On-the-Fly Calculation of Performance Metrics with Adaptive Time Resolution for HPC Compute Jobs

Konstantin Stefanov, Vadim Voevodin

cstef@parallel.ru, vadim@parallel.ru

Research Computing Center
M.V.Lomonosov Moscow State University

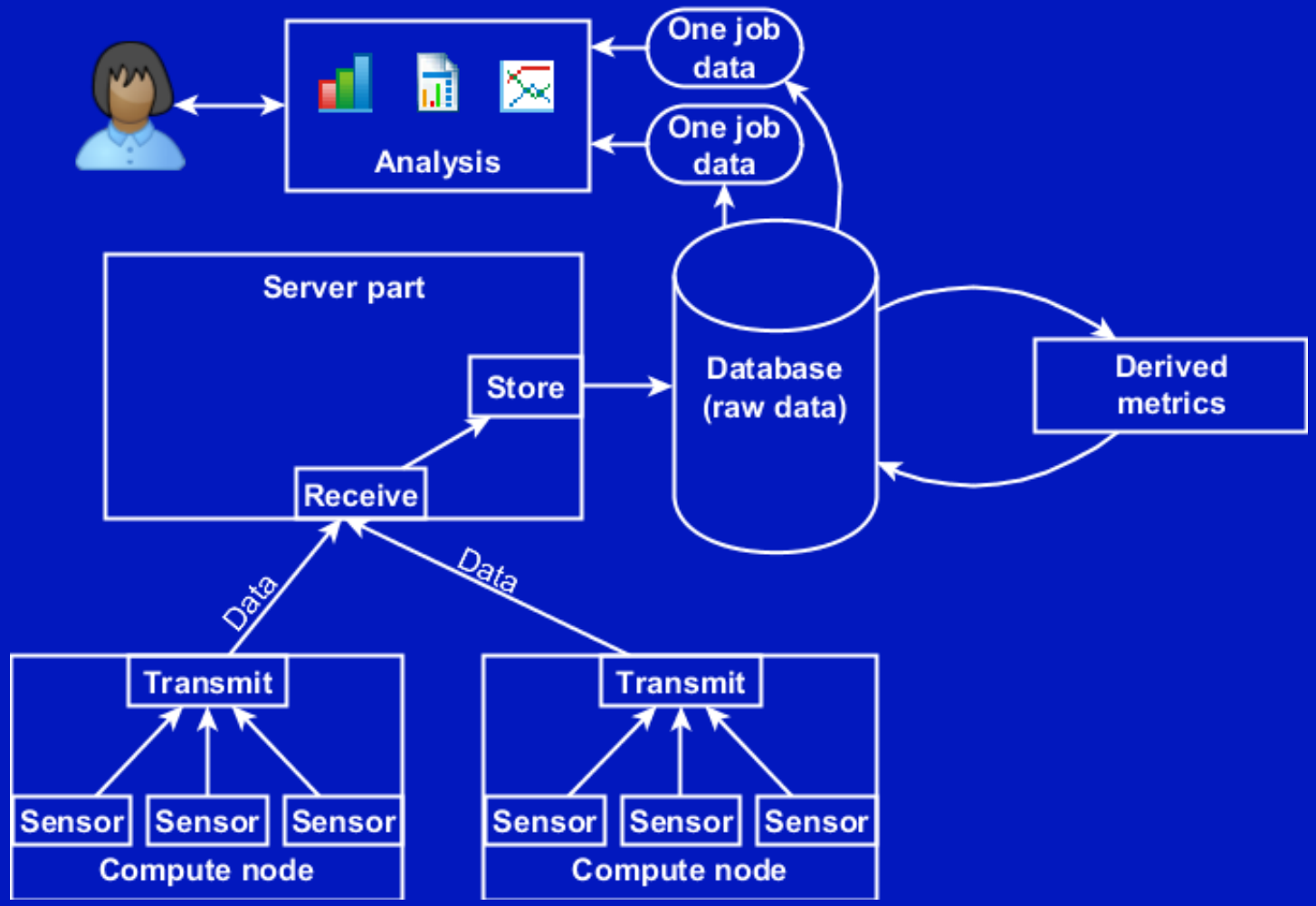
Russian Supercomputing Days
Moscow
24.09.2018

