



CHEP 2015 - April 13-17 - Okinawa

Integrated Monitoring-as-a-service for Scientific Computing Cloud applications using the ElasticSearch ecosystem

Speaker: Sara Vallero

Bagnasco Stefano
Berzano Dario
Guarise Andrea
Lusso Stefano
Masera Massimo

*The present work is partially funded under contract 20108T4XTM of
Programmi di Ricerca Scientifica di Rilevante Interesse Nazionale (Italy).*



Istituto Nazionale
di Fisica Nucleare



The INFN Torino Private Cloud



WLCG Tier2

(ALICE, LHCb, biomed, CTA, Panda, Belle2)


Virtual Farms on-demand:


theory
Compass
medical imaging
JLab
...


Customised VMs:


ufsd
nuclear plant simulation
...

R&D



 1.3k cores

 1.6k TB (gross)

 1-10 Gbps LAN
10 Gbps WAN

DIRAC Tier2

(BESIII)



Virtual Analysis Facility

(ALICE)

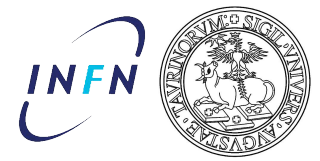


Elastic Farm

(ALICE)



The problem to be solved



What we want to achieve:

- real-time information on resource utilisation and system health
→ proactive monitoring
- historical information on resource usage for individual tenant
→ accounting/billing
- insight in application activity

but...

- several tenants
- application autoscaling → see poster-session A (contribution 387, booth 28)
- heterogeneous data sources
- towards monitoring-as-a-service (even more unpredictable data sources)

Elasticsearch:

- search and analytics engine (Apache Lucene)
- entries are stored as JSON documents, all fields can be indexed and used in a single query
- full-text search on unstructured data
- API driven: you can use any RESTful API using JSON over http
- horizontally scalable

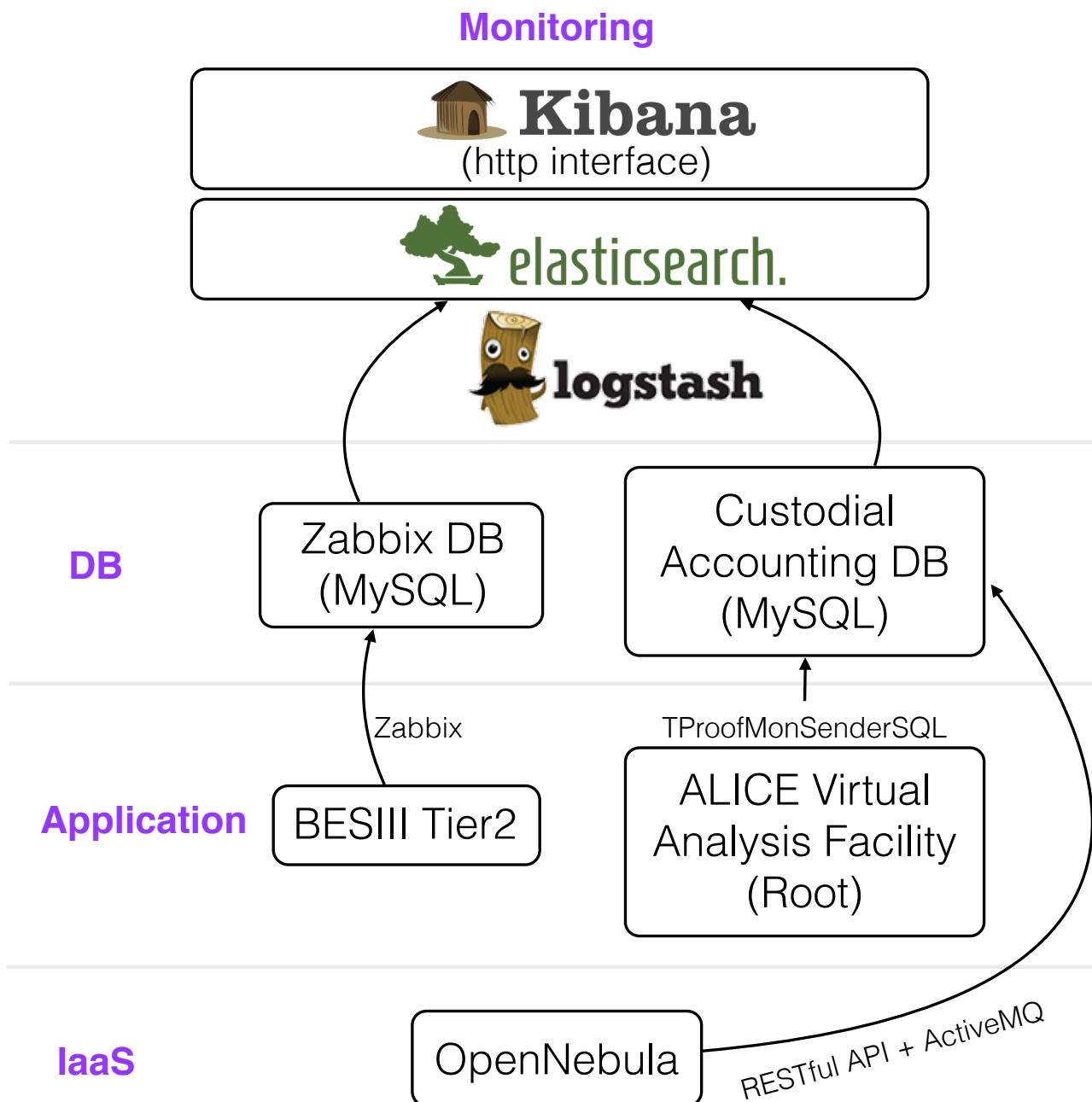
Logstash:

- tool to collect and parse events and log-files to a central service
- easily customisable via plugins

Kibana:

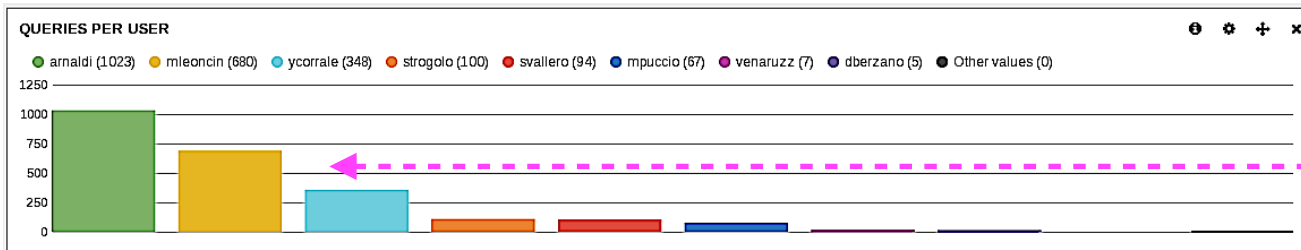
- GUI for displaying/searching ElasticSearch data
- implementation of interactive dashboards in few mouse-clicks

General set-up

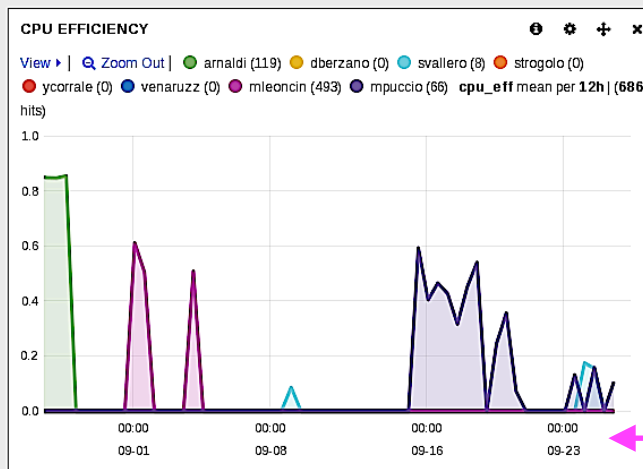
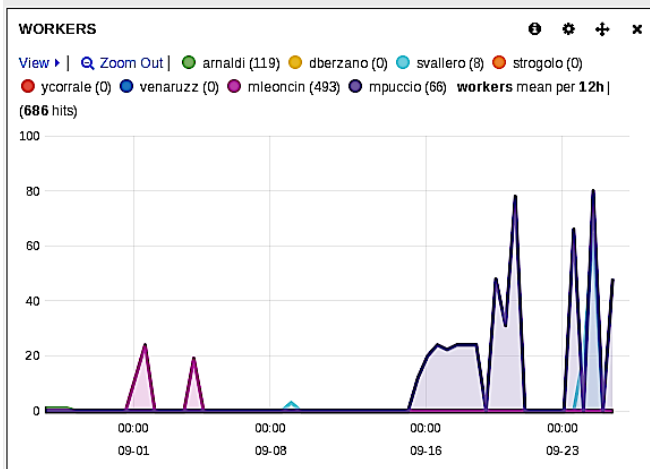


