HiGRID Database

Diego Torres

Goal

Bring Advanced Power and Energy Program’s data into a distributed system

- Using a distributed system increases data organization
- Accessing data remotely
- Data can be shared
- The database can be expanded for newer files
- Files are secure

MySQL

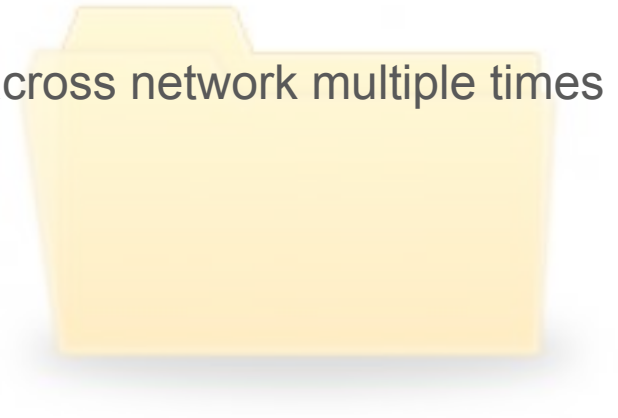
MySQL is a relational database management system(RDBMS)

- Stores tables of information
- The information has security checks for removing, adding or changing it
- Stores data in one central location
- Provides useful mechanisms for querying large sets of data with constraints
- Latency for querying data from remote location
- Database connectivity for applications is not always supported or free so accessing data can be inconvenient

Grabbag

APEP Grabbag is a network drive

- Used to share and communicate files
- Convenient and fast to use
- Useful for unique, single instance files
- Easy to expand
- Files that are exactly the same, yet distributed across network multiple times is poor for space and organization.
- Data can be susceptible to corruption
    - Malicious activity or accidental



MySQL vs Grabbag

Using a DBMS is more robust and generally accepted as the better way of storing sets of tabular data. However accessing the grabbag has it's performance benefits in the context of this research project.

- MySQL database is best for tabular datasets
- Grabbag is great for complex datasets
- MySQL is superior for managing and organizing the data
- Grabbag is more recognizable to existing members of APEP

…..

…….

MySQL W/ Grabbag

Considering each approach has its own benefits why not use both

- MySQL
  - Typical data used in HiGRID’s operation can be stored in the DB
- Grabbag
  - Complex data sets can be stored in the Grabbag

Initial Plans

Week	Module	API	DB
1	Renewable Generation	Read matlab code and design data functions	Design layout of tables
2	Renewable Generation	Write data retrieval functions, concurrently test program	Import tables with addition of each data retrieval script
3	Balance Generation	Read matlab code and design data functions	Design layout of tables
4	Balance Generation	Write data retrieval codet, concurrently test program	Import tables with addition of each data retrieval script
5	Cost Generation	Read matlab code and design data functions	Design layout of tables
6	Cost Generation	Write data retrieval function, concurrently test program	Import tables with addition of each data retrieval script
7	Dispatchable Load	Read matlab code and design data functions	Design layout of tables
8	Dispatchable Load	Write data retrieval function, concurrently test program	Import tables with addition of each data retrieval script
9	Migration of database onto a permanent server	Ensuring functions can call from new location	Moving tables, and finalizing table names, and columns
10	Final production server testing	Testing data retrieval functionality	Testing query performance

What still needs to be done?

As of 8/20/2018 :

Transfer existing database to a more permanent location

- Non-technical task but requires time to complete

Final server testing

- Having everyone use the server and running performance benchmarks, for optimization later on

How have things changed?

The **initial goal** : to solely use the MySQL database to store all of the relevant data the HiGRID uses to properly operate

Realization : .m(mat) files are difficult and weird to store on a database

Resolution : Use the Grabbag to supplement this area, where the database doesn’t work

Solution : Store location of mat files onto the database

Work Done

- **Week 1** - Continue files associated with the renewables
- **Week 2** - Finished files associated with renewables
- **Week 3** - Start dispatchable load module
- **Week 4** - Continue dispatchable load module
- **Week 5** - Finished dispatchable load module
- **Week 6** - Start Balance Generation module
- **Week 7** - Finish Balance Generation
- **Week 8** - Cost module completed
- **Week 9** - Work on mat files
- **Week 10** - Introduced directories in the database as well

Current *higrid* Database

60 Tables in MySQL

The database stores most files used within HiGRID

Table Names are the same as excel file names

Directory location of the matfiles (.m) are stored

Current *matfile* database

- This database stores some tables located within the .mat files
- *matfile* will be discarded because it doesn't play any role in future use
  - All contents of the .mat files will be stored in the Grabbag

How to access data?

- Accessing data is fairly simple
- More documentation can be found here
- Simple function principle
  - getData(func_name,{string_arg1,string_arg2,...})
  - A single MATLAB function is used to interface a developer of HiGRID and the Database. This 'getData' function allows specific commands and arguments to grab relevant data. The structure of this function is as shown
- example
  - getData(DB.getLocationStreamTable,{'tehachapi'})
  - In this example the getData function retrieves the excel Location_STREAM_hour for different locations. In this case the table data is retrieved for the location 'tehachapi'.

Adding new files to the database?

- Adding new files to the database is fairly straightforward
- A new *PyMatSQL Interface* class should be made access the data
- Pushing the data is simple with MySQL Data Import Wizard
- Altering a matlab-python interface class to recognize the new *PyMatSQL interface*

Permanent Database Server

- Need to find a permanent database server
  - Still under progress
- Identify some of the hardware specifications
  - Optimize for the given hardware constraints
- Make sure database permissions are set so that only certain people have the ability to alter the tables

Packaging

- There needs to be someway to package the python code so that it can be distributable
  - Libraries used in python code can be packaged with code
  - Ensures no installation of libraries from users of data retrieval code
  - Allows everyone to have the same version of libraries
- Connection of Matlab to Python
  - There should be a connection between matlab and python that is pre existent in MATLAB.

How will it be used?

```
%Load Wind Vectors

%database method

tehachapi = getData(DB.getLocationStreamTable,{'tehachapi'});

palmdale  = getData(DB.getLocationStreamTable,{'palmdale'});

beaumont  = getData(DB.getLocationStreamTable,{'beaumont'});

gmountain = getData(DB.getLocationStreamTable,{'gmountain'});

windvec1  = getData(DB.getMegaWattWindTable,{});

wind_local= getData(DB.getLocalWindTable,{});
```

Plans for the future

- The database is something that should be constantly updated upon discovery of a bug
- Expanding the database is simple
  - Including information specifically to the operation of HiGRID should be stored whenever possible
  - The grabbag should be used minimally
- Optimizing code when there are obvious signs of improvement

Info

- Official Documentation - documentation
  - Read up on how the system works and why kind of functionality it supports
- Contact - diegot1@uci.edu
  - Report any bugs or serious concerns to this email address
- A.P.E.P Website - http://www.apep.uci.edu/
  - Discover what is research organization is all about