Mage: Online and Interference-Aware Scheduling for Multi-Scale Heterogeneous Systems

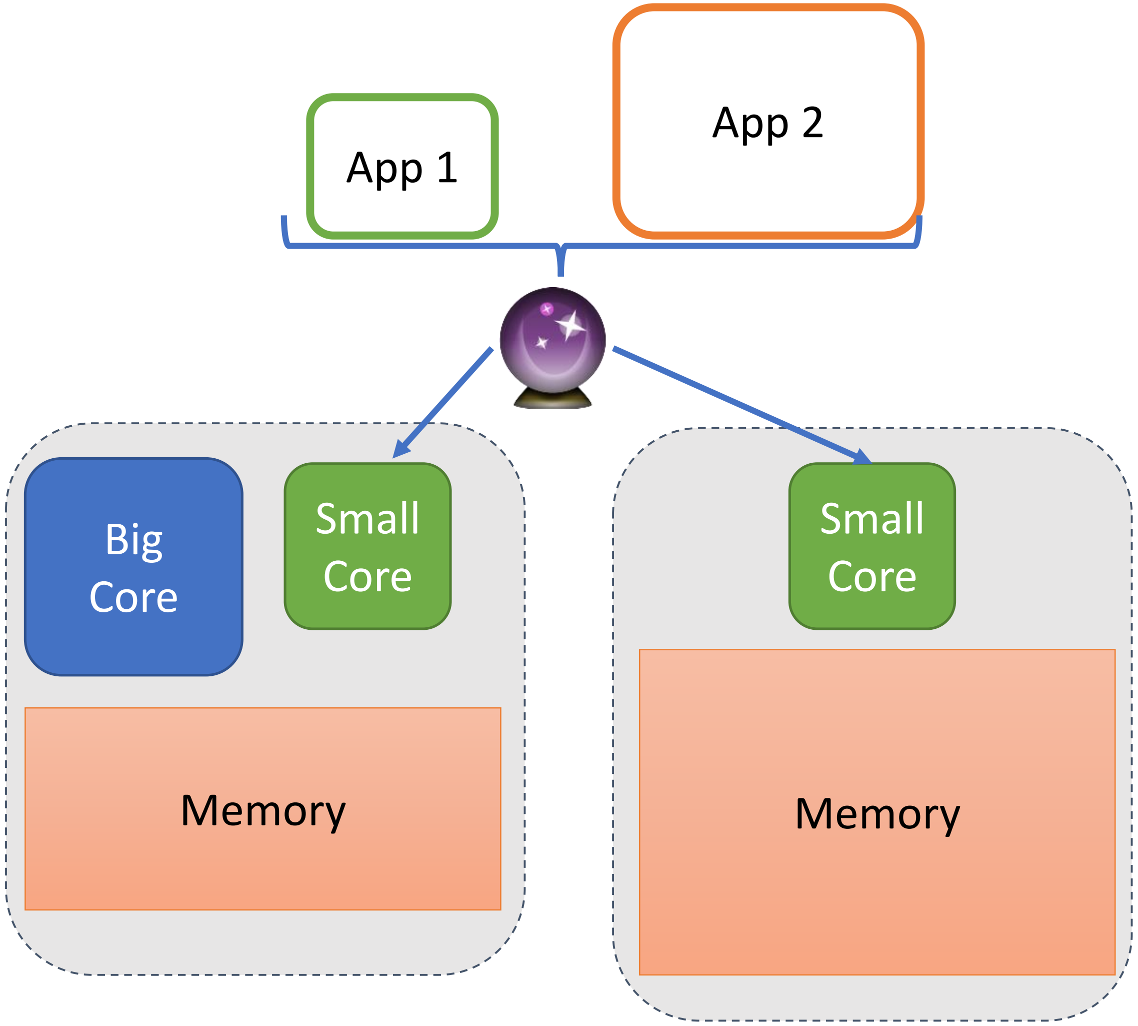
Francisco Romero[1] and Christina Delimitrou[2]

[1]Stanford University, [2]Cornell University

PACT – Session 4a – November 2, 2018

Motivation

- Heterogeneity is becoming more prevalent
  - Different server generations
  - Advanced management features, e.g., power management
- Allows for systems to better match applications to the underlying hardware
- **Challenge**: How do we maximize application performance *and* maintain high resource utilization?



