MRM: Delivering Predictability and Service Differentiation in Shared Compute Clusters

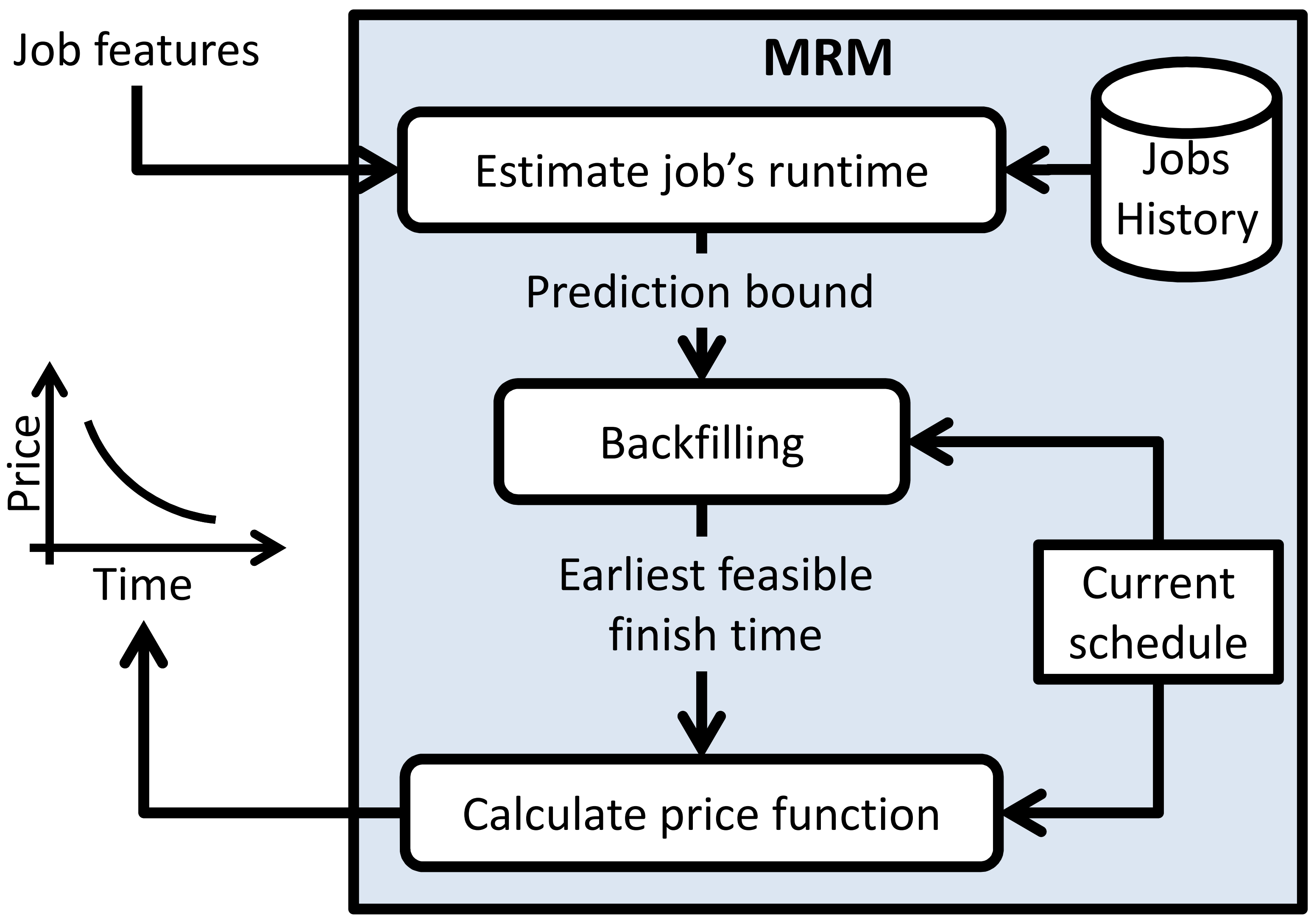
Masoud Moshref, Abhishek B. Sharma, Harsha V. Madhyastha, Leana Golubchik, Ramesh Govindan

Motivation

- Scheduling jobs in a shared cluster
  - Predictability: To know job finish time
  - Service differentiation: to finish sooner than previously enqueued jobs
- Current practice
  - FCFS: Predictability, ~~Service differentiation~~
  - Priority queue: ~~Predictability~~, Service differentiation

MRM Solution

1. **Predict each job's duration using history**
   - Jobs run multiple times with pseudo-similar features
   - Find prediction upper bound using confidence interval of error
2. **Find earliest feasible finish time based on**
   - Predicted duration bound of new job
   - Predicted duration bound and deadlines of currently scheduled jobs
3. **Present a price-deadline curve to user**
   - Pricing motivates users to select later deadlines
   - Calculated based on
     - Slack of a deadline
     - Scheduled jobs (load) in the system



