Università della Svizzera italiana

Faculty of Informatics

Jopera
Process Support for Web Services

Exploiting Multicores to Optimize Business Process Execution

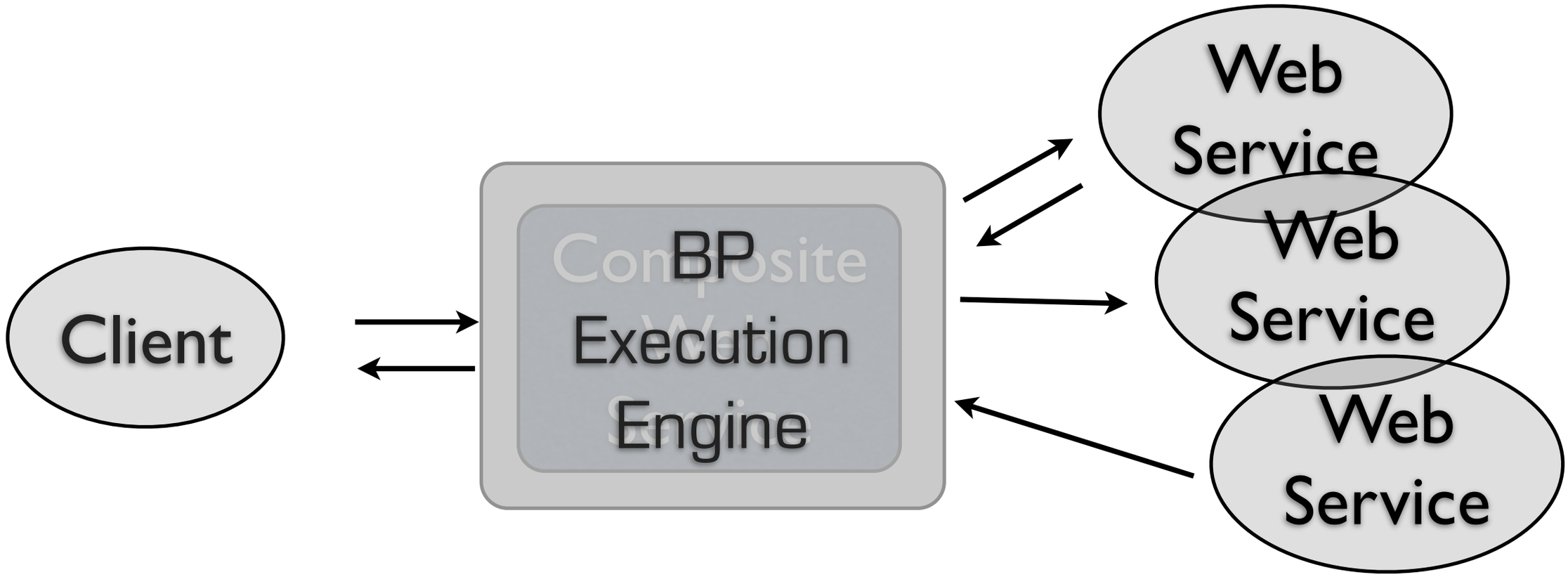
Daniele Bonetta,
Achille Peternier, Cesare Pautasso, Walter Binder

Faculty of Informatics
University of Lugano - USI
Switzerland
http://sosoa.inf.usi.ch

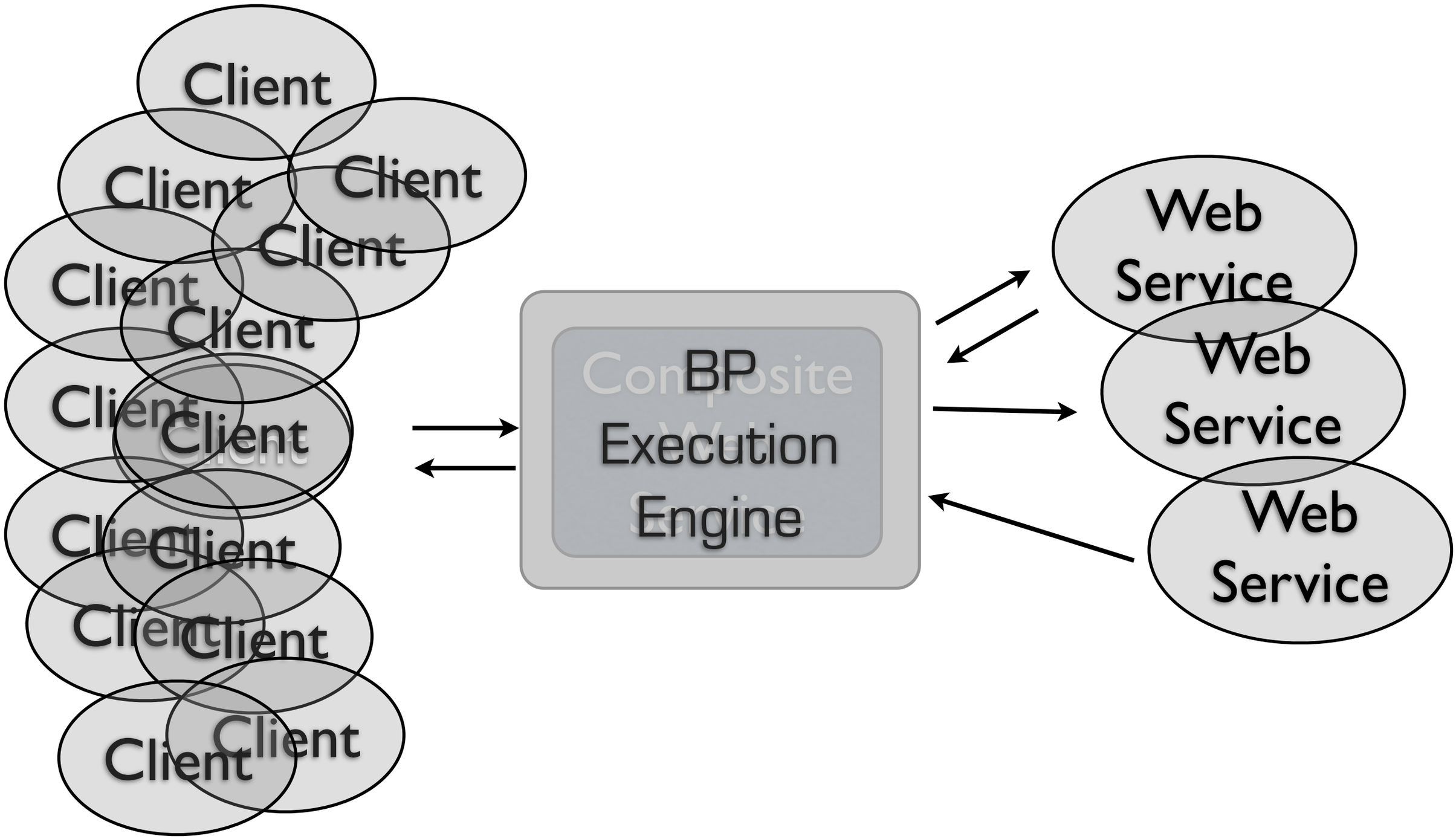
daniele.bonetta@usi.ch

BP Execution Engine

Focus: Business Process Runtime Execution Environment

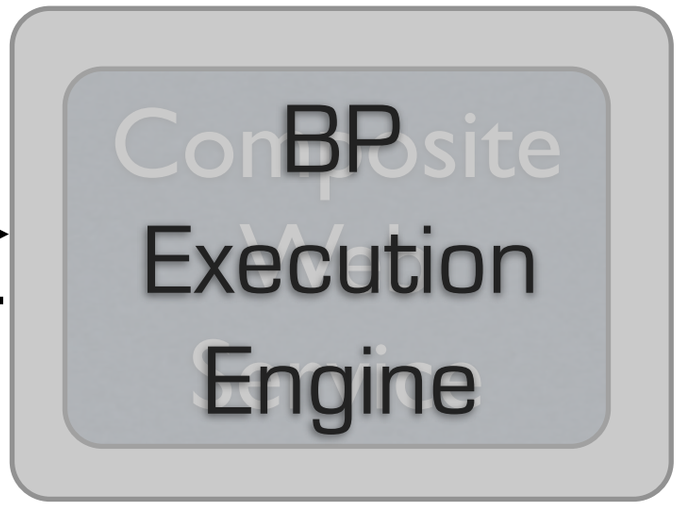


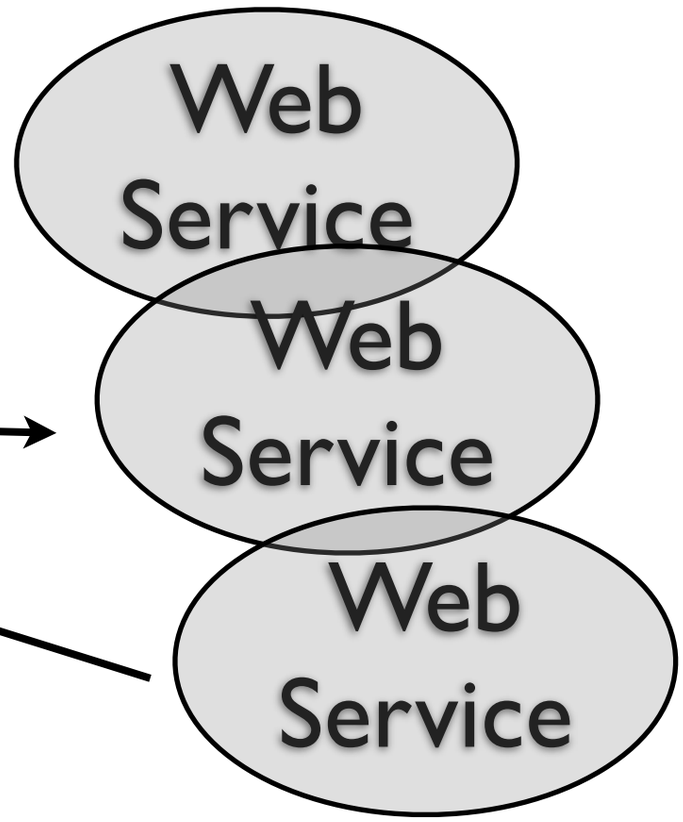
How to *scale*?


Client
Client
Client
Client
Client
Client
Client
Client
Client
Client
Client
Client
Client
Client
Client
BP
Execution
Engine
Web
Service
Web
Service

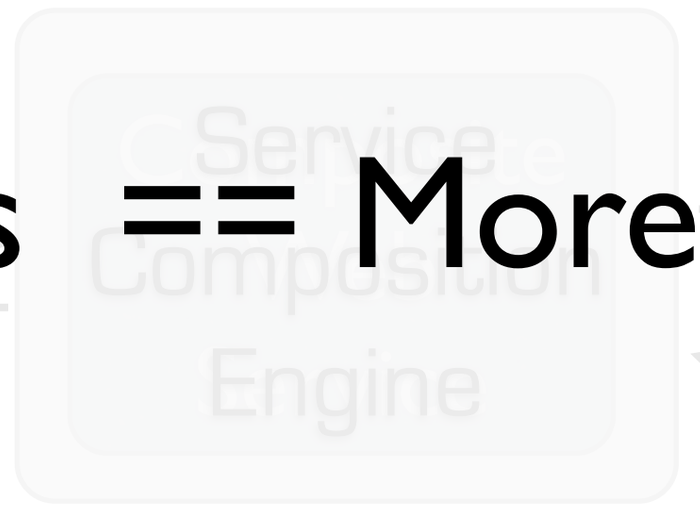
How to *scale*?

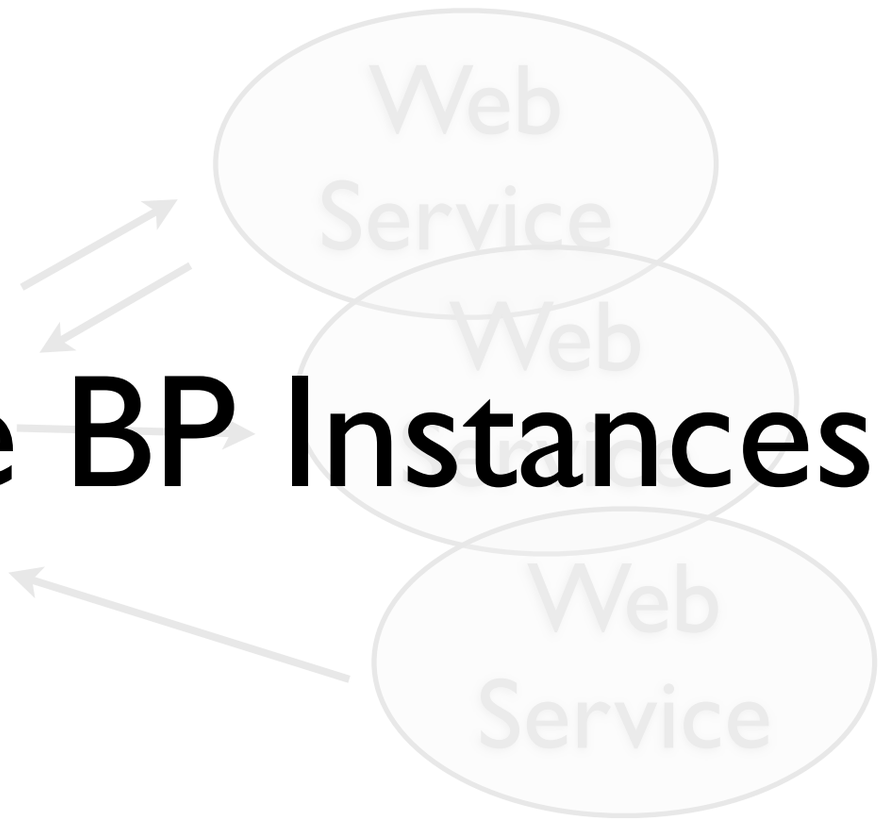

Client


BP
Execution
Engine


Web
Service
Web
Service
Web
Service

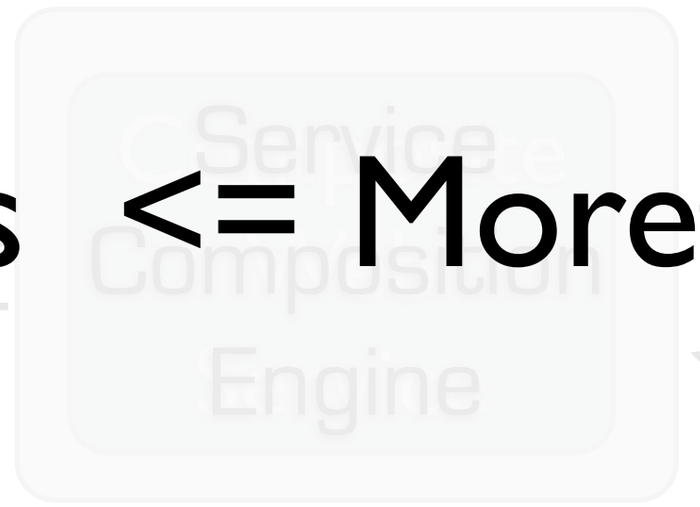
How to *scale*?





