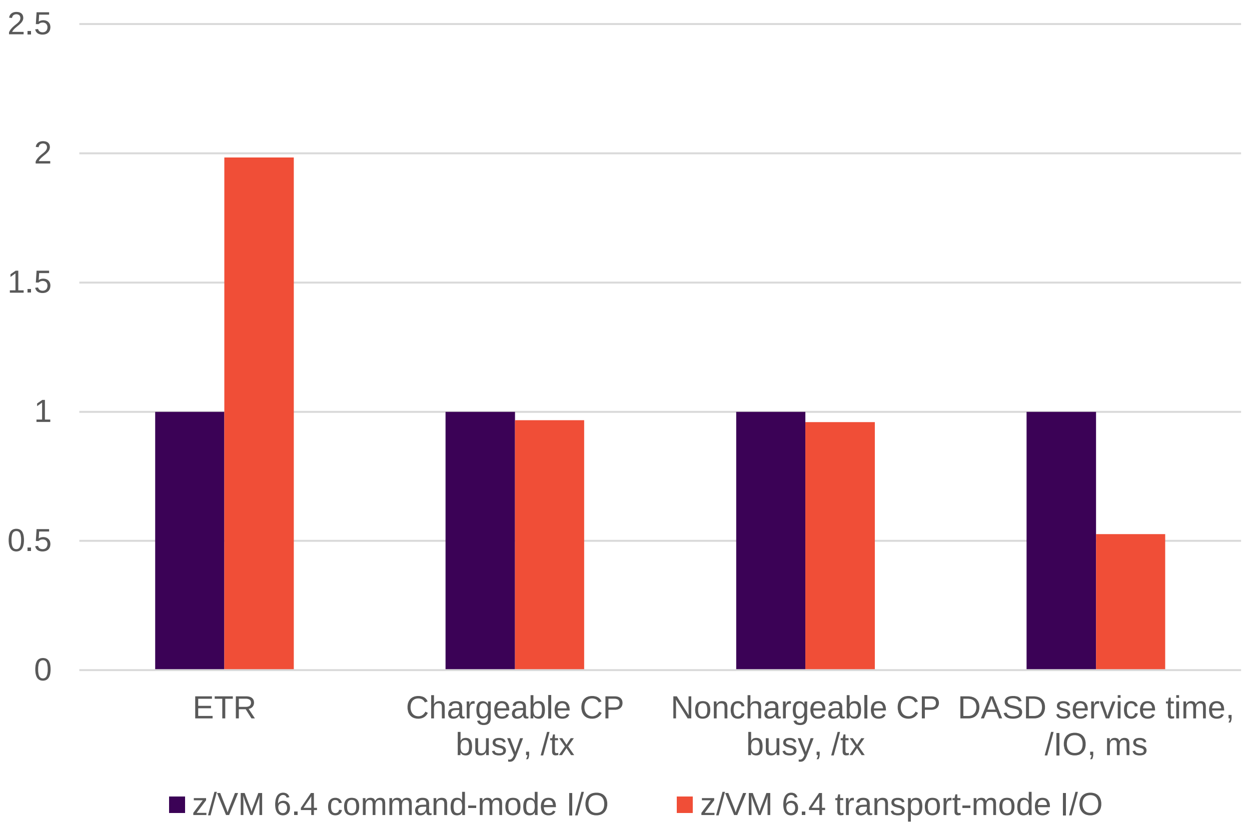
- The improvements were evaluated using a memory-thrashing workload based upon our internal tool called “Virtual Storage Exerciser” (VIRSTOR)

- Read the article: http://www.vm.ibm.com/perf/reports/zvm/html/640hpp.html

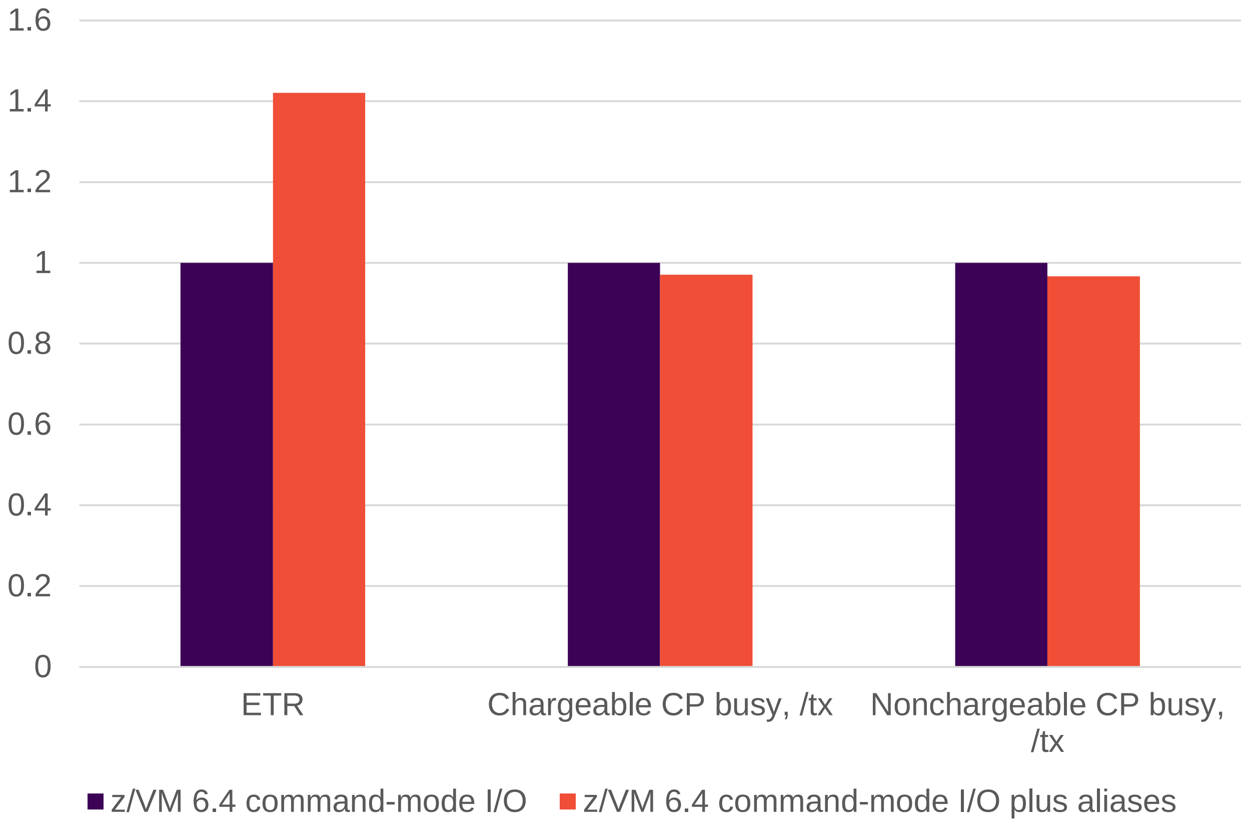
Paging: Effect of General Improvements



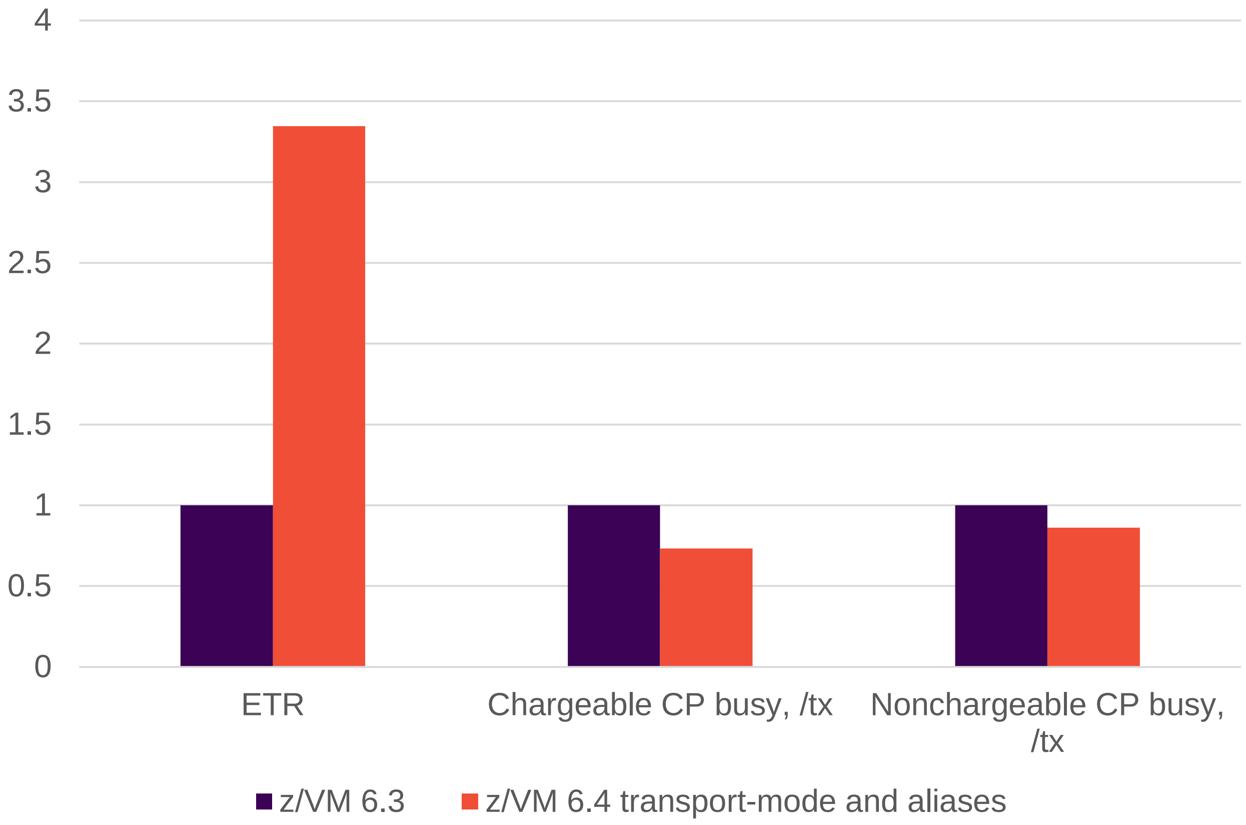
Paging: Effect of zHPF I/O (SET PAGING HPF ON)



Paging: Effect of HyperPAV (SET PAGING ALIAS ON)



Paging: All z/VM 6.4 Improvements in Play



Paging: Should I Exploit HyperPAV Aliases?

