Oracle as a Service

Shared Database Platform

Nicholas Tan
Head of Infrastructure & Platform Solutions

Oracle OpenWorld 2010
September 2010
Commonwealth Bank

The Commonwealth Bank is one of Australia’s leading providers of integrated financial services including retail, business and institutional banking, funds management, superannuation, insurance, investment and broking services. The Bank is one of the largest listed companies on the Australian Stock Exchange.
Introduction

- In 2007 CBA set out to create an Oracle database shared service shared offering for the bank
  - The offering has been highly successful by several measures
  - Oracle-as-a-Service has continued to be developed through several iterations

- In 2010 we have expanded this effort, and are currently implementing a range of Platform-as-a-Service offerings, providing:
  - Self Service Platform Provisioning
  - On Demand Platform Provisioning
  - Mass Standardisation
  - Better Utilization of Computing Resources
  - Utility Chargeback Models
  - Multi-provider, hybrid Cloud deployment topology: private internal, private external & public
Infrastructure Platforms – more than IaaS

- Service Orchestration
- Application Workload Software
- Platform (Oracle, Weblogic, etc)
- CBA SOE (Standard operating environment)
- Operating System
- Hypervisor
- Physical Compute Resources (Processor, Memory, Storage)
- Network Infrastructure

**IaaS**
Infrastructure delivery model such that consumers can rent virtualized resources (compute, storage, network) maintained, operated and supported by the internal or external provider. IaaS includes hypervisor, if necessary.

**iPaaS**
Infrastructure Platform-as-a-Service - iPaaS is defined as “thick” infrastructure that can be accessed over the network. Platforms are pre-integrated resource assemblies that can be auto-provisioned and serve as building blocks for new solutions.
Infrastructure Platforms as a Service

- Infrastructure Platforms are pre-integrated software assemblies
- A set of standard, pre-built containers into which we build and run applications and services, delivered as-a-Service.
- Centralised, Unified management software which provides a single point of control over all our Infrastructure Platforms
Oracle as a Service (OaaS) Implementation

1. Provide Oracle database services via the Platform as a Service model.
   - Build a shared infrastructure and software platform
   - Uniform, standardised service offering
   - Oracle database services “on tap”

2. Consolidate 300+ small to medium database environments on to 3 Grids
   - Centralise management of Oracle systems
   - Significant reduction in servers and associated license & hosting charges
   - Clean up the “rats & mice”

3. Operationalise
   - Define common hosting standards and support arrangements
   - A dedicated team of operational DBAs manage the Platform, not an application
   - Develop a charge back model for cost recovery


5. OaaS v2 – hardware in production July 2010, product launch this year
OaaS Evolution at CBA

Cluster of Enterprise-Class Sun Servers, integrated by CBA
(CommSee, NetBank)

Cluster of Commodity-Class Sun Servers, integrated by CBA
(OaaS v1)

Exadata
(OaaS v2)
OaaS v2

- OaaS v2 is being implemented on the Sun Oracle Database Machine
  - Oracle market it as a high performance DB machine
  - We see it as an ideal consolidation platform – the engineering is already done!
  - Expect this trend to continue – pre-built "Platform Servers" or "Private Cloud Servers": just plug in and go.
  - The first Exadata Machine was delivered to CBA in December 2009, second in May 2010.
  - First application migrated to OaaS v2 is Peoplesoft Financials.
Oracle as a Service In A Nutshell

- Host many Oracle database applications on a cluster of hardware
- Processor consolidation
  - Run each server hotter
  - Take advantage of complimentary workload peaks.
- Higher Availability
  - Load balancing
  - HA failover for component failure
  - Standby DR
  - Most apps do not implement these features – too expensive
- Cost Reduction
- Better Service
  - Full time experts
  - Always on-call
- Reduced Risk
  - Whole environment is managed
  - Operated as a “business”
Grid and Virtualization

Virtualization within a resource

- App A
- App B
- App C
- App D

Virtualization Layer

Large, multi-CPU server

Virtualization across resources

- App A
- App B
- App C
- App D

Virtualization Layer

Make lots of separate small servers look like one big server (think SETI)

Make one big server look like many smaller servers (Hypervisors, LPARs, Containers etc)

- "Grid is a logical extension of virtualization to encompass both workload and information virtualization across a distributed infrastructure." §
- For a grid platform the unit of provisioned service is *not* a VM – no hypervisor.