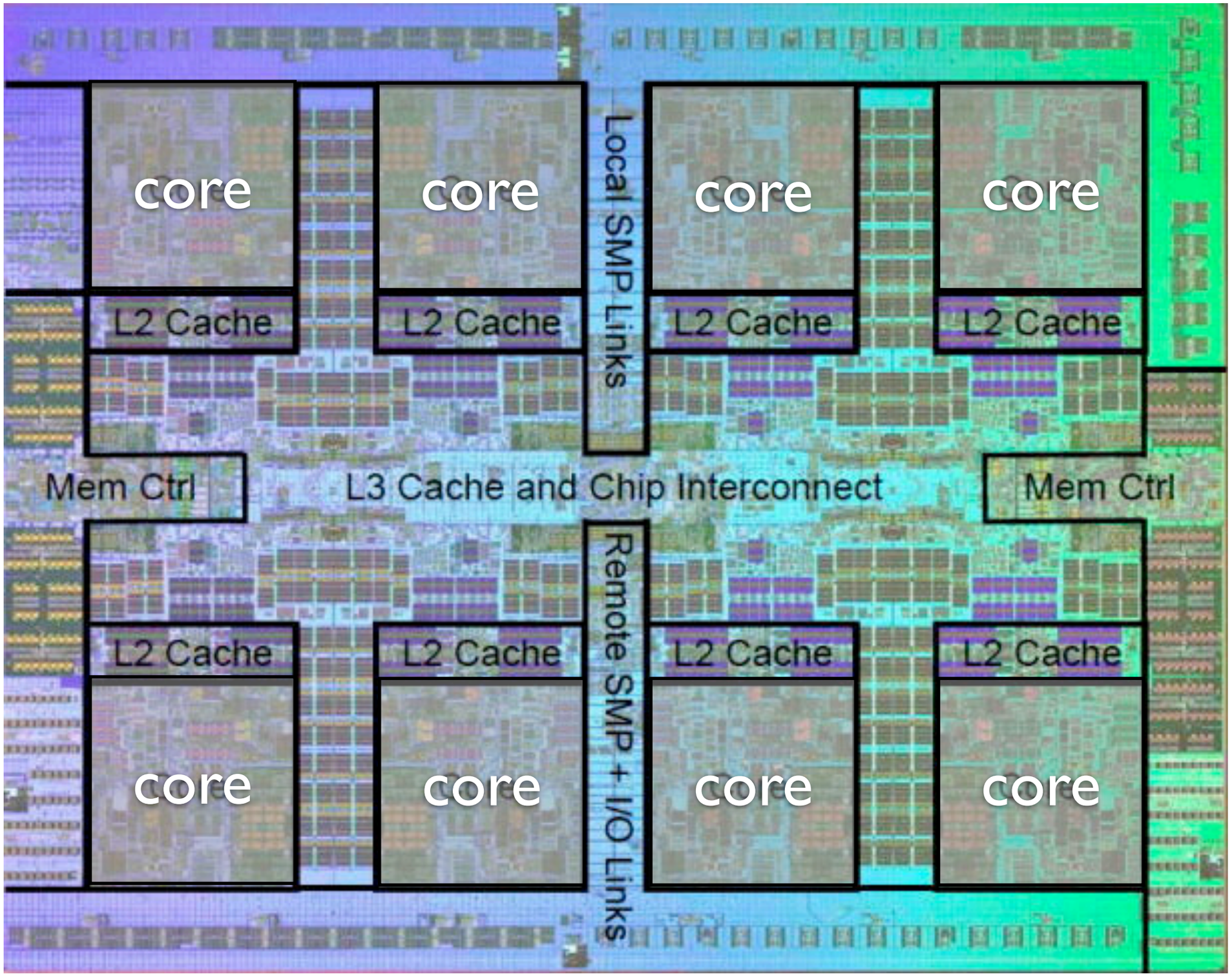
More clients == More BP Instances

How to *scale*?



More clients <= More BP Instances

Multicores



IBM Power7

Outline

1. Multicore Issues

2. JOpera Business Process Execution Engine

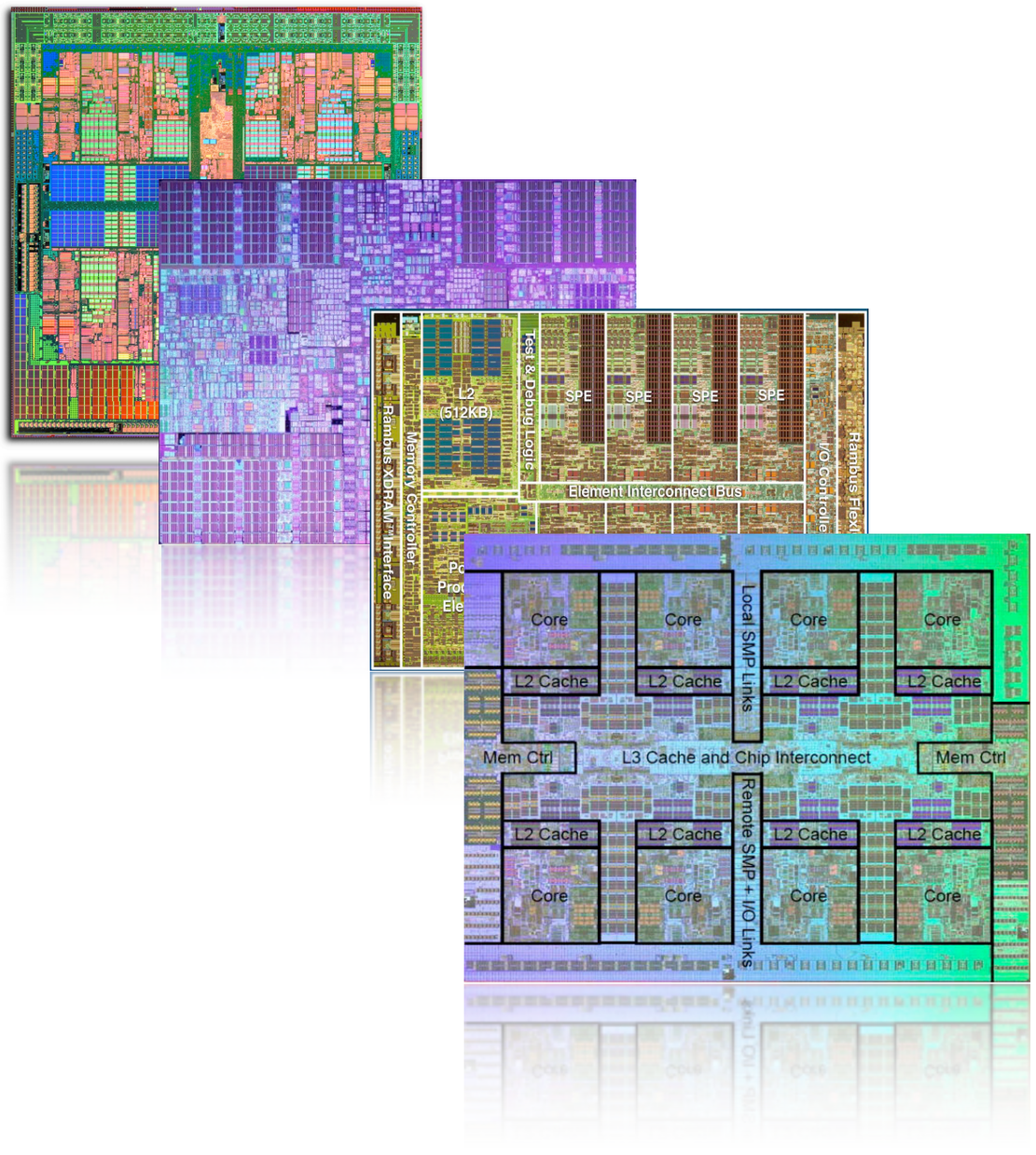
    1. Thread Level Parallelism

    2. CPU/Core Level Parallelism

3. Experimental Results

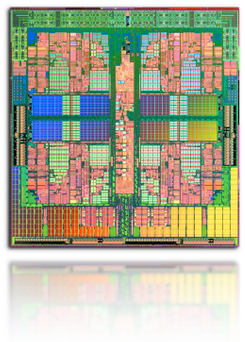
4. Conclusion

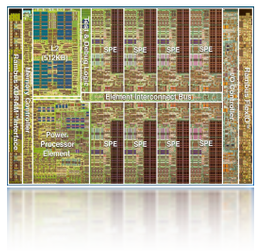
Multicore Issues

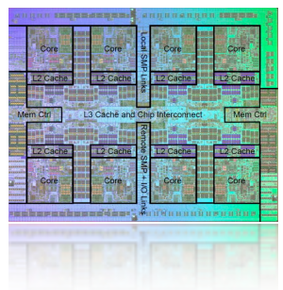


- Number of cores
- Type of cores (e.g. SMT)
- On Chip Caching Layout (e.g. L2, L3...)
- On Board Memory Layout (e.g. NUMA, NUMA-CC, ...)

