TetriSched: Space-Time Scheduling for Heterogeneous Datacenters

Alexey Tumanov

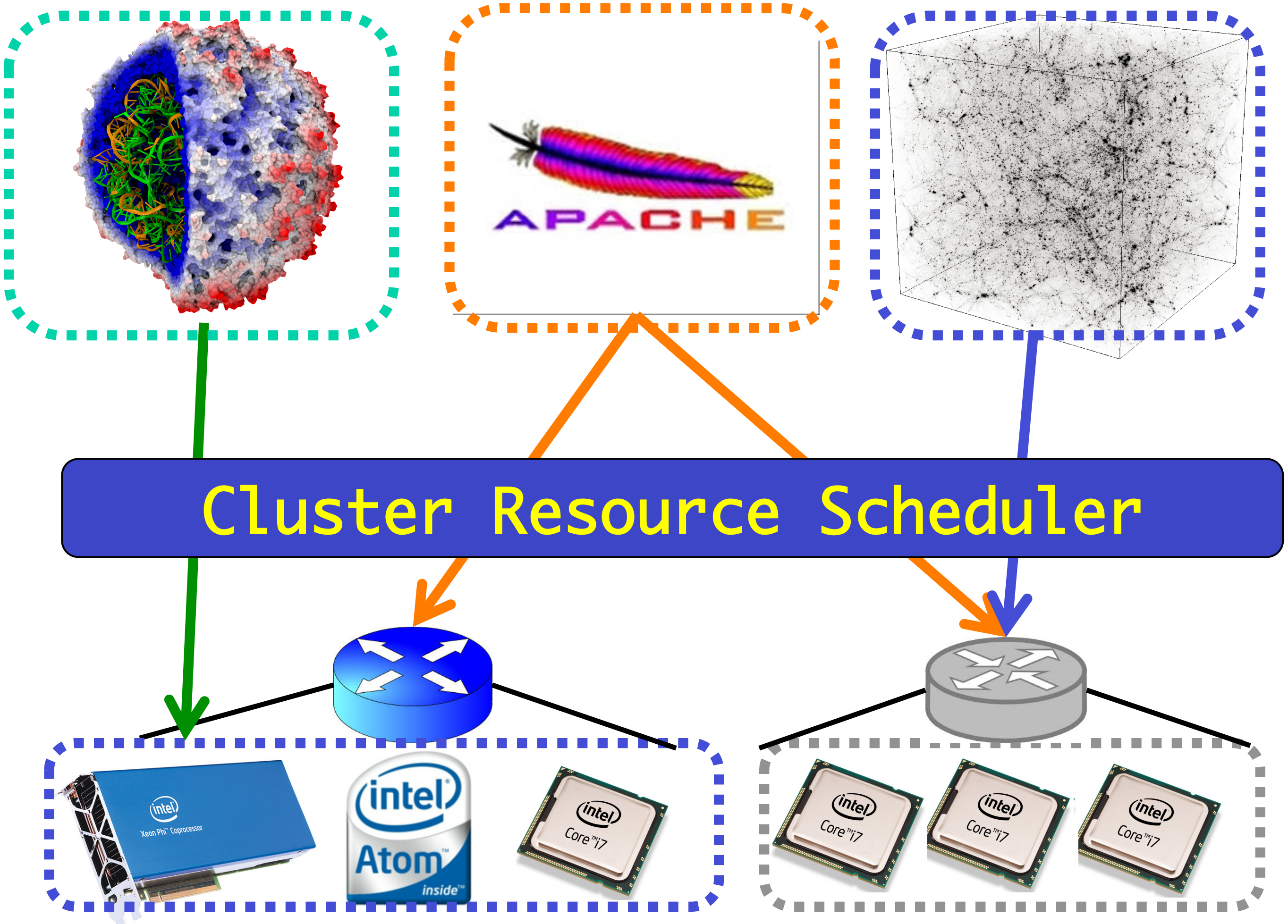
Timothy Zhu, Michael Kozuch,
Mor Harchol-Balter, Greg Ganger

