HP Escalation Team Performance Troubleshooting Techniques and Tools

Ken Johnson **Pat Kilfoyle**

Jan Weaver **Mike Chisholm**

HP Escalation Engineers

HP WORLD 2003
Solutions and Technology Conference & Expo

Americas Escalation Team
Performance troubleshooting

Ken Johnson

Escalation Engineer

Hewlett-Packard Company

19410 Homestead Rd. MS 4345

Cupertino, CA   95014

E-mail: ken.johnson@hp.com

Phone: (408) 447-1307

Americas Escalation Team Performance troubleshooting

- To share the strategies, tactics and tools used by the HP Americas Escalation Team (AET) to resolve performance escalations – using real world examples and case studies
- We will not deal with system tuning, capacity planning or benchmarking

Outline

1. The AET Perspective: Emergency Room
2. Defining the Performance Problem
3. Metrics and Tools
4. Is there a Bottleneck ?
5. Is the Work Necessary?
6. Looking for Anomalies
7. Isolating Components
8. Knowing Your System
9. Rules of Thumb

The AET perspective: emergency room

- ER the TV show - we do triage to stop the bleeding
- Stabilize the system as fast as possible
- Quickly identify the first steps
  - Is this an HP defect / config issue / 3rd party issue ?
  - Often our value-add is to point in the right direction
- We have a system perspective
  - Understand interactions between HW, OS, Network, DB, Application
  - We train our engineers for a system perspective

Defining the performance problem

Things we want to know in the first minutes

- When did the performance problem start ?
- How do you know you have a problem ?
  - Is this a user/business impacting problem or a metric-only issue ?
  - Is the problem quantified ?
- Is the hardware and OS base stable and consistent ?
- What changed ?

Defining the performance problem

- Quantification
  - Allows you to measure the objective effect of changes
  - Define the current state and the goal
- Changing only one thing at a time
- Characterize and focus the problem
  - System wide or particular application?
  - All the time or specific time of day?
  - Network access or local access?
  - NFS mounts or local disks?
  - Consistent or erratic?

Metrics & Tools

- What are the thermometers really measuring
- Metrics are simply statistics produced by software
  - Some of our escalations are with performance tools
- Always have more than one data point and always use more than one tool
- Your tools can affect the environment

Metrics & Tools

Be sure what a metric is really measuring

- Wait time/service time
- Page out/swap out/deactivation
- Run queue/load average
- Inode table utilization

Metrics & Tools
first tools

- uptime
- model
- uname –a
- sysdef
- top

Metrics & Tools
first metric: system/user cpu

- What is system CPU?

- Why is it important?
  - Points at initial directions to pursue root cause
  - HP owns this code

- High system CPU can point to:
  - High number of system calls
  - Memory I/O problems
  - Thrashing and spinning in the kernel

- AET has visibility into system CPU utilization
  - There are utilities we use to do kernel profiling on production systems

Metrics & Tools
sar – cpu report

```
$ sar 5 5

HP-UX karoo B.11.00 A 9000/820    05/09/03

17:47:15     %usr     %sys     %wio    %idle
17:47:20       12       23        7       58
17:47:25        1        6        6       87
17:47:30        0        0        1       98
17:47:35        0        0        2       98
17:47:40        0        4        1       94

Average         3        7        3       87
```

Metrics & Tools
sar – disk report

```
# sar -d 1 2

HP-UX cecl3 B.11.00 U 9000/800    07/08/03

14:33:20  device   %busy  avque r+w/s  blks/s  avwait  avserv
14:33:21  c1t2d0   79.21  1.48    208    2259    6.05    7.33
14:33:22  c1t2d0   77.00  0.93    272    2244    5.53    4.21
          c1t0d0    2.00  0.50      2       4    1.37    6.17

Average   c1t2d0   78.11  1.17    240    2252    5.76    5.57
Average   c1t0d0    1.00  0.50      1       2    1.37    6.17
```

Metrics & Tools
measureware tools

- Tightly integrated with HPUX kernel
- On most mission critical systems
- Trial version available for escalations
- Glance / GPM
- Scope / PerfView

Metrics and Tools
Glance

```
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
```

```
B3692A GlancePlus C.03.70.00     10:26:35  bokmaai 9000/820    Current  Avg   High
------------------------------------------------------------------------------------
CPU  Util   SAU                         L                     | 51%     7%     56%
Disk Util   F                                                 |  2%     5%     53%
Mem  Util   S             SU      UB                       B  | 95%    94%     95%
Swap Util   U                  UR      R                      | 55%    54%     55%
------------------------------------------------------------------------------------
                                  PROCESS LIST                     Users=      3
                              User       CPU Util     Cum     Disk              Thd
Process Name    PID    PPID Pri Name   ( 200% max)    CPU    IO Rate      RSS   Cnt
------------------------------------------------------------------------------------
memm3          24951  24926 239 kenj    99.8/77.6    12.8  0.0/ 0.3    692kb     1
glance         24919  24905 154 kenj     0.8/ 1.1     1.7  0.0/ 0.0    4.9mb     1
glance         24632  23393 154 root     0.8/ 0.8     8.1  0.0/ 0.0    5.0mb     1
midaemon        1303      1 -16 root     0.2/ 0.1  4373.4  0.0/ 0.0    9.3mb     3
diaglogd        1574   1161 168 root     0.0/ 0.0    88.5  0.0/ 0.0    440kb     1
nfsd            1529   1522 154 root     0.0/ 0.0     0.0  0.0/ 0.0    472kb     1
nfsd            1530   1522 154 root     0.0/ 0.0     0.0  0.0/ 0.0    472kb     1
nfsd            1531   1522 154 root     0.0/ 0.0     0.0  0.0/ 0.0    472kb     1
registrar       1581    636 154 root     0.0/ 0.0     0.7  0.0/ 0.0    388kb     1
nfsd            1532   1522 154 root     0.0/ 0.0     0.0  0.0/ 0.0    472kb     1
nfsd            1533   1521 154 root     0.0/ 0.0     0.0  0.0/ 0.0    472kb     1
nfsd            1534   1521 154 root     0.0/ 0.0     0.0  0.0/ 0.0    468kb     1
                                                                    Page 1 of 12
------------------------------------------------------------------------------------
 Process    CPU     Memory    Disk     37     1   Next    Select    Help    Exit
  List     Report   Report   Report               Keys    Process          Glance
```

B3692A GlancePlus C.03.70.00 10:27:17 bokmaai 9000/820 Current Avg High

		Current	Avg	High
CPU Util	SAU L	52%	16%	56%
Disk Util	F	2%	5%	53%
Mem Util	S SU UB B	95%	94%	95%
Swap Util	U UR R	55%	54%	55%

SYSTEM TABLES REPORT Users= 3

System Table	Available	Used	Utilization	High(%)
Proc Table (nproc)	5620	143	3	3
File Table (nfile)	10539	640	6	6
Shared Mem Table (shmmni)	200	13	7	7
Message Table (msgmni)	50	2	4	4
Semaphore Table (semmni)	64	31	48	48
File Locks (nflocks)	200	37	19	19
Pseudo Terminals (npty)	60	0	0	0
Buffer Headers (nbuf)	na	72930	na	na

Page 1 of 2

Process CPU Memory Disk 37 1 Next Select Help Exit

```
B3692A GlancePlus C.03.70.00    10:27:47  bokmaai 9000/820    Current  Avg  High
--------------------------------------------------------------------------------
CPU  Util   SSAU                           U                   | 54%   21%   56%
Disk Util   F                                                  |  2%    4%   53%
Mem  Util   S             SU      UB                       B   | 95%   94%   95%
Swap Util   U                  UR      R                       | 55%   54%   55%
--------------------------------------------------------------------------------
                              SYSTEM TABLES REPORT                   Users=    3

System Table                 Available    Requested        Used          High
--------------------------------------------------------------------------------
Inode Cache (ninode)              6488           na           0             0
Shared Memory                   12.5gb       30.2mb
Message Buffers                  800kb           na         0kb           0kb
Buffer Cache                   512.0mb           na     512.0mb            na
Buffer Cache Min                51.2mb
Buffer Cache Max               512.0mb
DNLC Cache                       11608

Model     : 9000/820/D380    Phys Memory : 1024mb  Network Interfaces :   4
OS Name   : HP-UX            Number CPUs :    2    Number Swap Areas  :   2
OS Release: B.11.11          Number Disks:   11    Avail Volume Groups:   4
OS Kernel Type: 64 bits      Mem Region Max Page Size: 64.0mb
                                                                  Page 2 of 2
--------------------------------------------------------------------------------
Process    CPU     Memory    Disk       37    1   Next    Select    Help   Exit
 List     Report   Report   Report                Keys    Process          Glance
```

```
B3692A GlancePlus C.03.70.00     10:29:13  bokmaai 9000/820     Current  Avg  High
-----------------------------------------------------------------------------------
CPU  Util   SAU                          U                      | 51%   30%   56%
Disk Util                                                       |  0%    4%   53%
Mem  Util   S             SU      UB                         B  | 95%   95%   95%
Swap Util   U                  UR       R                       | 55%   54%   55%
-----------------------------------------------------------------------------------
                          GLOBAL SYSTEM CALLS                        Users=    3
System Call Name          ID     Count      Rate    CPU Time    Cum CPU
-----------------------------------------------------------------------------------

read                       3        12       2.3    0.00066    0.00145
write                      4       161      30.9    0.00441    0.01187
open                       5         4       0.7    0.00061    0.00168
close                      6         4       0.7    0.00061    0.00162
time                      13       634     121.9    0.00132    0.00333
brk                       17         0       0.0    0.00000    0.00004
lseek                     19         7       1.3    0.00003    0.00006
getuid                    24         0       0.0    0.00000    0.00000
alarm                     27         0       0.0    0.00000    0.00002
access                    33         1       0.1    0.00005    0.00025
stat                      38        17       3.2    0.00124    0.00982

        Cumulative Interval:       11  secs
                                                                     Page 1 of 5
___________________________________________________________________________________
 Global   Global    System           37     1   Next   Netwk By  NFS     NFS By
  Waits  Syscalls   Tables                      Keys   Intrface  Global  System
```

B3692A GlancePlus C.03.70.00 10:29:45 bokmaai 9000/820 Current Avg High

		Current	Avg	High
CPU Util	SAU U	51%	32%	56%
Disk Util	F	1%	3%	53%
Mem Util	S SU UB B	95%	95%	95%
Swap Util	U UR R	55%	54%	55%

GLOBAL WAIT STATES Users= 3

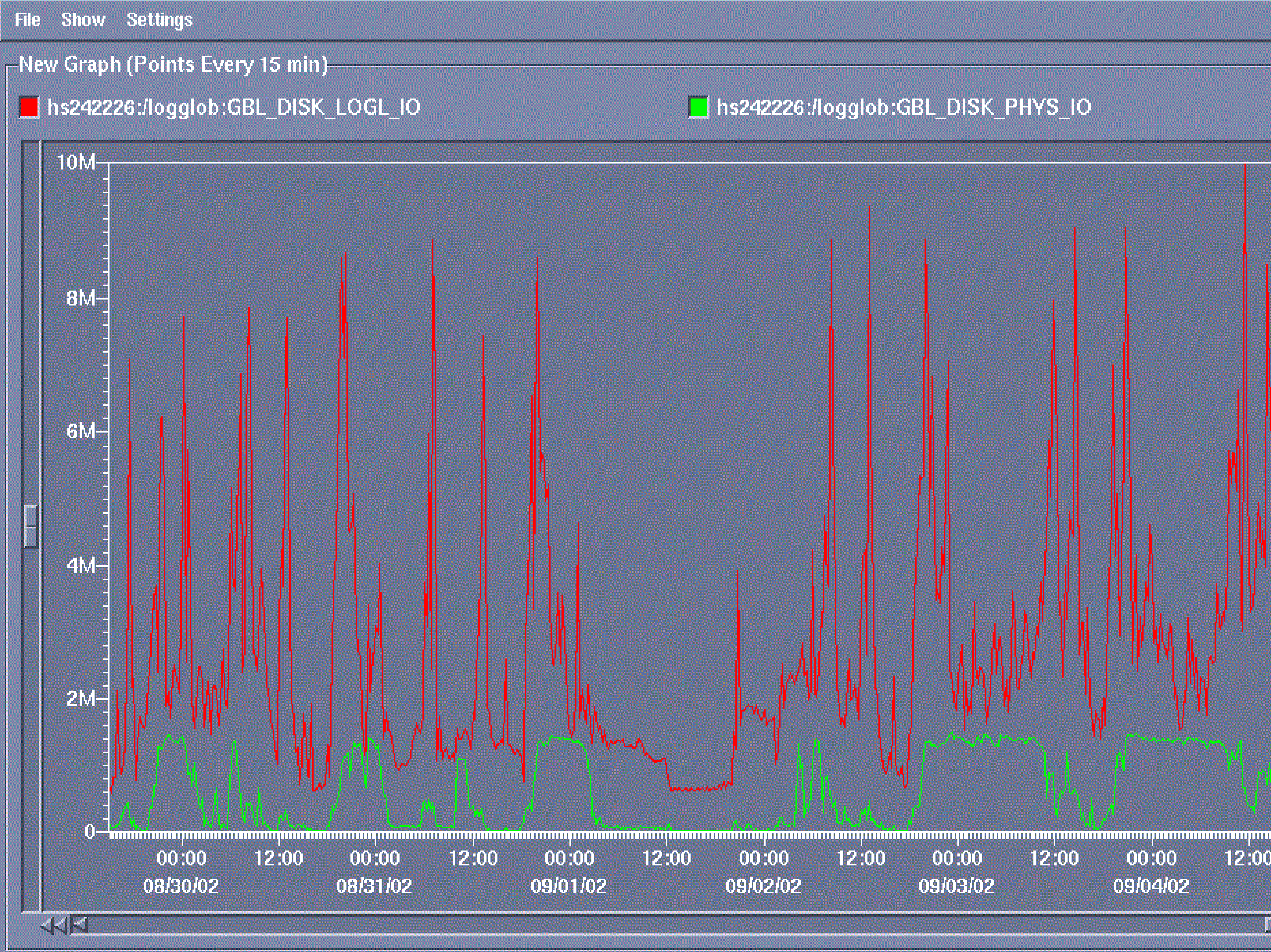
Event	%	Time	Procs/ Threads	Blocked On	%	Time	Procs/ Threads
IPC	0.7	10.48	2.0	Cache	0.0	0.00	0.0
Job Control	0.0	0.00	0.0	CDROM IO	0.0	0.00	0.0
Message	0.0	0.00	0.0	Disk IO	0.0	0.00	0.0
Pipe	0.3	5.24	1.0	Graphics	0.0	0.00	0.0
RPC	0.0	0.00	0.0	Inode	0.0	0.00	0.0
Semaphore	0.3	5.25	1.0	IO	0.6	9.80	1.9
Sleep	32.3	498.18	94.9	LAN	0.0	0.00	0.0
Socket	2.0	31.45	6.0	NFS	0.0	0.00	0.0
Stream	3.0	47.09	9.0	Priority	0.0	0.07	0.0
Terminal	0.3	5.24	1.0	System	42.8	660.81	125.9
Other	17.2	265.57	50.6	Virtual Mem	0.0	0.00	0.0

