

Using Cohort Scheduling to Enhance Server Performance

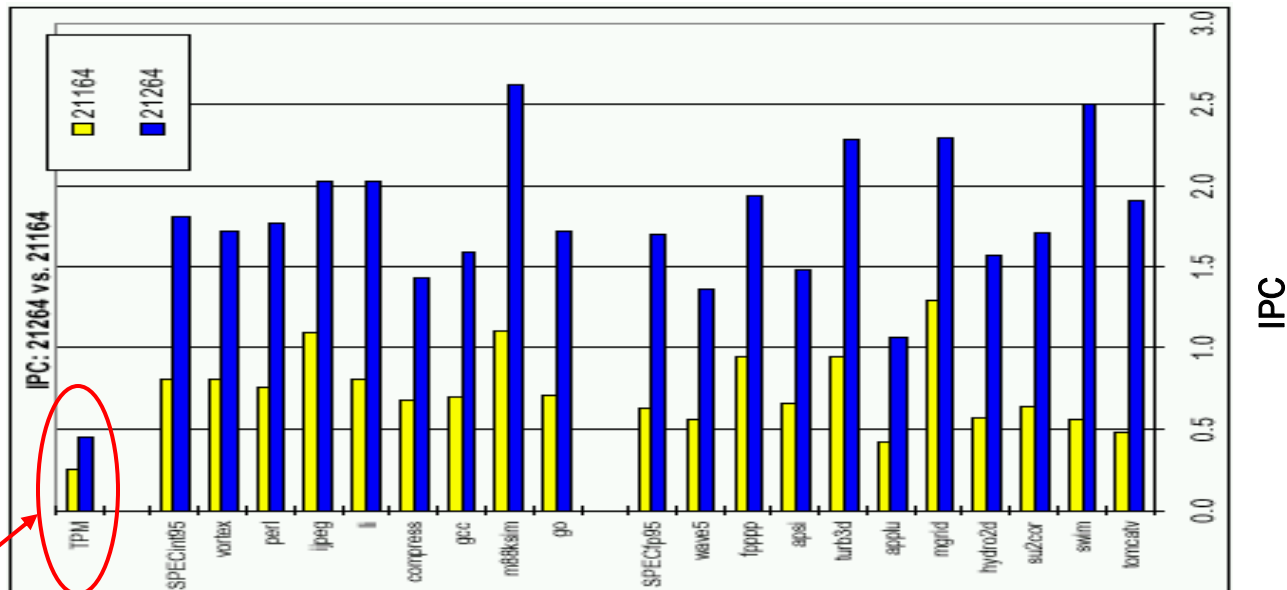
James Larus
Michael Parkes

Microsoft Research

Usenix Technical Conference
June 2002

Performance Problem

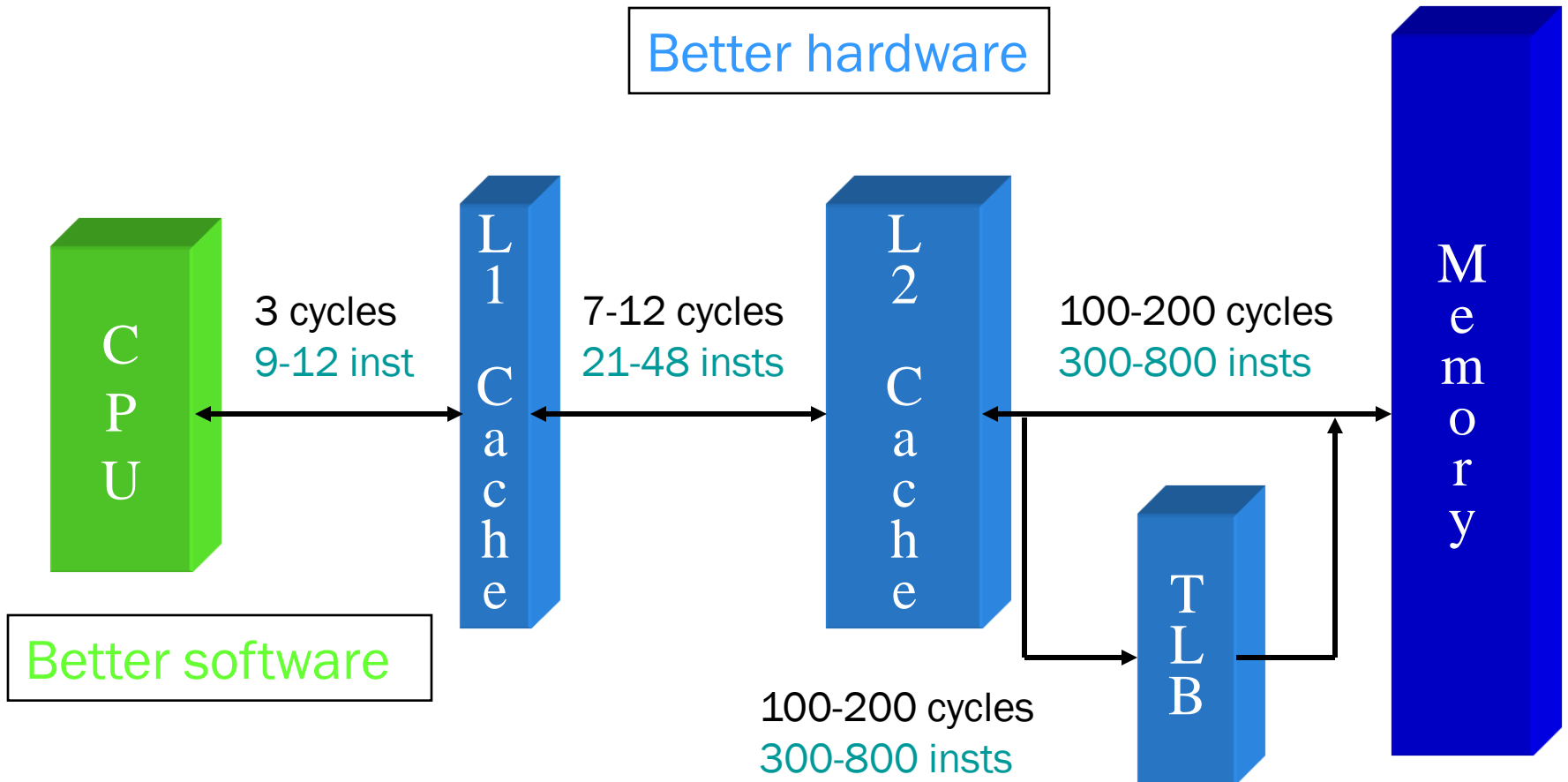
- Servers use fraction of modern processors
 - symptom: low IPC on multiple issue machine
 - consequence: busy processor, less work done



