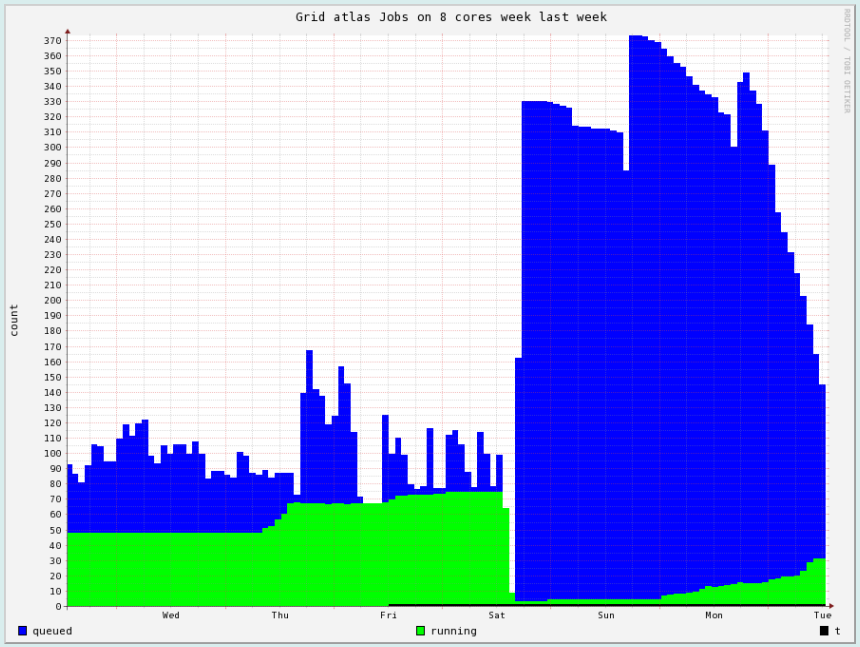
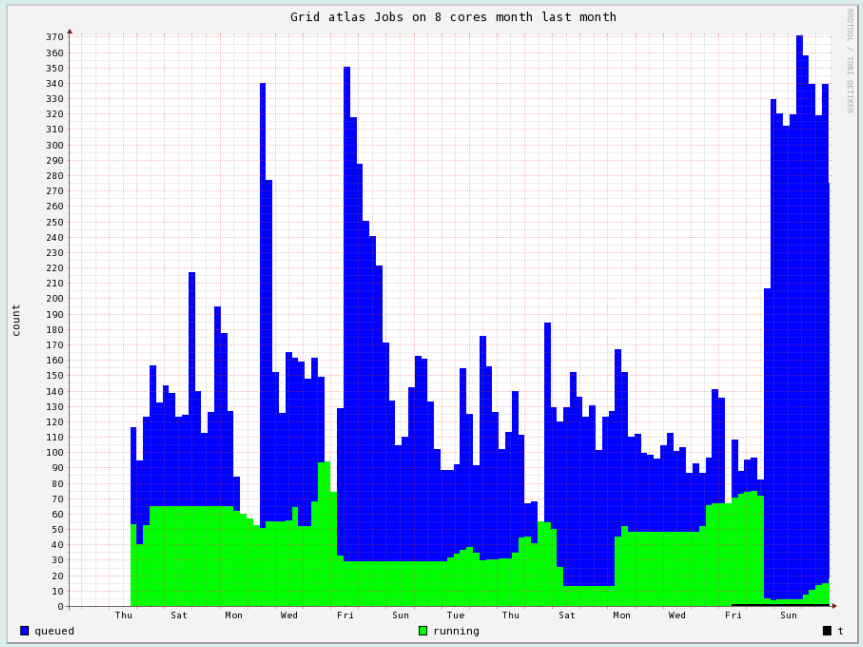
KIT: MCore Status

- batch system: SGE with dynamic PE on one queue
- observed waves of ATLAS multi job slot jobs
- occasionally freed nodes manually

2014.02.10-2014.02.17



2014.01.20-2014.02.17



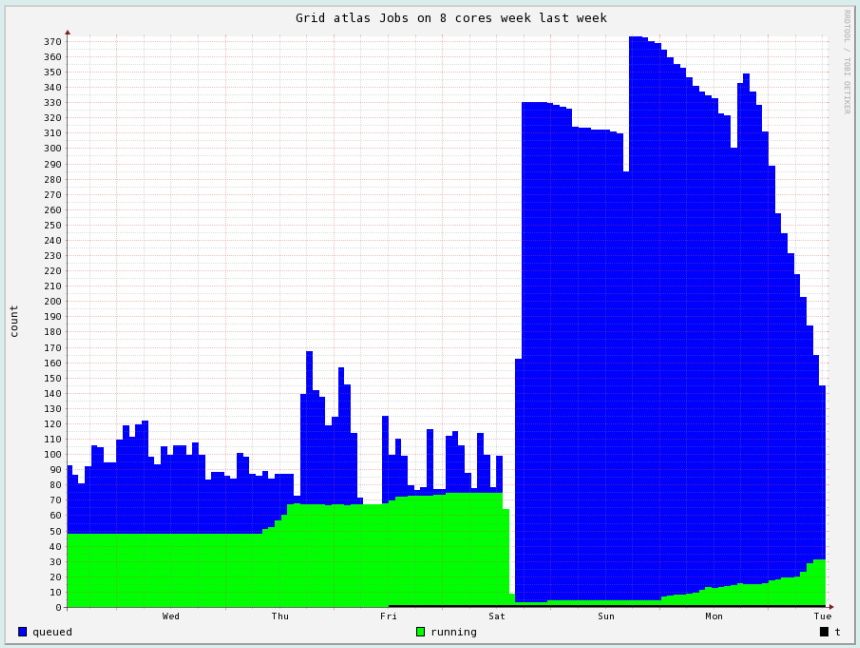
KIT: MCore Status

- getting experiences with job slot reservations

$\rightsquigarrow$ need more statistics for impact on utilization/efficiency

- what effect on utilization acceptable for ramp-ups?
  (How often will ramp-ups occur? Eff. drops shared between all VOs?)

