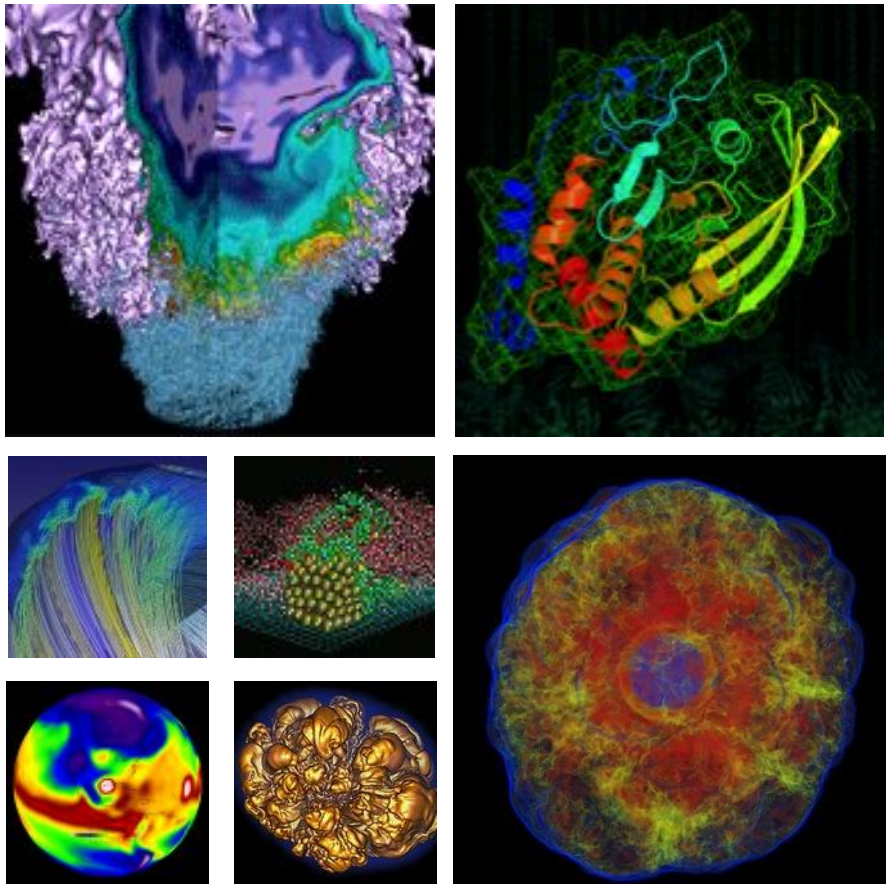
Databases



Wahid Bhimji
(wbhimji@lbl.gov)

https://www.nersc.gov/users/data-analytics/data-management/

Databases

- **Relational / SQL Databases**
  - NERSC host/support MySQL and PostgreSQL DBs for users
- **NoSQL / Schema-less Databases**
  - MongoDB

To request a database:
https://www.nersc.gov/users/science-gateways/science-gateway-databases/

SQL Databases

- **Good for:**
  - Your data is structured (you have a ‘Schema’)
  - Relational (tables of rows and columns)
  - Mid-Size, <=several GB in total
  - transactional operations (ensuring DB is consistent)
- Single databases on single servers so multiple connections not served in parallel.
- **PostgreSQL:** Object relational, some powerful features and extensions as well as SQL standards
- **MySQL:** Very popular, open-source, relational database

Accessing SQL DBs

- **Postgres:**

  `psql –h scidb1.nersc.gov yourdb –U dbuser`

- **Mysql:**

  `mysql yourdb –u dbuser –h scidb1.nersc.gov -p`

- **Very basic sql:**

  SET PASSWORD = PASSWORD(‘password');

  USE yourdb;

  CREATE TABLE yourtable (a_id INTEGER PRIMARY KEY ,b VARCHAR(10));

  SELECT * from yourtable WHERE yourtable.b = ‘bob’;

- Learning SQL try **http://sqlzoo.net/**

MongoDB

- **'NoSQL', document-oriented database**
- JSON-like documents (key: value)
- Queries are javascript expressions
- Memory-mapped files – queries can be fast
- Though not configured here for very frequent/ high-volume writes or very many connections
- **Good For:**
  - Un-Structured Data ('Schema-less')
  - Mid-Size to Large, e.g. 10 GB of Text

Accessing MongoDB

- Use mongo client

```
mongo -u yourdb_admin –p password
mongodb01.nersc.gov/yourdb
```

- Create a collection ; put a document in it and find it

```
doc1 = {name: “bob”, friends:5 }
yourdb.acollection.insert(doc1)
db.customers.find({name:”bob"})
```

- Use pymongo for Python:

```
import pymongo
client= pymongo.MongoClient(‘mongodb01.nersc.gov’)
client.admin.authenticate(yourdb_admin, args.passwd)
client.yourdb.acollection.insert([{"name": ”bob”, ”friends": 5}])
```

Parting Personal thoughts on I/O and databases

- **Access databases via command line or code or Science gateway apps**
- **When files and Databases (some personal observations)**
  - Massively parallel HPC programs -> Files
  - Instrument data distributed around the world -> Files
  - Large 100-1000 user collaborations -> Files
  - metadata (e.g. about conditions in which data was collected) -> Database (SQL if schema known)
  - Multi-source, aggregated, instrument metadata -> NoSQL DB

- **Database and file I/O documentaton:**
  - https://www.nersc.gov/users/data-analytics/data-management/

- **Database Request Form**
  - https://www.nersc.gov/users/science-gateways/science-database-request-form/