Site Description

Alessandro De Salvo

Jaroslava Schovankova

06-03-2018

Short summary of the strategy

- **The goal is to achieve a better site description with respect to what we currently have in Panda**

- **First approach: using GLUE 2**
  - **https://docs.google.com/document/d/1x_mqr_-VxosQvhvRNa26qcfNo6P9lVA9pckeU5_i2LQ/edit?usp=sharing**
  - **Not suitable for our purposes, values are not reliable enough**

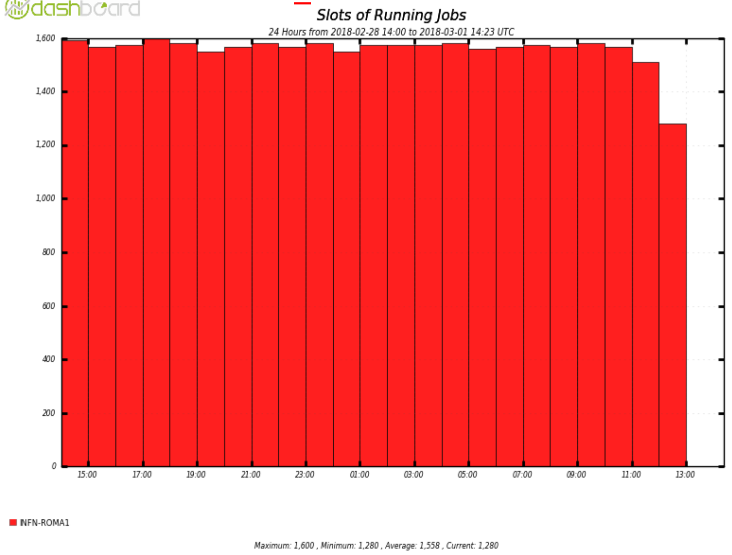
- **Second approach: building custom maps directly from jobs**
  - **No need to touch the pilot, just embedding callbacks in other infrastructures**
  - **Only when autosetup is called with the panda resource name we need to send out the data**
    - **Ensures that data is collected only for grid jobs**
  - **Data is totally custom, so we need to write plugins/providers for the different batch systems we want to support**
  - **Can send data using curl, complete freedom on the info to send out and the collector**
  - **We can achieve both a deeper view of the batch queues and a deeper view of the nodes, associated to the panda resources they belong to**
    - **Info not ataached to jobs but to nodes and batch queues**

Current status of the collector

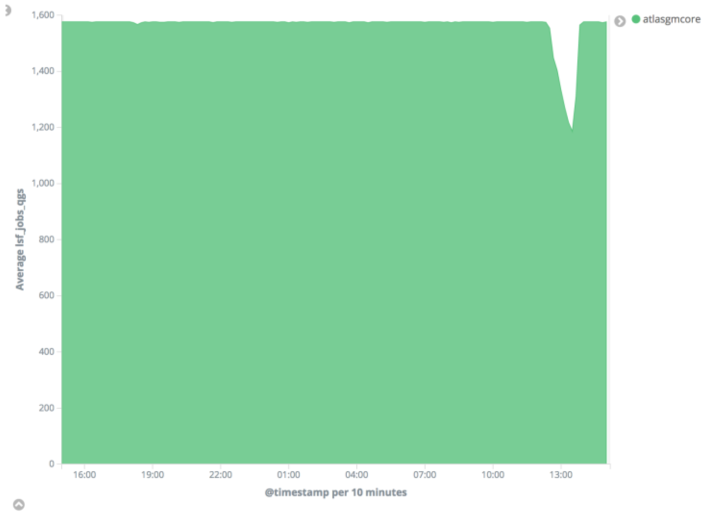
- Initial protototype of the collector
  - Embedded in autosetup
  - Supporting a subset of the batch systems
    - LSF, PBS and Condor
    - SLURM, SGE, and PBS experts are needed, as well as experts on “exotic” batch systems needed (is “arc” a batch system? Apparently yes, looking at AGIS)
  - Shipping data via curl into rabbitMQ -> logstash -> ES in Roma
  - Storing data for 1/10 of the job started
  - CSV data shipped via CURL
    - Low CPU usage for logstash, single instance in Roma can handle all ATLAS nodes with a fraction of CPU used (could not do the same if parsing via grok/regexp)
  - Many info already available via kibana
    - https://atlas-kibana.roma1.infn.it/goto/c8437edb46b281cd5446640f075bfba0
    - Example
      - Node address, name
      - Gateway (in case of natted nodes)
      - ATLAS site, Panda Site, Panda Resource -> node name
      - CPU model
      - Memory
      - # of CPUs
      - Queue name
      - Jobmanager type
      - Jobs pending/running/suspended in the queue
      - Total number of available slots (calculated, based on the internal nodes, for now available only for LSF and Condor)
      - …

