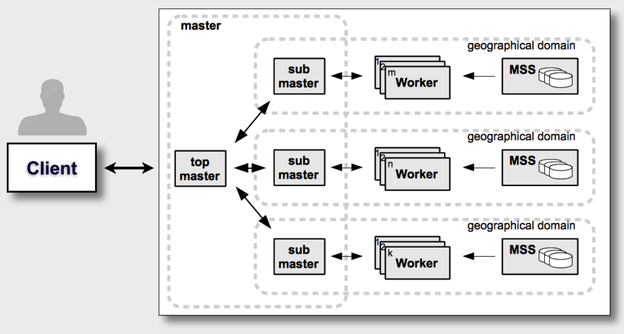
PROOF - Status and New Developments

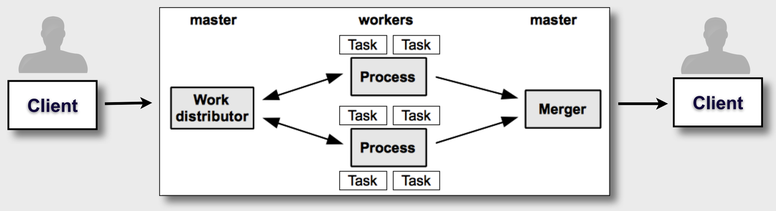
Gerardo Ganis, Fons Rademakers
CERN

What is PROOF?

PROOF - Parallel ROOT Facility - is an extension of ROOT [1] enabling interactive analysis in parallel on clusters of computers or many-core machines. PROOF provides an alternative to the traditional batch-oriented exploitation of distributed computing resources. PROOF uses a multi-process approach to implement basic parallelism at event-level, which naturally fits the case of HEP analysis.
PROOF has a flexible multi-tier client-master-workers architecture, supporting the possibility to run in a cluster of geographically separated clusters:



The applications running in the master and worker nodes are real ROOT sessions forked via a **dedicated protocol** plugged in the main component of the **XROOTD** daemon [2],
Dynamic load-balancing is achieved by using a pull model for work distribution (workers ask for more work as soon as they are ready)



The master is also in charge of merging the output it collects from the worker nodes, so that the client receives a single set of output objects, as would have been the case for local processing.

Where PROOF is used?

The LHC data will be primarily stored in ROOT format[3]. To be analysed in reasonable times, the large amount of data expected require a system of distributed resources. The PROOF dynamic approach allows for better adaptation to the varying and unpredictable work-load typical of the end-user analysis phase, and represents a natural candidate for Tier 3 or departmental analysis facilities. In addition, The lite version of PROOF provides a quick and basically unique way to exploit today multi-cores desktops .

The ALICE collaboration requires a PROOF service as integral part of its computing model [4]. Currently the main ALICE installations are the CAF at CERN, the SKAF in Slovakia, GSIAF at Darmstadt. CAF is also used for prompt calibration and reconstruction for very fast QA response.
The ATLAS US groups are prototyping Tier 3 models based on PROOF or having PROOF as service since 2007 [5]. The main ATLAS facilities are at UW-Madison and BNL; test facilities exist also in europe at LMU-Munich and in Spain.
CMS groups at Oviedo, Spain, Florence, Italy and DESY[6], are also testing the system.

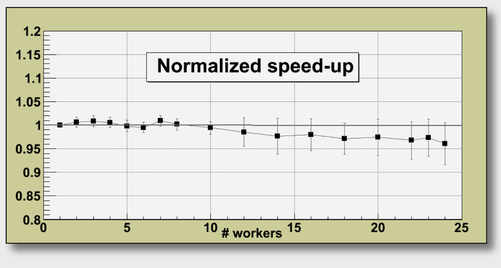
PROOF-Lite: exploiting the cores of desktops and laptops

PROOF-Lite is a 2-tier implementation of the PROOF paradigm designed for multi-core machines. In the lite version, the client is directly in control of the workers