Page 1 of 1

Global Waits | Global Syscalls | System Tables | 37 1 | Next Keys | Netwk By Intrface | NFS Global | NFS By System

Metrics and Tools
Scope/Perfview

- Scopeux logs data from the MI database once a minute
- The data is summarized at 5 minute intervals
- Good for trend analysis
- Can help with focused troubleshooting



File
Show
Settings
New Graph (Points Every 15 min)
hs242226:/logglob:GBL_DISK_LOGL_IO
hs242226:/logglob:GBL_DISK_PHYS_IO
10M
8M
6M
4M
2M
0
00:00
08/30/02
12:00
00:00
08/31/02
12:00
00:00
09/01/02
12:00
00:00
09/02/02
12:00
00:00
09/03/02
12:00
00:00
09/04/02
12:00

File Configure

☐ Enable Filters ☐ Enable Highlights

Process Name	Date and Time	System CPU %	User CPU %	CPU %	Phys IO Rt	IO Byte Rate	Stop Reason
PSAPPSRV	Mon Sep 2 18:45:00 2002	0.06	8.00	8.05	3.7	72.0	MESG
PSRUN	Mon Sep 2 18:45:00 2002	1.07	8.00	9.08	0.7	32.1	
autocons	Mon Sep 2 18:45:00 2002	16.73	6.00	22.73	0.0	0.0	SOCKT
midaemon	Mon Sep 2 18:45:00 2002	4.19	0.00	12.25	0.0	0.0	SYSTM
ora_arc0_ASRPRD8	Mon Sep 2 18:45:00 2002	0.01	0.00	0.01	11.4	394.1	IO
ora_ckpt_ASRPRD8	Mon Sep 2 18:45:00 2002	1.11	79.00	80.11	0.9	7.6	PRI
ora_dbw0_ASRPRD8	Mon Sep 2 18:45:00 2002	0.05	0.00	0.05	29.3	234.9	SEM
ora_dbw1_ASRPRD8	Mon Sep 2 18:45:00 2002	0.06	0.00	0.06	32.1	258.4	SEM
ora_dbw2_ASRPRD8	Mon Sep 2 18:45:00 2002	0.03	0.00	0.03	24.6	197.5	SEM
ora_dbw3_ASRPRD8	Mon Sep 2 18:45:00 2002	0.05	0.00	0.05	22.8	186.2	SEM
ora_lgwr_ASRPRD8	Mon Sep 2 18:45:00 2002	1.11	0.00	1.11	31.3	679.0	IO
ora_lgwr_ASRSCHD	Mon Sep 2 18:45:00 2002	0.01	0.00	0.01	5.6	49.3	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.21	6.00	6.21	2.0	45.8	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.09	7.00	7.09	7.8	212.0	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	7.28	7.00	14.28	218.4	5120.0	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.08	6.00	6.08	0.4	3.8	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.09	7.00	7.09	5.5	162.8	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.14	6.00	6.14	0.1	1.2	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.12	12.00	12.12	15.3	488.8	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.04	5.00	5.04	6.1	222.8	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.08	7.00	7.08	1.6	48.0	CACHE
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.00	0.00	0.00	8.2	80.6	
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.08	7.00	7.08	1.9	59.3	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.13	12.00	12.13	8.1	275.1	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	1.16	10.00	11.17	17.9	682.2	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.13	7.00	7.13	15.7	518.1	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.13	9.00	9.13	6.8	148.0	PRI
oracleASRPRD8	Mon Sep 2 18:45:00 2002	0.09	7.00	7.09	5.3	194.9	SEM
oracleASRPRD8	Mon Sep 2 18:45:00 2002	6.56	66.00	72.56	0.4	3.7	SLEEP

Select the metrics you wish to display in this window.

PROC_PROC_NAME	PROC_DISK_FS_IO_RATE	PROC_MEM_RES
PROC_CPU_SYS_MODE_UTIL	PROC_DISK_FS_READ	PROC_MEM_VIRT
PROC_CPU_USER_MODE_UTIL	PROC_DISK_FS_READ_RATE	PROC_MEM_WAIT_PCT
PROC_CPU_TOTAL_UTIL	PROC_DISK_FS_WRITE	PROC_MEM_WAIT_TIME
PROC_DISK_PHYS_IO_RATE	PROC_DISK_FS_WRITE_RATE	PROC_MINOR_FAULT
PROC_DISK_LOGL_READ_RATE	PROC_DISK_LOGL_IO_CUM	PROC_NFS_WAIT_PCT
PROC_DISK_LOGL_WRITE_RATE	PROC_DISK_LOGL_IO_RATE_CUM	PROC_NFS_WAIT_TIME
PROC_IO_BYTE_RATE	PROC_DISK_LOGL_READ	PROC_OTHER_IO_WAIT_PCT
PROC_STOP_REASON	PROC_DISK_LOGL_WRITE	PROC_OTHER_IO_WAIT_TIME
INTERVAL	PROC_DISK_PHYS_IO	PROC_OTHER_WAIT_PCT
PROC_APP_ID	PROC_DISK_PHYS_IO_CUM	PROC_OTHER_WAIT_TIME
PROC_CPU_CSWITCH_TIME	PROC_DISK_PHYS_IO_RATE_CUM	PROC_PARENT_PROC_ID
PROC_CPU_CSWITCH_UTIL	PROC_DISK_SUBSYSTEM_WAIT_PCT	PROC_PRI
PROC_CPU_INTERRUPT_TIME	PROC_DISK_SUBSYSTEM_WAIT_TIME	PROC_PRI_WAIT_PCT
PROC_CPU_INTERRUPT_UTIL	PROC_DISK_SYSTEM_IO	PROC_PRI_WAIT_TIME

OK Defaults Cancel Help

File Configure Help

Enable Filters Enable Highlights

Date and Time	Device Name	Disk %	Req Queue	Phys IO Rt
Tue Jun 3 02:55:00 2003	1/10/0/0.97.29.19.0.5.0	0.04	0.00	0.0
Tue Jun 3 02:55:00 2003	0/0/2/0.6.0	4.48	0.00	7.6
Tue Jun 3 02:55:00 2003	0/0/2/1.6.0	3.98	0.00	7.0
Tue Jun 3 02:55:00 2003	0/4/0/0.100.9.19.0.3.1	36.37	0.09	49.6
Tue Jun 3 02:55:00 2003	0/4/0/0.100.9.19.0.3.2	2.78	0.06	2.9
Tue Jun 3 02:55:00 2003	0/4/0/0.100.9.19.0.3.0	25.21	0.47	44.1
Tue Jun 3 02:55:00 2003	1/10/0/0.100.9.19.0.3.4	0.14	0.04	0.3
Tue Jun 3 02:55:00 2003	0/4/0/0.100.9.19.0.3.3	0.21	0.00	0.4
Tue Jun 3 03:00:00 2003	0/4/0/0.100.9.19.0.3.2	3.06	0.00	2.5
Tue Jun 3 03:00:00 2003	0/4/0/0.100.9.19.0.3.1	65.21	0.12	100.1
Tue Jun 3 03:00:00 2003	0/4/0/0.100.9.19.0.3.0	74.06	0.18	139.9
Tue Jun 3 03:00:00 2003	1/10/0/0.100.9.19.0.3.0	76.01	0.19	142.4
Tue Jun 3 03:00:00 2003	0/4/0/0.97.29.19.0.5.0	0.08	0.00	0.0
Tue Jun 3 03:00:00 2003	0/4/0/0.100.9.19.0.3.4	0.19	0.00	0.3
Tue Jun 3 03:00:00 2003	1/10/0/0.100.9.19.0.3.2	3.35	0.01	2.7
Tue Jun 3 03:00:00 2003	1/10/0/0.97.29.19.0.5.0	0.14	0.00	0.1
Tue Jun 3 03:00:00 2003	0/0/2/1.6.0	4.95	0.06	8.9
Tue Jun 3 03:00:00 2003	0/0/2/0.6.0	5.95	0.06	10.0
Tue Jun 3 03:00:00 2003	1/10/0/0.100.9.19.0.3.1	66.57	0.10	102.0
Tue Jun 3 03:00:00 2003	0/4/0/0.100.9.19.0.3.3	2.95	1.88	3.4
Tue Jun 3 03:00:00 2003	1/10/0/0.100.9.19.0.3.4	0.16	0.00	0.3
Tue Jun 3 03:00:00 2003	1/10/0/0.100.9.19.0.3.3	2.60	1.85	3.4
Tue Jun 3 03:05:00 2003	0/4/0/0.100.9.19.0.3.2	14.61	0.00	80.3
Tue Jun 3 03:05:00 2003	0/4/0/0.100.9.19.0.3.1	43.15	0.13	55.6
Tue Jun 3 03:05:00 2003	0/4/0/0.100.9.19.0.3.0	100.00	0.20	206.9
Tue Jun 3 03:05:00 2003	1/10/0/0.100.9.19.0.3.0	100.00	0.20	199.0
Tue Jun 3 03:05:00 2003	0/4/0/0.97.29.19.0.5.0	0.04	0.00	0.0
Tue Jun 3 03:05:00 2003	0/4/0/0.100.9.19.0.3.4	0.18	0.00	0.3
Tue Jun 3 03:05:00 2003	1/10/0/0.100.9.19.0.3.2	14.65	0.01	81.2


Select the metrics you wish to display in this window.
BYDSK_DEVNAME
BYDSK_UTIL
BYDSK_REQUEST_QUEUE
BYDSK_PHYS_IO_RATE
BYDSK_PHYS_READ_RATE
BYDSK_PHYS_BYTE_RATE
BYDSK_PHYS_BYTE
BYDSK_AVG_SERVICE_TIME
BYDSK_DIRNAME
BYDSK_FS_READ
BYDSK_FS_READ_RATE
BYDSK_FS_WRITE
BYDSK_FS_WRITE_RATE
BYDSK_ID
BYDSK_LOGL_READ
BYDSK_LOGL_READ_RATE
BYDSK_LOGL_WRITE
BYDSK_LOGL_WRITE_RATE
BYDSK_PHYS_IO
BYDSK_PHYS_READ
BYDSK_PHYS_READ_BYTE
BYDSK_PHYS_READ_BYTE_RATE
BYDSK_PHYS_WRITE
BYDSK_PHYS_WRITE_BYTE
BYDSK_PHYS_WRITE_BYTE_RATE
BYDSK_PHYS_WRITE_RATE
BYDSK_RAW_READ
BYDSK_RAW_READ_RATE
BYDSK_RAW_WRITE
BYDSK_RAW_WRITE_RATE
BYDSK_SYSTEM_IO
BYDSK_SYSTEM_IO_RATE
BYDSK_VM_IO
BYDSK_VM_IO_RATE
INTERVAL
OK
Defaults
Cancel
Help

Metrics & Tools
internal tools - tusc

- Trace Unix System Calls (tusc) - like truss on Solaris
- Traces all system calls made and signals received for a process
- Displays arguments in a symbolic way

Tusc [options] command [args] | pid [pid …]

Useful options:

-c	Summary of syscall counts, errors and CPU time
-ccc	CPU time for every syscall
-T %T	Print a timestamp before every trace
-rall	Display read buffer for all reads
-wall	Display write buffer for all writes
-f	Follow fork()s
-l	Print thread id
-p	Print pid
-n	Print process name

Metrics & Tools
internal tools - tusc

```
# tusc -o tusc.out 5263

open("/dev/telnet/", O_RDONLY,) ......... = 6
fcntl(6, F_SETFD, 1) .................... = 0
brk(0x400c0000) ......................... = 0
open("myfile", O_RDONLY, 02624).......... ERR#2 ENOENT
getdents(6, 0x4009f328, 8192)............ = 48
stat("/dev/", 0x6fff27f0)................ = 0
close(6) ................................ = 0
poll(0x6fff37c0, 1, 0) .................. = 1
setsockopt(13, SOL_SOCKET, SO_KEEPALIVE, 0xc000000023b49898,
4) = 0
recv(13, "sqAW8BPQAAsqlexe ", 4096, 0) .. = 371
ioctl(6, FIONBIO, 0xc00000023a915b8) ... = 0
```

Metrics & Tools
internal tools - tusc

```
# tusc -o tusc.out -c find /etc -name fred

Syscall                    Seconds      Calls     Errors

exit                          0.00          1
read                          0.06        242
write                         0.10        227
open                          0.00          9          2
close                         0.00          6
brk                           0.00          4
lseek                         0.00          7
execve                        0.00          1
umask                         0.00          2
mmap                          0.00         11          1
fstat                         0.00          4
sysconf                       0.00          2
stat64                        0.00          4          2
-----                        -----      -----      -----
Total                         0.17        534          6
```

