Enabling data analysis à la PROOF on the Italian ATLAS Tier-2s using PoD

CHEP2012 – Computing in High Energy and Nuclear Physics 2012, May 21-25 2012 New York, United States

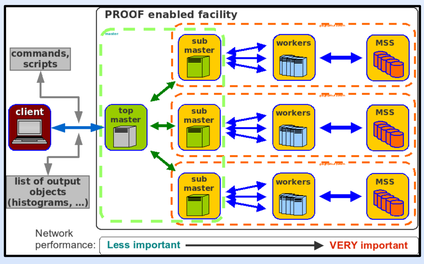
R. Di Nardo[1], G. Ganis[2], E. Vilucchi[1], A. Annovi[1], M. Antonelli[1]
G. Carlino[3], A. De Salvo[4], A. Doria[3], A. Manafov[5], A. Martini[1], M. Testa[1]

[1] INFN Laboratori Nazionali di Frascati, [2] CERN/PH-SFT, [3] INFN Napoli, [4] INFN Roma1 and Università La Sapienza,
[5] GSI Helmholtzzentrum für Schwerionenforschung, Darmstadt

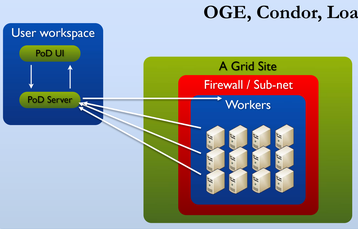
On behalf of the ATLAS Collaboration

PROOF and PoD

- PROOF (Parallel ROOT Facility) allows interactive analysis on a set of distributed resources using a multi-tier master-worker model to achieve dynamic workload-balancing.



- PoD (PROOF on Demand) → tool-kit to setup a PROOF cluster on any Resource Management System (RMS)
  - Currently supported backends: LSF, PBS, OGE, Condor, LoadLeveler and gLite-WMS



Test setup

- PoD and ROOT setup from the CVMFS ATLAS distribution.
  - PoD version 3.10
  - ROOT 5.32/02

PROOF daemons on Tier2 cluster with PoD

- Distributed analysis back-ends:
  - PanDA, WMS, CREAM and ARC
- PoD submission on the Grid via the gLite-WMS plug-in
  - Future development : PoD plugin for PanDA back-end
    - ATLAS central accounting system reports PanDA jobs only

```
$ pod-server start
Starting PoD server...
updating xproofd configuration file...
starting xproofd...
starting PoD agent...
preparing PoD worker package...[...]
-----------------------
XPROOFD [27630] port: 21001
PoD agent [27653] port: 22001
PROOF connection string: evilucch@atlas-ui-02.roma1.infn.it:21001
-----------------------
$ pod-submit -r glite -q atlasce2.lnf.infn.it:8443/cream-pbs-atlas_short -n 100
$ pod-info -n
45
```

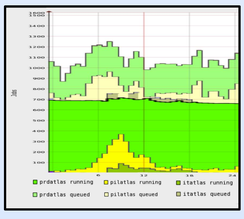
- User accesses Grid resources connecting to a User Interface (UI)
- Starts PoD server on the UI
- Allocation of n Tier-2 Worker Nodes
  - Submission of n jobs to the CREAM CE via WMS
- Check the number of allocated nodes :
  - WN with a PROOF daemon running ready for the analysis job
- Starts the analysis with PROOF on the available nodes
  - […] `TProof *p = TProof::Open("pod://");` […]

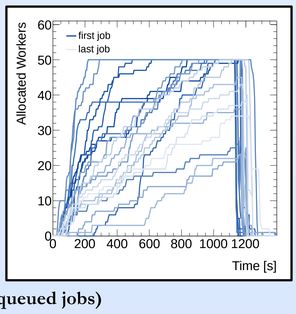
Startup Latency

- Startup latency: time necessary to allocate a certain number of nodes with PoD before to run PROOF analysis.
- Startup latency tested in Frascati. The latency depends on the share allocated in the scheduler and many other parameters, e.g. cluster load.

