IBM Systems and Technology Group

Linux on zSeries Performance Tools

SHARE 107 Technical Conference in Baltimore, Maryland

August 13-18, 2006

Session 2592/9302

Oliver Benke
IBM Germany Lab
Email: benke@de.ibm.com

ON DEMAND BUSINESS™

eServer Systems Management

Trademarks

The following are trademarks of the International Business Machines Corporation in the United States and/or other countries.

IBM*
the IBM logo*
OS/390*
Parallel Sysplex*
MVS
z/OS*

RACF*
RMF
zSeries
Tivoli*
CICS
IMS

DB2*
WebSphere*
Domino
e business(logo)*
e(logo)server
e(logo)businss

Lotus*
Tivoli(logo)*
z/VM*
z/Architecture
zSeries*

* Registered trademarks of IBM Corporation

The following are trademarks or registered trademarks of other companies.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. See Java Guidelines

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

SET and the SET Logo are trademarks owned by SET Secure Electronic Transaction LLC.

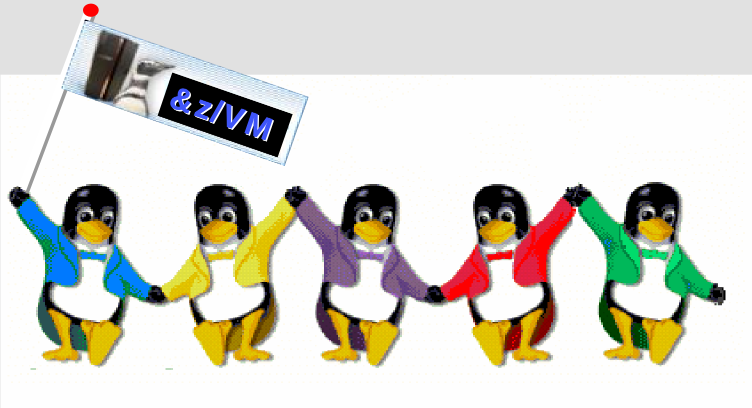
LINUX is a registered trademark of Linus Torvalds

* All other products may be trademarks or registered trademarks of their respective companies.

Agenda

1. **Performance Management, zSeries Architecture, …**
   Base concepts

2. **Performance Tools with Usage Examples**



Some basics

- **Performance Management**
- **Resource Sharing, Overcommitted Resources, Virtualization**
  - CPU Resources in a virtualized environment
- **zSeries Mainframes: what’s different?**
- **Performance base concepts**
  - Load Average
  - System/User CPU Consumption
- **The /proc filesystem**

Recent highlights

- System z LPAR, Channel and Device metrics added to the distros
- % Stolen metric for “correct” CPU reporting
- Extensions in SBLIM CIM infrastructure, cluster concept for gathering infrastructure
- I/O Wait Time metric

Performance Management

- **Online Monitoring, Problem drill-down; 1 day history (or 3 days for the weekend) needed**
  - –May be automated, using asynchronous events
  - –Online performance data may be used by autonomic software components, like VMRM and IRD on zSeries
- **Long-term monitoring and capacity planning**
  - –Understand whether growth of resource consumption is bug driven or business driven
  - –Estimate by when you need to invest in new hardware
- **Self-optimization**
  - –First implementations of workload management and
  - –load balancing available for Linux

Mainframe Linux: Any Advantages?

§ **Leading-edge Virtualization**

- –z/VM or LPAR virtualization technologies
- –Possibility to virtualize and share CPUs, Channels (=I/O) and probably Memory (iff running under VM)

§ **Advanced Resource Sharing**

- –Workload Management using *Intelligent Resource Director IRD* or *z/VM VMRM*

§ **Optimized for Server Workloads**

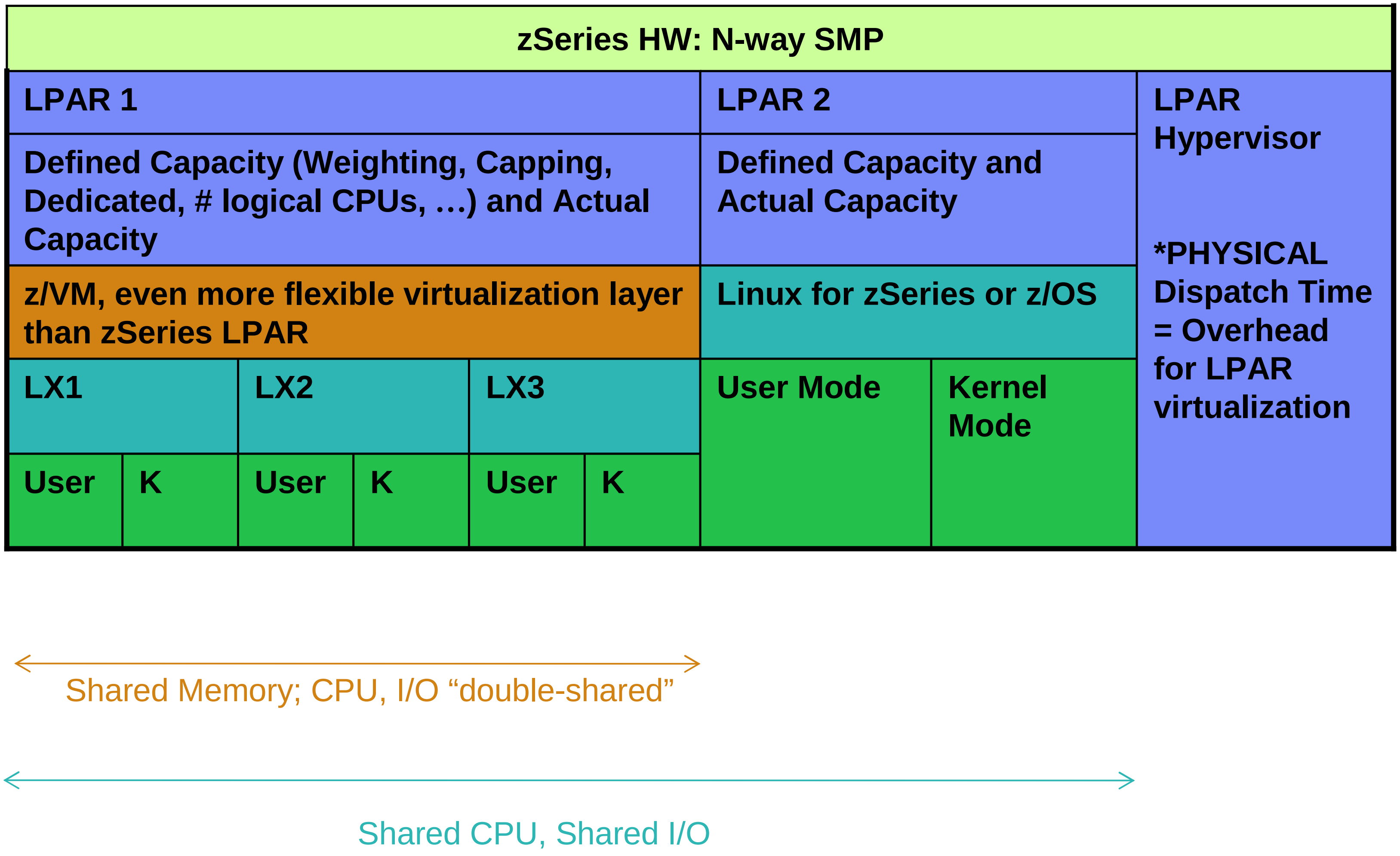
- –Reliability – Availability – Scalability
- –Horizontal and vertical scaling
- –High I/O performance, fast memory

§ **Internal Networking Facilities**

- –Memory-based networking using HiperSockets (LPAR) or GuestLAN (z/VM)

§ **Server consolidation**

Resource Sharing of CPU resources: the zSeries way



zSeries HW: N-way SMP

LPAR 1

Defined Capacity (Weighting, Capping, Dedicated, # logical CPUs, …) and Actual Capacity

z/VM, even more flexible virtualization layer than zSeries LPAR

LX1

User

K

LX2

User

K

LX3

User

K

LPAR 2

Defined Capacity and Actual Capacity

Linux for zSeries or z/OS

User Mode

Kernel Mode

LPAR Hypervisor

*PHYSICAL Dispatch Time = Overhead for LPAR virtualization

Shared Memory; CPU, I/O “double-shared”

Shared CPU, Shared I/O

Idle time

- **In the last picture, idle is not shown. Depending on whether CPU resources are dedicated or not, idle time cannot be attributed to single operating systems, as the zSeries box is only idle if and only if all of the running operating systems are idle concurrently. So for a well used system, you may not see any idle time.**
- **However, if a CPU is dedicated to one operating system, it is used completely by this operating system, so it would make sense to charge this idle time to the operating system which has the dedicated resources.**

Virtual Resources

- **… can be shared between several instances which do not even know about each other, like several companies hosted by the same data center**
- **… can be over-committed to a certain degree. However, this does not mean there are no limits, performance of over-committed systems can be very unpleasant. The useful capacity limit of virtual resources depends on the given workload mix you are running**
- **… can be created “out of nothing”, so as an example, you may go create a whole network infrastructure with router, switches, links, and servers – all virtual, all inside z/VM. No cabling, no hardware configuration changes, pure software. Virtual test floor.**

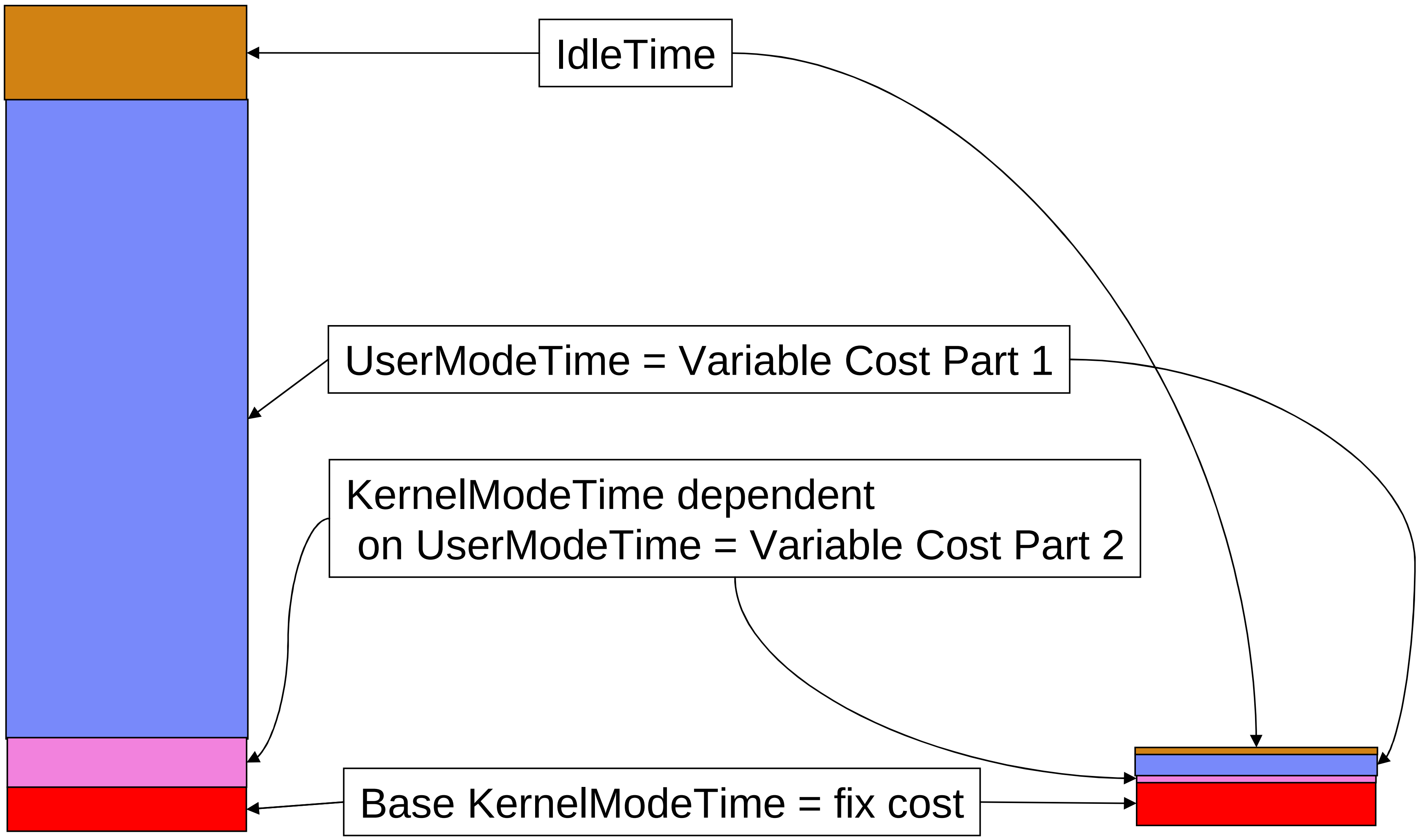
Resource Sharing and Virtualization: Effects

- **No idle resources if any virtual server has useful work to be executed**
  - This way, a mainframe can drive most resources to their capacity limits without penalties to the response times of critical business workloads
- **Different workload may compete for resources with each other, so performance tuning more challenging**
- **For severe over-commitment of resources, overall performance may degrade if no proper workload management and tuning is in place (like thrashing effects)**
- **Re-configuration of virtual data center very flexible; z/VM configuration changes instead of network cabling and hardware changes**

Internal Virtual Networks

- **HiperSockets: zSeries Hardware, can be used to communicate between different LPARs running z/VM, z/OS, Linux for zSeries, Linux under z/VM**
- **For TCP/IP socket-based applications, this is transparent.**
- **Alternative under z/VM 4.2 and higher: Guest LAN - HiperSockets simulated in software, useful for communication of several guests running inside the same z/VM**
- **Connect a “virtual network” (Guest LAN, HiperSockets) with a Linux router to the outside world; of course, this router could be a “hot spot”, so carefully watch it**
- **Older z/VM technologies: IUCV, vCTC**

CPU Usage: Variable cost and Fixed cost



IdleTime

UserModeTime = Variable Cost Part 1

KernelModeTime dependent
on UserModeTime = Variable Cost Part 2

Base KernelModeTime = fix cost

User-mode and kernel-mode CPU time consumption

- **If *UserModeTime / KernelModeTime* is relatively high and IdleTimePercentage is near zero, this can be an indicator that the underlying z/VM has a contention for CPU**
- **This happens because if Linux is constrained for CPU, it may only be able to execute the most important kernel daemons and at the time it would probably start doing some useful work, the CPU is taken away**
- **If KernelModeTime is relatively high, the system overhead is high, and this is usually a bad sign**
- **However, as always, it depends; there are some workloads which simply need high amount of KernelModeTime CPU, and for those workloads, high KernelModeTime values are just normal**

Timer Interrupt and Jiffies

- Derived from PC timer interrupt (100 Hz)
- Every time a timer interrupt occurs (100 times per second), the jiffies variable is incremented by one; that’s one timer tick
- CPU usage is accounted on in jiffies
- If a process is running at the time the timer interrupt occurs, its CPU usage counter is incremented
- Measurements based on 100 Hz timer are accurate on average if sampling is not biased; however, as the clock also drives scheduling, sampling is unfortunately very biased
- Jiffie-based performance measurement is currently wrong if running under z/VM
- Work-around solution: correlate information from LPAR Hypervisor, z/VM and Linux

- On demand timer patch: for an idle Linux image running under z/VM, CPU resources are used up mainly for generating the jiffies. With this patch, jiffies are generated on demand, significantly reducing system load. For newer Linux distribution, you just need to do

  cat 0 > /proc/sys/kernel/hz_timer

  in order to make sure time interrupts are generated on demand instead of 100 times a second

New CPU timer patch (in current 2.6 kernel)

- **In addition to the on-demand timer patch, another step away from the PC 100 Hz timer interrupt with the jiffies concept**
- **Based on zSeries CPU timer instead of 100 Hz timer**
- **Gives you accurate numbers for CPU consumption even if running under LPAR and z/VM**
- **Adds new field** *"CPU steal time" – time Linux wanted to run, but z/VM gave the CPU to some other guest*
- **Officially part of Linux kernel 2.6.11 (generic); hopefully, distributions will pick it up for zSeries within at least 2006**
- **This field will be very useful to understand CPU performance characteristics from within Linux, and much more precise than doing complicated correlation with out-of-band z/VM performance data**

```
top - 09:50:20 up 11 min,  3 users,  load average: 8.94, 7.17, 3.82
Tasks:  78 total,   8 running,  70 sleeping,   0 stopped,   0 zombie
 Cpu0 : 38.7%us,  4.2%sy,  0.0%ni,  0.0%id,  2.4%wa,  1.8%hi,  0.0%si, 53.0%st
 Cpu1 : 38.5%us,  0.6%sy,  0.0%ni,  5.1%id,  1.3%wa,  1.9%hi,  0.0%si, 52.6%st
 Cpu2 : 54.0%us,  0.6%sy,  0.0%ni,  0.6%id,  4.9%wa,  1.2%hi,  0.0%si, 38.7%st
 Cpu3 : 49.1%us,  0.6%sy,  0.0%ni,  1.2%id,  0.0%wa,  0.0%hi,  0.0%si, 49.1%st
 Cpu4 : 35.9%us,  1.2%sy,  0.0%ni, 15.0%id,  0.6%wa,  1.8%hi,  0.0%si, 45.5%s
 Cpu5 : 43.0%us,  2.1%sy,  0.7%ni,  0.0%id,  4.2%wa,  1.4%hi,  0.0%si, 48.6%st
Mem:    251832k total,   155448k used,    96384k free,     1212k buffers
Swap:   524248k total,    17716k used,   506532k free,    18096k cached
```

CPU %stolen: how it works

- **States of a logical CPU as Linux can see it:**
  - a) A physical CPU is attached and Linux uses the CPU
  - b) A physical CPU is available, but Linux is idle
  - c) Linux is not idle, but involuntarily lost the CPU because the hypervisor(s) attached it to another image

  - If CPU is lost due to virtualization (LPAR or z/VM), this is recorded in CPU stolen time.

  - With this patch, you don’t need a z/VM monitor any longer to understand what CPU resources are available to Linux, but you can understand this with pure Linux facilities.

Real CPU instead of just virtual CPU

Two alternatives if you’d like to see Linux “real” CPU numbers instead of virtual CPUs, where “real” CPU numbers are milliseconds spend on real hardware and virtual CPU numbers are fractions of virtual server size (which is dynamic)

- **Use IBM z/VM PT, Tivoli OMEGAMON for z/VM or some other vendor’s tools**
- **Wait until distributions integrate *“% cpu stolen”* metric and exploit this new, highly precise kernel level data. So Linux kernel development has solved this problem finally, and I think the solution is really great! Precise data, not complicated correlation of z/VM and Linux data.**

I/O wait time

- **If a processor is idle *and* a process on the run queue of the given processor has an outstanding I/O request, the processor is waiting for I/O completion**
- **In other words, this is a new I/O contention indicator – high I/O wait time means the processors are "idle" because they are waiting for I/O completion, so the I/O subsystem cannot keep up with the CPUs**
- **With older kernels, this is reported as idle time**
- **Beginning with kernel 2.6, this can be seen in Linux**

Load Average

- Average number of processes on the run queue
- A runnable process is one that is ready to consume CPU resources right now
- A high load average value (in relation to the number of physical processors) is an indicator for latent demand for CPU. The processes waiting on the run queue are not waiting for I/O or other processes, they are waiting for CPU and they are otherwise ready to run.
- load averages are available in various places; you may obtain it by typing
  - *cat /proc/loadavg*

  or using program like *xload*

Linux Page Cache

- **The page cache contains pages of memory mapped files - page I/O related system calls like *generic_file_read.* That's "cached" in /proc/meminfo.**
- **It may contain files which can be freed, and the kernel actually discards those pages if it runs out of free memory.**
- **Linux rarely has free space; everything not used is allocated for Page Cache, so** even if Linux does not really need it all, it uses all available memory **up to the last few percent up to now. "Active" and "Inactive" fields in /proc/meminfo give better information on what parts of memory are actively used.**
- **Linux does not have any special memory regions to do I/O. The size of the memory used for I/O is in "buffers"**

Linux process memory: basic terms

- SIZE**: size of the address space seen by the process, virtual size**
- RSS**: Resident Set Size**
  actual amount of memory that the process is using in RAM
- SHARE**:**
  portion of the RSS that is shared with other processes, such as shared libraries

Note that the implementation of CMM1 and CMM2 will change the way Linux uses memory in a virtualized environment

zSeries-specific tuning

- A nice summary of information can be found at
  - http://www-128.ibm.com/developerworks/linux/linux390/perf/tuning_rec.html

- For example, enabling fixed I/O buffers reduces the number of pages used by z/VM for I/O, and this can significantly increase overall performance.
- As with all hypervisor environments, having too many logical CPUs active mainly increases hypervisor overhead and decreases system throughput.
- For Linux under z/VM, it's crucial to limit memory to what's really needed, as memory is actually virtualized – but it cannot be overcommitted over a certain degree.

Sources for Performance Data on zSeries

- **zSeries Hardware**
  - HMC SNMP interface
- **z/VM**
  - CP MONITOR records, z/VM Performance Toolkit
- **Linux**
  - SYSSTAT package (sar, sadc) and standard LINUX/UNIX tools
  - BSD Accounting records
  - RMF Data Gatherer for Linux (rmfpms)
  - APPLDATA kernel module
  - SBLIM Project (OpenPegasus, CIM)
- **z/OS (SMF, RMF, CIM, …)**
- **Applications**

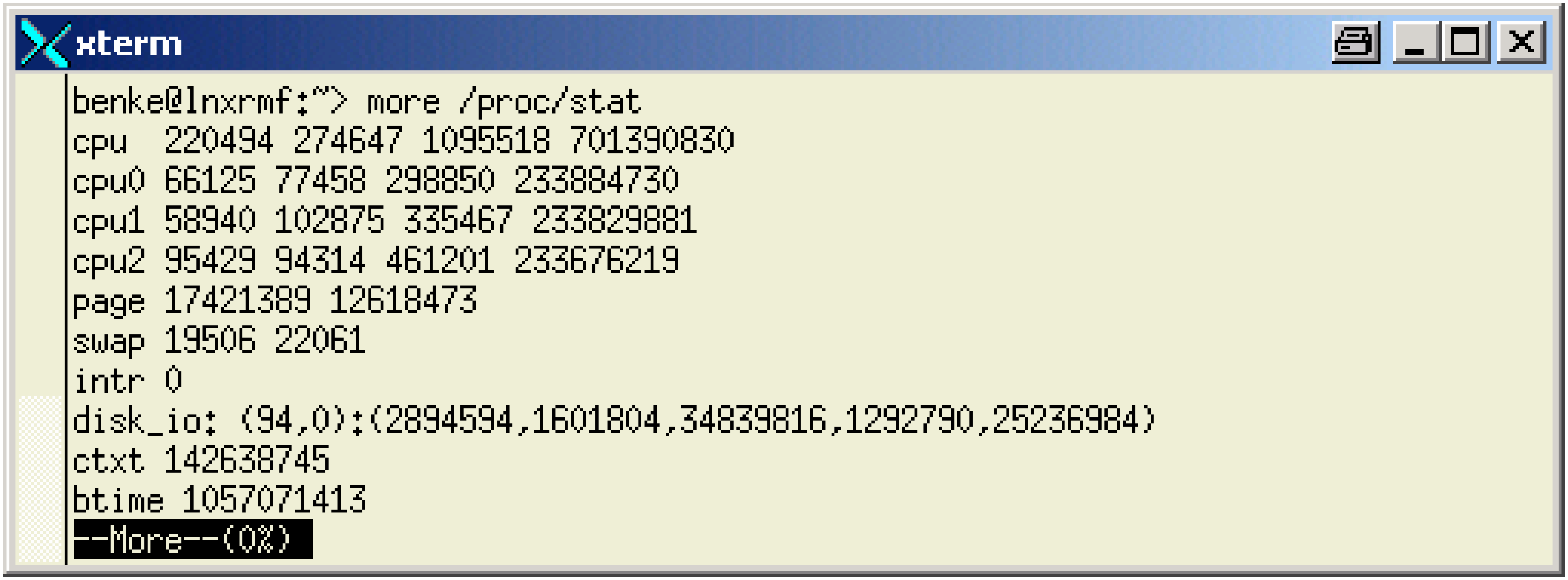
The /proc filesystem

- **Virtual filesystem**
- **One of the interfaces between kernel space and user space; if the user gives a command like**

  cat /proc/stat

  the kernel executes some function to generate the needed "virtual file"
- **Parts of the /proc filesystem are human readable**
- **Most performance measurement tools for Linux are based on /proc filesystem**

/proc/stat Example



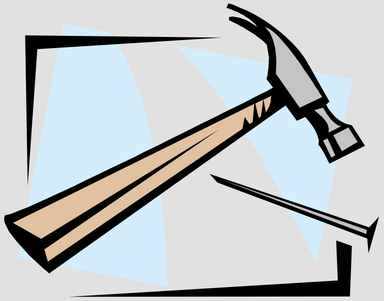
```
benke@lnxrmf:~> more /proc/stat
cpu  220494 274647 1095518 701390830
cpu0 66125 77458 298850 233884730
cpu1 58940 102875 335467 233829881
cpu2 95429 94314 461201 233676219
page 17421389 12618473
swap 19506 22061
intr 0
disk_io: (94,0):(2894594,1601804,34839816,1292790,25236984)
ctxt 142638745
btime 1057071413
--More--(0%)
```

Redbook Paper „Accounting and monitoring for z/VM Linux guest machines“

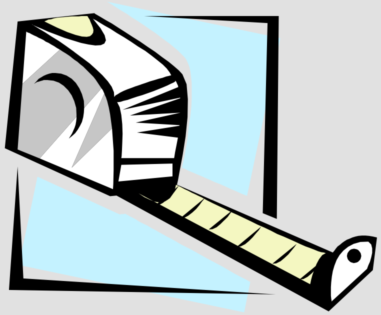
- **Collects CP *MONITOR data and Linux sysstat data (REXX sample code)**
- **Provides this data using a web browser front-end**
- **Sample code can be adjusted**
- **It is possible to correlate z/VM and Linux data; e.g. Linux may think it is 100% CPU busy, but z/VM at the same time may have given Linux only, say, 20% CPU …**
- ***http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp3818.html?Open***

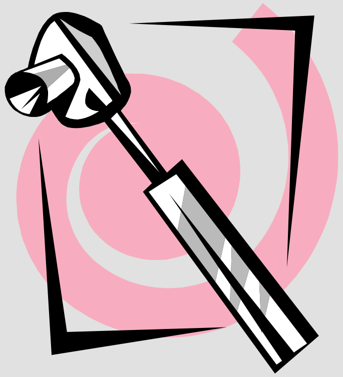
- **Apart from that, there are vendor applications like Tivoli Decision Support with some support for the combination of z/OS, z/VM and Linux on zSeries**

Linux Performance Tools



- **Standard UNIX Tools for performance-related problem analysis: *top, ps, time, netstat, free, vmstat, iostat, strace, df, du, ping, traceroute***
- ***sysstat* package (sar, sadc) for long-term data collection**
- **BSD accounting**
- **NET-SNMP**
- **SBLIM**
- **RMF for Linux, VM Performance Toolkit**