Metrics & Tools
internal tools - tusc

```
# tusc -o tusc.out -ccc mycmd myargs

<0.000285> open("/dev/telnet/", O_RDONLY,) = 6
<0.000123> fcntl(6, F_SETFD, 1) .......... = 0
<0.000296> brk(0x400c0000) ............... = 0
<0.000199> open("myfile", O_RDONLY, 02624).. ERR#2 ENOENT
<0.000309> getdents(6, 0x4009f328, 8192).. = 48
<0.000223> stat("/dev/", 0x6fff27f0)...... = 0
<0.000266> close(6) ...................... = 0
<0.000218> poll(0x6fff37c0, 1, 0) ........ = 1
<0.000118> setsockopt(13, SOL_SOCKET, SO_KEEPALIVE,
0xc00000023b49898, 4) = 0
<0.000165> recv(13, "sqAW8BPQAAsqlexe ", 4096, 0) = 371
<0.000189> ioctl(6, FIONBIO, 0xc00000023a915b8) = 0
```

Metrics & Tools
internal tools - tusc

```
#tusc  -o <filename>  -ccc -f -l -n -p -v -T "%H:%M:%S" <pid>

10:17:25  connect(53, 0x7f0025f0, 16) ... ERR#245 EINPROGRESS
                   sin_family: AF_INET
                     sin_port: 3206
               sin_addr.s_addr: 172.18.187.8

10:17:25  poll(0x415532dc, 122, 5000)................... = 1

                 poll[52].fd:  52
             poll[52].events:  POLLOUT
            poll[52].revents:  POLLOUT

                 poll[53].fd:  53
             poll[53].events:  POLLIN|POLLPRI
            poll[53].revents:  0
```

Case Study

Pat Kilfoyle
Hewlett Packard

HP WORLD 2003
Solutions and Technology Conference & Expo

Case Study

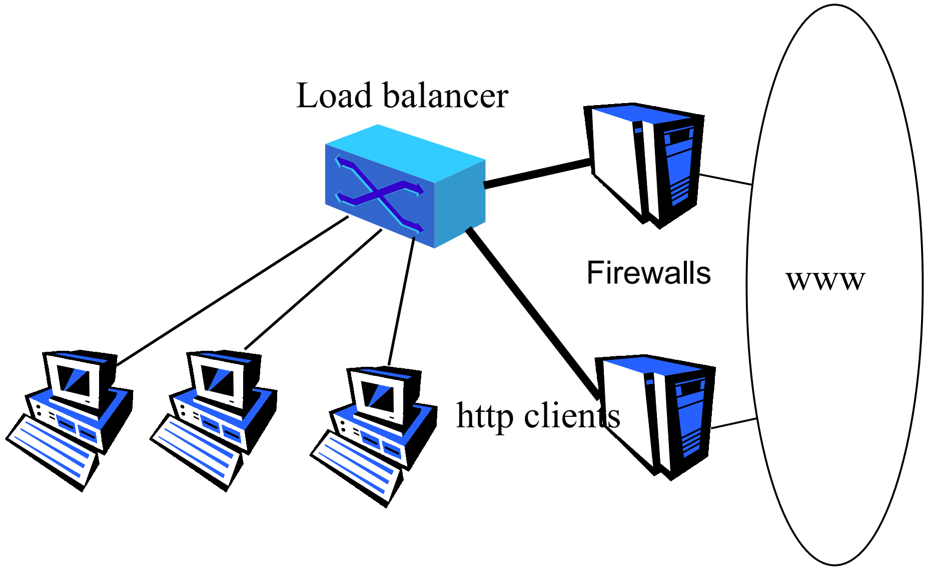
Problem -

- **Poor firewall performance – http traffic**
  - A multiprocess, multithreaded http daemon on a firewall was having slow connection handling stats according to an external network load balancer device.
  - Performance was compared with another HW vendor running the same revision of firewall product.

Case Study

Application details -

- http daemon had 10 processes with 8 kernel threads each.
- load balancer algorithm was ‘assign new connection to firewall with fewest active connections’
- New installation on 11.0



Case Study

Questions to be answered & tools to consider -

- *How do you find/ID an intermittent slow connection amongst 80 different threads spread among 10 processes?*
  - *http daemon logs that record time of transaction*
  - *A lot of network tracing and luck*
- *How do you measure/trace where a process threads spends its time?*
  - *glance process detail screens*
  - *nettl tracing at the IP layer to trace the network traffic*
  - *tusc syscall tracing …all threads traced at the same time.*
  - *kitrace syscall/kernel tool*
  - *kgmon tool to enable kernel profiling*
  - *Application logging with excruciating detail – wishful thinking.*

Case Study – tools

Tools used and the data they provided -

- http daemon logs showed which connections were delayed, but they seemed too few and infrequent to account for the overall slow performance.
  - Typically a failed DNS lookup was seen in the ***nettl*** IP layer traces.
  - The other vendors system would be subject to the same issue so this was ruled out as a root cause.
- ***tusc*** syscall trace, one tusc invocation for each process
  - Showed the thread interaction for each process
  - Searching for timestamp gaps in the syscall trace entries we were able to spot 'slow responses'.
  - recv() and ksleep() syscalls seemed to account for most of the thread delay time.
  - The tusc data showed an unexpected sequence of DNS lookups holding off other threads within the same process, calling kwakeup immediately after getting the DNS reply.
- sample threaded code was written to duplicate the DNS interaction outside of the http daemon…a simpler environment to debug.

Case Study
tusc – sample output

```
6.514689 [11972]{12562} <0.000046> socket(AF_INET, SOCK_DGRAM, 0) = 4

6.514860 [11972]{12562} <0.000033> connect(4, 0x400e0970, 16) = 0
                     sin_family: AF_INET
                       sin_port: 53
                sin_addr.s_addr: 201.155.160.51

6.515042 [11972]{12562} <0.000017> send(4, "\00201\0\001\0\0\0\0\0\0\ai p 2 ".., 25, 0) = 25

6.559376 [11972]{12562} <0.000020> select(5, 0x7f7918f0, NULL, NULL, 0x7f7918e8) = 1
                        readfds: 4
                       writefds: NULL
                       errorfds: NULL

6.559555 [11972]{12562} <0.000013> recv(4, "\0028183\001\0\0\001\0\0\ai p 2 "..,
1024, 0) = 100

6.559762 [11972]{12562} <0.000030> close(4) ..... = 0

6.563612 [11972]{12562} <0.000017> kwakeup(PTH_CONDVAR_OBJECT, 0x40001340,
WAKEUP_ONE, 0x7f790298) = 0             threads awakened: 1

6.563731 [11972]{12558} <0.000029> ksleep(PTH_CONDVAR_OBJECT, 0x40001340,
0x40001348, NULL) = 0
```

Case Study – resolution

Single threaded DNS code path found

- *The tusc output showed us an unexpected interaction among threads within the same process doing DNS queries.*
  - *It appeared to be a deliberately single threaded code path.*
  - *gdb debugger on the sample code showed us that the mutex lock was occurring in the DNS code within libnss_dns.1*
  - *Code review of the specific routines involved found old protection code in place from the days when the DNS resolver back end routines were not thread safe.*
  - *PHNE_27795 for 11.0 now contains the fix.*

Case Study

Key points - The tools and methodologies used are trying to answer the following:

- *Where is the thread/process spending it's time?*
  - *Kernel code active or sleeping?*
  - *User space active or sleeping?*
- *What is the process/thread doing?*
  - *What kernel code is it executing?*
  - *What user space code is it executing?*
- *Whatever it's doing, is it suppose to be doing it this way?*
  - *Between the application developers, the customer and HP, somebody had better know.*

Metrics & Tools
Looking inside the kernel - KItrace

- Presents the raw data behind Glance
- System-wide:
  - Syscall tracing
  - Event tracing through stubs
  - High resolution timestamps
  - Hardclock traces
- Example tracepoints for scheduling:
  - SETRQ, RESUME, SWTCH
- Example tracepoints for IO:
  - GETNEWBUF, ENQUEUE, SWTCH, QUEUESTART, QUEUEDONE, SETRQ, RESUME, BRELSE, read/write syscall

Metrics & Tools
internal tools - Kitrace

tracing tool that uses the same trace points as measureware

```
4.494196  cpu=1 pid=14031 ktid=753854 utid=0 ENQUEUE
  merged=0 dev_t=31/0x022000 q_len=1 MRG=0 wr=read len=8192
  bp=0x4fd653c0 blkno=0x7486f0 b_vp=0x0
  pid-u/a=14031/14031 ktid-u/a=753854/753854 utid-u/a=0/0
  b_flags=call/ndelay/busy/bcache/read/pftimeout/
  b2_flags=b2_null bptype=DATA/vxfs

4.494200  cpu=1 pid=14031 ktid=753854 utid=0 QUEUESTART
  merged=0 dev_t=31/0x022000 bp=0x4fd653c0 region=0x0
  len=8192 blkno=0x7486f0 sect=0xe90de0

4.504957  cpu=2 pid=-1 ktid=-1 utid=-1 QUEUEDONE
  dev_t=31/0x022000 retries=0 qs=0.010757
  MRG=0 wr=read len=8192 res=0 q_len=0 bp=0x4fd653c0
  lvdev=64/0x000004 b_site=0 blkno=0x7486f0 sect=0xe90de0
  pid-u/a=14031/14031 ktid-u/a=753854/753854 utid-u/a=0/0
```

Metrics & Tools
kparse

- K shell script that automates common Kitrace analyses
- Produces html report / has email capability
- Tracetypes
- Global switches
- Hardclock records
- Device service times
- Setrq records
- BRELSE blocks
- Device frequency
- Fname references
- Error codes reported

Metrics & Tools
internal tools - Kparse

Frequency of trace types...

Freq	Trace_type	Percent_of_total
202344	gettimeofday	28.0
59598	BRELSE	8.3
49687	select	6.9
46865	pstat	6.5
37841	SWTCH	5.2
37841	SETRQ	5.2
34693	RESUME	4.8
32530	sigprocmask	4.5
30506	kill	4.2
17158	time	2.4
16050	read	2.2
15786	times	2.2
15093	write	2.1

Metrics & Tools
internal tools - Kparse

Wait symbols...

Freq	Percent	Kernel_Symbol
16531	47.8%	real_nanosleep
7151	20.7%	read_sleep
4994	14.4%	semop
1467	4.2%	PIC3
1336	3.9%	lvmkd_daemon
898	2.6%	ksleep_one
831	2.4%	select
216	0.6%	ogetblk
182	0.5%	pm_sigwait
162	0.5%	sleep_spinunlock
160	0.5%	biowait
153	0.4%	poll

Metrics & Tools
internal tools - Kparse

```
Searching BRELSE records for hotblocks..

Freq    Block            bptype        operation
4167,   bp=0x1cabc3c00,  DATA/vxfs     wr=read
3189,   bp=0x15e843480,  INDBK/vxfs    wr=read
2967,   bp=0x1cabaeb00,  INDBK/vxfs    wr=write
 405 ,  bp=0x1567f7080,  DATA/vxfs     wr=write
 405,   bp=0x12d085500,  DATA/vxfs     wr=write
 327,   bp=0x206087580,  DATA/vxfs     wr=write
 327,   bp=0x1e6b0e100,  DATA/vxfs     wr=write
 327,   bp=0x15d59f780,  DATA/vxfs     wr=write
 222,   bp=0x1f1a67080,  DIR/vxfs      wr=read
 213,   bp=0x159896400,  DIR/vxfs      wr=read
```

Metrics & Tools
internal tools - Prospect

Prospect is a performance analysis tool based on based on KI tracing and Kernel Timing Clocks

System Summary

- System wide activity
- Per-CPU counters
- Per-Process summary

How each thread of each process spends its time

- Profiling, both user and system mode for every thread
- Instruction level profiling

Application profile requires no special compilation

- No source code access needed
- Very lightweight

Metrics & Tools
internal tools - Prospect

- Download from: http://www.hp.com/go/prospect

- To run, start daemon:

  `# prospect -P`

- Then execute command under prospect:

  `# prospect -V4 -e -f prospect.txt <command [args]>`

- Useful options:
- **-V2** Trace only the specified command
- **-V3** Trace command and descendants
- **-V4** Trace every process
- **-Vk** System-wide kernel profile
- **-e** Extended instruction-level tracing
- **-f** Specify output file

Metrics & Tools
internal tools - Prospect

```
System time
  select          0.22   32%
  gettimeofday    0.18   25%
  sigprocmask     0.09   13%
  write           0.05    8%
  ioctl           0.05    7%
  read            0.03    5%

User time
  main            4.41   69%   Dev=0x40000005,Inode=4078
  XTextExtents    0.41    6%   /usr/lib/X11R5/libX11.1
  memcmp          0.17    3%   /usr/lib/libc.1
  _isspace        0.13    2%   /usr/lib/libc.1
  MatchBranchHead 0.08    1%   /usr/lib/X11R5/libXt.1
  memmove         0.06    1%   /usr/lib/libc.1
```

Metrics & Tools
internal tools - Kgmon

- Activates kernel trace points
- Can provide flat or full profiles

%time	seconds	calls	name
24.6	133.99		IN_USER_MODE
15.4	83.96		prod_fullgprof_intercept
10.4	56.43		_mcount
8.2	44.89	3105	idle_nonpset_loop
6.2	33.70	35968151	soo_select
5.7	30.88	14384885	hpstreams_select_int2
2.5	13.72		asm_spinlock
2.4	13.19		spinlock_usav
1.9	10.48	126528	pollscan
1.9	10.41	14891228	mp_socket_lock
1.3	6.91		binit
1.2	6.27	14891199	sounlock

Metrics and Tools
Internal tools - p4 tools

- A set of tools that are compiled with the libp4 library
- Libp4 provides a quick way to write c programs that can access kernel structures
- Developed by GSE/WTEC organization
- Development focus is on dump analysis, we use on live systems

Metrics and Tools
p4 tools - kmeminfo

```
Physical memory usage summary (in page/byte/percent):

Physmem          =   262144    1.0g 100%  Physical memory
  Freemem        =    14572   56.9m   6%  Free physical memory
  Used           =   247572  967.1m  94%  Used physical memory
    System       =   208176  813.2m  79%  By kernel:
      text       =     2361    9.2m   1%   text
      data       =      418    1.6m   0%   data
      bss        =      348    1.4m   0%   bss
      Static     =    16874   65.9m   6%   for text/static data
      Dynamic    =    59617  232.9m  23%   for dynamic data
      Bufcache   =   131072  512.0m  50%   for buffer cache
      Eqmem      =       19   76.0k   0%   for equiv.mapped memory
      SCmem      =      594    2.3m   0%   for critical memory
    User         =    44381  173.4m  17%  By user processes:
      Uarea      =     2456    9.6m   1%   for thread uareas
    Disowned     =        8   32.0k   0%  Disowned pages
```

