ANL Tier3 infrastructure and PC farm prototype

Sergei Chekanov, Rik Yoshida

(*Analysis support center, ANL)*

http://atlaswww.hep.anl.gov/asc/

US ATLAS Distributed Facilities Meeting
March 3, 2009

Tier3 tasks

- Submission of jobs to the grid
- Retrieval of results (Ntuples, skimmed AODs, DPD's) from the grid
- Running over AOD, DPDs, Ntuples (batch mode)
- Interactive analysis of ROOT trees (PROOF) +histograms
- Generate Monte Carlo events (mainly truth levels for corrections)
- Generate NLO QCD predictions

NFS file server
with data

interactive worker
node with ssh login

desktop PCs

Can we increase capabilities of Tier3s for the post-grid analysis without significant investments?

Tier3 PC farm?

Advantages compare to the standard "desktop" Tier3 approach:

- **Jobs run 2 orders of magnitude faster compare to standard desktops (2-4 cores)**
  - + takes out load from Tier1-2 by enabling high performance at Tier3
- **Can deal with tens of TBs of data**
- **Better interactivity and full local control of processing of large datasets**
- **Generating large MC sample & CPU-consuming NLO predictions**

Needed characteristics:

- **Cost effective - tens of $k, preferably commodity PCs**
- **Low maintenance – max 0.5 FTE**
- **Scalability**
- **Low network load (assume commodity 1 Gb networking)**
- **Extension of the desktop rather than Tier2**

ANL PC farm fulfills all these characteristics

ANL PC farm prototype (24 cores)

NFS file server

- ssh user login
- interactive worker
- Condor master
  (no job submission)

Condor slaves

8-core
2TB

8-core
2TB

8-core
2TB

Minimum requirements for slave nodes:

- 8 cores

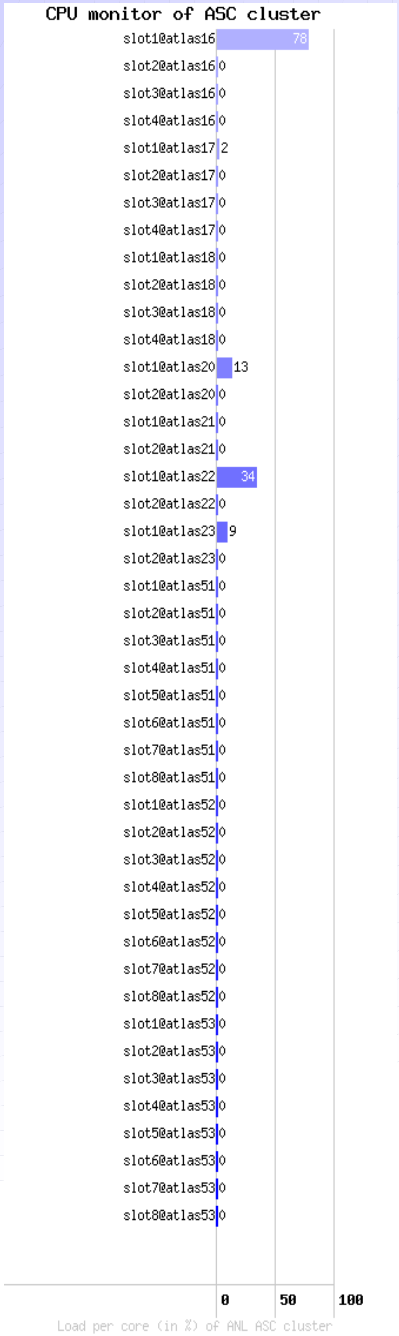
- 1 system disk
/atlas/release/14.5.1
/home/condor/ (local)
/users/ (mount point to NFS)

- 2 data 1 GB disks:
/data/dataset1/
/data/dataset2/

4 cores per 1 TB

Prototype build in Sep. 2008. Since then 200 job submissions (~5000 per core)

PC farm prototype in action



CPU slot uptime

http://atlaswww.hep.anl.gov/asc/admin/cpu-monitor/

public ssh login servers

Desktop PCs

PC farm prototype

Web monitoring:
WWW server PC runs Condor slave
(several Condor services disabled)

