Pacific Northwest
NATIONAL LABORATORY
Proudly Operated by Battelle *Since 1965*

CRT Data Analysis Tutorial

VIKAS BANSAL, MALACHI SCHRAM, JAN STRUBE

Pacific Northwest National Laboratory

CRT Data Analysis Tutorial @ US Belle II Summer School August 2016, PNNL

Glossary

- CRT – Cosmic Ray Test

- DIRAC – Distributed Infrastructure with Remote Agent Control
- API – Application Programming Interface
- REST – Representational State Transfer
- DISET – DIRAC Secure Transport
- DMS – Data Management System (inside DIRAC)
- WMS – Workload Management System (inside DIRAC)
- JDL – Job Description Language
- LFN – Logical File Name

- SE – Storage Element (resource that belongs to a site)
- CE – Compute Element (resource that belongs to a site)
- VM – Virtual Machine
- CLI – Command Line Interface
- CAF – CRT Analysis Framework
- TOP – Time of Propagation (sub-detector; sometimes called iTOP with i for imaging)
- DB – Data Base
- ROOT – RXXX Object Oriented Technology (RXXX – Rooted, Rapid)

DISCLAIMER

- This tutorial is specifically designed to showcase vanilla DIRAC using an example based on analyzing TOP CRT Data.
- Belle II grid computing provides an extension to DIRAC called BelleDIRAC. BelleDIRAC also provides an upper level interface to communicate to DIRAC. It is called gbasf2.
- gbasf2 tutorials have been given in previous B2GMs.
- In this school, we will learn a bit about DIRAC.
  - A precursor to gbasf2

TIP

- This tutorial shows some bash-shell code.
- Line/code in RED color can be executed in bash shell or in CLI where applicable.
  - Replace <var> with some actual value
  - E.g. <Job ID> by ’50’ etc. (without single quotes)

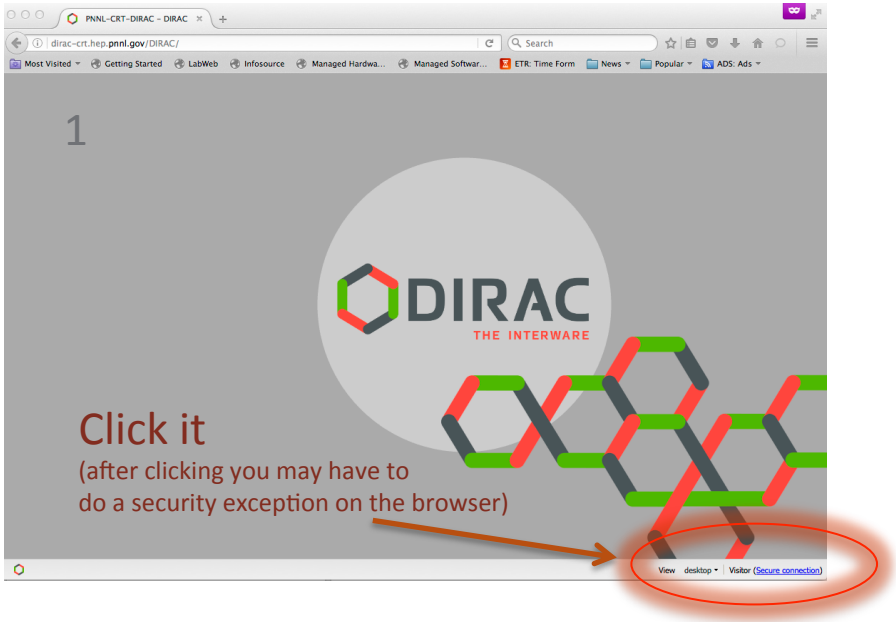
Pre-requisite: Certificate Registration

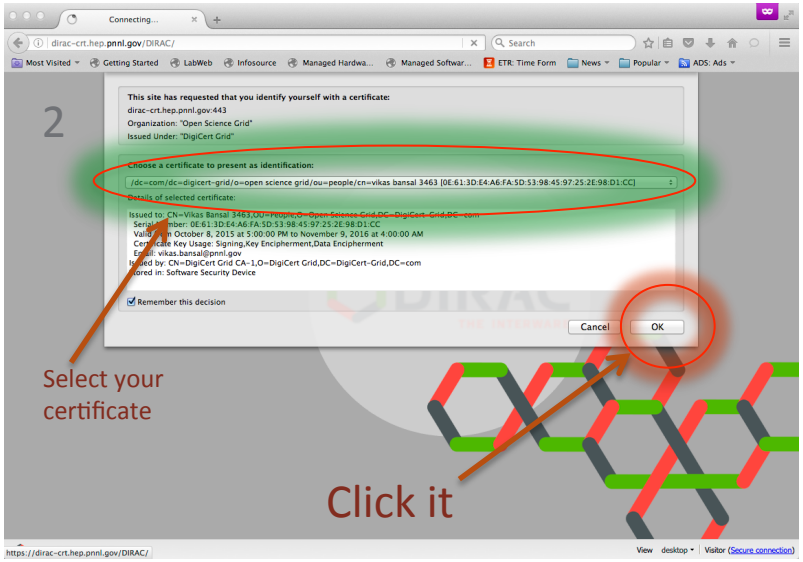
- We assume you already have a valid certificate and it is setup
  - In $HOME/.globus and is imported in firefox web browser.
- Register it with DIRAC by providing output of
  - $ openssl x509 –noout –in ~/.globus/usercert.pem -subject
- Provide a username, ideally same as you have at KEK