Multicore Issues





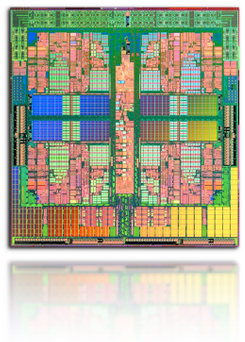


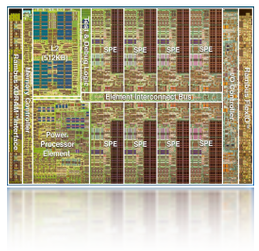
- Cores Num
- Cores Type

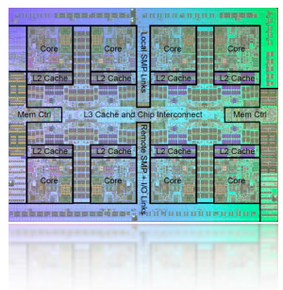
→ Th Migrations, Ctx Switches

- Cache Layout
- Memory Layout

Multicore Issues







- Cores Num
- Cores Type

→ Th Migrations, Ctx Switches

- Cache Layout
- Memory Layout

→ Data Locality, Contention

BP Execution Engine

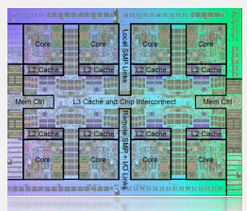
Java Business Process Execution Engine

3 Layers Approach

Concurrent Business Process Instances

OS Threads

Hardware Cores



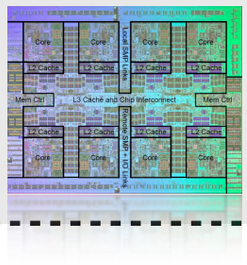
Abstraction Layers

Concurrent Business Process Instances

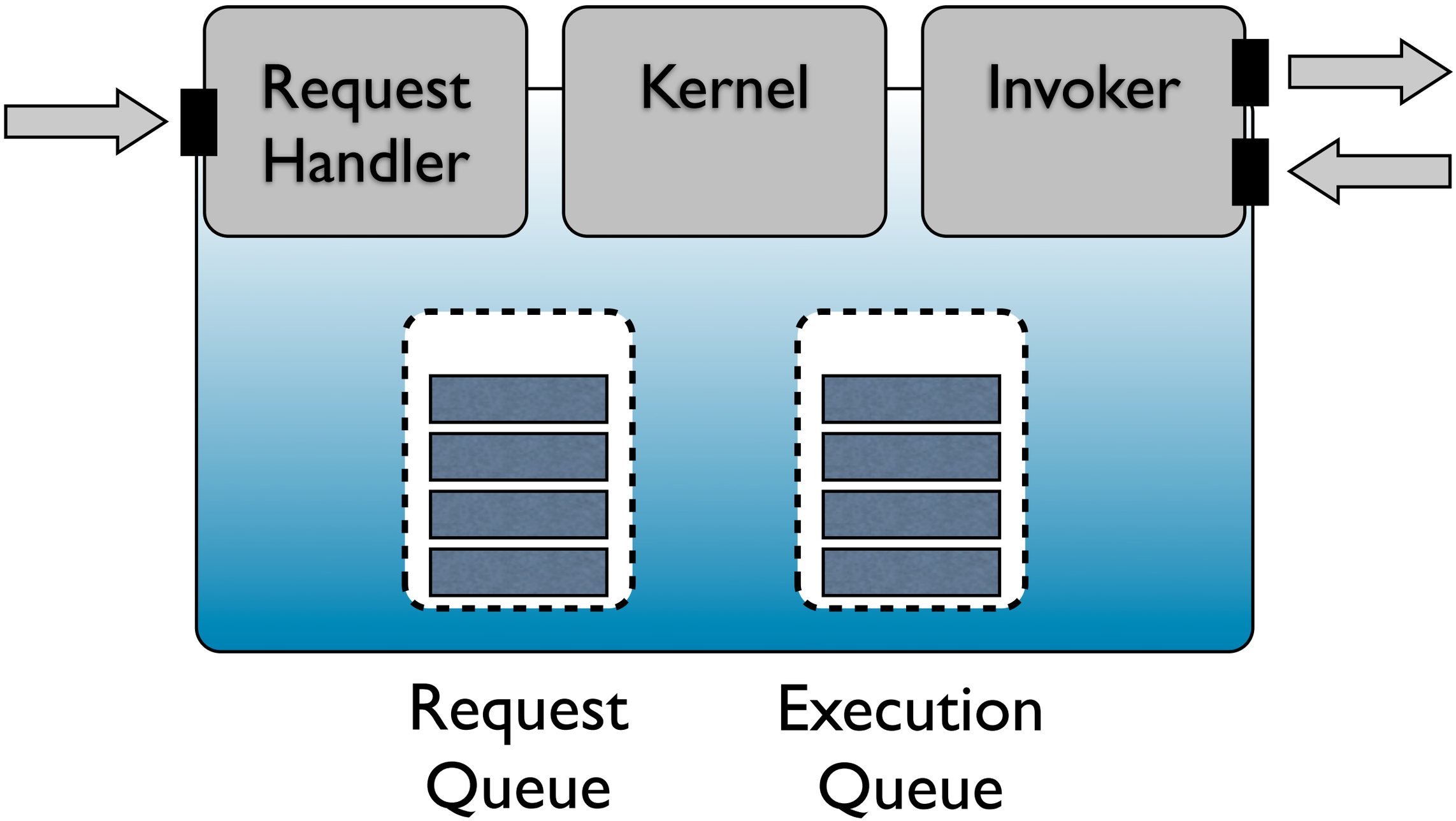
Jopera
Process Support for Web Services

OS Threads

Hardware Cores



Engine Architecture



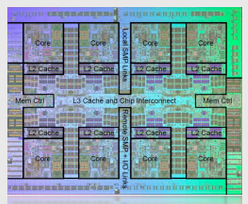
Abstraction Layers

Concurrent Business Process Instances

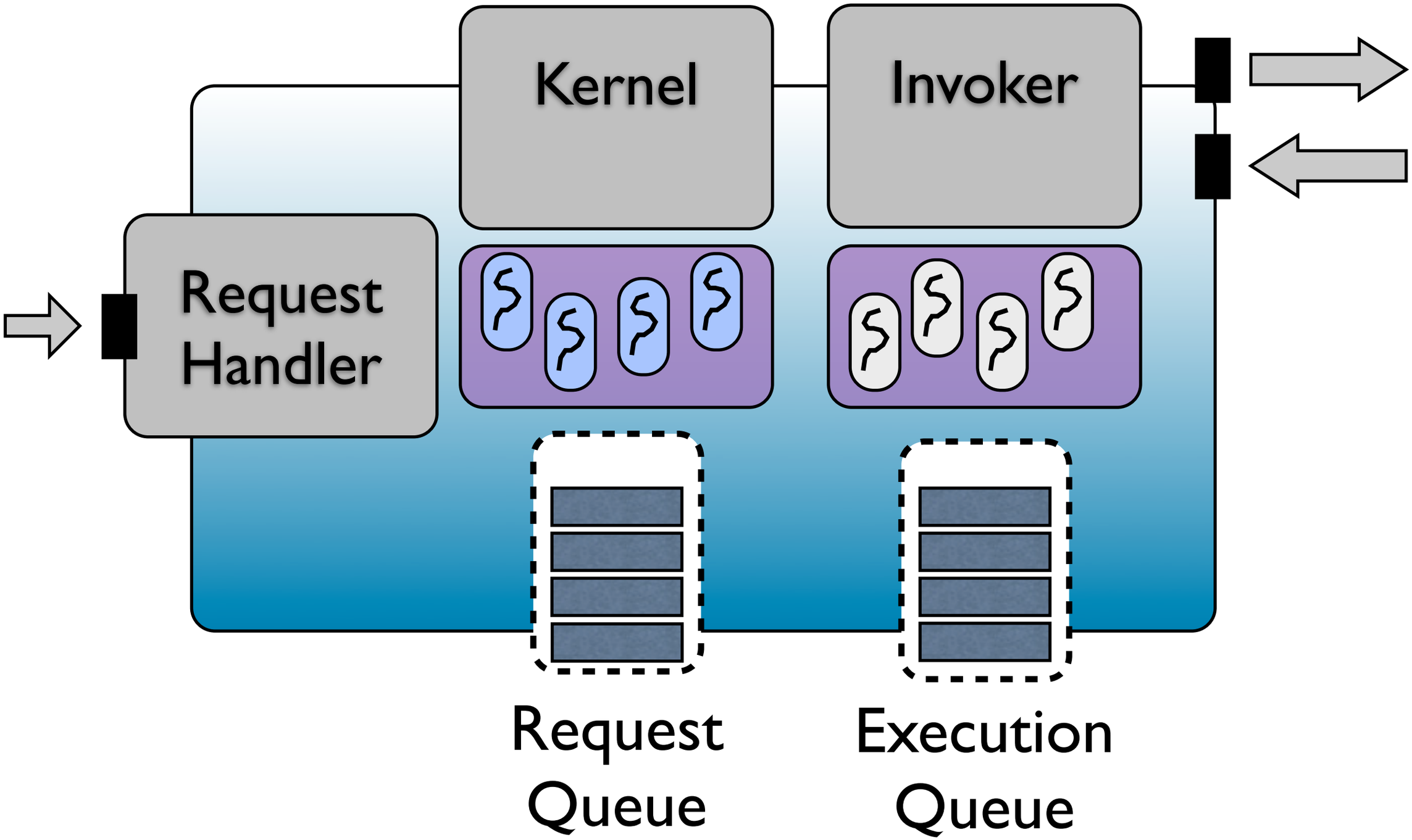
Jopera
Process Support for Web Services

OS Threads

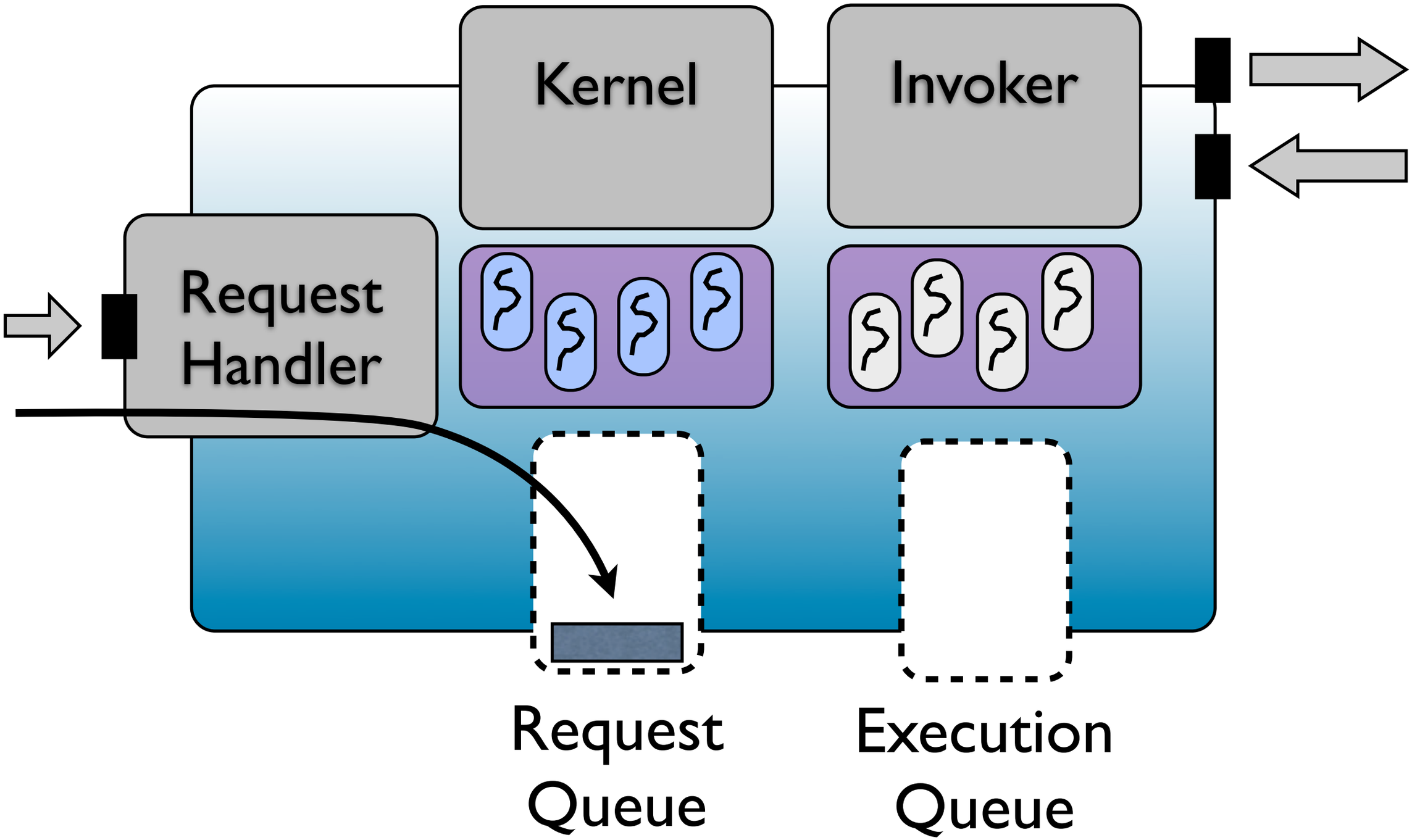
Hardware Cores



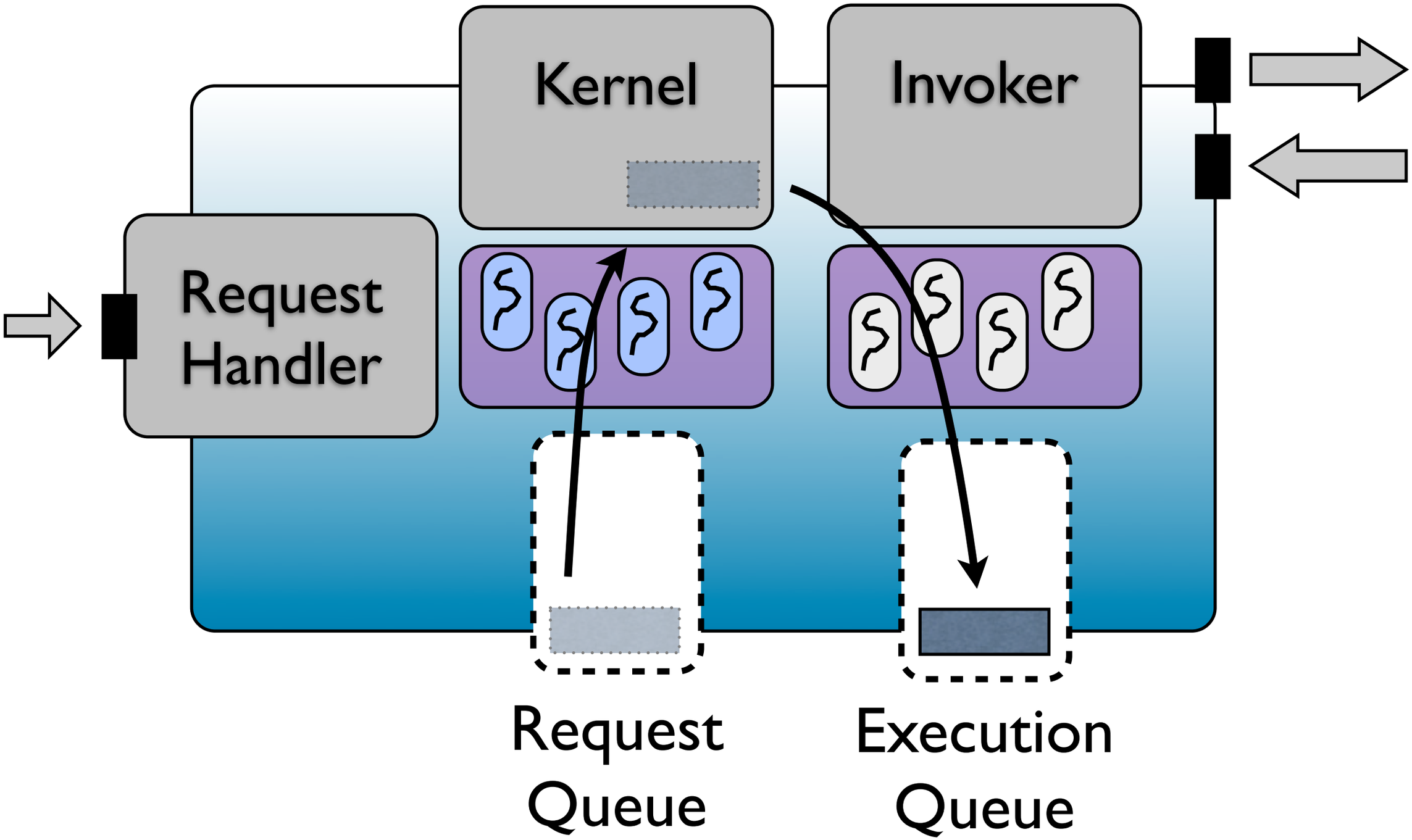
Engine Architecture



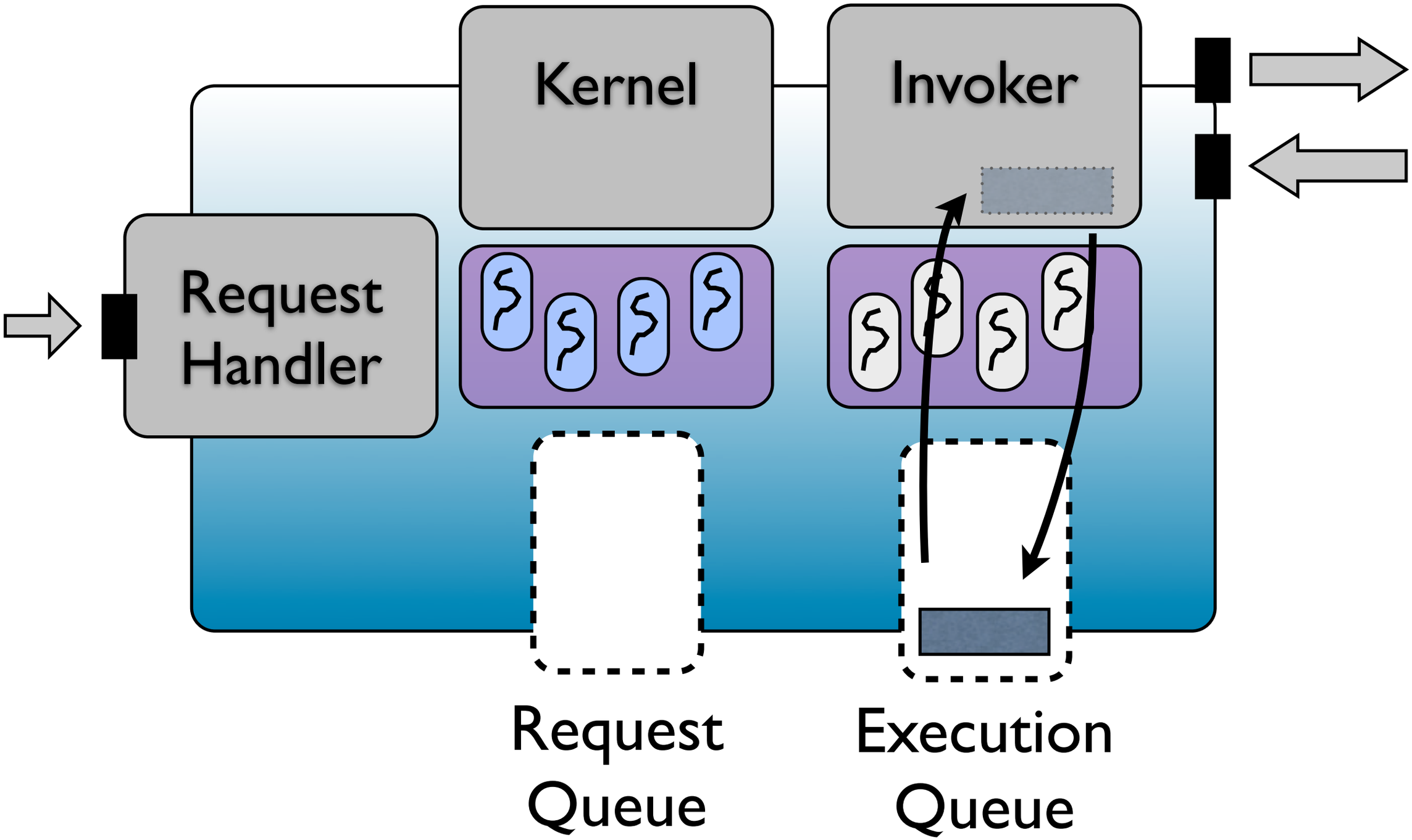
BP Execution