- Have I configured extra paging volumes just to achieve paging I/O concurrency, or rather did I really need the space?

- Am I forced to use large (e.g., mod-27) paging volumes, so I really do need >1 I/O in flight to each paging volume concurrently?

- Does Perfkit FCX146 AUXLOG show evidence of queueing?

  Does Perfkit FCX109 DEVICE CPOWNED show evidence of queueing?
  - (Note: on z/VM 6.3, paging I/O queueing does not show up in FCX108)

- Does Perfkit FCX109 DEVICE CPOWNED show elevated MLOAD values?
  - "Good" values here are around 1 msec

- Remember to inspect INTERIM reports or log-style reports to look for peaks that might otherwise go unnoticed

- Classic guidance on paging configuration still applies:
  - Use all the same model of volume (e.g., -9, -27)
  - Spread across chpids, across LCUs, across ranks within real controllers, and across real controllers

Paging: Alias Sharing Within an LCU

- During the design phase there was concern that within a given LCU, paging I/O's use of aliases could dominate the use of aliases, thereby preventing minidisk I/O from using aliases

- The CP SET CU command was changed to let the administrator specify *relative shares* for paging I/O and for minidisk I/O
  - Corresponding change to CU statement of system configuration file

- These relative shares work just like LPAR weights and z/VM relative shares
  - Entitlement, excess distribution, and so on

Suppose the LCU has 15 SYSTEM-attached HyperPAV aliases

Use	Share	Entitlement
Paging I/O	100	15 * 100/300 = 5
Minidisk I/O	200	15 * 200/300 = 10

- Perfkit isn't ready yet, so for now use the HPALIAS package:
  - http://www.vm.ibm.com/download/packages/descript.cgi?HPALIAS

Paging: Alias Sharing Between Minidisk I/O and Paging I/O

Hybrid workload:

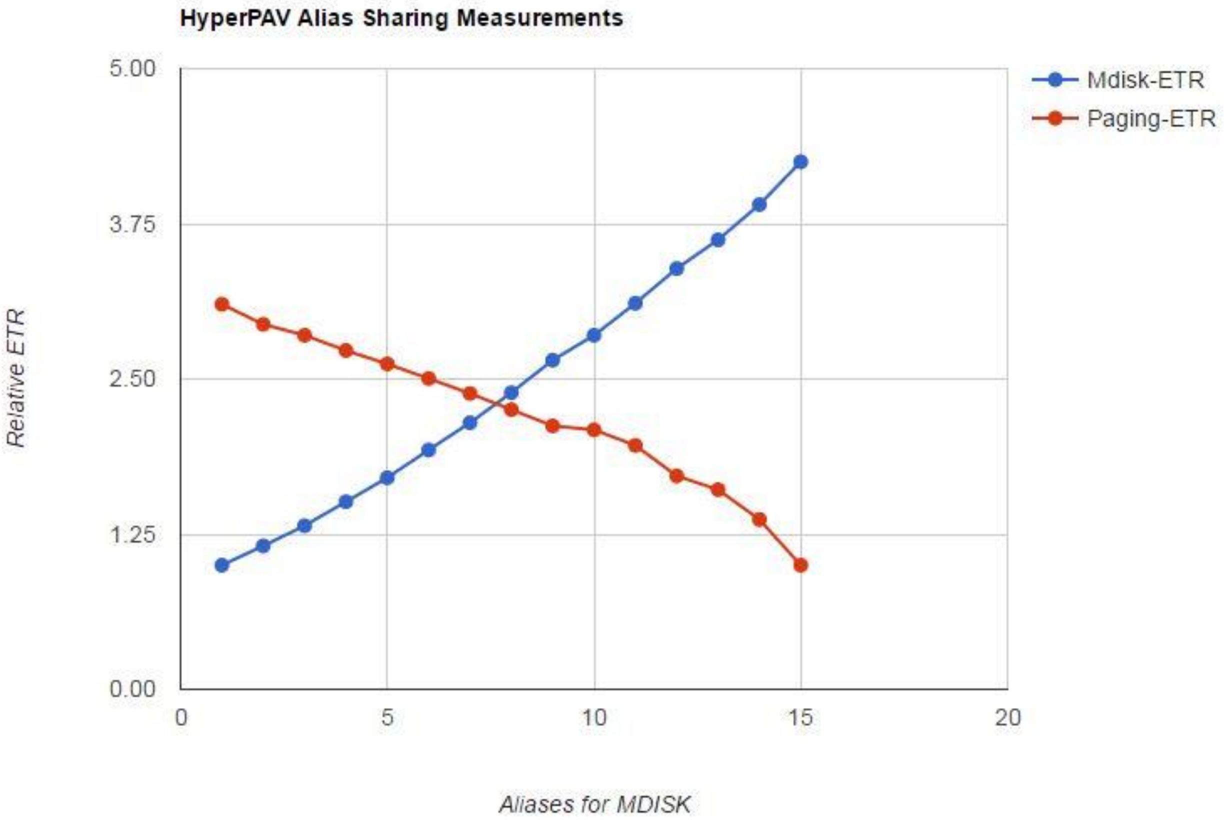
a. A farm of guests protected by SET RESERVE, intensive on minidisk I/O

b. A farm of guests running a page-fault-intensive storage exerciser

c. All DASD in a single LCU

We ran 15 measurements.

We used CP SET CU to “steer” the power of the HyperPAV aliases from minidisk I/O to paging I/O.



Paging: Balancing Alias Use

- FCX108 shows I/O queueing activity (sampled)
- The HPALIAS package shows alias shares, alias usage, and I/Os queued (continuously tracked)
- Only you know which of your application(s) might be suffering because of an alias imbalance

Perfkit FCX108 DEVICE excerpt:

```
.    .                     .      ___     .    .    .    .    .    .    .    .
<-- Device Descr. -->  Mdisk Pa- <-Rate/s-> <------- Time (msec) -------> Req.
Addr Type   Label/ID   Links ths  I/O Avoid Pend Disc Conn Serv Resp CUWt Qued
>> All DASD <<          ....  ..  183    .0 .232 .003 .934 1.17 .826 .078  .34
BE0E 3390-9 QRMD00        32   4 2149    .0 .229 .003 .193 .426 6.57 .000 13.2   minidisk
BE0F 3390-9 QRMD01        32   4 2142    .0 .230 .003 .194 .427 7.04 .000 14.2   minidisk
BE12 3390-9 ATP003         0   4  748    .0 .233 .003 1.06 1.30 1.30 .101  .00   paging
BE14 3390-9 ATP003         0   4  743    .0 .233 .002 1.07 1.30 1.30 .103  .00   paging
BE1F 3390-9 ATP000         0   4  742    .0 .233 .002 1.07 1.31 1.31 .102  .00   paging
BE1A 3390-9 ATP003         0   4  741    .0 .232 .001 1.08 1.31 1.31 .101  .00   paging
```

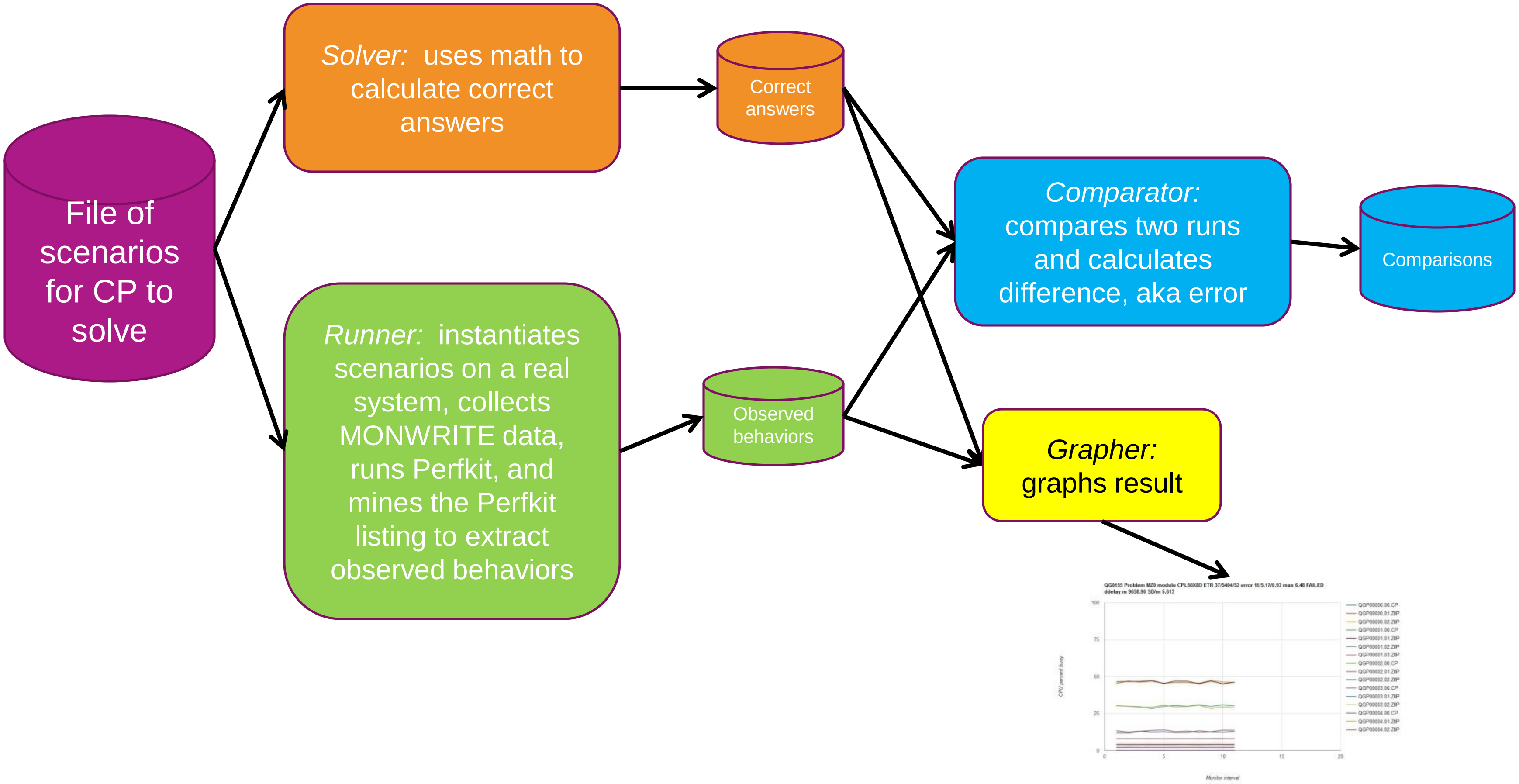
HPALIAS tool excerpt:

```
__ISO-UTC__________ SSID Pool __Type__ _Share__ _EntMnt_ _InUse__ _Queued_
2016-09-01,22:02:18 0600    0 MDISK          20     2.00     2.37    27.18
2016-09-01,22:02:18 0600    0 PAGING        140    14.00    13.63     5.19
2016-09-01,22:03:18 0600    0 MDISK          20     2.00     2.29    27.26
2016-09-01,22:03:18 0600    0 PAGING        140    14.00    13.71     6.18
2016-09-01,22:04:18 0600    0 MDISK          20     2.00     2.32    27.21
2016-09-01,22:04:18 0600    0 PAGING        140    14.00    13.68     6.31
2016-09-01,22:05:18 0600    0 MDISK          20     2.00     3.99    25.30
2016-09-01,22:05:18 0600    0 PAGING        140    14.00    12.01     3.78
```