… lots of useful point solutions for performance management

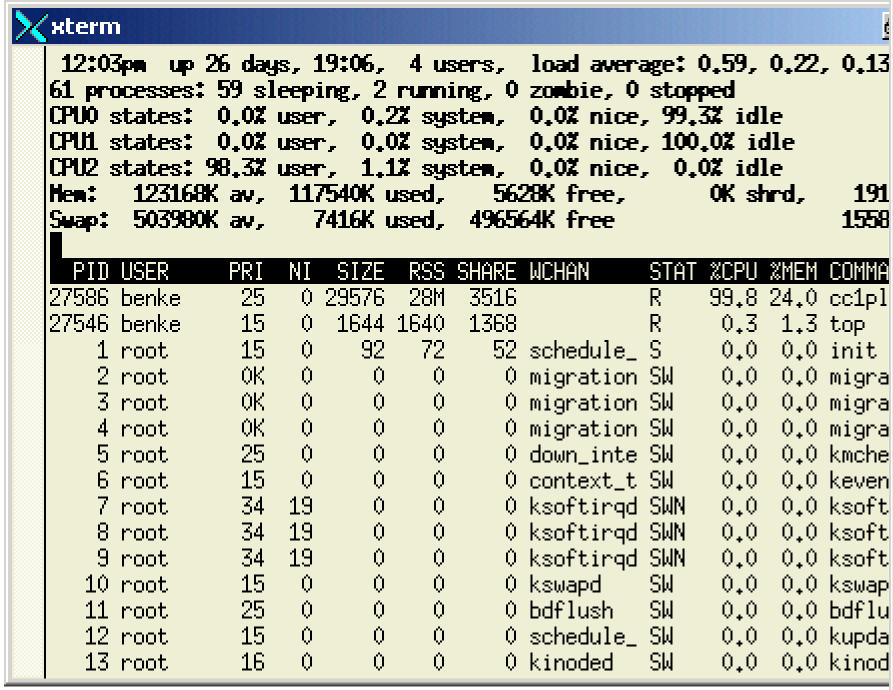
Advantages of good old UNIX standard tools

- **Can be used in own (shell) programs, in order to automate systems management (considered dangerous by some installations)**
- **Very flexible**
- **Available on every UNIX system (but one needs to be careful if it should run on both e.g. AIX as well as on Linux)**
- **Usually quite fast and low impact on system performance**
- **Nice for people who like to code**
- **In any case, at least for problem drill-down analysis, you should know about the standard UNIX tools**

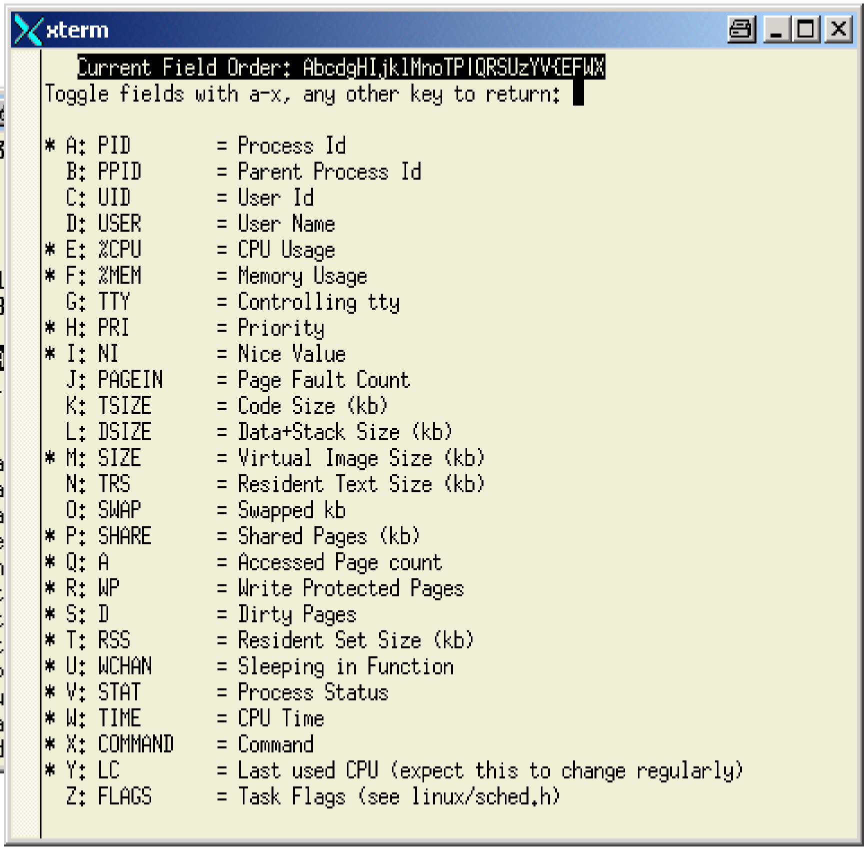
Hard to learn, but everything is explained in man pages (well, almost everything ;-)

top

- **Nice option: in interactive mode, enter** *<f>, <u>, <return>* **to see what the process is waiting for**



```
 12:03pm  up 26 days, 19:06,  4 users,  load average: 0.59, 0.22, 0.13
61 processes: 59 sleeping, 2 running, 0 zombie, 0 stopped
CPU0 states:  0.0% user,  0.2% system,  0.0% nice, 99.3% idle
CPU1 states:  0.0% user,  0.0% system,  0.0% nice, 100.0% idle
CPU2 states: 98.3% user,  1.1% system,  0.0% nice,  0.0% idle
Mem:   123168K av,  117540K used,    5628K free,       0K shrd,    191
Swap:  503980K av,    7416K used,  496564K free                   1558

  PID USER     PRI  NI  SIZE  RSS SHARE WCHAN      STAT %CPU %MEM COMMA
27586 benke     25   0 29576  28M  3516            R    99.8 24.0 cc1pl
27546 benke     15   0  1644 1640  1368            R     0.3  1.3 top
    1 root      15   0    92   72    52 schedule_ S     0.0  0.0 init
    2 root      0K   0     0    0     0 migration SW    0.0  0.0 migra
    3 root      0K   0     0    0     0 migration SW    0.0  0.0 migra
    4 root      0K   0     0    0     0 migration SW    0.0  0.0 migra
    5 root      25   0     0    0     0 down_inte SW    0.0  0.0 kmche
    6 root      15   0     0    0     0 context_t SW    0.0  0.0 keven
    7 root      34  19     0    0     0 ksoftirqd SWN   0.0  0.0 ksoft
    8 root      34  19     0    0     0 ksoftirqd SWN   0.0  0.0 ksoft
    9 root      34  19     0    0     0 ksoftirqd SWN   0.0  0.0 ksoft
   10 root      15   0     0    0     0 kswapd    SW    0.0  0.0 kswap
   11 root      25   0     0    0     0 bdflush   SW    0.0  0.0 bdflu
   12 root      15   0     0    0     0 schedule_ SW    0.0  0.0 kupda
   13 root      16   0     0    0     0 kinoded   SW    0.0  0.0 kinod
```

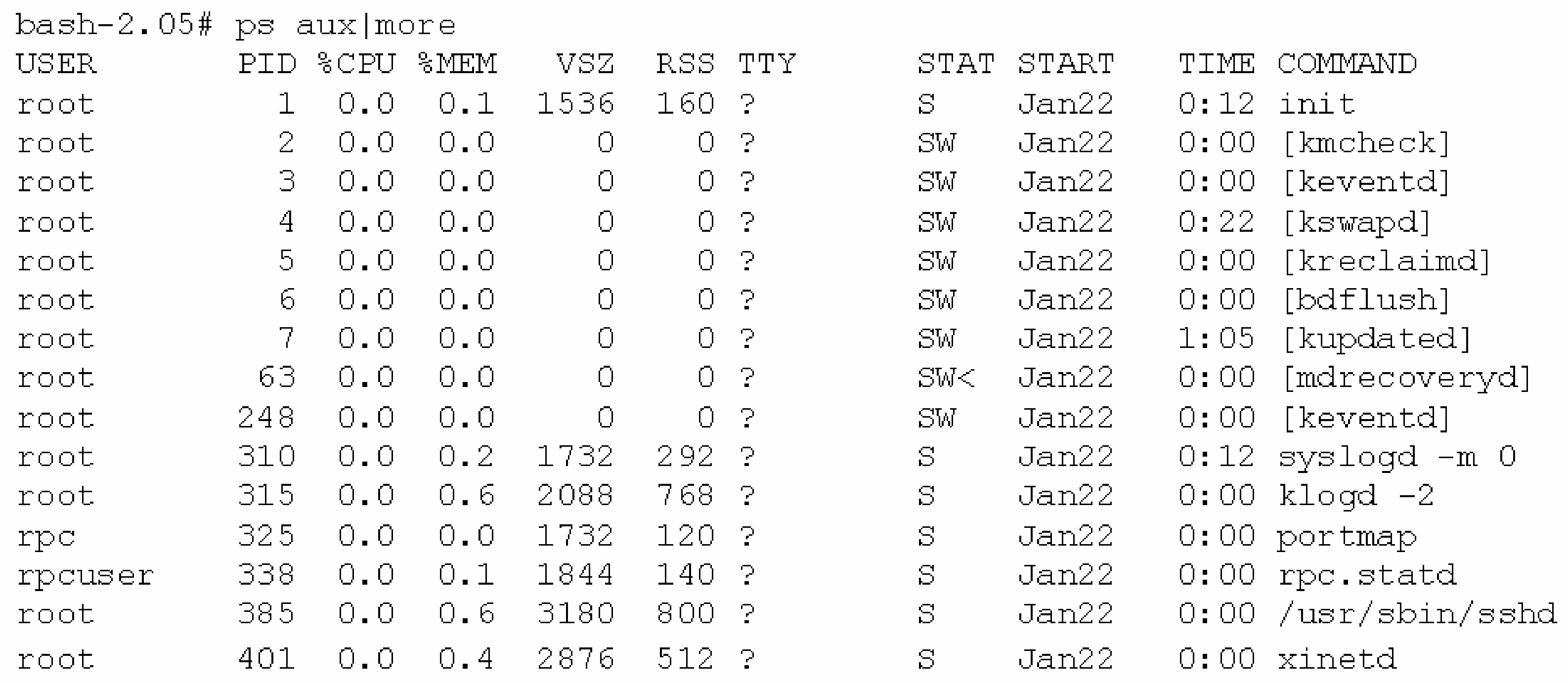


```
   Current Field Order: AbcdgHIjklMnoTP|QRSUzYV{EFWX
Toggle fields with a-x, any other key to return:

* A: PID        = Process Id
  B: PPID       = Parent Process Id
  C: UID        = User Id
  D: USER       = User Name
* E: %CPU       = CPU Usage
* F: %MEM       = Memory Usage
  G: TTY        = Controlling tty
* H: PRI        = Priority
* I: NI         = Nice Value
  J: PAGEIN     = Page Fault Count
  K: TSIZE      = Code Size (kb)
  L: DSIZE      = Data+Stack Size (kb)
* M: SIZE       = Virtual Image Size (kb)
  N: TRS        = Resident Text Size (kb)
  O: SWAP       = Swapped kb
* P: SHARE      = Shared Pages (kb)
* Q: A          = Accessed Page count
* R: WP         = Write Protected Pages
* S: D          = Dirty Pages
* T: RSS        = Resident Set Size (kb)
* U: WCHAN      = Sleeping in Function
* V: STAT       = Process Status
* W: TIME       = CPU Time
* X: COMMAND    = Command
* Y: LC         = Last used CPU (expect this to change regularly)
  Z: FLAGS      = Task Flags (see linux/sched.h)
```

ps - report process status

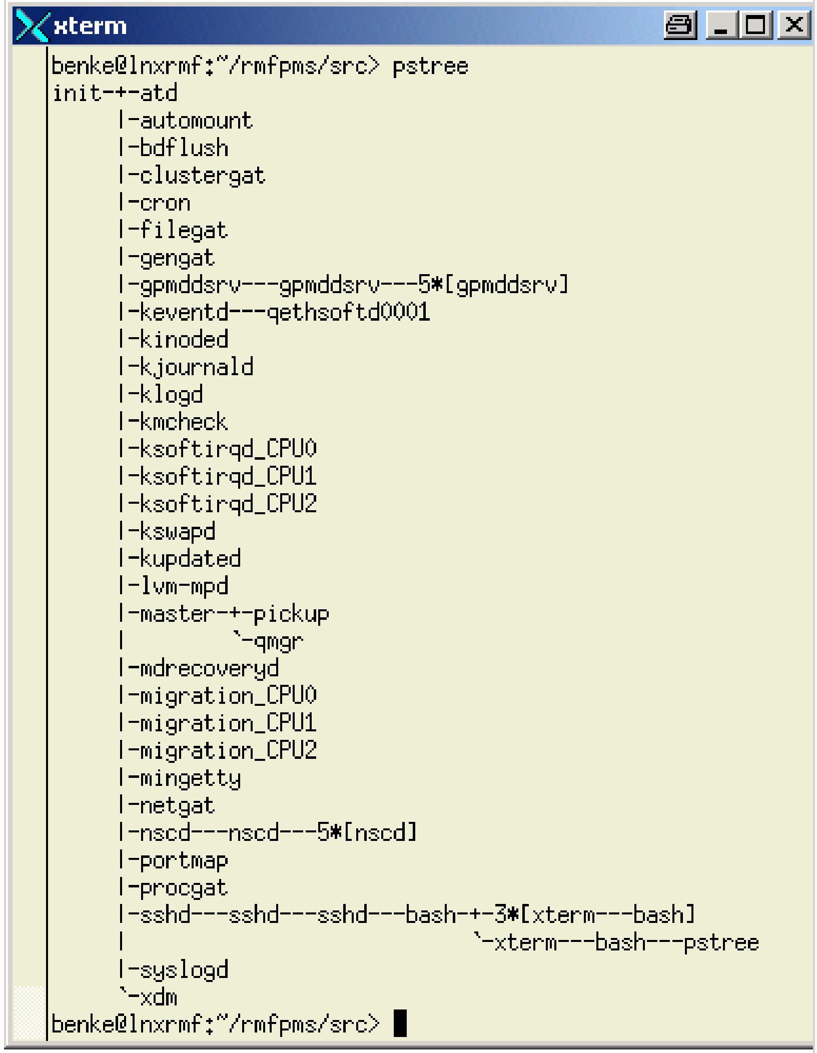
- **common set of parameters:**
  ps aux

- **single out a user:**
  ps u --User apache

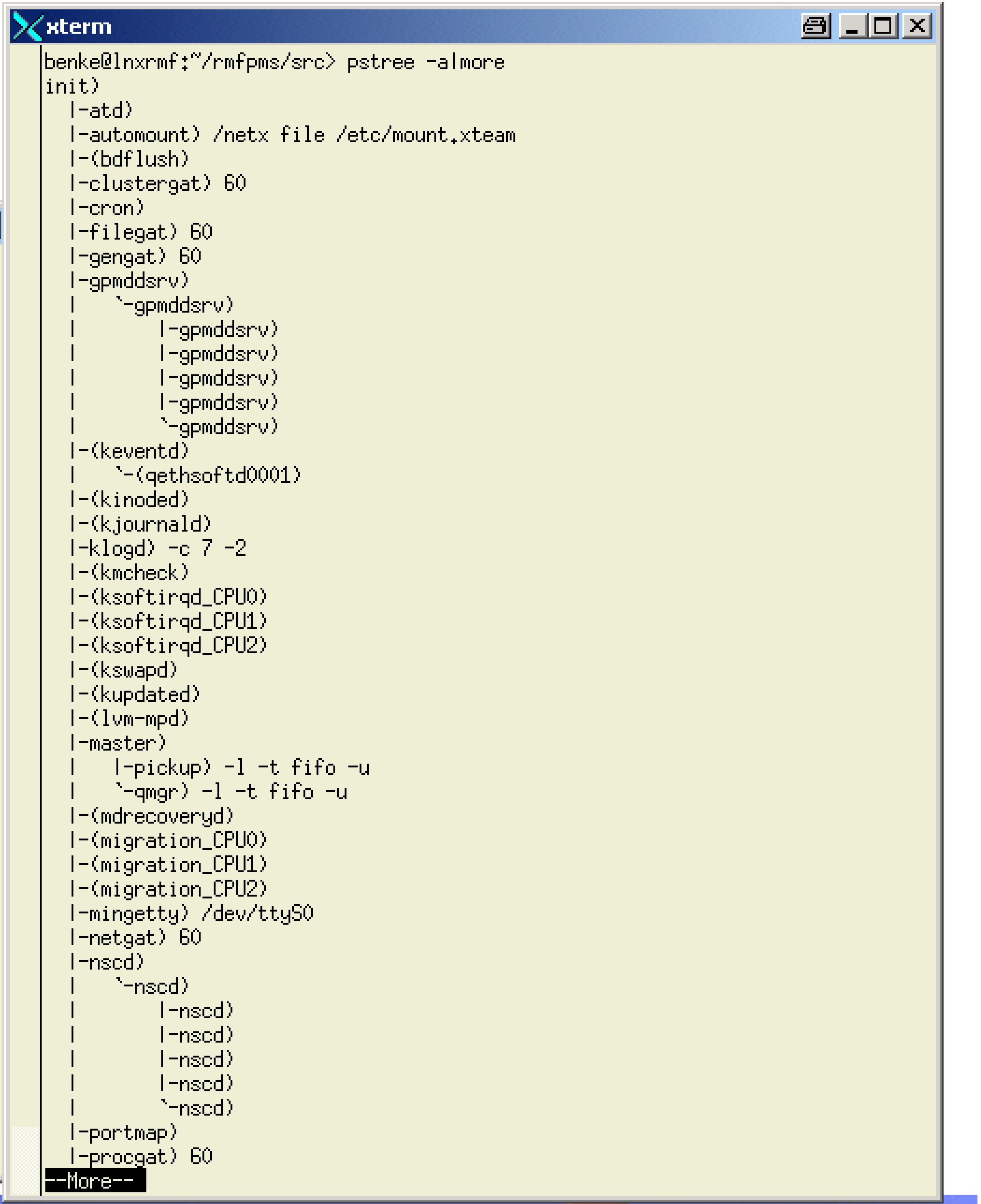


```
bash-2.05# ps aux|more
USER        PID %CPU %MEM   VSZ  RSS TTY      STAT START   TIME COMMAND
root          1  0.0  0.1  1536  160 ?        S    Jan22   0:12 init
root          2  0.0  0.0     0    0 ?        SW   Jan22   0:00 [kmcheck]
root          3  0.0  0.0     0    0 ?        SW   Jan22   0:00 [keventd]
root          4  0.0  0.0     0    0 ?        SW   Jan22   0:22 [kswapd]
root          5  0.0  0.0     0    0 ?        SW   Jan22   0:00 [kreclaimd]
root          6  0.0  0.0     0    0 ?        SW   Jan22   0:00 [bdflush]
root          7  0.0  0.0     0    0 ?        SW   Jan22   1:05 [kupdated]
root         63  0.0  0.0     0    0 ?        SW<  Jan22   0:00 [mdrecoveryd]
root        248  0.0  0.0     0    0 ?        SW   Jan22   0:00 [keventd]
root        310  0.0  0.2  1732  292 ?        S    Jan22   0:12 syslogd -m 0
root        315  0.0  0.6  2088  768 ?        S    Jan22   0:00 klogd -2
rpc         325  0.0  0.0  1732  120 ?        S    Jan22   0:00 portmap
rpcuser     338  0.0  0.1  1844  140 ?        S    Jan22   0:00 rpc.statd
root        385  0.0  0.6  3180  800 ?        S    Jan22   0:00 /usr/sbin/sshd
root        401  0.0  0.4  2876  512 ?        S    Jan22   0:00 xinetd
```

Show running processes as a tree



```
benke@lnxrmf:~/rmfpms/src> pstree
init-+-atd
     |-automount
     |-bdflush
     |-clustergat
     |-cron
     |-filegat
     |-gengat
     |-gpmddsrv---gpmddsrv---5*[gpmddsrv]
     |-keventd---qethsoftd0001
     |-kinoded
     |-kjournald
     |-klogd
     |-kmcheck
     |-ksoftirqd_CPU0
     |-ksoftirqd_CPU1
     |-ksoftirqd_CPU2
     |-kswapd
     |-kupdated
     |-lvm-mpd
     |-master-+-pickup
     |        `-qmgr
     |-mdrecoveryd
     |-migration_CPU0
     |-migration_CPU1
     |-migration_CPU2
     |-mingetty
     |-netgat
     |-nscd---nscd---5*[nscd]
     |-portmap
     |-procgat
     |-sshd---sshd---sshd---bash-+-3*[xterm---bash]
     |                           `-xterm---bash---pstree
     |-syslogd
     `-xdm
benke@lnxrmf:~/rmfpms/src> █
```

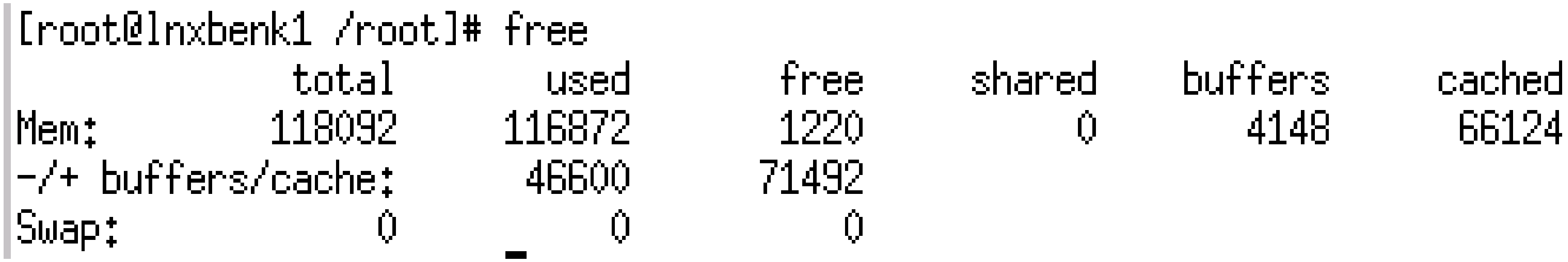


```
benke@lnxrmf:~/rmfpms/src> pstree -almore
init)
  |-atd)
  |-automount) /netx file /etc/mount.xteam
  |-(bdflush)
  |-clustergat) 60
  |-cron)
  |-filegat) 60
  |-gengat) 60
  |-gpmddsrv)
  |   `-gpmddsrv)
  |       |-gpmddsrv)
  |       |-gpmddsrv)
  |       |-gpmddsrv)
  |       |-gpmddsrv)
  |       `-gpmddsrv)
  |-(keventd)
  |   `-(qethsoftd0001)
  |-(kinoded)
  |-(kjournald)
  |-klogd) -c 7 -2
  |-(kmcheck)
  |-(ksoftirqd_CPU0)
  |-(ksoftirqd_CPU1)
  |-(ksoftirqd_CPU2)
  |-(kswapd)
  |-(kupdated)
  |-(lvm-mpd)
  |-master)
  |   |-pickup) -l -t fifo -u
  |   `-qmgr) -l -t fifo -u
  |-(mdrecoveryd)
  |-(migration_CPU0)
  |-(migration_CPU1)
  |-(migration_CPU2)
  |-mingetty) /dev/ttyS0
  |-netgat) 60
  |-nscd)
  |   `-nscd)
  |       |-nscd)
  |       |-nscd)
  |       |-nscd)
  |       |-nscd)
  |       `-nscd)
  |-portmap)
  |-procgat) 60
--More--
```

ON DEMAND BUSINESS™

free

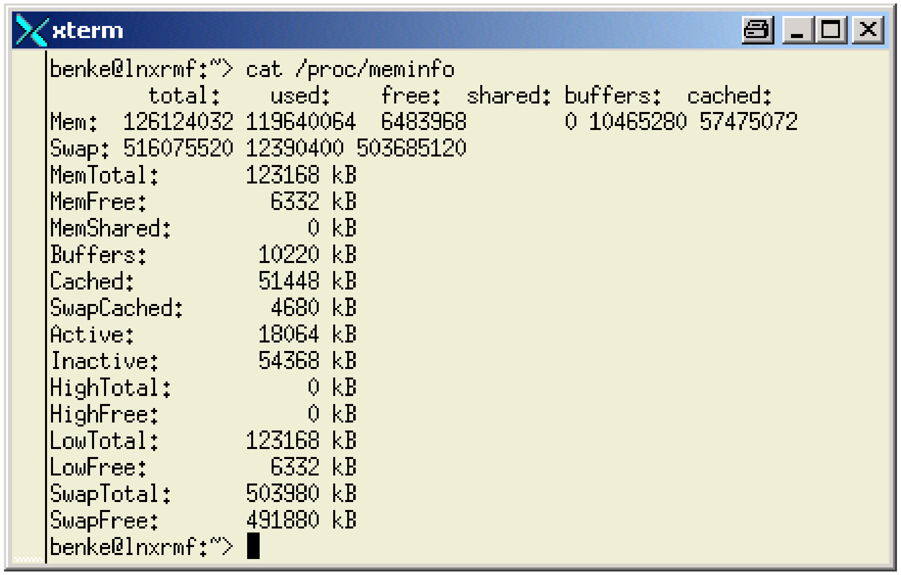
- Give free memory;
  important is the second line, as buffer/cache memory is not really needed by Linux



```
[root@lnxbenk1 /root]# free
             total       used       free     shared    buffers     cached
Mem:        118092     116872       1220          0       4148      66124
-/+ buffers/cache:      46600      71492
Swap:            0          0          0
```

/proc/meminfo

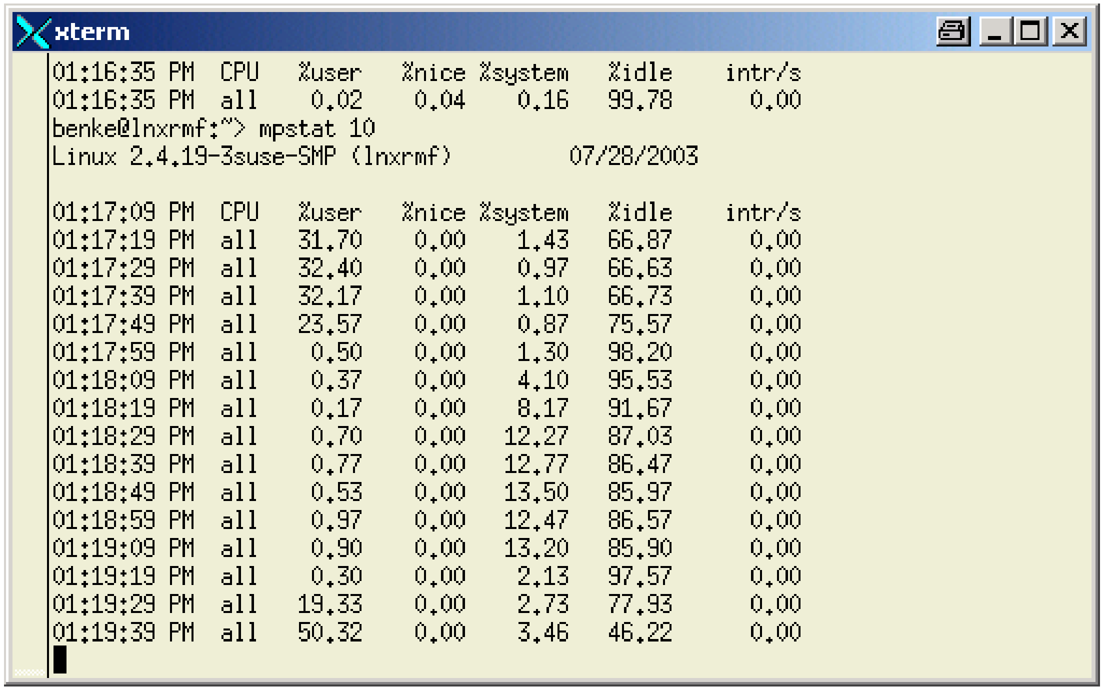
- MemShared: **0 (available for compatibility reasons only)**
- SwapCached: **memory which is both in swap space (=on disk) as well as in main memory (=usable); it’s easier to page memory from the SwapCache out, as there is already a copy in the swap file**
- Active: **memory which was recently used**
- Buffers, Cached: **memory in buffers and in cache**
- Mem, Swap: **physical memory, swap space**



```
benke@lnxrmf:~> cat /proc/meminfo
        total:    used:    free:  shared: buffers:  cached:
Mem:  126124032 119640064  6483968        0 10465280 57475072
Swap: 516075520 12390400 503685120
MemTotal:       123168 kB
MemFree:          6332 kB
MemShared:           0 kB
Buffers:         10220 kB
Cached:          51448 kB
SwapCached:       4680 kB
Active:          18064 kB
Inactive:        54368 kB
HighTotal:           0 kB
HighFree:            0 kB
LowTotal:       123168 kB
LowFree:          6332 kB
SwapTotal:      503980 kB
SwapFree:       491880 kB
benke@lnxrmf:~> █
```

mpstat

- mpstat **is used to display CPU relatded statistics.**
- mpstat 0**: display statistics since system startup (IPL)**
- mpstat N**: display statistics with N second interval time**