- data stored in high-availability MySQL server
- redundant step, but allows for more flexibility
- Italian Grid accounting now dismissing MySQL
- the framework was developed to monitor user activity within the VAF
- as a proof of concept also retrieve info from Zabbix DB, some custom view was created to ease indexing

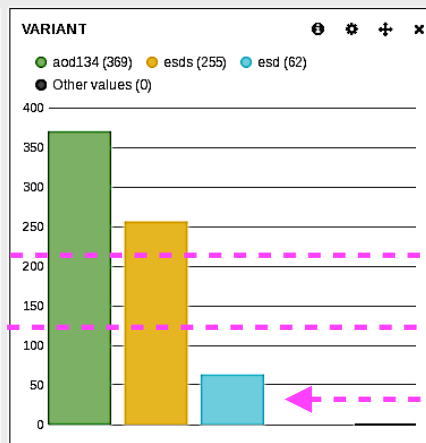
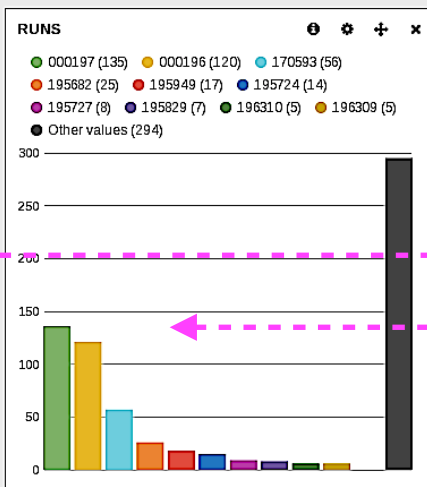
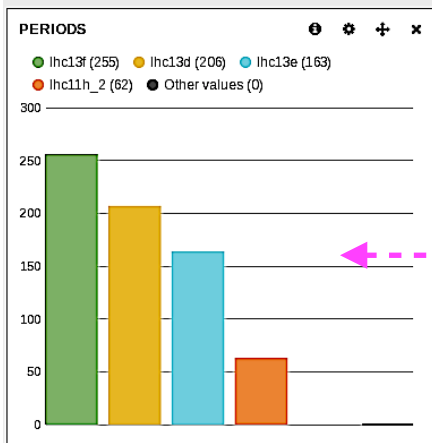
Applications monitoring: VAF dashboard