Carnegie Mellon University

Intel Science & Technology
Center for Cloud Computing

http://www.istc-cc.cmu.edu/

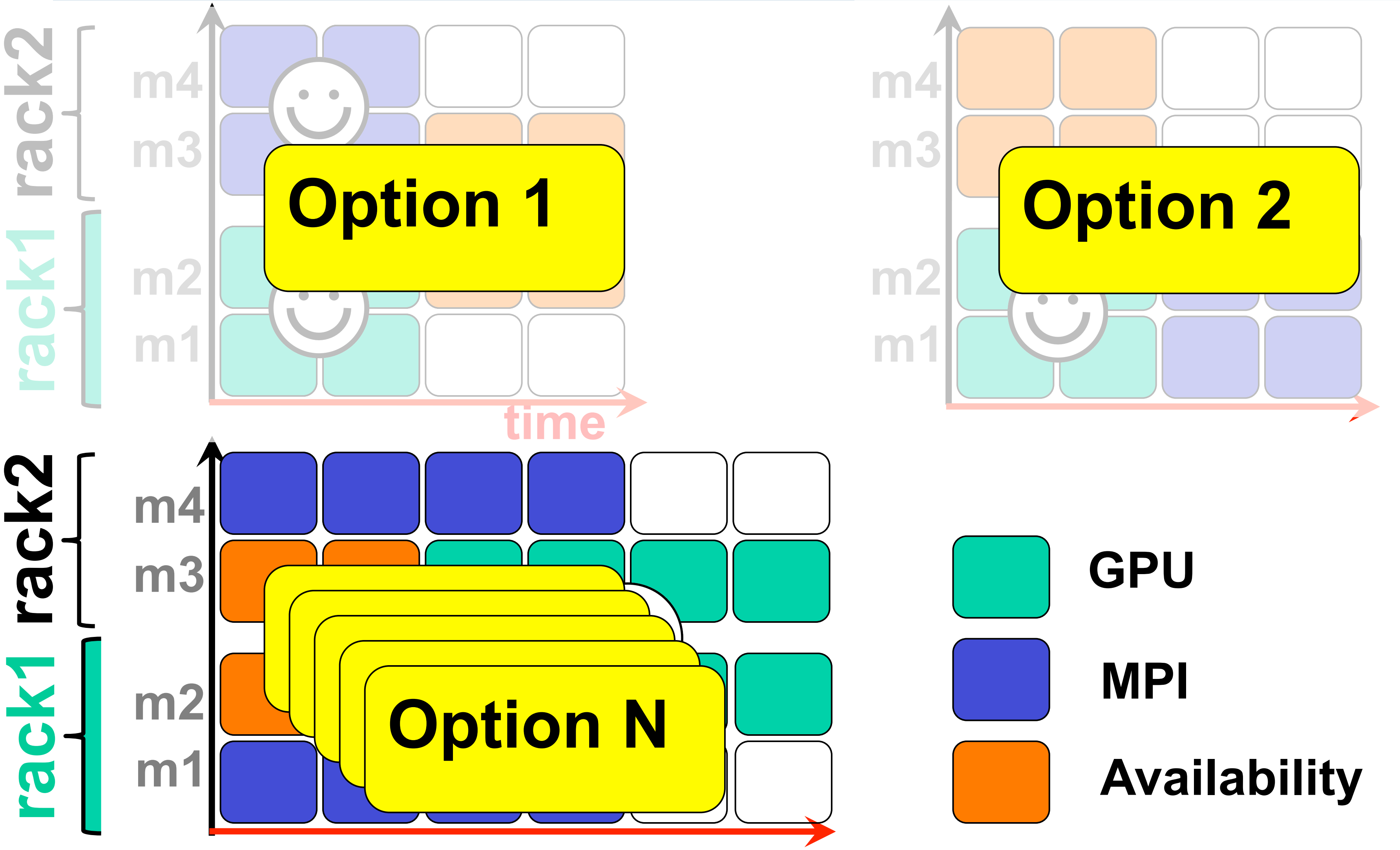
The Future of Datacenters



Consolidation

Specialization

Explosion of Choices + Tradeoffs

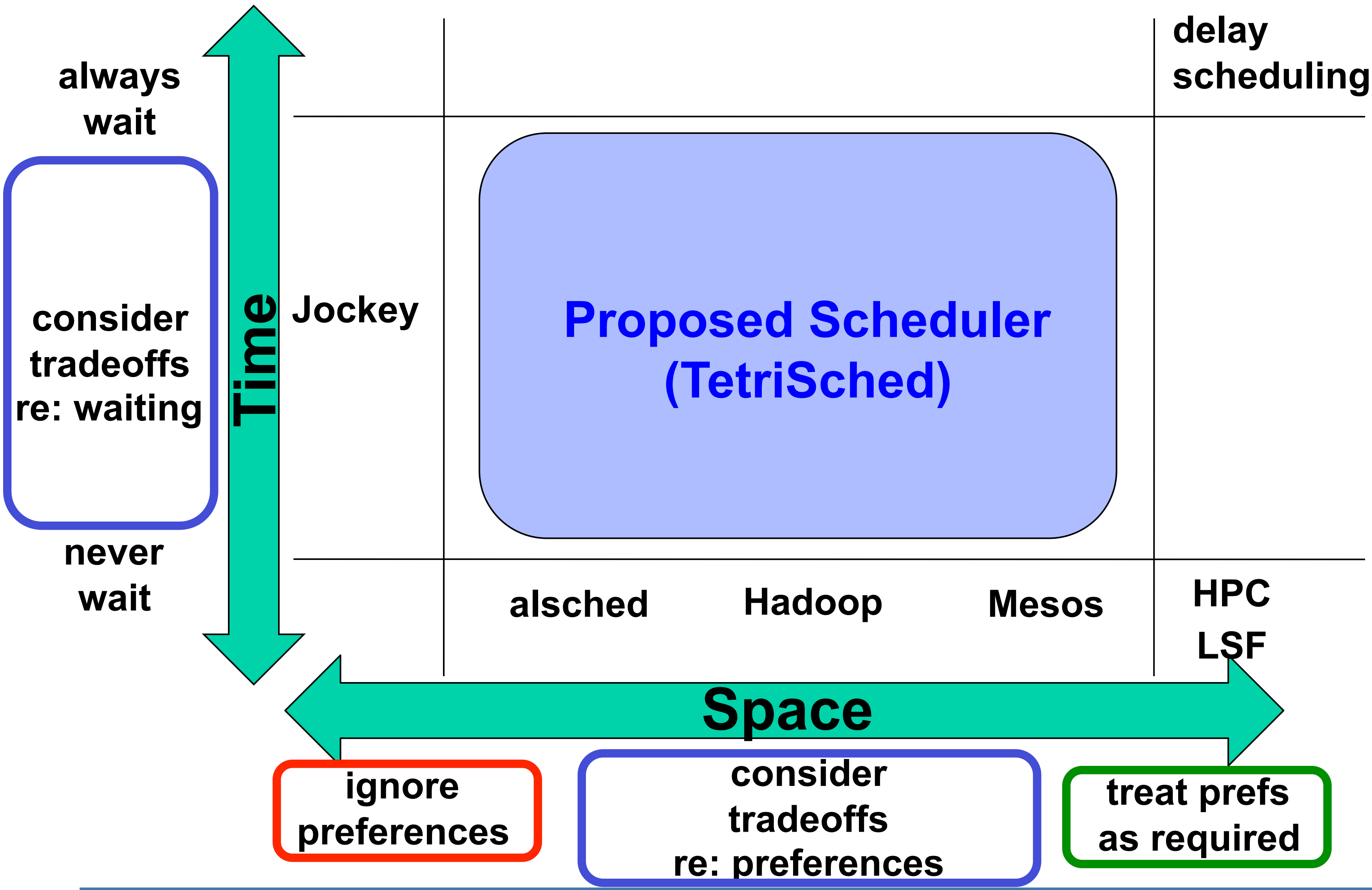


Necessary Ingredients

First-class support needed for:

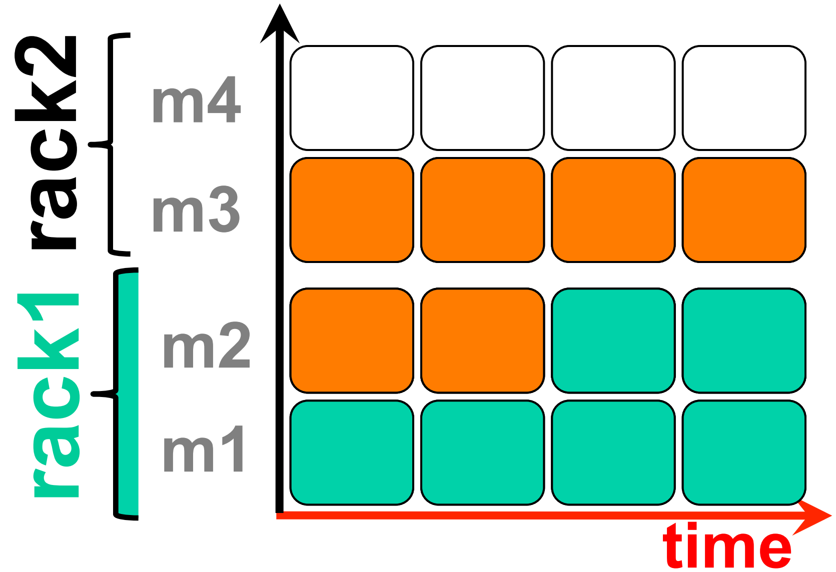
- Server-type preferences (FPGA > GPU > CPU)
- Server-set preferences (same rack <> diff racks)
- Timing preferences (right now == in 5 min > in 30 min)