DB: 1/2 Inst./Cycle

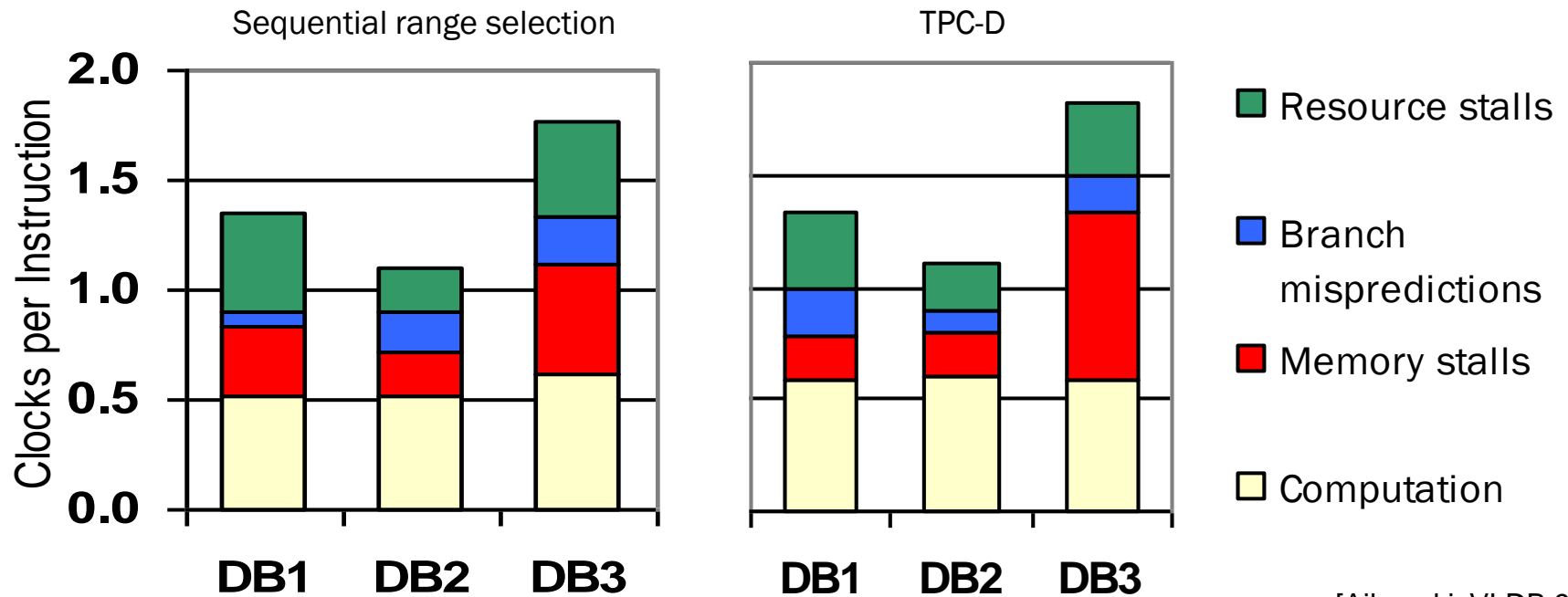
[Cvetanovic, ISCA 2000]

Processor-Memory Systems



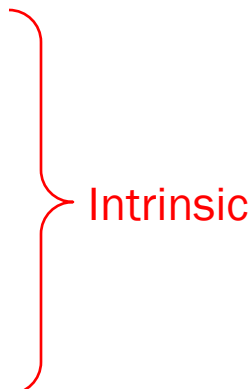
Poor Locality = Poor Performance

- Caches, branch predictors, etc. exploit program locality

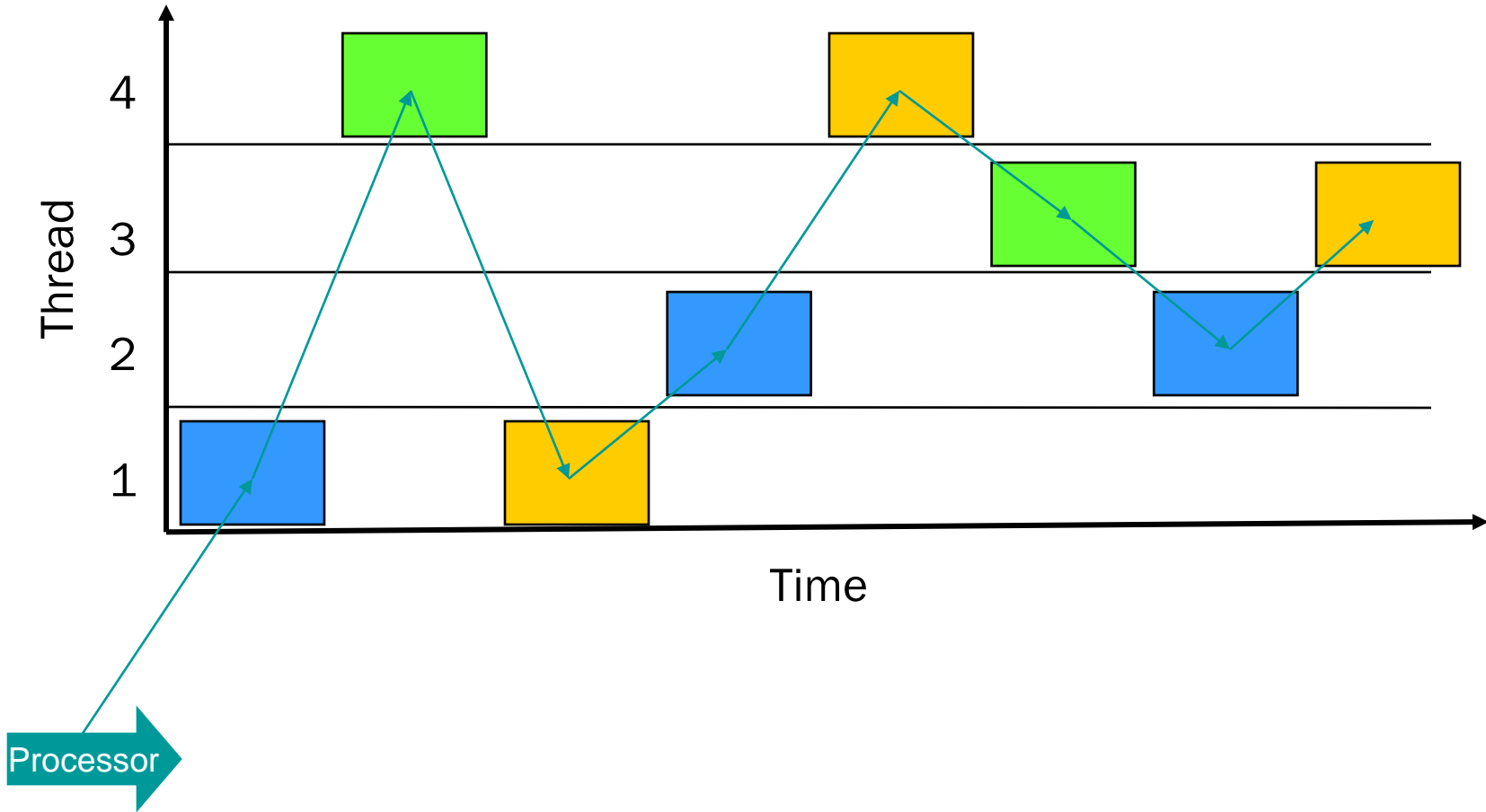


[Ailamaki, VLDB 99]

Why do Servers Have Poor Locality?