Checking collected data: LSF running jobs

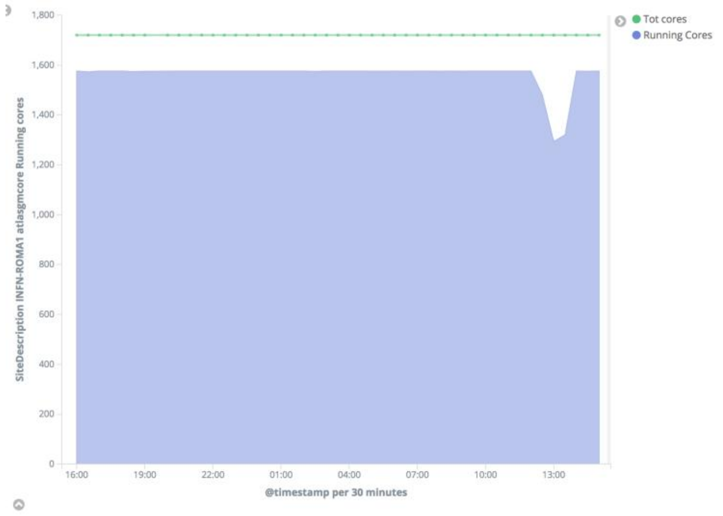
INFN-ROMA1_MCORE ATLAS Dashboard



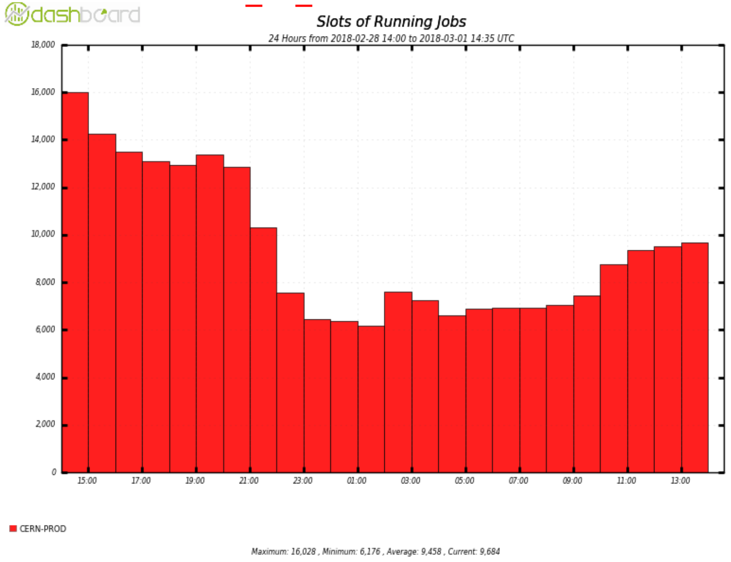
INFN-ROMA1_MCORE Local Monitoring



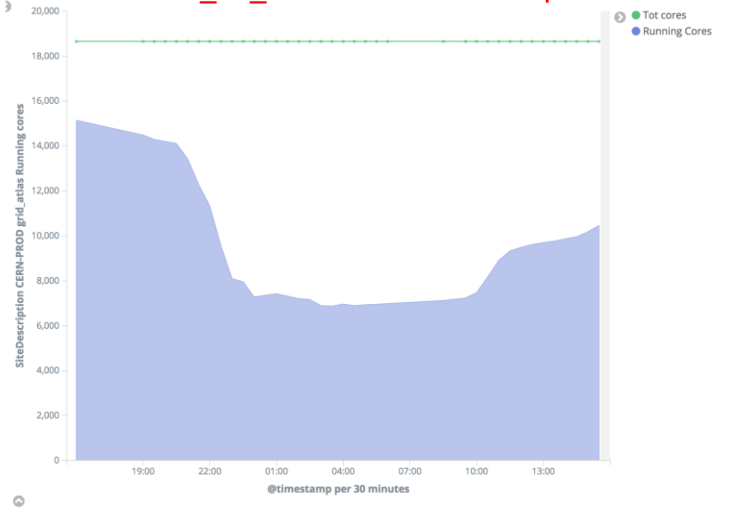
INFN-ROMA1_MCORE Site Description



CERN-PROD_T0_4MCORE ATLAS Dashboard



