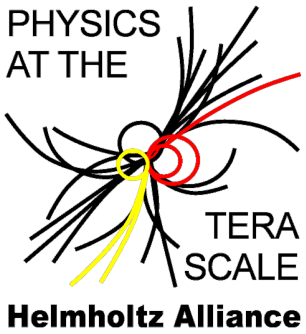
LHC Data Analysis with PROOF



Günter Duckeck (LMU), Wolfgang Ehrenfeld (DESY),
Hartmut Stadie (Hamburg)

Grid Project Session
5th Annual Helmholtz Alliance Workshop on
"Physics at the Terascale"

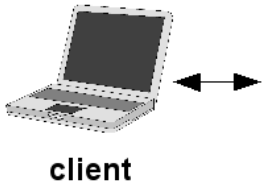
Bonn, 8.12.2011

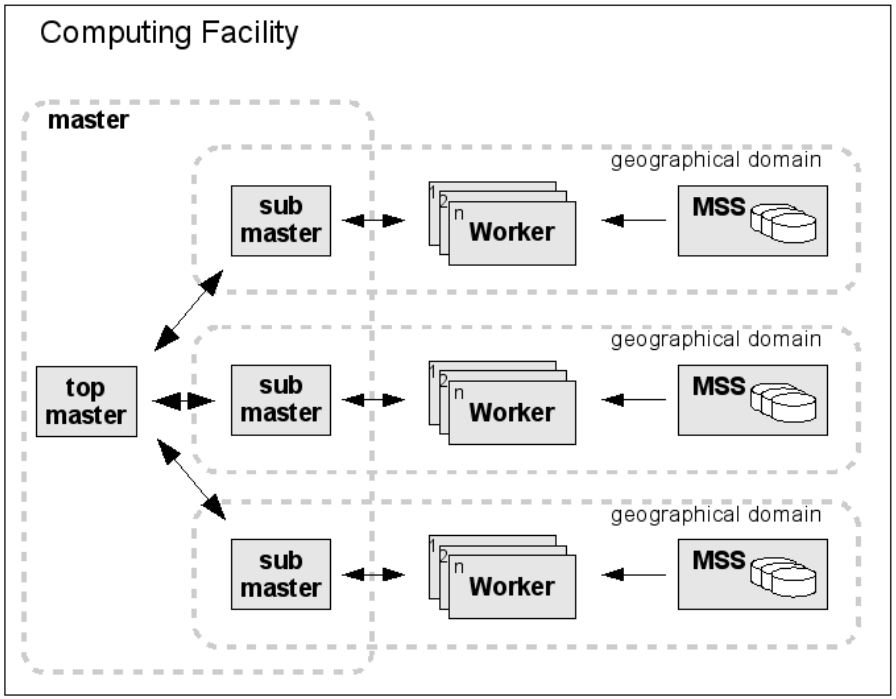
HELMHOLTZ
ASSOCIATION

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 at runtime
  - transparent merging of histograms and ntuples
  - interactive session
  - PROOF Lite for multicore machines (desktops)





- Where is PROOF used in the analysis chain?
  - mainly for Ntuple analysis (~100 GB - ~1 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 Cluster

- 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 by the experiment support**

PROOF on Batch / PROOF on Demand

- We developed and maintain a set of tools to easily configure and operate a dedicated PROOF cluster on a batch system
  - for details see last years report
  - mainly used by ATLAS on the NAF
  - needs man power to maintain and adjust to PROOF developments
  - needs some adjustment if used at other sites
- PROOF on Demand (PoD)
  - http://pod.gsi.de/
  - PROOF on Demand (PoD) is a tool-set developed at GSI, which sets up a PROOF cluster on any resource management system. PoD is a user oriented product with an easy to use GUI and a command-line interface. It is fully automated. No administrative privileges or special knowledge is required to use it.
  - used by ATLAS (LMU) and CMS (Uni Hamburg/NAF)

PROOF on Demand on the NAF (CMS)

PROOF on Demand

- replaced old “proofcluster”-script with PROOF on Demand developed at GSI
- easy to create and use a large PROOF cluster on the batch farm
- workers via array jobs using the special “proof” parallel environment of the NAF batch
- worker inherit server environment
- documentation for CMS users