Metrics & Tools
p4 tools - shminfo

```
Global 32-bit shared quadrants:
===============================
        Space        Start          End  Kbytes Usage
Q4 0x0f1a0000.0xc0000000-0xc0008fff      36 OTHER
Q4 0x0f1a0000.0xc0009000-0xc0009fff       4 SHMEM id=0
Q4 0x0f1a0000.0xc000a000-0xc000bfff       8 SHMEM id=2
Q4 0x0f1a0000.0xc000c000-0xc00e0fff     852 OTHER
Q4 0x0f1a0000.0xc00e1000-0xc00f0fff      64 SHMEM id=1 locked
Q4 0x0f1a0000.0xc00f1000-0xc02c8fff    1888 OTHER
Q4 0x0f1a0000.0xc02c9000-0xc02cefff      24 SHMEM id=407
Q4 0x0f1a0000.0xc02cf000-0xc034efff     512 OTHER

Limits for 32-bit SHMEM allocation:
===================================

Maximum shmem segment:       65536 Kbytes (shmmax)
Largest free  segment:     1048576 Kbytes (Window id 0 (global)
quadrant 2)
Available swap        :     772492 Kbytes (swapspc_cnt)
Available pseudo-swap:       20616 Kbytes (swapmem_cnt)
```

Metrics & Tools
p4 tools - seminfo

```
        semmni      64      # of semaphore identifiers
        semmns     128      # of semaphores in system
        semmnu      30      # of undo structures in system
        Semmsl    2048      max # of semaphores per ID
        semopm     500      max # of operations per semop call
        semume      10      max # of undo entries per process
        semusz     104      size in bytes of undo structure
        semvmx   32767      semaphore maximum value
        semaem   16384      adjust on exit max value

Pending semaphore operations:

kthread at 0x4209f040 sleeping in semop():
        cmd   = "ntl_reader"
        proc  = 0x4209e040 (pid 480)
        wchan = 0xbbdae4 (sem 0xbbdae0, n-waiter)
        semid = 2
        semundo = 0xc3cd48
        struct sembuf ops[1] at 0xd4e1800.0x7f7f0620:
                op sem_num sem_op sem_flg
```

Metrics and Tools
internal tools – timer9

- Developed to debug Service Guard cmcld hangs
- We now use it as a trigger for any short-term hang issue
- Original design assigned a process to each cpu, then it would report delays
- Buddy system assigns a process to keep track of another process and report the delay
- Large amount of data collection options

Metrics and Tools
internal tools – timer9

```
-r make process realtime (-32 (strongest) to 127 (weakest))
-t make process timeshare (not realtime)
-m memory lock process
-s how many secs between process wakeups - default 0.5
-c if process has not run for this many secs then report – default 1
-l write to logfile instead of stdout/stderr
-v filename containing list of kernel variables
-a report all processes which have used at least percent of CPU
-k toggle kernel profiling every secs seconds
-b do "buddy" -k,-p,-a and -P handling on MP systems
-C cause box to crash (panic) when delay of >= secs occurs
-B make all other CPUs check cpunum for -b and -C options
-P run program after a delay has occcured
-p send a -S's signo to pid after delay occurs
-S signo to be sent to -p's PID (default SIGUSR1)
-g ensure that there is a minimum gap of this number of secs
   between signal/exec of -p/-P process and -D delay time
   profiling (default 5).
```

Is there a bottleneck ?

- This is the supply side of performance
- Easiest to look at – easiest to fix

- IO
  – Is there queuing on any drives?
  – Are there long service/wait times on any drives?
- CPU
  – Is there a significant load average?
  – Is system CPU high?
  – Are processes priority waited?
- Memory
  – Is there any paging or deactivations?
  – Is there significant swap utilization?

Is the work necessary ?

- Is the I/O demand efficient?
- Are the CPU cycles necessary?
- Is the application efficient?
- Is the memory utilization necessary?

Case Study

Jan Weaver
Hewlett Packard

HP WORLD 2003
Solutions and Technology Conference & Expo

Case Study

Problem:

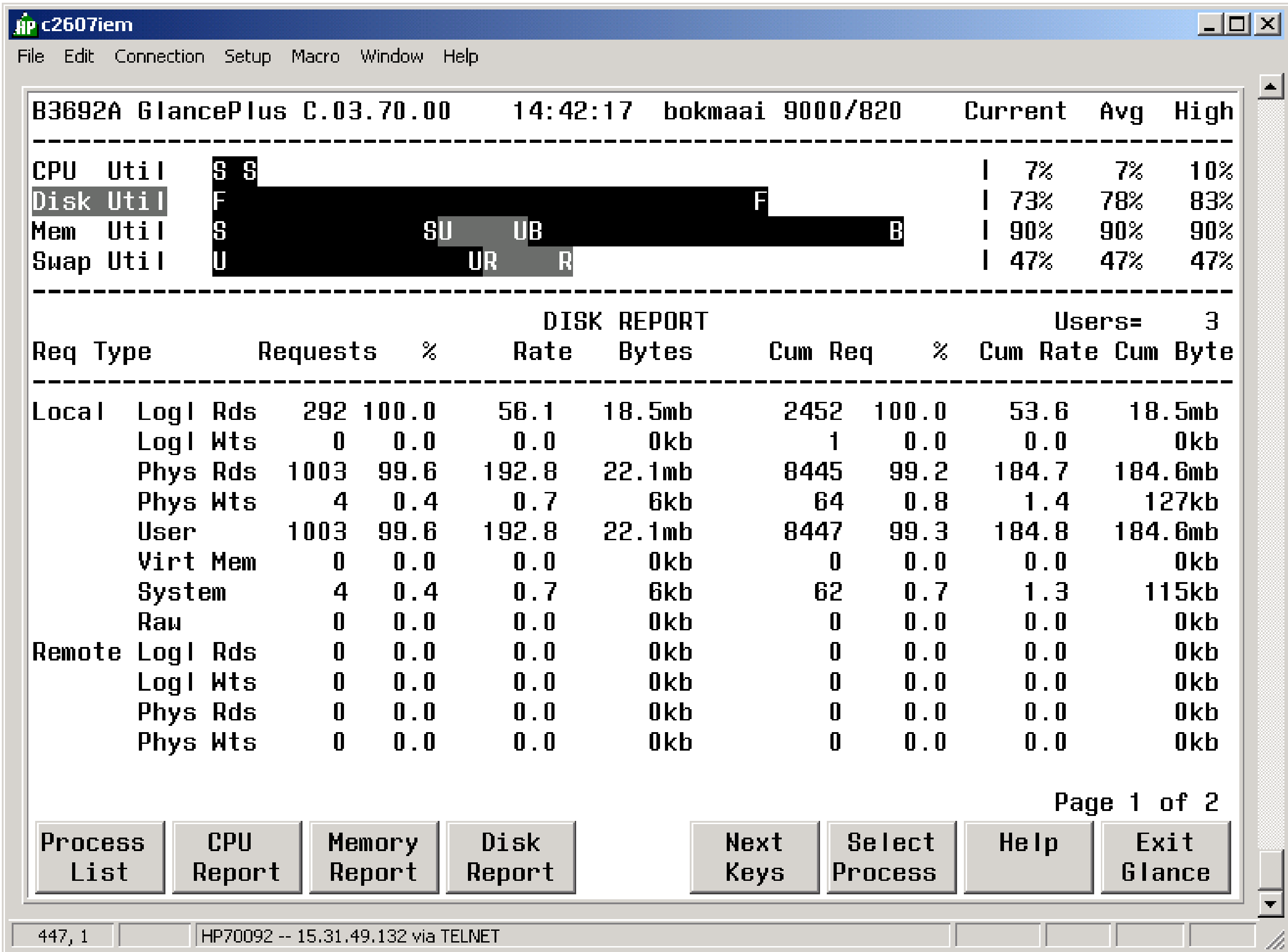
Customer upgraded from JFS 3.1 to JFS 3.3 or upgraded from HPUX 11.0 to HPUX 11i and he now has performance problems with his application and/or system.

He notices an increase in the disk activity.

Case Study

Glance shows a high level of physical disk activity and a low buffer cache hit rate.

Case Study


c2607iem

File Edit Connection Setup Macro Window Help

B3692A GlancePlus C.03.70.00 14:42:17 bokmaai 9000/820 Current Avg High

		Current	Avg	High
CPU Util	S S	7%	7%	10%
Disk Util	F F	73%	78%	83%
Mem Util	S SU UB B	90%	90%	90%
Swap Util	U UR R	47%	47%	47%

DISK REPORT Users= 3

Req Type		Requests	%	Rate	Bytes	Cum Req	%	Cum Rate	Cum Byte
Local	Logl Rds	292	100.0	56.1	18.5mb	2452	100.0	53.6	18.5mb
	Logl Wts	0	0.0	0.0	0kb	1	0.0	0.0	0kb
	Phys Rds	1003	99.6	192.8	22.1mb	8445	99.2	184.7	184.6mb
	Phys Wts	4	0.4	0.7	6kb	64	0.8	1.4	127kb
	User	1003	99.6	192.8	22.1mb	8447	99.3	184.8	184.6mb
	Virt Mem	0	0.0	0.0	0kb	0	0.0	0.0	0kb
	System	4	0.4	0.7	6kb	62	0.7	1.3	115kb
	Raw	0	0.0	0.0	0kb	0	0.0	0.0	0kb
Remote	Logl Rds	0	0.0	0.0	0kb	0	0.0	0.0	0kb
	Logl Wts	0	0.0	0.0	0kb	0	0.0	0.0	0kb
	Phys Rds	0	0.0	0.0	0kb	0	0.0	0.0	0kb
	Phys Wts	0	0.0	0.0	0kb	0	0.0	0.0	0kb

Page 1 of 2

Process List | CPU Report | Memory Report | Disk Report | Next Keys | Select Process | Help | Exit Glance

447, 1 | HP70092 -- 15.31.49.132 via TELNET

Case Study

c2607iem

File Edit Connection Setup Macro Window Help

```
B3692A GlancePlus C.03.70.00    06:32:00  bokmaai 9000/820    Current   Avg   High
-----------------------------------------------------------------------------------
CPU   Util   S  S                                              |   7%     5%    79%
Disk  Util   F                                   F             |  72%    76%    91%
Mem   Util   S              SU    UB                      B    |  90%    90%    90%
Swap  Util   U                 UR    R                         |  47%    47%    47%
-----------------------------------------------------------------------------------
                                    DISK REPORT                    Users=      3
Req Type            Requests      Rate       Cum Req    Cum Rate   High Rate
-----------------------------------------------------------------------------------
Read Cache Hits       2072        26.7        455822        46.9       100.0
Write Cache Hits         2        25.0           635        27.8
DNLC Hits                0         0.0             0         0.0         0.0
DNLC Longs               0         0.0             0         0.0         0.0

                                                                     Page 2 of 2
```

Process List	CPU Report	Memory Report	Disk Report		Next Keys	Select Process	Help	Exit Glance

88, 1 | HP70092 -- 15.31.49.132 via TELNET

Case Study

c2607iem

File Edit Connection Setup Macro Window Help

B3692A GlancePlus C.03.70.00 14:46:41 bokmaai 9000/820

		Current	Avg	High
CPU Util	S S	6%	6%	10%
Disk Util	F F	79%	78%	91%
Mem Util	S SU UB B	90%	90%	90%
Swap Util	U UR R	47%	47%	47%

IO BY DISK Users= 3

Idx	Device	Util	Qlen	KB/Sec	Logl IO	Phys IO
1	8/4.5.0	1/ 2	0.0	1.3/ 3.2	na/ na	0.6/ 1.3
2	8/4.11.0	0/ 0	0.0	0.0/ 0.0	na/ na	0.0/ 0.0
3	8/4.8.0	18/ 22	4.8	487.2/ 795.9	na/ na	32.7/ 41.8
4	8/4.9.0	79/ 78	10.4	2163.6/ 2481.1	na/ na	136.1/132.6
5	8/4.10.0	2/ 3	0.0	45.4/ 92.1	na/ na	2.2/ 4.9
6	8/4.0.0	0/ 0	0.0	0.0/ 0.1	na/ na	0.0/ 0.0
7	8/4.0.3	0/ 0	0.0	0.0/ 0.0	na/ na	0.0/ 0.0
8	8/16/5.2.0	0/ 0	0.0	0.0/ 0.0	0.0/ 0.0	0.0/ 0.0
9	8/8.1.0	0/ 0	0.0	0.0/ 0.0	0.0/ 0.0	0.0/ 0.0
10	8/8.2.0	0/ 0	0.0	0.0/ 0.0	0.0/ 0.0	0.0/ 0.0
11	8/8.3.0	0/ 0	0.0	0.0/ 0.0	0.0/ 0.0	0.0/ 0.0

Top disk user: PID 3665, pseudo 171.1 IOs/sec S - Select a Disk

Page 1 of 1

IO By File Sys	IO By Disk	IO By Logl Vol	Swap Space	Next Keys	CPU By Processr	Alarm History	Select

447, 1 | HP70092 -- 15.31.49.132 via TELNET

Case Study

We need to focus on the IO – who is doing it and why.

Kitrace can be used to look at the individual IO's and the system calls made by the process

In this case kitrace shows mostly random IO – lseek, read,lseek,read

However, occasionally we see sequential IO – lseek, read, read

Case Study

```
pid=3665 read ret1=8192
pid=3665 lseek ret1=365633536
pid=3665 read ret1=8192
pid=3665 lseek ret1=284893184
pid=3665 read ret1=8192
pid=3665 lseek ret1=466845696
pid=3665 read ret1=8192
pid=3665 lseek ret1=262332416
pid=3665 read ret1=8192
pid=3665 lseek ret1=118677504
pid=3665 read ret1=8192
pid=3665 read ret1=8192
pid=3665 lseek ret1=204439552
pid=3665 read ret1=8192
pid=3665 lseek ret1=229343232
```

Case Study

When the sequential reads occur, we see lots of physical IO being launched to the disks.