Prior Work

System	Heterogeneous Clusters	Heterogeneous CMPs
Paragon	✓	✗
Whare-map	✓	✗
Bubble-flux	✓	✗
Composite cores	✗	✓
Hass	✗	✓
PIE	✗	✓

The Problem with "Sum of Schedulers"

Heterogeneous **Cluster** Scheduler → Heterogeneous **CMP** Scheduler

- Suboptimal performance
- Revisit several scheduling decisions

Need a data-driven approach to avoid exhaustive search

Heterogeneous Cluster + CMP Scheduler

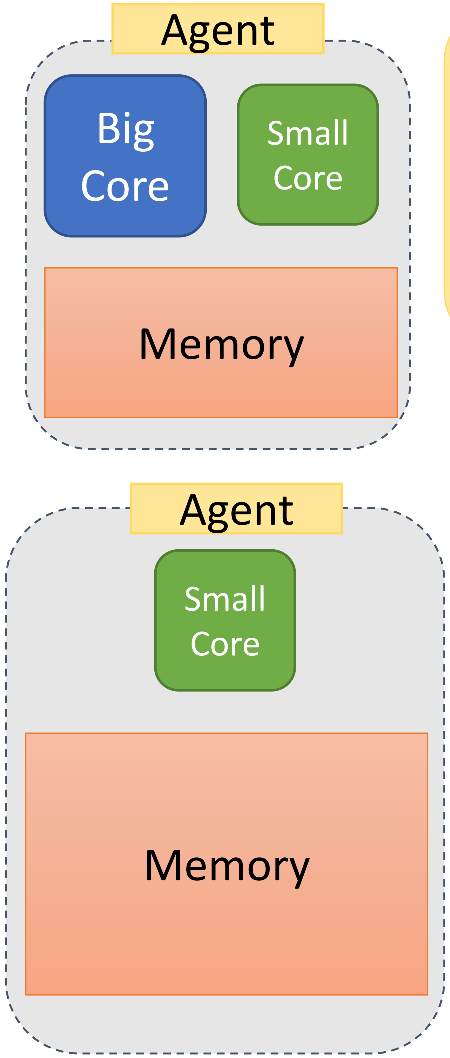
- High overhead
- Not scalable

Mage

- Tiered runtime scheduler that considers inter- and intra-server heterogeneity jointly
- Leverages fast and online data mining to quickly explore the space of application placements
- Lightweight application monitoring and rescheduling
- Heterogeneous CMPs: **38% average improvement** compared to a greedy scheduler
- Heterogeneous Cluster: **30% average improvement** compared to a greedy scheduler and **11% average improvement** compared to a heterogeneity- and interference-aware scheduler