Test I

A job requiring 50 cores submitted every 30 minutes for a total of 21 submissions. The proxy had VOMS group /atlas/it and 25% of computing share.





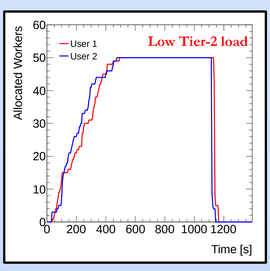
- Left: Cluster load during the job submissions from Ganglia (darker colors: running jobs; lighter colors: queued jobs)
  - Green: Monte Carlo production jobs; yellow: PanDA analysis; olive green: PoD submissions as /atlas/it
- Right: available workers as a function of the time since submission in a 20 minutes window; the color scale is proportional to the job submission time (from darker to lighter)

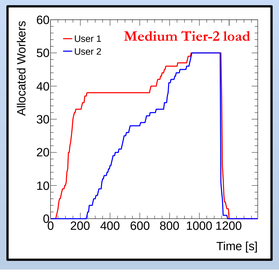
The results of the test are in agreement with the expectations from the cluster load and fairshares.
First submissions suffer from resource competition with PanDA analysis of generic ATLAS user jobs, whose computing fairshare was 25%

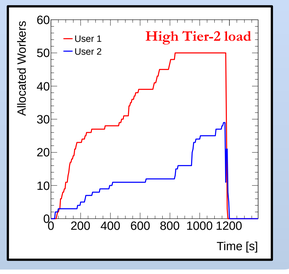
Test II

Two users with the same VOMS credential in competition for the same resources.

As expected, for light loads there is no resource competition, while for heavy loads one of the users is not even able to allocate all the requested nodes in the monitoring time window (20 minutes).
However, even in the worst case, the user can start working with 10 workers after 7 minutes.







Readout Performance

Input rate in MBytes/second using the PROOF statistics tools as a function of the number of workers.

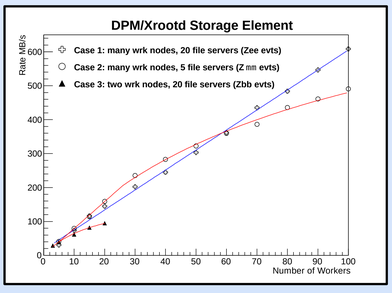
- Test performed at Roma1, Frascati and Napoli Tier-2 with DPM (Disk Pool Manager) as SRM with Xrootd access protocol (configured for local access only).
- Workers nodes and file servers connected with 1Gbit/s and 10Gbit/s network switches, respectively.

Three typical configurations, depending on cluster load, fairshare and dataset:

Case 1: Workers distributed over many nodes, dataset files distributed over many file servers

Case 2: Workers distributed over many nodes, dataset files concentrated over few file servers

Case 3: Workers on few nodes, dataset files over many file servers



- Slope for small number of workers measures the rate of reading and decompressing the event per worker.
  - 8 MBytes/s per process, depend on the type of analysis and on the structure of the event read and built in memory.
- Scalability indicates how the system react to increasing worker loads (e.g. increasing number of users) .
  - In Case 1, the effective network bandwidth is large because both storage and processing elements are widely distributed; good scalability over the range of numbers of workers tested;
  - In Case 2, deviations from linear scalability start to be visible due to network bandwidth limitations on the file servers (in the test, the dataset was distributed over 5 servers, 3 out of which temporary connected with 1 Gbit/s switches);
  - In Case 3, saturation starts much earlier because of the 1 Gbit/s network connections of workers (in the test, the PoD fairshare was set to 5% and workers were concentrated on two nodes).

The results underline the importance of a good network setup for efficient data serving to multiple processes. Even distributions for files across data servers and for processes across workers nodes allow to better exploit the available resources.

References

- http://root.cern.ch/drupal/content/proof
- http://pod.gsi.de
- C.Aguado-Sanchez *et al*, *"Studying ROOT I/O performance with PROOF-Lite"*, 2011 *J. Phys.: Conf. Ser.* **331** 032010