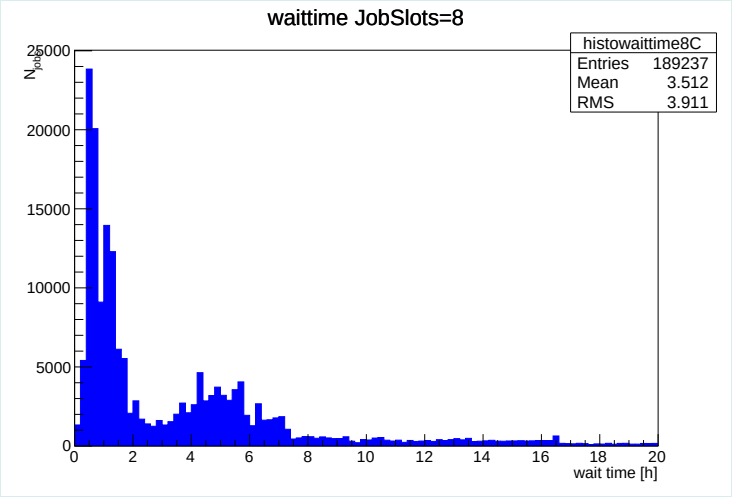
running/queues mcores



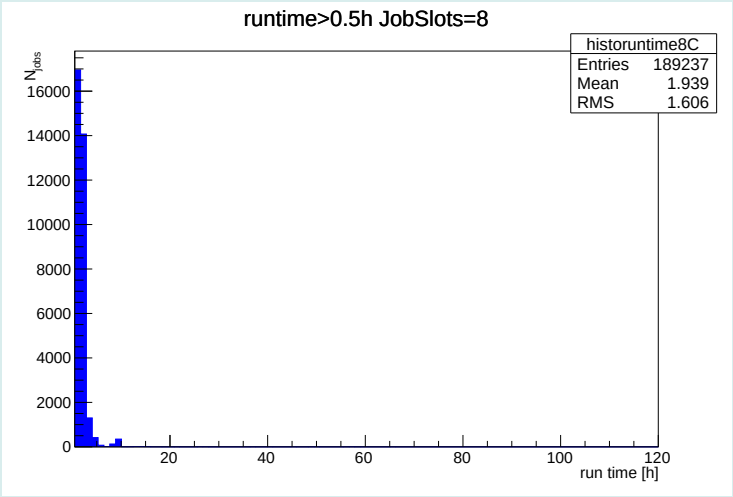
allocated job slots

KIT: MCore 2014. Jan - mid Feb

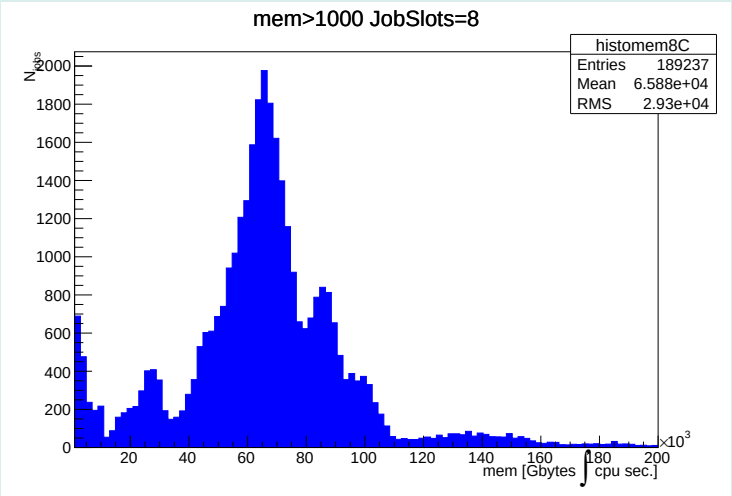
wait time



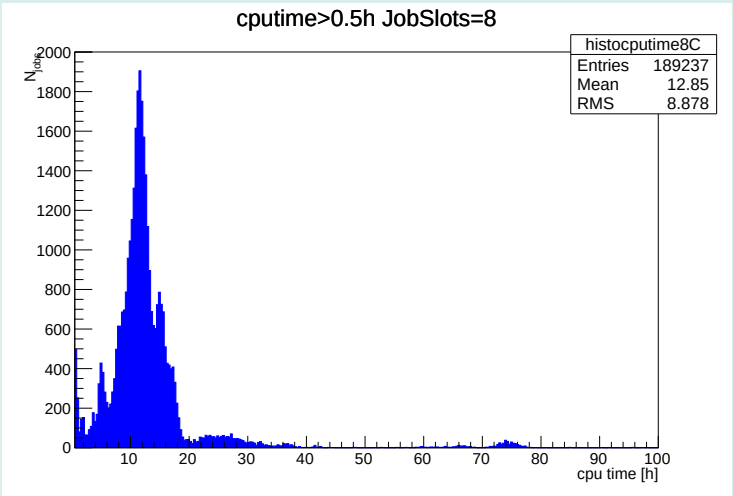
wall time (queue length=120h)



Memory usage

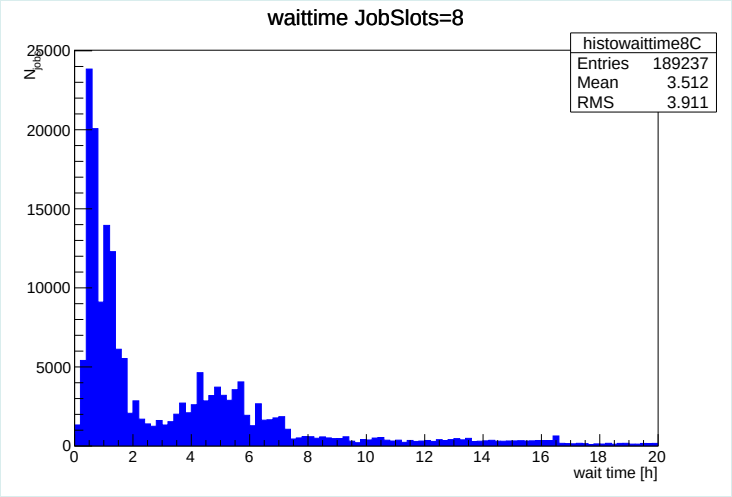


CPU time (∼ ×#Cores)

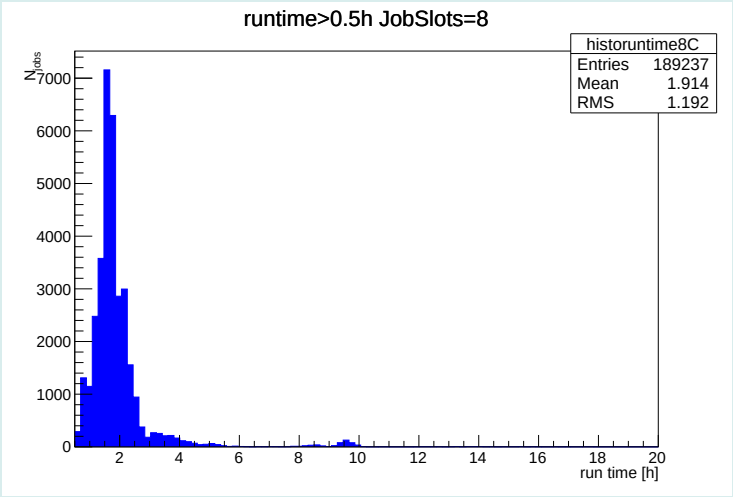


KIT: MCore 2014. Jan - mid Feb

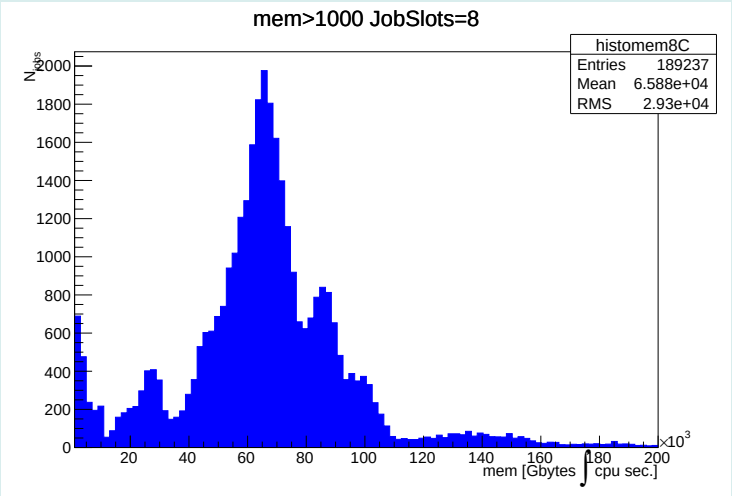
wait time



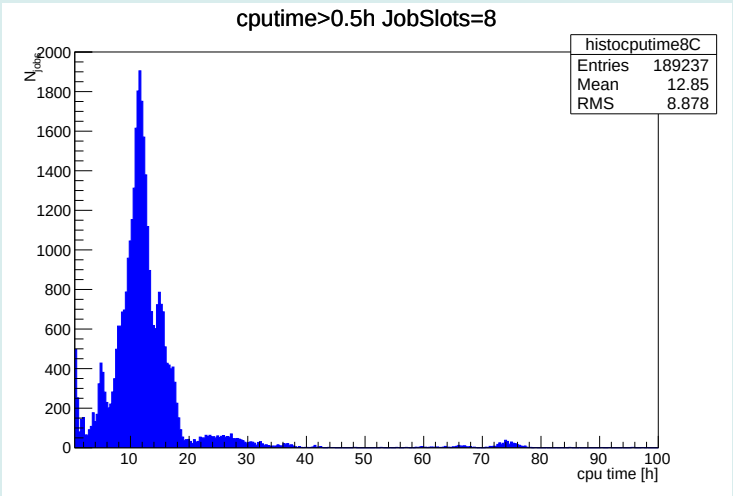
wall time (queue length=120h)(zoom 20h)



Memory usage

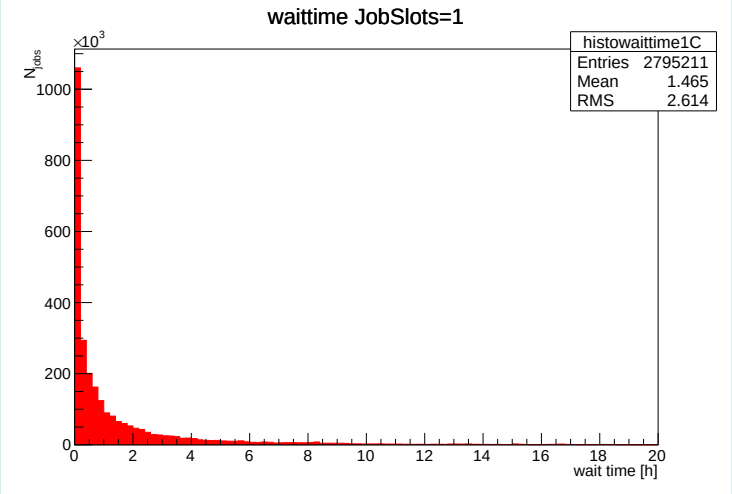


CPU time (∼ ×#Cores)