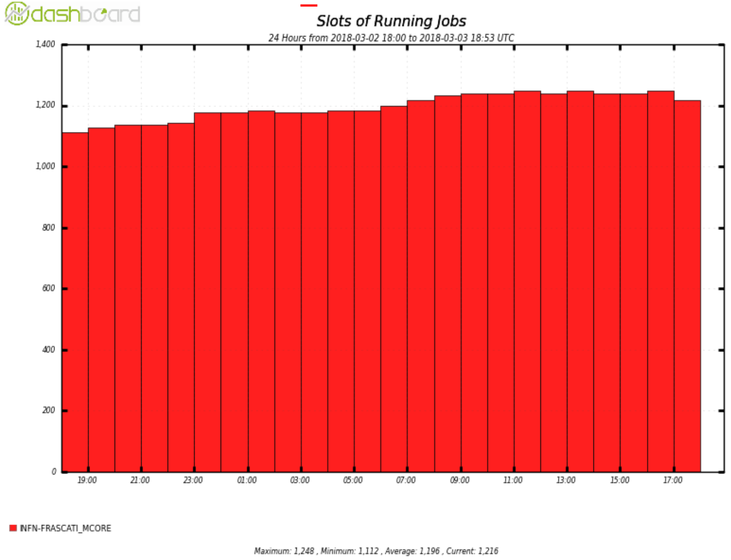
CERN-PROD_T0_4MCORE SiteDescription



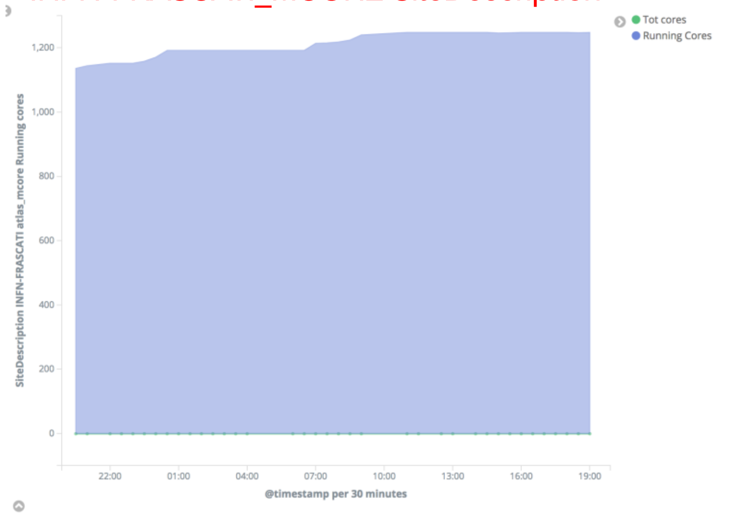
- **Good agreement between the collector and the dashboard**
  - **Big advantage since we have the inner view of the queues, including the max number of jobs**

