Techniques for implementing & running robust and reliable DB-centric Grid Applications

International Symposium on Grid Computing 2008
11 April 2008
*Miguel Anjo, CERN - Physics Databases*
Outline

• Robust and DB-centric applications

• Technologies behind
  – Oracle Real Application Clusters
  – Oracle Streams
  – Oracle Data Guard
  – DNS Load balancing

• Implement robust applications
  – What to expect
  – What to do
  – Other planning
Robust & DB-centric Applications

- **Robust**: *(adj.)* vigorous, powerfully built, strong
  - **Resilient**: *(adj.)* an ability to recover from or adjust easily to misfortune or change

- **DB-centric applications**: essential data is stored in a database
Technologies behind

- Oracle Real Application Clusters
- Oracle Data Guard
- DNS Load Balancing
Oracle RAC architecture

Oracle Real Application Clusters 10g - Foundation for Grid Computing

Figure 1: Oracle RAC – clustering database servers – foundation for Enterprise Grid Computing delivering high availability, scalability and flexibility.
Oracle RAC Architecture

- Applications consolidated on large clusters
- Redundant and homogeneous HW across each RAC
Architecture
Architecture (Oracle services)

- Resources distributed among Oracle services
  - Applications assigned to dedicated service
  - Applications components might have different services
- Service reallocation not always completely transparent

<table>
<thead>
<tr>
<th></th>
<th>CMS_COND</th>
<th>ATLAS_DQ2</th>
<th>LCG_SAM</th>
<th>LCG_FTS</th>
<th>CMS_SSTRACKER</th>
<th>CMS_PHEDEXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preferred</td>
<td>A1</td>
<td>A2</td>
<td>A3</td>
<td>A4</td>
<td>Preferred</td>
<td>Preferred</td>
</tr>
<tr>
<td>A5</td>
<td>A6</td>
<td>A7</td>
<td>A1</td>
<td>A2</td>
<td>Preferred</td>
<td>Preferred</td>
</tr>
<tr>
<td>A3</td>
<td>A4</td>
<td>A5</td>
<td>Preferred</td>
<td>Preferred</td>
<td>A1</td>
<td>Preferred</td>
</tr>
<tr>
<td>A2</td>
<td>A4</td>
<td>A5</td>
<td>Preferred</td>
<td>Preferred</td>
<td>A1</td>
<td>Preferred</td>
</tr>
<tr>
<td>A1</td>
<td>A2</td>
<td>A3</td>
<td>Preferred</td>
<td>Preferred</td>
<td>Preferred</td>
<td>Preferred</td>
</tr>
<tr>
<td>A2</td>
<td>A3</td>
<td>A4</td>
<td>Preferred</td>
<td>Preferred</td>
<td>Preferred</td>
<td>Preferred</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CMS RAC Node #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>CMS_COND</td>
<td>A1</td>
<td>A2</td>
<td>A3</td>
<td>A4</td>
<td>A5</td>
<td>A6</td>
<td>A7</td>
<td></td>
</tr>
<tr>
<td>ATLAS_DQ2</td>
<td>A2</td>
<td>A3</td>
<td>A4</td>
<td>A5</td>
<td>A6</td>
<td>A7</td>
<td>A1</td>
<td></td>
</tr>
<tr>
<td>LCG_SAM</td>
<td>A5</td>
<td>A3</td>
<td>A1</td>
<td>A2</td>
<td>Preferred</td>
<td>Preferred</td>
<td>Preferred</td>
<td></td>
</tr>
<tr>
<td>LCG_FTS</td>
<td>A4</td>
<td>A5</td>
<td>A6</td>
<td>A7</td>
<td>Preferred</td>
<td>A1</td>
<td>A2</td>
<td>A3</td>
</tr>
<tr>
<td>CMS_SSTRACKER</td>
<td>Preferred</td>
<td>Preferred</td>
<td>Preferred</td>
<td>Preferred</td>
<td>Preferred</td>
<td>Preferred</td>
<td>Preferred</td>
<td>Preferred</td>
</tr>
<tr>
<td>CMS_PHEDEXX</td>
<td>A2</td>
<td>Preferred</td>
<td>Preferred</td>
<td>A1</td>
<td>A3</td>
<td>A4</td>
<td>A5</td>
<td></td>
</tr>
</tbody>
</table>

