Ensuring Quality of Service in High Performance Servers

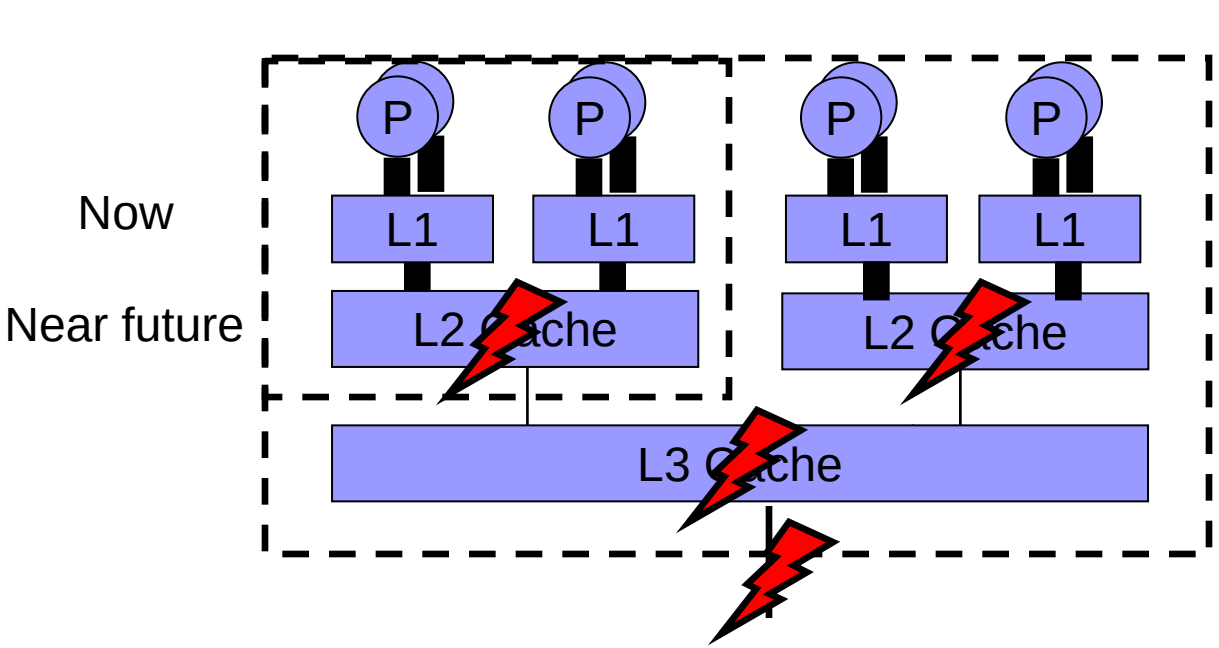
YAN SOLIHIN
Fei Guo, Seongbeom Kim, Fang Liu

Center of Efficient, Secure, and Reliable Computing (CESR)
North Carolina State University
solihin@ece.ncsu.edu
http://www.ece.ncsu.edu/arpers

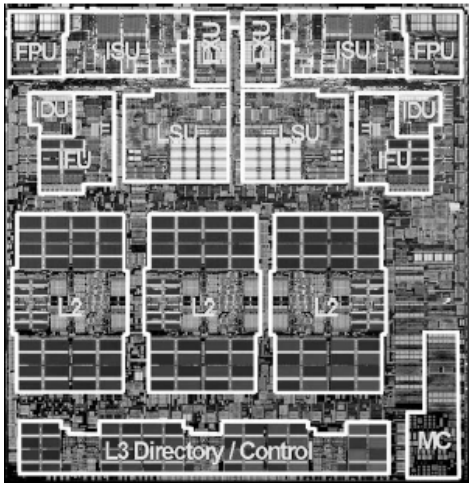
Motivation

- Two trends
  - Utility/On-demand computing
    - A server shared by programs from many clients
    - Programs need “Quality of Service” (QoS) guarantee
  - Multicore chips (Chip Multi-Processor/CMP)
    - Building blocks for utility computing servers
    - Do not naturally support QoS
    - Due to shared platform resources (caches, bandwidth)
- Goal
  - Build software and architecture support for providing QoS