Hardware configuration for condor slave PC

1	CUSTOMSERSERIAL	CYBERTRONPC		CUSTOM CONFIGURED SERVER	1
2	PRC-INT-X5410RA	INTEL	BX80574E5410A	XEON E5410 C4 2.33GHZ 771 RET	2
3	MBD-SPM-X7DVLE	SUPERMICRO	X7DVL-E BULK	771 V X8 6D2 2GL R DUAL XEON	1
4	MEM-GEN-2FB667	SUPERTALENT	MEM-SAM-2GEBA	2GB FB DDR2 ECC FB PC5400/667 MHZ	4
5				TOTAL 8 GB	
6	HDR-WDG-25AAKS	WESTERN DIGITAL	WD2500AAKS	250GB S2 7200 16MB	1
7				SYSTEM DISK	
8	HDR-SGT-1TBS2B	SEAGATE	ST31000340NS	1TB SATA2 7200RPM 32MB RAID EDITION	2
9				DATA DISKS	
10	CDR-LIT-16XDVDB	LITEON	DH-16D2P	16X DVD IDE BLACK	1
11	FLD-ALPS-144MBB	ALPS	DF35	1.44 MB FLOPPY DISK DRIVE BLACK	1
12	AD-VID-NOUPGRD			ONBOARD VIDEO	1
13	AD-NET6			DUAL 10/100/1000 GIGABIT NETWORK	1
14	SUP-CHN-BRKTNC	CHENBRO	84H312410-022	NACONA BRACKET SET OF 2	1
15	CAS-CHN-SR105	CHENBRO	SR105-BK(10569-BLACK)	SERVER TOWER BLACK	1
16	POW-SPK-460W	SPARKLE	FSP460-60PFN	460 WATT INTEL XEON CFT24 PIN	1
17	WARR-EXTENDED1			1 YEAR WARRANTY ON LABOR & PARTS	1
18				LIFETIME U.S. BASED TECHNICAL SUPPORT	
19	SHIPPFREE			FREE GROUND SHIPPING	1

Summary:

8 Xeon 2.33 GHz cores
8 GB RAM,
2 TB disks+ 1 system disk

CybetronPC quote: $2000 per box (Jan 2009 update)

Time to bring to a full operational mode ~ ½ day :

- SL4.6 installation
- starting necessary services (NIS, Condor, etc)
- configure condor home directory + iptables

Example performance for AOD

- **mc08.106070.PythiaZeeJet_Ptcut.recon.AOD.e352_s462_r541**
  - Release 14.2.21
  - 200k events. 800 AOD files. 266/per box, 33GB
  - Lumi=230 pb-1
  - Data equally distributed among 3 PC slaves
- **Program accessing:**
  - Jets,Photons,Muons,Electrons
  - Same for the truth level
  - 100 histograms + fill a ntuple with all objects
- **Processing time: 30 min + 4 min (compilation) on 24 cores** (110 ev/sec, 5 ev. sec).

10 fb-1 data: ~1 day of running on 24 cores, 6h on 80 cores
Data storage: 1.4 TB for data
x4 MC = 6-7 TB for MC and data

If ATLAS release and data located on NFS (ReadyNAS), a low performance due to I/O bottleneck is observed:

- **about 10 min to setup ATLAS release (24 cores hit NFS at the same time)**
- **factor ~2-3 slower during reading AOD events stored on NFS**
- **poor performance of desktops with NFS-based user home directories**

PC farm prototype performance for AOD

- **Assuming 10 fb-1, ~80 cores + 20 TB should be enough for storing and processing skimmed AODs**
-
  - **Inclusive direct photon analysis (PT(gamma)>50 GeV, signal ~ background)**
  - **Inclusive jets (PT>400-500 GeV)**
  - **Dijets (PT>200 GeV)**
  - **Z+jet, PT(jet)>40 GeV**
  - **.. all other processes with lower x-section (H->gg, etc..)**

  In all cases it is assumed that analysis data set consists of:

  - **Data and MC are in form of AODs or DPDs**
    - **for worst- case scenario when DPDs size = AOD size**
  - **Monte Carlo samples have 4 times larger statistics than (signal) data**

Estimates for 10 fb-1:

- **did not hit the limit ~10-20 TB for a single analysis**
- **processing time < 1 day for 80 cores in all cases**

Analyzing ntuples

- **200k events from the previous example analyzed using a compiled C++**
- **Ntuple structure and size:**
  - **Storing TLorentzVectors for:**
    - Photons, Muons, Cone4Jets, 10 vectors with doubles (PID for photons)
    - Same for MC truth
  - **Ntuple size: 75MB**
- **Processing 200k events takes 10 sec on one Xeon 2.33 CPU**
  - Filling ~10 histograms with invariant masses (jet-jet, γ-jet, γ-γ)
- **Similar checks where done for Z+jet analysis**

Estimates for 10 fb-1:

- **requires 3GB file storage**
- **processing time:**
  - 7 min on one core
  - ~ 20 sec on 24 cores
    (assuming no I/O bottleneck)

