Gemini: An Adaptive Performance-Fairness Scheduler for Data-Intensive Cluster Computing

Zhaojie Niu, ShanjiangTang, Bingsheng He
Nanyang Technological University

Outline

- **Background**
- Our Proposal: Gemini
- Experiments
- Summary

Hadoop YARN

- What is Hadoop YARN
  - New generation of Hadoop
  - An unified resource manager for data-intensive applications
- Schedulers in Hadoop YARN
  - FIFO: first come first service
  - Fair (DRF): assign resources fairly among users
  - Capacity: maximize the utilization of multi-tenant cluster
- Hadoop YARN grows quickly
  - Amazon, Cloudera, Hortonworks, IBM, et al.

Tradeoff between Performance and Fairness

User A (CPU intensive): <2 CPU cores, 1GB memory> A

User B (memory intensive): <2 CPU cores, 4GB memory> B

6 CPU cores | 9GB memory | 6 CPU cores | 9GB memory

Trade-off

Fairness-aware ⟷ Performance-aware

Tetris [**sigcomm'14**]

Problem Definition

- There is a trade-off between the performance and the fairness.

For the same fairness level, our system can achieve better performance, or vice versa.

Focus: optimize the performance
see the fairness optimization in our paper

Outline

- Introduction
- **Our Proposal: Gemini**
- Experiments
- Summary

Our Proposal: **Gemini**

- Gemini is a workload-aware scheduler which can adaptively decide the proper policy according to current running workload.

→ A model to characterize the workload and leverage it to guide the scheduling

→A adaptive scheduler which dynamically chooses the most proper policy according to the running workload and the optimization goal

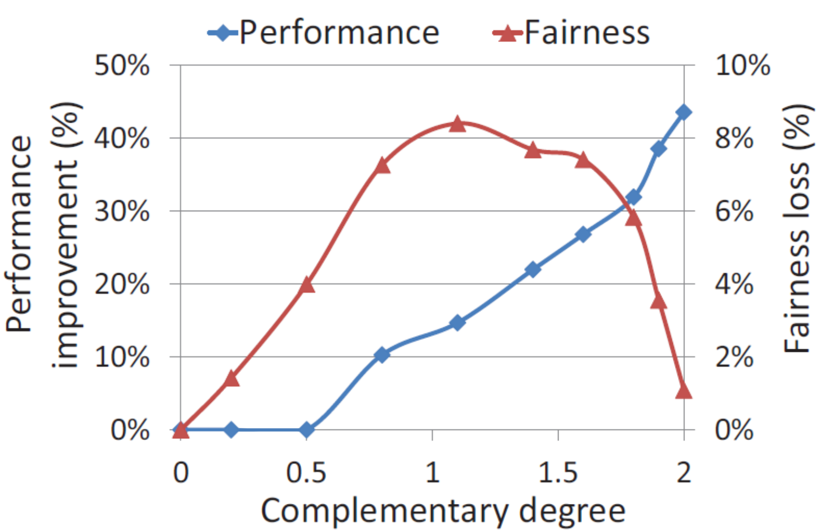
New Notion: Complementary Degree

- Applications have heterogeneous resource demand
  - Heterogeneity (complementarity) makes opportunities for bi-criteria optimization between performance and fairness
- Complementary degree
  - Quantify the complementarity for resource demands of all applications
  - Entropy-based approach
    - Entropy is used in information theory to characterize the randomness of information content
    - Treat resource demands as the information (randomness of the information → heterogeneity of the workload)

$$complementary\ degree = -\sum_{i \in R} P(i) \log_2 P(i),$$
$$R{:}\ all\ resource\ types,$$
$$P(i){:}\ the\ percentage\ of\ jobs\ whose\ dominant\ resource\ type\ is\ i.$$

Workload Characterization Model

- Build a model for the given scheduling policy with regression approach
  - input: the complementary degree of the workload
  - output: the performance improvement and the fairness loss