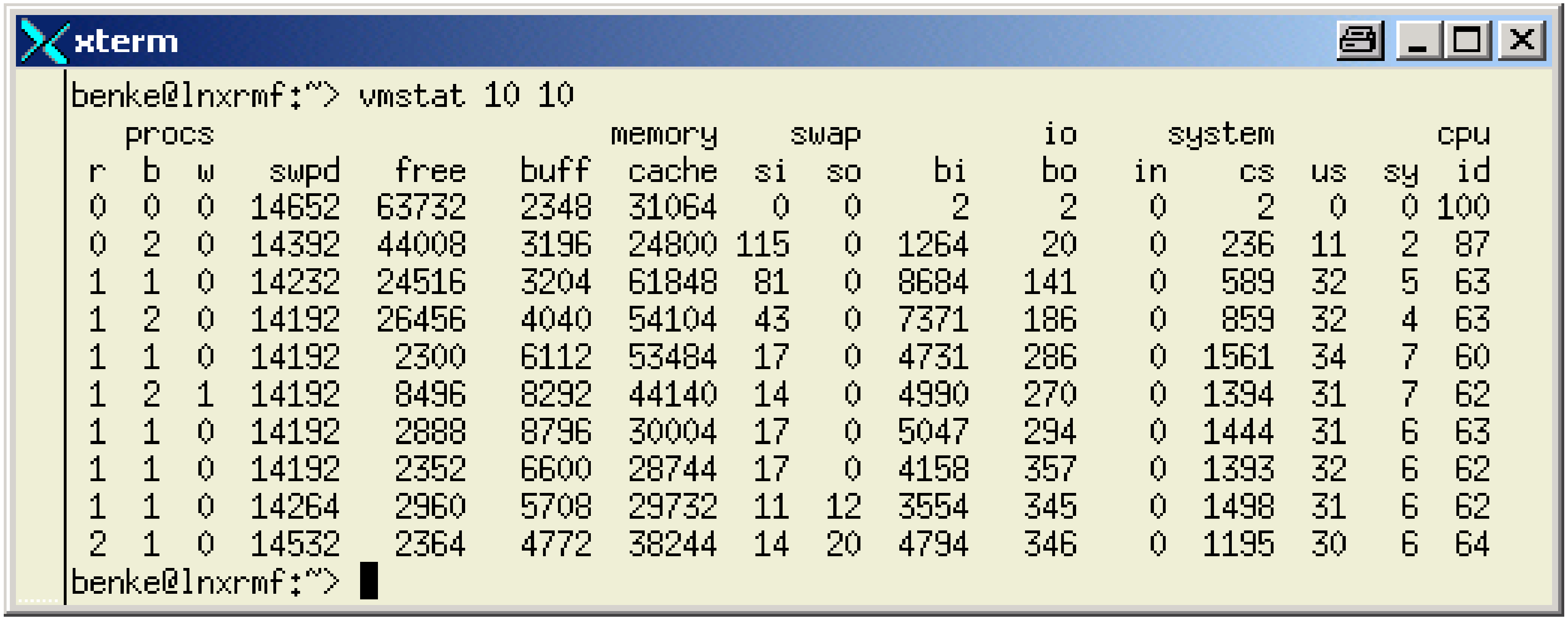


```
01:16:35 PM  CPU   %user   %nice %system   %idle    intr/s
01:16:35 PM  all    0.02    0.04    0.16   99.78      0.00
benke@lnxrmf:~> mpstat 10
Linux 2.4.19-3suse-SMP (lnxrmf)         07/28/2003

01:17:09 PM  CPU   %user   %nice %system   %idle    intr/s
01:17:19 PM  all   31.70    0.00    1.43   66.87      0.00
01:17:29 PM  all   32.40    0.00    0.97   66.63      0.00
01:17:39 PM  all   32.17    0.00    1.10   66.73      0.00
01:17:49 PM  all   23.57    0.00    0.87   75.57      0.00
01:17:59 PM  all    0.50    0.00    1.30   98.20      0.00
01:18:09 PM  all    0.37    0.00    4.10   95.53      0.00
01:18:19 PM  all    0.17    0.00    8.17   91.67      0.00
01:18:29 PM  all    0.70    0.00   12.27   87.03      0.00
01:18:39 PM  all    0.77    0.00   12.77   86.47      0.00
01:18:49 PM  all    0.53    0.00   13.50   85.97      0.00
01:18:59 PM  all    0.97    0.00   12.47   86.57      0.00
01:19:09 PM  all    0.90    0.00   13.20   85.90      0.00
01:19:19 PM  all    0.30    0.00    2.13   97.57      0.00
01:19:29 PM  all   19.33    0.00    2.73   77.93      0.00
01:19:39 PM  all   50.32    0.00    3.46   46.22      0.00
```

Btw the high %system values between 01:18:19 PM and 01:19:09 PM are no problem. I simply executed a file-system stress test, so there was lots of I/O and the operating system had lots to do…

vmstat

- Gives information about memory, swap usage, I/O activity and CPU usage. It really does a lot more than reporting virtual memory statistics …
- Please note that the first line contains a summary line since system start (IPL).
- First parameter: interval time, second parameter: number of parameters.



```
benke@lnxrmf:~> vmstat 10 10
   procs                      memory    swap          io     system         cpu
 r  b  w   swpd   free   buff  cache  si  so    bi    bo   in    cs  us  sy  id
 0  0  0  14652  63732   2348  31064   0   0     2     2    0     2   0   0 100
 0  2  0  14392  44008   3196  24800 115   0  1264    20    0   236  11   2  87
 1  1  0  14232  24516   3204  61848  81   0  8684   141    0   589  32   5  63
 1  2  0  14192  26456   4040  54104  43   0  7371   186    0   859  32   4  63
 1  1  0  14192   2300   6112  53484  17   0  4731   286    0  1561  34   7  60
 1  2  1  14192   8496   8292  44140  14   0  4990   270    0  1394  31   7  62
 1  1  0  14192   2888   8796  30004  17   0  5047   294    0  1444  31   6  63
 1  1  0  14192   2352   6600  28744  17   0  4158   357    0  1393  32   6  62
 1  1  0  14264   2960   5708  29732  11  12  3554   345    0  1498  31   6  62
 2  1  0  14532   2364   4772  38244  14  20  4794   346    0  1195  30   6  64
benke@lnxrmf:~>
```

vmstat fields explained

procs	r	Number of Processes waiting for CPU, Ready to run
	b	Number of Processes blocked in uninterruptable wait (usually for I/O)
	w	Number of Processes swapped out but otherwise ready to run
memory	swpd	Memory used in swap space, in KB
	free	Real memory not used
	buff	Memory used for Buffers
	cache	Memory used for Cache
swap	si	Memory swapped in per second, in KB
	so	Memory swapped out per second, in KB
io	b	Blocks read from block devices per second
	bo	Blocks written to block device per second
system	in	Number of interrupts per second
	cs	Number of context switches per second
cpu	us	User time percentage of total CPU
	sy	System time percentage of total CPU
	id	Idle time percentage of total CPU

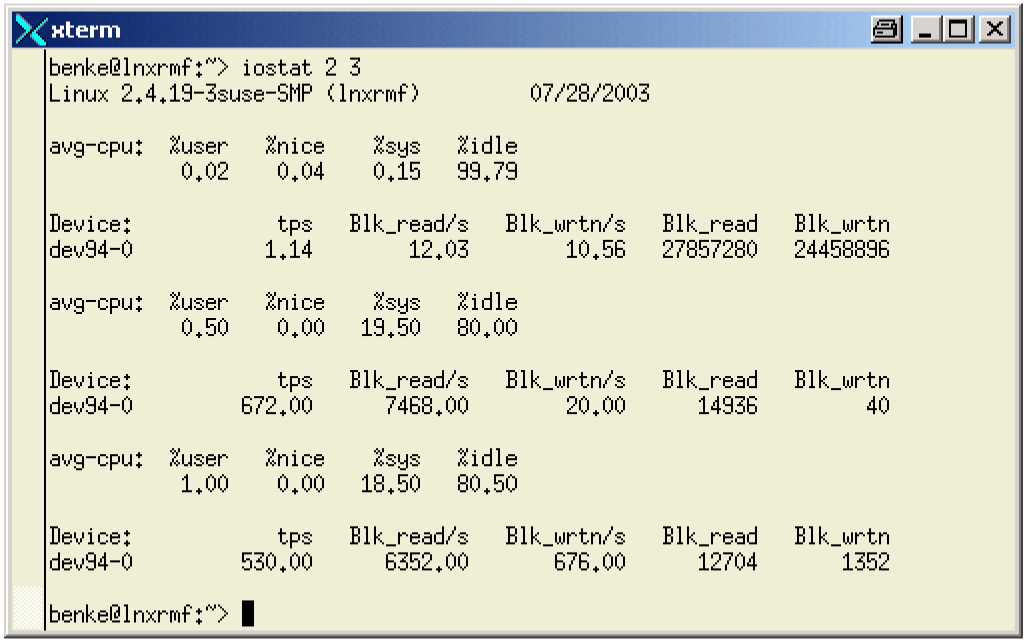
iostat

- ***iostat*** **is used to report CPU statistics and disk I/O statistics. The first parameter is the interval time in seconds, the second is the number of intervals to run, so “iostat 2 3” gives 3 samples with 2 seconds interval.**
- **As for vmstat, the first line reflects the summary of statistics since system IPL.**

tps: **number of I/O requests to the device per seconds**

Blk_read/s:**number of blocks (of indeterminate size) read per second**

Blk_wrtn/s**: number of blocks written per second**



```
benke@lnxrmf:~> iostat 2 3
Linux 2.4.19-3suse-SMP (lnxrmf)         07/28/2003

avg-cpu:  %user   %nice    %sys   %idle
           0.02    0.04    0.15   99.79

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
dev94-0           1.14        12.03        10.56   27857280   24458896

avg-cpu:  %user   %nice    %sys   %idle
           0.50    0.00   19.50   80.00

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
dev94-0         672.00      7468.00        20.00      14936         40

avg-cpu:  %user   %nice    %sys   %idle
           1.00    0.00   18.50   80.50

Device:            tps   Blk_read/s   Blk_wrtn/s   Blk_read   Blk_wrtn
dev94-0         530.00      6352.00       676.00      12704       1352

benke@lnxrmf:~> █
```

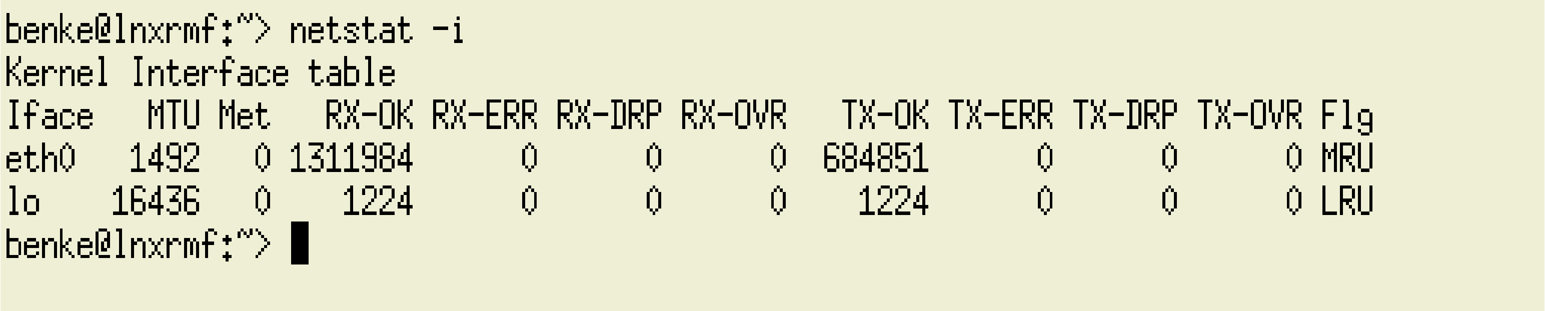
/proc/dasd/statistics

- **Only available in Linux for zSeries, kernel version 2.4**
- **Gathering of this information can be switched on and of, as it causes some overhead:**

```
echo set on > /proc/dasd/statistics
echo set off > /proc/dasd/statistics
```

- **Used in rmfpms to calculate the following metrics:**
  - dasd io average response time per request (in msec)
  - dasd io average response time per sector (in msec)
  - dasd io requests per second
- **More details can be found at**
  - http://www.ibm.com/developerworks/linux/linux390/perf/tuning_how_tools_dasd.html

Displaying Network Interface Statistics Overview



```
benke@lnxrmf:~> netstat -i
Kernel Interface table
Iface    MTU Met    RX-OK RX-ERR RX-DRP RX-OVR    TX-OK TX-ERR TX-DRP TX-OVR Flg
eth0   1492    0 1311984      0      0      0   684851      0      0      0 MRU
lo    16436    0    1224      0      0      0     1224      0      0      0 LRU
benke@lnxrmf:~> █
```

RX-OK, TX-OK: number of packets received/ transmitted without error

RX-ERR, TX-ERR: transfer with error

RX-DRP, TX-DRP: dropped packets

RX-OVR, TX-OVR: packets dropped because of overrun conditions

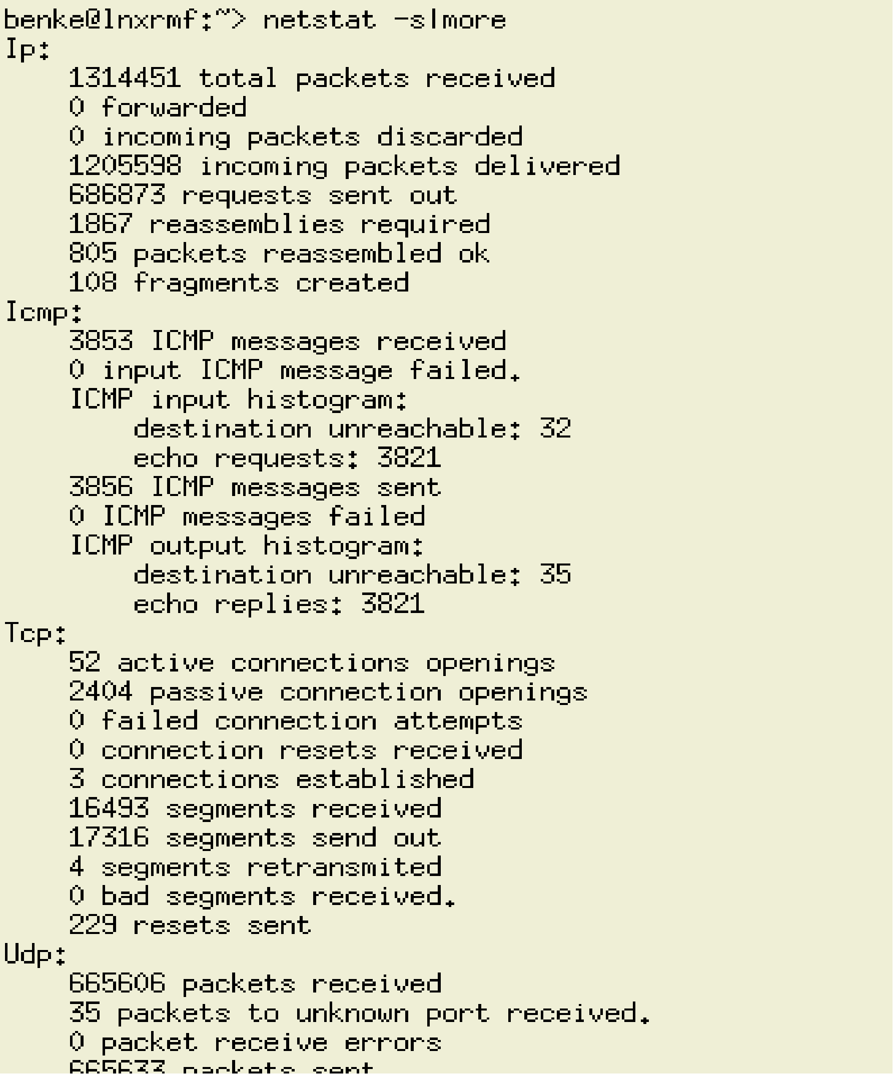
MTU, Met field: current MTU and Metric settings for this interface (Metric is used by the Routing Information Protocol RIP; MTU, Maximum Transmission Unit: max number of bytes transferred in one packet)

Flg: status, properties of the interface (R: running, U: up, …)

Iface: Name of the interface

Display Network Protocol Statistics

- **In contrast to “netstat –i”, which reports on network device level, “netstat –s” reports on network protocol level**
- **One advantage of this performance report is that it is less cryptic ;-) although there is a whole bunch on conditions gathered especially for the very important TCP protocol (not displayed here)**

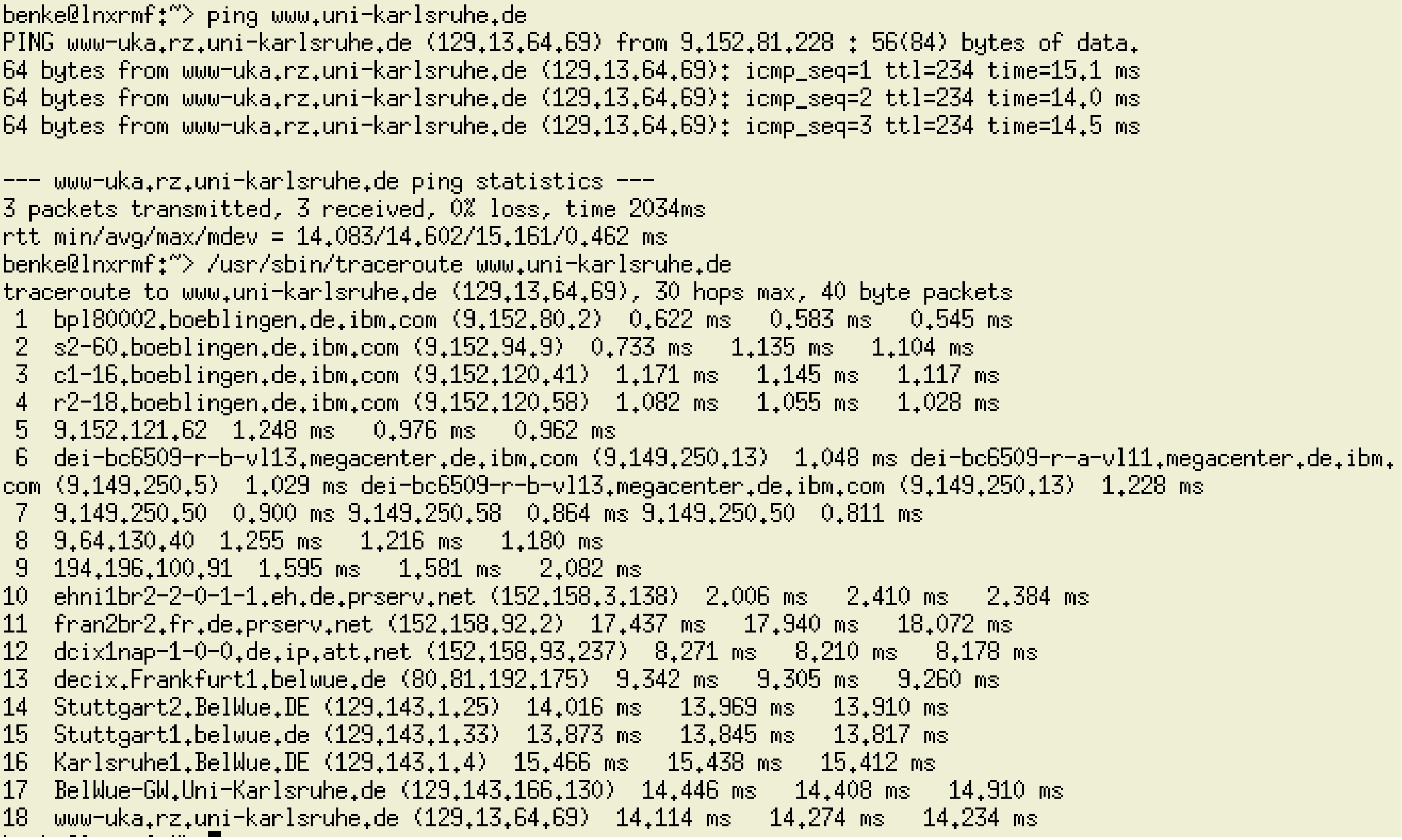


```
benke@lnxrmf:~> netstat -s|more
Ip:
    1314451 total packets received
    0 forwarded
    0 incoming packets discarded
    1205598 incoming packets delivered
    686873 requests sent out
    1867 reassemblies required
    805 packets reassembled ok
    108 fragments created
Icmp:
    3853 ICMP messages received
    0 input ICMP message failed.
    ICMP input histogram:
        destination unreachable: 32
        echo requests: 3821
    3856 ICMP messages sent
    0 ICMP messages failed
    ICMP output histogram:
        destination unreachable: 35
        echo replies: 3821
Tcp:
    52 active connections openings
    2404 passive connection openings
    0 failed connection attempts
    0 connection resets received
    3 connections established
    16493 segments received
    17316 segments send out
    4 segments retransmited
    0 bad segments received.
    229 resets sent
Udp:
    665606 packets received
    35 packets to unknown port received.
    0 packet receive errors
    665633 packets sent
```

ICMP Exploiter Applications

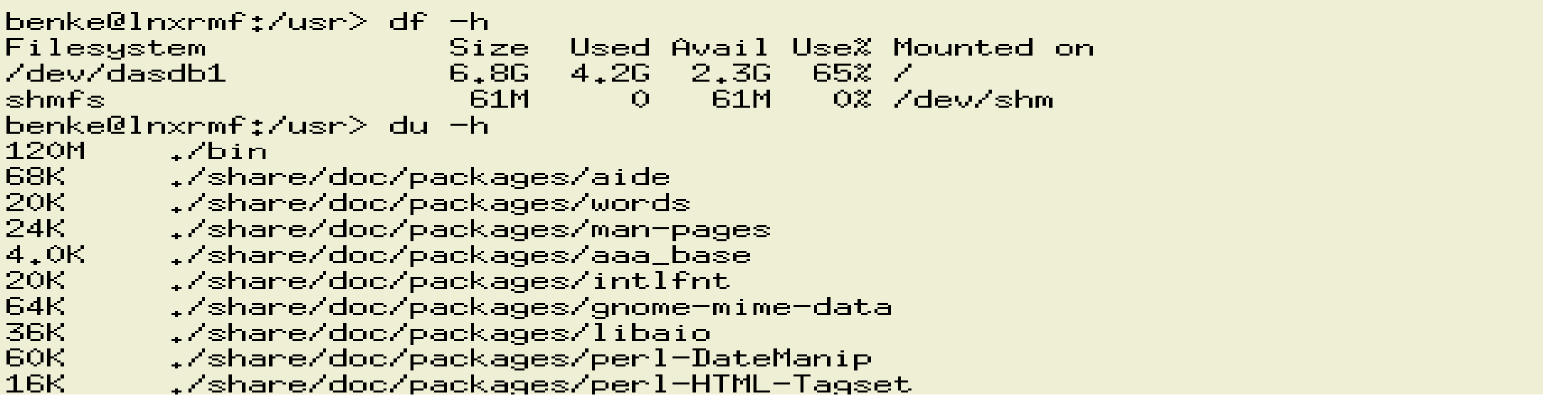
- **ICMP: Internet Control Message Protocol**
- ***ping* and *traceroute* are making use of the ICMP protocol in order to identify network problems.**
- ***ping* measures round-trip times between two hosts.**
- ***traceroute* – although a widely used UNIX command – is a hack, and so it does not always tell the truth. It tries to trace the way of packets through the network by sending around messages with short time to live (TTL) values.**
- **use “traceroute –q N” with N about 10 or higher if you want traceroute to sent more packets, in order to enhance precision of the reported numbers**

ping and traceroute examples



```
benke@lnxrmf:~> ping www.uni-karlsruhe.de
PING www-uka.rz.uni-karlsruhe.de (129.13.64.69) from 9.152.81.228 : 56(84) bytes of data.
64 bytes from www-uka.rz.uni-karlsruhe.de (129.13.64.69): icmp_seq=1 ttl=234 time=15.1 ms
64 bytes from www-uka.rz.uni-karlsruhe.de (129.13.64.69): icmp_seq=2 ttl=234 time=14.0 ms
64 bytes from www-uka.rz.uni-karlsruhe.de (129.13.64.69): icmp_seq=3 ttl=234 time=14.5 ms

--- www-uka.rz.uni-karlsruhe.de ping statistics ---
3 packets transmitted, 3 received, 0% loss, time 2034ms
rtt min/avg/max/mdev = 14.083/14.602/15.161/0.462 ms
benke@lnxrmf:~> /usr/sbin/traceroute www.uni-karlsruhe.de
traceroute to www.uni-karlsruhe.de (129.13.64.69), 30 hops max, 40 byte packets
 1  bpl80002.boeblingen.de.ibm.com (9.152.80.2)  0.622 ms   0.583 ms   0.545 ms
 2  s2-60.boeblingen.de.ibm.com (9.152.94.9)  0.733 ms   1.135 ms   1.104 ms
 3  c1-16.boeblingen.de.ibm.com (9.152.120.41)  1.171 ms   1.145 ms   1.117 ms
 4  r2-18.boeblingen.de.ibm.com (9.152.120.58)  1.082 ms   1.055 ms   1.028 ms
 5  9.152.121.62  1.248 ms   0.976 ms   0.962 ms
 6  dei-bc6509-r-b-vl13.megacenter.de.ibm.com (9.149.250.13)  1.048 ms dei-bc6509-r-a-vl11.megacenter.de.ibm.
com (9.149.250.5)  1.029 ms dei-bc6509-r-b-vl13.megacenter.de.ibm.com (9.149.250.13)  1.228 ms
 7  9.149.250.50  0.900 ms 9.149.250.58  0.864 ms 9.149.250.50  0.811 ms
 8  9.64.130.40  1.255 ms   1.216 ms   1.180 ms
 9  194.196.100.91  1.595 ms   1.581 ms   2.082 ms
10  ehni1br2-2-0-1-1.eh.de.prserv.net (152.158.3.138)  2.006 ms   2.410 ms   2.384 ms
11  fran2br2.fr.de.prserv.net (152.158.92.2)  17.437 ms   17.940 ms   18.072 ms
12  dcix1nap-1-0-0.de.ip.att.net (152.158.93.237)  8.271 ms   8.210 ms   8.178 ms
13  decix.Frankfurt1.belwue.de (80.81.192.175)  9.342 ms   9.305 ms   9.260 ms
14  Stuttgart2.BelWue.DE (129.143.1.25)  14.016 ms   13.969 ms   13.910 ms
15  Stuttgart1.belwue.de (129.143.1.33)  13.873 ms   13.845 ms   13.817 ms
16  Karlsruhe1.BelWue.DE (129.143.1.4)  15.466 ms   15.438 ms   15.412 ms
17  BelWue-GW.Uni-Karlsruhe.de (129.143.166.130)  14.446 ms   14.408 ms   14.910 ms
18  www-uka.rz.uni-karlsruhe.de (129.13.64.69)  14.114 ms   14.274 ms   14.234 ms
```