Example

```
. /afs/naf.desy.de/group/cms/proof/PoD/PoD_env.sh
pod-server start
#start 50 workers
pod-submit -q proof.q -n 50 -r ge
pod-info -n #number of workers
pod-server -c #connect string
```

Example: Z + jets

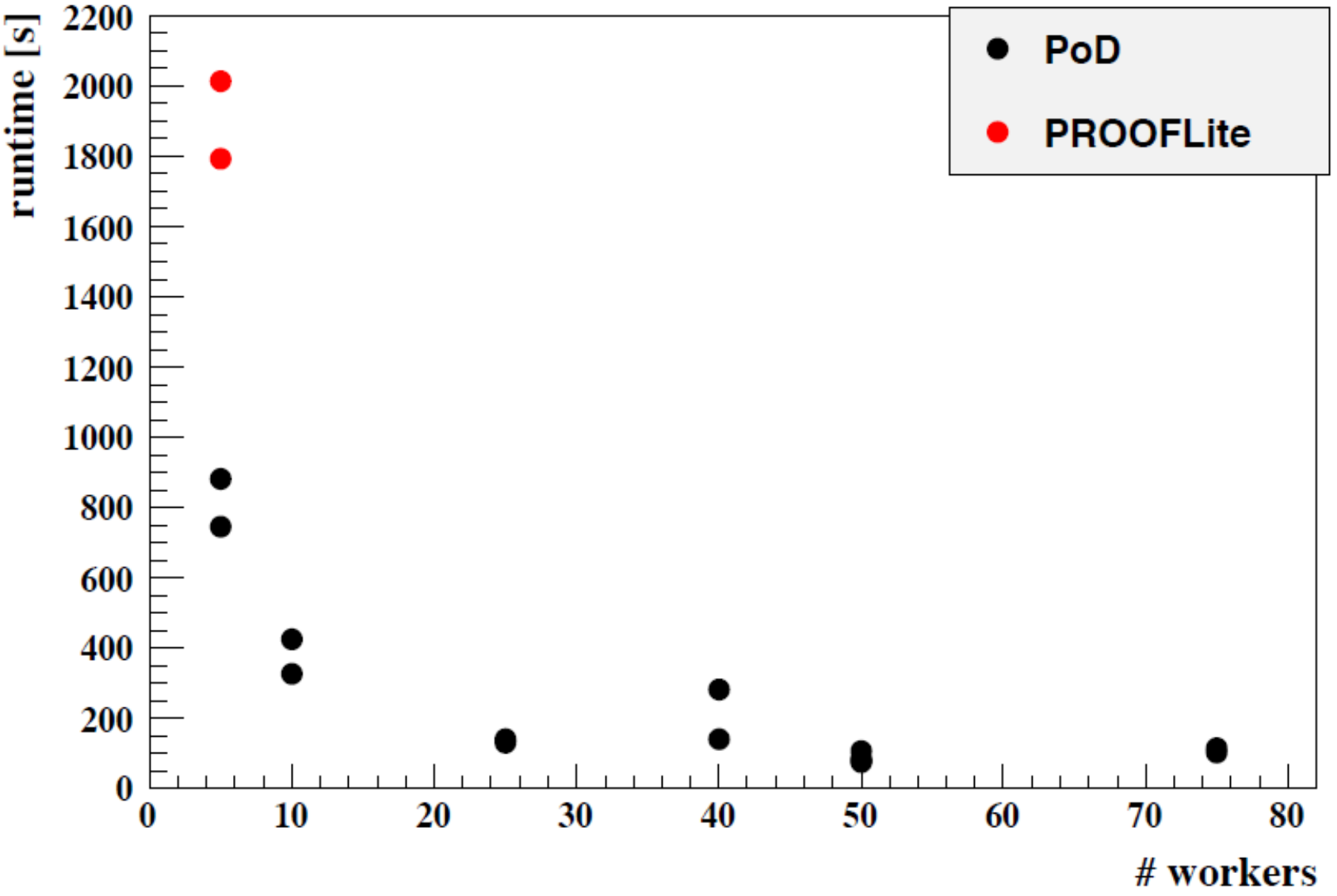
Example: $\mu\mu$ +jets

- private ROOT ntuple
- 21.2 M events
- 51 GB in 50 files (size: 300 MB - 2 GB)
- 2.4 kB/event

Tests:

