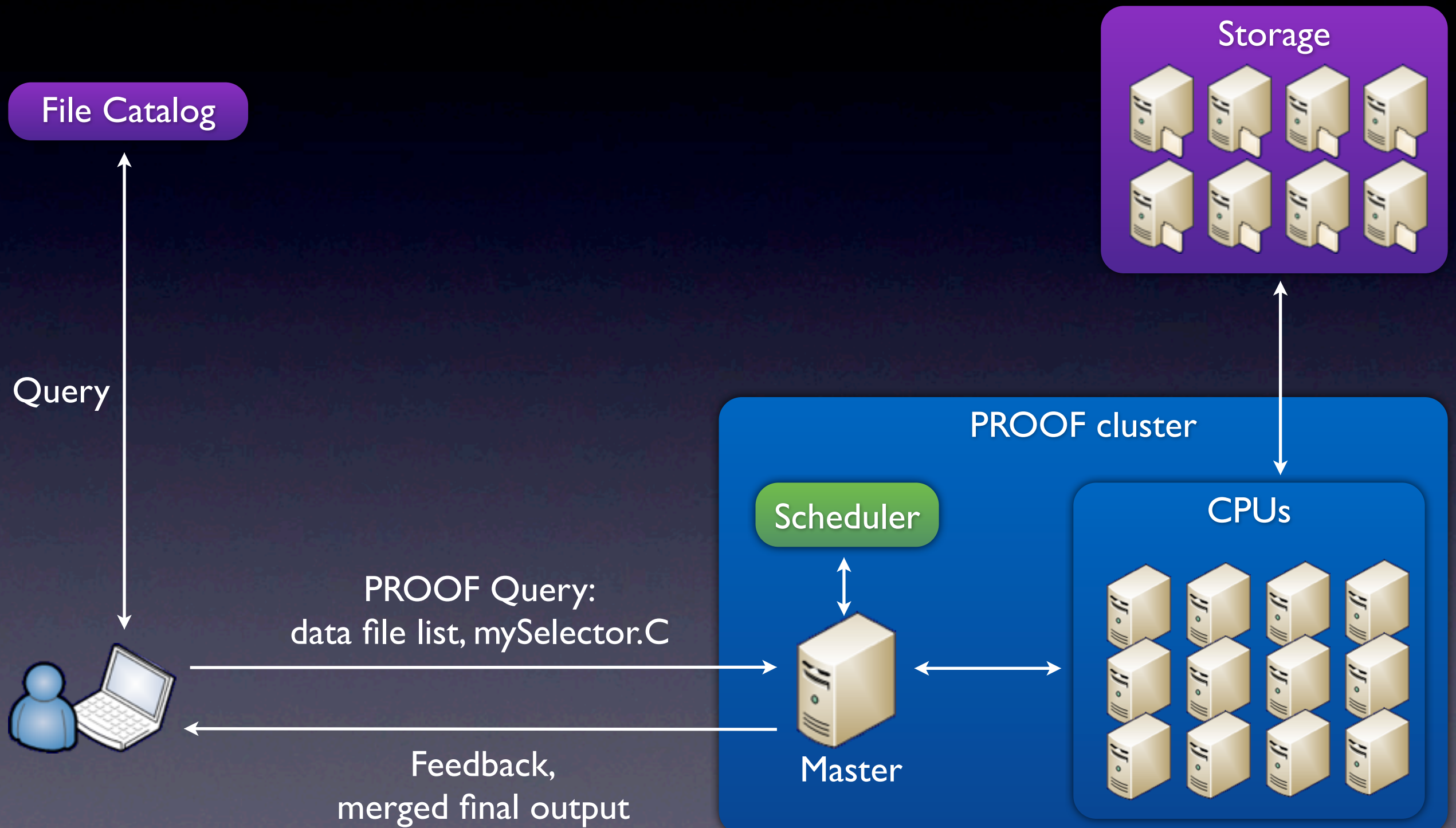


POD

PROOF on Demand

PROOF on Demand

PROOF



PROOF available via:

- standard, pre-installed PROOF cluster,
- PROOF-Lite,
- custom installers - wrapper scripts,
- PoD.

Motivation

The main idea is to have our resources shared between local jobs (“batch”), Grid jobs and PROOF.

We don't want to waste our resources and to let machines stay without a load.

The final product also should:

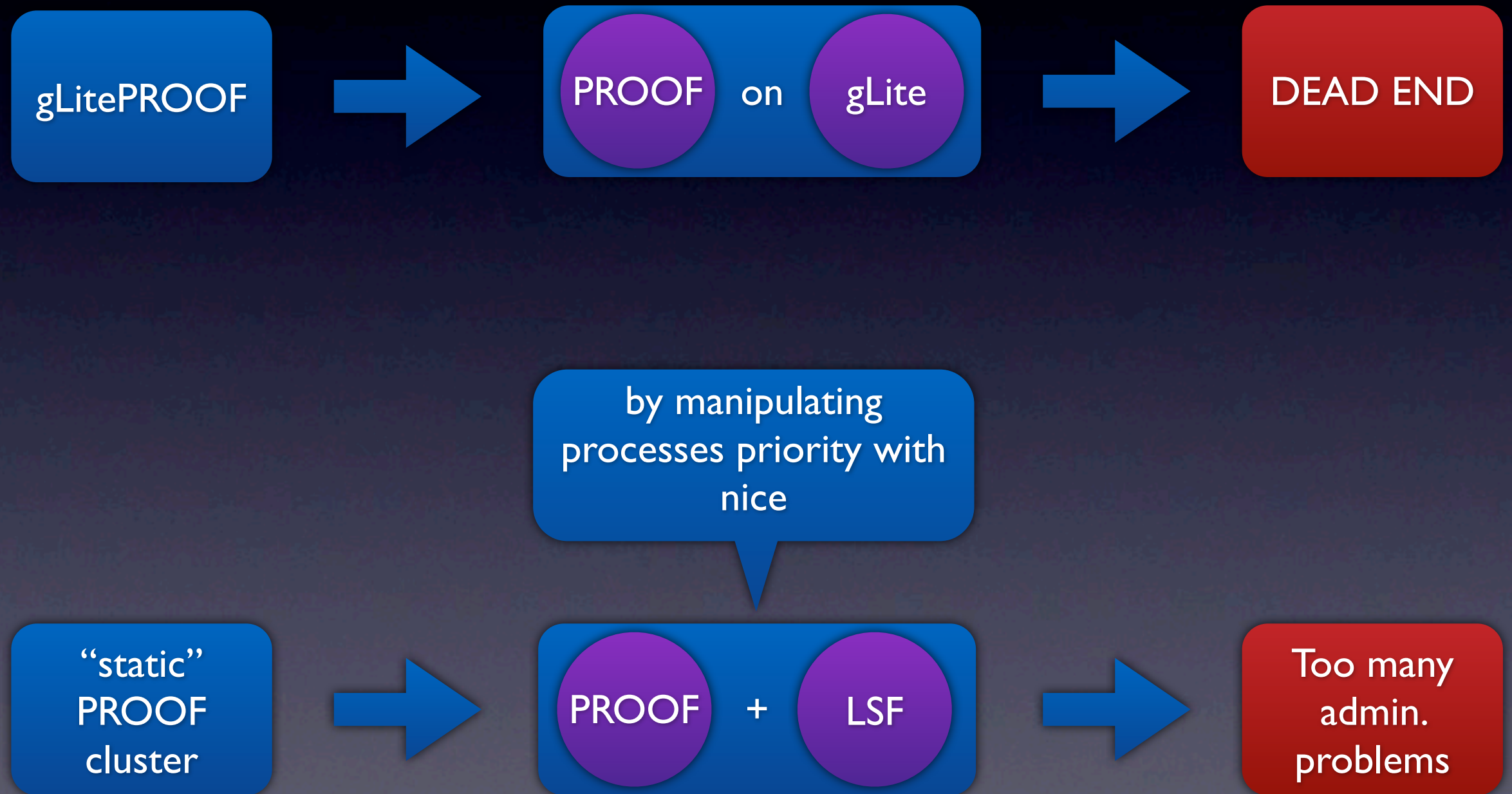
- not require administration,
- not require su privileges,
- be able to work on different RMS,
- provide automation and control of PROOF,
- be stable and private.

