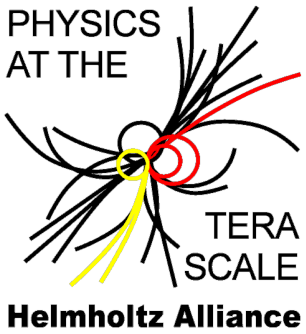
LHC Data Analysis with PROOF



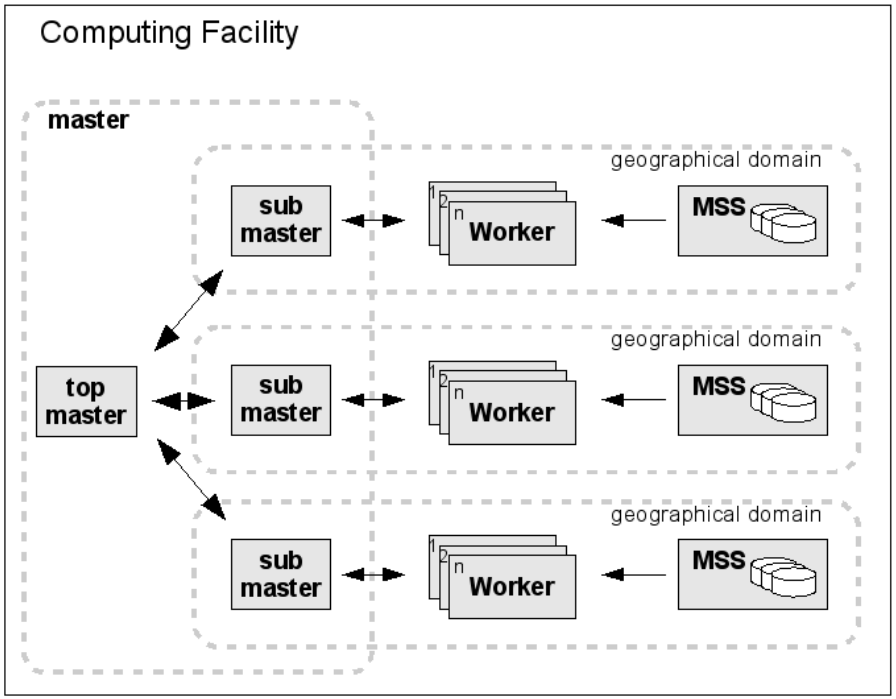
Wolfgang Ehrenfeld, Jörgen Samson (DESY)
Grid Project Session
4th Annual Helmholtz Alliance Workshop on
"Physics at the Terascale"
Dresden, 2.12.2010

HELMHOLTZ
ASSOCIATION

DESY

PROOF

- PROOF in a nut shell:
  automatic and intelligent distribution of work over PROOF cluster
  - file splitting done by PROOF, optimise throughput on client
  - transparent merging of histograms and ntuples
  - interactive session
  - PROOF Lite for multicore machines (desktops)



- Where does PROOF fits into the analysis chain?
  - Desktop/Laptop: ~1min, ~1GB data
  - PROOF analysis: ~10min, ~100GB data
  - Grid analysis: 1h – 1d, 1 - 10 TB

- PROOF is easy to use (besides having a PROOF cluster)
  - Derive from Tselector MakeClass() --> MakeSelector()
  - TSelector based analysis frameworks, e.g. SFrame

DESY

PROOF on Batch

- a dedicated PROOF cluster has some disadvantages
  - idle resources if unused
  - centrally installed and maintained by cluster admins
  - accounting, priorities, authentication
  - problems with jobs of one user can effect the whole cluster and hence all other users
  - choice of ROOT version
- start dedicated PROOF cluster on a number of batch nodes if requested by individual users
  - solves instantly accounting, priorities, authentication using the batch mechanisms
  - reduces unnecessary idle resources
  - jobs from user A can not interfere with jobs from user B
  - gives more choices of software versions to the user
  - **BUT setup tools and configuration needs to be developed and maintained**