Job features

MRM

Estimate job's runtime

Jobs History

Prediction bound

Backfilling

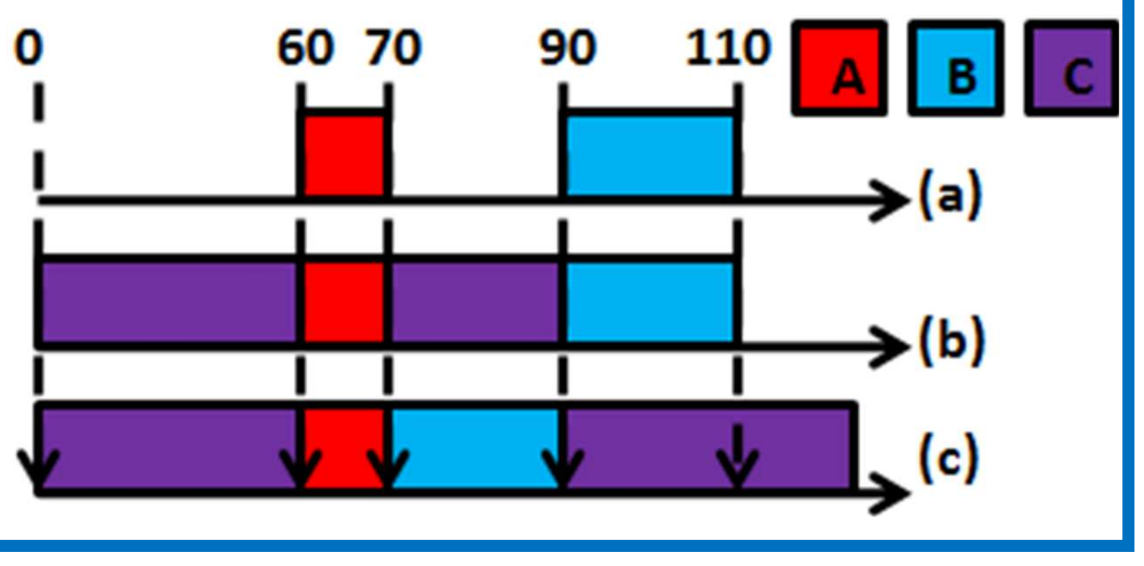
Earliest feasible finish time

Current schedule

Calculate price function

Predictability

- **Job features**
  - # Input records
  - # map and reduce tasks
  - Map and reduce reduction factor
- **Predict each job's duration + error of prediction**
  - Gaussian Process Regression: mean, error std
- **Find earliest feasible finish time**
  - Can have holes in schedule
  - Assume prediction error is Gaussian
  - Find 95% bound per job
  - Backfill the holes



Service Differentiation

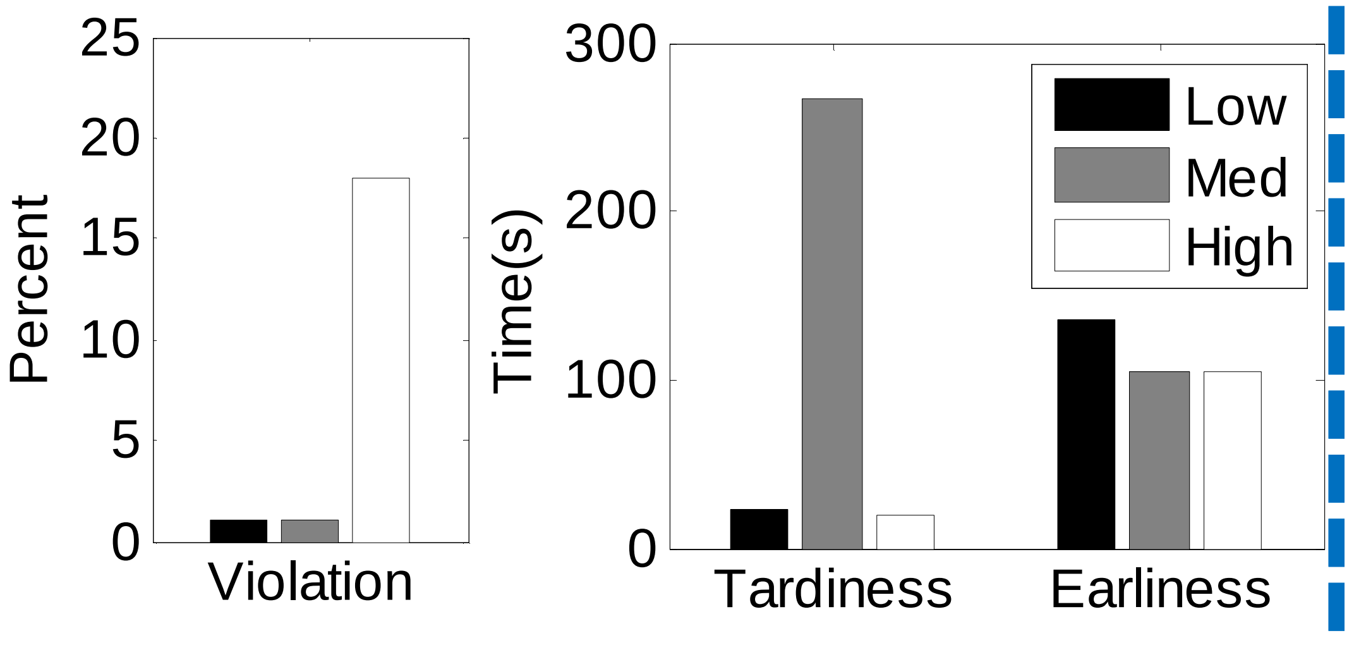
- **Design principles for price function**
  - A non-linear decreasing function of slack
  - Consider the load in system
  - Use predicted duration of jobs
  - Consider the purchasing power of users
- **Theoretical modeling on a simplified system**
  - Work conserving slotted system with job size=1
  - Deadline sensitive jobs
    - Only take slot 0 if free otherwise $1 penalty
  - Price deadlines for delay tolerant jobs such that
    - Compensate for penalties they may cause
- **Generalized function**