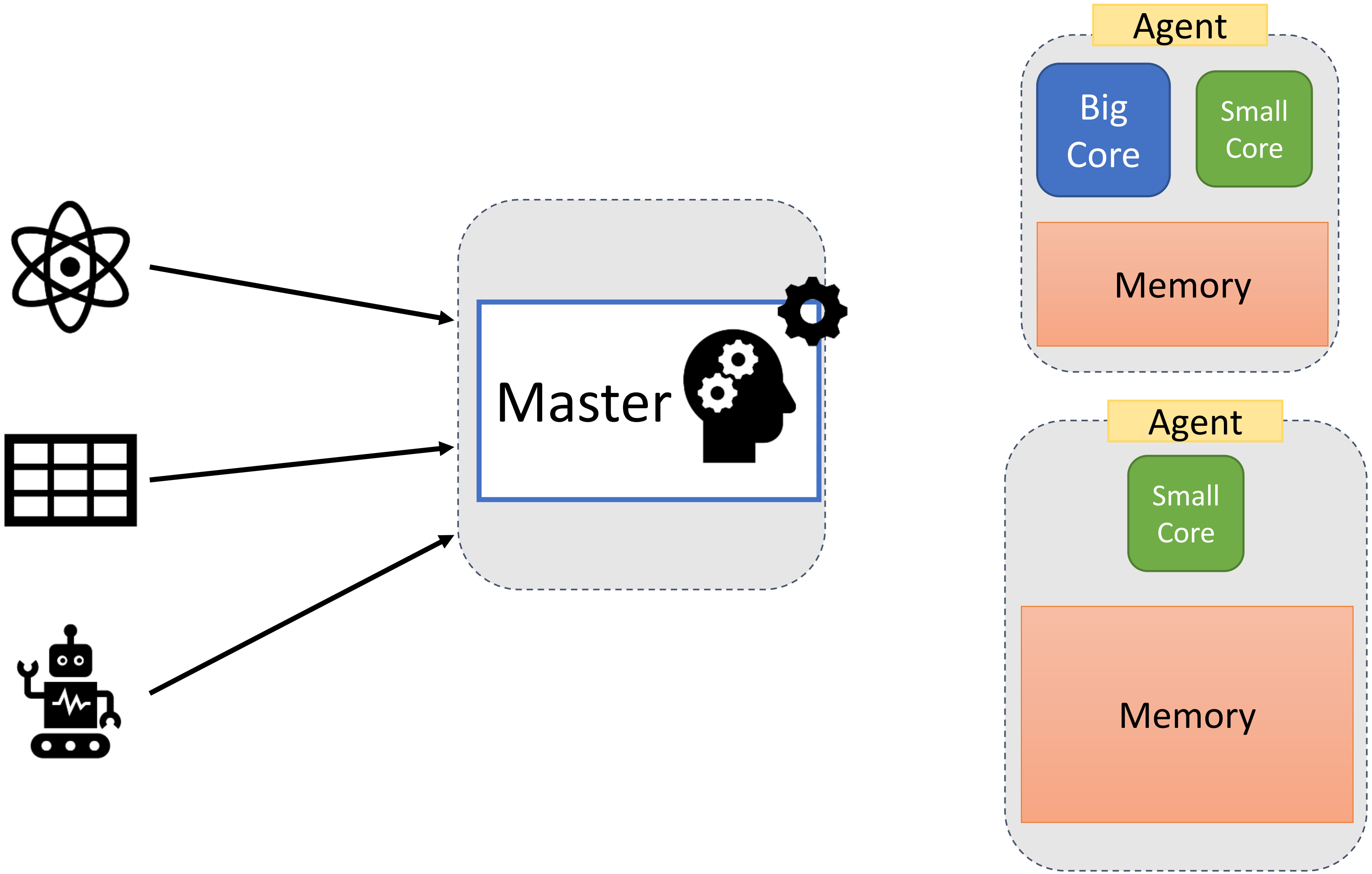
Mage Master and Mage Agents

Mage Master
- Runs inference
- Makes optimal application-to-*resource* scheduling decision
- Decides when applications should be migrated/rescheduled



Mage Agent
- Monitor the performance of all scheduled applications
- Notify the master when QoS violations occur

Application Arrival and Initial Scheduling