LHC Data Analysis with PROOF Project

- Common Project between DESY (W. Ehrenfeld, J. Samson), LMU (G. Duckeck) and University of Hamburg (H. Stadie)
- Develop a set of tools to easily configure and operated a dedicated PROOF cluster on a batch system
  - starting from a first version developed by Uni Hamburg for CMS for the NAF
  - add support for experiment software (ROOT and event data dictionaries) (LMU)
  - make tools more modular, robust and user friendly (usable for daily work) (DESY)
  - start with the SGE batch system at the NAF (Hamburg/DESY) and then add modular support for other batch systems, maybe even for the Grid
- Gain experience from real analysis
  - improve and optimize the developed tools and PROOF (DESY)
  - adaptation of analysis frameworks, e.g. Sframe (DESY)
  - study the limitations of PROOF (all)
- Educate and train users
  - ATLAS-D computing tutorials (2009) (LMU), (2010) (DESY)

DESY

proofcluster Script

- PROOF is written with a global cluster in mind, specially an always running PROOF master
  - some caching (compiled code, dataset information) is done on the master, which is lost after a restart
- some tuning of PROOF is needed to achieve the same overall performance
  - compilation of user code on worker nodes is not need if farm is homogenous or can be done beforehand by the user
  - validation of input files is not needed if checked by the user or a data distribution system beforehand
  - passing environment variables into the PROOF cluster is complicated and not straightforward

DESY

Current Usage

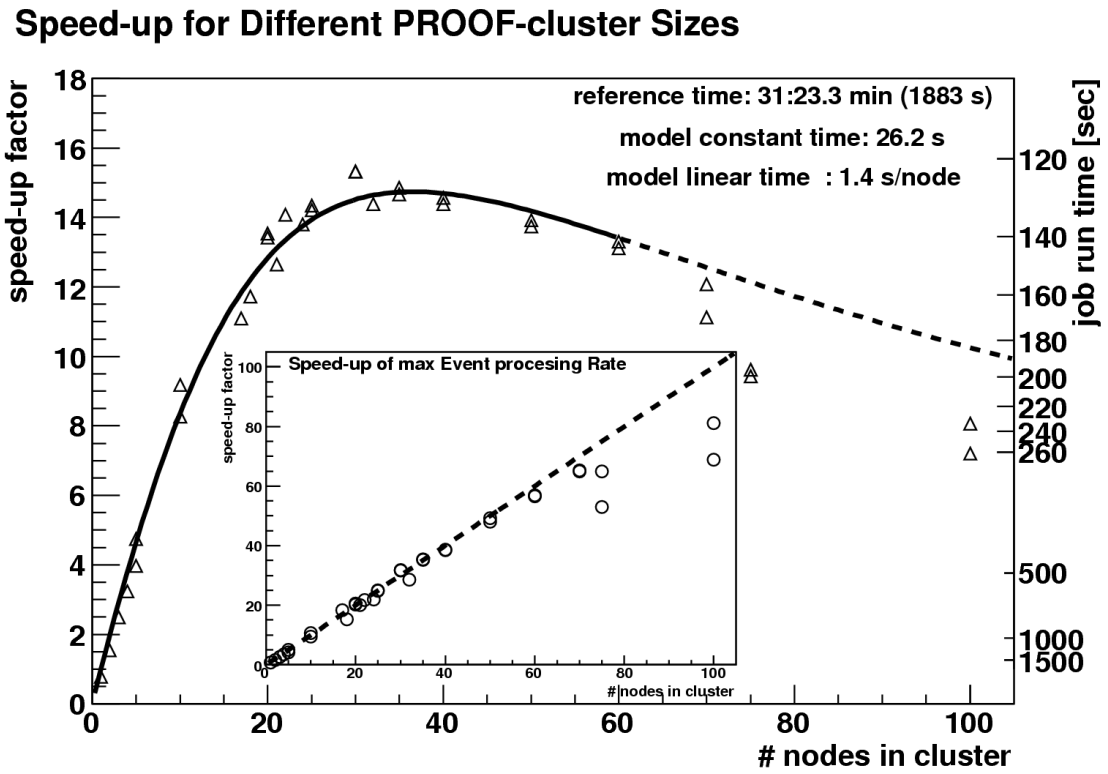
- proofcluster scripts used by ATLAS on the NAF
  - Accounting between 1.8. - 30.11.2010
  - 10 user with 952 proof sessions --> ~17 slots per proof session
  - ~1000 days of wall clock time --> ~0.6% of total NAF wall clock time
  - at least 75% of the clusters are stopped before hitting the CPU limit
  - a couple of users use the NAF multicore environment to run PROOF Lite (no problems with setting environment variables)
- some more features still expected as we better understand the interaction of PROOF with the user code and with the batch system

```
[efeld@tcx120]~% proofcluster config
Configuring PROOF cluster
-------------------------
PROOF master server name
Number of workers
Choose the site
Select queue
RAM required per worker (in MB)
xrootd port (1094)
xrootd protocol port (1093)
Path to ROOT installation or auto for automatic selection

[efeld@tcx120]~% proofcluster start
Starting master (tcx120)...
running on: tcx120.naf.desy.de
Starting clients...
Executing: qsub -pe proof 20 -notify -l h_cpu=00:59:00 -R y -l
h_vmem=2000M -l site=hh  -o Your job 2844814 ("proofjob")
Running!

[efeld@tcx120]~% proofcluster status
Master:
 5880 ?        Sl     0:00 xrootd
Workers:
2844814 proofjob efeld  r proof.q@tcx134.naf.desy.de  20

[efeld@tcx120]~% proof

[efeld@tcx120]~% proofcluster stop
Shutting down worker job-id 2844814
efeld has registered the job 2844814 for deletion
Shutting down xrootd, pid=5880
```


