Monitoring & Analyzing MySQL Performance with dim_STAT

Dimitri KRAVCHUK
Principal Benchmark Engineer
Sun Solution Center
Before we start...

- Few words about SSC :-)  
- Paris <= 10Mbit, 20ms latency => LLG
Sun Solution Centers near you

- [http://sun.com/ssc](http://sun.com/ssc)

Global Sun Solution Centers
- **View All Global Locations**
  There are 26 worldwide Sun Solution Centers providing Sun partners and customers with easy access to Sun skills and equipment to assist them in testing, integrating, and...

Authorized Sun Solution Centers
- **View All Authorized Locations**
  Sun's Authorized Sun Solution Centers are led by key partners who combine their Sun technology expertise with industry-specific solutions. The Authorized Sun Solution...
Agenda

• History / Main Idea
• Architecture Overview
• Main Features and Components
• Add-Ons Overview
• MySQL-oriented Add-Ons
• dbSTRESS & IObench
• Report Tool
• Live Demo
• Q & A
During the next 15min...

- Forget about MySQL ;-)  
- Think globally..  
- Think generally...  
- etc..
Preface: Close Look...
Preface: Close Look zoom--
Preface: Close Look zoom-- x10
Performance issues

• May come from anywhere...
• Sometimes you may wait them in a wrong place :-)
• Monitoring is the MUST :-)
• Monitoring without Analyzing = wasted time :-)
• Understanding from where the problem is coming is already a half of solution!
Why home-made tools?...

- Don't like stupid work...
- Tool adapted to Humans, and not Humans to the tool
- Best implementations when User = Developer
- Mind Pleasure
- Laziness = Power of Progress!
- Machines should help Humans!
## Monitoring & Analyzing Dilemma...

### Table Data

<table>
<thead>
<tr>
<th>r/s</th>
<th>w/s</th>
<th>kr/s</th>
<th>kw/s</th>
<th>wait</th>
<th>actv</th>
<th>wsvc_t</th>
<th>asvc_t</th>
<th>%w</th>
<th>%b</th>
<th>device</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1929.9</td>
<td>0.0</td>
<td>3859.7</td>
<td>0.1</td>
<td>1.0</td>
<td>0.0</td>
<td>0.5</td>
<td>5</td>
<td>75</td>
<td>c31t1d0</td>
</tr>
<tr>
<td>0.0</td>
<td>1622.3</td>
<td>0.0</td>
<td>3244.5</td>
<td>0.1</td>
<td>1.2</td>
<td>0.0</td>
<td>0.7</td>
<td>5</td>
<td>80</td>
<td>c5t4d1</td>
</tr>
<tr>
<td>0.0</td>
<td>1928.5</td>
<td>0.0</td>
<td>3857.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.0</td>
<td>0.5</td>
<td>5</td>
<td>74</td>
<td>c31t1d0</td>
</tr>
<tr>
<td>0.0</td>
<td>1570.1</td>
<td>0.0</td>
<td>3140.3</td>
<td>0.1</td>
<td>1.1</td>
<td>0.0</td>
<td>0.7</td>
<td>6</td>
<td>79</td>
<td>c40t1d0</td>
</tr>
<tr>
<td>0.0</td>
<td>1622.3</td>
<td>0.0</td>
<td>3244.7</td>
<td>0.1</td>
<td>1.2</td>
<td>0.0</td>
<td>0.7</td>
<td>5</td>
<td>79</td>
<td>c5t4d1</td>
</tr>
<tr>
<td>0.0</td>
<td>1589.6</td>
<td>0.0</td>
<td>3179.3</td>
<td>0.1</td>
<td>1.2</td>
<td>0.0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Extended Device Statistics

- SUN all disk(s) -- Read
- SUN all disk(s) -- Write

### Graphs

- I/O Activity Operations/sec: IObench 2/4/6/8/12/24 procs, MPXIO-off
- IObench 2/4/6/8/12/24 procs, MPXIO-off - [Write/s]
Performance Monitoring...

- System and application stats are our friends
- Only raw data – hard! (ex: days -> weeks load view)
- Only graph data – hard! (ex: need exact numbers)
- Command line interface (CLI) is most common
- Data presentation is most painful...
- Time interval is very important for accurate measure
- Too much data = out of control...
- Too few (avg) data = out of detail (problem)...
  > Ex: Avg( temperature ) in hospital is OK
Benchmark Center in 1997...

vmstat
mpstat
iostat
netstat
vxstat
sar
ps
etc...

sed
awk
grep
etc...

plot
Word/Exel
Star Office
Applix, etc...