Performance Monitoring

- Basic object (job) is dynamic – need to select related data from the whole stream
- Data volume is huge:
  - Poll frequency – the more HZ the better
  - Many different sensors (CPU load, LoadAvg, flops, network counters etc.)
- Much of the data are used only once (calculating aggregates for finished job)

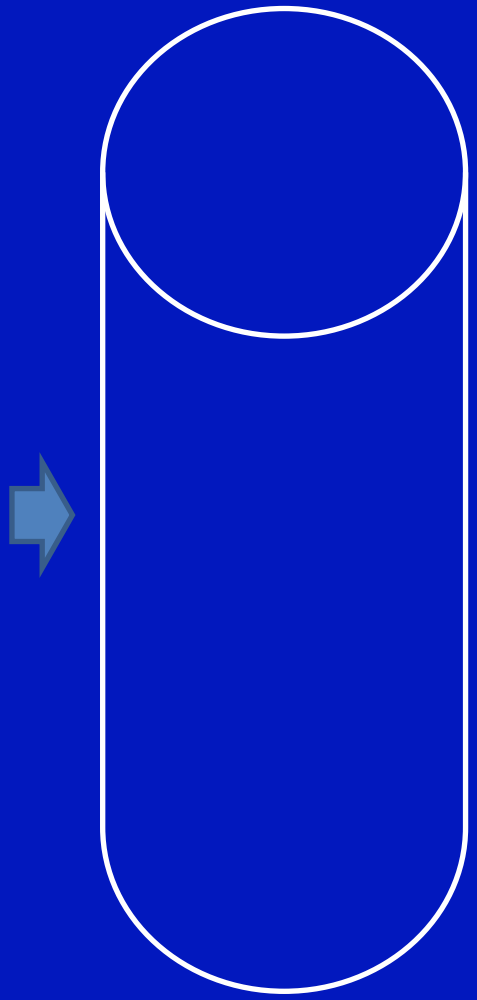
On-the-fly Calculation of Performance Metrics

