Little Shop of Performance Horrors

Brendan Gregg
Staff Engineer
Sun Microsystems, Fishworks

FROSUG 2009
Performance Horrors

• I usually give talks on:
  - how to perform perf analysis!
  - cool performance technologies!!
  - awesome benchmark results!!

  in other words, things going right.

• This talk is about things going wrong:
  - performance horrors
  - learning from mistakes
Horrific Topics

- The *worst* perf issues I've ever seen!
- Common misconfigurations
- The encyclopedia of poor assumptions
- Unbelievably bad perf analysis
- Death by complexity
- Bad benchmarking
- Misleading analysis tools
- Insane performance tuning
- The curse of the unexpected
The **worst** perf issues I've ever seen!
The worst perf issues I've ever seen!

- SMC
  - Administration GUI for Solaris
  - Could take 30 mins to load on first boot
The **worst** perf issues I've ever seen!

- **SMC**
  - Administration GUI for Solaris
  - Could take 30 mins to load on first boot

- **Problems:**
  - 12 Million mostly 1 byte sequential read()s of
    /var/sadm/smc/properties/registry.ser, a 72 KB file
  - 7742 processes executed
  - 9504 disk events, 2228 of them writes to the 72Kb registry.ser file.

- Happy ending – performance was improved in an update
The *worst* perf issues I've ever seen!

- SMC (cont.)
- Analysis using DTrace:
  - syscall frequency counts
  - syscall args

  This is “low hanging fruit” for DTrace

- Lesson: examine high level events.
- Happy ending – performance was improved in an update
The worst perf issues I've ever seen!

- nxge
  - 10 GbE network driver
  - tested during product development
The worst perf issues I've ever seen!

• nxge (cont.)
  – 10 GbE network driver
  – tested during product development

• Problems:
  – kstats were wrong (rbytes, obytes)
    this made perf tuning very difficult until I realized what was wrong!
  – CR: 6687884 nxge rbytes and obytes kstat are wrong

• Lessons:
  – don't trust statistics you haven't double checked
The worst perf issues I've ever seen!

- nxge (cont.)
  - 10 GbE network driver
  - tested during product development

- Problems (#2):
  - memory leak starving the ZFS ARC
  - The kernel grew to 122 Gbytes in 2 hours.
  - 6844118 memory leak in nxge with LSO enabled
  - Original CR title: “17 MB/s kernel memory leak...”

- Lessons:
  - Bad memory leaks can happen in the kernel too
The worst perf issues I've ever seen!

- **nxge (cont.)**
  - 10 GbE network driver
  - tested during product development

- **Problems (#3):**
  - LSO (large send offload) destroyed performance:
    
    Priority changed from [3-Medium] to [1-Very High]

    *This is a 1000x performance regression.*

    brendan.gregg@sun.com 2008-05-01 23:25:58 GMT

    - 6696705 enabling soft-lso with fix for 6663925 causes nxge to perform very very poorly

- **Lessons:**
  - All configurable options must be tested and retested during development for regressions (such as LSO)
Common Misconfigurations
Common misconfigurations

• ZFS RAID-Z2 with half a JBOD
  - half a JBOD may mean 12 disks. A RAID-Z2 stripe may be 12 disks in width, therefore this configuration acts like a single disk:
    • perf is that of the slowest disk in the stripe
    • with so few stripes (1), a multi-threaded workload is much more likely to scale

• Max throughput config without:
  - jumbo frames
  - 10 GbE ports (they do work!)

• sync write workloads without ZFS SLOG devices
Common misconfigurations

• Not running the latest software bits
  - perf issues are fixed often; always try to be on the latest software versions
• 4 x 1 GbE trunks, and < 4 clients
  - they won't map to all ports
The Encyclopedia of Poor Assumptions
The Encyclopedia of Poor Assumptions

- More CPUs == more performance
  - not if the threads don't scale
- Faster CPUs == more performance
  - not if your workload is memory I/O bound
- More IOPS capability == more performance
  - slower IOPS? Imagine a server with thousands of slow disks
- Network throughput/IOPS measured on the client reflects that of the server
  - client caching?
The Encyclopedia of Poor Assumptions

- System busses are fast
  - The AMD HyperTransport was the #1 bottleneck for the Sun Storage products

- 10 GbE can be driven by 1 client
  - may be true in the future, but difficult to do now
  - may assume that this can be done with 1 thread!

- Performance observability tools are designed to be the best possible

- Performance observability statistics (or benchmark tools) are correct
  - bugs happen!
The Encyclopedia of Poor Assumptions

- A network switch can drive all its ports to top speed *at the same time*
  - especially may not be true for 10 GbE switches

- PCI-E slots are equal
  - test, don't assume; depends on bus architecture

- Add flash memory SSDs to improve performance!
  - Probably, but really depends on the workload
  - This is assuming that HDDs are slow; they usually are, however their streaming performance can be competitive (~100 Mbytes/sec)
Unbelievably Bad Performance Analysis
Unbelievably bad perf analysis

• The Magic 1 GbE NIC!
• How fast can a 1 GbE NIC run in one direction?
Unbelievably bad perf analysis

- The Magic 1 GbE NIC!
- How fast can a 1 GbE NIC run in one direction?
- Results sent to me include:
  - 120 Mbytes/sec
  - 200 Mbytes/sec
  - 350 Mbytes/sec
  - 800 Mbytes/sec
  - 1.15 Gbytes/sec
- Lesson: perform sanity checks
Death by Complexity!
Death by complexity!