What we want

Application-to-*Resource*

How can Mage *quickly* and *accurately* generate this matrix?

Applications

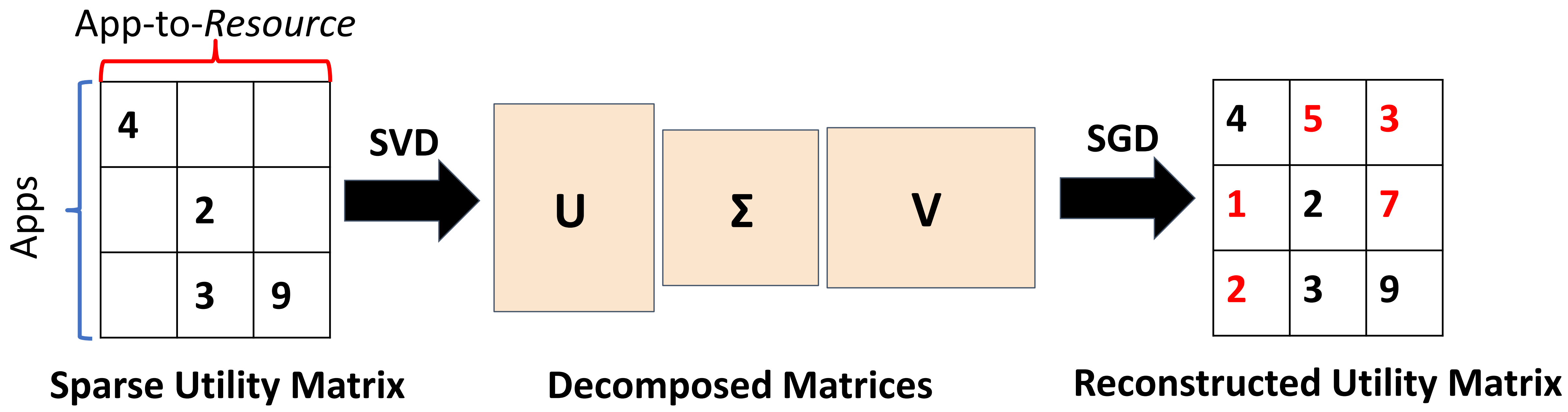
App3	$MIPS_{3,1}$	$MIPS_{3,2}$	…	$MIPS_{3,6}$

✓ Heterogeneous resources that benefit an application

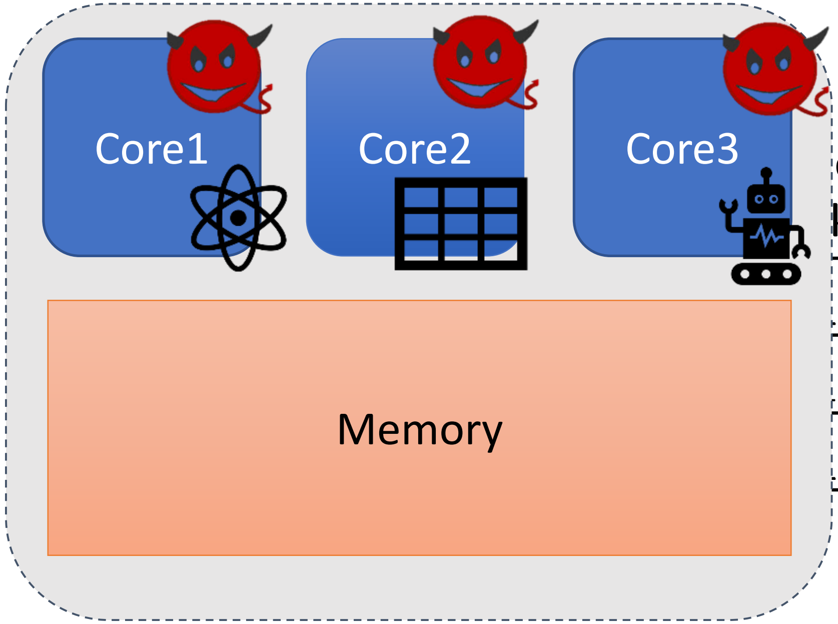
✓ Performance impact of co-scheduling applications