Case Study

```
pid=3665 lseek ret1=118677504
ENQUEUE pid=3665 wr=read len=8192
ENQUEUE pid=3665 wr=read len=8192
pid=3665 read ret1=8192
ENQUEUE pid=3665 wr=read len=8192
ENQUEUE pid=3665 wr=read len=24576
ENQUEUE pid=3665 wr=read len=16384
ENQUEUE pid=3665 wr=read len=8192
ENQUEUE pid=3665 wr=read len=32768
ENQUEUE pid=3665 wr=read len=8192
ENQUEUE pid=3665 wr=read len=32768
ENQUEUE pid=3665 wr=read len=8192
.
.
.
ENQUEUE pid=3665 wr=read len=24576
ENQUEUE pid=3665 wr=read len=65536
ENQUEUE pid=3665 wr=read len=40960
ENQUEUE pid=3665 wr=read len=8192
ENQUEUE pid=3665 wr=read len=8192
ENQUEUE pid=3665 wr=read len=57344
ENQUEUE pid=3665 wr=read len=32768
ENQUEUE pid=3665 wr=read len=8192
ENQUEUE pid=3665 wr=read len=8192
pid=3665 read ret1=8192
```

Case Study

System is doing read ahead when the sequential IO is detected.

Read ahead is more aggressive on JFS 3.3 than it was on JFS 3.1.

It is controlled by the vxtunefs parameters read_nstream and read_pref_io.

Case Study

```
>vxtunefs /data
Filesystem i/o parameters for /data
read_pref_io = 65536
read_nstream = 10
read_unit_io = 65536
write_pref_io = 65536
write_nstream = 1
write_unit_io = 65536
pref_strength = 10
buf_breakup_size = 131072
discovered_direct_iosz = 262144
max_direct_iosz = 655360
default_indir_size = 8192
qio_cache_enable = 0
max_diskq = 1048576
initial_extent_size = 4
max_seqio_extent_size = 2048
max_buf_data_size = 8192
```

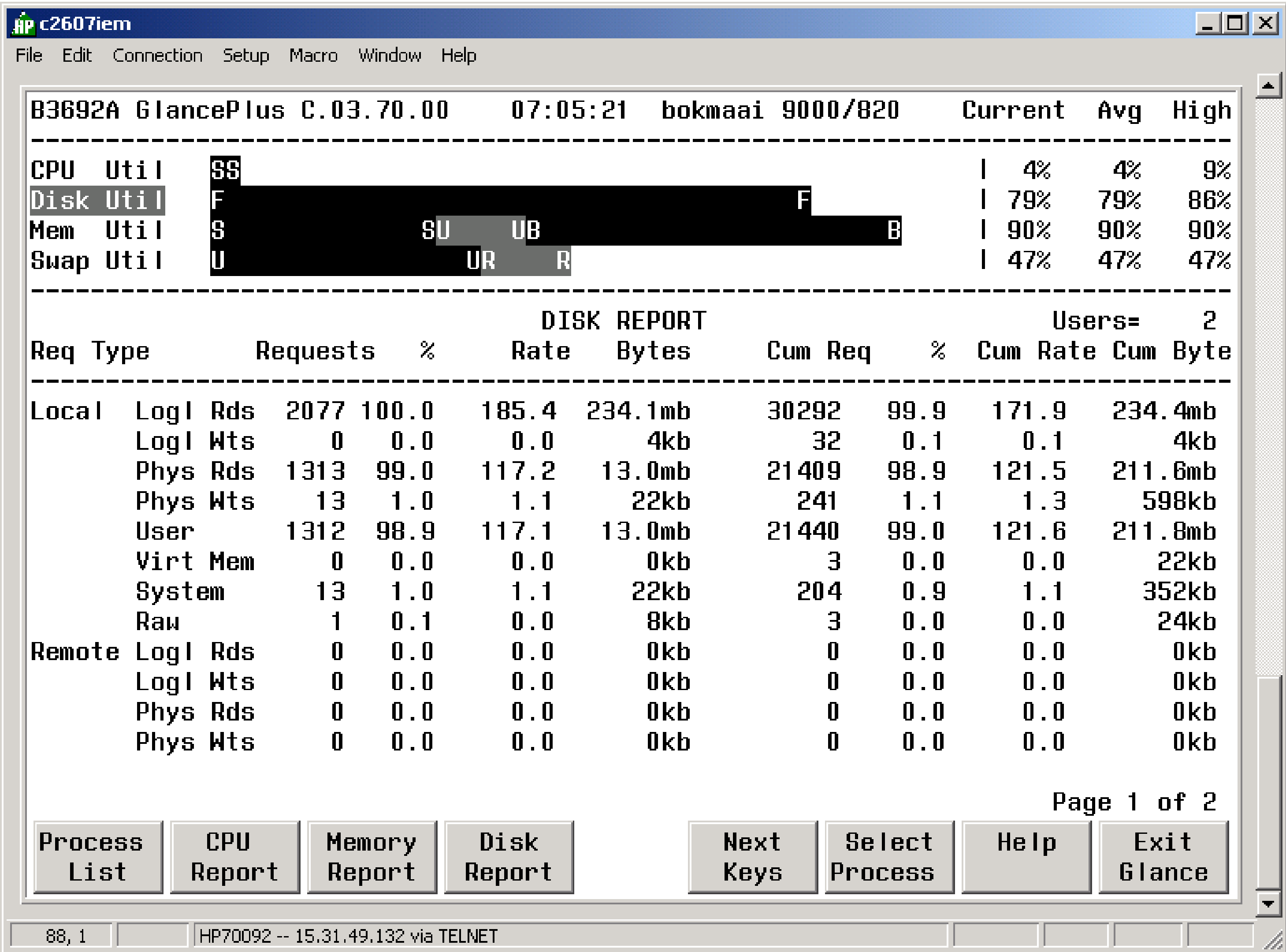
Case Study

Due to the generally random IO of the application the read ahead was unnecessary and in fact was likely harmful.

Filesystem parameters read_nstream and/or read_pref_io can be tuned to reduce the amount of read ahead that is performed.

Note that the application could also be changed to include code to advise the filesystem that the IO is random.

Case Study


c2607iem

File Edit Connection Setup Macro Window Help

B3692A GlancePlus C.03.70.00 07:05:21 bokmaai 9000/820

		Current	Avg	High
CPU Util	SS	4%	4%	9%
Disk Util	F F	79%	79%	86%
Mem Util	S SU UB B	90%	90%	90%
Swap Util	U UR R	47%	47%	47%

DISK REPORT Users= 2

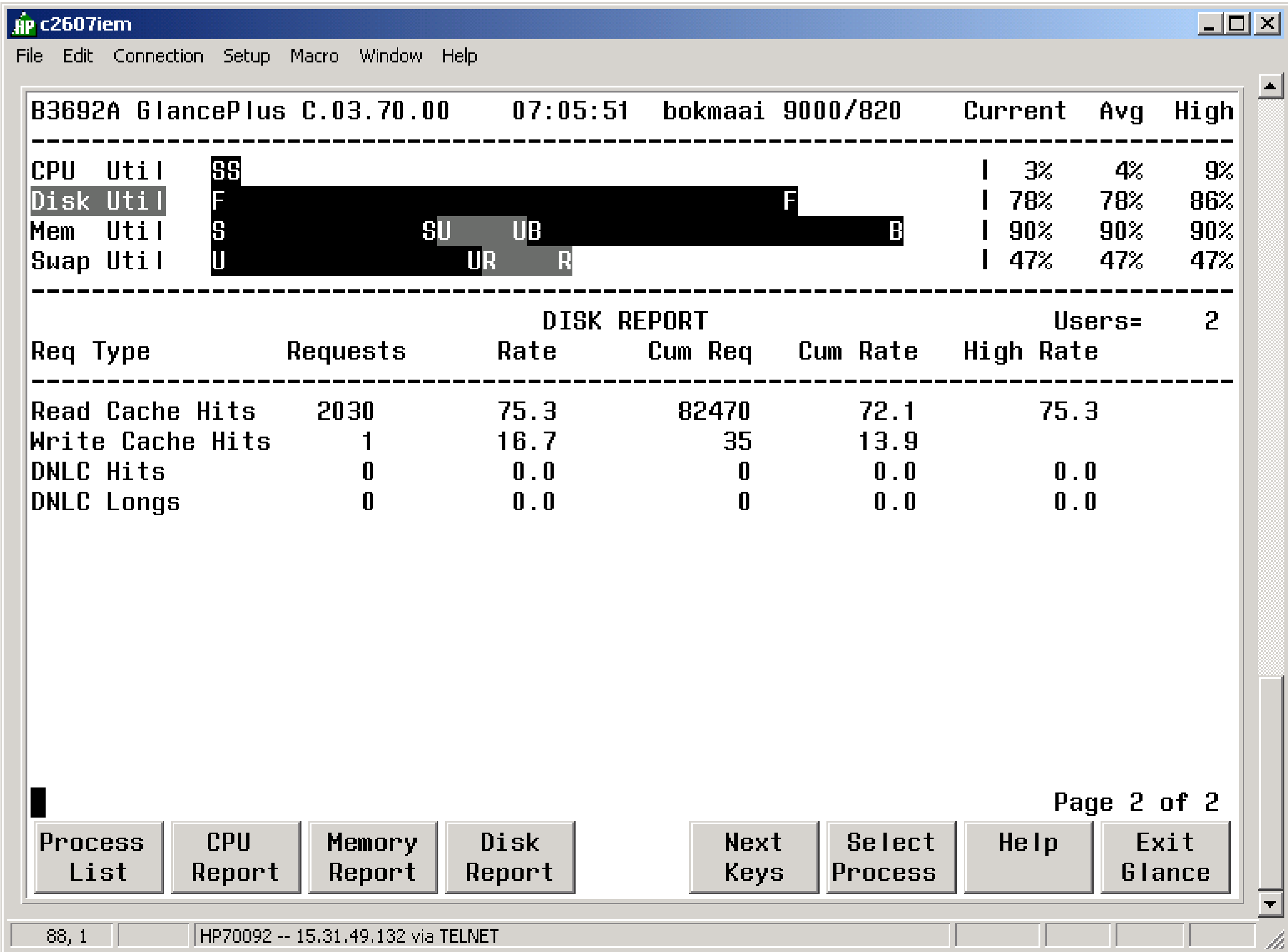
Req Type		Requests	%	Rate	Bytes	Cum Req	%	Cum Rate	Cum Byte
Local	Logl Rds	2077	100.0	185.4	234.1mb	30292	99.9	171.9	234.4mb
	Logl Wts	0	0.0	0.0	4kb	32	0.1	0.1	4kb
	Phys Rds	1313	99.0	117.2	13.0mb	21409	98.9	121.5	211.6mb
	Phys Wts	13	1.0	1.1	22kb	241	1.1	1.3	598kb
	User	1312	98.9	117.1	13.0mb	21440	99.0	121.6	211.8mb
	Virt Mem	0	0.0	0.0	0kb	3	0.0	0.0	22kb
	System	13	1.0	1.1	22kb	204	0.9	1.1	352kb
	Raw	1	0.1	0.0	8kb	3	0.0	0.0	24kb
Remote	Logl Rds	0	0.0	0.0	0kb	0	0.0	0.0	0kb
	Logl Wts	0	0.0	0.0	0kb	0	0.0	0.0	0kb
	Phys Rds	0	0.0	0.0	0kb	0	0.0	0.0	0kb
	Phys Wts	0	0.0	0.0	0kb	0	0.0	0.0	0kb

Page 1 of 2

Process List | CPU Report | Memory Report | Disk Report | Next Keys | Select Process | Help | Exit Glance