CPU

etc...
Main Idea...

vmstat
mpstat
iostat
netstat
vxstat
sar
har
psSTAT
etc...

Real Time Timestamped Correlated SQL Database

JDBC, ODBC, Native SQL, Pro*C, etc...
PHP, Perl, Tcl, Python, WebX, etc...
Word/Excel, StarOffice, Applix, etc...
etc...
Wish list for “STAT” tool

- Run on Solaris! (or Linux :-))
- Really small overhead!
- Easy to install
- Easy to use & administrate
- Easy to extend
- Easy to access internal data
- Unlimited history
- Application feed-back
- Without external cost - Why should we pay to analyze our own machines?...
dim_STAT Architecture Overview

Web Server (apache)

MySQL (database)

WebX (cgi-bin)

Collectors

STAT-service

STAT-service

STAT-service
Example Multi-host Analyzing...

dim_STAT Multi-Host Analyzer

CPU %Busy

netLOAD: SUM Packets/s
STAT-service (agent) details

- Controlled & logged access
- On-Demand Start/Stop service
- Listening on TCP/IP port and publishing available stats
- Includes command line kit (**EasySTAT**) to collect data locally and load later (paranoid or very protected sites)
- Default:
  - `vmstat, mpstat, iostat, netLOAD, ProcLOAD, UserLOAD, vxstat, ZoneLOAD, PoolLOAD, ProjLOAD`
- Extended:
  - `MySQL, PostgreSQL, Oracle, JVM, dbSTRESS, lobench`
- and any **New** one you want to add!
Main Page

Database: [New82]

dim_STAT v.8.2

Server Activity

Preferences / Bookmark(s)
Start New Collect
Restart Stopped Collect(s)
Analyze
Report Tool
Stop Active Collect(s)
Load STAT(s) from output file(s)
Save Collect(s) to tar.Z
Restore Collect(s) from tar.Z
Delete/Recycle Collect(s)
Modify Collect(s) attributes
LOG Messages Admin
Add-On STAT(s)

dimitri.kravtchuk@france.sun.com

http://localhost/cgi-bin/WebX.mysql[dim_STAT]/x.welcome
Preferences setting...

- Database
  - New Database: [New82]
  - Use Database: [New82]

- Host Name List
  - dimitri:5000
  - localhost
  - v890/LLG-free
  - v490/laplace.llg
  - uk-server/10.132.0.243

HTTP: `http://dimitri/cgi-bin/WebX.mysql/dim_STAT/x.preferences`
Start New Collect

```
dim_STAT v.8.2
```

Server Activity

![Graph showing server activity with bars for User and System]

- Preferences / Bookmark(s)
- Start New Collect
- Restart Stopped Collect(s)
- Analyze
- Report Tool
- Stop Active Collect(s)
- Load STAT(s) from output file(s)
- Save Collect(s) to tar.Z
- Restore Collect(s) from tar.Z
- Delete/Recycle Collect(s)
- Modify Collect(s) attributes
- LOG Messages Admin
- Add-On STAT(s)

dimitri.kravtchuk@france.sun.com

http://localhost/cgi-bin/WebX.mysql/dim_STAT/x.welcome
Start new STAT collect...

**dim_STAT Start New Collect(s)**

**LED Description:**
- **ok**: STAT-service is running on the host, compatible version
- **degraded**: STAT-service is running on the host, but old version, please, upgrade!
- **bad**: no STAT-service running or host is not accessible...

**NOTE**: your default STAT-service port is 5000

```bash
http://dimitri/cgi-bin/WebX.mysql[dim_STAT/x_client.pl]
```
Start new STAT collect (cont.)

Select stats

**dim_STAT** Start New Collect(s)

- **Collect BaseName:** STAT
- **Stat Title:** Probe Collect
- **Time Interval:** 20 sec.

<table>
<thead>
<tr>
<th>Host</th>
<th>Stat ID</th>
<th>STAT(s)</th>
<th>Client Log Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimitri</td>
<td>6</td>
<td><img src="checkboxes.png" alt="checkboxes" /></td>
<td></td>
</tr>
<tr>
<td>v890</td>
<td>7</td>
<td><img src="checkboxes.png" alt="checkboxes" /></td>
<td></td>
</tr>
<tr>
<td>uk-server</td>
<td>8</td>
<td><img src="checkboxes.png" alt="checkboxes" /></td>
<td></td>
</tr>
</tbody>
</table>

- **Make Collect Script ONLY...**
- **Start STAT(s) Collect Now!!!**

WARNING: Do not delete *STAT* files immediately after collect. Use the *Collect Script* to delete the files.

