Exploiting Hardware Heterogeneity for Interactive Services

Yuxiong He[2]

Joint work with Shaolei Ren[1],
Sameh Elnikety[2], Kathryn S McKinley[2]

[1]Florida International University
[2]Microsoft Research

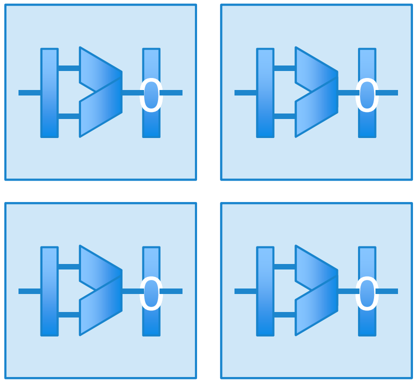
Interactive Services

- Applications
  - Web search, web server, finance server

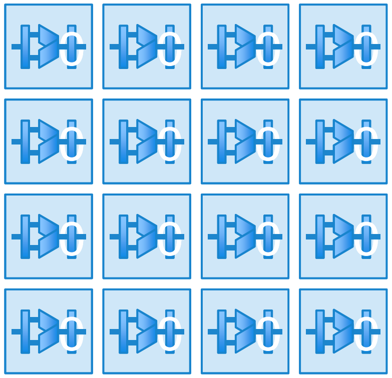
- Requirements
  - High quality, fast response
  - High throughput, low cost

Hardware for Interactive Services in Today’s Data Center

- Homogeneous servers



Few fast high-performance cores



Many slow energy-efficient cores

Variance of Job Service Demand

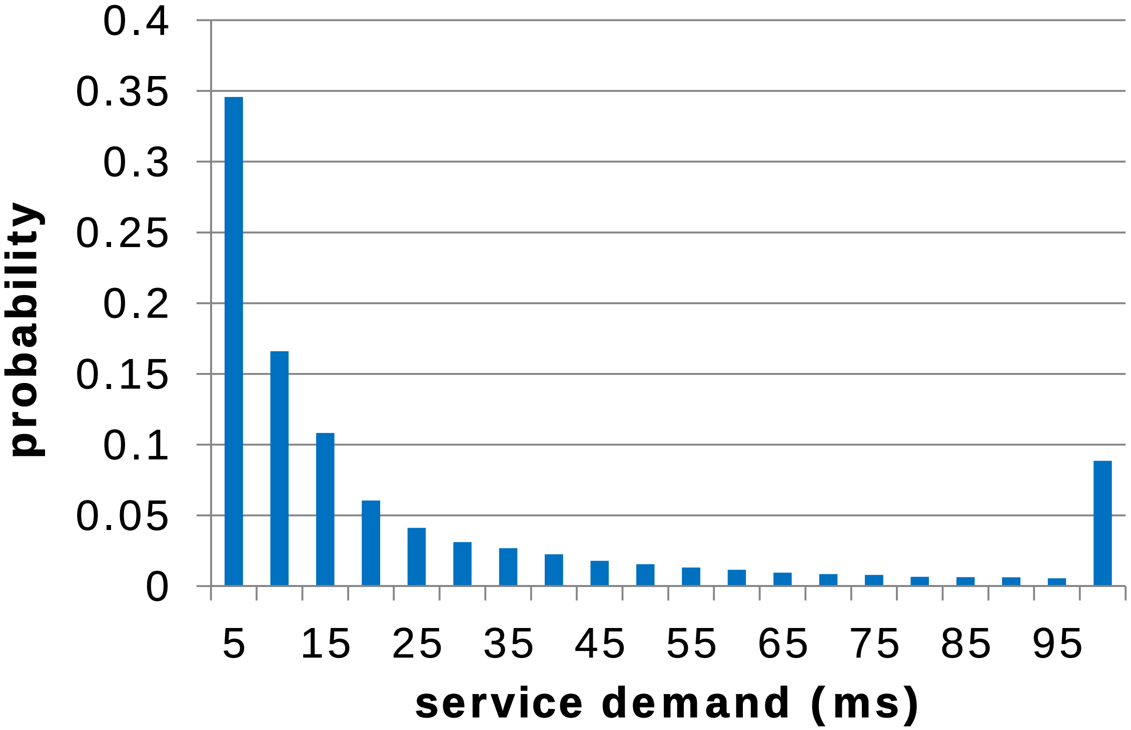


Figure. Measured Bing search service demand distribution

Homogeneous server with slow cores: cannot satisfy QoS of long requests

Homogeneous server with fast cores: meet QoS but energy consuming and lower throughput

