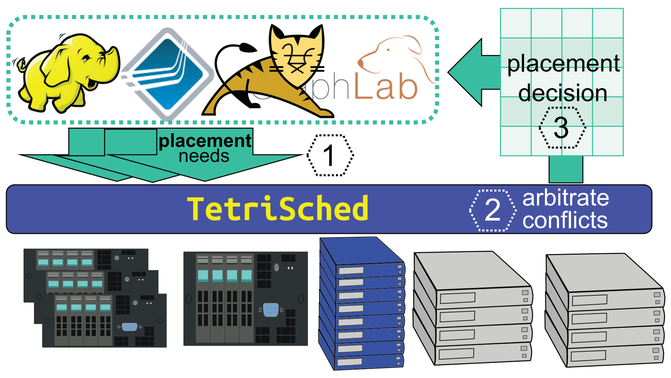
TetriSched: Space-time Soft Constraints in Heterogeneous Datacenters

Alexey Tumanov, Timothy Zhu, Jun Woo Park, Michael A. Kozuch*, Mor Harchol-Balter, Greg Ganger
Carnegie Mellon University, *Intel Labs

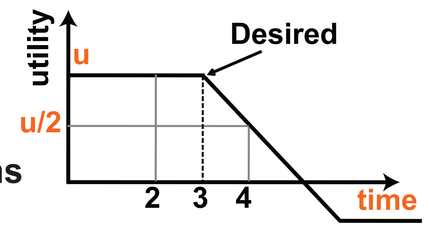
Background and Motivation

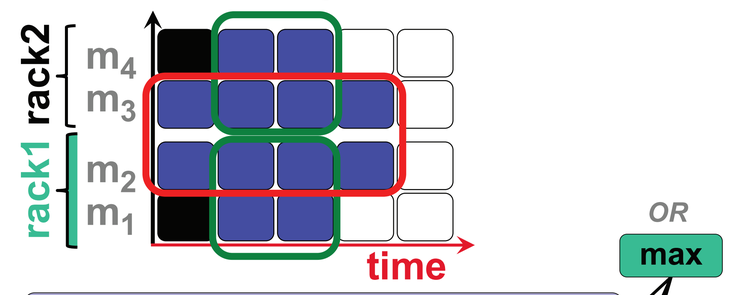


- Datacenters – increasingly heterogeneous
- Datacenter workloads – increasingly diverse
- User objectives – differ, conflict, change
- Cluster schedulers – map work to resources

Utility Functions

- User-defined utility functions
  - › Completion time
  - › Availability
  - › Queuing delay
- Scheduler-facing utility expressions
  - › "n Choose k" building blocks



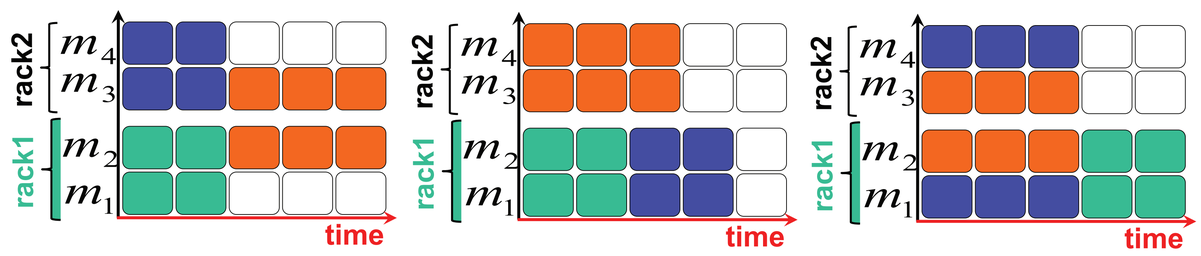


OR: max

rack1	nCk ($m_i \in$ rack1, k=2, s=1, d=2, u)
rack2	nCk ($m_i \in$ rack2, k=2, s=1, d=2, u)
anywhere	nCk ($\cup m_i$, k=2, s=0, d=4, u/2)