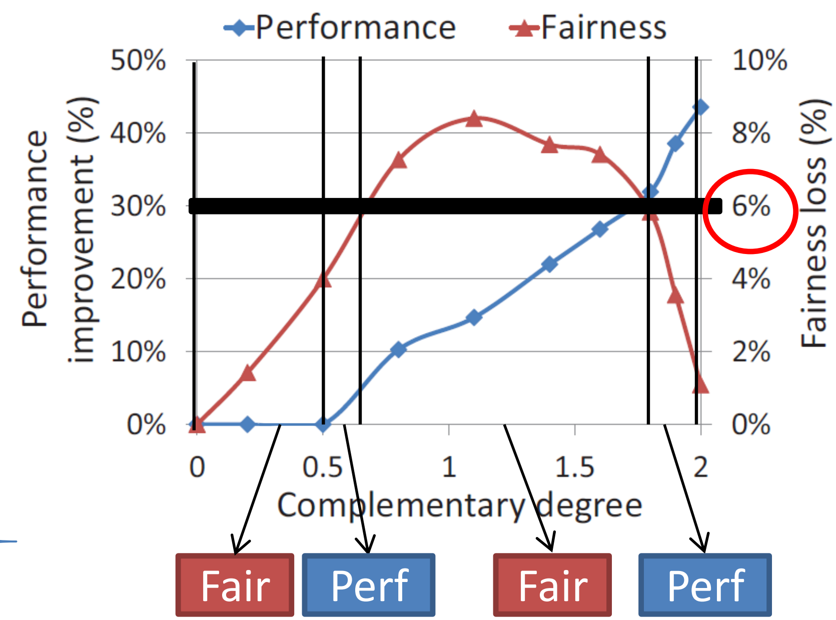


Adaptive Scheduling

- Scheduling policies in Gemini
  - Fairness-oriented policy (DRF)
  - Performance-oriented policy (enhance capacity scheduler with task-packing heuristic)
- Workload-aware scheduler

Goal

Optimize the performance when the fairness loss <= 6%



Fair fairness-oriented policy

Perf performance-oriented policy

Adaptive Scheduling Algorithm

- Decide the scheduling policy adaptively
  - Detect the change of the workload;
  - Calculate the complementary degree of the current running workload;
  - Predict the performance improvement and fairness loss with the model of performance-oriented policy
    - If performance improvement > 0 and fairness loss <= user-defined value
      - Apply performance-oriented policy
    - Else
      - Apply the fairness-oriented policy

Outline

- Introduction
- Our Proposal: Gemini
- **Experiments**
- Summary

Testbed Setup

- Cluster
  - 10 node (each with 12 CPU cores, 24GB memory and 500GB disk)
  - Connected with 10Gb/sec Ethernet
- Workload
  - Synthesized workload (100jobs) based on the trace provided by Facebook