Generating MC truth on PC farm prototype

- **Generate truth level for Z->e+e- events using the official MC production script: MC8.106070.PythiaZee*.py:**
  - Build Kt6TruthJets,Cone4TruthJets,Cone7TruthJets on-fly
  - Fill a ntuple with all jets + electrons
    - single-core: 100000 events for 110 min
    - 24 cores: 2.4M (Overall Lumi=2.4 fb-1)
  - Ntuple size with MC truth info: 446M

- **Second C++/ROOT program was developed to read Ntuple with MC truth**
  - Processing time using compiled C++ program:
    - **~2 min on single core**
    - **~20 sec on 24 cores (~10 sec runtime, ~10 sec Condor startup)**

Satisfactory performance for Tier~3

ArCond (Argonne Condor)

http://atlaswww.hep.anl.gov/asc/arcond/

- **Python frond-end for Condor for:**
  - job submission
  - data discovery
  - checking job status
  - merging outputs
- **Does not require installation & Atlas release**
- **No maintenance or extra service**
  - 1 cron job to build a static database with files (optional!)
- **Minimum requirement: OSG-client (for condor) and standalone ROOT**
- **Designed for analysis of data flatly distributed over multiple PCs**
  - **Example:**

    /data1/GammaJet/AOD1.root - 33% of data on atlas1.cern.ch

    /data1/GammaJet/AOD2.root - 33% of data on atlas2.cern.ch

    /data1/GammaJet/AOD3.root - 33% of data on atlas3.cern.ch

Stored data sets

- **Since Sep. 2008, we store 15422 AOD MC files**
  - ~ 4M Monte Carlo AOD events (+ few ESD sets)
  - Corresponds to **~25%** of the total capacity of the PC farm prototype

- **Data moved to each box after using dq2_get (ArCond provides such splitter).**
- **A prototype of dq2_get front end is ready to get data directly on each box by specifying fraction of AOD files necessary to copy**

/data1/mc/gamma_jet/pt17/AOD	atlas52	gamma+jet samples, r14.2, pt>17 GeV. Also available: pt40, pt8 pt600
/data1/mc/pythia_gfilter/pt17/AOD	atlas51	Filtered background sample, r14.2, pt>17 GeV. Also available: pt pt400, pt600
/data1/mc/PythiaZeegam25/AOD	atlas51-52	Z+gamma+X samples, r14.2, pt>25 GeV
/data1/mc/BaurZeegam/AOD	atlas51	Z+gamma+X, Baur MC, r14.2, pt>25 GeV, X-section=463.622 p each file
/data1/mc/mc08.105802.JF17_pythia_jet_filter.recon.AOD.e347_s462_r541/AOD	atlas51-53	~1.5 M events, inc.Pythia after JetFilter, r14.2, pt>17
/data1/mc/mc08.106070.PythiaZeeJet_Ptcut.recon.AOD.e352_s462_r541/AOD	atlas51-53	Z->e+e- + jet events, r14.2.20, 250 events in each file, 797 files, 968.637 pb, efficiency = 0.90
/data1/mc/mc08.106071.PythiaZmumuJet_Ptcut.recon.AOD.e352_s462_r541/AOD	atlas51-53	Z->mu+mu- + jet events, r14.2.20, 250 events in each file, 791 file 968.637 pb, efficiency = 0.90
/data1/mc/mc08.106072.PythiaZtautauJet_Ptcut.recon.AOD.e352_s462_r541/AOD	atlas51-53	Z->tau+tau- + jet events, r14.2.20, 250 events in each file, 759 file 968.637 pb, efficiency = 0.90
/data1 /mc/mc08.106379.PythiaPhotonJet_AsymJetFilter.recon.AOD.e347_s462_r541/AOD	atlas51-53	250k events, gamma+jet, ckin(3)>15 GeV
/data1/mc/MC08/JS0/ESD	atlas53	also JS1, JS2,JS3,JS4,JS5,JS6,JS7 available. Talk to Belen a
/data1/mc/mc08.107141.singlepart_pi0_Et40.recon.AOD.e342_s439_r546/AOD	atlas51	200 files, r14.2.20.3, single pi0
/data1/mc/mc08.107041.singlepart_gamma_Et40.recon.AOD.e342_s439_r546/AOD	atlas51	189 files, r14.2.20.3, single gamma
/data1/mc/mc08.107680.AlpgenJimmyWenuNp0_pt20.recon.AOD.e349_a68/AOD	atlas51-53	1202 files, r14.2.20, W->e+nu+0 partons
/data1/mc/mc08.107681.AlpgenJimmyWenuNp1_pt20.recon.AOD.e349_a68/AOD	atlas51	242 files, r14.2.20, W->e+nu+1 partons
/data1/mc/mc08.107682.AlpgenJimmyWenuNp2_pt20.recon.AOD.e349_a68/AOD	atlas51	624 files, r14.2.20, W->e+nu+2 partons
/data1/mc/mc08.107683.AlpgenJimmyWenuNp3_pt20.recon.AOD.e349_a68/AOD	atlas51	165 files, r14.2.20, W->e+nu+3 partons
/data1/mc/mc08.107684.AlpgenJimmyWenuNp4_pt20.recon.AOD.e349_a68/AOD	atlas51	48 files, r14.2.20, W->e+nu+4 partons
/data1/mc/mc08.107685.AlpgenJimmyWenuNp5_pt20.recon.AOD.e349_a68/AOD	atlas51	22 files, r14.2.20, W->e+nu+5 partons