Solution Space

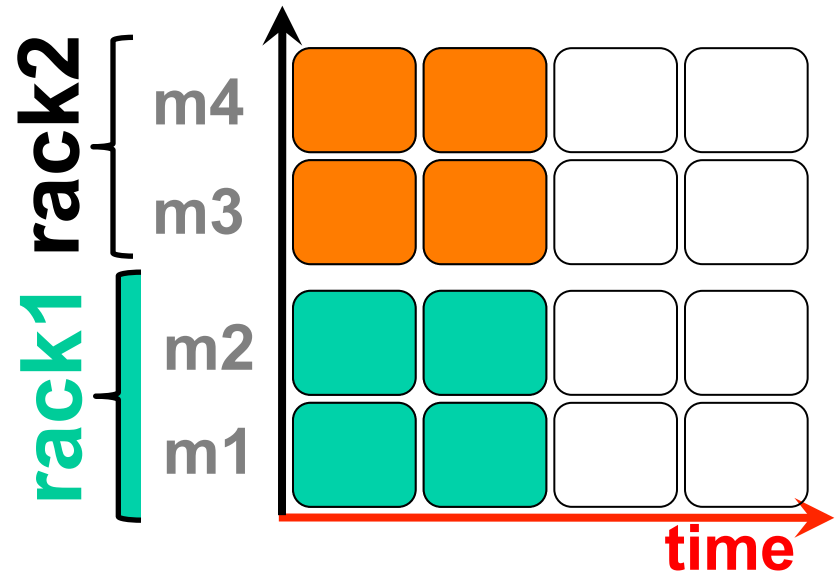


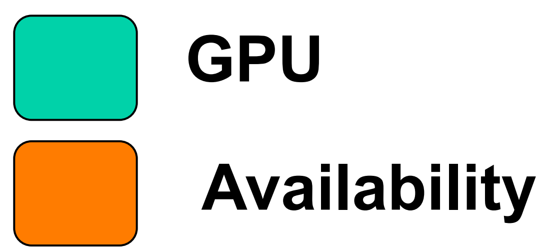
Benefits of Spatial Flexibility

Treat prefs as required



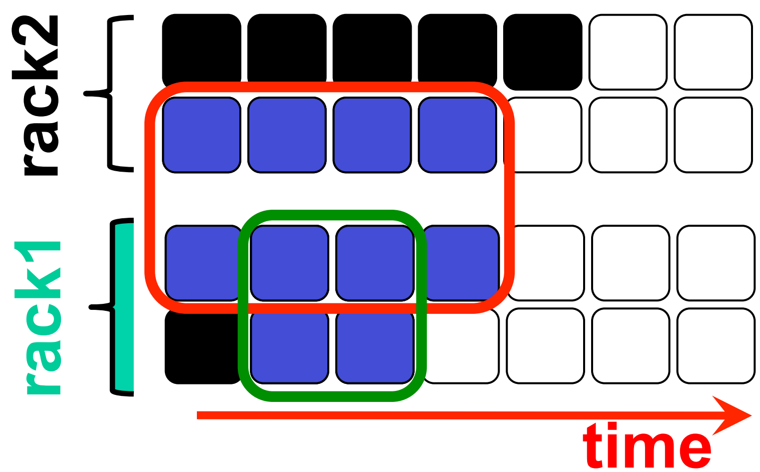
Consider tradeoffs





Benefits of Temporal Flexibility

- Previously: just look at current state
  - give each job the best option now, OR
  - wait indefinitely for the preferred placement
- Plan-ahead: estimate runtimes and choices
  - should this job wait for better placement?



