ArCond: A short user's Guide v1.0

S.Chekanov

(ANL)

August 20, 2008

0.1 Introduction to ArCond

ArCond (ARgonne CONDor) is a package to facilitate running multiple jobs in parallel using the Condor package [1]. The package was designed for a Tier-3 type of clusters consisting of a number of independent computer notes (PCs) with a local file storage on each node, and a common NFS-mounted user (or share) directories. A relatively modest commodity-type networking is assumed. It is also assumed that the Condor cluster software is already running on these PC noted in a slave mode.

The main advantage of the ArCond front-end of Condor is in possibility to run jobs in parallel using data samples stored locally on each computing node, rather than using a single file storage on the NFS server. This leads to significant performance improvement and the program can be used to work on a Linux cluster build from commodity PCs with several TB file storage.

In addition, any custom command sequence defined in a submission script can be executed in parallel. For example, one can run multiple athena option files, perform user-specific tests, file merging etc. directly on the condor linux boxes.

0.2 Expected PC farm configuration

ArCond can be used for the following environment:

- A PC farm should consist of N_{pc} computers with several TB file system. Each node should have N_{core}. The most optimal configuration for Tier-3 analysis facility is $N_{pc} > 10$ and $N_{core} \geq 8$.
- All PC's have running Condor in a slave mode. The home directory for Condor is /home/condor, with the size of 100-200 GB, while additional disk space should keep data files. The Condor master can be any PC without large file storage.
- a limited (1-2 TB) NFS-based file storage assumed to keep some shared libraries (like ATLAS releases) or user home directories. ATLAS or any other necessary software is mounted on NFS as well and accessible for each PC node.

0.3 How to run a bash script using an ANL PC farm

ArCond allows to execute a custom bash script to be executed on each Linux box. This can be used to monitor PC, to upload data on each box etc. To run a bush script, say "example.sh", just execute a statement "arc_exe -i example.sh.. You should see the output from jobs executed on each Linux box. For more verbose output, use the option "-iv". Use the option "-h" or help".

0.4 How to run an Athena job using a PC farm

Edit the file "arcond.conf". Specify atlas release, how many events to process, directory where data are located on each PC node. This directory should be on a local disk. Specify the root directory for a user program. Example: package_dir = /users/chakanau/testarea/14.2.21/PhysicsAnalysis/AnalysisCommon/UserAnalysis, such that inside "UserAnalysis" you have the standard ATLAS directory tree (cmt, src, share, etc). You do not need to compile the package; it will be compiled locally on each node.

Now, prepare configuration files in the "patterns" directory. For each PC node, there should be one file "schema.site.[PCNAME].cmd", where the [PCNAME] is a unique name for a Linux box. In each file, the "requirements" variable should be set to the hostname of the PC.

Now you are ready to submit Athena package to the node. Setup OSG software and then use the command "arcond". The following actions will be performed:

- the "patterns" directory will be scanned in order to determine which PC will be used;
- All local directories (specified in the "arcond.conf" on each PC will be scanned. Then the program will build a file list for Athena input ("database"). The result of this step will be found in the directory "DataCollector", which should be filled with configuration files for each PC box. Configuration files contain information about which data files are available on which Linux box, and how many processor cores are available;
- Data files specified in the configuration files in the "DataCollector" will be rescanned to make sure that all data files are unique. In case if there are more than one specified data file, it will be removed from the configuration file;
- The directory with the package ("UserAnalysis") will be tarred into a file "UserAnalysis.tgz" and will be put to the "Job" directory. If you send second time the arcond command, the program will ask you about should this tar file be recreated or not;
- the program will send Condor jobs on each PC core. Exactly $N_{pc} \times N_{core}$ jobs will be sent. It should be noted that if the program will not detect data on the specified input, the number of jobs will be less. The program will send jobs only to Linux boxes were the data are located;
- Condor will compile and will run jobs locally on each linix box, without any load on NFS server;
- Condor will copy the output files when jobs will be finished to the directories "Job/runN".

0.5 Ready-to-use example

This is one example of how to run a user athena program using data scattered among several Linux boxes:

- Check your inputs in "arcond.conf". For this example, you do not need to modify anything. Type "arcond". If OSG GRID is not set, set it and get the proxy. It will determine how many PC's are available (from the "pattens" directory) and will ask to build a database which will contain the configuration files with input data on each PC box. Always build this database if the configuration file "arcond.conf" is changed.
- If you answer "y", wait for about 1 min. The result of this step is a number of configuration files in "DataCollector" You may get some warning id some PC specified in the "patterns" directory is not accessible for the Condor. In this case fix this PC or move the data somewhere else.
- The next question is "Submit all collections to the condor?". Say "y". This will submit all jobs to the condor. The submission scripts will be located in the directory "Job".
- Wait when all jobs will be done. Use "condor_q" command. Also, you can use "arc_check"
- If all jobs are done, combine the outputs ("Analysis.root for this example) in one file. Use "arc_add" command.
- Clear the project as "arc_clean".

0.6 More details about job option files

The base for the option Athena file is located in the "user/" directory. The name of this file "Analysis_jobOptions_BASIC.py". You should adjust it to meet your needs. The output is "Analysis.root" (do not change this name).

The script which runs this option file is build using the "user/ShellScript_BASIC.sh" file. This is the main execution file to be run by the Condor. Again, one can correct it to fit your needs.

Both files are very important, since all submission directories in "Job/run*" contain scripts build using "Analysis_jobOptions_BASIC.py" and "ShellScript_BASIC.sh".

If you want to use a private option file, there are should be 2 modifications in the original athena file:

```
include("InputCollection.py")
ServiceMgr.EventSelector.InputCollections = dataCollection
```

and the output file should always has the name "Analysis.root", otherwise the mergin part will not work. This part of the program should look like:

```
# save ROOT histograms and Tuple
from GaudiSvc.GaudiSvcConf import THistSvc
ServiceMgr += THistSvc()
ServiceMgr.THistSvc.Output = [ "AANT DATAFILE='Analysis.root' OPT='RECREATE'" ]
from AnalysisTools.AnalysisToolsConf import AANTupleStream
topSequence += AANTupleStream()
AANTupleStream = AANTupleStream()
AANTupleStream.ExtraRefNames = [ "StreamESD","Stream1" ]
AANTupleStream.OutputName = 'Analysis.root'
AANTupleStream.WriteInputDataHeader = True
AANTupleStream.OutputLevel = WARNING
```

Always remove the statement starting with "theApp.EvtMax" which specifies the maximum number of events from your option file, since this part comes from the ArCond include file.

0.7 Resubmission

If some programs will fail, one can resubmit them. The script "arc_check" or "arc_add" will tell which jobs did not finish properly.

Resubmit them using the scripts located in "Job/runNUMBER" (runNUMBER will be given in the error message of the script "arc_check").

Bibliography

[1] THE CONDOR PROJECT.
URL http://www.cs.wisc.edu/condor/ 1