ArCond

a front-end framework for Condor and parallel data processing using a distributed data storage

Sergei Chekanov (ANL)

Tier3 meeting, ANL
October 2009

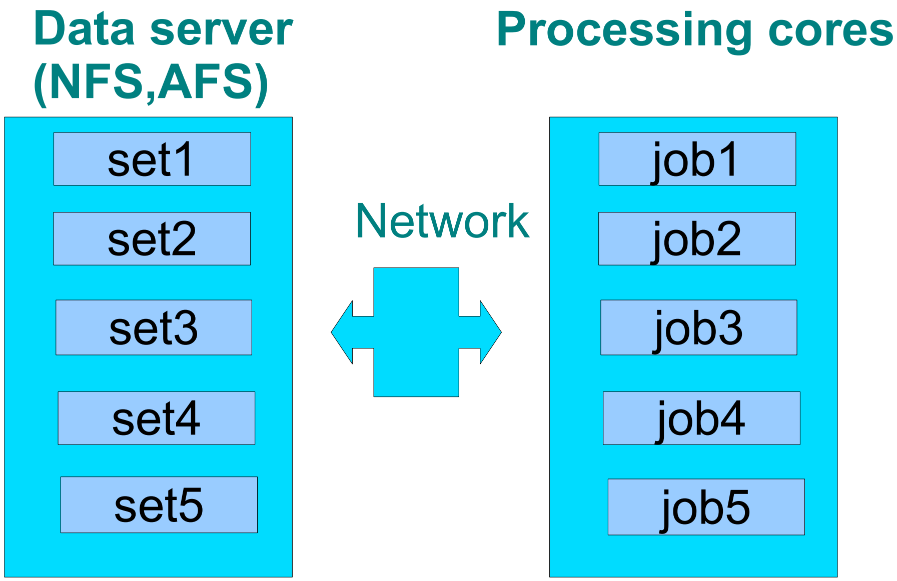
Condor

Condor is known package for batch clusters:

- Allows parallelization of jobs
- Provides scheduling policy
- Defines priorities
- Resource allocation

Ideal engine for Linux clusters
But when it comes to analysis using
input data, it requires additional features

Example:
Assume 100 files located on some storage. We want to run 5 parallel jobs.
Parallelism is achieved by slitting 100 files on 5 subsets, assigning each job to certain subset.



Data server (NFS,AFS)

Processing cores

Network

set1
set2
set3
set4
set5

job1
job2
job3
job4
job5

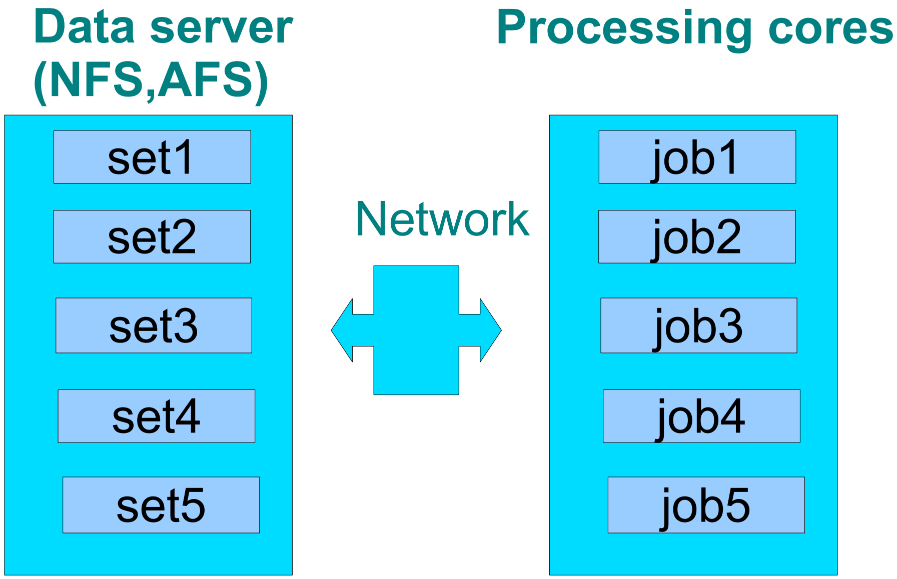
“Vertical scheme”

Needed features:
1) identify files on a file server,
2) split input file list
3) assign each core to each data list
4) return results and combine outputs

All of this should be done for Athena programs, ROOT/C++, Monte Carlo generation, NLO calculation etc. etc.

