Characterizing Workload thru an Oracle Database

Larry Klein (larry.klein@hotsos.com)
Hotsos Enterprises, Ltd.
Long Island Oracle User’s Group
Tuesday June 27, 2006

Presenter’s Background

Larry Klein
- 18+ years, IBM mainframe performance and capacity planning
- 12+ years, Oracle performance and capacity planning
- VP of Consulting, Hotsos Enterprises, Ltd.
Hotsos Enterprises, the company...

- Thought leadership
  - Optimizing Oracle Performance
  - www.hotsos.com Library
  - Method R

- Products
  - Hotsos Profiler
  - Laredo
  - HAWCS for eBusiness
  - SQL Test Harness

- Services
  - 1-week performance assessment
  - On-site consulting
  - Remote consulting

- Education
  - Oracle performance curriculum
  - Public and private events
  - Hotsos Symposium

Presentation Agenda

- An Early Case Study
- The Problem – Different People, Different Perspectives
- Seeking Common Ground - The “System” as a Factory
- Measuring the Factory’s Activities
- Another Case Study
- Questions?
A Case Study

<table>
<thead>
<tr>
<th>Challenge</th>
<th>Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Client XYZ Company</td>
<td>• Tune Logical Reads</td>
</tr>
<tr>
<td>• Custom Order Entry Application</td>
<td>• Identify/Trace Work</td>
</tr>
<tr>
<td>• Application not meeting needs of the</td>
<td>• Measure and report progress</td>
</tr>
<tr>
<td>business</td>
<td>• 5 week effort</td>
</tr>
<tr>
<td>• Database Server max’ed out</td>
<td></td>
</tr>
</tbody>
</table>

Project Progress Report to the Client’s CIO

**Total Daily Logical Read 86% Reduction, over 5 Weeks**
Client Feedback…

Client CIO, “Good Work, but What does it Mean to Me???”

How to Show Progress in Relevant Terms?

This is what I “knew” about the past 5 weeks:
• Custom “Order Entry” Application
• Orders are Everything
• Users Enter Orders as Fast as the System Permits
• Huge Order Entry Backlog – much latent demand

This is what I learned during the 5 weeks:
• Detailed Traces
• Data Model
• Mounds of Database and System-level Statistics
• How the Application “did its thing”

Ah – Hah!!!
Case Study's Alternative Ways to Report Success

**Daily Order Volume**

- Aug-95
- Sep-95
- 0
- 1000
- 2000
- 3000
- 4000
- 5000
- 6000

500% Increase
Over 5 weeks

**Cost per Order**

- Aug-95
- Sep-95
- 0
- 10
- 20
- 30
- 40
- 50

40:1 “Betterment”
Over 5 weeks

Client bought me dinner!!!

---

Lesson Learned - Different Views, Different Perspectives

**Business Mgr's, CIO's, End Users**

- availability
- return on investment
- transactions as “orders”

**Tech Support DBA, Admins, …**

- logical reads
- disk response time
- CPU utilization
- transactions as “commits”

System And its Health ???
Seeking Common Ground - the “System” as a Factory

Workers
- clock in
- perform tasks

Assembly Line
- enables workers
- consumes power
- transports WIP

Widgets
- outputs of tasks

The System as a Factory – Key Metrics

<table>
<thead>
<tr>
<th>Measured Attribute</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Workers</td>
<td>The “demanders” of services</td>
</tr>
<tr>
<td>Widgets</td>
<td>The “outputs” of services</td>
</tr>
<tr>
<td>Assembly Line Costs</td>
<td>The costs to support the Workers producing Widgets</td>
</tr>
<tr>
<td>Cost per Widget</td>
<td>A relative measure of Efficiency</td>
</tr>
</tbody>
</table>
A Factory Consisting of an Application and Oracle

Assembly Line Costs

(LIO, PIO)

Workers

Client Tool or Tier

Widgets

• Orders
• GL JE lines
• Checks
• …

• Online Users
• Batch Jobs
• Interfaces
• ETL’s
• 3rd party tools
• …

Once Measured, then Managed…

Factory Activities by Day, Normalized

Date

Normalized Percentages

Total Workers

Total Widgets

Total Assembly Line Costs
But What do you Do with this Capability?

The Case of…
Some Days Performance is Good,
But on Other Days…

Case Study #1 – A Past Day’s Anomaly – January 29?

Cost per Widget by Day
Pretty Normal Worker Pattern…

Workers by Day

Jan 29 Weekend Cost much higher than Next Weekend

Assembly Line Costs by Day
Jan 29 Cost Dominated by non-COMMON id Worker(s)

Assembly Line Cost by Worker Type

- Jan 29 Cost Dominated by non-COMMON id Worker(s)

Drilldown to non-COMMON Workers that Day...

