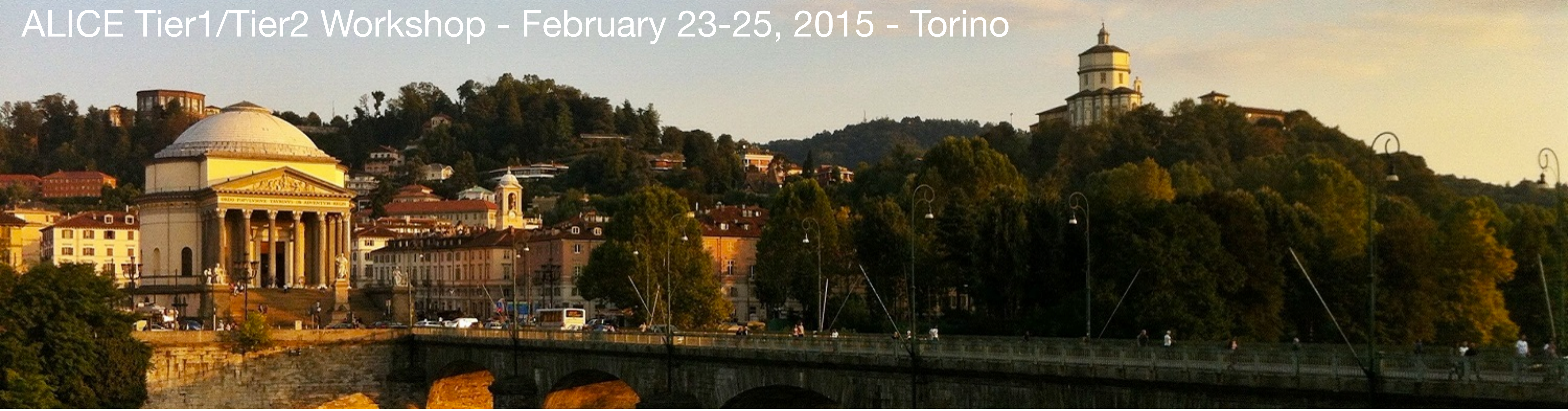


Outcomes from the PRIN STOA-LHC project
(inherited from HHLR-GU)

Speaker: Sara Vallero (Università degli Studi di Torino and INFN)
Dario Berzano
Grazia Luparello
Giorgia Miniello
Massimo Venaruzzo

INFN Istituto Nazionale di Fisica Nucleare

The PRIN STOA-LHC project

Scientific research program of relevant national interest

Development of computing technologies for the optimisation of access to LHC data and for the technology transfer towards other research areas using the grid and cloud computing approach.

Main Objectives:

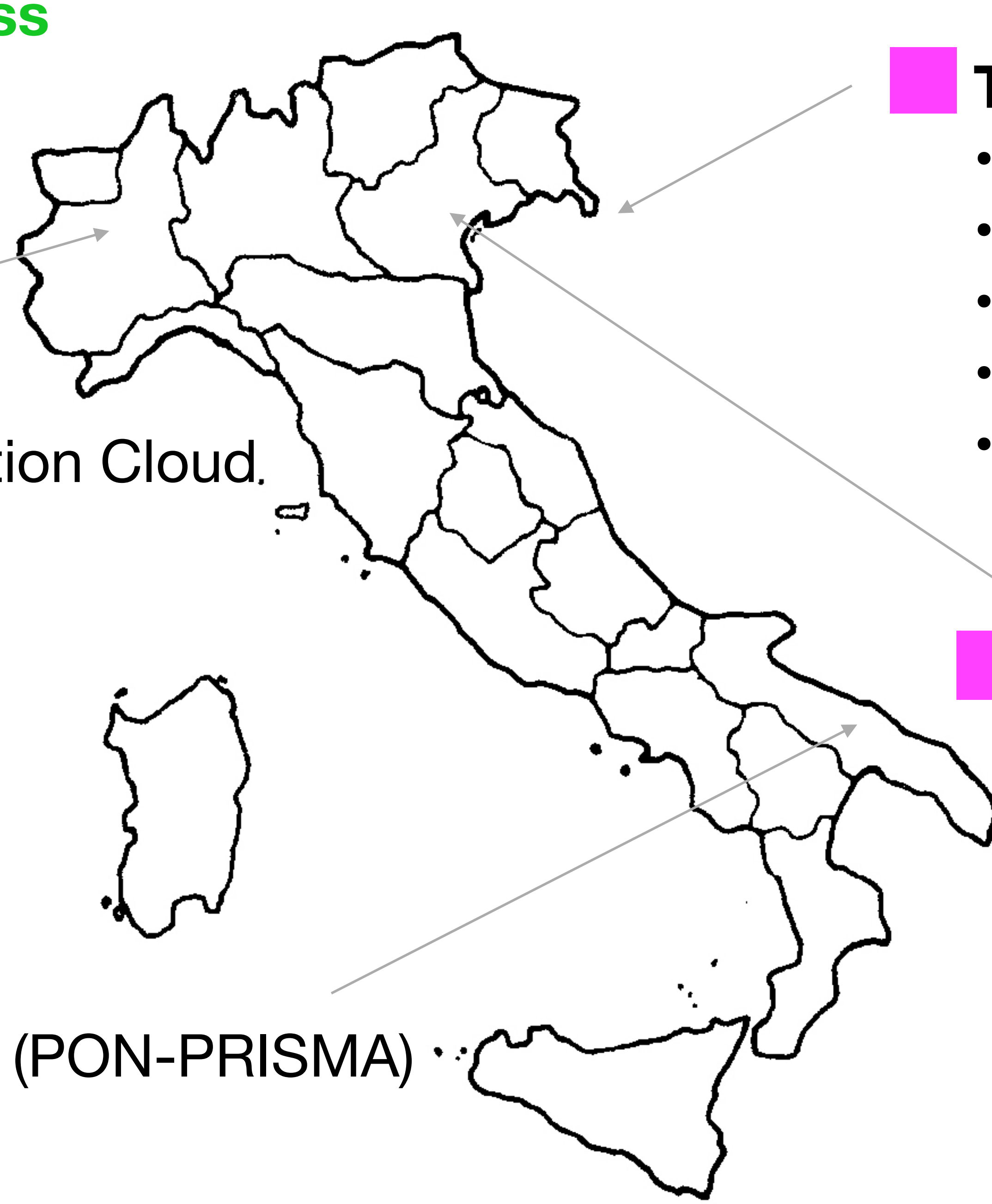
- improve the robustness of the existing LHC Italian infrastructure
- global effort to **ease data and resources access** to LHC users:
  - parallel and interactive analysis solutions (i.e. Virtual Analysis Facility for ALICE)
  - standard access to interactive resources of different local deployments (i.e. centralised authentication system)
  - federation among single analysis facilities to optimise distribution and access to remote data
- build a **uniform environment** capable of managing at once **interactive and batch** activities:
  - Cloud Computing paradigm (isolate applications, *elasticity*)
- allow **users outside high-energy physics** to fully exploit LHC computing infrastructures

Outline

Activities on interactive analysis on cloud infrastructures

Optimisation of data access

Monitoring



TRIESTE
- OpenStack test Cloud
- 24 cores
- 1.2 TB
- 1Gbps LAN
- 3 Gbps WAN

TORINO
- OpenNebula production Cloud
- 1.3k cores
- 1.6k TB (gross)
- 1-10 Gbps LAN
- 10 Gbit/s WAN

PADOVA-LEGNARO
- OpenStack test Cloud
- 100 cores
- 5 TB
- migrating to production

BARI
- OpenStack test-bed (PON-PRISMA)
- 600 cores
- 110 TB
- 10 Gbps LAN/WAN

Virtual Analysis Facility (VAF) for ALICE

The ingredients:

