

Using CRAB and Running Diagnostic Tests on the Cluster

Undergraduate Report for Research Credit

Instructor

Dr. Hohlmann

Student

Xenia Fave

Assisted by

Patrick Ford

Fall 2009

Using CRAB and Running Diagnostic Tests on the Cluster

Undergraduate Report for Research Credit

Introduction

In this report we will describe our experiences with CRAB and the different diagnostic tests we performed on the Florida Tech Cluster. We hope that this documentation will serve as a guide to anyone unfamiliar with either of these areas and help with the troubleshooting of errors.

CRAB

CMS Remote Analysis Builder (CRAB) is a tool designed by CMS to help with the running of jobs at sites either on or off location. It does not require the user to discover a compatible site for his configuration files and as such facilitates their completion. Additionally it assists in the publishing of data for widespread use.

We originally intended to use CRAB at the CERN UNIX account, lxplus.cern.ch, but after we ran into some errors, we opted to instead use it at our own Cluster here at Florida Tech. We have included the details for setting up and using CRAB at both locations, as well as the errors and solutions we encountered.

Crab at Lxplus

In order to submit jobs through lxplus, you have to have a CMS computing account. In order to get a CMS computing account you will need to first get private certificates, then a CMS account, and register with the CMS VO. Then you can log in to lxplus and sign up with hypernews. Once this is done, you can set up your CRAB environment and begin submitting jobs. The following sections will detail each of these steps along with the results and the obstacles we encountered.

Private Certificates

We chose to get the private certificates through DOEGrids.

Instructions are located at:

<http://www.doegrids.org/pages/cert-request.html#Globus>

Your request will be processed when an authorized agent verifies and validates the information in your request. In my case it became a little more difficult and Patrick e-mailed our DOE official several times before I was successfully issued my certificates.

Once you have imported your certificates you will need to export the keypair for use by globus grid-proxy init. Instructions for this are located at the same site as above.

CMS Accounts

The next step to using Lxplus is register with CMS and then obtain your CMS Computing Account.

To register with CMS simply fill out and submit the information requested at <http://cms.cern.ch/iCMS/jsp/sect/reg/regCMS.jsp>

You will then receive an email from the CMS secretariat with further instructions on registering. They will ask for some more information and then they ask your Institute Representative for approval. Dr. Baarmand is the current representative and he will e-mail you needing a few other pieces of information. Once this information has been given, and everything found satisfactory he will approve you at which point you will be registered with CMS.

You may then apply for the CMS Computing Account.

Instructions for this are located at <https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookGetAccount>

Once again it is merely a matter of emailing them the appropriate information and waiting for a positive response.

Registering with the CMS VO

You can register with the CERN VO as soon as you have your private certificates from the DOE and are registered with CMS. You do not have to wait until you have the CMS Computing Account.

Instructions are located at [https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideLcgAccess#How to register in the CMS VO](https://twiki.cern.ch/twiki/bin/view/CMS/SWGuideLcgAccess#How_to_register_in_the_CMS_VO)

After clicking on this link you will first be asked to log in to CERN. You can do so with either your grid certificates or your new CMS credentials. Then retype the above website into your address bar.

Hypernews

You may also want to register with HyperNews once you have finished the above steps. This will give you access to several forums, email lists, and contacts in case you need help with anything or want to learn more.

In order to register you must first

```
ssh lxplus.cern.ch
```

Use your new cern information to log in. Then

```
ssh hypernews.cern.ch
```

You will be automatically recognized and asked to choose a password for your HyperNews account. It may take a couple hours before the sitedatabase will recognize your HyperNews account.

More information about HyperNews is located at <https://twiki.cern.ch/twiki/bin/view/Atlas/HyperNews>

HyperNews forums are located at <https://espace.cern.ch/atlas-forums/default.aspx>

Setting up CRAB at Lxplus

Once you have access to Lxplus, the next step is to set up CRAB in your directory.