Collaborative Filtering

- Use Single Value Decomposition (SVD) with PQ-Reconstruction (SGD) to uncover:
  - Heterogeneous resources that benefit individual applications
  - Interference that can be tolerated between applications



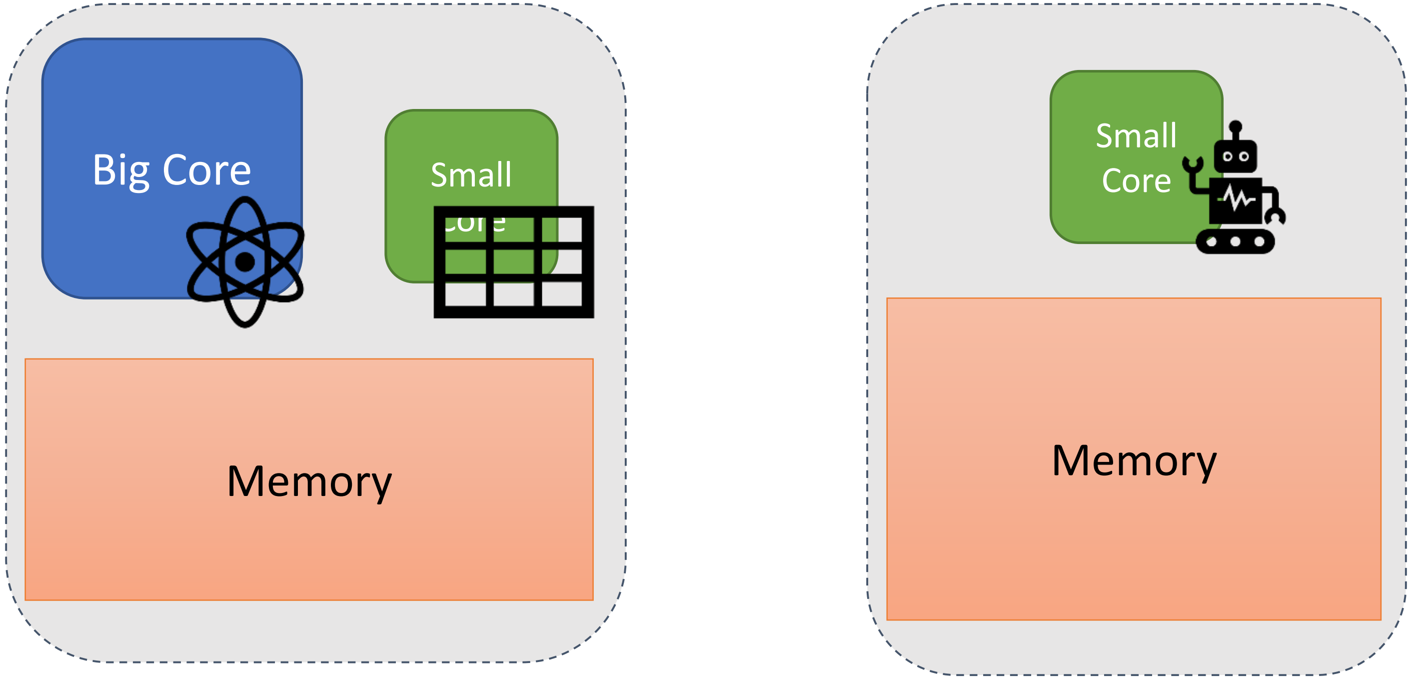
Contentious Kernel Profiling



el 1 k]	Cont. Kernel 2 [CPU]	...	Cont. Kernel *n* [Cache]
	$MIPS_{1,2}$	...	$MIPS_{1,n}$
	$MIPS_{2,2}$	...	$MIPS_{2,n}$
	$MIPS_{3,2}$	...	$MIPS_{3,n}$

Common reference point for the sensitivity of new applications to interference of shared resources