- Go Find and Talk with WILLFLE!!!
- On Jan 29 WILLFLE ran for the first time:
  - a new, monthly, ad-hoc report
  - that never went thru “code review”
### For Workers

<table>
<thead>
<tr>
<th>Questions</th>
<th>Common Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Who are my workers?</td>
<td>• 30% of yesterday’s workers were not considered in the sizing/buy</td>
</tr>
<tr>
<td>• Who are my most costly workers?</td>
<td>• That adhoc user Will cost me 45% of all yesterday’s LIO’s</td>
</tr>
</tbody>
</table>

### For the Assembly Line

<table>
<thead>
<tr>
<th>Questions</th>
<th>Common Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>• How were a past day’s costs compared to the trend?</td>
<td>• Spiked due to the following 12 batch jobs and 5 online forms that need to be evaluated and optimized…</td>
</tr>
<tr>
<td>• How do daily costs attribute to application modules?</td>
<td>• 52% of yesterday’s cost is charged to the Order Entry module</td>
</tr>
</tbody>
</table>
How can you use the Factory Measurements?

Data Analysis answers many Questions

For Widgets

<table>
<thead>
<tr>
<th>Questions</th>
<th>Common Answers</th>
</tr>
</thead>
<tbody>
<tr>
<td>• How was yesterday’s activity compared to the trend?</td>
<td>• It was a busy day – month end – 37% more widget activity than average</td>
</tr>
<tr>
<td>• Did yesterday’s stress test fairly mimic production activity?</td>
<td>• No – the test workload fell 78% short of generating transactions at the same rate as production</td>
</tr>
</tbody>
</table>

How to Measure the Factory – Instrumentation!!!

Factory Measurements

Workers
• Tier Logs
• Database Tables – Login, Authentication

Assembly Line
• Database Cost Metrics
  – Statspack
  – Logon Triggers
  – Database “audit session”

Widgets
• Transactions in Tables
### How to Measure the Factory – Instrumentation!!!

#### Factory Measurements

- Sometimes the instrumentation is built-in
  - Oracle eBusinessSuite
    - fn and other tables
- Sometimes the instrumentation can be reaped from tier logs
- Sometimes the instrumentation needs to be built into the application

#### Key Metrics

- **Worker related**
  - osuser, user, program, module, action
  - application user, form name, job name
- **Assembly Line Cost related**
  - logoff_lread, logoff_pread
- **Widget related**
  - new table rows by time interval, for the most important modules and functions
Putting Factory Measurements to Work – Case Study

How much like current “PROD” was the Stress Test of the Upgraded System?

Case Study #2 - How Valid is the Stress Test?

Typical ERP Client Needs to Upgrade, Will the Upgrade Sustain PROD Volumes?

- Clone PROD to TEST, then Upgrade TEST
- Stress TEST
  - Focus on current, important PROD activity from 2-4pm
  - Induce similar activity in TEST for 2 hours, monitor
  - No automated test scripts
  - Execute manual “Day in Life” in TEST to mimic PROD
  - Power users to log in, perform work, submit jobs, reports
- Determine if Day in Life (DIL) TEST came close to PROD
- Decide Upgrade go-live based on TEST “closeness” and TEST system performance
Measuring Day in Life TEST - Workers

<table>
<thead>
<tr>
<th></th>
<th>Workers Batch</th>
<th>Workers Online</th>
<th>Workers Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Count</td>
<td>Run Mins</td>
<td>Count</td>
</tr>
<tr>
<td>Prod 2-4pm</td>
<td>2546</td>
<td>397</td>
<td>3134</td>
</tr>
<tr>
<td>TEST</td>
<td>1990</td>
<td>1663</td>
<td>2669</td>
</tr>
</tbody>
</table>

**Hmmm – TEST**

- *light on Workers*
- *heavy on Batch Runtime Minutes*

Measuring Day in Life TEST - Widgets

<table>
<thead>
<tr>
<th></th>
<th>Application X</th>
<th>Application Y</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Table A</td>
<td>Table B</td>
<td>...</td>
</tr>
<tr>
<td>Prod 2-4pm</td>
<td>41670</td>
<td>1162</td>
<td>55982</td>
</tr>
<tr>
<td>TEST</td>
<td>10496</td>
<td>841</td>
<td>12212</td>
</tr>
</tbody>
</table>

**Hmmm – TEST was light on Widgets**
Measuring Day in Life TEST – Assembly Line Costs

<table>
<thead>
<tr>
<th></th>
<th>Logical Reads (millions)</th>
<th>Physical Reads (millions)</th>
<th>Total Logical Reads/Widgets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>COMMON</td>
<td>OTHER</td>
<td>COMMON</td>
</tr>
<tr>
<td>Prod 2-4pm</td>
<td>486</td>
<td>8</td>
<td>11</td>
</tr>
<tr>
<td>TEST</td>
<td>1038</td>
<td>2</td>
<td>17</td>
</tr>
</tbody>
</table>

Hmmm – TEST
- much higher Assembly Line costs
- much higher Cost per Widget

How Much like PROD was TEST?

Probably, Manual Stress TEST users
- “underloaded” transaction processing
- “overloaded” reporting

TEST was not valid!!!

Upon review, many TEST users
- used TEST as “pedal to the metal”
- disregarded instructions and pacing
- loaded up TEST with favorite longrunning month-end reports
Summary – Measuring the “System” as a Factory

Gives you Power, Proaction:

• Hunt for Anomalies
• Characterize the Workload
  • Know your Demanders
  • “Optimize” the Baseline
  • Plan for Capacity

Questions???
Thank You!

References

