



Survey of HP-UX Commands and Tools for System and Performance Monitoring



Logan Sankaran Ph.D.
Sr. Solution Architect, Americas Solution Center
Hewlett-Packard

© 2004 Hewlett-Packard Development Company, L.P.
The information contained herein is subject to change without notice



HP-UX Tools and Commands - Overview

- Tools for System Monitoring
 - glance, vmstat, sar, tusc
- Tools for Application Performance
 - caliper, prospect, chatr, mpsched, psrset, mpctl
- Miscellaneous Commands
 - iostat, netstat, ipcs, ipcrm, swapinfo, sysdef, dmesg
- Tools and Commands NOT covered here:
 - openview, prm, wlm, gdb, totalview

Glance – Powerful and Popular Tool



Tool: glance

Where: /opt/perf/bin/glance

Description:

GlancePlus is a powerful system monitoring tool

Features:

- on-line and easy-to-use
- available in character and graphical modes
- rich set of collectable metrics
- very low resource requirements
- runs on any terminal/workstation with serial interface and slow data communication links

Glance in character Mode – Examples of Usage ...



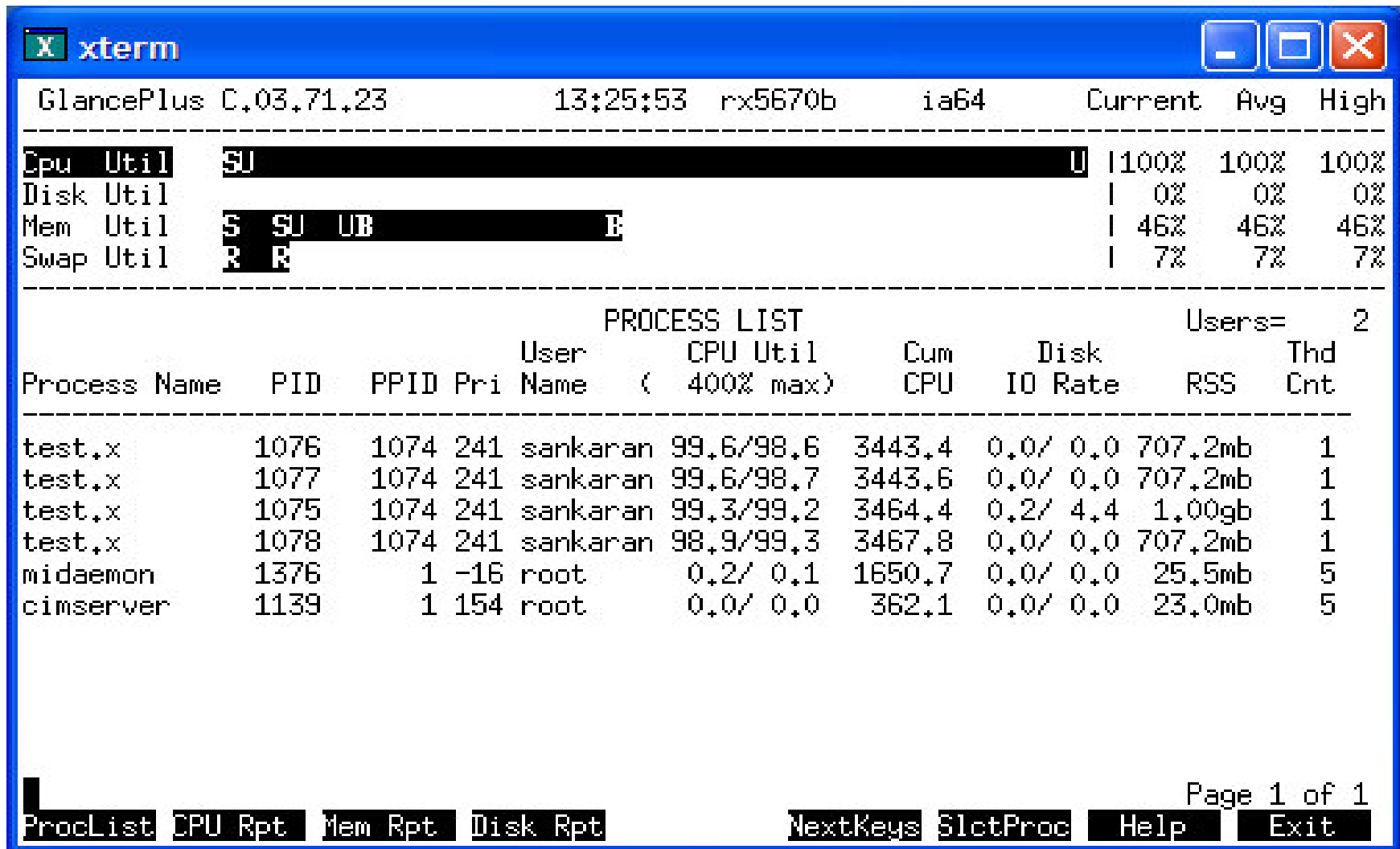
- glance : starting with default values
- glance -j 60 : screen refreshes every 60 sec.
- glance -p lp2 : sends screen output to printer lp2
- glance -j 7200 -p lp1 -w : prints the swap utilization every two hours on the printer lp1
- glance -nice -19 -maxpages 10 -nosort : execute glance at a higher priority, limits the max. pages by print command to 10 and do not sort processes
- glance -iterations 1000 : runs glance only 1000 times at the default refresh rate (5 sec).

