Our experience of the NoSQL database integration in PanDA infrastructure

M. Grigorieva, M. Golosova

BigPanDA monitor

- Analyze job failures
- Monitor progress of analysis/activity of a PanDA resource
- Organize/visualize data

- Separates data access layer and visualization
- Built around common key PanDA objects:

  jobs, resources, etc.
- BigPanDAMon based on django Framework
- Runs on top of SQL DB backends
- Modular and reusable monitoring:

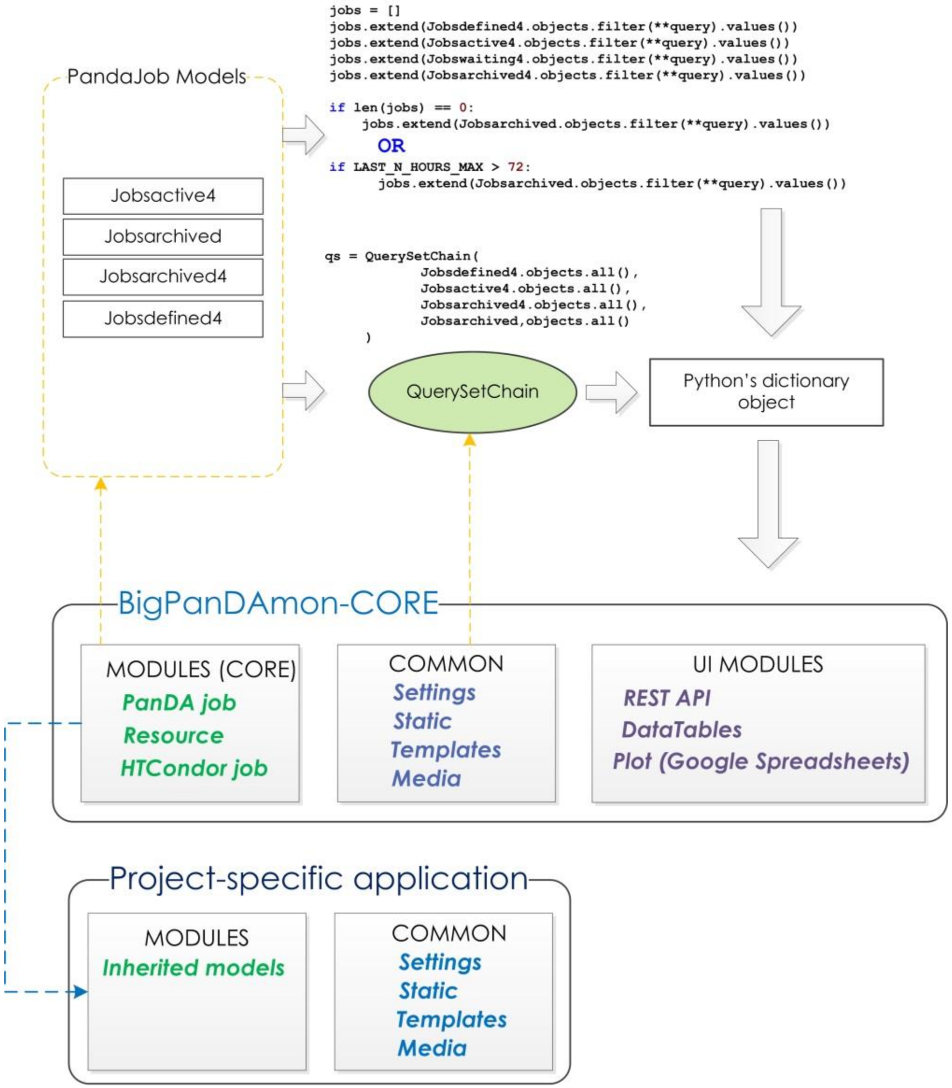
ATLAS EXPERIMENT HTCondor High Throughput Computing LSST Large Synoptic Survey Telescope

Goal

Adapt BigPanDA Monitor to work with both SQL and NoSQL DB backends
(SQL – operational data,
NoSQL – historical data):

Methods:

1. Enchance Django ORM to interact with both: SQL and NoSQL
2. Integration of **Hybrid SQL/NoSQL Storage** in BigPanDA Monitor



```
jobs = []
jobs.extend(Jobsdefined4.objects.filter(**query).values())
jobs.extend(Jobsactive4.objects.filter(**query).values())
jobs.extend(Jobswaiting4.objects.filter(**query).values())
jobs.extend(Jobsarchived4.objects.filter(**query).values())

if len(jobs) == 0:
    jobs.extend(Jobsarchived.objects.filter(**query).values())
      OR
if LAST_N_HOURS_MAX > 72:
      jobs.extend(Jobsarchived.objects.filter(**query).values())
```

```
qs = QuerySetChain(
            Jobsdefined4.objects.all(),
            Jobsactive4.objects.all(),
            Jobsarchived4.objects.all(),
            Jobsarchived,objects.all()
      )
```