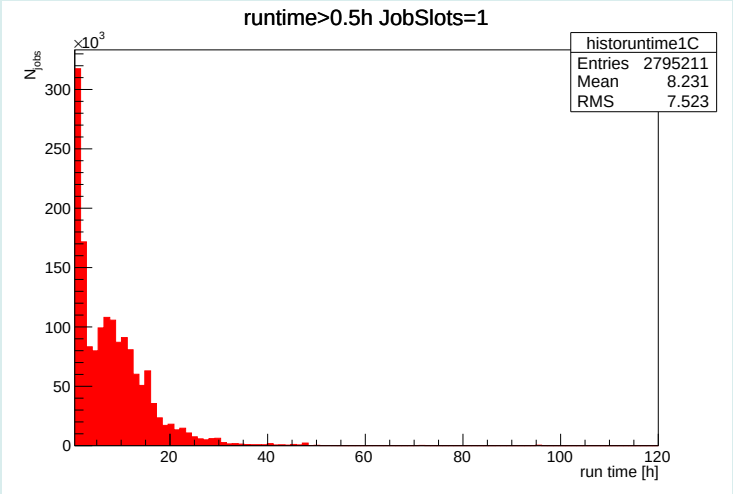


KIT: SingleCore 2014. Jan - mid Feb

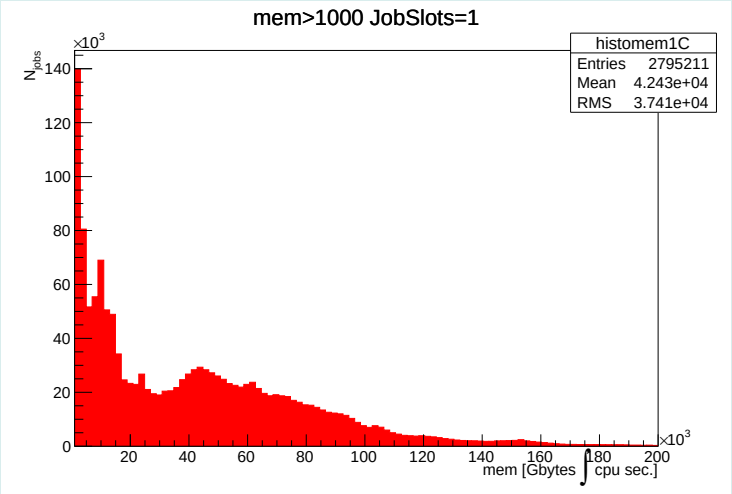
wait time



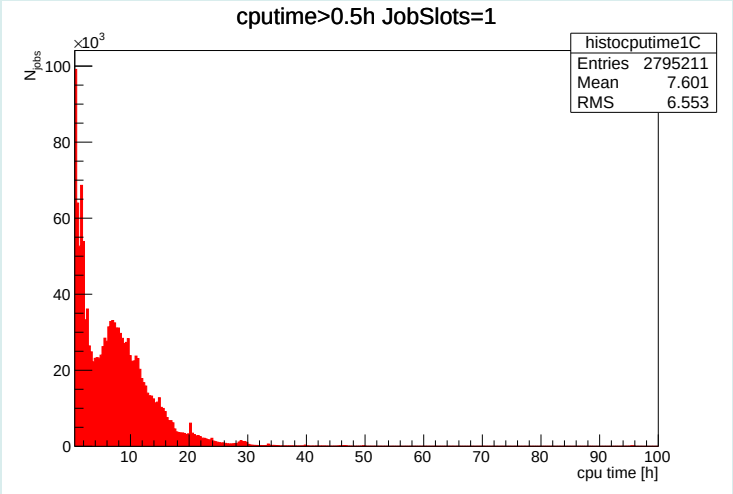
wall time (queue length=120h)



Memory usage

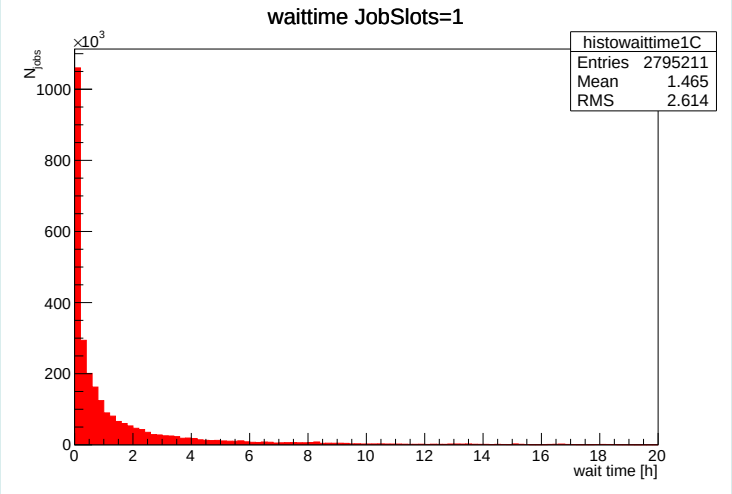


CPU time

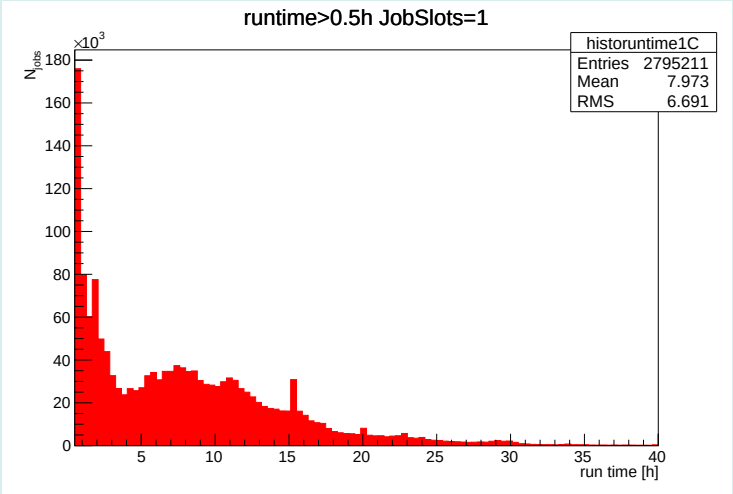


KIT: SingleCore 2014. Jan - mid Feb

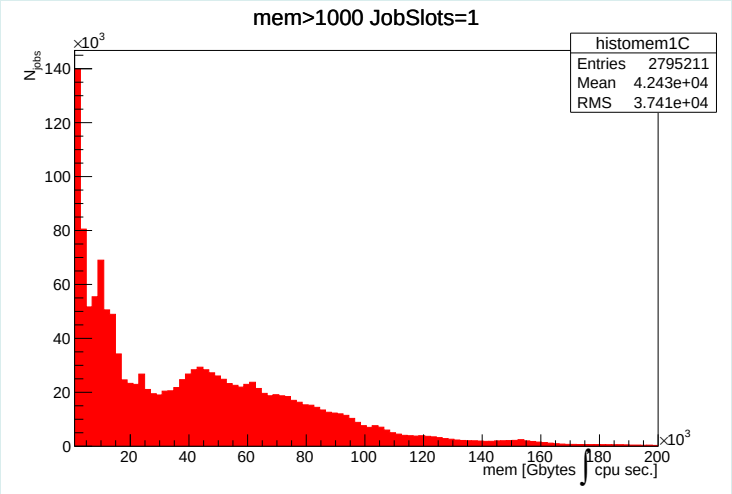
wait time



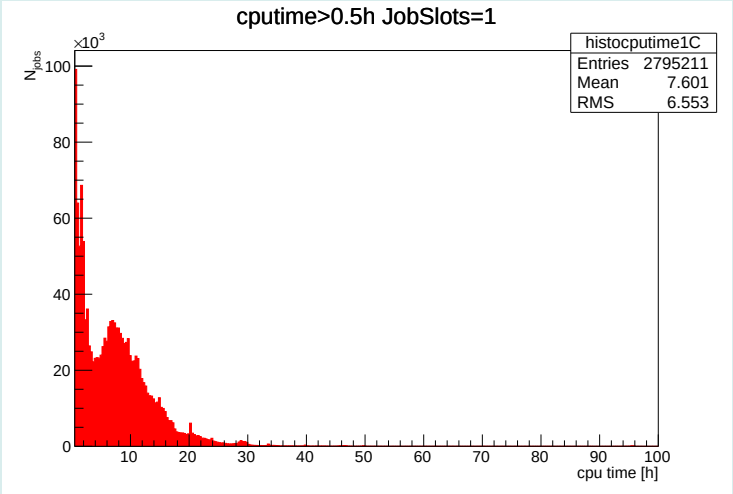
wall time (queue length=120h)(zoom 40h)



Memory usage



CPU time



HPC vs. WNs

- exchanged experiences with HPC team
  - $\rightsquigarrow$ cluster utilization and job wait times
- HPC cluster #1 (2848 cores)
  - for $\text{jobs}_{513..1024\ \text{cores}} \sim \text{cluster size}_{18\%..36\%}$
  - wait time: 10h..150h
  - cluster utilization: 80%..95%
- HPC cluster #2 (6560 cores)
  - for $\text{jobs}_{1025..2048\ \text{cores}} \sim \text{cluster size}_{15\%..31\%}$
  - wait time: 60h..280h
  - cluster utilization: 77%..89%

$\sim$ compare with GridKa WorkerNodes

- $\text{mcore}_{8\ \text{cores}} = 33\%$ on $\text{node}_{24\ \text{Cores}}$
- GridKa WN utilization: 94.5%

Questions/Discussion

- if scheduler going to utilize wall time prediction needs HEP-SPEC06 secs
  - how *exact* have HS06 to be for a VO/for a site?

  $\rightsquigarrow$ HS06 scores are designed to scale with the average performance of typical HEP job mix. Be aware that there is absolutely no warranty that it scales with every individual job!
- how are inefficiencies being accounted?
  - no official requirements on WLCG sites
  - is it reasonable to account a VO when a job's run time deviates by $\times\sigma$ from the prediction and spoiling the scheduler?
    - currently no wall time prediction provided per WLCG job
    - no duty of VOs to supply wall time predictions precise enough to avoid gaps/optimize scheduling
  - it is accountable to the VO submitting a mcore job?
  - is it solemnly a site issue?
- how large is the effect in the end?
  - how many ramp-up periods for how long
  - with steady stream of mcores negligible after x?

Questions/Discussion

- system states with efficient utilization of bare metal?
  - high entropy: many(?) short(m, h, ?) jobs filling free slots
    - what max. wall time prediction variance necessary for good scheduling? (necessary at all?)
  - low entropy: long(h, d, ?) with accurate wall time estimation
    - what max. wall time prediction variance necessary for *good* scheduling?
  - sites with mixed VO users
    - stable mixed state possible?
    - or implicit/explicit segmentation inevitable?