Checking collected data: PBS running jobs

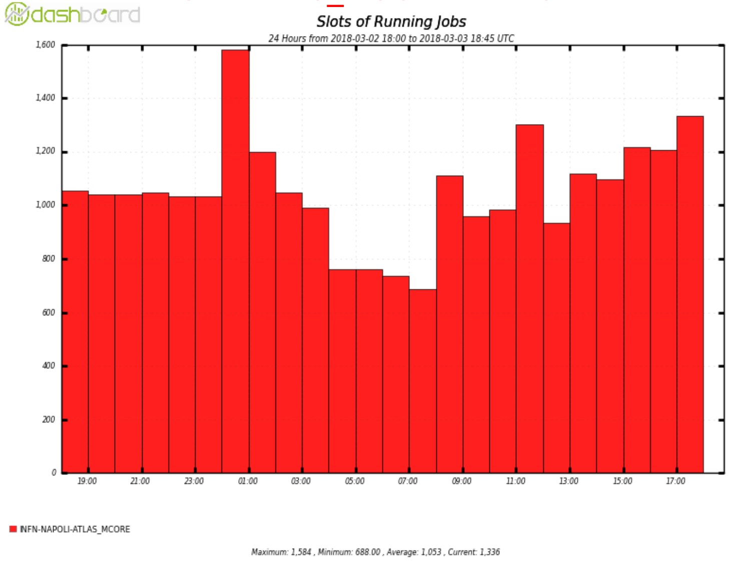
INFN-FRASCATI_MCORE ATLAS Dashboard



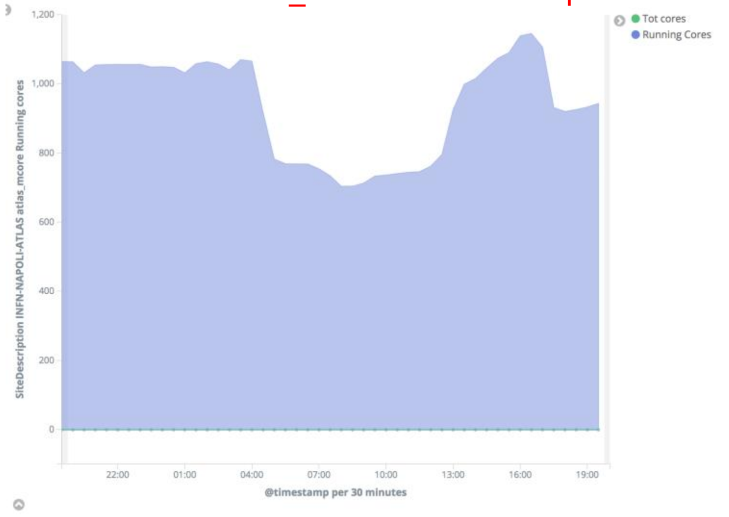
INFN-FRASCATI_MCORE SiteDescription



INFN-NAPOLI-ATLAS_MCORE ATLAS Dashboard



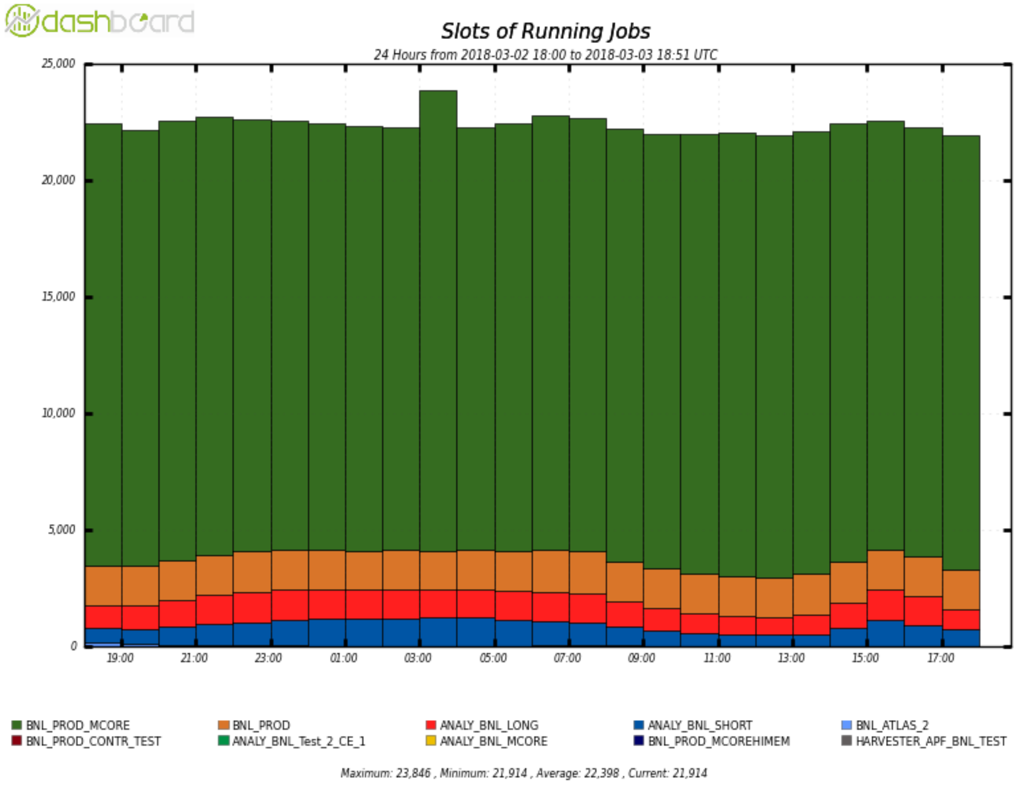
INFN-NAPOLI-ATLAS_MCORE SiteDescription



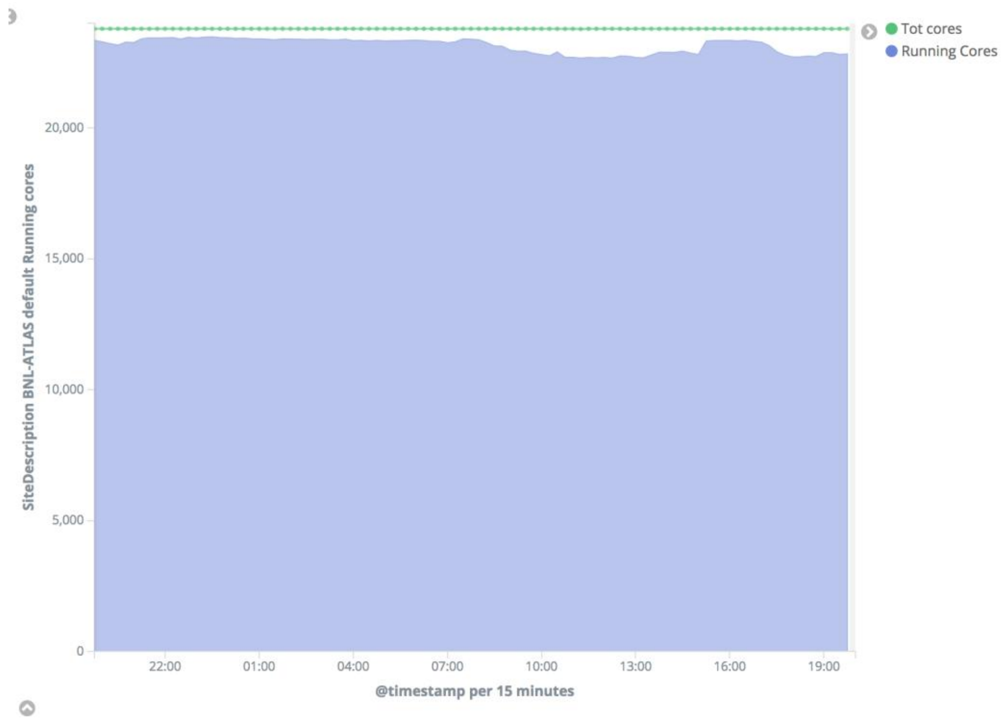
- **General good agreement between the collector and the dashboard**
  - **But it’s not possible (so far, at least) to derive e.g. the total number of slots without running privileged commands**
  - **Running privileged commands could need to establish an agent or cron job e.g. from the Computer elements, plus a bit of scripting**
  - **Some sites do not allow even the use of qstat in WN, in which case the probes are not effective**

Checking collected data: HTCondor running jobs

BNL-ATLAS ATLAS Dashboard



BNL-ATLAS SiteDescription



- **General good agreement between the collector and the dashboard**
  - **But not for all sites, still trying to understand why some of the sites are just reporting 0 running jobs, of sometimes twice**
  - **Very complex task for Condor (thanks to Jarka for providing the support for it!)**

What can we learn from this info?

- **Easy to derive several useful info**
  - Nuber of running/pending/suspended jobs in the internal batch queues
  - Nodes shared among several Panda Resources
  - Total number of slots (physical limit), but not in all cases if just running as unprivileged users
  - Real usage of the site queues (e.g. “are we really filling up all the defined nodes?”)
  - …
- **What can we also learn?**