You first need to import your grid certificates. Instructions are located at http://jcatmore.home.cern.ch/jcatmore/getting_grid_cert.html

Then you will need to set up your environment. Instructions are located at <https://twiki.cern.ch/twiki/bin/view/CMS/WorkBookRunningGrid>

Using CRAB at Lxplus

Every time you log into Lxplus you will have to rerun the following three commands to setup your CRAB environment:

```
source /afs/cern.ch/cms/LCG/LCG-2/UI/cms_ui_env.csh
cmsenv (this must be done from the CMSSW_#_#_#/src
directory)
source/afs/cern.ch/cms/ccs/wm/scripts/Crab/crab.csh
```

I then used the CRAB Tutorial located at <https://twiki.cern.ch/twiki/bin/viewauth/CMS/WorkBookRunningGrid>

This provided the configuration files necessary to run jobs. Then I used the following standard CRAB commands:

```
crab -create -cfg [name of config file]
```

This command creates the jobs as directed in your configuration file. Note, if your configuration file is the default of crab.cfg you do not need to specify it in the command.

```
crab -submit -c [Working Directory]
```

This command submits the created jobs to any open site unless you specified where the jobs should run in your configuration file. Note, if you do not specify a unique name for your working directory, the default will name it after the timestamp and you do not have to specify the directory. This is true for the following two commands as well.

```
crab -status -c [Working Directory]
```

This command will display the status of your jobs once created.

```
crab -getoutput -c [Working Directory]
```

This command will retrieve the output from your jobs once they have finished running. The output will be sent to the /res directory within your working directory.

```
crab -publish -c [Working Directory]
```

This command will publish the retrieved output to the location specified in the config file. Please note that the config file must give permission to publish the output.

```
crab -help
```

This command will display the different commands available for crab and other helpful information.

Results and Obstacles

I first ran this tutorial on September 3, 2009. Ten jobs were submitted. 8 jobs successfully completed but two failed. The second time I ran this tutorial, on the same day, ten jobs were submitted. 5 were successful and 5 failed.

On September ninth, I once again performed the same steps without changing any part of the configuration or python file. I was able to create the jobs but could not submit them to the network. When I typed crab -submit, I received the following,

```
“error creating remote directory”
```

I edited the config file to be identical to the one given in the tutorial but received the same error.

I tried creating and submitting entirely new jobs from a different config file that I received from Samir and received the same error.

At this point, it became more sensible to use CRAB at the FLTECH Cluster then to spend more time fixing this error.

CRAB at the Florida Tech Cluster

In order to use CRAB at our home cluster, I had to once again set up my environment, but then could immediately try running jobs. The commands for CRAB are the same as in the previous section and so will not be enumerated. I once again encountered obstacles although several of these were overcome.

Setting Up Crab at FLTECH

The instructions for setting up CRAB outside of CERN are located at

<https://twiki.cern.ch/twiki/bin/viewauth/CMS/SWGuideCrabHowTo>

Steps unique for our cluster are as follows

Within your home directory, type

```
export SCRAM_ARCH=slc4_ia32_gcc345
source /mnt/nas0/OSG/APP/cmssoft/cms/cmsset_default.sh
scramv1 project CMSSW CMSSW_#_#_#
```

change to your CMSSW_#_#_#/src directory,

```
cmsenv
```

Back within your home directory.

```
source /mnt/nas0/OSG/APP/crab/CRAB_2_6_1/crab.sh
```

Then, I recommend creating a folder specifically for your CRAB working directories and configuration files. It is also beneficial to add the following commands to your .bashrc file within your home directory. These commands must be run after every log-in in order to use CRAB.

```
source /mnt/nas0/OSG/APP/cmssoft/cms/cmsset_default.sh
cd CMSSW_3_1_2/src
cmsenv
cd ../..
source /mnt/nas0/OSG/APP/crab/CRAB_2_6_1/crab.sh
```

Once you have completed these steps you can begin creating and running jobs.

Using CRAB at FLTECH

Problem 1: When I first tried running the jobs on the cluster, I once again could create the jobs but could not submit them albeit for a different reason than the one I had gotten on lxplus. I received the response,

```
“importerror: libssl.so.4: file or directory does not exist”
```

This was fixed by creating a symbolic link between libssl.so.6 in the /usr/lib directory. In order to find which version of the library you have and where it is, I used the find command. The same solution worked when it next came back with

```
“importerror: libcrypto.so.4: file or directory does not exist”
```

Problem 2: After fixing these two errors, I once again tried to submit my jobs and got

```
“ASAP ERROR: unable to ship valid proxy to the server”
```

To fix this I went to the log file and did

```
asap-user-register --server vocms58.cern.ch --myproxy
```

