Karlsruhe Institute of Technology

On the Quality of Wall Time Estimates for Resource Allocation Prediction

Mehmet Soysal, Marco Berghoff, Dalibor Klucasek and Achim Streit

Steinbuch Centre for Computing (SCC) / Scientific Computing and Simulation (SCS)

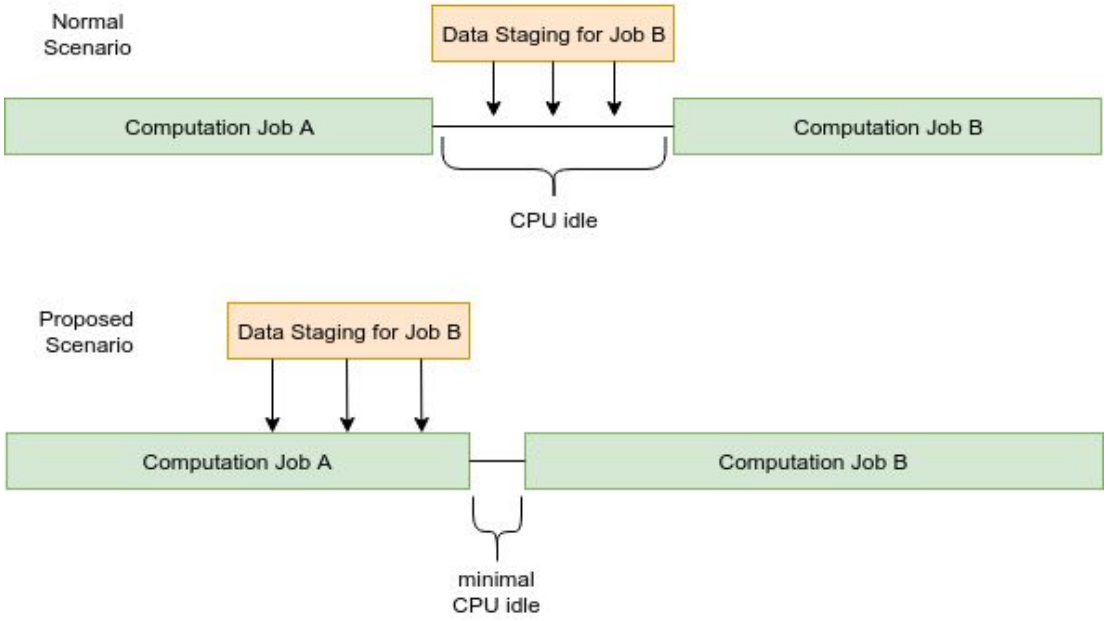
Overview

- Motivation
- Problem
- Metric
- Results
- Conclusion & take away

Motivation

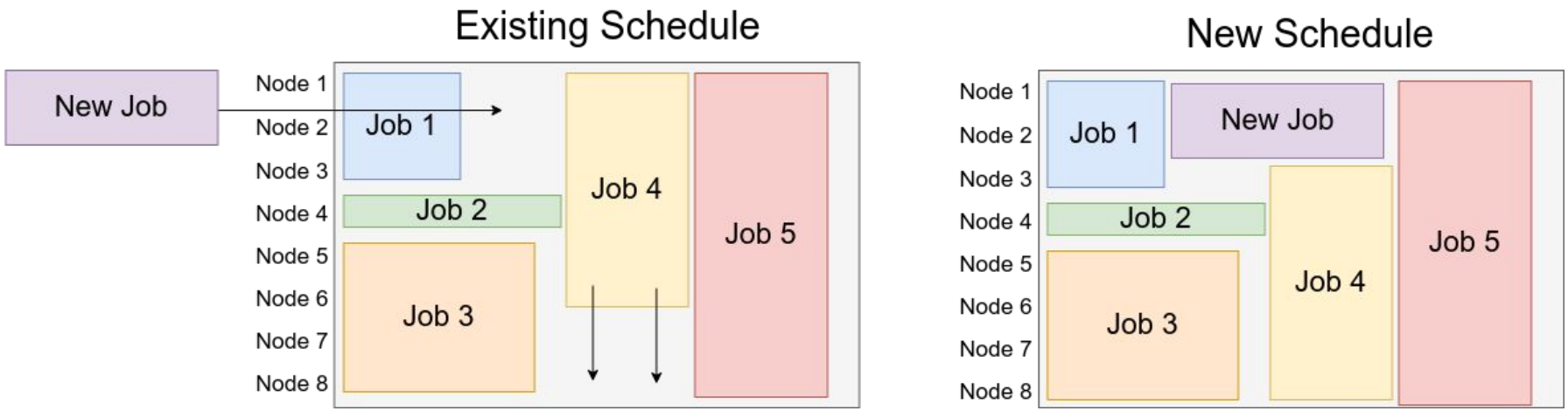
- The I/O Subsystem (parallel FS) is a bottleneck in HPC Systems
  - Bandwidth, metadata or latency
- Data Staging in advance to compute node
- Which nodes are going to be allocated?
- Wall times are far away from optimal
- How good wall time predictions have to be?