Filesystem Usage

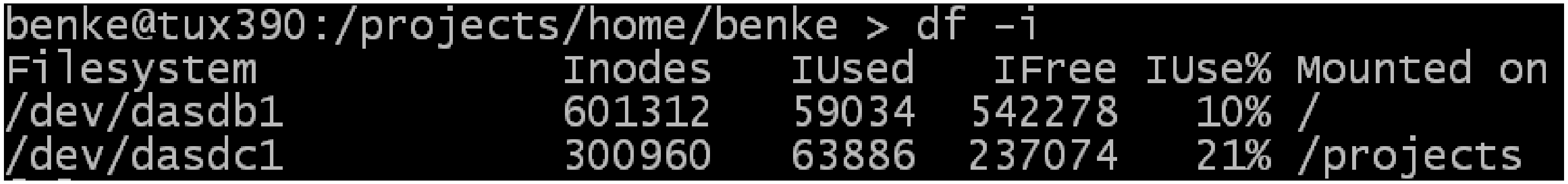


```
benke@lnxrmf:/usr> df -h
Filesystem            Size  Used Avail Use% Mounted on
/dev/dasdb1           6.8G  4.2G  2.3G  65% /
shmfs                  61M     0   61M   0% /dev/shm
benke@lnxrmf:/usr> du -h
120M    ./bin
68K     ./share/doc/packages/aide
20K     ./share/doc/packages/words
24K     ./share/doc/packages/man-pages
4.0K    ./share/doc/packages/aaa_base
20K     ./share/doc/packages/intlfnt
64K     ./share/doc/packages/gnome-mime-data
36K     ./share/doc/packages/libaio
60K     ./share/doc/packages/perl-DateManip
16K     ./share/doc/packages/perl-HTML-Tagset
```

- **The “-h” option stands for human readable. Without “-h”, reported numbers are bytes …**
- **The “df” command gives you a list of all mounted filesystems, corresponding to /dev/dasdxx devices.**
- **Using “du” you can see the amount of disk storage used in various directories. If you want a sum, use “-s” option.**

Inode Utilization

- In UNIX, an inode is a structure containing meta data about files and directories.
- The number of inodes is limited, can be changed at filesystem creation time.
- If you are running out of inodes, you can not store anything more on this filesystem.
- Check with "df -i" command:



```
benke@tux390:/projects/home/benke > df -i
Filesystem            Inodes   IUsed   IFree IUse% Mounted on
/dev/dasdb1           601312   59034  542278   10% /
/dev/dasdc1           300960   63886  237074   21% /projects
```

time

- **Find out how many CPU resources a command is using.**

Example:

```
$ > time make dep
...
72.52user 8.87system 2:03.72elapsed 65%CPU
(0avgtext+0avgdata 0maxresident)k 0inputs+0outputs
(131158major+106391minor) pagefaults 0swaps
$ >
```

elapsed**:** **real time elapse**

user**:** **time this command (and its children) have spent in user space**

sys**:** **time spent in kernel space**

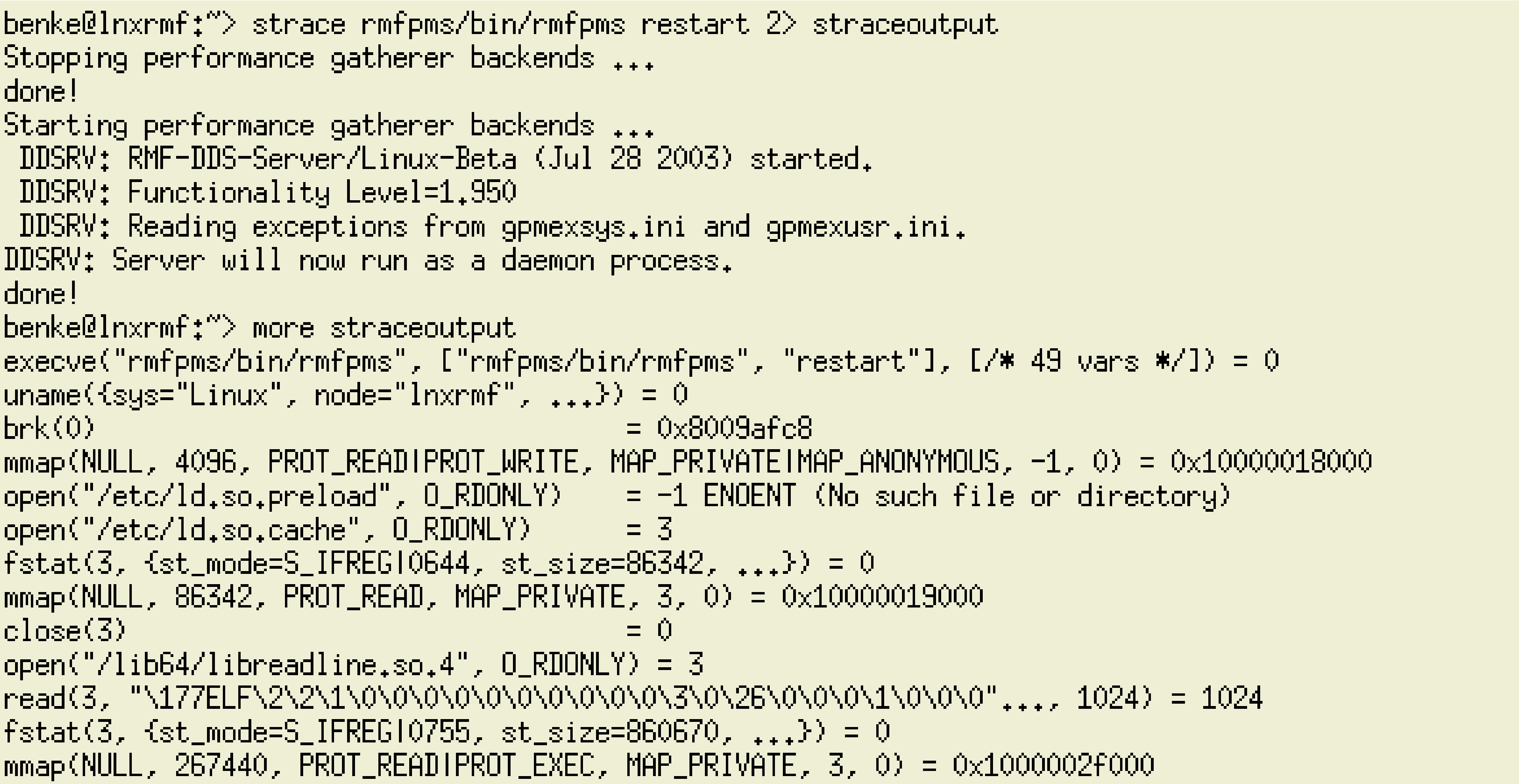
System Call Trace

- One of the commands more powerful than what we have for traditional mainframe operating systems, comes in very handy …
- strace allows to see the system calls a process is currently executing, so for example if you have the gut feeling a process with process ID PID 4711 is looping, you can execute

  strace –p 4711

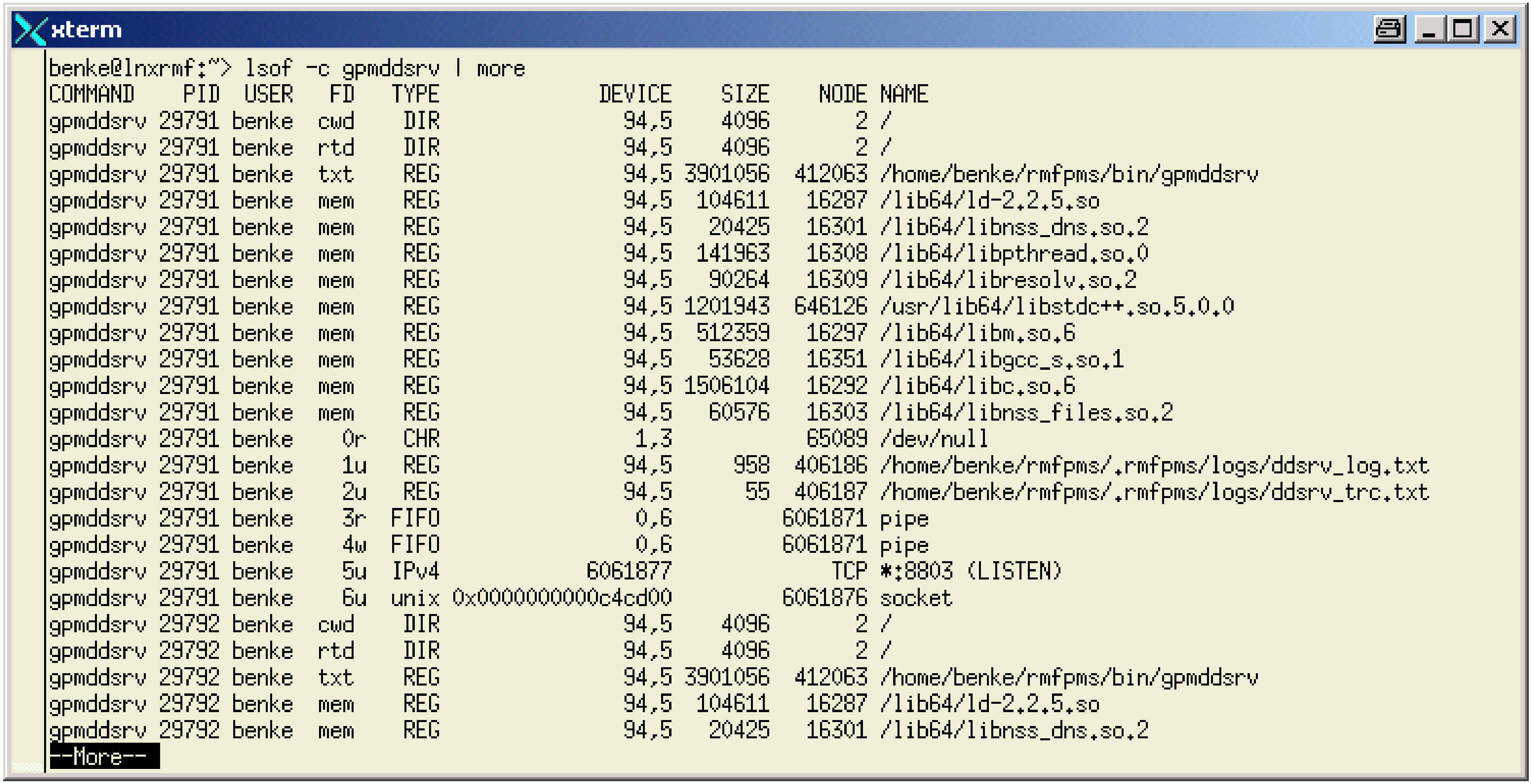
  in one terminal window; if it is a server process and it is not using any system calls but runs the CPU to 100% utilization, this is very suspicious, so you may think about killing this process

strace Example



```
benke@lnxrmf:~> strace rmfpms/bin/rmfpms restart 2> straceoutput
Stopping performance gatherer backends ...
done!
Starting performance gatherer backends ...
 DDSRV: RMF-DDS-Server/Linux-Beta (Jul 28 2003) started.
 DDSRV: Functionality Level=1.950
 DDSRV: Reading exceptions from gpmexsys.ini and gpmexusr.ini.
DDSRV: Server will now run as a daemon process.
done!
benke@lnxrmf:~> more straceoutput
execve("rmfpms/bin/rmfpms", ["rmfpms/bin/rmfpms", "restart"], [/* 49 vars */]) = 0
uname({sys="Linux", node="lnxrmf", ...}) = 0
brk(0)                                  = 0x8009afc8
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x1000018000
open("/etc/ld.so.preload", O_RDONLY)    = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=86342, ...}) = 0
mmap(NULL, 86342, PROT_READ, MAP_PRIVATE, 3, 0) = 0x1000019000
close(3)                                = 0
open("/lib64/libreadline.so.4", O_RDONLY) = 3
read(3, "\177ELF\2\2\1\0\0\0\0\0\0\0\0\0\0\3\0\26\0\0\0\1\0\0\0"..., 1024) = 1024
fstat(3, {st_mode=S_IFREG|0755, st_size=860670, ...}) = 0
mmap(NULL, 267440, PROT_READ|PROT_EXEC, MAP_PRIVATE, 3, 0) = 0x100002f000
```

List open files (*lsof*)



```
benke@lnxrmf:~> lsof -c gpmddsrv | more
COMMAND    PID  USER   FD   TYPE             DEVICE    SIZE    NODE NAME
gpmddsrv 29791 benke  cwd    DIR               94,5    4096       2 /
gpmddsrv 29791 benke  rtd    DIR               94,5    4096       2 /
gpmddsrv 29791 benke  txt    REG               94,5 3901056  412063 /home/benke/rmfpms/bin/gpmddsrv
gpmddsrv 29791 benke  mem    REG               94,5  104611   16287 /lib64/ld-2.2.5.so
gpmddsrv 29791 benke  mem    REG               94,5   20425   16301 /lib64/libnss_dns.so.2
gpmddsrv 29791 benke  mem    REG               94,5  141963   16308 /lib64/libpthread.so.0
gpmddsrv 29791 benke  mem    REG               94,5   90264   16309 /lib64/libresolv.so.2
gpmddsrv 29791 benke  mem    REG               94,5 1201943  646126 /usr/lib64/libstdc++.so.5.0.0
gpmddsrv 29791 benke  mem    REG               94,5  512359   16297 /lib64/libm.so.6
gpmddsrv 29791 benke  mem    REG               94,5   53628   16351 /lib64/libgcc_s.so.1
gpmddsrv 29791 benke  mem    REG               94,5 1506104   16292 /lib64/libc.so.6
gpmddsrv 29791 benke  mem    REG               94,5   60576   16303 /lib64/libnss_files.so.2
gpmddsrv 29791 benke    0r   CHR                1,3           65089 /dev/null
gpmddsrv 29791 benke    1u   REG               94,5     958  406186 /home/benke/rmfpms/.rmfpms/logs/ddsrv_log.txt
gpmddsrv 29791 benke    2u   REG               94,5      55  406187 /home/benke/rmfpms/.rmfpms/logs/ddsrv_trc.txt
gpmddsrv 29791 benke    3r  FIFO                0,6         6061871 pipe
gpmddsrv 29791 benke    4w  FIFO                0,6         6061871 pipe
gpmddsrv 29791 benke    5u  IPv4            6061877             TCP *:8803 (LISTEN)
gpmddsrv 29791 benke    6u  unix 0x0000000000c4cd00         6061876 socket
gpmddsrv 29792 benke  cwd    DIR               94,5    4096       2 /
gpmddsrv 29792 benke  rtd    DIR               94,5    4096       2 /
gpmddsrv 29792 benke  txt    REG               94,5 3901056  412063 /home/benke/rmfpms/bin/gpmddsrv
gpmddsrv 29792 benke  mem    REG               94,5  104611   16287 /lib64/ld-2.2.5.so
gpmddsrv 29792 benke  mem    REG               94,5   20425   16301 /lib64/libnss_dns.so.2
--More--
```

lsof explained

- **For UNIX, everything is a file. Directories, inter-process communication structures (like pipes), network sockets and regular files are all files. "lsof" can list all file usages.**
- **Some useful usage examples of lsof:**
  - List all files by processes with name "gpmddsrv":
    lsof –c gpmddsrv
  - List all TCP/IP v4 network connections to host "tux390.boeblingen.de.ibm.com":
  - **lsof –i4tcp@tux390.boeblingen.de.ibm.com**
  - List all files using /var/log:
  - **lsof –t /var/log**

Lock Contention

- **/var/lock is the standard location to place lock files, so have a look what's in it**
- **The "ipcs" gives a summary on shared memory segments, semaphores and message queues the calling user has read access to. As "ipcs" only displays locks the calling user has read access to, you may run it as user root.**
- **You may also check "/proc/locks" if you suspect there is some locking problem. Unfortunately, Linux supports several ways of locking, and I don't know a single place where all locks and lock contentions are displayed.**

BSD Accounting

- **Writes one accounting record per terminated process or thread (as threads are something like processes in Linux...)**
- **Information provided:**
  - –user ID, group ID, process name
  - –CPU resource consumption
  - –average memory usage, page faults, swap activity
- **An alternative to accounting Linux "from the inside" is accounting it "from the outside", with the aid of z/VM or z/OS performance tools**

“sysstat” package

- Contains sar and sadc, long term data collector
- Normally, it collects data about overall system activity like CPU usage, swapping; no data about processes
- start with

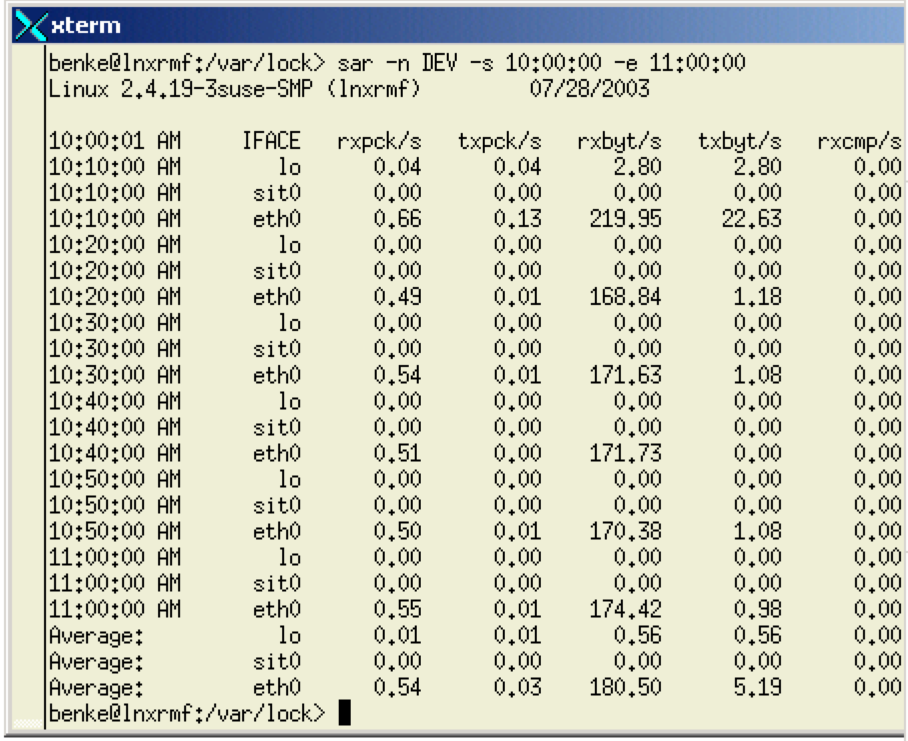
```
$ > sadc 60 /var/log/sa/sa25 &
```

- to let it generate one report every 60 seconds and write it in binary format to /var/log/sa/sa25
- *http://freshmeat.net/projects/sysstat/*

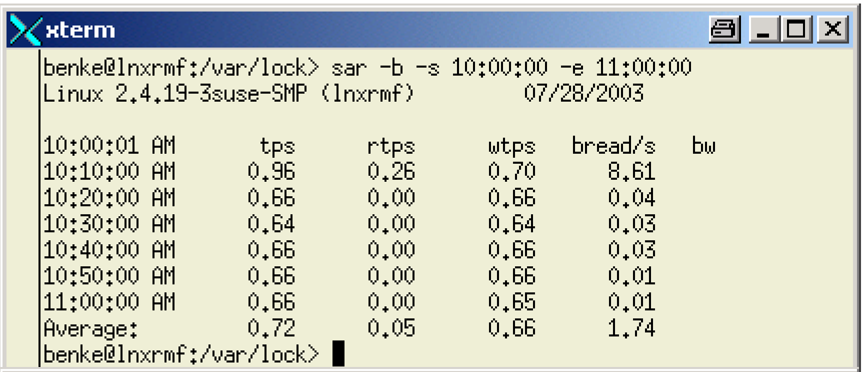
sar: some options

CPU	sar -u	CPU Utilization Data: %user, %nice, %system, %idle
	sar –U <n>	Like “sar –u”, but only for CPU number <n>
	sar –c	Process creation rate
	sar –w	Context switch rate
Mem	sar –r	Memory and swap space utilization
	sar –R	Memory usage statistics (buffer growth, …)
	sar -B	Paging statistics
	sar –w	Swapping activity
I/O	sar –b	I/O and transfer rate statistics
	sar –d	Block device statistics
	sar –n DEV	Network device statistics
	sar –n EDEV	Network device error rates
	sar –n SOCK	Socket statistics

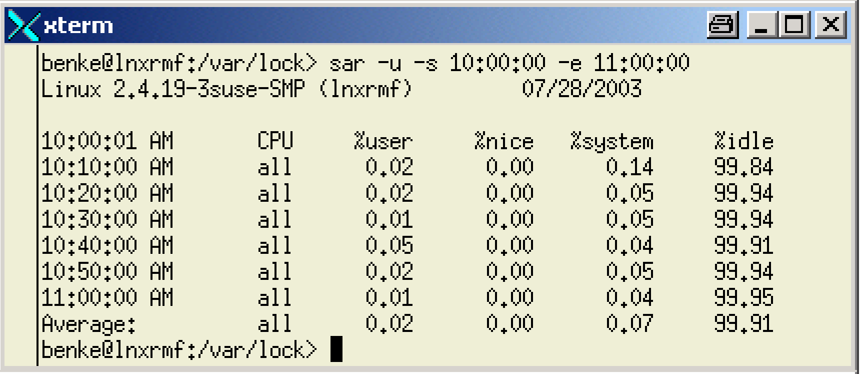
sar: some examples



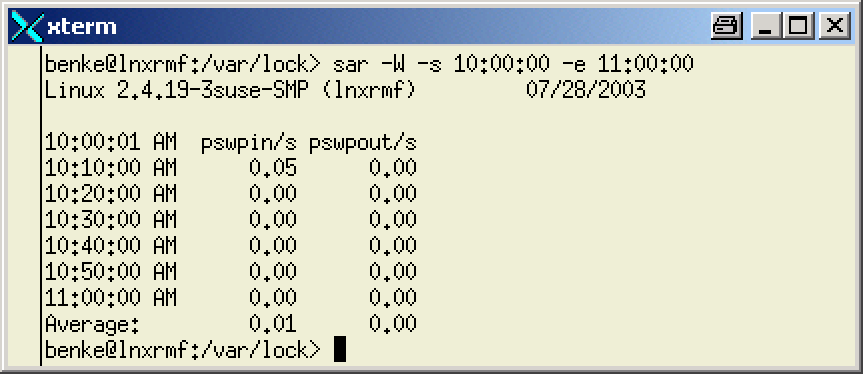
```
benke@lnxrmf:/var/lock> sar -n DEV -s 10:00:00 -e 11:00:00
Linux 2.4.19-3suse-SMP (lnxrmf)          07/28/2003

10:00:01 AM     IFACE   rxpck/s   txpck/s   rxbyt/s   txbyt/s   rxcmp/s
10:10:00 AM        lo      0.04      0.04      2.80      2.80      0.00
10:10:00 AM      sit0      0.00      0.00      0.00      0.00      0.00
10:10:00 AM      eth0      0.66      0.13    219.95     22.63      0.00
10:20:00 AM        lo      0.00      0.00      0.00      0.00      0.00
10:20:00 AM      sit0      0.00      0.00      0.00      0.00      0.00
10:20:00 AM      eth0      0.49      0.01    168.84      1.18      0.00
10:30:00 AM        lo      0.00      0.00      0.00      0.00      0.00
10:30:00 AM      sit0      0.00      0.00      0.00      0.00      0.00
10:30:00 AM      eth0      0.54      0.01    171.63      1.08      0.00
10:40:00 AM        lo      0.00      0.00      0.00      0.00      0.00
10:40:00 AM      sit0      0.00      0.00      0.00      0.00      0.00
10:40:00 AM      eth0      0.51      0.00    171.73      0.00      0.00
10:50:00 AM        lo      0.00      0.00      0.00      0.00      0.00
10:50:00 AM      sit0      0.00      0.00      0.00      0.00      0.00
10:50:00 AM      eth0      0.50      0.01    170.38      1.08      0.00
11:00:00 AM        lo      0.00      0.00      0.00      0.00      0.00
11:00:00 AM      sit0      0.00      0.00      0.00      0.00      0.00
11:00:00 AM      eth0      0.55      0.01    174.42      0.98      0.00
Average:           lo      0.01      0.01      0.56      0.56      0.00
Average:         sit0      0.00      0.00      0.00      0.00      0.00
Average:         eth0      0.54      0.03    180.50      5.19      0.00
benke@lnxrmf:/var/lock>
```



```
benke@lnxrmf:/var/lock> sar -b -s 10:00:00 -e 11:00:00
Linux 2.4.19-3suse-SMP (lnxrmf)          07/28/2003

10:00:01 AM       tps      rtps      wtps   bread/s    bw
10:10:00 AM      0.96      0.26      0.70      8.61
10:20:00 AM      0.66      0.00      0.66      0.04
10:30:00 AM      0.64      0.00      0.64      0.03
10:40:00 AM      0.66      0.00      0.66      0.03
10:50:00 AM      0.66      0.00      0.66      0.01
11:00:00 AM      0.66      0.00      0.65      0.01
Average:         0.72      0.05      0.66      1.74
benke@lnxrmf:/var/lock>
```



```
benke@lnxrmf:/var/lock> sar -u -s 10:00:00 -e 11:00:00
Linux 2.4.19-3suse-SMP (lnxrmf)          07/28/2003

10:00:01 AM       CPU     %user     %nice   %system     %idle
10:10:00 AM       all      0.02      0.00      0.14     99.84
10:20:00 AM       all      0.02      0.00      0.05     99.94
10:30:00 AM       all      0.01      0.00      0.05     99.94
10:40:00 AM       all      0.05      0.00      0.04     99.91
10:50:00 AM       all      0.02      0.00      0.05     99.94
11:00:00 AM       all      0.01      0.00      0.04     99.95
Average:          all      0.02      0.00      0.07     99.91
benke@lnxrmf:/var/lock>
```



```
benke@lnxrmf:/var/lock> sar -W -s 10:00:00 -e 11:00:00
Linux 2.4.19-3suse-SMP (lnxrmf)          07/28/2003

10:00:01 AM  pswpin/s pswpout/s
10:10:00 AM      0.05      0.00
10:20:00 AM      0.00      0.00
10:30:00 AM      0.00      0.00
10:40:00 AM      0.00      0.00
10:50:00 AM      0.00      0.00
11:00:00 AM      0.00      0.00
Average:         0.01      0.00
benke@lnxrmf:/var/lock>
```