- Large programs
 - Large data sets
 - Multiple, concurrent tasks
 - Threads
- 
- Intrinsic

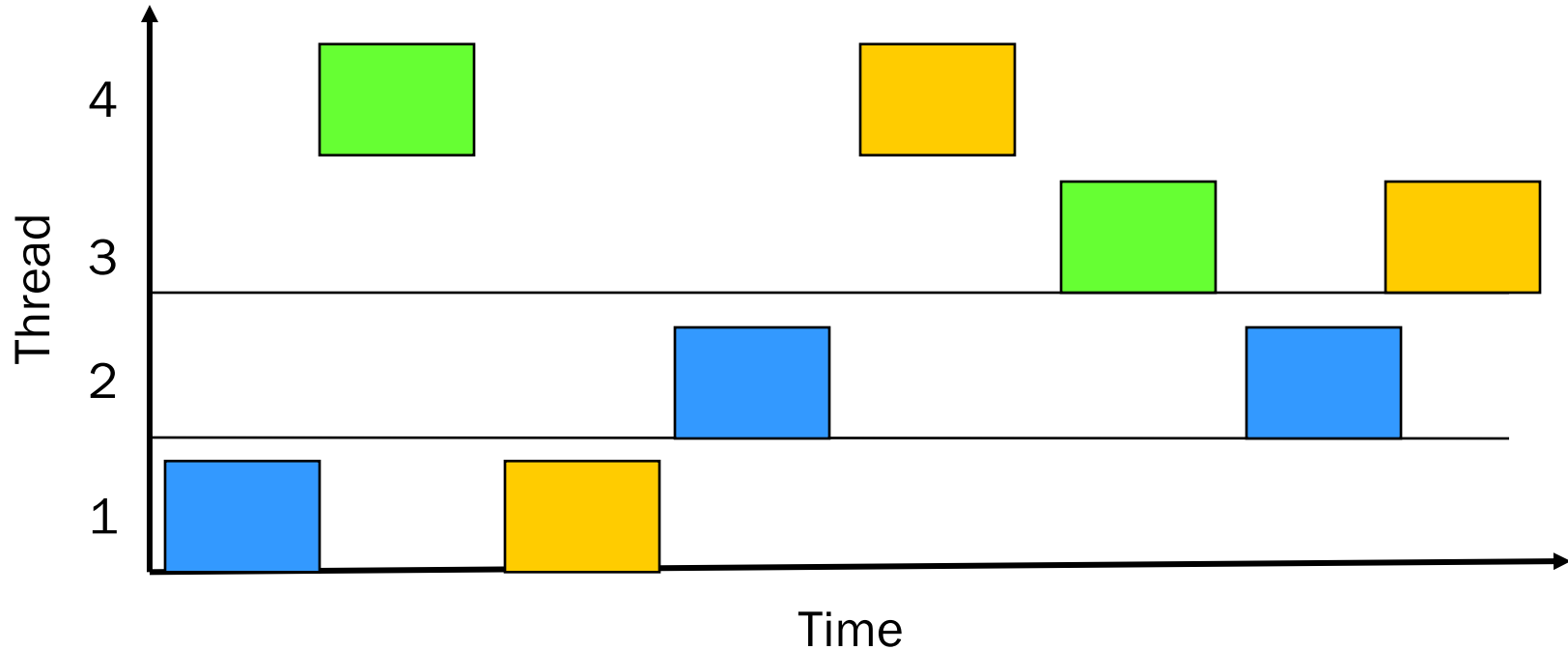
Processor-Thread Interaction



Talk Outline

- Cohort scheduling
- Staged computation
- StagedServer library
- Experiments

Cohort Scheduling



Reorder tasks into cohorts

Processor

June 2002

Cohort Scheduling — James Larus

8

Key Insight

- Cohort scheduling natural for servers
- Concurrently executing many tasks
 - common operations in different tasks
 - assemble these operations into cohorts
- Reorder task wrt other tasks
 - preserve order within task