Goal: Data staging in advance



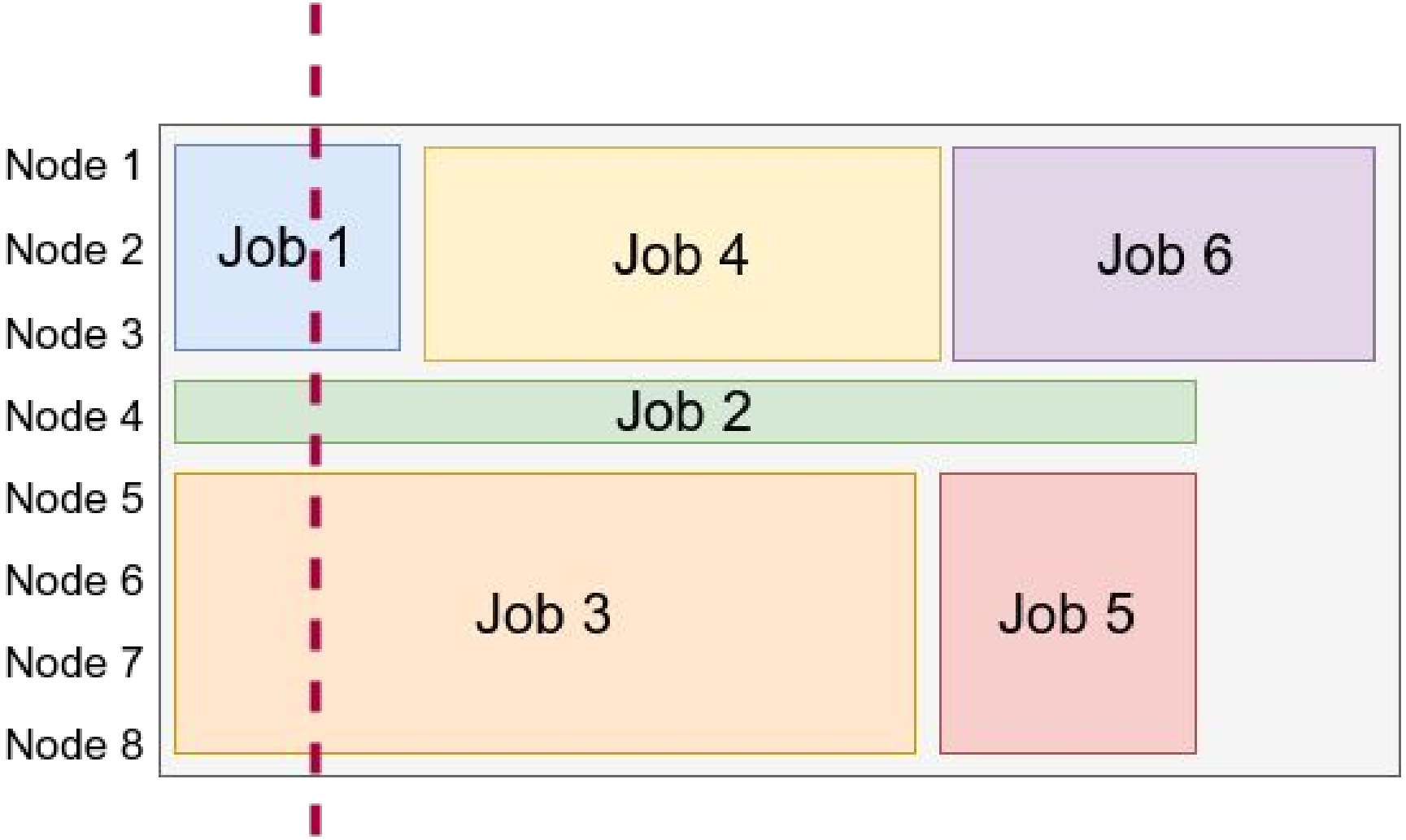
- Based on the allocation prediction
- No modification on scheduling behaviour
- How reliable is the schedule?

