21st International Conference on Computing in High Energy and Nuclear Physics **CHEP2015** Okinawa Japan: April 13 - 17, 2015

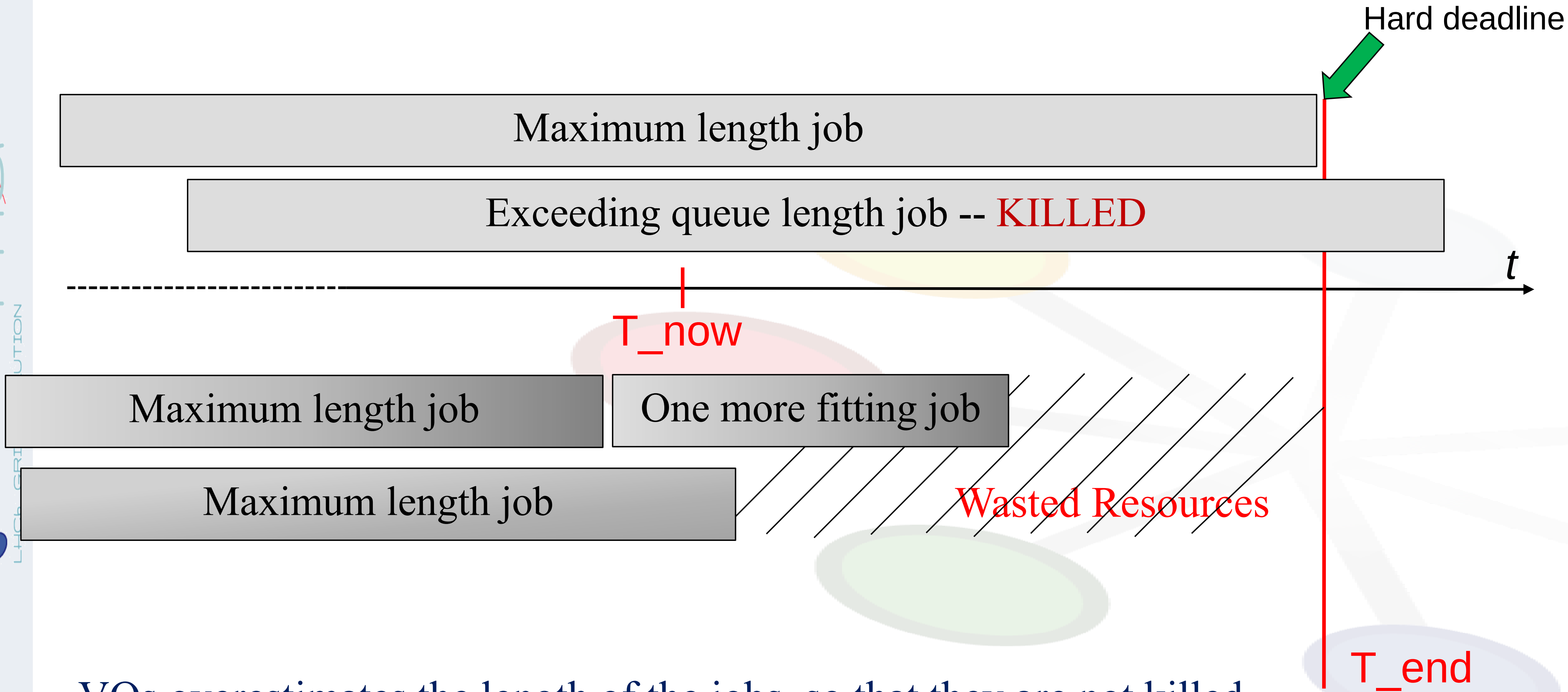
Jobs masonry with elastic Grid Jobs

Federico Stagni, Philippe Charpentier

On behalf of the LHCb collaboration

The masonry problem

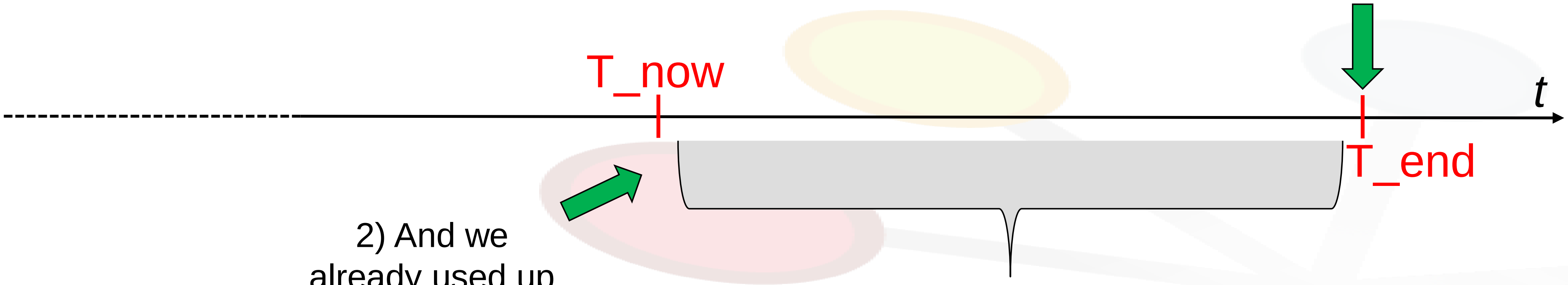
The jobs masonry problem (*)



VOs overestimates the length of the jobs, so that they are not killed

* copyright: A. McNab

Elastic Simulation jobs



1) We (just) discovered that we can run up to here..

2) And we already used up to here..

3) we'll try and fit a job there, no matters what

LHCb simulation jobs become elastic

- All started as a requirement for BOINC jobs
  - That should be short…
  - Realized that we could apply this everywhere
- All simulation jobs request (in the JDL) a short CPUTime
  - Enough to produce 25 events
  - So, matched easily “everywhere”
- The actual amount of events produced is determined at run time:
  - Given the CPUwork available and CPUe (CPU-work, in HS06.s for producing one event), we can compute how many events may be simulated
    - 20% safety margin to cope with uncertainties in CPU power estimate

$$eventsToProduce = \frac{CPUwork}{CPUe} * 0.8$$

$$CPUwork = CPUtime * CPUNormalizationFactor$$

Determining CPUe

- Simulation productions go through a testing phase before being submitted
- A limited amount of jobs are created and submitted to a site chosen for testing
  - □ each job produces a fixed amount of events

Mgmt of sim prods in LHCb:
G.Corti, Track 2, Thu, 9:15

- Productions undergoing a testing phase are monitored by a dedicated agent
  - □ when all jobs are finished, an evaluation takes place:
    - If all jobs failed, the production request is rejected
    - If jobs are successful, we are evaluate CPUe
      - Job description is modified: CPUe is added, destination is changed…
- Simulation productions have to produce at least a requested amount of events
  - □ In case not enough events have been produced, simulation productions are automatically extended