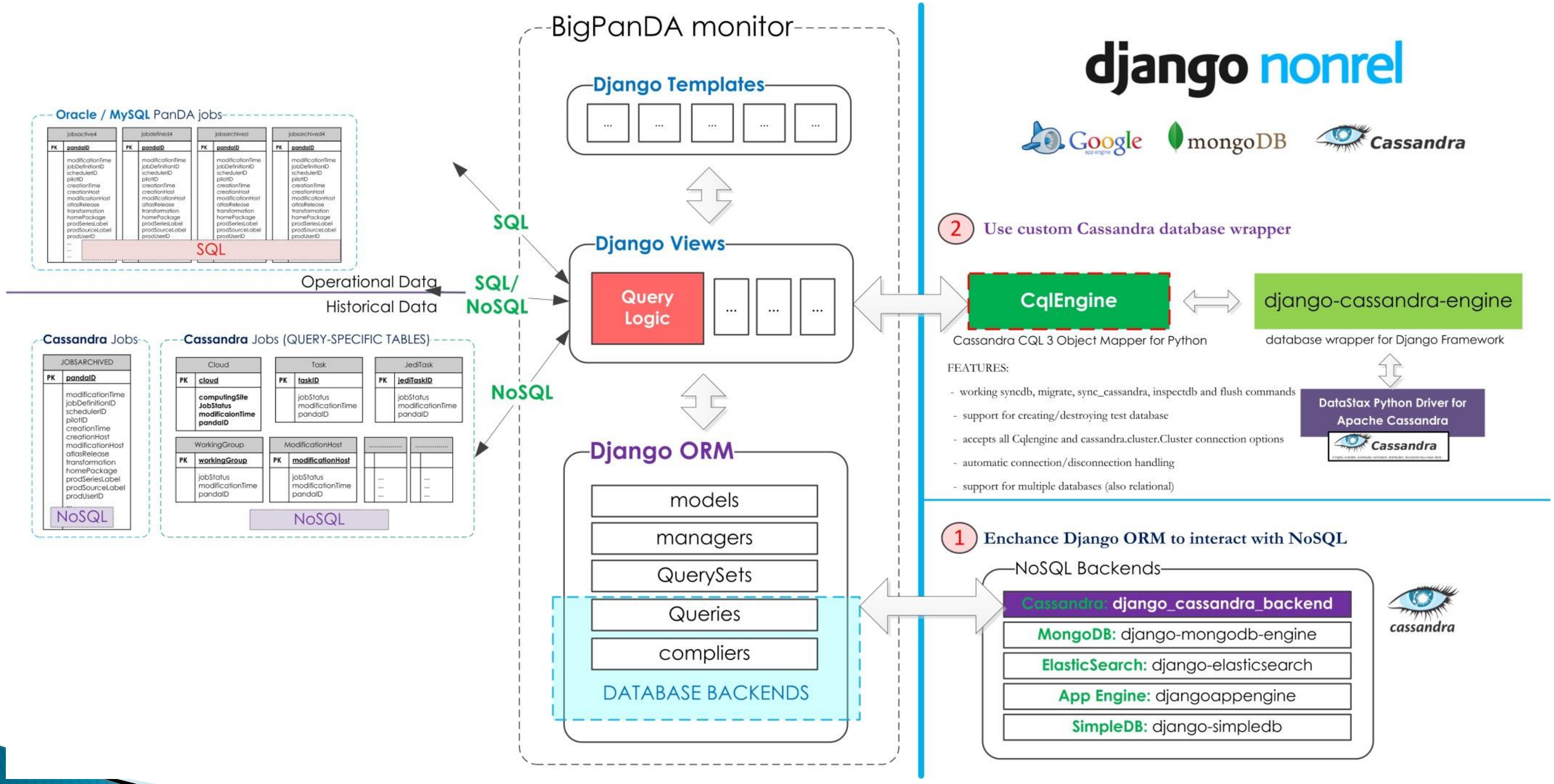
Django-nonrel

- Django-nonrel is an independent branch of Django that adds **NoSQL** database support to the ORM.
- The long-term goal is to add NoSQL support to the official Django release.