Scenario 1: Backfill-Scheduling

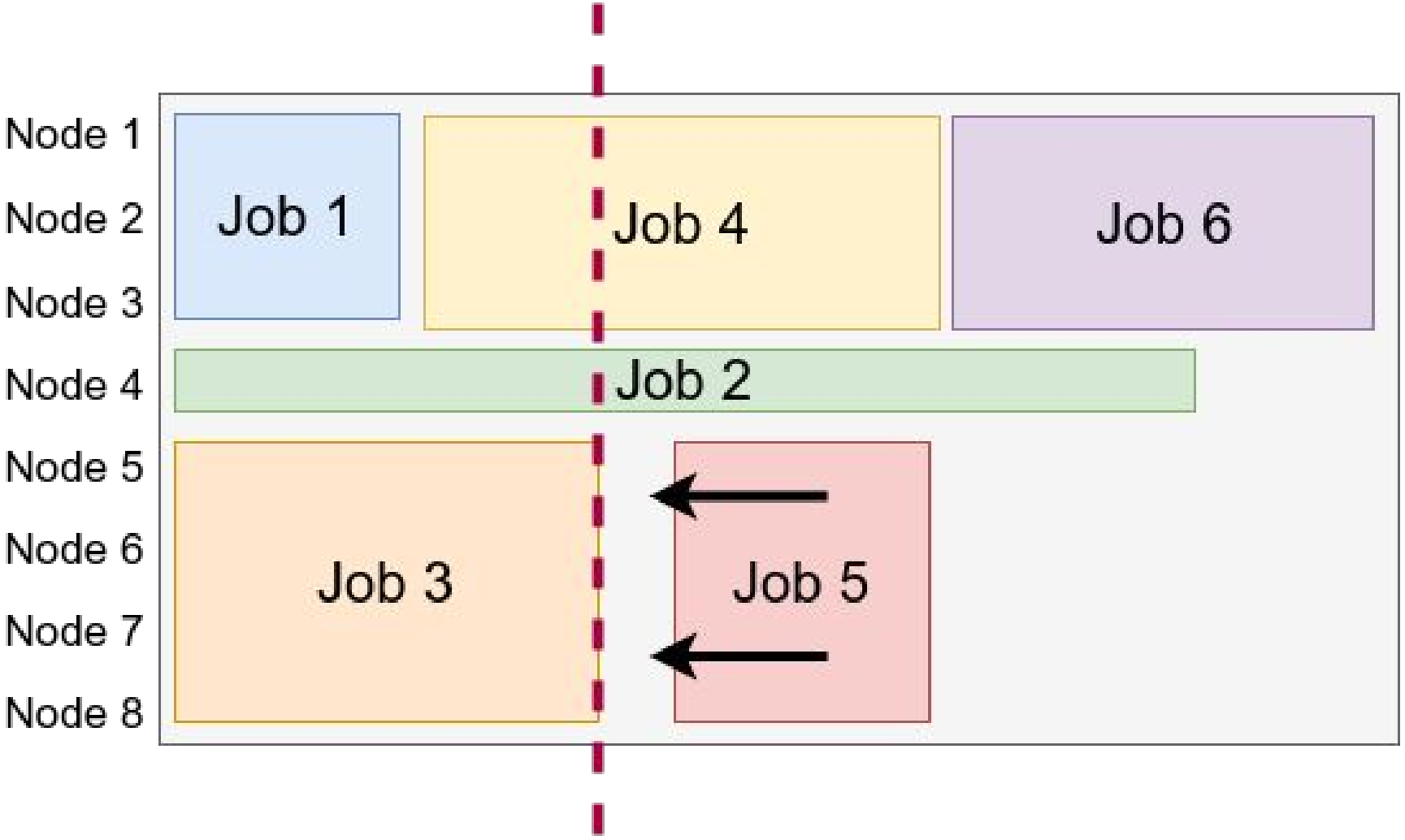


- Backfilling re-shuffles planned schedule

Scenario 2: Forward jump in schedule



Jump forward (2)



How to solve

- Many other cases cause reschedule
  - Node failure
  - Other nodes earlier free
  - High priority jobs
- Need accurate wall time estimates
  - Reduces need for back-filling
  - No jumping forward in schedule
  - Keep cluster utilization high ←
- Many approaches to predict wall time estimates