Determining CPUNormalizationFactor

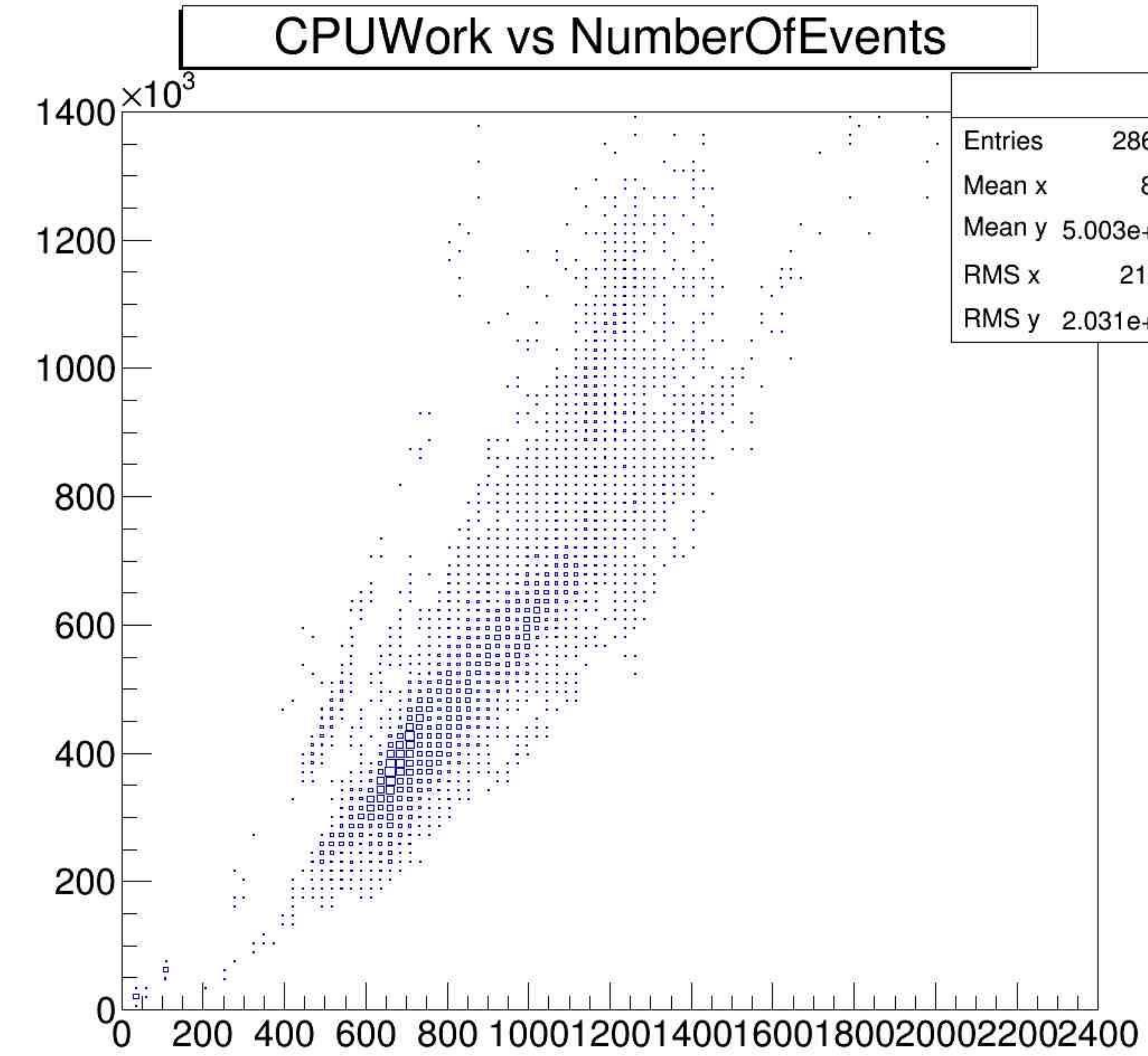
- CPUNormalizationFactor = power of the machine
- Highly discussed subject
  - Power of the machine is not deterministic (HT, power saving, BIOS settings…)
- LHCb (DIRAC) goes for the simplest solution
  - Every pilot jobs runs a quick benchmark script
    - "Proven" to be rather reliable
  - So, we don't use BDII info
    - (also, they are available only for a fraction of resources)
  - We would like to use MachineFeatures
    - And soon we will for those sites that provide it already
      - And at that point we'll have the possibility to make a comparison…

