

Table of Contents

Analyzing Millions of Gigabyte of LHC Data for CMS - Discover the Higgs on OSG.....	1
Introduction.....	1
Setup environment and prepare user area.....	1
Analysis code preparation.....	2
Dataset discovery.....	2
Analysis job execution on OSG.....	2
Finalize analysis: histograms.....	2
Monitoring.....	3

Analyzing Millions of Gigabyte of LHC Data for CMS - Discover the Higgs on OSG

Introduction

- Demo is showing analysis workflow: **discover the Higgs on OSG**
 - ◆ Analysis code was prepared in the CMSSW framework:
 - ◇ EDAnalyzer accessing reconstructed tracks and writing out ROOT file with histograms:
 - transverse momentum of reconstructed tracks: p_T [GeV]
 - di-track invariant mass: $m_{\mu,\mu}$ [GeV]
 - invariant mass of two di-track-objects: m_{ZZ} [GeV]
 - ◆ Dataset discovery: use DBS/DLS discovery page to check availability and location of datasample: **Higgs->ZZ->4mu**
 - ◆ Analysis job execution on the GRID using CRAB
- Used components of the CMS software and computing environment
 - ◆ **CMSSW**: CMS software framework and EDM
 - ◆ **DBS/DLS discovery webpage**:
 - ◇ **DBS**: Dataset Bookkeeping System, database of datasets and their files
 - ◇ **DLS**: Dataset Location Service, database of location(s) of datasets (*which dataset is available at which site*)
 - ◆ **CRAB**: CMS Remote Analysis Builder, user tool to submit and control batch analysis jobs to the GRID
- **CMSSW**:
 - ◆ Based on a bus model, user schedules modules which are run by the main framework application: *cmsRun*
 - ◆ User interaction with the framework application is done through configuration file called *parameter-set*
 - ◆ Parameter-set instantiates modules, instance is labeled by the *module label*
 - ◆ 4 different types of modules, two main user modules:
 - ◇ **EDProducer**: uses input from the event and produces new output which is stored in the event
 - ◇ **EDAnalyzer**: uses input from the event and performs operations on input, does not store anything in the event (*preparation shown in this demo*)
- Locations:
 - ◆ **User interface (UI)**: interactive login nodes at Fermilab (*UAF*)
 - ◆ **GRID sites**: one of the seven US-CMS T2 sites
 - ◇ University of Nebraska, Lincoln (UNL, OSG middleware)
 - ◇ University of Wisconsin, Madison (Wisconsin, OSG middleware)
 - ◇ California Institute of Technology (Caltech, OSG middleware)
 - ◇ Massachusetts Institute of Technology (MIT, OSG middleware)
 - ◇ Purdue University (Purdue, OSG middleware)
- Format conventions

Setup environment and prepare user area

- Setup CMS software environment

Analysis code preparation

- Code preparation

Dataset discovery

- Data discovery on DBS/DLS discovery page <http://cmsdbs.cern.ch/discovery/>
 - ◆ keyword search for:
 - ◇ **Higgs-ZZ-4mu**

Analysis job execution on OSG

- Setup CRAB
- prepare CRAB configuration file

```
[CRAB]
jobtype           = cmssw
scheduler         = condor_g

[CMSSW]
datasetpath       = <dataset name discovered with discovery page
pset              = <parameter-set for analysis code>
total_number_of_events = 100
events_per_job    = 10
output_file       = <histogram file name>

[EDG]
se_white_list     = <destination site>
virtual_organization = cms
lcg_catalog_type  = lfc
lfc_host          = lfc-cms-test.cern.ch
lfc_home          = /grid/cms
```

- create jobs

```
crab -create
```

- submit jobs

```
crab -submit all -continue
```

- status check

```
crab -status -c
```

- output retrieval

```
crab -getoutput -c
```

Finalize analysis: histograms

- post processing: add histogram files of individual jobs using ROOT tool

```
cd crab?_*_*/res
hadd histograms.root *.root
```

- [Show histograms](#)

```
root histograms.root
pt->Draw();
mmumu->Draw();
mzz->Draw();
```

Monitoring

- [DashBoard](#)
-

This topic: [Main > MidWestGridWorkshop2007](#)

Topic revision: r3 - 2007-03-25 - [OliverGutsche](#)



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)