PLCC Cluster Manual

Table of Contents

About

This is a short manual for *PLCC cluster*, a collection of scripts which allows you to update the PTGL v3 server with fresh PDB data.

PLCC cluster runs the normal *plcc* java software for many PDB files on a computer cluster or on several machines in a LAN.

Note that the update is independent of the PTGL server, and this manual assumes that the update is run on different machines. It would take way too long to run the update on a single machine, and it would make the PTGL server very slow for users while the update is running. You can copy the data (and the database) to the PTGL server after the update.

Versions

Available versions

Using PLCC cluster, you can run an update of the PTGL in different technical setups:

1. SINGLE: on a single machine (not recommended, most likely takes several weeks)
2. LAN: on some machines connected via LAN, which have a shared network drive (but no queue management software or any other cluster stuff installed)
3. CLUSTER: on a computer cluster which runs the openPBS queue. For this, two sub versions exits:
    1. CLUSTER_MPI: using openMPI. One job contains many (like 200) PDB files, and all cores

of the node which is assigned the job work on them until all are done. The job is done when all PDB files are done.

2. CLUSTER_NOMPI: without openMPI. Each job contains only a single PDB file, and MPI is not used.

Which version to use

Here is a rough comparison of the versions:

	System setup	Job management	Speed
SINGLE	Very easy	None	Very poor
LAN	Easy	None	Medium to OK
CLUSTER_MPI	Very complicated, MPI-specific tech problems	Full queue support, wall time, etc	Very good
CLUSTER_NOMPI	Complicated	Full queue support, wall time, etc	Very good

As of April 2015, we use the cluster version without openMPI (3.2 in the list above) to update the PTGL, and this is the recommended way to do it.

Preface: PTGL server first-time setup

This section describe how to install the required standard software on the PTGL server. It is only needed when you want to setup the PTGL server from scratch, i.e., you wiped the OS and re-installed a new, bare linux on it. What you need to install and do is the following:

- Install & secure the server as appropriate: ssh access, firewalling before it, etc. Keep in mind it will be a public server, exposed to the internet.
- Create a unix user *ptgl* (or use you own, but only if you are the only one responsible for the server). This account will **not** run the web or database server (the standard accounts created for these services under linux will do this), but it can be used to quickly add single PDB files to the database and manually run advanced SQL queries against the DB.
  - Install DSSP, PLCC and SplitPDB to this account. I usually checkout the source code repo to do this.
  - Run `plcc NONE --create-config` once to have it createits config file at ~/.plcc_config, and edit the config file with the correct database server credentials.
- Install the Apache 2 web server, and configure it:
  - set a web directory for the PTGL3 (e.g., /srv/www/htdocs/PTGL3/)
  - a vhost for the PTGL server may make sense

  - make sure that php-zip support and -postgresql support is installed
- Install the PostgreSQL database server, and configure a *vplg* database user, and a database owned by him. Enable local password authentication over TCP/IP for him, deny any other login methods (see pg cfg files *postgresql.conf* and *pg_hba.conf* in */var/lib/pgsql/data/*).

Running the Update

To run the update, you need to prepare some stuff first, independent of which method you intend to use. These preparations are describe in the following.

Common preparations

Before you start, ensure that PostgreSQL database server is installed correctly and running on the update machine or another server which is reachable from it. You should also create a database user and a database for update there (usually user *vplg*, database *vplg* owned by him). Then do the following:

- **Install PLCC cluster**: On the update machine (cluster head or whatever machine, usually NOT the PTGL server!), copy the plcc.jar file, splitpdb.jar file and the 'lib/' directory from your VPLG installation to the 'plcc_run' subdirectory. (You can also use subversion to get the latest version onto that machine.)
- **Install DSSP**: Install dsspcmbi on the machine, from the DSSP website
- **Get PDB data**: Get a copy of the PDB via rsync or update your local copy if required. You can use the 'update_files_via_rsync.sh' script for this.
- **Get DSSP data**: Get a copy of the DSSP via rsync or update your local copy if required. If you do not do this, the update procedure will have to run dsspcmbi for each PDB file to generate the DSSP data, which takes some additional time. It is recommended to download the data instead!
- **Update PLCC Cluster config**: Edit the config file 'settings_statistics.cfg' to match your local settings. Be sure to set the paths to dsspcmbi, and to the PDB and DSSP data. Also the plcc command line options should be chosen with care, especially output directory. Ensure it is reachable from all machines and writable for the user who runs the update.
- **Update PLCC config file**: Edit the PLCC config file (at '~/.plcc_settings', run 'plcc --help' once if you do not have that file yet) and configure the connection to the database server (machine, port, user, password).
- **Prepare database**: Change to plcc_run/ and execute './plcc NONE --recreate-tables' once to create the DB structure (and test DB connectivity). WARNING: This will delete all contents from the database in case anything is in there already.

Running the update using the CLUSTER_NOMPI version

To run the update once you have prepared everything, do the following:

- On the update machine, change into the *plcc_cluster* dir, then into OpenPBD_NoMPI_version.
- Run the job submit script: ./start_all_jobs_on_all_nodes_whole_PDB_openPBS_noMPI.sh

Watch the cluster status using Ganglia or other monitoring software, and watch the PDB job queue using the openPBS commands to ensure update is running. Nodes should pikcup load. Also watch for output and error files of openPBS and check them for errors. If everything seems ok, leave it running and check back once per day.

After the update

It depends on your setup what to do after the update, but usually you will need to do these things to get the PTGL server up(dated):

- **Copy data files to PTGL web directory**: copy the data to the PTGL server, usually via rsync. This is a lot of data to copy, and will take many hours even over a fast network. Ensure that the target disk has enough free space and inodes! A lot of small files get created there! Check the number of inodes now. It is advised to have a separate disk for the data dir, and format the disk for small files (see mkfs documentation). The default directory where to copy the data is *<PTGL3_web>/data/*, which should be somewhere in the web root (usually */srv/www/*). The command could look like this:

  ```
  root@ptgl3:> /usr/bin/rsync -rhe ssh
  youruser@192.168.185.11:/shares/modshare/vplg_all_nodes_output
  /srv/www/PTGL3/newdata/ 2>&1  > /root/ptgl3_rsync_MAI-08-
  2015.log
  ```

- **Copy database contents**: You will have to export the database, copy the dump file to the live database system the PTGL server uses, and re-import the data on it. With the currently used PostgrSQL version, it works like this:

  ```
  # dump the database, on the update db server (ixion41 atm):

  postgres@clusterdbsrv>: pg_dump
  --file=/shares/modshare/vplg_db_dump/vplg.dump vplg

  # copy the file to the PTGL3 server. On the PTGL3 server:

  root@ptgl3:> rsync -rhe ssh
  youruser@192.168.185.11:/shares/modshare/vplg_db_dump/vplg.dump
  ```

```
/tmp/
```

- **Load DB on PTGL server**: You have to provide an empty *vplg* database owned by the user *vplg*, so drop the old one or, way better, just rename it in case stuff goes wrong while importing the new data afterwards. Then, you can re-import the database on the production db server:

  ```
  # re-import the database on the PTGL3 server
  ```

  ```
  psql@ptgl3>: psql vplg < /tmp/vplg.dump
  ```

- **Run maintenance PHP scripts on PTGL web server**: You should run the maintenance tasks in *maintenance.php* from a web browser. Also check for possible warnings (e.g., missing file system permissions on *temp_downloads* folder) which may be listed on that page.

That's it. The server should be up and running.