Certificate Registration : Check

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

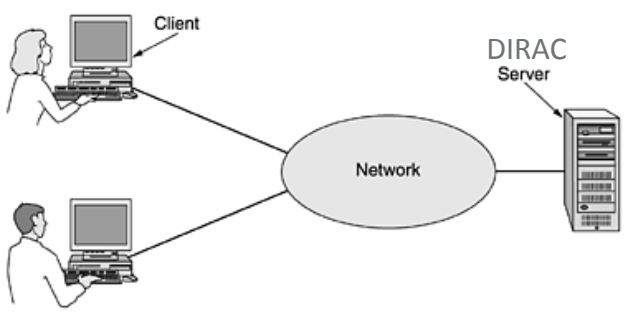
Goto: http://dirac-crt.hep.pnnl.gov/DIRAC/





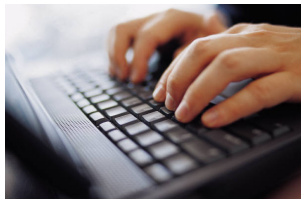
How to setup DIRAC Client?

- We will use DIRAC client to talk to DIRAC server
- Lets setup a DIRAC client. We will use bash shell



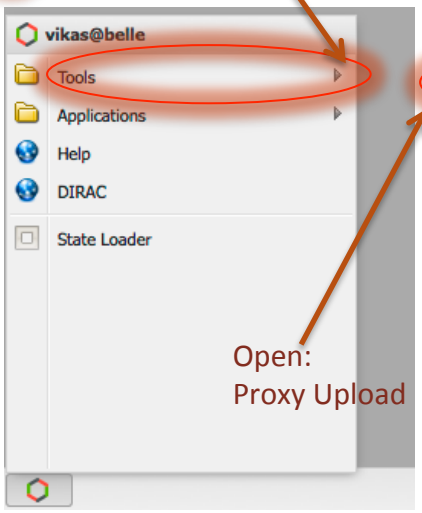
- Make a new directory – $ mkdir $HOME/CRT_PNNL_DIRAC_CLIENT

- Download and configure DIRAC client (One time step only)
  - $ cd $HOME/CRT_PNNL_DIRAC_CLIENT
  - $ wget http://belleweb.pnnl.gov/forDIRAC/setupDIRACClientFirstTime.sh
  - $ source setupDIRACClientFirstTime.sh
  - It will ask for your grid certificate password in the very end.

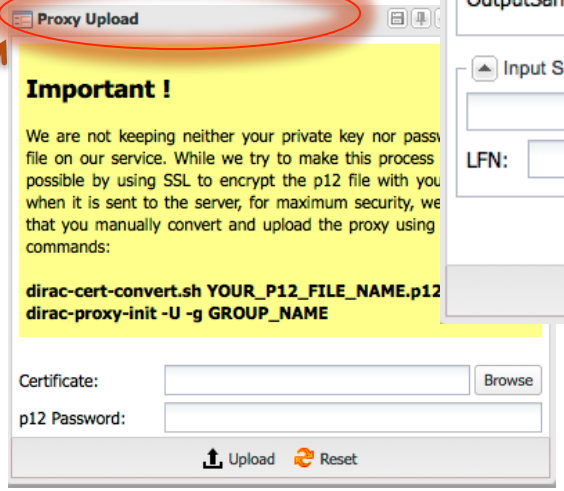


- Every time in a new shell, activate DIRAC client env.
  - $ cd $HOME/CRT_PNNL_DIRAC_CLIENT ; source bashrc
  - $ dirac-proxy-init –g belle -U

