

Performance Forensics

**Bob Sneed
Staff Engineer
Sun Microsystems, Inc.
Performance & Availability Engineering (PAE)
Bob.Sneed@Sun.Com**

**February 14th, 2003
Version 0.4**

Premises

- Medical analogies are extremely appropriate to discussing system performance
 - Skills and division of labor
 - Processes and record-keeping
 - Health versus pathology
- Solving “crimes” resembles fo•ren•sics
 - Cambridge Dictionary - “The analysis of physical information connected with a crime.”
 - Synthesis - “The science and practice of collection, analysis, and presentation of information relating to a crime in a manner suitable for use in a court of law.”

(Bob's) Background Experiences

- **Writing**
 - “ Sun/Oracle Best Practices”
 - “ Oracle I/O: Supply & Demand”
- **PAE “ASAP” Program**
 - PAE ASAP was a Sun internal process/project
 - Mail alias with all-star cast - interface customer issues to engineering, track through resolution
 - ASAP “Facilitator”: gather data; slam-dunk simple cases; broker top talent for harder cases; manage communications

Overview

- **Medical Analogues & Mixing Metaphors**
- **Tools and Record Keeping**
- **Common Analytical Errors**
- **Diagnosis: Tools and Techniques**
- **Conclusion**

Medical Analogues & Mixing Metaphors

Medical Truths

- Doctors do not take care of patients – nurses and staff do. You get 10 minutes with the doctor after they have your patient folder.
- Despite the fact that the knowledge base is commonly available, people employ doctors.
- The E.R. is always open, but does not specialize in total patient care.
- “Public Health” pertains to statistics over a large sample size.
- All drugs have contraindications and warnings.
- Doctors diagnose and prescribe, pharmacists dispense, surgeons cut and sew.
- Diet and exercise are of paramount importance!

Medical Protocols

- **First Things First – The patient folder!**
 - Age, gender, etc.?
 - Address, phone number, emergency contact?
 - Known risk factors? (Smoker?)
 - Drug allergies?
 - Current medications?
 - Medical history?
 - Who's paying?
 - Historical lab results & trends?
 - Current complaint and vital signs?
- **Normal care: General Practitioner or Primary Care Physician (and their staff) handles the routine; refer as needed to Specialists.**

Medical Protocols, Continued

- **Emergency Care**
 - First Things First (Patient records)
 - Triage (coarse-grain diagnosis & stabilization)
 - First Aid (expedient treatment)
 - Referral (for detailed diagnosis and management)
- **Medical Practice**
 - Hippocratic oath (classical versus modern)
 - Test according to established protocols
 - Treat according to established protocols
 - Use the least intrusive effective treatment
- Each specialty has its own protocols.

Customer Syndromes

- Failure to instrument business metrics
- Bad record keeping, poor configuration management, changing too many things at once
- Not seeing the doctor, skipping physicals, waiting until you are really sick, suffering in silence
- Self-medicating, throwing iron at the problem, chaotically turning knobs (or conversely, being too mired in established local practices)
- Diet & Exercise (Diet -> Best Practices ...
 - Exercise -> Stress tests, rehearsals, benchmarks)
- Lack of reproducible workloads & inexperience with benchmarking techniques
- Hypochondria & Paranoia

Consultant Syndromes

- Failing to begin with a full assessment of configuration practices
- Being compelled to find problems in their particular area of expertise
- Failing to understand platform dependencies
- Attempting to master all specialties
- Failing to refer or collaborate
- Treating knowledge as “consulting assets” versus scientific norms of open sharing and peer review
- Poor configuration management, changing too many things at once, bad record keeping

Service Organization Syndromes

- Failing to begin with a full assessment of configuration factors and parameters
- Failing to maintain “patient folder” data – focus instead on “case management” data
- Failing to understand 3rd party dependencies
- Failing to refer or collaborate
- Propensity to close (performance) cases as ...
 - Not a bug
 - No trouble found
 - Cannot reproduce
 - Needs tuning / needs a consultant

Engineer Syndromes

- Poor “Bedside manner”.
- Narrow technical focus.
- Difficulty relating to customer experiences and constraints.
- Viewing the world through product engineering processes versus customer processes, constraints, and business goals.
- Frequently only offered a limited view of an alleged issue, rather than the entire 'patient file'.