Shared Cache in CMP Server

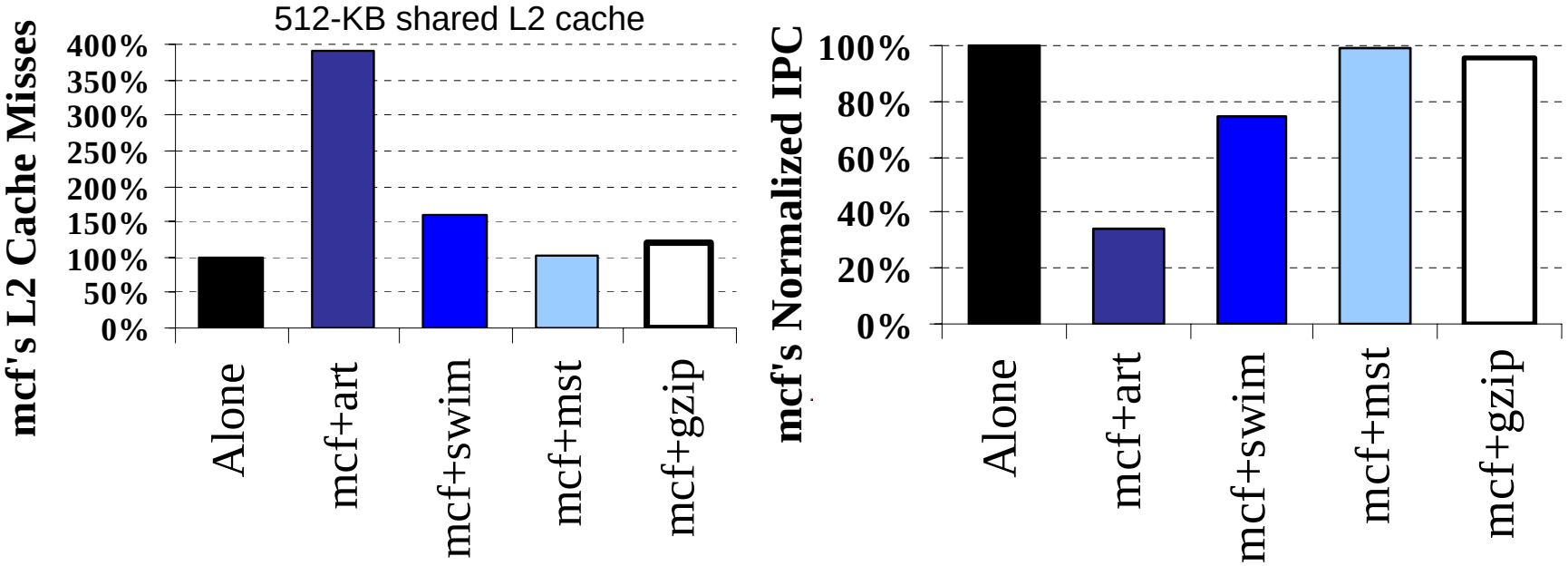


IBM Power5



- Shared resources present new performance and fairness challenges

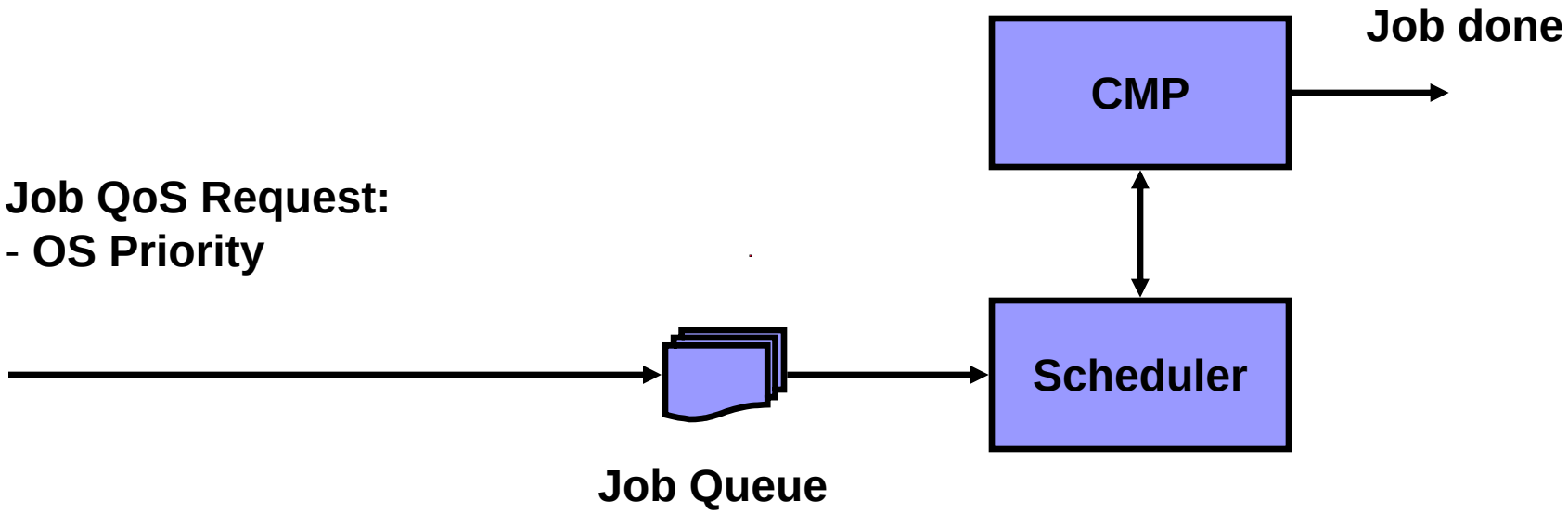
Impact of Cache Space Contention



- Application-specific and Coschedule-specific
- Significant: Up to 4X cache misses, 65% IPC reduction
- Other benchmarks affected, too