Submitting a simple job to DIRAC

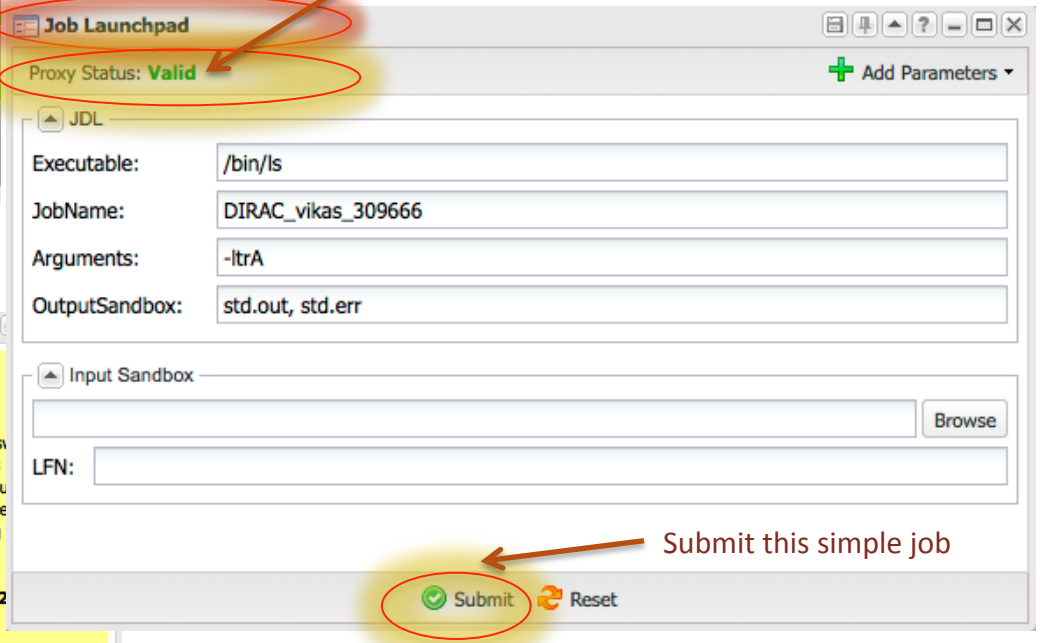


Open:
Proxy Upload



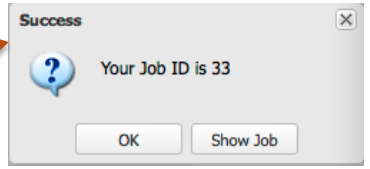
Open:
Job Launchpad

Make sure Proxy Status is VALID

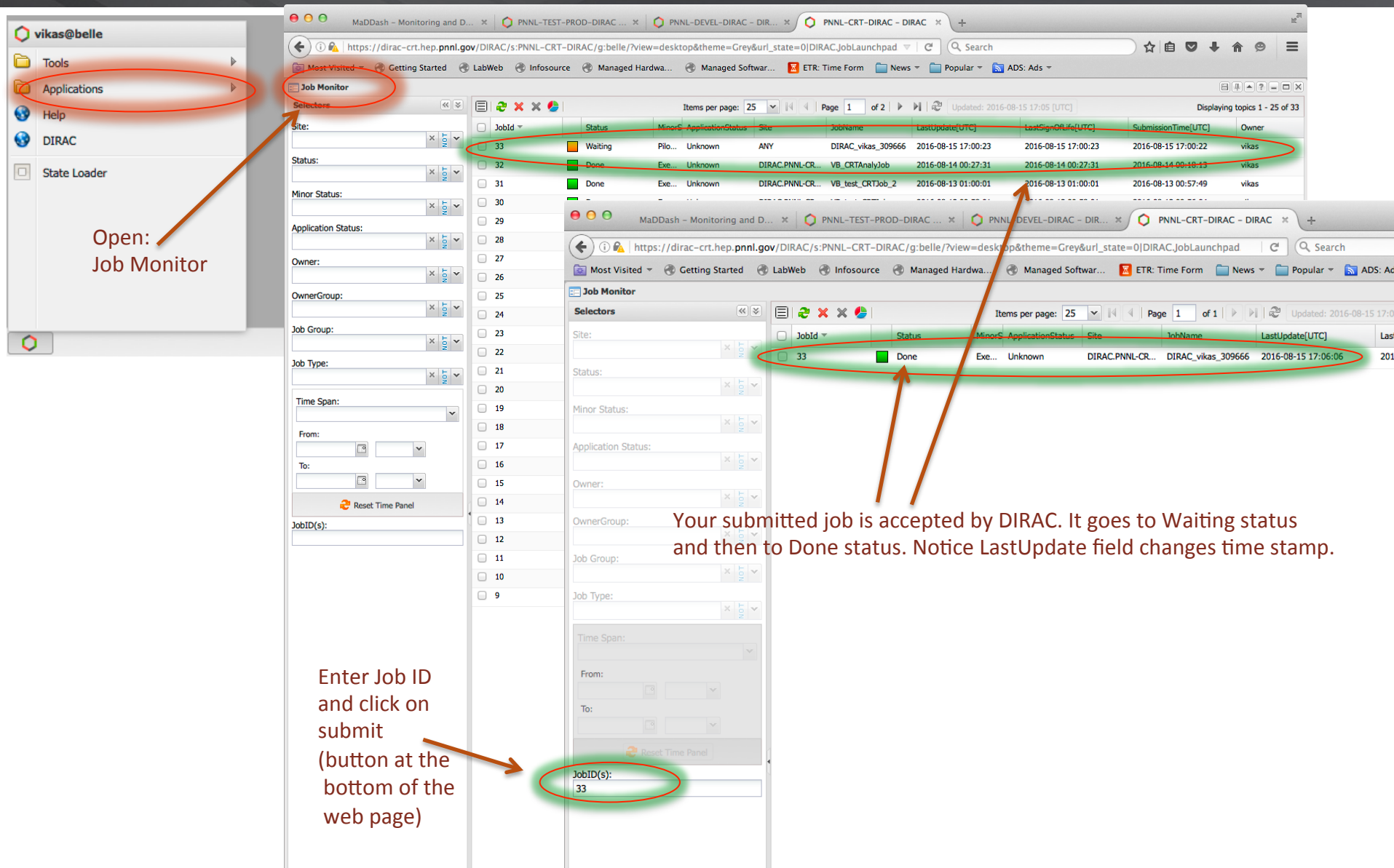