Performance Monitoring: Common Approach



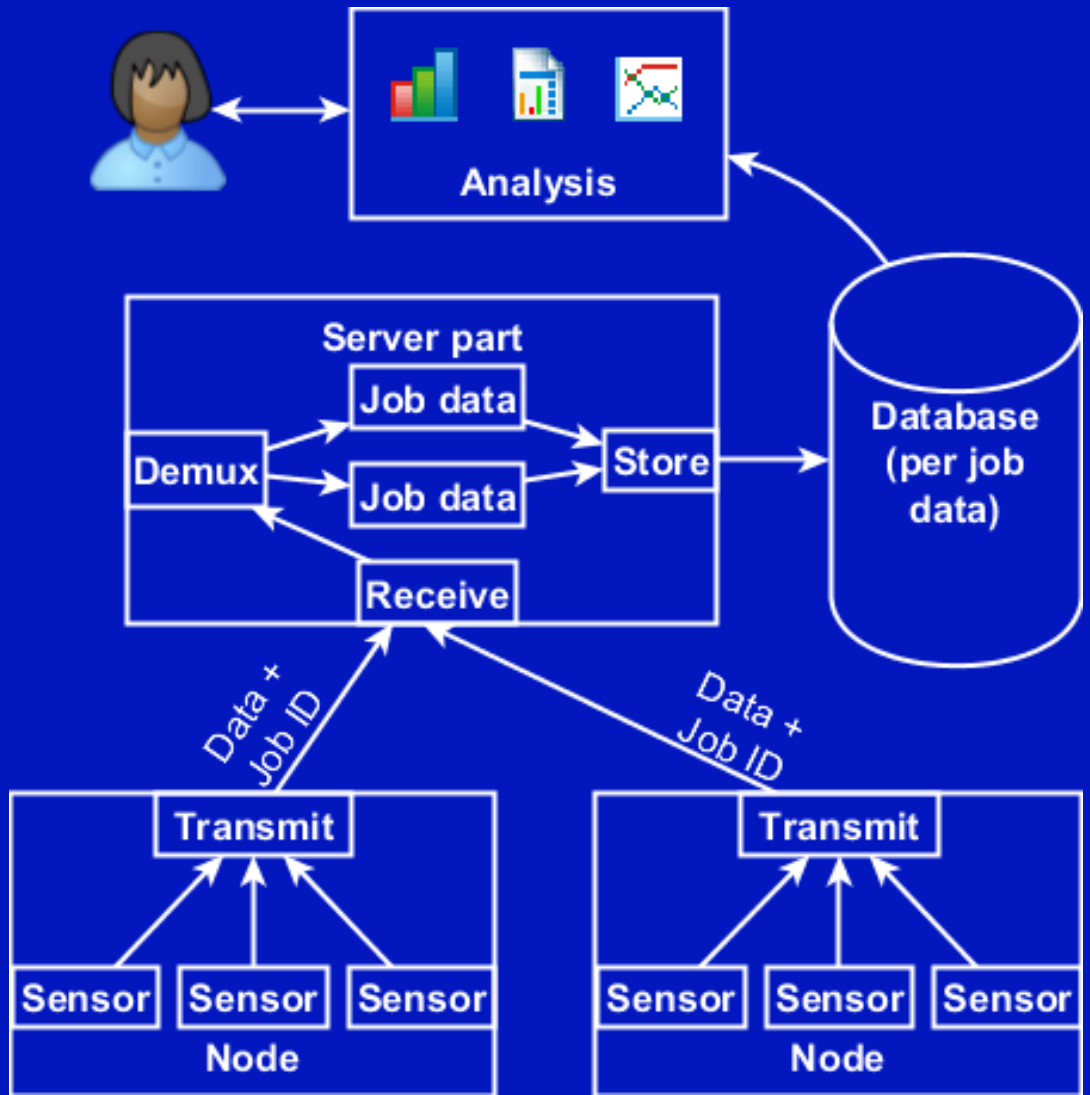
Performance Data Stream: Random IO

Time	Node	Sensor	Value
21:34:45	1	CPULoad	45
		IB bytes in	2679
	2	CPULoad	49
		IB bytes in	2179
	3	CPULoad	35
		IB bytes in	1629
	4	CPULoad	45
		IB bytes in	1629
21:34:55	1	CPULoad	45
		IB bytes in	2679
	2	CPULoad	45
		IB bytes in	2679
	3	CPULoad	45
		IB bytes in	2679
	4	CPULoad	45
		IB bytes in	2679