click to select entries for specific user only



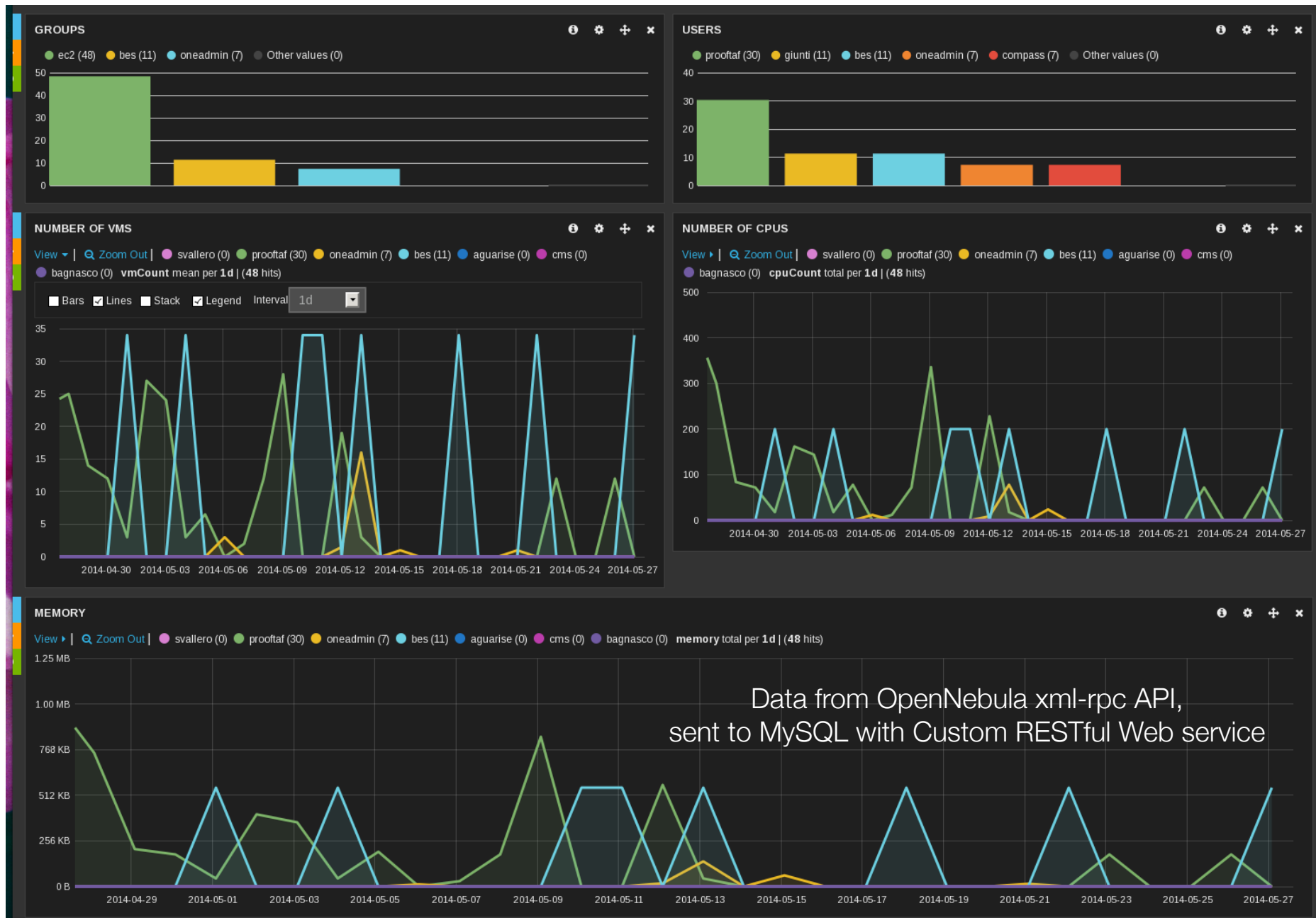
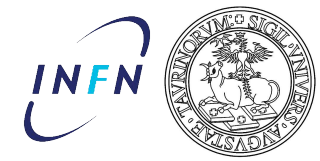
dynamic time range (the selection is applied to all plots)



example of ES analysis on a complex string

path/to/file

IaaS monitoring



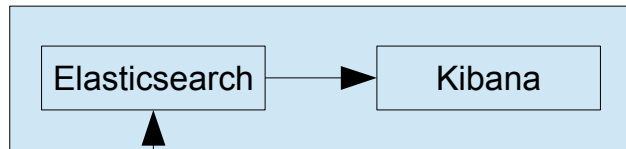
Italian Grid accounting



New System (Work in Progress)