Glance in character Mode – Examples of Usage



- glance –adviser_only : start glance in the background with no screen output
- glance –adviser_only –syntax ./temp –j 60 > glance.out &
- where ./temp contains
- ‘print gbl_stattime, “runq=”, gbl_run_queue’
- this approach can be incorporated into scripts to monitor and collect metrics while your applications are running.

Glance in character Mode – Main Screen



Glance in character Mode – Command Summary ...



```
X Terminal on snick.rsn.hp.com

GlancePlus C.03.71.23      13:51:32  rx5670b   ia64      Current  Avg  High
-----
CPU Util    SU                                     U 100% 100% 100%
Disk Util   | 0% 1% 1
7 Util      S  SU  UB                                     | 46% 46% 46%
Swap Util   R  R                                     | 7% 7% 7%
-----

GlancePlus Commands Menu

h - Online Help          q - exit (or e)          A - Application List
g - Process List         d - Disk Report          P - PRM Group List
a - CPU By Processor     i - IO By File System    Y - Global System Calls
c - CPU Report           u - IO By Disk           F - Process Open Files
m - Memory Report        v - IO By Logical Volume M - Process Memory Regions
t - System Tables        N - NFS Global Activity  R - Process Resources
w - Swap Space           n - NFS By System        W - Process Wait States
B - Global Waits         l - Network By Interface L - Process System Calls
Z - Global Threads       T - Trans Tracker        y - Renice Process
G - Process Threads      H - Alarm History        s - Select Process
I - Thread Resource      J - Thread Wait
S - Select Disk/NFS/Appl/Trans/Thread

Enter command or function key:

ProcList CPU Rpt Mem Rpt Disk Rpt      NextKeys SlectProc Help Exit
```

Glance in character Mode – Command Summary



```
X Terminal on snick.rsn.hp.com

GlancePlus C.03.71.23      13:52:39  rx5670b  ia64      Current  Avg  High
-----
CPU  Util  SU [99% bar] U | 99% 100% 100%
Disk Util  F [1% bar]  | 1%  1%  7%
Mem  Util  S  SU  UB [46% bar] E | 46% 46% 46%
Swap Util  z  R [7% bar]  | 7%  7%  7%
-----

GlancePlus Control Key Menu

? - Commands Menu      b - Page Backward (or -)    < - Display Previous Screen
! - Invoke Shell       f - Page Forward (or +,space) > - Display Next Logical Scr
h - Online Help        q - exit (or e )          z - Reset Statistics to Zero
p - Print Toggle      r - Refresh Screen (or ^L)  <cr>- Update Current Screen
j - Adjust Interval  o - Threshold Screen Options

Enter command or function key:

ProcList CPU Rpt Mem Rpt Disk Rpt      NextKeys SlotProc Help Exit
```

Glance in character Mode – CPU Report



X Terminal on snick.rsn.hp.com

GlancePlus C.03.71.23 13:58:06 rx5670b ia64 Current Avg High

Resource	Util	Current	Avg	High
Cpu Util	SU	99%	100%	100%
Disk Util	F	1%	1%	9%
Mem Util	S SU UB	46%	46%	46%
Swap Util	R R	7%	7%	7%

CPU REPORT Users= 2

State	Current	Average	High	Time	Cum Time
User	98.1	98.9	100.0	5.15	756.11
Nice	0.0	0.0	0.0	0.00	0.00
Negative Nice	0.0	0.0	0.0	0.00	0.31
RealTime	0.0	0.0	0.0	0.00	0.16
System	0.2	0.3	5.0	0.01	2.41
Interrupt	0.6	0.5	0.7	0.03	3.98
ContextSwitch	0.0	0.1	0.2	0.00	0.84
Traps	0.0	0.0	0.0	0.00	0.32
Vfaults	0.0	0.0	0.0	0.00	0.04
Idle	0.8	0.0	1.0	0.04	0.05

Top CPU user: PID 1075, test.x 99.0% cpu util

Active CPUs: 4

Page 1 of 2

ProcList CPU Rpt Mem Rpt Disk Rpt NextKeys SlctProc Help Exit

Glance in character Mode – Memory Report



X Terminal on snick.rsn.hp.com

GlancePlus C.03.71.23 14:44:03 rx5670b ia64 Current Avg High

Event	Current	Cumulative	Current Rate	Cum Rate	High Rate
Cpu Util	SU				
Disk Util	F				
Mem Util	S SU UB				
Swap Util	R R				