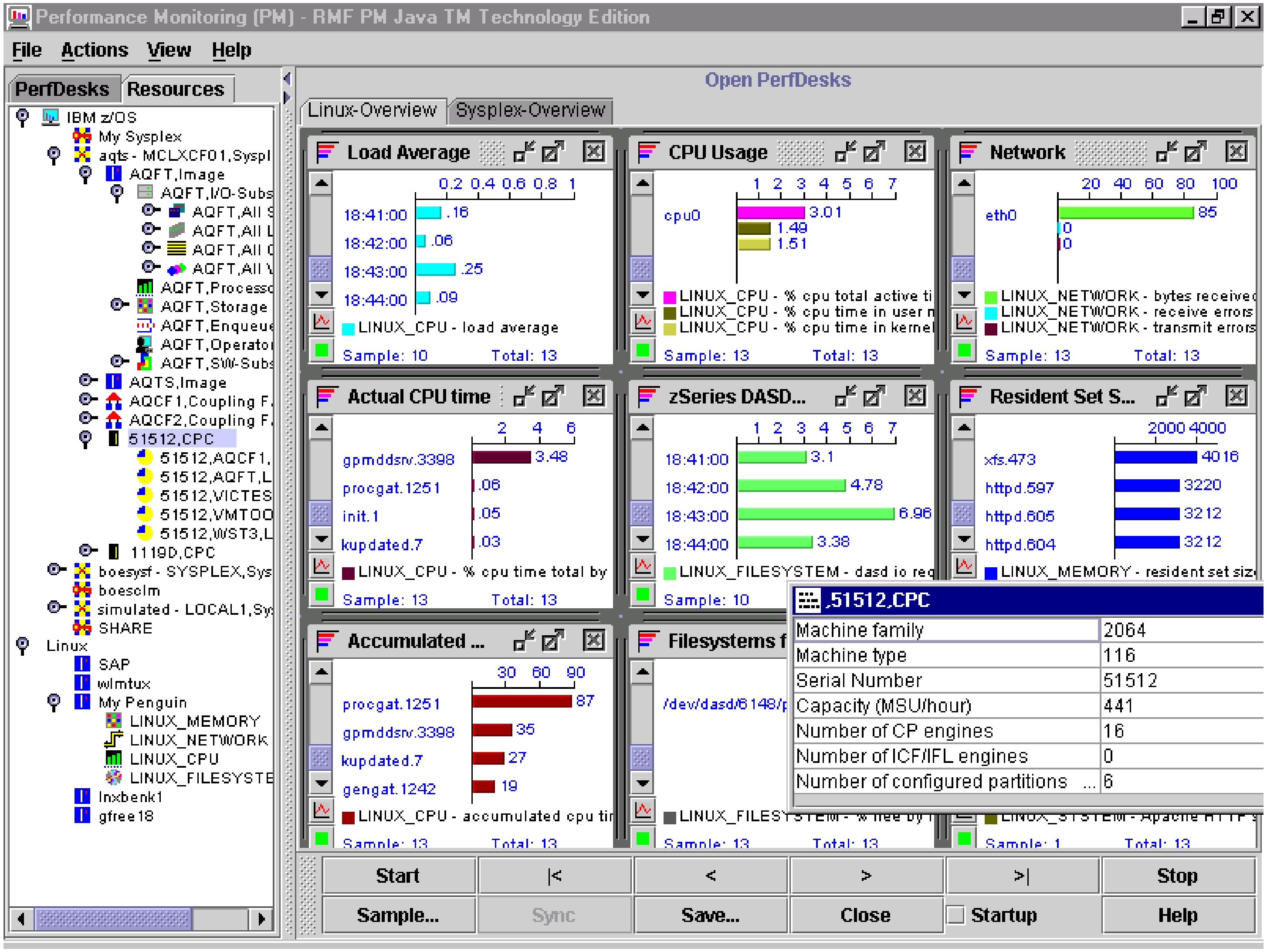
RMFPMS

- Long term data gathering
- XML over HTTP interface
- independent from z/OS; with z/OS, you can also have an LDAP interface to Linux performance data
- Modular architecture
- zSeries specific information (like LPAR data) can be obtained using existing z/VM or z/OS code
- Integrated with z/OS RMF PM and z/VM Performance Toolkit
- see
  - *http://www.ibm.com/eserver/zseries/zos/rmf/rmfhtmls/pmweb/pmlin.htm*

rmfpms (Linux data gathering) – recent updates

- **New script to automatically start Linux gatherer at Linux guest IPL (boot) time (“*enable_autostart*”); in addition, this scripts moves rmfpms to */var/opt/rmfpms* and */opt/rmfpms* in conjunction with Linux standards and it user user ID nobody for security reasons**
- **New “*delete_old_perfdata*” script to delete old Linux performance data archives**
- **Automatic repository compression now also applied for those customers which did not install a specific *cronjob* as described in the documentation**

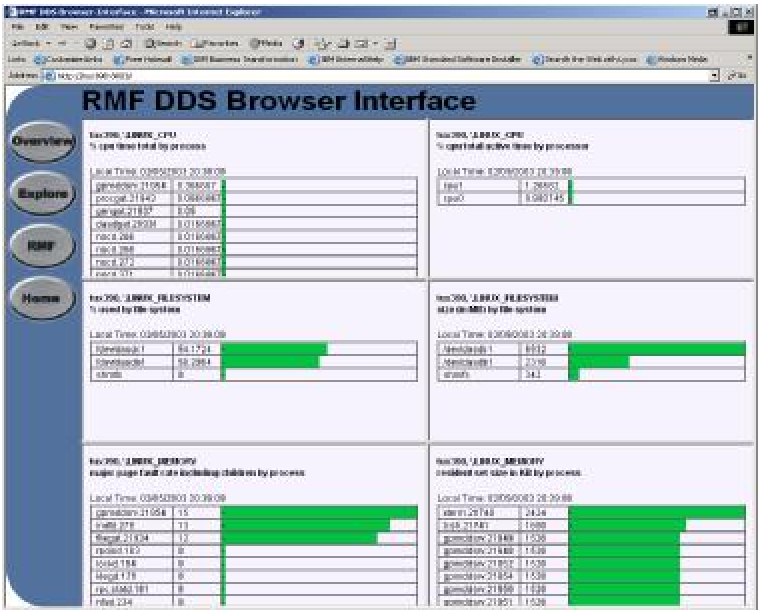
RMF PM Java Client



Machine family	2064
Machine type	116
Serial Number	51512
Capacity (MSU/hour)	441
Number of CP engines	16
Number of ICF/IFL engines	0
Number of configured partitions ...	6

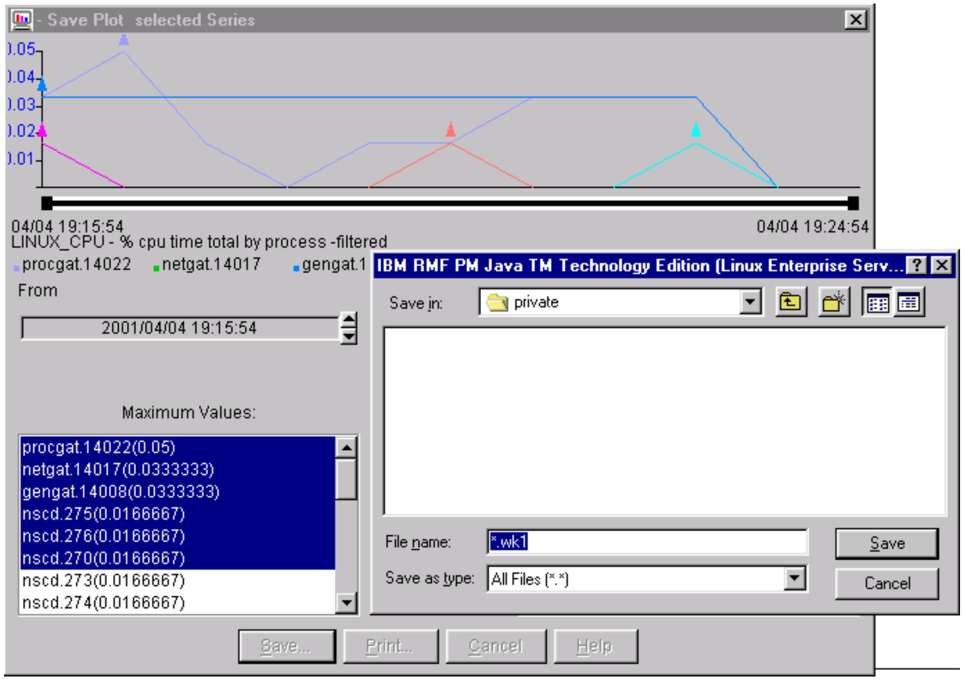
RMF PM Java Client: Features

- Positioned for online performance analysis and problem drill-down
- Can monitor multiple Linux server and multiple z/OS or OS/390 Sysplexes at the same time, in one application
- The performance analysis scenario can be saved
- Alternatively, you may use the web browser interface of the Distributed Data Server (DDS)



RMF DDS Browser Interface

RMF PM: Spreadsheet Data


- Save Plot selected Series
).05
).04
).03
).02
).01
Save...
Print...
Cancel
Help
IBM RMF PM Java TM Technology Edition (Linux Enterprise Serv...
Save
Cancel

04/04 19:15:54

04/04 19:24:54

LINUX_CPU - % cpu time total by process -filtered

procgat.14022 netgat.14017 gengat.1

From: 2001/04/04 19:15:54

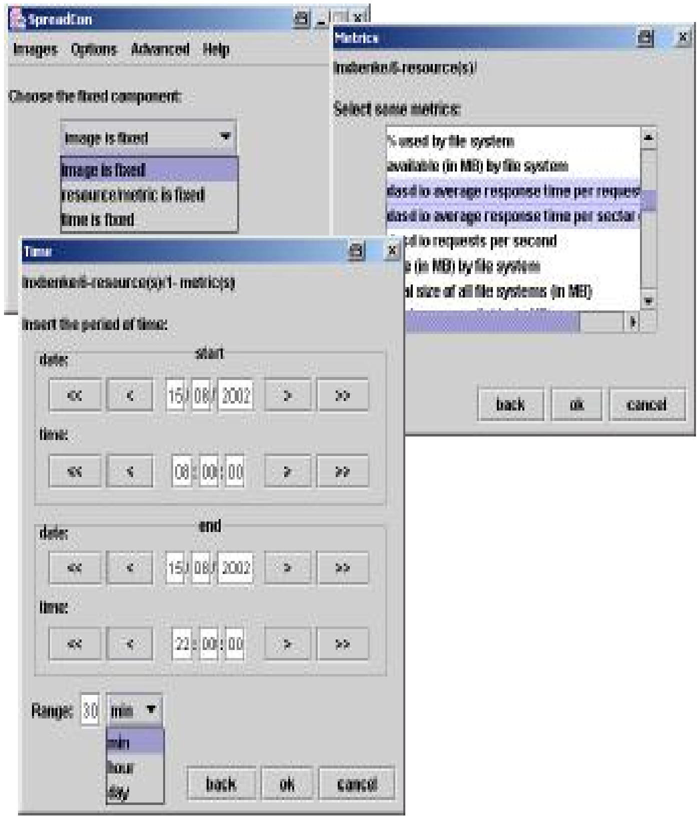
Maximum Values:

procgat.14022(0.05)
netgat.14017(0.0333333)
gengat.14008(0.0333333)
nscd.275(0.0166667)
nscd.276(0.0166667)
nscd.270(0.0166667)
nscd.273(0.0166667)
nscd.274(0.0166667)

Save in: private

File name: *.wk1

Save as type: All Files (*.*)



Choose the fixed component: image is fixed

image is fixed
resource/metric is fixed
time is fixed

lnxbenk/6-resource(s)/

Select some metrics:

% used by file system
available (in MB) by file system
dasd io average response time per reques
dasd io average response time per sector
dasd io requests per second
(in MB) by file system
total size of all file systems (in MB)

lnxbenkei-resource(s)/1- metric(s)

Insert the period of time:

start

date: 15/08/2002

time: 08:00:00

end

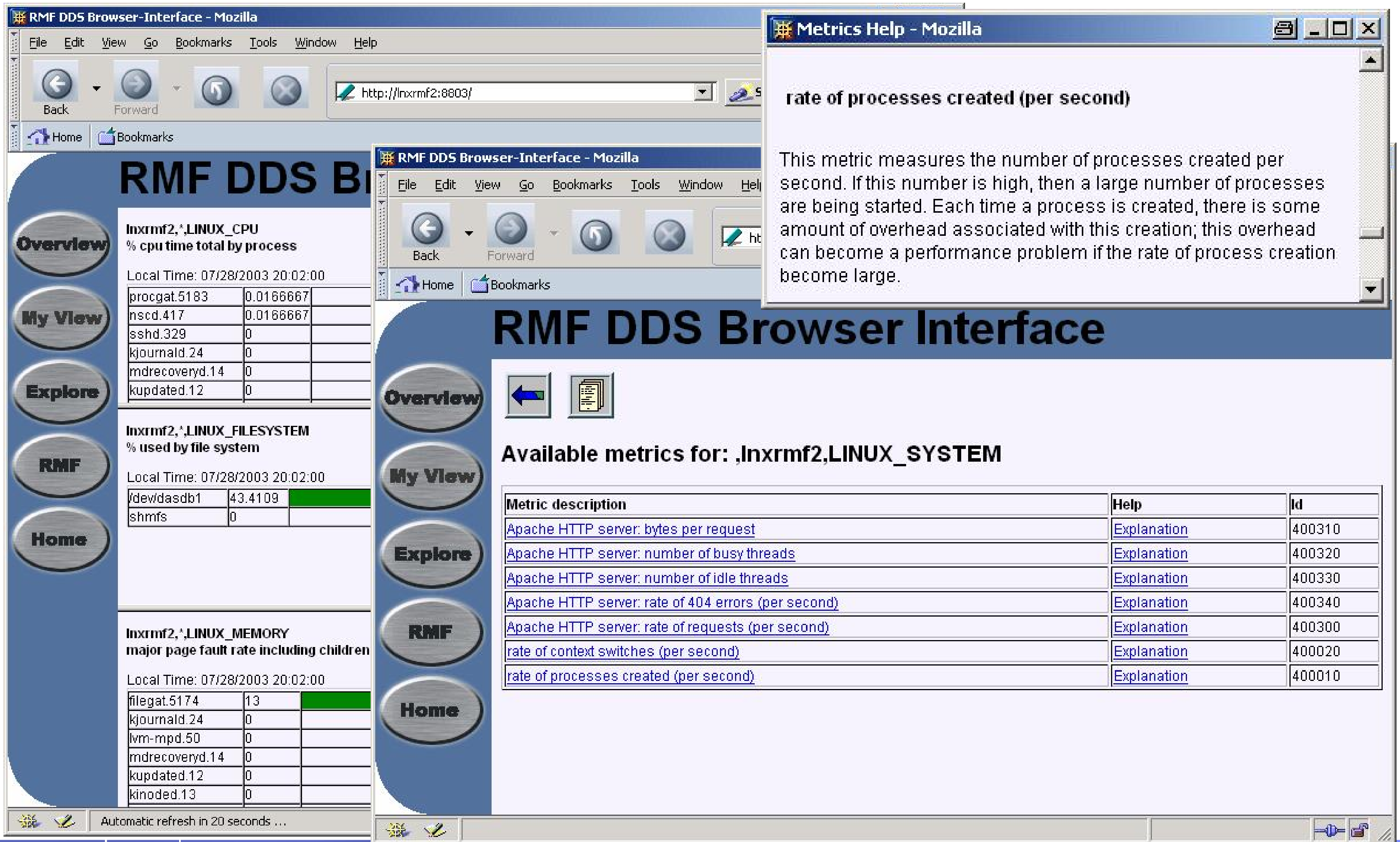
date: 15/08/2002

time: 22:00:00

Range: 30 min

min
hour
day

Enhanced RMFPMS Web Browser Interface


RMF DDS Browser-Interface - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back
Forward
http://lnxrmf2:8803/
Home
Bookmarks
RMF DDS Br
Overview
My View
Explore
RMF
Home
Automatic refresh in 20 seconds ...
RMF DDS Browser Interface
Metrics Help - Mozilla

lnxrmf2,*,LINUX_CPU
% cpu time total by process

Local Time: 07/28/2003 20:02:00

procgat.5183	0.0166667
nscd.417	0.0166667
sshd.329	0
kjournald.24	0
mdrecoveryd.14	0
kupdated.12	0

lnxrmf2,*,LINUX_FILESYSTEM
% used by file system

Local Time: 07/28/2003 20:02:00

/dev/dasdb1	43.4109
shmfs	0

lnxrmf2,*,LINUX_MEMORY
major page fault rate including children

Local Time: 07/28/2003 20:02:00

filegat.5174	13
kjournald.24	0
lvm-mpd.50	0
mdrecoveryd.14	0
kupdated.12	0
kinoded.13	0

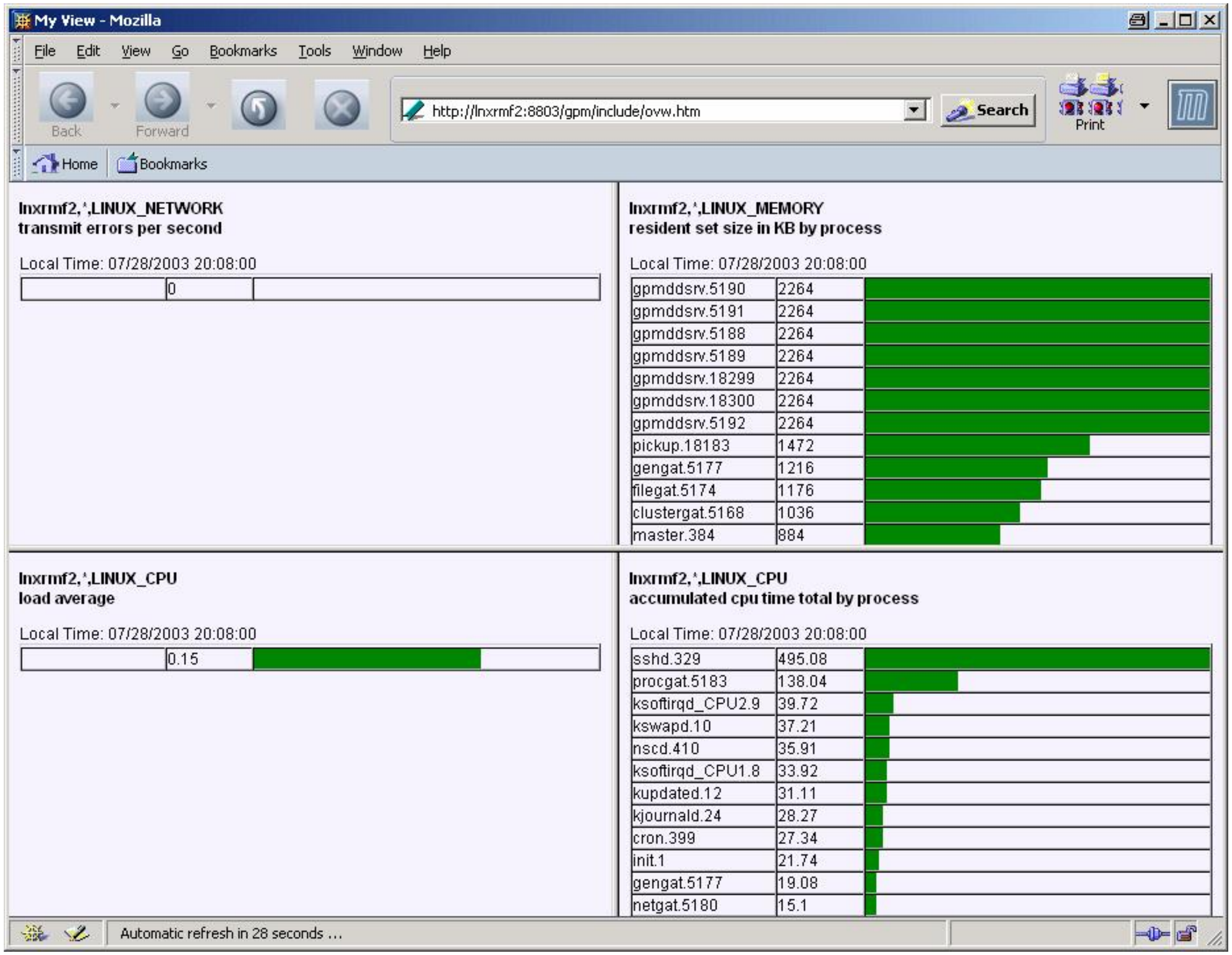
Available metrics for: ,lnxrmf2,LINUX_SYSTEM

Metric description	Help	Id
Apache HTTP server: bytes per request	Explanation	400310
Apache HTTP server: number of busy threads	Explanation	400320
Apache HTTP server: number of idle threads	Explanation	400330
Apache HTTP server: rate of 404 errors (per second)	Explanation	400340
Apache HTTP server: rate of requests (per second)	Explanation	400300
rate of context switches (per second)	Explanation	400020
rate of processes created (per second)	Explanation	400010

rate of processes created (per second)

This metric measures the number of processes created per second. If this number is high, then a large number of processes are being started. Each time a process is created, there is some amount of overhead associated with this creation; this overhead can become a performance problem if the rate of process creation become large.

… you can now create your own customizable view even in a Web browser like Mozilla, Explorer, Netscape



http://lnxrmf2:8803/gpm/include/ovw.htm

lnxrmf2,*,LINUX_NETWORK
transmit errors per second

Local Time: 07/28/2003 20:08:00

	0

lnxrmf2,*,LINUX_MEMORY
resident set size in KB by process

Local Time: 07/28/2003 20:08:00

gpmddsrv.5190	2264
gpmddsrv.5191	2264
gpmddsrv.5188	2264
gpmddsrv.5189	2264
gpmddsrv.18299	2264
gpmddsrv.18300	2264
gpmddsrv.5192	2264
pickup.18183	1472
gengat.5177	1216
filegat.5174	1176
clustergat.5168	1036
master.384	884

lnxrmf2,*,LINUX_CPU
load average

Local Time: 07/28/2003 20:08:00

	0.15

lnxrmf2,*,LINUX_CPU
accumulated cpu time total by process

Local Time: 07/28/2003 20:08:00

sshd.329	495.08
procgat.5183	138.04
ksoftirqd_CPU2.9	39.72
kswapd.10	37.21
nscd.410	35.91
ksoftirqd_CPU1.8	33.92
kupdated.12	31.11
kjournald.24	28.27
cron.399	27.34
init.1	21.74
gengat.5177	19.08
netgat.5180	15.1

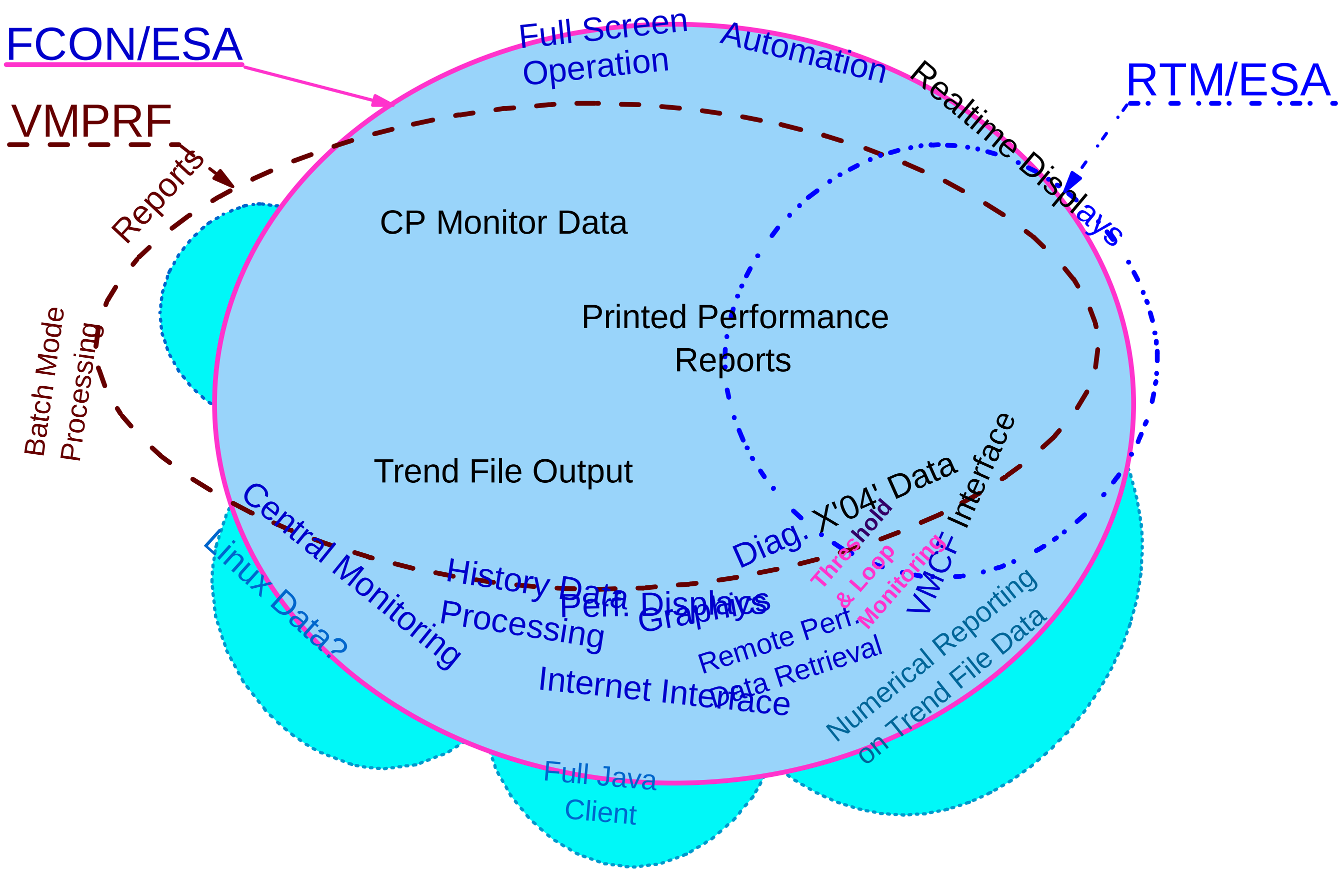
Automatic refresh in 28 seconds ...