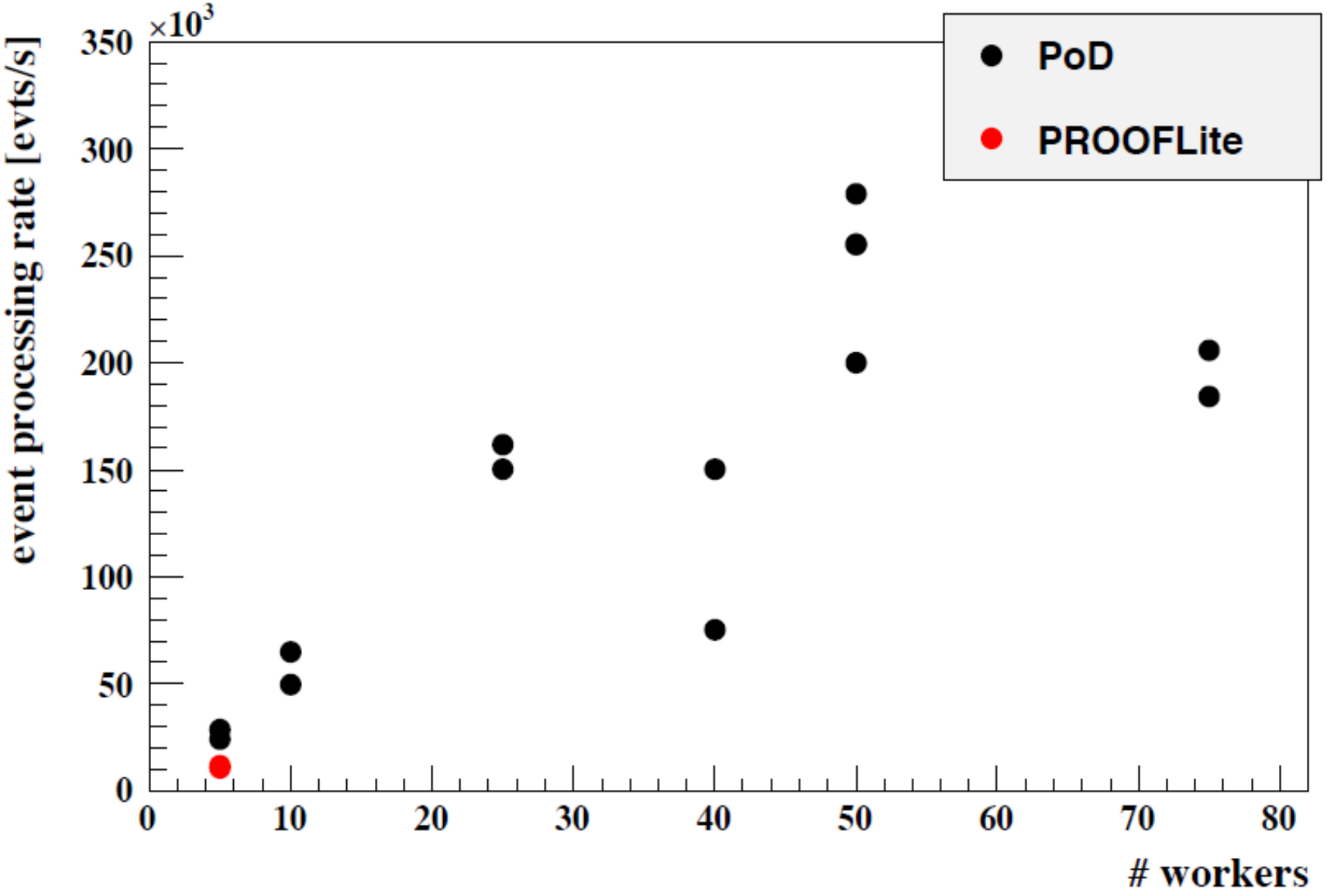
- setup PROOF cluster with PoD or PROOFLite on a CMS wgs on the NAF
- read files from NAF Lustre space (dCache via dcap/xroot slower)
- measured run time, calculated event rate and read speed (the PROOF output is unreliable (includes uncompressed size?))
- CAVEAT: results depend heavily on other users/load

Runtime


runtime [s]
2200
2000
1800
1600
1400
1200
1000
800
600
400
200
0
0
10
20
30
40
50
60
70
80
workers
PoD
PROOFLite