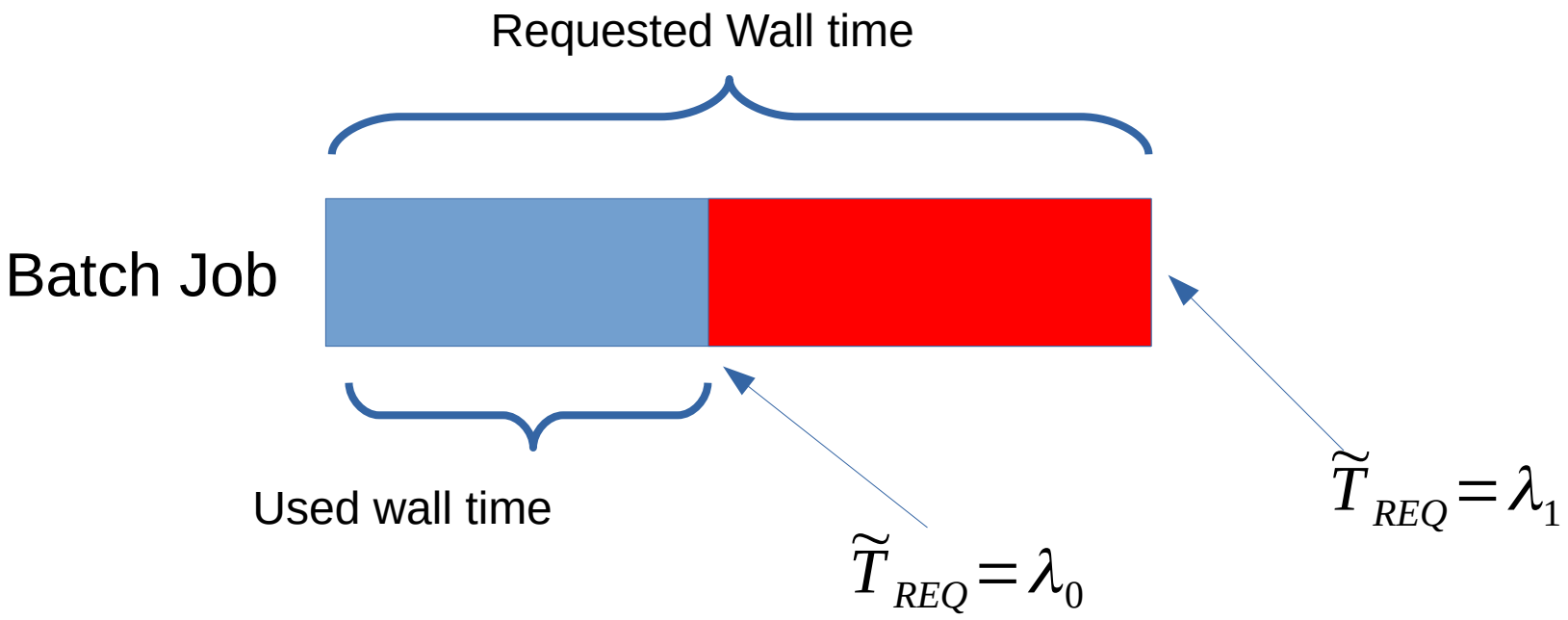
  Simple Rules, Machine Learning (ML), Automatic ML, Deep learning

Evaluation

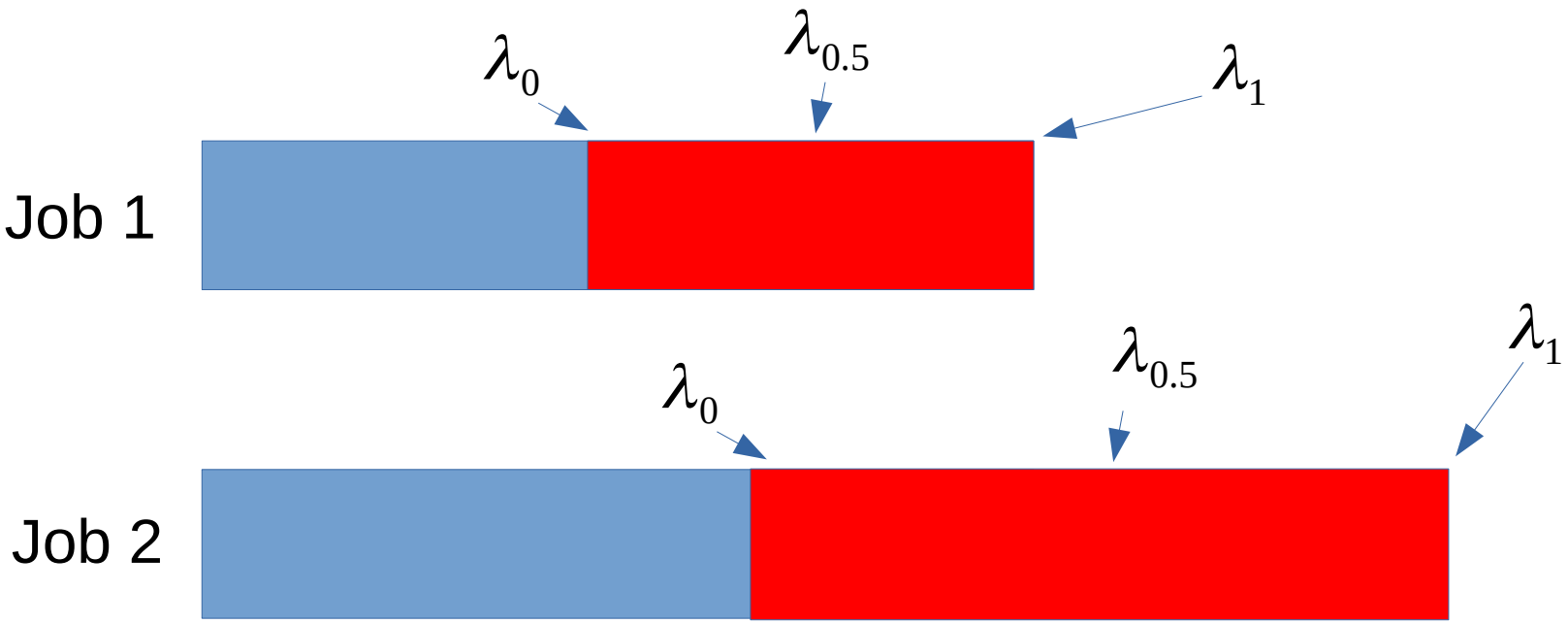
- Not another wall time predictor
  - Impact of accurate wall time on node prediction
- Improve wall times artificially (“redefined” requested wall time)
  - No under-estimations
- Workloads from the parallel workload archiv
  - CTC, SDSC, KTH, ForHLR II*
- ALEA new feature developed
  - Node allocation tracking