$\rightarrow$ how efficient could an ideal scheduler be under which variances in which state?

i.e. how large would the inefficiency in node utilization become under which variance?

- if the wall time prediction is binned, i.e., in finite numbers of attainable run times/queues, how does the efficiency evolve with the number of prediction time slots?

Addendum

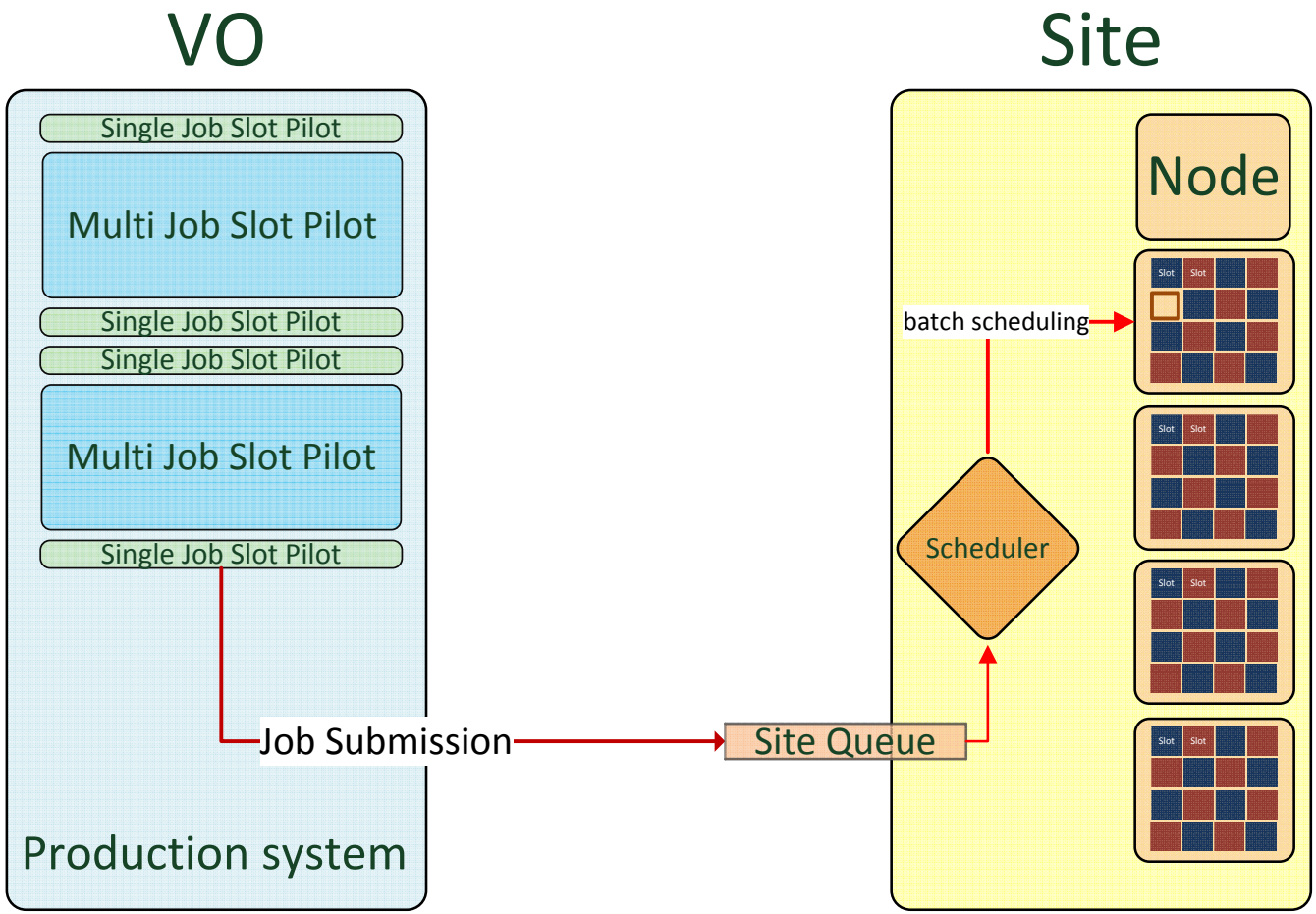
Addendum

some illustrations

(got the impression that sometimes discussions were going talking at cross purposes)

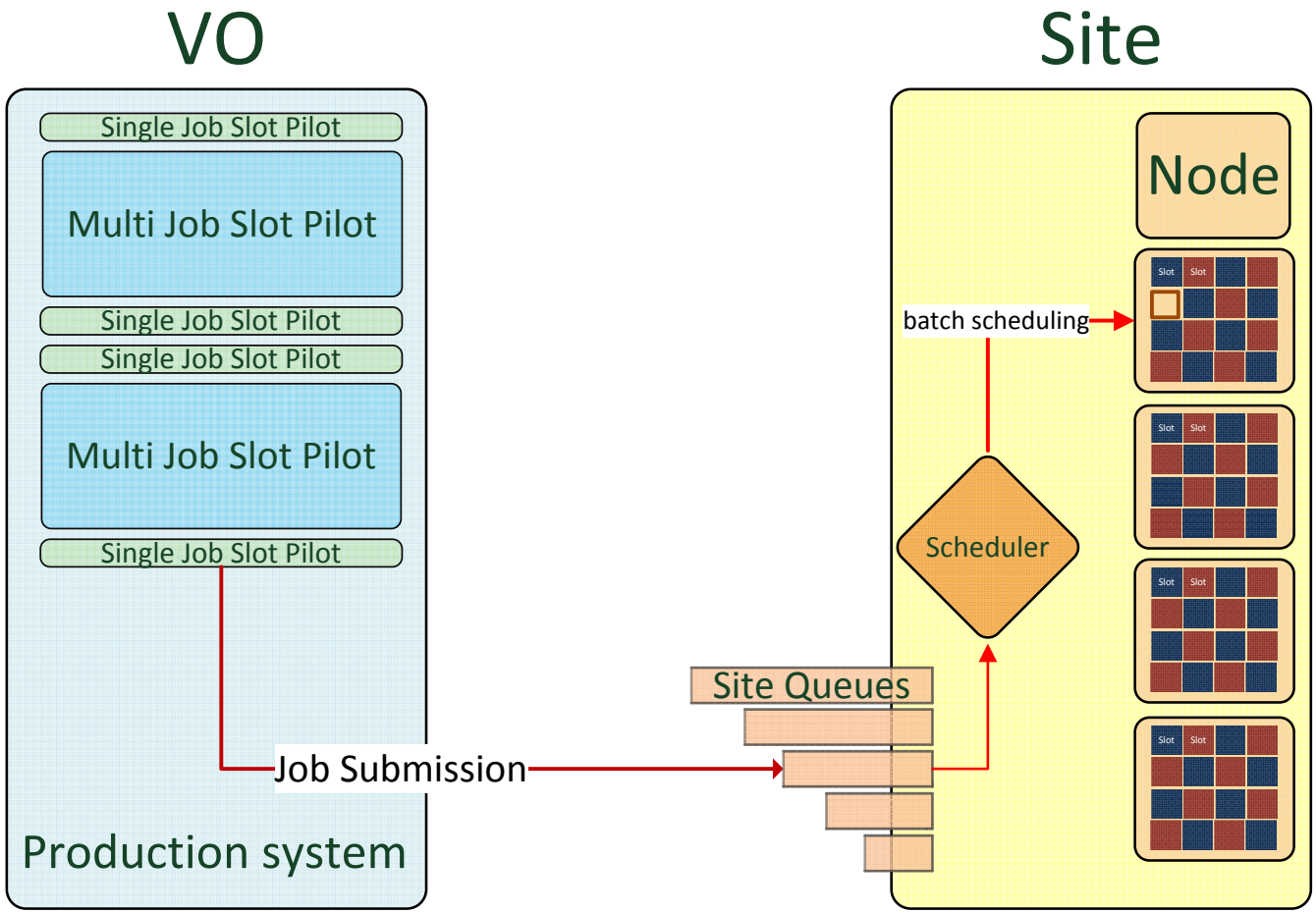
VO-Site job submission

- site setup: one long run queue
  - crucial: batch system efficiency
  - inefficient $\rightsquigarrow$ oscillating fair share adjustment/job allocation