- Performance isn't that hard, however it often isn't that easy either...
- TCP/IP stack performance analysis
  - heavy use of function pointers
- ZFS performance analysis
  - I/O processed asynchorously by the ZIO pipeline
Bad Benchmarking
Bad benchmarking

- SPEC-SFS

- cp
  - Copying a file from a *local* filesystem to an NFS share, to performance test that NFS share

- various opensource benchmark tools that don't reflect your intended workload

- Lesson: don't run benchmark tools blindly; learn everything you can about what they do, and how close they match your environment
Misleading Analysis Tools
Misleading analysis tools

- **top**

  load averages: 0.03, 0.03, 0.03
  17:05:29

  236 processes: 233 sleeping, 2 stopped, 1 on cpu

  CPU states: 97.7% idle, 0.8% user, 1.6% kernel, 0.0% iowait, 0.0% swap

  Memory: 8191M real, 479M free, 1232M swap in use, 10G swap free

<table>
<thead>
<tr>
<th>PID</th>
<th>USERNAME</th>
<th>LWP</th>
<th>PRI</th>
<th>NICE</th>
<th>SIZE</th>
<th>RES</th>
<th>STATE</th>
<th>TIME</th>
<th>CPU</th>
<th>COMMAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>101092</td>
<td>brendan</td>
<td>1</td>
<td>49</td>
<td>0</td>
<td>93M</td>
<td>25M</td>
<td>sleep</td>
<td>187:42</td>
<td>0.28%</td>
<td>realplay.bin</td>
</tr>
<tr>
<td>100297</td>
<td>root</td>
<td>26</td>
<td>100</td>
<td>-20</td>
<td>182M</td>
<td>177M</td>
<td>sleep</td>
<td>58:13</td>
<td>0.14%</td>
<td>akd</td>
</tr>
<tr>
<td>399362</td>
<td>brendan</td>
<td>1</td>
<td>49</td>
<td>0</td>
<td>95M</td>
<td>28M</td>
<td>sleep</td>
<td>53:56</td>
<td>0.12%</td>
<td>realplay.bin</td>
</tr>
<tr>
<td>115306</td>
<td>root</td>
<td>1</td>
<td>59</td>
<td>0</td>
<td>0K</td>
<td>0K</td>
<td>sleep</td>
<td>21:30</td>
<td>0.06%</td>
<td>dtrace</td>
</tr>
<tr>
<td>100876</td>
<td>brendan</td>
<td>1</td>
<td>59</td>
<td>0</td>
<td>0K</td>
<td>0K</td>
<td>sleep</td>
<td>103:52</td>
<td>0.05%</td>
<td>Xorg</td>
</tr>
</tbody>
</table>

- What does %CPU mean? Are they all CPU consumers?
- What does RSS mean?
Misleading analysis tools

- vmstat

```
# vmstat 1
kthr      memory            page            disk          faults      cpu
r b w   swap  free  re  mf pi po fr de sr s0 s1 s2 s3   in   sy   cs us sy id
0 0 0  10830436  501464  54  91  2  0  0  0  0  5  18  18  1  1835  4807  2067  3  3  94
0 0 0  10849048  490460  9  245  0  0  0  0  0  0  16  16  0  1824  3466  1664  0  4  96
0 0 0  10849048  490488  0  0  0  0  0  0  0  0  0  0  1470  3294  1227  1  1  99
0 0 0  10849048  490488  0  0  0  0  0  0  0  0  0  0  1440  3315  1226  0  1  99
0 0 0  10849048  490488  0  0  0  0  0  0  0  0  0  0  1447  3278  1236  1  1  98
```

- What does swap/free mean?
- Why do we care about de, sr?
Insane Performance Tuning
Insane performance tuning

- **disabling CPUs**
  - turning off half the available CPUs can improve performance (relieving scaleability issues)

- **binding network ports to fewer cores**
  - improves L1/L2 CPU cache hit rate
  - reduces cache coherency traffic

- **reducing CPU clock rate**
  - if the workload is memory bound, this may have little effect, but save heat, fan, vibration issues...
Insane performance tuning

• less memory
  - systems with 256+ Gbytes of DRAM – codepaths that walk DRAM

• warming up the kmem caches
  - before benchmarking, a freshly booted server won't have its kmem caches populated. Warming them up with any data can improve performance by 15% or so.
The Curse of the Unexpected
The Curse of the Unexpected

- A switch has 2 x 10 GbE ports, and 40 x 1 GbE ports. How fast can it drive Ethernet?
  - Unexpected: some cap at 11 Gbit/sec total!

- Latency
  - Heat map discoveries
  - DEMO (http://blogs.sun.com/brendan)
Thank you!

Brendan Gregg
Staff Engineer
brendan@sun.com
http://blogs.sun.com/brendan

“open” artwork and icons by chandan:
http://blogs.sun.com/chandan