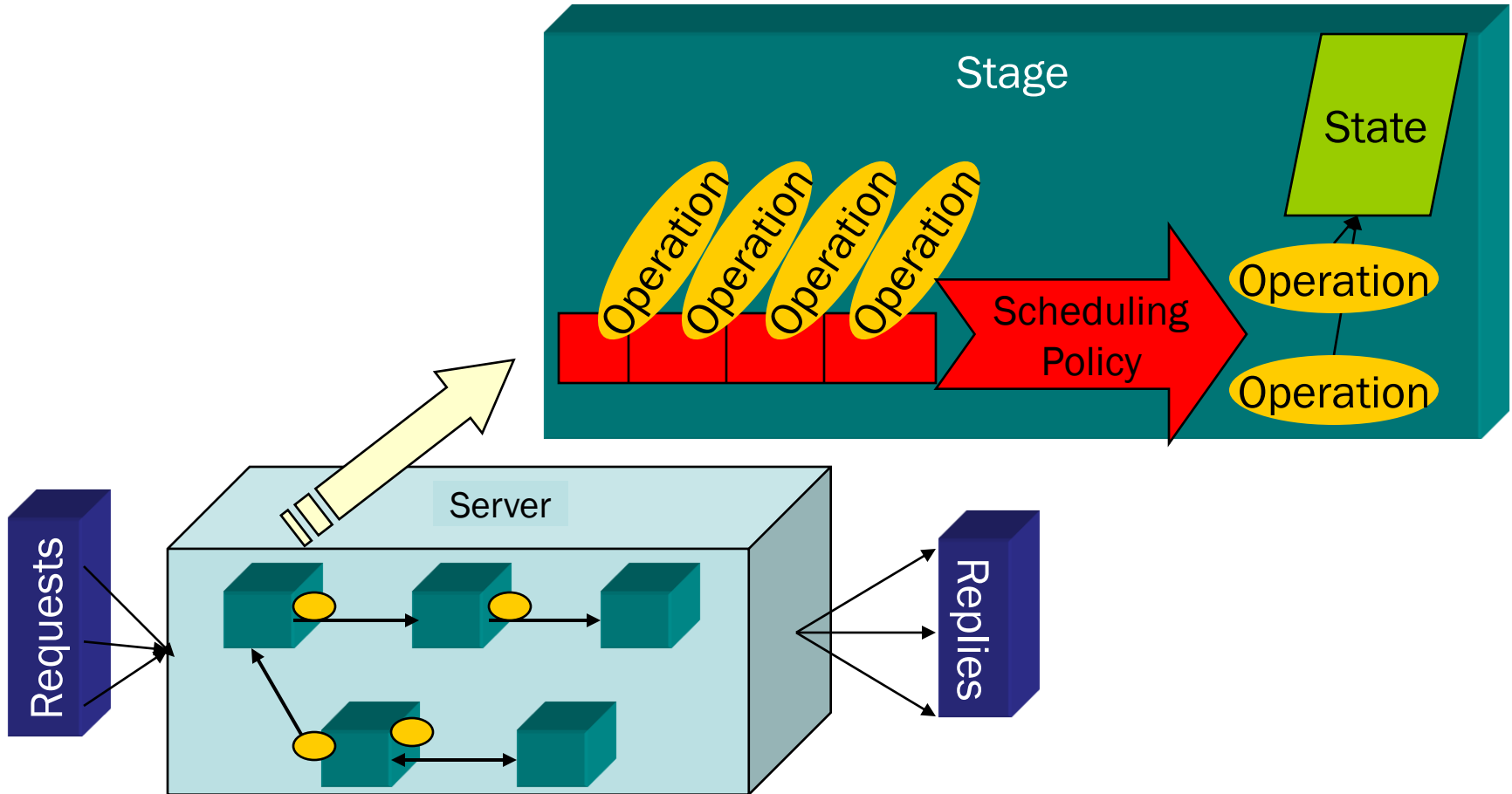
Talk Outline

- Cohort scheduling
- Staged computation
- StagedServer library
- Experiments

Staged Computation

- Programming model
- Support cohort scheduling
 - delimit operations and dependencies
- Better than threads
 - more structured and modular
 - less expensive, difficult synchronization
 - easier to verify

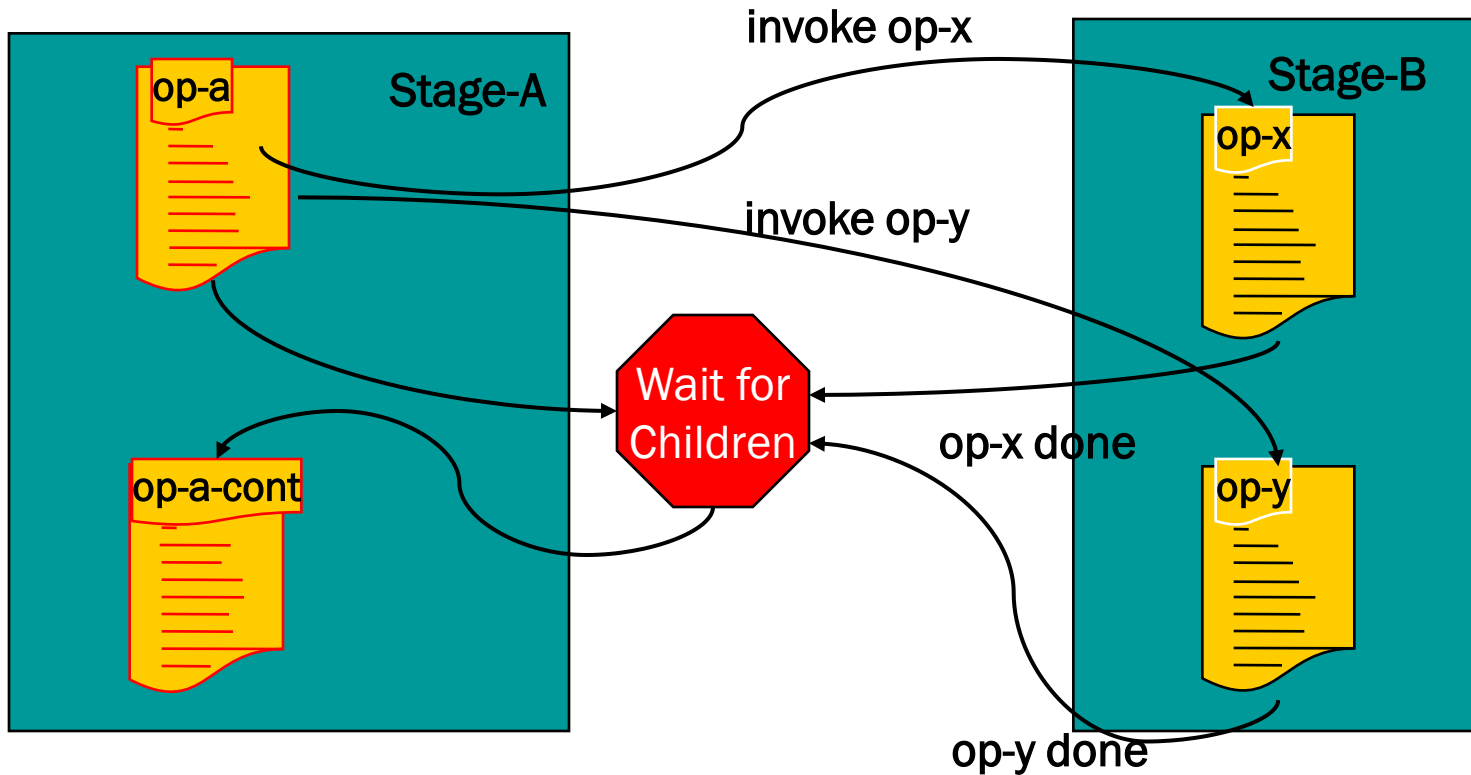
Staged Computation



Asynchronous Programming

- Actions in operation
 - compute (conventional—synchronous)
 - invoke (asynchronously) operations
 - wait for results

Staged Computation



Concurrency in a Stage

- Stage controls internal concurrency
 - supplant synchronization
- **Exclusive stage** (uniprocessor)
 - execute one operation at a time
 - access local data without synchronization
- **Shared stage** (SMP)
 - operations run concurrently on processors
- **Partitioned stage**
 - send operations to processor based on key
 - partition data, so processor can access w/o sync

Discussion

- Stages similar to distributed system
 - stage \sim SMP node
- Key difference: shared address space
 - pass around references
 - but, avoid communication through side effects
- Open question: benefits of uniformity