History

History



History



POD

PROOF on Demand

PROOF on Demand

3 steps to set your private PROOF cluster up

Start
PoD Server

3 steps to set your private PROOF cluster up



3 steps to set your private PROOF cluster up



3 steps to set your private PROOF cluster up



step #1

```
pod-server start
```

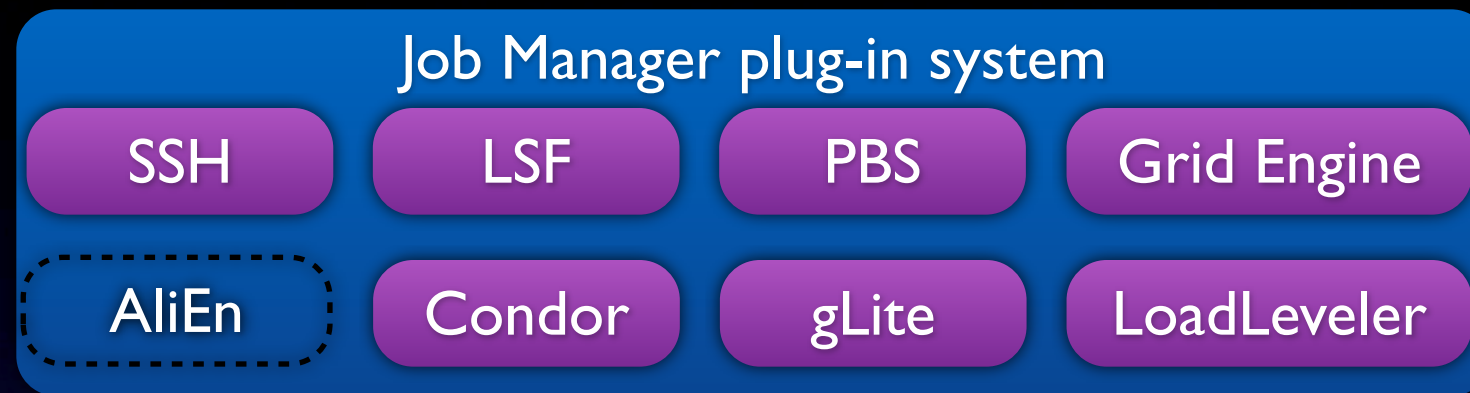
step #2

```
pod-submit -r [lsf | ge | pbs | condor ] -q my_queue -n 100  
or  
pod-ssh -c pod_ssh.cfg --submit
```

step #3

```
pod-info -n and TProof::Open("pod://")
```

Different job managers



PoD is shipped with a number of plug-ins, which cover all major RMSs, such as local cluster systems and Grid.

If you don't have any RMS, then the SSH plug-in can be used.

The SSH plug-in is also used to setup PROOF clusters on Clouds.

PoD vs “Static” approach

a User

- can entirely control his/her dynamic cluster,
- can setup and use it on demand,
- can dynamically change an amount of workers,
- can select a preferable master host,
- doesn't need admins to take an action,
- doesn't disturb other users,
- is free to choose a ROOT version for services.

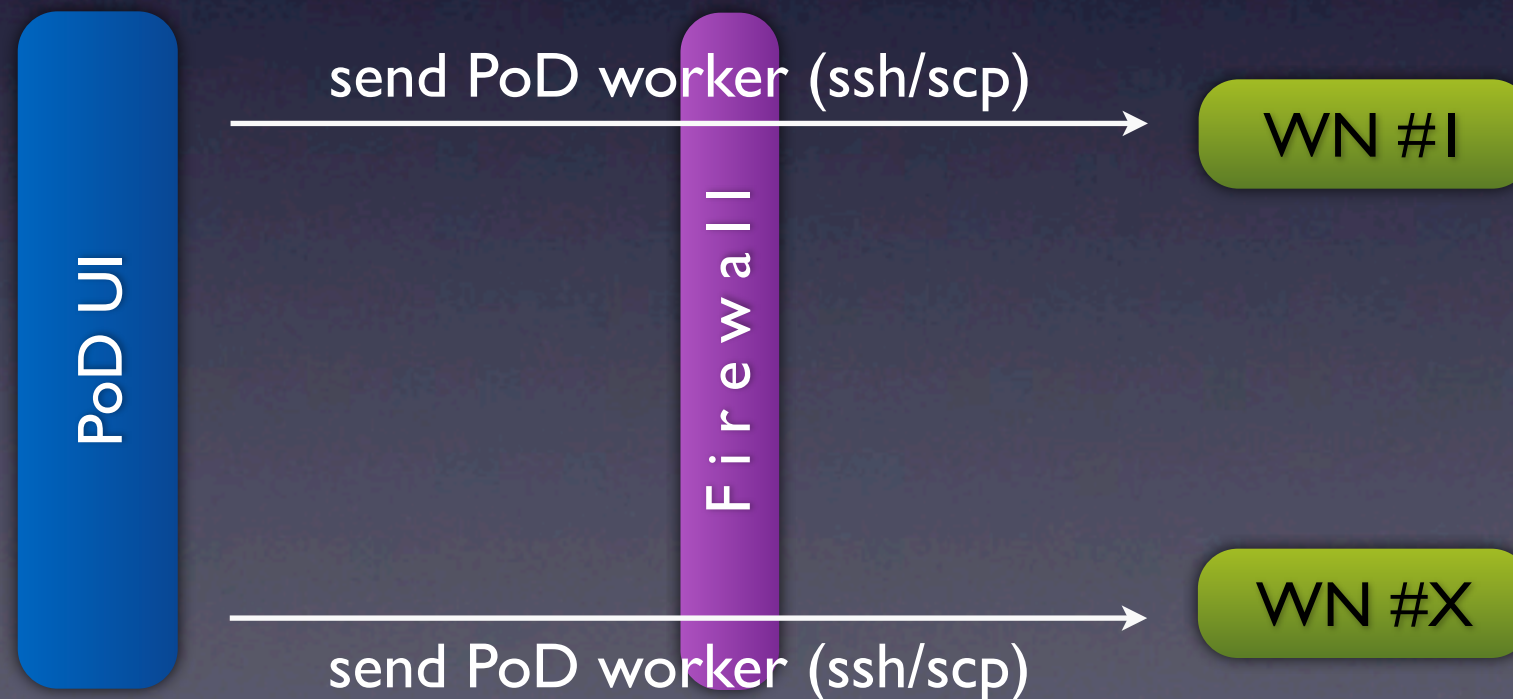
PoD supports:

- Mac OS X and Linux,
- local and central installations,
- direct and packet-forwarded connections.

SSH plug-in

a simple CSV file as an input to the plug-in:

| <i>id</i> | <i>, login@host</i> | <i>, ssh_params</i> | <i>, wn_dir</i> | <i>, num_of_workers</i> |
|-----------|----------------------|---------------------|-----------------|-------------------------|
| r1 | , anar@lxcg27.gsi.de | , -p24 | , /tmp/test | , 10 |
| a2 | , user@lxi001 | , | , ~/pod_wn | , 8 |
| 125 | , doom@host.my | , -p22 | , /opt/pod | , 16 |



SSH plug-in & cloud

Possible workflow:

- distribute data files,
- prepare an OS image, which includes PoD's worker package (made by pod-prep-worker),
- send the image to a cloud provider, requesting an ssh access to nodes,
- pass the list of workers to PoD ssh plug-in.

Enjoy your cloud based PROOF cluster.

PoD: a “Green-PC” concept

The PoD SSH plug-in helps to consume workstations around you.

PoD could be perfectly combined with CernVM, for example, to let users intentionally share cores of their workstations.

pod-remote: a thin client concept



pod-remote

- helps to control remote PoD instances,
- uses only secured channels for communications,
- automatically creates and maintains SSH tunnels for PROOF and PoD connections.

