

Tools for Measuring System and Application Performance



- **Introduction**
- **GlancePlus**
 - Introduction
 - Glance Motif
 - Glance Character Mode
 - Glance Command-line
- **Xverbosegc**
- **HPjmeter and –Xeprof**
- **Other Tools**
- **gdb**

The Other Tools

Java Stack Traces

Instrumented pthreads

netstat

tusc

swapinfo

JProbe

Prospect



Java Stack Traces Java and Native

- Display using jdb
- Display using gdb (in 1.3.1)
 - gdb macros available from support or internally at:
 - http://pdlweb.cup.hp.com/~ecaspole/read_stack_trace.html
 - <http://pdlweb.cup.hp.com/app-dev>

Java Stack Traces

JVM Thread Stack Trace

- Generate with: `kill -s SIGQUIT <pid>`
- Dumps stack trace of every Java thread
 - Includes monitor information
 - waiting on
 - locked
 - Includes operating system thread id
 - Maps between Java and Glance
- Easy to read, but much information

Java Stack Traces

JVM Thread Stack Trace: locking

"Worker Thread 17" prio=9 tid=0x1310b70 nid=41 **lwp_id=14165**
suspended [0x1194d000..0x11948478]

at fields.FieldPropertiesLibraryLoader.forClass(FieldPropertiesLibraryLoader.java:67)

- waiting to lock <0x3ca45848> (a java.lang.Object)

at fields.FieldsServiceImpl.getFpl(FieldsServiceImpl.java:75)

at fields.FieldsServiceImpl.getFpl(FieldsServiceImpl.java:64)

at base.core.BaseObject.getFpl(BaseObject.java:2930)

at base.core.BaseObject.getFieldProperties(BaseObject.java:2661)

at core.BaseObject.getFieldProperties(BaseObject.java:2670)

at fields.FieldProperties.getFieldsInGroup(FieldProperties.java:1157)

at fields.FieldProperties.getFieldsInGroup(FieldProperties.java:1107)

Java Stack Traces

JVM Thread Stack Trace: waiting

"Reader Thread 12" prio=10 tid=0x135c290 nid=85 **lwp_id=14212**
waiting on monitor [0xed1c000..0xed1c478]

at java.lang.Object.wait(Native Method)

- **waiting on** <0x3df92650> (a rpc.server.ClientReader)

at java.lang.Object.wait(Object.java:424)

at server.ClientReader.run(ClientAcceptLoop.java:145)

- **locked** <0x3df92650> (a rpc.server.ClientReader)

at java.lang.Thread.run(Thread.java:479)

Instrumented libpthread

- Find contention
 - In Java VM
 - In user native code
- Replace libpthread.sl in \$SHLIB_PATH
- mutex_perf tool produces sorted text file
 - had waiters
 - max waiters
- Use pthread_mutex_setname_np() to name mutexes in the source code
 - Otherwise, you will spend a great deal of time in the debugger trying to track down the location where each mutex is created

netstat & ndd

- `netstat -p tcp`
- Toward the end of the output, look for:
 - connect requests dropped due to full queue
- Fix if necessary with:
 - `ndd -set /dev/tcp tcp_conn_request_max 1024`

tusc

- tusc - trace Unix system calls
 - Lightweight method of monitoring
 - Easy to use

```
tusc -l -p -f -E -o /tmp/tusc_file.out
```

- When starting java, use `$DEBUG_PROG=tusc`
- Shows call's parameters and return value
- Shows thread identifier of thread making system call

tusc Output

```
{4318} sched_yield() ..... = 0
{4313} sched_yield() ..... = 0
{4276} kwakeup(PTH_CONDVAR_OBJECT, 0x20c250, WAKEUP_ALL, 0x6fff1248) = 0
{4323} ksleep(PTH_CONDVAR_OBJECT, 0x20c250, 0x20c258, NULL) ..... = 0
{4317} sched_yield() ..... = 0
{4304} sched_yield() ..... = 0
{4296} sched_yield() ..... = 0
```

swapinfo

- Prints device and file system paging space
 - Available
 - Used
 - Free
 - Reserve

swapinfo Output

```
#
# swapinfo -m
          Mb      Mb      Mb      PCT  START/      Mb
TYPE     AVAIL    USED    FREE  USED  LIMIT RESERVE  PRI  NAME
dev       4096      0     4096   0%    0      -      1  /dev/vg00/lvol2
dev       2960      0     2960   0%    0      -      1  /dev/vg00/lvol5
reserve   -      1344  -1344
memory    3123     1251   1872  40%
```

