QUEST 2008 Performance Test Monitoring for Free!

Jack R. Frank
Managing Consultant
Test Automation Practice Manager
Mosaic, Inc.
205 N. Michigan Ave., Ste. 2211
Chicago, IL 60601

www.mosaicinc.com
312-819-2220

Who is Mosaic?

- Incorporated in 1988 with Headquarters in Chicago
- Specialize in Software Risk Management
- Service Areas
  - Functional & Performance Test Planning and Execution
  - Test Automation Design and Implementation
  - Quality Assurance
  - Metrics and Measurement
  - Large/High-Risk Project Support
  - End User Support
- Products
  - MSTAR®
  - QSTAR™
  - TR Sizer ™
  - Object Profile Builder™
  - DSTAR™

2

Performance Test Monitoring

- No Tool Used or Described Herein Requires a Payment (except Excel)
- Performance Testing 101
- Measurement - What to Measure
- Metrics - What to Extract from Measurements
- How to Do It
  - Built in functions
  - Open source, scripting
- Examples, Links, Favorites

MOSAIC

3

Performance Testing 101

- Define Your Requirements
  - Fast… oh, within 2 seconds, 95% of the time
    - Even with 1.5 times normal load
    - Never more than 5 seconds unless…
  - CPUs under 80%, bandwidth < 75%, etc.
- How Are We Testing?
  - Vendor tools, home-grown harnesses, injectors
  - Users pushing buttons, production feeds, etc.
- Where Are We Testing?
  - Full test environment or pre-production system?
  - What isn't like production? What's missing?

MOSAIC

4

Typical Measurements

- Computers
  - CPU, memory usage/paging, disk I/O
  - Network card saturation, file system capacity
  - Windows, UNIX, mainframe, mobile, etc.
    - Servers, desktops, appliances…
- In Between
  - Message queues, FTP processes, others:
    - JMS, RV, AMQP
  - Routers, switches, internet, intranet, etc.
- Databases
  - Server statistics - persist-commit time, cache hits
  - SQL efficiencies, lock contentions, deadlocks
  - Memory usage, file system size, file system waits…

5

Your Measurements

- Your Applications: Technically
  - Framework - garbage collect, memory usage
  - Clustering effectiveness, load balancing
  - System to system times - Tinker→Evers→Chance
  - Internal times - glove→hand→pivot→throw
- Your Applications: Functionally
  - End user experiences
    - Request→Response→Respond→Response
    - Here, there, anywhere - Global
  - Other performance requirements
    - Batch times, end of day tasks, etc.

6

Your Metrics

- Unique to Your Business Rules and Roles
- Know the Metric Ranges
- Must Correlate with System Metrics
- Big Risk of Uncertainty
  - System tools tested by thousands
  - Applications tools tested by just you and yours
- Let Development Languages/Frameworks Do Some of the Heavy Lifting:
  - Log4j, Log4net (open source!), etc.
  - Efficient, handles max. sizes, roll-over
  - Leveling - info, warn, error, perf, debug

MOSAIC

7

What You Want From Metrics

- No Uncertainty…
- Passing the Torch
  - Some identifier for the transaction
  - Consistency!
  - End to end tracking
- Synchronization
  - All system must be within x milliseconds
  - Need to check synchronization every z seconds/minutes
    - Desktops clocks vary… so do your eyes
- Granularity
  - hh:mm:ss.sss please, UTC (GMT) is always nice
  - Elapsed time not as good, but use milliseconds
- Recorded Throughout Testing:
  - IN A FORMAT THAT'S EASY TO PROCESS!

MOSAIC

8

Synchronizing Windows Time

- Control Panel - Date & Time: Internet Tab
  - External time source (NIST, Microsoft)
  - Once a day
- W32tm.exe Command - Built into Windows
  - /? - provides 'help'
  - /resync - re-synchs your computer
  - /computer:me,you,yours - many computers
  - /domain:our_domain - when you have one
  - Can identify drifting machines

9

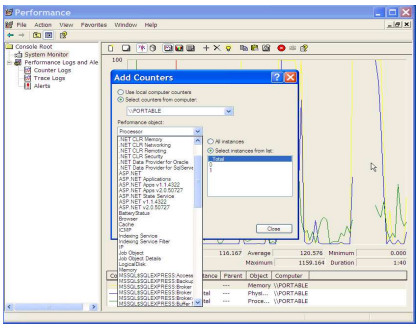
UNIX Time

- Slight Variations Amongst Linux(s), UNIX, Solaris…
  - Order of defaults and output may vary
  - But greater variety, uncertainty usually less
- What Time Is It?
  - NTP (network time protocol) always in UNIX
  - Configuration includes systems, domains, etc.
    - Works with Windows too (TCP/IP stack)
    - Best to have 1 Windows domain server synch to UNIX
      - Rest of Windows synch to that server