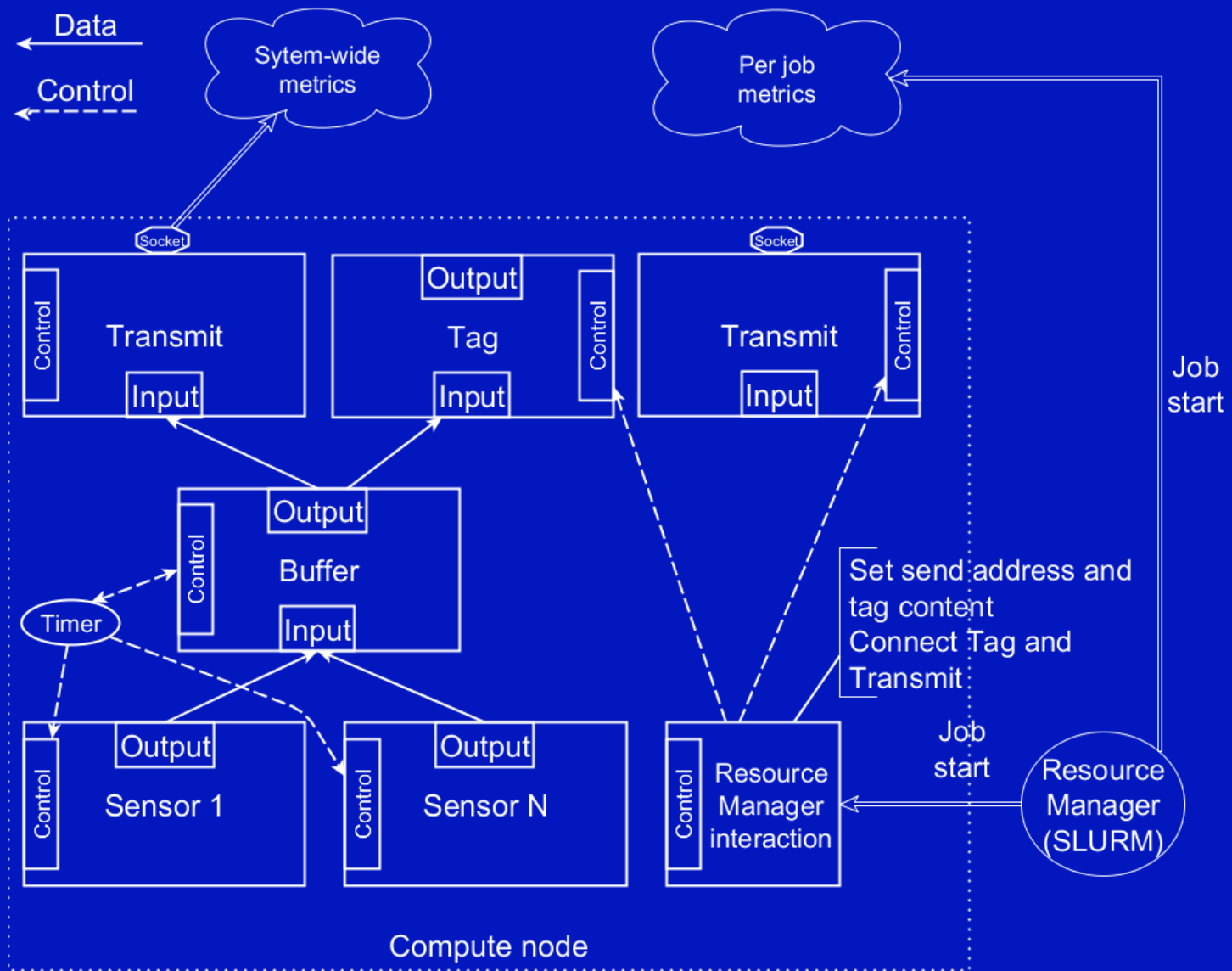


Job	Start	Finish	Nodes
1	20:30:07	21:34:50	1, 3
2	21:05:37	21:48:23	2, 4
3	21:34:53	22:08:21	1, 3