- Proof On Demand (**PoD**)
- **HTCondor** as batch system (cloud-aware)
- **Elastiq** daemon (optimisation of resource usage)
- **CernVM Online** for cluster contextualisation

Activities:

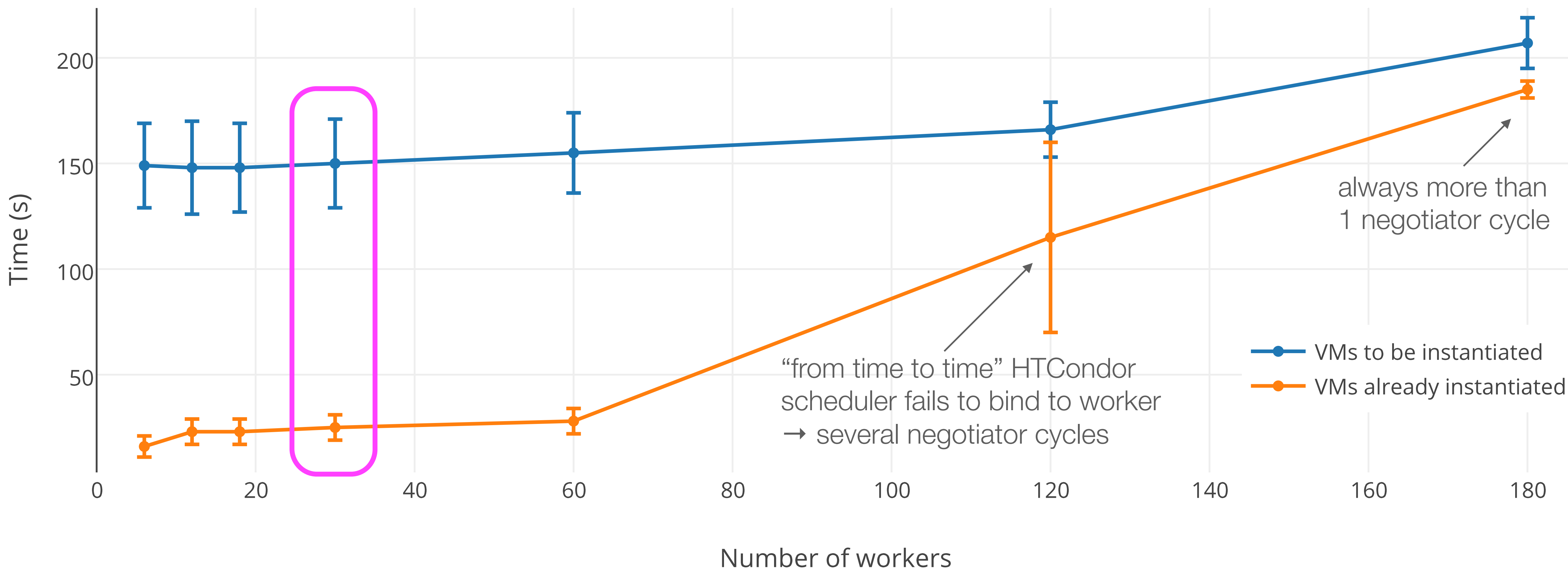
- benchmarking activities at all sites (common analysis task and data-set)
- tests on local data storage access (Trieste)
- application monitoring with the ElasticSearch ecosystem (Torino, Padova)
- in production at the Torino site:
  - in operation since November 2013
  - 5 active users
  - 60 TB of dedicated storage (GlusterFS, Xrootd)
  - up to ~100 workers
  - mainly analysis on *ntuples* (TSelector)