CMP-based server needs to provide Quality of Service

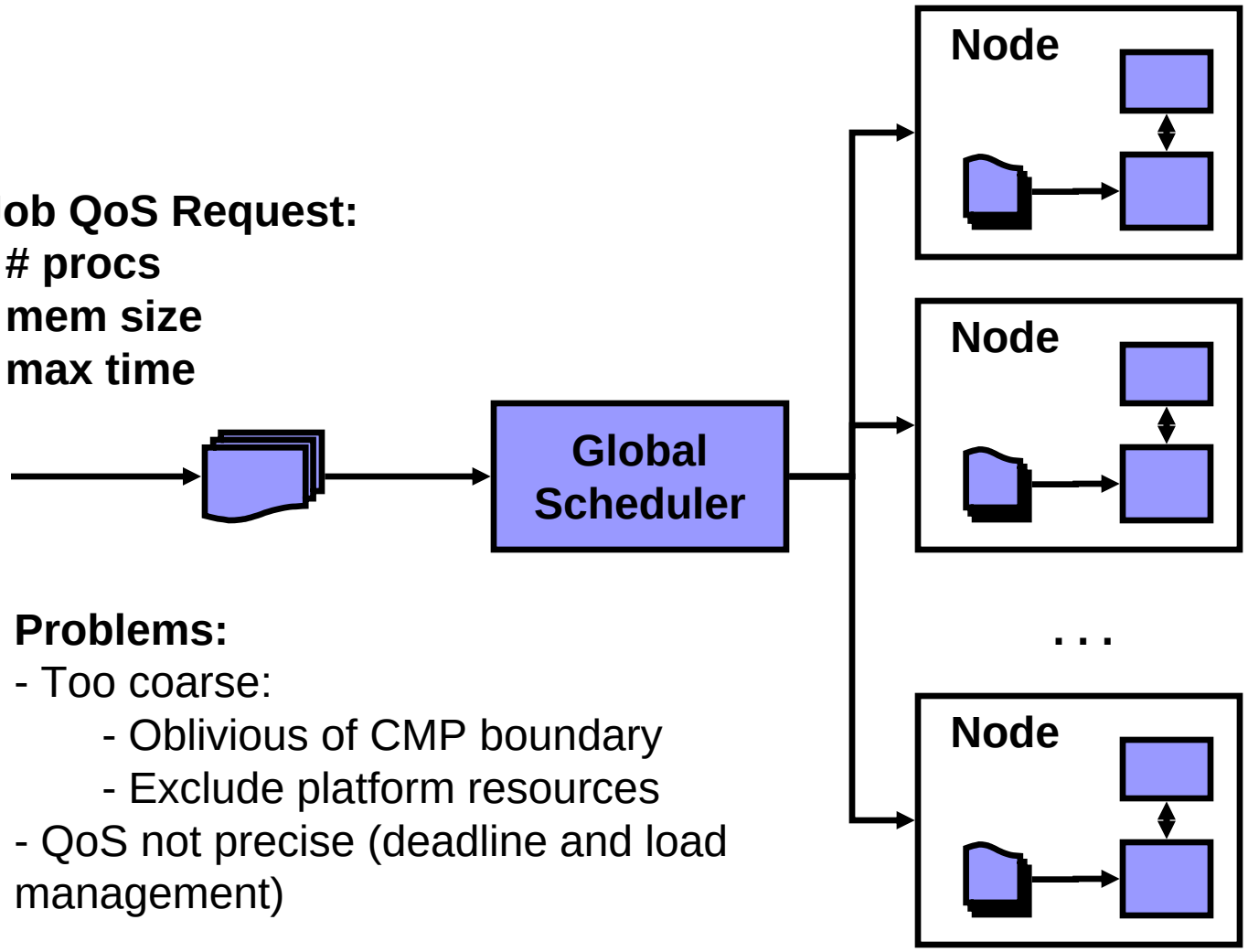
Current Multicore Chip (CMP) Node



Problems:
- OS Priority does not provide “guarantee”
- OS Priority does not control platform resources
- The same priority yields different performance under different load

Current Server

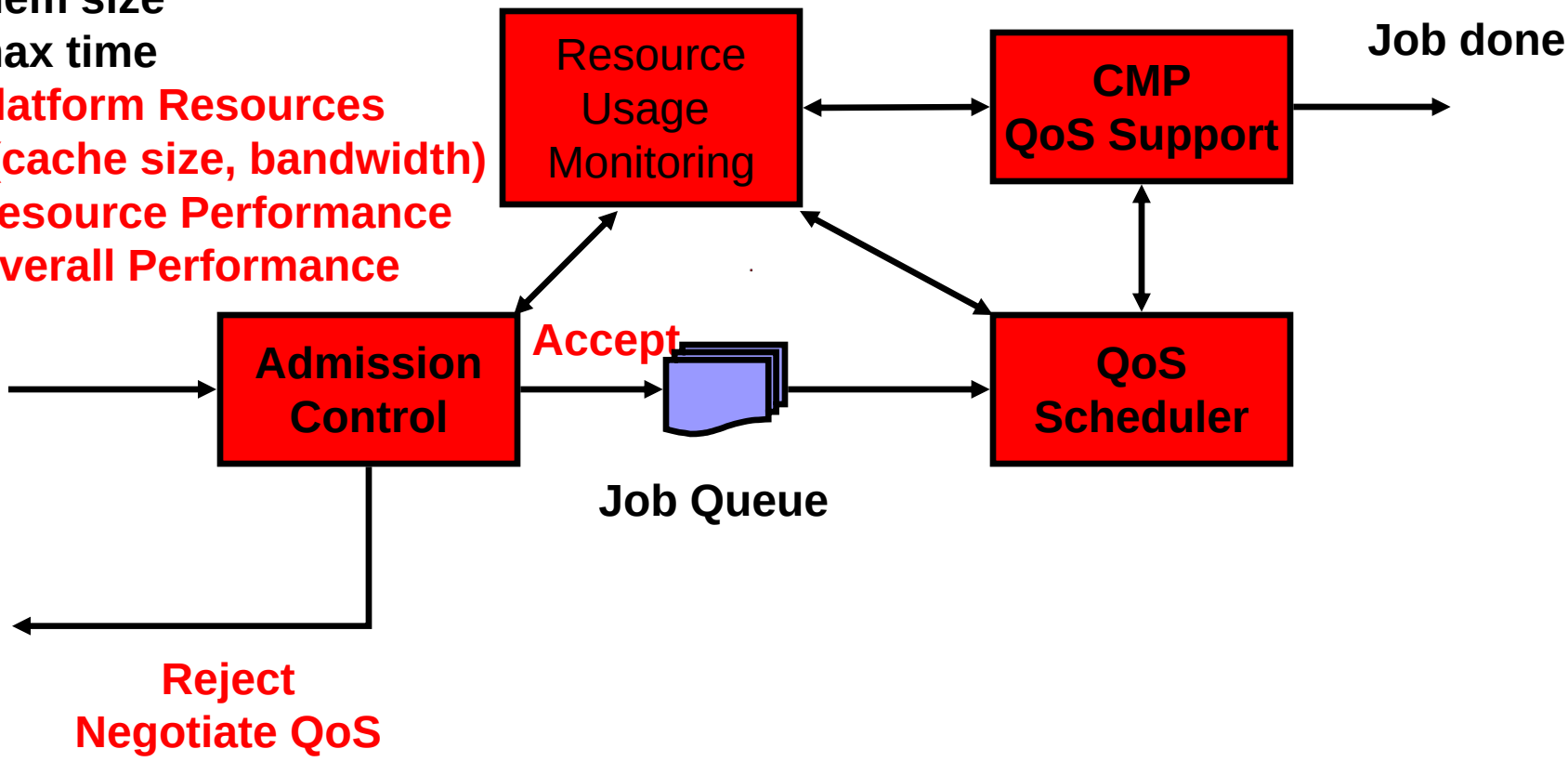
Job QoS Request:
- **# procs**
- **mem size**
- **max time**