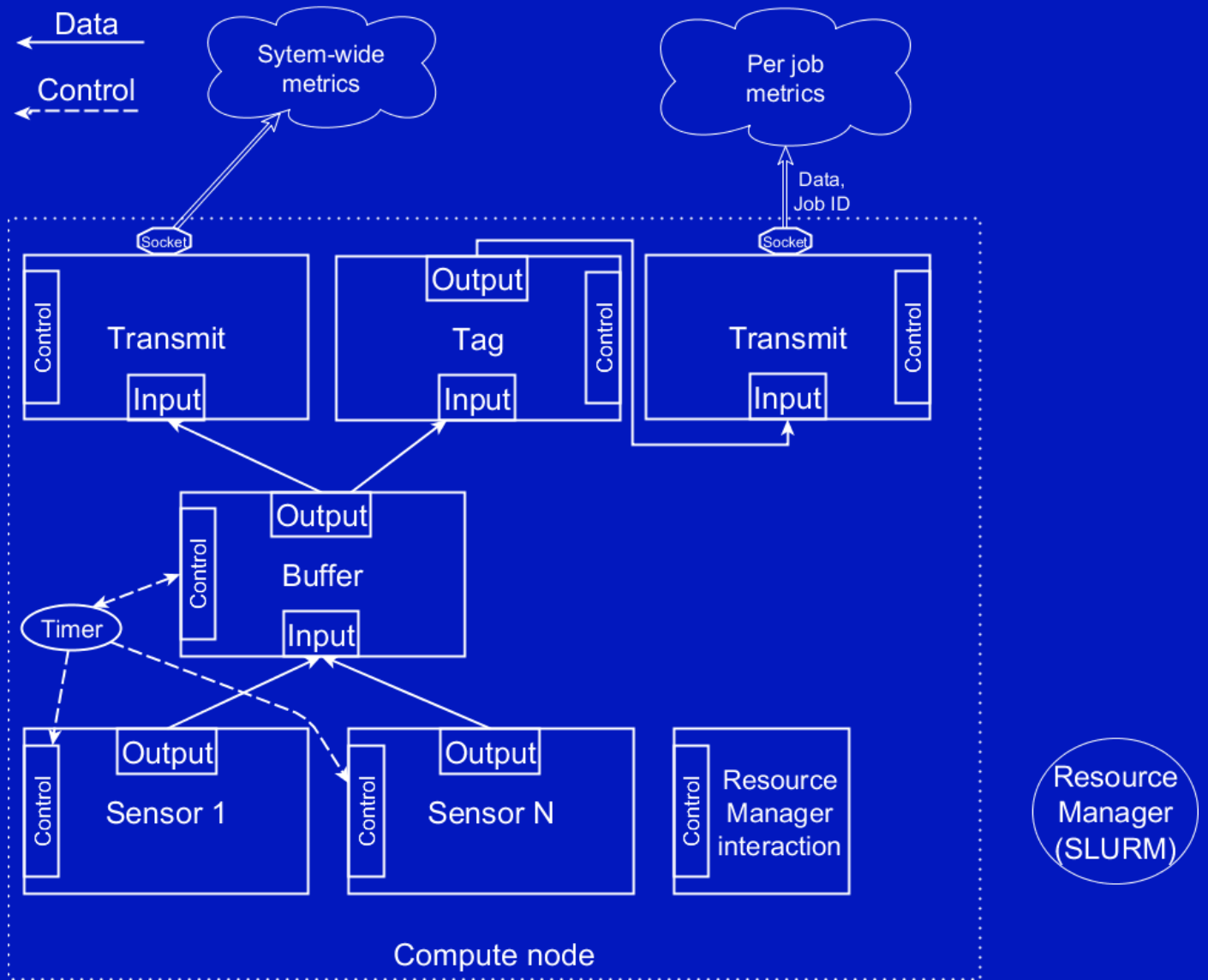
Performance Monitoring: Proposed Approach