What Doctors Do

Health Maintenance and Disease Remediation

- **Process**
 - Hippocratic Oath (community, referrals)
 - Division of labour, records keeping
- **Knowledge**
 - Education in medical sciences
 - Legal factors
 - Risk management
- **Skills**
 - Professional experience & judgement
- **Tools**
 - Lab tests as appropriate
- **Collaboration**

Tools and Record Keeping

Survey Tools (Statics)

- Sun Explorer
 - /etc/system
 - Software and patches
 - Storage Factors
 - Network tuning
- Oracle files
 - init.ora (& possible ifile)
 - alert.log
 - trace files

Survey Tools (Dynamics)

- Common goal: establish “normal” and “trend”
- iostat, vmstat, mpstat, netstat, nfsstat, sar
 - Cumbersome
 - timestamps difficult to correlate
- SE Toolkit & “oracalator.se”
- Memtool
- Teamquest
- Other 3rd Party

Diagnostic Tools (Dynamics)

- Common goal: find the “root cause”.
- iostat, vmstat, mpstat, netstat, nfsstat, sar – same tools as used for characterization and trending.
- truss, sotruss, appttrace, cputrack.
- lockstat, trapstat, busstat, cpustat, etc.
- Unreleased or experimental engineering tools.
- Post-processors or 'helper programs' for analysis of logged data.
- Programmer's tools (eg: collect, analyzer)

Record Keeping

- Make your own “patient file”
 - Historical operational data
 - Periodic configuration snapshots
- Don't run Explorer on busy production systems
- Supply “patient file” along with trouble reports, whether or not asked

Common Analytical Errors

Stats: Cache Hit Rate

- Q: What is the practical difference between a 95% cache hit rate and a 96% cache hit rate?

Stats: Cache Hit Rate

- Q: What is the practical difference between a 95% cache hit rate and a 96% cache hit rate?
- A: 20% (at least in principle) ... because the misses are the real performance concern.

Stats: Limitations of Averages

- Averages hide temporal factors.
 - 10 events in 1 second is 10/sec
 - The same 10 events in 1 second is 1 per second if averaged over 10 seconds.
- If one is late 50% of the time and early 50% of the time one is never - “on average” - “on time”.

Logic: Post-Hoc Causal Fallacy

- In Latin: “*Post-hoc, ergo prompter hoc.*”
- In Plain English: “*After that, therefore because of that*”.
- Statistics-speak: “*Correlation does not imply causation*”.
- MORALE: Any credible hypothesis for a phenomenon must involve a clear chain of causality.

Logic: Why is 100% Busy Bad?

- 100% busy is not necessarily bad.
- “Headroom” is not equivalent to “idle time”.
- Service Level Agreements (SLA's) or Critical To Quality indicators (CTQ's in Six Sigma-speak) might be greatly exceeded at 100% CPU usage.
- “Usage” != “Utilization”

Experiment Design: All Other Things are Rarely Equal

- The devil is in the details
- It is common to upgrade multiple system elements at once
- Storage factors are very difficult to assess as being “the same”
- “Equal” is not always “good” - alternate tuning may be expected with newer products

Experiment Design: Initial State can Matter

- Ideally, benchmark methodology will assure each test iteration has a controlled initial state.
- System reboot is rarely indicated, but does lead to consistent test results.
- “Warm caches” can have huge impact on back-to-back runs of the same task.
- There are often “first use” versus “subsequent use” performance deltas.
- Testing can leave “footprints”.

Diagnosis Tools and Techniques

Diagnostic Guidelines

- Always start analysis with review of configuration state factors (patches, features, and parameters) ... or in other words “the patient file”.
- Analyze top-down, starting with business metrics.
 - For Oracle per se, that means STATSPACK
 - For apps layered atop Oracle, it may be up to the user to add instrumentation of business metrics
- You can always find where the time went, and often the bigger challenge is finding why.
- Keep in mind that 80% of illness relates back to “diet and exercise”.
- Keep in mind that the answer is often “bad code”.

Diagnostic Strategies

- Configuration state review.
- What's going around? Clinical signs?
- Mine data for specific problem indicators.
- Where did the time go?
 - STATSPACK
 - truss -c, truss -f, sotruss, apptrace, time
 - collect (example: case study)
- Why did the time go?
 - Hot functions, MT-hot functions
 - lockstat, trapstat, busstat, prstat
- Data aggregation & visualization.
 - Orca
 - iostat post-processing (example: iobal)

Diagnostic Skills

- Knowing what's 'normal'. Is the complaint valid?
- Knowing how the thing actually works (medical doctors study chemistry, math, logic, physics, anatomy, physiology, pathology, etc).
- Avoiding common errors of reasoning.
- Using the right tools at the right time.
- Knowing when to refer to a specialist.
- How doctors do it: process, knowledge, skills, tools, collaboration.