Problem 3: Latest problem appeared when I again tried to submit

```
“-asap-user-register: error while loading shared libraries:  
libgridsite.so.1.1: cannot open shared object file: No such file or  
directory”
```

Fixed by creating a symbolic link between libgridsite.so.1.1

Problem 4: Then it again failed to submit the jobs. No error was returned but within the crab.log file it said,

```
“executing: asap-user-register --server vocms58.cern.ch –  
myproxy  
SOAP 1.1 fault: SOAP-ENV: Server [no subcode] "getProxyReq"  
Detail: Not a member of the CMS VO”
```

Fixed with the following command that manually checks your registration to the CMS VO with

```
voms-proxy-init -voms cms
```

Problem 5: The next error displayed a long traceback of responses and within the crab.log file said

```
“ Calling ServerConfig cern”
```

No Solution was found for this problem and we decided to try creating and submitting jobs based off one of Samir’s configuration files instead.

Problem 6: Samir sent me an email with a .cfg and .py file and once these were copied onto the cluster, I tried to create the jobs. I received the following

```
"Your config file is not valid python: No Module named  
HLTrigger/Configuration/HLT_1E31_cff”
```

This merely required updating my CMSSW version. The steps for which are as follows

In your home directory type the following, which will display a list of CMSSW versions already installed on the cluster

```
scramv1 list CMSSW
```

Then to update type, (changing # to whichever version you need)

```
scramv1 p CMSSW CMSSW_#_#_#
```

Once it is finished building change to your new CMSSW directory and go to the /src directory and type

```
cmsenv
```

This time everything worked. The jobs were created, submitted, and ran successfully. I was even able to retrieve the output.

Publishing Data

Now that CRAB seemed to be working successfully on our cluster we wanted to begin publishing some of Samir's Data. I used the same configuration file Samir had originally emailed me with a few changes.

In the configuration file I changed the following variables

```
return_data = 0
copy_data=1
storage_element = T3_US_FIT
publish_data =1
publish_data_name=yourname_data
dbs_Url_for_publication=
https://cmsdbsprod.cern.ch:8443/cms_dbs_prod_local_09_writer/s
ervlet/DBSServlet
```

This time after retrieving the output I used the CRAB publishing command as well.

The first time all the jobs failed while running, so obviously could not be published.

I checked that the CMSSW version fit the requirements and resubmitted. Once again the jobs aborted.

It was at this point in the semester that the Florida Tech Cluster began to have a problem because a RAID card needed to be replaced. It was

running incredibly slow, and every time I submitted my jobs they were aborted. Several fixes were tried but to no avail. After the cluster was fixed I tried to run the original configuration file without the latest changes. Everything worked fine. I then put the adjustments back into the configuration file and all of the jobs completed successfully. I was able to retrieve the output but the publish command did not work.

I received the following error when I tried to publish

```
error: DBSAPI.dbsapiException.DbsConnectionError: failed to
connect in 03 attempts
call to DBS server
(https://cmsdbsprod.cern.ch:8443/cms_dbs_prod_local_09_writer/
servlet/DBSServlet) failed
Http ERROR status '403'
Status detail :'access to the specified resource (DN provided not
found in grid-mapfile.) has been forbidden
```

I returned to lxplus in order to determine whether it was an issue with the cluster or the website we were trying to use to publish. I had to update my CMSSW version but then received a new error

```
problems trying remote dir check... please check stage out
configuration paramters
```

Samir received an email from a professor who was receiving the same error. The professor realized that new URL was needed and emailed that out. I changed my configuration file to match the new website

```
https://cmsdbsprod.cern.ch:8443/cms_dbs_ph_analysis_02_writer
/servlet/DBSServlet
```

I was still unable to publish the jobs. They were completing with the Wrapper Exit Code=60303 which the CRAB website says means that the files already exist on the SE. I logged into the SE and removed all the CRAB output I had there, but I still received the same error when I tried to publish the jobs. This is the problem I am currently working on solving in relation to CRAB.