§ Matt Haynos, Director Grid Strategy, IBM
OaaS Changes Operational Economics

- Consolidate many individual databases onto a OaaS platform
  - Centralised management of database systems
  - Consistent, standardised platform
  - Significant reduction in servers and associated operational charges
  - A dedicated team of operational DBAs manage the OaaS platform, not an application database
  - Platform Economics

![Graph showing the comparison between Traditional silo approach and Grid computing model](image-url)
## Where are the Savings?

<table>
<thead>
<tr>
<th>Areas to Reduce Costs</th>
<th>How Cost Reductions are Achieved</th>
</tr>
</thead>
</table>
| **Hardware**          | Significantly Reduce Hardware Requirements:  
  • Remove need for expensive servers, cluster lower cost servers  
  • Buy what you need now, scale as processing requirements grow, JIT |
| **Operating System**  | Adopt a single O/S build to reduce operating system software & maintenance charges |
| **3rd Party Software**| Leverage standard features of the database software stack where possible to minimise the need for much 3rd party software.  
  e.g. Standardise the design and software for:  
  • System Administration & Monitoring  
  • Replication/DR  
  • Clustering Software  
  • File System/Volume Management |
| **Maintenance**       | Reduction in Hardware Maintenance (less dependency on 24x7 support)  
  Reduction in Software Maintenance (on 3rd party software and O/S) |
## Where are the Savings?

<table>
<thead>
<tr>
<th>Areas to Reduce Costs</th>
<th>How Cost Reductions are Achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operations</strong></td>
<td>Reduced number of environments to manage:</td>
</tr>
<tr>
<td></td>
<td>• Standard operating environments cost less to manage</td>
</tr>
<tr>
<td></td>
<td>Adoption of Enterprise Management tools &amp; processes:</td>
</tr>
<tr>
<td></td>
<td>• Automate system administration tasks that are performed manually</td>
</tr>
<tr>
<td></td>
<td>• Standard Operating Procedures leveraged across for hosted applications</td>
</tr>
<tr>
<td></td>
<td>• Improved governance of environments</td>
</tr>
<tr>
<td><strong>Database</strong></td>
<td>There are opportunities for long term savings:</td>
</tr>
<tr>
<td></td>
<td>• Higher utilisation of assets =&gt; fewer CPUs =&gt; Reduction in license fees</td>
</tr>
<tr>
<td></td>
<td>• Fewer Oracle environments =&gt; reduced operational tasks.</td>
</tr>
<tr>
<td></td>
<td>• Fewer FTEs required to manage consolidated environment</td>
</tr>
<tr>
<td><strong>Others</strong></td>
<td>Project Costs:</td>
</tr>
<tr>
<td></td>
<td>• Much faster provisioning of new database environments for projects. A new dev environment is available within hours, instead of weeks/months.</td>
</tr>
<tr>
<td></td>
<td>• HA solution design done! Production already built.</td>
</tr>
</tbody>
</table>
Hardware asset is charged per server.

Database support may be independent and charged per instance.

Database licensing is independent and charged per CPU, regardless of usage.

OS hosting is independent and charged per instance.

Hardware asset is charged per server.

Each implementation needs to pay for connectivity to LAN or WAN.

Data is charged on allocated storage, not on actual usage.
Service Delivery Transformation

- Simple and quick to provision an environment
- A “bundled” offering
- No infrastructure design required per project:
  - No need for expensive SME resource
  - Makes best practice design available to all

Whole stack is now charged as a single Resource Unit in line with actual consumption.