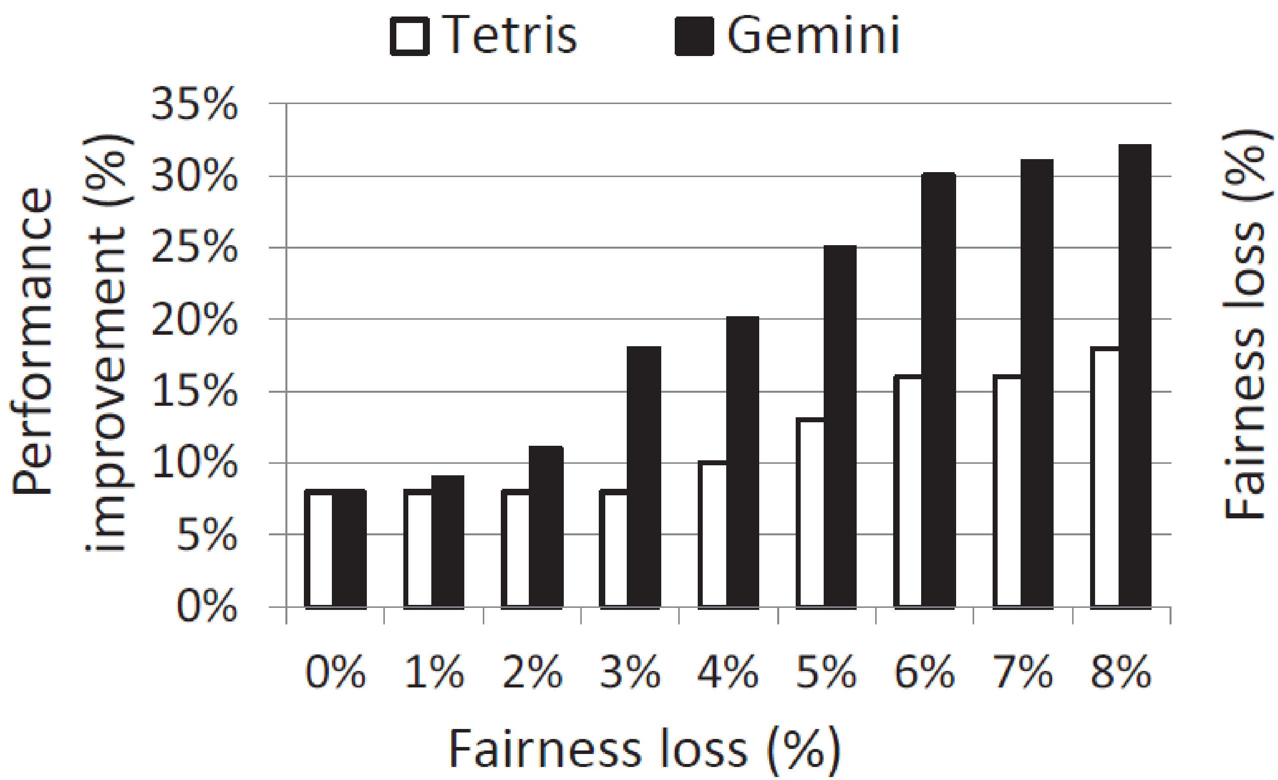
Bin	Job Type	Map Tasks		Reduce Tasks		# Jobs
		#	Demand	#	Demand	
1	rankings selection	1	<1,1 GB>	NA	NA	38
2	grep search	2	<1, 1.5 GB>	NA	NA	18
3	uservisits aggregation	10	<2, 0.5 GB>	2	<4,2 GB>	14
4	rankings selection	50	<4, 1 GB>	NA	NA	10
5	uservisits aggregation	100	<2, 1.5 GB>	10	<2, 2 GB>	6
6	rankings selection	200	<3, 2 GB>	NA	NA	6
7	grep search	400	<2, 1 GB>	NA	NA	4
8	rankings-uservisits join	400	<1, 2 GB>	30	<2, 0.5 GB>	2
9	grep search	800	<2, 0.5 GB>	60	<1, 3 GB>	2

- Metrics
  - Performance: percentage reduction on the makespan
  - Fairness: average reduction of job completion times

