Tyrex: Size-based Resource Allocation in MapReduce Frameworks

CCGRID 2016

Bogdan Ghit and Dick Epema

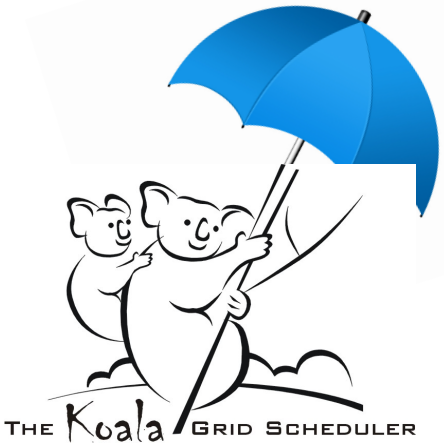
Delft University of Technology

About me

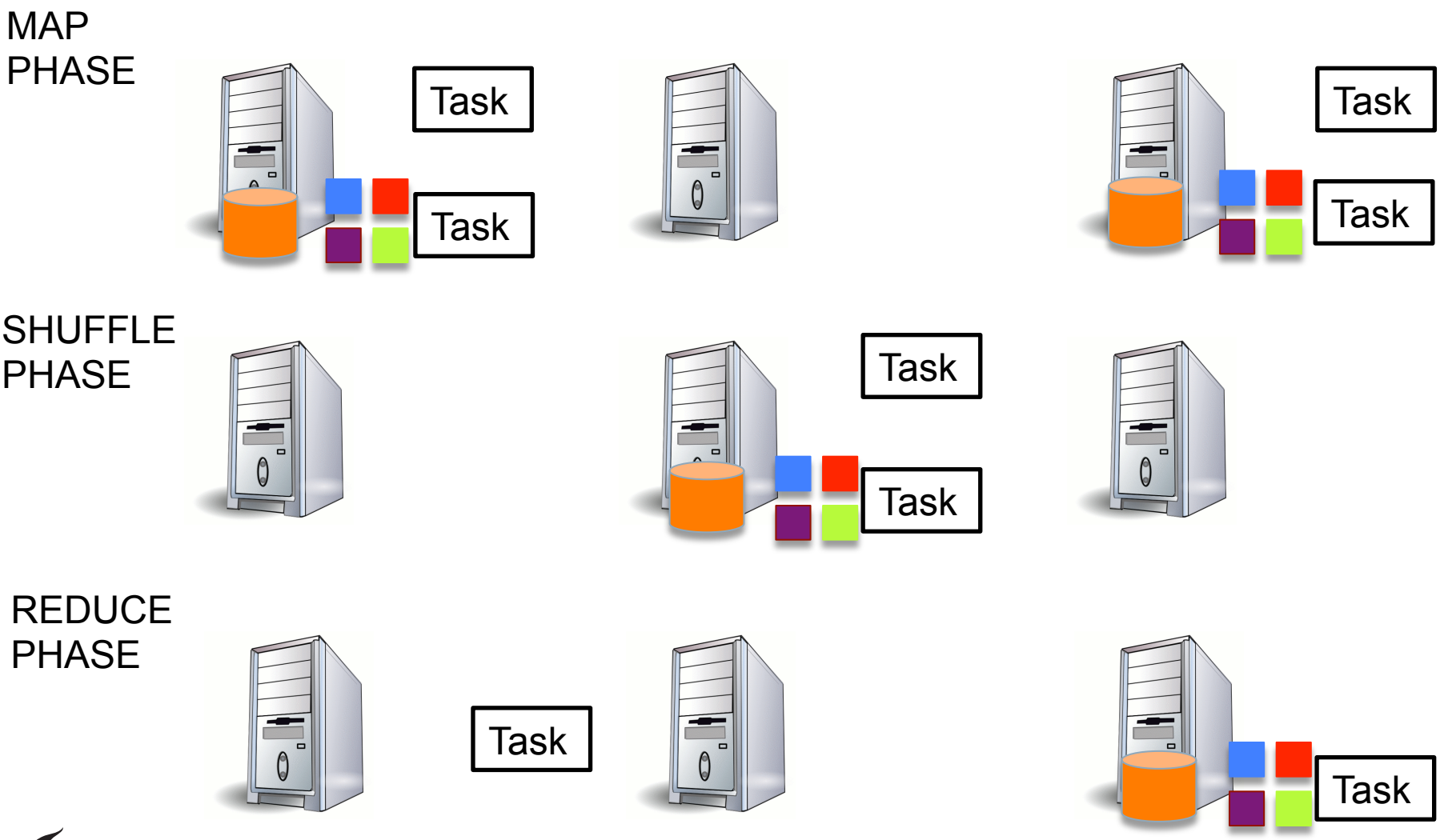
PhD candidate at TU Delft, advised by Dick Epema.

Thesis topic: optimizing the performance of data analytics frameworks.

Member of the DS group (see tag cloud below).



A job model for datacenters



Datacenter workloads are heavy-tailed

- Very variable job sizes:
  - Google, Facebook, Bing, Yahoo! clusters.

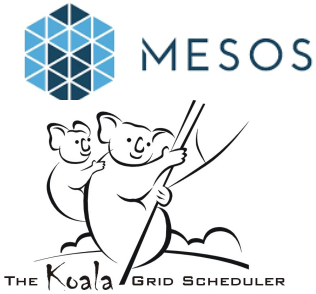
- Our experience with BTWorld:
  - Less than 15% of the jobs account for 80% of the total load.
  - More than 65% of the jobs complete in a minute.

- Short jobs experience long delays due to large jobs ahead of them.

Bogdan Ghit et. al., “V for Vicissitude: The Challenge of Scaling Complex Big Data Workflows”, SCALE Challenge Winner, 2014.

MapReduce schedulers

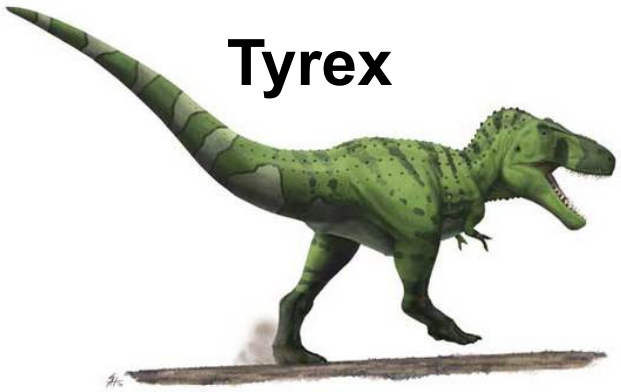
Performance:
- Locality: Delay scheduler [EuroSys'10], PAC-Man [NSDI'12]
- Stragglers: Late [OSDI'08], Hopper [SIGCOMM'15].



Fairness:
- Resource-sharing: FAIR scheduler [EuroSys'10], Mesos [NSDI'11], Fawkes [Sigmetrics'14], Koala-F [CCGrid'16]

Performance & fairness: This talk!