Workers deploy time

VAF benchmarking

- 10 measurements per point
- error-bar is the standard deviation

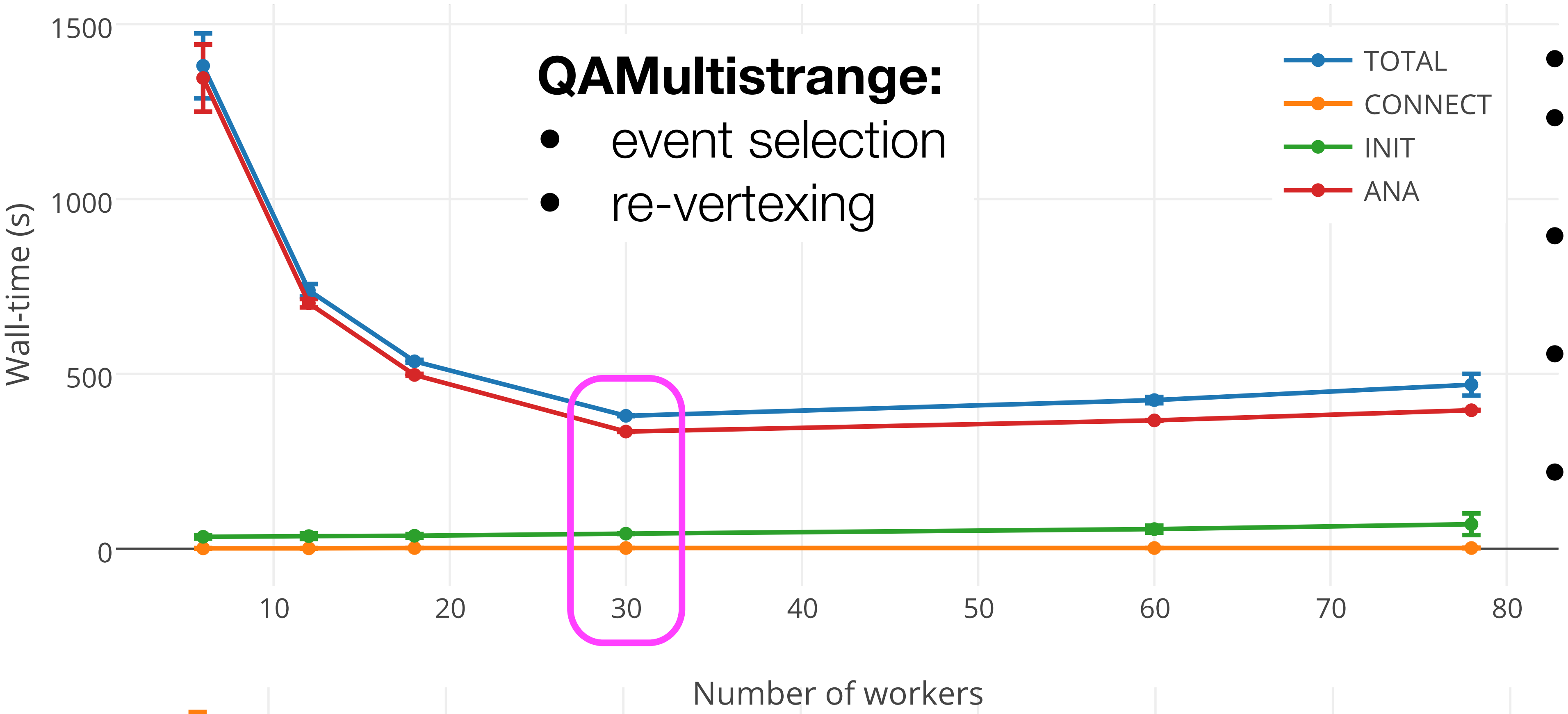
Similar study at all other sites



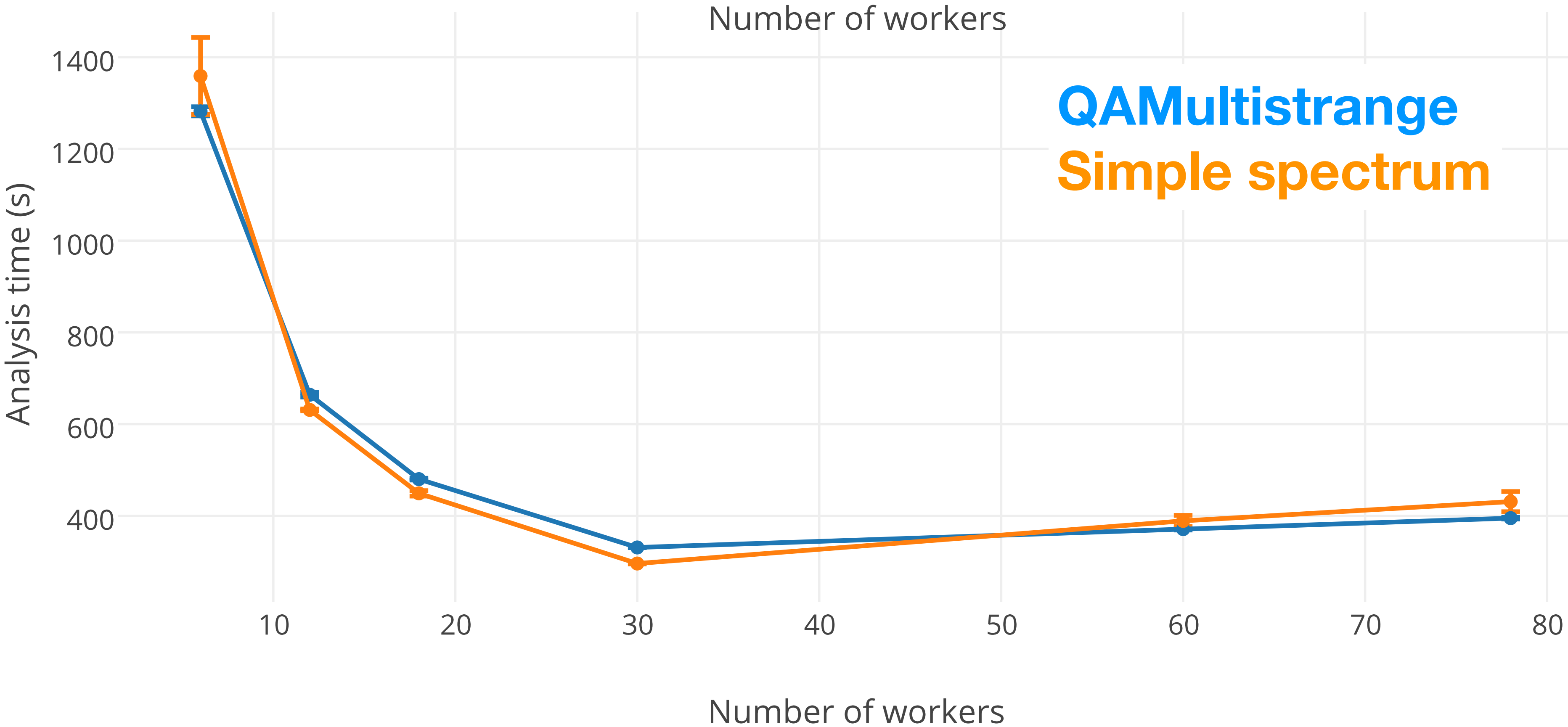
- if new **VMs need to be instantiated**, workers deploy time ranges from **2.5 min** to **3.5 min**
- if **VMs are already available**, workers deploy time ranges from **16 s** to **3 min**
- the golden number of **30 workers** (see later) is reached in **2.5 min** in the first case and **25 s** in the latter

Wall-time for different analysis steps

VAF benchmarking



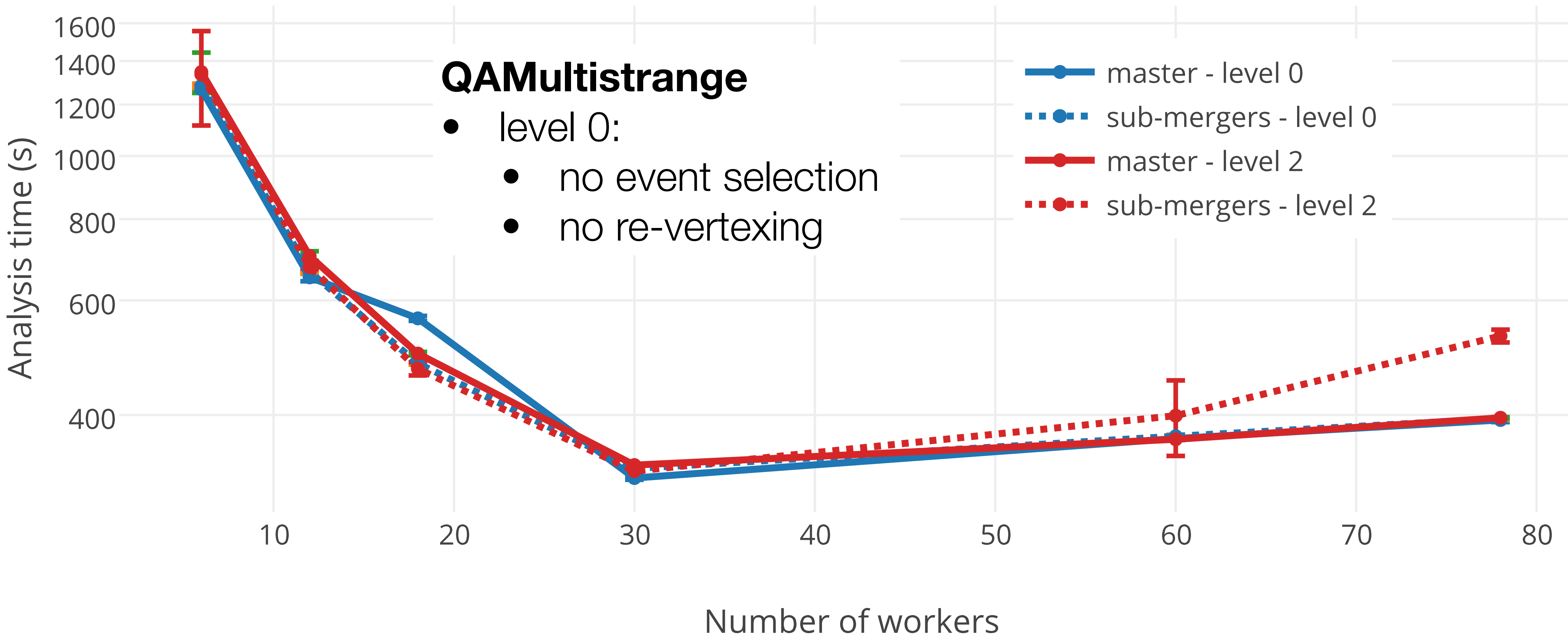
- 3 measurements per point
- error-bar is the standard deviation
- analysis task 1: **QAMultistrange**
- analysis task 2: **simple p_T spectrum**
- data sample:
  - **LHC10h** (PbPb)
  - run 139510
  - ~ 226k events



For this type of analysis and number of events, ~ **30 workers** is the optimal number

Wall-time is **comparable** for low and high CPU-intensive analyses

Compare merging strategies



- compare wall-time for analysis and merging: merging on master and with sub-mergers
- sub-mergers are activated in $HOME/.PoD/user_xpd.cf1:

    `xpd.putrc Proof.UseMergers 0` (0=calculate optimal number of mergers given the number of workers)