Users= 2

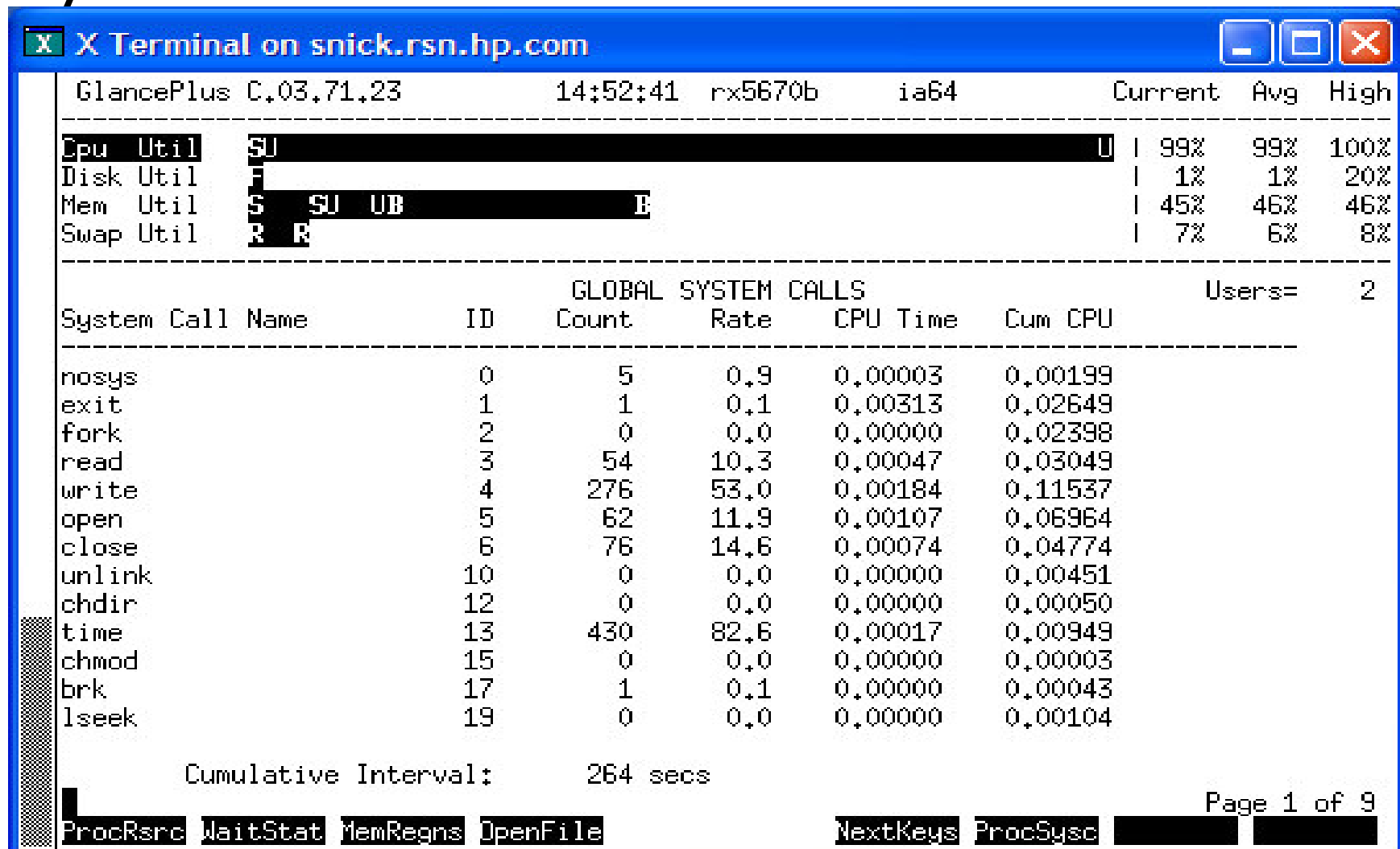
Event	Current	Cumulative	Current Rate	Cum Rate	High Rate
Page Faults	0	31726	0.0	9.0	180.5
Page In	0	8266	0.0	2.3	53.8
Page Out	0	0	0.0	0.0	0.0
KB Paged In	0kb	4.2mb	0.0	1.2	829.2
KB Paged Out	0kb	0kb	0.0	0.0	0.0
Reactivations	0	0	0.0	0.0	0.0
Deactivations	0	0	0.0	0.0	0.0
KB Deactivated	0kb	0kb	0.0	0.0	0.0
VM Reads	0	70	0.0	0.0	15.5
VM Writes	0	112	0.0	0.0	0.4

Total VM : 3.43gb Sys Mem : 3.91gb User Mem: 3.41gb Phys Mem: 48.0gb
Active VM: 3.30gb Buf Cache: 14.4gb Free Mem: 26.3gb