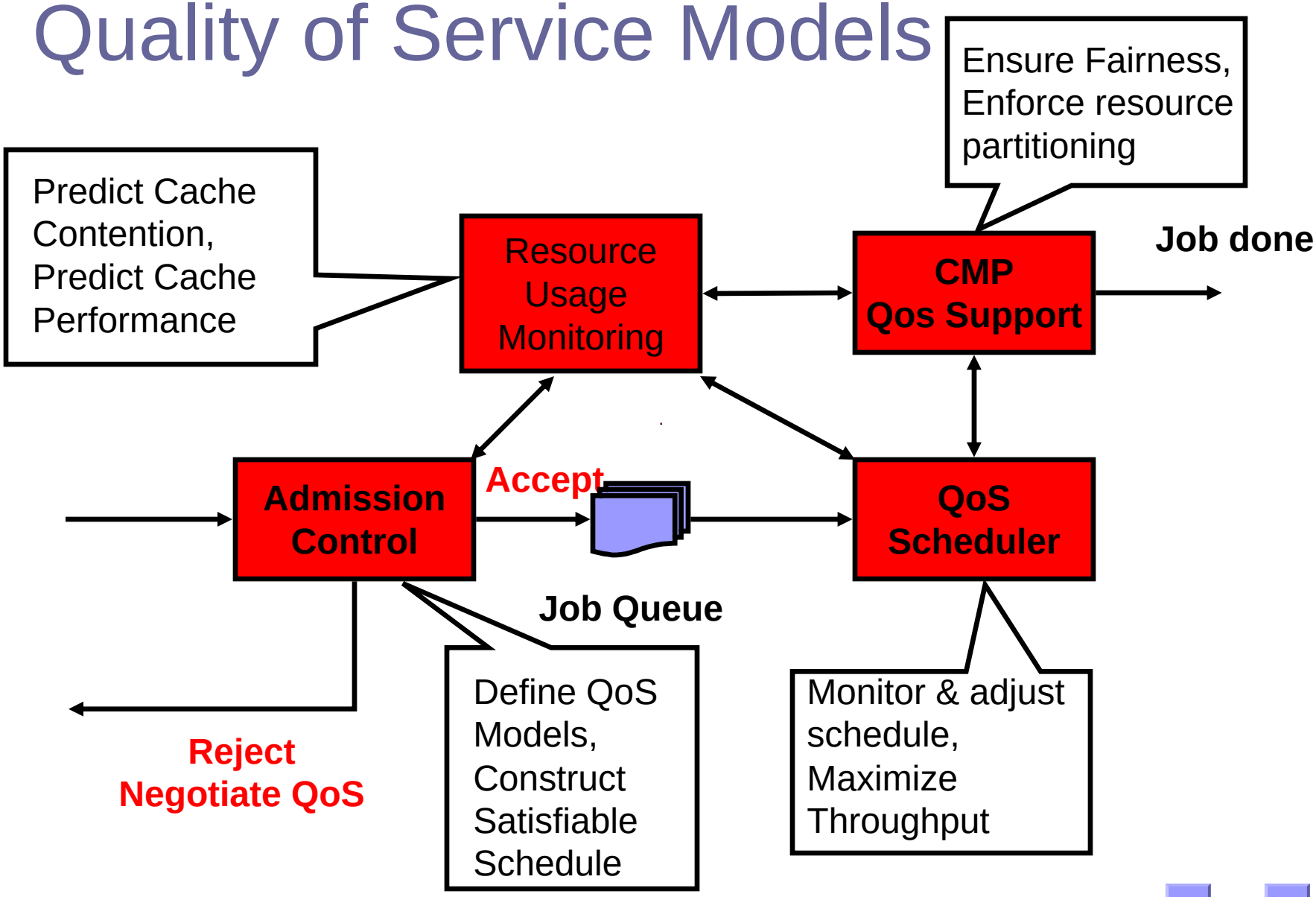
Problems:
- Too coarse:
  - Oblivious of CMP boundary
  - Exclude platform resources
- QoS not precise (deadline and load management)