Configuration State Analysis

- Start with Explorer (or subset of it)
- patchdiag (<http://sunsolve.sun.com>) - compare installed patches to available patches.
- /etc/system – rationale of all settings known? All situational “Best Practices” applied?
- Filesystem factors – buffering modes used? Write concurrency addressed? VxFS blocksize?
- Gross use of memory – SGA size (ipcs -b)?
- Solaris messages, Oracle alert file – indications of abnormalities?
- Oracle init.ora – rationale for all settings known? Any settings contrary to “Best Practices”?
- Storage – brand and model, VM factors?

Performance Complaints

- “Slow”, relative to ...
 - yesterday, last week, last year
 - before previous upgrade
 - brand X
 - model Y
 - expectations
 - the business requirement
- “Variance” ...
 - pauses, hangs, slowdowns
 - distribution of response times
- Scaling issues – degradation with increasing load.

Performance “Diseases”

- Best Practice deviations (Configuration and Operational errors)
- Bugs
- Algorithm design
- Inefficient coding
- Component failures
- Degraded operational modes
- Resource contention and queuing
- Performance versus availability tradeoffs
- Space versus speed tradeoffs
- Resource saturation
- Deadlock and race conditions
- Pathological interactions with target architecture

Intrusiveness of Tools

- `truss -c` (for a brief interval) – fairly low
- `collect` – fairly low
- `truss`, `sotrust`, `apprtrace` – fairly high
- `profile code` – fairly high
- `lockstat`, `trapstat`, `kstat` – low
- `iostat`, `sar` – fairly low
- Oracle timed statistics & `STATSPACK` - low
- `debug trails` – varies with verbosity, # users, possibility of file contention
- Tool overhead can accumulate when too many tools are used.
- Measurement and data logging can comprise a large part of the workload.

busstat, cpustat

- For wizards only! (Like “microcellular biology and pathology”.)
- Requires intimate engineering knowledge of hardware architectures and embedded counters.
- Other more friendly tools are percollating within engineering (eg: cachestat)

netstat

- `netstat -s` – loads of network-related counters, many of which are not available as `kstats`.
- `netstat -i` – per-interface summary

iostat

- **iostat -nxzTd** – a favourite, on 5, 10 or 60 second sample intervals – higher sample rates are more intrusive.
- Reams of data hard to interpret without a helper tool (example: 'iobal')
- Common diagnoses – slow component, “hot spots”, saturated component (HBA or disk/LUN)

prstat

- Exploits Solaris 'microstate accounting' and allows thread-level analysis (prstat -mL)
- Breakdown of run state information
 $usr+sys+trp+tfl+dfl+lck+slp+lat = 100\%$
- ICX – Involuntary ConteXt switching – schedule quantum expiry or preemption by higher priority thread
- VCX – Voluntary ConteXt switches – blocking system call or yield().
- SCL – syscall rate
- SIG – signal rate
- Does not reveal behaviour of kernel (versus user-mode) threads

lockstat

- Reveals overhead of instrumented system-level locking overhead
- Allows statistical sampling of kernel execution (wizards only)
- Output is cryptic, but will make some sense to analysts with some programming experience

trapstat

- Stuff most people never see – mostly the domain of wizards
- Only interesting if trap handling comprises a high percent of resources over the measured interval
- Diagnoses include - pathological memory behaviour (TLB thrashing); pathological math exceptions
- In Solaris 9. Available, but poorly documented for Solaris 8 (-i for interrupts, -t or TLB data)

collect

- Part of the SunONE(TM) Studio tools (Formerly branded 'Forte')
- Allows analysis of non-instrumented binaries
- Rather programmer-geeky stuff, but extremely powerful.
- Has GUI analysis engine, as well as CLI interface.

Conclusions

Conclusions

- Keep your own “patient profile”
- Use top-down approach for problem resolution
- Master triage and collaboration first

Resources

- <http://www.sun.com/blueprints>
- <http://www.sun.com/sun-on-net/performance>
- <http://www.setoolkit.com>
- <http://www.solarisinternals.com>
- <http://www.orcaaware.com/orca>
- <http://sunsolve.sun.com>
- <http://metalink.oracle.com> (need CSI login)
- <http://otn.oracle.com> (free, online docs)