[![URL](url.png)](http://dimitri/cgi-bin/WebX.mySQL)
Analyze

Database: [New82]

dim_STAT v.8.2

Server Activity

- Preferences / Bookmark(s)
- Start New Collect
- Restart Stopped Collect(s)
- Analyze
- Report Tool
- Stop Active Collect(s)
- Load STAT(s) from output file(s)
- Save Collect(s) to tar.Z
- Restore Collect(s) from tar.Z
- Delete/Recycle Collect(s)
- Modify Collect(s) attributes
- LOG Messages Admin
- Add-On STAT(s)

dimitri.kravtchuk@france.sun.com

http://localhost/cgi-bin/WebX.mysql/dim_STAT/x.welcome
Analyzer interface

- Single or Multi-Host view
  - Single: more detailed, in depth analyze
  - Multi: global view, relative to hosts activity
- Choose Host(s)
- Choose time period
- Choose STATs and presentation mode (text, graph)
- Go!
Example: Multi-Host Analyze

Multi host

hosts, title pattern
Example: Multi-Host Analyze (cont.)

```
<table>
<thead>
<tr>
<th>ID</th>
<th>Host</th>
<th>Title</th>
<th>Started</th>
<th>Interval</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>top-A</td>
<td>Probe new Multi-host (@10.128.4.70)</td>
<td>2007-07-06 21:59:19</td>
<td>15</td>
<td>Finished</td>
</tr>
<tr>
<td>4</td>
<td>top-C</td>
<td>Probe new Multi-host (@10.128.4.74)</td>
<td>2007-07-06 21:59:22</td>
<td>15</td>
<td>Finished</td>
</tr>
</tbody>
</table>
```

**Time period**

- Last 30 minutes (current time)
- Between 2007-09-19 00:00 and 2007-09-19 13:36
- After LOG Message 30 minutes
- Between LOG Messages

**Interval**

- [22:16 2007.07.06] top-C: STOP!

**Database:** [New82]

**Webx Page:**

- [http://dimitri/cgi-bin/WebX.mySQL](http://dimitri/cgi-bin/WebX.mySQL)
Example: Multi-Host Analyze (cont. 2)

Select stats

Values

Per host

Grouped AVG by 4 first/last letters in host name

Grouped SUM by 4 first/last letters in host name

Grouped MAX by 4 first/last letters in host name

Grouped MIN by 4 first/last letters in host name

Note: if value already contains SUM/AVG/etc. grouping, host's aggregate will be ignored.

Graphics

Mode: PNG Image

Style: Bold ContGraph

Background: White

Size: 750 x 206

Title:

Values:

CPU: Usr%

CPU: Sys%

CPU: Busy%

RAM: Free List (KB)

Run queue

Blocked processes

U/O: Read KB/s

http://dimitri/cgi-bin/WebX.mysql

Checks/Uncheck All

Show ALL Host's LOG Messages Matching:

Show Task(s) from ALL Hosts

Refresh every 60 sec.

Start Clean

http://dimitri/cgi-bin/WebX.mysql
Example: Multi-Host Analyze (cont.3)

Database: [New82]

*dim_STAT* Multi-Host Analyzer

**CPU %Busy**

- **top-A**
- **top-C**

**oraEXEC: Exec/sec**

- **top-C**
Example: Single-Host Analyze

Select host, click on STAT

Dim_STAT Analyzer

<table>
<thead>
<tr>
<th>ID</th>
<th>Host</th>
<th>Title</th>
<th>Started</th>
<th>Interval</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>top-A</td>
<td>Probe new Multi-host (@10.128.4.70)</td>
<td>2007-07-06 21:59:19</td>
<td>15</td>
<td>Finished</td>
</tr>
<tr>
<td>4</td>
<td>top-C</td>
<td>Probe new Multi-host (@10.128.4.74)</td>
<td>2007-07-06 21:59:22</td>
<td>15</td>
<td>Finished</td>
</tr>
</tbody>
</table>

Use LogFile Messages from: the same host
Matching: *
Example: Single-Host Analyze (cont.)