88, 1 | HP70092 -- 15.31.49.132 via TELNET

Case Study

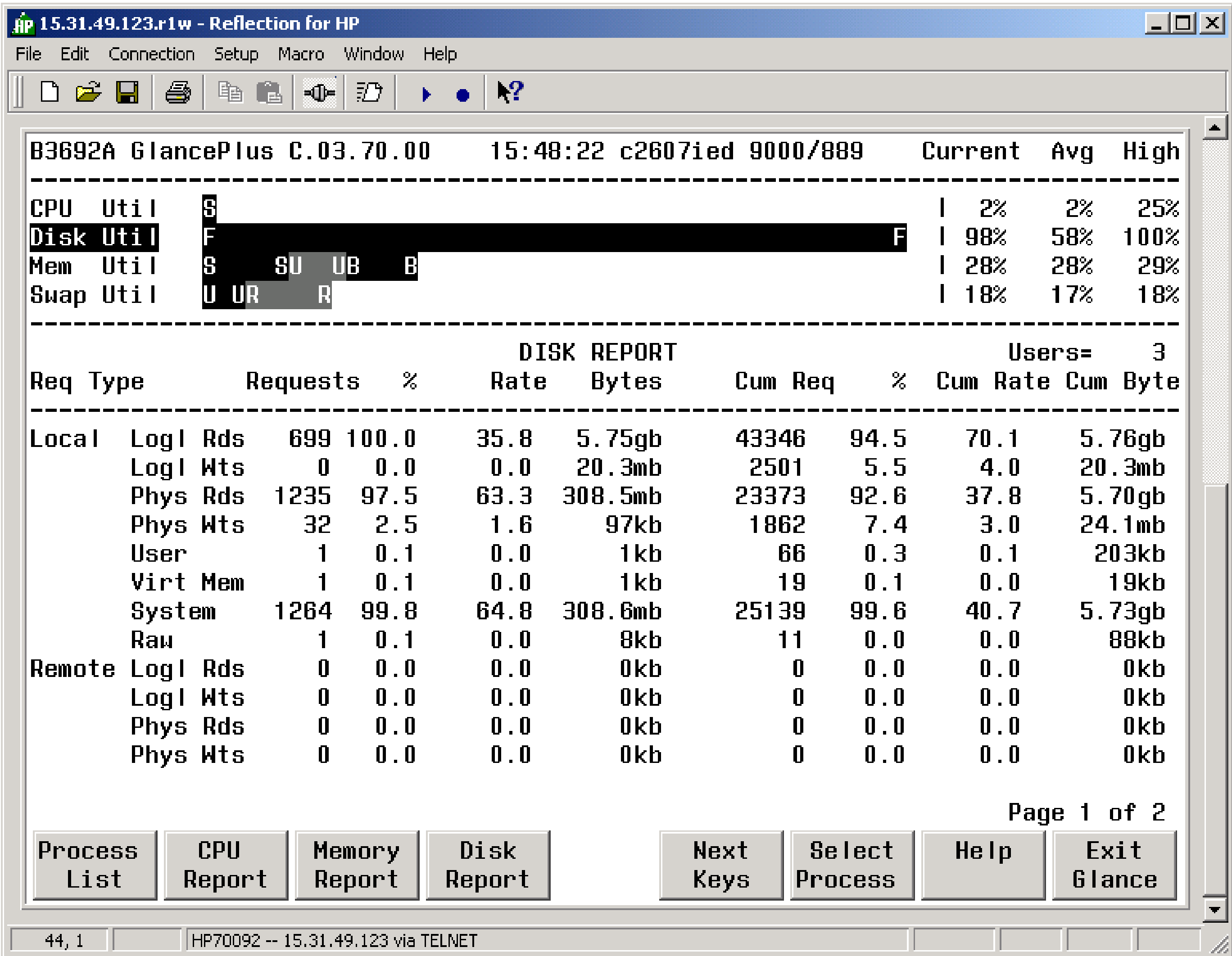

c2607iem
File Edit Connection Setup Macro Window Help
B3692A GlancePlus C.03.70.00 07:05:51 bokmaai 9000/820 Current Avg High
CPU Util SS | 3% 4% 9%
Disk Util F F | 78% 78% 86%
Mem Util S SU UB B | 90% 90% 90%
Swap Util U UR R | 47% 47% 47%
DISK REPORT Users= 2
Req Type Requests Rate Cum Req Cum Rate High Rate
Read Cache Hits 2030 75.3 82470 72.1 75.3
Write Cache Hits 1 16.7 35 13.9
DNLC Hits 0 0.0 0 0.0 0.0
DNLC Longs 0 0.0 0 0.0 0.0
Page 2 of 2
Process List
CPU Report
Memory Report
Disk Report
Next Keys
Select Process
Help
Exit Glance
88, 1
HP70092 -- 15.31.49.132 via TELNET

Case Study

Customer sees a similar performance slowdown after adding online JFS.

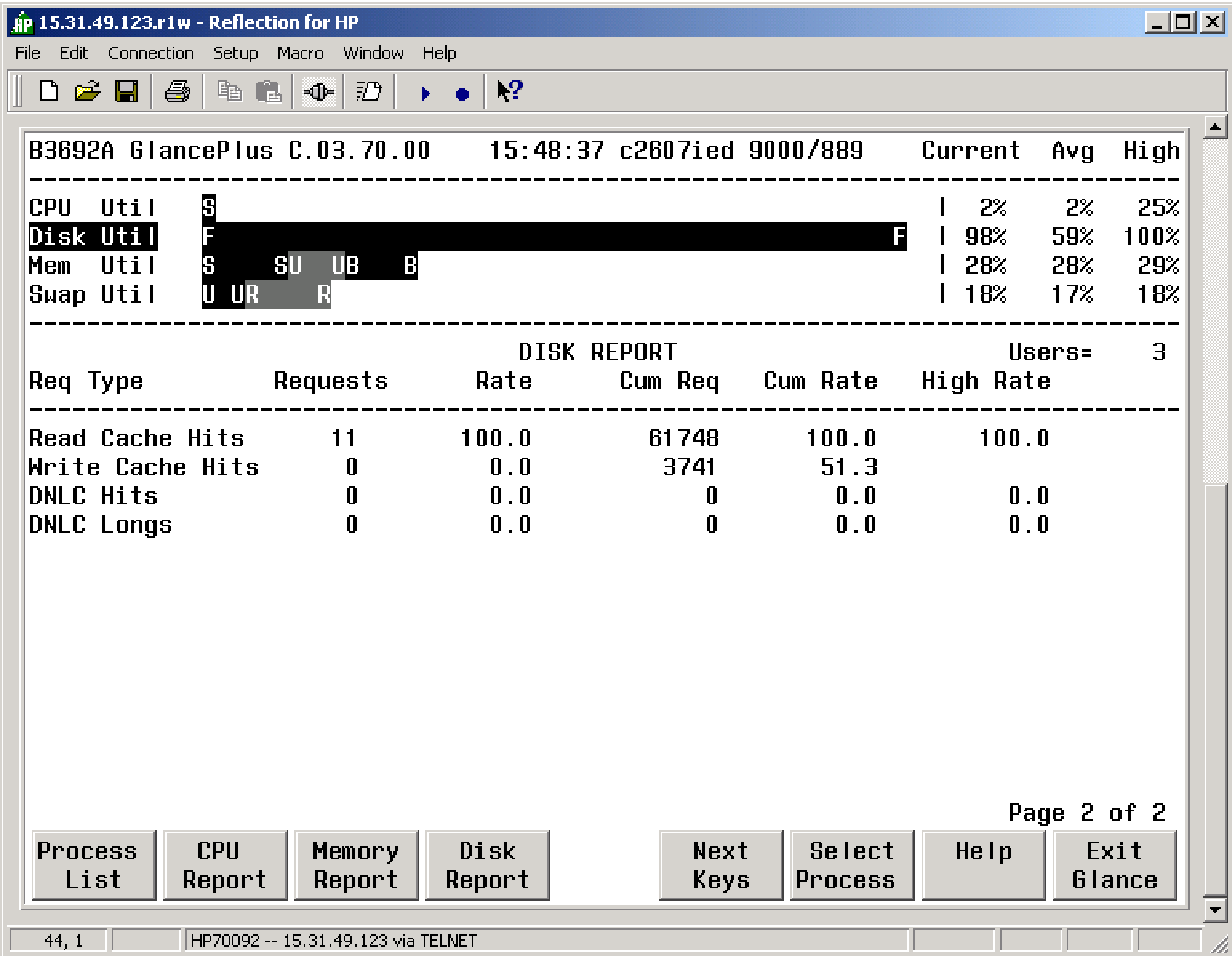
Applications run slower and there is more physical IO than seen previously.

Case Study

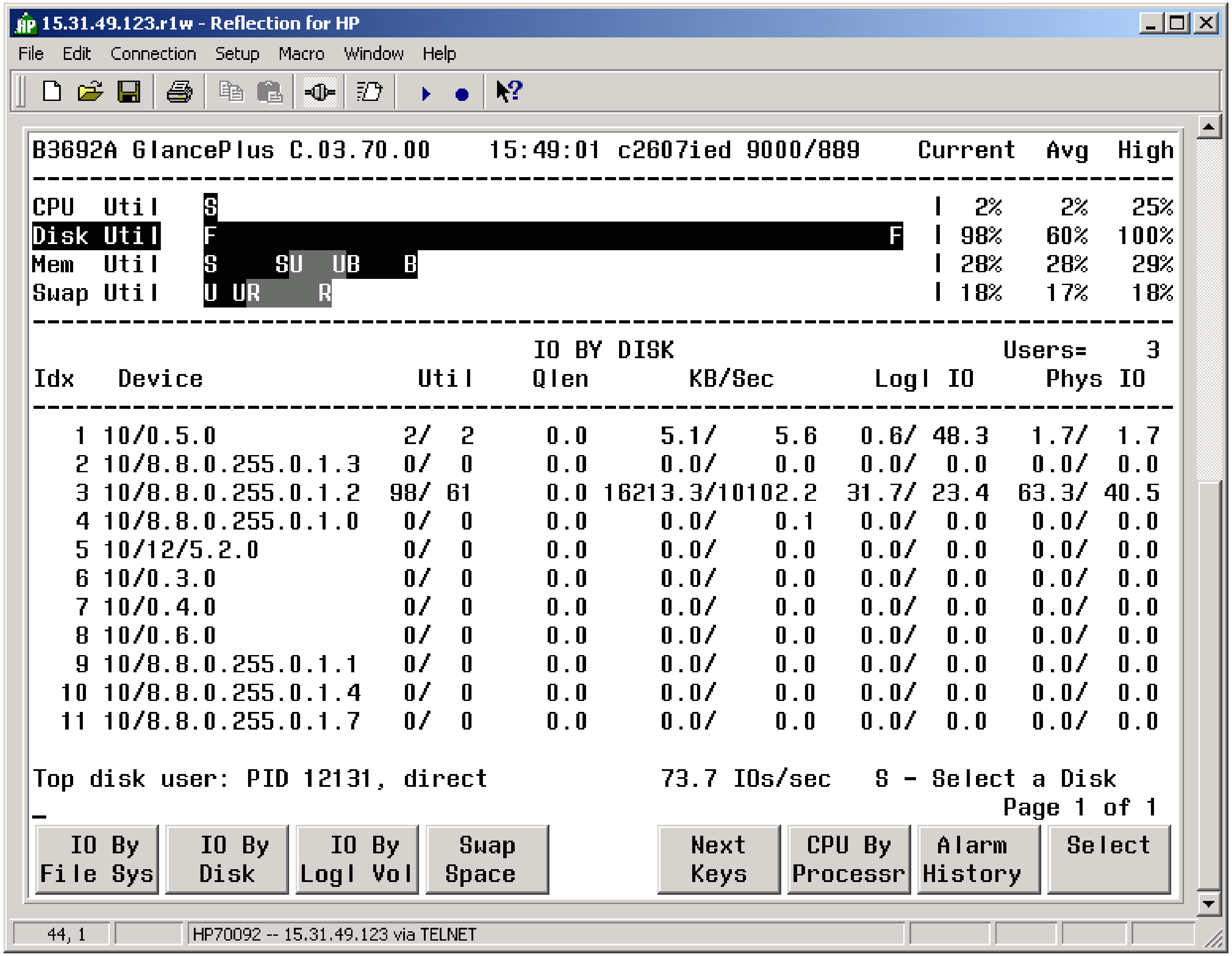


```
15.31.49.123.r1w - Reflection for HP
File  Edit  Connection  Setup  Macro  Window  Help

B3692A GlancePlus C.03.70.00    15:48:22 c2607ied 9000/889    Current  Avg  High
--------------------------------------------------------------------------------
CPU  Util   S                                                 |  2%     2%   25%
Disk Util   F                                               F | 98%    58%  100%
Mem  Util   S    SU  UB   B                                   | 28%    28%   29%
Swap Util   U UR     R                                        | 18%    17%   18%
--------------------------------------------------------------------------------
                                  DISK REPORT                       Users=     3
Req Type          Requests   %     Rate   Bytes      Cum Req    %  Cum Rate Cum Byte
--------------------------------------------------------------------------------
Local  Logl Rds      699 100.0     35.8   5.75gb      43346   94.5     70.1   5.76gb
       Logl Wts        0   0.0      0.0   20.3mb       2501    5.5      4.0   20.3mb
       Phys Rds     1235  97.5     63.3  308.5mb      23373   92.6     37.8   5.70gb
       Phys Wts       32   2.5      1.6     97kb       1862    7.4      3.0   24.1mb
       User            1   0.1      0.0      1kb         66    0.3      0.1    203kb
       Virt Mem        1   0.1      0.0      1kb         19    0.1      0.0     19kb
       System       1264  99.8     64.8  308.6mb      25139   99.6     40.7   5.73gb
       Raw             1   0.1      0.0      8kb         11    0.0      0.0     88kb
Remote Logl Rds        0   0.0      0.0      0kb          0    0.0      0.0      0kb
       Logl Wts        0   0.0      0.0      0kb          0    0.0      0.0      0kb
       Phys Rds        0   0.0      0.0      0kb          0    0.0      0.0      0kb
       Phys Wts        0   0.0      0.0      0kb          0    0.0      0.0      0kb

                                                                    Page 1 of 2
Process   CPU      Memory   Disk               Next    Select   Help    Exit
 List     Report   Report   Report             Keys    Process          Glance

44, 1        HP70092 -- 15.31.49.123 via TELNET
```

Case Study


15.31.49.123.r1w - Reflection for HP
File Edit Connection Setup Macro Window Help
B3692A GlancePlus C.03.70.00 15:48:37 c2607ied 9000/889 Current Avg High
CPU Util S | 2% 2% 25%
Disk Util F F | 98% 59% 100%
Mem Util S SU UB B | 28% 28% 29%
Swap Util U UR R | 18% 17% 18%
DISK REPORT Users= 3
Req Type Requests Rate Cum Req Cum Rate High Rate
Read Cache Hits 11 100.0 61748 100.0 100.0
Write Cache Hits 0 0.0 3741 51.3
DNLC Hits 0 0.0 0 0.0 0.0
DNLC Longs 0 0.0 0 0.0 0.0
Page 2 of 2
Process List
CPU Report
Memory Report
Disk Report
Next Keys
Select Process
Help
Exit Glance
44, 1
HP70092 -- 15.31.49.123 via TELNET

Case Study


15.31.49.123.r1w - Reflection for HP
File Edit Connection Setup Macro Window Help
B3692A GlancePlus C.03.70.00 15:49:01 c2607ied 9000/889 Current Avg High
CPU Util S | 2% 2% 25%
Disk Util F F | 98% 60% 100%
Mem Util S SU UB B | 28% 28% 29%
Swap Util U UR R | 18% 17% 18%
IO BY DISK Users= 3
Idx Device Util Qlen KB/Sec Logl IO Phys IO
1 10/0.5.0 2/ 2 0.0 5.1/ 5.6 0.6/ 48.3 1.7/ 1.7
2 10/8.8.0.255.0.1.3 0/ 0 0.0 0.0/ 0.0 0.0/ 0.0 0.0/ 0.0
3 10/8.8.0.255.0.1.2 98/ 61 0.0 16213.3/10102.2 31.7/ 23.4 63.3/ 40.5
4 10/8.8.0.255.0.1.0 0/ 0 0.0 0.0/ 0.1 0.0/ 0.0 0.0/ 0.0
5 10/12/5.2.0 0/ 0 0.0 0.0/ 0.0 0.0/ 0.0 0.0/ 0.0
6 10/0.3.0 0/ 0 0.0 0.0/ 0.0 0.0/ 0.0 0.0/ 0.0
7 10/0.4.0 0/ 0 0.0 0.0/ 0.0 0.0/ 0.0 0.0/ 0.0
8 10/0.6.0 0/ 0 0.0 0.0/ 0.0 0.0/ 0.0 0.0/ 0.0
9 10/8.8.0.255.0.1.1 0/ 0 0.0 0.0/ 0.0 0.0/ 0.0 0.0/ 0.0
10 10/8.8.0.255.0.1.4 0/ 0 0.0 0.0/ 0.0 0.0/ 0.0 0.0/ 0.0
11 10/8.8.0.255.0.1.7 0/ 0 0.0 0.0/ 0.0 0.0/ 0.0 0.0/ 0.0
Top disk user: PID 12131, direct 73.7 IOs/sec S - Select a Disk
Page 1 of 1
IO By File Sys
IO By Disk
IO By Logl Vol
Swap Space
Next Keys
CPU By Processr
Alarm History
Select
44, 1
HP70092 -- 15.31.49.123 via TELNET

Case Study

Again we can use KItrace to see the characteristics of the IO.