Co-Scheduling Sensitivity



Co-Scheduling Sensitivity

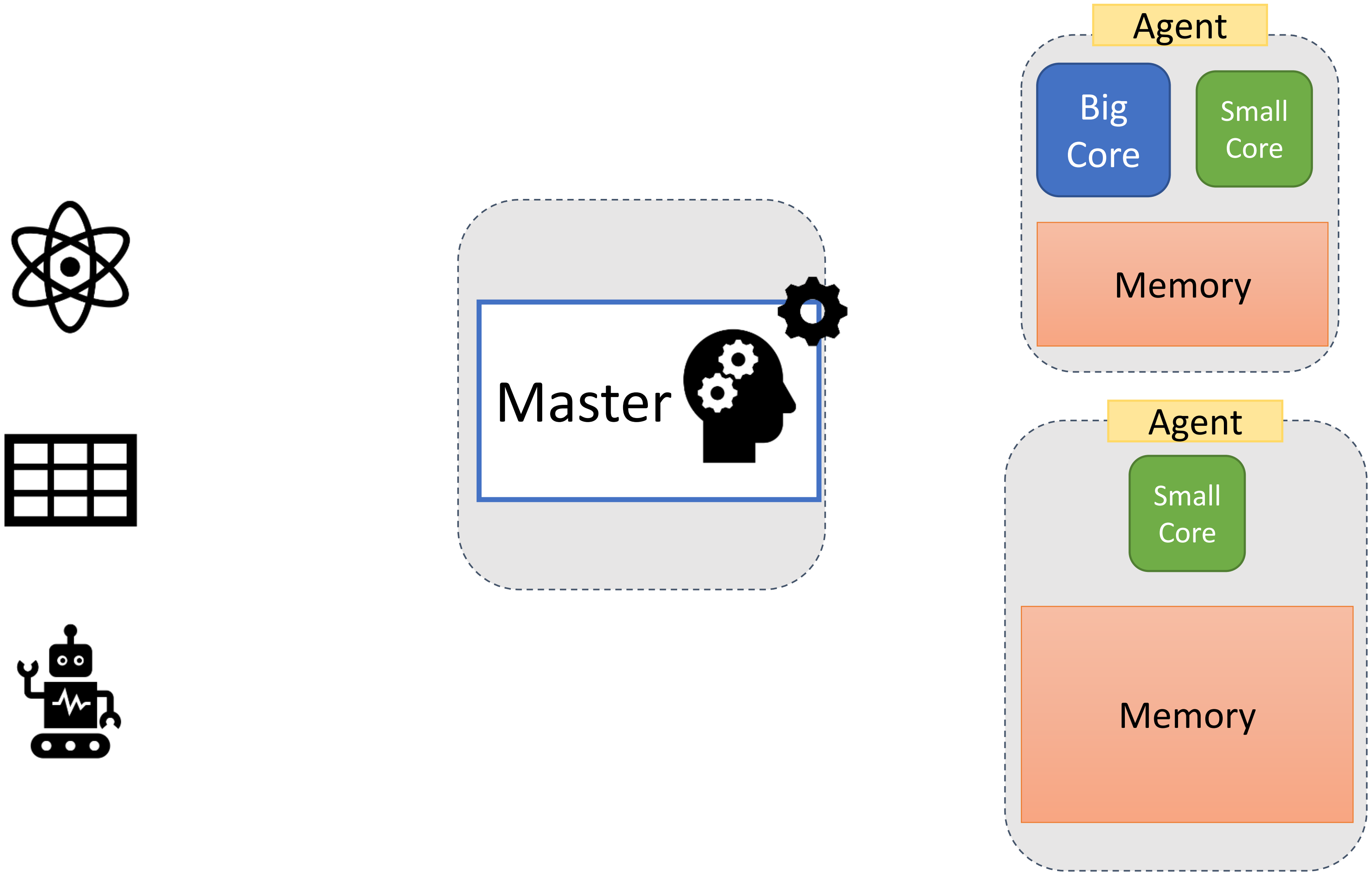
	App1:Core1 App2:Core2 App3:Core3	App1:Core1 App2:Core3 App3:Core2	App1:Core2 App2:Core1 App3:Core3	App1:Core2 App2:Core3 App3:Core1	App1:Core3 App2:Core1 App3:Core2	App1:Core3 App2:Core2 App3:Core1
App1	$MIPS_{1,1}$	$MIPS_{1,2}$	?	?	?	?
App2	$MIPS_{2,1}$	?	?	?	?	$MIPS_{2,6}$
App3	$MIPS_{3,1}$	?	$MIPS_{3,3}$	?	?	?

Co-Scheduling Sensitivity

	App1:Core1 App2:Core2 App3:Core3	App1:Core1 App2:Core3 App3:Core2	App1:Core2 App2:Core1 App3:Core3	App1:Core2 App2:Core3 App3:Core1	App1:Core3 App2:Core1 App3:Core2	App1:Core3 App2:Core2 App3:Core1
App1	$MIPS_{1,1}$	$MIPS_{1,2}$	$MIPS_{1,3}$	$MIPS_{1,4}$	$MIPS_{1,5}$	$MIPS_{1,6}$
App2	$MIPS_{2,1}$	$MIPS_{2,2}$	$MIPS_{2,3}$	$MIPS_{2,4}$	$MIPS_{2,5}$	$MIPS_{2,6}$
App3	$MIPS_{3,1}$	$MIPS_{3,2}$	$MIPS_{3,3}$	$MIPS_{3,4}$	$MIPS_{3,5}$	$MIPS_{3,6}$