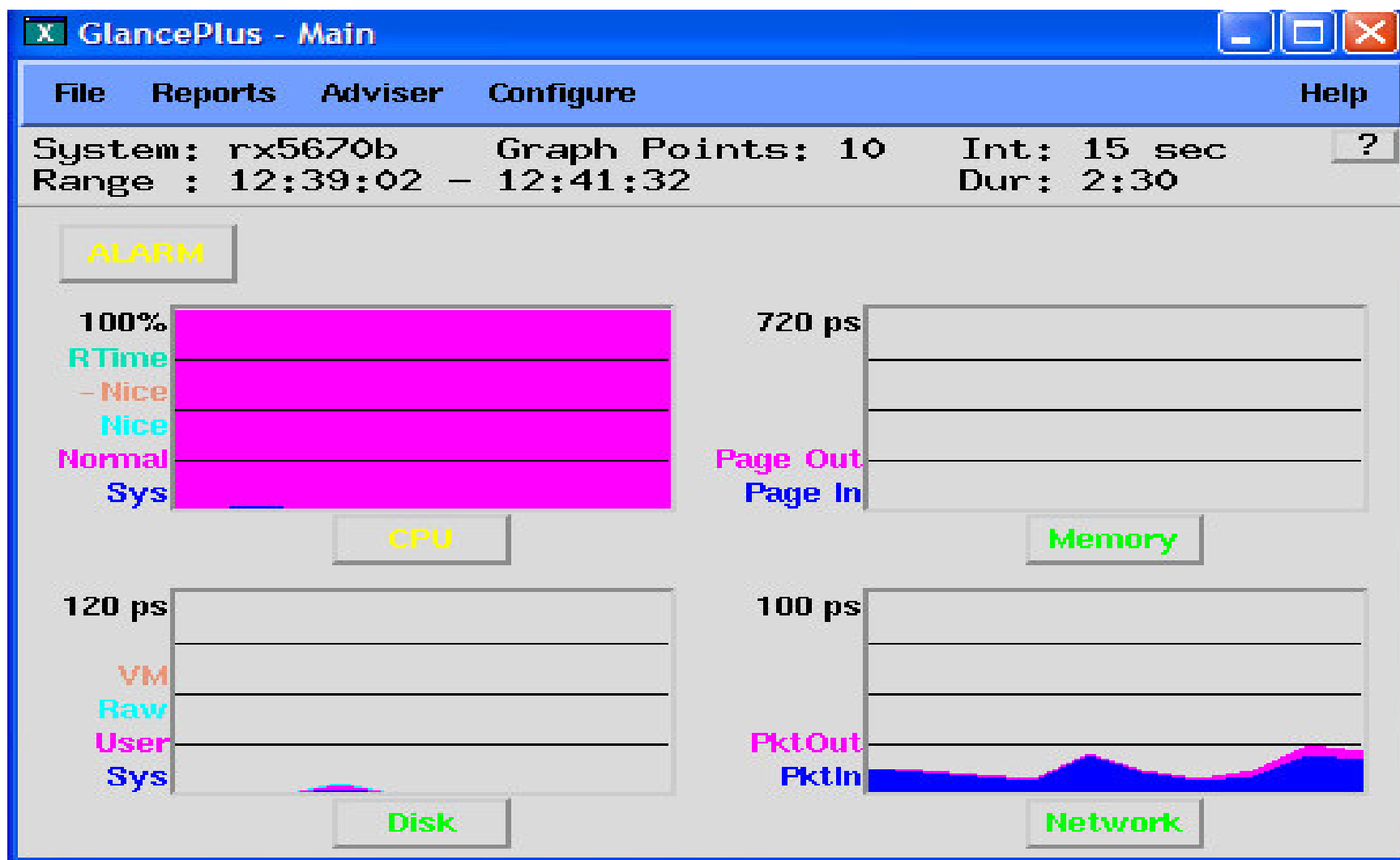
Page 1 of 1

ProcList CPU Rpt Mem Rpt Disk Rpt NextKeys SlctProc Help Exit

Glance in character Mode – Global System Calls



Glance in gpm Mode – Main Screen



Glance in gpm Mode – Processor List



GlancePlus - CPU By Processor

File

Reports

Configure

Help

System: rx5670b Last Update: 12:37:32 Int: 15 sec T ?

CPUs: 4

CPU ID	CPU State	Total CPU %	Normal CPU %	Nice CPU %	NNice CPU %	RealTm CPU %	SysCall CPU %
3	Enable	100.0	98.9	0.0	0.1	0.0	0.3
2	Enable	100.0	99.4	0.0	0.0	0.0	0.1
0	Enable	100.0	98.9	0.0	0.0	0.0	0.5
1	Enable	100.0	98.9	0.0	0.0	0.0	0.4

Glance in gpm Mode – Memory Report



GlancePlus - Memory Report						
File Reports						Help
System: rx5670b		Last Update: 12:32:47		Int: 15 sec		?
Event	Current	Cumulative	Curr Rate	Cum Rate	High Rate	
Page Faults	299	1588	19.9	8.7	198.7	
Page Ins	80	594	5.3	3.2	191.2	
Page Outs	0	0	0.0	0.0	0.0	
Paged In (kb)	0	92	0.0	0.5	6.1	
Paged Out (kb)	0	0	0.0	0.0	0.0	
Reactivated Proc	0	0	0.0	0.0	0.0	
Deactivated Proc	0	0	0.0	0.0	0.0	
Deactivated (kb)	0	0	0.0	0.0	0.0	
VM Reads	0	6	0.0	0.0	0.4	
VM Writes	1	6	0.0	0.0	0.1	
Total VM : 3.44gb		Phys Mem : 48.0gb		Sys Mem : 3.91gb		
Active VM: 3.29gb				Buf Cache: 14.4gb		
				User Mem : 3.42gb		
				Free Mem : 26.3gb		

Glance in gpm Mode – Disk Report



GlancePlus - Disk Report									
File Reports									
System: rx5670b		Last Update: 15:02:20		Int: 15 sec					
Req Type	Requests	%	Rate	Bytes	Cum Req	% Cum	Rate	Cum	Bytes
Local Logl Reads	52	56.5%	3.4	13.6mb	11300	91.5%	37.4	13.6mb	
Local Logl Writes	40	43.5%	2.6	144kb	1050	8.5%	3.4	144kb	
Local Phys Reads	0	0.0%	0.0	0kb	0	0.0%	0.0	0kb	
Local Phys Writes	26	100.0%	1.7	9kb	508	100.0%	1.6	1.3mb	
Local User	0	0.0%	0.0	0kb	95	18.7%	0.3	1.1mb	
Local Virtual Mem	0	0.0%	0.0	0kb	8	1.6%	0.0	0kb	
Local System	8	30.8%	0.5	9kb	176	34.6%	0.5	106kb	
Local Raw	18	69.2%	1.2	0kb	229	45.1%	0.7	40kb	
Remote Logl Reads	0	0.0%	0.0	32kb	1433	100.0%	4.7	32kb	
Remote Logl Writes	0	0.0%	0.0	0kb	0	0.0%	0.0	0kb	
Remote Phys Reads	0	0.0%	0.0	0kb	3	100.0%	0.0	0kb	
Remote Phys Writes	0	0.0%	0.0	0kb	0	0.0%	0.0	0kb	
Remote User	0	0.0%	0.0	0kb	0	0.0%	0.0	0kb	
Remote Virtual Mem	0	0.0%	0.0	0kb	0	0.0%	0.0	0kb	
Remote System	0	0.0%	0.0	0kb	3	100.0%	0.0	0kb	
Remote Raw	0	0.0%	0.0	0kb	0	0.0%	0.0	0kb	
Event	Current	Cum	Curr %	Avg %	High %				
Read Cache Hits	120	17331	100.0	100.0	100.0				
Write Cache Hits	52	2020	74.3	86.2					
DNLC Hits	0	0	0.0	0.0	0.0				
DNLC Longs	0	0	0.0	0.0	0.0				