Opportunity of Heterogeneity

Heterogeneous server: combine fast and slow cores

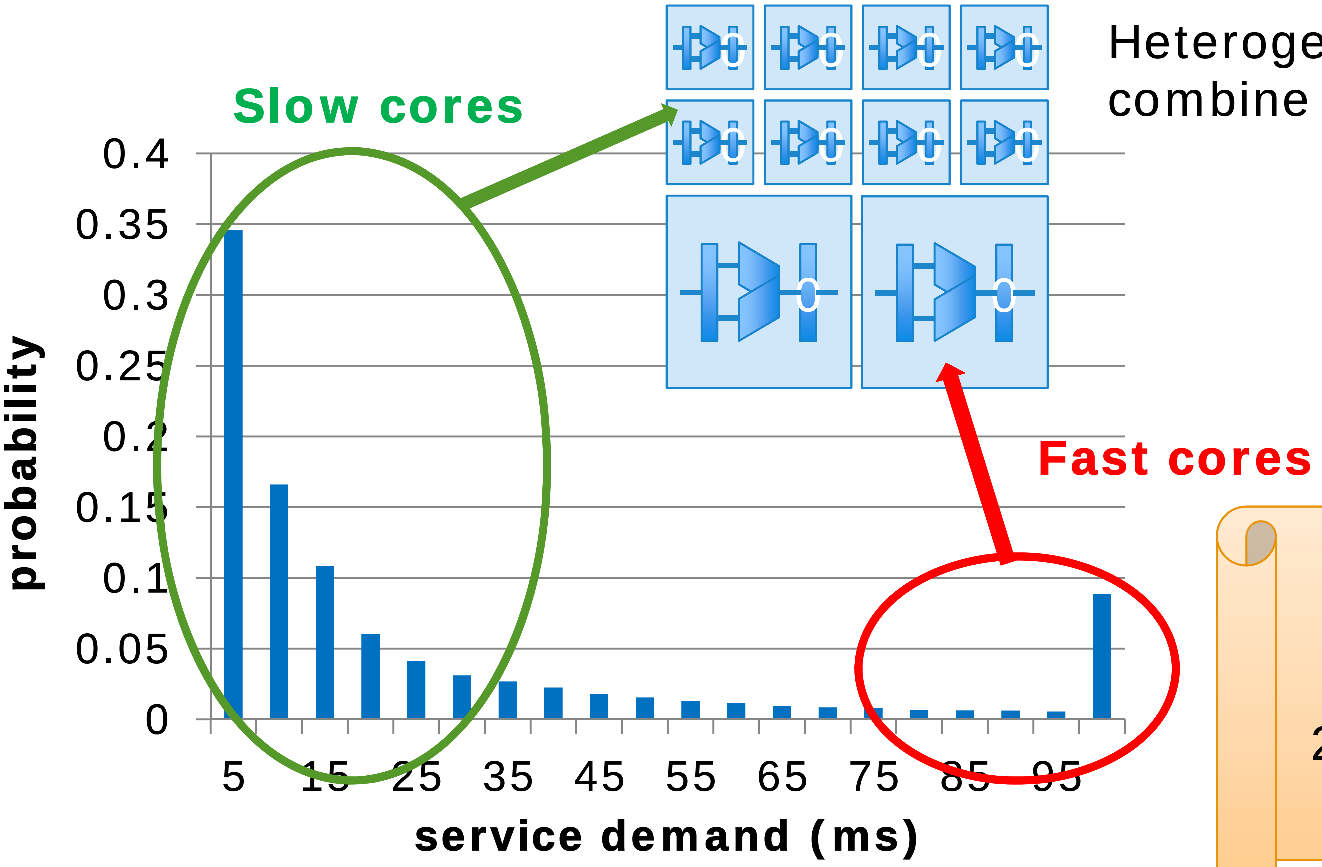


Figure. Measured Bing search service demand distribution

Challenges:

1. Service demand is unknown.
2. Jobs compete for cores.

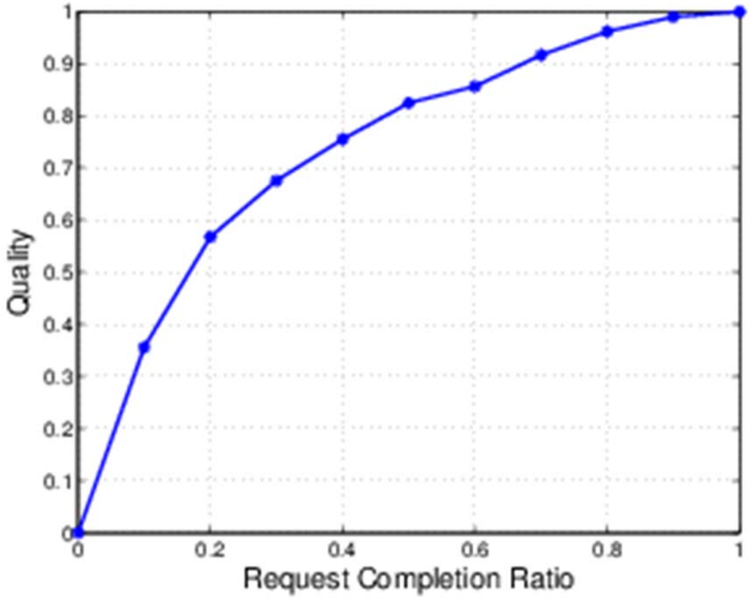
Contributions

- FOF scheduler for heterogeneous servers
- Bing search server simulation
  - Double throughput while meeting QoS

- FOF for servers with SMT (Simultaneous Multithreading)
- Finance server implementation
  - 16% higher throughput than default OS scheduler

Scheduling Model

- Inputs
  - Queue of jobs
  - Job service demand unknown
  - Job deadline
  - Partial results



Measured Bing search quality profile

Scheduling Model

- Inputs
  - Queue of jobs
  - Job service demand unknown
  - Job deadline
  - Partial results
- Outputs
  - Assign jobs to fast/slow cores
  - Decide processing time of jobs
- Objective
  - Maximize total quality of all jobs

Challenge I.
Unknown Service Demand

- How can we assign long jobs to fast cores and short jobs to slow cores?

- Key insight: Slow to Fast
  - Migrate a job from slower to faster cores
  - Short jobs complete on slow cores
  - Leave fast cores for long jobs

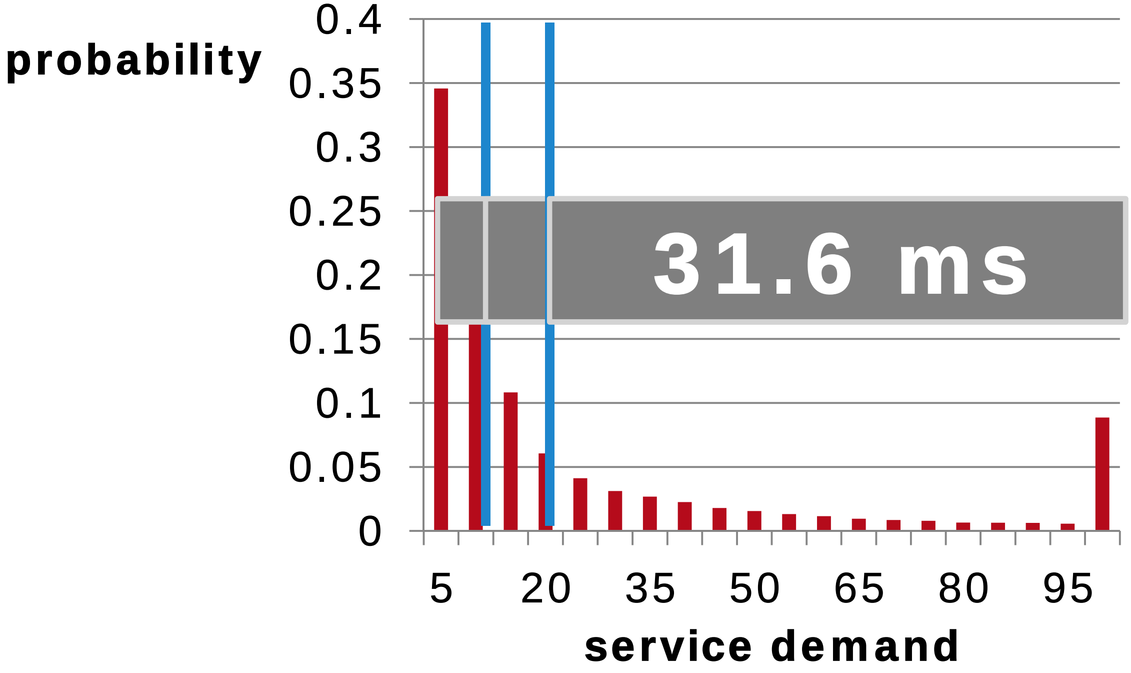
Challenge II.
Jobs Compete for Cores

- Which jobs should be processed by fast cores?