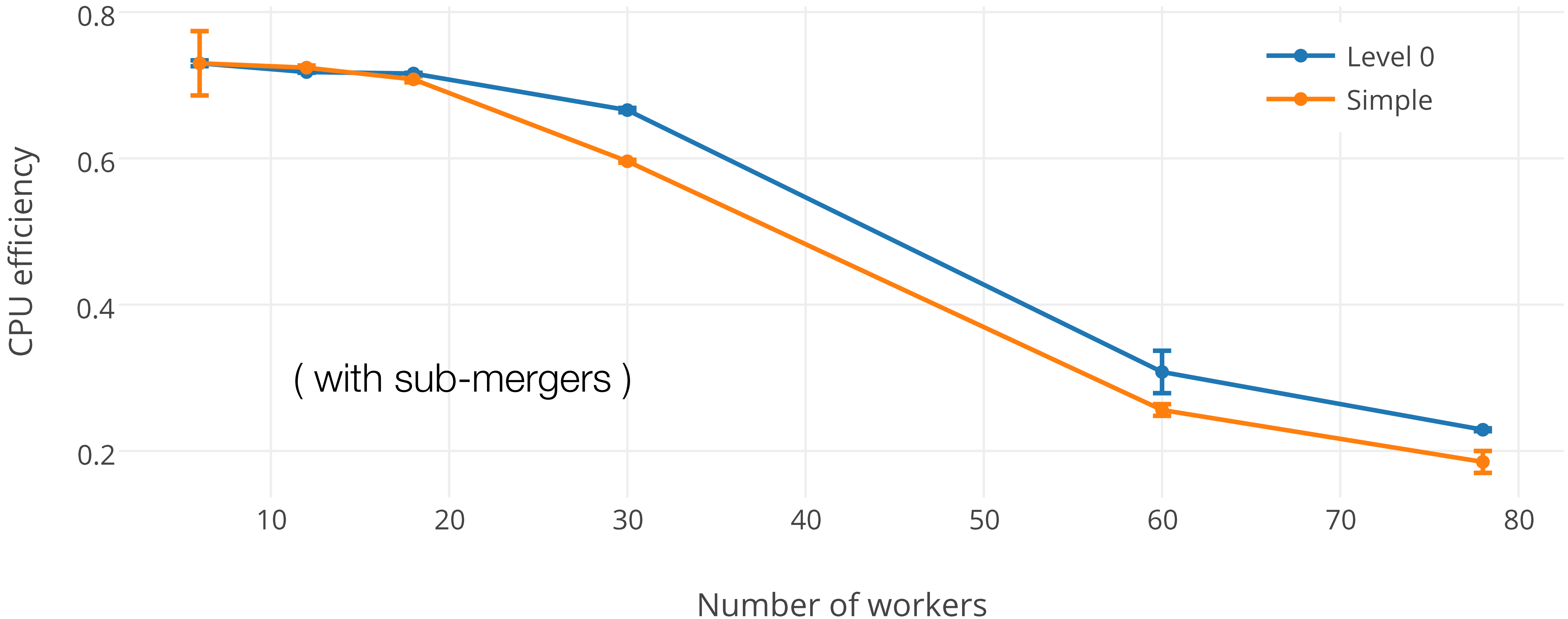
- no striking difference in wall-time
- BUT **sub-mergers avoid crashes on master due to too high memory consumption**

VAF benchmarking

CPU efficiency during analysis and merging

VAF benchmarking

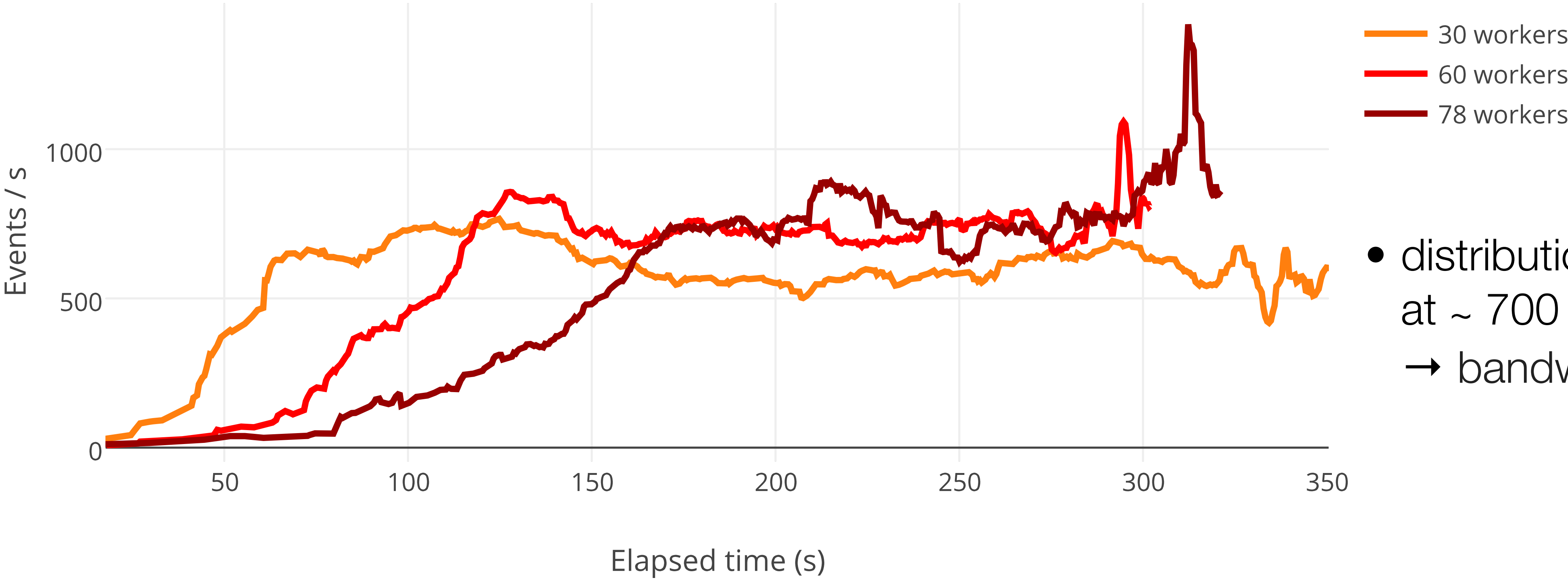
$$\text{EFFICIENCY} = \frac{1}{\text{total wall-time at master}} \frac{\sum \text{CPU-time at workers}}{\text{\# of workers}}$$