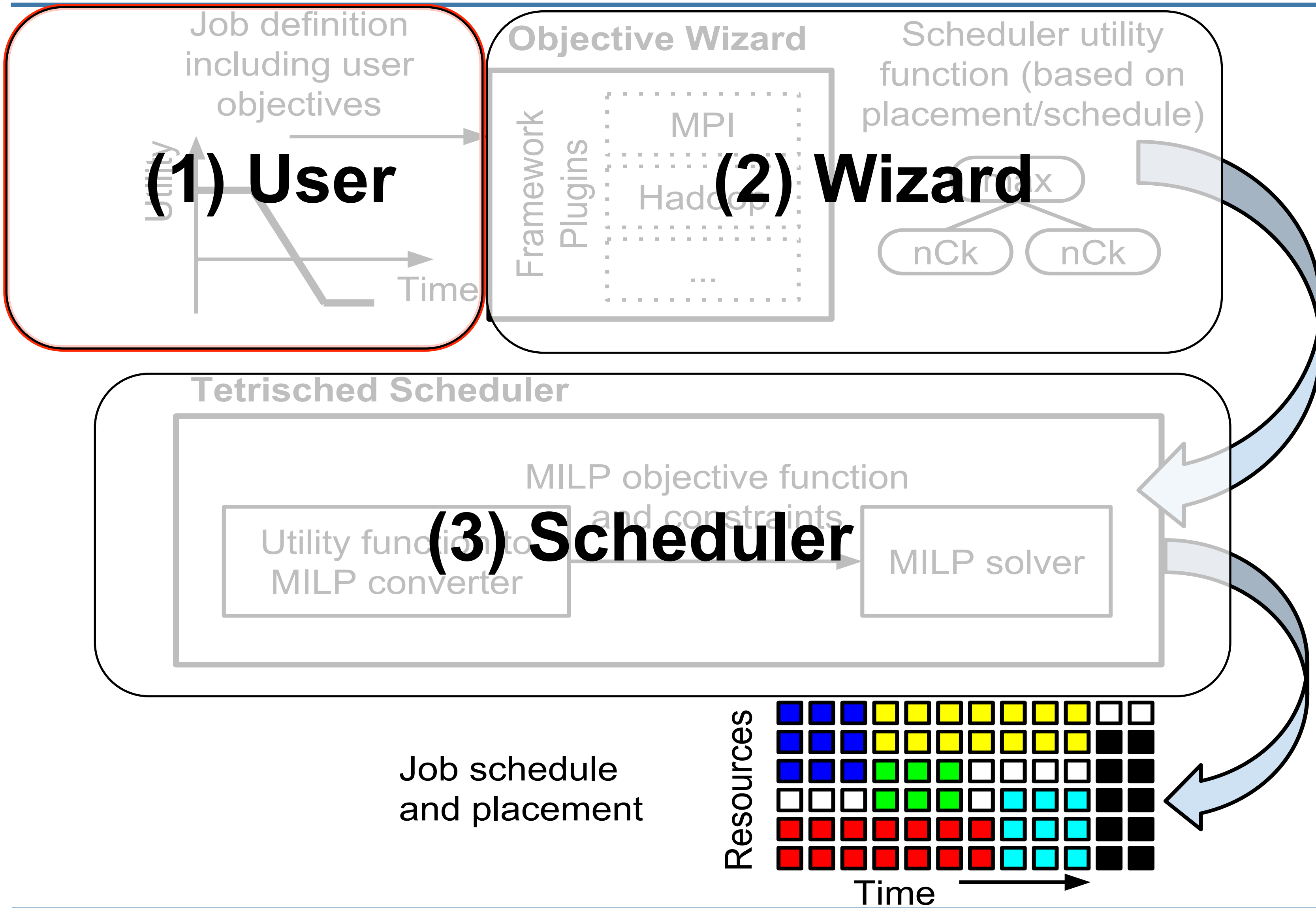
Key Questions

1. How does the user specify placement prefs and associated tradeoffs?

2. How can the scheduler efficiently cope with the combinatorial explosion of placement options?

3. How can the scheduler solve the aggregate, combinatorially complex scheduling problem?

TetriSched System Model



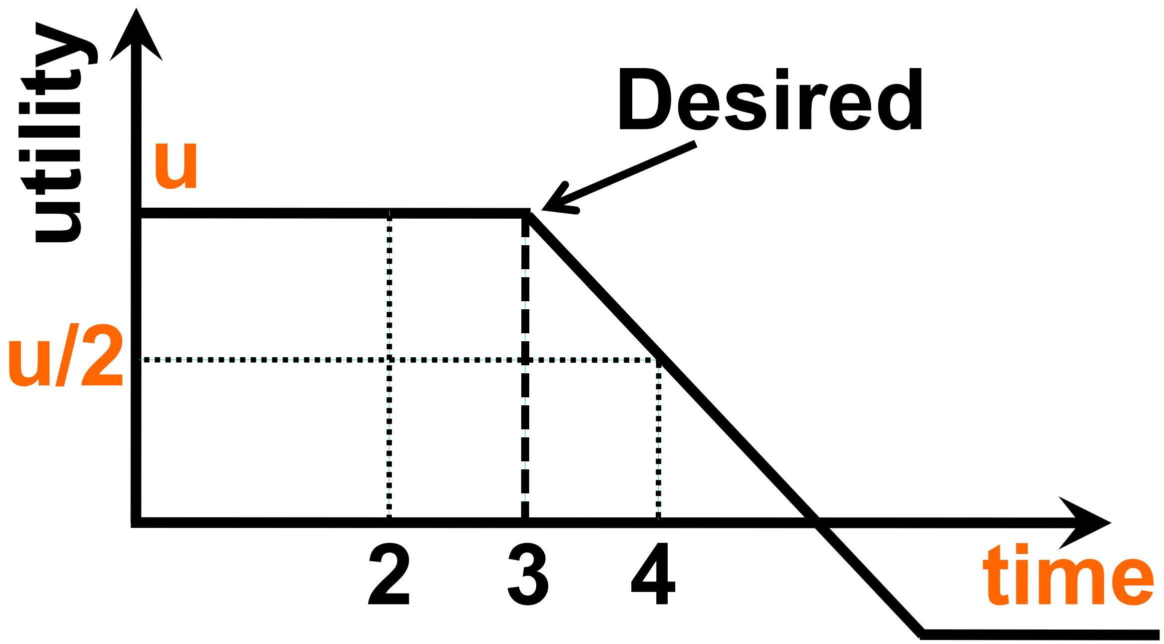
Utility

- User Objectives
  - response time (completion - arrival)
  - queueing delay (start - arrival)
  - availability (susceptibility to failure)