**Setup criteria**

Database: [New82]

[dim_STAT] Analyzer


Disk(s)

- c0t10d0
- c0t11d0
- c0t3d0
- c0t9d0
- c14t60x10xA0Cd0
- c14t60x11xB3Ff0
- c14t60x1xB6D4d0
- c14t60x2xB43Ed0
- c14t60x3xB7C4d0
- c14t60x5xB685d0
- nfs1
- nfs2

Inversed Selection

- Select TOP-10 DISK(s)
- Accept ONLY data with Read/s
- Use Select Pattern

Values

- Last 100 measurements
Example: Single-Host Analyze (cont2.)
Bookmarks (or rename-it-as-you-like)

Select stats
Bookmarks (cont.)

Direct Links to other STATs for the same time period...

Database: [New82]

**dim_STAT Analyzer**

Probe new Multi-host (@10.128.4.74) (top-C)

**CPU %Busy:** BMK Workload

- **Cpu Usr%**
- **Cpu Sys%**
- **Cpu Idle%**

**Top-10 Reads/sec Disks:** BMK Workload - [Read/s]

[CPU] [CPU_CrossCalls] [CPU_DL_Switch] [CPU_ThMigration] [FreeMEM] [I/O-KB/s] [I/O-Op/s] [NetByte/s] [NetByteALL/s] [Net_Collis/s] [Net_Error/s] [Net_NoCanout] [Net_Pack/s] [Net_PackALL/s] [ORA_Commit/s] [ORA_Exe/s] [ORA_Sessions] [Paging] [PsScan] [RuncQueue] [SpinMx] [SpinW] [SysCalls] [Top-10_Read/s] [Top-10-BusyDisks] [Top10Busy_Actv] [Top10Busy_SrvTm] [Top10Busy_Wait] [Top10_PrcCpu] [Top10_PrcNumB] [Top10_PrcSysTm] [Top10_PrcUsrTm]
THINK MYSQL SINCE NOW :-)


Add-Ons

- Add-On = a framework to integrate any new stat commands
- Near any stats you want :-)
- Type:
  > Single-Line
  > Multi-Line
- Regular output, well structured
- Integration via Web Interface
Add-Ons

Database: [New82]

*dim_STAT v.8.2*

Server Activity

- Preferences / Bookmark(s)
- Start New Collect
- Restart Stopped Collect(s)
- Analyze
- Report Tool
- Stop Active Collect(s)
- Load STAT(s) from output file
- Save Collect(s) to tar.Z
- Restore Collect(s) from tar.Z
- Delete/Recycle Collect(s)
- Modify Collect(s) attributes
- LOG Messages Admin
- Add-On STAT(s)

dimitri.kravtchuk@france.sun.com

http://localhost/cgi-bin/WebX.mysql/dim_STAT/x.welcome

37
Add-Ons: new stats integration

_dim_STAT - Add-On STAT(s)_

Current database: [New82]

- Integrate New Add-On-STAT
- Save Add-On STAT(s) Description
- Restore Add-On STAT(s) Description
- Delete Add-On STAT(s)

<table>
<thead>
<tr>
<th>Add-On STAT(s) -- Standard Kit</th>
</tr>
</thead>
<tbody>
<tr>
<td>ForkExec</td>
</tr>
<tr>
<td>HPiostat</td>
</tr>
<tr>
<td>HPvmstat</td>
</tr>
<tr>
<td>IObench</td>
</tr>
<tr>
<td>LProcLOAD</td>
</tr>
<tr>
<td>LUsrLOAD</td>
</tr>
<tr>
<td>LcpuSTAT</td>
</tr>
<tr>
<td>LioSTAT</td>
</tr>
<tr>
<td>LnetLOAD</td>
</tr>
<tr>
<td>LpsSTAT</td>
</tr>
<tr>
<td>Lvmstat</td>
</tr>
<tr>
<td>MEMSTAT</td>
</tr>
<tr>
<td>PoolLOAD</td>
</tr>
<tr>
<td>ProcLOAD</td>
</tr>
<tr>
<td>ProjLOAD</td>
</tr>
<tr>
<td>T3stat</td>
</tr>
<tr>
<td>TaskLOAD</td>
</tr>
<tr>
<td>TaskLOAD</td>
</tr>
</tbody>
</table>
Add-On Example

$ MyPerf.sh 5

<table>
<thead>
<tr>
<th>Calls/s</th>
<th>Users</th>
<th>AvgTM</th>
<th>MaxTM</th>
<th>Waits</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>25</td>
<td>5.5</td>
<td>75.3</td>
<td>5</td>
</tr>
<tr>
<td>179</td>
<td>29</td>
<td>5.9</td>
<td>74.1</td>
<td>7</td>
</tr>
<tr>
<td>344</td>
<td>43</td>
<td>6.3</td>
<td>98.2</td>
<td>11</td>
</tr>
<tr>
<td>153</td>
<td>31</td>
<td>5.7</td>
<td>81.1</td>
<td>8</td>
</tr>
</tbody>
</table>

... dim_STAT Integrate New Add-On STAT ...