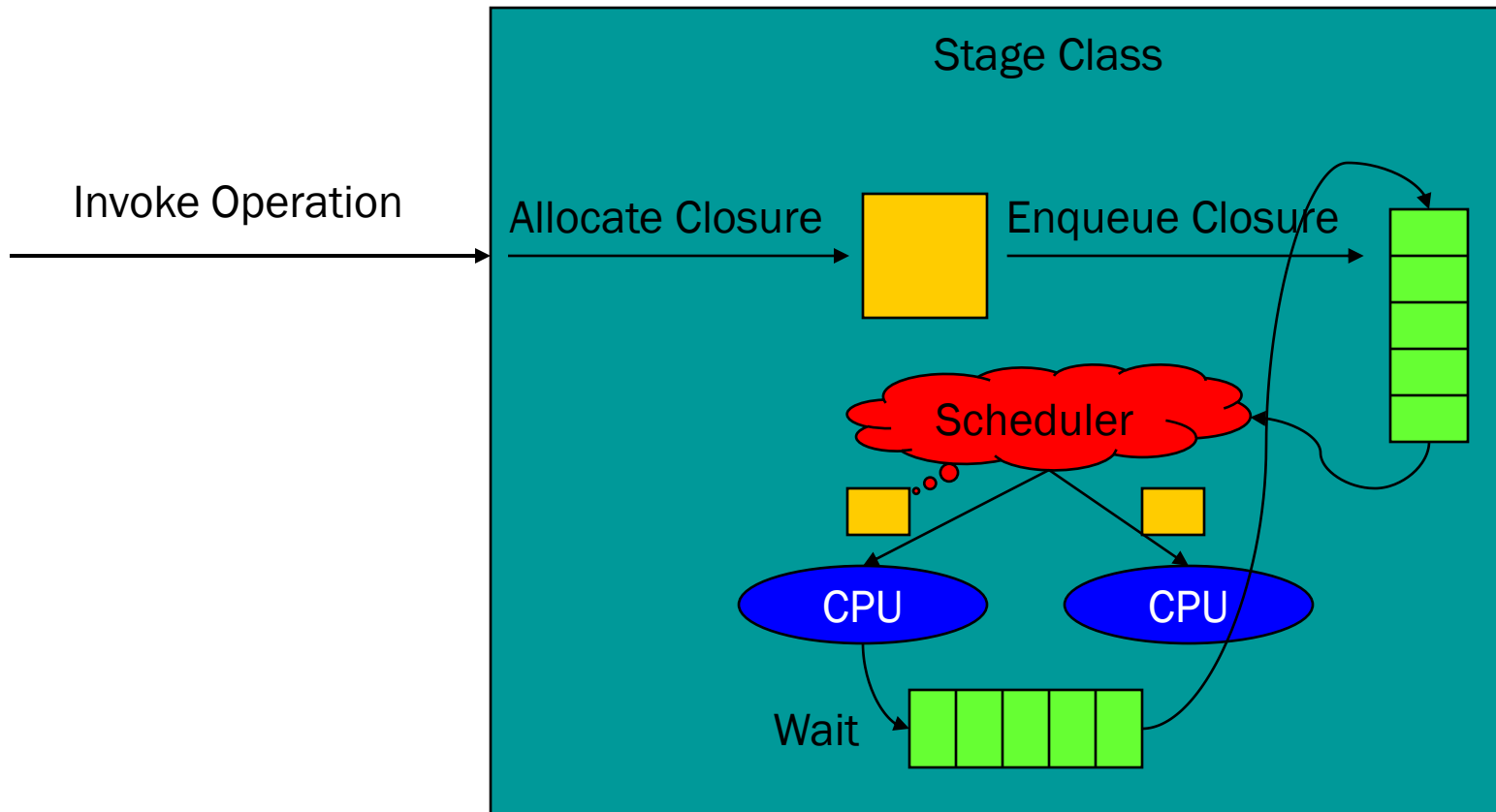
Talk Outline

- Cohort scheduling
- Staged computation
- StagedServer library
- Experiments

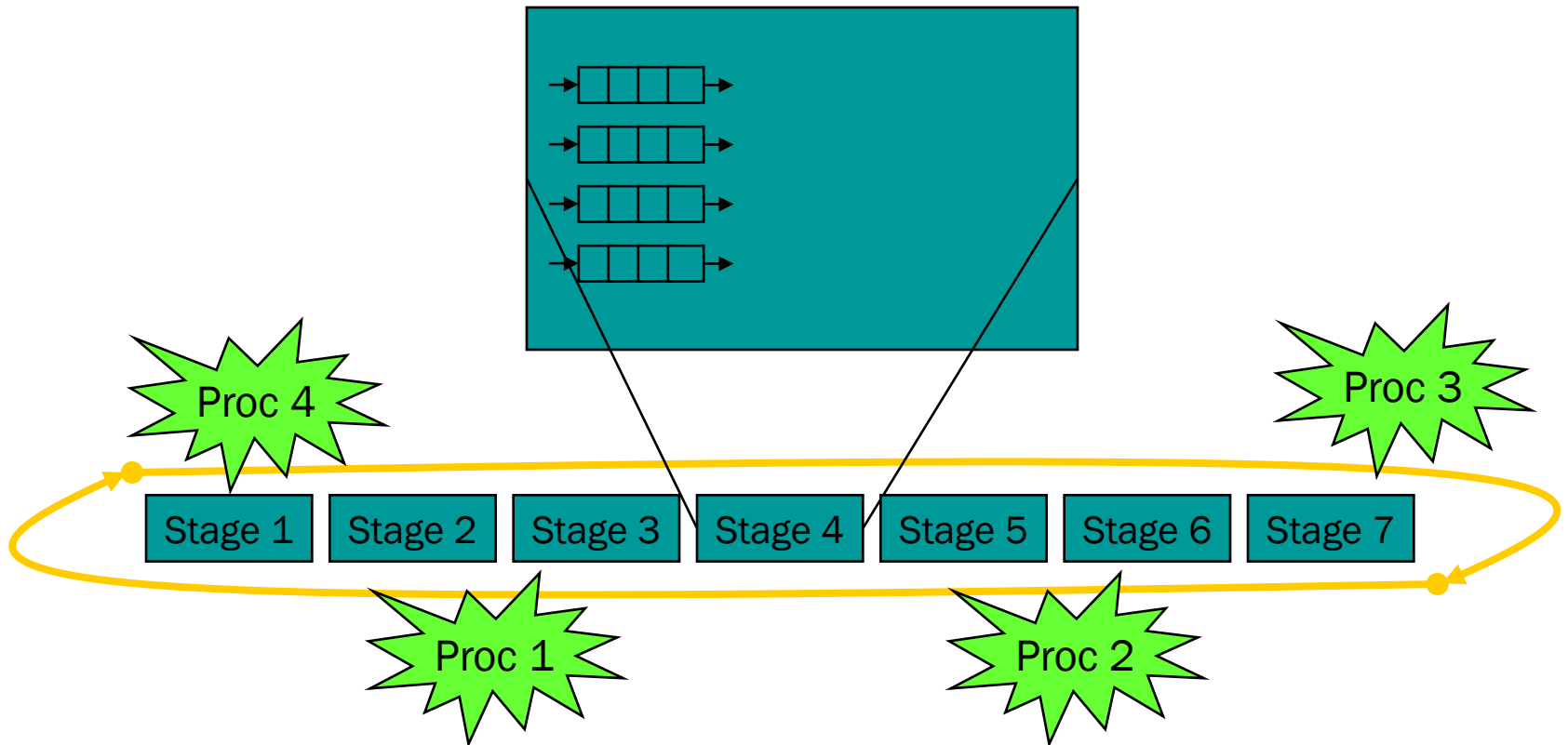
StagedServer

- C++ library
 - uniprocessor or SMP
 - framework for staged computation
 - aggressive cohort scheduling
- Two template classes
 - stage
 - closure
- Also COM version

Stage and Closure Objects



Assigning Processors to Stages



Aggressive Cohort Scheduling

- Processor affinity
 - Keep operation & children on processor