- CMS Cond Node 1
- ATLAS DQ2 Node 2
- LCG SAM Node 3
- LCG FTS Node 4
- CMS SSTRACKER Node 5
- CMS PHEDEX Node 6

[Diagram of servers and databases]
Service’s *connection string* mentions all virtual IPs
- It connects to a random virtual IP (client load balance)
- Listener sends connection to least loaded node where service runs (server load balance)

```bash
$ sqlplus db_user@LCG_FTS
```

```
listener srv1
listener srv2
listener srv3
listener srv4
```

**Virtual IP**
Architecture (load balancing)

- Used also for rolling upgrades (patch applied node by node)
- Small glitches might happen during VIP move
  - no response / timeout / error
  - applications need to be ready for this → catch errors, retry, not hang

```bash
$ sqlplus db_user@LCG_FTS
```

```
 srv1-v
|↓|
listener srv1
```

```
 srv2-v
```

```
 srv3-v
|↓|
listener srv3
```

```
 srv4-v
|↓|
listener srv4
```

```
 LCG_FTS
```

Virtual IP
Oracle Streams

- Streams data to external databases (Tier1)
  - Limited throughput
  - Can be used for few applications
  - Create read-only copy of DB
  - Application can failover to copy
Oracle Data guard

- Use as on-disk backup
  - Physical stand-by RAC with small lag (few hours)
  - Can be open read-only to recover from human errors
  - Switch to primary mode as Fast disaster recovery mechanism
DBA Main concerns (based on experience)

- Human errors
  - By DBA on administrative tasks
    - Use and test procedures, not always easy task
  - By developers
    - Restrict access to production DB to developers

- Logical corruption / Oracle SW bugs
  - Data inserted in wrong schemas
    - Patches better tested on pilot environments before deployment in production

- Oracle software Security
  - Quarterly security patches released by Oracle

- Increasing amount of stored data
  - Tapes slow as 5 years ago, backups take longer
    - Move to backup on disks
    - Prune old redundant data/summarizing
The Databases reality at CERN

- Databases for the world’s biggest machine: particle collider
- 18 database RACs (up to 8 nodes)
  - 124 servers, 150 disk arrays (+1700 disks)
  - Or: 450 CPU cores, 900GB of RAM, 550 TB of raw disk space(!)
- Connected to 10 Tier-1 sites for synchronized databases
  - Sharing policies and procedures
- Team of 5 DBAs + service coordinator and link to experiments
- 24x7 best effort service for production RACs
- Maintenance without downtime within RAC features
  - 0.02% services unavailability (average for 2008) = 1.75 hours/year
  - 0.32% server unavailability (average for 2008) = 28 hours/year
  - Patch deployment, broken hardware
DNS Load balancing

Q: What is the IP address of application.cern.ch?

A: application.cern.ch resolves to:
   node4.cern.ch
   node1.cern.ch
   node2.cern.ch
   node3.cern.ch

Connecting to node4.cern.ch
DNS Round Robin

- Allows basic load distribution

```
lxplus001 ~ > host lxplus.cern.ch
lxplus.cern.ch has address 137.138.4.171 (1)
lxplus.cern.ch has address 137.138.4.177 (2)
lxplus.cern.ch has address 137.138.4.178 (3)
lxplus.cern.ch has address 137.138.5.72  (4)
lxplus.cern.ch has address 137.138.4.169 (5)
```

```
lxplus001 ~ > host lxplus.cern.ch
lxplus.cern.ch has address 137.138.4.177 (2)
lxplus.cern.ch has address 137.138.4.178 (3)
lxplus.cern.ch has address 137.138.5.72  (4)
lxplus.cern.ch has address 137.138.4.169 (5)
lxplus.cern.ch has address 137.138.4.171 (1)
```

