Swingbench

Dominic Giles
Oracle UK.
Agenda

• About the author
• Brief History
• Overview of Swingbench
• Swingbench Family
• Supplied Benchmarks
• Running and Installing Swingbench
• Other Useful Tools
• What Next
• Questions
About the Author
About the Author

- Dominic Giles
- http://www.dominicgiles.com
- 18 years at Oracle UK
- Database specialist
- The UK “Database Solutions” team
Brief History
Brief History

- Development started out of a need to demonstrate a load on Real Application Clusters
- Many of the commercial database solutions were simply an overkill and too difficult to obtain
- It started as a throw away project
- Swingbench 2.2 is currently production
- Swingbench 2.3 is now functionally complete.
Overview of Swingbench
Overview of Swingbench

• Simple to use load generator for Oracle databases
• Three different front ends
  • Swingbench
  • Charbench
  • Minibench
• Four different benchmarks.
• Trivial to add your own transactions.
• Written in Java so should run on any platform with a 1.5 JVM.
Overview of Swingbench

• Typically used as a standalone load generator
• However a Coordinator process allows multiple load generators to work together.
• ClusterOverview aggregates all of the results together.
• It's free....
Overview of Swingbench

• Simple Architecture

jdbc based transactions

optional system monitor reports on cpu and disk

Single instance database
Overview of Swingbench

• Advanced Architecture
Swingbench Family
Swingbench Family

- Several front ends to the same kernel
- Configurations are saved to an xml based file
  - XML schema included
- Results are typically saved in xml to a results file
- Choose the frontend that suits your needs
Swingbench Family

Swingbench

• Rich graphical front end
• Real time charting
• Editing of all configuration parameters (new in 2.3)
• Useful for demonstrations
Swingbench Family

Minibench

• Small footprint graphical front end
• Simple real time charting
• Results of benchmark are placed in file
• Useful for accessing in the impact of an operation such as a backup.
Swingbench Family

Charbench

- Character based
- Capable of vmstat/sar like output
- Wide range of command line options
- Used in “serious” benchmarking tests
Usage: parameters:
-D <variable=value> use value for given environment variable
-a run automatically
-be <stopafter> end recording statistics after. Value is in the form hh:mm
-bs <startafter> start recording statistics after. Value is in the form hh:mm
-c <filename> specify config file
-co <hostname> specify/override coordinator in configuration file.
-com <comment> specify comment for this benchmark run (in double quotes)
-cpuloc <hostname> specify/override location of the cpu monitor.
-cs <connectstring> override connect string in configuration file
-dt <drivertype> override driver type in configuration file. Value is either "thin" or "oci"
-h,--help print this message
-i run interactively (default)
-ld <milliseconds> specify/override the logon delay (milliseconds)
-max <milliseconds> override maximum think time in configuration file
-min <milliseconds> override minimum think time in configuration file
-p <password> override password in configuration file
-r <filename> specify results file
-rr specify/override refresh rate for charts in secs
```
[dgiles@macbook bin]$ ./charbench -a -v users,tpm,tps,cpu,disk -min 10 -max 100 -rt 0:05
Author: Dominic Giles
Version: 2.3.0.261

Results will be written to results.xml.

<table>
<thead>
<tr>
<th>Time</th>
<th>Users</th>
<th>TPM</th>
<th>TPS</th>
<th>User</th>
<th>System</th>
<th>Wait</th>
<th>Idle</th>
<th>Bi</th>
<th>Bo</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:48:09 PM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1:48:11 PM</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1:48:12 PM</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>21</td>
<td>10</td>
<td>1</td>
<td>69</td>
<td>10</td>
<td>73</td>
</tr>
<tr>
<td>1:48:13 PM</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>21</td>
<td>10</td>
<td>1</td>
<td>69</td>
<td>10</td>
<td>73</td>
</tr>
<tr>
<td>1:48:14 PM</td>
<td>5</td>
<td>26</td>
<td>7</td>
<td>21</td>
<td>10</td>
<td>1</td>
<td>69</td>
<td>10</td>
<td>73</td>
</tr>
<tr>
<td>1:48:15 PM</td>
<td>5</td>
<td>26</td>
<td>7</td>
<td>21</td>
<td>10</td>
<td>1</td>
<td>69</td>
<td>10</td>
<td>73</td>
</tr>
<tr>
<td>1:48:16 PM</td>
<td>5</td>
<td>62</td>
<td>17</td>
<td>21</td>
<td>10</td>
<td>1</td>
<td>69</td>
<td>10</td>
<td>73</td>
</tr>
<tr>
<td>1:48:17 PM</td>
<td>5</td>
<td>62</td>
<td>17</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>82</td>
<td>16</td>
<td>150</td>
</tr>
<tr>
<td>1:48:18 PM</td>
<td>5</td>
<td>83</td>
<td>18</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>82</td>
<td>16</td>
<td>150</td>
</tr>
<tr>
<td>1:48:19 PM</td>
<td>5</td>
<td>83</td>
<td>18</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>82</td>
<td>16</td>
<td>150</td>
</tr>
<tr>
<td>1:48:20 PM</td>
<td>5</td>
<td>112</td>
<td>13</td>
<td>9</td>
<td>4</td>
<td>5</td>
<td>82</td>
<td>16</td>
<td>150</td>
</tr>
<tr>
<td>1:48:21 PM</td>
<td>5</td>
<td>112</td>
<td>13</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>87</td>
<td>0</td>
<td>83</td>
</tr>
<tr>
<td>1:48:22 PM</td>
<td>5</td>
<td>133</td>
<td>9</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>87</td>
<td>0</td>
<td>83</td>
</tr>
<tr>
<td>1:48:23 PM</td>
<td>5</td>
<td>133</td>
<td>9</td>
<td>8</td>
<td>5</td>
<td>0</td>
<td>87</td>
<td>0</td>
<td>83</td>
</tr>
<tr>
<td>1:48:24 PM</td>
<td>5</td>
<td>155</td>
<td>11</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>84</td>
<td>12</td>
<td>125</td>
</tr>
<tr>
<td>1:48:25 PM</td>
<td>5</td>
<td>155</td>
<td>11</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>84</td>
<td>12</td>
<td>125</td>
</tr>
</tbody>
</table>
```
Swingbench Family