- **Django-dbindexer**
- use SQL features on NoSQL databases and abstract the differences between NoSQL databases
- denormalization, JOINs, and other important features
- Currently, this project is in an early development stage.

BigPanDA Monitor with NO SQL backend

1. Enchance Django ORM to interact with NoSQL
2. Use Cassandra database wrapper



Oracle / MySQL PanDA jobs

jobsactive4		jobdefined4		jobsarchived		jobsarchived4	
PK	pandaID	PK	pandaID	PK	pandaID	PK	pandaID
	modificationTime jobDefinitionID schedulerID pilotID creationTime creationHost modificationHost atlasRelease transformation homePackage prodSeriesLabel prodSourceLabel prodUserID		modificationTime jobDefinitionID schedulerID pilotID creationTime creationHost modificationHost atlasRelease transformation homePackage prodSeriesLabel prodSourceLabel prodUserID		modificationTime jobDefinitionID schedulerID pilotID creationTime creationHost modificationHost atlasRelease transformation homePackage prodSeriesLabel prodSourceLabel prodUserID		modificationTime jobDefinitionID schedulerID pilotID creationTime creationHost modificationHost atlasRelease transformation homePackage prodSeriesLabel prodSourceLabel prodUserID

SQL

Cassandra Jobs

JOBSARCHIVED	
PK	pandaID
	modificationTime jobDefinitionID schedulerID pilotID creationTime creationHost modificationHost atlasRelease transformation homePackage prodSeriesLabel prodSourceLabel prodUserID

NoSQL

Cassandra Jobs (QUERY-SPECIFIC TABLES)

Cloud		Task		JediTask	
PK	cloud	PK	taskID	PK	jediTaskID
	computingSite **JobStatus** **modificaionTime** **pandaID**		jobStatus modificationTime pandaID		jobStatus modificationTime pandaID

WorkingGroup		ModificationHost		....		....	
PK	workingGroup	PK	modificationHost				
	jobStatus modificationTime pandaID		jobStatus modificationTime pandaID		... ...		... ...

NoSQL

2 Use custom Cassandra database wrapper

Cassandra CQL 3 Object Mapper for Python

database wrapper for Django Framework

FEATURES:

- working syncdb, migrate, sync_cassandra, inspectdb and flush commands
- support for creating/destroying test database
- accepts all Cqlengine and cassandra.cluster.Cluster connection options
- automatic connection/disconnection handling
- support for multiple databases (also relational)

1 Enchance Django ORM to interact with NoSQL

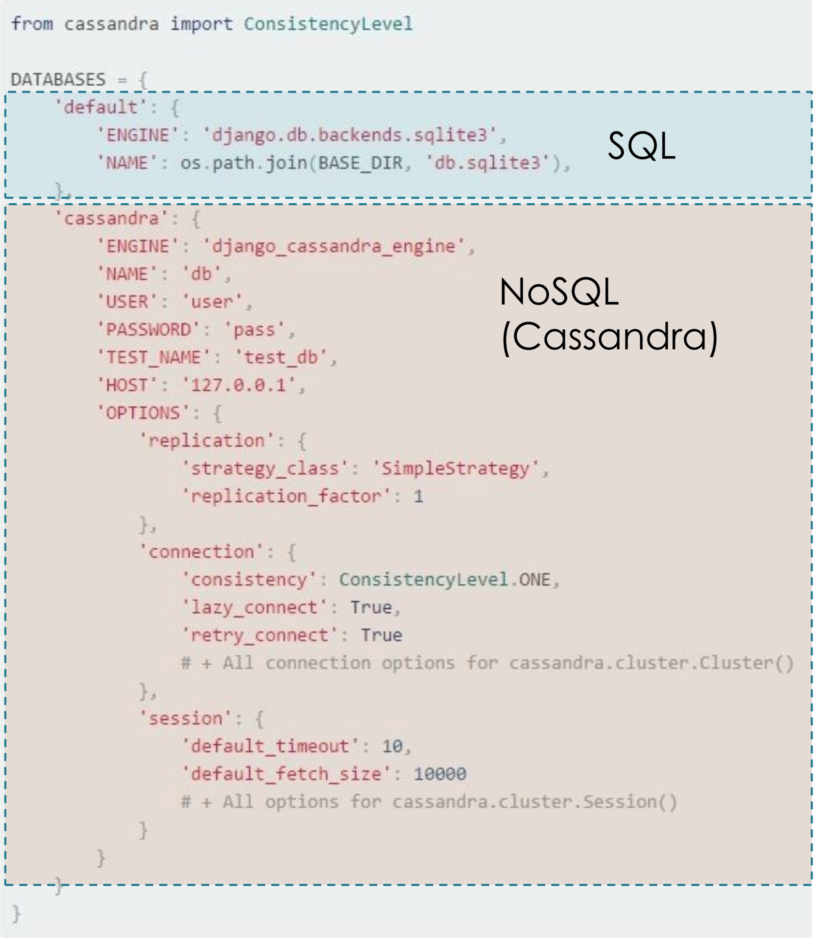
NoSQL Backends

- Cassandra: django_cassandra_backend
- MongoDB: django-mongodb-engine
- ElasticSearch: django-elasticsearch
- App Engine: djangoappengine
- SimpleDB: django-simpledb