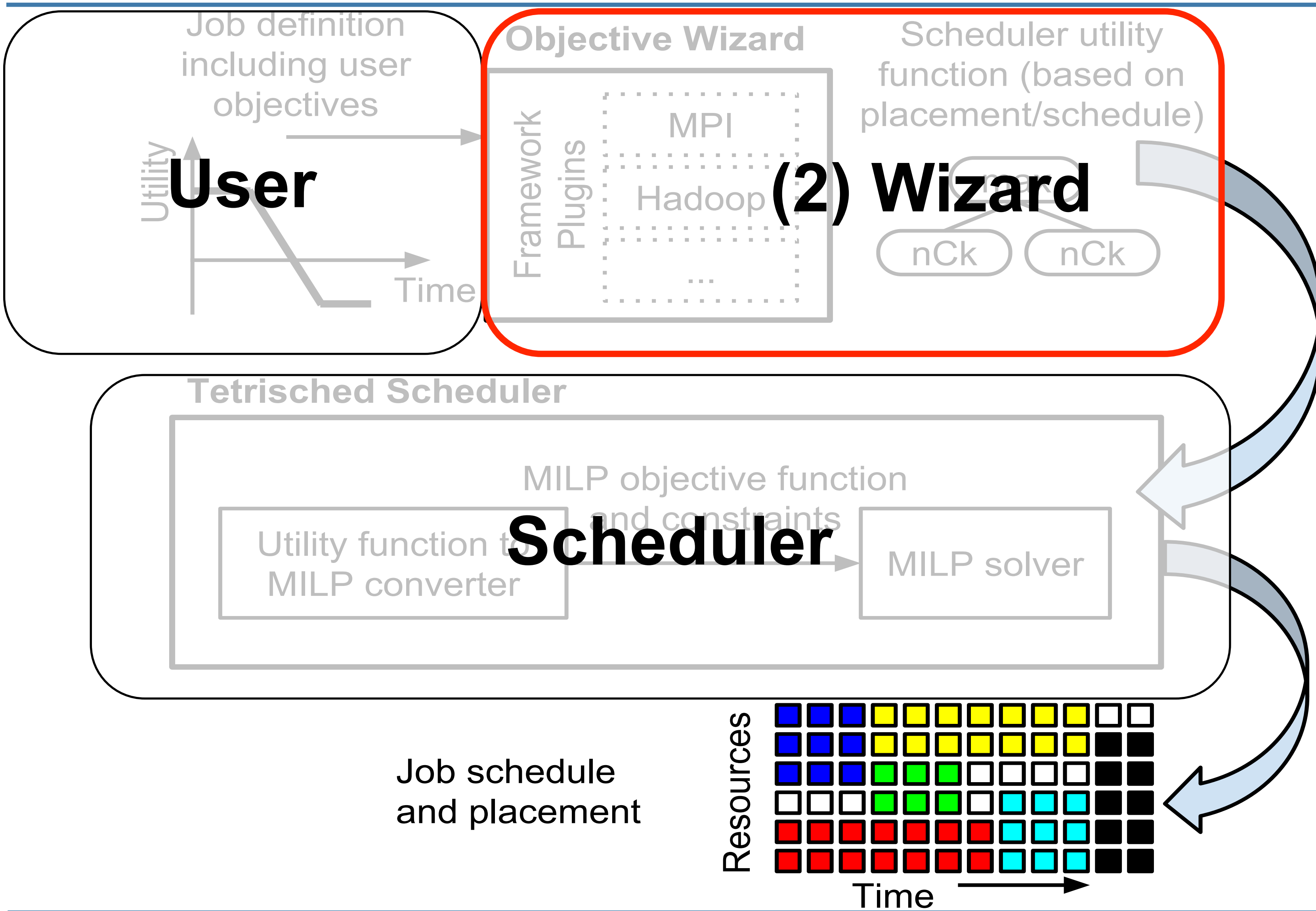
- Utility – common currency
  - a way of normalizing diverse objectives
  - each objective translated to generic utility
    - what am I willing to pay for …

Utility Example: MPI

- Example: MPI
  - fast duration (2) → higher utility
  - slow duration(4) → lower utility



System Model



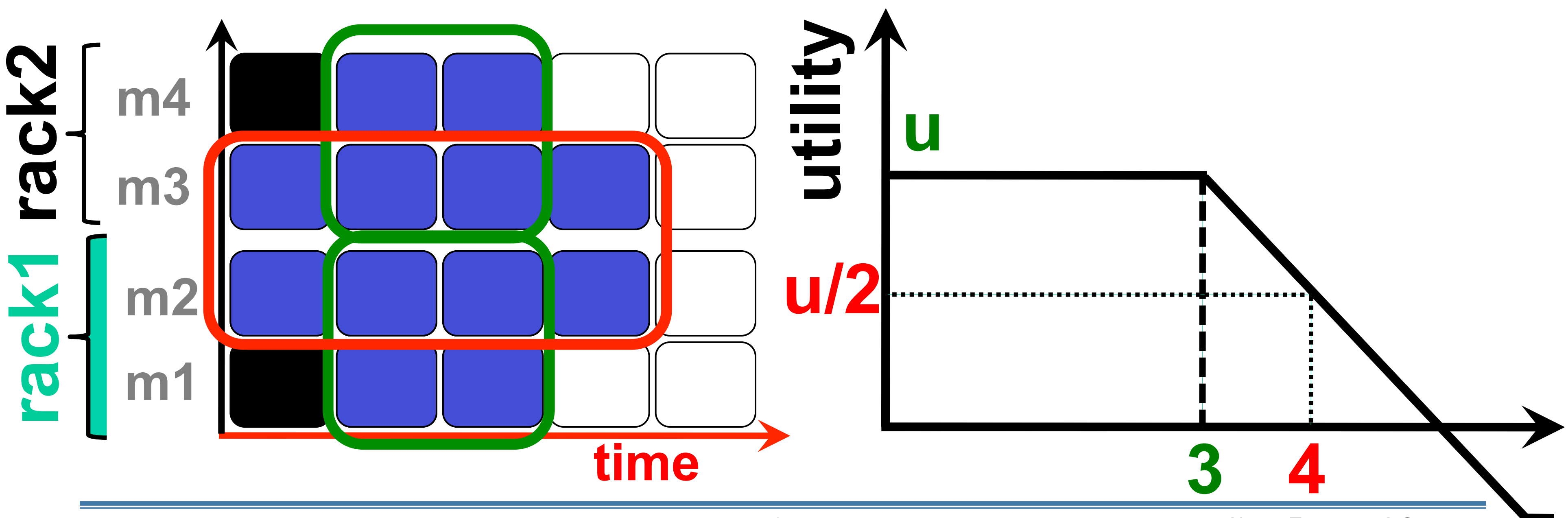
Objective Wizard

- Input: high-level user objectives
- Output: scheduler-centric utility functions
  - encoded in an algebraic expression language
- Translates user inputs to internal form
  - the form needed for the sched. optimization