Tyrex



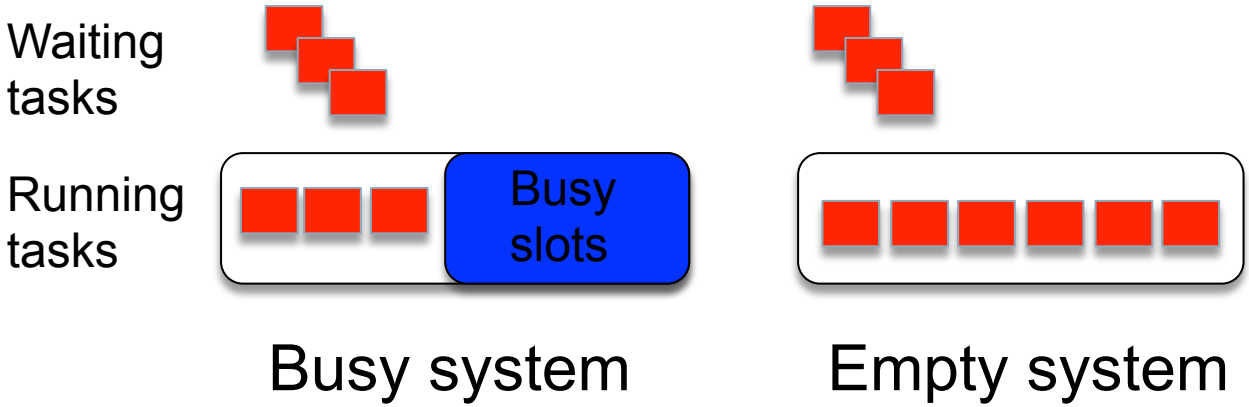
This work

(1) Formulate the goals of our scheduler.
Job slowdown variability.

(2) Design of Tyrex.
Two scheduling policies.

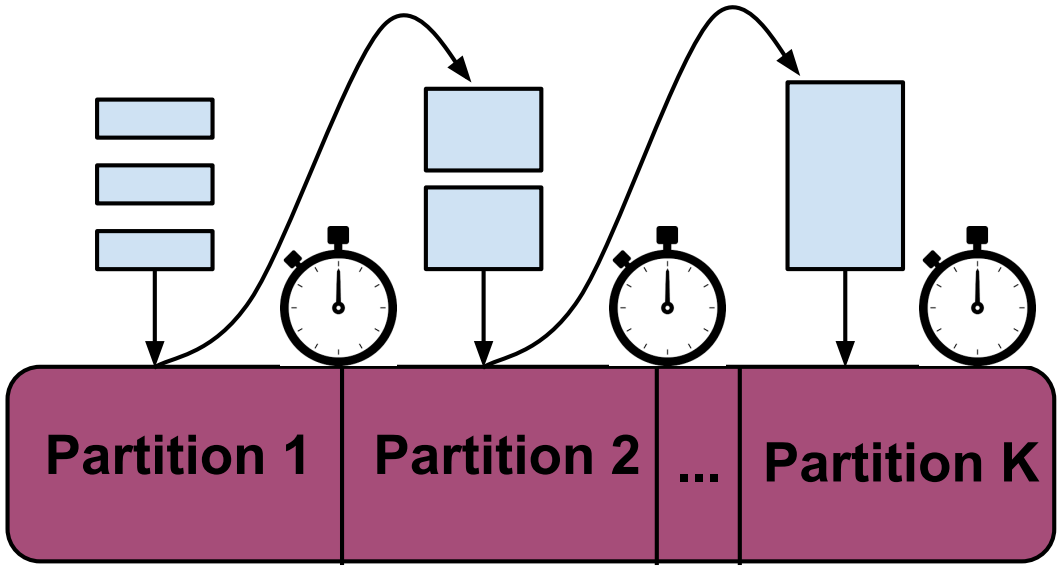
(3) Experimental evaluation on the DAS.
Job slowdown performance.

Job slowdown variability



- Job slowdown = response time / wall-clock time.
- Job slowdown variability: 95th percentile/median.

The queueing model



- Resource partitioning: avoid job interference.
- Timer-based job migration: distinguish job sizes.

Design considerations

Inspired by the TAGS policy, but with three key differences:

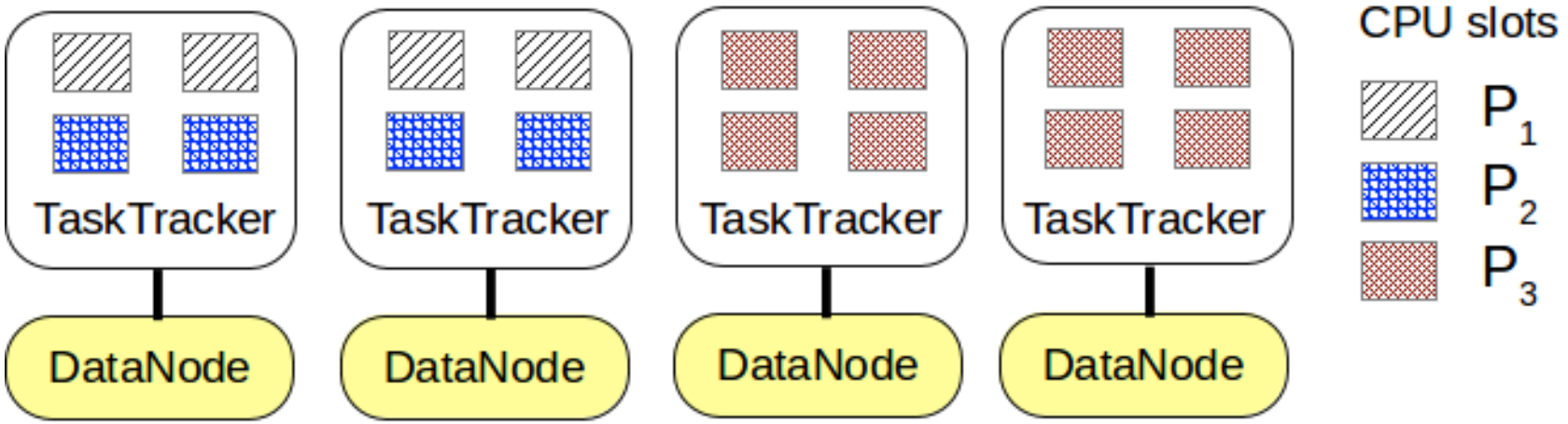
1. **Datacenter environment**

2. **Complex job model**

3. **Work-conserving system**

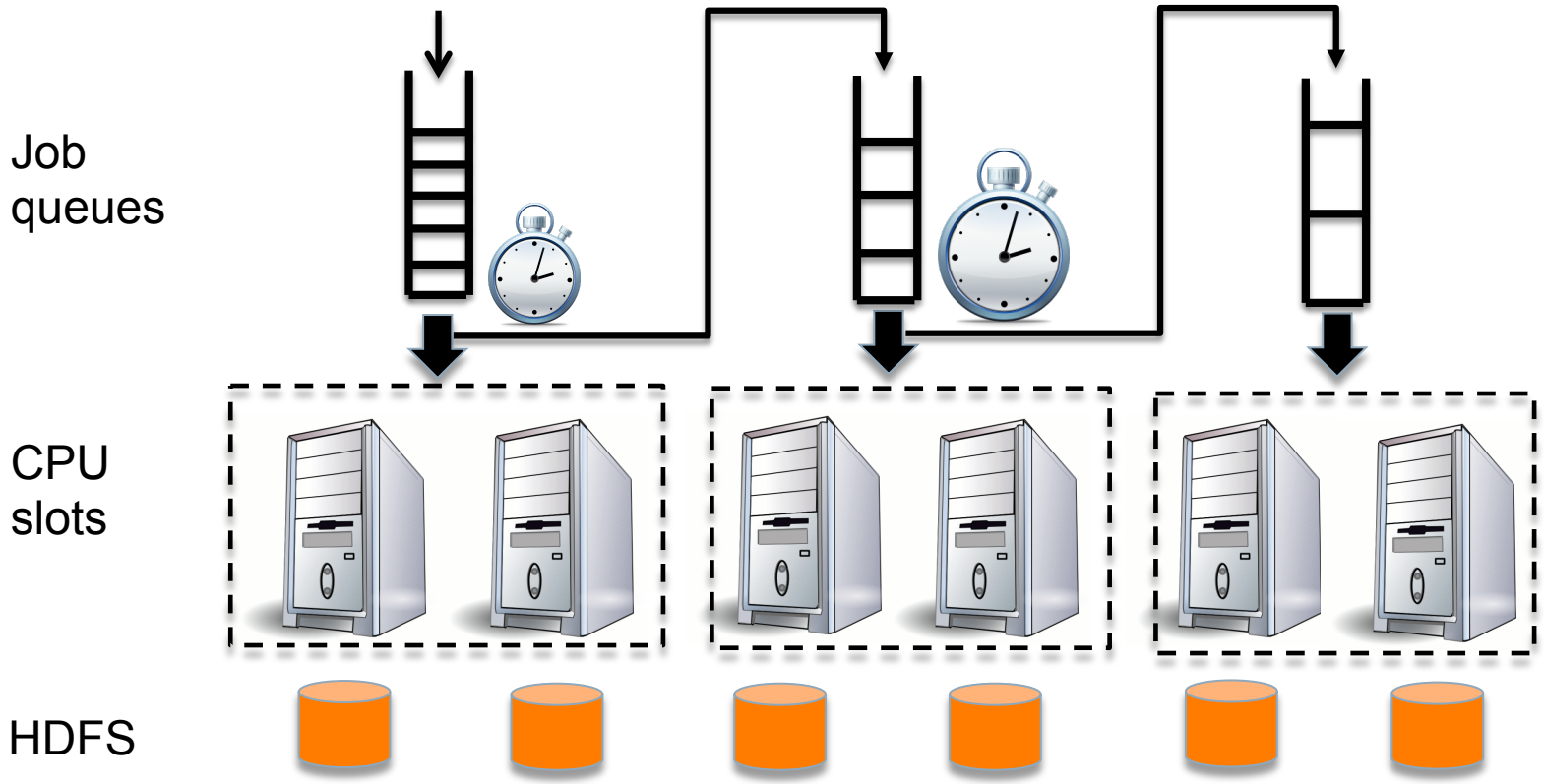
M. Harchol-Balter, “Task assignment with unknown durations”, Distributed Computing Systems, 2001.

System model



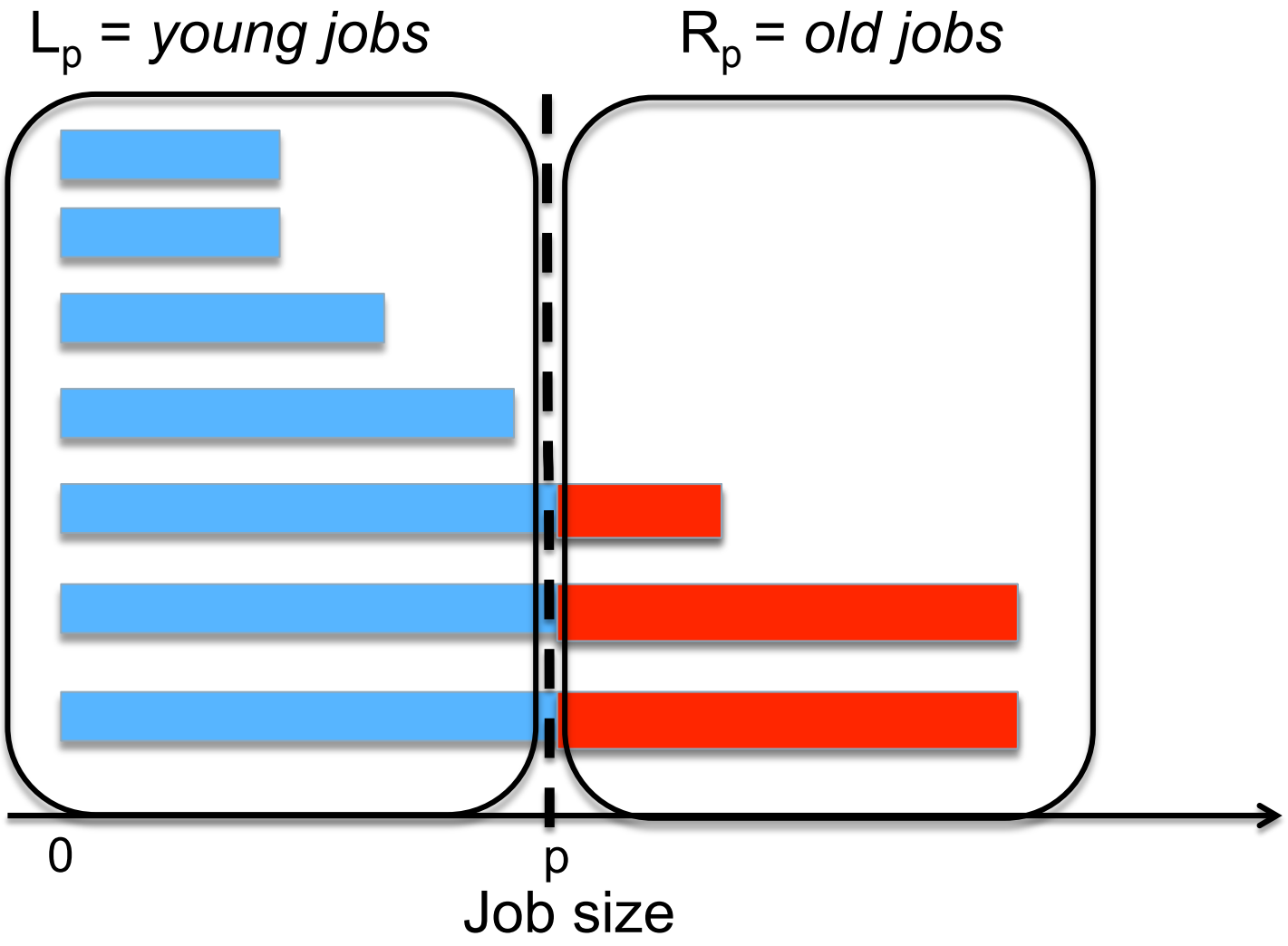
- Job migration with no overhead.