Quality of Service Models

Job QoS Request:
- **# procs**
- **mem size**
- **max time**
- **Platform Resources (cache size, bandwidth)**
- **Resource Performance**
- **Overall Performance**



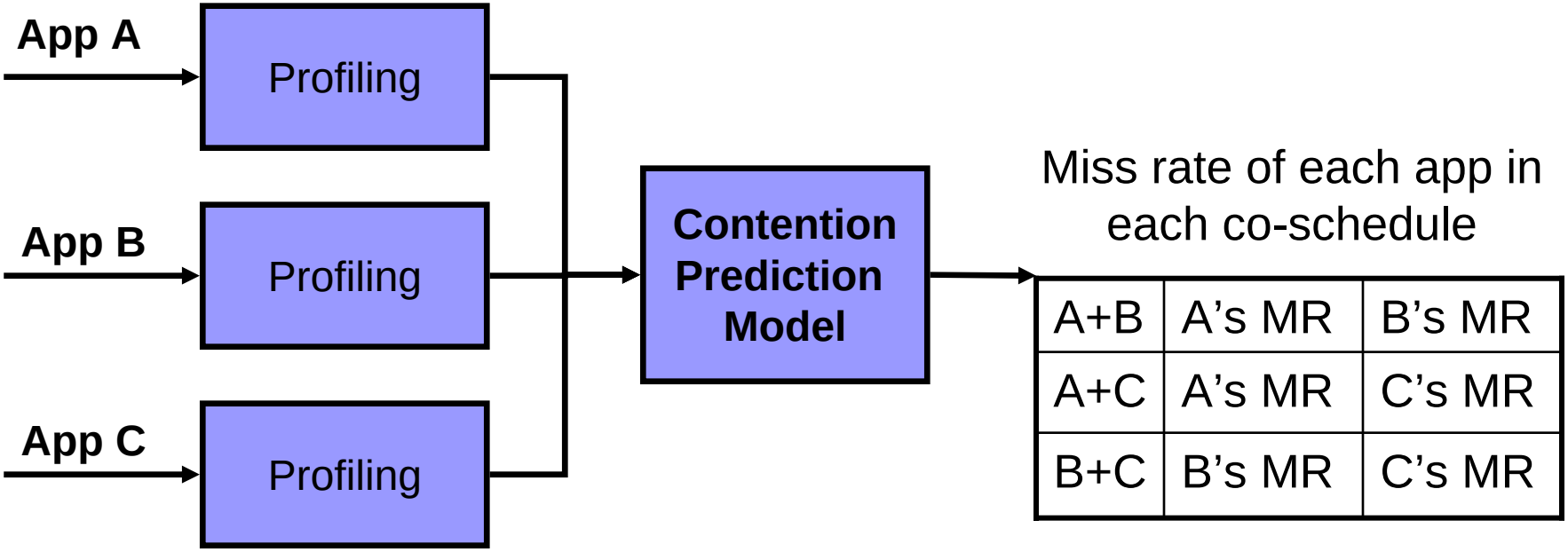
Quality of Service Models



QoS Models

- Multiple Models
  - Resource Usage Model (RUM)
    - Cache size, bus bandwidth, etc.
  - Resource Performance Model (RPM)
    - e.g. miss rate
  - Overall Performance Model (OPM)
    - e.g. exec time, TPS
- Multiple Service Tiers, e.g. for RUM:
  - Gold: 16GB Mem, 4MB Cache, 6GB/sec bandwidth
  - Silver: 4GB Mem, 2MB Cache, 3GB/sec bandwidth
  - Standard: 2GB Mem, 1MB Cache, 2GB/sec bandwidth