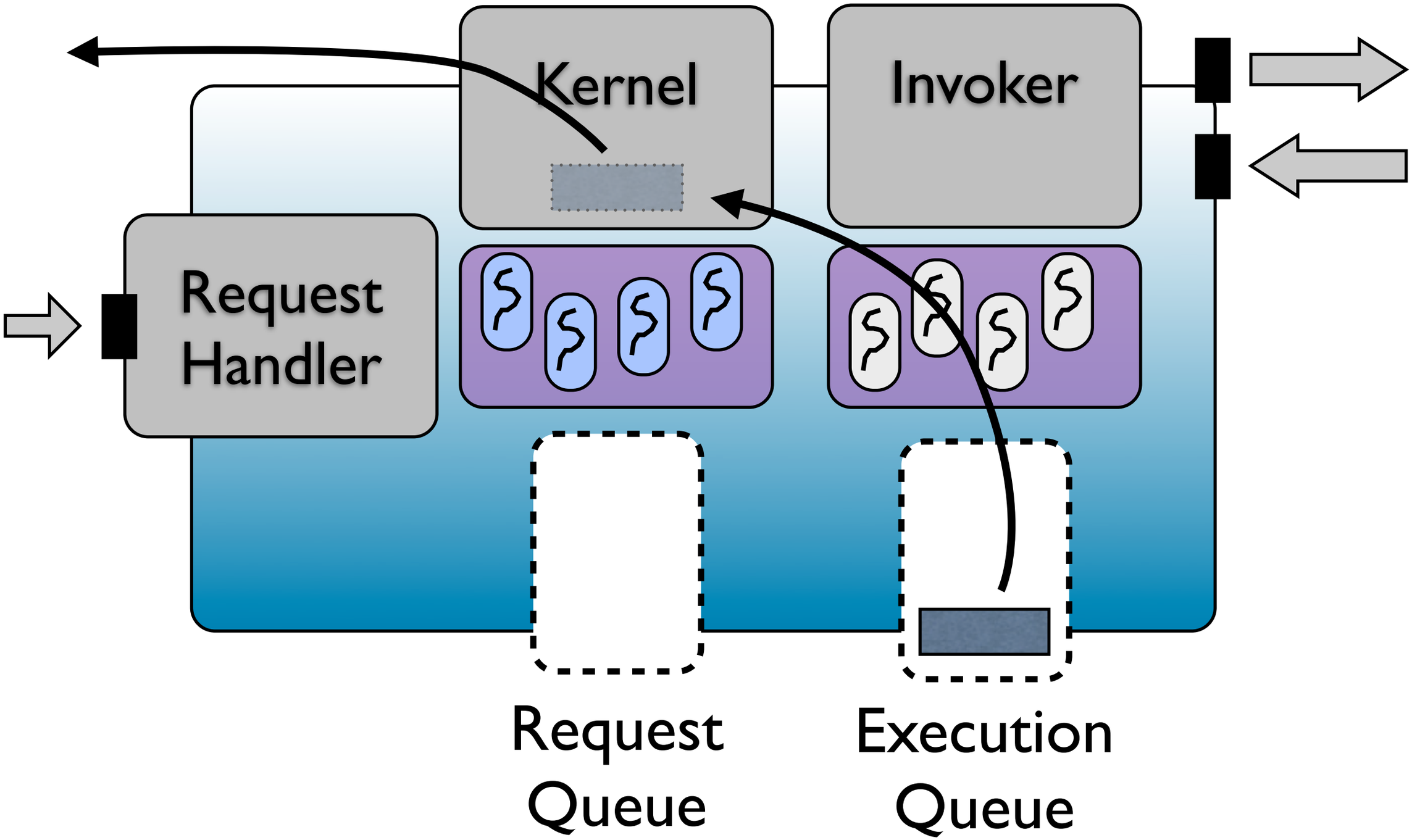
BP Execution



BP Execution



BP Execution

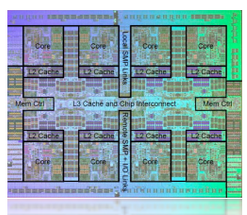


Abstraction Layers

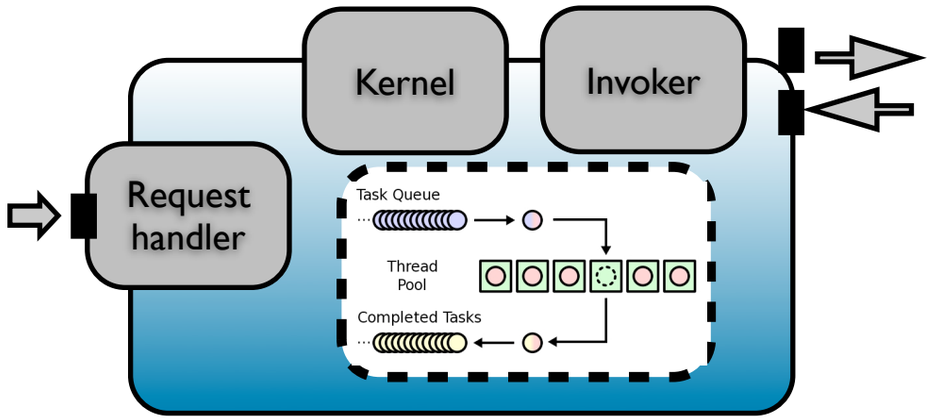
Concurrent Business Process Instances

OS Threads

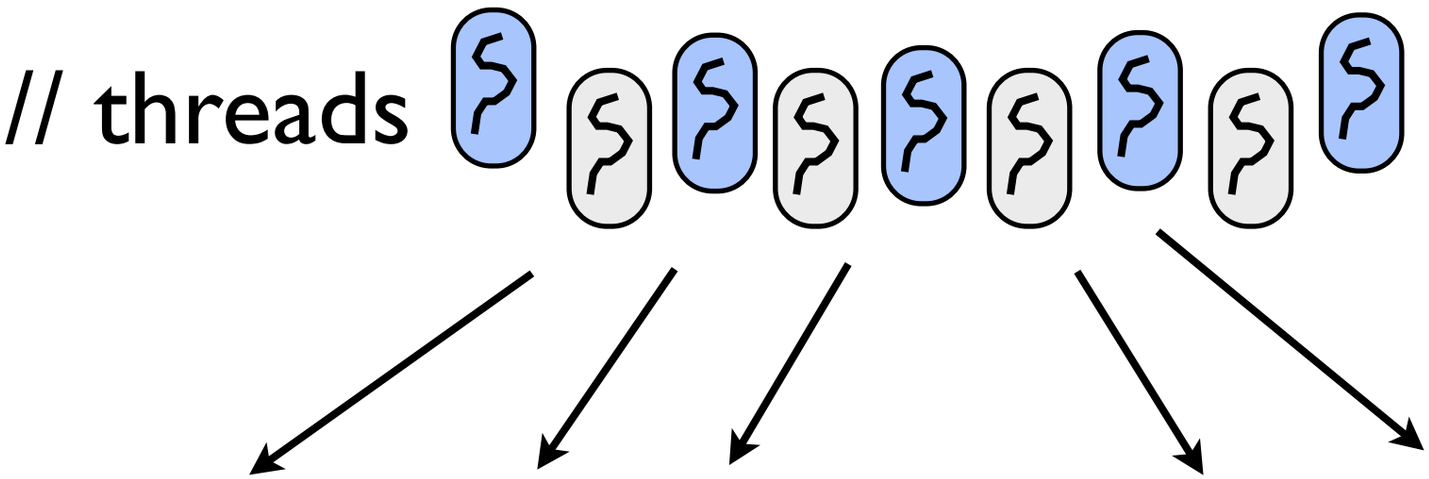
Hardware Cores

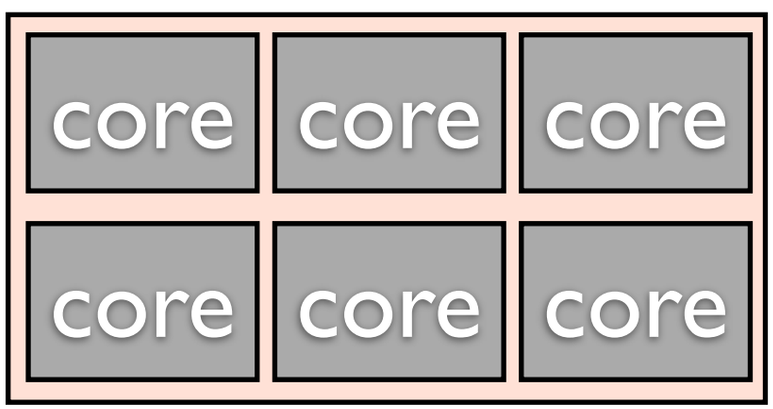


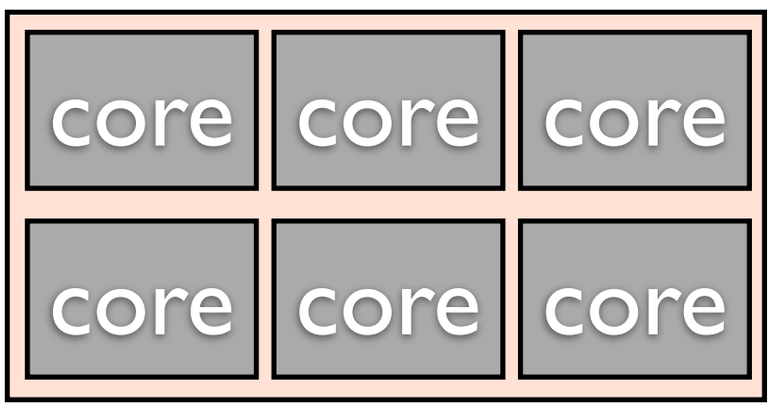
Deployment on Multicores



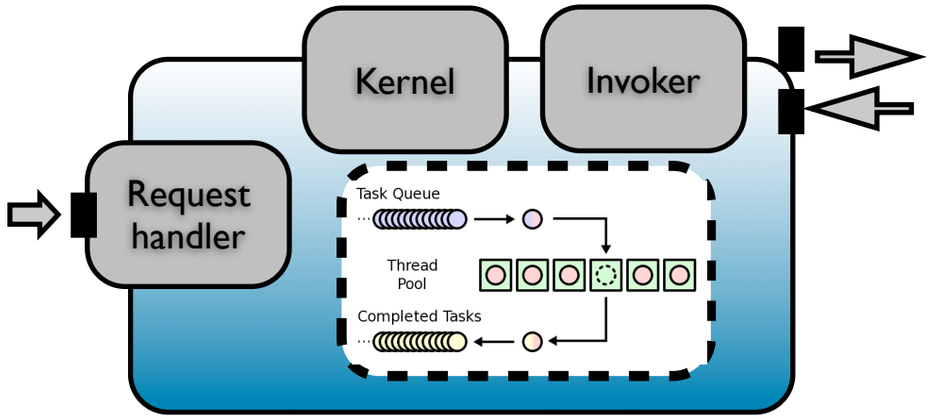
// threads





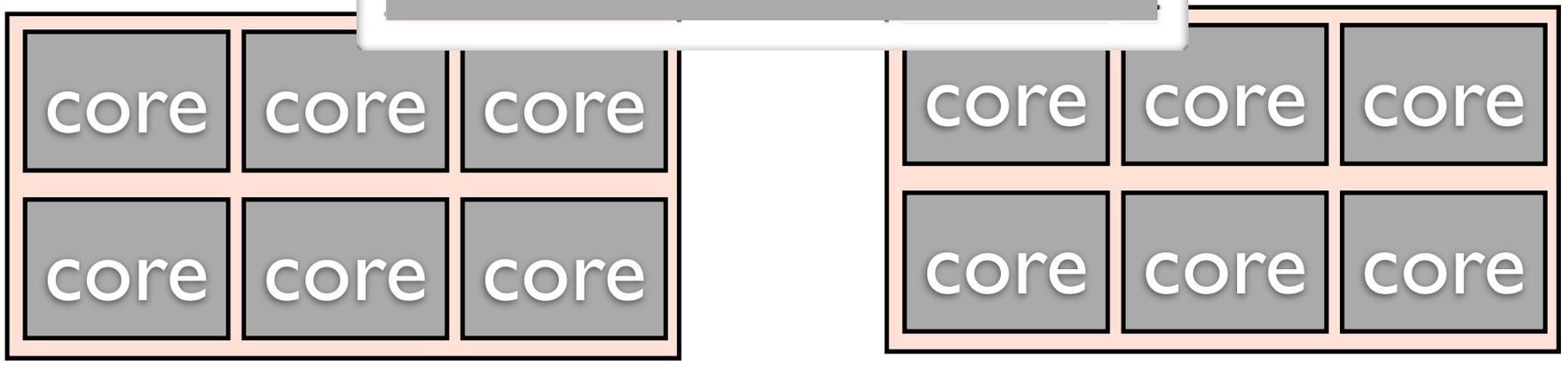


Deployment on Multicores

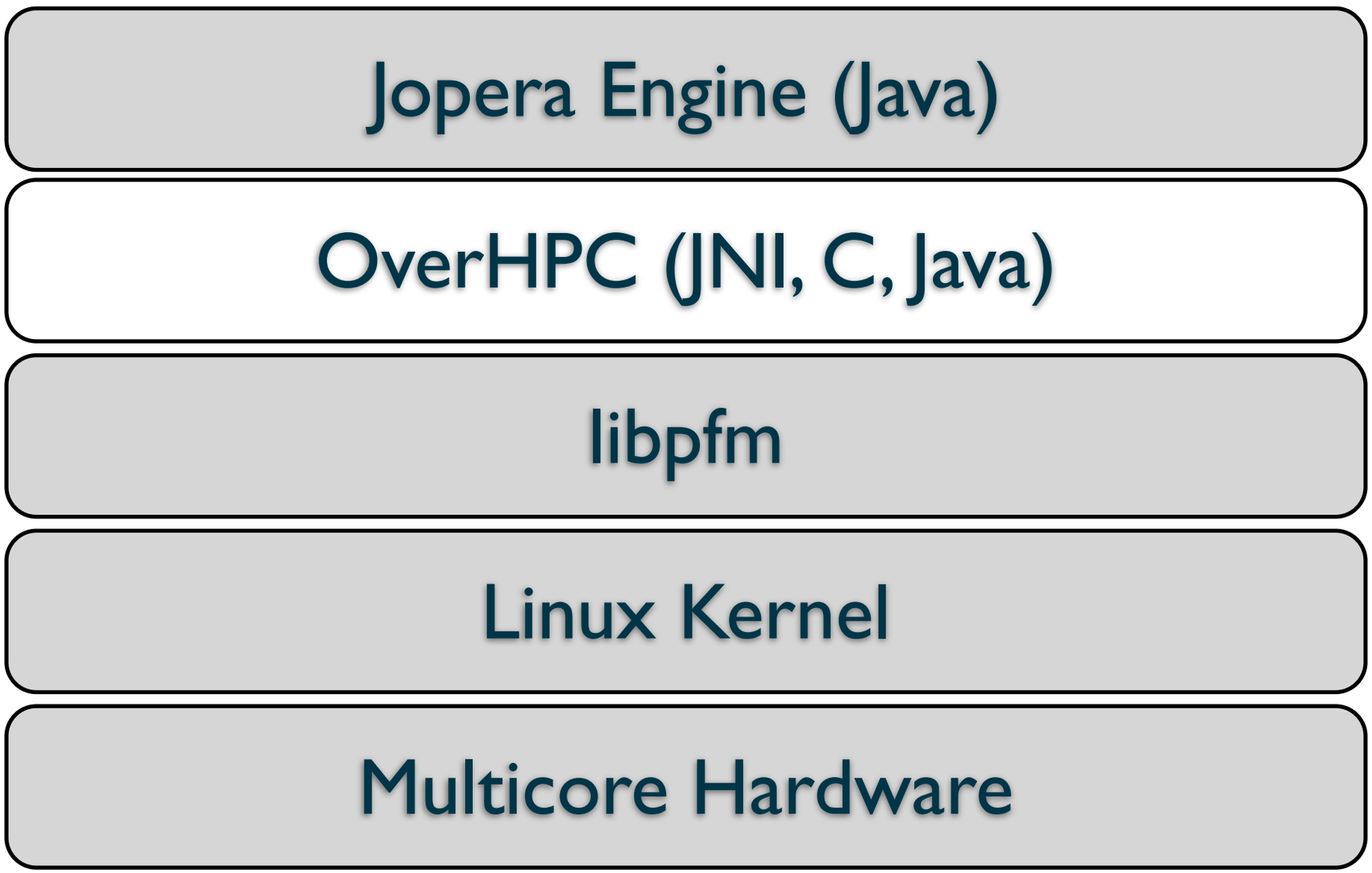


// threads

How?



OverHPC Library



OverHPC Library

1. Control and *Change* per-thread scheduling

2. *Measure* low level thread performance data

OverHPC Library API

1) Control and *Change* per-thread scheduling

Thread-Core Dynamic Affinity Binding

getThreadPID()
getThreadAffinity()
setThreadAffinity()
getAffinityInfo()

OverHPC Library API

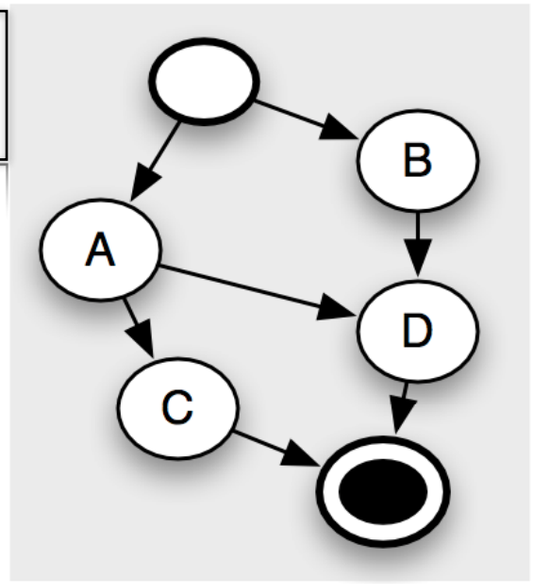
2) *Measure* low level thread performance:

Hardware Performance Counters

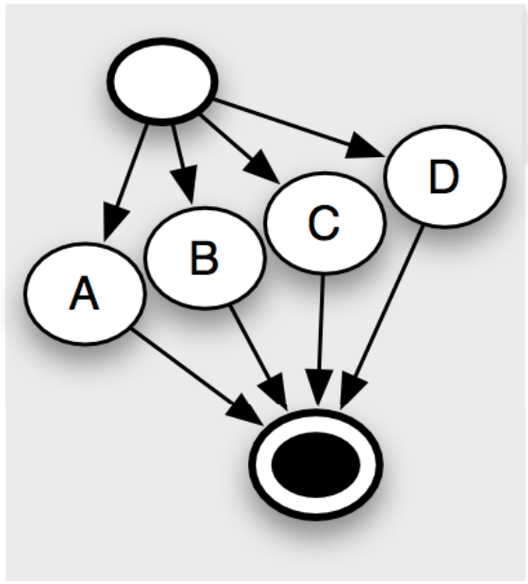
getEventsFromCache()
getEventsFromThread()
bindEventsToCore()
bindEventsToThread()