$
Add-On Example (2)

Add-On STAT Name: MyPerf

Description: MyPerf Statistic(s)

Shell Command: MyPerf %i

where: %h - hostname, %i - interval(sec.), %p - parameter(s)

Ignore Line(s): matching pattern(s): *Calls*

<table>
<thead>
<tr>
<th>Database Column Name</th>
<th>Data Type</th>
<th>Column# in input Line</th>
<th>Short Name</th>
<th>Full Name</th>
<th>Use in Multi-Host Analyze</th>
</tr>
</thead>
<tbody>
<tr>
<td>calls_sec</td>
<td>Float</td>
<td>1</td>
<td>Calls/sec</td>
<td>MyPerf Calls/sec</td>
<td>Yes</td>
</tr>
<tr>
<td>users</td>
<td>Float</td>
<td>2</td>
<td>Users</td>
<td>MyPerf Users</td>
<td>Yes</td>
</tr>
<tr>
<td>avgresptm</td>
<td>Float</td>
<td>3</td>
<td>AvgRespTM</td>
<td>MyPerf AvgRespTM</td>
<td>No</td>
</tr>
<tr>
<td>maxresptm</td>
<td>Float</td>
<td>4</td>
<td>MaxRespTM</td>
<td>MyPerf MaxRespTM</td>
<td>No</td>
</tr>
<tr>
<td>waits</td>
<td>Float</td>
<td>5</td>
<td>Waits</td>
<td>MyPerf Waits</td>
<td>No</td>
</tr>
</tbody>
</table>
MySQL Information sources

• SHOW status / innodb status / mutex status !!
• PROCESSLIST
• INFORMATION SCHEMA !!
• PERFORMANCE SCHEMA !!
• INNODB LOKCS!! / INNODB LOCK WAITS !!
• PBXT stats !!
• Userstats !!
• Etc. etc. etc...
Already Available Add-Ons for MySQL

- **mysqlSTAT** => detailed data from “show status”
- **innodbSTAT** => detailed data from “innodb status”
- **innodbIOSTAT** => I/O activity via DTrace (Solaris)
- **mysqlLOAD** => short data from “show status”
- **innodbMUTEX** => mutex waits from “mutex status”
Example: innodbSTAT

```bash
# /etc/STATsrv/bin/innodbSTAT.sh 5

STAT-name                     STAT-value
db-server-online               1
...                             
History-len                    0
OS-file-reads                  81833
OS-file-reads/sec              0.000000
OS-file-writes                 212
OS-file-writes/sec             0.000000
OS-fsyncs                      212
OS-fsyncs/sec                  0.000000
Insert-Buffer-size             1
Insert-Buffer-free-len         0
Insert-Buffer-seg-size         2
Insert-Buffer-merges           0
Hash-Index-size                36958711
Hash-Index-buffers             7577
Log-age                        2194.000000
Log-age-MB                     0.002092
Checkpoint-age                 16462.000000
Checkpoint-age-MB              0.015699
Log-writes                     177
Log-writes/sec                 0.000000
...
```
**Example: innodbSTAT (2)**

**dim_STAT Analyzer**

**MySQL Load... (hydra01)**

InnoDB Statistic(s) Started: 2010-04-12 14:51:47 Timeout: 10 sec.