Glance in gpm Mode – Memory Report



GlancePlus - Memory Report						
File Reports						Help
System: rx5670b		Last Update: 12:32:47		Int: 15 sec		?
Event	Current	Cumulative	Curr Rate	Cum Rate	High Rate	
Page Faults	299	1588	19.9	8.7	198.7	
Page Ins	80	594	5.3	3.2	191.2	
Page Outs	0	0	0.0	0.0	0.0	
Paged In (kb)	0	92	0.0	0.5	6.1	
Paged Out (kb)	0	0	0.0	0.0	0.0	
Reactivated Proc	0	0	0.0	0.0	0.0	
Deactivated Proc	0	0	0.0	0.0	0.0	
Deactivated (kb)	0	0	0.0	0.0	0.0	
VM Reads	0	6	0.0	0.0	0.4	
VM Writes	1	6	0.0	0.0	0.1	
Total VM : 3.44gb		Phys Mem : 48.0gb		Sys Mem : 3.91gb		
Active VM: 3.29gb				Buf Cache: 14.4gb		
				User Mem : 3.42gb		
				Free Mem : 26.3gb		

Customizing Glance – Driver Script



```
#!/bin/csh -f
```

```
#
```

```
# intv = collection intervals in sec. default is 5 sec.
```

```
setenv intv $1
```

```
# count = how many times to collect the metrics.
```

```
setenv count $2
```

```
#
```

```
/opt/perf/bin/glance -iterations $count -i $intv -  
adviser_only -syntax ./thread.info.2
```

```
#
```

Customizing Glance – Adviser Syntax



```
process loop {  
  if ((proc_user_name == "sankaran" ) and (proc_cpu_total_util > 4) )  
  then  
    thread loop {  
      if ( (thread_proc_name == "standard.x") and (thread_cpu_total_util >  
6) ) then  
        print gbl_stattime, " ",  
          thread_proc_name, " ",  
          thread_cpu_total_util, " ",  
          thread_cpu_total_time_cum, " ",  
          thread_mem_res, " ",  
          thread_mem_virt  
      }  
    }  
  }  
}
```

Customizing Glance – Sample Output



TIME	PROCESS	CPU %	CPUTIME	RSS	VM
08:38:55	standard.x	98.3	187.2	1.67gb	2.03gb
08:38:55	standard.x	50.5	7.6	1.67gb	2.03gb
08:39:10	standard.x	99.3	202.1	2.12gb	2.49gb
08:39:10	standard.x	99.0	22.5	2.12gb	2.49gb
...					
08:49:56	standard.x	92.3	822.8	4.95gb	5.32gb
08:49:56	standard.x	98.7	467.6	4.95gb	5.32gb
08:50:11	standard.x	97.3	837.3	4.95gb	5.32gb
08:50:11	standard.x	78.5	479.3	4.95gb	5.32gb

vmstat – virtual memory statistics

Tool: vmstat

Where: /bin/vmstat

Description:

vmstat reports statistics about process, virtual memory, trap and CPU activity

Features:

- simple and easy-to-use interactive tool
- available only in character mode
- very little overhead

vmstat – Examples of Usage

vmstat [options] [interval] [count]

vmstat recognizes the following options:

- d : report disk transfer information
- n : separate memory and cpu info. in 80-column format
- S : report swapped in (si) and swapped out (so) processes
- f : report no. of forks and no. of pages of virtual memory involved since boot-up
- s : prints paging related events since boot-up or it is cleared by -z option
- z : clears all accumulators in the kernel sum structures

vmstat – Sample Outputs

- rx5670b% vmstat -n
- VM
- memory
- page
- faults
- avm free re at pi po fr de sr in sy cs
- 925893 6807822 5 0 1 0 0 0 0 0 1063 1792 167
- CPU
- cpu
- procs
- us sy id r b w
- 5 0 95 5 1 0
- 5 0 95
- 5 0 94
- 6 0 94

vmstat – Sample Outputs

- rx5670b% vmstat -n 2 2
- VM
- memory page faults
- avm free re at pi po fr de sr in sy cs
- 926544 6807818 5 0 1 0 0 0 0 0 0 1063 1792 167
- CPU
- cpu procs
- us sy id r b w
- 5 0 95 5 0 0
- 5 0 95
- 5 0 94
- 6 0 94
- 926544 6807738 2 0 0 0 0 0 0 0 0 1173 505 350
- 94 6 0 5 0 0
- 100 0 0
- 100 0 0
- 100 0 0

vmstat – Sample Outputs



- rx5670b% vmstat -d
-
-
-
-

procs			memory				page				faults				cpu		
r	b	w	avm	free	re	at	pi	po	fr	de	sr	in	sy	cs	us	sy	id
5	0	0	922600	6807818	5	0	1	0	0	0	0	0	1063	1791	167	5	0

95

- Disk Transfers
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-
-

device	xfer/sec
c0t0d0	0
c4t0d0	0
c5t1d0	0
c4t2d0	0
c5t3d0	0
c4t4d0	0
c5t5d0	0
c4t6d0	0
c5t7d0	0
c4t8d0	0
c5t9d0	0
c0t2d0	0

sar – system activity reporter

Tool: sar

Where: /bin/sar

Description:

sar samples cumulative activity counters in the operating system to report user, system and idle time waiting for I/O etc.