Evaluation

<flow>



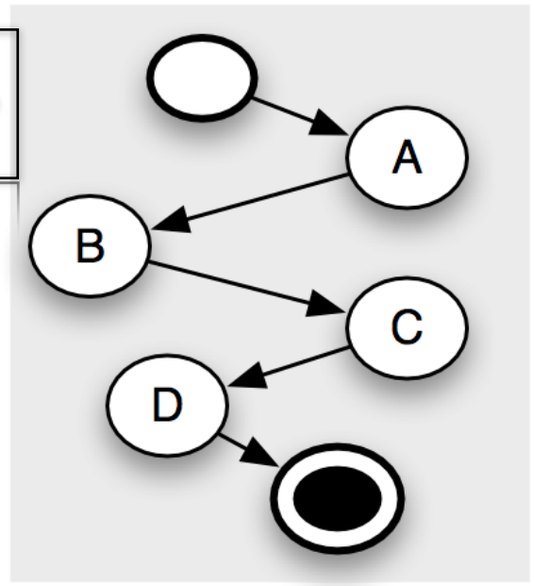
DAG



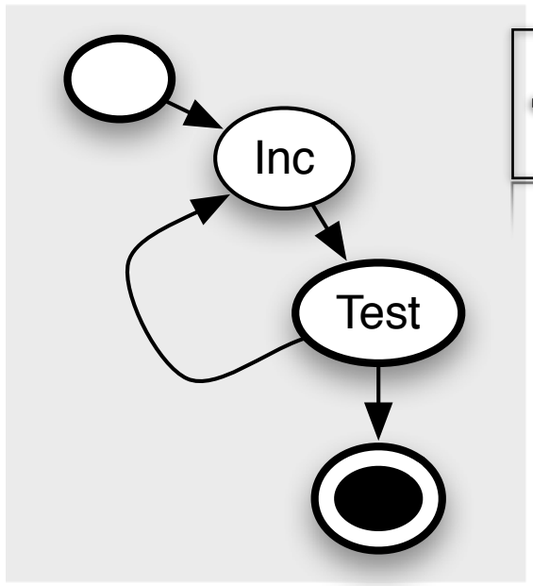
Parallel

<flow>

<sequence>



Sequential



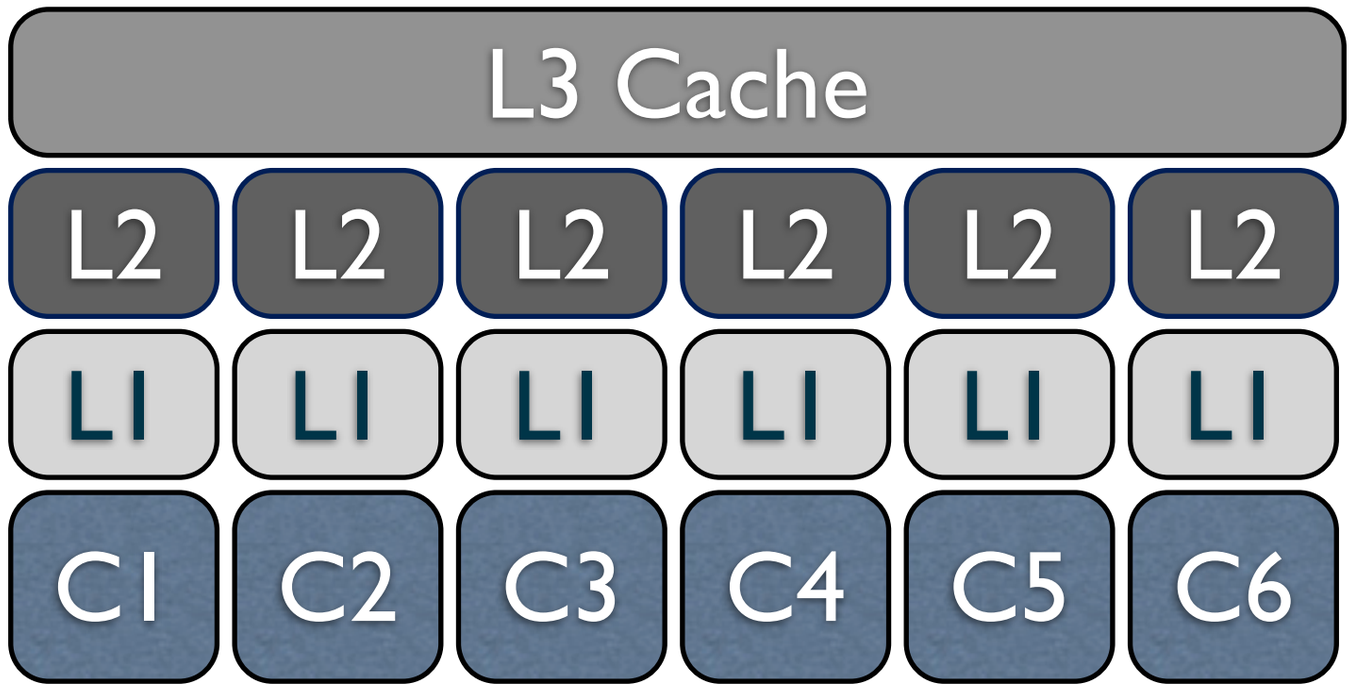
Loop

<while>

Hardware Setup

6 cores, 3 cache levels, 1 last level cache

2 x



Experimental Setup

Concurrent Business Process Instances

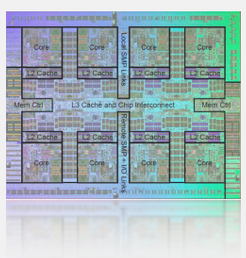
Jopera
Process Support for Web Services

Up to 30'000

OS Threads

k

Hardware Cores



12

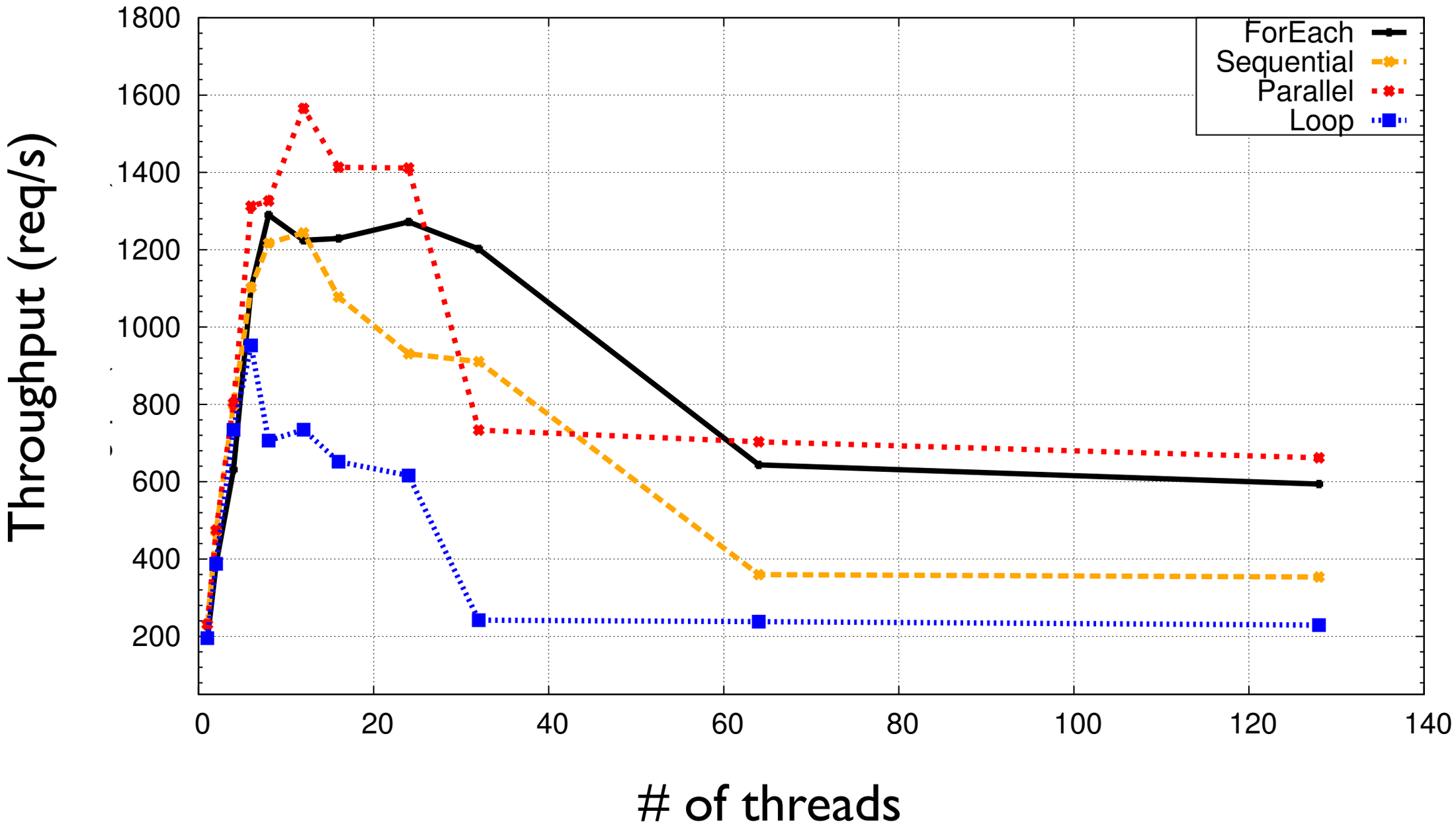
Thread-level Parallelism

How many *threads*?

Just *increase* the number of parallel concurrent threads in the pools for an increasing number of instances?

Thread-level Parallelism

Just increasing the number of threads...



Experimental Setup

Concurrent Business Process Instances

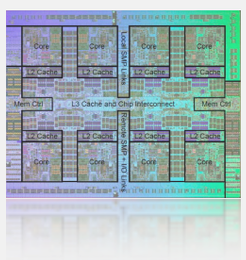
Jopera
Process Support for Web Services

Up to 30'000

OS Threads

24

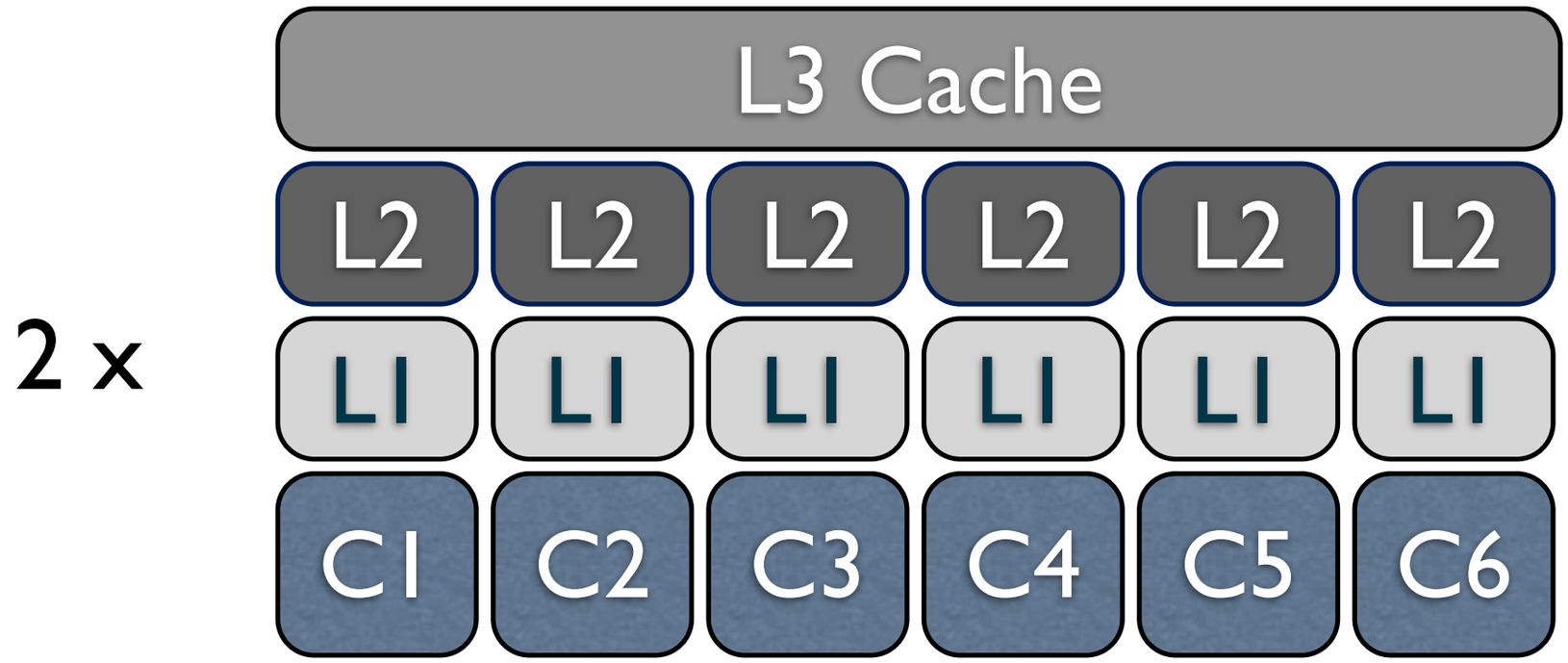
Hardware Cores



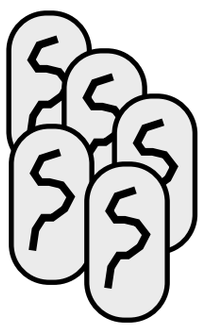
12

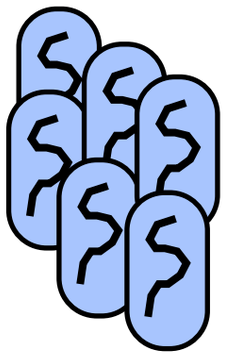
Experimental Setup

6 cores, 3 cache levels, 1 last level cache



2 Thread pools:

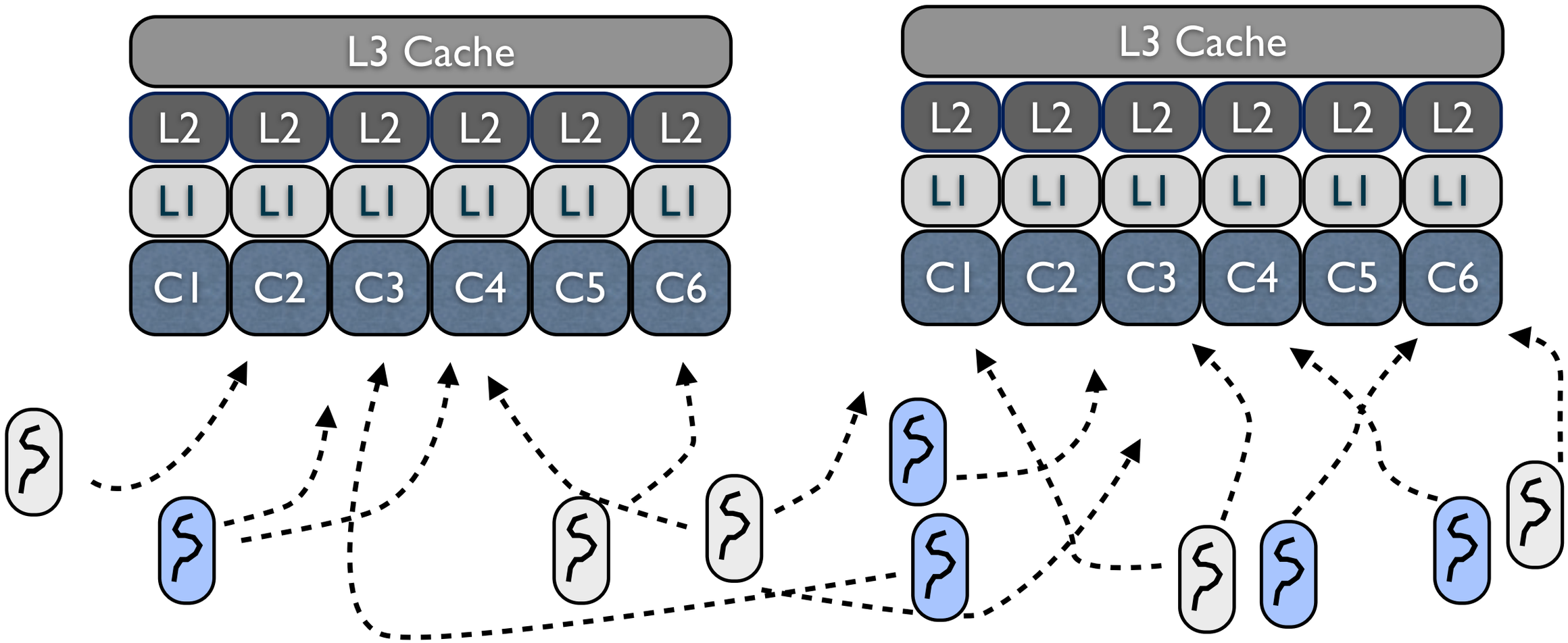
Kernel 

Invoker 

CPU Affinity Binding

Policy 1: Default

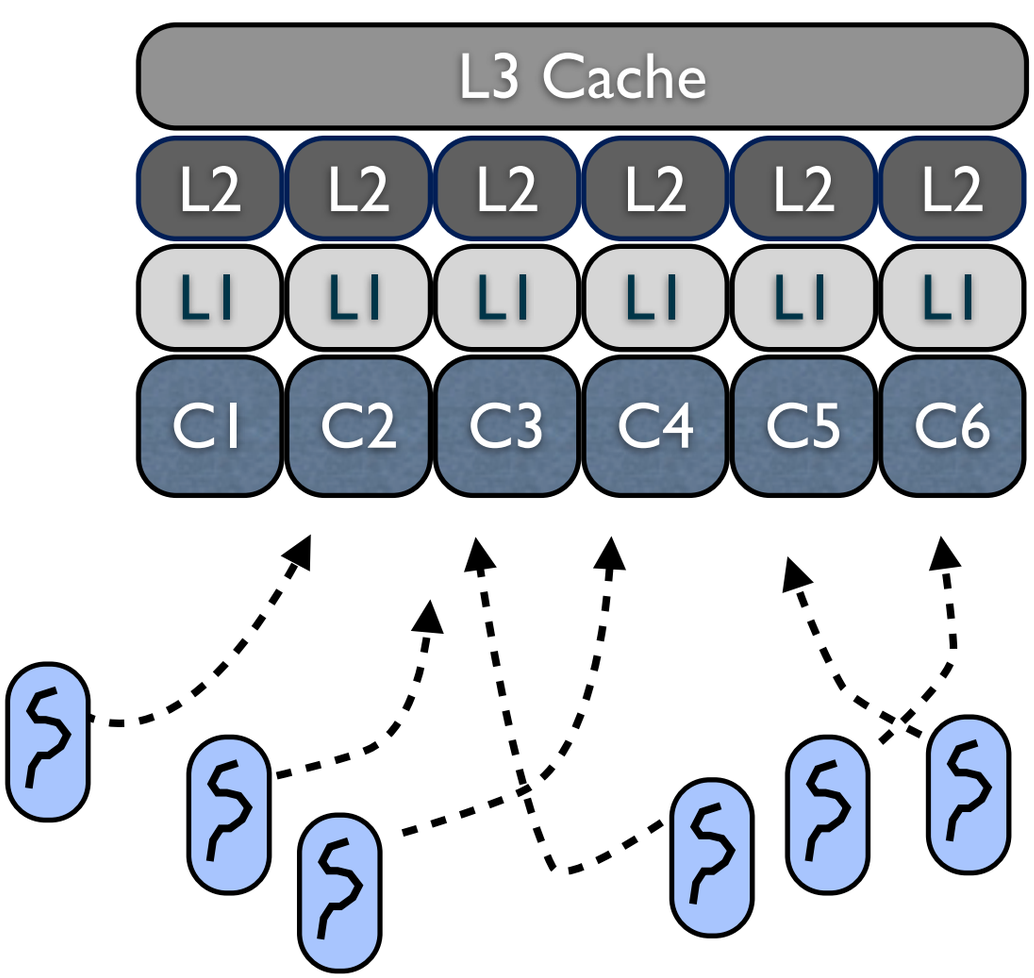
Unconstrained scheduling of threads by the OS