<table>
<thead>
<tr>
<th>Name(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Additional-pool-MB</td>
</tr>
<tr>
<td>Buffer-pool-hit</td>
</tr>
<tr>
<td>Buffer-pool-size</td>
</tr>
<tr>
<td>Checkpoint-age</td>
</tr>
<tr>
<td>Checkpoint-age-MB</td>
</tr>
<tr>
<td>Database-pages</td>
</tr>
<tr>
<td>db-server-online</td>
</tr>
<tr>
<td>Dictionary-memory-KB</td>
</tr>
<tr>
<td>Free-buffers</td>
</tr>
<tr>
<td>Hash-Index-buffers</td>
</tr>
<tr>
<td>Hash-Index-size</td>
</tr>
<tr>
<td>History-len</td>
</tr>
<tr>
<td>Insert-Buffer-free-len</td>
</tr>
<tr>
<td>Insert-Buffer-merges</td>
</tr>
<tr>
<td>Insert-Buffer-seg-size</td>
</tr>
<tr>
<td>Insert-Buffer-size</td>
</tr>
<tr>
<td>Log-age</td>
</tr>
<tr>
<td>Log-age-MB</td>
</tr>
<tr>
<td>Log-writes</td>
</tr>
<tr>
<td>Log-writes-MB</td>
</tr>
<tr>
<td>Log-writes-sec</td>
</tr>
<tr>
<td>Log-writes/sec</td>
</tr>
<tr>
<td>Modified-db-pages</td>
</tr>
<tr>
<td>Mutex-OS-waits</td>
</tr>
<tr>
<td>Mutex-OS-waits/sec</td>
</tr>
<tr>
<td>Mutex-rounds</td>
</tr>
<tr>
<td>Mutex-rounds/sec</td>
</tr>
<tr>
<td>Mutex-spin-waits</td>
</tr>
<tr>
<td>Mutex-spin-waits/sec</td>
</tr>
<tr>
<td>Old-db-pages</td>
</tr>
<tr>
<td>OS-file-reads</td>
</tr>
<tr>
<td>OS-file-reads/sec</td>
</tr>
<tr>
<td>OS-file-writes</td>
</tr>
<tr>
<td>OS-file-writes/sec</td>
</tr>
<tr>
<td>OS-fsyncs</td>
</tr>
<tr>
<td>OS-fsyncs/sec</td>
</tr>
<tr>
<td>Pages-create</td>
</tr>
<tr>
<td>Pages-create/sec</td>
</tr>
<tr>
<td>Pages-read</td>
</tr>
<tr>
<td>Pages-read/sec</td>
</tr>
<tr>
<td>Pages-write</td>
</tr>
<tr>
<td>Pages-write/sec</td>
</tr>
<tr>
<td>Queries-in-queue</td>
</tr>
<tr>
<td>Queries-inside</td>
</tr>
<tr>
<td>Rows-delete</td>
</tr>
<tr>
<td>Rows-delete/sec</td>
</tr>
<tr>
<td>Rows-insert</td>
</tr>
<tr>
<td>Rows-insert/sec</td>
</tr>
<tr>
<td>Rows-read</td>
</tr>
<tr>
<td>Rows-read/sec</td>
</tr>
<tr>
<td>Rows-update</td>
</tr>
<tr>
<td>Rows-update/sec</td>
</tr>
<tr>
<td>RW-excl-OS-waits</td>
</tr>
<tr>
<td>RW-excl-OS-waits/sec</td>
</tr>
<tr>
<td>RW-excl-spins</td>
</tr>
<tr>
<td>RW-excl-spins/sec</td>
</tr>
<tr>
<td>RW-shared-OS-waits</td>
</tr>
<tr>
<td>RW-shared-OS-waits/sec</td>
</tr>
<tr>
<td>RW-shared-spins</td>
</tr>
<tr>
<td>RW-shared-spins/sec</td>
</tr>
<tr>
<td>Total-memory-MB</td>
</tr>
<tr>
<td>Wait-Array-reservations</td>
</tr>
<tr>
<td>Wait-Array-reservations/sec</td>
</tr>
<tr>
<td>Wait-Array-signals</td>
</tr>
<tr>
<td>Wait-Array-signals/sec</td>
</tr>
</tbody>
</table>

More...
Example: innodbSTAT (3)