CP Scheduler Improvements

- In VM65288 a customer reported CP did not enforce relative share settings
  - For example: four guests, equal relative share, all infinitely hungry, should get equal amounts of CPU time, but they did not
- IBM answered the APAR as *FIN* aka *fixed-if-next*
- While developing z/VM 6.4 we studied the behavior of the scheduler and made several repairs
- In the next few charts we'll show you some scenarios
- Read the article: http://www.vm.ibm.com/perf/reports/zvm/html/640srp.html

How One Tests This



File of scenarios for CP to solve

Solver: uses math to calculate correct answers

Correct answers

Runner: instantiates scenarios on a real system, collects MONWRITE data, runs Perfkit, and mines the Perfkit listing to extract observed behaviors

Observed behaviors

Comparator: compares two runs and calculates difference, aka error

Comparisons

Grapher: graphs result

QG0155 Problem MZ0 module CPL50X8D ETR 37/5404/52 error 11/5.17/0.93 max 6.48 FAILED
ddelay m 9658.90 SD/m 5.613

Problem: Infinite Demand, Unequal Share

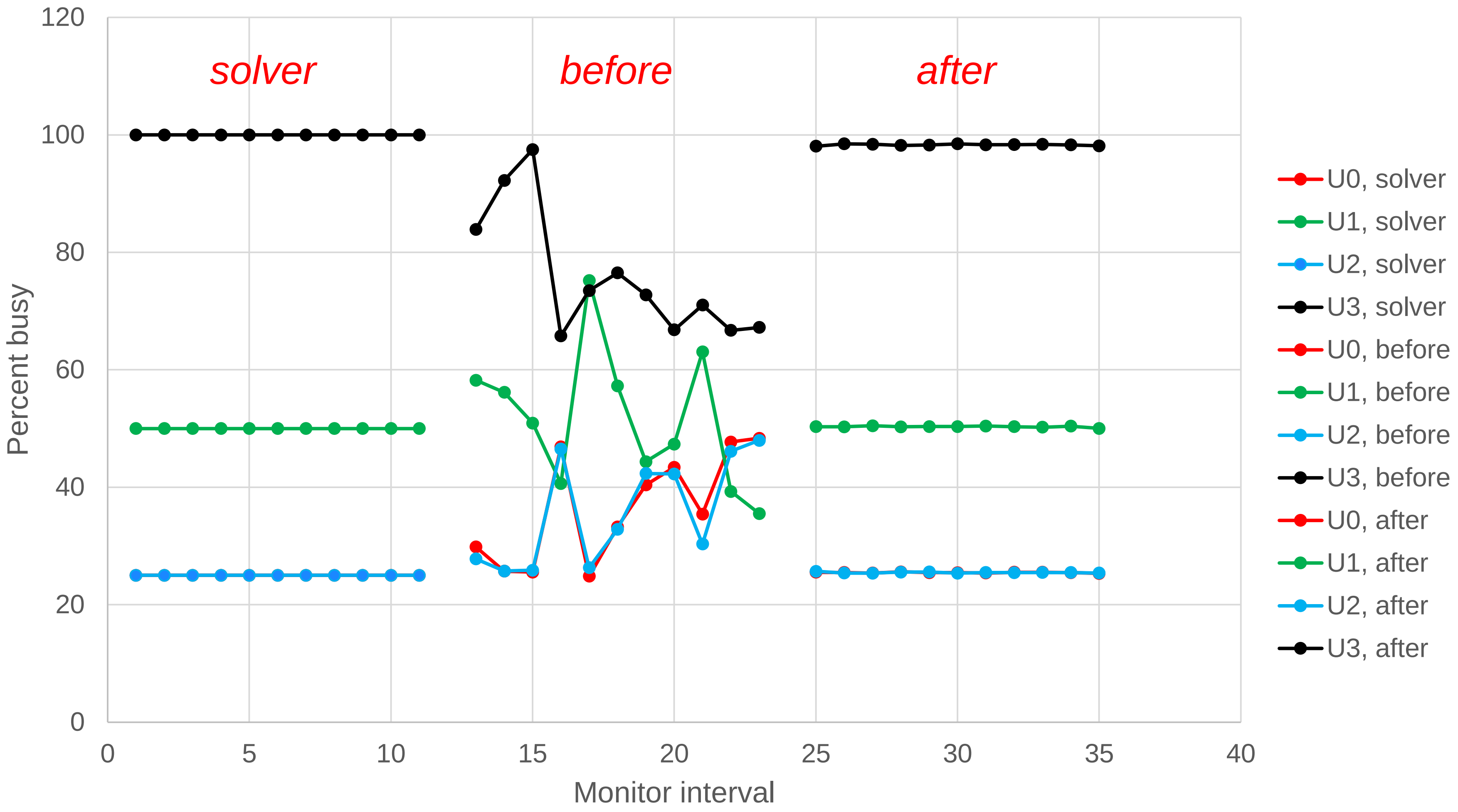
- LPAR has two logical processors, so it has 200% to give

- All four users want as much power as they can get
  - User 0 is relative 100
  - User 1 is relative 200
  - User 2 is relative 100
  - User 3 is relative 400

- Their utilizations should be in ratio 1:2:1:4
  - For a total of 200% this would be 25%, 50%, 25%, 100%   (sum = 200%)

- Let’s see what happened:
  - What the solver said
  - What happened before the fix
  - What happened after the fix

Infinite Demand, Unequal Share: Before and After



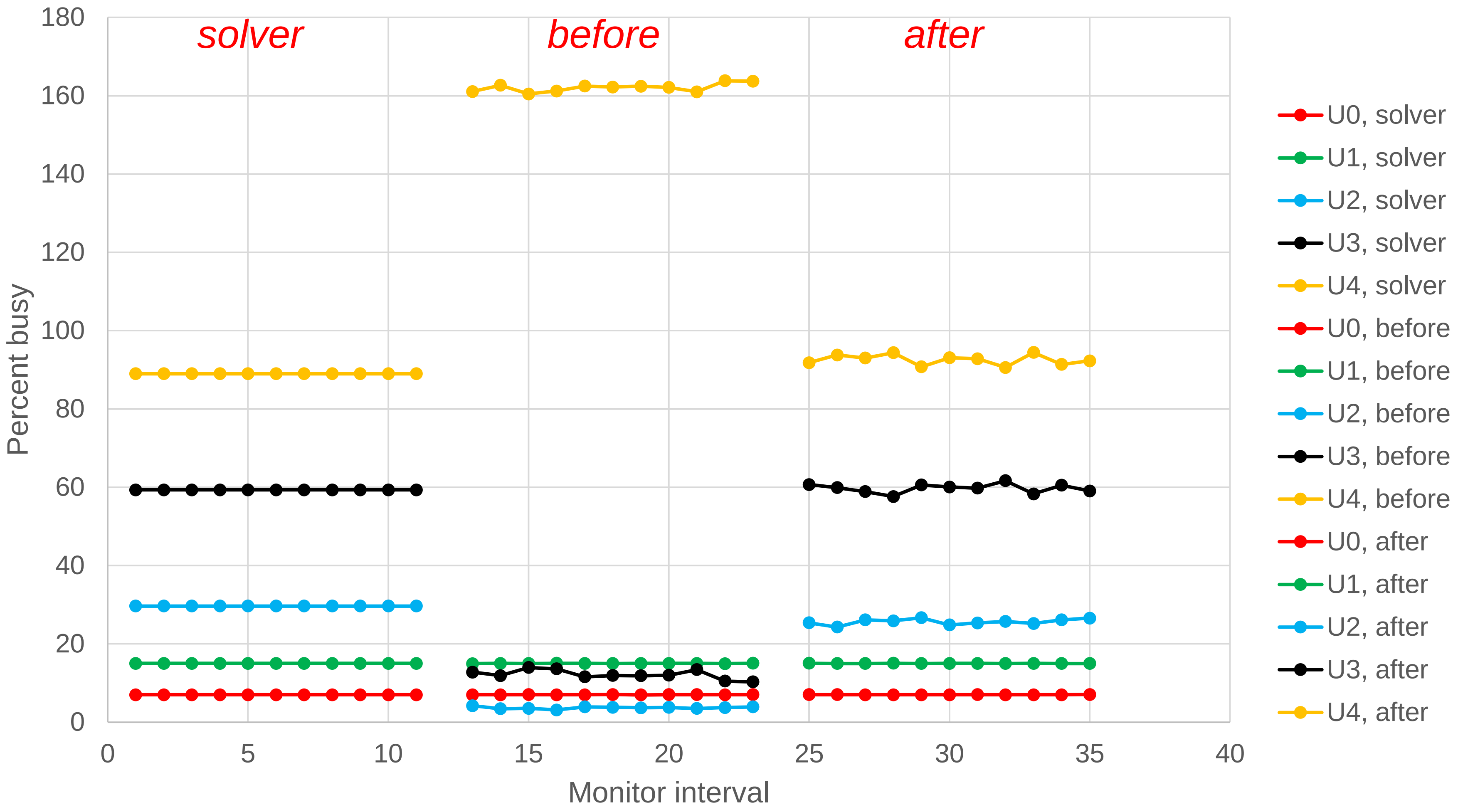
Problem: Distribution of Excess

- LPAR has two logical processors, so it has 200% to give

- Users 0 and 1 have relative 10000 but want almost nothing
- User 2 is relative 100 and wants all he can get
- User 3 is relative 200 and wants all he can get
- User 4 is relative 300 and wants all he can get