Trace-driven Setup

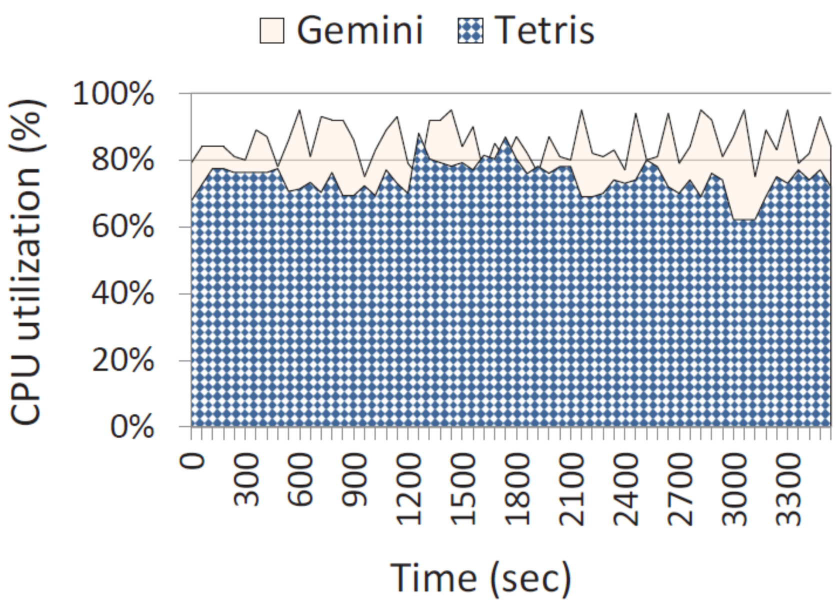
- Google trace
  - Over 900 users
  - 12.5k machines
  - One month
  - Task submission times, execution time and normalized CPU/Memory/Disk resource demands
- Simulation acceleration
  - 600 nodes
  - 60 users (equally share)
  - 24 hours

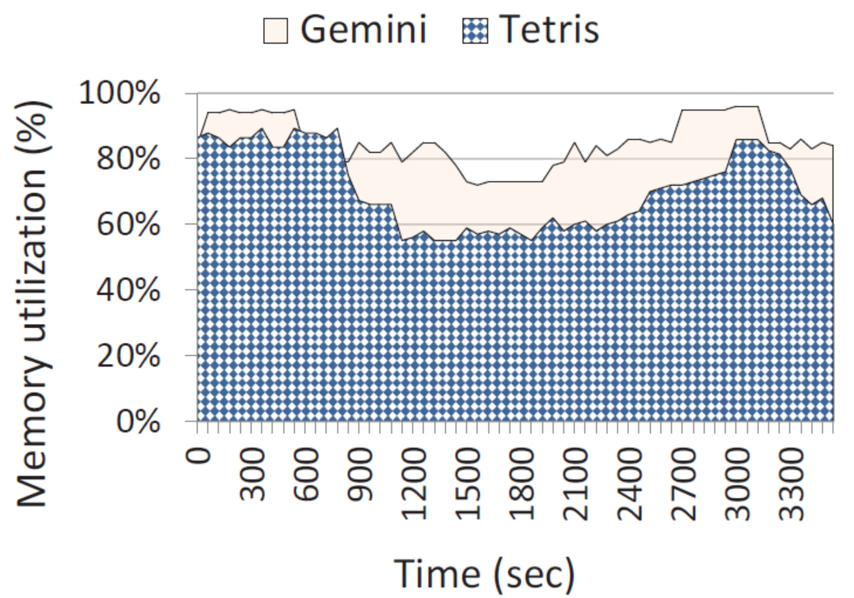
Testbed Experimental Result



Achieve better performance under the same fairness loss!