The StaticTAGS policy

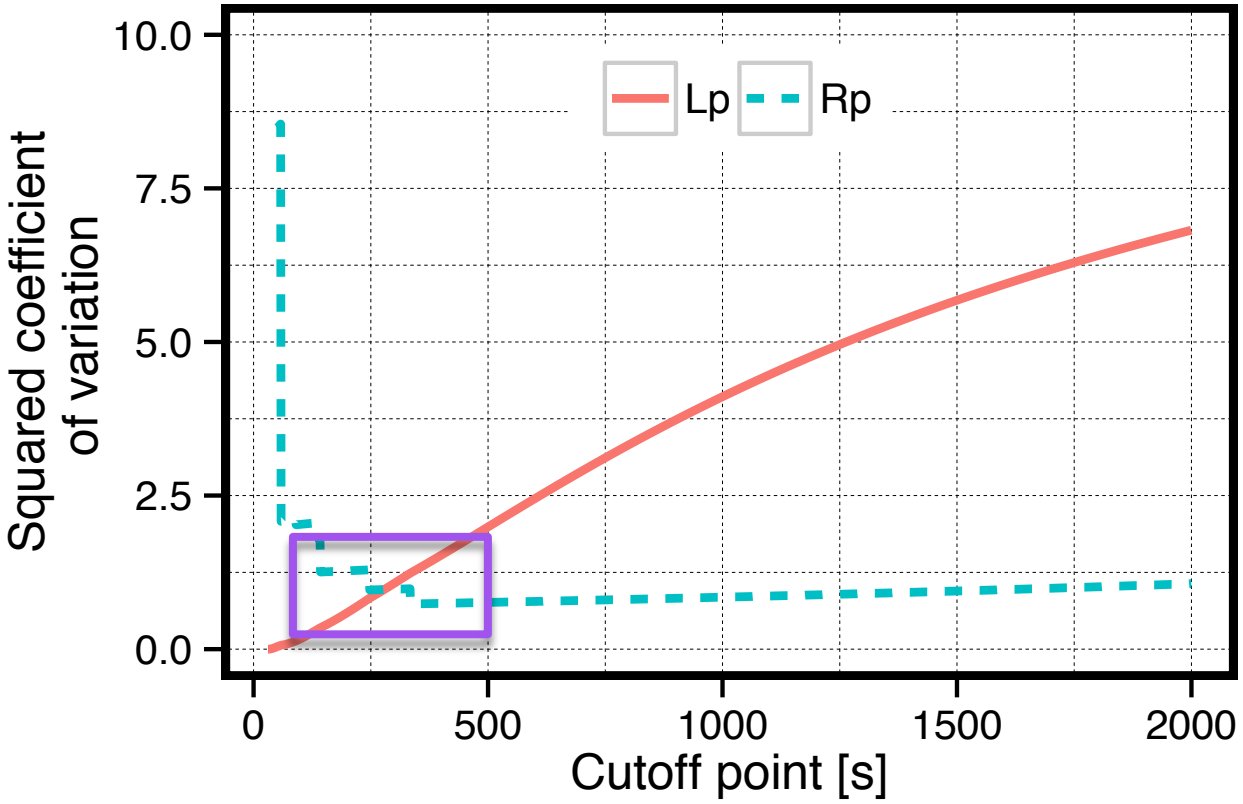


- Move a job to the **next queue** when it exceeds the timer using capacity from the **current partition**.

Identifying long jobs (1/2)

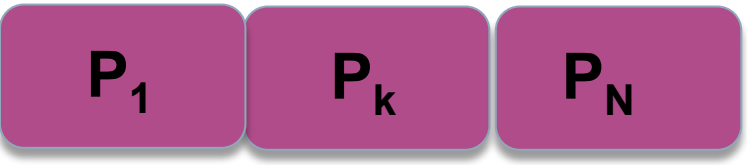


Identifying long jobs (2/2)

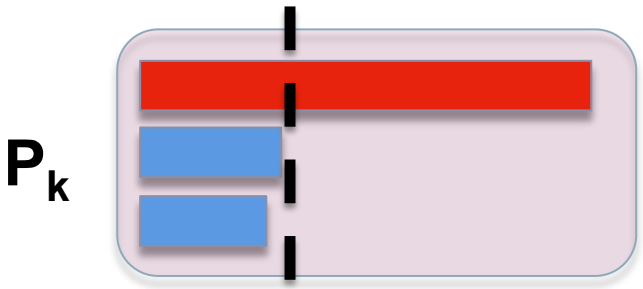


- Optimal cutoff point: balanced squared CVs.
- We aim for a squared CV in L_p that is lower than 2.
- No need for more than 2 partitions.

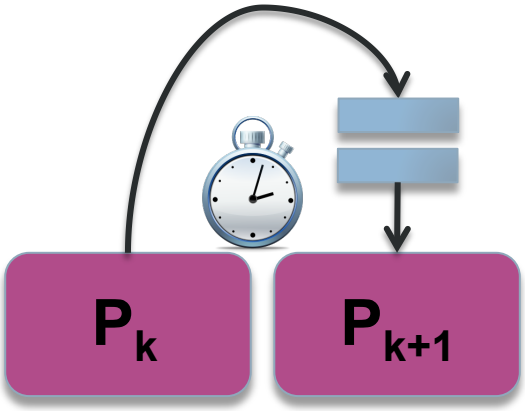
The DynamicTags policy



1. Find partitions with variable job sizes.



2. Determine the optimal cutoff point.



3. Set the timer to the optimal cutoff point.

Experimental setup

- DAS-4 multicluster system
  - 20 dual-quad core nodes, 24 GiB, Infiniband.
  - Hadoop 1.0, 4 map slots, 2 reduce slots per node.
  - Tyrex with two partitions.