Cache Performance Prediction model

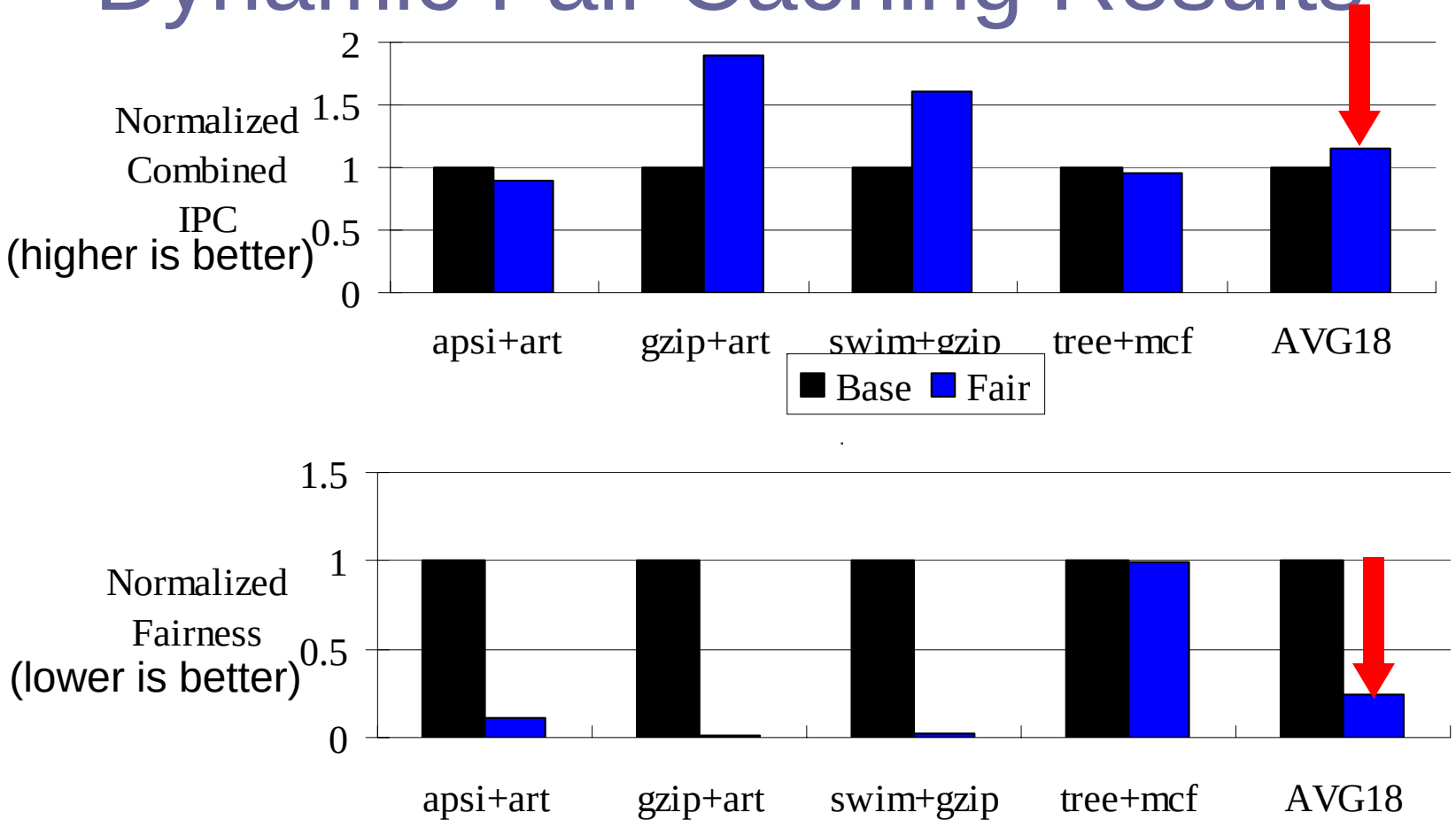


- Validated against detailed CMP simulation
- Average error of 3.9%
- Correctly identifies all hyper contention cases

QoS Support in CMP

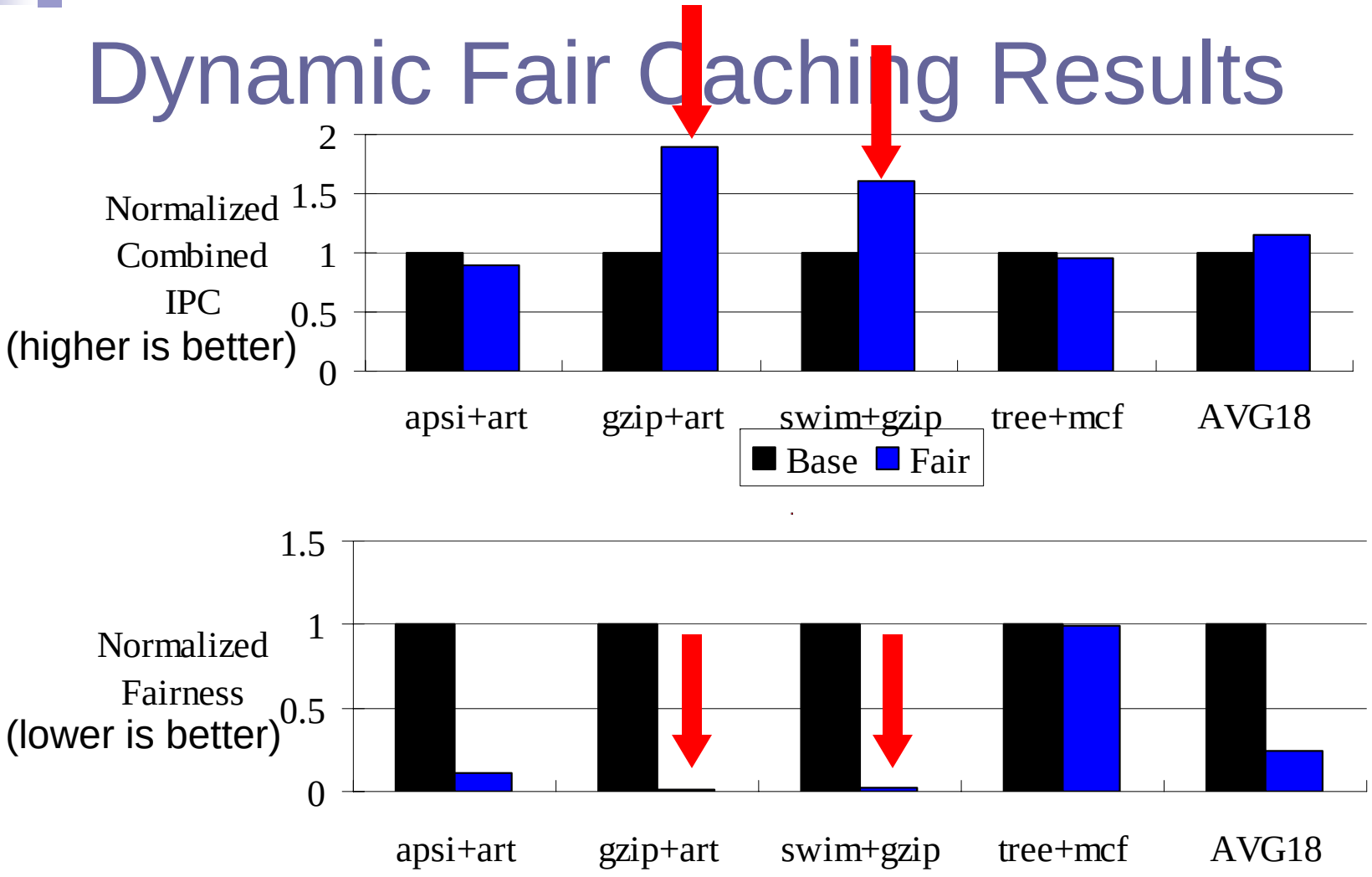
- Enforce Resource Partitioning Policies
  - Cache and Bandwidth Partitioning
- Enforce Fairness in Resource Usage
  - How is cache fairness measured? Cache fairness metrics
  - How is fairness improved?
    - Static and dynamic fair caching policies
  - What benefits fairness gives?
    - Improved fairness, but also improved throughput!