FDR2 reprocessed data: ||

/data1/mc/fdr08_run2.0052280.physics_Egamma.recon.AOD.o3_f47_r575/AOD	atlas51-53	FDR2 AOD data, release 14.2.24
/data1/mc/fdr08_run2.0052280.physics_Egamma.recon.DPD_CALOJET.o3_f47_r575/AOD	atlas51-53	FDR2 DPD data, release 14.2.24
/data1/mc/fdr08_run2.0052280.physics_Egamma.recon.DPD_EGAMMA.o3_f47_r575/AOD	atlas51-53	FDR2 DPD data, release 14.2.24
/data1/mc/fdr08_run2.0052280.physics_Egamma.recon.DPD_PHOTONJET.o3_f47_r575/AOD	atlas51-53	FDR2 DPD data, release 14.2.24
/data1/mc/fdr08_run2.0052280.physics_Jet.recon.AOD.o3_f47_r575/AOD	atlas51-53	FDR2 AOD data, release 14.2.24

ArCond PC farm submission

- **Pure python & bash. Does not need installation. Requires OSG-client (Condor)**
  - **`> arcond`**
    - Reads a configuration file (with atlas release version, input directory with AOD files on all boxes, package athena name)
    - Splits jobs to be run in parallel: N=N(PC boxes) x N(cores)
    - Data discovery using local storage. Builds a database with input files and associates each AOD file with specific box
    - Splits data lists, prepare submission scripts, submits to each box with local data
      - shell submission script contains anything you like, including multiple athena runs etc.
    - Compiles programs using either NFS-based ATLAS software release or locally installed release
    - Runs jobs using local condor home directory
    - When jobs are ready, the output is copied to submission directory
      - optional, depends what do you put in shell script
    - Output root files merged automatically
    - Automatic check new arcond version

Running arcond

- **Before submitting a job, prepare a configuration file (“ arcond.conf”)**

```
atlas_release=14.5.1

# events to process in each job
events = -1

# dir with input AOD files.
input_data = /data1/mc/mc08.105802.JF17_pythia_jet_filter.recon.AOD.e347_s462_r541/AOD

# package directory on NFS
package_dir = /users/chakanau/testarea/14.2.21/analysis/PromptGamma
```

scan all subdirectories

- **Prepare the job option file**
- **Check data availability as:**
  - `arc_ls <dataset>`

Ready to submit!

Submitting job..

```
chakanau@atlas16:submit$ ./arcond
################## ARCOND v1.0 ####################
##                ANL ASC                      ##
#################################################
 Input configuration=arcond.conf
---> Input data located at =
/data1/mc/mc08.105802.JF17_pythia_jet_filter.recon.AOD.e347_s462_r541/AOD
---> Checking computing cores
    -->1 PC node=atlas51.hep.anl.gov with=8 cores found
    -->2 PC node=atlas52.hep.anl.gov with=8 cores found
    -->3 PC node=atlas53.hep.anl.gov with=8 cores found
---> Total number of found cores= 24
Start data ArCond data discovery tool?
-> To discover data on-fly, type "f"
-> To discover data using ArCond static database created every 24h, say "s"
-> Do not discover data, say "n"
---> Checking claimed CPUs
---> Total number of claimed CPU cores= 0
---> Building the database on all nodes with input AOD/DPD files
---> Checking for duplicate input data files
   --> PC node= atlas53.hep.anl.gov   has  1987  input files
   --> PC node= atlas51.hep.anl.gov   has  1964  input files
   --> PC node= atlas52.hep.anl.gov   has  1722  input files
   --> ## SUMMARY: Total number of input files = 5673
Project file:/users/chakanau/work/submit/Job/PromptGamma.tgz was found.
Do you want to rebuild it (y/n)? y
---> Package submission file = Job/PromptGamma.tgz
---> Package submission log file = Job/PromptGamma.log
---> Number of events in one job = -1
---> Atlas release = 14.5.1
---> 24  jobs will be submitted to = 3 PC boxes
Do you want to prepare the submission scripts (y/n)? y
Submit all prepared jobs to the PC farm? (y/n)
```

only for first submission! (see next slide)

it was found since I've sent this package before

To run ArCond in silent mode use: "arcond -allyes"

