UW-ATLAS Experiences with Condor

M.Chen, A. Leung, B. Mellado
Sau Lan Wu and N. Xu

Paradyn / Condor Week, Madison, 05/01/08
Outline

• Our first success story with Condor
• CRONUS system
  - A success story of using Condor glide-in
• A Tier3 model
  - Use of multi-layer Condor system
• The integration of PROOF and Condor
  - Distributed Analysis for ATLAS
Center of mass $E$ | 14 TeV
---|---
Design Luminosity | $10^{34}$ cm$^{-2}$ s$^{-1}$
Luminosity Lifetime | 10 h
Bunch spacing | 25 ns

ATLAS at the LHC
ATLAS Physics/Computing

Online System

~PByte/sec

~100-400 MBytes/sec

Offline Farm, CERN Computer Ctr
~25 TIPS

CERN/Outside Resource Ratio ~1:2
Tier0/(Σ Tier1)/(Σ Tier2) ~1:1:1

Tier 0 +1

Tier 1

France
UK
Italy

Tier 2

Tier 3

Tier 4

Physics data cache
Workstations

~2.5+ Gbps

Institute ~0.25TIPS

100 - 10000 Mbits/sec

Physicists work on analysis “channels”
Each institute has ~10 physicists working on one or more channels
Our first success story with Condor

• Our group started running ATLAS full simulation jobs on GLOW since October 2004
• First large production was from December 2004 to May 2005. Production extended to August 2005
• With help from Condor team, 12.5 million fully simulated Monte Carlo (MC) events was produced in 9 months
  – Each event takes 10-20 minutes in 2 GHz cpu
• The Wisconsin group was the largest single contributor of Higgs MC, providing over 90% of all the events produced within ATLAS Higgs Working Group.
Number of simulated events done in Madison

Date

Events

Dec 5
Dec 17
Dec 29
13-Jan
25-Jan
8-Feb
20-Feb
7-Mar
19-Mar
31-Mar
12-Apr
25-Apr
7-May
21-May
2-Jun
14-Jun
9-Jul
27-Jul
17-Aug

0
2000000
4000000
6000000
8000000
10000000
12000000
14000000
CRONUS system
A success story of using Condor glide-in

- CRONUS is the first Condor Glide-in Based ATLAS Production Executor.
- The development started from 2006 by Sanjay Padhi
Introduction to CRONUS – ATLAS Virtual Computing Cluster

The concept of late binding is intrinsic to Condor via the ClassAds

Only one communication language among all - ClassAds
Example of the CPU usage

Glide-In Scalability

ATLAS - CRONUS using ~ 5200 CPUs in parallel [http://txb2170.cern.ch/condor_view/]

Cronus Eff. - 99.08%
A Tier3 model
Use of multi-layer Condor system

- UW has a leading role in defining the computing model for universities in the ATLAS collaboration
- Accommodate Several requirements
  - Combination of CPU intensive with less CPU intensive jobs
  - Combination of I/O intensive jobs with less I/O intensive jobs
  - Distributed data analysis jobs (PROOF)
Typical Model

GRID

Computing Pool
Computing nodes with small local disk.

Storage Pool
Centralized storage servers (NFS, Xrootd, Dcache, CASTOR)

The gatekeeper
Takes the production jobs from Grid and submits to local pool.

The users
Submit their own jobs to the local pool.

Batch System
Normally uses Condor, PBS, LSF, etc.

Dedicated PROOF pool
CPUs cores + big disks.

CPUs are idle most of the time.

Heavy I/O Load
The way we want to go...

Job priority

**The gatekeeper**
Production jobs from Grid and submits to local pool.

**Local jobs**
Users’ own jobs to the whole pool.

**Proof jobs**

**Condor Pool**

+PROOF/Xrootd Pool
cpu + storage

Computing nodes
no storage

Storage Pool (Mainly for backups)

Less I/O Load
Multi-layer Condor System

- **Local Job Queue**: For Private Jobs, No number limitation, No run time limitation, Cover all the CPUs, Higher priority.
- **Fast Queue**: For high priority private Jobs, No number limitation, run time limitation, cover all the CPUs, with high priority.
- **PROOF COD Queue**: For PROOF Jobs, Cover all the CPUs, no affective to the condor queue, jobs get the CPU immediately.
- **I/O Queue**: For I/O intensive jobs, No number limitation, No run time limitation, Cover the CPUs in Xrootd pool, Higher priority.
- **Production Queue**: No Pre-emption, Cover all the CPUs, Maximum 3 days, No number limitation.

- **Local job Submission**: Users’ own jobs to the whole pool.
- **Proof jobs**: Users’ PROOF jobs to the Xrootd pool.
- **The gatekeeper**: Takes the production jobs from Grid and submits to local pool.
Production with PANDA

• Following new ATLAS production rules, we integrated into ATLAS central production system PANDA
• Currently, about dedicated 100 CPU cores are used by PANDA
The Principle of the I/O Queue

0. The cronjob provide all the file location in the Xrood pool.
1. Submission node ask Mysql database for the input file location.
2. Database provide the location for file and also the validation info of the file.
3. Submission node add the location to the job requirement and submit to the condor system.
4. Condor sends the job to the node where the input file stored.
5. The node runs the job and put the output file also to the local disk.
0. The cronjob provide all the file location in the Xrood pool.
I/O Queue Test Configuration

- Input file (ESD files) size ~700MB
- Output File (CBNTAA) size ~35MB
- Each machine has ~10 ESD files
- 42 running nodes
- 168 CPUs cores
Test Results

Direct Access vs Xrootd

Average running time (sec)

<table>
<thead>
<tr>
<th>Number of jobs</th>
<th>Average running time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>~800</td>
</tr>
<tr>
<td>50</td>
<td>~700</td>
</tr>
<tr>
<td>100</td>
<td>~600</td>
</tr>
<tr>
<td>150</td>
<td>~500</td>
</tr>
<tr>
<td>200</td>
<td>~400</td>
</tr>
</tbody>
</table>

Time save per job: ~230 sec
PROOF: Dynamic approach to end-user HEP analysis on distributed systems exploiting the intrinsic parallelism of HEP data
The end Point: Scalability

![Graph showing the relationship between processing rate and number of workers. The graph indicates an increasing trend as the number of workers increases.]
PROOF+COD Model

Normal Production Condor jobs

Condor Master

COD requests

PROOF requests

Xrootd Redirector

Condor + Xrootd + PROOF pool

The local storage on each machine

Local storage on each machine
Single user with 2 sessions
Integration of Condor and PROOF

• The new Condor+PROOF model try to address following issues:
  – Multi-session scheduling
  – Automatic dataset stage-in and stage-out of data

• See talk by G.Ganis
PROOF+CONDOR Model

- Active jobs
- Ready jobs, waiting for resource
- Waiting for input dataset to be staged in

PROOF sessions queue

File stager

PROOF pool

PROOF Master
Condor Master

D-Cache, CASTOR, Xrootdfs, Or DDM(Grid)
Outlook and Conclusions

• Computing demand in High Energy Physics as reached unprecedented levels

• The UW-ATLAS group uses capabilities of Condor quite extensively
  – Massive Monte Carlo production
  – Development of a production system
  – Development of model for Tier3
  – Development of distributed analysis system
    • Integration of Proof/Condor