- CP should distribute User 0’s and User 1’s excess to Users 2, 3, and 4 in correct proportion
  - Users 2, 3, and 4 each get their entitlement plus their share of the excess
- Let’s see what happened:
  - What the solver said
  - What happened before the fix
  - What happened after the fix

Problem: Distribution of Excess



Changes to SSL

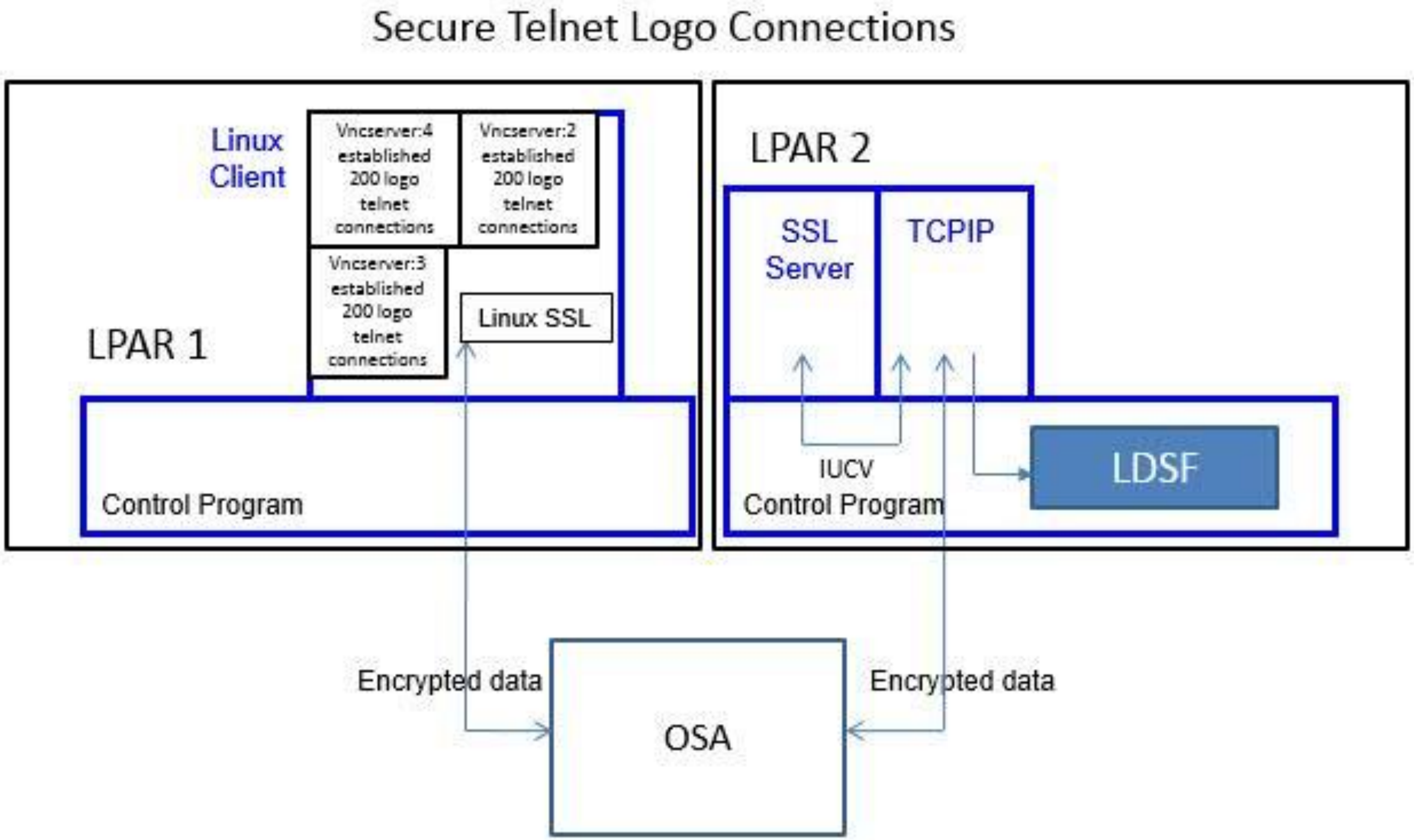
- Which default settings changed?

z/VM	6.3	6.3 + APARs (1)	6.4
TLS	1.0	1.2	1.2
Cipher	RSA_AES_256	RSA_AES_256	RSA_AES_128_SHA256
System SSL	V1.13	V2.1	V2.2

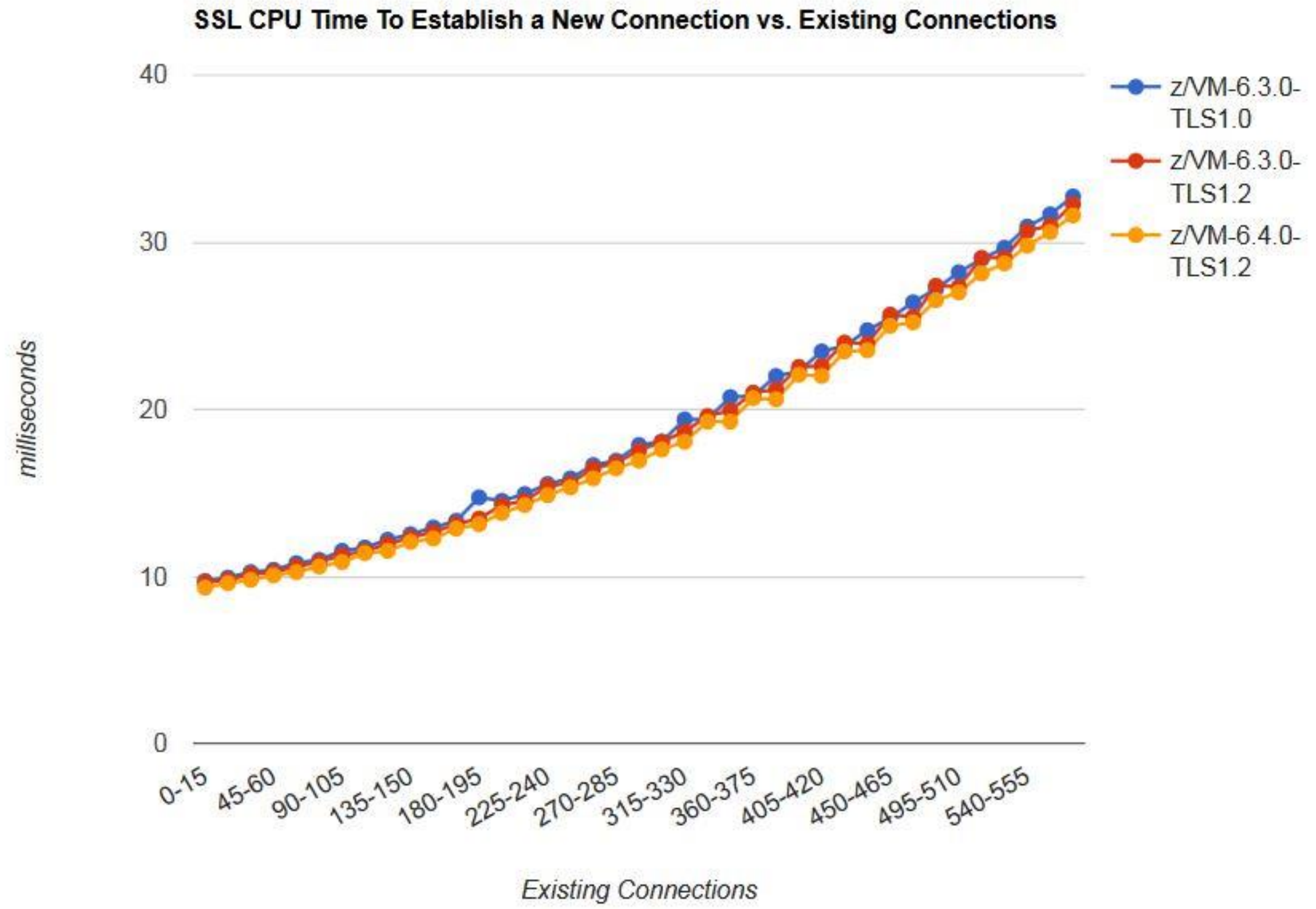
- Two scenarios studied:
  - 600 remote Linux Telnet connections established
  - 200 remote Linux Telnet connections doing data transfer

(1) PI40702 to TCP/IP, VM65717 to CMS, and VM65718 to LE.