Dynamic Fair Caching Results



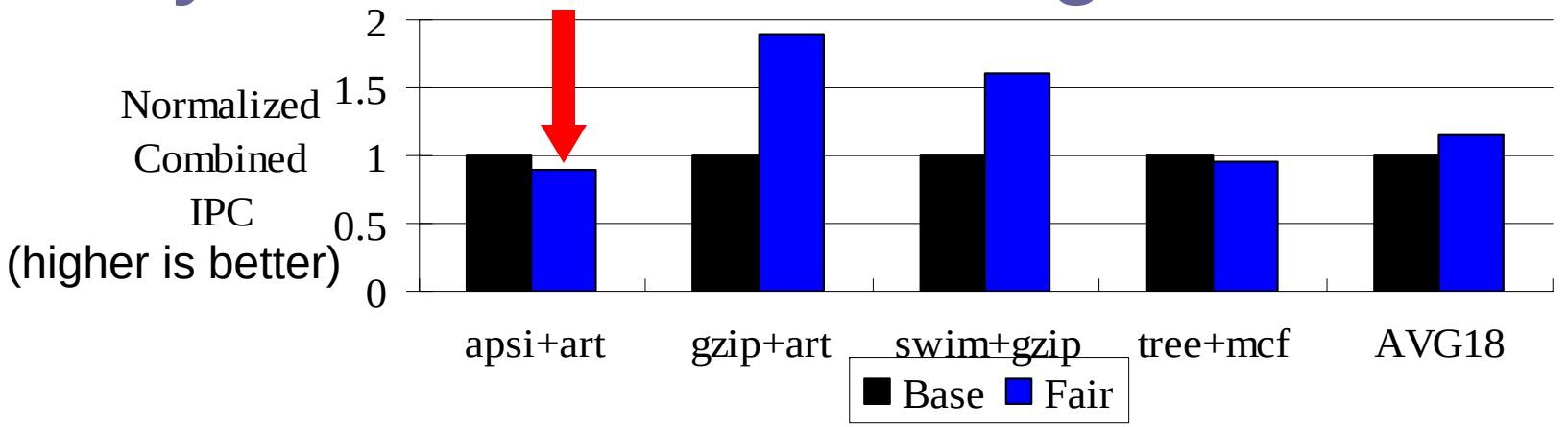
Fair improves both fairness and throughput

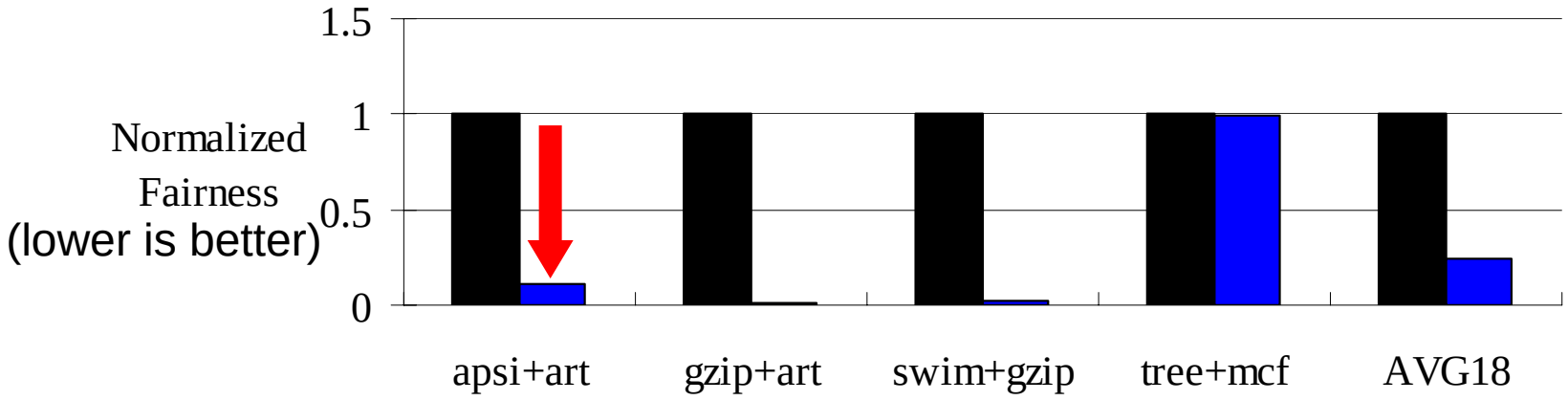
Dynamic Fair Caching Results



Better fairness ⇒ better throughput (16/18 cases)

