How to provide your own DBaaS: a practical story

Ruben.Gaspar.Aparicio@cern.ch
on behalf Database on Demand Team

CERN, IT Databases group

PostgreSQL Conference Europe, Vienna 30th Oct. 2015

Proton Antiproton collision leading to discovery of W and Z particles. 1984 Nobel Prize: Carlo Rubbia & Simon van der Meer.
Agenda

- CERN introduction
- CERN DBaaS Rationale & evolution
- Architecture, software components description
- Network architecture
- Storage setup
- Backup & Recoveries
- Upgrades
- HA
- Monitoring
- Future development
Agenda

- CERN introduction
- CERN DBaaS Rationale & evolution
- Architecture, software components description
- Network architecture
- Storage setup
- Backup & recovery
- Upgrades
- HA
- Monitoring
- Future development
CERN

- CERN - European Laboratory for Particle Physics
- Founded in 1954 by 12 Countries for fundamental physics research in the post-war Europe
- Today 21 member states + world-wide collaborations
  - About ~1000 MCHF yearly budget
  - 2’300 CERN personnel
  - 10’000 users from 110 countries
Fundamental Research

- What is 95% of the Universe made of?
- Why do particles have mass?
- Why is there no antimatter left in the Universe?
- What was the Universe like, just after the "Big Bang"?
Who Changed My Plan?
The Large Hadron Collider (LHC)

Largest machine in the world
27km, 6000+ superconducting magnets

Fastest racetrack on Earth
Protons circulate 11245 times/s (99.9999991% the speed of light)

Emptiest place in the solar system
High vacuum inside the magnets

Hottest spot in the galaxy
During Lead ion collisions create temperatures 100 000x hotter than the heart of the sun;
WLCG

- The world’s largest scientific computing grid

More than 100 Petabytes of data stored and analysed. Increasing: 20+ Petabytes/year

CPU: over 250K cores
Jobs: 2M per day

160 computer centres in 35 countries

More than 8000 physicists with real-time access to LHC data
CERN IT Databases group

- Database on Demand (DBoD) single instances
  - 226 MySQL Open community databases (5.6.17)
  - 36 Postgresql databases (9.2.9 -> 9.4.5)
  - 8 Oracle databases (11.2.0.4 -> 12.1)
- ~100 Oracle databases, most of them RAC
  - 100% NAS storage
  - ~750 TB of data files for production DBs in total
  - Using a variety of Oracle technologies: Active Data Guard, Golden Gate, Cluster ware, etc.
- Examples of critical production DBs:
  - LHC logging database ~315 TB, expected growth up to ~180 TB / year
  - 13 production experiments’ databases ~15-25 TB in each
  - Read-only copies (Active Data Guard)
Agenda

- CERN introduction
- **CERN DBaaS Rationale & evolution**
- Architecture, software components description
- HA
- Network architecture
- Storage setup
- Upgrades
- Monitoring
- Future development
DBaaS providers

Legend
- MySQL
- Oracle
- PostgreSQL
- B&R
- HA
- Upgrades
- Monitoring

HP Helion Public Cloud
CERN DBaaS Manifesto

- Making users database owners
  - Full DBA privileges
- Covers a demand from CERN community not addressed by the Oracle service
  - Different RDBMS: MySQL, PostgreSQL and Oracle (private use)
- No access to underlying hardware
- Foreseen as single instance service
- No DBA support or application support
- No vendor support (except for Oracle)
- It provides tools to manage DBA actions: configuration, start/stop, upgrades, backups & recoveries, instance monitoring
DBoD instance per type

Database on Demand instances per DBMS

- MySQL: 226
- Oracle: 8
- PostgreSQL: 36

Database on Demand instances per Department

- IT: 149
- PH: 68
- BE: 15
- GS: 12
- EN: 8
- TE: 7
- DG: 5
- FP: 2
- DGS: 1
- HR: 1
- PF: 1
- External: 1

Many critical services running on our DBaaS

DBoD instances space consumption

<table>
<thead>
<tr>
<th>Data</th>
<th>WALs/binlogs/archivelogs</th>
<th>Snapshots</th>
</tr>
</thead>
<tbody>
<tr>
<td>~ 5TB</td>
<td>~8TB</td>
<td>~1.4 TB</td>
</tr>
</tbody>
</table>

Biggest: 375GB  Average: 16.2GB
https://cern.ch/dbondemand
Agenda

• CERN introduction
• CERN DBaaS Rationale & evolution
• Architecture, software components description
• Network architecture
• Storage setup
• Backup & recovery
• Upgrades
• HA
• Monitoring
• Future development
Architecture: DBoD 2.0