Data location

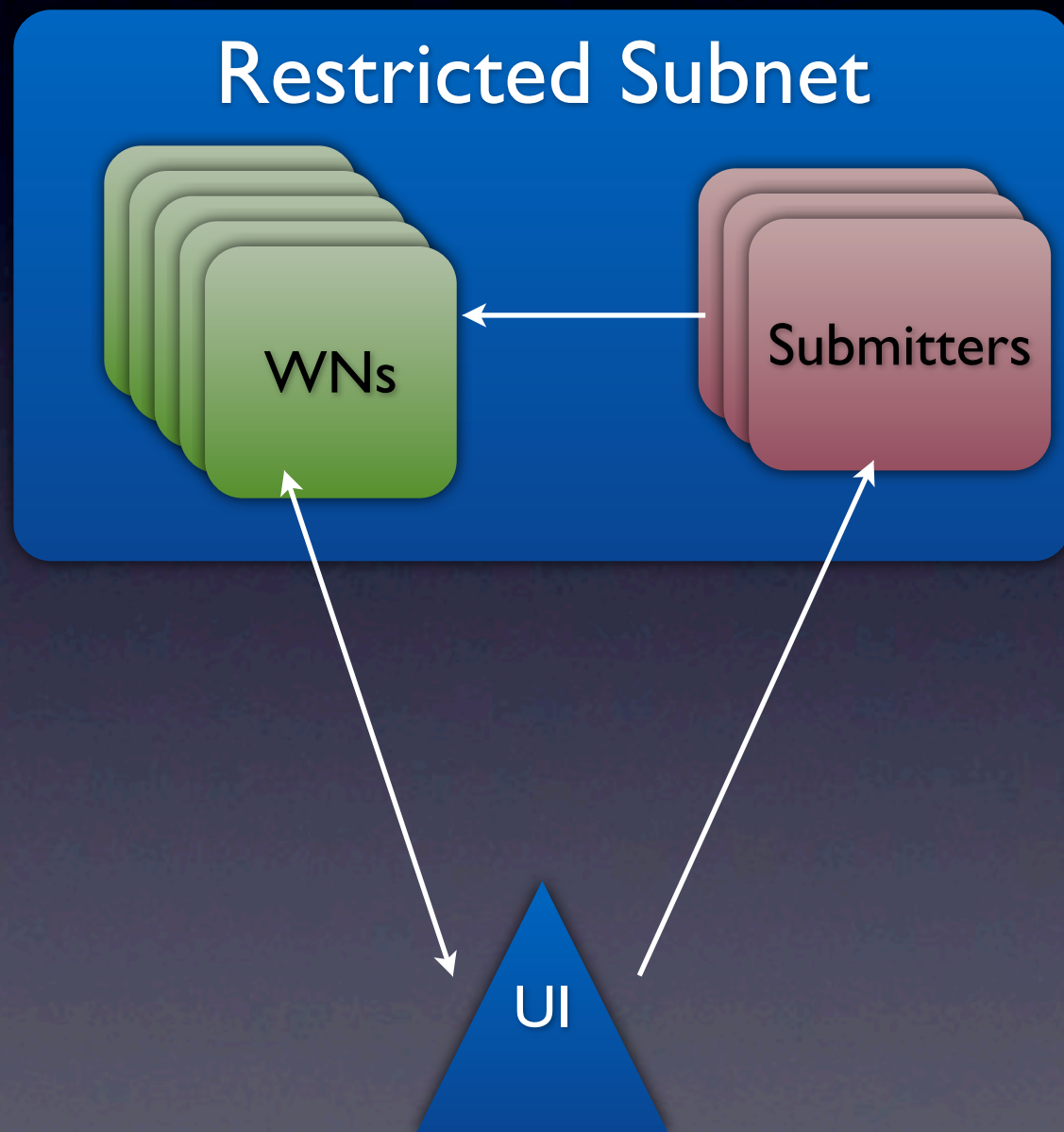
For the dynamic analysis model is the best to keep data remote from WNs, but local for the site.

Interesting study:

ISGC2011, "Investigation of storage options for scientific computing on Grid and Cloud facilities" (Fermi National Accelerator Laboratory, US)

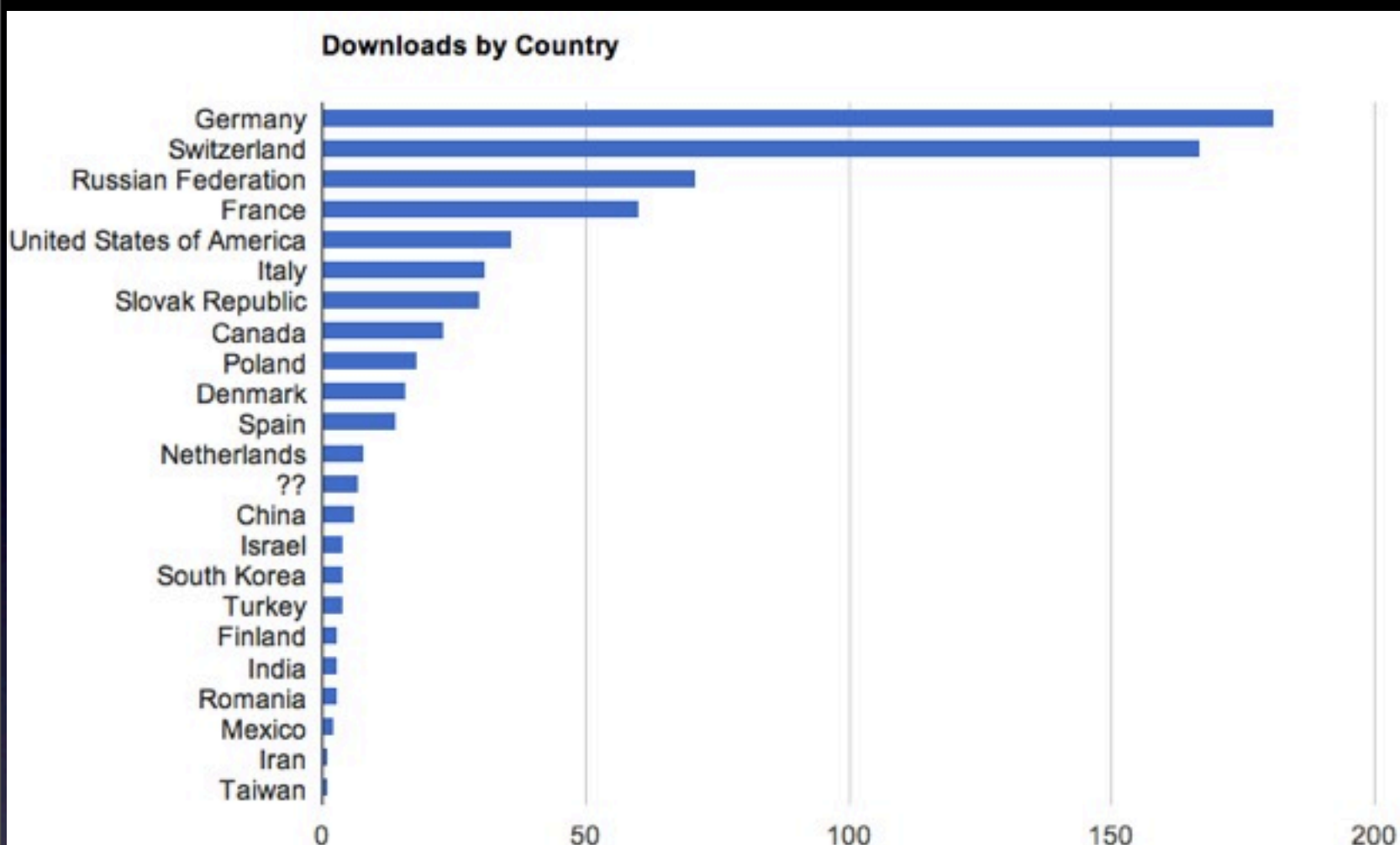
<http://www2.twgrid.org/event/isgc2011/slides/GridsandClouds/2/Storage-Evaluation-ISGC-2011.pdf>

New PoD development: Dynamic Master architecture



- use pod-remote to submit PoD jobs,
- all PoD WNs connect back to PoD UI,
- PoD UI dynamically assigns a suitable WN to be a PoD server and a PROOF master,
- all WNs connect to that dynamic PROOF master,
- PoD automatically creates its packet-forwarding tunnel to redirect PROOF traffic between UI and the Master.