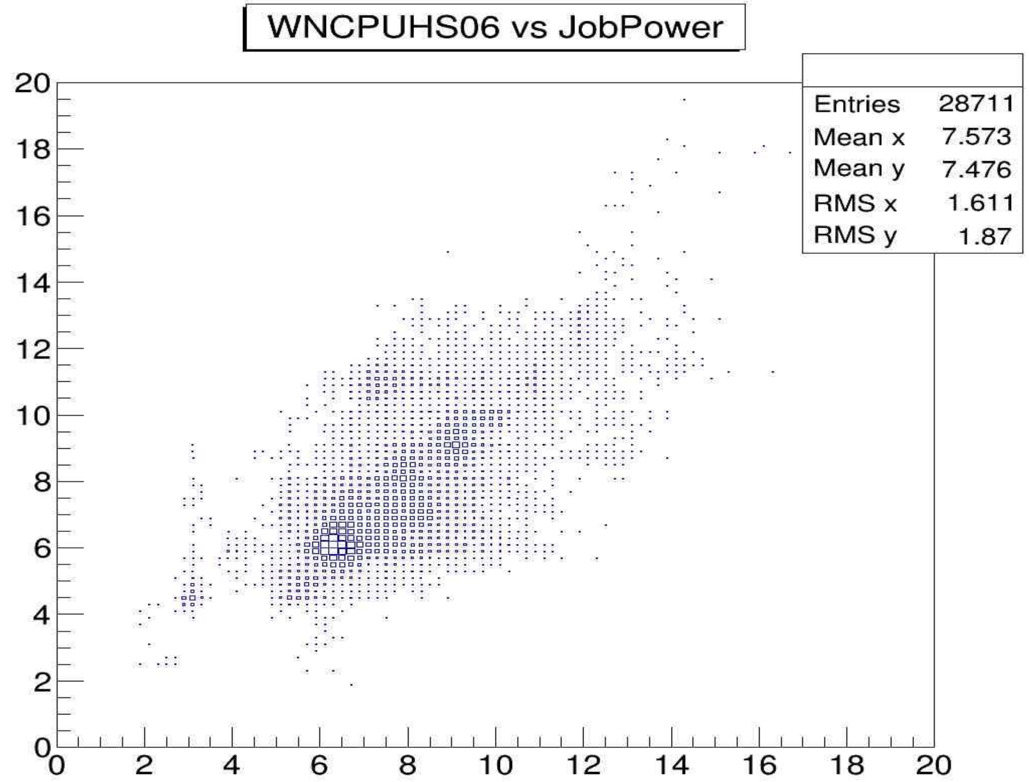
A history-based estimation for LHCb job requirements:
N.Rauschmayr, Track 4, Mon, 14:45

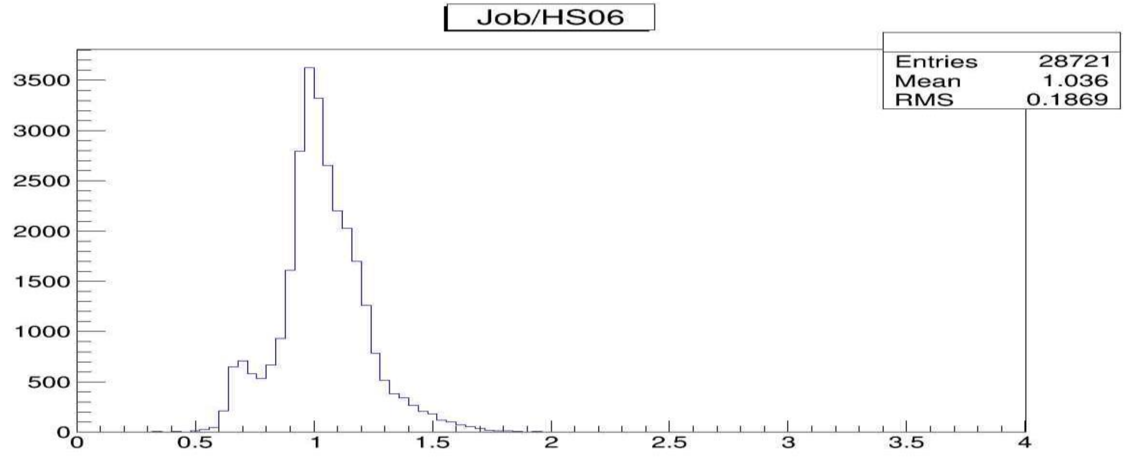
Determining (CPU)Time left

- Pilot checks time left before matching any job
  - Using Job Features where available
    - Right now only in VMs, not completely integrated in the pilot
  - Interrogating directly the batch system
    - if supported by DIRAC

Elastic MC jobs in practice







Summary and prospects

- LHCb simulation jobs are elastic
  - For about a year now
- Simple concept, rather important implications
  - Simulations jobs can run “everywhere”
    - Low priorities, back-filling jobs
- Power calculations non-deterministic

- Next steps
  - Better control of when to end producing.
    - for VM: look for shutdown messages
  - We can send signals to the simulation applications to stop
    - needs anyway to know CPUe (for precise calculations)
  - Run-time adjustment of CPUe

Thanks!

Question, comments

?