Changes to SSL: Establishing Telnet Connections

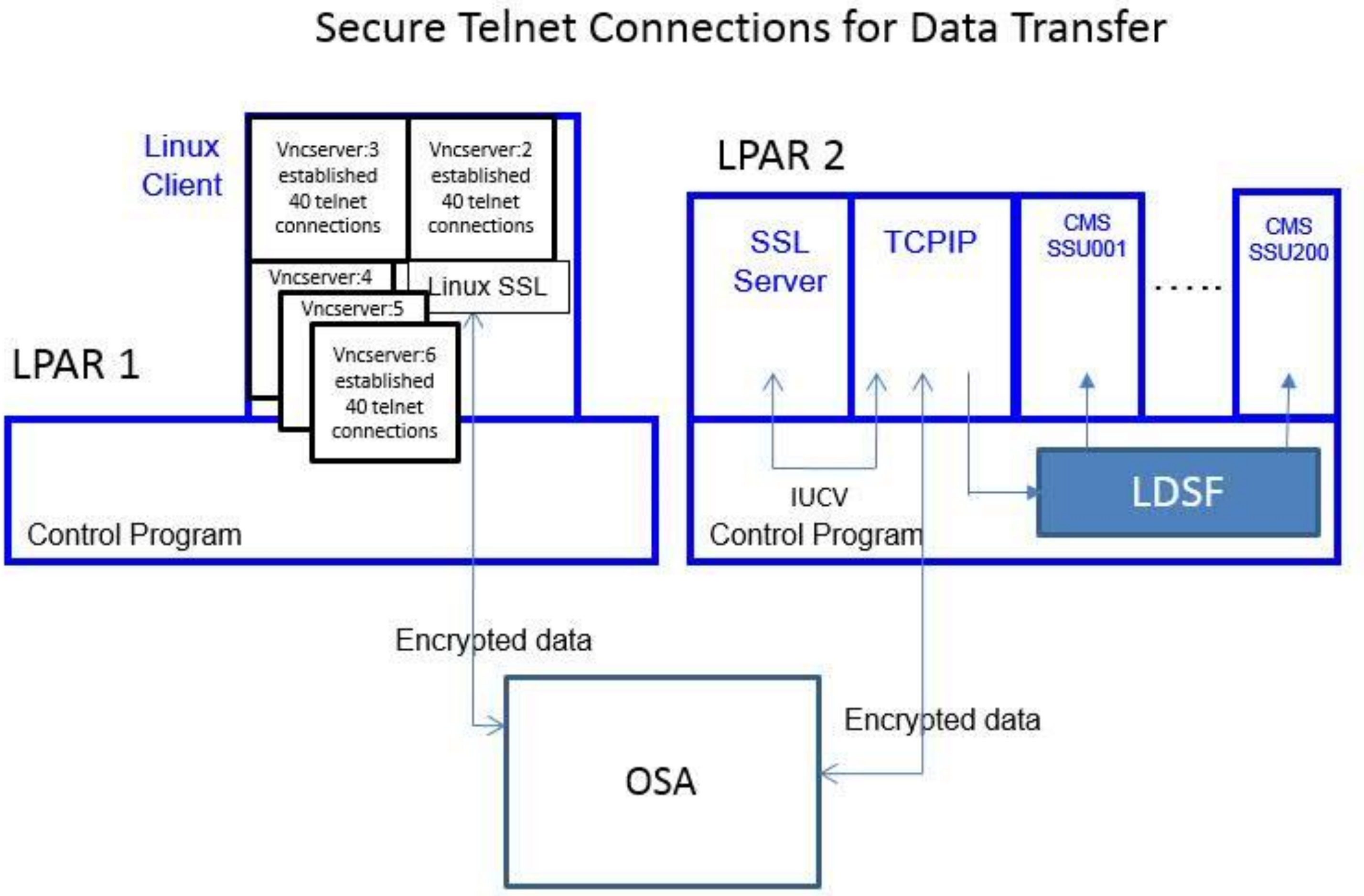

Secure Telnet Logo Connections
Linux Client
Vncserver:4 established 200 logo telnet connections
Vncserver:2 established 200 logo telnet connections
Vncserver:3 established 200 logo telnet connections
Linux SSL
LPAR 1
Control Program
LPAR 2
SSL Server
TCPIP
IUCV
Control Program
LDSF
Encrypted data
OSA
Encrypted data

Changes to SSL: 600 Logo Connections Results



Ciphers used: TLS 1.0 was RSA_AES_256; TLS 1.2 was RSA_AES_128_SHA256.

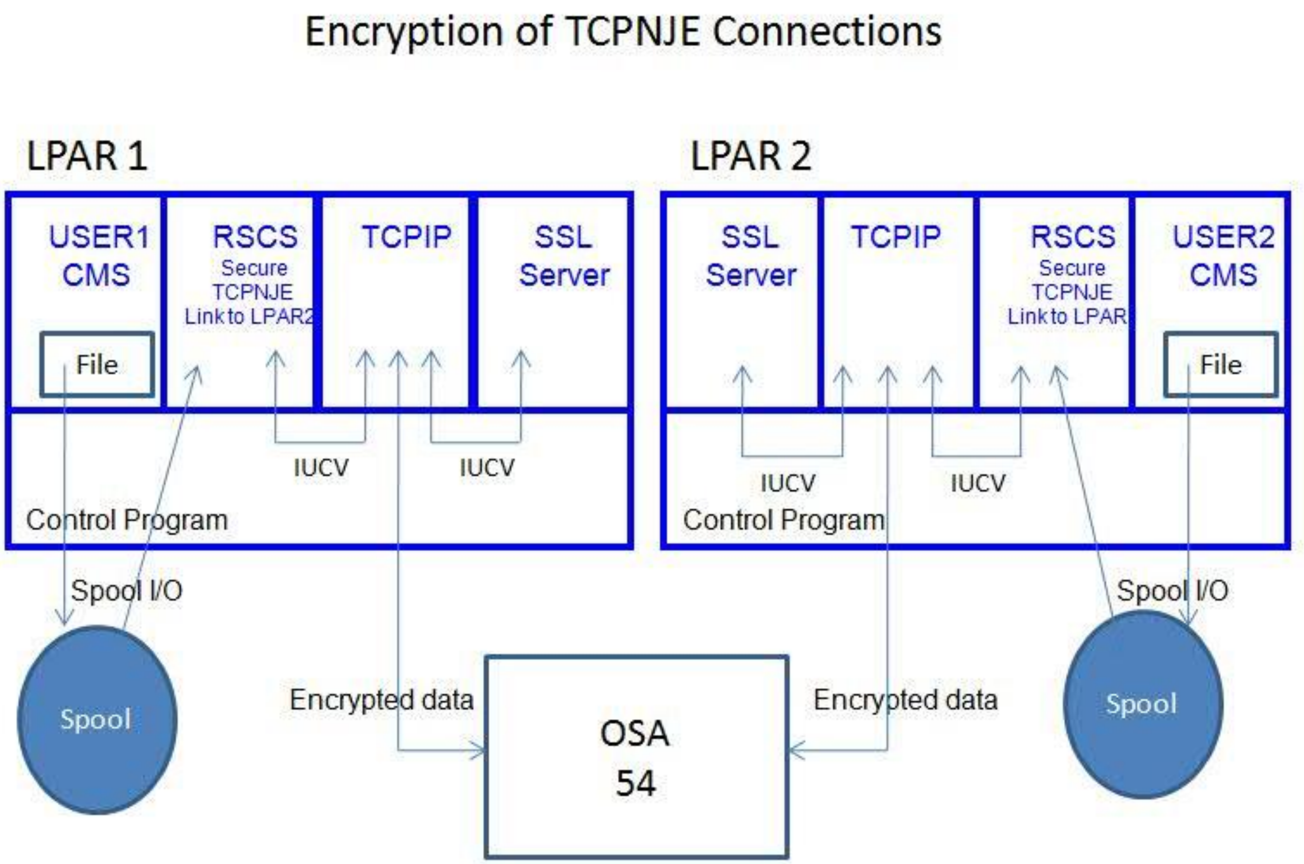
Changes to SSL: Telnet Data Transfer


Secure Telnet Connections for Data Transfer
Linux Client
Vncserver:3 established 40 telnet connections
Vncserver:2 established 40 telnet connections
Vncserver:4
Linux SSL
Vncserver:5
Vncserver:6 established 40 telnet connections
LPAR 1
Control Program
LPAR 2
SSL Server
TCPIP
CMS SSU001
. . . . .
CMS SSU200
IUCV
Control Program
LDSF
Encrypted data
OSA
Encrypted data

Changes to SSL: 200 Telnet Connections Doing Data Transfer

- With z/VM 6.3, results showed a 1.1% decrease in CPU/tx when changing the cipher from RSA_AES_256 to RSA_AES_128_SHA256
- With z/VM 6.4 and SSL V2.2, results showed a 13.6% increase in CPU/tx for the SSL server, compared to z/VM 6.3 and SSl V2.1. Most of the increase was observed within the SSL server. (emulation)
- Read the article: http://www.vm.ibm.com/perf/reports/zvm/html/640cip.html

RSCS TCPNJE Encryption: What We Ran



TCPNJE is the RSCS line driver that uses a TCP connection to move the data.