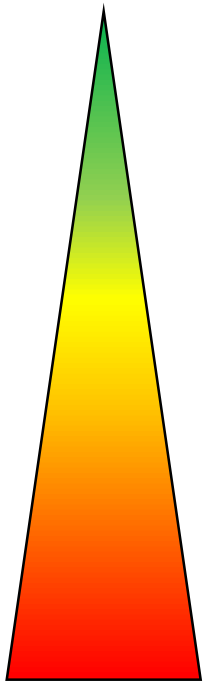
Profile of the impact of co-scheduling applications on all combinations of resources

Initial Application Placement



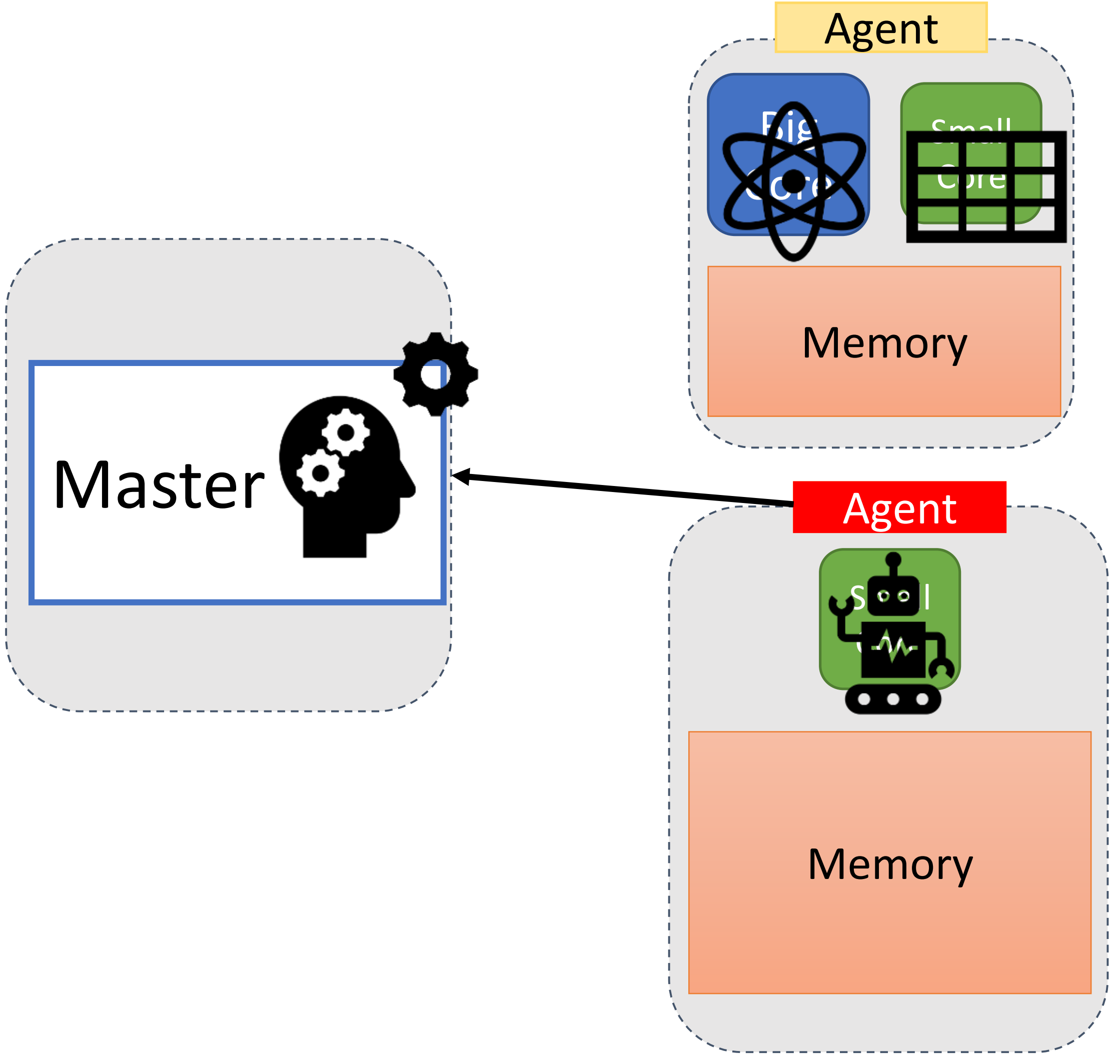
Runtime Monitoring and Rescheduling

Least invasive



Most invasive

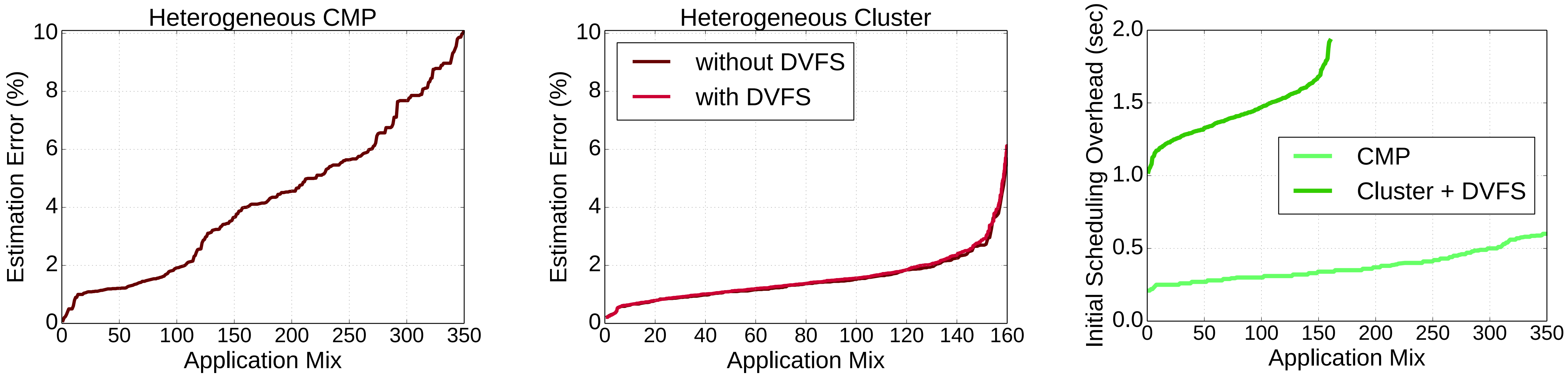
- Increase resources locally
- Migrate from smaller core to bigger core
- Migrate across servers



Evaluation

- **Workloads**
  - Single- and multi-threaded benchmark suites
  - Latency-critical, interactive services
- **Execution scenarios**
  - Simulated heterogeneous 16-core CMP
  - Real 40-server heterogeneous cluster
  - Real cluster with core-level heterogeneity using power management (DVFS)
- **Comparison schedulers**
  - Greedy, Smallest-First, Mage-Static, PIE [ISCA’12], Paragon [ASPLOS’13]