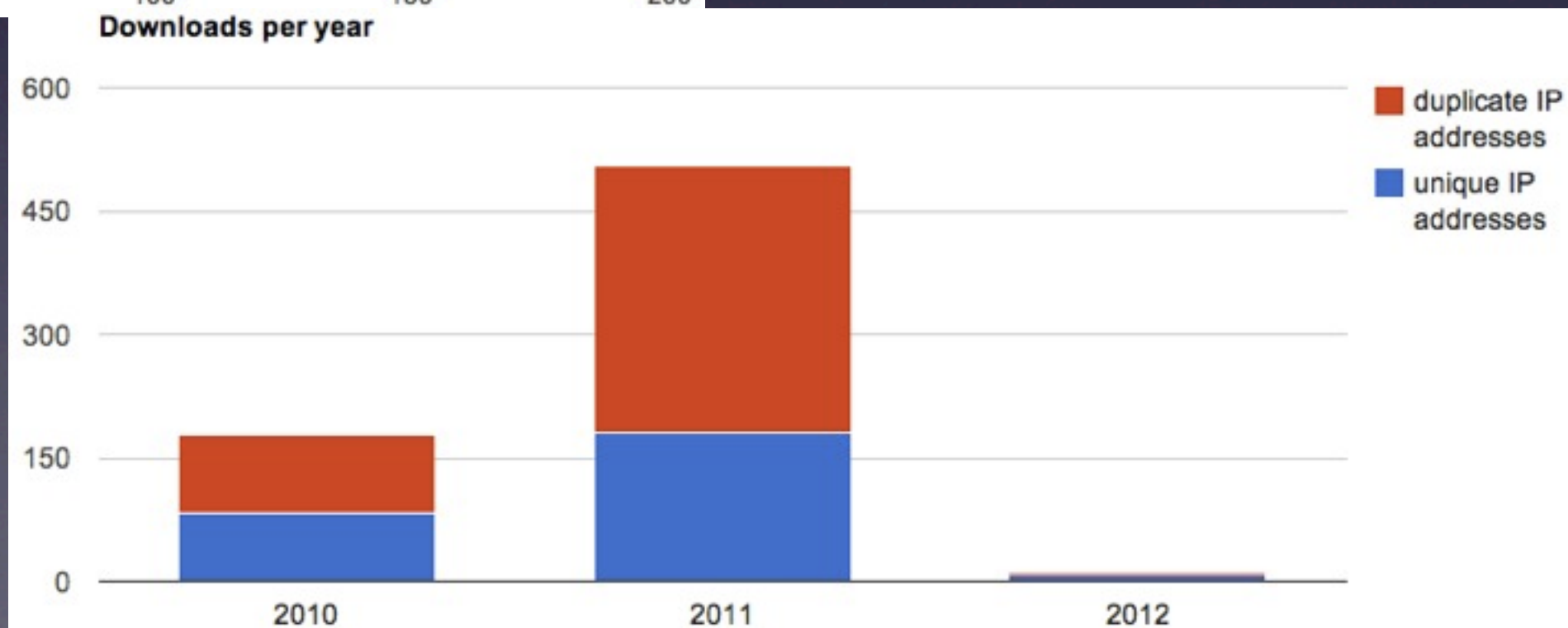
User community



~700 Downloads from 22 countries, since June 2010

ALICE, ATLAS, CMS sites + many individual users and small groups

http://pod.gsi.de/download_stat.html



backup slides

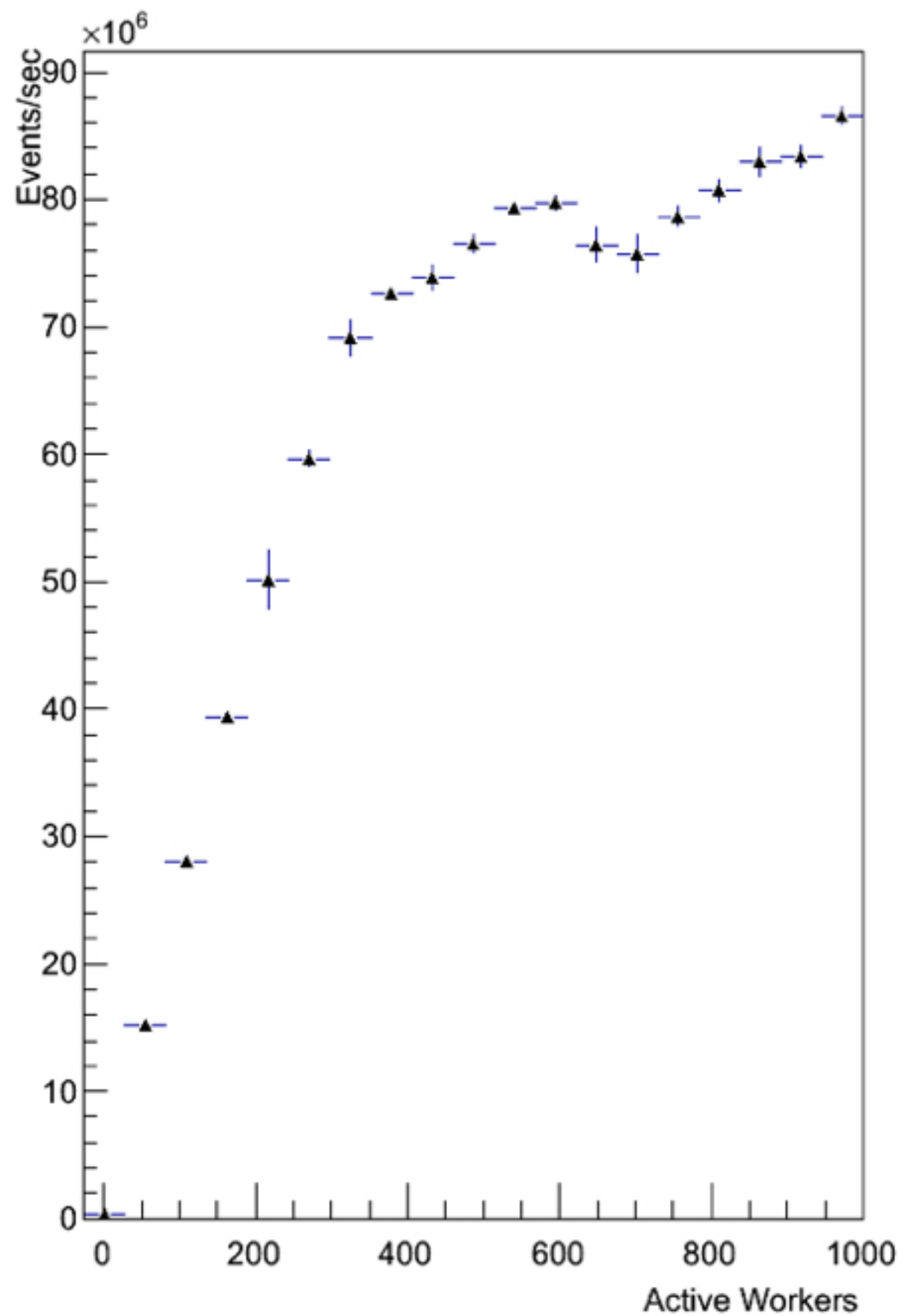
Fitness

These kind of exercises you can't easily make without PoD!

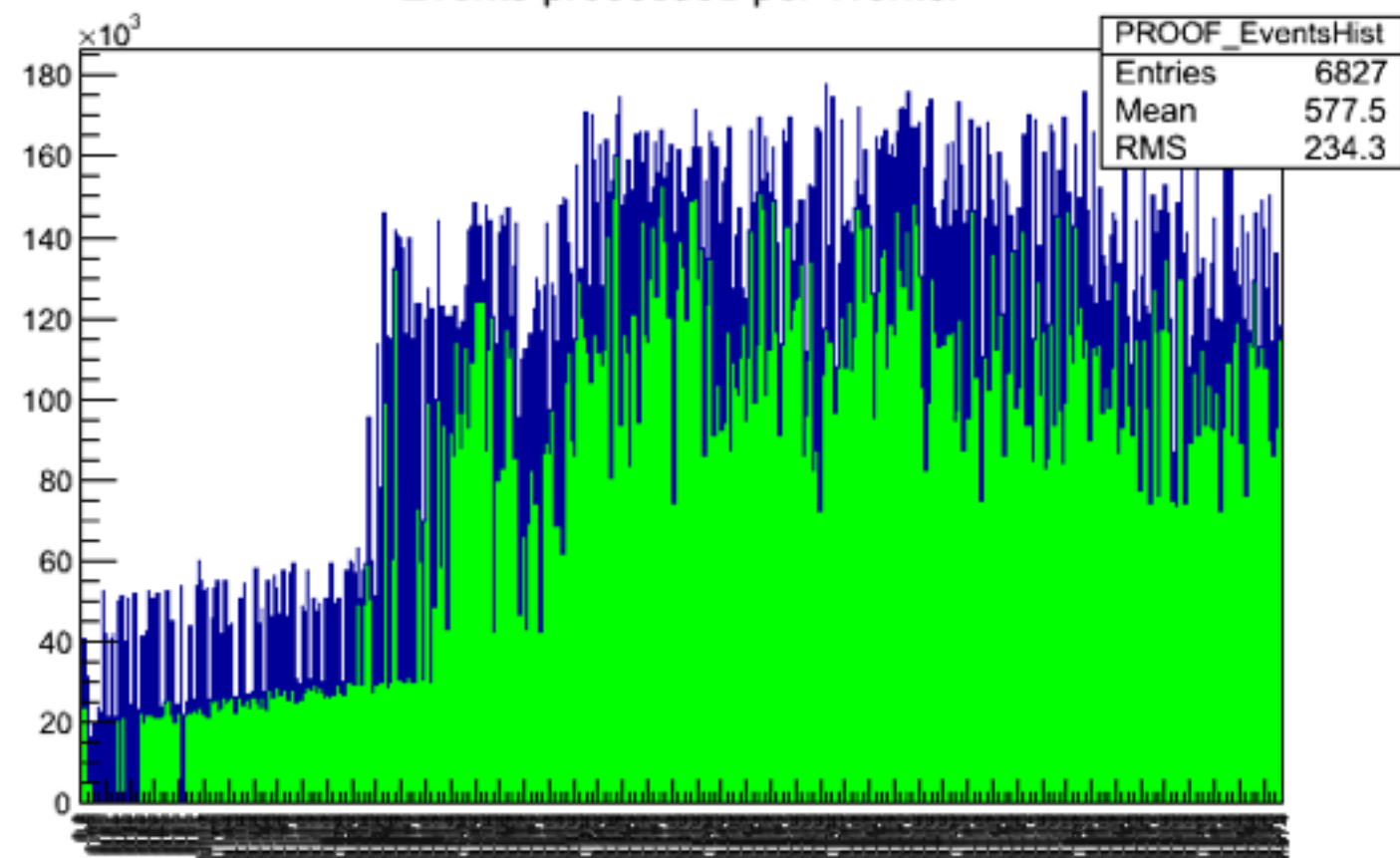
PoD has been used to setup a dynamic PROOF cluster of 1000 WNs on Cloud.