RSCS TCPNJE Encryption: What We Ran

- All measurements were completed on zEC12 with CPACF support

Case Number	Security	z/VM Level	SSL Version	TLS Protocol	Cipher (default)	Ratio, CPU/tx, incremental	Ratio, CPU/tx, since Case 1
case 1.	not secured	6.3 w/ VM65788	V2.1	na	na	1	1
case 2A.	secured	6.3 w/ VM65788	V2.1	1.0	RSA_AES_256	1.56	1.56
case 2B.	secured	6.3 w/ VM65788	V2.1	1.2	RSA_AES_128_SHA256	0.891	1.39
case 3.	secured	6.4	V2.2	1.2	RSA_AES_128_SHA256	1.105	1.54

- In table above, CPU/tx is summed over RSCS, TCP/IP, and the SSL server
- Read the article: http://www.vm.ibm.com/perf/reports/zvm/html/640nje.html

APARs and Small Fixes

Performance-related APARs Against z/VM 6.3 Since 1Q15

- VM64587: VDISK pages not stolen aggressively enough
- VM64770: The read-in of guest PGMBKs at logoff was inefficient
- VM64890: A bad loop counter caused excessive CPU in MDC
- VM64941: Guest's view of storage key change bit sometimes wrong for IBR page
- VM65097: PGMBK prefetch not applicable when Diag x'10's are for single pages
- VM65101: IBR pages on GAL unnecessarily rewritten
- VM65189: Excess storage management work stacked on SYSTEMMP
- VM65199: Master CPU stuck doing SYSTEMMP work; some eligible CPUs not signalled to do SYSTEMMP work
- VM65420: Frames that should have been stolen from MDC were not being stolen
- VM65692: FII intercept bit set by mistake caused excess simulation overhead
- VM65709: MDC processing being done even though MDC-inhibit flag was set
- VM65748: High Performance FICON features unavailable to guests
- VM65762: CP fails to deliver PCI thin interrupts to guests
- VM65794: MDC fails to work for RDEVs with device number > x'8000'
- VM65801: CP excessively redrives VSwitch uplink port
- VM65820: PGMBK reclaim exits without releasing lock, prohibiting further reclaim
- VM65824: SET MDC OFF for an RDEV inadvertently turned MDC back on
- VM65837: DASD recovery I/O queued at inopportune moment causes DASD I/O to stall
- VM65845: HyperSwap occurring at inopportune moment causes DASD I/O to stall
- VM65869: Excessive LOGOFF delay for QDIO-exploitive guests

These are all in z/VM 6.4.

Performance-related APARs Against z/VM 6.4

- VM65885 (1601): Perfkit needs deprecated HPF monitor fields
- VM65916: HiperSockets Guest LAN NIC lost initiative
- VM65992 (1701): HiperSockets performance issue on short busy
- VM65985 (1701): PRG004 or hang when MDC enabled for volumes used for z/OS guests
- VM66016: Abend during zHPF paging error recovery
- VM65644 (1701): SCSI monitor fields not filled
- VM65946: SECUSER output is slow
- VM65998 (1701): crypto polling too frequent
- VM65741 (1701): make all 3390-A eligible for MDC
- VM65886 (1601): CCW fast-trans incorrectly marked minidisk I/O as ineligible for HyperPAV aliases
- VM66026 : Monitor enhancements for HyperPAV and PAV aliases (went PE: also apply VM66036)
- VM65979: Removed unnecessary MDC purge done during HyperSwap
- VM65942: z14 support (includes new Monitor counters for priv ops by VCPU) (went PE: also apply VM66071)

(xxxx) is RSU number

Small Performance Fixes in z/VM 6.4

- **Excessive SCSI retries:** during IPL from a LUN, offline paths would cause delays. Fixed.
- **Lock hierarchy violation:** slowed down processing in VSwitch. Fixed.
- **Storage leak in QUERY PROCESSORS:** could slow the system if hit enough times. Fixed.
- **Unnecessary VSwitch redrives:** unnecessary redrives of bridge port or uplink port in some situations. Fixed.
- **Unnecessary emergency replenishment:** unnecessary frame table scans in some situations. The scans could hang the system. Fixed.
- **Incorrect dispatcher settings for SMT:** z/VM 1Q15 dispatcher accommodations made for SMT should have been in effect for only SMT-2. Fixed.
- **Unnecessary calls to MDC steal:** storage management was trying to steal frames from MDC even though it knew MDC had none. Fixed.

z/VM Performance Toolkit

z/VM Performance Toolkit: PTFs for z/VM 6.3 (1 of 4)

- VM65656: Perfkit has a CMS Pipelines input
  - Useful if you want to concatenate several MONWRITE files as a single Perfkit input

- VM65528: Support for Multi-VSwitch Link Aggregation
  - New report: FCX317 GLONACT Global Networking Object Activity
  - Changed reports
    - FCX155 MONDATA, counts new record
    - FCX185 IOCHANGE, changes for global VSwitch
    - FCX240 VSWITCH, same
    - FCX266 GVSWITCH, same
    - FCX267 EVSWITCH, same

z/VM Performance Toolkit: PTFs for z/VM 6.3 (2 of 4)

- VM65699: New Function
  - FCX215 FICON gets channel read and write speeds
  - FCX155 MONDATA now counts the events that happened after the last flight of samples
  - The whole family of wait-state reports (USTAT, etc.) has repaired headers
  - Nod to SMT: “LPU” now changed to “Core” in many places
  - FCX180 SYSCONF displays CEC tttt-mmm and MCI
  - FCX179 SYSLOG computes user-exit %busy correctly

z/VM Performance Toolkit: PTFs for z/VM 6.3 (3 of 4)

- VM65698: z13 GA2 and z13s
  - New reports for format-3 PCI functions:
    - FCX322 PCI Activity
    - FCX323 PCI Activity Log
  - Changed reports for format-3 PCI functions:
    - FCX310 PCI Menu
    - FCX311 PCI Function Config

z/VM Performance Toolkit: PTFs for z/VM 6.3 (4 of 4)

- VM65697: CPU Pooling, LPAR Group Capping, and Prorated Core Time
  - New reports:
    - FCX324 CPLMENU CPU Pooling Menu
    - FCX308 CPLCONF CPU Pooling Configuration
    - FCX309 CPLACT CPU Pooling Activity
  - Changed reports:
    - FCX124 MENU Performance Data Selection Menu
    - FCX226 UCONF User Configuration (adds CPU Pool name)
    - FCX126 LPAR LPAR Activity (adds MT and group cap fields)
    - FCX202 LPAR LPAR Log (adds MT and group cap fields)
    - FCX306 LSHARACT LPAR Share Activity (adds group cap fields)

z/VM Performance Toolkit – z/VM 6.4

- **Perfkit now requires z/CMS**
  - It **no longer runs on plain CMS**
  - It uses z/Architecture instructions
  - It is able to use memory above 2 GB

- Changed reports:
  - FCX124 MENU: choice 1 now goes to new CPUMENU
  - FCX265 LOCKLOG: not available for data from z/VM 6.4 or later (use new LOCKACT)

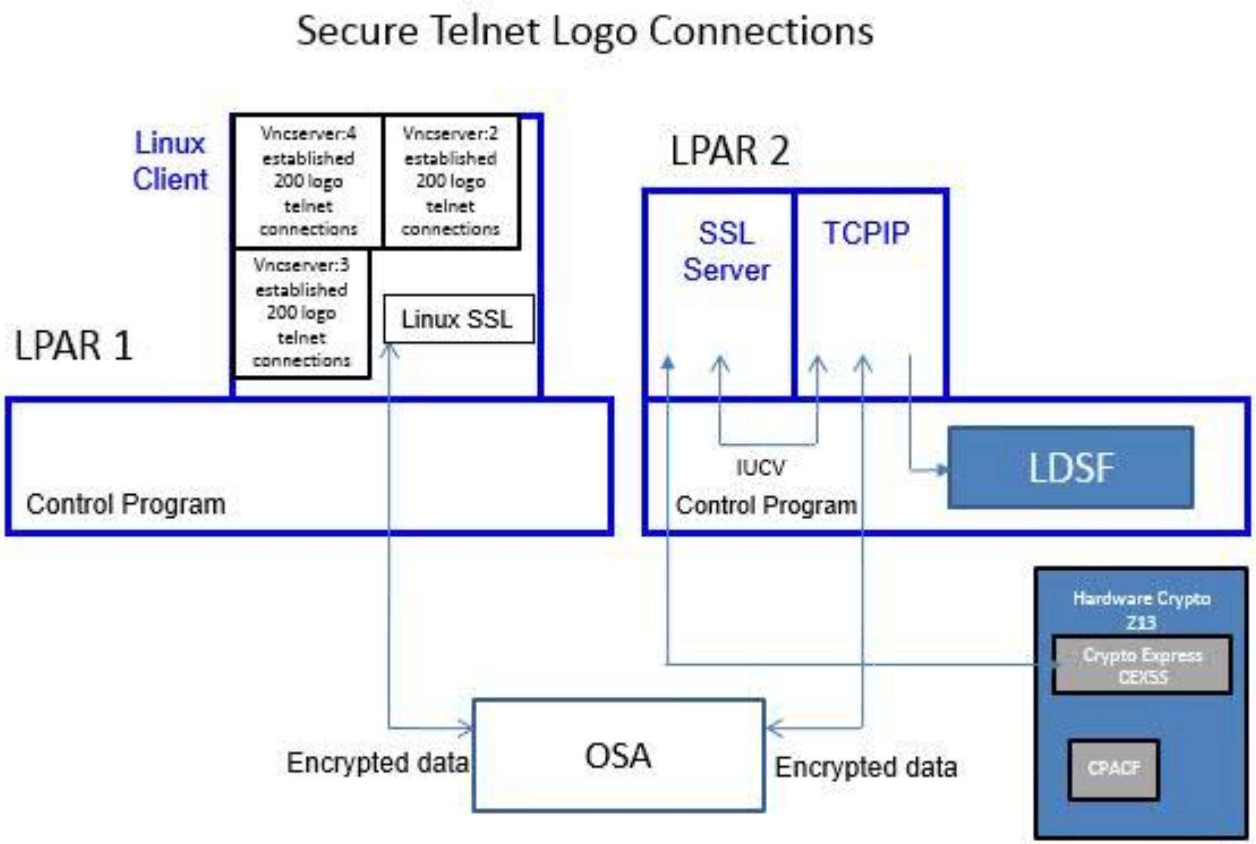
- New reports:
  - FCX325 CPUMENU: shows a menu of CPU-related options
  - FCX326 LOCKACT: spin lock activity report

