Introduction to MongoDB

NoSQL vs SQL

SQL vs NoSQL

http://www.sitepoint.com/sql-vs-nosql-differences/

Introduction to MongoDB from MongoDB shell

https://docs.google.com/presentation/d/1gLCq7JiGnULKRQopcINVko688z60Rw4s4-qKFs5UFxg/edit#slide=id.p

MongoDB real world example

...

Provide simulation runs provenance with Python

Provenance for simulations

- Simulations running concurrently on a cluster
- Each with different configuration parameters

Want to:

- Save intermediate results
- Save final results
- Keep track of configuration parameters
- Keep track of software properties

Simulator configuration command line arguments

aprun python simulator_cal.py --chtag LFI23S --input_cal='DX11D' --input_map_polarization='False' --datarelease='dx11_delta' --pencil_input='False' --init_dipole_fit='False' --input_map='/global/project/projectdirs/planck/data/mission/DPC_maps/dx11_delta/lfi/LFI_SkyMap_%03d_1024_DX11D_full.fits' --mask_filename='/global/project/projectdirs/planck/software/zonca/dev/chi2cal/destripingmask_30.fits' --scale_sol_dip_straylight='1.0' --remove_dipoles_signal='False' --remove_polarization='False' --scale_straylight='1' --preremove_sol_dip='False' --white_noise_scale='1.0' --ddx9data='False' --only_orb_dip='False' --nside='128' --pencil='False' --tag='full' --dipole_constraint='sol_dip' --unknown_straylight='False' --precond='True' --destripe='True' --correct_main_beam_eff='False' --straylight='False'

Parse as a Python dictionary

```
config = {'chtag': 'LFI23S', 'input_cal': 'DX11DDVV', 'destripe': True,
'init_dipole_fit': False, 'ddx9data': False, 'dipole_constraint': '',
'input_map_polarization': False, 'correct_main_beam_eff': False,
'pencil': False, 'remove_dipoles_signal': False, 'precond': True,
'datarelease': 'dx11_delta', 'unknown_straylight': False,
'preremove_sol_dip': False, 'only_orb_dip': False, 'remove_polarization':
False,    'straylight': False, 'tag': 'full', 'white_noise_scale': 0.0}
```

Configure pymongo

```
client = pymongo.MongoClient('yourmongodb.instance.com')
db = client["databasename"]
db.authenticate("username", "password")
gainsdb = db.gains
```

Store results

```
config["pids"] = my_numpy_array.tolist()
config["date"] = datetime.datetime.now()
config["iteration"] = iteration_number
gainsdb.insert(config)
```

Software provenance

```
from git import Repo

repo = Repo("path/to/folder")

git_hash = repo.head.commit.hexsha


config["code_hash"] = git_hash
```

Close connection

```
client.close()
```

Query results

```
git_hash = gainsdb.find({'chtag': 'LFI23S', 'precond': True})

for record in records:

    gains = record["gains"]
```

Other examples

- Natural language processing:
  - ingest paper abstracts with metadata into MongoDB
  - update automatically every day
  - flexibility to add more metadata.
- Geophysics:
  - ingest earthquakes data into MongoDB every day
  - use 2D geospatial queries
  - add more servers as data grows.

Conclusion

- Provenance tracking for:
  - configuration parameters
  - software version
  - data
- Easy: dump a dict into MongoDB
- Powerful query language for analyzing results