Linux monitor stream support for z/VM

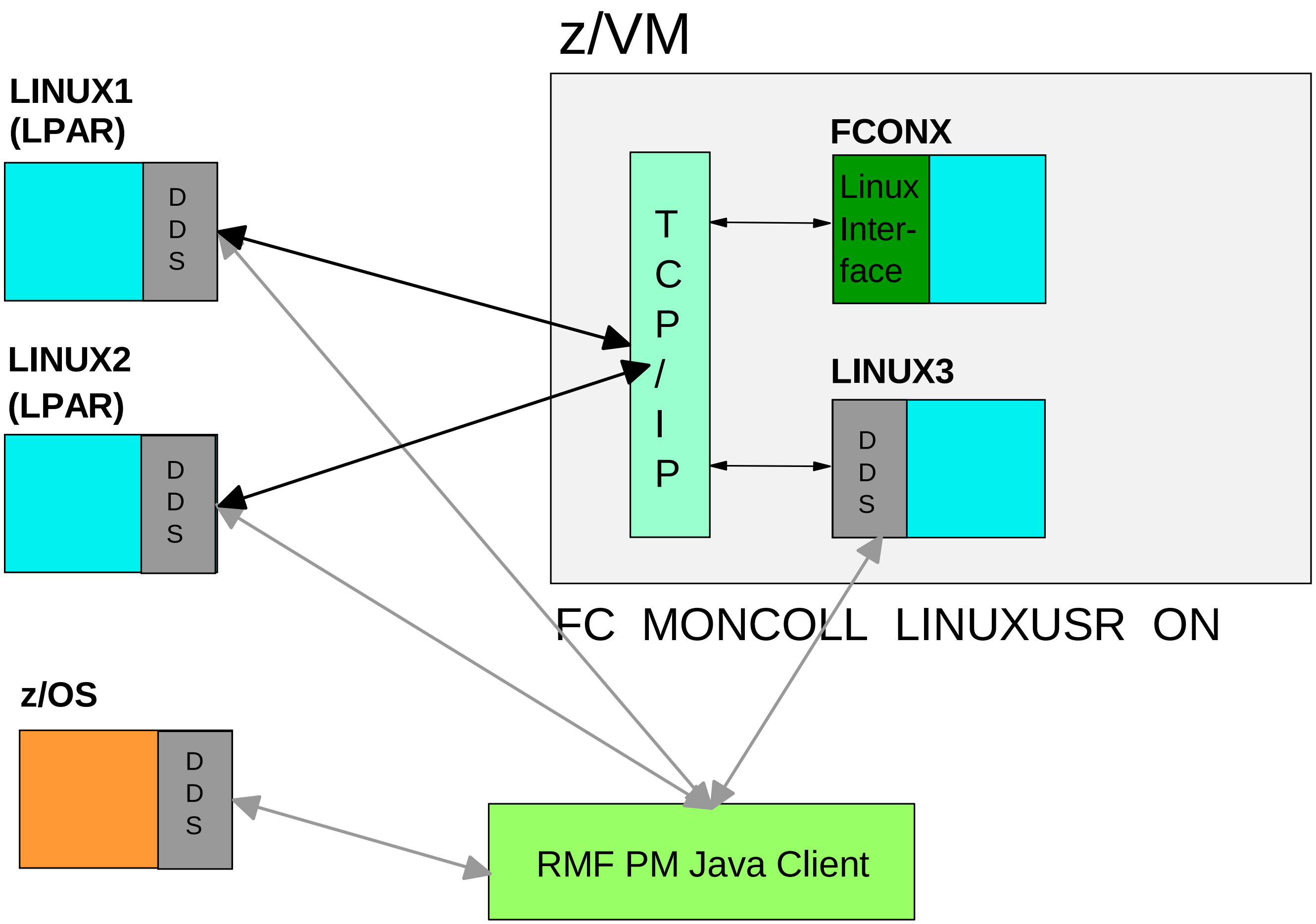
- **Based on virtual CPU timer**
  - This timer only ticks if the Linux image consumes CPU resources
  - Advantage: you consume a given percentage of a virtual server's CPU resources for monitoring, not a given percentage of the physical box (this way, reducing scalability by doing performance monitoring)
  - Expect more like this to come
- **Feed Linux performance data into normal z/VM performance monitoring infrastructure (APPLDATA interface)**

z/VM FCON

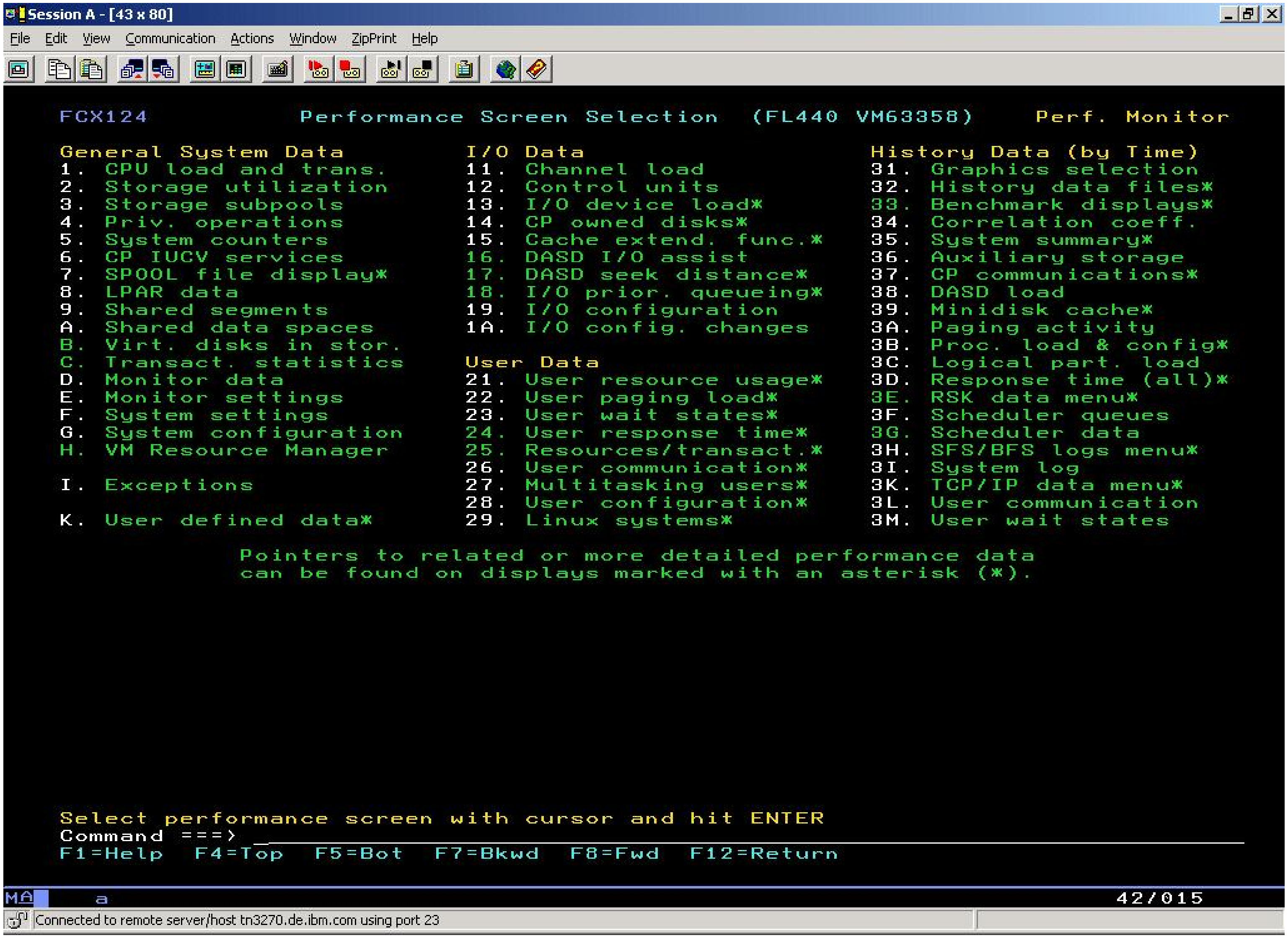
Linux patch for z/VM Performance Toolkit:
http://oss.software.ibm.com/developerworks/opensource/linux390/index.shtml



Accessing Linux Performance Data: Concept



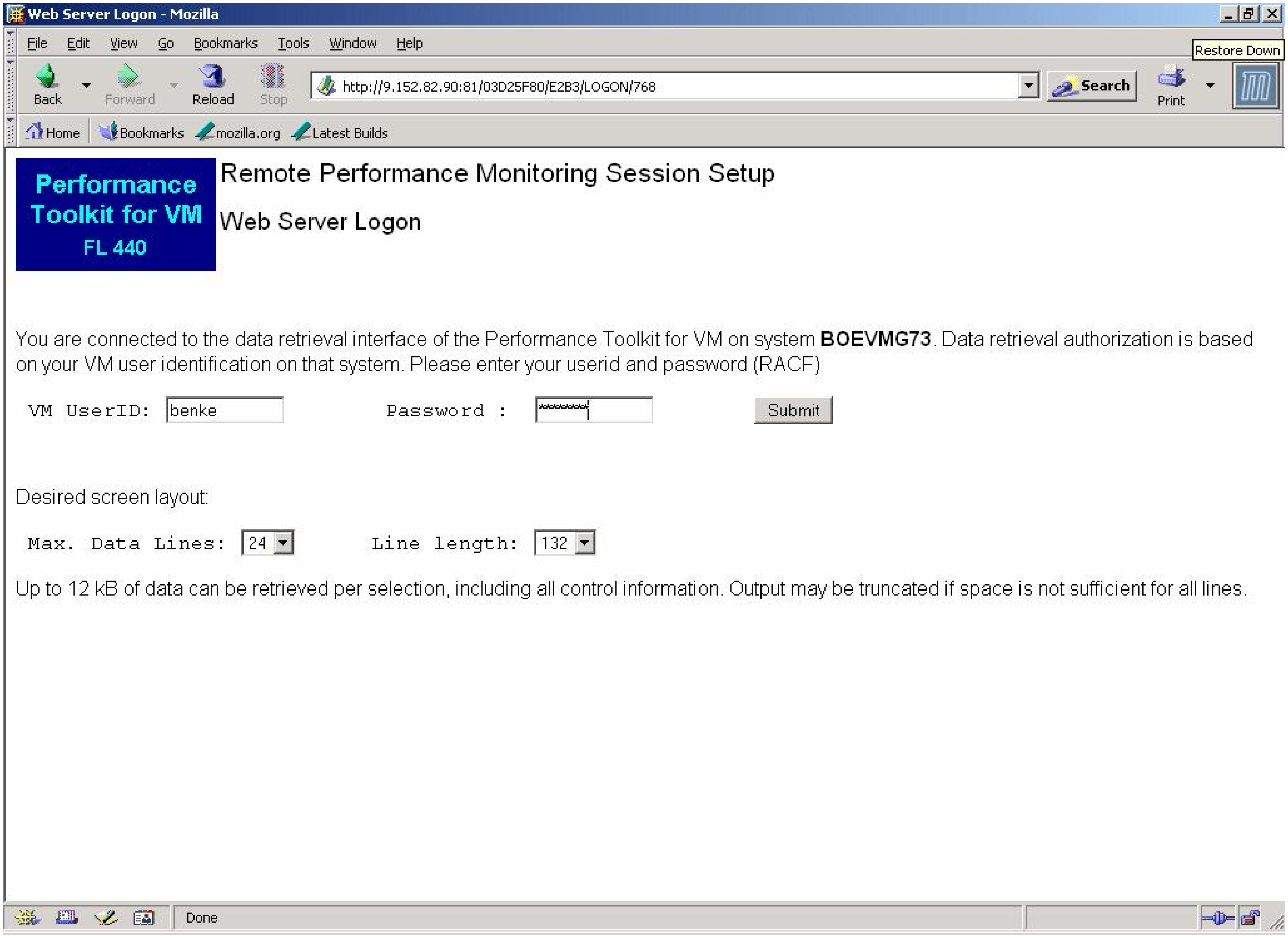
z/VM Performance Toolkit 3270 Startup Screen



```
FCX124          Performance Screen Selection  (FL440 VM63358)    Perf. Monitor

General System Data        I/O Data                   History Data (by Time)
1. CPU load and trans.     11. Channel load           31. Graphics selection
2. Storage utilization     12. Control units          32. History data files*
3. Storage subpools        13. I/O device load*       33. Benchmark displays*
4. Priv. operations        14. CP owned disks*        34. Correlation coeff.
5. System counters         15. Cache extend. func.*   35. System summary*
6. CP IUCV services        16. DASD I/O assist        36. Auxiliary storage
7. SPOOL file display*     17. DASD seek distance*    37. CP communications*
8. LPAR data               18. I/O prior. queueing*   38. DASD load
9. Shared segments         19. I/O configuration      39. Minidisk cache*
A. Shared data spaces      1A. I/O config. changes    3A. Paging activity
B. Virt. disks in stor.                               3B. Proc. load & config*
C. Transact. statistics    User Data                  3C. Logical part. load
D. Monitor data            21. User resource usage*   3D. Response time (all)*
E. Monitor settings        22. User paging load*      3E. RSK data menu*
F. System settings         23. User wait states*      3F. Scheduler queues
G. System configuration    24. User response time*    3G. Scheduler data
H. VM Resource Manager     25. Resources/transact.*   3H. SFS/BFS logs menu*
                           26. User communication*    3I. System log
I. Exceptions              27. Multitasking users*    3K. TCP/IP data menu*
                           28. User configuration*    3L. User communication
K. User defined data*      29. Linux systems*         3M. User wait states

            Pointers to related or more detailed performance data
            can be found on displays marked with an asterisk (*).












Select performance screen with cursor and hit ENTER
Command ===> _
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return
MA    a                                                          42/015
```

Connected to remote server/host tn3270.de.ibm.com using port 23

Connect to z/VM PT Web Browser Interface


Web Server Logon - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://9.152.82.90:81/03D25F80/E2B3/LOGON/768
Search Print
Restore Down
Home Bookmarks mozilla.org Latest Builds
Performance Toolkit for VM FL 440
Done

Remote Performance Monitoring Session Setup

Web Server Logon

You are connected to the data retrieval interface of the Performance Toolkit for VM on system **BOEVMG73**. Data retrieval authorization is based on your VM user identification on that system. Please enter your userid and password (RACF)

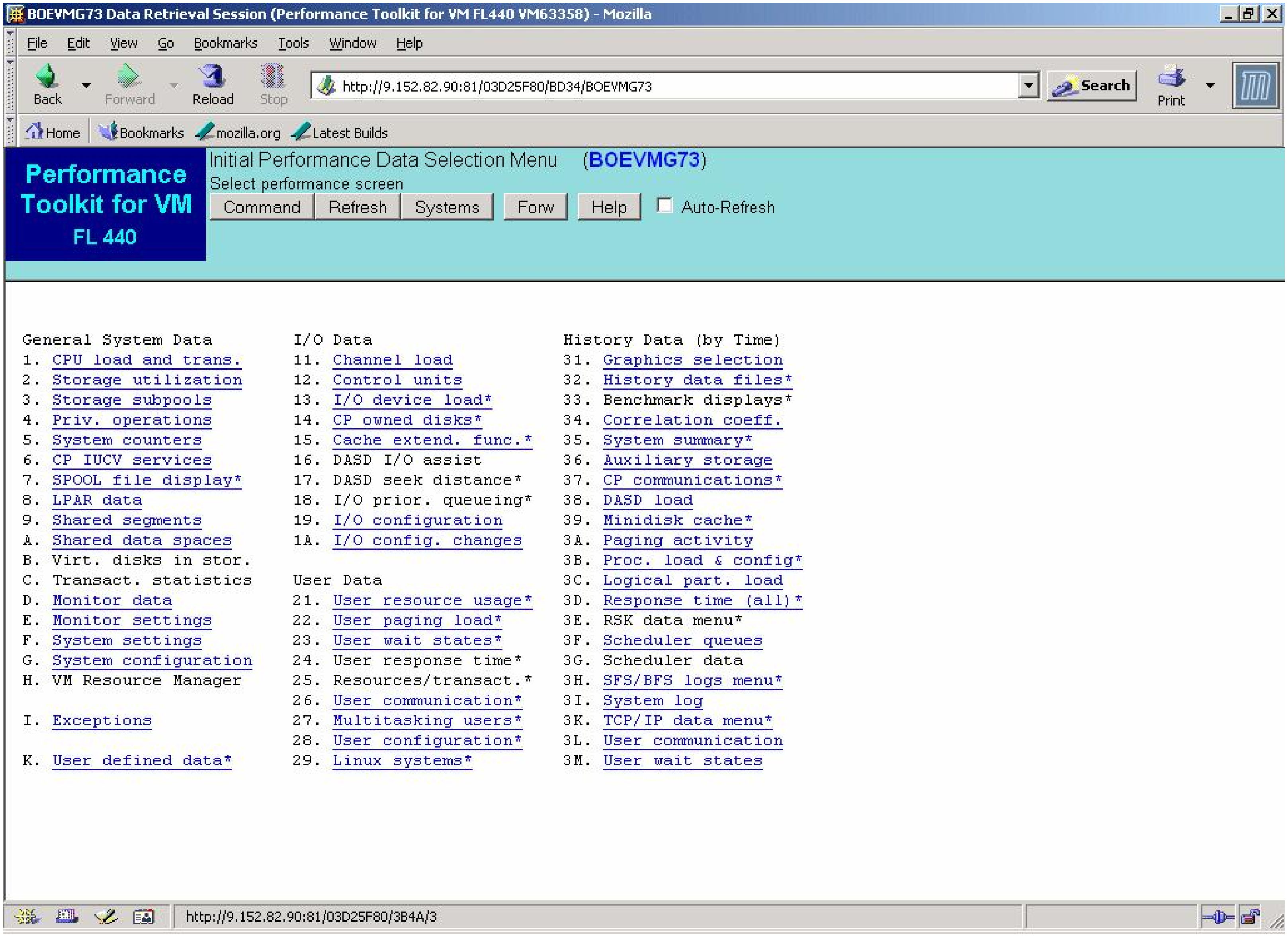
VM UserID: benke Password : ******* Submit

Desired screen layout:

Max. Data Lines: 24 Line length: 132

Up to 12 kB of data can be retrieved per selection, including all control information. Output may be truncated if space is not sufficient for all lines.

z/VM PT Web Browser Main Menu



BOEVMG73 Data Retrieval Session (Performance Toolkit for VM FL440 VM63358) - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back
Forward
Reload
Stop
http://9.152.82.90:81/03D25F80/BD34/BOEVMG73
Search
Print
Home
Bookmarks
mozilla.org
Latest Builds
Performance Toolkit for VM FL 440
Command
Refresh
Systems
Forw
Help
http://9.152.82.90:81/03D25F80/3B4A/3

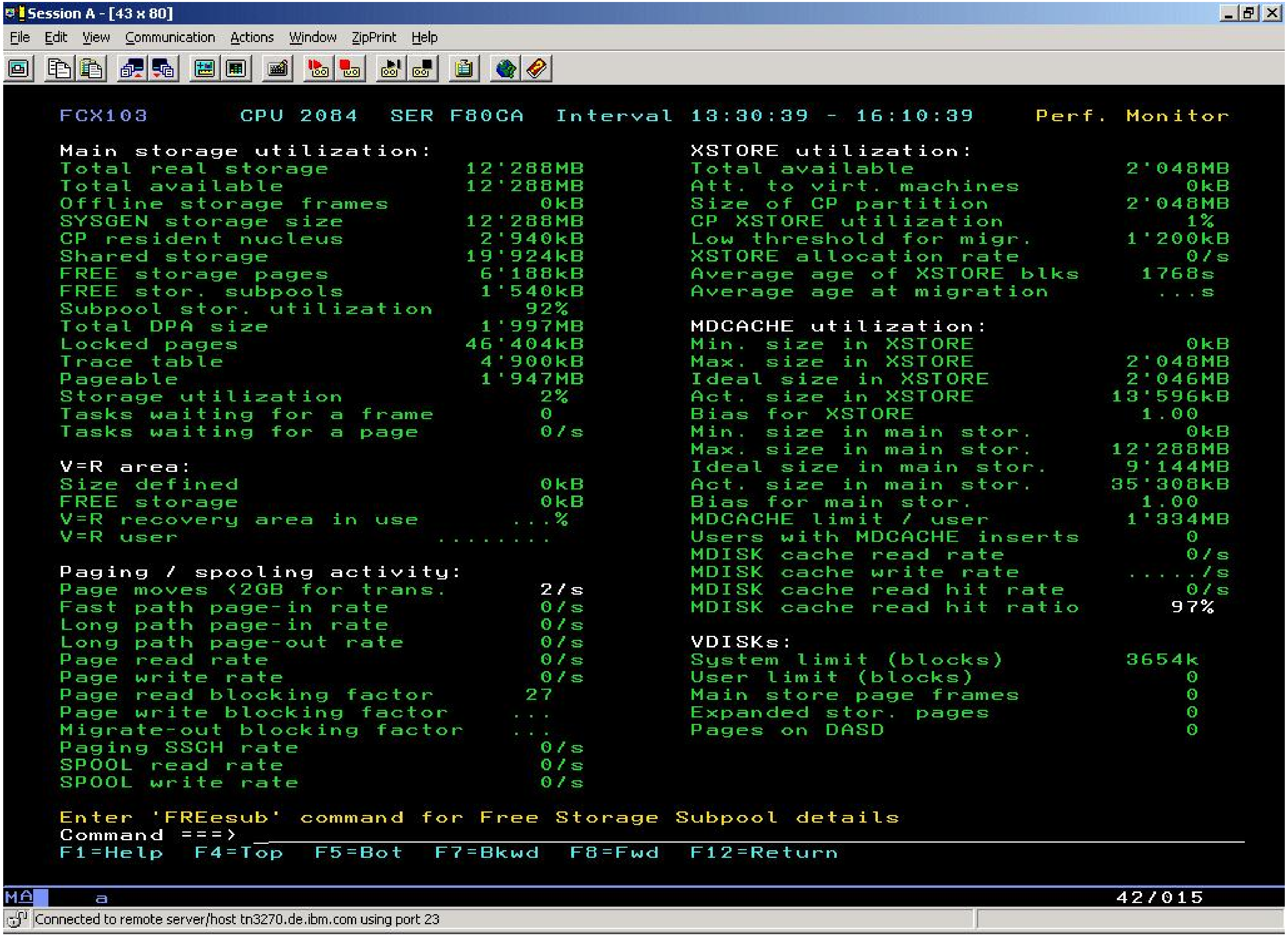
Initial Performance Data Selection Menu (BOEVMG73)

Select performance screen

□ Auto-Refresh

```
General System Data          I/O Data                  History Data (by Time)
1. CPU load and trans.       11. Channel load          31. Graphics selection
2. Storage utilization       12. Control units         32. History data files*
3. Storage subpools          13. I/O device load*      33. Benchmark displays*
4. Priv. operations          14. CP owned disks*       34. Correlation coeff.
5. System counters           15. Cache extend. func.*  35. System summary*
6. CP IUCV services          16. DASD I/O assist       36. Auxiliary storage
7. SPOOL file display*       17. DASD seek distance*   37. CP communications*
8. LPAR data                 18. I/O prior. queueing*  38. DASD load
9. Shared segments           19. I/O configuration     39. Minidisk cache*
A. Shared data spaces        1A. I/O config. changes   3A. Paging activity
B. Virt. disks in stor.                                3B. Proc. load & config*
C. Transact. statistics      User Data                 3C. Logical part. load
D. Monitor data              21. User resource usage*  3D. Response time (all)*
E. Monitor settings          22. User paging load*     3E. RSK data menu*
F. System settings           23. User wait states*     3F. Scheduler queues
G. System configuration      24. User response time*   3G. Scheduler data
H. VM Resource Manager       25. Resources/transact.*  3H. SFS/BFS logs menu*
                             26. User communication*   3I. System log
I. Exceptions                27. Multitasking users*   3K. TCP/IP data menu*
                             28. User configuration*   3L. User communication
K. User defined data*        29. Linux systems*        3M. User wait states
```

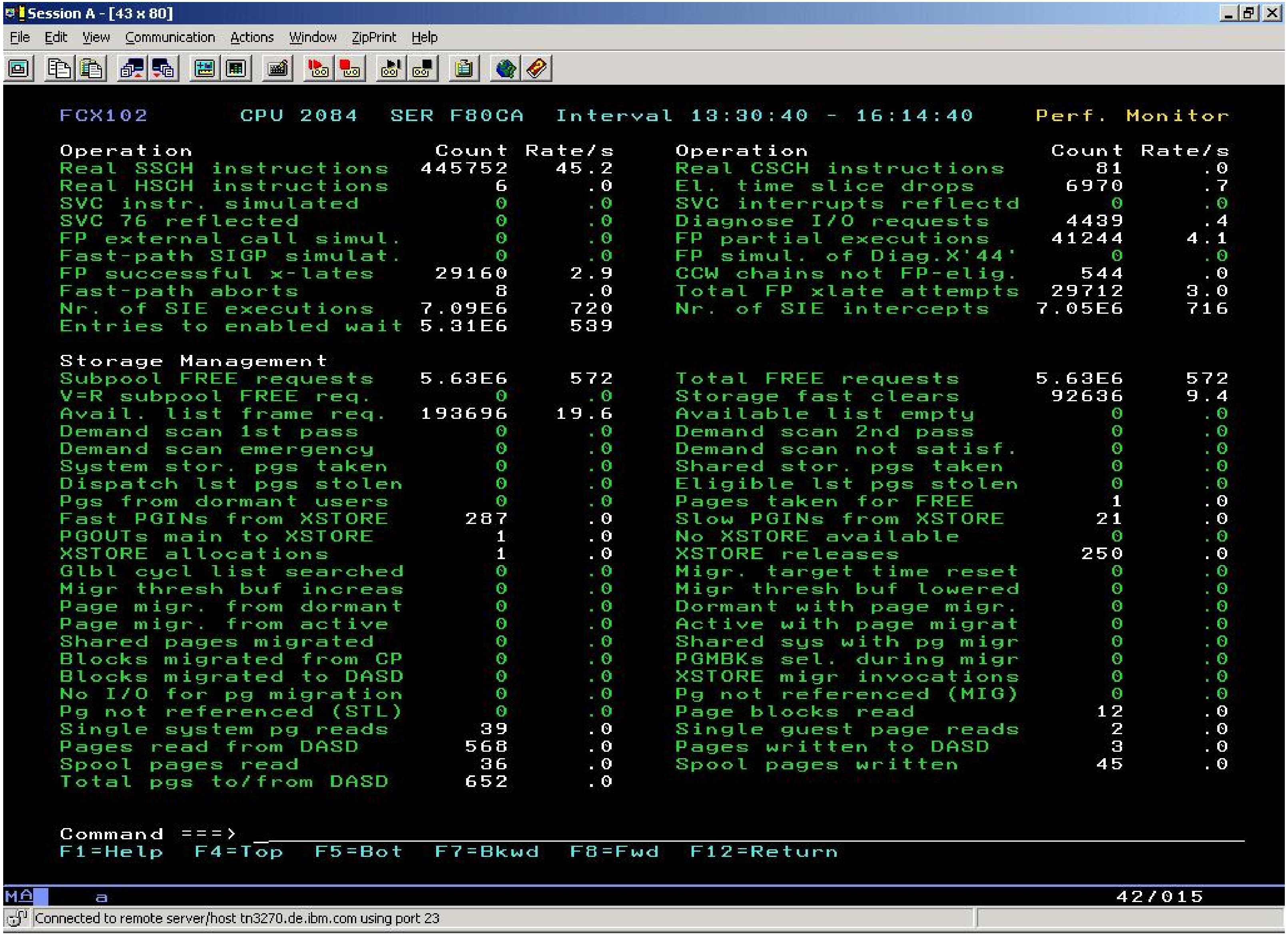
z/VM PT: Storage Utilization



```
FCX103      CPU 2084  SER F80CA  Interval 13:30:39 - 16:10:39    Perf. Monitor

Main storage utilization:                 XSTORE utilization:
Total real storage         12'288MB       Total available              2'048MB
Total available            12'288MB       Att. to virt. machines           0kB
Offline storage frames          0kB       Size of CP partition         2'048MB
SYSGEN storage size        12'288MB       CP XSTORE utilization            1%
CP resident nucleus         2'940kB       Low threshold for migr.      1'200kB
Shared storage             19'924kB       XSTORE allocation rate           0/s
FREE storage pages          6'188kB       Average age of XSTORE blks    1768s
FREE stor. subpools         1'540kB       Average age at migration        ...s
Subpool stor. utilization      92%
Total DPA size              1'997MB       MDCACHE utilization:
Locked pages               46'404kB       Min. size in XSTORE              0kB
Trace table                 4'900kB       Max. size in XSTORE          2'048MB
Pageable                    1'947MB       Ideal size in XSTORE         2'046MB
Storage utilization            2%         Act. size in XSTORE         13'596kB
Tasks waiting for a frame      0          Bias for XSTORE               1.00
Tasks waiting for a page        0/s       Min. size in main stor.          0kB
                                          Max. size in main stor.     12'288MB
V=R area:                                 Ideal size in main stor.     9'144MB
Size defined                    0kB       Act. size in main stor.     35'308kB
FREE storage                    0kB       Bias for main stor.           1.00
V=R recovery area in use      ...%        MDCACHE limit / user         1'334MB
V=R user                 ........         Users with MDCACHE inserts       0
                                          MDISK cache read rate            0/s
Paging / spooling activity:               MDISK cache write rate       ...../s
Page moves <2GB for trans.      2/s       MDISK cache read hit rate        0/s
Fast path page-in rate          0/s       MDISK cache read hit ratio      97%
Long path page-in rate          0/s
Long path page-out rate         0/s       VDISKs:
Page read rate                  0/s       System limit (blocks)         3654k
Page write rate                 0/s       User limit (blocks)              0
Page read blocking factor     27          Main store page frames           0
Page write blocking factor    ...         Expanded stor. pages             0
Migrate-out blocking factor   ...         Pages on DASD                    0
Paging SSCH rate                0/s
SPOOL read rate                 0/s
SPOOL write rate                0/s

Enter 'FREesub' command for Free Storage Subpool details
Command ===> _
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return

MA      a                                                                42/015
```