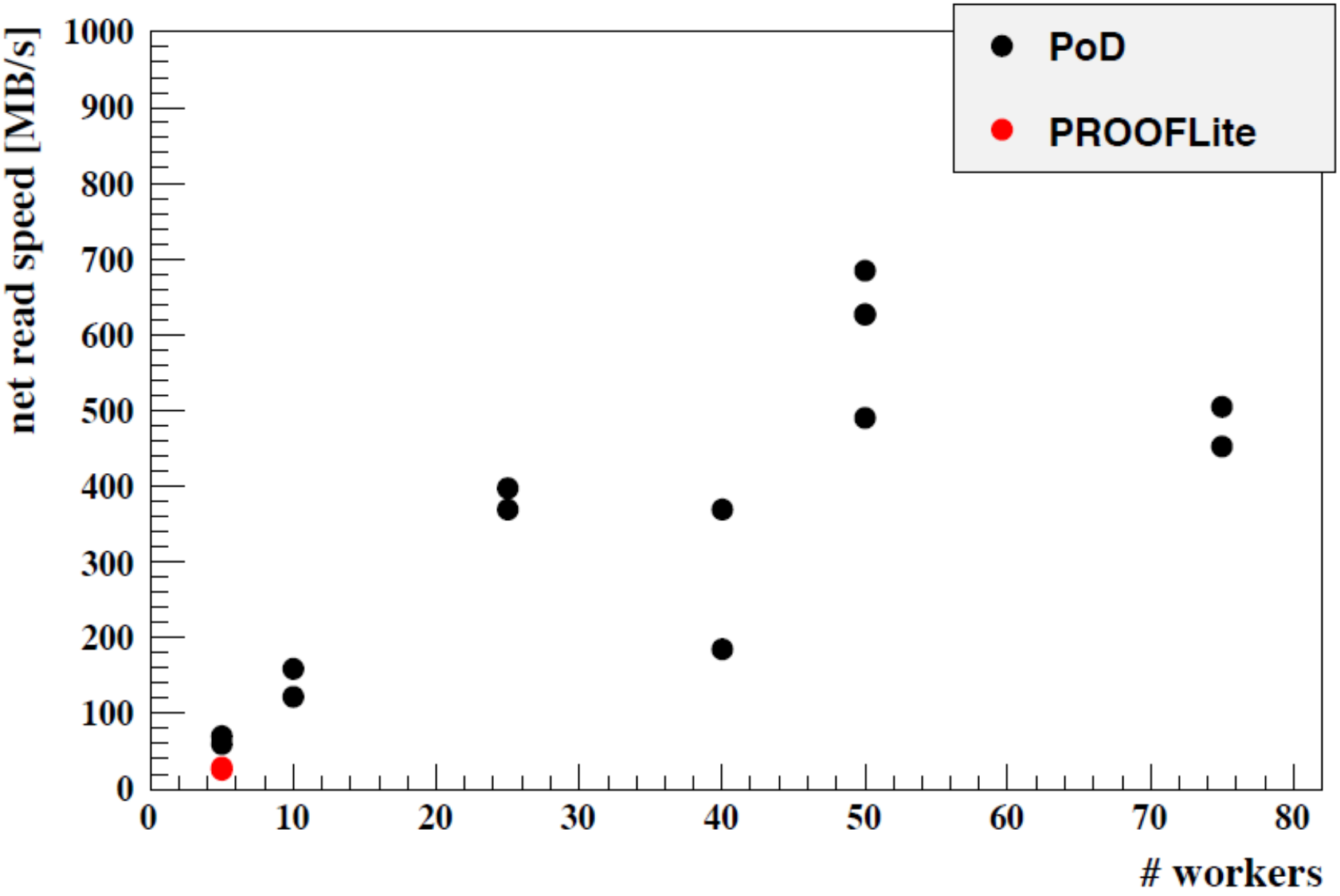
Event Processing Rate

event processing rate = number of events / runtime



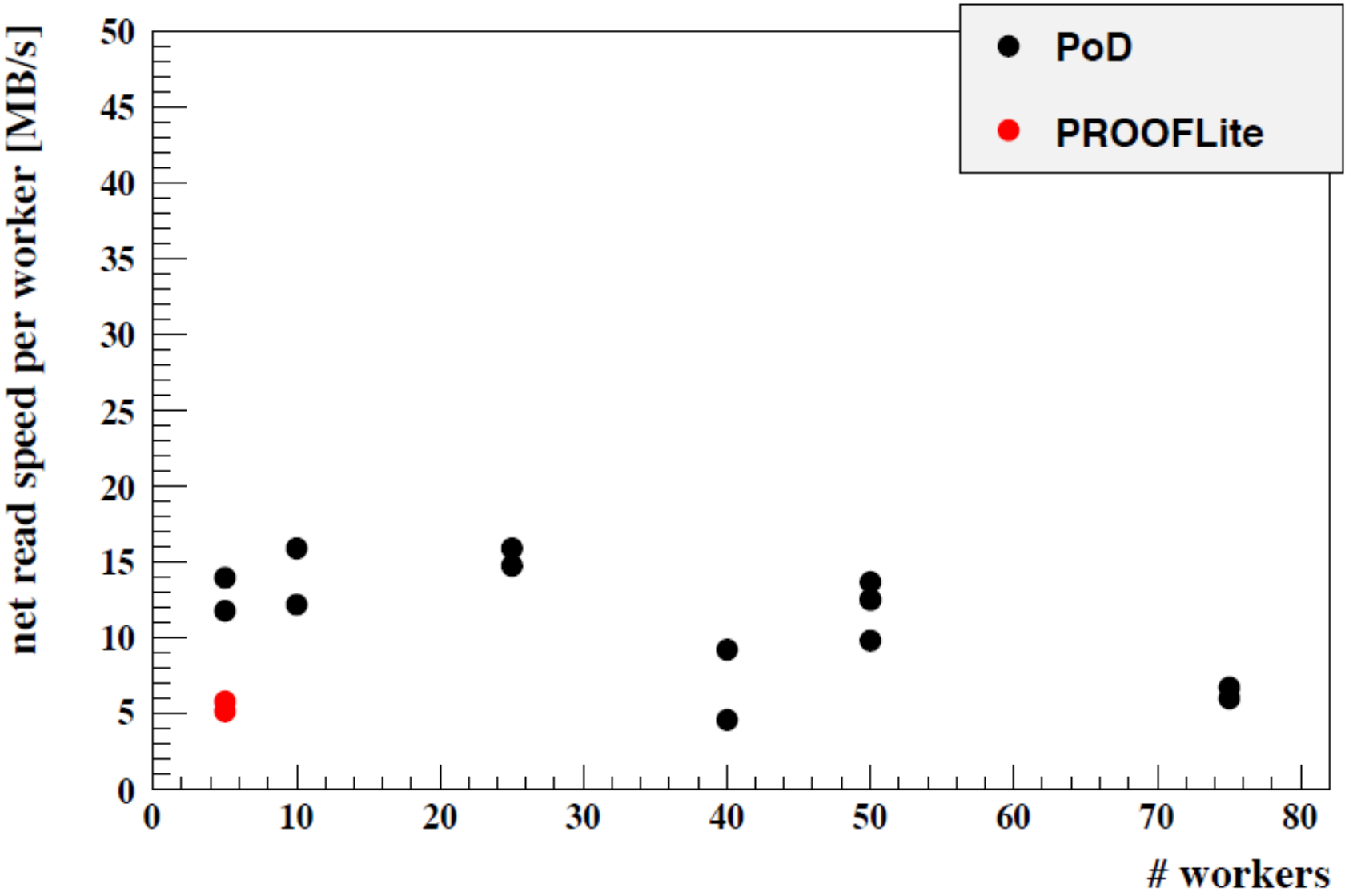
Net Read Speed

net read speed = overall size / runtime



Net Read Speed per Worker

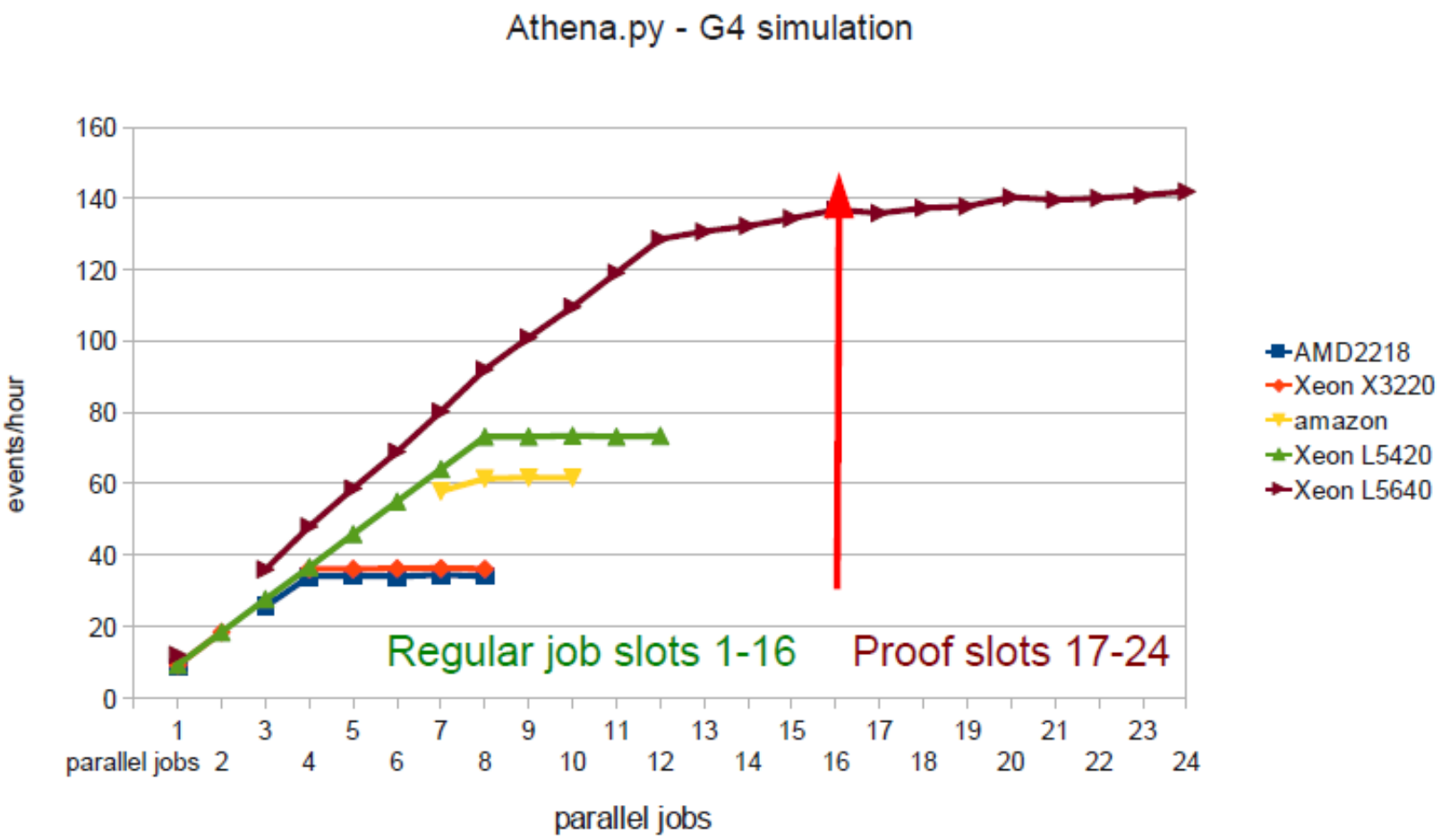
net read speed = overall size / runtime /# workers



PROOF on Demand (ATLAS/LMU)

- ProofOnDemand since ~1 year
  - Very actively used by LMU physics group
  - ca 30 sessions/day
- local cluster setup: Proof start via PoD-ssh
  - ~20 4-core desktops
  - 10 Gb link to dCache storage (LOCALGROUPDISK)
  - simple start/end scripts for users
- T2/3 cluster setup at LRZ-LMU
  - ~200 job slots reserved, PoD-SGE job-array submission
  - Important: Impose vmem limit (1.5 GB) on proof processes to not interfere with parallel running grid production/analysis jobs