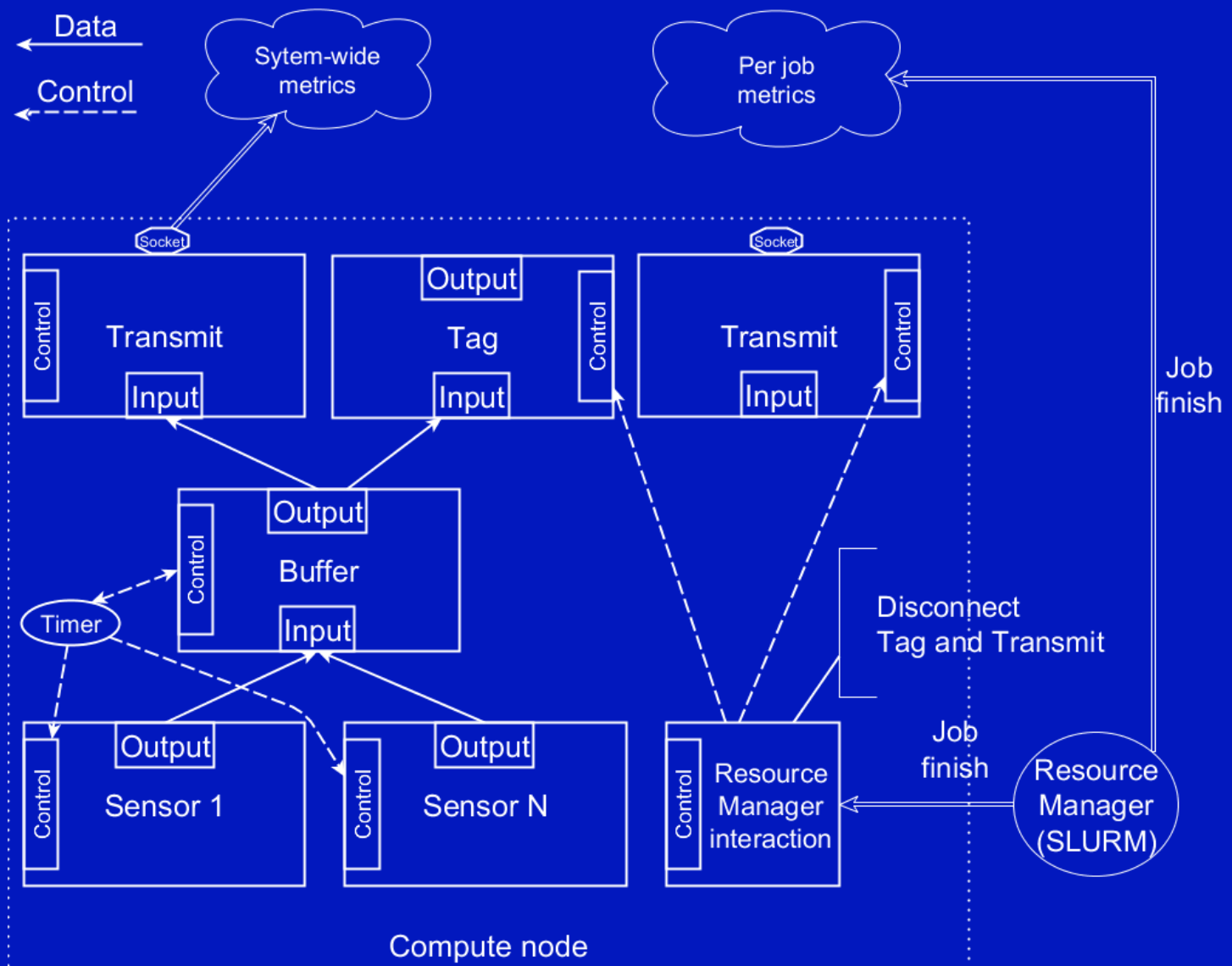
Reconfiguring on-the-fly to Calculate per-Job Metrics: Job Start



Reconfiguring on-the-fly to Calculate per-Job Metrics: Job Running



Reconfiguring on-the-fly to Calculate per-Job Metrics: Job finish



Issue:
Restart Monitoring System Parts

- Restart is a big issue:
  - Node agent saves node ID
  - Server part saves job data after the job is finished

Issue:
Restart Node Agent

- Restart is done after a job is finished from a system-wide SLURM epilog script

Issue:
Restart Server Part

(not implemented yet)

- 2 server parts, both have the same data
  - Main part works as described above, save the data right after the job is finished
  - Standby part tries to save job data in 30s after the job is finished
    - If the record for the job is already there, the data are discarded
- Restart both parts with interval longer than the job time limit

Making Time Resolution Adaptive

Problem description and solution

- The more data points the better (especially for short jobs)
- Graphs with many data points are hard to visualize
- Need a trade-off

- Make dynamic resolution:
  - Fine for short jobs
  - Coarse for long jobs

Short Job

Time	23:00:01	23:00:02	23:00:03	23:00:04							
CPU user, %	100	94	90	90							

...