Redefined requested wall time (1)

$$\widetilde{T}_{REQ} = T_{REQ} + \lambda_x \left(T_{REQ} - T_{USED} \right), for\ x \in [0,1]$$

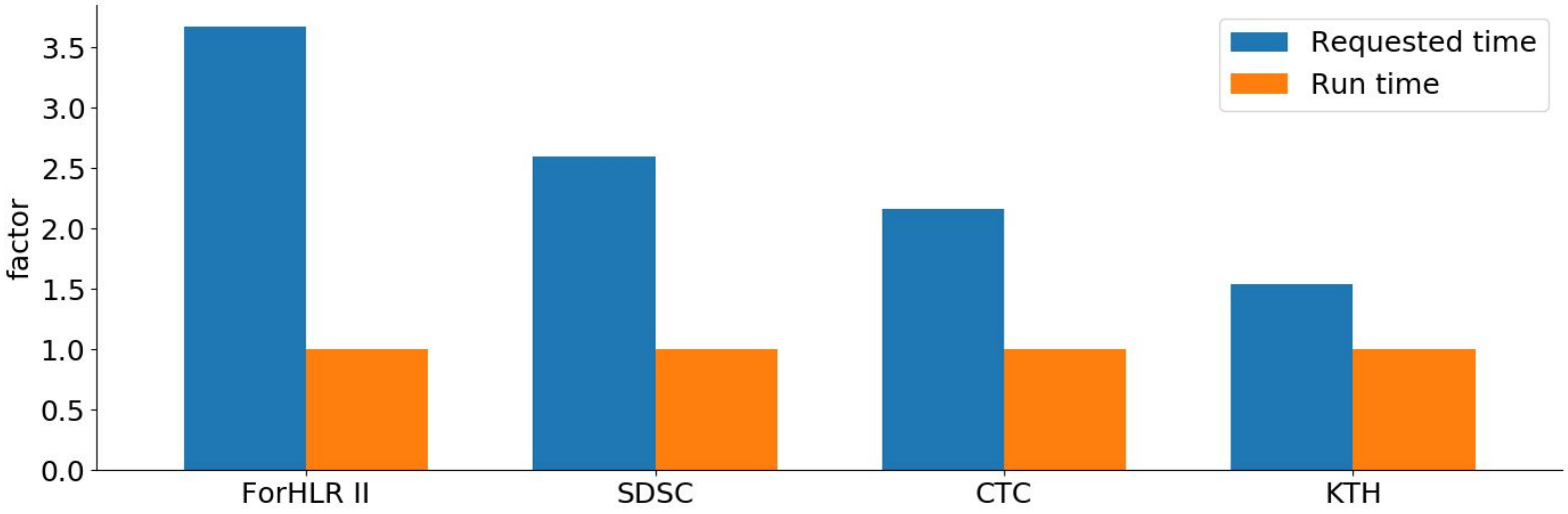


Redefined requested wall time (1)