Submit this simple job

A pop-up confirming job submission and providing Job ID



Monitoring the DIRAC job



- Congratulations
- You successfully submitted a job to DIRAC

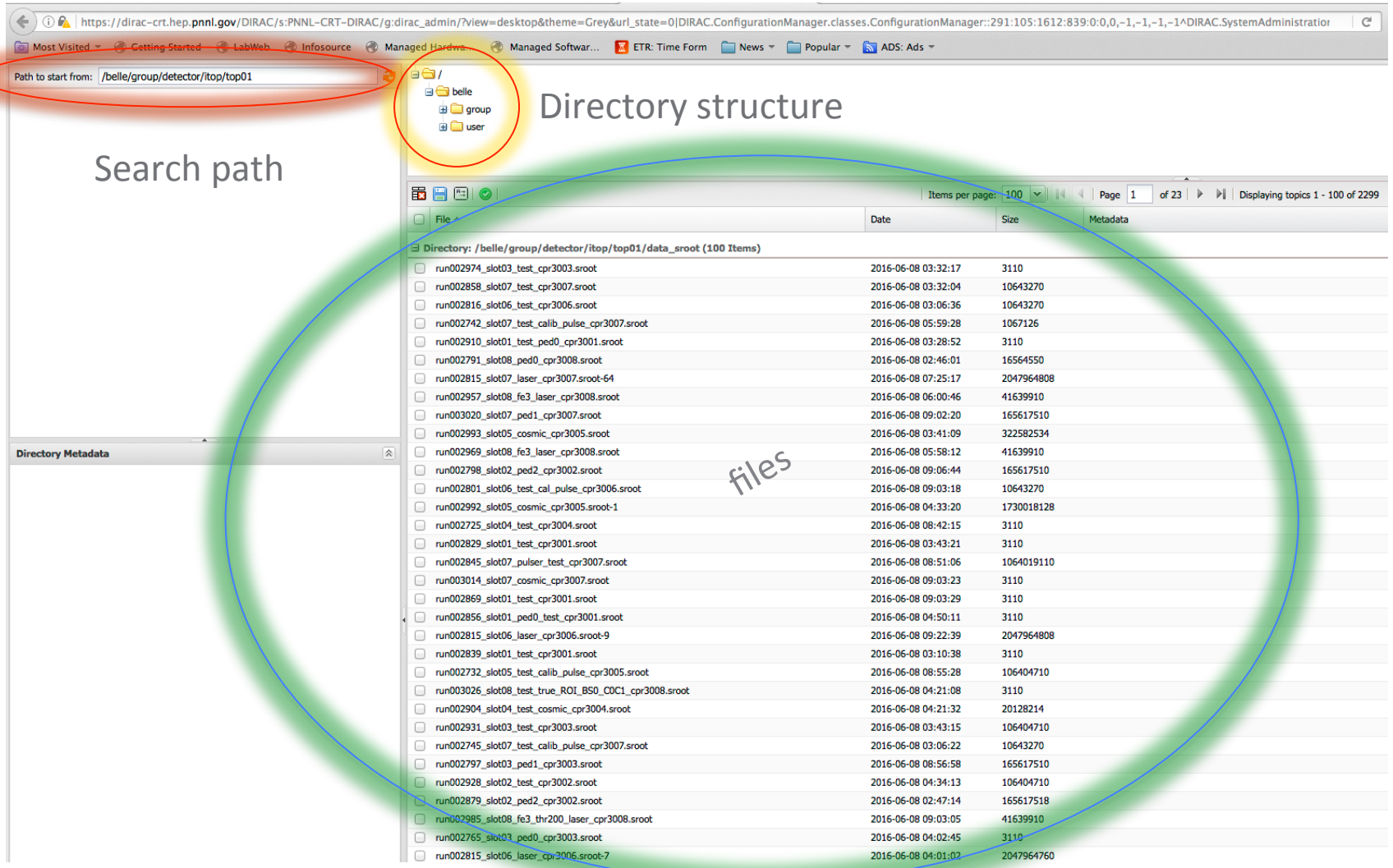
Where is CRT (Cosmic Ray Test) Data?