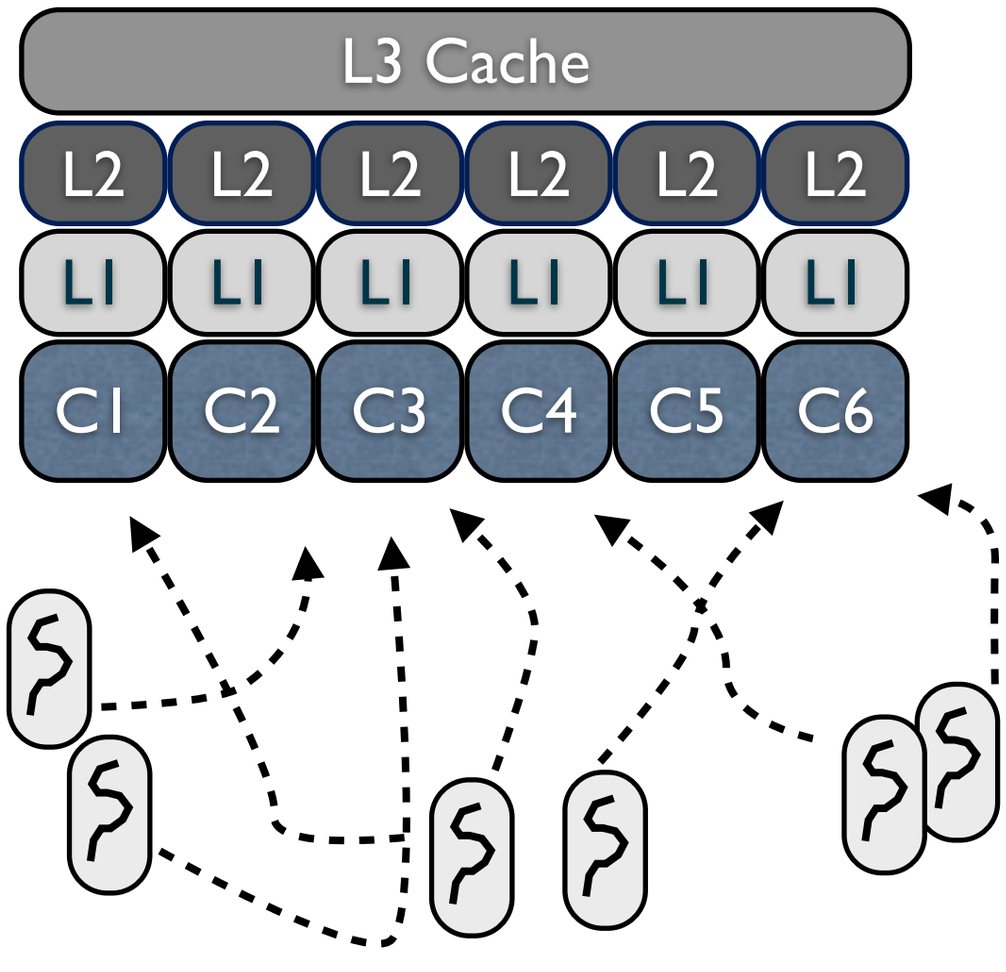


CPU Affinity Binding

Policy 2: per CPU

Constrain each thread pool within a CPU

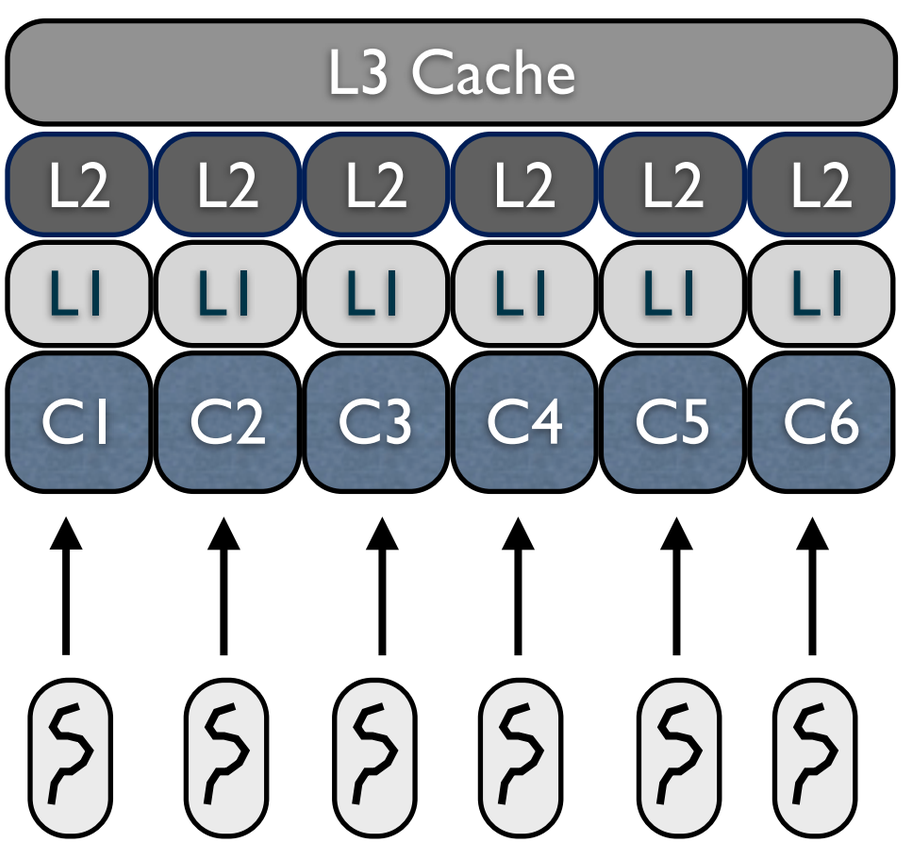


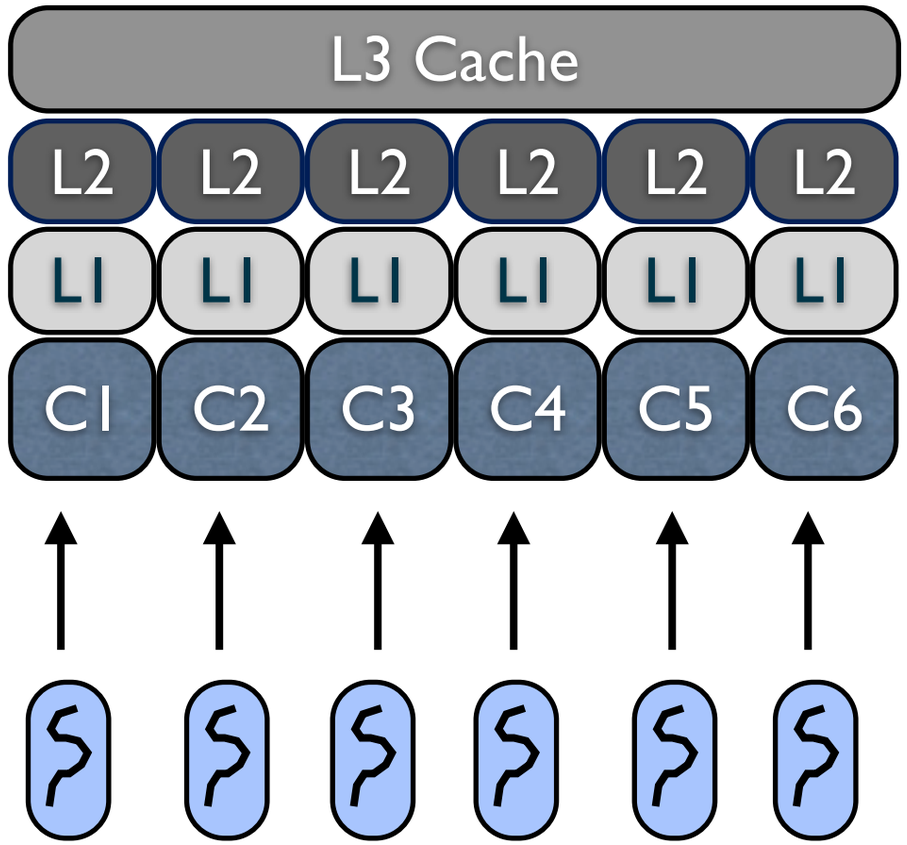


CPU Affinity Binding

Policy 3: per Core

Policy 2 + Constrain each thread on a specific core

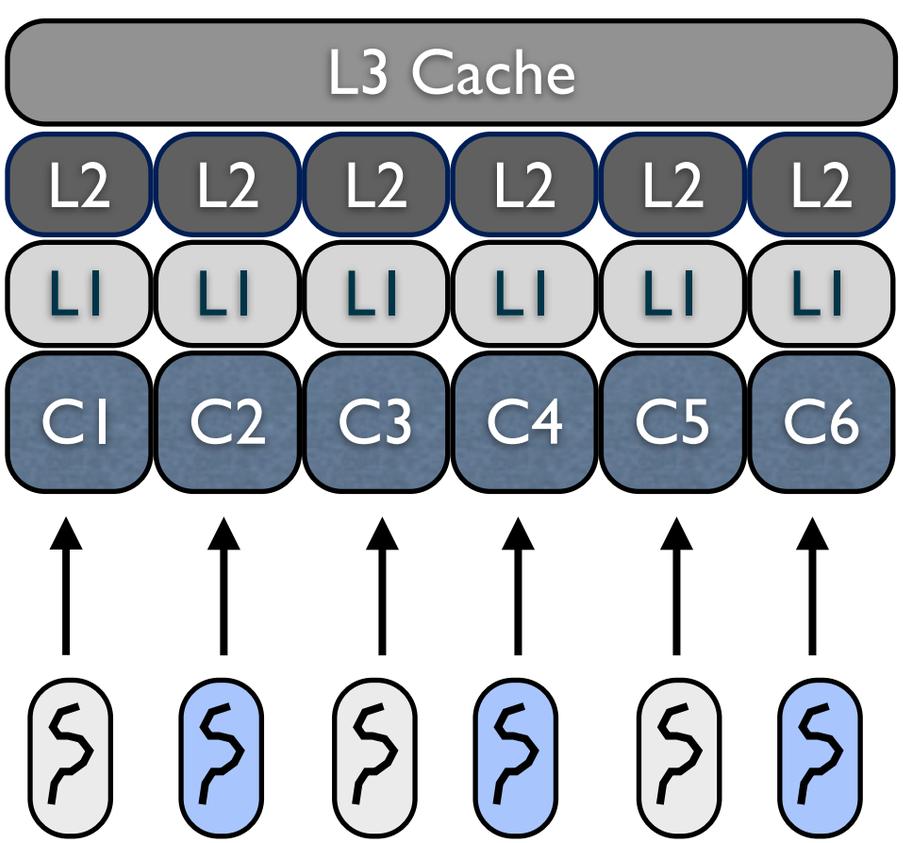


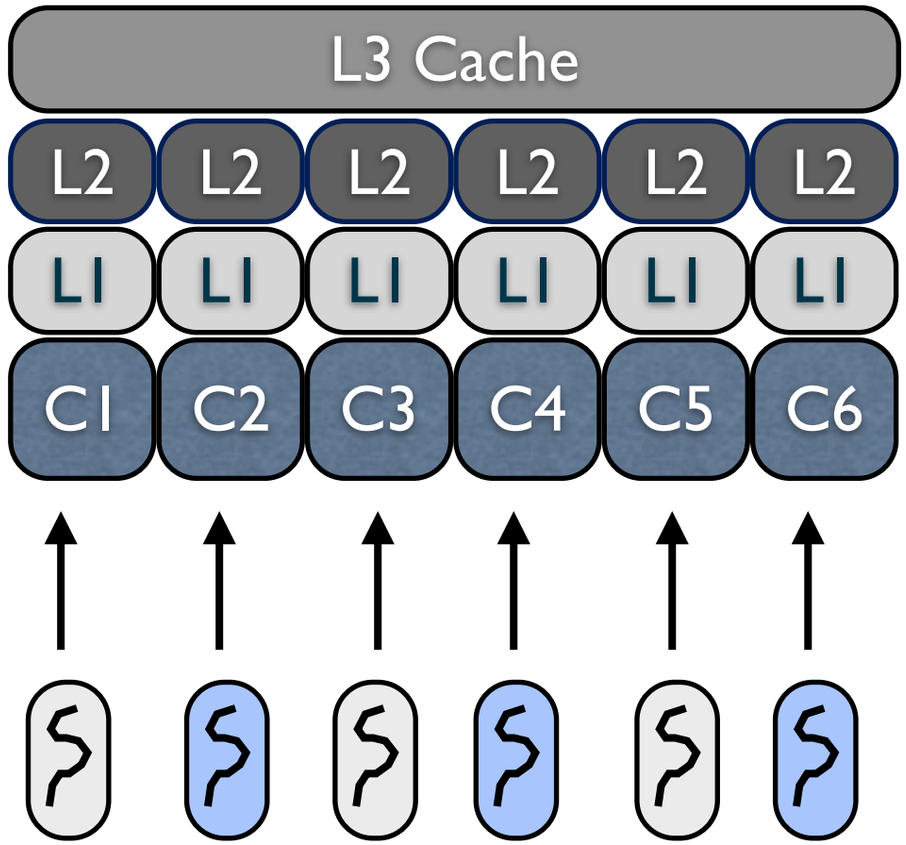


CPU Affinity Binding

Policy 4: Interleaved

Mix thread pools across CPUs





Experimental Setup

Concurrent Business Process Instances

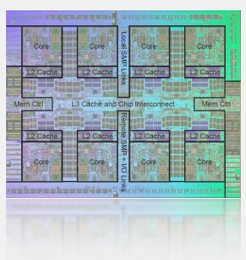
Jopera
Process Support for Web Services

Up to 30'000

OS Threads

24

Hardware Cores



12

Performance Layers

Concurrent Business Process Instances

5'000 - 30'000

Throughput, Walltime, ...

OS Threads

24

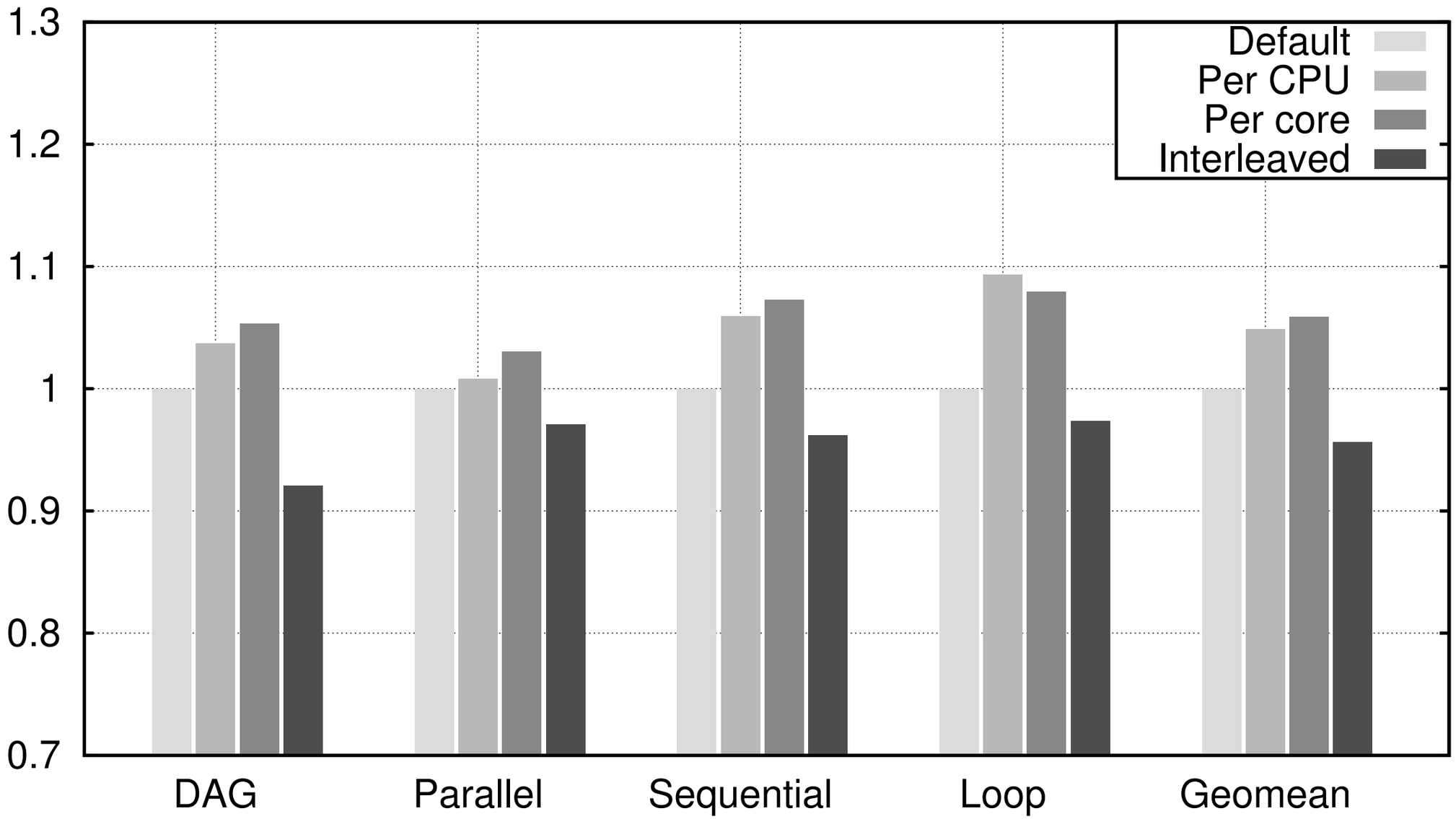
Hardware Performance Counters:
Cache miss, Thread Migrations, Context sw, ...

