AliEn New Catalogue Structure - v2.20

Dushyant Goyal
The LNM Institute of Information Technology, India

November 23, 2011
Outline

- New Catalogue Schema
- Catalogue Optimization
- Job Quota Tests - Fixes
Current schema

- Currently we have the ALICE_CATALOGUE_DATABASE as a combination of 3 databases: alice_users alice_data alien_system
- copyCatalogue.pl script runs through all the 3 databases and combines the 3 databases into a single database i.e. alice_users
- It also removes L#L_QUOTA and L#L_broken tables
- Advantages:
  - Single database much easier to handle
  - Removes the use of HostIndex from the INDEXTABLE and GUIDINDEX being in single database
  - Takes around 2-3 hours
Current schema

- alien_system
- alice_data
- alice_users

Relationship:
- alien_system
  - alice_data
  - alice_users

Intermediate node:
- copyCatalogue.pl

Output node:
- alice_users
Removing some attributes from catalogue tables

- Right now we have attributes `owner varchar(20)`, `gowner varchar(20)`
  - It would be nice to replace them with `ownerId MEDIUMINT unsigned`, `gownerId MEDIUMINT unsigned`
- Creating 2 more tables
  - `USERS`: mapping `ownerId`, `owner`
  - `GROUPS`: mapping `gownerId`, `gowner`
- Modifying the present table `GROUPS` (currently maps `owner` and `gowner`)
  - `UGMAP`: mapping `ownerId`, `gownerId`
- `MEDIUMINT UNSIGNED`: 3 bytes
  - Maximum numbers of users: `8,388,607`
  - PS: We currently have just above 1000 users.
Experiments with the database

- **Script to update the database written:**

- **Some results:**
  - takes around **50 hours**
  - around 10GB of disk space is saved: **5%**

- **Intuitively:**
  - There are around 300M (297,532,376) entries in all L#L tables
  - On average each entry has 10B + 10B for owner varchar (20) & gowner varchar(20)
  - Replacing them with 3Bytes ownerId + 3B gownerId (MEDIUMINT)
  - So, 20-6=14B per entry we save

  - Finally, (14*300M )/(10^9) \(\approx\) **4.2GB**
Experiments with the database

- Throwing some numbers for required queries and for the SELECT queries
  - ALTER TABLE L1308L ADD (uld Int(11), gId Int(11))
  - Query OK, 16202680 rows affected (35 min 38.10 sec)
  - UPDATE L1308L join USERS ON L1308L.owner=USERS.user SET L1308L.uld= USERS.uld;
  - Query OK, 16202680 rows affected (9 min 32.82 sec)
  - ALTER TABLE L1308L DROP COLUMN uld, DROP COLUMN gId
  - Query OK, 16202680 rows affected (36 min 3.65 sec)

- Checking for the SELECT queries
  - SELECT lfn,owner FROM L1308L LIMIT 1000000;
  - 1.34 sec
  - SELECT lfn,ownerId FROM L1308L LIMIT 1000000;
  - 1.34 sec
  - SELECT lfn,user FROM L1308L JOIN USERS ON USERS.uld=L1308L.ownerId LIMIT 1000000
  - 1.53 sec

- After creating indexes on UIlds From USERS table and ownerId from L#L tables.
  - SELECT lfn,user FROM L1308L JOIN USERS ON USERS.uld=L1308L.ownerId LIMIT 1000000
  - ≈ 1.4 sec
Faster Script for updating Database
- using UPDATE_TIME from information_schema table
- only updating the tables which have been modified
- working for the Test_Database
  - P.S. Results still to be checked for the ALICE_CATALOGUE_DATABASE
Experiments with database optimization

- Script to optimize the ALICE_CATALOGUE_DATABASE
- Running di_optimize for optimizing L#L and G#L tables

```
[pcalice74.cern.ch:33871 /pcalice74/user/d/dushyant/] -> di -help
Gives the number of entries in the Catalogue tables i.e. L#L, G#L, G#L_PFN tables and optimizes them
Usage:
   di <options_1> <max_lim> <min_lim> <dir>
   di <options_2>
Options:

   optimize: Optimizes the the L#L LFN tables wrt number of entries in the table (all the L#L tables)
   optimize_dir: Optimizes the the L#L LFN tables wrt number of entries in the table in the path specified
   optimize_guid: Optimizes the G#L and corresponding G#L_PFN tables wrt number of entries in the table
   max_lim: Maximum limit of number of entries to be present in a table
   min_lim: Maximum limit of number of entries to be present in a table
options 2: l => L#L,
   options 2: g => G#L,
   options 2: gp => G#L_PFN,
```

- By optimization we mean combining/splitting the tables to keep an optimized number of entries in the L#L, G#L, G#L_PFN tables
- Optimization of L#L done on the basis of **depth of directory structure**
- Optimization of GUIDs done based upon the **guidTime**
Experiments with database optimization

Optimization Parameters
- Maximum Lim: 5M
- Minimum Lim: 1M

For L#L tables
- Time Taken: 2days (approx)

<table>
<thead>
<tr>
<th></th>
<th>Before Optimization</th>
<th>After Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total No. of tables</td>
<td>1320</td>
<td>400</td>
</tr>
<tr>
<td>No. of New tables</td>
<td>0</td>
<td>20</td>
</tr>
<tr>
<td>Table with Max Entries</td>
<td>76M</td>
<td>20M</td>
</tr>
<tr>
<td>Table with Min Entries</td>
<td>1</td>
<td>13K</td>
</tr>
</tbody>
</table>

For G#L tables
- Time Taken: 2days (approx)

<table>
<thead>
<tr>
<th></th>
<th>Before Optimization</th>
<th>After Optimization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total No. of tables</td>
<td>75</td>
<td>115</td>
</tr>
<tr>
<td>No. of New tables</td>
<td>0</td>
<td>40</td>
</tr>
<tr>
<td>Table with Max Entries</td>
<td>52M</td>
<td>5M</td>
</tr>
<tr>
<td>Table with Min Entries</td>
<td>0</td>
<td>4.9M (approx)</td>
</tr>
</tbody>
</table>
Summary
Fixes

- File Quotas Tests work now !!
- Job Quotas Tests fixed (almost)
Documentation

- Description about AliEn Service (Start, Stop, Status)
- Updating the details about the various logs for better debugging for newcomers
- For ex: Job/Proc logs, alien_tests logs etc
- For ex: Central Services logs (JobManager, JobInfoManager, etc)
- Documentation Page
- Other suggestions are welcome .. !!
Thanks Everyone !!

Special Thanks to Pablo, Steffen, Costin, Latchezar .. !!