Hyperthreading and Job Slots @ LRZ



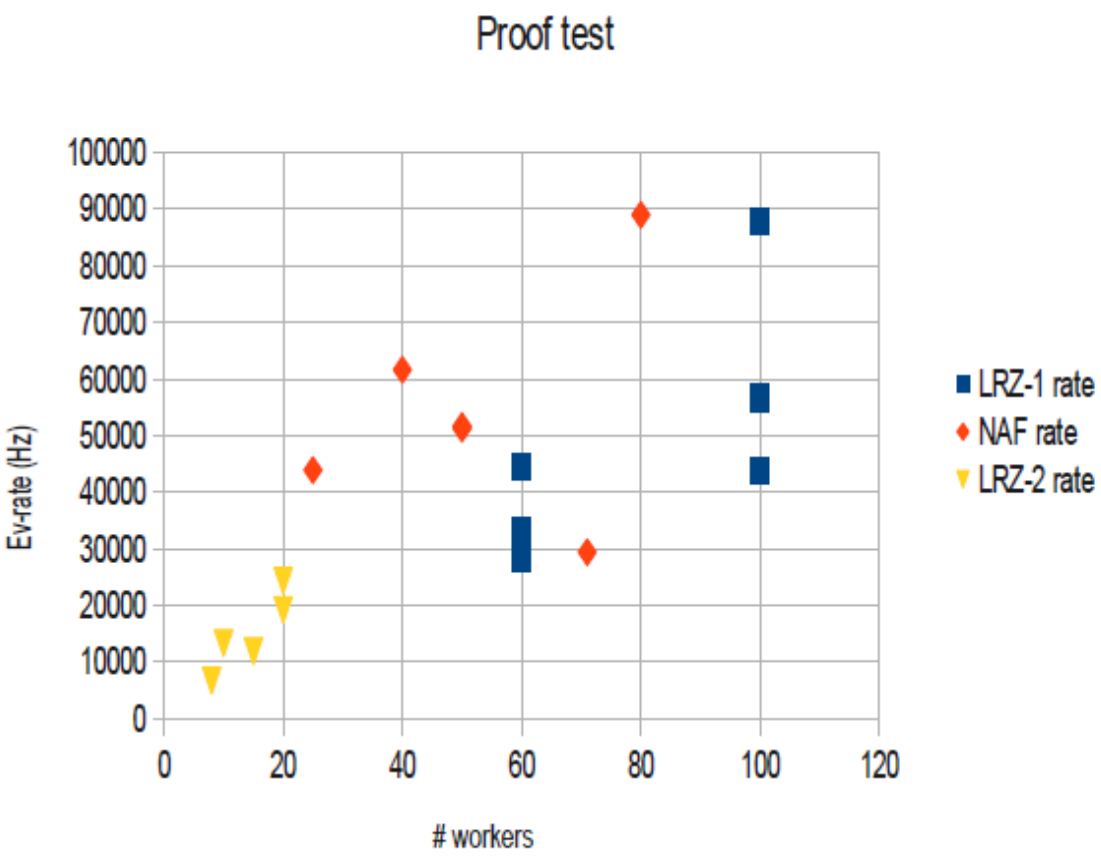
•PROOF & dCache

- We use (mostly) direct access via dcap protocol to Tier-2 dCache storage@LRZ
  - ~350 TB ATLAS controlled storage (DATADISK, GROUPDISK, …)
  - ~140 TB LMU controlled (LOCALGROUPDISK)
- Powerful ATLAS DDM tools to get ATLAS & User datasets replicated
- Interplay dCache & Proof a bit delicate
  - Excessive validation in case of standard Proof-TChain Processing loops
    - Slow start, processing overhead
    - dcache file open rather slow (~0.1 s) compared to to NFS/Lustre open
  - Much better to register as proof dataset
    - proof->RegisterDataSet(“myDS”,<File-collection>,"OV")
    - One-time validation (~0.5 s / file)
      Dataset w/ 1k files ~10 mins (if all goes well)
    - Dataset listing w/: proof->ShowDataSets()
    - many time processing: proof->Process(“myDS”,”MySelector”)

PROOF & dCache 2

- Fully automatized python script :
  - search dq2-datasets at sites,
  - convert SURL → TURL (srm://lcg-lrz-se... → dcap://lcg-lrz-dcache...)
  - register as Proof-DataSet
- All done at dataset level
  - users don't need to handle files
- Proof datasets can be shared between users – only 1 registration