<table>
<thead>
<tr>
<th>Bookmark(s)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBS_Clients</td>
<td>dbSTRESS ACTIVE! Clients Number</td>
</tr>
<tr>
<td>DBS_ReadTM</td>
<td>dbSTRESS Read Resp.Time</td>
</tr>
<tr>
<td>DBS_Trans/s</td>
<td>dbSTRESS TPS</td>
</tr>
<tr>
<td>DBS_WriteTM</td>
<td>dbSTRESS Write Resp.Time</td>
</tr>
<tr>
<td>InnoDB_CheckpointAge</td>
<td>InnoDB Checkpoint Age</td>
</tr>
<tr>
<td>InnoDB_CheckpointAge_MB</td>
<td>InnoDB Checkpoint Age (MB)</td>
</tr>
<tr>
<td>InnoDB_HistLen</td>
<td>InnoDB History Length</td>
</tr>
<tr>
<td>InnoDB_LogAge</td>
<td>InnoDB Log Age</td>
</tr>
<tr>
<td>InnoDB_LogAge_MB</td>
<td>InnoDB Log Age (MB)</td>
</tr>
<tr>
<td>InnoDB_Log_MB/s</td>
<td>InnoDB Log Write MB/s</td>
</tr>
<tr>
<td>InnoDB_Log_W/s</td>
<td>InnoDB Log Writes/sec</td>
</tr>
<tr>
<td>InnoDB_Mutex</td>
<td>InnoDB Mutex Stats</td>
</tr>
<tr>
<td>InnoDB_MutexWt</td>
<td>InnoDB Mutex Waits</td>
</tr>
<tr>
<td>InnoDB_OS_file/s</td>
<td>InnoDB OS-file Operations/sec</td>
</tr>
</tbody>
</table>
Example: innodbSTAT (4)
Example: innodbMUTEX
Analyzing Partitions...

- Using partitions to lower Index Mutex contention..
- Hm... No difference??..
Analyzing Partitions... (2)

• New contention! :-) => update stats log
Analyzing Partitions... (3)

- Fixed! :-)

---

```
dbSTRESS TPS: MySQL 5.5.4 / 5.5.4+3partitions / +fixed

12/04/10 13/04/10 02:39 05:03

ALL-tps -- TR_all
ALL-tps -- TR_Read
ALL-tps -- TR_Write

InnoDB Top-7 Mutex Waits/s: MySQL 5.5.4 / 5.5.4+3partitions / +fixed - [os_waits/s]

btr/btr0sea.c:181
buf/buf0buf.c:1122
combined_buf/buf0buf.c:820
log/log0log.c:769
srv/srv0srv.c:973
dict/dict0dict.c:1584
dict/dict0dict.c:634
```
Next Step...

Customer oriented?
Report Tool

Database: [New82]

dim_STAT v.8.2

Server Activity

[Graph showing server activity with bars for User and System]

- Preferences / Bookmark(s)
- Start New Collect
- Restart Stopped Collect(s)
- Analyze
- Report Tool
- Stop Active Collect(s)

- Load STAT(s) from output file(s)
- Save Collect(s) to tar.Z
- Restore Collect(s) from tar.Z
- Delete/Recycle Collect(s)
- Modify Collect(s) attributes
- LOG Messages Admin

- Add-On STAT(s)

dimitri.kravtchuk@france.sun.com

http://localhost/cgi-bin/WebX.mysql/dim_STAT/x_welcome
Reporting needs...

- Several versions of the same story:
  > Confidentiality level (Customer, Partner, Internal, etc.)
  > Language (English, French, German, etc.)
- Copy & Paste graphs is too painful...
- Automate whatever possible...
- Simplified formatting, teamwork and publishing...
- etc...
Report Tool

- Web Interface, Database driven
- Report Import / Export
- **Mostly Automated** Processing
- Supported data types:
  > dim_STAT Collect
  > SysINFO
  > Text, HTML
  > Image, Binary
  > HTAR (tar archive with HTML documents)

- **Several Views** of the same report
- HTML and/or PDF output
Example of Report page...

**Test: WebMethods replacing SPL**

The same tests were executed using WebMethods. But due lack of time, further optimization for WebMethods was not possible and there was no test finished without error... So, don’t take care comparing Cordapix actions between WebMethods and SPL solutions; all actions are going faster with WebMethods because they are never arrived till the end... :)

**Full Detailed STATs for every Final Test and some other**

- Test IS1M 1000us
- Rf Test 600us SPL
- Final Test 600us SPL
- Test 600us WebMethod
- Final 500us WebMethod
- Final Test Consultation 600us SPL
- Test Consultation 600us WebMethod
- Test Consultation 600us WebM Software

**Additional Tests**

OK, we have to admit we are really crazy...

Discussing about obtained results everybody was surprised we used 16CPU for every Siebel server instead of initially recommended 8CPU. Looking on the CPU usage with 16 processes (70%) it’s easy to understand there is no way to keep the same workload with 8. But what about 12CPU?...

As we don’t like to speak a lot in Benchmark, we just got in few minutes an agreement from everybody to start few additional tests: the most heavy (factorization) using 13 and after 8CPU on Siebel machines... :)

And even joking that this test we are running ”just to make a pleasure for Sun team”, everybody well understand that this test will put the last dot in the final platform and architecture design.
Report Tool architecture...