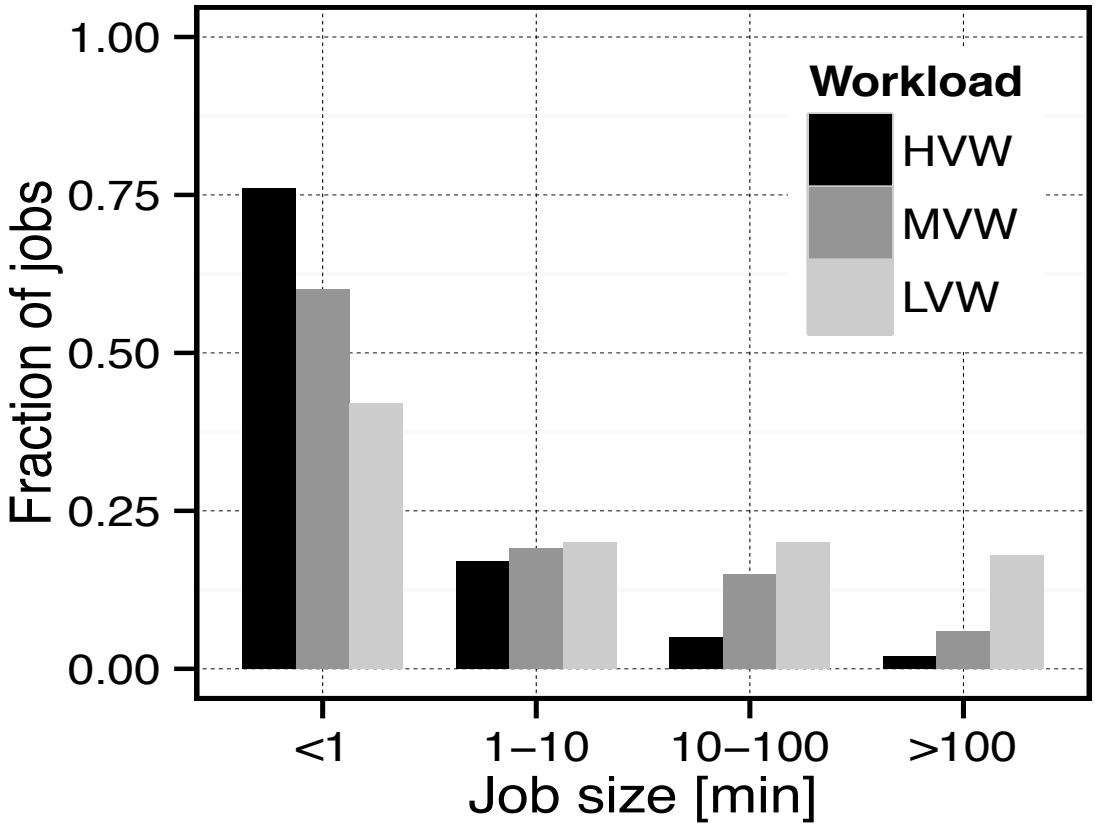
- Applications
  - CPU intensive: Wordcount, PiEstimator
  - Disk intensive: Sort, Grep
  - Complex workflows: BTWorld

MapReduce workloads (1/2)

Statistics	HVW	MVW	LVW
Total jobs		300	
Squared CV	20	10	4
BTWORLD jobs	33	45	10
Total maps	6,139	11,866	30,576
Total reduces	788	1,368	3,089
Temporary data [GB]	573	693	1,062
Persistent data [GB]	100	92	303
Total CPU time [h]	63.6	124.6	306.9
Total runtime [h]	3.51	3.98	5.31

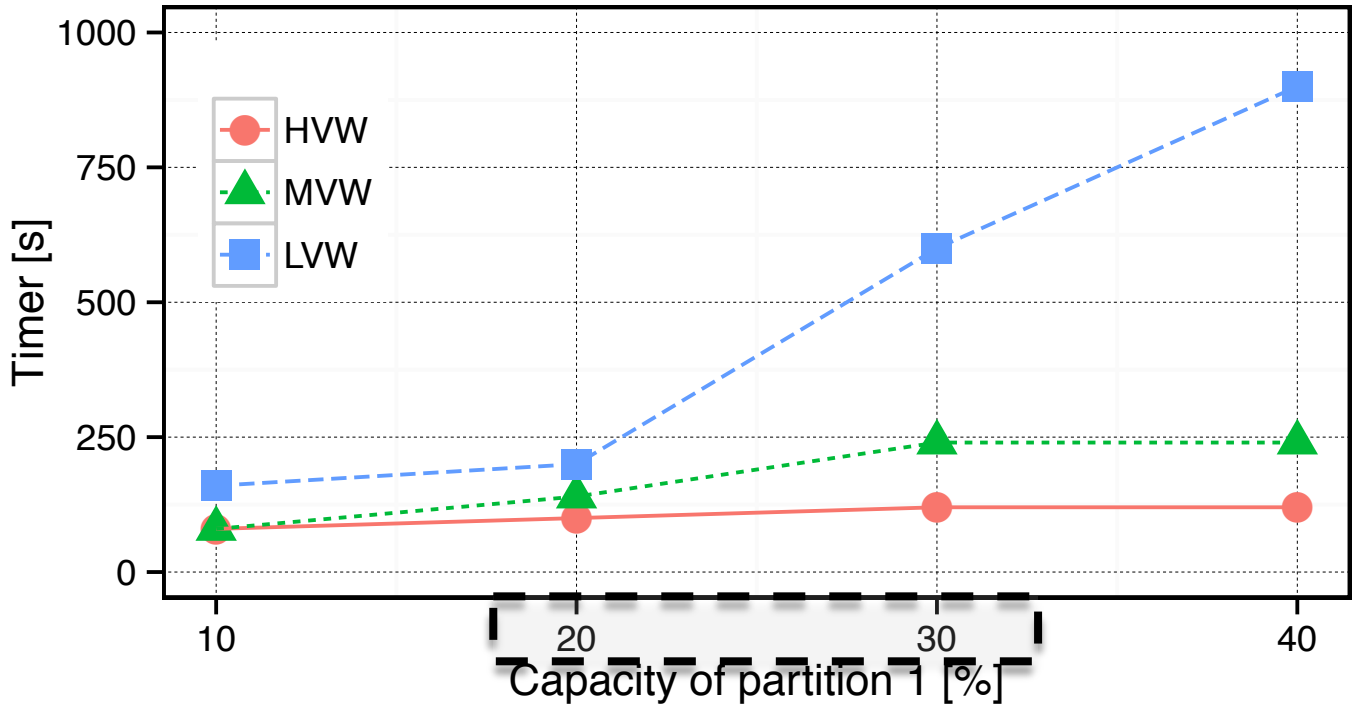
- Stream of 300 jobs with Poisson arrivals.
- Average system load: 70%.

MapReduce workloads (2/2)