 - Ex: explicit placement, partitioning
- Distinguish local/remote work
 - Local: push on stack, process LIFO
 - Remote: queue (lock)
- Process local work first
 - Enhance locality

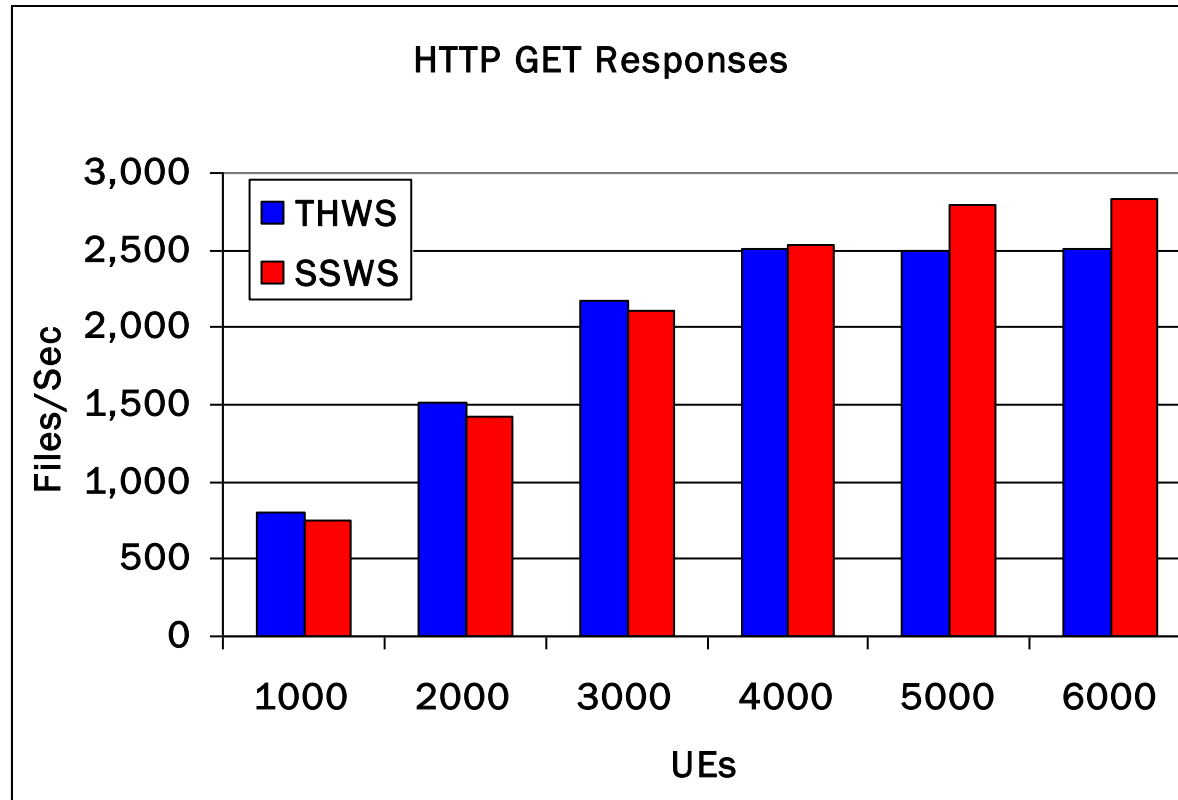
Talk Outline

- Cohort scheduling
- Staged computation
- StagedServer library
- Experiments

Tale of Two Servers

- I/O-bound server
 - simple, static web page
- Compute-bound server
 - publish-subscribe
- For each:
 - threaded “best practices” server
 - StagedServer server