- The data is available at PNNL Storage Element (SE)
- One can use command line interface (CLI) or web app to browse

```
$ dirac-dms-filecatalog-cli
Starting FileCatalog client

File Catalog Client $Revision: 1.17 $Date:

FC:/> ls /belle/group/detector/itop/top01/data_sroot/run003118_slot07_laser_cpr3007.sroot
-rwxrwxr-x 0 schram belle 2047967918 2016-06-26 12:43:32 run003118_slot07_laser_cpr3007.sroot
FC:/> help
FC:/> exit
```

File Catalog Web App

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965



TOP CRT Data File Name Convention

- run003102_slot07_cosmic_test_cpr3007.sroot
- run003109_slot01_ped1_cpr3001.sroot
- run003110_slot01_ped2_cpr3001.sroot
- run003111_slot01_laser_cpr3001.sroot
- run003112_slot01_laser_cpr3001.sroot
  - run003112_slot01_laser_cpr3001.sroot-1

File format: .sroot (sequential ROOT)

Sometimes run continues in multiple files .sroot-1, .sroot-2, etc.

Which Run : run#

Which TOP module: slot#

Run Type : ped, laser, cosmic_test, etc.

Which COPPER board took data: cpr3XXX

- Lets get started with CRT Data Analysis with DIRAC

Thanks to Sam Cunliffe for testing this tutorial beforehand

Prepare files for Analysis Job on CRT Data

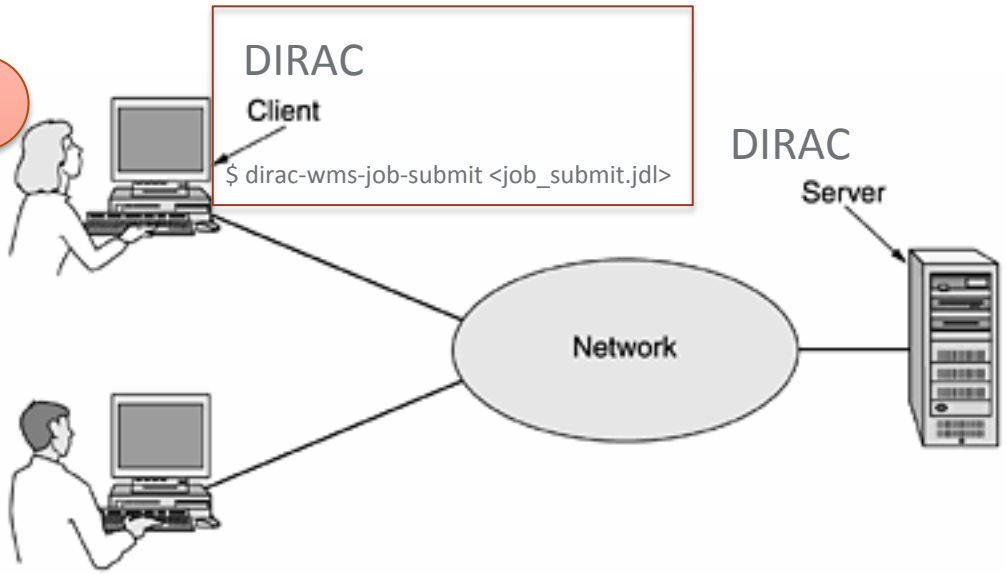
- Prepare a working directory
  - $ mkdir $HOME/CRT_PNNL_DIRAC_CLIENT/CRTDataAnalysis
  - $ cd $HOME/CRT_PNNL_DIRAC_CLIENT/CRTDataAnalysis
- Prepare analysis script
  - $ wget http://belleweb.pnnl.gov/forDIRAC/submitITopAnalysis.sh
  - This script is the same script one would run inside basf2 if data and basf2 were available locally
- Prepare DIRAC JDL
  - $ wget http://belleweb.pnnl.gov/forDIRAC/job_submit_01.jdl
  - This file, AKA DIRAC JDL, gives specifics of the job to DIRAC

DIRAC JDL

- DIRAC client submits a job, defined in DIRAC JDL, to the server.