10

Windows Metrics Galore

- Windows' Wonderful Performance Monitor
  - Control panel/Admin tools/Performance
    - Or run: perfmon.msc at command line

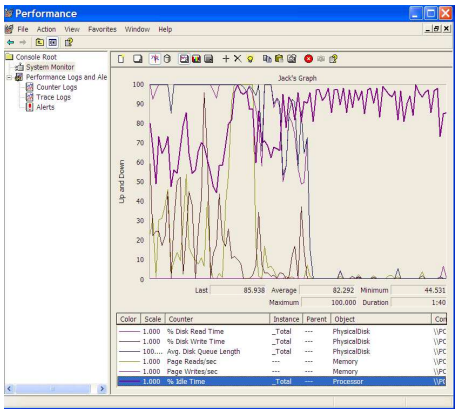


MOSAIC

11

My Metrics… For Example

- Jack's Chart
  - Just my metrics
  - Colors, title, etc.
  - Scales, periods

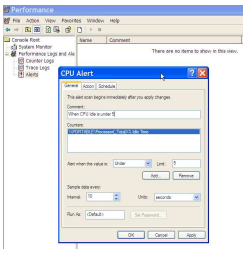


MOSAIC

12

Windows - Alerts

- Also Under Performance Monitor
- My Alerts, Action, Schedule
  - If CPU idle < 5%, check every 5 seconds - Alert!
  - Can log events, run a command
  - Can be scheduled for your timeframe only



MOSAIC

13

Window Logs For Analysis

- Still in Perfmon… Counter and Trace Logs
- Set Parameters
  - Your metrics
  - Set output to delimited, not binary
  - Make sure max size is reasonable
  - Make sure file system has enough space!
  - Schedule gathering during your test times
- Beware of Uncertainty
  - Get functional performance metrics w/o logging
  - Benchmark download - Fresh Diagnose software
    - www.freshdevices.com/freshdiag.html
  - Run with logging - what is impact?
  - Lower time intervals, # of metrics, roll-overs
- See Example at End of Presentation

MOSAIC

14

UNIX CPU Metrics

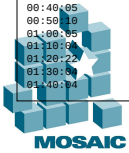
- Real Time CPU - top
- Logged CPU - sar
  - Also can have disk
  - And other devices
- Kind of Text-y Though

```
Terminal — top — 89x27
Processes:  50 total, 3 running, 1 stuck, 46 sleeping... 146 threads   08:22:57
Load Avg:  1.20, 0.40, 0.27     CPU usage:  67.8% user, 33.8% sys, 0.0% idle
SharedLibs: num =  117, resident = 30.1M code, 3.23M data, 10.1M LinkEdit
MemRegions: num =  6019, resident =  180M + 14.5M private,  131M shared
PhysMem:  60.5M wired,  295M active,  148M inactive,  504M used, 7.27M free
VM: 3.24G + 94.5M   28486(137) pageins, 2974(0) pageouts

  PID COMMAND      %CPU  TIME    #TH #PRTS #MREGS RPRVT  RSHRD  RSIZE  VSIZE
  491 Microsoft    0.0%  0:00.46   2    47   130  1.84M  10.5M  12.6M  101M
  490 screencapt   0.0%  0:00.01   1    27    30   212K   888K   912K  59.3M
  488 Microsoft   11.3%  0:02.45   2    69   410  16.8M+ 51.0M+ 42.1M+ 185M+
  487 iTunes      37.2%  0:07.84  15   258   361  18.1M  29.4M+ 30.0M-  166M
  486 top          4.5%  0:03.87   1    28    26   288K   464K   672K  27.1M
  472 bash         0.0%  0:00.06   1    12    15   152K   920K   752K  18.2M
  471 login        0.0%  0:00.03   1    13    37   132K   456K   516K  26.9M
  410 Terminal     0.3%  0:18.41   3    67   166  1.98M  19.1M+ 20.1M-  133M
  404 sipd         0.0%  0:00.02   6    29    37   240K  1.06M   920K  30.4M
  387 lookupd      0.0%  0:00.53   2    35    80  492K+ 1.01M  1.27M  28.5M
  365 nmbd         0.0%  0:00.53   1    17    35   240K   952K  1.04M  27.9M
  351 System Eve   0.0%  0:00.90   1    57    96  1.05M  4.65M+ 3.17M  100M
  349 firefox-bi  22.7% 88:12.45   9   113   700   128M  49.3M+  146M  319M
  346 IIDCAssist   0.0%  0:00.08   2    29    34   316K  2.01M+ 1.75M  28.9M
  338 automount    0.0%  0:00.03   2    29    26   228K   932K   940K  28.3M
  335 automount    0.0%  0:00.01   2    28    26   224K   932K   960K  28.3M
  332 rpc.lockd    0.0%  0:00.00   1     9    16    80K   420K   144K  17.7M
  323 nfsiod       0.0%  0:00.00   5    29    23    96K   364K   156K  19.6M
  305 ntpd         0.0%  0:00.66   1    10    18   136K   560K   340K  17.9M
```

```
Linux 2.4.21-27.0.2.x (jack_server)          01/12/08

00:00:22          CPU     %user     %nice   %system   %iowait     %idle
00:10:04          all      0.25      0.00      0.39      1.86     97.49
00:20:10          all      0.19      0.00      0.36      1.74     97.71
00:30:10          all      0.20      0.00      0.45      1.74     97.61
00:40:05          all      0.20      0.00      0.40      1.75     97.65
00:50:10          all      0.20      0.00      0.48      1.69     97.63
01:00:05          all      0.21      0.00      0.39      1.76     97.64
01:10:04          all      0.21      0.00      0.37      1.80     97.62
01:20:22          all      0.25      0.00      0.38      1.85     97.53
01:30:04          all      0.20      0.00      0.37      1.81     97.62
01:40:04          all      0.19      0.00      0.36      1.76     97.69
```