New instances installed by Puppet
DBoD Web Interface

- Allows the user to manage its instances without specific knowledge about databases:
  - Startup and shutdown the database.
  - Backups and recoveries.
  - Logs.
  - Upgrades.
  - Monitoring.
  - Configuration.
  - Jobs history
DBoD Web Interface

- [https://github.com/cerndb/dbod-webapp](https://github.com/cerndb/dbod-webapp) 18.5k lines of code checked by Coverity
- ZK framework 7.0.2. Application running on Tomcat 7
- Role Based Access Control using e-groups and LDAP Authentication.
- Access to the Monitoring Server is secured using an internal proxy and security tokens.
- The owner of the instance can manage the e-groups with access from the Web Interface.
Backend for metadata

- Based on PostgreSQL 9.4. Heavy use of JSON types, it will replace Oracle.
- **RUNDECK**
  - To manage execution of scheduled jobs (before managed with scheduled jobs on the Oracle database)
  - Useful also for any task that should be run as job
  - REST API
DBOD WS API

- https://github.com/cerndb/dbod-api
- A single point of access to the service data for all components
- REST interface
- Developed in Python (using Tornado)
- Interfaces to:
  - The service backend database
  - IT-DB LDAP directory and API
DBoD 2.0

- Simplification

<table>
<thead>
<tr>
<th>Current implementation</th>
<th>DBoD 2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>88 Perl files ~ 15kloc</td>
<td>44 Perl file ~ 6kloc (80% functionality implemented)</td>
</tr>
<tr>
<td>19 Bash scripts ~ 1.2kloc</td>
<td>None</td>
</tr>
<tr>
<td>58 support files: configuration, templates, SQL</td>
<td>24 support files, Puppet managed</td>
</tr>
</tbody>
</table>

- Perl 5.20.2. Using Perl::Moose as a foundation

- [https://github.com/cerndb/DBOD-core](https://github.com/cerndb/DBOD-core)
CI: Testing

- Using Travis-CI free for OSS projects
- Problematic to mock complex operations
- Functional testing
  - Latest builds
  - Using RUNDECK
# Unit testing: Travis CI

Travis CI - Help make Open Source a better place and start building better software today!

## cerndb / DBOD-core

<table>
<thead>
<tr>
<th>Build History</th>
<th>Current</th>
<th>Branches</th>
<th>Pull Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.63 Merge pull request #25 from cerndb/travis-ci</td>
<td><img src="Green" alt="Green Checkbox" /></td>
<td><img src="Green" alt="Green Checkbox" /></td>
<td><img src="Green" alt="Green Check" /></td>
</tr>
<tr>
<td>icot committed</td>
<td><img src="Green" alt="Green Check" /></td>
<td><img src="Green" alt="Green Check" /></td>
<td><img src="Green" alt="Green Check" /></td>
</tr>
<tr>
<td>master Merge pull request #25 from cerndb/travis-ci</td>
<td><img src="Green" alt="Green Checkbox" /></td>
<td><img src="Green" alt="Green Checkbox" /></td>
<td><img src="Green" alt="Green Check" /></td>
</tr>
<tr>
<td>icot committed</td>
<td><img src="Green" alt="Green Check" /></td>
<td><img src="Green" alt="Green Check" /></td>
<td><img src="Green" alt="Green Check" /></td>
</tr>
<tr>
<td>travis-ci TRAVIS-CI: Change default config path and set up configuration file.</td>
<td><img src="Green" alt="Green Checkbox" /></td>
<td><img src="Green" alt="Green Checkbox" /></td>
<td><img src="Green" alt="Green Check" /></td>
</tr>
<tr>
<td>icot committed</td>
<td><img src="Green" alt="Green Check" /></td>
<td><img src="Green" alt="Green Check" /></td>
<td><img src="Green" alt="Green Check" /></td>
</tr>
</tbody>
</table>

- Number of tests passed: 143 passed
- Number of tests passed: 142 passed
- Number of tests passed: 140 passed
- Time taken: 1 minute 24 seconds
- Time taken: 1 minute 53 seconds
- Time taken: 1 minute 28 seconds
- Commit: 387106d
- Commit: fc7a772
- Commit: a day ago
Functional tests: RUNDECK