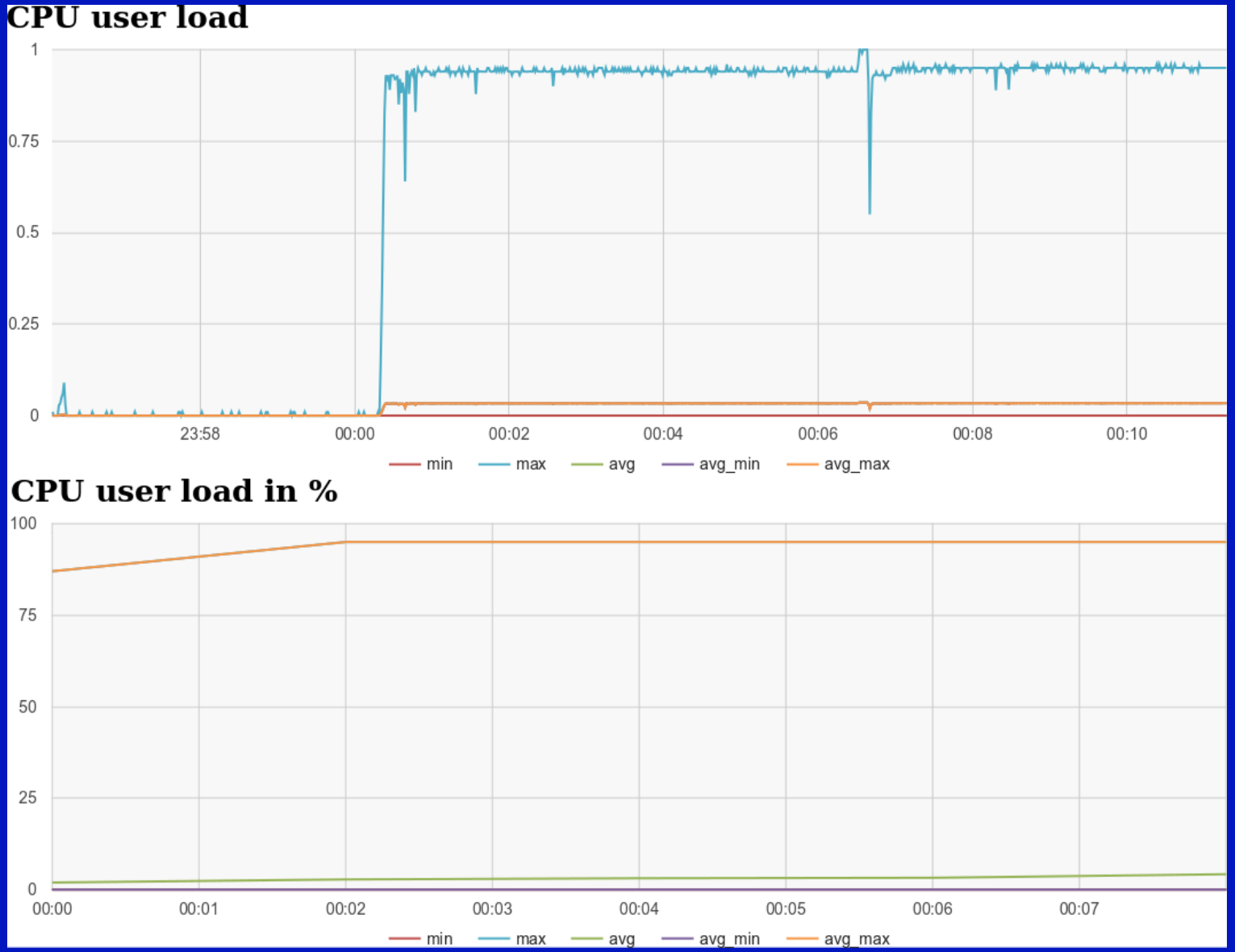
Time	23:00:01	23:00:02	23:00:03	23:00:04	...					23:16:39	23:16:40
CPU user, %	100	94	90	90						50	55