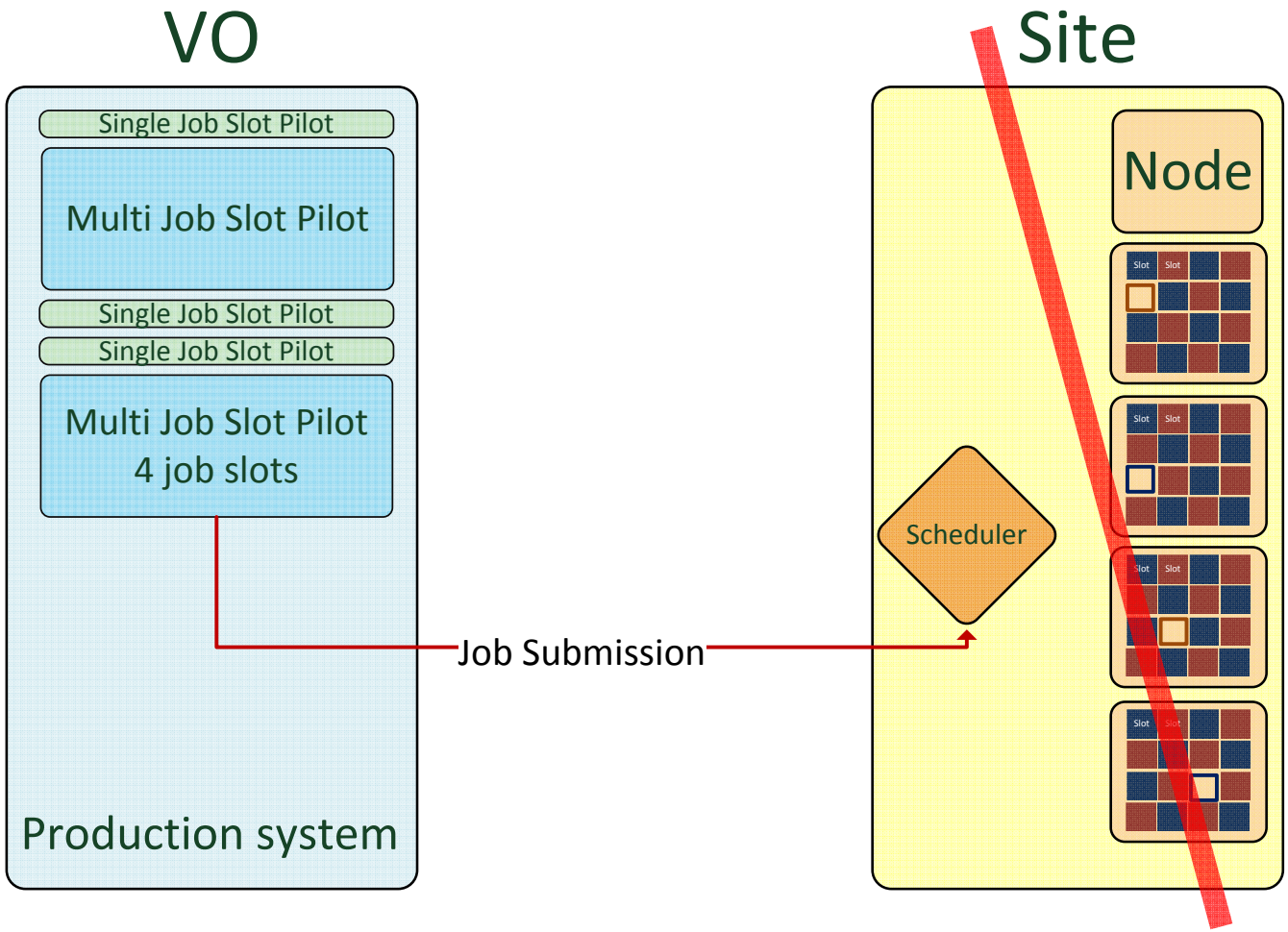
VO-Site job submission

- site setup: multiple queues various length
  - administration effort
  - non-WLCG users?