Diagnositics

We were receiving a lot of complaints about the speed of the Florida Tech Cluster. As a result we decided to determine what the most efficient block size for reading and writing would be. In order to test the abilities of the cluster, I used the two programs, dd and iperf. The results were collected in tables that have been attached and comprise the final section of this report.

dd

dd stands for dataset definition. We used it to measure the throughput between the NAS and the front end and then the NAS and the NAS. We were able to specify what size block we wanted used and then kept track of the speed at which information was written and copied

Using dd

The following two commands are the only ones you need for dd. Simply change the output file (of) to reflect the destination you would like to test and vary the block size (bs) to see which works best for you. We did five trials at 2,4,8,16,32, and 64 kb.

```
time dd if=/dev/zero of=/mnt/nas0/testfile bs=16k count=16384
time dd if=/mnt/nas0/testfile of=/dev/null bs=16k
```

We then logged into the nas and performed all the trials again to measure the throughput there as well.

Iperf

Iperf sets up a server and a client and then measures the bandwidth between each end. We originally designated the Frontend as the server with the Nas as the client. We checked both the UDP and TCP bandwidth. Then we switched the Frontend to be the client and the Nas to be the server and again measured both the UDP and the TCP.

Using iperf

To begin using iperf, log in to the host that you want to be the server and type the following to set up the server to start listening for the client.

```
iperf -s -i5
```

Note the default is TCP. To use UDP simply add a -u to the command. The -i5 determines the interval in seconds between periodic bandwidth reports.

Then log in to the client and type the following, once again just add a -u to use UDP.

```
iperf -c HOSTNAMEof SERVER -w 1M
```

The -w sets the window size, which in this case was 1 Megabyte.

Then on the client it will print out one line with the bandwidth, and on the server it will print out three bandwidths. If you are testing it with UDP, you can expect the results to be slower and you should also take note of the jitter and loss.

Conclusion

During the course of this semester, we worked extensively with CRAB in an attempt to publish some of Samir's data. We have run into several errors along the way and are still working on the latest one. We also tested our cluster with both dd and iperf in an attempt to speed it up. After our first set of diagnostic tests were completed we changed the block sizes for reading and writing to 64k. Then we performed all the tests again for comparison and noticed a significant improvement.

APPENDICES: Results from Tests

TABLE 1: Using dd before fixing the cluster

On Frontend					On NAS				
Block size	WRITE(s)	MB/s	READ	GB/s	Block size	WRITE(s)	MB/s	READ(s)	GB/s
64k	102.9	10.4	0.45	2.4	64k	1.11	961	0.387	2.8
	94.5	11.4	0.49	2.2		2.2	486	0.386	2.8
	288.5	3.7	0.49	2.2		1.14	939	0.673	1.6
	244.89	4.4	0.49	2.2		1.13	942	0.937	1.1
	135	7.9	0.49	2.2		1.15	933	0.391	2.7
32K	82	6.5	0.24	2.2	32K	0.599	895	0.225	2.4
	58	9.2	0.25	2.1		0.639	839	0.206	2.6
	37.5	14.3	0.25	2.1		17.3	30.9	0.227	2.4
	59.2	10.1	0.25	2.1		17.2	31.1	0.5	2.1
	41.9	12.8	25	2.1		0.602	891	0.227	2.4
16k	68	3.9	0.13	2	16k	13.7	19.5	0.12	2.2
	28.8	9.3	0.13	2		0.324	826	0.12	2.2
	32.7	8.2	0.13	2		0.323	831	0.12	2.2
	75	3.6	0.13	2		7.8	36.9	0.12	2.2
	28.4	9.4	0.13	2		15	17.8	0.16	1.6
8k	10.6	12.6	0.07	1.9	8k	10.55	12.7	0.06	2.2
	13.8	9.7	0.07	1.9		3.99	33.6	0.06	2
	12.5	10.7	0.07	1.9		4.08	32.9	0.06	2
	11.1	12	0.06	2		3.9	34.4	0.06	2
	17.3	7.8	0.07	1.9		4.3	31.2	0.09	1.4
4k	11.6	6.1	0.04	1.6	4k	0.1	666	0.04	1.6
	6.5	10.2	0.04	1.6		0.1	639	0.04	1.6
	7.5	9.2	0.04	1.6		2	33.5	0.04	1.6
	8.5	7.9	0.04	1.6		2.59	25.9	0.04	1.6
	9.6	7	0.04	1.6		2.26	29.6	0.04	1.6
2k	3.8	8.8	0.02	1.1	2k	1.3	24.5	0.02	1.2
	3.7	9	0.03	1.1		1.24	27	0.02	1.1
	5.5	6.1	0.03	1.1		2.13	15.8	0.03	1.1
	5.1	6.5	0.02	1.1		1.7	18.7	0.03	1.1
	2.3	14.4	0.03	1.1		1.5	21.9	0.029	1.1