Large Virtual Memory Pages

- 4 KB is the default page size
- HP-UX supports large text and data pages
- HP's Java is chartr'ed to use large pages
 - 1 Megabyte pages specified

swapinfo

- Java Heap is allocated “lazy swap”
- Swap space is not committed on allocation
 - Pages committed as needed
 - Saves swap space in system
 - Can cause sudden failure if system runs out of swap
- Can toggle with `-XX:+ForceMmapReserved`
 - Lazy swap (default) will force use of 4 KB pages
 - For large heaps, use `-XX:+ForceMmapReserved`
 - Example: 1 GB Java Heap:
 - 4 KB pages: 250,000 pages required
 - 4 MB pages: 250 pages required
 - Significantly lower pressure on the Translation Lookaside Buffer (TLB)

OS Tools

vps_stat

- Analyzes page allocations for system and/or a process
- Displays page sizes for:
 - Process DATA (includes C Heap)
 - Java Heap
- Used to optimize **T**ranslation **L**ookaside **B**uffer (**TLB**) usage
- Same information is available in Glance's Memory Regions window

C Heap Data Region Page Counts

VERSION	HINT	Page	SIZE[Count]
-----	-----	----	-----
Java 1.1.3	default 4K page size	16K	4K[2] 16K[7705 pages]
Java 1.1.7	default 4K page size	16K	4K[2] 16K[6066]
Java 1.1.7	64M page size	64M	4K[2] 16K[1] 256K[3] 1M[3] 4M[3] 16M[3] 64M[1]