$$f(\delta) = \frac{\kappa}{\delta + 1} \qquad c(\delta_j, p_j) = p_j f(\delta_j) + \sum_{i \in Q} p_i \Delta f_i$$

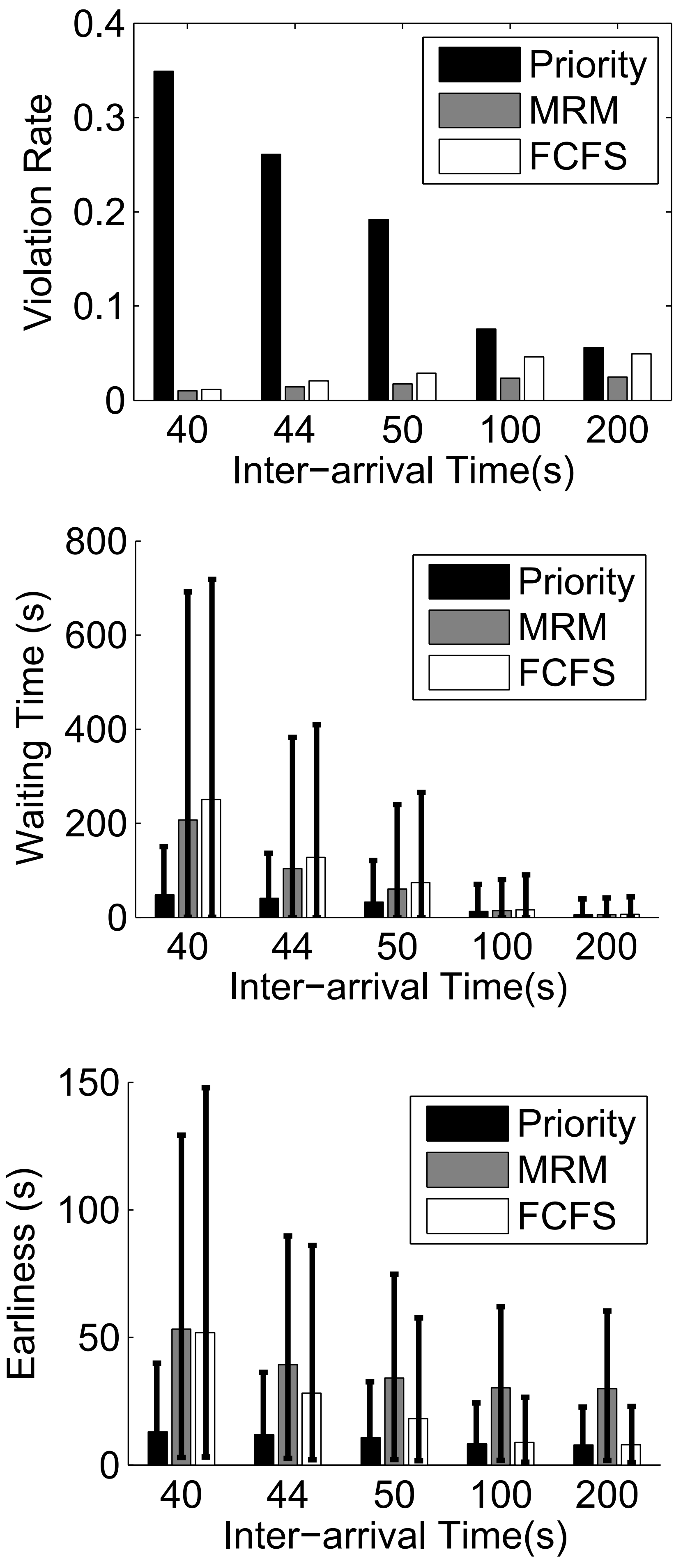
Evaluation

- Evaluation setting: History of Map-Reduce jobs (Grep, word-count, Pi estimator, Sort) on 40 servers
- Earliness (How loose was deadline) is also important