DB and Search

Dashboard

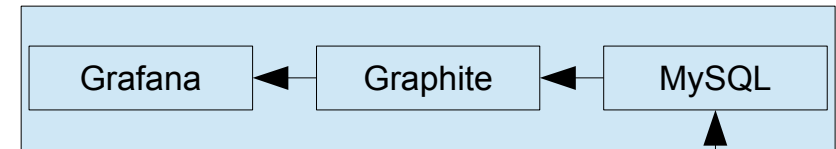


Current System in Production

Dashboard

Search

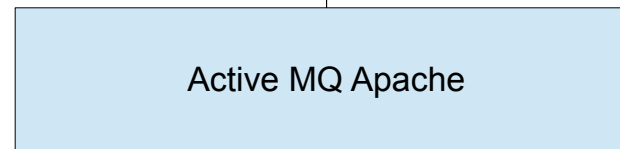
DB



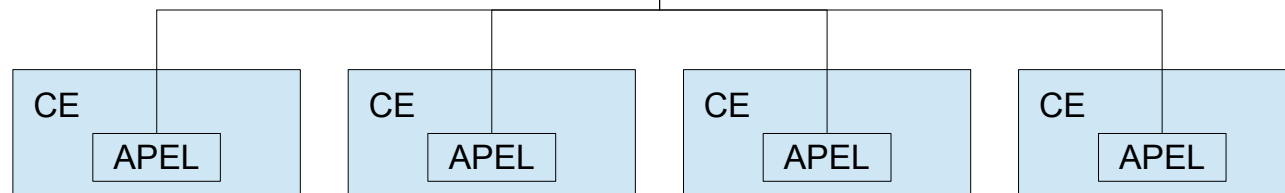
A/O

Accounting
Informations

Queue system



Workload Logs



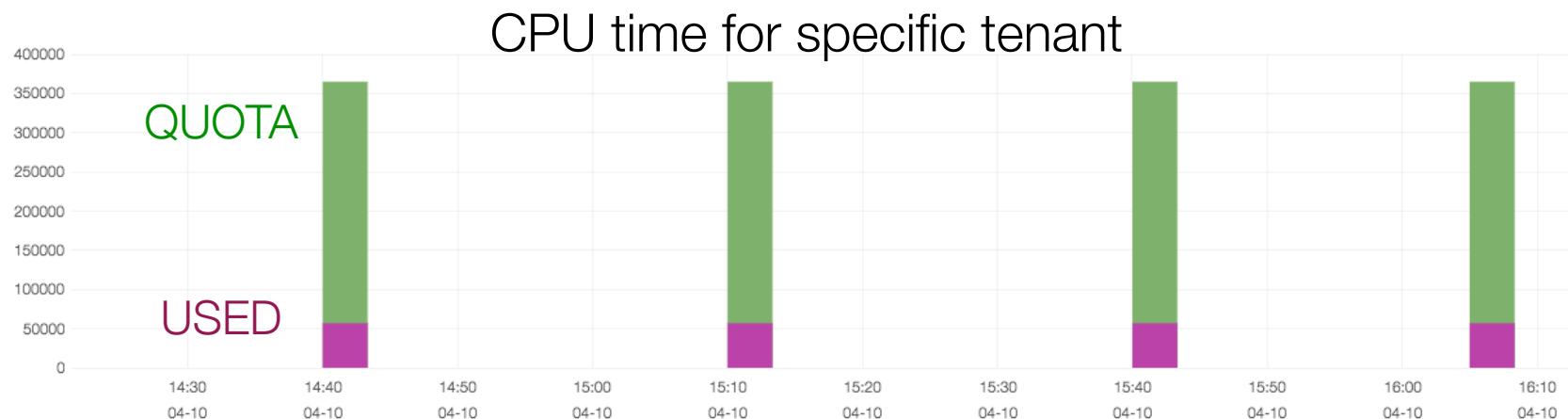
Computing Elements in Batch Farms of different Grid Sites

Billing



- force users to **release unused resources** (when auto-scaling is not there)
- OpenNebula implements accounting but we need finer tuning
- need for alarms or to trigger actions if quota is exceeded
- implemented **custom service*** with asynchronous two-daemons logic:
 - gather resource usage metrics with the ON xml-rpc API and publish to message queue (RabbitMQ)
 - process requests and insert data in DB or send **e-mail in case of quota exceeded for user**

* <https://github.com/svallero/cloud-accounting>



Conclusions and Outlook



- we have implemented a prototype **uniform monitoring system across service levels**
- the **ELK stack** allows gathering and digesting heterogeneous data from many sources
- the ELK stack proved to be well suited for the task at hand
→ the Italian Grid accounting infrastructure is migrating to this model
- in our case (a medium size scientific computing centre) **“billing”** is mostly a way to **stimulate users to release unused resources**
→ a natural evolution is the implementation of fair-share scheduling in cloud controllers (one of the tasks of the INDIGO-DataCloud project)
- the next step is to **generalise this monitoring service**, providing a set of default sensors that each tenant can extend according to their needs