Features:

- simple and easy-to-use interactive tool
- available only in character mode
- very little overhead

sar – Examples of Usage ...

sar [options] [-p psetid] [-o file] interval count

sar [options] -f file (to extract data from
previously recorded in 'file')

where options are one or more of:

- u : report cpu utilization (default)
- b : report buffer activity
- d : report activity for each block device
- y : report tty device activity
- c : report system calls

sar – Examples of Usage ...

- w : report system swapping and switching
- a : report use of file access system routines
- v : report of status of text, process, inode and file tables
- m : report message and semaphore activities
- p : report specified pset info. (psetid needed)
- S : report no. of System V select() calls per sec.
- M : report per cpu data on a multi-processor system
- A : report data using all possible options

sar – Examples of Usage

`sar 2 5` : watch cpu activity for 10 seconds

`sar -o temp 60 10` : watch cpu activity for 10 min. at 1 min interval and save data (binary)

`sar -M 30 100 > sar.log` : save cpu activity for 50 min. at 30 sec. interval in a text file

`sar -d -f temp` : review disk and tape activity from the previously saved (binary) data file 'temp'

`sar -u -M -f temp` : review cpu utilization on a multi-processor system later

sar – Sample Outputs

- rx5670b% sar 5 6
- HP-UX rx5670b B.11.23 U ia64 07/08/04
- | 13:54:40 | %usr | %sys | %wio | %idle |
|----------|------|------|------|-------|
| 13:54:45 | 100 | 0 | 0 | 0 |
| 13:54:50 | 100 | 0 | 0 | 0 |
| 13:54:55 | 100 | 0 | 0 | 0 |
| 13:55:00 | 100 | 0 | 0 | 0 |
| 13:55:05 | 100 | 0 | 0 | 0 |
| 13:55:10 | 100 | 0 | 0 | 0 |
| Average | 100 | 0 | 0 | 0 |

sar – Sample Outputs



- rx5670b% sar -M 5 2
- HP-UX rx5670b B.11.23 U ia64 07/08/04
- | 13:57:09 | cpu | %usr | %sys | %wio | %idle |
|----------|--------|------|------|------|-------|
| 13:57:14 | 0 | 100 | 0 | 0 | 0 |
| | 1 | 100 | 0 | 0 | 0 |
| | 2 | 100 | 0 | 0 | 0 |
| | 3 | 100 | 0 | 0 | 0 |
| | system | 100 | 0 | 0 | 0 |
| 13:57:19 | 0 | 100 | 0 | 0 | 0 |
| | 1 | 100 | 0 | 0 | 0 |
| | 2 | 100 | 0 | 0 | 0 |
| | 3 | 100 | 0 | 0 | 0 |
| | system | 100 | 0 | 0 | 0 |
| Average | 0 | 100 | 0 | 0 | 0 |
| Average | 1 | 100 | 0 | 0 | 0 |
| Average | 2 | 100 | 0 | 0 | 0 |
| Average | 3 | 100 | 0 | 0 | 0 |
| Average | system | 100 | 0 | 0 | 0 |

sar – Sample Outputs

- rx5670b% sar -d -o temp 5 2
- HP-UX rx5670b B.11.23 U ia64 07/08/04
- 14:01:49 device %busy avque r+w/s blks/s avwait avserv
- 14:01:54 c0t0d0 1.00 0.50 2 24 0.00 7.39
- c4t0d0 0.20 0.50 0 3 0.00 10.82
- 14:01:59 c0t0d0 0.40 0.50 0 6 0.01 6.14
- Average c0t0d0 0.70 0.50 1 15 0.00 7.14
- Average c4t0d0 0.10 0.50 0 2 0.00 10.82

tusc – trace unix system calls

Tool: tusc

Where: /homegrown/bin/tusc

Description:

tusc traces the system calls a process invokes and the signals it receives. It can attach to a live process by providing PID and is similar to 'truss' (Solaris)

Features:

- relatively easy and useful tool to track system calls
- available only in character mode
- little overhead

tusc – Options and Arguments ...

tusc [options] command [args ...] | pid [pid ...]

where options and arguments:

- a : show exec arguments
- A : append to output file
- b bsize: dump 'bsize' max bytes (-r/-w)
- c : count syscalls instead of printing trace
- e : show environment variables
- E : show syscall entries
- f : follow forks
- F : show kernel's ttrace feature level
- g : don't attach to members of my session
- h : show state of all processes when idle

tusc – Options and Arguments ...