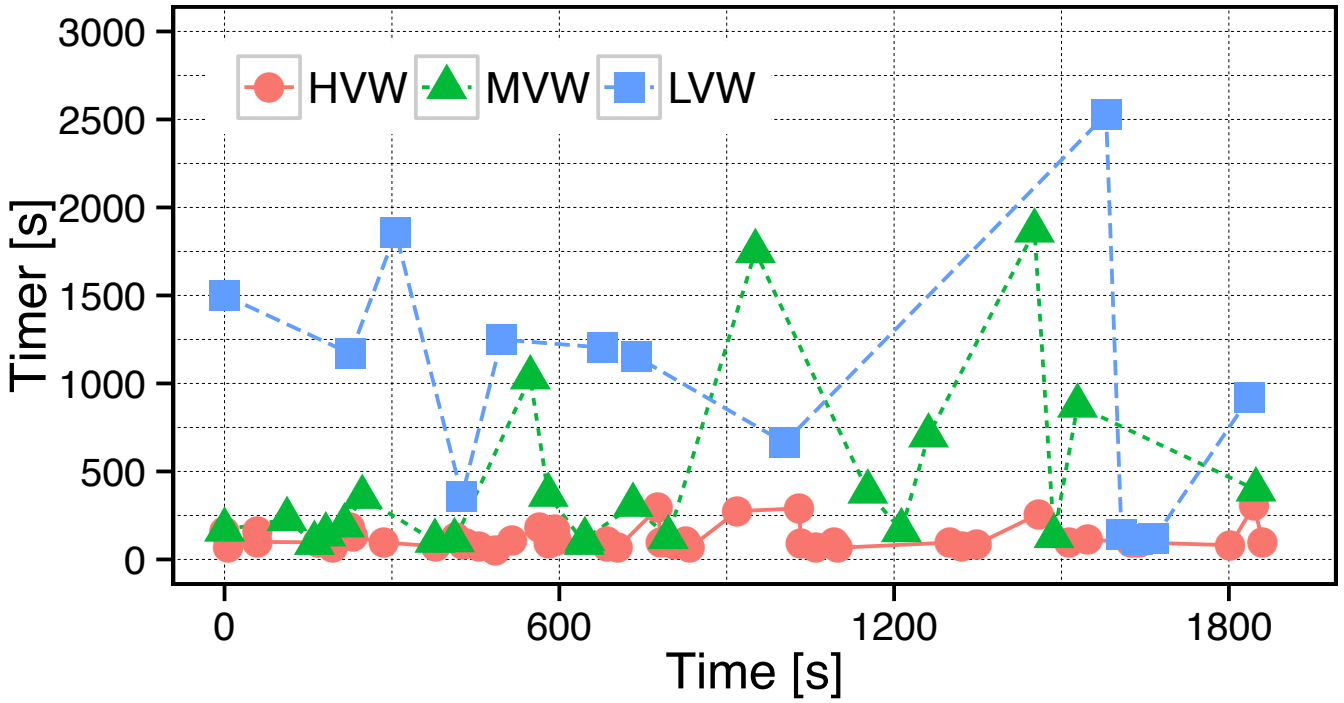
- Large fraction of short jobs in all workloads.

StaticTags: setting capacities and timers



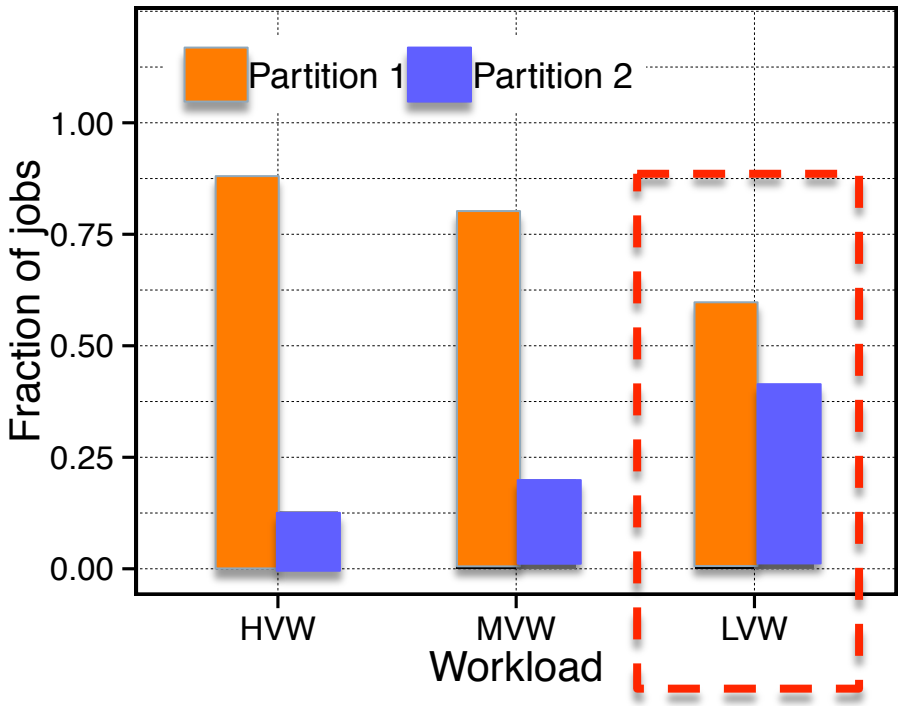
- Optimal timer is insensitive to the partition capacity when the job size variability is high.
- We set the capacity of partition 1 to 30%

DynamicTags: evolution of dynamic timers

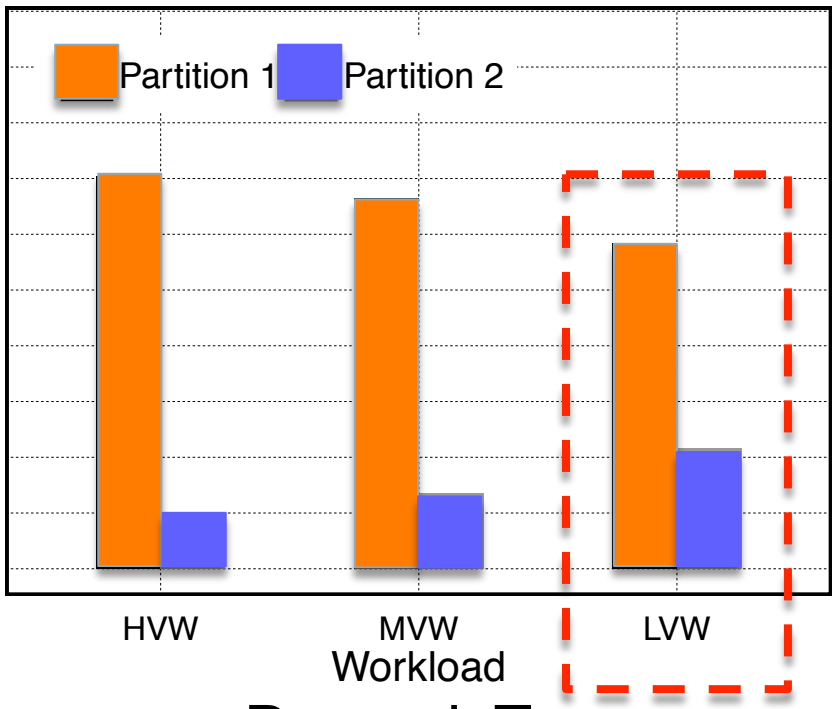


- Low values when the job size variability is high.
- Increasingly wider and higher ranges for MVW and LVW.