Hitting the network limits

- For a central file storage, network is used for I/O (at run-time)
- For typical Tier3, network bandwidth is 1 Gb or less
  - 1Gp usually means 400-500 Mbs due to various other limitations
- > 20 running athena jobs accessing data on the same file storage causes a significant performance penalty
  - Typical speed ~3 times slower compare to jobs accessing local disk I/O
  - ~20 running jobs accessing ANL NFS leads to non-operational cluster
  - For ROOT ntuples, the network I/O limit is 4-5 jobs



"Vertical scheme"

Solution?

- Equally partition data

- Redistribute data portions between computers with similar specifications

- Run jobs on data stored on local disks

- Do not use network for data access

Improving performance

“Divide and conquer” principle for low-cost clusters with commodity networks (<1 Gb):

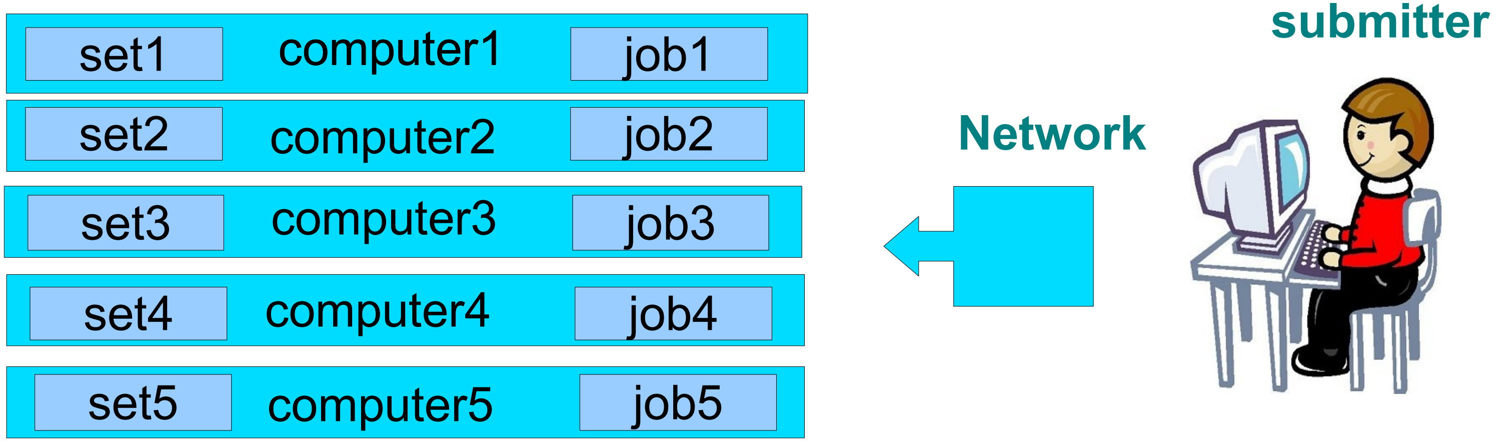
- Jobs should run on the same computer where data are
- Avoid any network load at run time. Use network for job submission/retrieval only
- Things are getting more complicated for Condor submissions

Features to be added for Condor processing:

1) identify files located on **different computers**
   - data discovery tool

2) Upload data on different computers
   - data splitter

Processing cores and data disks on the same computer



set1	computer1	job1
set2	computer2	job2
set3	computer3	job3
set4	computer4	job4
set5	computer5	job5

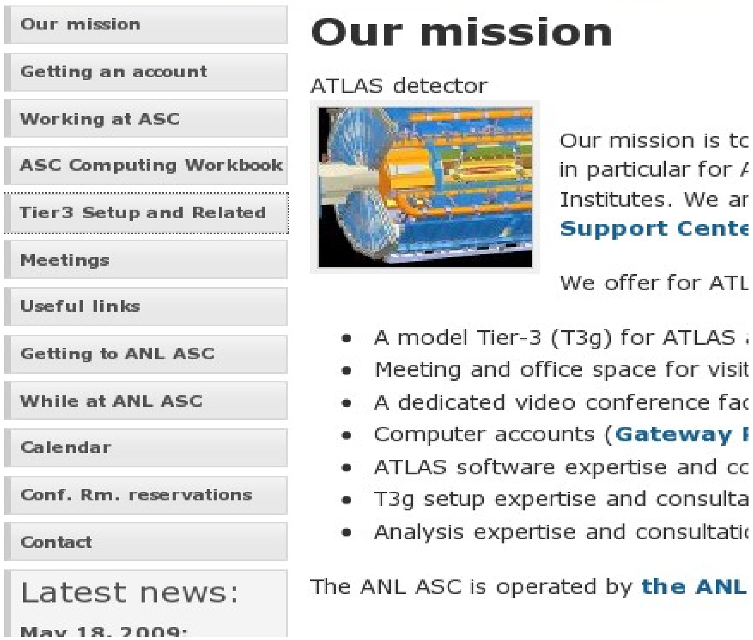
Network

submitter

“Horizontal scheme”