- No withdrawal of overloaded or failed nodes
DNS Load Balancing and Failover

• Requires an additional server = arbiter
  – Monitors the cluster members
  – Adds and withdraw nodes as required
  – Updates are transactional
    • Client never sees an empty list

```
lxplus001 ~ > host lxplus.cern.ch
lxplus.cern.ch has address 137.138.4.168
lxplus.cern.ch has address 137.138.4.168
lxplus.cern.ch has address 137.138.5.74
lxplus.cern.ch has address 137.138.5.72
lxplus.cern.ch has address 137.138.5.76
```

Application Load Balancing

![Diagram of Application Load Balancing]

- **Application Cluster**
- **Load Balancing Arbiter**
- **DNS Server**

**Node Metrics**
- node1: metric=24
- node2: metric=48
- node3: metric=35
- node4: metric=27

**2 best nodes for application.cern.ch:**
- node1
- node4

**Q:** What is the IP address of application.cern.ch?

**A:** application.cern.ch resolves to:
- node4.cern.ch
- node1.cern.ch

Connecting to node4.cern.ch

Connecting to node1.cern.ch
DB-centric Applications

- Development cycle
- What to expect
- What to do
- How to survive planned interventions
Apps and Database release cycle

- **Applications’ release cycle**
  - Development service
  - Validation service
  - Production service

- **Database software release cycle**
  - Validation service version 10.2.0.(n+1)
  - Production service version 10.2.0.n
  - Production service version 10.2.0.(n+1)
What to expect

• Network glitches
  – Any network failure

• Disconnects
  – Idle time, network failure

• Failover
  – Rolling upgrades, server failure

• Interventions (planned, unplanned)
  – Upgrades, patches
What application should do

- **General guidelines for DB apps**
  - Primary keys, foreign keys, indexes, bind variables

- **Reconnect**
  - Catch disconnect errors
  - Reconnect before giving error

- **Handle errors**
  - Catch DB common errors
  - Rollback transaction and re-execute, if needed

- **Throttle retries**
  - After 2/3 consecutive failures, wait before retry

- **Timeout calls**
  - If load is too high DB might stop responding
  - Error better than no results
  - Timeout connection to database
Survive problems and interventions

- Have some cache/log when possible
  - Next few transfers, results of last/common requests
  - Create queue on server for later apply

- Keep last results as static (for graphs...)

- Have a read-only copy of the DB somewhere
  - Oracle Streams being tested at CERN
  - Application should be able to failover and failback
Check space constraints

- Space/disk is almost free but...
- ...fragmentation is still around
- ...bad organisation can lead to slowness

- Keep DB tidy
  - Drop unnecessary data
  - Archive it (compressed, not indexed)
  - Aggregate it (keep only summaries)
  - Oracle partitioning used at CERN
Monitoring and logging

- Application responsiveness
  - Time to perform a operation
- Database availability
- Log errors and check them
- Automatise most of the error correction
  - Pass it to operators
- Weekly reports on application and database
  - Ask your DBAs
  - Easy to spot changes in behaviour
Summary

- Cluster database using redundant hardware
  - Split applications at database, isolate if needed
- Oracle DataGuard as “on disk backup” for fast recoveries, human errors
- Application stateless as possible
- DNS load balancing with Arbiter to get better servers
- Application should target to run also during DB unavailability
  - Local cache/queues
- Predict space usage in future (and cleanup)
- Monitor application usage, errors, times
- Involve your DBA as much as possible
Questions?

- ...and answers.

References:
- CERN Physics Databases wiki:
  - General advice
  - Connection management
  - [http://cern.ch/phydb/wiki](http://cern.ch/phydb/wiki)
- WLCG Service Reliability Workshop (Nov07)
  - DNS Load balancing *(slides from Vlado Bahyl)*
  - Case studies
  - [http://indico.cern.ch/conferenceOtherViews.py?confId=20080](http://indico.cern.ch/conferenceOtherViews.py?confId=20080)
- WLCG Collaboration Workshop (Tier0/Tier1/Tier2)(Apr08)
  - [http://indico.cern.ch/conferenceOtherViews.py?confId=6552](http://indico.cern.ch/conferenceOtherViews.py?confId=6552)