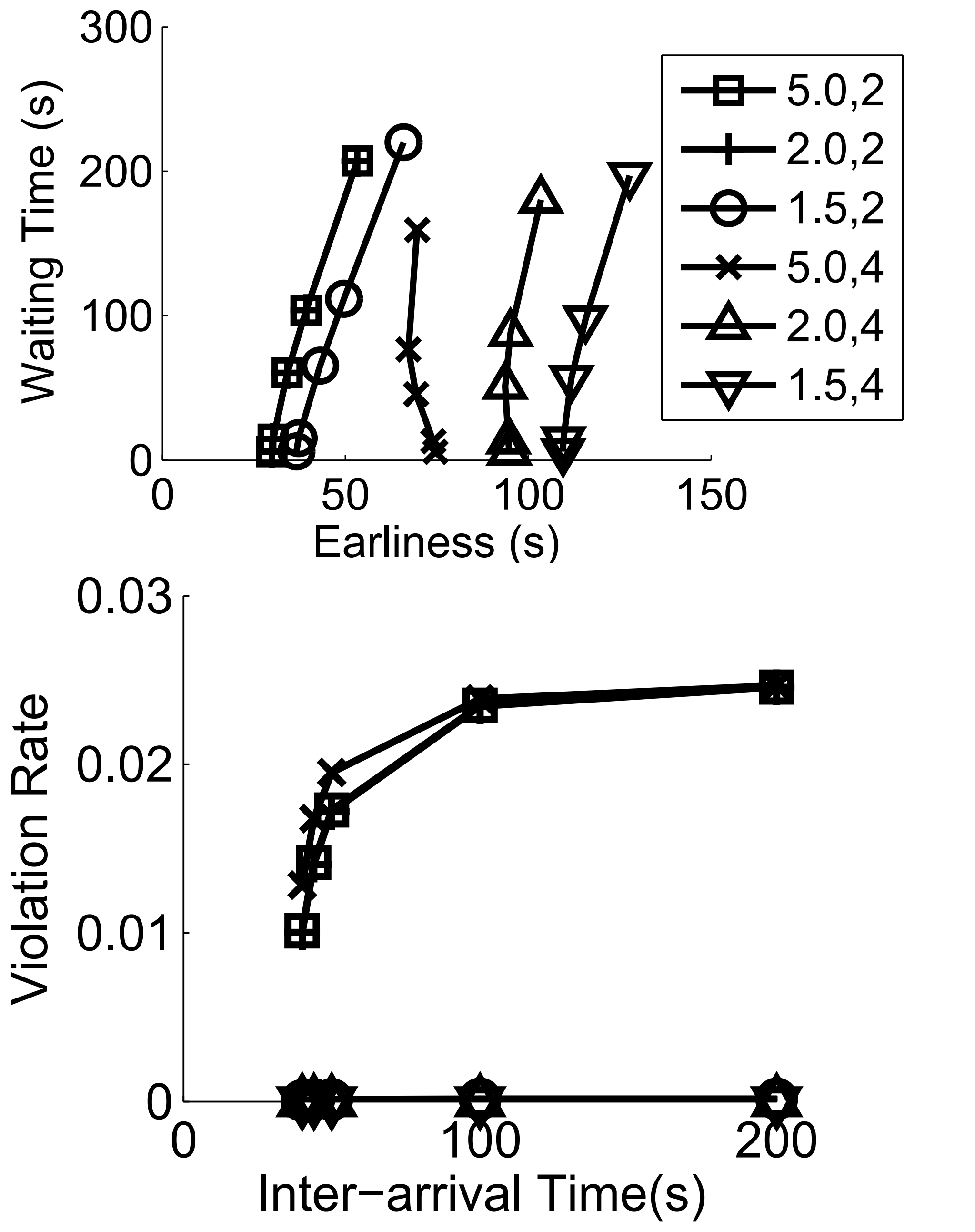
Predictability on real cluster with zero slack (FCFS)



Price function trace-based simulation



Purchasing power effect



Conclusion

- MRM provides predictability and service differentiation
- A design point between FCFS and priority queue
- **Future work**
  - Consider failure in job processing time
  - Feedback deadline violations to scheduler
  - Evaluate on more complex jobs
  - More specific job types with a richer feature set

USC NSL

Contact: moshrefj@usc.edu