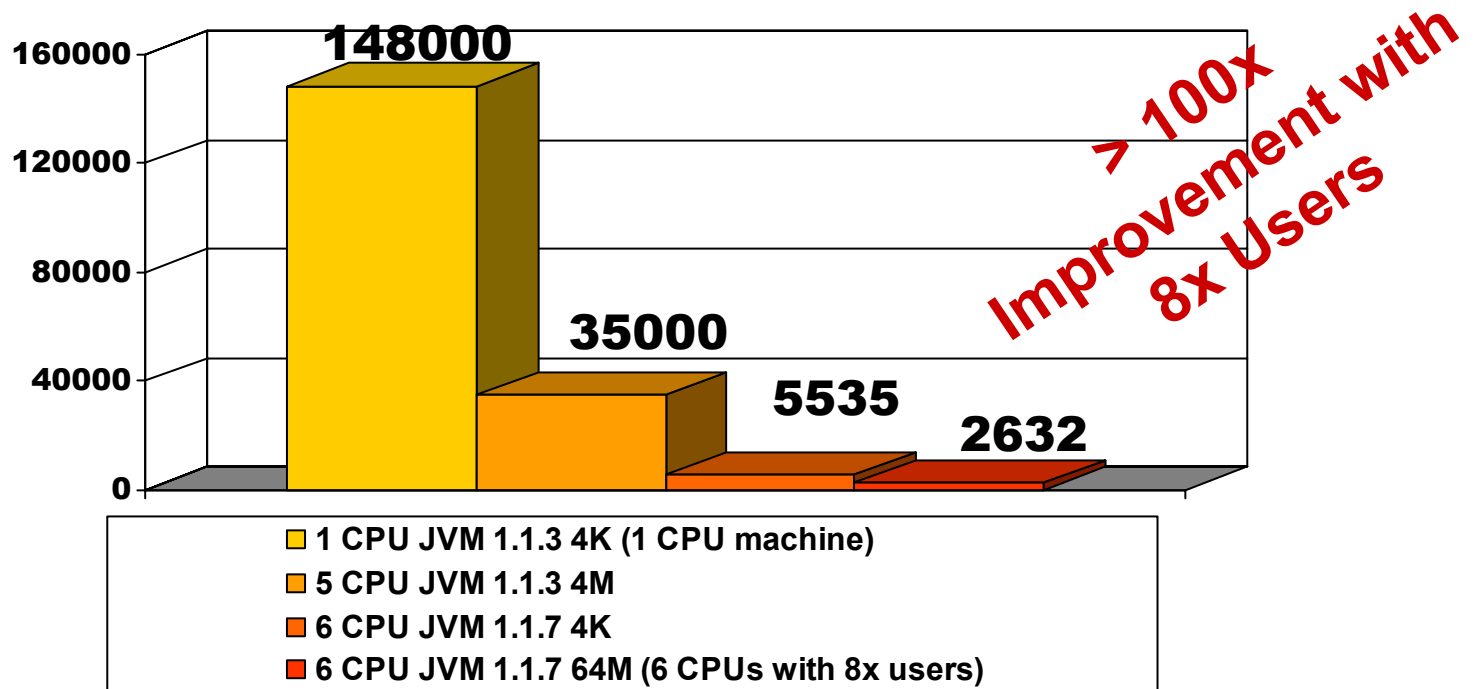
16 Pages Total

Java Heap Region Page Counts

VERSION	HINT	Page SIZE[Count]
-----	----	-----
Java 1.1.3 default 4K page size	16K	4K[4] 16K[49151 pages]
Java 1.1.7 default 4K page size	64M	4K[4] 16K[3] 64K[3] 256K[3] 1M[3] 4M[3] 16M[3] 64M[11]
Java 1.1.7 64M page size	64M	64M[12]
		12 Pages Total

OS Tools cyclometer - TLB Misses

Translation Lookaside Buffer Misses per Second



OS Tools

vmstat - System Overview

- CPU and Virtual Memory
 - -z zeroes counters
 - -S display swap information also
- Metric groupings:
 - Queues: run, blocked, runnable but swapped
 - Virtual memory metrics: page fault, swap rate
 - Scheduler metrics: context switch, interrupts
 - CPU usage: user, system, idle

OS Tools

vmstat - usage

vmstat -s (summary of system)

vmstat -dS 5 20 (20 collections at 5 sec.)

procs	memory	page	faults	cpu
r b w	avm free	si so	pi po fr de sr in	sy cs us sy id

Disk Transfers

device xfer/sec

First line output is average since bootup

OS Tools

vmstat - usage

- procs
 - r - in **run** queue
 - b - **blocked** for resources (I/O, paging, ...)
 - w - **waiting** - runnable or short sleep but swapped
- memory
 - avm - active virtual memory pages
 - text + data + stack for all processes (no shared)
 - free - size of the free list

OS Tools

vmstat - usage

- page
 - re - virtual memory page reclaims
 - at - number of virtual memory page faults
 - **si/so** - processes swapped in and out
 - pi/po - pages paged in/out
 - fr - pages freed per second
 - de - anticipated short term memory shortfall
 - sr - pages scanned by clock algorithm

OS Tools

vmstat - usage

- faults
 - in - device interrupts per second
 - sy - system calls per second
 - **cs** - context switches per second
- cpu
 - us/sy - user/system CPU usage
 - id - CPU idle

HP and Sitraka JProbe Server Side Suite

- The premier performance tuning suite for Java
 - Analyzes performance bottlenecks and memory leaks in your Java code
 - Detects deadlocks, stalls and race conditions
 - Locates untested Java code
 - Released on HP-UX for JProbe versions 2.8.1+
 - Works with the latest release of HP's J2SE
 - <http://www.sitraka.com>

Prospect Sampling-based Profiler

- Early version of Prospect with Java support
- Attach/Detach from Java process
- Collects profile information
 - Native code
 - Runtime compiled Java methods