Data discovery

PC farm users have several choices for data discovery:

- **"s" - to discover data using a small flat-file database**
  - Updated every night
  - Implementation: Each slave note runs a cron job
    - (based on `find "/data1/ -type f > /users/condor/$date.txt"`)
    - for 10000 AOD files, run time is 3-5 sec.
  - Copied and stored on NFS
  - When a user runs "./arcond", always the latest database is used
  - Also can be used to recover data when PC box fails (do not have experience yet)
- **"f" - to discover data "on-fly"**
  - If data have been copied recently, the database may not exists
  - Then arcond sends a small script on each PC boxes and brings data list back
  - Usually takes ~20-30 sec (assuming that Condor is not busy)
- **"n" if the user selected "s" and "f" from previous runs, there is no need to discover data (previous data list will be used)**

Simple and robust. So far required no attention from admin.

Checking and getting jobs back

- Run condor commands: condor_status or condor_q
- Your jobs are in “idle” state?
  - check who is running on the farm as:
    - condor_status -submitters (OR) condor_q -global
- Check output files as: arc_check
- If arc_check tells that all output files “Analysis.root” are ready, combine output files to one file using arc_add. This creates “Analysis_all.root”

- To debug program and check errors:
  - ./Job/runXXX/Analysis.log - athena log file
  - ./Job/runN_atlasXXX/Job.ShellScript.atlasXXX/job.local.out - execution log

Open questions

- **How to bring data to the PC nodes?**
  - Copy to /tmp or an external file server and re-distribute data?
  - Copy data directly using `dq2_get` on each PC node?
    - front-end of dq2-get is ready, but not well tested yet
  - Ideally, it would be good to run `dq2-get` automatically on each box
    - cron job, condor job?
- **Did not study data recovery in case of a faulty disk (assuming no RAID):**
  - Attach an external USB and sync data? Add extra disk?
  - Copy data again from the grid using ArCond static database?
    - need to develop a script based on dq2-get for data retrial using ArCond database
- **Arcond was designed to run jobs over locally distributed data, with little attention to event generation (MC, NLO, etc).**
  - Presently ArCond uses all 24 cores only if each PC node has a “dummy” input file
    - will be improved soon (new option for ArCond submissions)

Summary

- **24-CPU PC farm prototype is fully functional**
  - ~$6k investment last year
  - Man power: 0.5 FTE, which dropped to 0.2 FTE after the setup
- **Since Sep 1, ~5000 jobs completed on each CPU node**
  - ~ 200x24-core completed jobs.
  - Most of ANL analysis were done using the PC farm prototype
- **No any failures reported:**
  - Small problem if Condor master is busy (at present runs on the worker PC)
    - 1-2 cores are not identified correctly by Condor → lower efficiency
    - a dedicated Condor master should be installed
- **With extra $14k investment, the PC farm could be extended to ~T3g**
    - **Goal:** 80 CPUs with 20 TB data storage

Appendix: "typical" T3g

- reliable
- simple
- safe
- scalable

~24 port 1Gb
manageable
$2.5k – Cisco (or)
$800 - LinkSys

weekly sync

NIS server

NFS File server
SL5.2
No login

4-cores
>4 TB RAID5

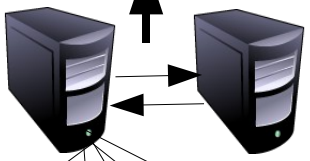
/home/users
/share/
(mounted on all boxes)
$4k each

WWW server
(optional)

daily sync

"Interactive worker" node

SL4.6
SSH login
8 cores
> Xeon 2.33 GHz
> 4 TB RAID5 work directories
16 GB RAM
~ $6k each



Master (+ master clone)
(condor, gridFTP? SRM?)

Desktop PCs
(optional)

8-cores
2 TB
$2K each

PC farm

Condor slaves

~$50k T3g building from scratch
(20 TB, 80-core farm)
~$35k for minim. config (no backup for worker & file server)