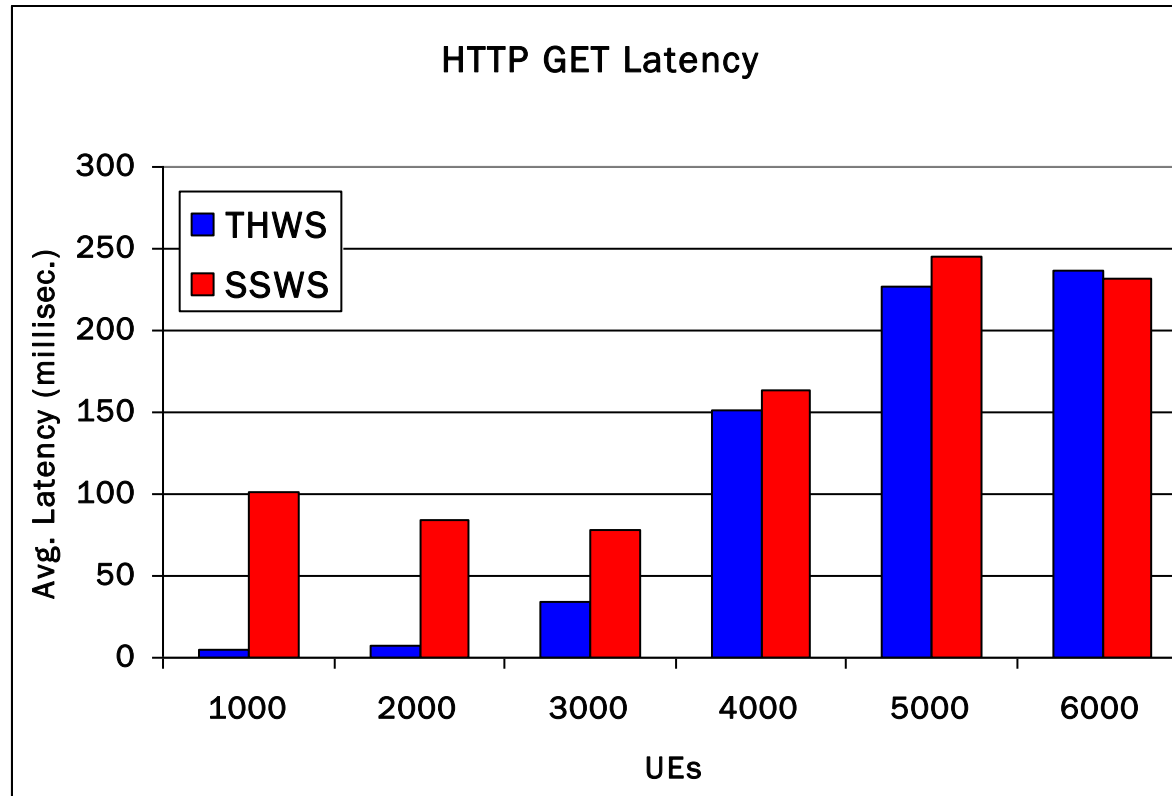
Web Server Bandwidth



Server: 4x700MHz Pentium III-Xeon
Clients: (3) 4x400MHz Pentium II Xeon

SURGE benchmark
1,000,000 pages (20.1GB)
6,638,449 requests

Web Server Latency



Publish-Subscribe

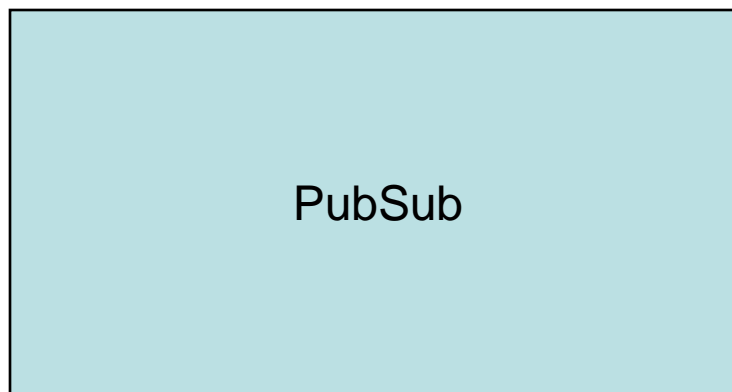
Events

SELECT FPD INDUSTRY STOCKS		
Company	Price	Change
App Films	24.08	-0.84
Corning	7.52	-0.26
DuPont	47.34	0.24
Kodak	30.76	-0.29
ellagin	0.68	0.01
iFine	1.13	0.07
Infocus	17.90	-0.10
JRCO	6.06	-0.15
Kopin	9.40	0.38
Kyocera	68.33	0.43
Microvisn	11.41	-0.14
Phillips	30.43	-0.48
Photon Dyn	48.88	-2.47
Pixtech	0.08	0.00
Sage	22.98	0.00
Sil. Image	8.60	0.01
Supertex	20.53	0.03
Three-Five	14.39	-0.11
UDC	9.35	-0.22