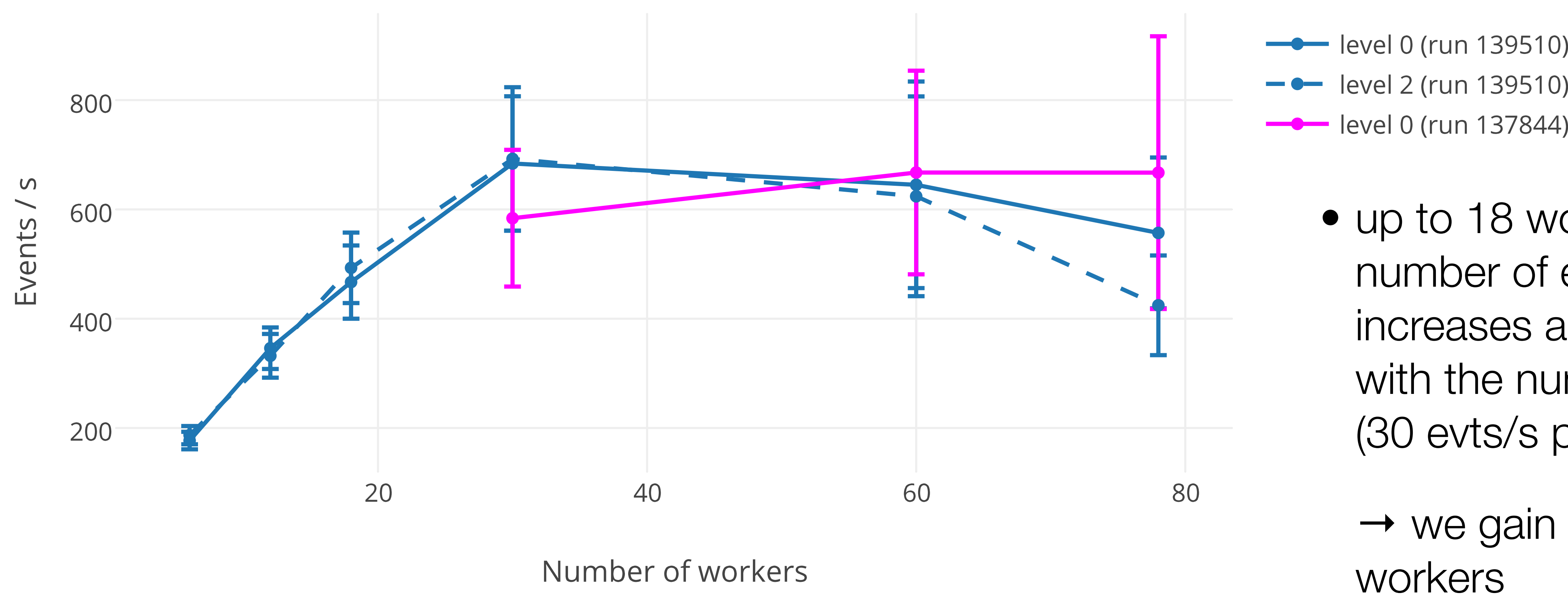
- information from monitoring database
- cpu-intensive analysis mode is slightly more cpu-efficient
- **cpu-efficiency decreases with increasing number of workers**

Events analysed per second

VAF benchmarking



- distributions reach a plateau at ~ 700 evts/s
  → bandwidth limitation



- up to 18 workers the mean number of events analysed increases almost linearly with the number of workers (30 evts/s per worker)

→ we gain by adding more workers

Local data storage tests

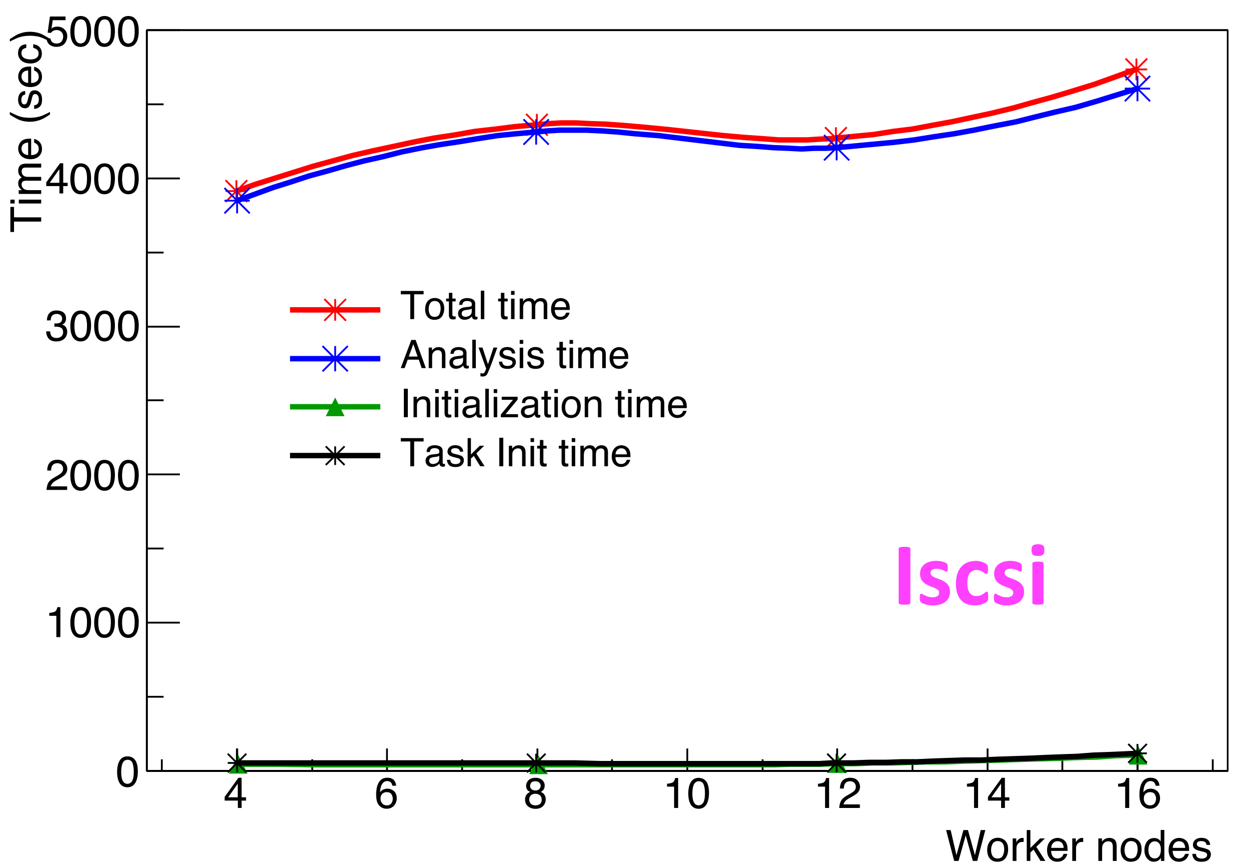
Trieste

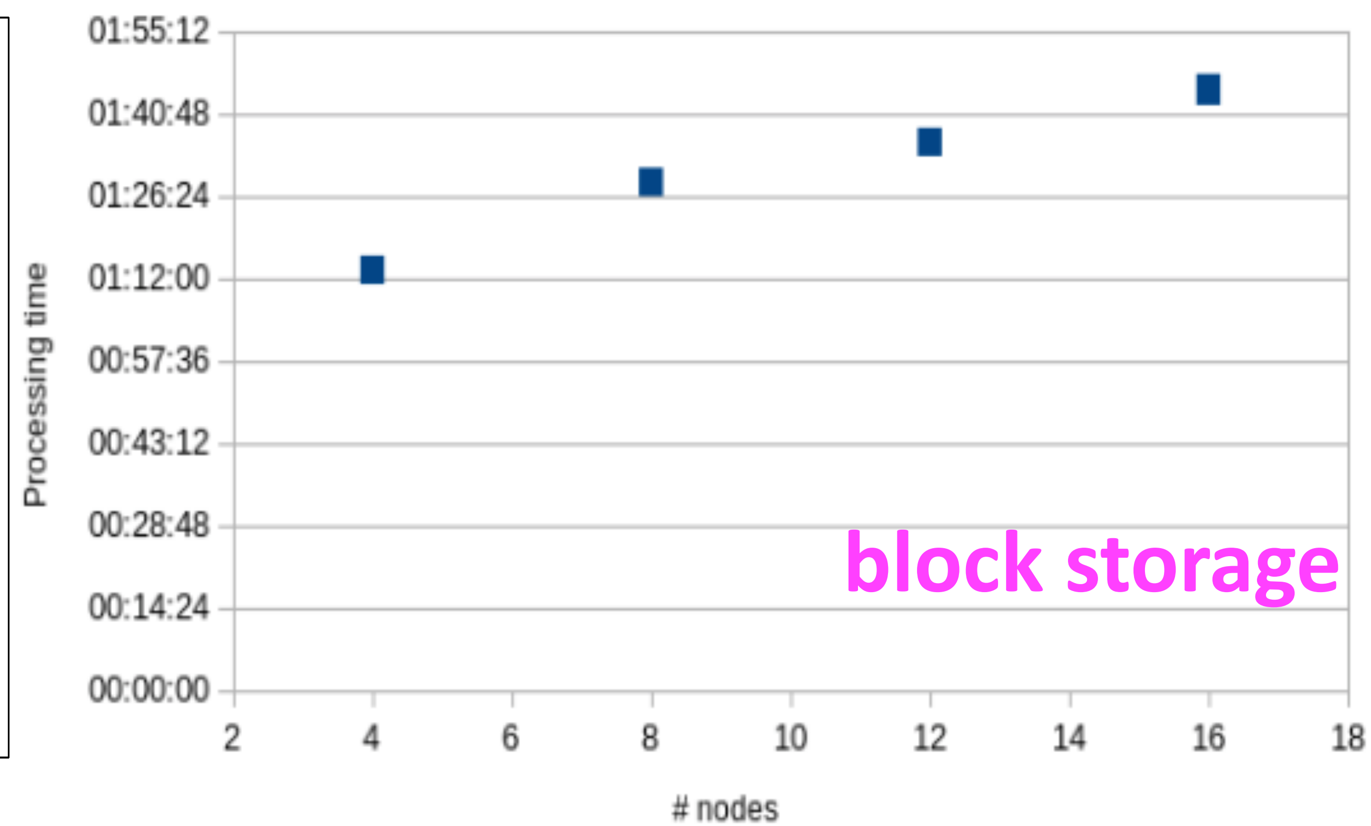
3 possibilities have been explored:

