Design for the Distributed Data Locator Service for Multi-site Data Repositories

presented by Nakanishi H.
National Institute for Fusion Science (NIFS), NINS, Japan

on behalf of NIFS, NII, and QST research collaborators

12th IAEA-TM CODARP in Daejeon, Korea, 13-17 May 2019
Backgrounds and Objectives

▪ Massive data analyses need **high-bandwidth, low-latency data access** to storage.
  ➢ Need a super-computer cluster together with a huge, local data storage

▪ **For ITER** huge data analyses,
  ➢ JA-DA will prepare huge computer & storage at ITER REC in Rokkasho.
  ➢ Inter-continental data replication method has been well tested.  ➢ *cf. MMCFTP*

▪ **Fusion Virtual Laboratory** in Japan gathers data from **LHD + 3 remote sites**.
  ➢ FVL shares a central storage & index DB  ➢ will be a “**SPoF**” in accidents
  ➢ Multi-tier storage can queue data at every stage, but Index DB should be always on service.
  ➢ Index DB must be a redundant, distributed service by using **multi-master DB**.

▪ In this study, bi-directional replication between multi-master index DB has been designed and tested by using the LHD data system on FVL.
  ➢ Bi-directional replication is enabled by **“BDR extension”** module for PostgreSQL version 9.4 and higher.
ITER UDA structure

- ITER on-site data repository is a single substance. If having a replica, ...

cf. UDA architecture diagram “UDA user manual” (TPLTKG v2, 2018)
ITER UDA structure with remote repository

- Long-distance communications for accessing the Index DB would be a "risk" of data accesses in remote site.
- Index DB would be a single point of failure (SPoF).

12th IAEA-TM CODARP in Daejeon, 13-17 May 2019
ITER UDA structure with “replicated index”

- SPoF of Index DB will be solved by “redundant” Index DB. It can improve Index data safety, site independence, ...

- Bi-directional index “synchronization” is needed.
How implement ?

- **Postgres BDR** has a loosely coupled shared-nothing multi-master design.

- Bi-directional replication **(BDR)** is an extension package of PostgreSQL version 9.4 and higher.
  - can be introduced into standard PgSQL by “CREATE EXTENSION bdr” command

- As BDR is based on the PgSQL **logical replication**, data will be modified by “**row-based replication**”, neither by high-level **statement-based** nor by low-level **log-based** manners.
  - SQL statement based $\rightarrow$ via SQL proxy, such as “Pgpool-II”
    - cf. Trigger based $\rightarrow$ Daemons run on both C/S, such as “Slony-I” & “Bucardo”
  - Log (i.e. binary block) based $\rightarrow$ PostgreSQL streaming replication

- BDR still has some constraints:
  i. “**Primary key**” must be defined in every table. “**OID**” cannot be used.
  ii. Data Definition Language (DDL) commands are not fully supported in BDR.
    - e.g. CREATE/DROP/ALTER DATABASE/ROLE/USER/GROUP/TABLESPACE/TABLE_TYPE ...
  iii. BDR solves transaction conflicts using a simple “**last-update-wins**” strategy.
    - Replication interval is set to **2 seconds** for better throughput. $\rightarrow$ quasi- Real-time sync.
LHD data system

- Different from ITER UDA, LHD adopts “recommend” type of Facilitator model.
  - 2-step data access with different protocols
- LHD storage has 3 layers: ① SSD array, ② HDD raid cluster, ③ Blu-ray library.

- Archiving data can be queued and also served in every stage.
- Real-time data streaming uses a simple C/S model and served independently from the discrete data service.
“Facilitator Model” for tripartite systems

Broker type

1. advertise
2. query
3. request
4. answer
5. answer

Recommend type

1. advertise
2. query
3. recommend
4. request
5. answer

Recruit type

1. advertise
2. query
3. request
4. answer

Client
Facilitator
Server

12th IAEA-TM CODARP in Daejeon, 13-17 May 2019
LHD data system having 3 remote sites (now)

- Proxy cannot relay requests if primary DB service or network is down, even though archiving data can be queued at remote archivers.
LHD data system having 3 remote sites (mod.)

- If accident happened, each remote site can operate separately from the primary storage & index DB.
- When connection is back, queued data & index changes will be re-synchronized.

![Diagram of LHD data system with 3 remote sites](image)
Performances of Postgres BDR

- BDR throughputs have been investigated between NIFS, Toki and REC, Rokkasho.
  - **Round-trip time = 16.2 ms**, connected via 20 Gbps – 100 Gbps – 10 Gbps link
  - **Postgres BDR 9.4 servers:**
    - cpu: Xeon E5-2650 v4 2.2 GHz, 12c/24t
    - mem: 128 GB
    - xfs: Samsung NVMe SSD 960 PRO 512GB

- Replicating a single record may take **negligible small time (<< RTT)** on average for usual operations, excepting 2 second queuing.

<table>
<thead>
<tr>
<th>Table Name (5)</th>
<th># of records</th>
<th>elapsed time</th>
<th>per record</th>
<th>note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ex_Note (5)</td>
<td>144 772</td>
<td>243.6 s</td>
<td>$16.8 \times 10^{-4}$ s</td>
<td>--inserts</td>
</tr>
<tr>
<td>↑ BDR</td>
<td>144 772</td>
<td>23.1 s</td>
<td>$1.60 \times 10^{-4}$ s</td>
<td>(--copy)</td>
</tr>
<tr>
<td>↑ no BDR</td>
<td>144 772</td>
<td>0.978 s</td>
<td>$6.76 \times 10^{-6}$ s</td>
<td>(local)</td>
</tr>
<tr>
<td>Setup (167)</td>
<td>11 577 821</td>
<td>971.6 s</td>
<td>$0.84 \times 10^{-4}$ s</td>
<td>(local)</td>
</tr>
<tr>
<td>↑ no BDR</td>
<td>11 577 821</td>
<td>62.15 s</td>
<td>$5.37 \times 10^{-6}$ s</td>
<td>(local)</td>
</tr>
<tr>
<td>Index (22)</td>
<td>207 911 053</td>
<td>35 654 s</td>
<td>$1.72 \times 10^{-4}$ s</td>
<td>-F c -j 3</td>
</tr>
<tr>
<td>↑ no BDR</td>
<td>237 544 798</td>
<td>581.5 s</td>
<td>$2.45 \times 10^{-6}$ s</td>
<td>-F c -j 3  (local)</td>
</tr>
</tbody>
</table>
Conclusions and Future works

- In order to put the replicated data repositories of practical use for massive data analyses, metadata Index DB should be also replicated for each repository site.

- Considering the compatibility of PostgreSQL, bi-directional replication extension Postgres BDR has been investigated and tested by using LHD Index data and FVL environment.

- BDR performance seems sufficient for usual data operations, excepting some global DDL commands.

In future works,

- A selection scheme for the most appropriate data repository site will be implemented soon in data retrieving client API and tested on LHD & FVL.
  - List of possible Index DBs can be stored and served by Index DB itself.
  - The best server can be found by practically measuring the network round-trip time (RTT) between C/S.
    - e.g. ICMP echo reply (ping) or TCP SYN+ACK response can be used.
    - DNS top domains or GeoIP resolvers provide answers with a very limited precision.
Thank you!