Job_submit.jdl

```
JobName = ”CRTDataAnalyJob";

Executable = "submitITopAnalysis.sh";

Arguments = "run003149_slot07_laser_cpr3007.sroot";

StdOutput = "StdOut";

StdError = "StdErr";

InputSandbox = {"submitITopAnalysis.sh”};

OutputData = {"StdOut","StdErr","hits.root"};

OutputSandbox = {"StdOut","StdErr","hits.root"};

Site = "DIRAC.PNNL-CRT.us";

OutputSE = "PNNL-CRT-DIP";
```



TOP CRT data analysis example script

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

```
#!/bin/bash
#### Some hard coded variables
export TOPDATADIR="/hep/belle2/dip-crt/belle/group/detector/itop/top01/data_sroot"

######
#set up BASF2 environment for this job
echo "Setting up the topcaf env."
. /cvmfs/belle.cern.ch/sl6/tools/setup_belle2.sh
pushd /home/stru821/Belle2/itopAnalysis
setuprel
cd -
#######

filename=$1
inputRun=${TOPDATADIR}/${filename}

echo "Running basf2 to prepare pedestal file"
basf2 /home/stru821/Belle2/itopAnalysis/topcaf/example/topcaf_itop_sroot_ped.py --arg --inputRun=${inputRun} --arg --output=
$PWD/$(basename ${inputRun/.sroot/ped.root}) -l ERROR -n 500

echo "Now Running basf2 to output ROOT files with Belle2 objects."
basf2 /home/stru821/Belle2/itopAnalysis/topcaf/example/topcaf_itop_sroot_hits.py --arg --inputRun=${inputRun} --arg --ped=$PWD/$
(basename ${inputRun/.sroot/ped.root}) --arg --output=$PWD/$(basename ${inputRun/.sroot/_x.root}) -n 500

echo "Now analyze hits file and make plots"
analyzeHits_AsicByAsic $PWD/$(basename ${inputRun/.sroot/_x.root})
#### hits.root will be produced.
```

Use bash shell.
Define site specific variable for data location
(In general a script can be written without it.
Demo in second example)

Setup basf2 env. for running TOP CAF example

Accept which TOP CRT data file to analyze
Provide name as argument

Run TWO basf2 scripts on the input file (Daisy chain)

Finally make histograms on intermediate output files

Submit/Monitor a DIRAC job via CLI

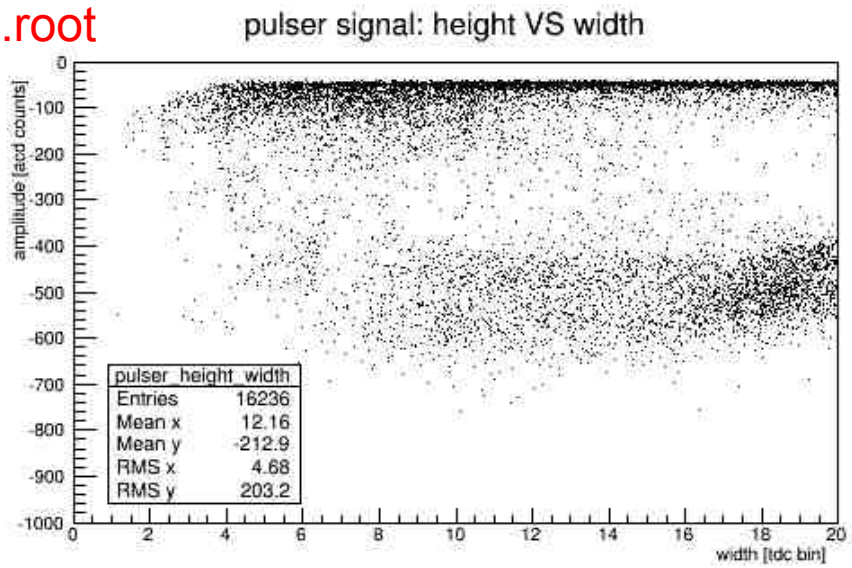
- $ cd $HOME/CRT_PNNL_DIRAC_CLIENT/CRTDataAnalysis
- Make sure the directory is prepared and has all relevant files

- $ dirac-wms-job-submit job_submit_01.jdl
  - Outputs : JobID = <Job ID>
- $ dirac-wms-job-status <Job ID>
  - JobID=<Job ID> Status=**Waiting**; MinorStatus=Pilot Agent Submission; Site=DIRAC.PNNL-CRT.us;
- $ dirac-wms-job-status <Job ID> (later time)
  - JobID=<Job ID> Status=**Running**; MinorStatus=Application; Site=DIRAC.PNNL-CRT.us;
- $ dirac-wms-job-status <Job ID> (later later time)
  - JobID=<Job ID> Status=**Done**; MinorStatus=Execution Complete; Site=DIRAC.PNNL-CRT.us;

Download/view Successful Job Output Files

- DIRAC server submits the user submitted job to an actual CE
  - Defined executable in JDL runs on the CE.
  - When executable has finished running, DIRAC flags the job as “Done”
  - If executable was successful, the output files are registered in a file catalog
- User can simply download the output files by
  - $ dirac-wms-job-get-output <Job ID>
- Analyze the output root file on your laptop/pc
  - $ root –l <Job ID>/hits.root