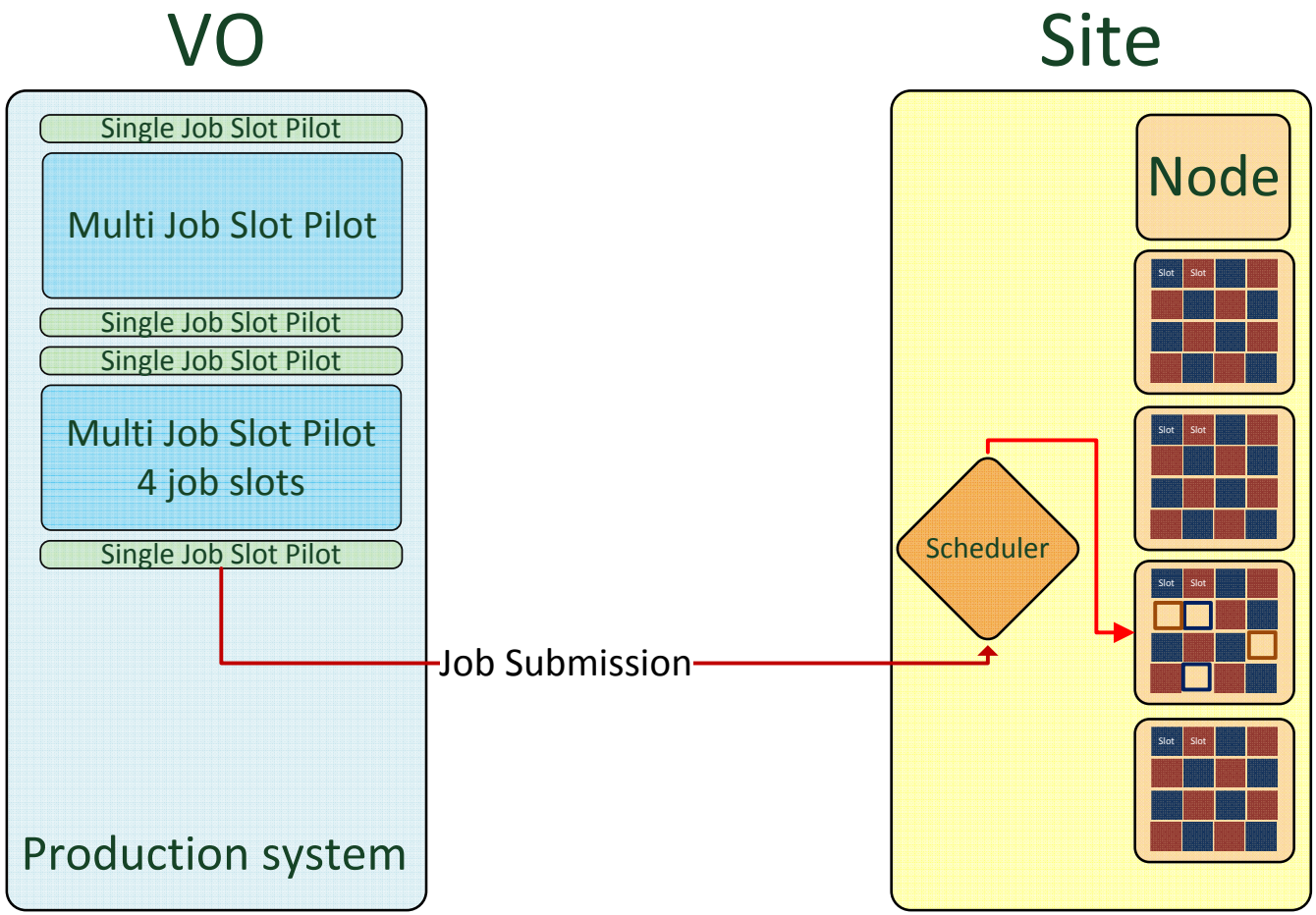
Multi Job Slot Jobs

- request for multiple job slots for one job
  - constraint: congruent job slots



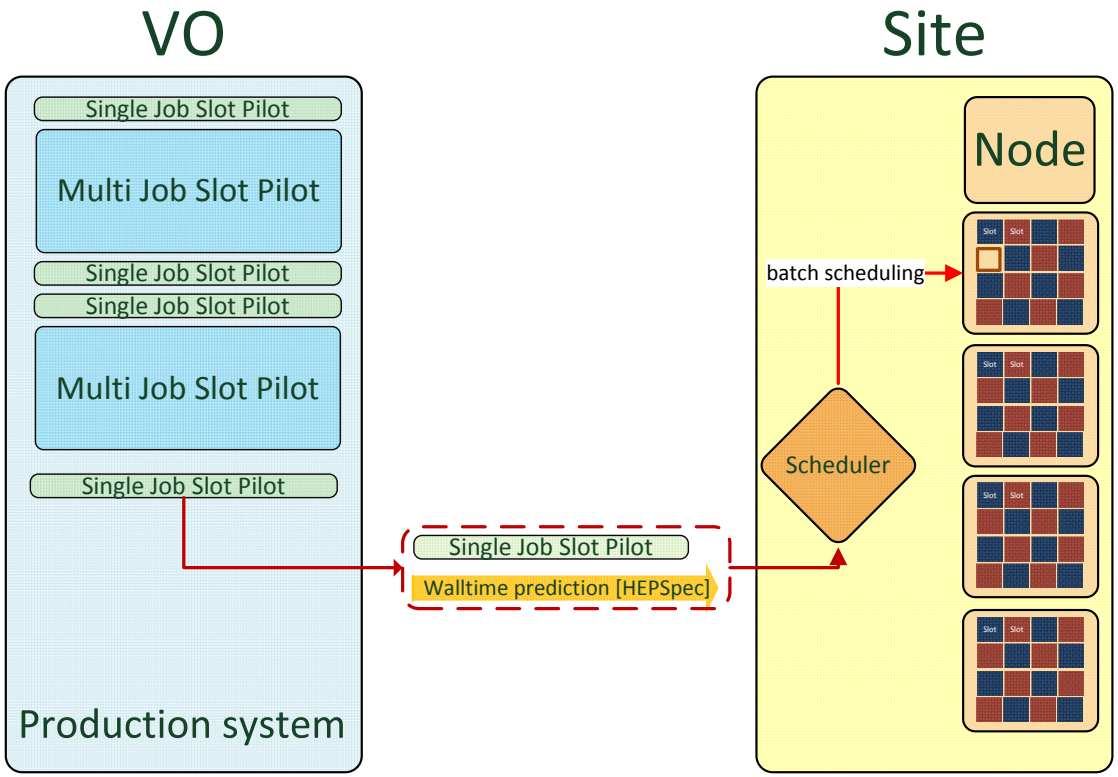
Multi Job Slot Jobs

- all job slots on one node
- farm $\neq$ HPC cluster



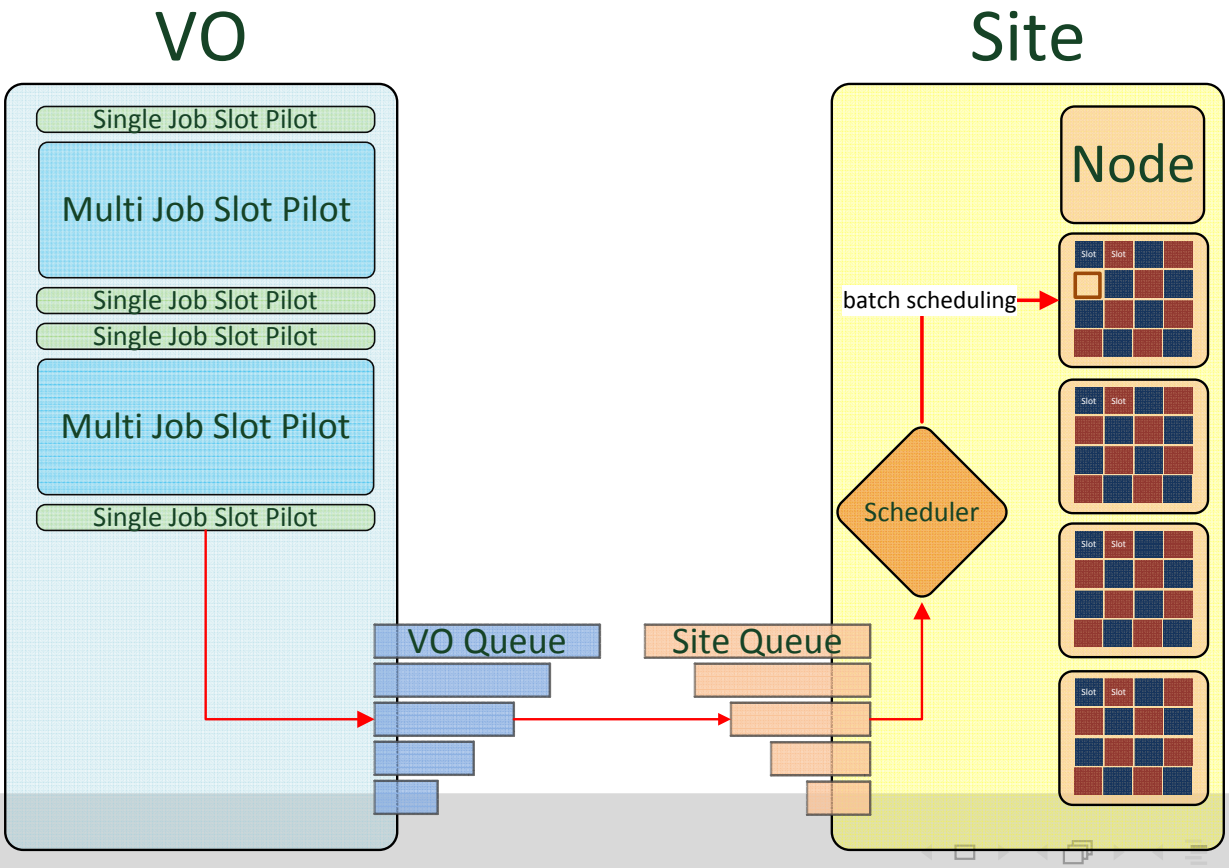
Job scheduling

- efficient scheduling [valid for scheduling either at site and at VO]
  - → crucial: reliable wall time prediction for job
- scheduling at site
  - would need to scale wall time in HS06 seconds!
  - BDII published HEP-SPEC06 accurate enough?



Scheduling@Site: *prediction emulation*

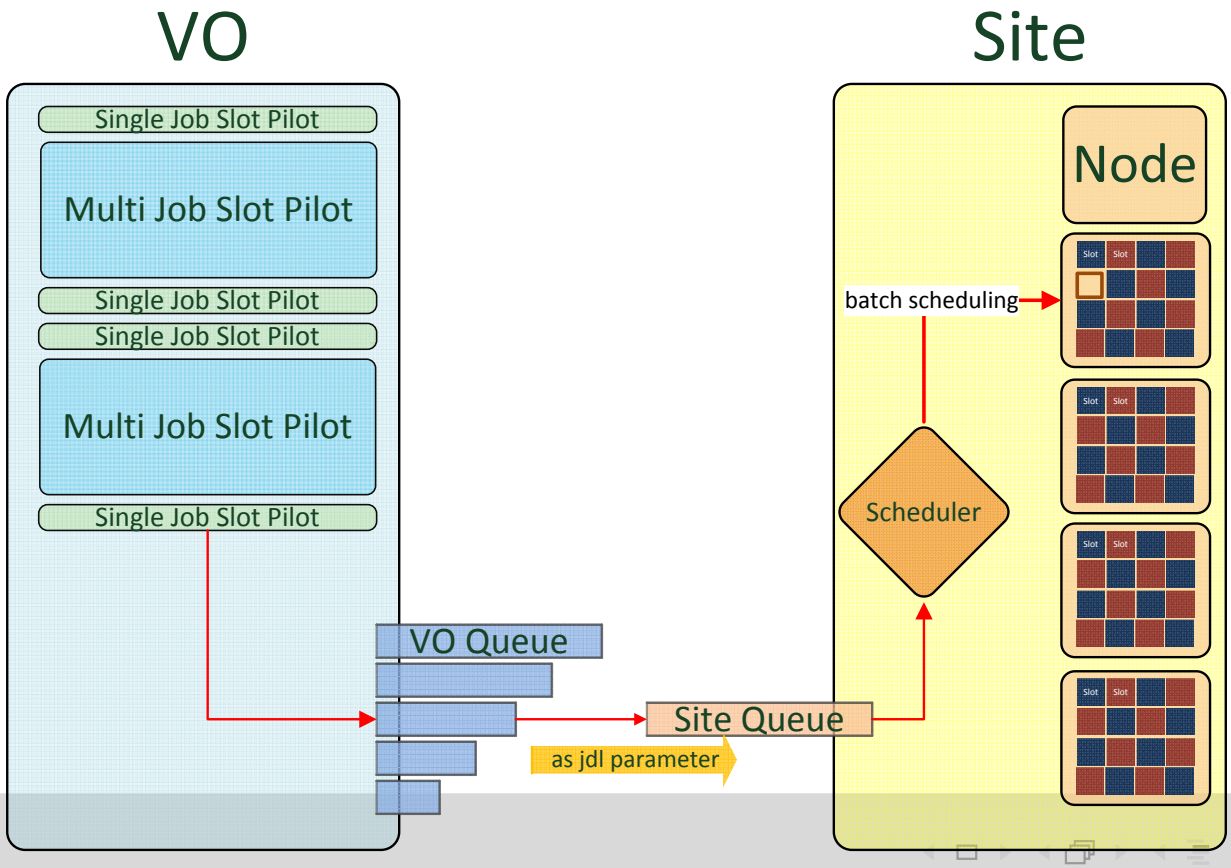
- ⇝ no per job wall time prediction
  - scheduling@site via queue length
    - ≈ more coarse wall time prediction ↔ multiple queues with increasing length
      - multiple VO and site queues