z/VM 6.4 First Quarter 2017

z/VM 6.4 1Q17 SPE Stack: Regression Behavior

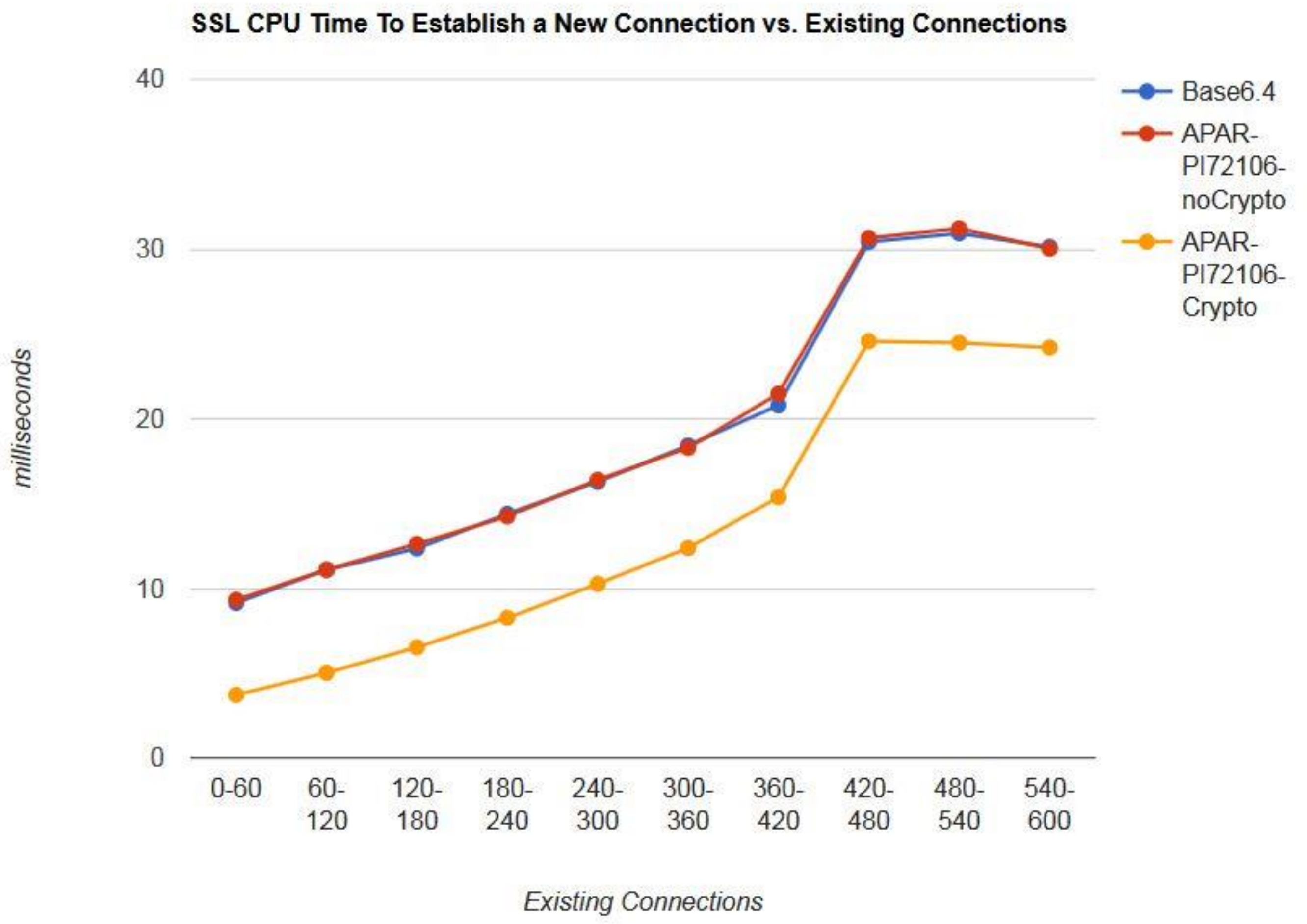
- Compared back to z/VM 6.4 GA on same hardware (z13), our suite experienced:
  - ETRR: mean (µ) 0.98, standard deviation (σ) 0.14
  - ITRR: µ 0.98, σ 0.14

CRYPTO APVIRT Support in TLS/SSL Server and LDAP/VM


Secure Telnet Logo Connections
Linux Client
Vncserver:4 established 200 logo telnet connections
Vncserver:2 established 200 logo telnet connections
Vncserver:3 established 200 logo telnet connections
Linux SSL
LPAR 1
Control Program
LPAR 2
SSL Server
TCPIP
IUCV
Control Program
LDSF
Hardware Crypto Z13
Crypto Express CEX5S
CPACF
Encrypted data
OSA
Encrypted data

APAR is PI72106 (z/VM 6.4) to TCP/IP and LDAP/VM.

CRYPTO APVIRT: CPU Time to Establish a New Connection



Read the article: http://www.vm.ibm.com/perf/reports/zvm/html/640cip.html

Concurrent I/O Support for XIV

- Lets CP overlap I/Os to XIV EDEVs
  - Guests' I/Os
  - Its own I/Os

- APAR VM65929, PTF UM35080 for z/VM 6.4

- We evaluated this with a heavy-paging workload configured to page to XIV EDEVs

- Results for our workload:
  - 96% increase in ETR
  - 83% increase in pages/sec to EDEVs
  - Transfers/sec to each EDEV about doubled

- Read the article: http://www.vm.ibm.com/perf/reports/zvm/html/640xiv.html

Dump Channel Program Improvements

- Changes structure of channel programs used for dumps
  - PSW restart dumps
  - SNAPDUMPs

- APAR VM65989, PTF UM35132 for z/VM 6.4

- We evaluated this using SNAPDUMPs of several different sized LPARs

- Results for our workload: dump rate improved 260%

- Read the article: http://www.vm.ibm.com/perf/reports/zvm/html/640500.html

z/VM 6.4 Third Quarter 2017

z/VM 6.4 3Q17 SPE Stack: Regression Behavior

- Compared back to z/VM 6.4 1Q17 on same hardware (z13), our suite experienced:
  - ETRR: mean (μ) 1.02, standard deviation (σ) 0.52
  - ITRR: μ 1.02, σ 0.44

New Shared/Exclusive Spin Lock Manager

- We continue to look for ways to reduce the MP-effect penalty

- The scheduler lock is a kind of a lock called a “shared/exclusive lock”
  - Many processors can concurrently hold it “shared” – no fair changing the protected data
  - OR, one processor can hold it “exclusive” – I am changing the protected data

- We changed how we use cache lines:
  - Old manager: all acquirers hit a single line read/write
  - New manager: only the exclusive acquirers hit that line read/write

- Results:
  - A DayTrader-based workload with high scheduler lock content: ETRR 6.29, ITRR 5.00
  - A VIRSTOR-based workload with moderate scheduler lock content: ETRR 1.03, ITRR 1.44
  - Your results will depend highly upon:
    - How heavily your workload drives the scheduler lock, and
    - What fraction of that is demand for shared holds

- APAR is VM65988, PTF UM35214 for z/VM 6.4

z/VM 6.4 Fourth Quarter 2017

VM66063: High PR/SM LPAR Management Time

```
1FCX302  Run 2017/06/26 16:24:06         PHYSLOG
                                         Real Core Utilization Log
 From 2017/06/26 10:10:00
 To   2017/06/26 10:46:00
 For   2160 Secs 00:36:00                "This is a performance report"
 ______________________________________________________________________________

  Interval      <PhCore> Shrd  Total
  End Time Type Conf Ded Log. Weight %LgclC %Ovrhd LCoT/L %LPmgt %Total TypeT/L
  >>Mean>> IFL    86   0   95   1000 3240.9 244.65  1.075 1804.8 5290.4   1.632
  >>Mean>> >Sum   86   0   95   1000 3240.9 244.65  1.075 1804.8 5290.4   1.632

  10:11:00 IFL    86   0  116   1000 4238.0 340.31  1.080 1714.0 6292.3   1.485
  10:11:00 >Sum   86   0  116   1000 4238.0 340.31  1.080 1714.0 6292.3   1.485

  10:12:00 IFL    86   0  115   1000 4164.1 343.12  1.082 1911.5 6418.7   1.541
  10:12:00 >Sum   86   0  115   1000 4164.1 343.12  1.082 1911.5 6418.7   1.541
```

%LPmgt is time spent in PR/SM and not chargeable to the LPARs.

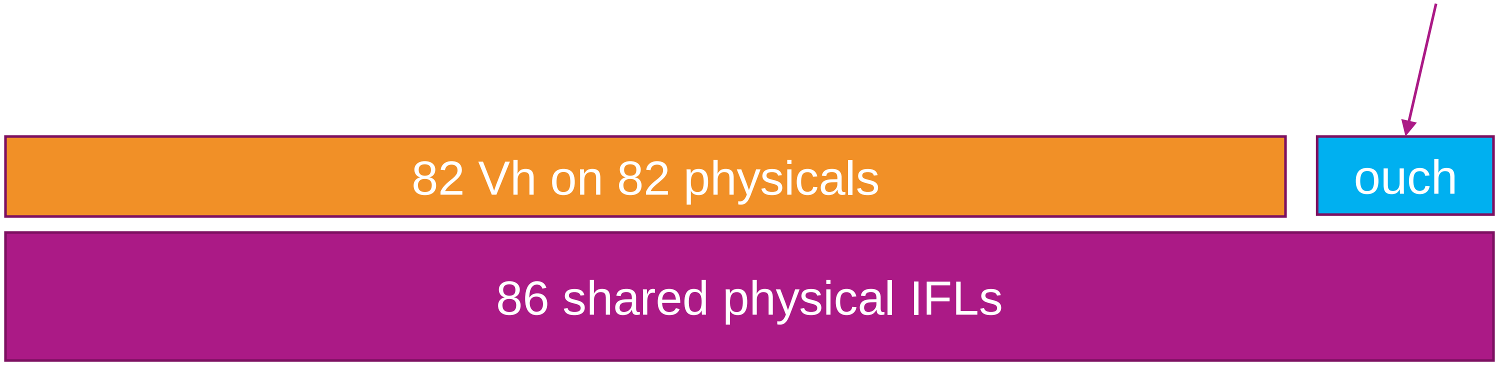
One contributing factor in this situation was found to be that
the LPARs of the CPC were running with too many vertical-lows unparked.

VM66063: Too Many Vertical-Lows? What Does That Mean?