One of the many plots in hits.root

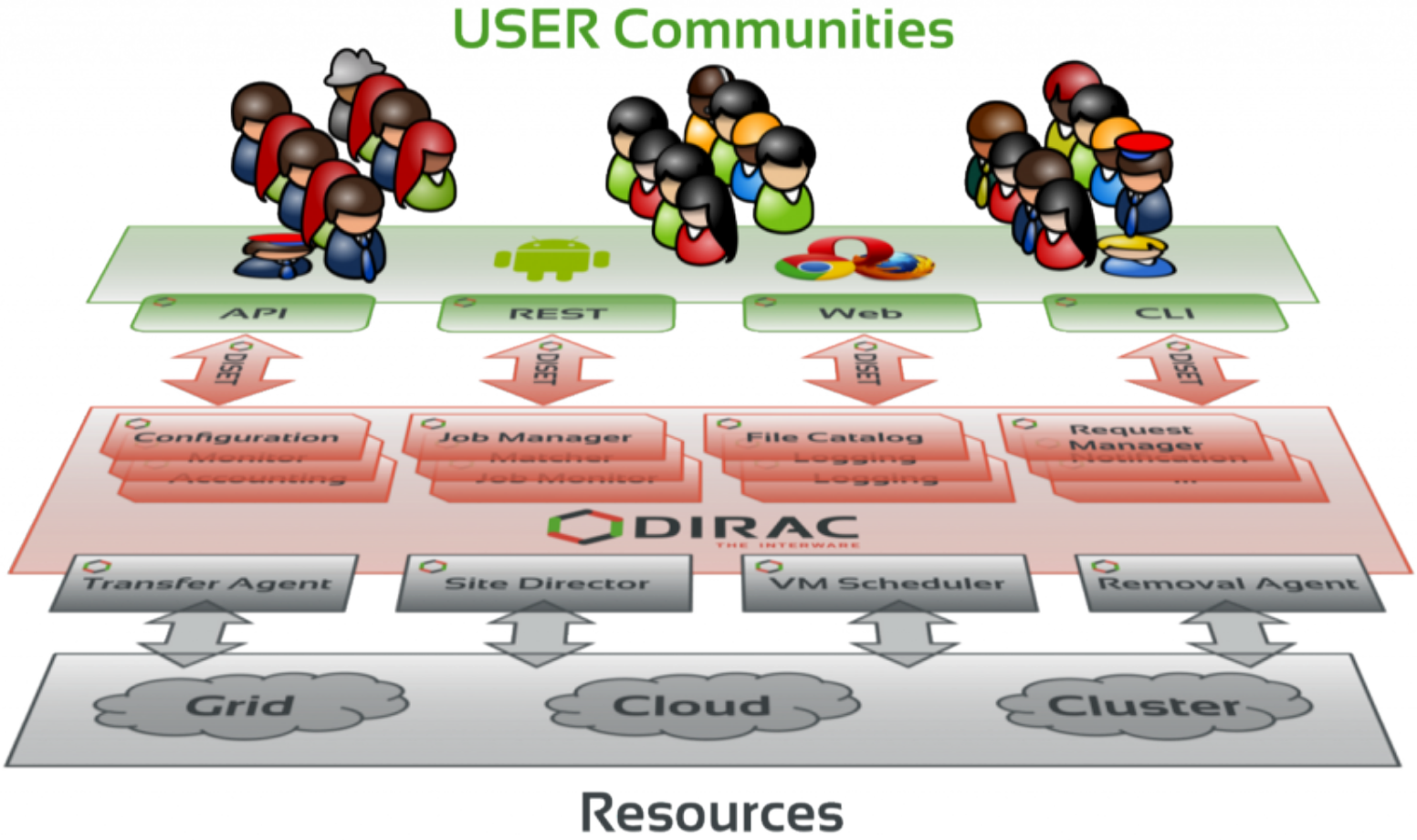


Congratulations

- You successfully submitted a DIRAC job for TOP CRT Data Analysis
  - Fetched the output file
  - Fetched histograms of interest

  - Salient features of user job submission to DIRAC :
    - Users did not have to worry about basf2 software, conditions DB, or setting up of any environment
    - Users did not have to worry about input data file