For tests we use
the new PROOF Benchmark Suite.

Profile CPU QueryResult Event - TSelHist_Hist1D



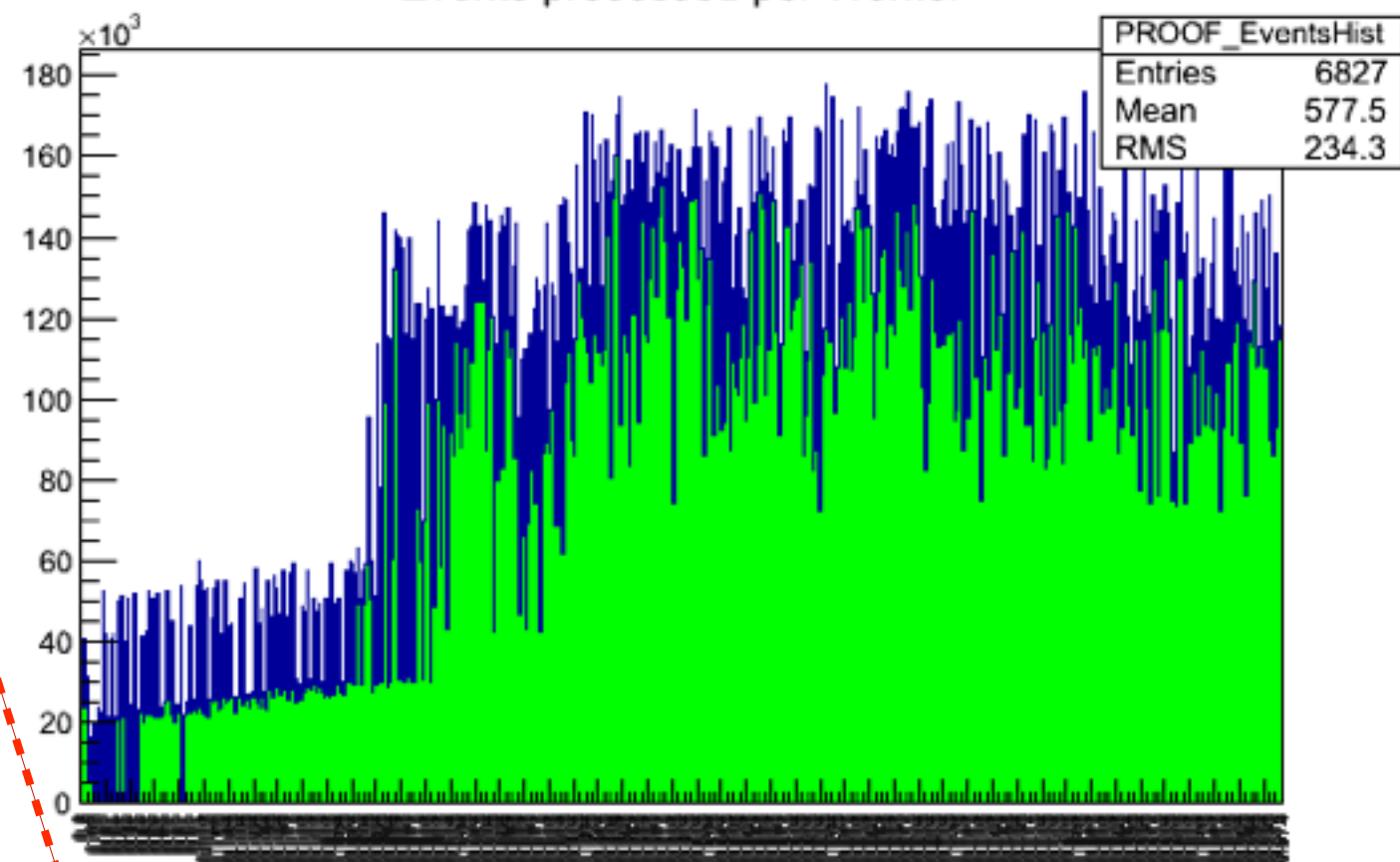
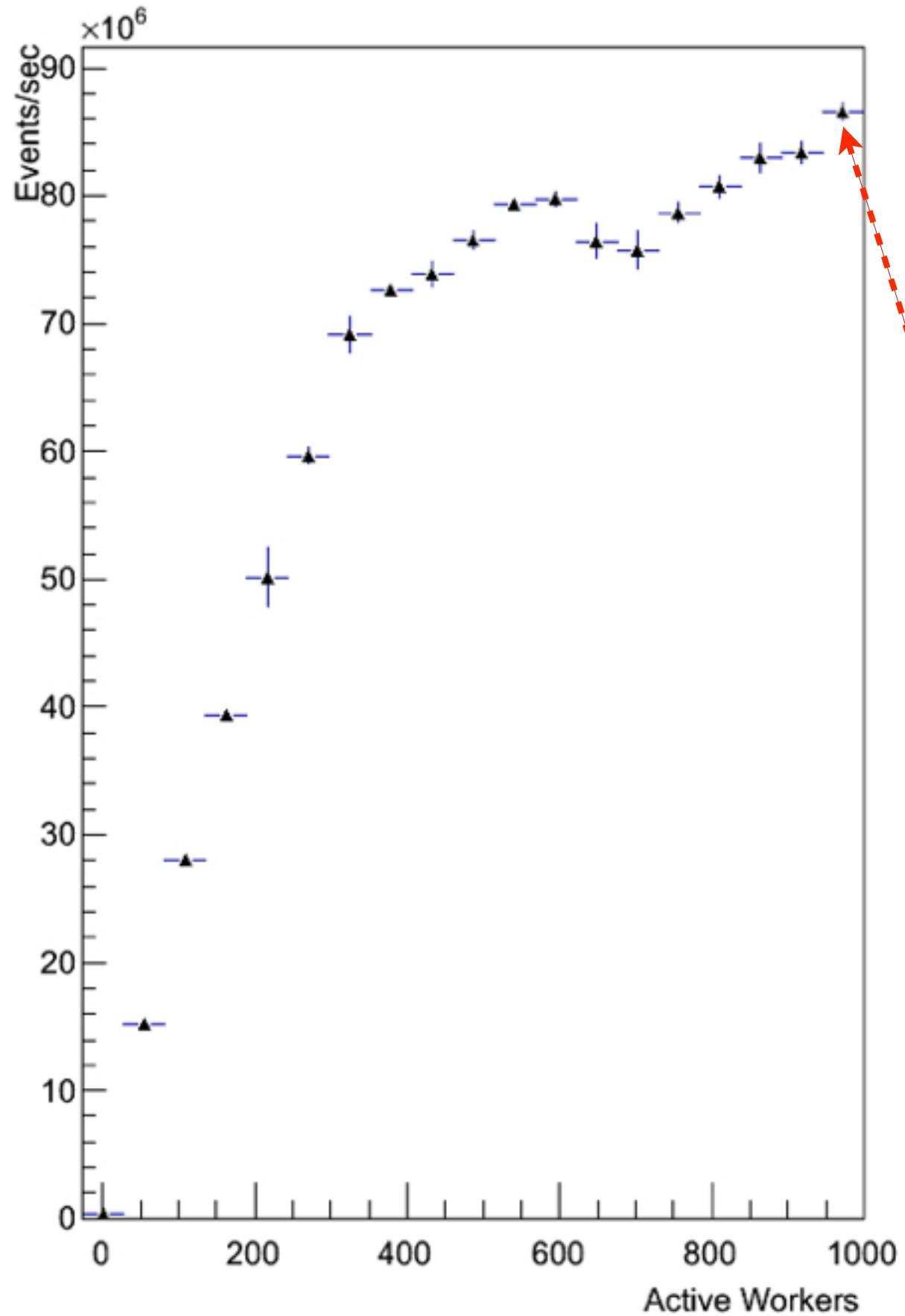
Events processed per Worker