-l :	don't display interruptible syscalls
-l start[/stop]:	single-step and show instructions
-k :	keep alive (wait for *all* processes)
-l :	print lwpsids
-L [!] <i>lwps</i> :	[un]select these <i>lwps</i>
-n :	print process names
-o [<i>file</i> <i>fd</i>]:	send trace output to file or <i>fd</i>
-p :	print pids
-Q :	be quiet about some warnings
-r [!] <i>[fd all]</i> :	dump read buffers
-R :	show syscall restarts

tusc – Options and Arguments

-s [!]syscalls:	[un]select these syscalls
-S [!]signals:	[un]select these signal
-t :	detach process if it becomes traced
-T timestamp:	print time stamps
-u :	print user thread IDs (pthreads)
-v :	verbose (some system calls only)
-V :	print version
-w [!][fd all]:	dump write buffers
-x :	print raw (hex) arguments
-z :	only show failing syscalls

tusc – Examples of Usage ...

tusc date : trace the date command

tusc 1 : trace init process, if superuser

tusc -f -o make.trace make : trace make and its descendents and save the output in a file

tusc -s read,write cat /etc/passwd : show the read and write syscalls from 'cat /etc/passwd'

tusc -s !sig,mmap sleep 10 : show all system calls except mmap and those having 'sig' in their name

chatr – change program's attributes



Tool: chatr

Where: /bin/chatr

Description:

Changes program's internal attributes for 32-bit and 64-bit applications

Features:

- easy to use without needing any special build
- significant performance improvements of applications with large no. of TLB misses
- many new features now

chatr – Examples of Usage

chatr {segment info.} [options] file, where options are:

- s : perform its operations silently
- z/-Z : enable/disable dereferencing null pointers
- +id enable : enable use of inter-leaved memory
- +k enable : enable kernel assisted branch prediction
- +r enable : enable static branch prediction
- +l enable : enable instrumentation by caliper
- +pd M : hint for virtual data page size of M
- +pi N : hint for virtual instruction [text] page size of N

mpsched – multi-processor scheduler



Tool: mpsched

Where: /bin/mpsched

Description:

Provides control to a process on which processor (spu) or locality domain (ldom) it executes. It is accomplished by binding a process to spu or ldom or using a different launch policy.

Features:

- very easy to use
- significant performance improvements

mpsched – Examples of Usage ...



mpsched [options] command, where options are

- h : prints a help message
- s : returns hardware configuration (no other)
- g : enables gang-scheduling (no other)
- q : query about process/thread bindings
- c spu : binds process to a processor, spu
- l ldom : binds process to a ldom
- p pid : specifies process by pid
- T policy : applies policy to a team of threads
- P policy : applies policy to processes

mpsched – Examples of Usage

mpsched -P policy-p -T policy-t

where policy-p or policy-t can be one of:

FILL	:	fill-first
FILL_TREE	:	fill-first for parent and descendents
LL	:	least-loaded
PACKED	:	packed launch
RR	:	round robin
RR_TREE	:	round robin for parent and descendents
NONE	:	default launch policy

mpsched – Sample Results



Application: AVL/FIRE V 8.2.2

System: RX8620

OS: HP-UX 11.23

Parallel Model	No. of CPUs	Scheduling Policy	Elapsed Time (hh:mm:ss)
MPI	8	Default	1:00:05
MPI	8	-P FILL	1:00:25
MPI	8	-P RR	0:58:38
SMP	8	Default	2:42:25
SMP	8	-T FILL	2:38:08
SMP	8	-T RR	2:54:60

mpsched – Sample Results



Application: EXA/PowerFlow V3.4p3

System: Integrity Superdome (64 CPUs)

OS: HP-UX 11.23

No. of CPUs	Scheduling Policy	Data Page Size	Elapsed Time (hh:mm:ss)
64	-P FILL	Default	2:07:54
64	-P RR	Default	2:05:58
64	-P RR	256 MB	1:46:50

mpsched – Sample Results



Application: ISV Codes

System: RX5670

OS: HP-UX 11.23

Application	No. of CPUs	Elapsed Time (s) with mpsched	Elapsed Time (s) w/o mpsched
ABAQUS	4	1352	1440
FLUENT	4	1934	2179
LS970	4	756	888

mpsched – Sample Throughput Results



Running 4 copies of 4-way Abaqus (Standard) jobs on 4 cells

```
#!/bin/csh -f
```

```
mpsched -P RR runtput
```

```
#!/bin/csh -f
```

```
foreach rdir(1 2 3 4)
```

```
mpsched -P PACKED T$dir/runit4
```

```
end
```

Log File Name	No. of CPUs	Elapsed Time with mpsched	Elapsed Time w/o mpsched
runit4.log.1	4	14:55:47	16:01:00
runit4.log.2	4	14:37:44	15:58:35
runit4.log.3	4	14:35:25	15:58:22
runit4.log.4	4	14:43:14	15:58:23



Pset – processor sets

Tool: psrset

Where: /usr/sbin/psrset

Description:

Create and manage processor sets

Features:

- optional HP-UX product
- dynamic configuration and application binding
- dedicated resources to batch and special needs
- integration with PRM, nPars and VParS
- hardware and platform independent

Pset – before you use



- System is booted with a default Pset with all cpus
- Superuser is always the default owner of default Pset
- Default Pset can never be destroyed
- A processor can belong to only one Pset at any time
- Processor 0 is always part of default Pset
- Users can create and destroy their own Psets
- Superuser or user in PRIV_PSET privilege group can only reassign processors (except 0) to user Psets
- Default Pset is always available to all apps/users
- Destroying user Psets moves applications and processors to default Pset

Pset – Examples of Usage

psrset options [command], where options are:

- c creates a Pset with default psetid
- i 1 queries attributes of Pset 1
- d 1 2 destroys Pset 1 and 2
- d all destroys all Psets of a user
- c 2 4 6 creates Pset 2 and assigns cpus 4 & 6
- a 3 5 7 assigns cpus 5 & 7 to Pset 3
- b 1 432 binds pid 432 to Pset 1
- U 2 21 binds process of uid 21 to Pset 2
- e 3 a.out executes a.out in Pset 3
- t 1 PERM=700 changes permission of Pset 1 to 700