UNIX and Open Source Tools

- No Single Tool Like Windows
  - Yet, there could be one on the web somewhere!
- System Utilization Commands
  - vmstat - virtual machine memory - also CPU
  - iostat - I/O metrics
  - netstat - network device usage
  - prstat - Sun Solaris only - nice mix of stats
- Open Source Tools
  - Endless… **http://ltp.sourceforge.net/tooltable.php**
  - Windows too… **http://sourceforge.net/ (searchable)**

Few Comments on Open Source

- From E-Week 10-Oct-2007:
  - Not a secret society
  - Not just for coders
  - It's Windows-friendly too
  - You can ignore the source!
- Corporate Policies…
  - Analysis tools - installed just for me
    - Temporarily, I promise.
  - Incorporating tools into corporate test systems requires forward-looking mgmt…

17

Summary

- Performance Testing Is a Technical Effort
- Planning Will Make or Break the Effort
  - Know your technical measurements
  - Know your application measurements
    - Dig, cajole, nag them out of the team!
  - Know how to convert measurements into metrics
- Keep Your Eyes Open for New Tools
- Keep Your Technical Skills Current

18

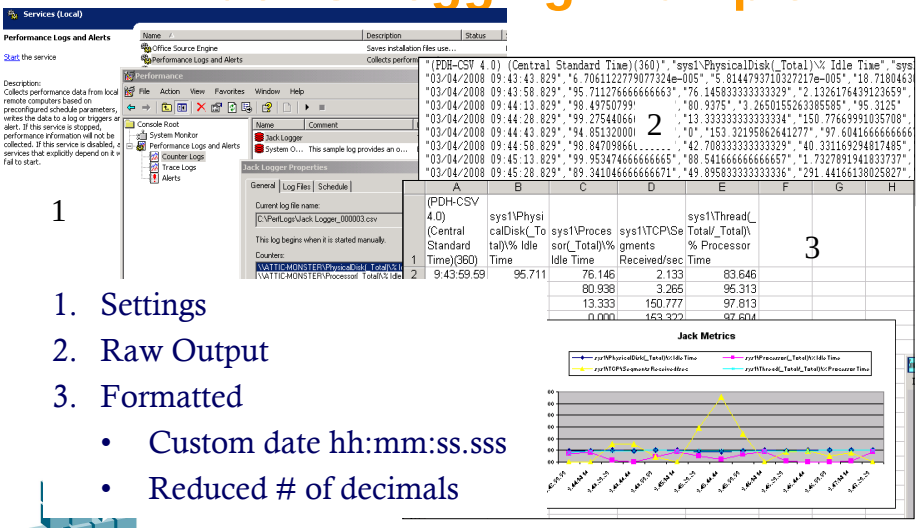
Examples and Favorites

- The Next Two Slides Show a Logging Example
  - First shows Windows' logging and the processing of a log with MS Excel
  - Second shows three UNIX utilities used to create a log that is then a processed by a Python script that produces a textual report
- After the Example Slides is a Slide Listing Some of My Favorite tools
  - Some may nag you for a donation…
- Final Slide has a Few Cool UNIX Commands
  - Will work with Cygwin under Windows
  - A bit complex if you are not familiar with UNIX