Resource utilization





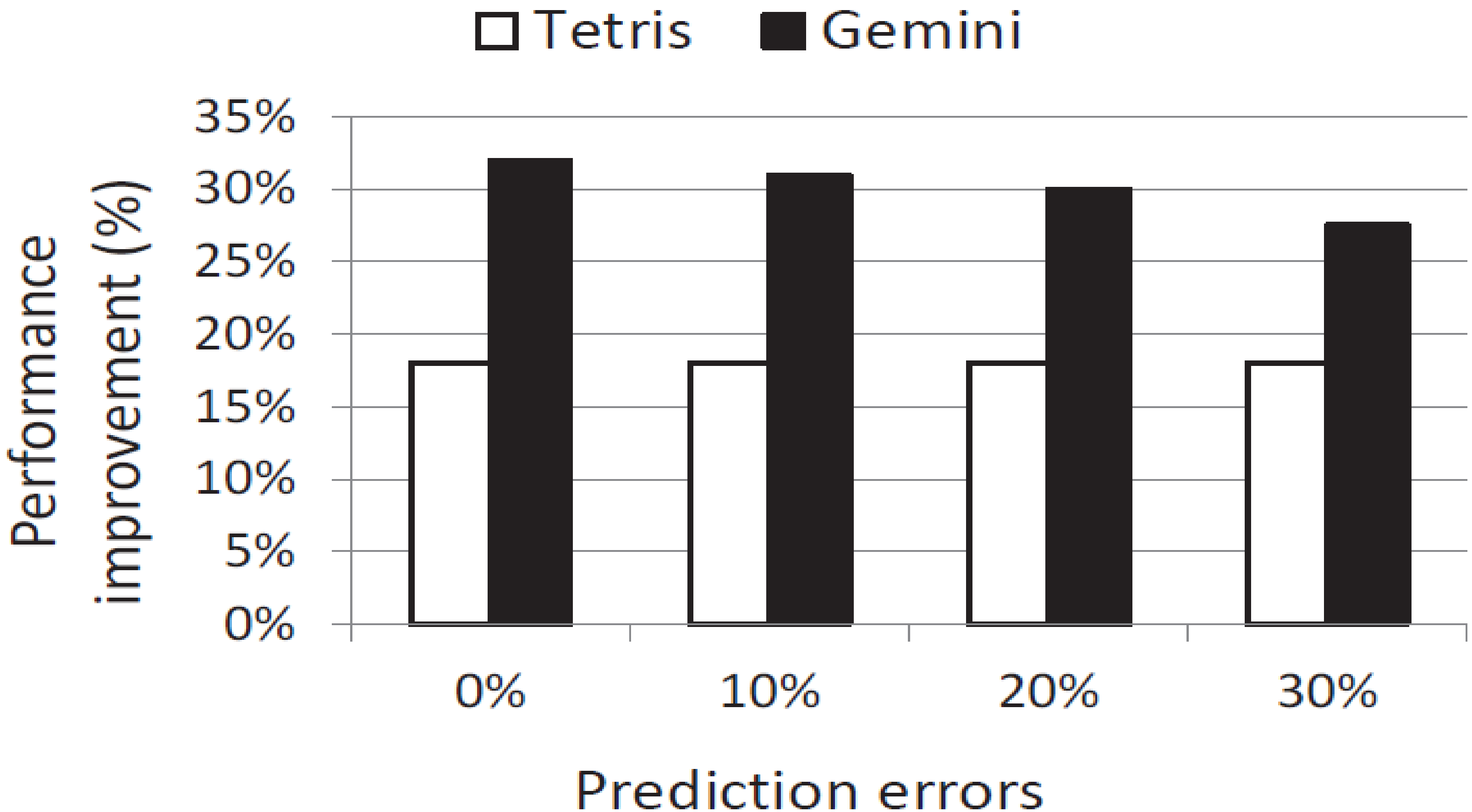
Given one fixed fairness loss, the reason Gemini achieves better performance is that it can utilize the resources more efficiently.

Overhead Analysis

	Hadoop Fair Scheduler 10K (50K) tasks	Tetris 10K (50K) tasks	Gemini 10K (50K) tasks
Scheduling overhead	.05ms (.18ms)	.078ms (.19ms)	.08ms (.19ms)