Suppose the CPC has 86 shared physical IFLs and these LPARs:

1. One LPAR with entitlement 688 and 6 Vh, 1 Vm, 5 Vl
2. Two LPARs, each with entitlement 1978 and 19 Vh, 1 Vm, 4 Vl
3. Two LPARs, each with entitlement 1978 and 19 Vh, 1 Vm, 8 Vl

This makes 82 Vh and 34 non-Vh, like so:



34 non-Vh on
4 physicals!

If we could get rid of the unneeded Vls we could improve the situation.

VM66063: The Old Unparking Heuristic

- When Global Performance Data Control is ON,

- z/VM 6.4 unparks logical cores according to perceived *capacity*.
  - How many of my cores do I think PR/SM will power? Let's unpark that many.
  - Then spread the workload over all those unparked cores.

- This strategy can be a mistake in some situations:
  - Across the whole CPC, it results in unparking too many vertical-lows
  - Especially if the LPARs have lots of vertical-lows (*e.g.*, entitlement 1900% with 26 logical cores)

- The excessive unparking of vertical-lows can cause:
  - Increased PR/SM LPAR management time (FCX302 PHYSLOG %LPmgt)
  - Increased logical core suspend time (FCX304 PRCLOG %Susp)
  - Increased nonchargeable CP time ("system time") (FCX304 PRCLOG Syst)
  - Increased CP use of Diag x'9C' (FCX239 PROCSUM Diag9C/sec)
  - Increased guest use of Diag x'9C' (FCX104 PRIVOP Diagnose X'9C')

- The solution is to introduce some new unparking heuristics that can help reduce the MP level by parking unneeded logical cores

VM66063: Now, Three Unparking Heuristics

Large: Runs in all the logical cores it appears will be powered. (today’s behavior)
This can tend to unpark vertical-lows.

Medium: Runs in the Vh and Vm cores plus only the needed-and-powered Vl cores.
This can tend to reduce the number of unparked vertical-lows.

Small: Runs in only the needed-and-powered cores.
This can tend to reduce the number of unparked cores.

Effect: N(small) <= N(medium) <= N(large)
Medium has potential to dispose of Vls and of some MP effect.
Small has potential to dispose of Vl, Vm, and Vh and of even more MP effect.

VM66063: Externals

(All of this applies only to running with GPD on)

To change the unparking model:

```
>--SET SRM UNPARKING--+--LARGE---+--><
                      +--MEDIUM--+
                      +--SMALL---+
```

To query which model is in effect:

```
>--QUERY SRM--+-----------+--><
              +UNPARKING-+
```

To shut off use of vertical-lows (this is mostly a safety switch):

```
>--SET SRM EXCESSUSE--+--HIGH---+--><
                      +--MEDIUM-+
                      +--LOW----+
                      +--NONE---+
```

There are corresponding statements for the z/VM system configuration file.

z/VM 6.4 First Quarter 2018

VM65993: Encrypted Paging

- Encryption of guest data by the Control Program as data is paged out to paging volumes owned by z/VM. Includes primary guest address space, data space, and VDISK space.
- Supported on the z14; Uses the Central Processor Assist for Cryptographic Functions (CPACF)
- Available Cipher Strength
  - Advanced Encryption Standard 128-bit (AES128), AES 192-bit (AES192) and AES 256-bit (AES256)
- Performance Key Findings
  - As cipher strength increased, total CPU used on encryption and decryption increased
    - CPU time used to encrypt a page increased
    - CPU time used to decrypt a page decreased
  - On average, decryption costs less than encryption
  - Despite the extra cost of encryption, the z14 with encrypted paging enabled performed better when compared back to a z13
  - The CPU cost of encrypted paging is a function of the paging rate rather than the size of the LPAR.
- Performance Report: http://www.vm.ibm.com/perf/reports/zvm/html/640EP.html

VM65987: Pause-Reduced Garbage Collection for Java

- *Pause time* is the amount of time Java application threads are stopped during a garbage collection (GC) event.
  - Can impact response times and cause application failures.
- Java can now exploit the new Guarded Storage Facility (GSF) architecture for the z14.
  - Java application threads can now run concurrently during most GC events.
- Requires IBM Java SR5 and Linux support (TBD)
- A Java Store Inventory and Point of Sale application was used to evaluate this feature
- Results
  - Response-time-constrained throughput increased 67%
    - Transactions that satisfy certain response-time-constrained SLAs
  - Avg. pause time for a GC event decreased from 221 ms to 32 ms
  - No difference in z/VM performance
- Performance Report: http://www.vm.ibm.com/perf/reports/zvm/html/640jgc.html

Summary

Summary

- z14 offers a good performance bump over the z13
- z/VM 6.4 offers good regression behavior compared to z/VM 6.3
- z/VM 6.4 offers improved performance and capacity compared to z/VM 6.3
  - The Control Program can now exploit a 2 TB LPAR
  - The Control Program can now exploit HyperPAV aliases for paging
  - The Control Program can now exploit High Performance FICON for paging
- z/VM 6.4 offers improved scheduler behavior compared to z/VM 6.3
  - Relative shares are enforced more accurately than they were previously
- z/VM 6.4 offers increased cipher strength compared to z/VM 6.3
- z/VM 6.4 offers many small improvements compared to z/VM 6.3
- There are a number of new or changed z/VM Performance Toolkit screens
  - Perfkit now requires a z/Architecture virtual machine and z/CMS
- There are a number of new or changed monitor records
- Visit us on the web at http://www.vm.ibm.com/perf/reports/zvm/html/

Thank you!

Send feedback to:
Brian Wade, bkw@us.ibm.com

Also, visit our z/VM Performance Report:
http://www.vm.ibm.com/perf/reports/zvm/html/

Appendix: Monitor Record Changes

z/VM 6.4 GA Monitor Record Changes, 1 of 4

D and R	Name	Long Name	N=new; C=changed; D=deleted
Domain 0, System			
D0 R1	MRSYTSYP	System data (per processor)	C
D0 R2	MRSYTPRP	Processor data (per processor)	C
D0 R4	MRSYTRSP	Real storage data (per processor)	C
D0 R5	MRSYTXSP	Expanded storage data (per processor)	D
D0 R14	MRSYTXSG	Minidisk cache data (global)	C
D0 R15	MRSYTCUG	Logical partition configuration	C
D0 R21	MRSYTSXG	System execution space (global)	C
D0 R23	MRSYTLCK	Formal spin lock data	C
Domain 1, Monitor			
D1 R4	MRMTRSYS	System configuration data	C
D1 R6	MRMTRDEV	Device configuration data	C
D1 R7	MRMTRMEM	Memory configuration data	C
D1 R16	MRMTRSCH	Scheduler settings	C
D1 R17	MRMTRXSG	Expanded storage data	D
D1 R19	MRMTRQDC	QDIO device configuration	C
D1 R20	MRMTRHPP	HyperPAV pool definition	C
D1 R31	MRMTRSRV	CP service configuration	N

z/VM 6.4 GA Monitor Record Changes, 2 of 4

D and R	Name	Long Name	N=new; C=changed; D=deleted
Domain 2, Scheduler			
D2 R4	MRSCLADL	Add user to dispatch list	C
D2 R5	MRSCLDDL	Drop user from dispatch list	C
D2 R6	MRSCLAEL	Add user to eligible list	C
D2 R7	MRSCLSRM	SET SRM changes	C
D2 R13	MRSCLALL	Add VMDBK to the limit list	C
D2 R14	MRSCLDLL	Drop VMDBK from the limit list	C
Domain 3, Storage			
D3 R1	MRSTORSG	Real storage management (global)	C
D3 R3	MRSTOSHR	Shared storage management (per NSS or DCSS)	C
D3 R4	MRSTOASP	Auxiliary storage management	C
D3 R8	MRSTOBPG	Block paging data	C
D3 R9	MRSTOXSG	Expanded storage data	D
D3 R10	MRSTOXSU	Expanded storage data (per user)	D
D3 R11	MRSTOASS	Auxiliary shared storage management	C
D3 R14	MRSTOASI	Address space information record	C
D3 R16	MRSTOSHD	NSS/DCSS/SSP removed from storage	C
Domain 4, User			
D4 R2	MRUSELOF	User logoff data	C
D4 R3	MRUSEACT	User activity data	C
D4 R9	MRUSEATE	User activity data at transaction end	C

z/VM 6.4 GA Monitor Record Changes, 3 of 4

D and R	Name	Long Name	N=new; C=changed; D=deleted
Domain 5, Processor			
D5 R1	MRPRCVON	VARY ON processor	C
D5 R2	MRPRCVOF	VARY OFF processor	C
D5 R11	MRPRCINS	Instruction counts (per processor)	C
D5 R18	MRPRCDHF	Dispatch vector high frequency data	C
D5 R20	MRPRCMFM	MT CPUMF counters	C
D5 R21	MRPRCSMT	SMT configuration change event	N
Domain 6, I/O, 1 of 2			
D6 R1	MRIODVON	VARY ON device	C
D6 R3	MRIODDEV	Device activity	C
D6 R4	MRIODCAD	Cache activity data	C
D6 R10	MRIODALS	Automated tape library statistics	C
D6 R22	MRIODVSF	Virtual switch failover	
D6 R23	MRIODVSR	Virtual switch recovery	C
D6 R25	MRIODQDA	QDIO device activation event	C
D6 R27	MRIODQDD	QDIO device deactivation event	C
D6 R28	MRIODHPP	HyperPAV pool activity	C

z/VM 6.4 GA Monitor Record Changes, 4 of 4

D and R	Name	Long Name	N=new; C=changed; D=deleted
Domain 6, I/O, 2 of 2			
D6 R30	MRIODLPT	LSS PAV transition	C
D6 R32	MRIODHPF	Indicates an HPF feature change	C
D6 R34	MRIODBPD	Virtual switch bridge port deactivation	C
D6 R40	MRIODPDS	Guest disables a PCI function	C
D6 R42	MRIODPAD	PCI function added to the system	C
D6 R45	MRIODPON	Real PCI function varied on	C
Domain 8, Virtual Network			
D8 R1	MRVNDSES	Virtual NIC session activity	C
D8 R2	MRVNDLSU	Virtual NIC guest link state – link up	C
D8 R3	MRVNDLSD	Virtual NIC guest link state – link down	C

End