Not-so-Short Job

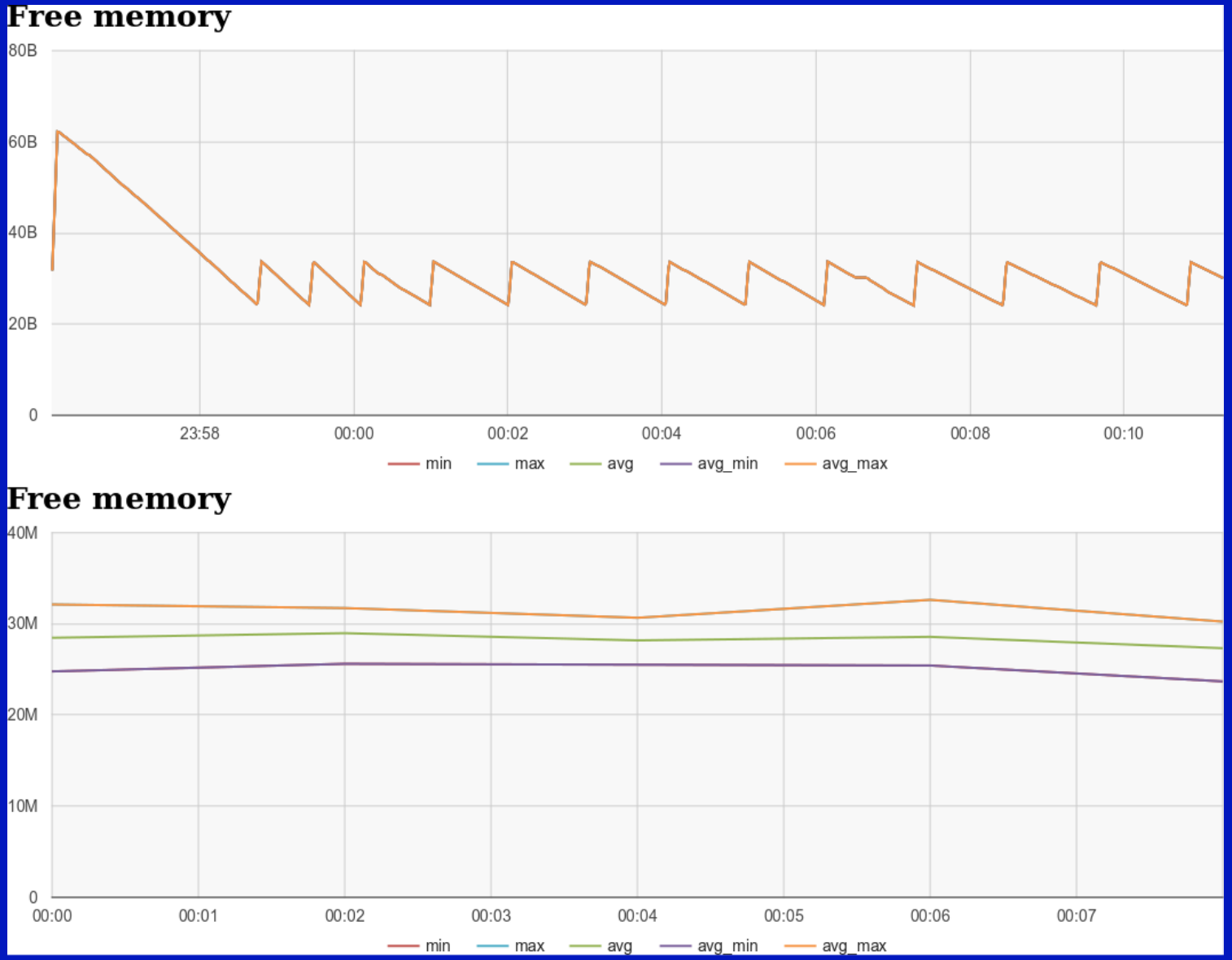
Time	23:00:01	23:00:02	23:00:03	23:00:04	…					23:16:39	23:16:40
CPU user, %	100	94	90	90						50	55

Time	23:00:01	23:00:03	…		23:16:39						
CPU user, %	97	90			52,5						

Results:
Dynamic vs Fixed (2 min) Resolution



Results:
Dynamic vs Fixed (2 min) Resolution



Thank you for your attention