Scheduling@Site: *prediction emulation*

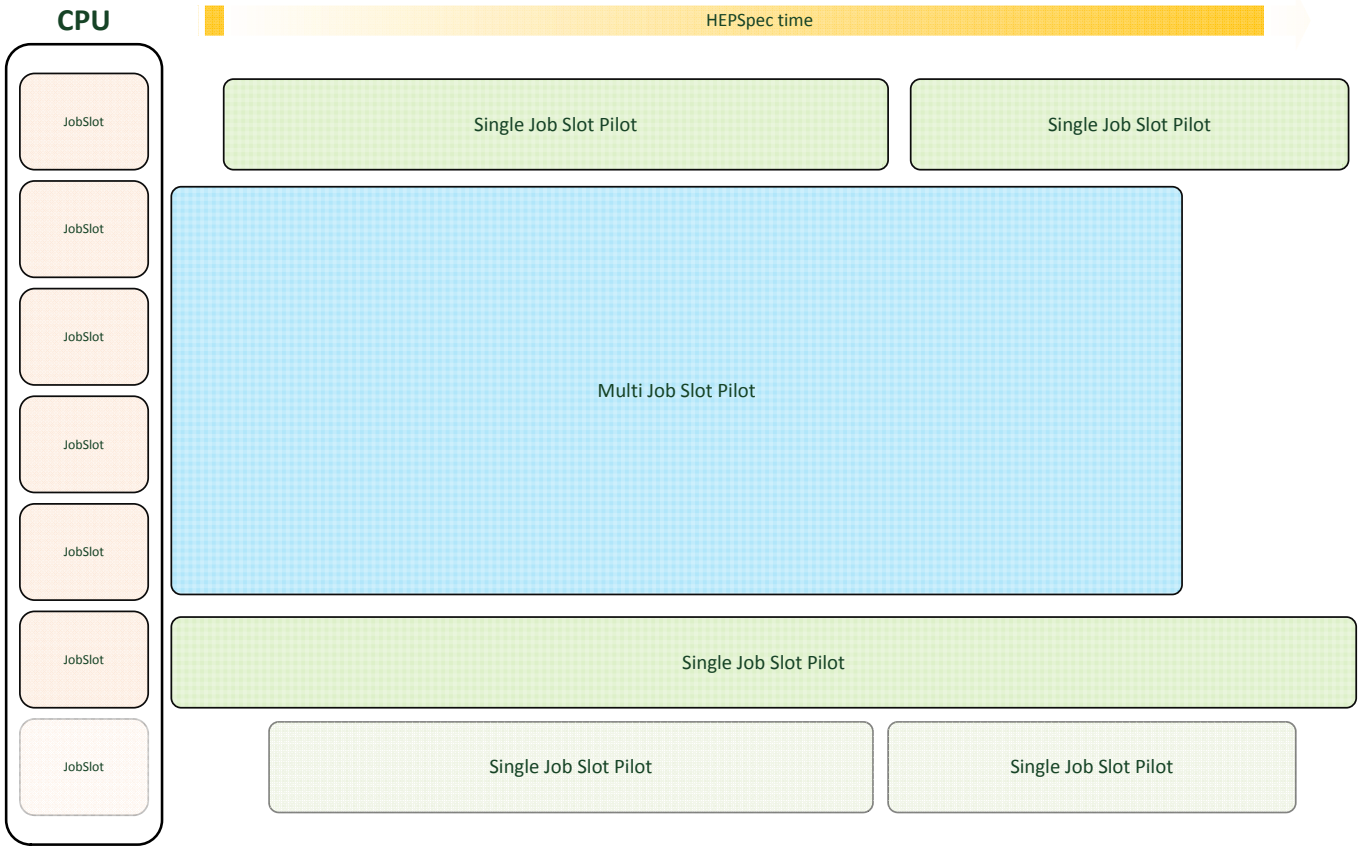
Karlsruhe Institute of Technology

- $\rightsquigarrow$ no per job wall time prediction
  - scheduling@site via queue length
    - $\approx$ more coarse wall time prediction $\leftrightarrow$ multiple queues with increasing length
      - multiple VO queues to job parameter translation

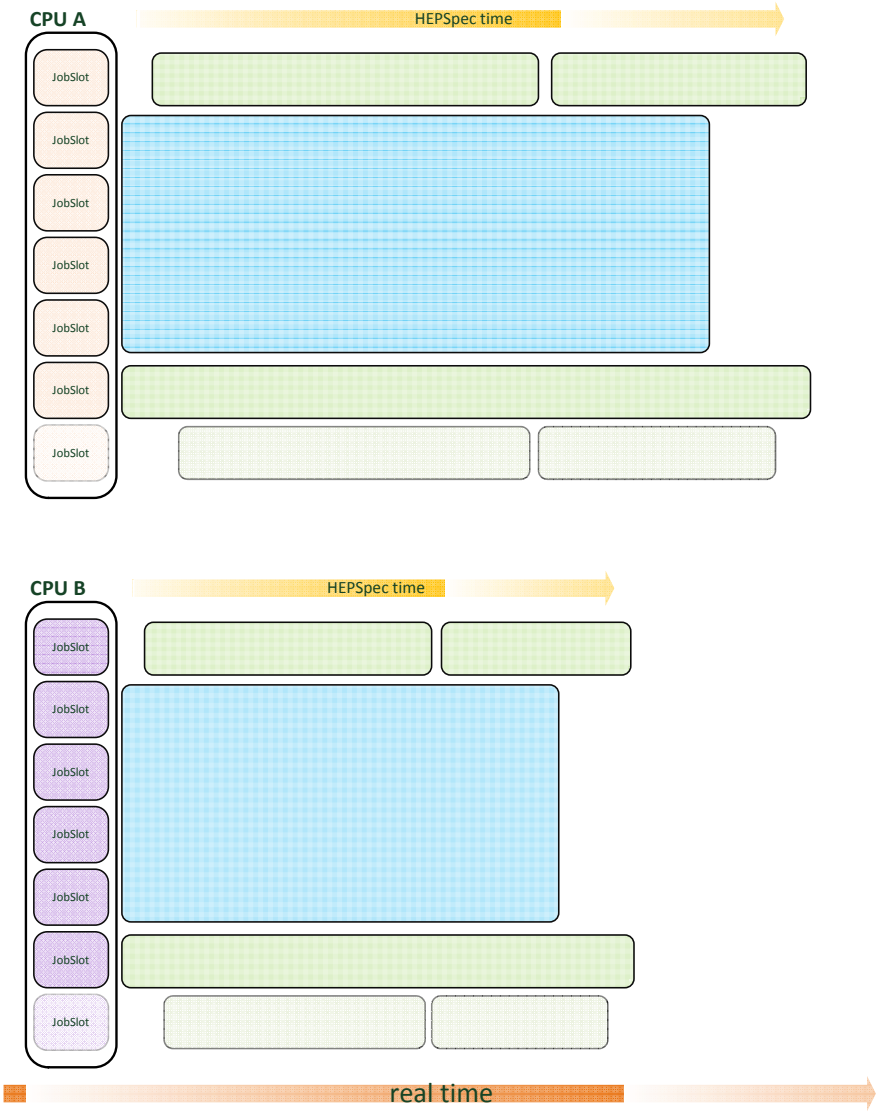


Node view of single & multi job slot jobs

- CPUs with n job slots
- multiple single & multi job slot jobs from different VOs
- time in measured in HEP-SPEC06 sec.!



Real time vs HS06 time

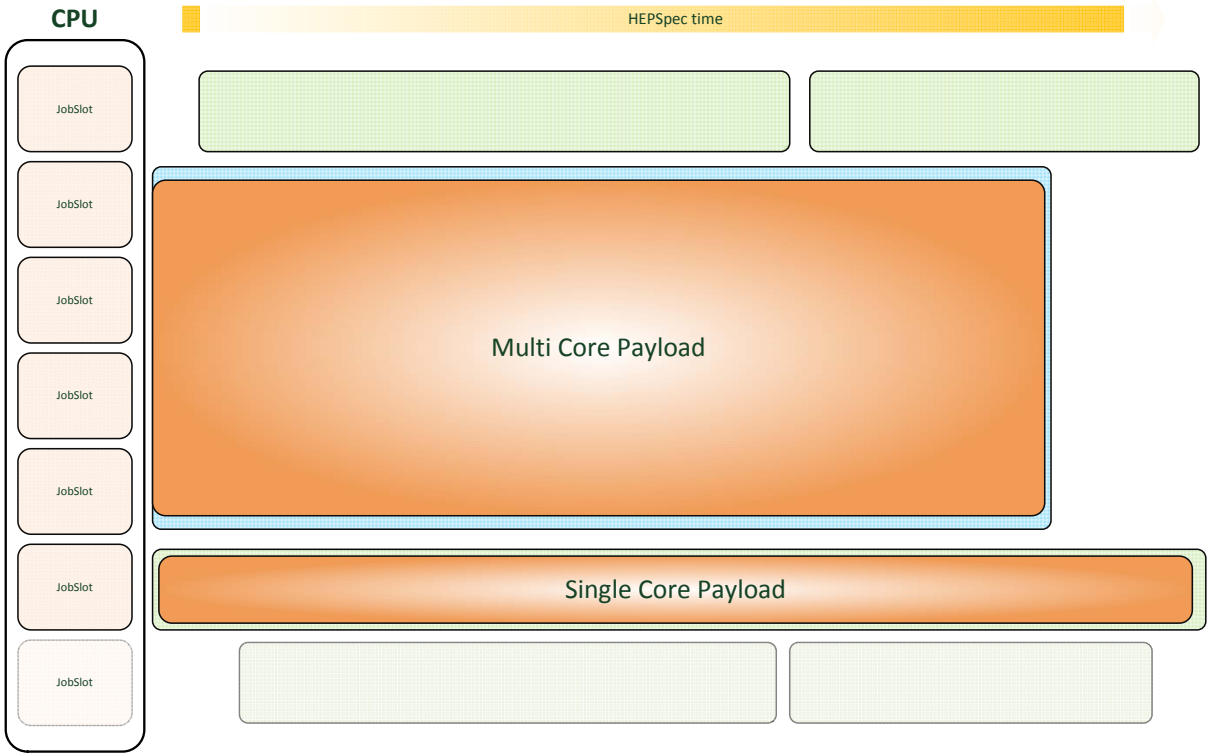


- wall time etc in HS06 sec.
- batch system would need to scale according to HS06

