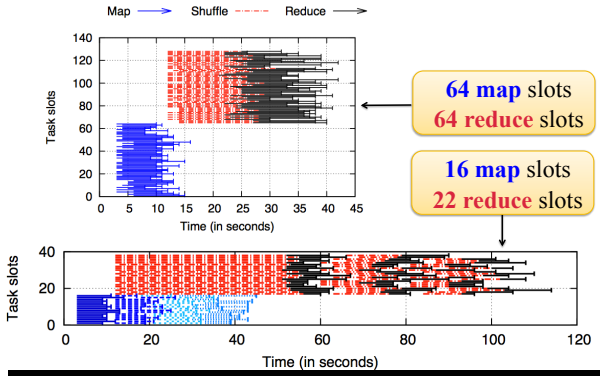
Three Pieces of the MapReduce Workload Management Puzzle

Abhishek Verma*, Ludmila Cherkasova#, Vijay S. Kumar#, Roy H. Campbell*
*{verma7, rhc}@illinois.edu University of Illinois at Urbana-Champaign, #{lucy.cherkasova, vijay.s.kumar}@hp.com HP Labs, Palo Alto

Motivation

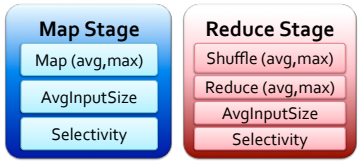
- **Problem:** Existing job schedulers do not support Service Level Objectives
- Often MapReduce applications are a part of critical business pipelines and require job completion time guarantees (SLOs)
- **Goal:** Design a workload management framework for efficient processing of MapReduce jobs with completion time goals in **shared** environments

Job Execution with Different Resources



Job Profiles and MapReduce Performance Model

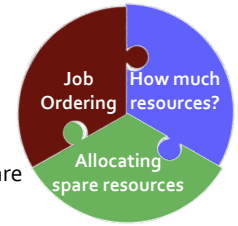
- Job Profiles compactly summarize performance metrics of different job stages collected from logs



- Automatic Resource Inference and Allocation (ARIA) with **novel performance** models:
  - Can predict job completion time = f(resources)
  - Given a deadline for job, compute minimum resources

Three Pieces of the Puzzle

1. **Job Ordering**
   - How to order jobs?
2. **Tailoring amount of resources**
   - How many slots to allocate?
3. **Allocating spare resources**
   - How to allocate and de-allocate spare resources?



Job Scheduling using Different Mechanisms

1. **Earliest Deadline First**
   - Allocate all resources to the job with EDF
2. **Min-EDF**
   - Compute and allocate minimum resources
3. **Min-EDF-WC**
   - Allocate any spare resources among running jobs
   - When new job arrives, compute if enough slots will be released in the future to satisfy current job
   - If not, cancel spare tasks of the currently running jobs

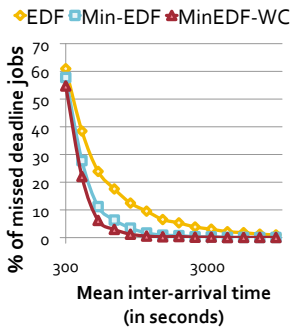
Evaluation Setup and Workloads

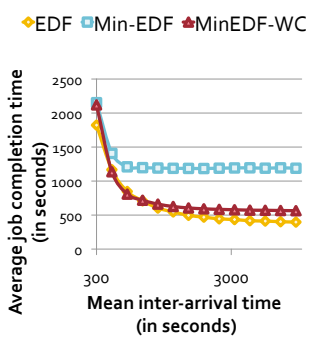
- **Testbed Setup**
  - 66 HP machines: 2 masters + 64 workers
  - Four 2.39 GHz cores, 8 GB RAM, 2 x 160 GB hard disks
- **Workloads**
  - **Real testbed trace** of 1000 jobs with combinations of: Wordcount, Sort, Bayesian classification, TF-IDF, WikiTrends, Twitter on 3 different datasets
  - **Synthetic Facebook trace**: generated using LogNormal distribution fit to 6 months of jobs

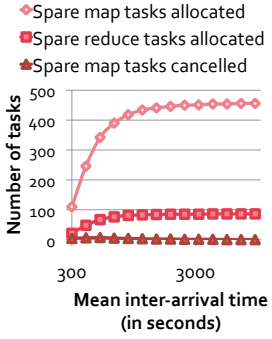
Simulator SimMR

- **Replay traces using *SimMR***
  - Discrete event simulator replays job traces at task-level
- **Speed**
  - Can replay two week workload in 2 seconds
- **Accuracy > 95%**
  - Simulated job completion time within 5% of real completion time

Evaluation







The simulation results with the synthetic Facebook trace are similar and reflect the same conclusions.

Conclusion & Future Work

- All three mechanisms are required for deadline-based workload management
- **Dynamic resource adjustment**
  - Compare expected behavior against observed behavior and adjust
  - Deal with stragglers
  - Input data skew