Latest code
Or
RDBMS binaries
RUNDECK: job execution summary
Agenda

- CERN introduction
- CERN DBaaS Rationale & evolution
- Architecture, software components description
- **Network architecture**
- Storage setup
- Backup & recovery
- Upgrades
- HA
- Monitoring
- Future development
Network architecture

- Just first element of each type is shown cabled
- Each switch is in fact a set of switches (4 in our latest setup) managed as one by HP Intelligent Resilient Framework (IRF)
- ALL our databases run with same network architecture.
- NFSv3 is used for data access
- Transtec Database server: 128GB RAM & 2x eight-core Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz
Agenda

• CERN introduction
• CERN DBaaS Rationale & evolution
• Architecture, software components description
• Network architecture
• Storage setup
• Backup & recovery
• Upgrades
• HA
• Monitoring
• Future development
NetApp evolution at CERN (last 8 years)

FAS3000
100% FC disks
DS14 mk4 FC
Data ONTAP®
7-mode

scaling up

FAS6200 & FAS8000
Flash pool/cache = 100% SATA disk + SSD
DS4246
6gbps

scaling out

Data clustered
ONTAP®
2gbps
6gbps
Data protection

- Storage is monitored: NetApp tools + home made tools
- Multipath access to disks (redundancy + performance) → disks are seen by two controllers (HA pair) → Transparent interventions
- RAID6
- Automatic scrubbing (based on checksum)
- Rapid RAID Recovery + Disk Maintenance Center
DBoD instance storage setup

- 2 file systems
- Snap reservation enabled
- Snapshot auto-deletion enabled
- Data volume located in hybrid volume: HDD + SSD caching

Clustered ONTAP

DBOD instances

MySQL

PostgreSQL

Datafiles

Binary logs / WALs

10GbE

20GbE

Dual SAS loop and double stack
24gbps

2 file systems
Snap reservation enabled
Snapshot auto-deletion enabled
Data volume located in hybrid volume: HDD + SSD caching
Mount options

- Oracle and MySQL are well documented
  - Mount Options for Oracle files when used with NFS on NAS devices (Doc ID 359515.1)
  - Best Practices for Oracle Databases on NetApp Storage, TR-3633
  - What are the mount options for databases on NetApp NFS? KB ID: 3010189

- PostgreSQL not popular with NFS, though it works well if properly configured
  - MTU 9000, reliable NFS stack e.g. NetApp NFS server implementation
  - Don’t underestimate impact
Mount options (II)

MySQL and PostgreSQL single instance:

```bash
-- PostgreSQL and MySQL, data and binlogs or WAL file systems
controller-lif2:/ORA/dbs02/CPGSQL on /ORA/dbs02/CPGSQL type nfs (rw,noatime,bg,hard,nointr,tcp,vers=3,timeo=600,rsize=65536,wsize=65536)
controller-lif4:/ORA/dbs03/CPGSQL on /ORA/dbs03/CPGSQL type nfs (rw,noatime,bg,hard,nointr,tcp,vers=3,timeo=600,rsize=65536,wsize=65536)
```

With old mount point option `actime=0`  
With mount point option `noatime`  

**lagging puppetdb, slave running pgsq1 9.2.9**
IO histograms on NetApp

- Based on [http://cern.ch/go/knb9](http://cern.ch/go/knb9)
- PostgreSQL 9.4.5, under HammerDB workload
Agenda

• CERN introduction
• CERN DBaaS Rationale & evolution
• Architecture, software components description
• Network architecture
• Storage setup
• **Backup & recovery**
• Upgrades
• HA
• Monitoring
• Future development
Backup management

- Same backup procedure for all RDBMS. Only **data volume** is snapshot.

- Backup workflow:

  ```
  mysql> FLUSH TABLES WITH READ LOCK;
  mysql> FLUSH LOGS;
  or
  Oracle> alter database begin backup;
  or
  Postgresql> SELECT pg_start_backup('$SNAP');
  ```

  ```
  mysql> UNLOCK TABLES;
  or
  Oracle> alter database end backup;
  or
  Postgresql> SELECT pg_stop_backup(), pg_create_restore_point('$SNAP');
  ```

  NetApp drawings

  ```
  pubstg: 280GB size, ~ 1 TB archivelogs/day
  ```
Secondary storage: Tape backups

- Driven by demand/need, not initially in the plans
- Likely to be removed
- Possible only on PostgreSQL and MySQL
  - Oracle12c solution comes already with a tape backup!
- Consistent snapshot + redo logs sent to tape
- Database activity is not impacted
- Tape backups are not validated
- Manual process to set them up, need to contact us (DBOD + TSM service)
- Likely to be replaced by CERN CEPH cluster (S3) or
Instance restore