$\rightsquigarrow$ job wall time predictions in HS06

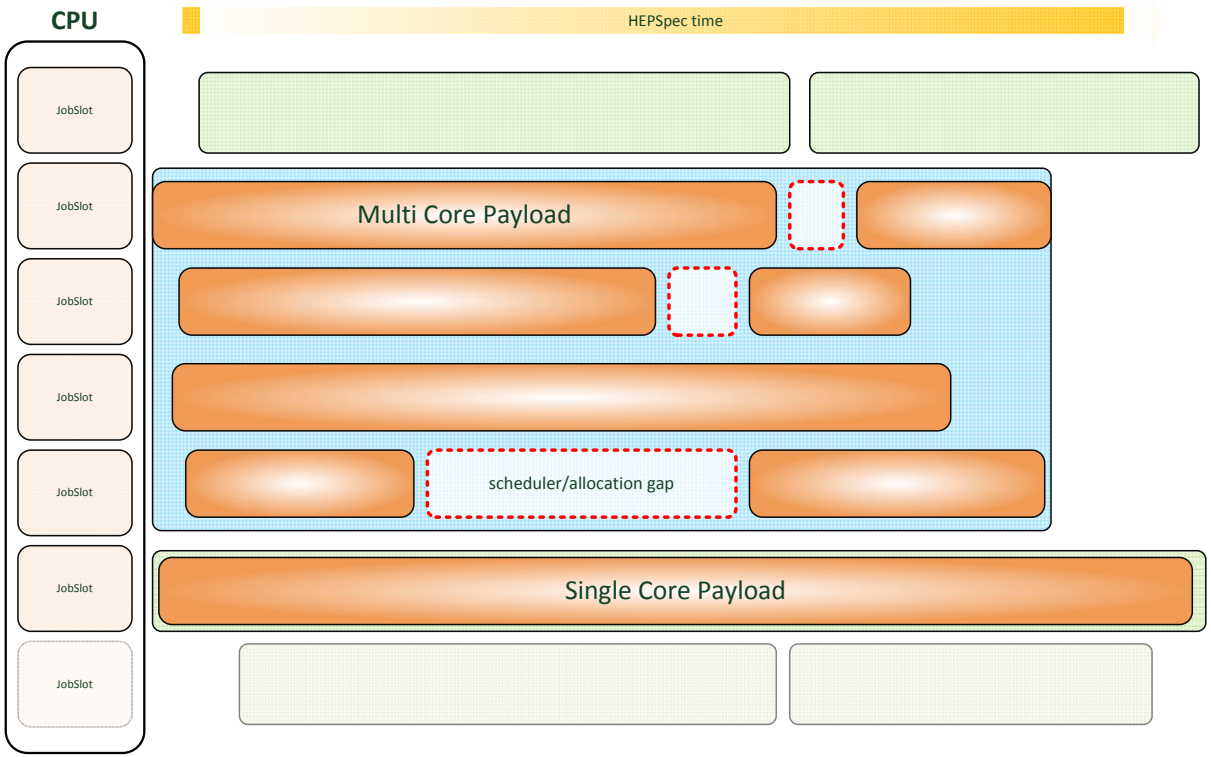
Congruent Payloads

- Congruent Payloads
- one payload per single job slot job
- one payload per multi job slot job



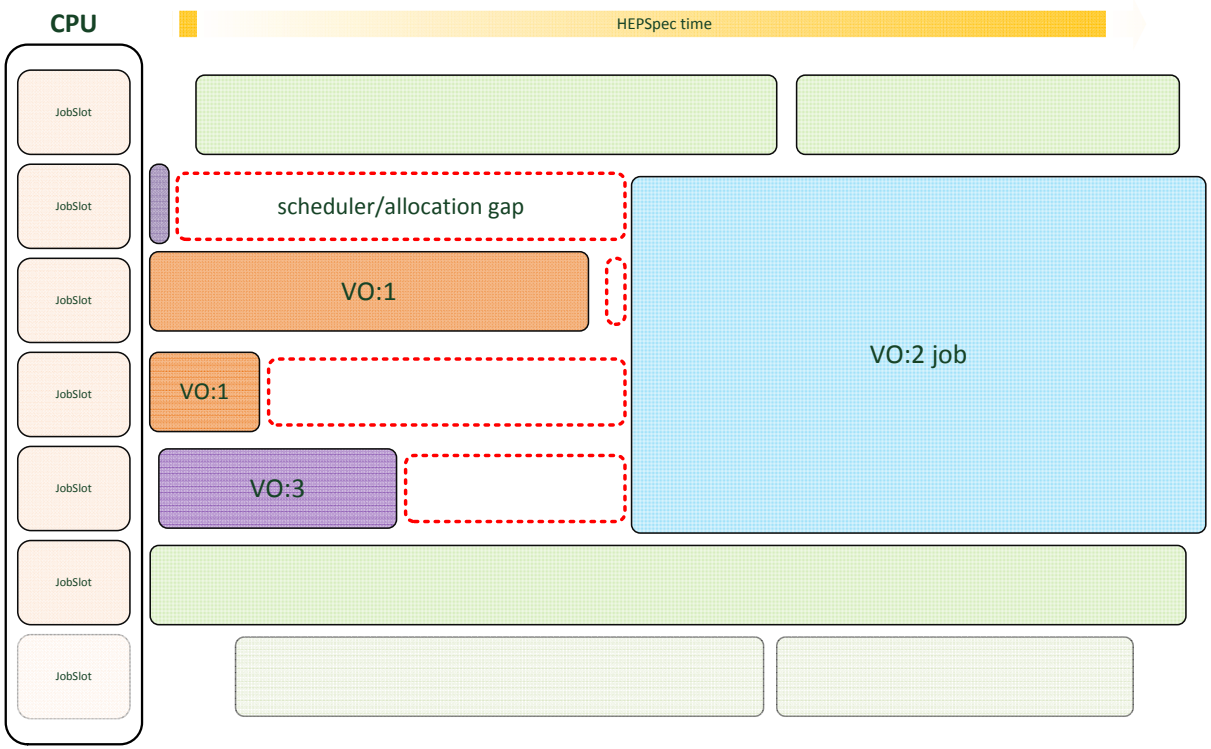
VO Scheduler: Multiple Payloads per Multi Job Slot Job

- multiple payloads reloaded in multi job slot pilot
- scheduler within pilot
- scheduling/job allocation gaps within VO



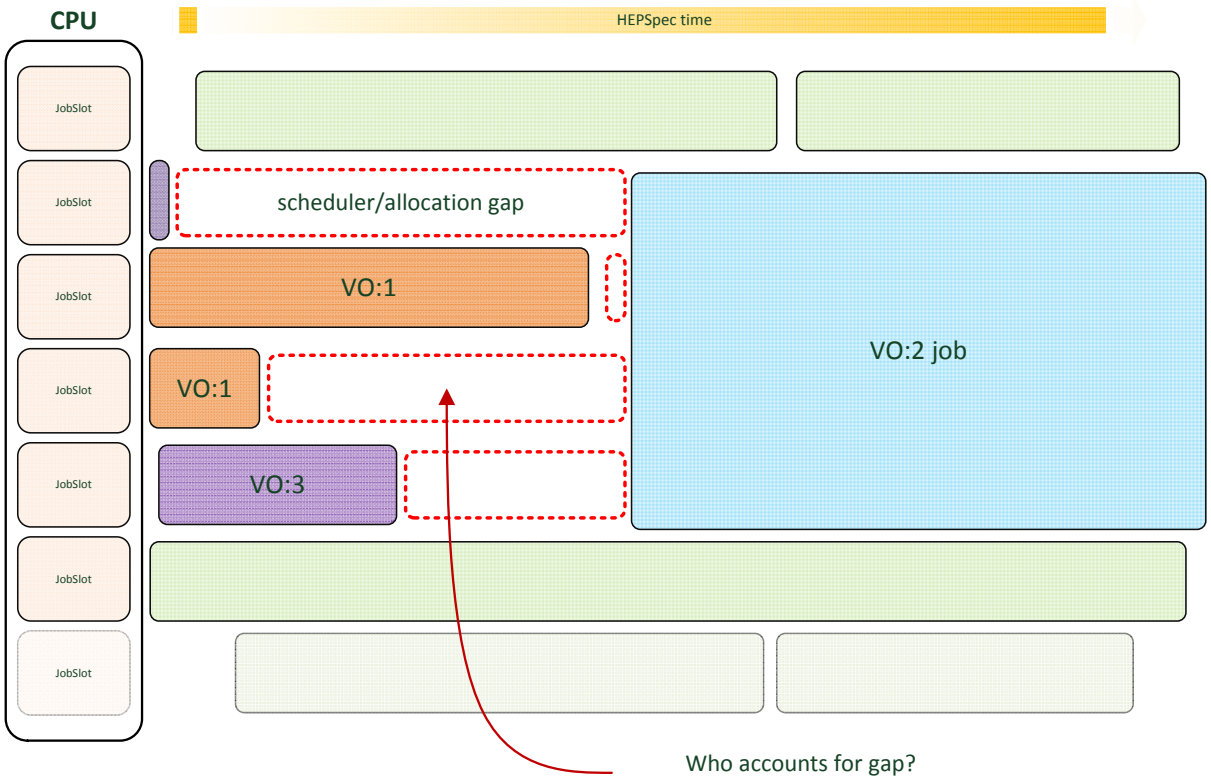
Site Scheduler

- scheduling multi job slot job at site
- applying to both: single/multiple payloads per multi job slot job
- constraint: x *congruent* job slots, i.e., x free slots on a *discrete* node
- for optimal scheduling: scheduler depends on accurate wall time prediction in HS06



Site Scheduler

- who is accounted for scheduling gaps?
  - VO requesting multi job slot resource?
  - *VO of preceding single job slot job?*



Site Scheduler

i.e. how to handle in-accurate wall time predictions

- VO requesting resource for xh while job submission
- actual resource usage is yh $< x$h

- sites with WLCG and non-WLCG users?