Subscriptions

Subscribe:
MSFT=100 & IBM=10

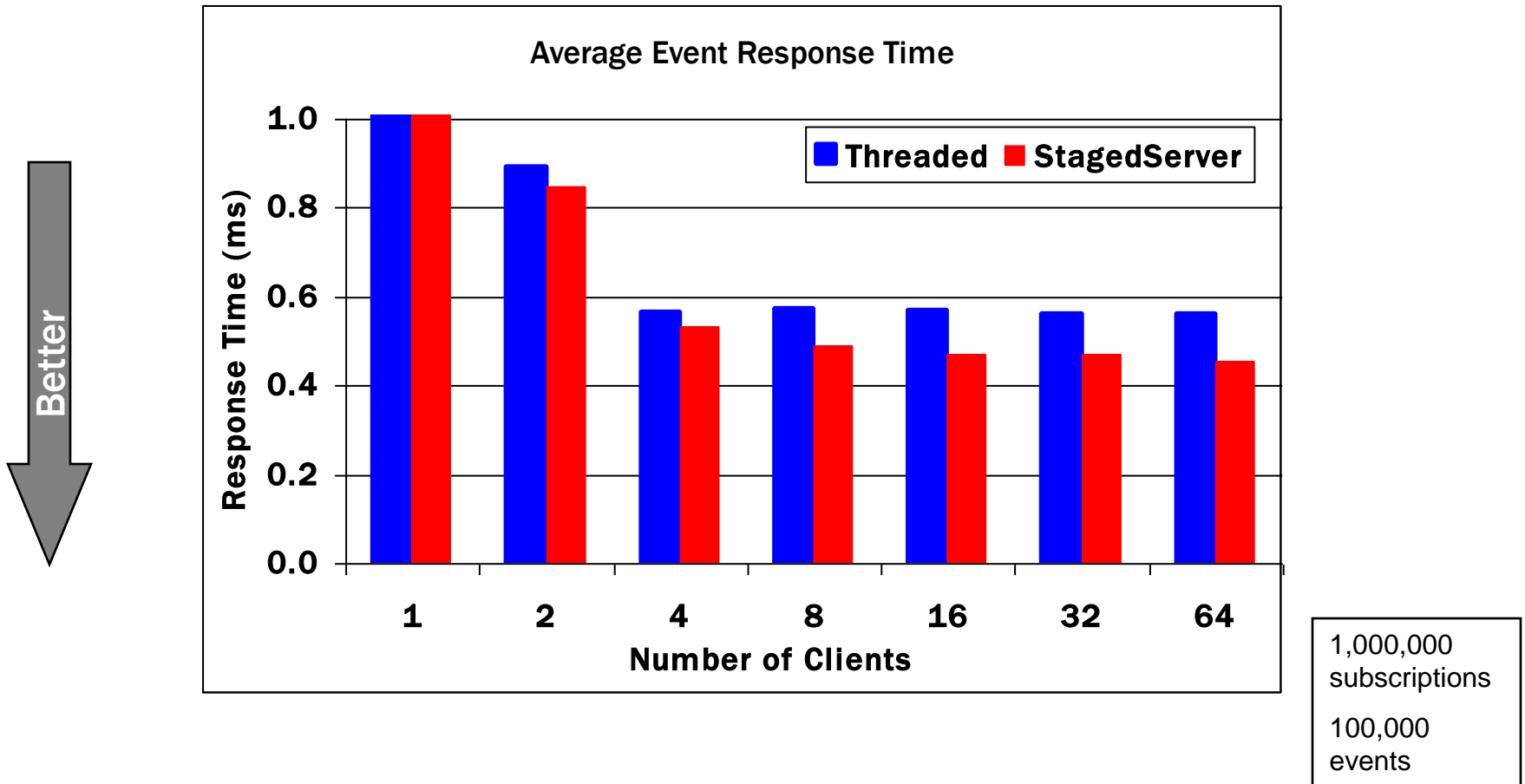
Subscribe:
RHAT=50 & MSFT = 5



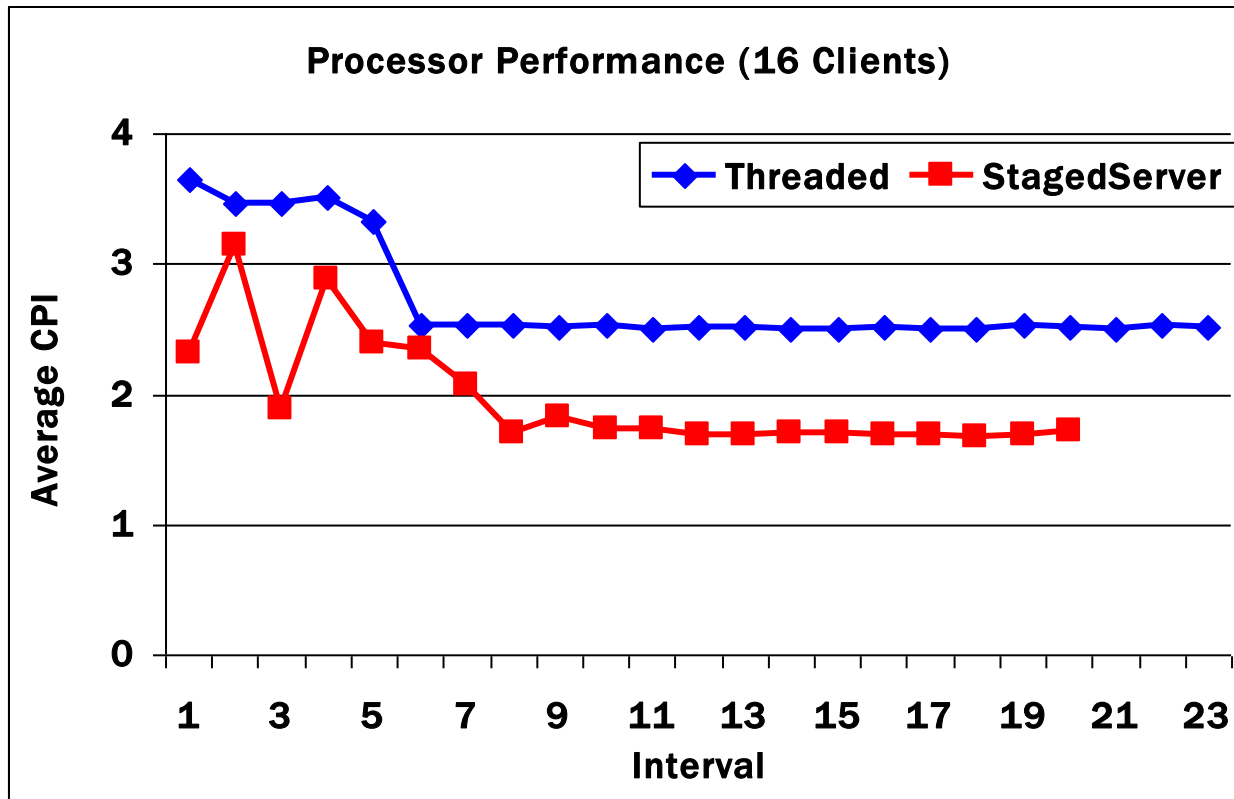
Notifications

Notify: MSFT=100
& IBM=10

PubSub Latency



PubSub IPC



Future Work

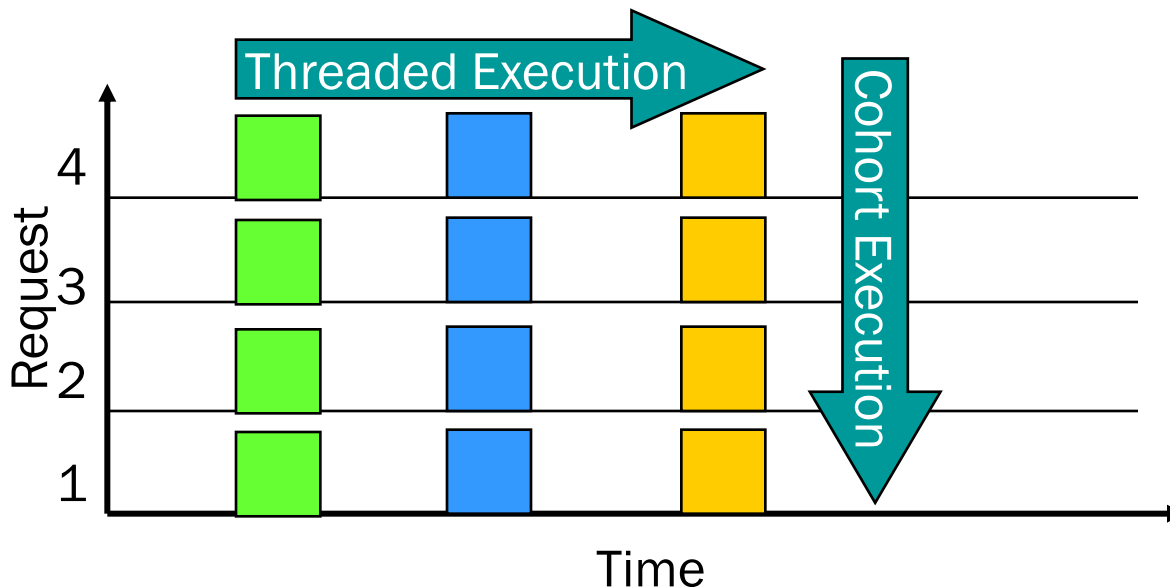
- Stage coordination language
 - describe system-wide communication and stage behavior
 - verify system properties
 - deadlock freedom, progress, don't lose work,...
- Extend to clusters
 - same semantics shared/non-shared memory
 - reconfigure without rewriting

Key Points

- Processor performance is a software problem
 - better hardware helps, but ...
 - hardware exploits predictability and locality
- Programs with little locality perform poorly
- Think beyond threads
 - well-known programming difficulties
 - little intrinsic locality

Cohort Scheduling

- Enhance locality by grouping similar operations



Staged Computation

- Identifies cohorts
- Supports cohort scheduling
- Reduces synchronization