LSF HPC – Custom LSF with Psets Functionality



- Seamless use simplifies processor selection and use
- LSF HPC integrates Psets through LSF job submission
- HPC version selects cpus; creates Psets; runs job; destroys Psets at the end
- To use Pset, add "pset" to the resource list and submit jobs with 'bsub' as:

-ext "PSET[CELLS=num_cells | PTILE=cpus_per_cell"

bsub -n 6 ... CELLS=4 will place 2,2,1,1 cpus on 4 cells

bsub -n 6 ... PTILE=2 will place 2,2,2 cpus on 3 cells

- Useful environment variables:

DEFAULT_EXTSCHED=PSET[CELLS=m | PTILE=n]

MANDATORY_EXTSCHED=PSET[CELLS=m PTILE=n]

where m=num_cells and n=cpus_per_cell



Processor Placement with MPCTL

- Mpctl can query no. of cpus or ldoms on a system and assign processes/threads to cpus and ldoms
- Multiprocessor control highly dependent on hardware
- Source level mechanism to manage cpus and ldoms
- Examples of mpctl usage:

```
#include <sys/mpctl.h>
```

```
int mpctl(mpc_request_t request, spu_t spu, pid_t pid)
```

```
int mpctl(mpc_request_t request, ldom_t ldom, pid_t pid)
```

where request specifies action to be taken:

MPC_GETNUMCPUS_SYS: total no. of cpus on a system

MPC_GETNUMLDOMS_SYS: no. of locality domains on a system

Caliper – source level profiling tool



Tool: caliper

Where: /opt/caliper/bin/caliper

Description:

Measure and report application profiling data

Features:

- No special preparation of measured program
- Uses Itanium's PMU counters for some metrics
- Other metrics are collected by probe code inserted into the application by caliper
- Application can be written in C, C++, Fortran and assembly or combination of these

Caliper – Examples of Usage ...



- Data Collection Mode:

```
caliper config_file [collect_opts] [report_opts]  
           program [program_args]
```

configuration files: total_cpu, cgprof, fprof, pbo
dcache_miss, dtlb_miss etc.

- Report Only Mode

```
caliper report -d datafile [report_opts]
```

report formats: ascii, html or both

- Information only Mode

```
caliper info [info_options]
```

Caliper – Examples of Usage

- `caliper total_cpu --process=all a.out`
measure and report total instructions, nops and cpu cycles expended from all processes
- `caliper total_cpu -o total_cpu.report a.out`
- `caliper cgprof a.out`
measure and report an extended call graph profile
- `caliper pbo a.out`
run, measure and produce 'flow.data' file which can be used by compiler (+Oprofile=use) to generate a highly optimized code
- `caliper fprof --process=all --module-exclude=libc -o fprof.report`

prospect – tool to measure performance



Tool: prospect

Where: /apps/prospect/prospect_[ver]/prospect

Description:

Prospect uses Kernel Instrumentation (KI) tracing and Kernel Timing Clock (KTC) to gather statistical profile of all processes as well as the system

Features:

- collects info. over an interval of time
- works in two modes: 'super' time command and statistical profiler
- profiles many applications both in HP-UX and Linux

prospect – Examples of Usage ...



Prospect [options] command

Where options are:

- v : time using KTC clocks including sys. Calls
- V2 : time using KI tracing of just 'command'
- V3 : time 'command' and its descendents
- V4 : time *all* processes using KI tracing
- Vk : generate /stand/vmunix profile summary
- M : merge multi-threaded profiles into one process-wide profile
- F filename: write an ascii output file (default:none)
- ?? : more advanced command line options

prospect – Examples of Usage



prospect -V2 -f output2 prog1

outputs information only about prog1

prospect -V3 -f output3 prog2

outputs information about prog2 and all its descendents

prospect -V4 -f output4 prog3

outputs information about prog3 and all other processes that ran during that interval

prospect -T Tfile1 prog4

records all traces during prog4 ran into a binary file

Prospect -t Tfile1 -f output5

read the trace file Tfile1 and write output file output5

Miscellaneous Commands ...



- iostat : reports I/O statistics for each active disk
- ps : displays info. about selected processes
- time : reports elapsed, user and system times
- top : displays real-time top processes
- uptime, w: shows system uptime and current activities
- acct : accounting program for system resources
- netstat: displays general networking statistics
- netsstat: displays NFS networking statistics
- perfview: helps viewing MeasureWare data in GUI

Miscellaneous Commands



- `ipcs` : shows status info. about IPC activities
- `ipcrm` : removes specified message queue, semaphore set or shared-memory segment
- `nice` : run a command at non-default priority
- `renice` : alters the priority of a running program
- `bdf`, `df` : displays the amount of free disk space
- `dmesg` : shows the most recent diagnostic messages in system buffer
- `swapinfo`: prints info. About device and file system paging space
- `sysdef`: analyzes and reports tunable configuration parameters from a running system

Conclusions



- HP-UX offers a wide range of tools and commands
- Complexity come into play with HP's great choices
- Knowledge and appropriate use of these tools and commands can identify and remedy system bottlenecks
- Careful use of performance tools can greatly enhance application performance
- Applying these tools and commands can increase the overall productivity of HP systems
- Finally, we can provide total customer experience with HP's high-tech solutions but at low cost

HP WORLD 2004

Solutions and Technology Conference & Expo

Co-produced by:



RECOMMENDED TRAINING VENUE FOR THE
HP Certified Professional