Using CRAB and Running Diagnostic Tests on the Cluster | 12/1/2009

TABLE 2: Using dd after fixing the cluster

On Frontend					On Nas				
Block size	WRITE(s)	MB/s	READ(s)	GB/s	Block size	WRITE(s)	MB/s	READ(s)	GB/s
64k	12.7	84.4	0.42	2.5	64k	1.1	933	0.38	2.8
	13.1	81.7	0.45	2.3		1.1	947	0.38	2.8
	13.1	81.5	0.48	2.2		1.1	927	0.39	2.8
	14	76.5	0.53	2		1.1	933	0.38	2.8
	12.6	84.8	0.46	2.3		1.1	934	0.38	2.8
32K	6.5	81.7	0.22	2.4	32K	1.7	302	0.2	2.6
	6.4	83.1	0.22	2.4		1.8	289	0.2	2.5
	6.4	82.8	0.22	2.4		1.7	301	0.2	2.6
	6.4	83.1	0.22	2.4		1.7	312	0.2	2.6
	6.5	82.2	0.22	2.3		1.7	303	0.2	2.6
16k	3.18	84.3	0.12	2.2	16k	0.79	336	0.1	2.6
	3.2	83.8	0.12	2.2		0.76	350	0.11	2.4
	3.17	84.5	0.12	2.1		0.83	320	0.1	2.4
	3.14	85.2	0.12	2.1		0.8	334	0.12	2.2
	3.18	84.3	0.12	2.1		0.76	352	0.1	2.4
8k	1.6	82.6	0.06	2	8k	0.36	366	0.06	2.2
	1.6	80.3	0.06	2		0.39	340	0.06	2.2
	1.5	85.5	0.06	2		0.36	368	0.06	2.2
	1.5	85.8	0.07	1.9		0.36	370	0.06	2.2
	1.5	85.3	0.06	2		0.38	345	0.06	2.1
4k	0.82	81.4	0.04	1.6	4k	0.21	309	0.03	1.7
	0.79	84.7	0.04	1.6		0.19	344	0.03	1.7
	0.79	84.6	0.04	1.6		0.18	354	0.03	1.7
	0.78	85.5	0.04	1.6		0.21	319	0.03	1.7
	0.8	83.5	0.04	1.6		0.2	335	0.03	1.7
2k	0.38	87	0.02	1.2	2k	0.1	309	0.02	1.2
	0.39	84.2	0.03	1.1		0.12	273	0.02	1.1
	0.39	86	0.02	1.2		0.1	313	0.02	1.2
	0.39	84.8	0.02	1.2		0.1	312	2	1.2
	0.39	84.1	0.02	1.1		0.09	336	0.02	1.2

TABLE 3: Using Iperf before fixing the cluster

Server	TCP(Mbits/s) :Server	TCP(Mbits/s): Client	UDP: jitter (ms)	lost	(Mbits/sec)
frontend	753	754	0.11	0	1.05
	912	913	0.022	0	1.05
	896	897	0.034	0	1.05
	891	892	0.393	0	1.05
	888	889	1.751	0	1.05
Server	TCP(Mbits/s) Server	TCP (Mbits/s): Client	UDP: jitter(ms)	lost	(Mbits/s)
nas-0-0	941	942	0.023	0	1.05
	941	942	0.022	0	1.05
	936	933	0.022	0	1.05
	941	942	0.023	0	1.05
	941	942	0.022	0	1.05

TABLE 4: Using Iperf after fixing the cluster

Server	TCP(Mbits/s): Server	TCP(Mbits/s): Client	UDP: jitter(ms)	lost	(Mbits/s)
frontend	941	942	0.048	0	1.05
	939	940	0.025	0	1.05
	935	937	0.022	0	1.05
	930	931	0.023	0	1.05
	941	942	0.025	0	1.05
Server	TCP(Mbits/s): Server	TCP(Mbits/s): Client	UDP: jitter(ms)	lost	(Mbits/s)
nas-0-0	936	937	0.081	0	1.05
	941	942	0.022	0	1.05
	941	942	0.022	0	1.05
	941	942	0.022	0	1.05
	941	942	0.021	0	1.05