Automatic snapshots

Data files

Binary logs

Manual snapshot

Point-in-time recovery

Now
Agenda

• CERN introduction
• CERN DBaaS Rationale & evolution
• Architecture, software components description
• Network architecture
• Storage setup
• Backup & recovery
• Upgrades
• HA
• Monitoring
• Future development
Upgrade testing: from 9.2.9 to 9.4.5

- **9.2.9**
  - DBOD-core: pg_snapshot
  - pgreplay -f
  - DBOD-core: pg_clone
  - 1..N
  - DBOD-core: pg_restore
  - pgreplay -r

- **9.4.5**
  - DBOD-core: pg_upgrade
  - DBOD-core: pg_snapshot
  - pgbadger report
  - 1..N
  - DBOD-core: pg_restore
  - pgbadger report
Agenda

- CERN introduction
- CERN DBaaS Rationale & evolution
- Architecture, software components description
- Network architecture
- Storage setup
- Backup & recovery
- Upgrades
- HA
- Monitoring
- Future development
High Availability

- Driven by demand/need, not initially in the plans
- Not relying on virtualization features so far (it may change in the future, as OpenStack evolves)
- 4 node clusters
  - Nowadays two clusters running under Oracle clusterware 12.1.0.1.
- Clusterware controls:
  - Virtual IP
  - RDBMS instance
    - PostgreSQL and MySQL instances can co-exist, different versions supported.
High Availability

Testing the cluster (MySQL & Postgresql instances)

<table>
<thead>
<tr>
<th>Failover test</th>
<th>Downtime</th>
<th>Avg. (s)</th>
<th>Min. (s)</th>
<th>Max (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kill process</td>
<td></td>
<td>16.9</td>
<td>4</td>
<td>39</td>
</tr>
<tr>
<td>Kill process (different node)</td>
<td></td>
<td>21.7</td>
<td>10</td>
<td>34</td>
</tr>
<tr>
<td>Network down</td>
<td></td>
<td>39.9</td>
<td>37</td>
<td>47</td>
</tr>
<tr>
<td>Server down</td>
<td></td>
<td>37</td>
<td>33</td>
<td>43</td>
</tr>
<tr>
<td>Relocate</td>
<td></td>
<td>6.2</td>
<td>5</td>
<td>7</td>
</tr>
</tbody>
</table>

- For instances running on an Oracle cluster ware, care must be taken in case of server crash for MySQL instances.

  "InnoDB: Unable to lock ./ibdata1, error: 11" Error Sometimes Seen With MySQL on NFS (Doc ID 1522745.1)
Agenda

• CERN introduction
• CERN DBaaS Rationale & evolution
• Architecture, software components description
• Network architecture
• Storage setup
• Backup & recovery
• Upgrades
• HA
• Monitoring
• Future development
# Monitoring: current tools

<table>
<thead>
<tr>
<th>Monitoring</th>
<th>Alerts</th>
<th>Archive</th>
<th>Displays</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Data</td>
<td>Email</td>
<td>Temporal &amp; long term for offline analysis</td>
<td>Host Data</td>
</tr>
<tr>
<td>Service Metrics</td>
<td>SMS</td>
<td></td>
<td>Service Metrics</td>
</tr>
<tr>
<td>Service Availability</td>
<td>ServiceNow Ticket</td>
<td></td>
<td>Service Availability</td>
</tr>
<tr>
<td>Logs</td>
<td></td>
<td></td>
<td>Logs</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Processed Data</td>
</tr>
</tbody>
</table>

- **MySQL**
- **PostgreSQL**
- **ORACLE**

- GNI
- elastic
- hadoop
- kibana

- AppDynamics
- ORACLE
- ORACLE ENTERPRISE MANAGER
Agenda

- CERN introduction
- CERN DBaaS Rationale & evolution
- Architecture, software components description
- Network architecture
- Storage setup
- Backup & recovery
- Upgrades
- HA
- Monitoring
- Future development
Future development

- Cloning
  - Based on NetApp feature
- Replication
- Containers
  - Limit CPU and memory
  - CLI access to instances
- Architecture enhancements done in order to expand model of single instance
  - NoSQL solutions too!
Acknowledge

- DBoD Team
  - Ignacio Coterillo
  - David Collados
  - Jose Andrés Cordero Benitez
  - Charles Delort
- IT-DB colleagues
Questions