Data storage charge is based on actual consumption.
What savings are we realising?

- P&L breakeven in Year 1, cashflow positive Year 2
- 150% ROI over five years – and that’s for the consolidation only
  - If you factor in cost avoidance – costs not incurred by new applications – ROI is higher again
- Per application OaaS OpEx charge is 40% – 50% of a standalone environment
Many ways to apportion cost
- No standard measure of chargeable resource unit
- How do you measure workload?
- Each to their own for the moment!

Settled on a CPU Month measure of resource usage
- Simple to understand
- Set a minimum monthly charge of 0.5 CPU month – the base hosting fee

Pay-as-go utility charge back
- No upfront charge or ongoing commitment
- Pricing variability was an issue – BU finance preferred budget certainty
- First year, billed in 0.5 CPU Month increments, now moving to 0.1 increments

The service is "overbooked" – recover 89 CPUs worth of capacity; only have 72!

**Charge Back Model**

<table>
<thead>
<tr>
<th>Service Name</th>
<th>DB Time (s)</th>
<th>DB CPU (s)</th>
<th>Physical Reads</th>
<th>Logical Reads</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSPA_MITG</td>
<td>12,300.50</td>
<td>5,144.90</td>
<td>1,438,859</td>
<td>99,811,632</td>
</tr>
<tr>
<td>OSPA_DCM</td>
<td>3,163.90</td>
<td>2,141.80</td>
<td>114,736</td>
<td>46,540,055</td>
</tr>
<tr>
<td>OSPA_CCL</td>
<td>2,496.30</td>
<td>1,455.40</td>
<td>127,937</td>
<td>64,295,226</td>
</tr>
<tr>
<td>OSPA_THL</td>
<td>984.50</td>
<td>725.70</td>
<td>32,184</td>
<td>5,404,057</td>
</tr>
<tr>
<td>OSPA_CPI</td>
<td>339.10</td>
<td>160.40</td>
<td>16,673</td>
<td>1,671,850</td>
</tr>
<tr>
<td>OSPA_MDC</td>
<td>154.90</td>
<td>85.50</td>
<td>13,638</td>
<td>1,473,399</td>
</tr>
<tr>
<td>OSPA_IFW</td>
<td>16.00</td>
<td>10.50</td>
<td>225</td>
<td>17,895</td>
</tr>
<tr>
<td>OSPA_PFR</td>
<td>16.80</td>
<td>6.70</td>
<td>1,291</td>
<td>85,457</td>
</tr>
</tbody>
</table>
Reduce Risk, Improve Time to Market

For new Projects:

- Remove a phase from the project – infrastructure already in place
- Remove reliance on expensive/scarcie SME resources for design and build
- No longer need to manage risk associated with procurement and build
- Time to instantiate a new Production quality environment: 3 months -> 2 minutes.

Example: New ISV Application introduced into our Online Share Trading platform

- Required to test performance under the workload and data volume conditions projected in 2 years time.

<table>
<thead>
<tr>
<th></th>
<th>Dedicated Infrastructure</th>
<th>OaaS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implementation Time</td>
<td>3-4 months</td>
<td>few hours</td>
</tr>
<tr>
<td>$ Cost to Project</td>
<td>Several hundred thousand</td>
<td>&lt; $10K</td>
</tr>
<tr>
<td>On Project Completion</td>
<td>Under-utilized asset remains</td>
<td>Environment turned-off</td>
</tr>
</tbody>
</table>
Thoughts on Implementing PaaS

1. Take the time to get the right technical/commercial solution for your business
   • Different virtualisation techniques have different densities – resulting in different economics

2. Must have buy-in from Application owners
   • Plan of when and how to migrate applications
   • Internal sales function needed to rustle up demand

3. Go for quick wins
   • Migrate / host the easiest apps first

4. Invest in Governance and Operational Process Improvement
   • Much, much more than a technology solution

5. Have a clear, consistent, accurate sales pitch
   • Beware the FUD factor; can derail many an initiative
Questions?