- Virtual block storage with GlusterFS exported by Cinder to the worker nodes through:
  - nfs
  - Xrootd

networking filtered by the cloud controller
→ servers configured inside the cloud

- Volume **Iscsi** exported by Xrootd server





- for 16 workers the Iscsi option takes roughly 70% of the time than the Xrood one

Data access

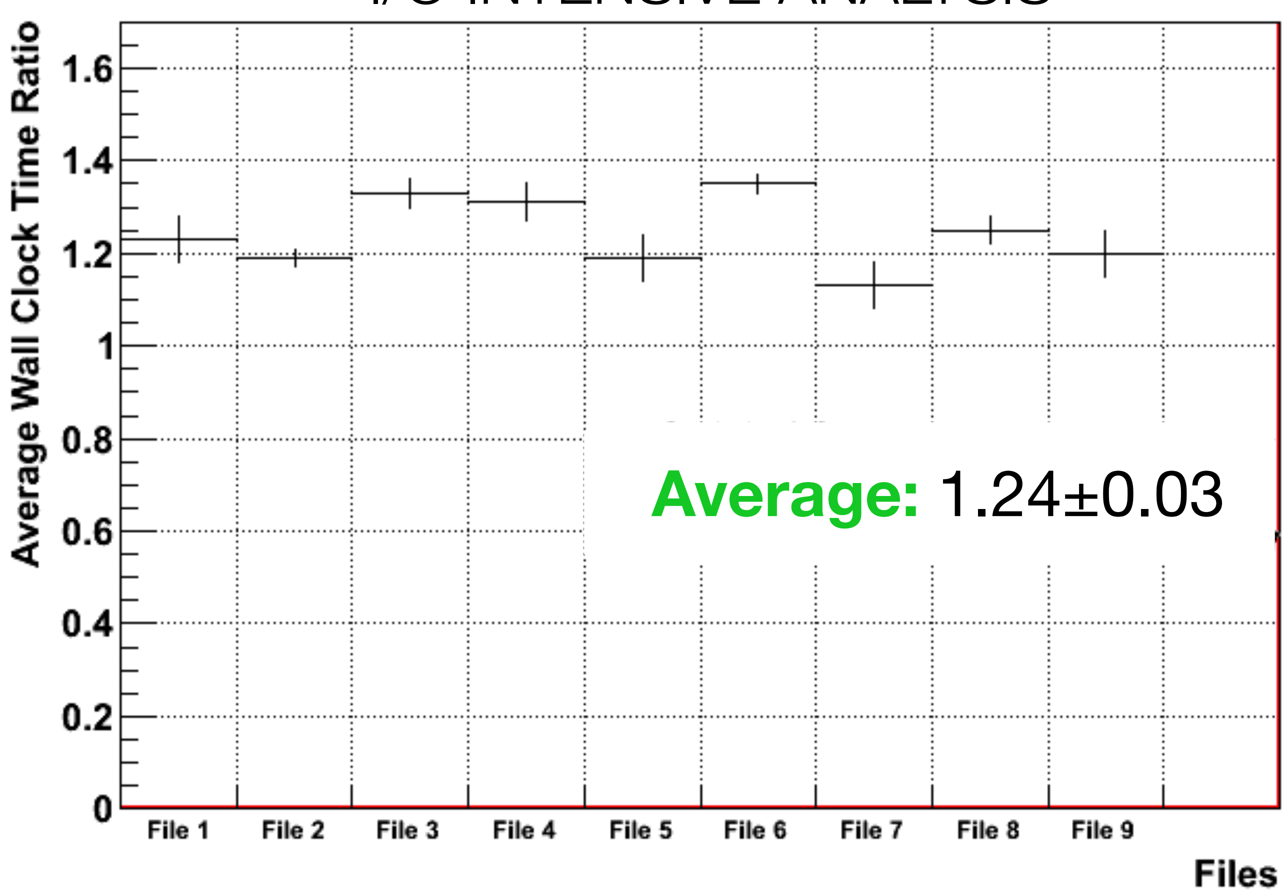
A Distributed Storage and Data Federation for VAF

- distribute and share data using a unique **XRootD Italian redirector** is under investigation
- two steps of a test-analysis:
  1. 75% I/O intensive and 25% CPU intensive
  2. 17% I/O intensive and 83% CPU intensive