- Key insight: Fast Old
  - Assign fast cores to old jobs.

“Fast Old” insight

- Older job has closer deadline.
- Older job has more work left.
- “Fast old” improves response quality



FOF Scheduler: Fast Old & First

1. **Fast first: always use the fastest available core**
2. **Fast old: promote old jobs slow to fast**

Slow
Medium
Fast

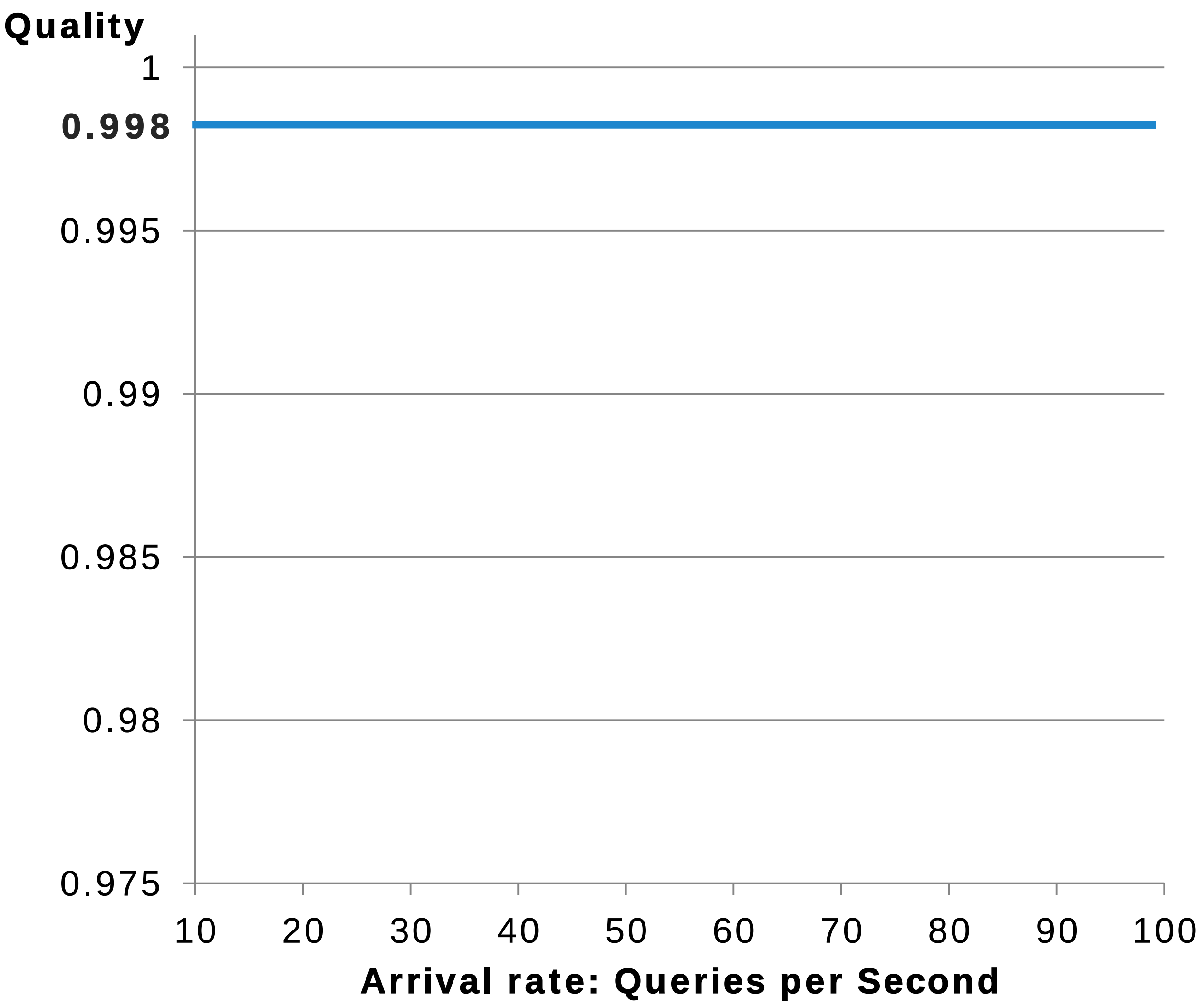
Evaluation

- Simulation modeling Bing search workload
- Hardware:

  4 servers configurations with same design time power budget

  A: 2 Big cores (Sandy Bridge)
  B: 10 Medium cores (Nehalem)
  C: 24 Small cores (AtomD)
  D: 1 B + 4 M + 2 S

Homogeneous Fast vs Slow Cores



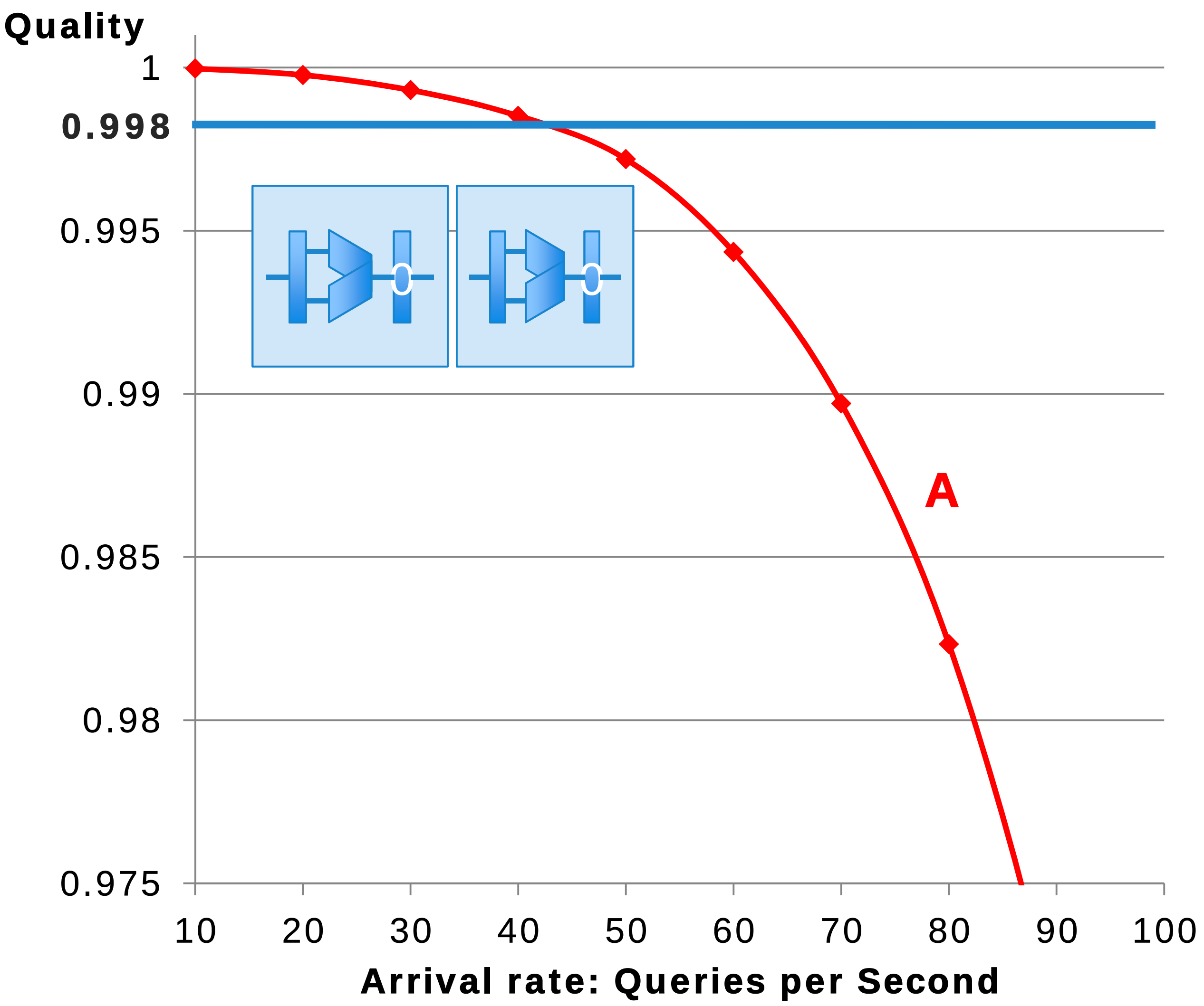
Homogeneous

A. 2 Fast

B. 10 Medium

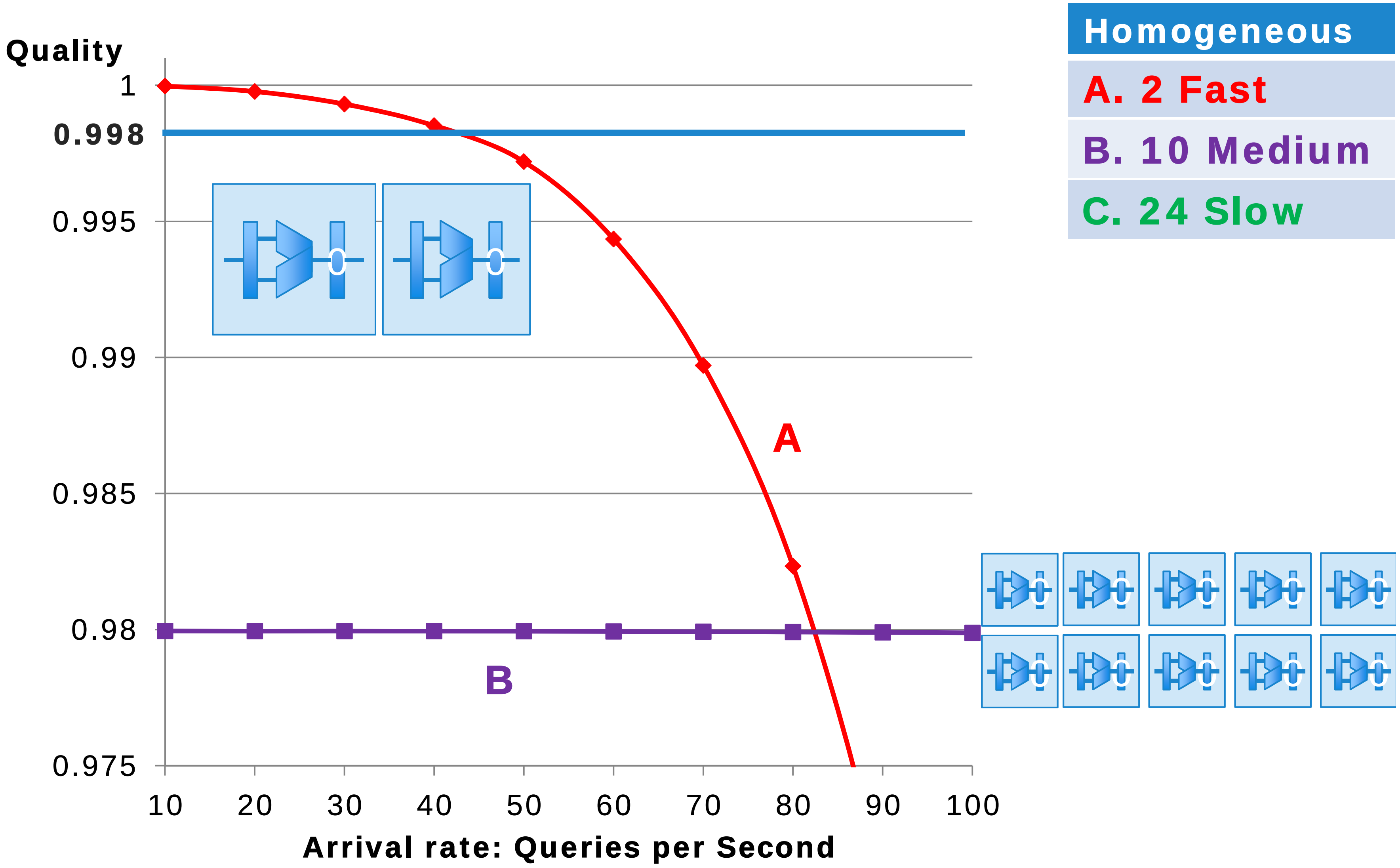