PC farm challenge for T3g sites

- **A complete T3G PC farm setup is given on the ANL ASC page (atlaswww.hep.anl.gov):**



Latest news:

May 18, 2009:

Our mission

ATLAS detector

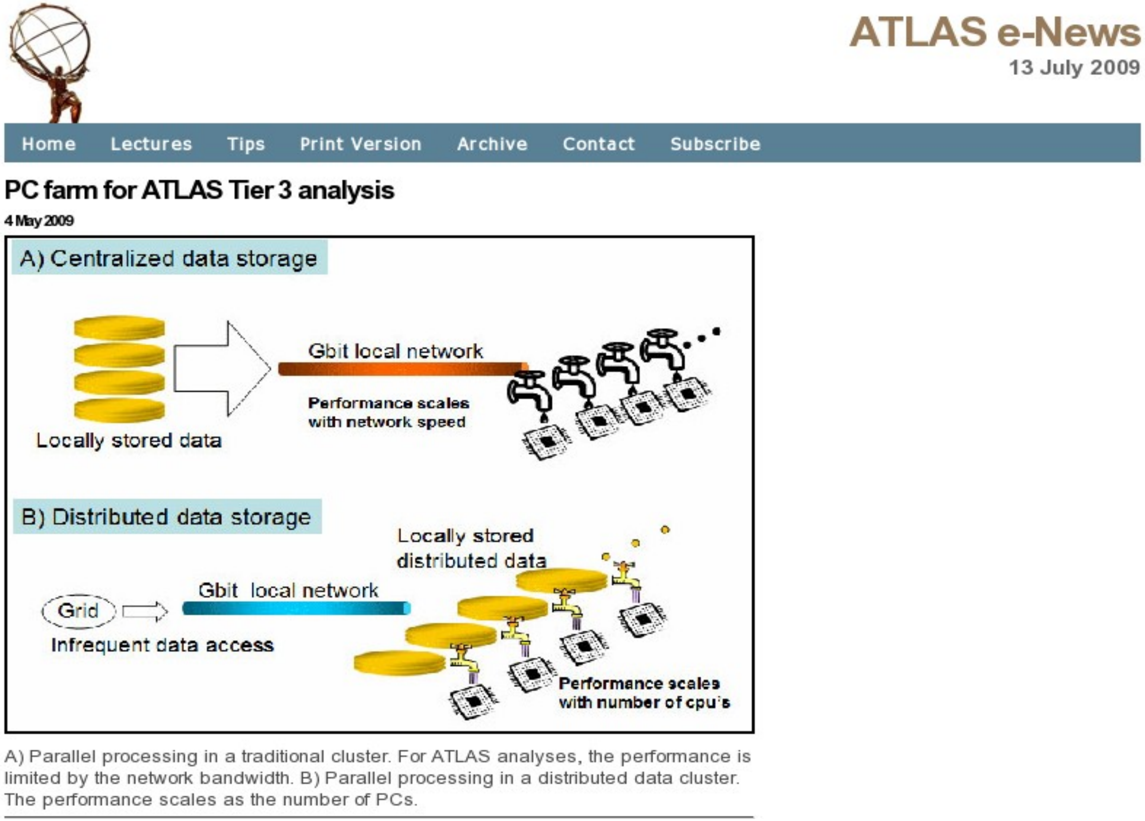
Our mission is tc in particular for A Institutes. We ar **Support Cente**

We offer for ATL

- A model Tier-3 (T3g) for ATLAS ,
- Meeting and office space for visit
- A dedicated video conference fac
- Computer accounts (**Gateway F**
- ATLAS software expertise and cc
- T3g setup expertise and consulta
- Analysis expertise and consultati

The ANL ASC is operated by **the ANL**

Article in ATLAS e-News


ATLAS e-News
13 July 2009
Home
Lectures
Tips
Print Version
Archive
Contact
Subscribe
A) Centralized data storage
Locally stored data
Gbit local network
Performance scales with network speed
B) Distributed data storage
Grid
Infrequent data access
Gbit local network
Locally stored distributed data
Performance scales with number of cpu's

PC farm for ATLAS Tier 3 analysis

4 May 2009

A) Parallel processing in a traditional cluster. For ATLAS analyses, the performance is limited by the network bandwidth. B) Parallel processing in a distributed data cluster. The performance scales as the number of PCs.