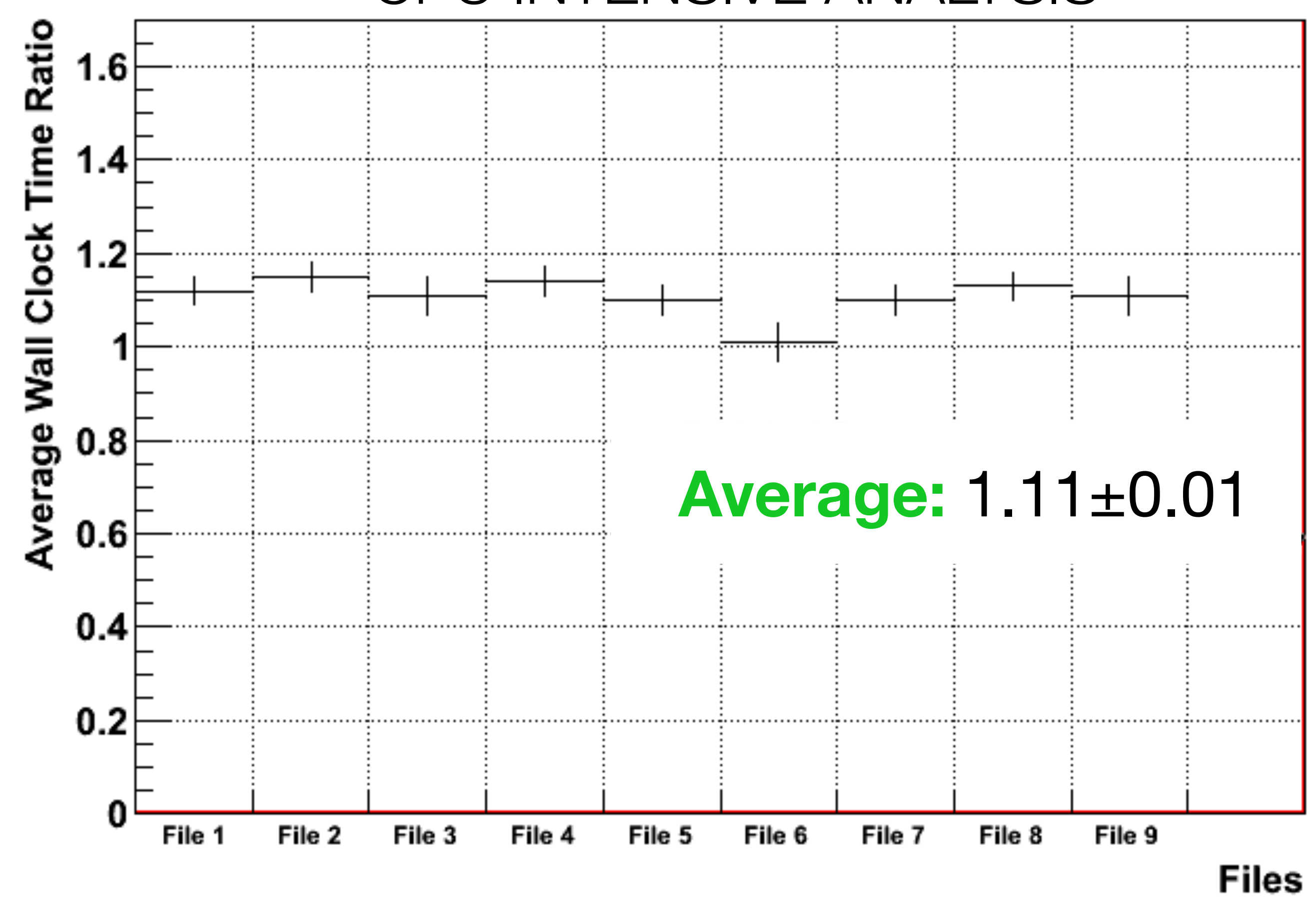
Bari

Ratio between wall time of jobs accessing files via XROOTD-IT and locally

I/O INTENSIVE ANALYSIS



CPU INTENSIVE ANALYSIS



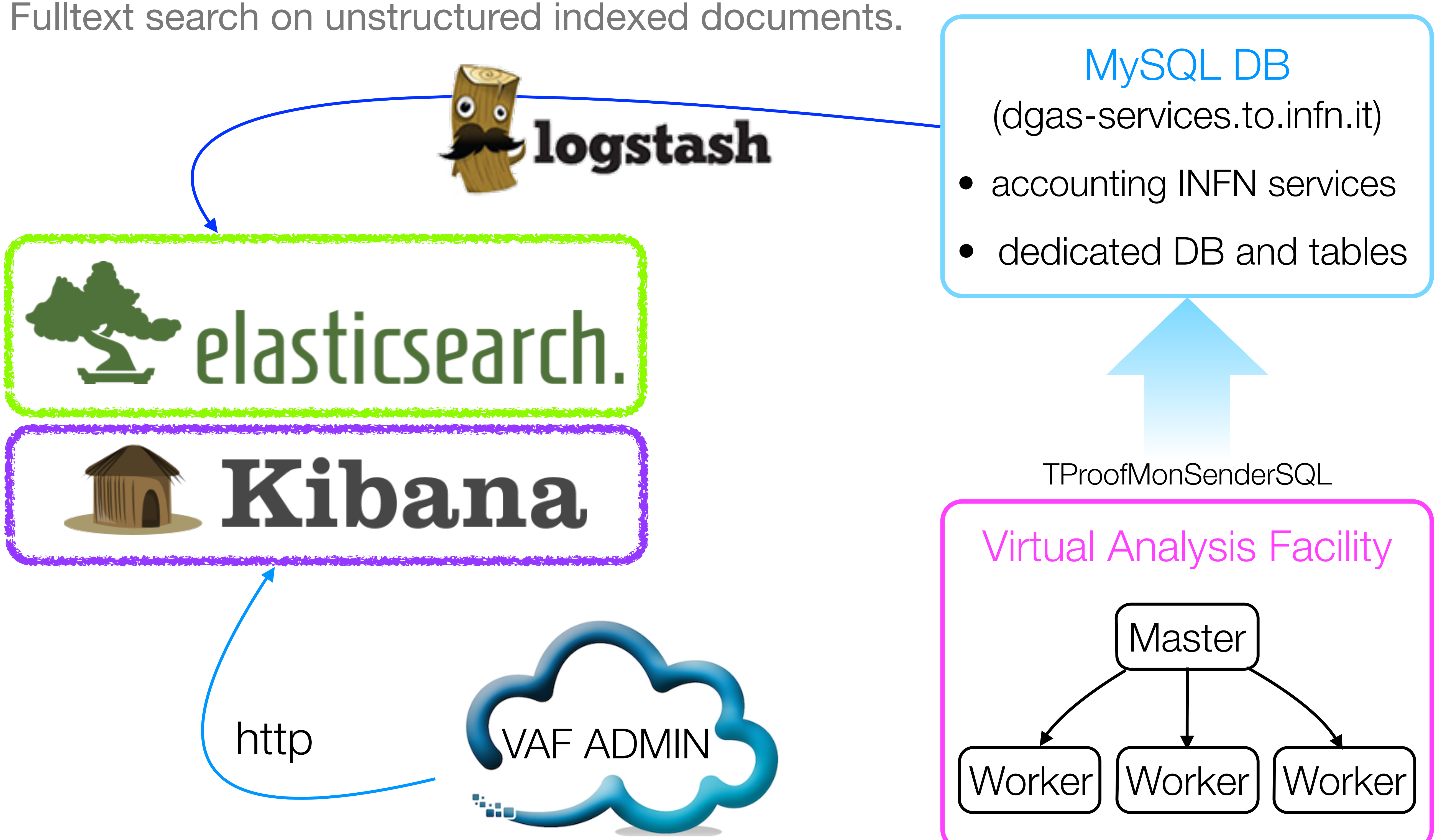
- **difference within 10-20%** at most, even for I/O intensive jobs
- encouraging to further develop the VAF data federation using such XRootD option