PROOF-Lite is a 0-config product: no need for daemons or configuration files. PROOF-Lite runs the same code run in a PROOF standard cluster: it can be used to test/debug the code before moving to a larger facility

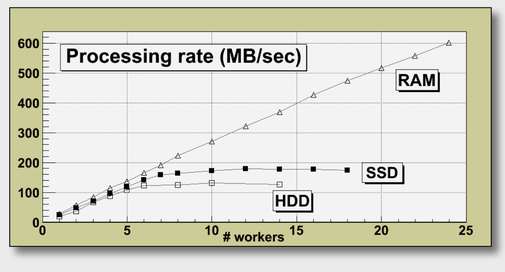
The normalized speed-up for CPU-bound tasks is shown in the figure below; the vertical scale is zero-suppressed. Non-linearities are due to the serial phases, notably output merging.

The machine used here has 24 cores and 48 GB RAM.



For I/O-bound tasks, PROOF-Lite exploits the full available IO bandwidth. The impact of different IO hardware speeds on performance scalability is shown in the figure.

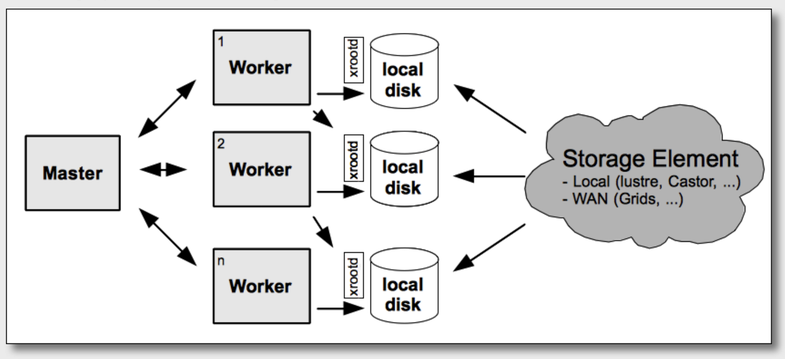
- **HDD:** RAID0 array of SAS disks 10 Krpm/s
- **SSD**: RAID0 array of solid state disks;
- **RAM**: tasks become CPU bound



PROOF-enabled LAN Analysis Facilities

Standard LAN clusters

The typical configuration for a PROOF-enabled analysis facility (PAF) currently considered is shown below



Typical nodes have 8 or 16 cores, at least 2 GB/core of RAM and a fair amount O(TB) of local disk to be used as pool cache. The cluster gets the data from a storage element which can be the Grid or a local high performance storage system (e.g. Castor for the ALICE CAF@CERN, Lustre for the GSI-Darmstatd facility).
Local caching of data, together with RAM caching, can help for frequently accessed files. A good network hardware layout may increase the number of effective independent I/O devices and increase the available I/O bandwidth.

The issues addressed recently for PAFs related mostly to dataset managent, multi-user running and merging/ handling-of large outputs.

Dataset management

The concept of *dataset* is common to all HEP experiments and helps end-users to properly define their analysis samples. PROOF provides an easy-to-use and flexible way to access to dataset meta information via small ROOT files, which can be easily filled using with the tools provided by the experiments (e.g. ATLAS DQ2 [7] or the ALICE AliEn catalogue [4]). The same technology can be used to access the official datasets and to create / handle the private datasets produced during the analysis. This functionality will be extended to handle *entry-list*, bit-based lists of events passing some filters.
The dataset information is used by ALICE to steer staging in / out of data to / from the local pool.

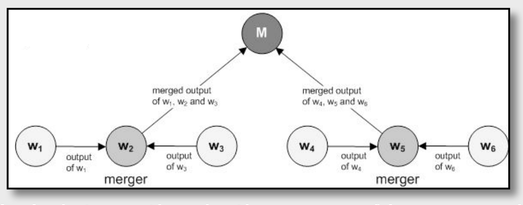
Multi-user/priority management

PROOF provides a way to conrol the number of users running concurrently by either refusing access or delaying processing. It also provides the possibility to control the priority of the running sessions which can be centrally steered by a monitoring systems.