ClusterOverview

- Rich graphical front end
- Real time charting
- Typically used to test Oracle Real Application Clusters
- Can be used to coordinate large loads against large SMP machines.
Whats New in 2.3

• Updated UI for Minibench and Swingbench
• Improved overview chart
• TimesTen support
• Benchmark windows
• Jobs
• More command line options
• DSS benchmark
• Faster benchmark schema creation
Quick demo of Swingbench
Supplied Benchmarks
Supplied Benchmarks

- Swingbench comes with four benchmarks
- Configuration files located in the sample directory

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Description</th>
<th>Read/Write Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>OrderEntry</td>
<td>Classic Order Entry Benchmark. TPC-C Like</td>
<td>60/40</td>
</tr>
<tr>
<td>Calling Circle</td>
<td>Telco based self service application</td>
<td>70/30</td>
</tr>
<tr>
<td>Stress Test</td>
<td>Simple Insert / Update / Delete / Select benchmark</td>
<td>50/50</td>
</tr>
<tr>
<td>Sales History</td>
<td>DSS benchmark</td>
<td>100/0</td>
</tr>
</tbody>
</table>
Benchmarks

Order Entry

• TPC-C Like.
• Uses Oracle10g's “oe” schema as its basis
• Grows over time
• Does not require pre-generation of data for the benchmark.
• Stored procedure and native code versions
• Choice of using partitioning
• Maximum size 100GB*

* If using the order entry creation wizard
Welcome to the Order Entry Install Wizard

This wizard will walk you through the steps to install a schema for the order entry benchmark. You will need a logon with DBA privileges to create the needed tablespace, users, tables etc.

Author: Dominic Giles
Email: dominic.giles@oracle.com

* Wizards can be run in “lights out” (character) mode
Benchmarks
Calling Circle

- Telco based self service benchmark
- Relatively static in size
- Requires the generation of a new set of data files for each new benchmark run
- The schema will eventually become “worn out” and will need to be rebuilt
- Maximum size 100GB

* If using the order entry creation wizard
* Wizards can be run in “lights out” (character) mode
Benchmarks

Stress Test

- Simplest of all the benchmarks
- Performs simple Insert/Update/Delete/Select operations
- Schema created at run time
- Useful for quick tests
- Can be run against TimesTen

* If using the order entry creation wizard
Benchmarks

DSS

• A Data Warehousing class benchmark
• Still under testing
• Based on the Oracle10g “Sales History” schema
• No wizard at present
  • Priority to build one
• Uses “Datagenerator” to populate the database.
  • Scripts ship with Datagenerator
• Maximum size limited by disk and time
Benchmarks