More details: "A PC farm for ATLAS Tier3 analysis"
S.C., R.Yoshida, ATL-COM-GEN-2009-016

***ArCond – Ar**gonne's **Cond**or*

- ArCond does all above and complements Condor for data-intensive jobs with input data.
- Python front-end for Condor for:
  - job submission, data discovery, results retrieval
- Can be used for athena packages, ROOT/C++ jobs, MC generation, etc.
- Can be used for a **central** storage or **distributed** storage
- Does not requite extra services. No maintenance
  - Data discovery is done using Condor itself ('pilot jobs')
  - Better solution - cron jobs to build lists with local files (optional)
- Developed and supported at ANL ASC
  - available version 1.4
  - Web site: http://atlaswww.hep.anl.gov/asc/arcond/

ArCond – Argonne's Condor

```
> arcond
```

- Reads a configuration file with:
  - ATLAS release version
  - input directory with input files
  - athena package name
- Splits jobs to be run in parallel: N=N(PC boxes) x N(cores)
- 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 defines execution sequence
  - may include multiple athena runs etc.
- Compiles programs using either NFS-based ATLAS software release or locally installed release
- When jobs are ready, the output is copied to the submission directory
  - optional, depends what do you put in shell script
  - output root files merged automatically

Benchmarks

Types of job submissions & benchmarks for 24 cores Harpertown Xeon (5400), 2.2 Ghz

- **Running over AOD files**
  - 0.5M events /h
- **Fast MC simulation and on the fly analysis**
  - 1.5M events /h
- **Running over C++/ROOT ntuples**
  - 1000M events /h (1M events / min for 1 core)
- **Generating MC truth ntuples**
  - 2.5M events /h
- **AOD production (generating & reconstructing MC events)**
  - 120 events /h

Note: 5500 (Nehalem) processors are ~100% faster than Harpertown Xeon (5400)

- 5000 jobs since 2008. Tested by ~20 users

~80% athena programs
~15% ROOT/C++
~5% Fast MC simulation and full MC reconstruction

