

MySQL on GPC

Ramses van Zon

SciNet HPC Consortium

16 October 2013

Data in all shapes and forms:

- Standard binary data (hdf5, netcdf, grib2, jpg, pdb, ...)
- Standard/semi-structured text data (xml, json, csv, fasta)
- Proprietary binary data format
- Unstructured text data (logs, output and error files)
- **Databases**

Database: an organized collection of data for storing, managing and retrieving information

- Often back-end to a website or application.
- Typically structured as tables.
- Each row expresses a *relation* between columns.
- SQL: Structured Query Language.
- Implementations often have access control on databases and tables.

Exception to this structure: *NoSQL*. Not in this TechTalk.

Why use a relational database?

- You like/are familiar with SQL.
- Application relies on a RDBMS.
- You want to be able express complex queries, such as:

List the directories of stored simulations in which the pressure was between 1 and 2 atm and the number of water molecules was between 4,000 and 50,000.

Why use a relational database?

- You like/are familiar with SQL.
- Application relies on a RDBMS.
- You want to be able express complex queries, such as:

List the directories of stored simulations in which the pressure was between 1 and 2 atm and the number of water molecules was between 4,000 and 50,000.

```
SELECT B.directory FROM simulations_attributes A
INNER JOIN simulation_paths B WHERE
A.simulation_id=B.simulation_id AND
A.pressure>=1 AND A.pressure<=2 AND
A.molecules>=4000 AND A.molecules<=50000;
```

Why use a relational database?

- You like/are familiar with SQL.
- Application relies on a RDBMS.
- You want to be able express complex queries, such as:

List the directories of stored simulations in which the pressure was between 1 and 2 atm and the number of water molecules was between 4,000 and 50,000.

```
SELECT B.directory FROM simulations_attributes A
INNER JOIN simulation_paths B WHERE
A.simulation_id=B.simulation_id AND
A.pressure>=1 AND A.pressure<=2 AND
A.molecules>=4000 AND A.molecules<=50000;
```

Note that you rarely store your binary data in the DB, but you could store paths and filenames with metadata.

Some Relational Database Management Systems

- Commercial:
 - Oracle Database
 - IBM DB2
 - Microsoft SQL Server
 - SAP Sybase
- Open Source:
 - PostgreSQL
 - SQLite
 - MySQL

What is MySQL?



- Open-source RDBMS
- Was owned by MySQL AB, now by Oracle
- Often used by Free-software/open-software projects requiring RDBMS
- Command-line interface (`mysql`)
- Interfaces for C, python, php, perl, R, ...
- Third party graphical interfaces

Typical MySQL Deployment

- Database server runs as root on some server
- A 'root' user can setup privileges on the databases and tables
- Application/web service runs on client (sometimes server=client)
- and contacts the server as a non-root user through a port or socket.

Why This Setup Doesn't Work on SciNet:

- Don't want to maintain a database server accessible to 1000+ users:
 - **Scalability:** Many nodes may try to access DB
 - **Conflicts:** Users may run same application, access the same databases/tables.
 - **Security:** What if privileges are not set correctly?
 - **Stability:** If DB goes down, or becomes slow, all jobs go with it.
 - **File systems:** RDBMS may not play nice with file system (iops-heavy).
- Users cannot start root processes or setup privileges.

Why This Setup Doesn't Work on SciNet:

- Don't want to maintain a database server accessible to 1000+ users:
 - **Scalability:** Many nodes may try to access DB
 - **Conflicts:** Users may run same application, access the same databases/tables.
 - **Security:** What if privileges are not set correctly?
 - **Stability:** If DB goes down, or becomes slow, all jobs go with it.
 - **File systems:** RDBMS may not play nice with file system (iops-heavy).
- Users cannot start root processes or setup privileges.

Luckily it is possible to install MySQL in user space.

User space MySQL Deployment

- Database server runs as user on some machine.
- Creates its own 'root' user, sets up privileges on databases and tables
This is a database user called 'root', which does not have, and should not, have anything at all to do with the linux root user.
- Application runs on another or the same machine
Port numbers and sockets must be assigned properly to avoid conflicts with other users' databases
- Application contacts this server as a non-'root' user through the properly assigned port or socket.

- MySQL user deployment is a bit of a pain, and would mean a lot of duplication of effort.
- So we've done that already, and wrote a few scripts to help you.
- They are in the 'mysql' module.

- MySQL user deployment is a bit of a pain, and would mean a lot of duplication of effort.
- So we've done that already, and wrote a few scripts to help you.
- They are in the 'mysql' module.

Let's demonstrate how this works...