Dynamic Fair Caching Results



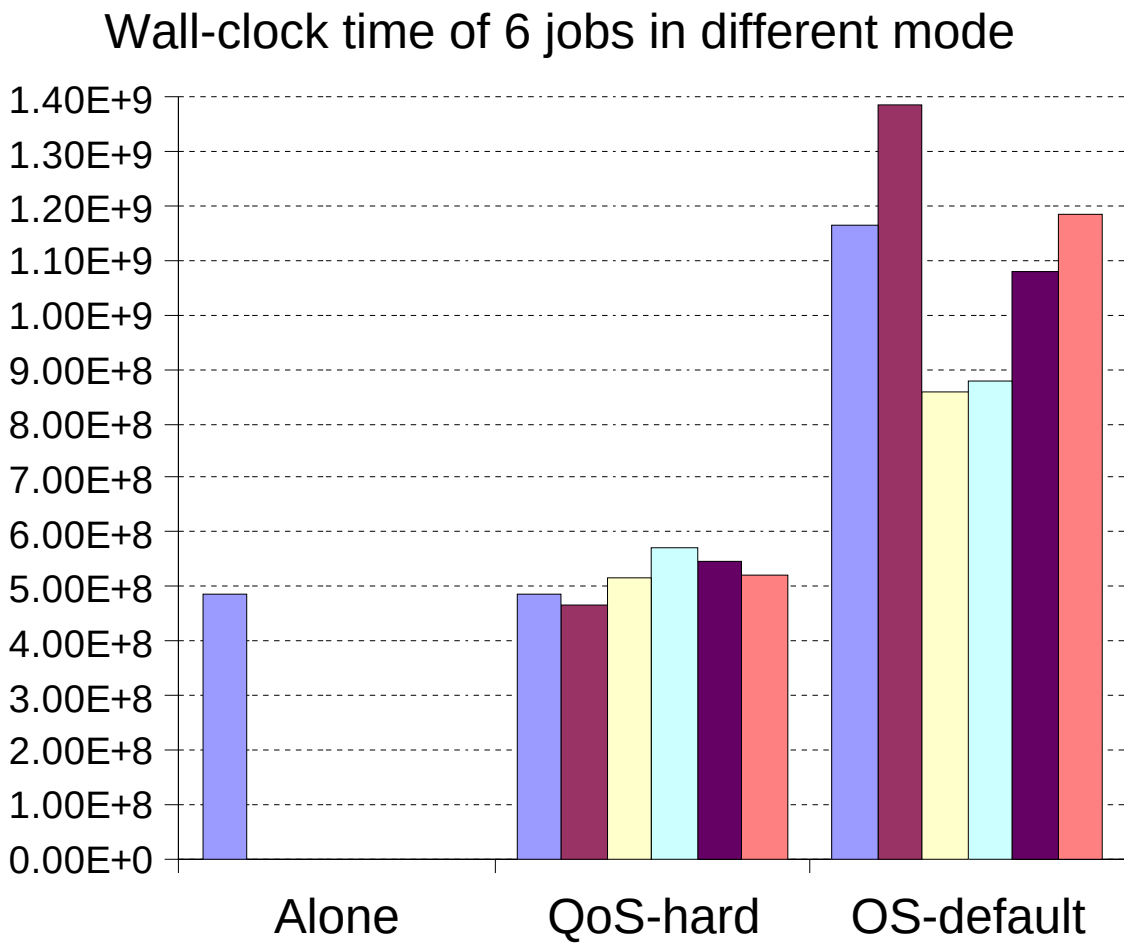


Better fairness $\not\Rightarrow$ better throughput (2/18 cases)

State of the Project

- Completed
  - Performance Monitoring and Modeling
    - Cache Space Contention in CMP **[*HPCA 2005*]**
    - Replacement Policy Performance Model **[*Sigmetrics 2006*]**
    - Performance impact of OS activities **[*ISPASS 2007*]**
  - QoS Models and Architecture
    - Fair Caching, Cache partitioning **[*PACT 2004*]**
    - QoS Models' Impact on Performance **[*Sigmetrics 2007*]**
  - Prototype of OS with QoS Support **[*dasCMP 2007*]**
- Ongoing
  - Job Admission Control and Scheduler
- Broader Impact
  - HPCA 2007 Tutorial
  - Software: Analytical Cache Performance Prediction (ACAPP) Toolset
  - Industry Collaboration with Intel and RedHat

System Integration of QoS Components



Acknowledgement

- Collaborators:
  - Ravi Iyer, Intel
  - Li Zhao, Intel
  - Will Cohen, RedHat
- Students:
  - Seongbeom Kim, PhD (VMWare)
  - Fei Guo, PhD
  - Fang Liu, PhD
  - Radha Venkatagiri, MST
- Other Research @ ARPERS (Architecture Research for PErformance, Reliability, and Security)
  - Helper Threads
  - Secure Architecture and Operating System
  - Support for Software Reliability

Q & A

- Thank you