<0.01 failure rate

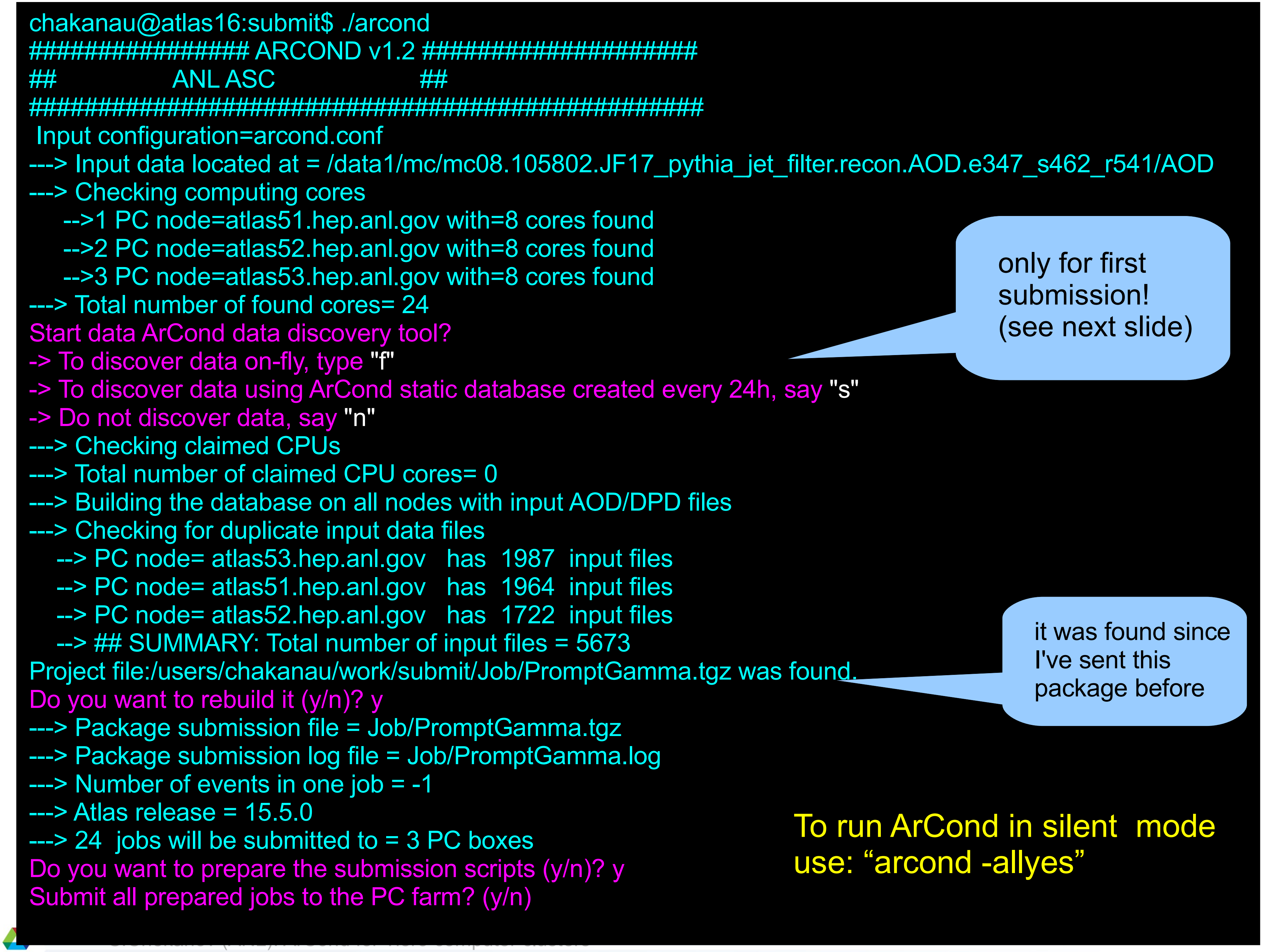
Running arcond

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

```
atlas_release=15.5.0

# events to process in each job
events = -1

# dir with input AOD files.
input_data = 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

Submitting job..



```
chakanau@atlas16:submit$ ./arcond
################# ARCOND v1.2 ######################
##                   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 = 15.5.0
---> 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

Several choices:

- **“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 exist
  - Arcond sends 'pilot' jobs on each PC boxes and generates lists with input files
  - 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)**

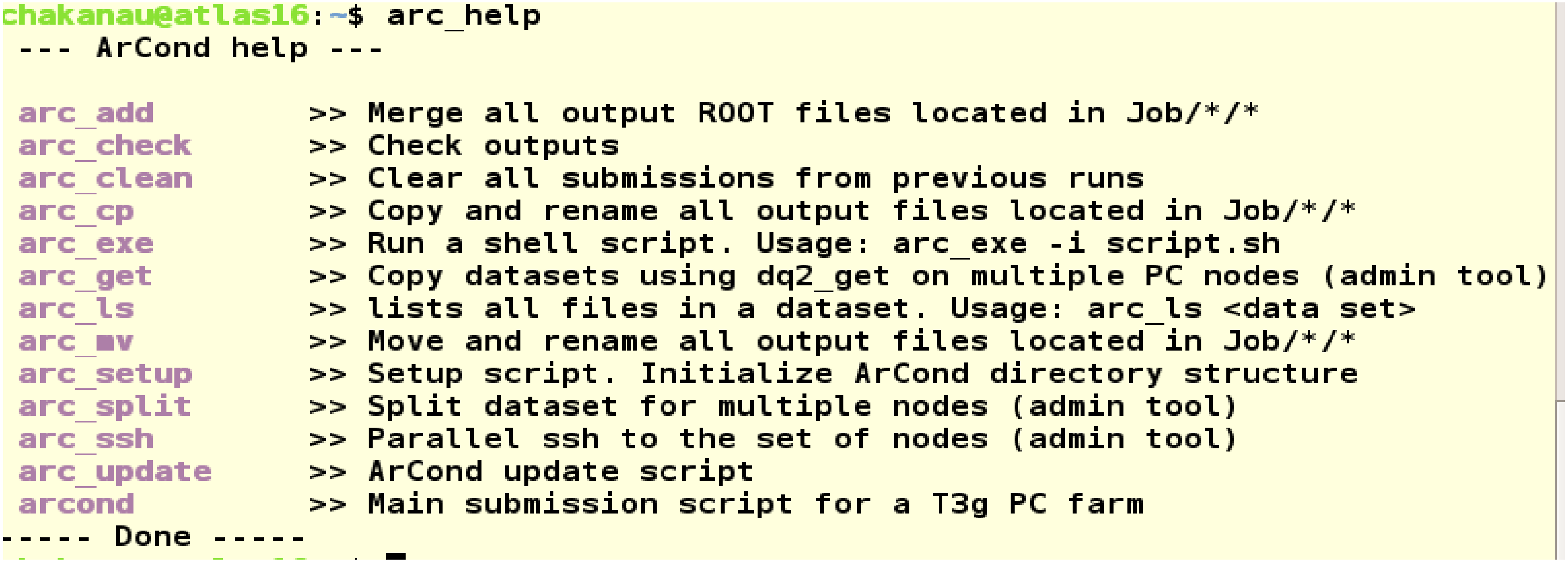
Simple and robust. So far required no attention from admin.