Using django-cassandra-engine

settings.py

```
INSTALLED_APPS = ('django_cassandra_engine',) +
INSTALLED_APPS
```



```
from cassandra import ConsistencyLevel

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    },
    'cassandra': {
        'ENGINE': 'django_cassandra_engine',
        'NAME': 'db',
        'USER': 'user',
        'PASSWORD': 'pass',
        'TEST_NAME': 'test_db',
        'HOST': '127.0.0.1',
        'OPTIONS': {
            'replication': {
                'strategy_class': 'SimpleStrategy',
                'replication_factor': 1
            },
            'connection': {
                'consistency': ConsistencyLevel.ONE,
                'lazy_connect': True,
                'retry_connect': True
                # + All connection options for cassandra.cluster.Cluster()
            },
            'session': {
                'default_timeout': 10,
                'default_fetch_size': 10000
                # + All options for cassandra.cluster.Session()
            }
        }
    }
}
```

SQL

NoSQL
(Cassandra)

models.py

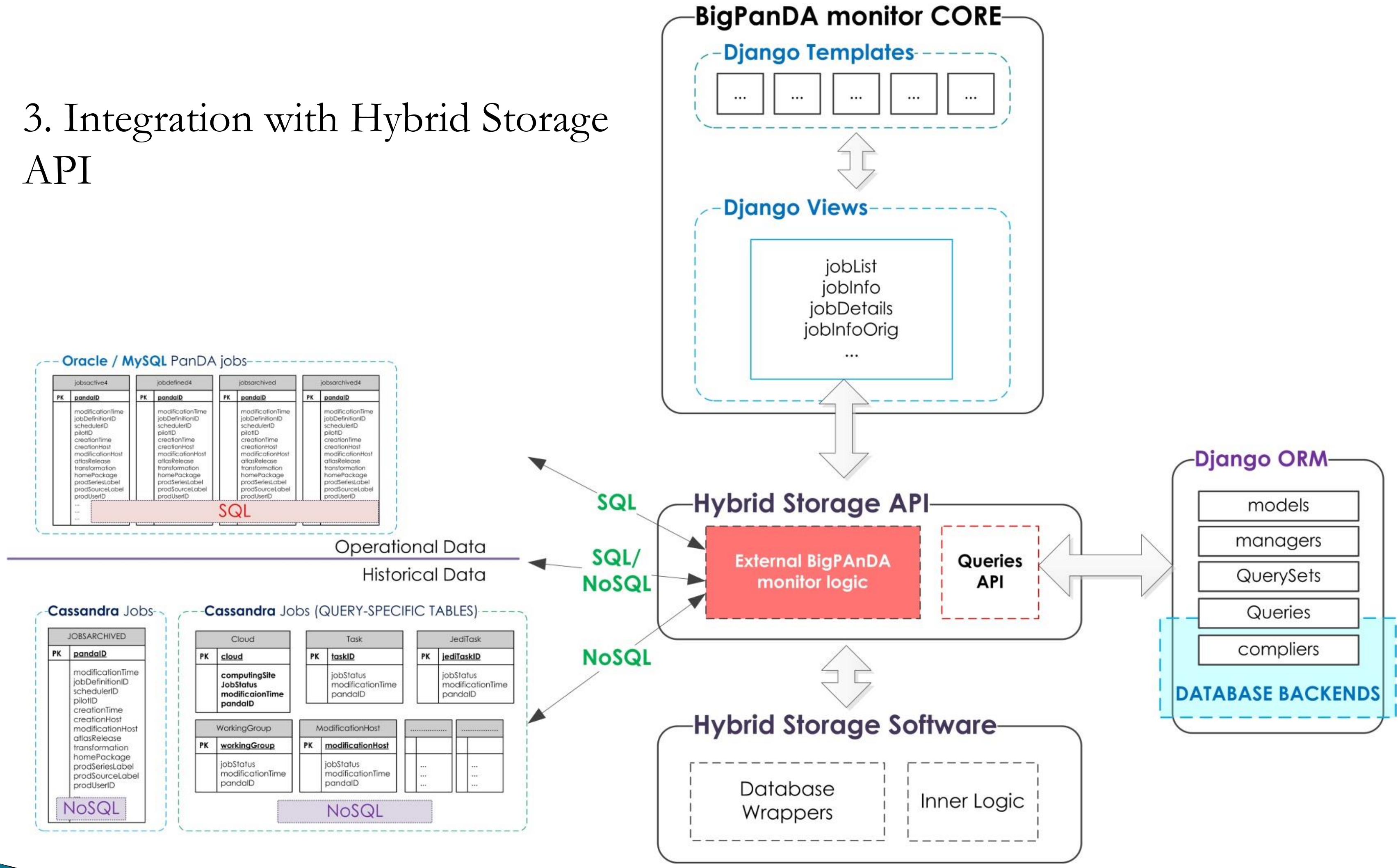
```
from cqlengine import columns
from cqlengine.models import Model

# main table
class PandaJobArchived(Model):
    pandaid = columns.BigInt(primary_key=True)
    modificationtime = columns.DateTime()
    jobdefinitionid = columns.BigInt()
    schedulerid = columns.Text(max_length=384)
    pilotid = columns.Text(max_length=600)
    creationtime = columns.DateTime()
    creationhost = columns.Text(max_length=384)
    modificationhost = columns.Text(max_length=384)
    ……………………………………
    ……………………………………
# dependent tables
class task_status(Model):
    task_id = columns.Integer(partition_key = True)
    job_status = columns.Text(partition_key = True)
    modification_time = columns.DateTime(primary_key = True)
    panda_id = columns.BigInt(primary_key = True)

…………
```