C. 24 Slow

Homogeneous Fast vs Slow Cores

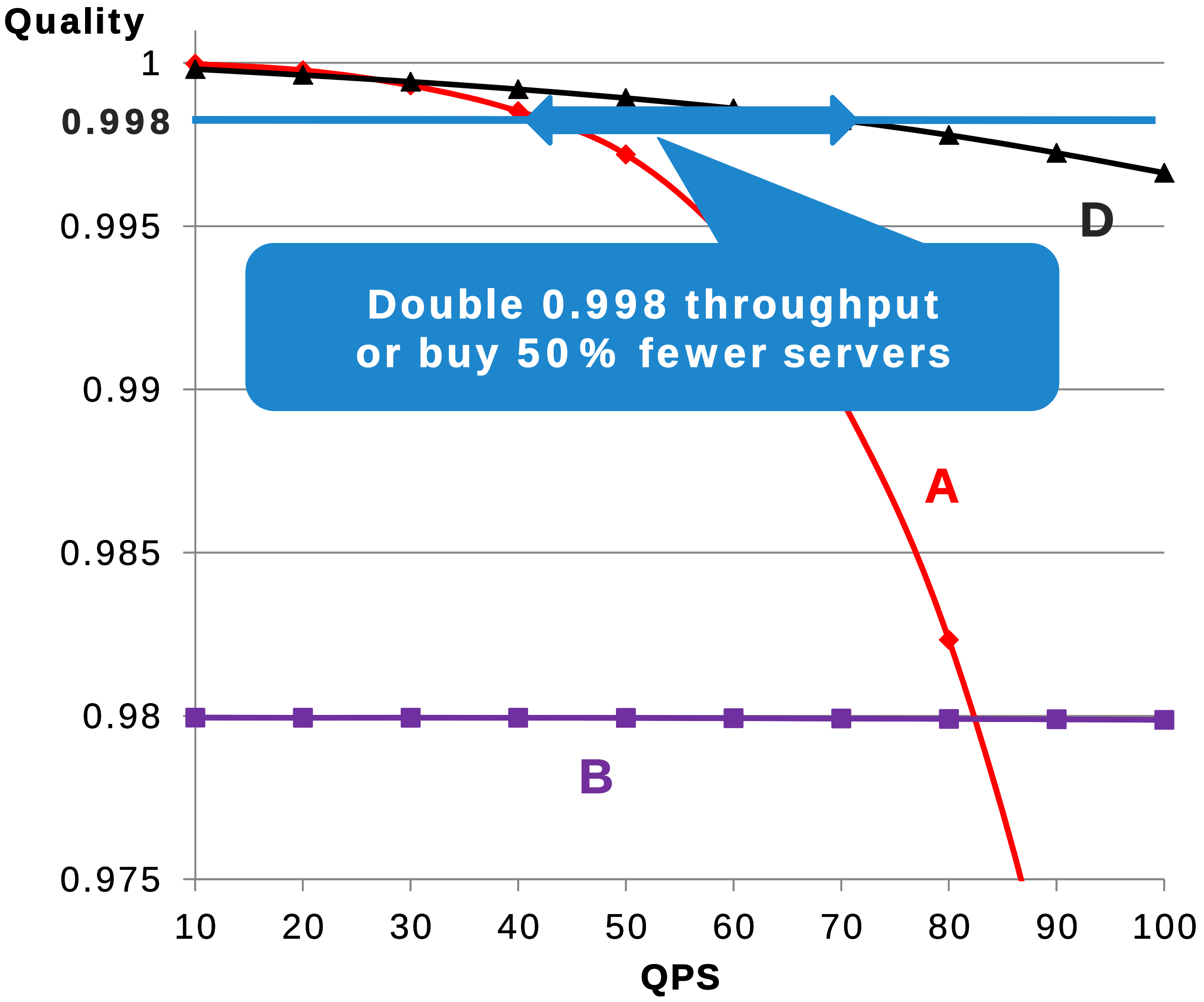


Homogeneous
A. 2 Fast
B. 10 Medium
C. 24 Slow

Homogeneous Fast vs Slow Cores



Heterogeneous vs. Homogeneous



Double 0.998 throughput
or buy 50% fewer servers

Homogeneous

A. 2 Big

B. 10 Medium

C. 24 Small

FOF on Heterogeneous

D. 1 Fast
+ 4 Medium
+ 2 Slow

Opportunities on Existing Data Center Hardware

- SMT (Simultaneous Multithreading) or Hyperthreading
- SMT creates asymmetry among cores
  - Fast core: a physical core only runs one job
  - Slow core: two logical cores belonging to the same physical core both run jobs