Connected to remote server/host tn3270.de.ibm.com using port 23

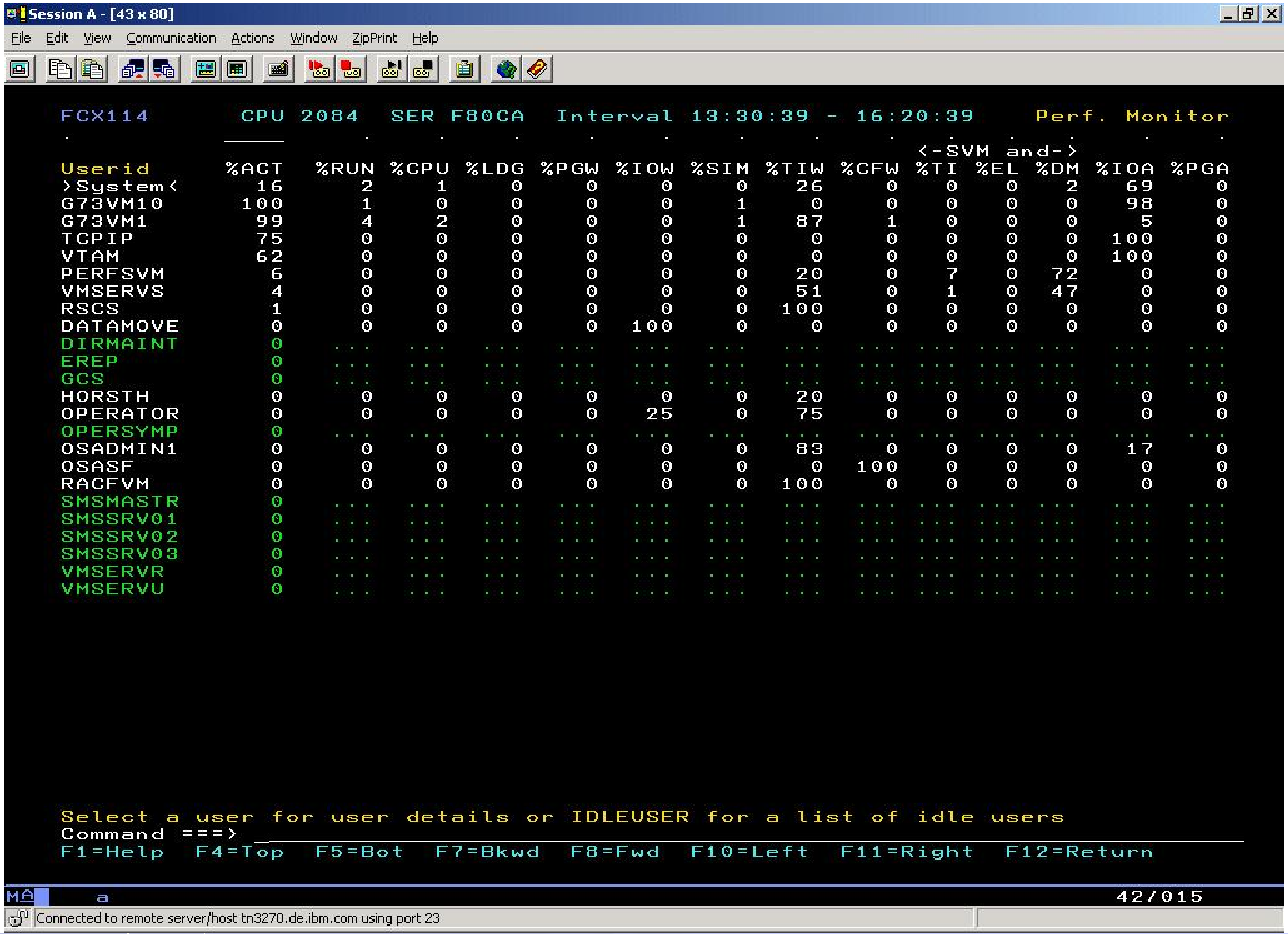
z/VM PT: System Counters



```
FCX102      CPU 2084   SER F80CA   Interval 13:30:40 - 16:14:40    Perf. Monitor

Operation                Count Rate/s    Operation                Count  Rate/s
Real SSCH instructions  445752   45.2    Real CSCH instructions      81      .0
Real HSCH instructions       6     .0    El. time slice drops      6970      .7
SVC instr. simulated         0     .0    SVC interrupts reflectd      0      .0
SVC 76 reflected             0     .0    Diagnose I/O requests     4439      .4
FP external call simul.      0     .0    FP partial executions    41244     4.1
Fast-path SIGP simulat.      0     .0    FP simul. of Diag.X'44'      0      .0
FP successful x-lates    29160    2.9    CCW chains not FP-elig.    544      .0
Fast-path aborts             8     .0    Total FP xlate attempts  29712     3.0
Nr. of SIE executions   7.09E6    720    Nr. of SIE intercepts   7.05E6     716
Entries to enabled wait 5.31E6    539

Storage Management
Subpool FREE requests   5.63E6    572    Total FREE requests     5.63E6     572
V=R subpool FREE req.        0     .0    Storage fast clears      92636     9.4
Avail. list frame req.  193696   19.6    Available list empty         0      .0
Demand scan 1st pass         0     .0    Demand scan 2nd pass         0      .0
Demand scan emergency        0     .0    Demand scan not satisf.      0      .0
System stor. pgs taken       0     .0    Shared stor. pgs taken       0      .0
Dispatch lst pgs stolen      0     .0    Eligible lst pgs stolen      0      .0
Pgs from dormant users       0     .0    Pages taken for FREE         1      .0
Fast PGINs from XSTORE     287     .0    Slow PGINs from XSTORE      21      .0
PGOUTs main to XSTORE        1     .0    No XSTORE available          0      .0
XSTORE allocations           1     .0    XSTORE releases            250      .0
Glbl cycl list searched      0     .0    Migr. target time reset      0      .0
Migr thresh buf increas      0     .0    Migr thresh buf lowered      0      .0
Page migr. from dormant      0     .0    Dormant with page migr.      0      .0
Page migr. from active       0     .0    Active with page migrat      0      .0
Shared pages migrated        0     .0    Shared sys with pg migr      0      .0
Blocks migrated from CP      0     .0    PGMBKs sel. during migr      0      .0
Blocks migrated to DASD      0     .0    XSTORE migr invocations      0      .0
No I/O for pg migration      0     .0    Pg not referenced (MIG)      0      .0
Pg not referenced (STL)      0     .0    Page blocks read            12      .0
Single system pg reads      39     .0    Single guest page reads      2      .0
Pages read from DASD       568     .0    Pages written to DASD        3      .0
Spool pages read            36     .0    Spool pages written         45      .0
Total pgs to/from DASD     652     .0


Command ===> _
F1=Help  F4=Top   F5=Bot   F7=Bkwd   F8=Fwd   F12=Return

MA      a                                                          42/015
```

Connected to remote server/host tn3270.de.ibm.com using port 23

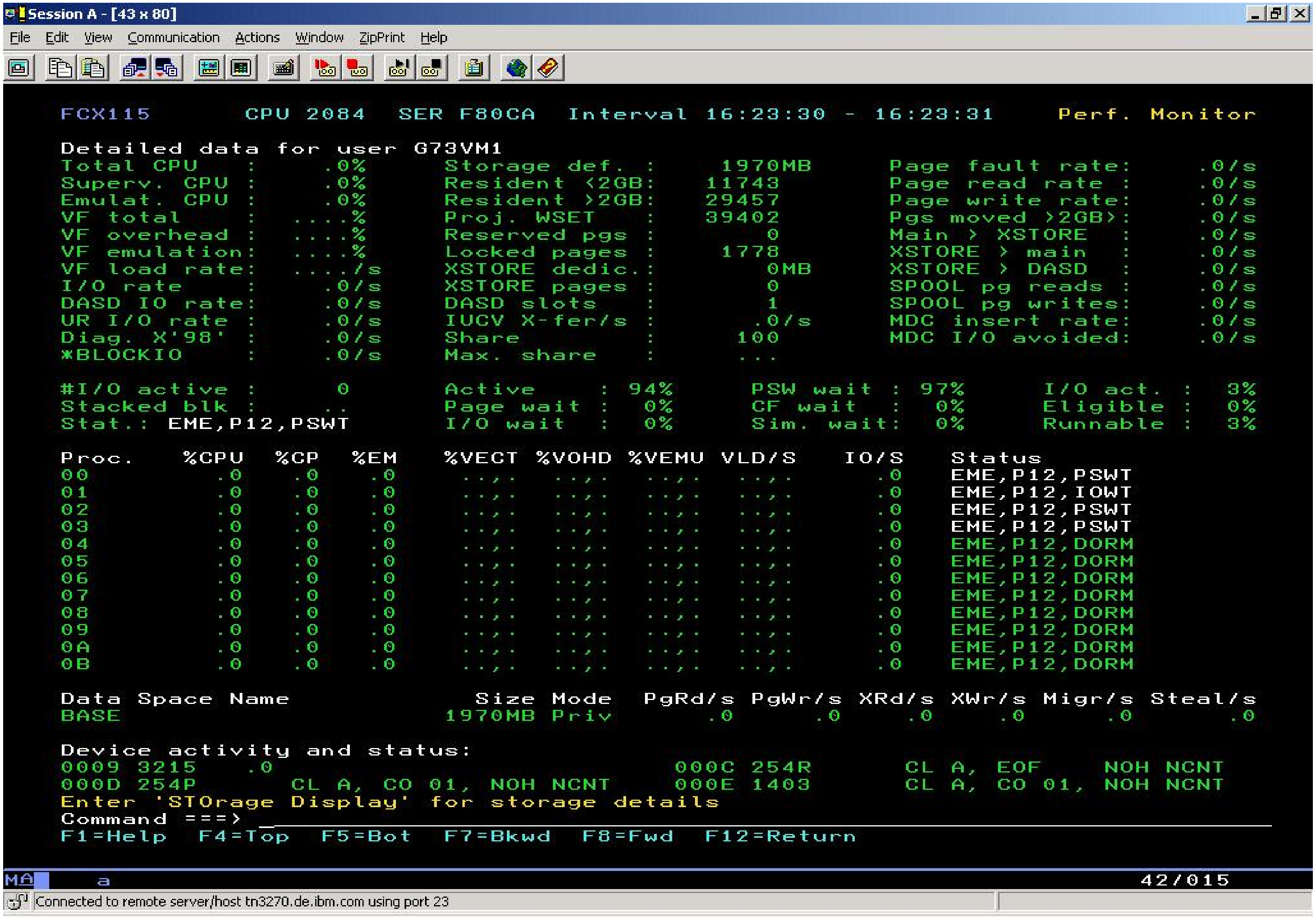
z/VM PT: %using and %delay – like states …



```
FCX114      CPU 2084  SER F80CA  Interval 13:30:39 - 16:20:39   Perf. Monitor
.           ____   .    .    .    .    .    .    .    .    .    .    .    .    .
                                                        <-SVM and->
Userid      %ACT  %RUN %CPU %LDG %PGW %IOW %SIM %TIW %CFW %TI  %EL  %DM %IOA %PGA
>System<      16     2    1    0    0    0    0   26    0    0    0    2   69    0
G73VM10      100     1    0    0    0    0    1    0    0    0    0    0   98    0
G73VM1        99     4    2    0    0    0    1   87    1    0    0    0    5    0
TCPIP         75     0    0    0    0    0    0    0    0    0    0    0  100    0
VTAM          62     0    0    0    0    0    0    0    0    0    0    0  100    0
PERFSVM        6     0    0    0    0    0    0   20    0    7    0   72    0    0
VMSERVS        4     0    0    0    0    0    0   51    0    1    0   47    0    0
RSCS           1     0    0    0    0    0    0  100    0    0    0    0    0    0
DATAMOVE       0     0    0    0    0  100    0    0    0    0    0    0    0    0
DIRMAINT       0   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
EREP           0   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
GCS            0   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
HORSTH         0     0    0    0    0    0    0   20    0    0    0    0    0    0
OPERATOR       0     0    0    0    0   25    0   75    0    0    0    0    0    0
OPERSYMP       0   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
OSADMIN1       0     0    0    0    0    0    0   83    0    0    0    0   17    0
OSASF          0     0    0    0    0    0    0    0  100    0    0    0    0    0
RACFVM         0     0    0    0    0    0    0  100    0    0    0    0    0    0
SMSMASTR       0   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
SMSSRV01       0   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
SMSSRV02       0   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
SMSSRV03       0   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
VMSERVR        0   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...
VMSERVU        0   ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...  ...




Select a user for user details or IDLEUSER for a list of idle users
Command ===> _____________________________________________________________________
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F10=Left  F11=Right  F12=Return
```

Connected to remote server/host tn3270.de.ibm.com using port 23

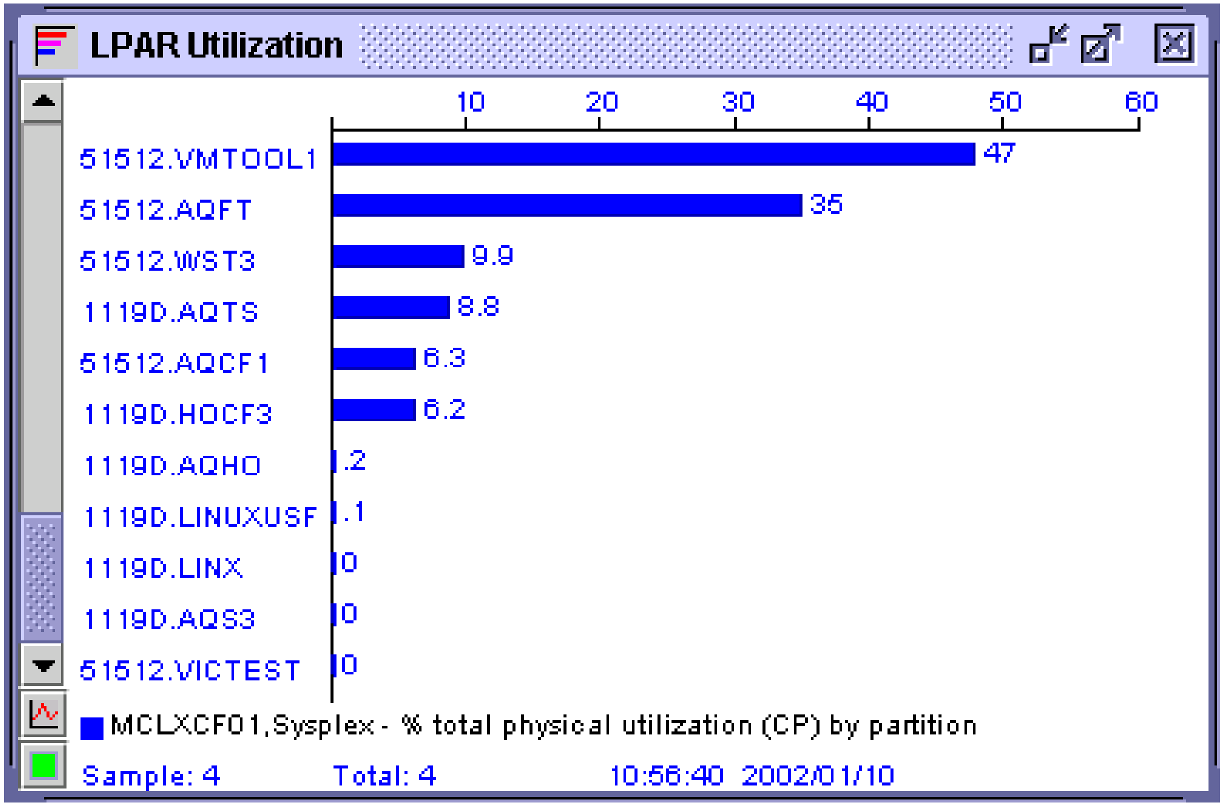
z/VM PT: User Details



```
FCX115      CPU 2084  SER F80CA  Interval 16:23:30 - 16:23:31   Perf. Monitor

Detailed data for user G73VM1
Total CPU   :     .0%     Storage def. :    1970MB    Page fault rate:    .0/s
Superv. CPU :     .0%     Resident <2GB:   11743      Page read rate :    .0/s
Emulat. CPU :     .0%     Resident >2GB:   29457      Page write rate:    .0/s
VF total    :   ....%     Proj. WSET   :   39402      Pgs moved >2GB>:    .0/s
VF overhead :   ....%     Reserved pgs :       0      Main > XSTORE  :    .0/s
VF emulation:   ....%     Locked pages :    1778      XSTORE > main  :    .0/s
VF load rate:   ..../s    XSTORE dedic.:       0MB    XSTORE > DASD  :    .0/s
I/O rate    :     .0/s    XSTORE pages :       0      SPOOL pg reads :    .0/s
DASD IO rate:     .0/s    DASD slots   :       1      SPOOL pg writes:    .0/s
UR I/O rate :     .0/s    IUCV X-fer/s :      .0/s    MDC insert rate:    .0/s
Diag. X'98' :     .0/s    Share        :     100      MDC I/O avoided:    .0/s
*BLOCKIO    :     .0/s    Max. share   :     ...

#I/O active :      0      Active     : 94%     PSW wait : 97%     I/O act.  :  3%
Stacked blk :     ..      Page wait  :  0%     CF wait  :  0%     Eligible  :  0%
Stat.: EME,P12,PSWT       I/O wait   :  0%     Sim. wait:  0%     Runnable  :  3%

Proc.   %CPU  %CP  %EM   %VECT %VOHD %VEMU VLD/S   IO/S   Status
00        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,PSWT
01        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,IOWT
02        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,PSWT
03        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,PSWT
04        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,DORM
05        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,DORM
06        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,DORM
07        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,DORM
08        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,DORM
09        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,DORM
0A        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,DORM
0B        .0   .0   .0    ..,.  ..,.  ..,.  ..,.     .0   EME,P12,DORM

Data Space Name            Size Mode  PgRd/s PgWr/s XRd/s XWr/s Migr/s Steal/s
BASE                     1970MB Priv      .0     .0    .0    .0     .0      .0

Device activity and status:
0009 3215   .0                          000C 254R      CL A, EOF    NOH NCNT
000D 254P      CL A, CO 01, NOH NCNT    000E 1403      CL A, CO 01, NOH NCNT
Enter 'STOrage Display' for storage details
Command ===> _
F1=Help  F4=Top  F5=Bot  F7=Bkwd  F8=Fwd  F12=Return

MA     a                                                                 42/015
```

Connected to remote server/host tn3270.de.ibm.com using port 23

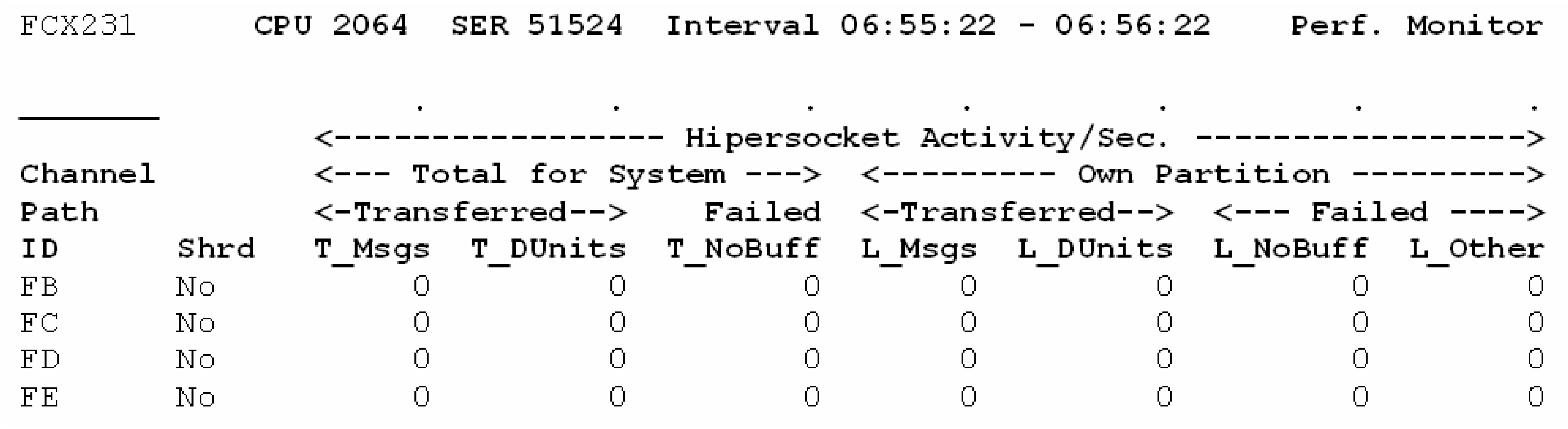
LPAR partition data from z/OS RMF



■ MCLXCF01,Sysplex - % total physical utilization (CP) by partition

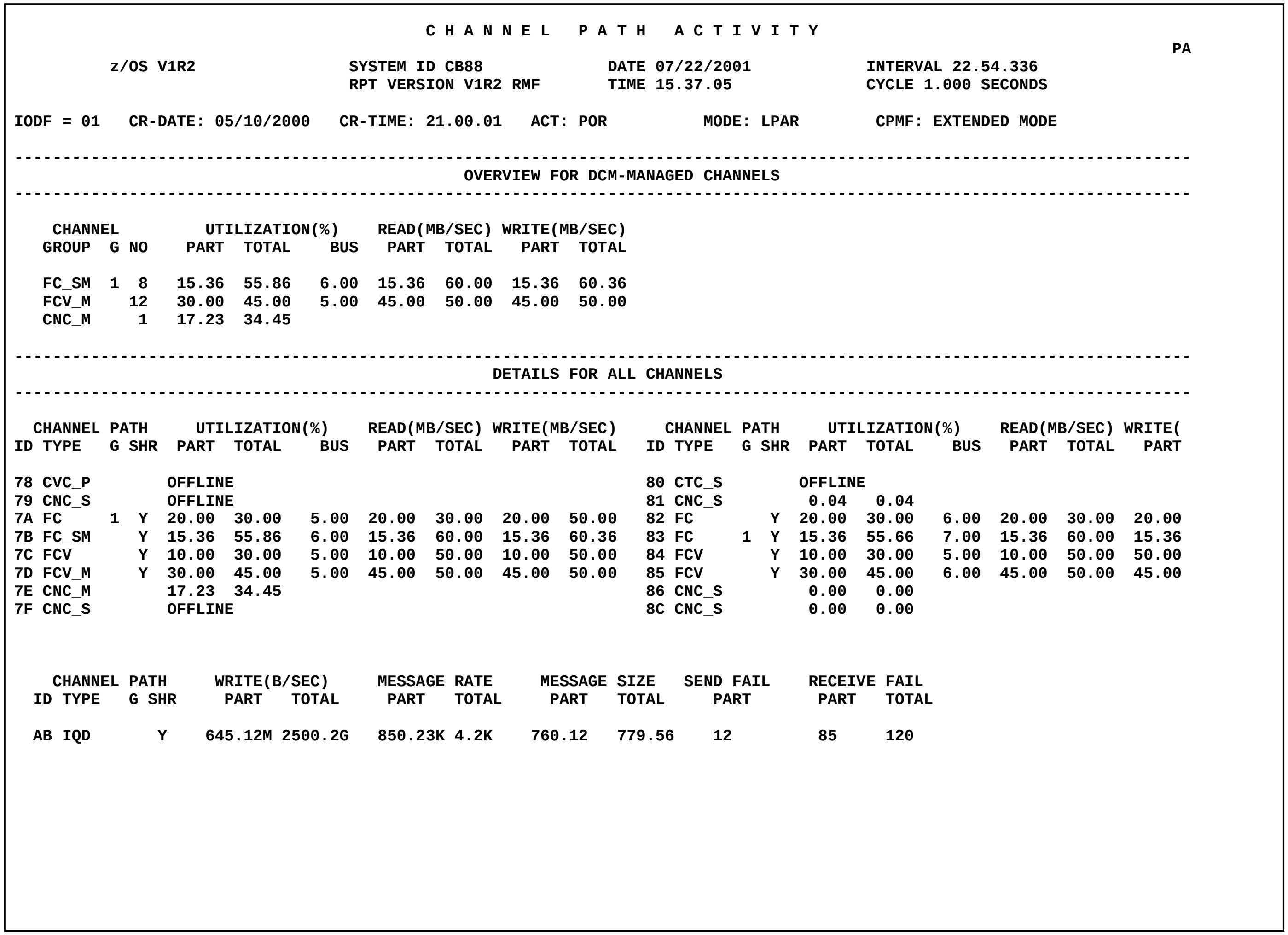
Sample: 4 Total: 4 10:56:40 2002/01/10

HiperSockets display in z/VM FCON



```
FCX231      CPU 2064  SER 51524  Interval 06:55:22 - 06:56:22    Perf. Monitor

_______             .         .         .       .         .         .        .
               <----------------- Hipersocket Activity/Sec. ----------------->
Channel        <--- Total for System --->  <--------- Own Partition --------->
Path           <-Transferred-->    Failed  <-Transferred-->  <--- Failed ---->
ID      Shrd   T_Msgs  T_DUnits  T_NoBuff  L_Msgs  L_DUnits  L_NoBuff  L_Other
FB      No          0         0         0       0         0         0        0
FC      No          0         0         0       0         0         0        0
FD      No          0         0         0       0         0         0        0
FE      No          0         0         0       0         0         0        0
```

… and in z/OS RMF



```
                                           C H A N N E L   P A T H   A C T I V I T Y
                                                                                                                         PA
          z/OS V1R2                SYSTEM ID CB88             DATE 07/22/2001            INTERVAL 22.54.336
                                   RPT VERSION V1R2 RMF       TIME 15.37.05              CYCLE 1.000 SECONDS

IODF = 01   CR-DATE: 05/10/2000   CR-TIME: 21.00.01   ACT: POR          MODE: LPAR        CPMF: EXTENDED MODE

---------------------------------------------------------------------------------------------------------------------------
                                               OVERVIEW FOR DCM-MANAGED CHANNELS
---------------------------------------------------------------------------------------------------------------------------

    CHANNEL         UTILIZATION(%)    READ(MB/SEC) WRITE(MB/SEC)
   GROUP  G NO    PART  TOTAL    BUS   PART  TOTAL   PART  TOTAL

   FC_SM  1  8   15.36  55.86   6.00  15.36  60.00  15.36  60.36
   FCV_M    12   30.00  45.00   5.00  45.00  50.00  45.00  50.00
   CNC_M     1   17.23  34.45

---------------------------------------------------------------------------------------------------------------------------
                                                  DETAILS FOR ALL CHANNELS
---------------------------------------------------------------------------------------------------------------------------

  CHANNEL PATH     UTILIZATION(%)    READ(MB/SEC) WRITE(MB/SEC)     CHANNEL PATH     UTILIZATION(%)    READ(MB/SEC) WRITE(
ID TYPE   G SHR  PART  TOTAL    BUS   PART  TOTAL   PART  TOTAL   ID TYPE   G SHR  PART  TOTAL    BUS   PART  TOTAL   PART

78 CVC_P        OFFLINE                                           80 CTC_S        OFFLINE
79 CNC_S        OFFLINE                                           81 CNC_S         0.04   0.04
7A FC     1  Y  20.00  30.00   5.00  20.00  30.00  20.00  50.00   82 FC        Y  20.00  30.00   6.00  20.00  30.00  20.00
7B FC_SM     Y  15.36  55.86   6.00  15.36  60.00  15.36  60.36   83 FC     1  Y  15.36  55.66   7.00  15.36  60.00  15.36
7C FCV       Y  10.00  30.00   5.00  10.00  50.00  10.00  50.00   84 FCV       Y  10.00  30.00   5.00  10.00  50.00  50.00
7D FCV_M     Y  30.00  45.00   5.00  45.00  50.00  45.00  50.00   85 FCV       Y  30.00  45.00   6.00  45.00  50.00  45.00
7E CNC_M        17.23  34.45                                      86 CNC_S         0.00   0.00
7F CNC_S        OFFLINE                                           8C CNC_S         0.00   0.00


    CHANNEL PATH     WRITE(B/SEC)     MESSAGE RATE     MESSAGE SIZE   SEND FAIL    RECEIVE FAIL
  ID TYPE   G SHR     PART   TOTAL     PART   TOTAL     PART   TOTAL     PART       PART   TOTAL

  AB IQD       Y    645.12M 2500.2G   850.23K 4.2K    760.12   779.56    12         85     120
```

CP IND interface in Linux

- Interface between Linux kernel and z/VM CP
- CP device driver, developed by Neale Ferguson; interface between Linux and z/VM
- http://penguinvm.princeton.edu/programs (cpint.tar.gz)
- "#cp ind user" in Linux console:

```
CP IND
AVGPROC-069% 07
XSTORE-000037/SEC MIGRATE-0000/SEC
MDC READS-000001/SEC WRITES-000000/SEC HIT RATIO-094%
STORAGE-024% PAGING-0000/SEC STEAL-000%
Q0-00071 Q1-00000          Q2-00000 EXPAN-001 Q3-00000 EXPAN-001
```

  - … giving information like the 7 logical CPUs were utilized to 69%

Example scenario if not using “% Stolen” metric

- The following Linux image may be completely idle:

```
$ > top 12:30pm
up 4 min,  2 users,  load average: 0.02, 0.07, 0.03
24 processes: 23 sleeping, 1 running, 0 zombie, 0 stopped
CPU0 states:  0.1% user,  19.1% system,  0.0% nice, 80.8% idle
CPU1 states:  0.0% user,  23.2% system,  0.0% nice, 76.8% idle
```

- ... as z/VM is heavily loaded and does not give Linux many resources, so even for simple tasks, Linux needs about 20% of its CPU resources just to do almost nothing:

```
$ > #CP IND

AVGPROC-099% 07
```

z/VM MONWRITE

- **You can extract z/VM monitor records without any z/VM performance monitor; details are described on**
  - http://www.vm.ibm.com/perf/tips/collect.html

The NET-SNMP Project

- **SNMP (*Simple Network Management Protocol*) is a standard for performance data interchange. It is especially strong in TCP/IP network management. It is standardized by the IETF (Internet Engineering Task Force).**
- **SNMP has a simple Manager-Agent architecture. Standard protocol used is UDP (connectionless, delivery not guaranteed)**
- **Simple hierarchical data model**
- **Some security concerns for versions before v3**
- **NET-SNMP provides a free SNMP implementation, also usable for Linux for zSeries. The OSA adapter provides some performance information using SNMP.**

See *http://net-snmp.sourceforge.net/*

What is CIM ?