  - Many sites are exposing strange values in AGIS
    - Example: different nomenclature HTCondor, HTCondorCE, condor for the same batch type
    - What is the “arc” jobmanager?
  - Other sites are publishing a wrong JM type
    - Example, DESY is publishing to be pbs, while it seems it has UGE
    - Need to improve the batch systems autodetect features
- **Other questions**
  - How can we make an efficient use of this info from Panda?
  - How to extend to the other batch systems?
    - We need SLURM, SGE and PBS experts to help building or improving the providers

Conclusions and next steps

- The initial collector prototype is able to give deeper views of the site internals and setup
  - But more coverage and batch experts needed
  - Extensible infrastructure, very easy to add more info, if available or possible to derive

- Next steps
  - Stabilize the current implementation of the probes
    - Extend the batch system types coverage
    - Crosscheck with problematic sites
    - Understand how to derive privileged informations
  - Migrate to the official ES/Kibana (Analitics Platform)
    - Not difficult to achieve, everything should be already in place, just needs some coordination
    - Not a big amount of data, but we'll have to monitor and pack as needed
    - Information can be easily accessed via python, jypter notebooks, etc, and possibly injected in Panda for further usage (or used directly)
  - Include the site description probes in HC, and eventually operate them from there
    - Lighter approach for sites, but we need to be sure the site coverage is complete in this way