Scheduler Expression Language

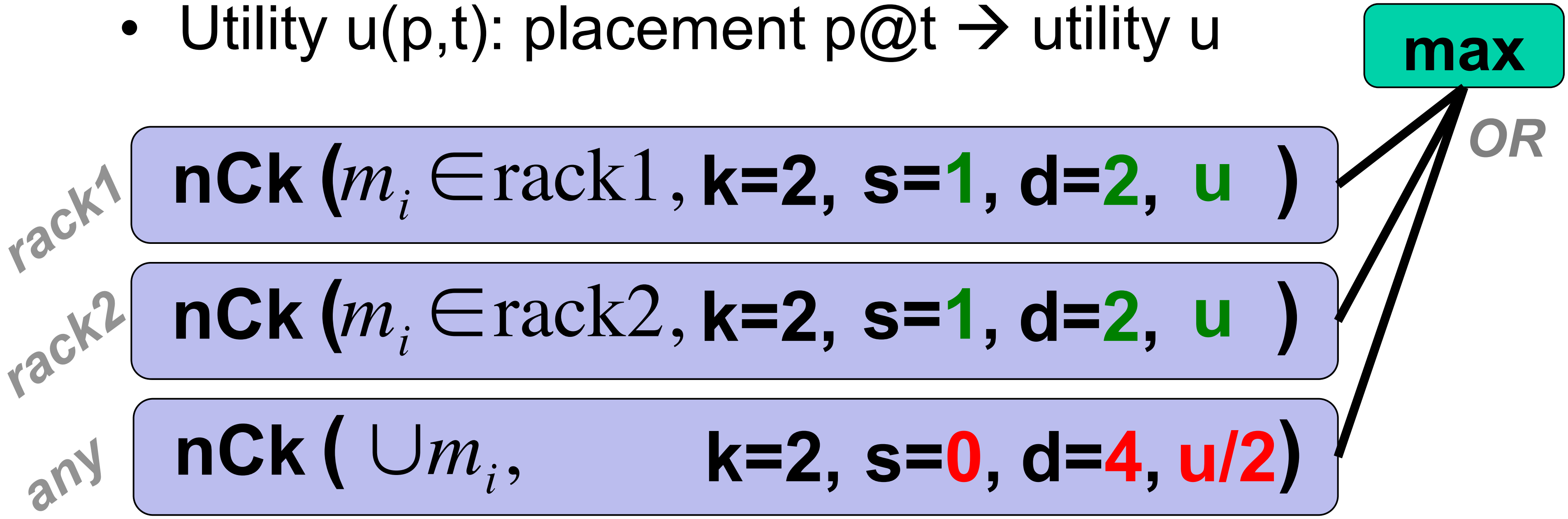
- Utility u(p,t): placement p@t → utility u

- **"n Choose k" (nCk)**
  n → refers to a group of nodes to choose from
  k → how many nodes to choose