MOSAIC

19

Windows Logging Example



1. Settings
2. Raw Output
3. Formatted
   - Custom date hh:mm:ss.sss
   - Reduced # of decimals

MOSAIC

20

UNIX Logging Example

```
# 12:40 to 13:30
vmstat 1 2 >  `sys1`-`date +%m%d%y`.txt
df -k >> `sys1`-`date +%m%d%y`.txt
sar -f /var/adm/sa/sa12 -s 12:40:00  -e 13:30:00 >> `sys1 - date +%
```

1

```
procs                  memory      swap          io
 r  b   swpd   free   buff  cache   si   so    bi    bo
 1  0   1276  24432 221428 2016236    0    0     1
 0  0   1276  24452 221428 2016236    0    0     0
Filesystem           1K-blocks      Used Available Use%
/dev/hda2              8254272    296616   7538360   4%
/dev/hda6              4127076    916412   3001020  24%
/dev/hda7              4127076    547304   3370128  14%
/dev/hda9             11369416   5446968   5344916  51%

Linux 2.4.21-27.0.2.x (sys1)        06/12/07

12:40:11          CPU     %user     %nice   %system   %iowait     %idle
12:45:21          all      0.88      0.00      0.43      0.09     98.60
12:50:05          all      0.77      0.00      0.54      0.09     98.61
```

2

```
for line in f:
    ts=line[0:
  if ts >= star
      flds=lin
      el_no=le
      str_metr
      if str_m
          recs=
          CPU_a
  ecs > 0:
  utl=+100.0-
  print 'Util
  if u
      p
  else
      p
else:
  prin
f.close
```

3

```
VMSTAT Instant Test Environment Metrics for Testing done in:  sys
              Thu, 12 Apr  12:40:00  /  13:30:00 +0000
sys1      ->: CPU  25.0 % -> Mem Used/Free     455400     24708  94.9%  *** CPU +++ Memory
sys09     ->: CPU   4.0 % -> Mem Used/Free      89088     32752  73.1%   +++ Memory
sys05     ->: CPU   5.0 % -> Mem Used/Free    7905192   5107720  60.7%   +++ Memory
sys09     ->: CPU   0.0 % -> Mem Used/Free   36357632  31051792  53.9%
sys11     ->: CPU   1.0 % -> Mem Used/Free    6016912   7813560  43.5%
sys23     ->: CPU  20.0 % -> Mem Used/Free    8581944   9299840  48.0%  *** CPU
                           -- end of report --
```

4

1. Shell Script to Gather 3 Metrics
2. Output Example
3. Python Script to Parse
4. Report with Aggregates from 6 Servers

MOSAIC

21

Some Favorite Tools

- Wireshark (Ethereal) - Monitor Network Traffic
- Firefox HTTP Headers - Monitor, Capture and Replay
- Curl - Command Repeat (Loops Scripts, Code)
- Numpy or MATLAB
  - Math functions for Python and other languages (larger arrays, easy to do the math)
- Cygwin - UNIX command line for Windows (for File Processing)
- MS Excel… >64k Rows in 2008
  - FYI Java date to Excel date: (javadate/86400000)+25569
    - +/- for your time zone vs UTC…
- Textpad - Huge File Capacities, Sorting, Column Mode
- UNIX - grep, awk and Other Geeky Things
- ping - Raw How Long Measure

MOSAIC

22

Favorite UNIX commands

- grep what where
  - grep error huge_log.txt
  - grep 'elapsed.[0-9]\{4,\}' my_log.txt
    - Show word elapsed. Followed by a 4 digit number
      - Finds 1000-9999 milliseconds
- grep and awk - find, parse, do math
  - grep metric_a my_log.txt | awk '{split($1,parts,","); if(parts[5]>30000) print $1;}' > big.txt
    - Find metric_a entries, break into pieces based on commas, check 5th item for values > 30k, put those lines into a file called big.txt
  - grep metric_a my_log.txt | awk '{split($1,parts,",")} {tot += parts[5] } END {print tot/NR}'
    - Find metric_a entries, break into pieces based on commas, sum up all 5th items, when done print to the console the average value for 5th item (NR is number of records)
- Note - These Can Run in Real-time as Monitors
  - tail -f my_log.txt | grep 'elapsed.[0-9]\{4,\}'

MOSAIC

23