SciNet's MySQL Module on the GPC: Basic Example

```
$ module load mysql
$ mysqlsetup
Creating directory (tree) $SCRATCH/.mysql-5.6.12/default for c
Creating directory $SCRATCH/mysqldata/default for databases
Creating local configuration
Initializing local mysql database
Enter mysql root password:
Confirm mysql root password:
Starting mysql service
131011 15:05:31 mysqld_safe Logging to '$SCRATCH/mysqldata/def
131011 15:05:31 mysqld_safe Starting mysqld daemon with databa
Setting password
Cleaning up
Stopping mysql service
131011 15:05:36 mysqld_safe mysqld from pid file $SCRATCH/mysc
$
```

SciNet's MySQL Module on the GPC: Basic Example

```
$ module load mysql
$ mysqlsetup
...
$ mysqlstart
131011 15:09:39 mysqld_safe Logging to '$SCRATCH/mysqldata/def
131011 15:09:39 mysqld_safe Starting mysqld daemon with databa
$ mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
...
mysql> \q
$ mysqlstop
131011 15:18:34 mysqld_safe mysqld from pid file $SCRATCH/mysc
$
```


What happened?

- Stuff was setup on scratch, so we can use this on compute nodes too.
- Setup root password
- Directories created:
 - `$$SCRATCH/.mysql-5.6.12/default`: contains configuration files
 - `$$SCRATCH/mysqlldata/default`: contains database files
- Configuration is such that non-default sockets and ports are used: multiple users can use this module simultaneously on devel nodes.
- Tip: do not use the root user, create a new user.

Note default in the above?

- It's possible to have other configurations, using different directories. This is required if you want to run a database server for each job.
- Each setup is called a 'profile'.
- Each profile has its own configuration and data directory.
- Each profile has its own root password.
- Data directory can even be on ramdisk!
(but you'll have to copy it if you want to save your results).

Using Profiles: Profile Example

```
$ module load mysql
$ mysqlsetup -P compute
$ mysqlstart -P compute
$ export MYSQL_HOME=$MYSQL_HOMES/compute
$ mysql -u root -p
mysql> CREATE DATABASE computedb
mysql> GRANT ALL PRIVILEGES ON computedb.* TO computeuser
mysql> FLUSH PRIVILEGES
mysql> \q
$ mysqlstop -P compute
```

Using Profiles: Ramdisk Example

```
$ mkdir -p /dev/shm/rzon/mysqldata
$ mysqlsetup -P ramdisk -D /dev/shm/rzon/mysqldata
$ mysqlstart -P ramdisk
$ mysqlstart -P default
$ mysqlstatus
PROFILE          PID
ramdisk          5243
default          5740
$ export MYSQL_HOME=$MYSQL_HOMES/ramdisk
$ mysql -u root -p
$ export MYSQL_HOME=$MYSQL_HOMES/default
$ mysql -u root -p
$ mysqlstop -P ramdisk
$ cd /dev/shm/rzon/mysqldata
$ tar cf $SCRATCH/ramdiskdb.tar .
```

Using Profiles: Ramdisk Example

Next run can just do:

```
$ module load mysql
$ cd /dev/shm/rzon/mysqldata
$ tar xpf $SCRATCH/ramdiskdb.tar
$ mysqlstart -P ramdisk
$ export MYSQL_HOME=$MYSQL_HOMES/ramdisk
$ ...
$ mysqlstop -P ramdisk
$ tar cf $SCRATCH/ramdiskdb.tar /dev/shm/rzon/mysqldata
```

Note that if there are several runs at the same time, you'll have to have different tar files on the file system and may have to figure out how to merge databases.

Using Profiles: Multi-node Example

A script that uses the mysql database but does not start it

```
#!/bin/bash
# file: compute.sh
module load mysql
export MYSQL_HOME=$MYSQL_HOMES/compute
export mysqlhost=$1
export thiscase=$2
mysql -h $mysqlhost -u computeuser computedb ... $thiscase ..
EOF
```

Using Profiles: Multi-node Example

```
#!/bin/bash
#PBS -l nodes=2:ppn=8,walltime=24:00:00
cd $PBS_0_WORKDIR
module load gnu-parallel/20130422 mysql
mysqlstart -P compute
export mysqlhost=$(hostname)
parallel -j2 --slf $PBS_NODEFILE --workdir $PWD <<EOF
compute.sh $mysqlhost 1
compute.sh $mysqlhost 2
compute.sh $mysqlhost 3
...
EOF
mysqlstop -P compute
#end of job script
```

```
$ qsub jobscript
```



Summary

- The `mysql` module allows you to run `mysql` server as a user.
- Without bothering other users.
- Can run multiple `mysql` servers using profiles.
(must use different directories).
- Can run on ramdisk.
- Can serve a multinode job
- Scripts from the `mysql` module:
 - `mysqlsetup`
 - `mysqlstart`
 - `mysqlstatus`
 - `mysqlstop`
- More detailed info to be put on the wiki.

Summary

- The `mysql` module allows you to run `mysql` server as a user.
- Without bothering other users.
- Can run multiple `mysql` servers using profiles.
(must use different directories).
- Can run on ramdisk.
- Can serve a multinode job
- Scripts from the `mysql` module:
 - `mysqlsetup`
 - `mysqlstart`
 - `mysqlstatus`
 - `mysqlstop`
- More detailed info to be put on the wiki.

Questions?