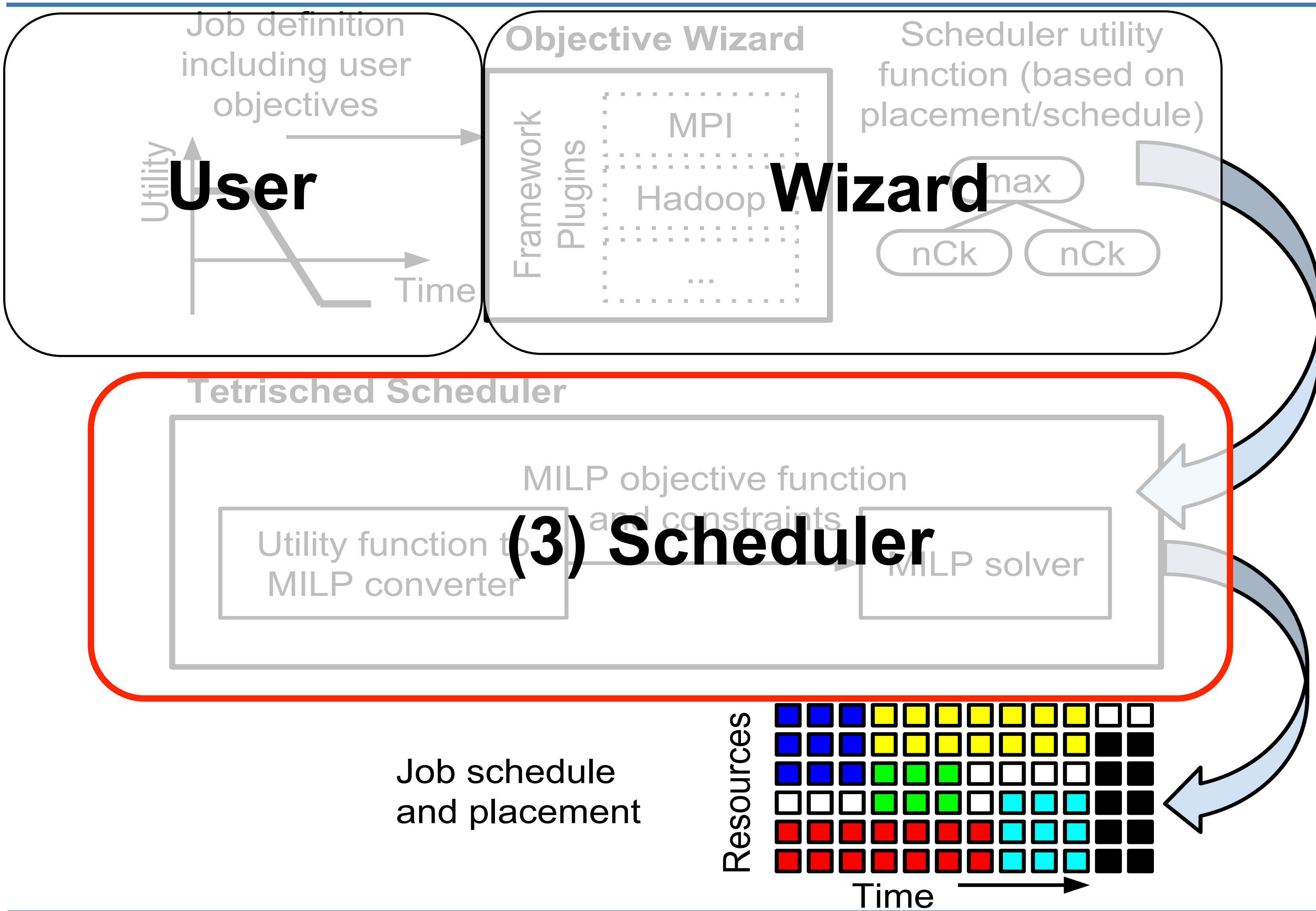


Scheduler Expression Composition

- Utility u(p,t): placement p@t → utility u



System Model

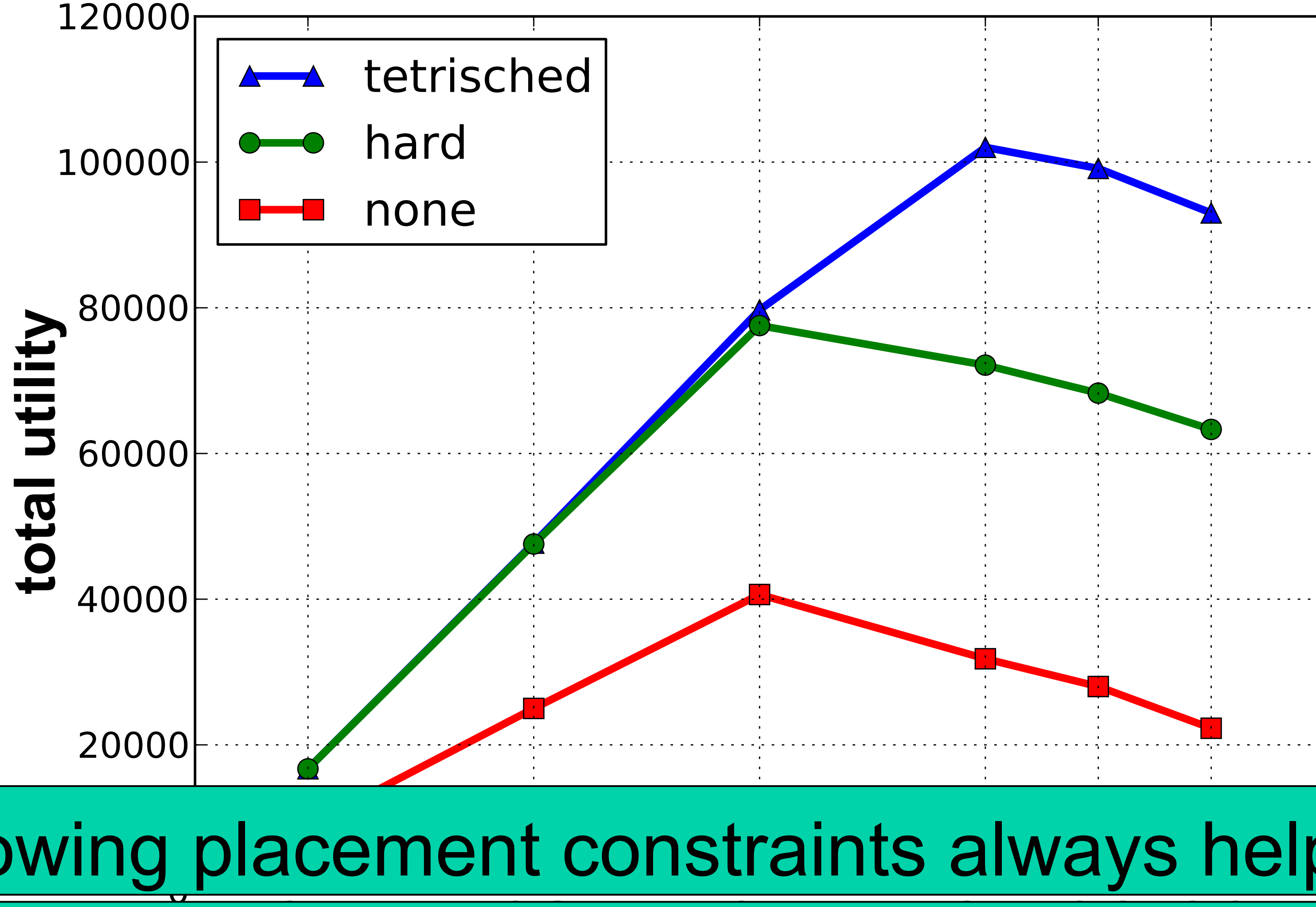


Experimental Results

- Real results from YARN-based TetriSched prototype on 73-node cluster
  - with soft constraints, plan-ahead, and gang-scheduling, TetriSched outperforms YARN
  - details on poster

- Simulation results for a 1000-node cluster
  - parameter sweeps for load, burstiness, plan-ahead, slowdown
  - variable workload compositions
  - sensitivity to job runtime mis-estimation