Kparse will take the KItrace output and extract such things as disk service times, queue lengths and disk block frequency.

Case Study

From the Kparse report:

```
Disk block frequency...
Freq    Dev                     Block
597     dev_t=31/0x031200       blkno=0xb37340  wr=read
597     dev_t=31/0x031200       blkno=0xb37240  wr=read
4       dev_t=31/0x031200       blkno=0x538     wr=write
2       dev_t=31/0x031200       blkno=0xbee378  wr=write
2       dev_t=31/0x031200       blkno=0xb4f2ec  wr=write
2       dev_t=31/0x031200       blkno=0xb4f2cc  wr=write
2       dev_t=31/0x025000       blkno=0x3fb2b4  wr=write
2       dev_t=31/0x025000       blkno=0x30badc  wr=write
```

Case Study

We see the same physical blocks being read from the disk multiple times during the short (20 second) data collection.

Why are these blocks being continuously read from the disk when the file system should be using the buffer cache and therefore the block should be available in the buffer cache?

Case Study

If we look at a particular pid doing IO we can see what the IO looks like:

```
 pid=12131 ktid=13338 lseek err=0 ret1=0
ENQUEUE dev_t=31/0x031200  pid-u/a=12131/12131 wr=read blkno=0xb37240
b_flags=call/ndelay/busy/read/pftimeout/phys/
ENQUEUE dev_t=31/0x031200  pid-u/a=12131/12131 wr=read blkno=0xb37340
b_flags=call/ndelay/busy/read/pftimeout/phys/
  pid=12131 ktid=13338 read err=0 ret1=524288
  pid=12131 ktid=13338 lseek err=0 ret1=0
ENQUEUE dev_t=31/0x031200  pid-u/a=12131/12131 wr=read blkno=0xb37240
b_flags=call/ndelay/busy/read/pftimeout/phys/
ENQUEUE dev_t=31/0x031200  pid-u/a=12131/12131 wr=read blkno=0xb37340
b_flags=call/ndelay/busy/read/pftimeout/phys/
  pid=12131 ktid=13338 read err=0 ret1=524288
  pid=12131 ktid=13338 lseek err=0 ret1=0
ENQUEUE dev_t=31/0x031200  pid-u/a=12131/12131 wr=read blkno=0xb37240
b_flags=call/ndelay/busy/read/pftimeout/phys/
ENQUEUE dev_t=31/0x031200  pid-u/a=12131/12131 wr=read blkno=0xb37340
b_flags=call/ndelay/busy/read/pftimeout/phys/
  pid=12131 ktid=13338 read err=0 ret1=524288
```

Case Study

Here we see the same blocks being read repeatedly by the application (lseek to position 0, read), the reads rather large (524288 bytes) and the IO bypassing the buffer cache (b_flags=phys)

```
 pid=12131 ktid=13338 lseek err=0 ret1=0
ENQUEUE dev_t=31/0x031200  pid-u/a=12131/12131 wr=read blkno=0xb37240
b_flags=call/ndelay/busy/read/pftimeout/phys/
ENQUEUE dev_t=31/0x031200  pid-u/a=12131/12131 wr=read blkno=0xb37340
b_flags=call/ndelay/busy/read/pftimeout/phys/
  pid=12131 ktid=13338 read err=0 ret1=524288
```

Case Study

This is the discovered_direct_io feature of Online JFS.

Large reads typically are done once (backups or copies) and do not need to be kept in the buffer cache.

However, in this case the reads were repeated. The discovered_direct_io parameter should be tuned for this application.

Case Study

```
# vxtunefs /home/jan
Filesystem i/o parameters for /home/jan
read_pref_io = 65536
read_nstream = 1
read_unit_io = 65536
write_pref_io = 65536
write_nstream = 1
write_unit_io = 65536
pref_strength = 10
buf_breakup_size = 262144
discovered_direct_iosz = 262144
max_direct_iosz = 1048576
default_indir_size = 8192
qio_cache_enable = 0
max_diskq = 1048576
initial_extent_size = 2
max_seqio_extent_size = 2048
max_buf_data_size = 8192
```

Case Study

```
15.31.49.123.r1w - Reflection for HP
File  Edit  Connection  Setup  Macro  Window  Help

B3692A GlancePlus C.03.70.00    16:38:19 c2607ied 9000/889    Current  Avg  High
--------------------------------------------------------------------------------
CPU  Util   S            SU                                  | 27%    20%   27%
Disk Util   FF                                               |  3%     3%    3%
Mem  Util   S      SU   UB    B                              | 30%    30%   30%
Swap Util   U  UR       R                                    | 19%    19%   19%
--------------------------------------------------------------------------------
                               DISK REPORT                        Users=     3
Req Type          Requests  %     Rate   Bytes     Cum Req    %  Cum Rate Cum Byte
--------------------------------------------------------------------------------
Local  Logl Rds   1968 100.0   378.4   3.43gb      7177  100.0   273.9    3.43gb
       Logl Wts      0   0.0     0.0      0kb         1    0.0     0.0       0kb
       Phys Rds      1   7.7     0.1      8kb        43   41.7     1.6     2.0mb
       Phys Wts     12  92.3     2.3     39kb        60   58.3     2.2     209kb
       User          0   0.0     0.0      0kb         4    3.9     0.1       9kb
       Virt Mem      1   7.7     0.1      1kb         1    1.0     0.0       1kb
       System       11  84.6     2.1     38kb        97   94.2     3.7     2.2mb
       Raw           1   7.7     0.1      8kb         1    1.0     0.0       8kb
Remote Logl Rds      0   0.0     0.0      0kb         0    0.0     0.0       0kb
       Logl Wts      0   0.0     0.0      0kb         0    0.0     0.0       0kb
       Phys Rds      0   0.0     0.0      0kb         0    0.0     0.0       0kb
       Phys Wts      0   0.0     0.0      0kb         0    0.0     0.0       0kb

                                                                   Page 1 of 2
Process   CPU     Memory   Disk               Next   Select    Help    Exit
 List    Report   Report   Report             Keys   Process          Glance

500, 1    HP70092 -- 15.31.49.123 via TELNET
```

Are the CPU cycles necessary ?

System: T600 8-way 10.20
2 GB memory
Development system
Compiling and source code management

Symptoms: 3 year installation
Recently seeing slow overall performance
Intermittent
High system CPU and high context switch rates

Diagnosis: Files used for compiling were located in one directory

Large number of files and very volatile

Contention around the directory file itself (25 MB)

Spinning while waiting for shared resource caused unnecessary context switching

Is the application efficient ?

System: N4000 4-way 11.0
4 GB memory
Web server

Symptoms: New installation
Server throughput was never acceptable
High CPU utilization with mostly user CPU
Load average was reasonable and good system response time

Diagnosis: Identified large # of semop calls in bolt-on application

Application was in the critical path for the server

Allowed vendor to identify configuration problem

Is the memory utilization necessary ?

System:	V2500 16-way 11.0 4 GB memory Database server
Symptoms:	Memory utilization at 100% High page out and deactivation rates
Diagnosis:	Default 50% buffer cache had been used maxuser had been set very high – affects many other kernel variables
	Final solution was to add memory and to tune kernel variables

Looking for anomalies

- System call rates/CPU utilization
- I/O patterns
  - By device
  - By time of day
  - By process
- Wait states
  - Global and per process

Anomalies
system call rates / CPU util

System: K460 4-way 10.20
2 GB memory
Legacy shell script-based application
Files ftp'd in, processed, then put in a directory for pickup

Symptoms: Suddenly application throughput was down
No changes to the application
System CPU way up

```
B3690A GlancePlus C.02.40.00     06:26:36 P1000147 9000/785     Current  Avg   High
-------------------------------------------------------------------------------
CPU  Util   S                                                  |  2%     2%    14%
Disk Util                                                      |  0%     0%    10%
Mem  Util   S   SU                UB  B                        | 50%    49%    50%
Swap Util   UUR          R                                     | 20%    20%    20%
-------------------------------------------------------------------------------
                              GLOBAL SYSTEM CALLS                  Users=    1
System Call Name          ID     Count       Rate    CPU Time   Cum CPU
-------------------------------------------------------------------------------
exit                       1         0        0.0     0.00000   0.03828
fork                       2         0        0.0     0.00000   0.02793
read                       3       392       87.1     0.00144   0.13783
write                      4       119       26.4     0.00103   0.08626
open                       5         4        0.8     0.00018   0.03305
close                      6         4        0.8     0.00012   0.00746
wait                       7         0        0.0     0.00000   0.00009
unlink                    10         0        0.0     0.00000   0.00105
chdir                     12         0        0.0     0.00000   0.00006
time                      13       199       44.2     0.00012   0.00180
brk                       17         0        0.0     0.00000   0.00162

        Cumulative Interval:        50 secs
                                                                  Page 1 of 9
-------------------------------------------------------------------------------
  Global    Global    DCE      System     68    1    Next    Netwk By   NFS      NFS By
   Waits   Syscalls  Global    Tables                Keys    Intrface  Global   System
```

Anomalies
system call rates / CPU util

System:	K460 4-way 10.20 2 GB memory Legacy shell script-based application Files ftp'd in, processed, then put in a directory for pickup
Symptoms:	Suddenly application throughput was down No changes to the application System CPU way up
Diagnosis:	vfork() was very large CPU consumer Identified shell script that was in a loop

Anomalies
IO patterns by device, time, process

System:	N4000 4-way 11.0 4 GB memory Database server for web front-end
Symptoms:	New installation System response was good Unacceptable database performance DB connections were short-lived Analysis showed that delay was in DB disconnect

```
B3690A GlancePlus C.02.40.00     06:35:44 F1000147 9000/785     Current   Avg   High
--------------------------------------------------------------------------------
CPU  Util   SUL                                             |   5%     2%    14%
Disk Util                                                   |   0%     0%    22%
Mem  Util   S  SU                UB  B                      |  50%    50%    51%
Swap Util   UUR       R                                     |  20%    20%    20%
--------------------------------------------------------------------------------
Open Files PID: 21113, netscape       PPID: 21112 euid:   101 User: kenj
                                                        Open  Open
FD  File Name                                      Type   Mode  Count       Offset
--------------------------------------------------------------------------------
 12 <reg,vxfs,/home,/dev/vg00/lvol4,inode:80>      reg    rd/wr   1         131072
 13 <reg,vxfs,/home,/dev/vg00/lvol4,inode:81>      reg    rd/wr   1          16384
 14 <reg,vxfs,/home,/dev/vg00/lvol4,inode:93>      reg    rd/wr   1            260
 15 <reg,vxfs,/home,/dev/vg00/lvol4,inode:83>      reg    rd/wr   1            260
 16 <fifo,pipe,inode:0>                            fifo   read    1              0
 17 <fifo,pipe,inode:0>                            fifo   write   3              0
 18 /dev/null                                      chr    write  22           1250
 19 /dev/null                                      chr    write  22           1250
 20 <reg,vxfs,/home,/dev/vg00/lvol4,inode:136>     reg    rd/wr   1            194
 21 <socket: inet,tcp,0x009f5e00>                  socket rd/wr   1          16878
 22 <socket: inet,tcp,0x02387400>                  socket rd/wr   1          16043
 23 <socket: inet,tcp,0x009d0800>                  socket rd/wr   1          25478
                                                                     Page 2 of 3
--------------------------------------------------------------------------------
Process   Wait    Memory   Open     68    1   Next   Process
Resource  States  Regions  Files                Keys   Syscalls
```

Anomalies
IO patterns by device, time, process

System:	N4000 4-way 11.0 4 GB memory Database server for web front-end
Symptoms:	New installation System response was good Unacceptable database performance DB connections were short-lived Analysis showed delay was in DB disconnect
Diagnosis:	Used Glance to observe when user disconnected Found high rates of IO during disconnect IO was to 2 database trace files

Anomalies
IO patterns by device, time, process

System:	V2600 32-way 16 GB memory Database server
Symptoms:	Application queues building up intermittantly System response was good sar showed average service time was ok
Diagnosis:	Used kitrace to determine there were short bursts of IO causing the EMC a problem

Anomalies
IO patterns by device, time, process

```
IO completion

timestamp   device    service time
                       (ms)

0.330263    40500     11.154
0.479900    40500     11.167
0.516062    40500     11.557   <<< burst starts at .504507
0.543143    40500     34.816
0.566754    40500     53.129
0.588817    40500     73.247
0.605655    40500     85.149
0.617827    40500     94.898
0.628228    40500    104.251
0.634265    40500    110.166
0.680130    40500     16.511
0.803631    40500     13.441
```

Anomalies
IO patterns by device, time, process

```
IO initiation


timestamp device     ms since
                 last IO start

0.319111   40500
0.468733   40500          149
0.504507   40500           35
0.508329   40500            3
0.513627   40500            5
0.515572   40500            1
0.520509   40500            4
0.522931   40500            2
0.523979   40500            1
0.524101   40500            0
0.663621   40500          139
0.790192   40500          126
1.757730   40500          967
```