Profile CPU QueryResult Event - TSelHist_Hist1D

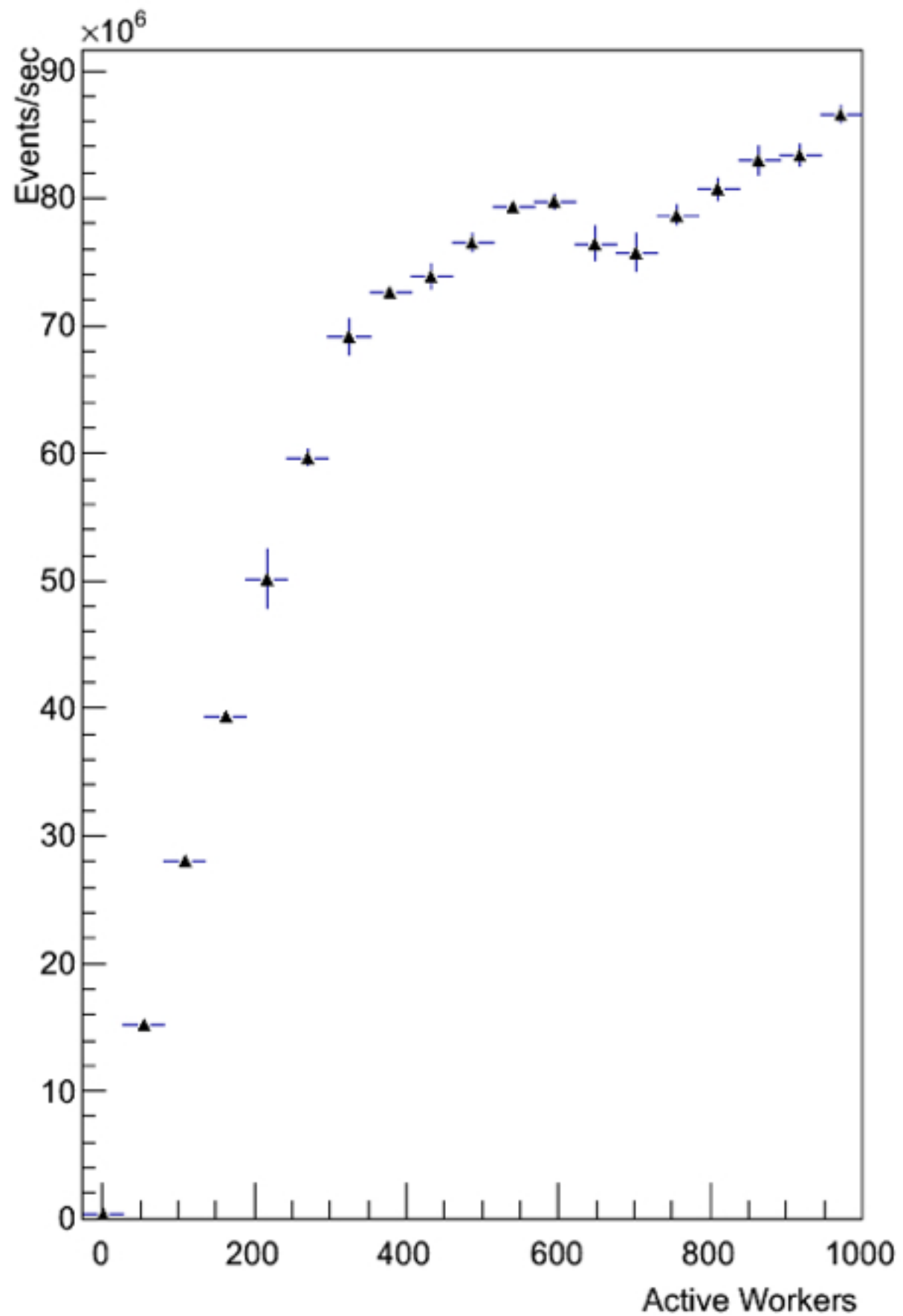
Events processed per Worker

| PROOF_EventsHist | |
|------------------|-------|
| Entries | 6827 |
| Mean | 577.5 |
| RMS | 234.3 |