views.py

```
def jobList(request, mode=None, param=None):
       jobs.extend(Jobsdefined4.objects.filter(**query).values())
       jobs.extend(Jobsactive4.objects.filter(**query).values())
       jobs.extend(Jobswaiting4.objects.filter(**query).values())
       jobs.extend(Jobsarchived4.objects.filter(**query).values())
       jobs.extend(Jobsarchived.objects.filter(**query).values()) #SQL
                                    ↓
       # NoSQL (Cassandra) query
       jobs.extend(PandaJobArchived.objects.filter(**query).values())
```

BigPanDA Monitor with NO SQL backend

3. Integration with Hybrid Storage API



jobsactive4

PK	pandaID
	modificationTime jobDefinitionID schedulerID pilotID creationTime creationHost modificationHost atlasRelease transformation homePackage prodSeriesLabel prodSourceLabel prodUserID ... ...

jobdefined4

PK	pandaID
	modificationTime jobDefinitionID schedulerID pilotID creationTime creationHost modificationHost atlasRelease transformation homePackage prodSeriesLabel prodSourceLabel prodUserID

jobsarchived

PK	pandaID
	modificationTime jobDefinitionID schedulerID pilotID creationTime creationHost modificationHost atlasRelease transformation homePackage prodSeriesLabel prodSourceLabel prodUserID

jobsarchived4

PK	pandaID
	modificationTime jobDefinitionID schedulerID pilotID creationTime creationHost modificationHost atlasRelease transformation homePackage prodSeriesLabel prodSourceLabel prodUserID

JOBSARCHIVED

PK	pandaID
	modificationTime jobDefinitionID schedulerID pilotID creationTime creationHost modificationHost atlasRelease transformation homePackage prodSeriesLabel prodSourceLabel prodUserID ...

Cloud

PK	cloud
	computingSite **JobStatus** **modificaionTime** **pandaID**

Task

PK	taskID
	jobStatus modificationTime pandaID

JediTask

PK	jediTaskID
	jobStatus modificationTime pandaID

WorkingGroup

PK	workingGroup
	jobStatus modificationTime pandaID

ModificationHost

PK	modificationHost
	jobStatus modificationTime pandaID

....

	... ...

....

	... ...

Discussion topics

- Database performance tests : SQL - NoSQL, NoSQL - NoSQL
- Technology evaluation tests results for NoSQL databases: MongoDB, HBase, Cassandra, Dremel, CouchDB, MariaDB
- Experience of using Hadoop/Spark/MapReduce in PanDA Infrasctucture. Use cases.
- Foreseen performance and possible changes in PanDA Oracle archived database schema during/after the LHC Run2
- Query routing strategy in BigPanDA applications (BigPanDA Monitor in particular)
- How to implement cross database requests in heterogeneous architecture
- Strategies of the data modelling for NoSQL databases