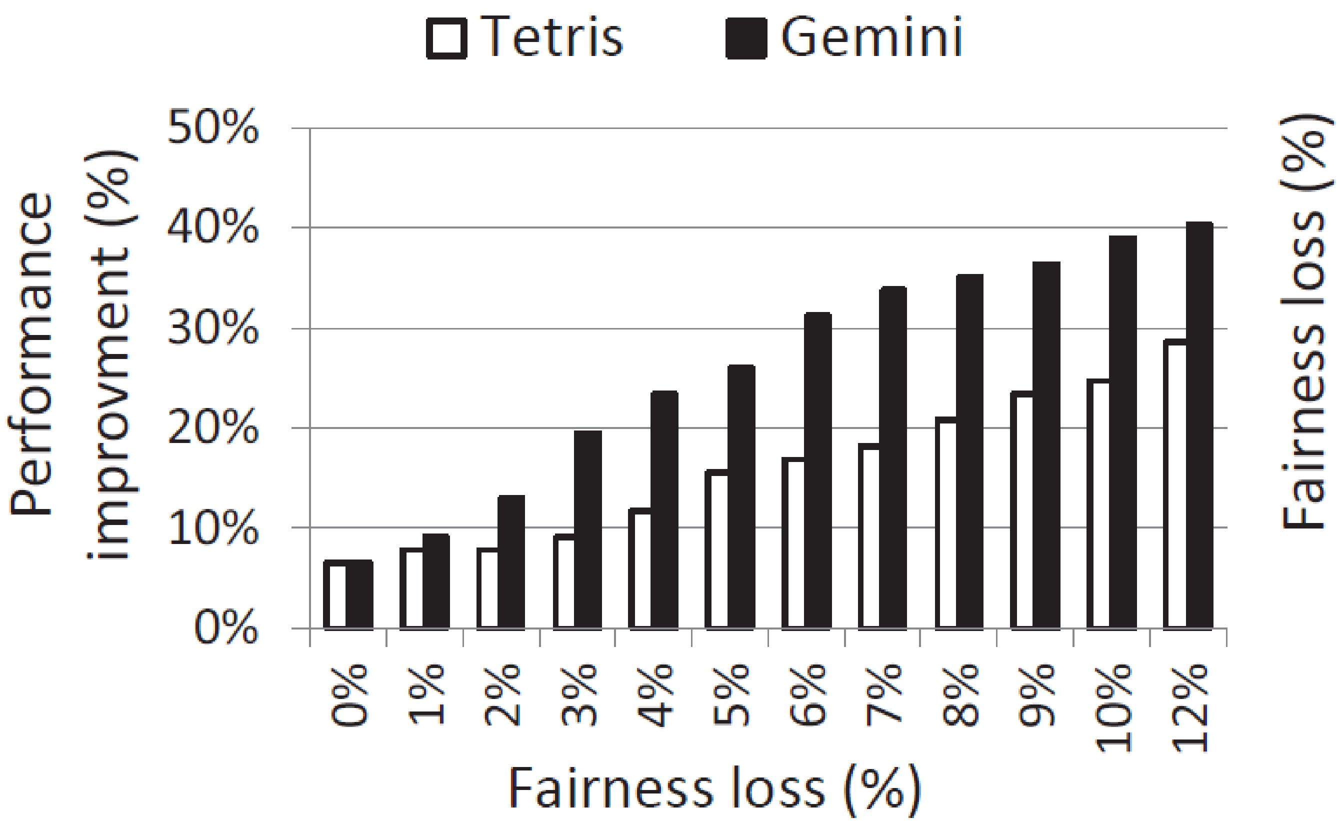
Our online algorithm design has little runtime overhead.

Sensitivity to Prediction Inaccuracy



The result demonstrates that Gemini is robust to the prediction errors.

Large-scale Simulation Result



Gemini can still achieve significant performance improvement in large-scale cluster.

Conclusion

- There is a tradeoff between the performance and fairness.
- We propose an workload-aware scheduler which can adaptively decide the most proper scheduling policy at runtime.
- The experiment on real clusters and simulations shows that our system achieves better performance as well as fairness than the state-of-the-art work.

Acknowledgement

This work is supported by a MoE AcRF Tier 1 grant (MOE 2014-T1-001-145) in Singapore.

Zhaojie's work is in part supported by the National Research Foundation, Prime Ministers Office, Singapore under its IDM Futures Funding Initiative and administered by the Interactive and Digital Media Programme Office.

Thank you and Q&A