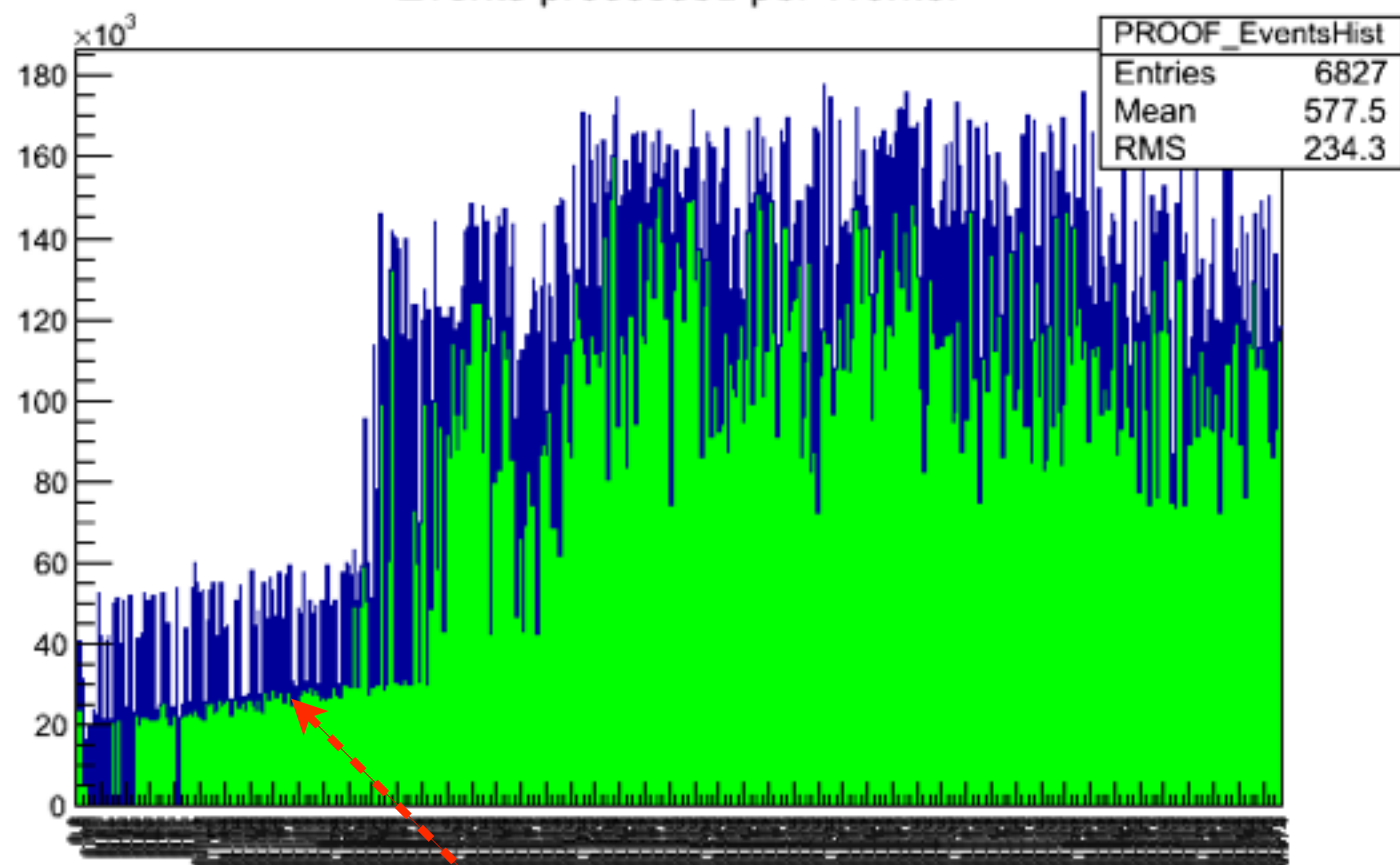


975 WNs/Master is the limit

Profile CPU QueryResult Event - TSelHist_Hist1D

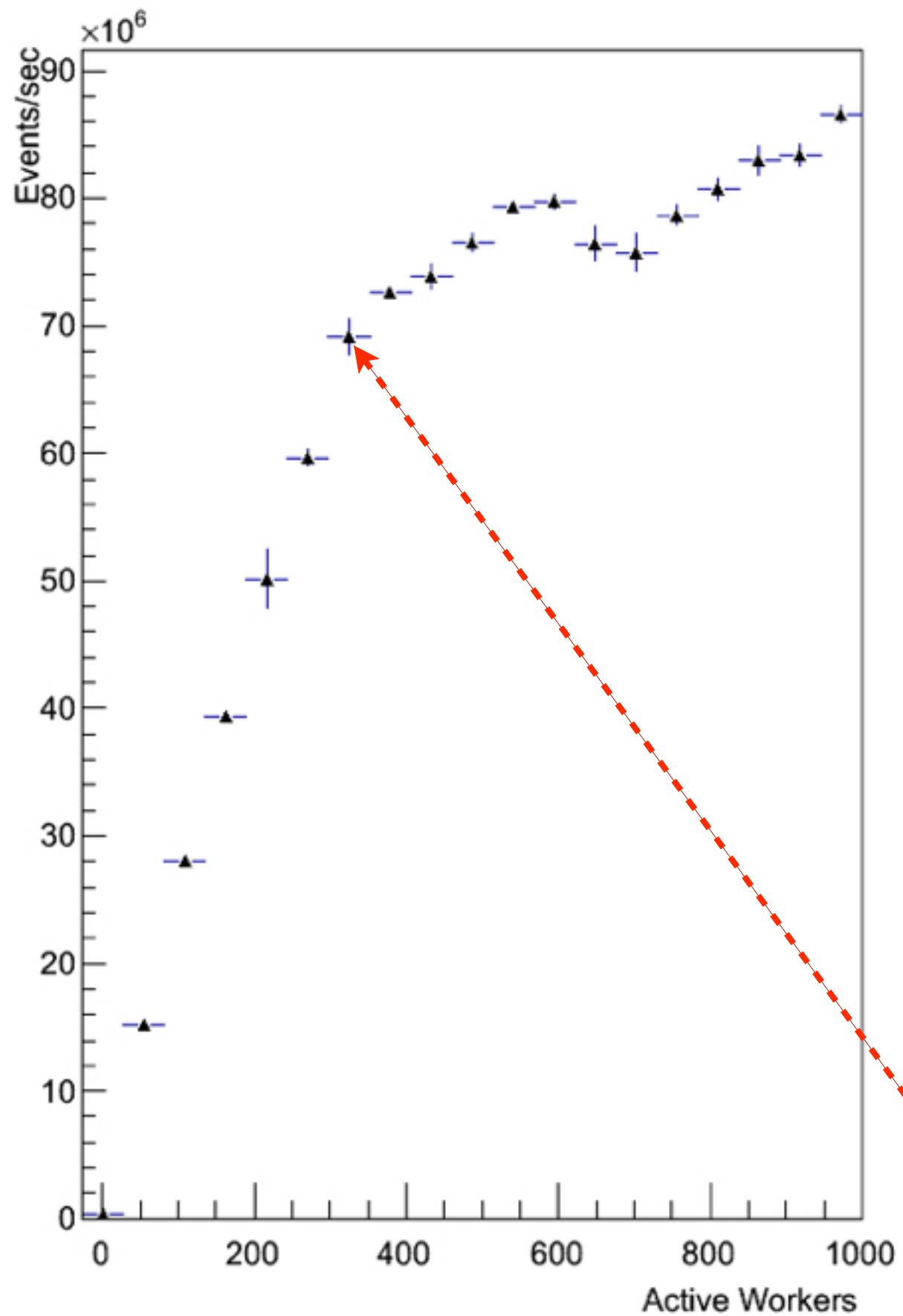


Events processed per Worker

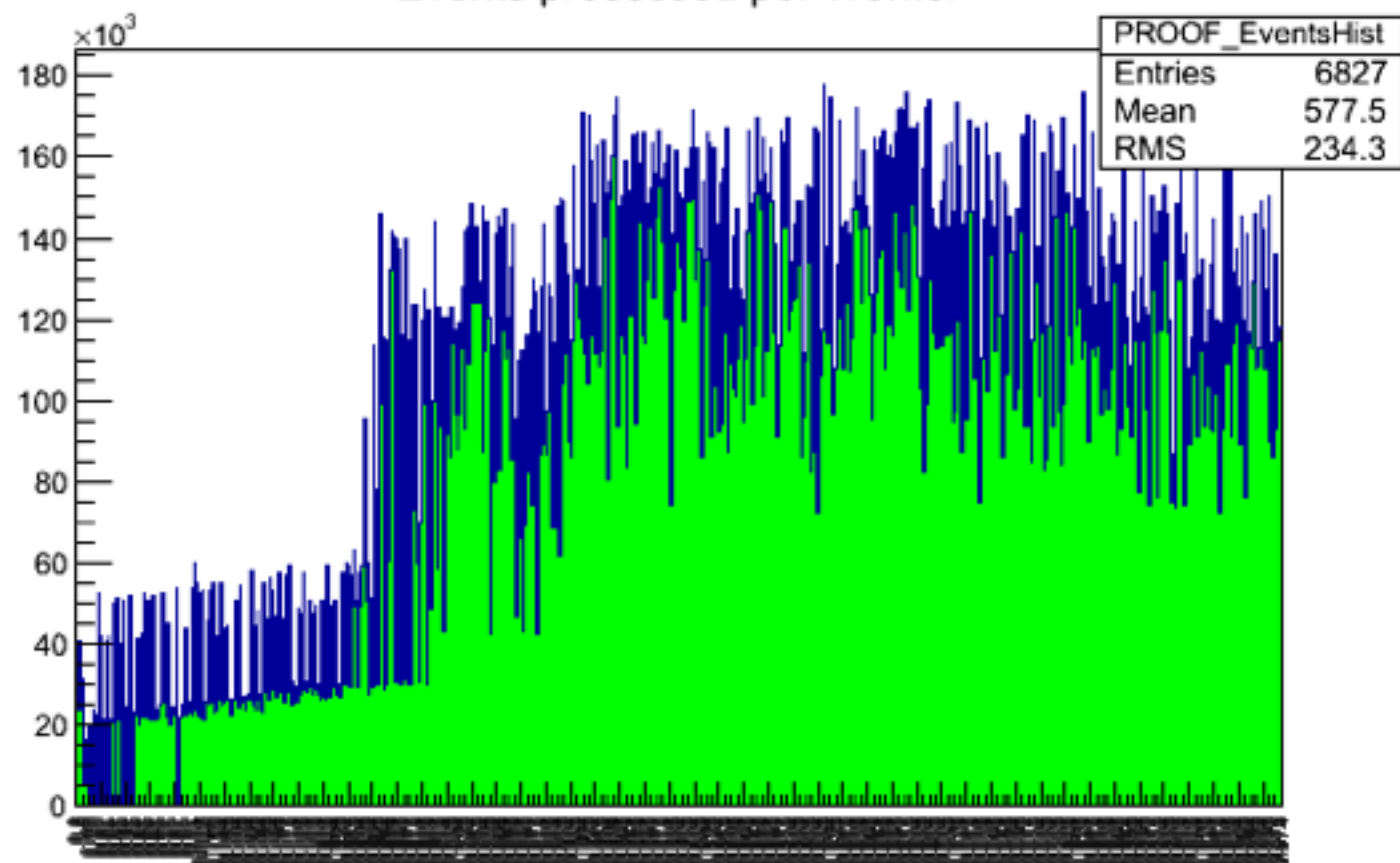


Slow single-threaded packetizer:
some workers are idling

Profile CPU QueryResult Event - TSelHist_Hist1D

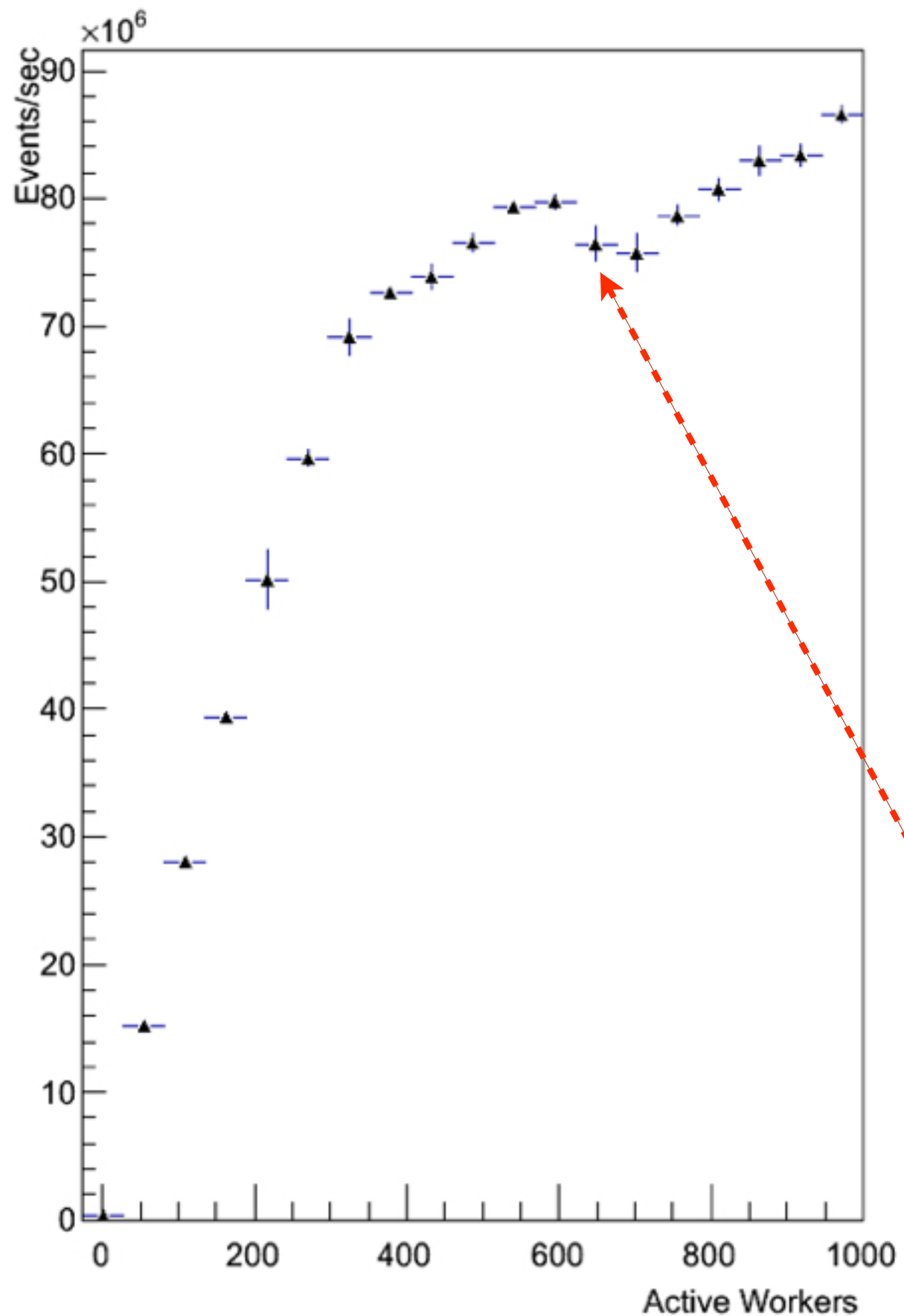


Events processed per Worker

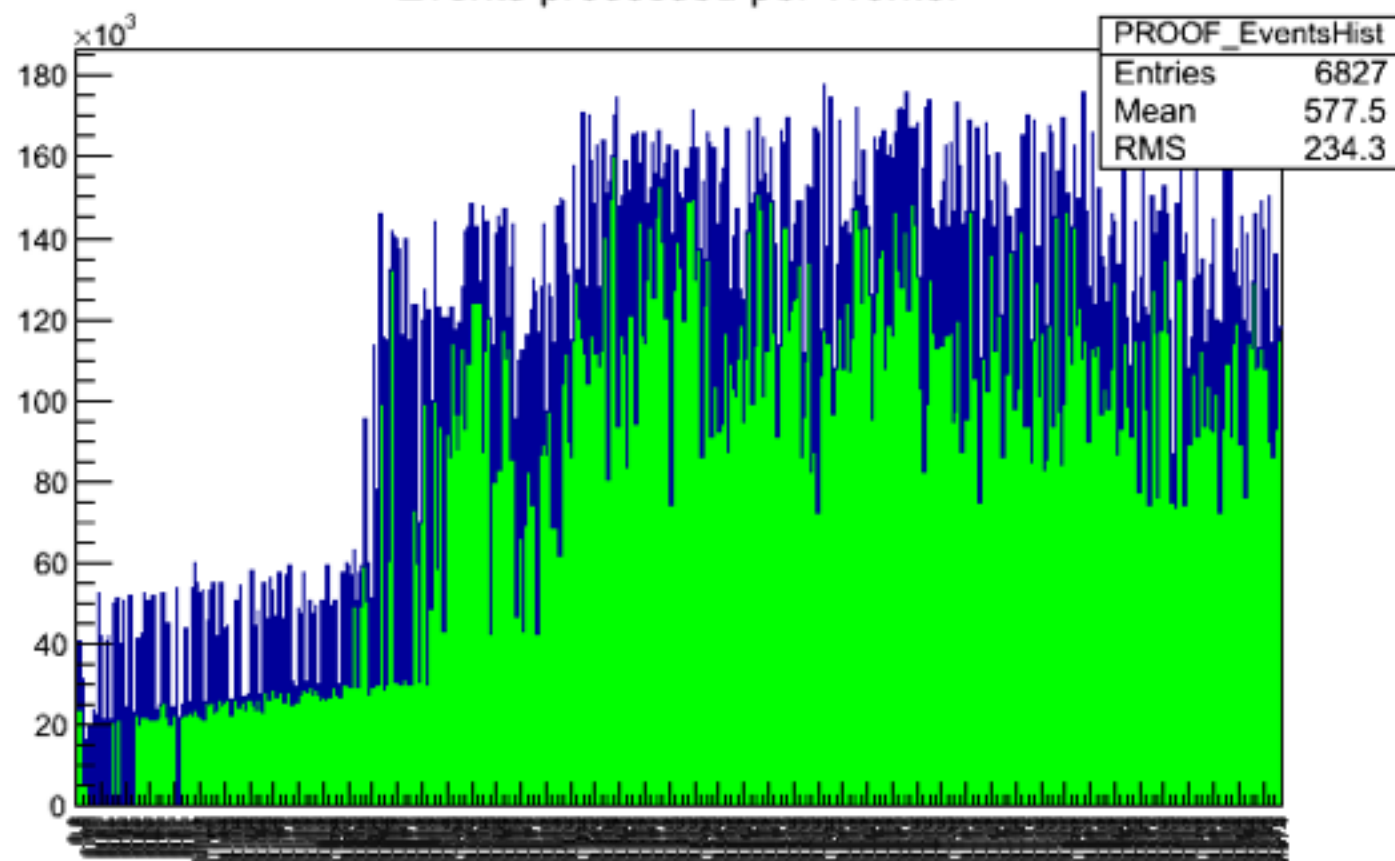


~350WNs per Master is the efficiency border

Profile CPU QueryResult Event - TSelfHist_Hist1D



Events processed per Worker



What is this? Is it a “Slow Packetizer” effect?
Perfectly reproducible, when going 500+ WNs...

Fitness: conclusions

PoD:

- is a scalable and a perfect tool for PROOF on the Cloud,
- more optimization is required to distribute WNs faster via SSH (when going over 600+ WNs).

PROOF:

- ~350WNs per Master is the efficiency border,
- a single-threaded PROOF packetizer is a bottleneck,
 - a new TThreadPool is in development. Can that help?
- should not be limited by only ~975 WNs per Master
 - can be solved by changing select() call to poll().