- CIM is a systems management standard provided by the DMTF (Distributed Management Task Force), a sub group of The Open Group. It is the dominant standard in SAN management, but also applicable to all other areas of systems management. It provides bridges to SNMP, e.g. for TCP/IP network management.

CIM Overview

- **One of the strength of CIM is the rich conceptual data model with about 1000 classes for major resources needed in the management of heterogeneous, distributed servers**
- **OpenPegasus, "C++ CIM/WBEM Manageability Services Broker", is the DMTF reference implementation of a CIMOM. It is published under the liberal MIT license in open source. See**
  http://www.openpegasus.org/

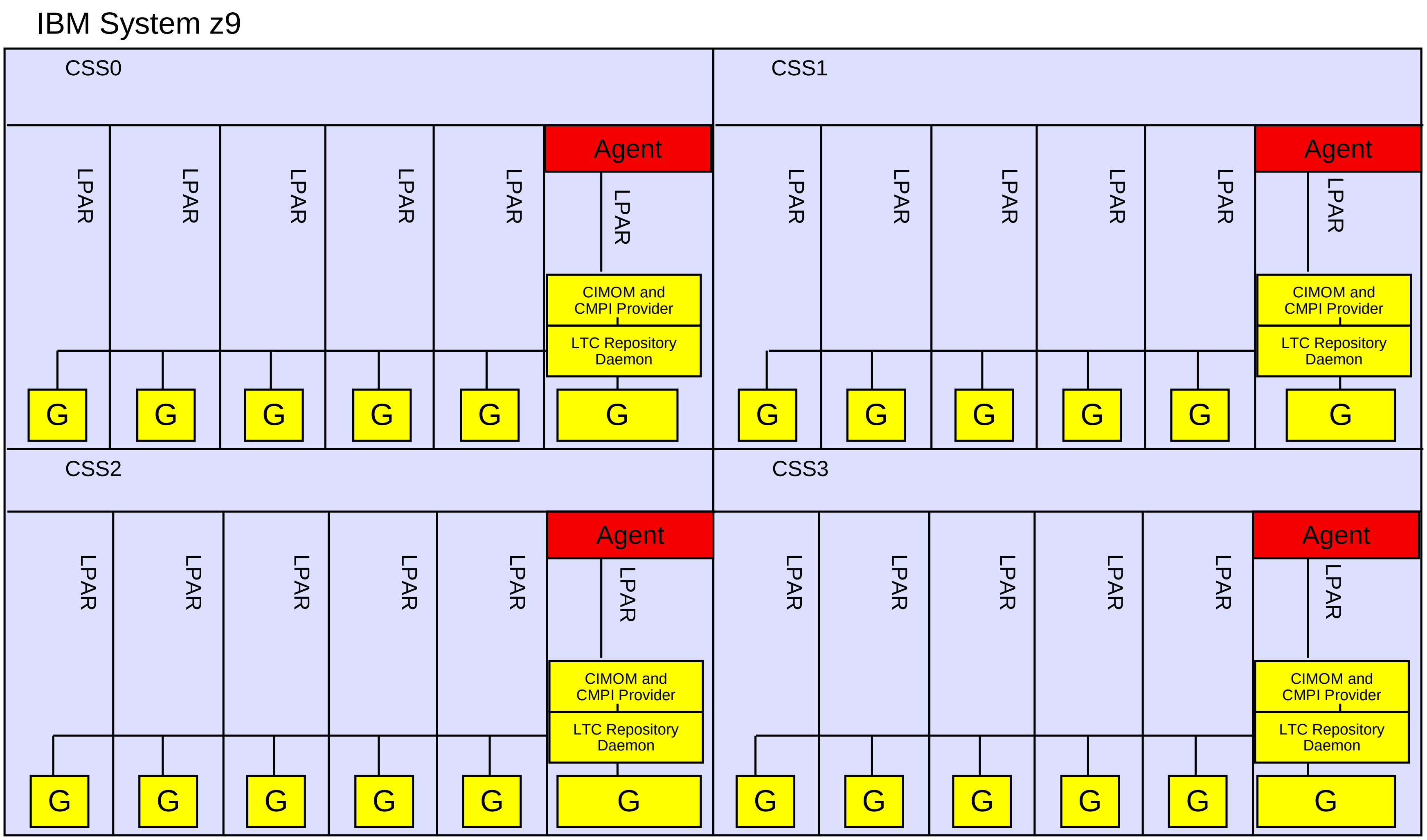
New System z specific metrics (SBLIM): CPU

- **LPAR data**
  - –Dispatch time
  - –LPAR management overhead time
  - –Number of processors
  - –… all directly from the hypervisor, so extremly precise, same data which is presented by z/OS RMF or z/VM PT

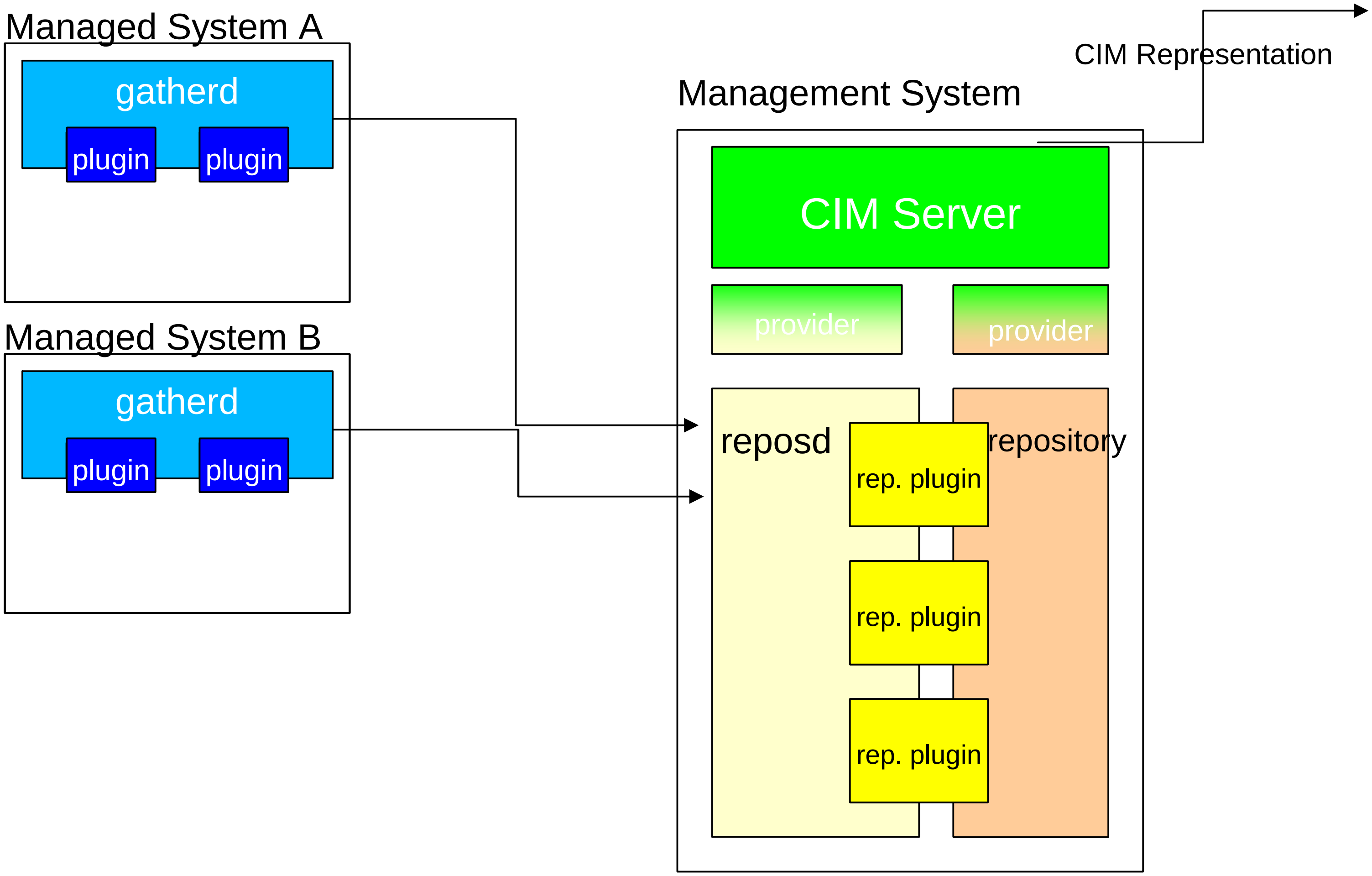
New System z specific metrics (SBLIM): IO

- **… all in RMF spirit from the semantics:**
- **Channel metrics**
  - Partition and CEC total utilization percentages, bandwidths for read/write transfer
- **FICON device metrics**
  - Connect, Disconnect, Pending times
  - Request rate, I/O intensity / I/O queue depths
  - Response time
  - Control Unit Queue time
  - Initial Command Response time

Possible Architecture: z9 box view



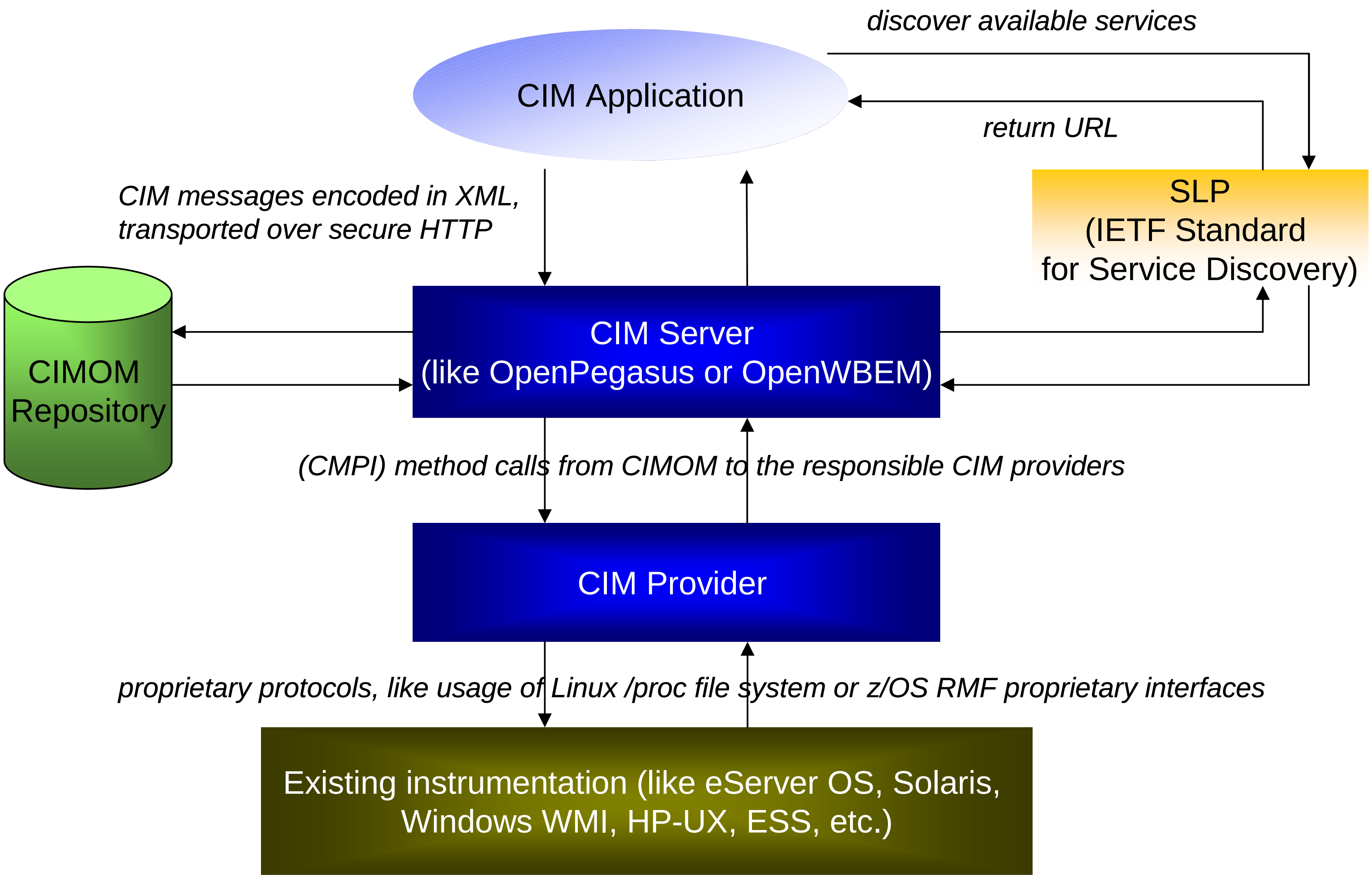
SBLIM Gatherer Topology for Distributed Systems



Platform independent

- **Smiliar infrastructure and metrics are currently also available for**
  - z/OS V1.7 and later
  - i5/OS
  - Xen

WBEM/CIM Architecture Overview



discover available services

CIM Application

return URL

SLP
(IETF Standard
for Service Discovery)

*CIM messages encoded in XML,
transported over secure HTTP*

CIMOM
Repository

CIM Server
(like OpenPegasus or OpenWBEM)

(CMPI) method calls from CIMOM to the responsible CIM providers

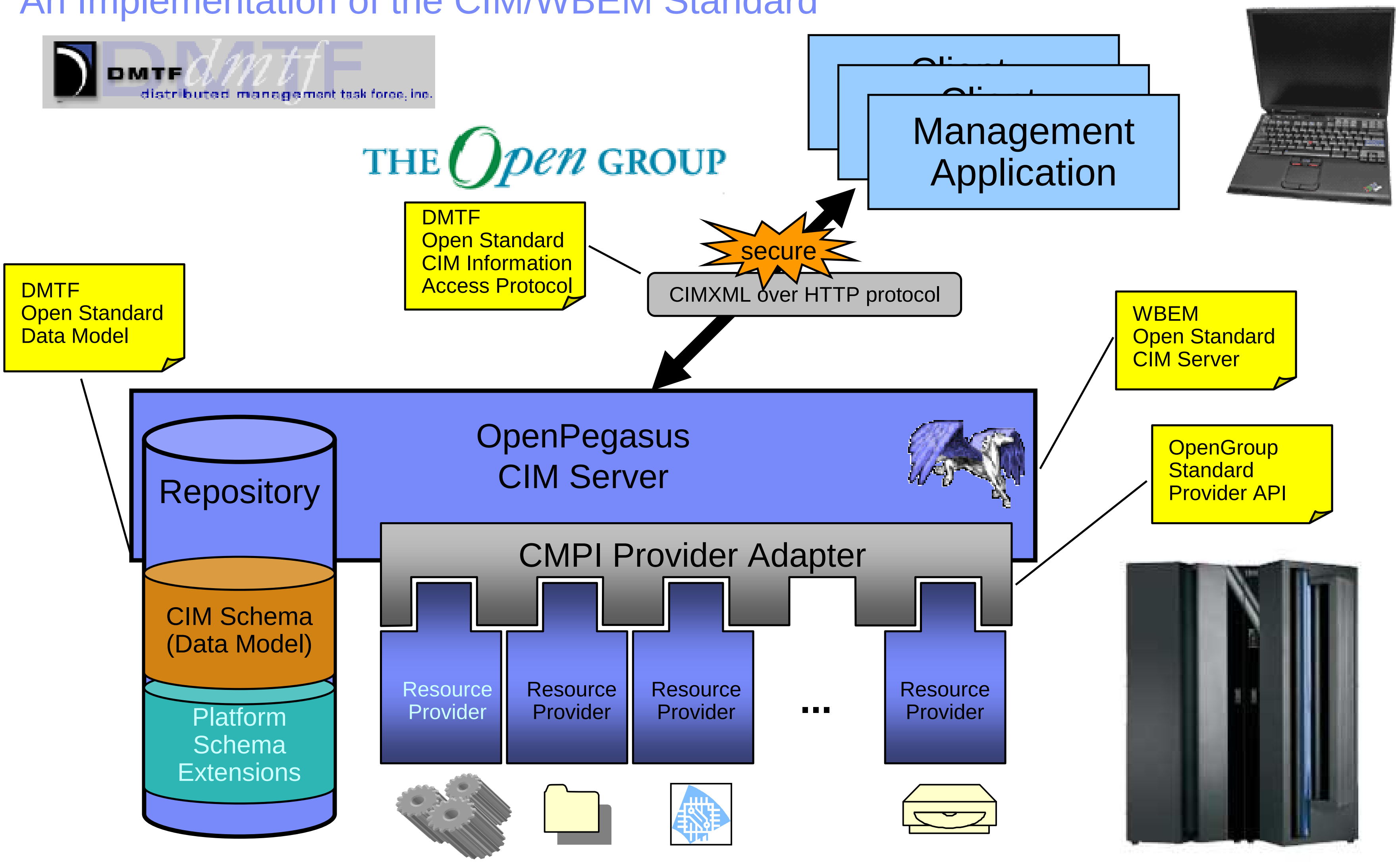
CIM Provider

proprietary protocols, like usage of Linux /proc file system or z/OS RMF proprietary interfaces

Existing instrumentation (like eServer OS, Solaris,
Windows WMI, HP-UX, ESS, etc.)

The OpenPegasus CIM Server

An Implementation of the CIM/WBEM Standard



DMTF distributed management task force, inc.

THE Open GROUP

Client

Client

Management Application

DMTF Open Standard CIM Information Access Protocol

secure

CIMXML over HTTP protocol

DMTF Open Standard Data Model

WBEM Open Standard CIM Server

OpenGroup Standard Provider API

OpenPegasus CIM Server

Repository

CIM Schema (Data Model)

Platform Schema Extensions

CMPI Provider Adapter

Resource Provider

Resource Provider

Resource Provider

...

Resource Provider

CIM/WBEM-based eServer OS management instrumentation

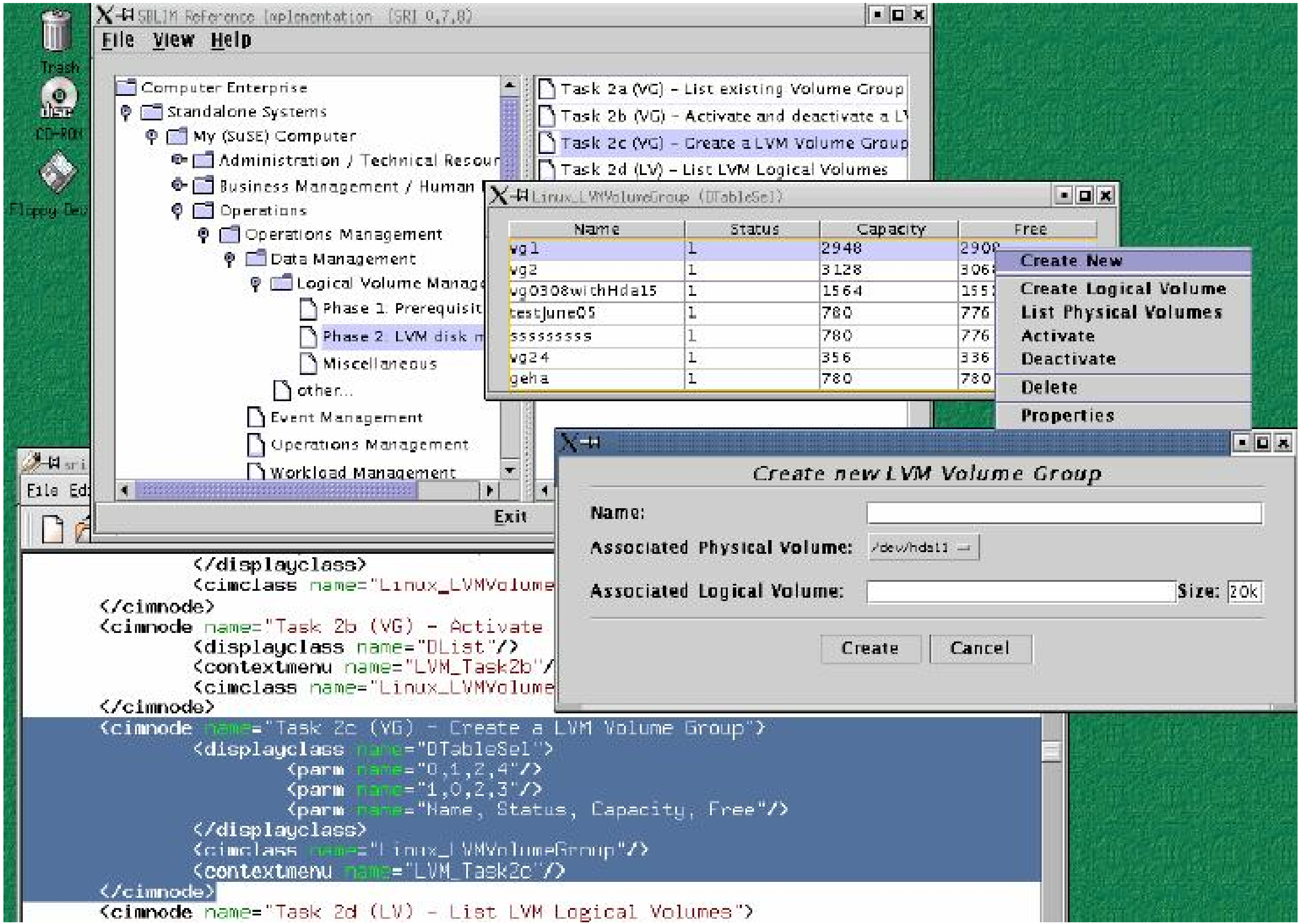
- **Common eServer model**
- **Open Standards**
- **Involved standardization bodies: The OpenGroup, DMTF, SNIA, etc.**

- **IBM TotalStorage CIM Agent for ESS:** http://www-1.ibm.com/servers/storage/support/software/cimess/planning.html
- **eServer CIM:** http://publib.boulder.ibm.com/infocenter/eserver/v1r1/en_US/index.htm?info/icmain.htm
- **pSeries / AIX:** http://publib.boulder.ibm.com/infocenter/pseries/index.jsp?topic=/com.ibm.aix.doc/aixbman/cim/overview.htm

SBLIM

- The goal of *WBEM (Web-based Enterprise Management)* is to provide interoperable technology based on the CIM standard. This standard is also driven by the DMTF.
- SBLIM is an Open-Source WBEM instrumentation project; see http://sourceforge.net/projects/sblim or http://www.sblim.org
- CMPI (*Common Manageability Programming Interface*) instrumentation interface (standardized API with CIM compliant semantics and operations) to make provider independent from CIMOM technology

SBLIM Reference Implementation



Name	Status	Capacity	Free
vg1	1	2948	2908
vg2	1	3128	3068
vg0308withHda15	1	1564	155
testJune05	1	780	776
sssssssss	1	780	776
vg24	1	356	336
geha	1	780	780

Create new LVM Volume Group

Name:

Associated Physical Volume: /dev/hda11

Associated Logical Volume: Size: 20k

Create Cancel

```
        </displayclass>
        <cimclass name="Linux_LVMVolume
</cimnode>
<cimnode name="Task 2b (VG) - Activate
        <displayclass name="DList"/>
        <contextmenu name="LVM_Task2b"/
        <cimclass name="Linux_LVMVolume
</cimnode>
<cimnode name="Task 2c (VG) - Create a LVM Volume Group">
        <displayclass name="DTableSel">
                <parm name="0,1,2,4"/>
                <parm name="1,0,2,3"/>
                <parm name="Name, Status, Capacity, Free"/>
        </displayclass>
        <cimclass name="Linux_LVMVolumeGroup"/>
        <contextmenu name="LVM_Task2c"/>
</cimnode>
<cimnode name="Task 2d (LV) - List LVM Logical Volumes">
```

Resources

- **z/VM Performance Resources:**
  http://www.vm.ibm.com/perf/
- **z/VM Performance Toolkit:**
  http://www.vm.ibm.com/related/perfkit/
- **RMF Linux Data Gatherer:**
  http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/rmftools.htm#pmlin
- **SBLIM Project: (OpenPegasus CIMOM based)**
  http://sourceforge.net/projects/sblim/
- **Accounting and Monitoring for z/VM Linux guest machines**
  http://publib-b.boulder.ibm.com/Redbooks.nsf/RedpaperAbstracts/redp3818.html?Open
- **Tuning Hints and Tips**
  http://www10.software.ibm.com/developerworks/opensource/linux390/perf/index.shtml

References

- **"Linux on IBM eServer zSeries and S/390: Performance Toolkit for z/VM" Redbook, SG24-6059**

- **Redbook Paper "Accounting and monitoring for z/VM Linux guest machines" by Erich Amrehn et al**

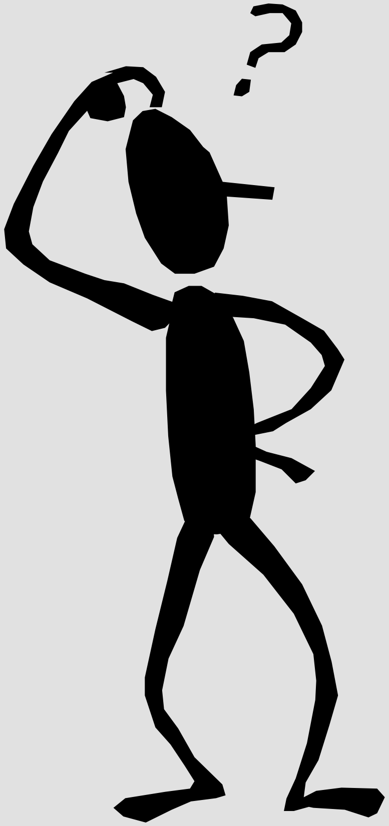
- **"Linux on IBM eServer zSeries and S/390: Performance Measurement and Tuning" Redbook, SG24-6926**

- **"Linux on zSeries and S/390: Systems Management Redbook, SG24-6820**

- **"Linux for IBM eServer zSeries and S/390: ISP/ASP Solutions" Redbook, SG24-6299**

- **Jason R Fink & Matthew D Sherer: "Linux Performance Tuning and Capacity Planning", SAMS 2001, ISBN 0-672-32081-9**

Questions?



Email:

benke@de.ibm.com