The ALICE experiment uses the information in MonAlisa [8] to calculate the usage history in a given time-window and modify the priorities accordingly [9].

Output merging optimizations

Depending on type, size and number of the output objects. merging the results of each worker can be an expensive operation, For large outputs PROOF provides the possibility to save the results in files on the workers. These files are optimally distributed for further processing and can be automatically registered as a new dataset. If a single file is needed, PROOF can run the optimized file merging procedure available in ROOT.
PROOF also provides the possibility to merge objects in parallel promoting some of the workers as *sub-mergers* at the end of the run following the schema



This is particularly interesting in the case of large numbers of small objects (e.g. histograms) whene the speed-up using sub-mergers can be substantial.

PROOF and Resource Management Systems

The idea behind the integration of PROOF with resource management systems (RMS) is a better utilization of available resources. Good interactive response required the resources to not be completely overloaded. However, it is not always possible to dedicate a set of machines to PROOF. A RMS can be used to partition the usage of resources so that one can profit from the PROOF technology when needed, yet having an underlying batch activity ready to take over.
Several models of PROOF-RMS integration have recently seen the light. The basic idea is to use the RMS to create a private PROOF cluster for each user. This has the advantage to run each user in its own space, automatically enforcing user encapsulation and the privacy of the working area, leaving to the RMS all related security issues. Typically, the PROOF sessions have higher priority and priority policies can/are insured via the RMS scheduler.

A model based on the SGE batch system has been developed at DESY for the NAF [10]. The ATLAS group at UW setup a prototype using the capabilities Condor system to launch PROOF daemons and control the fraction of *slots* dedicated to PROOF [11].
The most generic and flexible integration is the one provided by the PoD system (Proof-On-Demand) [12], developed at GSI-Darmstadt. PoD provides a virtual RMS interface to a RMS via a plug-in system. A nice GUI allows to start, monitor and control the active nodes.

Conclusions

PROOF is currently being widely adopted or seriously considered as Tier-3 solution by the LHC experiments.
Native PROOF on local clusters is steadily maturing, PROOF-Lite already attracted several end-users for desktop / laptop usage. The integration with resource management systems has much progressed and represents an attractive way to enable PROOF on existing facilities.
The PROOF system is becoming a full-featured solution for LHC analysis.

References

1. PROOF: http://root.cern.ch/drupal/content/proof; for ROOT see http://root.cern.ch .
2. xrootd: http://www.slac.stanford.edu/xrootd.
3. LCG TDR, CERN-LHCC-2005-024 and references therein.
4. ALICE TDR, CERN-LHCC-2005-018; for AliEn see http://alien.cern.ch/twiki/bin/view/AliEn/Home .
5. S.Panitkin et al., “Distributed analysis with PROOF in ATLAS collaboration”, to appear in JPCS/CHEP09.
6. A.Haupt, Y.Kemp, “The NAF: National Analysis Facility at DESY”, to appear in JPCS/CHEP09.
7. B.Fernando-Harald et al., “The ATLAS DQ2 accounting service”, to appear in JPCS/CHEP09.
8. MonALISA: http://monalisa.cacr.caltech.edu/monalisa.htm .
9. M.Meoni et al., “Status of the ALICE CERN Analysis Facility”, to appear in JPCS/CHEP09.
10. H.Stadie, W.Behrenhoff, “Integrating interactive PROOF into a Batch System”, to appear in JPCS/CHEP09.
11. Neng Xu, Univ. Wisconsin, private communication; for Condor see http://www.cs.wisc.edu/condor/.
12. PoD: http://pod.gsi.de/; see also A,Manafov et al, “PROOF on Demand”, to appear in JPCS/CHEP09.

February 2010