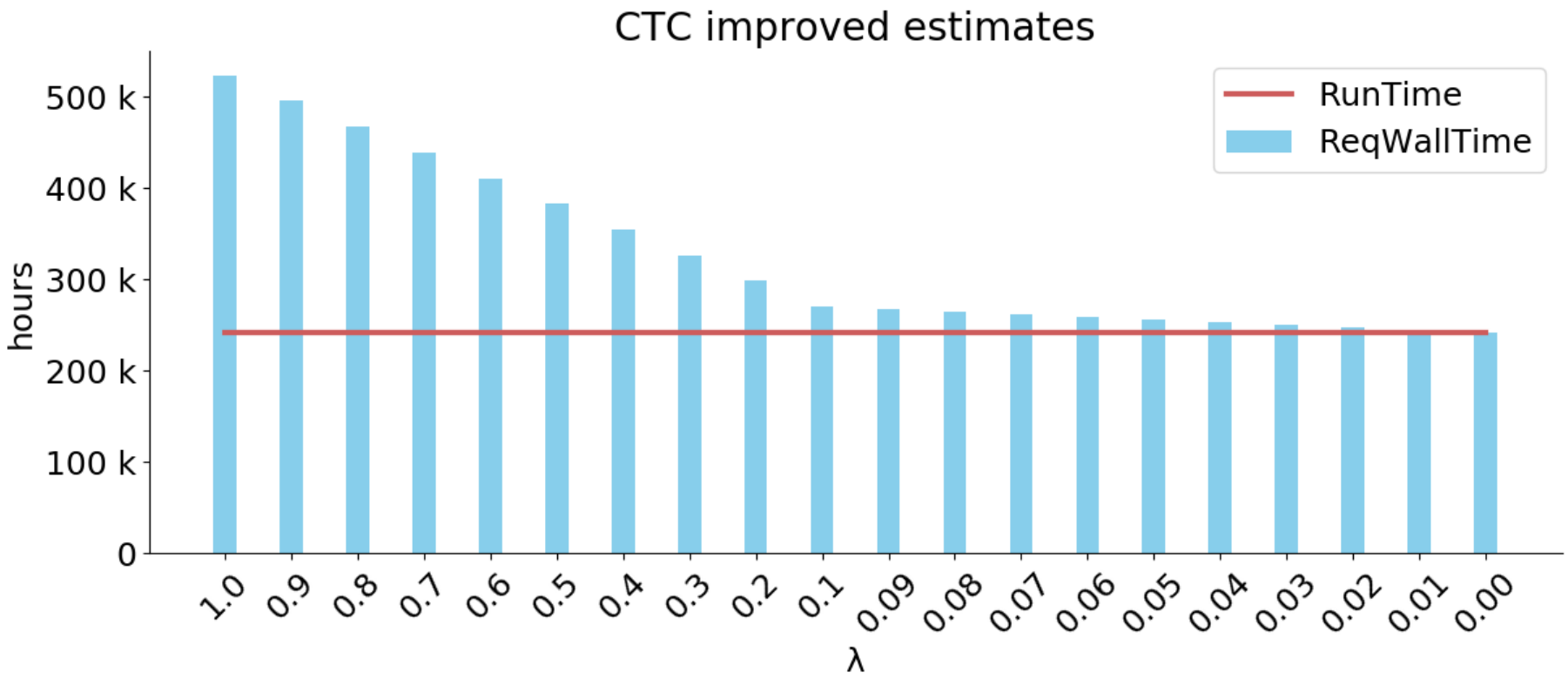
- Calculate for every job a new wall time estimate based on λ_x for given x

Requested / used walltime



- User request more than they use
- ForHLR 3.5X requested wall time than used wall time
  SDSC 2.5X
  CTC 2.0X
  KTH 1.5X

Improved wall times estimates



Metric

- Categorized into valid node allocation prediction time (T_{NAP})

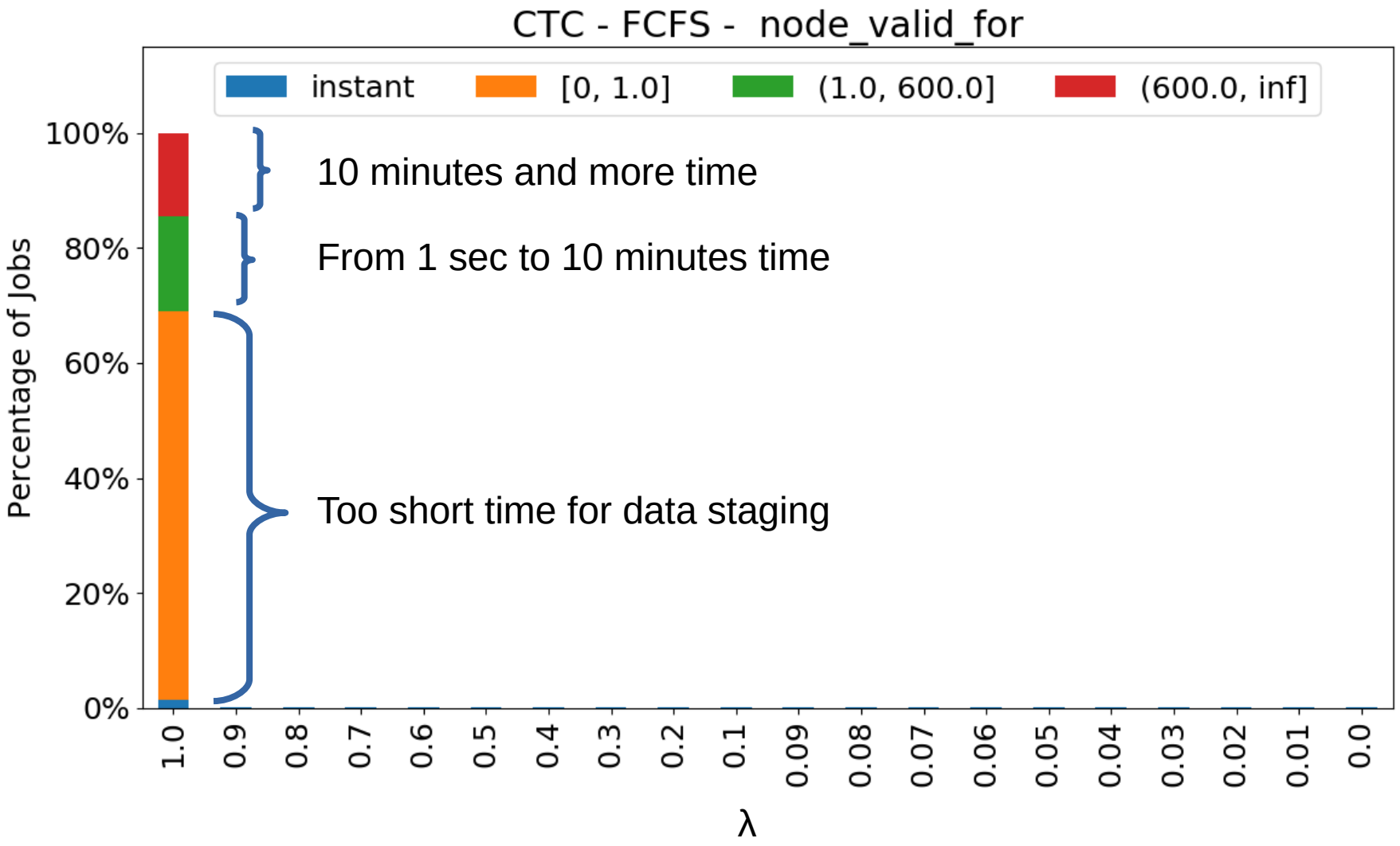
>10 minutes before job start the node allocation list is known