Getting data

- **So far is based on dq2-get:**
  - Works very well
  - Keeping about 20k AOD/DPD files, 45 MC data sets distributed between 3 computer nodes
- **Main issue is how to redistribute a data set between different nodes**
  - **Used solution:**
    - Get data on one node, use ArCond splitter to divide data. Copy sets on other nodes
- **A better solution is to add a “splitting” functionality to dq2-get**
- **ArCond provides a front-end of dq2-get which allows to divide sample during downloads**
  - *arc_ssh -h hosts-file -l <user-name> -o /tmp/log "exec send_dq2.sh”*
    - Gets a list of files. Splits in ranges depending on number of slaves.
    - Executes dq2-get on each slave node using this list.
  - Tested using 5 Linux boxes (five dq2-get threads). For many nodes, the number of threads included by dq2-get can be reduced
  - 3-4 TB/day for several Tier2 & BNL Tier1
- **Will explore other solutions “subsciption”?, xrootd? Setting a test cluster**

Other ArCond features

- Built-in help



```
chakanau@atlas16:~$ arc_help
 --- ArCond help ---

 arc_add          >> Merge all output ROOT files located in Job/*/*
 arc_check        >> Check outputs
 arc_clean        >> Clear all submissions from previous runs
 arc_cp           >> Copy and rename all output files located in Job/*/*
 arc_exe          >> Run a shell script. Usage: arc_exe -i script.sh
 arc_get          >> Copy datasets using dq2_get on multiple PC nodes (admin tool)
 arc_ls           >> lists all files in a dataset. Usage: arc_ls <data set>
 arc_mv           >> Move and rename all output files located in Job/*/*
 arc_setup        >> Setup script. Initialize ArCond directory structure
 arc_split        >> Split dataset for multiple nodes (admin tool)
 arc_ssh          >> Parallel ssh to the set of nodes (admin tool)
 arc_update       >> ArCond update script
 arcond           >> Main submission script for a T3g PC farm
----- Done -----
```

- Example: List data files on all nodes:
  - *arc_ls <dataset>*
  - Prints location of each file (computer, directory)

Data recovery

- **One feature of ArCond is simplicity**
  - No need to be an IT expert
  - Only basic knowledge of Linux
  - No any particular “file system”
  - Extends desktop environment + replace condor commands
  - All operations are transparent and do not require extra knowledge
- **Did not exercise data recovery when a disk or a computer fails**
  - datasets lost fractions of files (and less CPU)
  - ArCond does not offer solution. But data recovery is simple!
    - Main steps:
      - arc_ls /data1   - lists all files located on the disk /data1
      - Consider only files on failed node
      - Make a list with missing files
      - Get lost data from the grid (dq2-get) or a central storage
      - All of this require basic knowledge of bash,python,sed etc..
      - Can be provided in future

Summary

- **Experience with “distributed” analysis & ArCond model for more than a year**
  - No problems found, fault rate <0.01
  - Tested by ~20 users
  - Used for 24-core PC farm
- **One prominent feature – simplicity. No need to be an IT expert. No maintenance**
  - knowledge of Linux is sufficient
- **ArCond version 1.4**:
  - Tested for NFS3/NFS4 and most recent Condor (7.2.3)
- **A full-scale computer farm + Tier3 integration cluster are under development**
- **The most outstanding issue is how to redistribute data between computers**
  - ArCond offers several choices
  - Subscription, xrootd will be tested. Xrootd can be merged with ArCond