DESY

Scalability Studies

- Study scalability using real user analysis code (SUSY di-tau search)
  - based on flat ntuples (D3PD)
  - 7 Mev, ~300 files, ~650 MB
  - input files from dCache
  - reads 250 out of 3300 branches
  - creates ~600 histograms
- in this test, vary number of PROOF slaves in cluster
  - speed up from 30 minutes to 2 minutes possible
  - 20 slaves give already good performance without wasting much CPU
  - speed of larger cluster sizes dominated by overhead for starting up and merging.
  - optimal cluster size depends on number of events and number of files



Documentation, Training etc.

- Project webpage:
  - https://wiki.terascale.de/index.php/PROOFOnBatch
- Papers, Posters, etc.:
  - Poster at ICHEP2010: PO-MON-037: PROOF on a Batch System
    https://wiki.terascale.de/images/7/7f/CHEP10_PROOF.pdf
  - Diploma thesis: W. Behrenhoff: Entwicklung interaktiver Analysewerkzeuge für das CMS-Experiment
    http://www.desy.de/~wbehrenh/diplom.pdf
- ATLAS/CMS specific WIKIs:
  - https://wiki-zeuthen.desy.de/ATLAS/WorkBook/NAF/PROOF
  - https://twiki.cern.ch/twiki/bin/view/CMS/HamburgWikiComputingNAFPROOF
- User training:
  - ATLAS-D computing tutorials 2009 (Bonn), 2010 (Mainz)

Summary and Outlock

- PROOF is one way to easily parallelise user analysis
- PROOF on a batch system is a viable alternative to a global PROOF cluster
- within the 'LHC Data Analysis with PROOF' project a set of tools was developed to easily configure and create user proof clusters on the NAF batch system
- first experience with real user analysis and real collision data has been collected and the chosen approach is working quite well

- Next steps:
  - finalise development
  - evaluate Proof on Demand (DGrid project@GSI)
  - optimize interaction with storage systems (dCache, Lustre, NFS4.1)
  - more user training: both basic tutorials and support for complex analysis cases

DESY