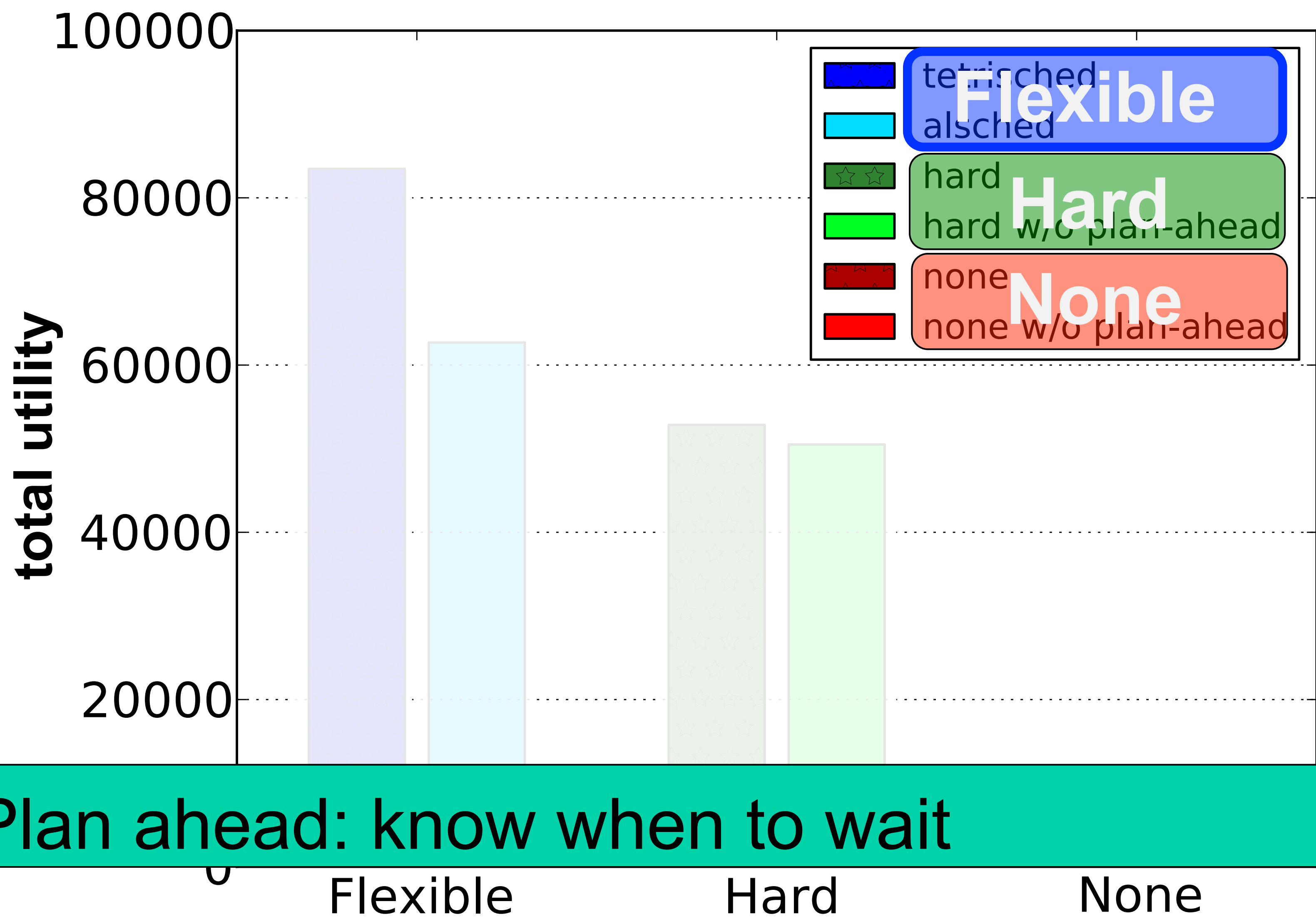
Benefits of Flexible Placement



Knowing placement constraints always helps

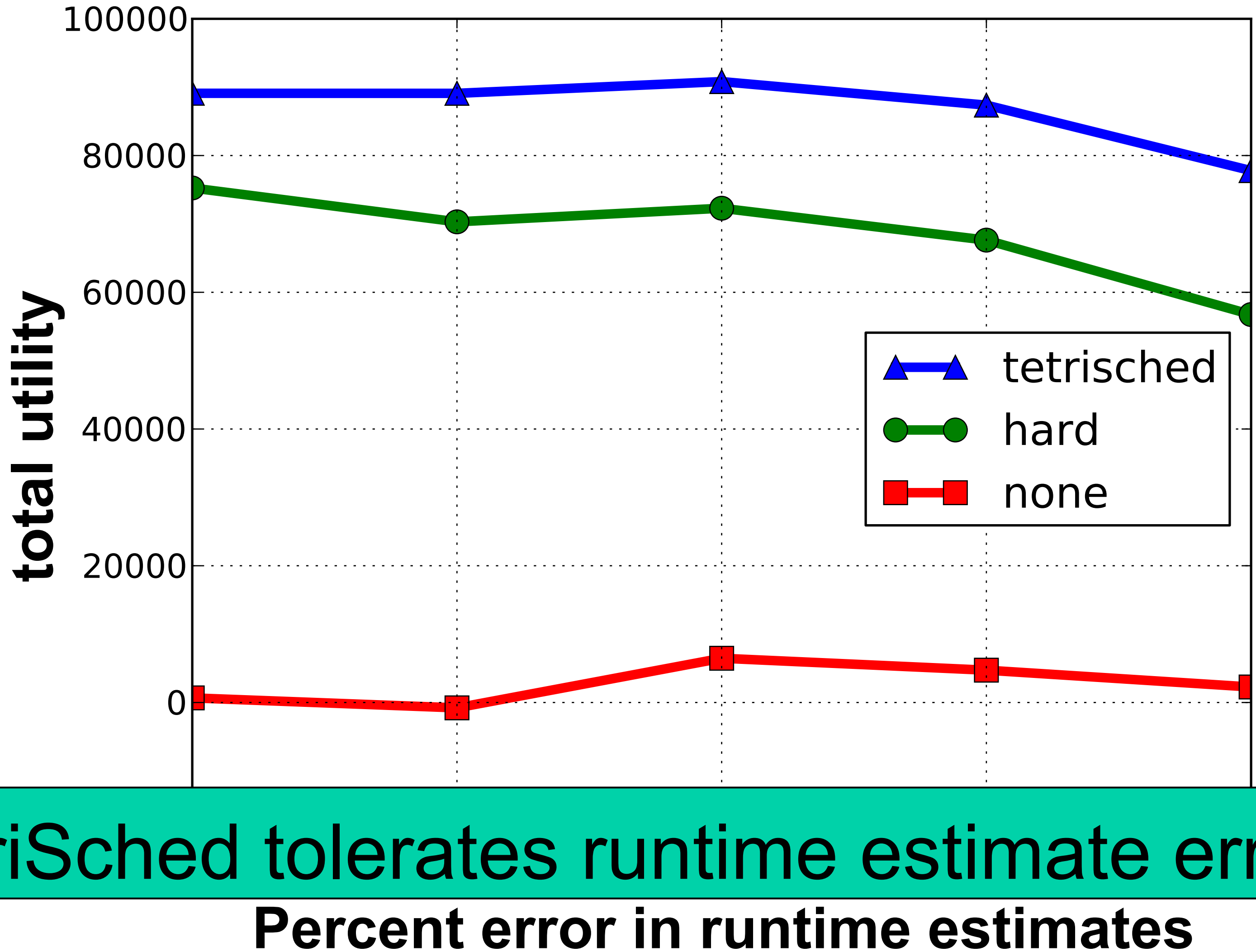
Placement flexibility enables better tradeoffs

Benefits of Plan-ahead



Plan ahead: know when to wait

When Users Err



TetriSched tolerates runtime estimate errors

Key Takeaways

- Problem: current schedulers don't cope with
  - increased heterogeneity in datacenters
  - explosion of tradeoffs and choices
- Solution: TetriSched explicitly enables
  - spatial flexibility aware scheduling (soft constraints)
  - temporal flexibility aware scheduling (plan-ahead)
- End result:
  - better schedules of heterogeneous mixes
  - easier adoption of specialized hardware

Related Work References

- A.Tumanov, J.Cipar, M.Kozuch, G.Ganger, "**alsched**: algebraic scheduling of mixed workloads in heterogeneous clouds", SoCC'12.
- C.Reiss, A.Tumanov, G.Ganger, R.Katz, M.Kozuch, "Heterogeneity and dynamicity of clouds at scale: Google trace analysis", SoCC'12.
- B.Hindman, A.Konwinski, M.Zaharia et al, "Mesos: a platform for fine-grained resource sharing in the data center", NSDI'2011.
- M.Jette, M.Grondona, "SLURM: simple Linux utility for resource management", Lawrence Livermore National Lab, ClusterWorld Conf&Expo'03, 2003.
- V.Vavilapalli et al, "Apache Hadoop YARN: yet another resource negotiator", SoCC'13, 2013.
- M.Schwarzkopf, A.Konwinski, M.Abd-El-Malek,J.Wilkes, "Omega: flexible, scalable schedulers for large compute clusters", Eurosys'13.
- A.Ghodsi, M.Zaharia, S.Shenker, I.Stoica, "Choosy: max-min fair sharing for datacenter jobs with constraints", EuroSys'13.
- J.Wilkes, "Utility functions, prices, and negotiation", HP Labs, Tech.Rep. HPL-2008-81, 2008.