VAF monitoring with the ELK stack

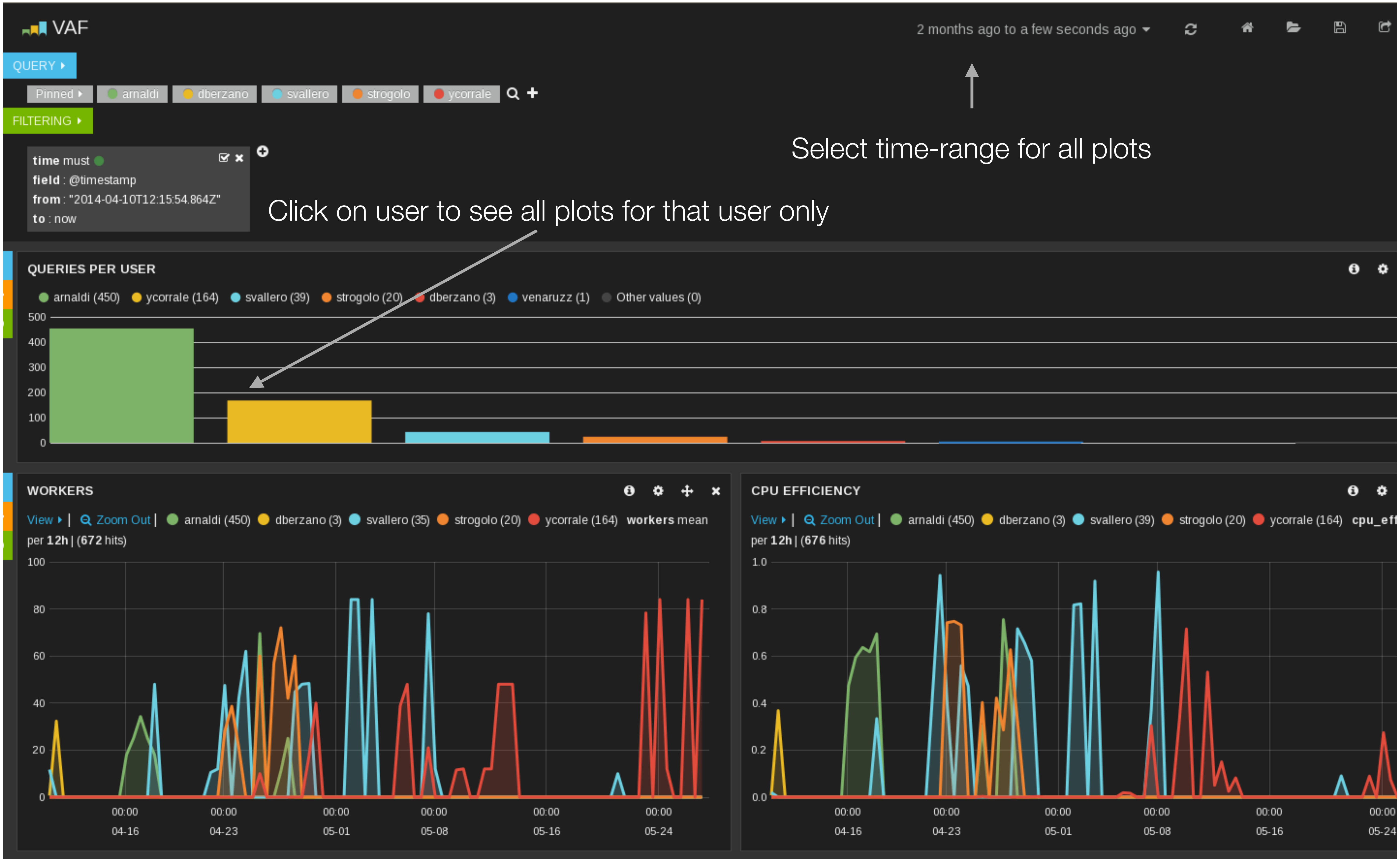
Monitoring

Elasticsearch: search and analytics engine.

Fulltext search on unstructured indexed documents.



The VAF dashboard

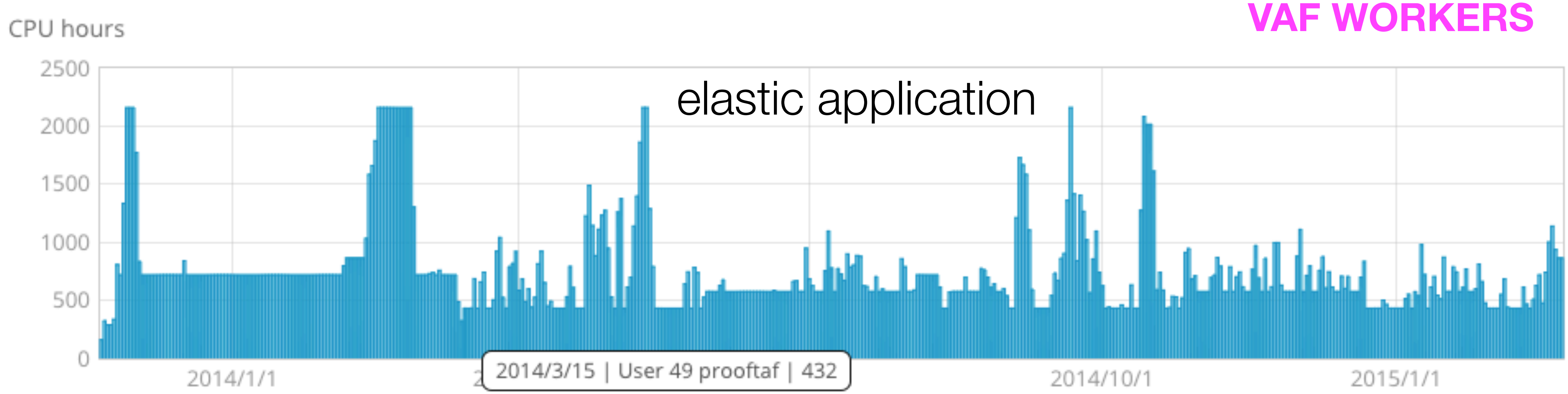


Monitoring

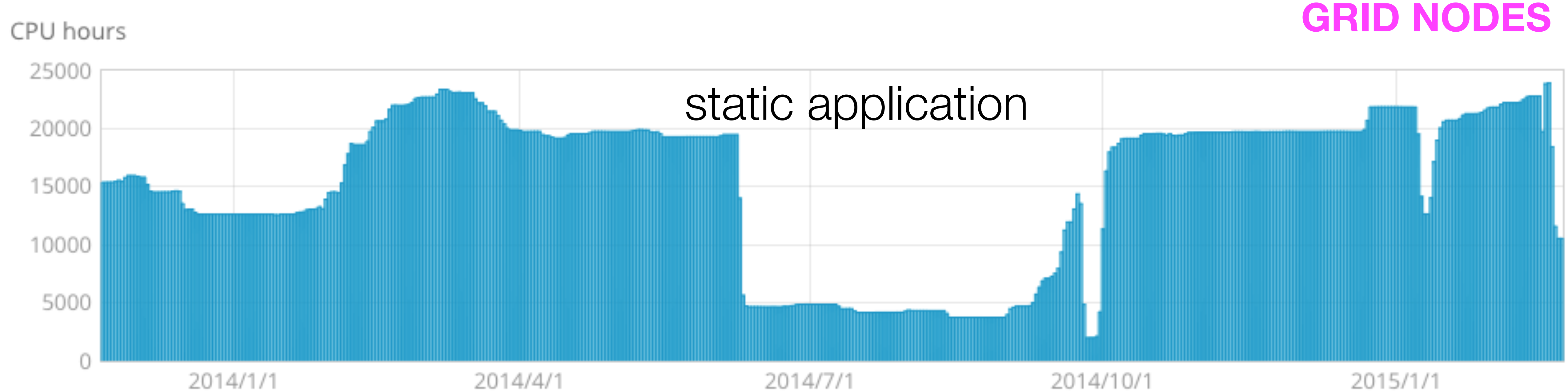
Actually many more monitored quantities: memory, datasets…

OpenNebula Accounting

VAF WORKERS



GRID NODES



- GRID nodes are roughly 60% of all virtual machines
- implement *elasticity* also for the (or part of) GRID nodes

Summary

- VAF operational in all sites at different levels of maturity
- Benchmarking results:
  - deploy time for 18 workers: 600s (BA), 400s (PD/LNL), 150s (TO)
  - analysis time does not depend by the CPU intensiveness of the task

    → data access is the dominant factor
  - CPU efficiency decreases with increasing number of workers
  - convenient to enable sub-merging on the workers
- Access to local data:
  - Iscsi exported via XRootD gives better performance than block storage exported in the same way
- Data federation:
  - encouraging indications to use XRootD
- Monitoring:
  - investigation of the ELK stack to handle heterogeneous data sources (applications, IaaS)
  - allows inspection of unstructured data
  - possible solution for Monitoring-as-a-Service

Outlook

- Ongoing work on:
  - data federation (BA, TS)
  - Monitoring-as-a-Service
  - Tier2 elasticity: ALICE, BESIII (TO)
  - elastic farm (non Proof based) for ALICE (TO)
- Open LHC computing infrastructures to non-HEEP users:
  - interest in the VAF system from Auriga-Virgo and CUORE groups (PD/LNL)
  - prototype of elastic cluster for medical imaging application (TO)