Fraction of jobs across partitions



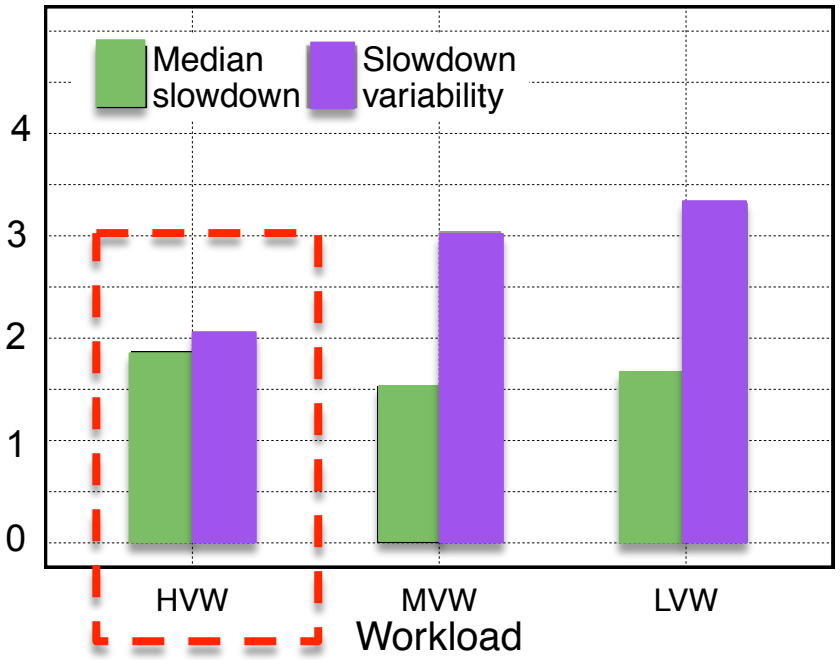
StaticTags



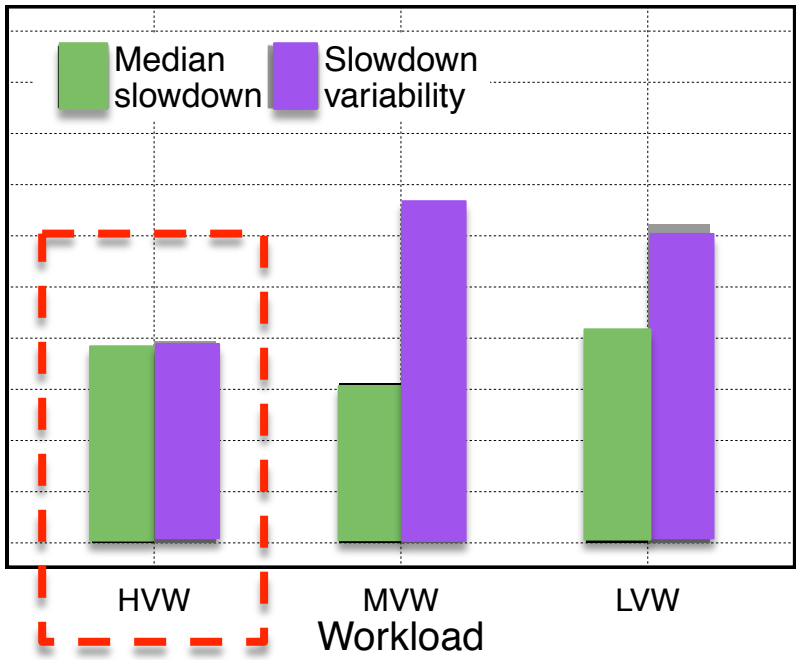
DynamicTags

- Both migrate more jobs to partition P2 as the workload variability decreases.
- DynamicTags is more conservative than StaticTags.

Job slowdown variability



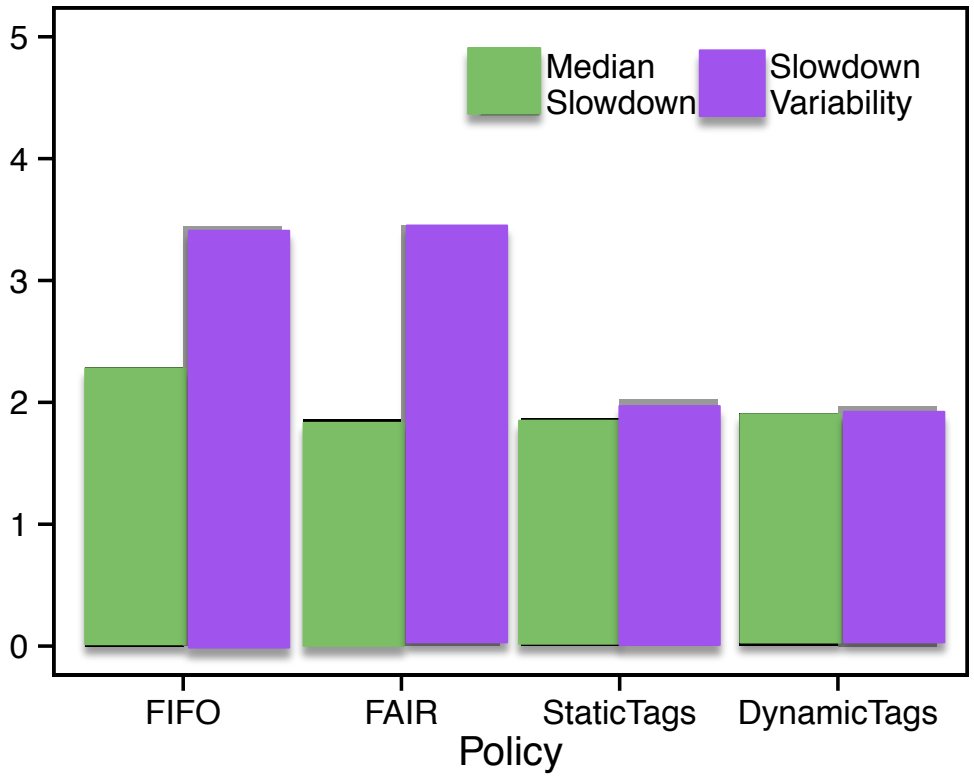
StaticTags



DynamicTags

- Similar results for StaticTags and DynamicTags
- Very good performance for HVW with both values below 2.

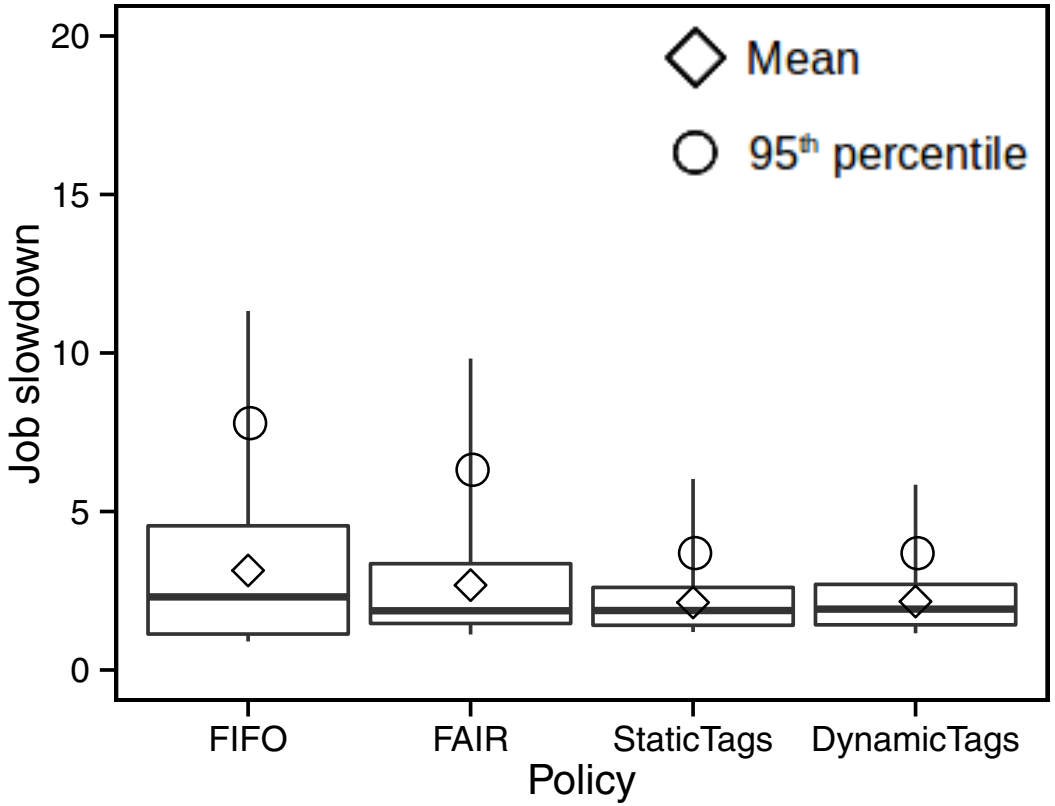
Improvements from Tyrex (1/2)