DIRAC in a picture



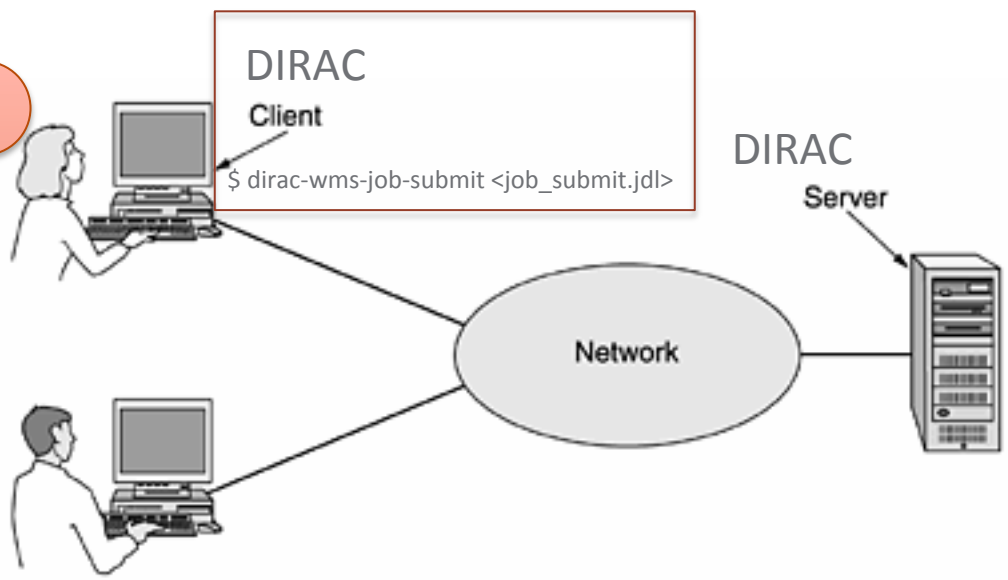
- Another example where exact file path at the site need not be specified.

DIRAC JDL 2

- DIRAC client submits a job, defined in DIRAC JDL, to the server.

Job_submit.jdl

```
JobName = ”CRTDataAnaly2Job";

Executable = "submitITopAnalysis2.sh";

StdOutput = "StdOut";

StdError = "StdErr";

InputSandbox = {"submitITopAnalysis2.sh”, "LFN:/belle/group/detector/itop/top01/data_sroot/run003149_slot07_laser_cpr3007.sroot"};

OutputData = {"StdOut","StdErr","hits.root"};

OutputSandbox = {"StdOut","StdErr","hits.root"};

Site = "DIRAC.PNNL-CRT.us";

OutputSE = "PNNL-CRT-DIP";
```



TOP CRT data analysis example script 2

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

```
#!/bin/bash

######
#set up BASF2 environment for this job
echo "Setting up the topcaf env."
. /cvmfs/belle.cern.ch/sl6/tools/setup_belle2.sh
pushd /home/stru821/Belle2/itopAnalysis
setuprel
cd -
#######

filename=$(basename *.sroot)
inputRun=${filename}

echo "Running basf2 to prepare pedestal file"
basf2 /home/stru821/Belle2/itopAnalysis/topcaf/example/topcaf_itop_sroot_ped.py --arg --inputRun=${inputRun} --arg --output=
$PWD/$(basename ${inputRun/.sroot/ped.root}) -l ERROR -n 500

echo "Now Running basf2 to output ROOT files with Belle2 objects."
basf2 /home/stru821/Belle2/itopAnalysis/topcaf/example/topcaf_itop_sroot_hits.py --arg --inputRun=${inputRun} --arg --ped=$PWD/$
(basename ${inputRun/.sroot/ped.root}) --arg --output=$PWD/$(basename ${inputRun/.sroot/_x.root}) -n 500

echo "Now analyze hits file and make plots"
analyzeHits_AsicByAsic $PWD/$(basename ${inputRun/.sroot/_x.root})
#### hits.root will be produced.
```

Setup basf2 env for running TOP CAF example

TOP CRT Data LFN was described in JDL. DIRAC automatically copies it to working directory.

Run TWO basf2 scripts on the input file (Daisy chain)

Finally make histograms on intermediate output files

Submit Analysis Job 2 on CRT Data

- Go to the working directory
  - $ cd $HOME/CRT_PNNL_DIRAC_CLIENT/CRTDataAnalysis
- Prepare/get analysis script
  - $ wget http://belleweb.pnnl.gov/forDIRAC/submitITopAnalysis2.sh
- Prepare/get DIRAC JDL
  - $ wget http://belleweb.pnnl.gov/forDIRAC/job_submit_02.jdl
- Submit this JDL to DIRAC as it was tried the first time.
  - $ dirac-wms-job-submit job_submit_02.jdl

BACK UP

Setting up of GRID Certificate

https://belle2.cc.kek.jp/~twiki/bin/view/Computing/GettingStarted

Converting P12 to PEM

1. Extract the certificate:
$ **openssl pkcs12 -in myCert.p12 -clcerts -nokeys -out $HOME/.globus/usercert.pem**

2. Extract the encrypted private key:
$ **openssl pkcs12 -in myCert.p12 -nocerts -out $HOME/.globus/userkey.pem**

3. Change permission to read-only for the private key
$ **chmod go-rw $HOME/.globus/userkey.pem**