Recent PROOF Tests with ATLAS Ntuple and dCache



ATLAS SM Ntuple Zmumu

- 641 Files, 6.4 M evts, 350 GB (~500 MB/file), 50 kB/evt

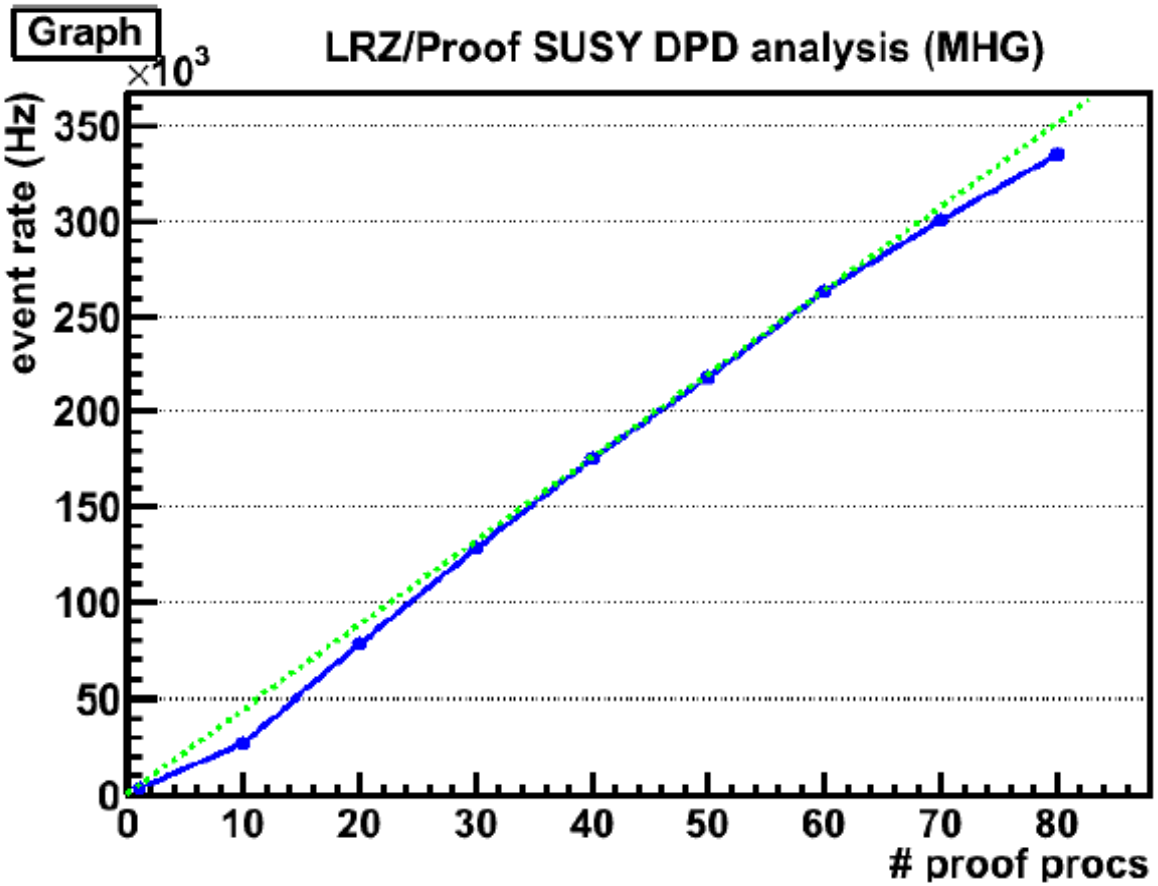
User analysis job for WW->mumu selection

- Only required branches are read → small sub-set
  - 500 MB → 10 MB (1 kB/ev)
- Root-TTreeCache (and dcap vector-read) ensure only needed stuff is transferred

In principle good scaling up to 100 slaves but big variations

- Depends on other activity
- Cold vs hot pools
- ...

Test at LRZ on Unloaded System (18 Month ago)



- Observe good scaling and fast startup

Documentation, Training etc.

- Webpages:
  - https://wiki.terascale.de/index.php/PROOFOnBatch
  - http://pod.gsi.de/
- 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), 2011 (Göttingen)

Summary and Outlock

- PROOF is one way to easily parallelise user analysis
- within the experiments PROOF is actively used
- user PROOF cluster on a batch system is a viable alternative to a global PROOF cluster
- moving from custom written tools to PoD