HVW

- Tyrex cuts in half the job slowdown variability.
- Tyrex maintains roughly the same median job slowdown.

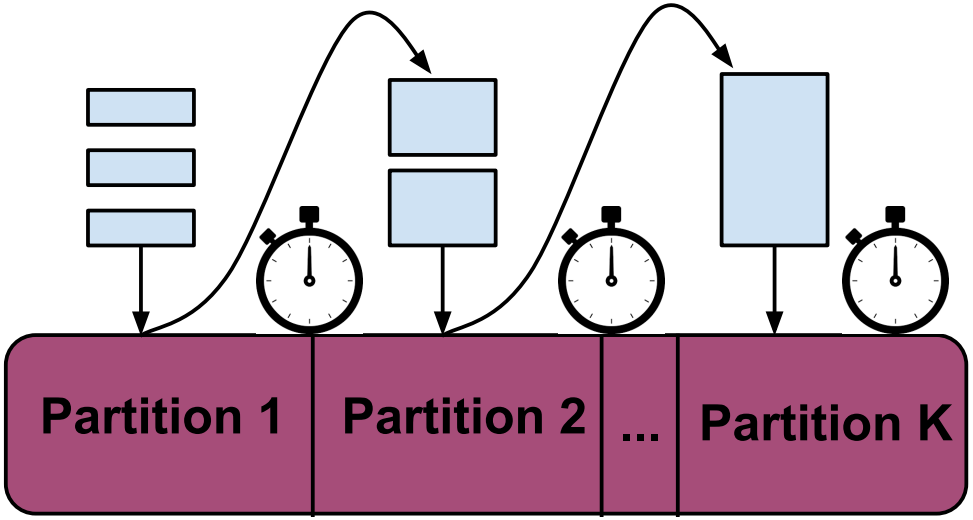
Improvements from Tyrex (2/2)



HVW

- Much smaller interquartile ranges and outliers with Tyrex.

Conclusions



- Main elements: resource partitioning and timers.
- Near-optimal performance when using dynamic timers.
- Tyrex cuts in half the job slowdown variability and preserves the median job slowdown.

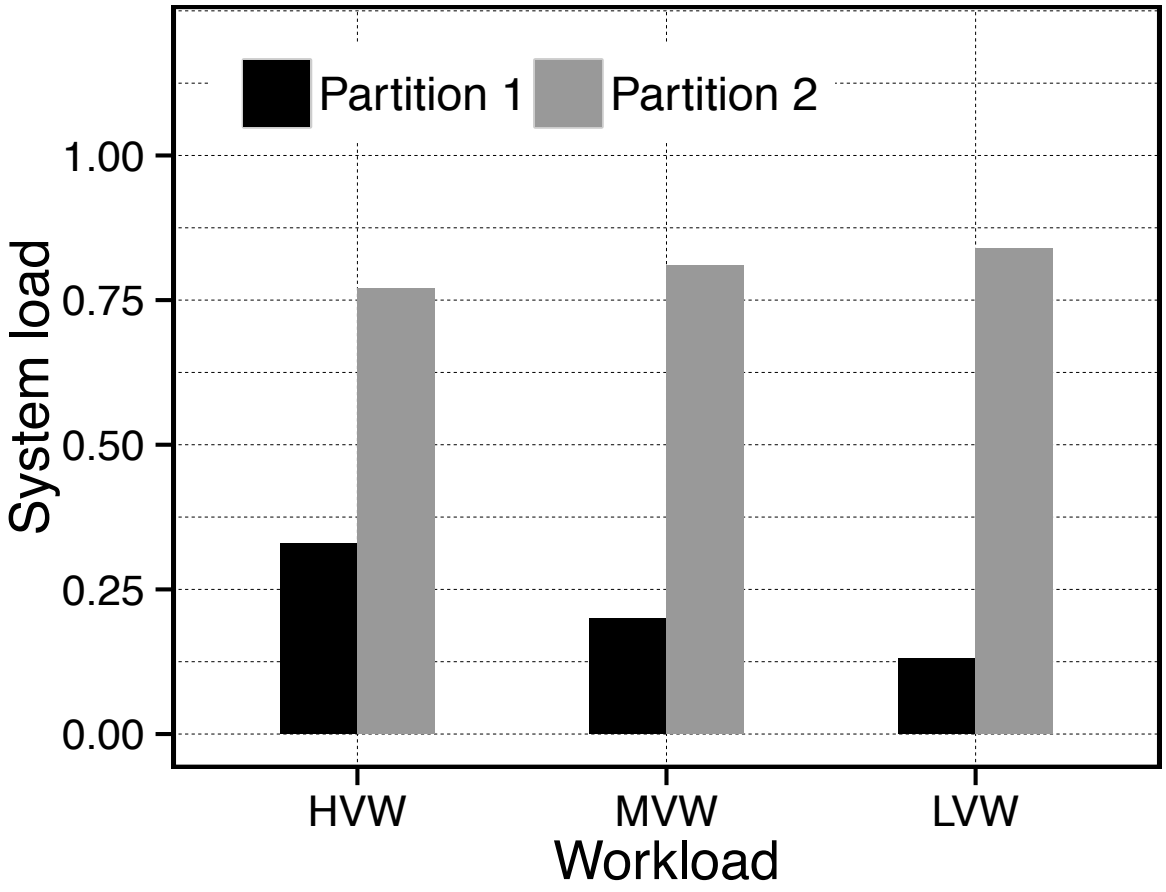
TUDelft Group at CCGrid 2016

	Tuesday	Thursday
10:00	BOGDAN GHIT (Venue A)	
10:30	BOGDAN GHIT (Venue C)	
14:00		A. Kuzmanovska (Venue B)
16:00	A. Kuzmanovska (Venue A)	

http://ds.ewi.tudelft.nl/

Backup slides

Unbalanced load across partitions



Good performance under high load