Hardware Cores

12

Experimental Results

Relative Speedup with 30k instances



2 x AMD Barcelona 6 cores processors with 2 LLC

HPC-Based Validation

Ineffective sw prefetches

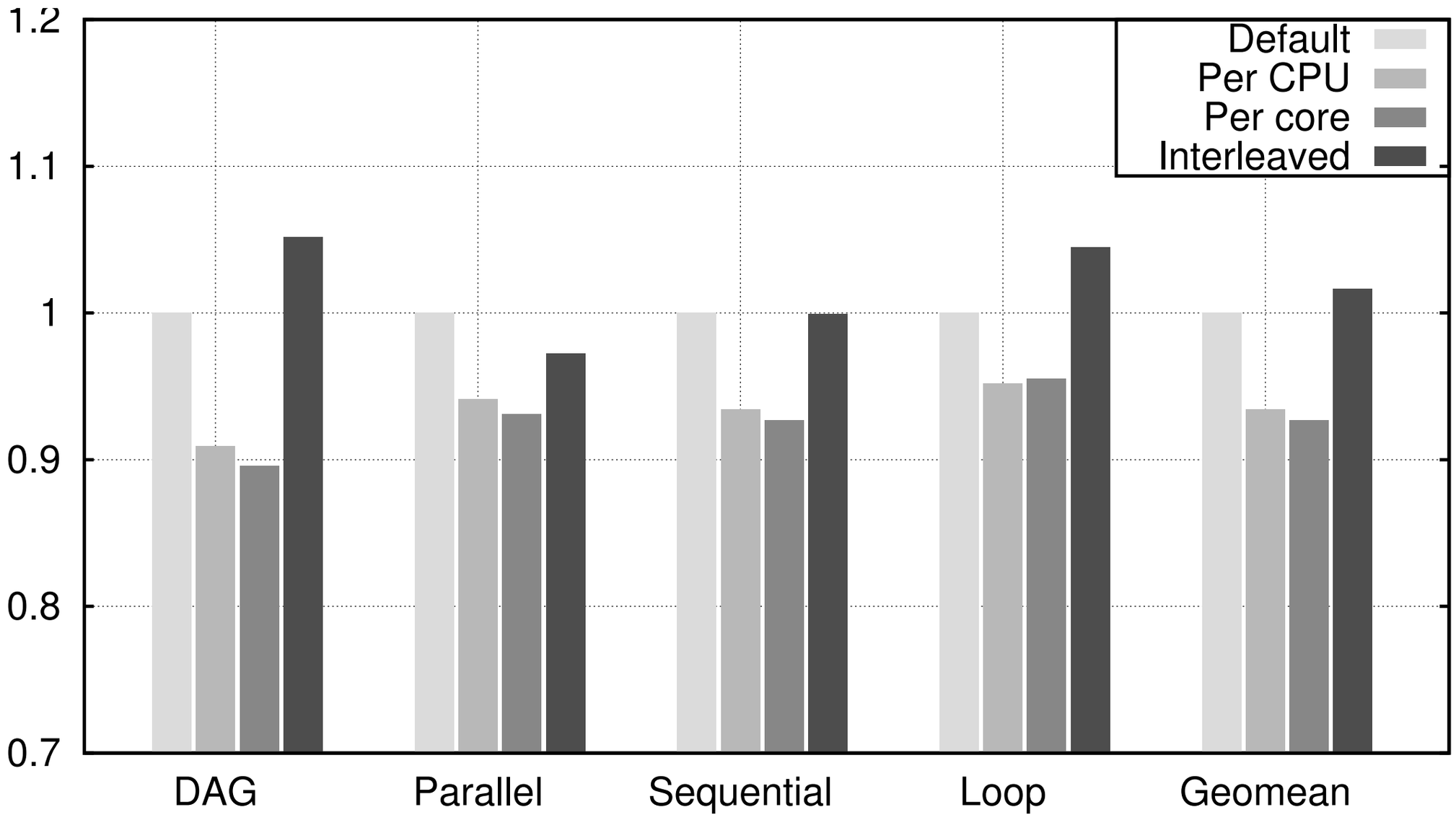
A prefetch request for a memory address already in the cache