Developing your own

• Swingbench ships with a “blank” benchmark that calls a stored procedure which can be modified to include your own transactions
• The Java source for all the transactions is shipped with swingbench.
  • An “Ant” script will compile your new or modified transactions
• Simple Java API allows for the creation of new transactions
Running and Installing Swingbench
Running and Installing Swingbench

- Download from

http://www.dominicgiles.com

- Installation requires the modification of either the swingbench.env file or swingbenchenv.bat file
- Simply set the SWINGHOME and JAVAHOME to reflect your environment
- Linux/Unix executables are in the bin directory
- Windows are in the winbin directory
Running and Installing Swingbench

- Swingbench, Minibench and Charbench can all override settings inside of the configuration file from the command line.

$> ./swingbench -c sample/ccconfig.xml -rt 1:30 -a

- You'll need to use double quotes on some commands on Windows

C:\> charbench.bat -a -v "users,tpm,tps" -rt 0:15
Running and Installing Swingbench

• It's possible to compare the results of several runs with the bmcompare tool

$> ./bmcompare -r results.xml,results0001.xml

• This will create a html file with a comparison of several runs
Other Useful Tools
Other Useful Tools

- To support various aspects of current and future functionality, TraceAnalyzer and Datagenerator came into existence.
- Datagenerator is designed to reverse engineer simple schemas and generate dummy data.
- TraceAnalyzer is designed to parse SQL trace files.
  - In future it will create transactions for Swingbench.
Datagenerator

Table

Name: SALES_Q3_1998
Target Table: 
Row Count: 67,320

Model Parent Child Relationship

Parent Table: COUNTRIES
Parent Table's Column: COUNTRY_ID
Child's Column: PROD_ID
Minimum Relationship Count: 1
Maximum Relationship Count: 1
Reset Child Column:

Retrieve Data From Database:
### TraceAnalyzer

#### SQL Summary

<table>
<thead>
<tr>
<th>SQL</th>
<th>Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. SELECT calendar_year, calendar_month, calendar_quarter, number, SUM(amount_sold) FROM sales, products, customers, countries WHERE sales_time_id = times.time_id AND sales.prod_id = products.prod_id</td>
<td></td>
</tr>
<tr>
<td>2. SELECT channels.channel_desc, calendar_month, calendar_quarter, number, SUM(amount_sold) FROM sales, products, customers, countries WHERE sales_time_id = times.time_id AND sales.prod_id = products.prod_id</td>
<td></td>
</tr>
<tr>
<td>3. SELECT calendar_year, calendar_month, calendar_quarter, number, SUM(amount_sold) FROM sales, products, customers, countries WHERE sales_time_id = times.time_id AND sales.prod_id = products.prod_id</td>
<td></td>
</tr>
<tr>
<td>4. SELECT calendar_year, calendar_month, calendar_quarter, number, SUM(amount_sold) FROM sales, products, customers, countries WHERE sales_time_id = times.time_id AND sales.prod_id = products.prod_id</td>
<td></td>
</tr>
</tbody>
</table>

#### Timings in Microseconds

<table>
<thead>
<tr>
<th>Operation</th>
<th>Count</th>
<th>Elapsed</th>
<th>CPU</th>
<th>Physical</th>
<th>Consistent</th>
<th>Current</th>
<th>Rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parse</td>
<td>299</td>
<td>1,225,368</td>
<td>200,010</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Execute</td>
<td>299</td>
<td>1,095,215</td>
<td>232,012</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Fetch</td>
<td>149,263,115</td>
<td>45,683,080</td>
<td>44,644</td>
<td>6,261,745</td>
<td>44,644</td>
<td>6,261,745</td>
<td>0</td>
</tr>
<tr>
<td>Total</td>
<td>150,561,283</td>
<td>45,906,163</td>
<td>44,644</td>
<td>6,261,745</td>
<td>44,644</td>
<td>6,261,745</td>
<td>0</td>
</tr>
</tbody>
</table>

### Ratings Key

- ★ Largest Elapsed Time
- ★★ Largest consumer of CPU resource
- ★★★ Highest number of physical I/O
- ★★★★★ Highest number of Consistent I/O
What Next?

- Swingbench 2.3 is functionally complete
- Testing, Bug Fixing, Documentation, Viewlets
- Testing of the DSS schema at large scale
What Next? Swingbench 2.4

- Wizard for the DSS benchmark
- TimesTen/Oracle Benchmark
- Application server based version of Swingbench
- TraceAnalyzer improvements
- Merge ClusterOverview with Swingbench
Questions?