Anomalies
IO patterns by device, time, process

```
timestamp   interval     type    service_time

0.504507        25       start                  *** first burst ***
0.508329         4       start
0.513627         5       start
0.515572         2       start
0.516062         0       complete        12
0.520509         4       start
0.522931         2       start
0.523979         1       start
0.524101         0       start
0.543143        19       complete        35
0.566754        24       complete        53
0.588817        22       complete        73
0.605655        17       complete        85
0.617827        12       complete        95
0.628228        10       complete       104
0.634265         6       complete       110
```

Anomalies
Wait states – global and per process

System:	V2250 8-way 11.0 8 GB memory Database server
Symptoms:	New installation Slow database throughput No system bottlenecks or high utilization

B3690A GlancePlus C.02.40.00   06:27:12 P1000147 9000/785   Current Avg High

CPU Util	SUL	6%	2%	14%
Disk Util		0%	0%	10%
Mem Util	S SU UB B	50%	49%	50%
Swap Util	UUR R	20%	20%	20%

GLOBAL WAIT STATES   Users= 1

Event	%	Time	Procs/ Threads	Blocked On	%	Time	Procs/ Threads
IPC	0.0	0.00	0.0	Cache	0.0	0.00	0.0
Job Control	0.0	0.00	0.0	CDROM IO	0.0	0.00	0.0
Message	0.0	0.00	0.0	Disk IO	0.0	0.00	0.0
Pipe	0.7	5.09	1.0	Graphics	0.0	0.00	0.0
RPC	0.0	0.00	0.0	Inode	0.0	0.00	0.0
Semaphore	0.0	0.00	0.0	IO	0.0	0.00	0.0
Sleep	45.9	353.71	69.6	LAN	0.0	0.00	0.0
Socket	0.0	0.01	0.0	NFS	0.0	0.00	0.0
Stream	0.7	5.09	1.0	Priority	0.0	0.09	0.0
Terminal	1.3	10.17	2.0	System	38.3	295.20	58.1
Other	13.2	101.60	20.0	Virtual Mem	0.0	0.00	0.0

Global Waits	Global Syscalls	DCE Global	System Tables	68 1	Next Keys	Netwk By Intrface	NFS Global	NFS By System

Anomalies
Wait states – global and per process

System:	V2250 8-way 11.0 8 GB memory Database server
Symptoms:	New installation Slow database throughput No system bottlenecks or high utilization
Diagnosis:	Identified high semop waits Database tuning required

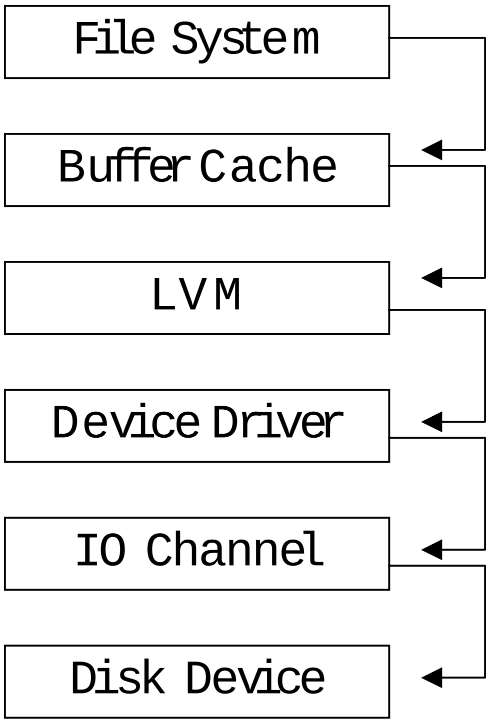
Isolating components

Make everything into a black box

Define and manipulate inputs and outputs

- The discrete elements of an I/O request
- Taking the network out of the picture
- OmniBack performance debugging techniques

Isolating components
discrete elements of an I/O request



mincache = direct

/dev/vg/rlvol

/dev/rdsk/cxtxdx

Isolating components
discrete elements of an I/O request

- Bottlenecks can happen at any of the layers in either direction

- Isolate the I/O test at one layer
  - mincache = direct
  - /dev/vg/rlvol
  - /dev/rdsk/cxtxdx

- Only test reading or writing

Isolating components
discrete elements of an I/O request

HP WORLD 2003 Solutions and Technology Conference & Expo

Code Fragment for Timing IO Requests

```
#include <sys/time.h>
#define delta_tv(tv_0, tv_1) \
    (tv_1.tv_sec - tv_0.tv_sec + (tv_1.tv_usec - tv_0.tv_usec)/1000000.0)

struct timeval xtv0, xtv1;
struct timezone tz;
double rdt = 0.0


main()
{
    gettimeofday(&xtv0, &tz);
    read(fd,buf,bufsize)
    gettimeofday(&xtv1, &tz);

    rdt = delta_tv(xtv0, xtv1);

    printf("milliseconds for read:%.3lf ms)\n", 1000*rdt);
}
```

Isolating components
taking the network out of the picture

- Multi-tiered applications (e.g. SAP) have large network components which can have a large impact on overall throughput

- Database access is often through sockets

- Techniques for isolation
  - Make local queries rather than client queries
  - With system issues execute problematic commands at the console
  - Use programs/benchmarks similar to those used for IO testing

Isolating components
OmniBack debugging techniques

- Understand the capabilities of each component in the configuration
- Isolate
  - Disk I/O
  - Network
  - Tape I/O
  - Updates to OmniBack database
  - Data compressibility

Case Study

Mike Chisholm
Hewlett Packard

HP WORLD 2003
Solutions and Technology Conference & Expo

Case Study

Problem

- Performance problem on restore of database files
- Restores take progressively more time to run on each iteration
- High system cpu utilization
- Reboot resolves problem temporarily
- Problem showed up after the customer applied a patch bundle containing many patches

Case Study

Client server backup/restore application

- Backup sets are located on local disk
- Multiple processes are spawned to read the data, decompress it and write it to a socket(discovered as part of the analysis)
- Other processes read the socket and write the destination database files(discovered as part of the analysis)
- Each restore is slower than the last, going from about 4 hours to restore the data up to about 8 hours
- Rebooting the server resets the performance profile

Case Study

Tools used -

- Glance
- Kitrace and kparse
- Prospect
- Kmeminfo
- Crashinfo on system TOC

Case Study

Glance -

- Glance global screen(default startup screen in character mode) showed increasing system mode CPU utilization during each run of the restore.
- Throughput in Kbytes/second("u" screen) dropped during each run.
- Number of system calls("s" to select PID, then "L" to observe system call rate) also dropped, less work was getting done.

Case Study

kitrace

- Kitrace was used to collect data during both a good and bad trace, then kparse was used to do a first pass analysis of the collected data
- Syntax

  ./kparse –R ki.<timestamp> > kp_<timestamp>.out

Case Study

kparse output from good case

```
Section3============= HARDCLOCK =============
Note: Check for sym= entries in HARDCLOCK traces
Note: Display frequency of kernel symbols when state=SYS
Note: Then breakdown percent each PID in what states
Checking and sorting on HARDCLOCK...
 Verifying version of kitrace has kernel syms..
Checking state=SYS,SSYS,& INTR and displaying freq on kernel syms..
Freq    KernSym
2421    b_na_loop
571     b_pcxu_loop
373     sul_pcxu_stop_here
311     lv_strategy
242     b_eight_word_loop
234     resume_cleanup
207     pdremap
189     biodone
```

Case Study

kparse output from bad case

```
Section3============= HARDCLOCK =============
Note: Check for sym= entries in HARDCLOCK traces
Note: Display frequency of kernel symbols when state=SYS
Note: Then breakdown percent each PID in what states
Checking and sorting on HARDCLOCK...
 Verifying version of kitrace has kernel syms..
Checking state=SYS,SSYS,& INTR and displaying freq on kernel syms..
Freq    KernSym
25461   kfree_one_to_superpage
1486    kalloc_from_superpage
1018    sul_pcxu_stop_here
489     b_na_loop
146     b_pcxu_loop
75      lv_strategy
71      b_eight_word_loop
59      biodone
```

Case Study

prospect data from good case

Prospect command line

```
./prospect_1100_2.2.1/prospect -Vkernel -f
  kern_profile.$(date +%m%d%H%M) sleep 20
```

```
Section 2.0: All Kernel Intersections with 100Hz System Clock
-------------------------------------------------------------------------------


Total KERNEL profile:
Total number hits: 6328     Of those, 704 are unique.

pcnt accum% Hits    Secs   Address   Routine name          Instruction  TEXT
 Filename

 40%    40% 2502   25.02  0x00036bc0 ulbcopy_gr_method        OBJECT    GLOBAL
 /stand/vmunix
  8%    48%  515    5.15  0x00036a80 ulbcopy_pcxu_method      OBJECT    GLOBAL
 /stand/vmunix
  5%    53%  324    3.24  0x000348f0 spinunlock               OBJECT    GLOBAL
 /stand/vmunix
  4%    57%  260    2.60  0x0010cab8 lv_strategy              FUNC      GLOBAL
 /stand/vmunix
  3%    60%  218    2.18  0x002023e0 resume_cleanup           FUNC      GLOBAL
 /stand/vmunix
```

Case Study

prospect data from bad case

```
Section 2.0: All Kernel Intersections with 100Hz System Clock
------------------------------------------------------------------------------

Total KERNEL profile:
Total number hits: 30567     Of those, 533 are unique.

pcnt accum% Hits    Secs   Address    Routine name          Instruction  TEXT
 Filename

 84%    84% 25535  255.35  0x00173fa8 kfree_one_to_superpage   FUNC       GLOBAL
  /stand/vmunix
  5%    89% 1543   15.43  0x00172060 kalloc_from_superpage    FUNC       GLOBAL
 /stand/vmunix
  4%    93% 1311   13.11  0x000348f0 spinunlock               OBJECT     GLOBAL
 /stand/vmunix
  2%    95%  601    6.01  0x00036bc0 ulbcopy_gr_method        OBJECT     GLOBAL
 /stand/vmunix
  1%    95%  170    1.70  0x00036a80 ulbcopy_pcxu_method      OBJECT     GLOBAL
 /stand/vmunix
  0%    96%  103    1.03  0x00038e60 bcopy_gr_method          OBJECT     GLOBAL
 /stand/vmunix
```

Case Study

Crashinfo, kmeminfo and system TOC

- We were closer, but still didn't understand completely what was happening
- The customer agreed to do a system crashdump to maximize information availability
- At this point, we could have run crashinfo and kmeminfo to get information from the live system and that would have been helpful, but in this particular case the customer had the luxury of being able and willing to do a full system dump

Case Study

kmeminfo output from system TOC

Kmeminfo and the superpage pool

- Typically, lots of time spent in kfree_one_too_superpage and kalloc_to_superpage are indicative of fragmentation of the superpage pool. This can be verified by use of the kmeminfo tool.
- This was not the case in this instance, details on next slide
- Run kmeminfo –c –kas

Case Study

kmeminfo output from system TOC

```
Kalloc              =    91499   357.4m    1%   kalloc()
   SuperPagePool    =    60154   235.0m    1%    Kernel superpage cache
size         count     free     highest
 0   4KB         0     1208     1614
 1   8KB         0     3553     4107
 2  16KB         0     3524     4424
 3  32KB         0     3110     3768
 4  64KB         0      312      482
 5 128KB         0      124      153
 6 256KB         0       13       43
 7 512KB         0        2       37
 8   1MB         0        1       36
 9   2MB         0        1       35
10   4MB         0        0       34
11   8MB         0        1       25
12  16MB         0        0       20
13  32MB         0        0       11
14  64MB         9        0        0

Total number of free page on pools:   60154
```

This isn't bad, we have seen cases where the number of free pages is > 500,000 and the "highest" is > 30,000

Case Study

crashinfo output from system TOC

- ./crashinfo
- We used the crashinfo tool to get a stacktrace of the processes at the time of the TOC – example

```
  SP          RP Return Name
0x400003ffffff16d0 0x000348f4 spinunlock+0x4
0x400003ffffff16d0 0x0017448c kfree_one_to_superpage+0x20c
0x400003ffffff1640 0x00174c34 kfree_to_superpage+0x34
0x400003ffffff15d0 0x00178ef4 kfree_common+0x294
0x400003ffffff14e0 0x0014c4a8 freeb+0x860
0x400003ffffff12b0 0x0008bb88 sbcompress+0x188
0x400003ffffff1230 0x00074f58 sbappend+0x20
0x400003ffffff11e0 0x000ac114 uipc_usrreq+0x9c
0x400003ffffff10f0 0x00147b0c sosend+0x49c
0x400003ffffff0f40 0x0027f480 soo_rw+0x80
0x400003ffffff0ed0 0x00164d24 write+0x104
0x400003ffffff0dc0 0x001656d4 syscall+0x28c
0x400003ffffff0c70 0x00035944 syscallinit+0x54c
```

Case Study

- In each case, the application was either in a sosend or a soreceive which in turn called freeb and got us into the superpage code
- From this point, GSE was able to look at the patches and determine changes that had been made to the ARPA stack that potentially had a direct impact on this
- It turns out that a change made to the ARPA stack to optimize typical socket writes had an adverse reaction with the superpage memory management when an application was doing large writes(>64k) to AF_UNIX/local sockets
- We had run into a corner case here because the application was restoring data from files locally on the same box as the destination database

Case Study

Resolution –

- Remove offending patch PHNE_26771 on 11.0 or PHNE_27063 on 11.11
- Install PHNE_28538 for 11.0 or PHNE_28895 for 11.11 to get optimizations along with fixes for the local socket corner case

Knowing your system

- Transaction reporting
  - Example: SAP instrumentation
    ARM instrumentation
- Maintain a history
  - sar, vmstat, scope, application measures
- Develop an intuition for your systems
- Watch it closely when its healthy
- Know the performance pattern over the day/week/month
- Internals knowledge of the application/database
- Internals knowledge of the OS

Rules of thumb

- CPU
- Memory
- I/O

Rules of thumb
CPU

- System CPU <= 30%
- Total CPU < 80%
- Small load average

Rules of thumb
Memory

- Never page out
- Never deactivate processes

Rules of thumb
IO

- Utilization < 50% on any drive
- Minimal queuing < 4
- Response time ~10 milliseconds

HP WORLD 2003

Solutions and Technology Conference & Expo

Interex, Encompass and HP bring you a powerful new HP World.

interex shared knowledge • shared power

encompass AN HP USER GROUP

hp invent