1 seconds – 10 minutes

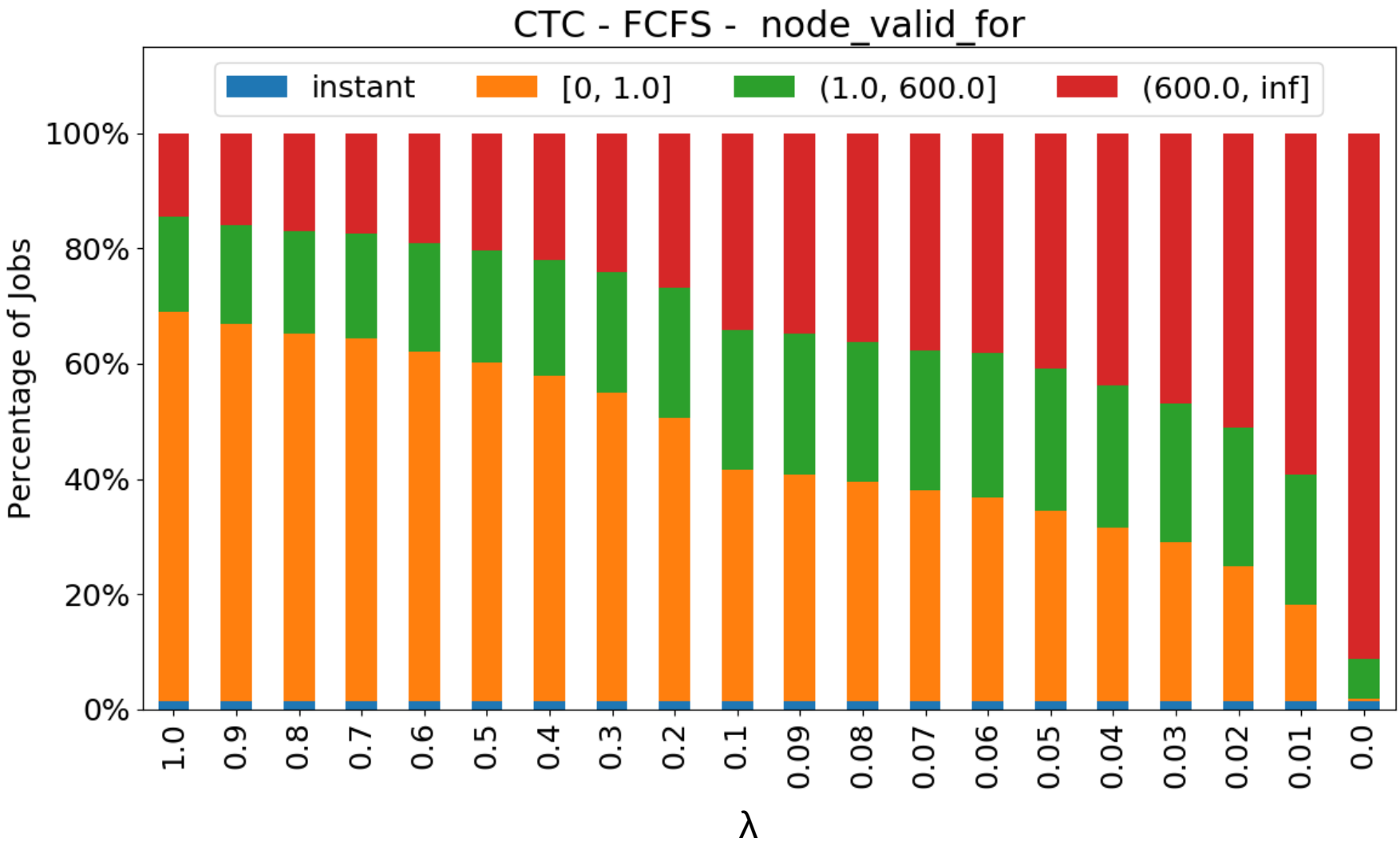
0 seconds – 1 second

Instant started jobs

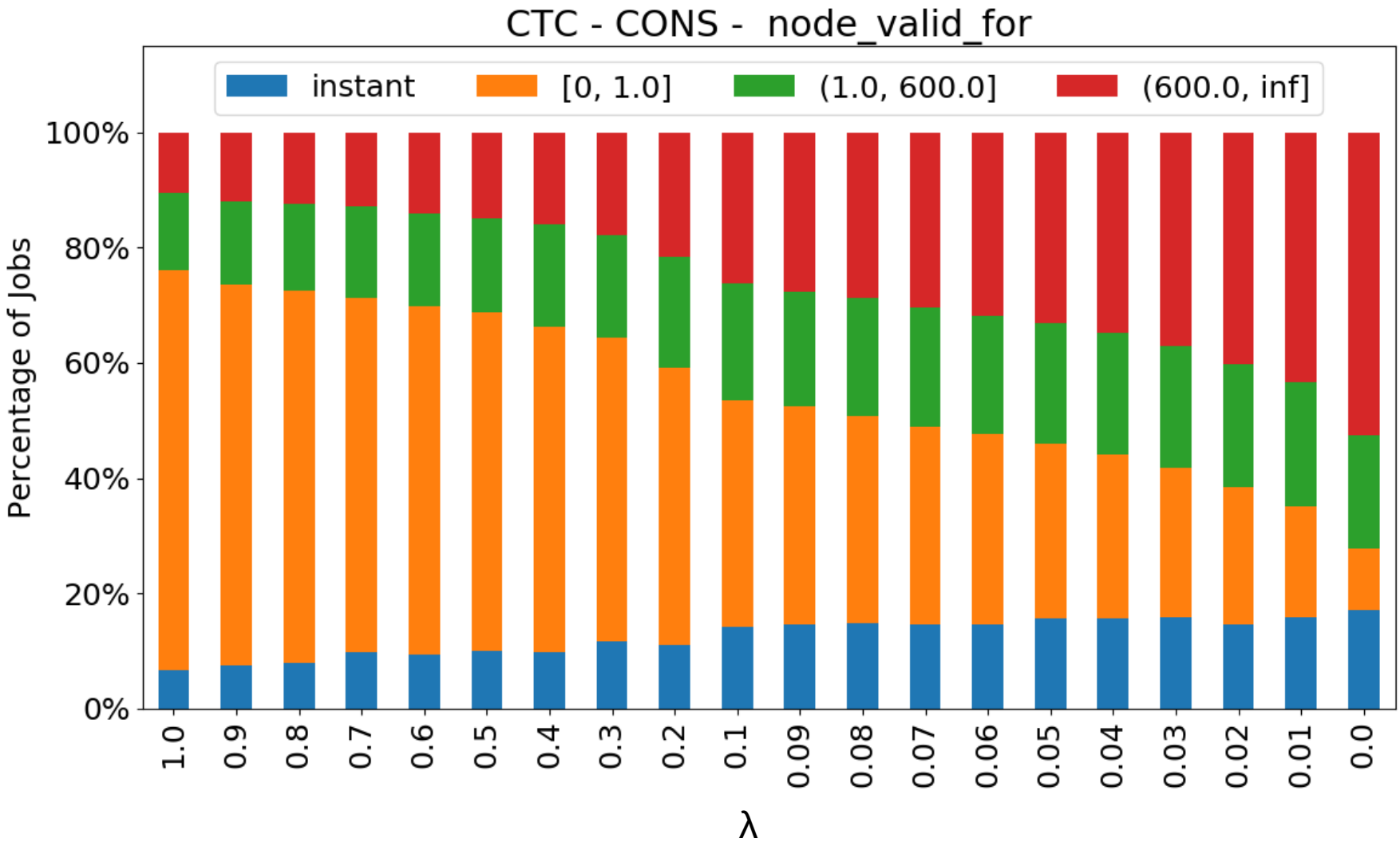
Results – CTC - FCFS



Results – CTC - FCFS



Results – CTC - Back-filling



Conclusion & Take Away

- With FCFS higher accuracy on node allocations
- Alea wall time predictor is quite good if user estimations are bad
- But still:
  Even with perfect wall times there is a huge uncertainty

- Alea can now simulate node allocation prediction
- Modification to scheduling needed for advanced data staging
  - Reservations
  - Slurm ODFS Burst buffer plugin

Acknowledgement

- ADA-FS Project
- DFG priority programm SPPEXA “Software for exascale Computing”
- Steinbuch Centre for Computing

- Contact: Mehmet.Soysal@kit.edu

Questions?