- **Data Type**
  - HTML
  - TXT
  - BIN
  - IMG
  - dim_STAT
  - SysINFO

- **Attribute(s)**
  - Confidentiality
    - Customer
    - Sun Internal
    - Bench Internal
    - etc...
  - LANGUAGE
    - English
    - French
    - German
    - Japanese
    - etc...
  - Etc...

- **Note**
  - Title, Owner info(s)
  - Attachment
  - Comments
Report Working cycle

- Report template
- Editing Report contents
- Export.tar.Z
- Preview
- Publish
- PDF
- Set Presentation options & Attributes criteria
- Report.tar.Z
Report: dim_STAT Collect

- **No** copy & paste!... :-)
- Just open new *dim_STAT Note*:
  > choose server + DB name
  > choose STATs/ Bookmarks
  > choose time periods
  > **GO!**

- Take your coffee and leave the machine working for you (finally)...
Thu, 14/Dec

Main events:
- Oracle: customer discovered a bug that caused a...-
- restart tests
- Test400ref+70promo:...-
- Test400ref alone: ok!
- Test100promo alone:...-
- Sun appserver: continues...

SysINFO_neel @

Warning: RAM Usage:

- 57 GBytes used
- 15 GBytes available

SysINFO_fourrier @

Stats: High load tests, etc.

- [2006-12-14]
- [2006-12-14]
- [2006-12-14]
Other pre-integrated stuff...

- dbSTRESS -- database stress framework
  > check your database performance issues ahead

- IObench -- I/O stress framework
  > test your I/O subsystem before database deployment
dbSTRESS Scenario

db_STRESS Test Scenario Generator v.1.7

**Scenario Name:** Stress Test

**DB Vendor:** MySQL

- **DB Name:** dbSTRESS (format: dbname[@Host[:Port]] or SID)
- **DB User:** dim
- **DB User's Password:** dimitri
- **DB Size:** 10000000 (number of OBJECT(s))

**Scenario(s):**

- **Concurrent Session(s) Load Level(s):** 1 2 4 8 16 32 64 128 256 (max: 2000)
- **Transaction Reads-per-Write Ratio(s):** 0 1 10 1000 (0 means Read-Only)

- **Timeout between transactions (ms):** 0 (think time)
- **Duration of each Load Level:** 120 seconds
- **Keep Max Load Level during:** 300 seconds
Coming features with v.9 ...

- Move to 5.5 :-)
- Customized Alerts
- Health Analyzer
- Dash board
- etc...
Alerts Management (prototype)

*dim_STAT* Alert Management

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Description</th>
<th>Frequency</th>
<th>Next Check</th>
<th>Alert Conditions</th>
<th>Subscriber’s ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CPU Add</td>
<td>Add CPU on Hot Server</td>
<td>300 sec.</td>
<td>ASAP</td>
<td>Idle &lt; 20 &amp;&amp; NCPU &lt; 40</td>
<td>6, 7</td>
</tr>
<tr>
<td>2</td>
<td>CPU Remove</td>
<td>Remove CPU from Cool Server</td>
<td>300 sec.</td>
<td>ASAP</td>
<td>Idle &gt; 80 &amp;&amp; NCPU &gt; 16</td>
<td>6, 7</td>
</tr>
</tbody>
</table>

- Any stat values may be involved and then Named
- Condition formula with Names executed to know if Alert conditions were matched
- Any defined Alert may be imported to the “Known problems” database and then help others to identify a problem...
Key Notes

- GPL/Freeware!
- Solaris/SPARC (since 2.6 and higher)
- Solaris 10 x86, Linux/x86
- 5 min. install & run
  > all software is pre-bundled!
- 60MB disk space, 0.1% CPU usage

Download:

- Internet: http://dimitrik.free.fr
- Mirrors??
Some Facts

- Over 12 Years! :-)
- Sun: ToolsCD, most BMK Centers, etc.
- Alcatel (World-wide and pre-installed by CRS)
- Orange, EDF, Renault, Nokia, Motorola, etc...
- Downloads...

“your program is one of Sun's best kept secrets”
Tom Alling, tecsol.com
Contact

• Dimitri KRAVTCUK
dimitri@sun.com

Sun Solution Center
Paris, FRANCE

N.B. A Tool will never replace an Engineer!!!