L3 cache evictions

Data that needs to be stored in the cache is bigger than free available space

Experimental Results

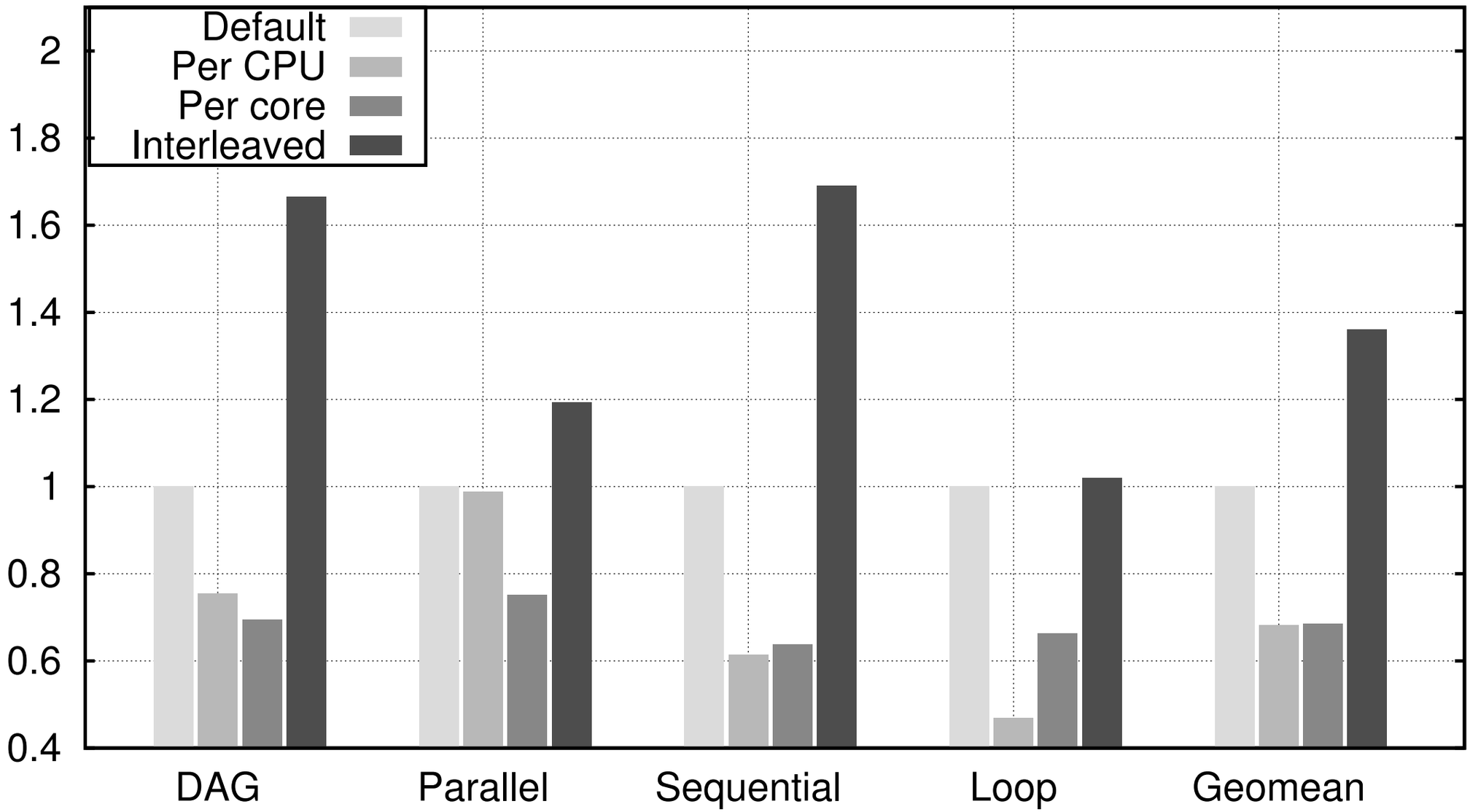
Ineffective SW prefetches



2 x AMD Barcelona 6 cores processors with 2 LLC

Experimental Results

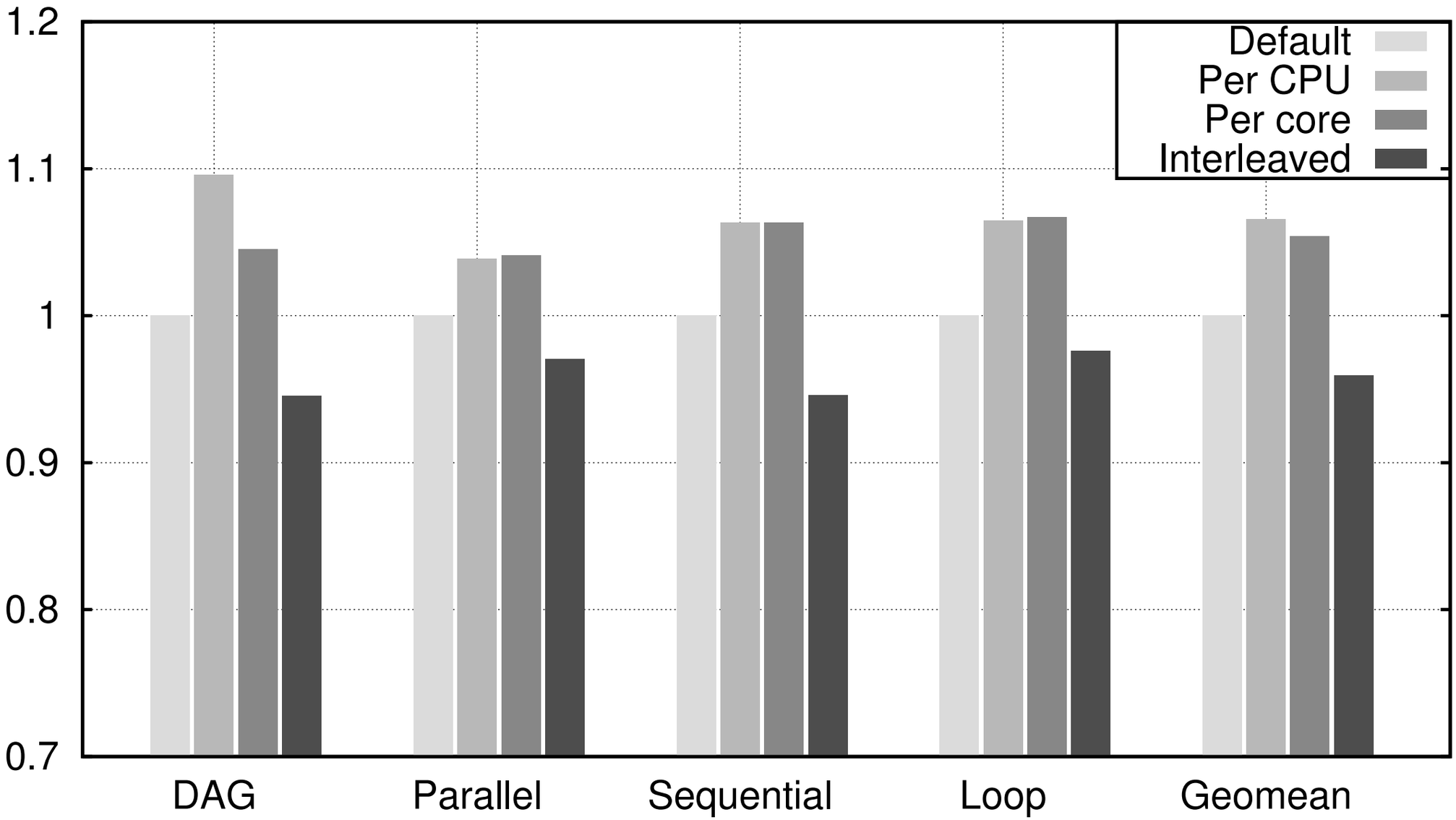
L3 Cache evictions



2 x AMD Barcelona 6 cores processors with 2 LLC

Experimental Results

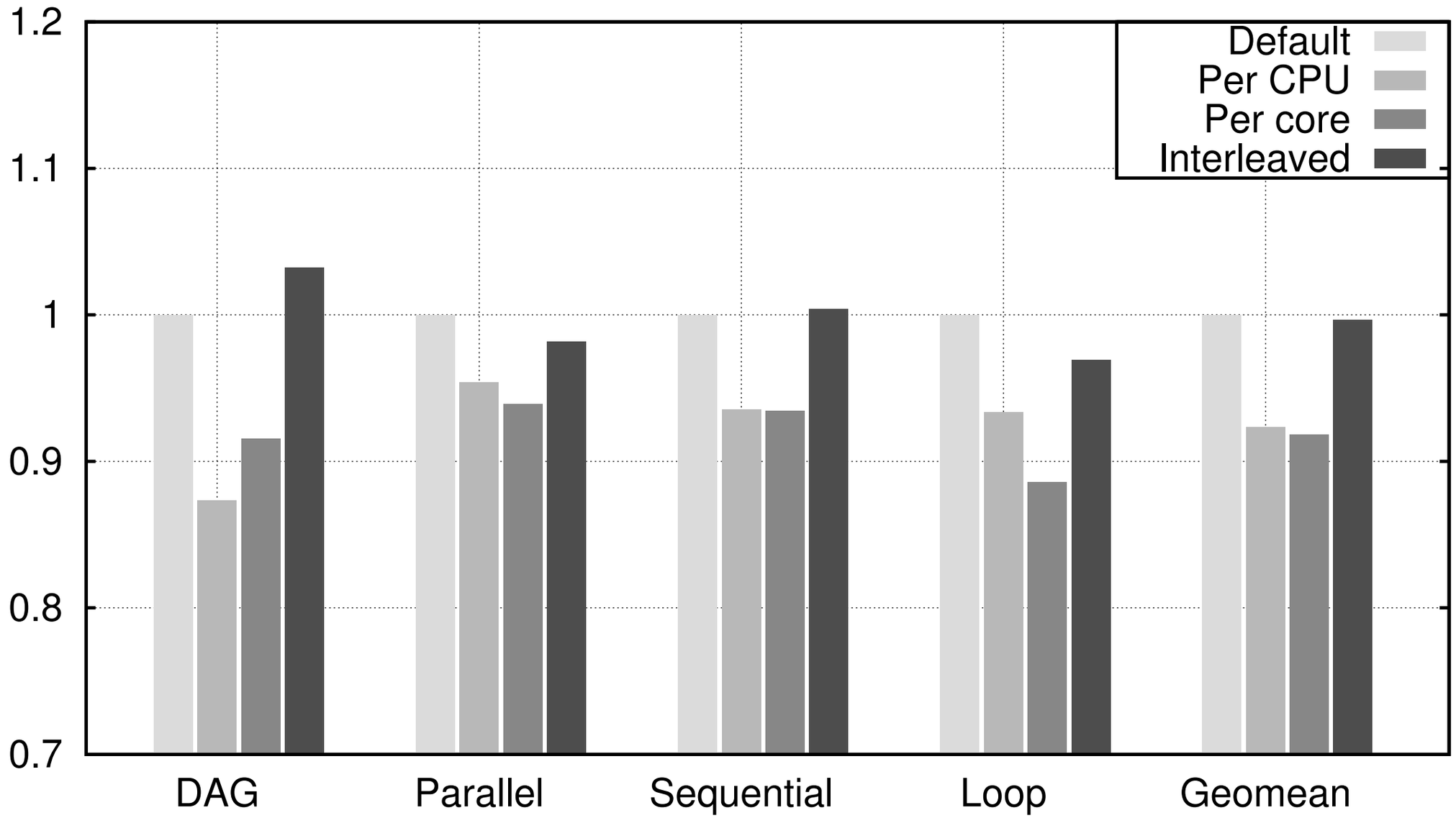
Relative Speedup with 5k instances



2 x AMD Barcelona 6 cores processors with 2 LLC

Experimental Results

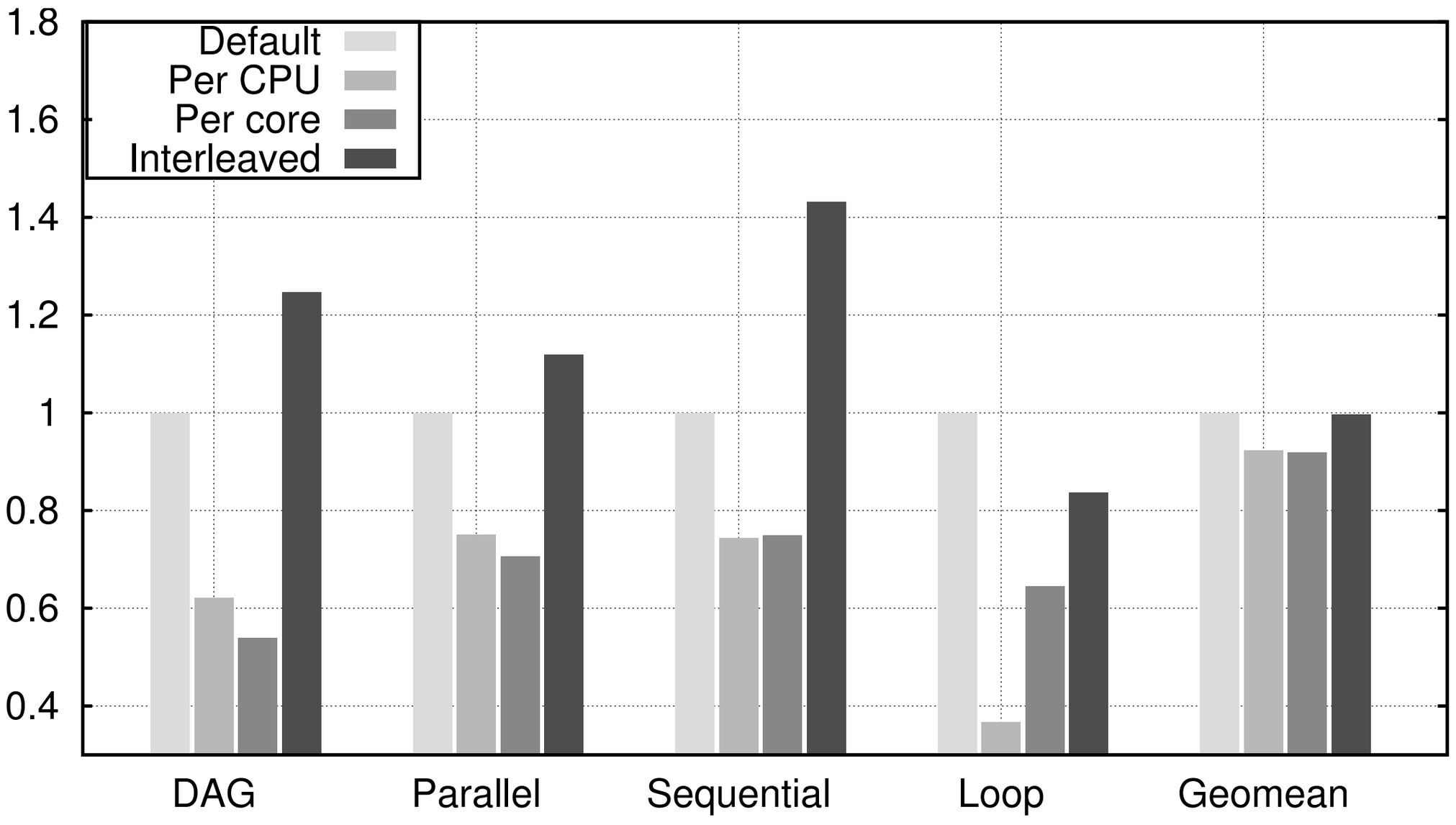
Ineffective SW prefetches



2 x AMD Barcelona 6 cores processors with 2 LLC

Experimental Results

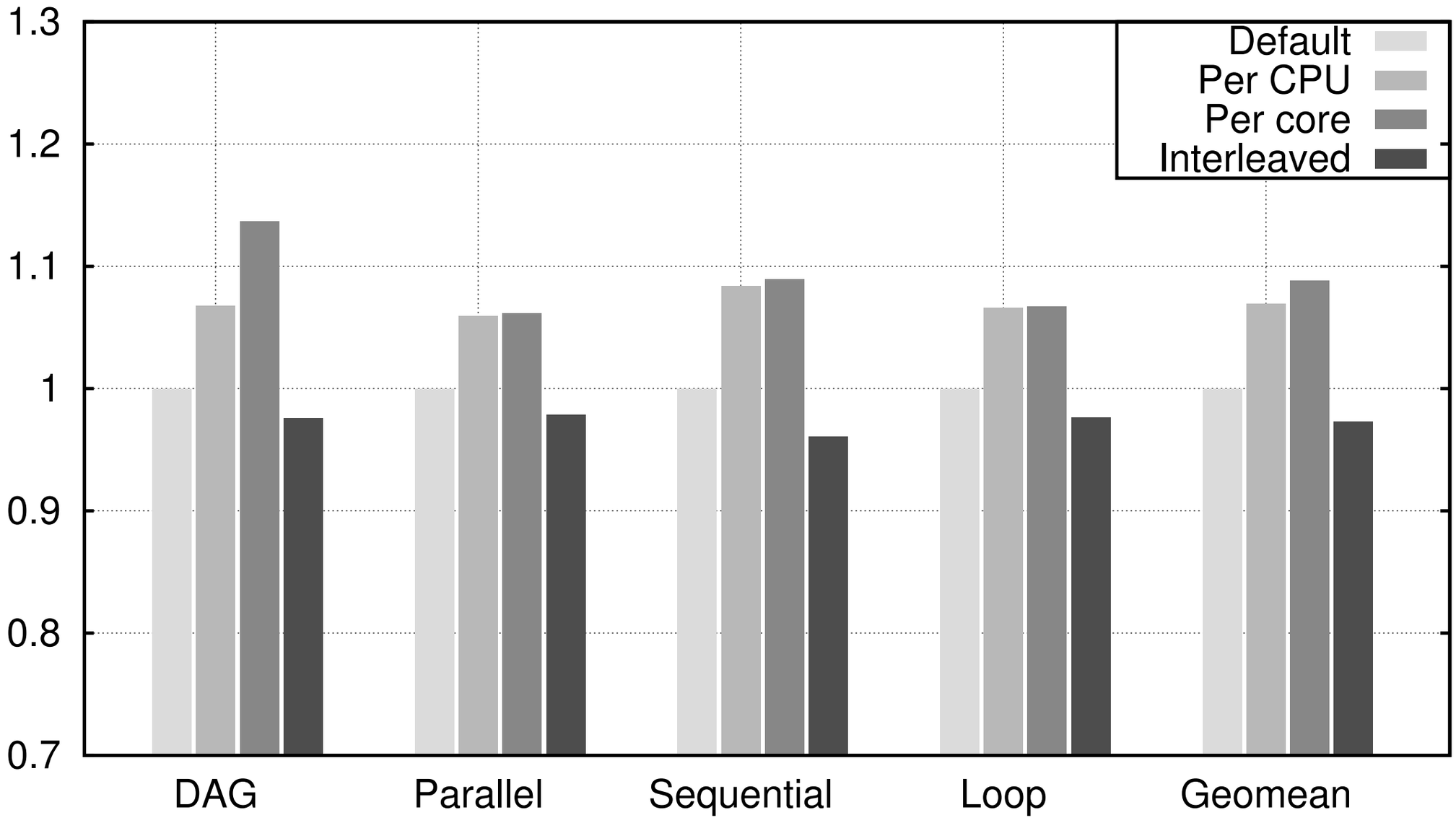
L3 Cache evictions



2 x AMD Barcelona 6 cores processors with 2 LLC

Experimental Results

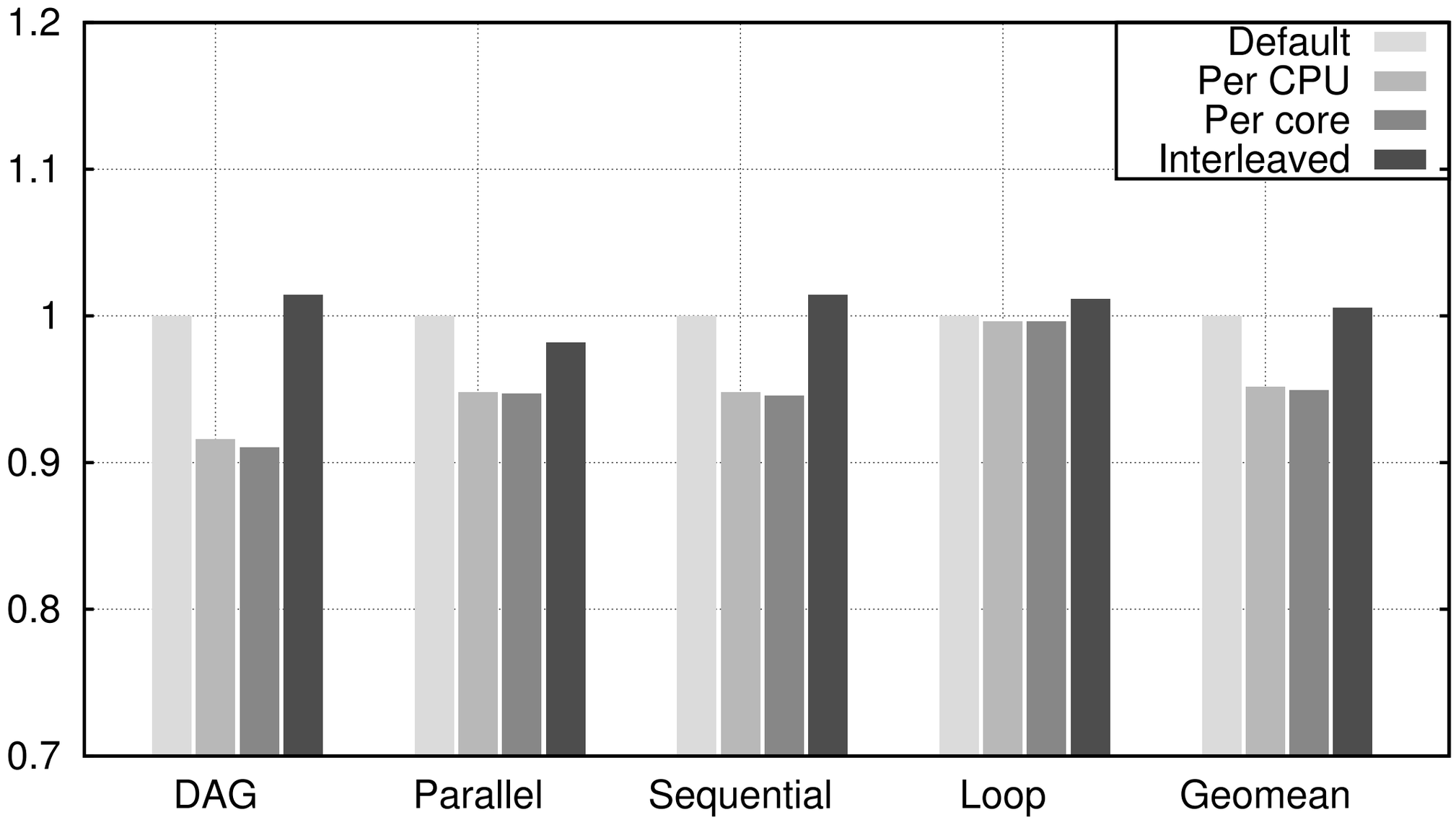
Relative Speedup with 10k instances



2 x AMD Barcelona 6 cores processors with 2 LLC

Experimental Results

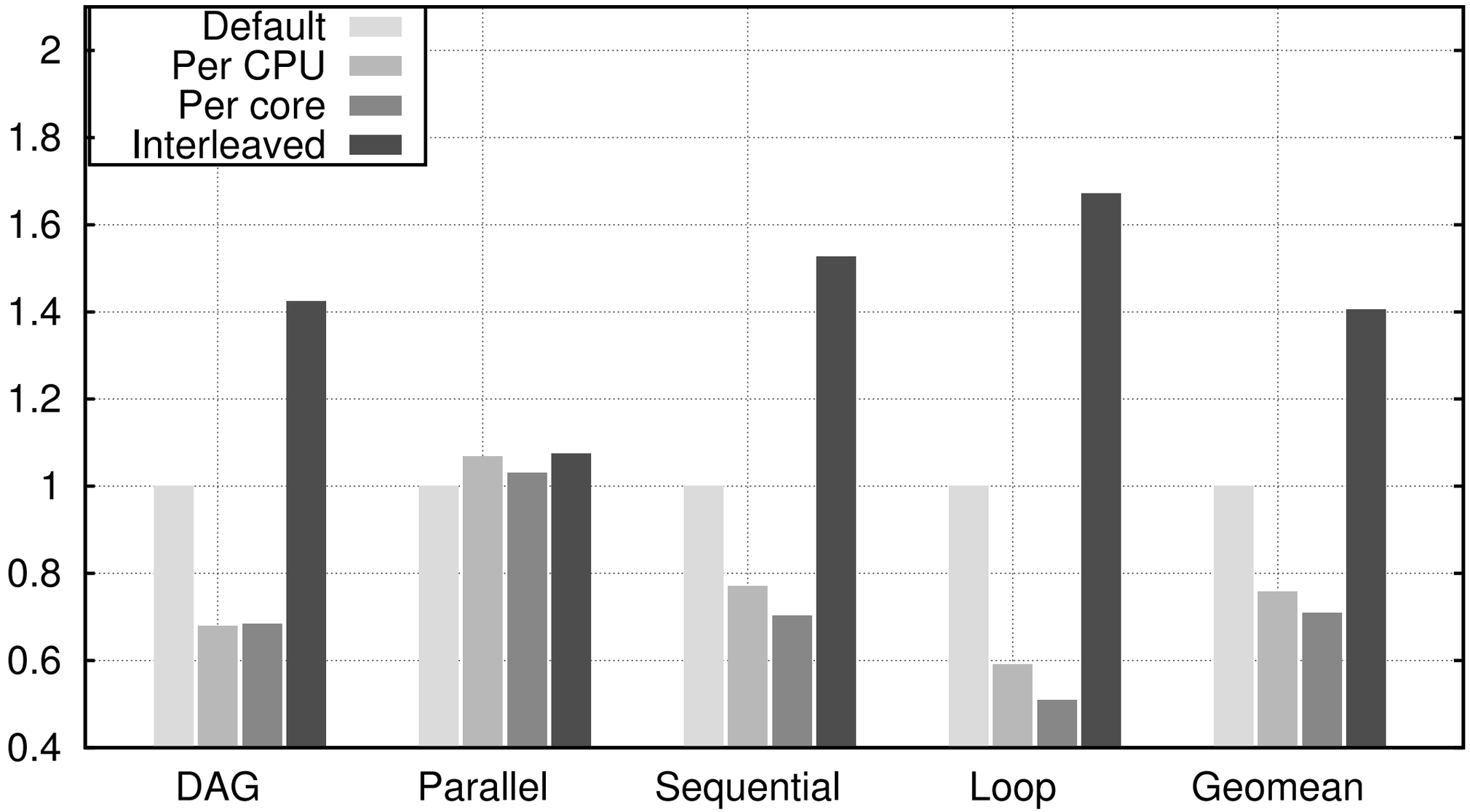
Ineffective SW prefetches



2 x AMD Barcelona 6 cores processors with 2 LLC

Experimental Results

L3 Cache evictions



2 x AMD Barcelona 6 cores processors with 2 LLC

Experimental Results

Correlation Coefficients (Hardware events - JOpera throughput)

Workload Size (Number of Instances)	Ineffective SW Pref	L3 Cache Evictions
5000	0.9842	0.9456
10000	0.9125	0.9883
30000	0.9661	0.9946

Conclusion

- Multicore machines offer powerful hardware parallelism, but what matters is not just the number of PEs
- The performance depends on *how* a limited amount of threads are mapped to the HW
- Multicore Aware Thread Scheduling significantly impacts the performance (up to 10% speedup)

Università della Svizzera italiana

Faculty of Informatics

Thank you!

Jopera
Process Support for Web Services

OverHPC Library:
http://sosoa.inf.usi.ch

JOpera business process execution engine:
http://www.jopera.org

Twitter:
@jopera_org

me:
daniele.bonetta@usi.ch