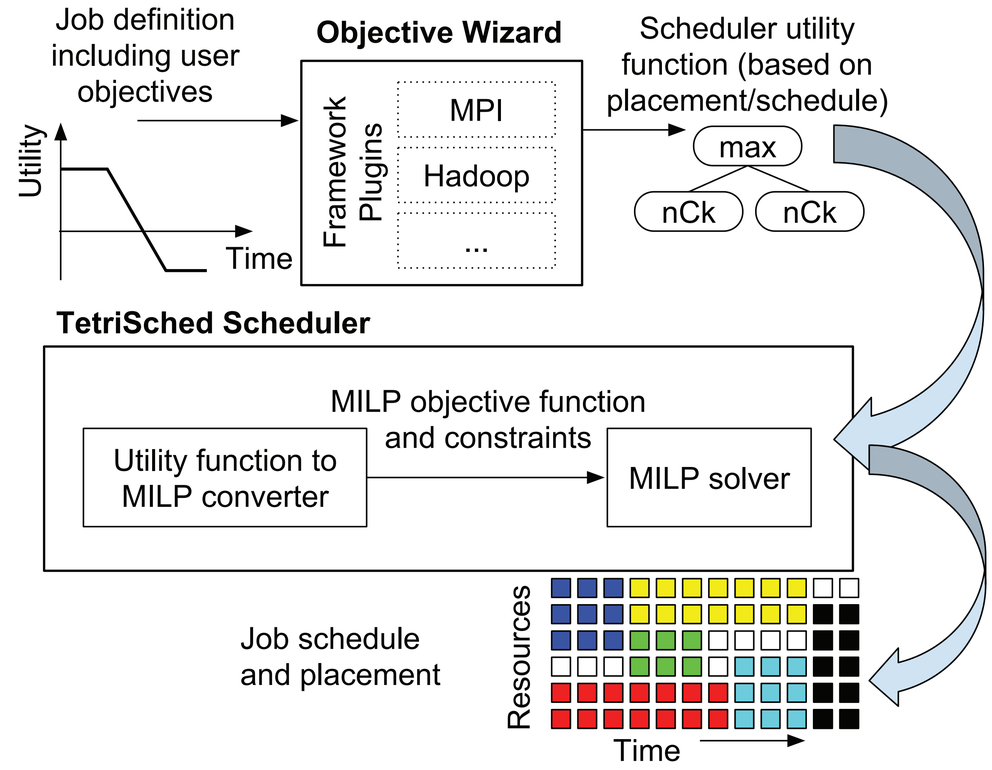
Problem Statement

- Heterogeneity results in many placement options
  - › Which resources/types to allocate (space)
  - › Run now or wait for better resource? (time)
- Existing schedulers don't leverage these options
  - › No interfaces to specify succinctly (or at all)
  - › No way to quantify the trade-offs
  - › Hard to efficiently solve : combinatorial solution space



- GPU: run 2 tasks on GPU nodes (rack1) if possible
- MPI: colocate 2 tasks on the same rack and complete ASAP
- Availability: place 2 tasks, each on a different rack

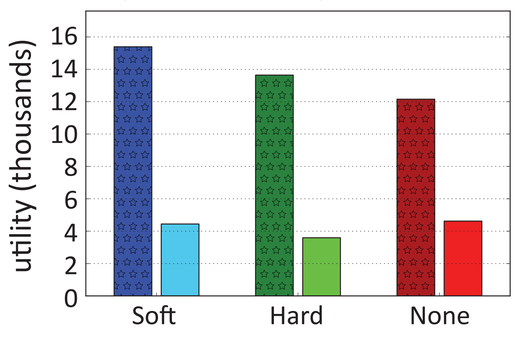
TetriSched System Model

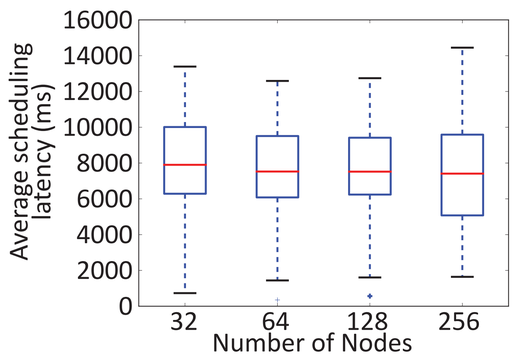


Real System Experiments

- TetriSched: outperforms YARN in all cases
- Hard & None ≥ Yarn-Hard & Yarn-None

[high load, heterogeneous]

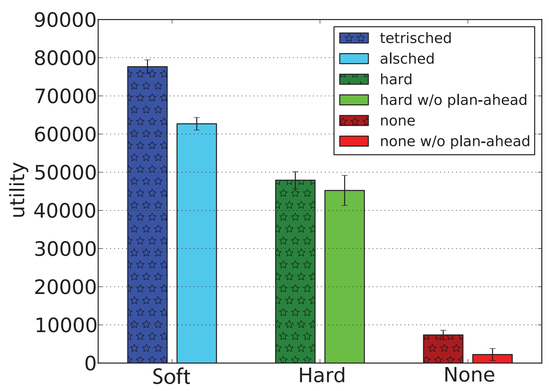


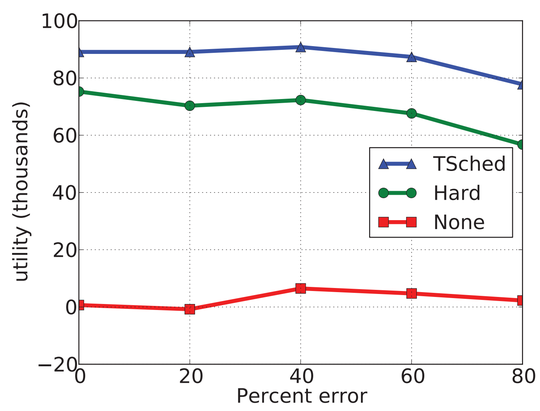


- TSched: TetriSched with soft constraints and plan-ahead (all in)
- YARN-Soft: Capacity Scheduler with AMs in relaxed locality mode
- Hard: placement constraints treated as required
- YARN-Hard: Capacity Scheduler with AMs in strict locality mode
- None: placement constraints ignored
- YARN-None: Capacity Scheduler with AMs in no locality mode

Simulation Results

- Simulated cluster: 1000 nodes, 25 racks
- Parameter sweeps for load, burstiness, plan-ahead, slowdown
- Variable workload compositions
- Robust to job runtime mis-estimation





Intel Science & Technology Center for Cloud Computing
NSERC CRSN
Carnegie Mellon University
Georgia Tech
intel
PRINCETON UNIVERSITY
UC Berkeley
UNIVERSITY of WASHINGTON