Insight
SMT = dynamic heterogeneous core

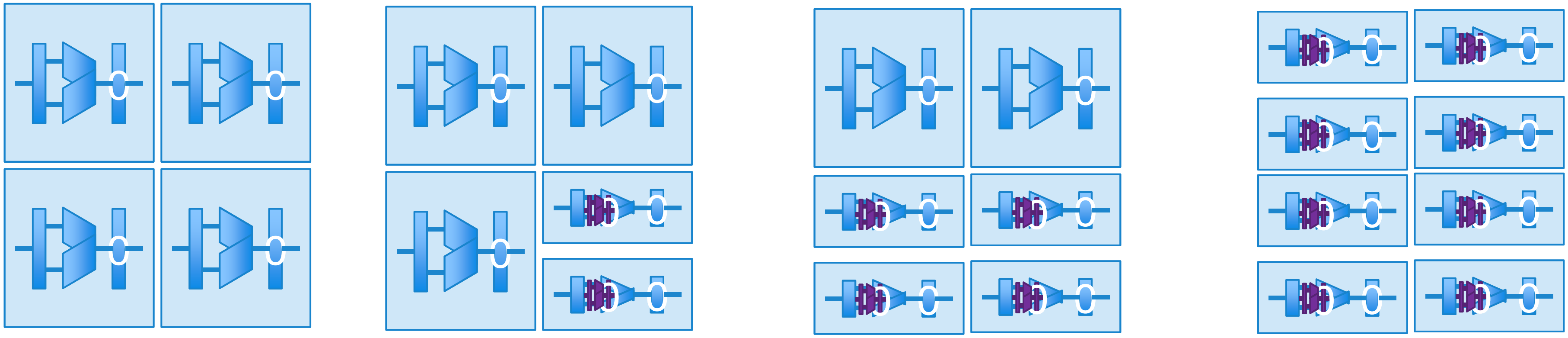
4 fast

3 fast+
2 slow

2 fast +
4 slow

…

8 slow



SMT off

SMT on
1 core

…

SMT on
all cores

Simultaneous Multithreading (SMT)

FOF Scheduler for SMT

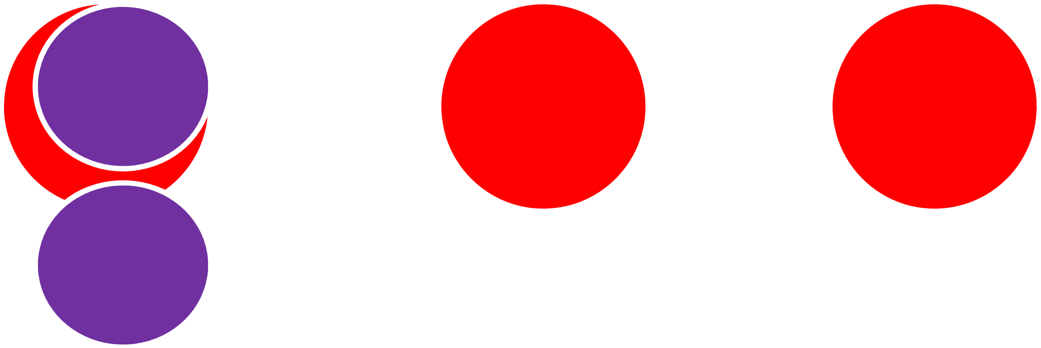
1. Fast first

   Fastest = unshared core

2. Fast old

   free core? Find shared pair (oldest, X)

   move X to free core



Evaluation

- Implementation on Finance application: Monte-Carlo computation for option price
- Hardware: 6 Core 2-way SMT 3.33 GHz Intel Xeon X5680
  - shared (slow) smt-core speed = 0.63 x unshared (fast) core speed
- FOF achieves
  - 16% higher throughput than default OS scheduler while meeting QoS

Conclusions

- FoF scheduler for interactive services
  - Exploit hardware heterogeneity
  - Achieve both high quality and high throughput
- Heterogeneous servers: Bing search simulation
  - Double throughput while meeting QoS
- SMT: Finance server implementation
  - 16% higher throughput than default OS scheduler

Thank you & Questions