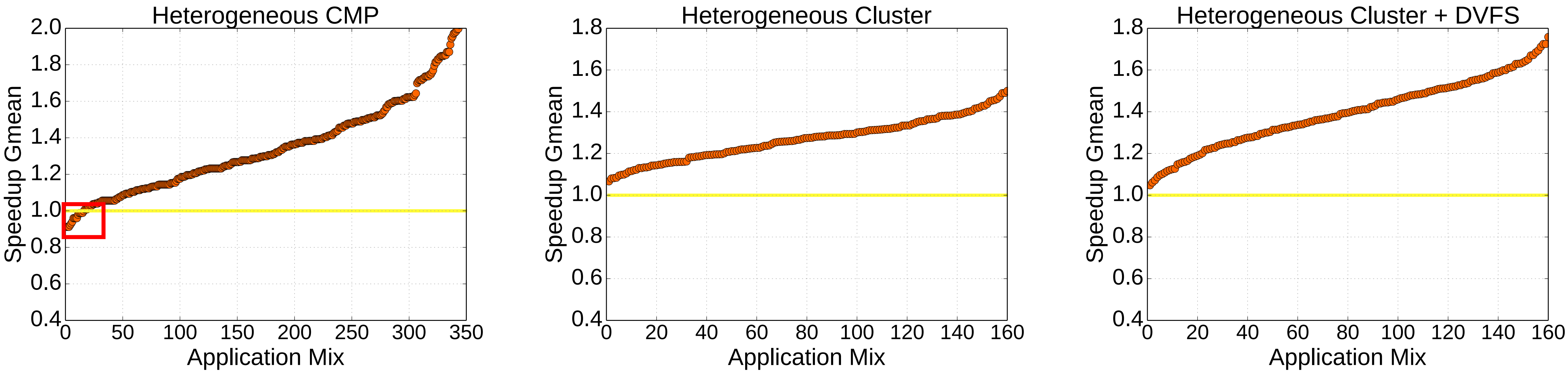
Low Error and Scheduling Overhead



Mage has low initial scheduling overhead and low estimation error

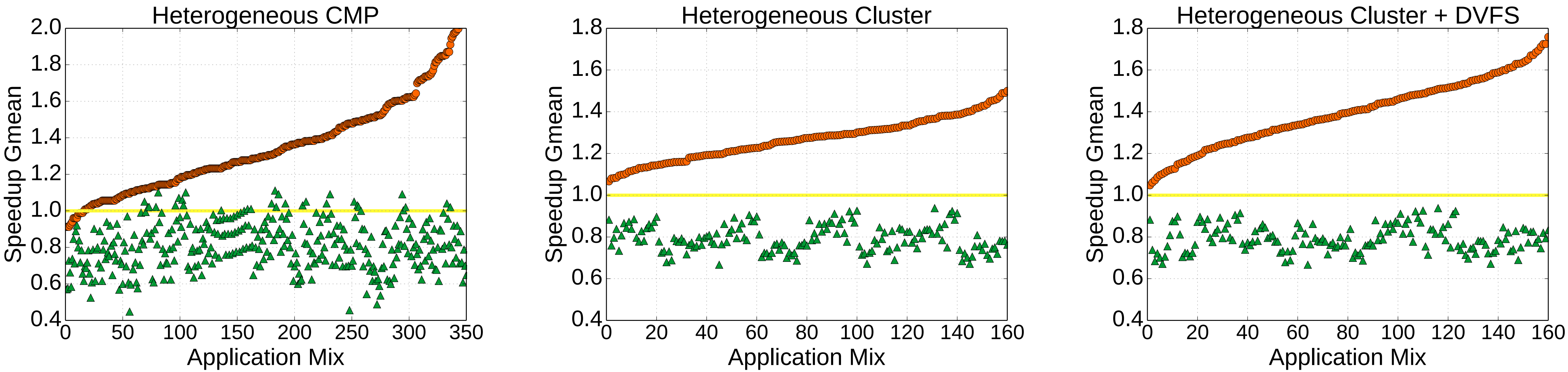
- Reduces the need to adjust scheduling decisions frequently during application lifetime

Versus Greedy



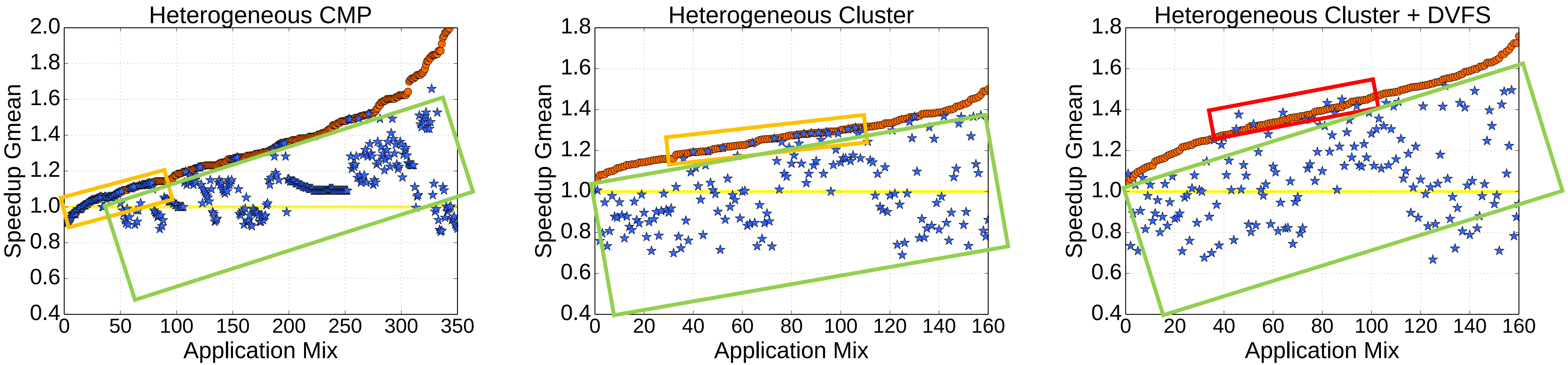
Mage outperforms the Greedy scheduler by only allocating the necessary resources to meet an application’s QoS

Versus Smallest-First



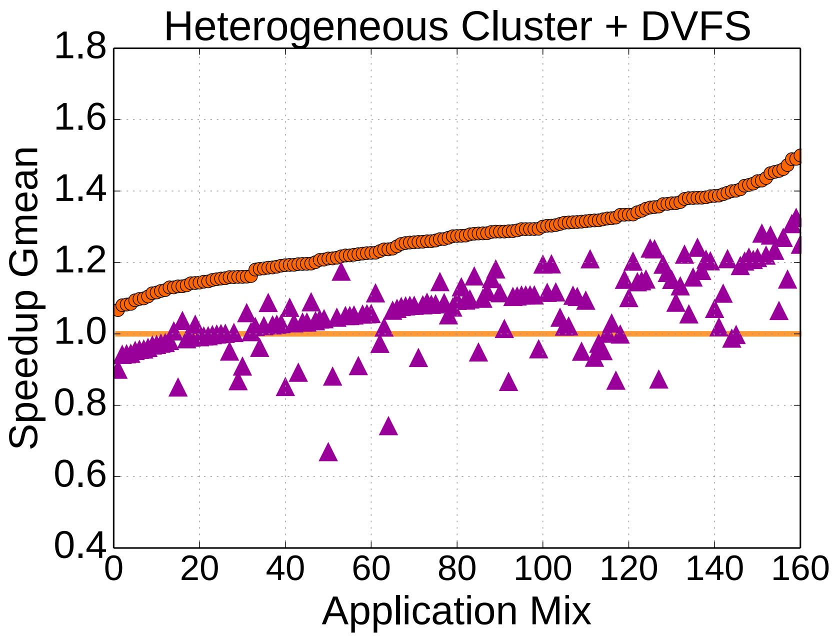
Mage outperforms the Smallest-First scheduler by not exacerbating contention in shared resources

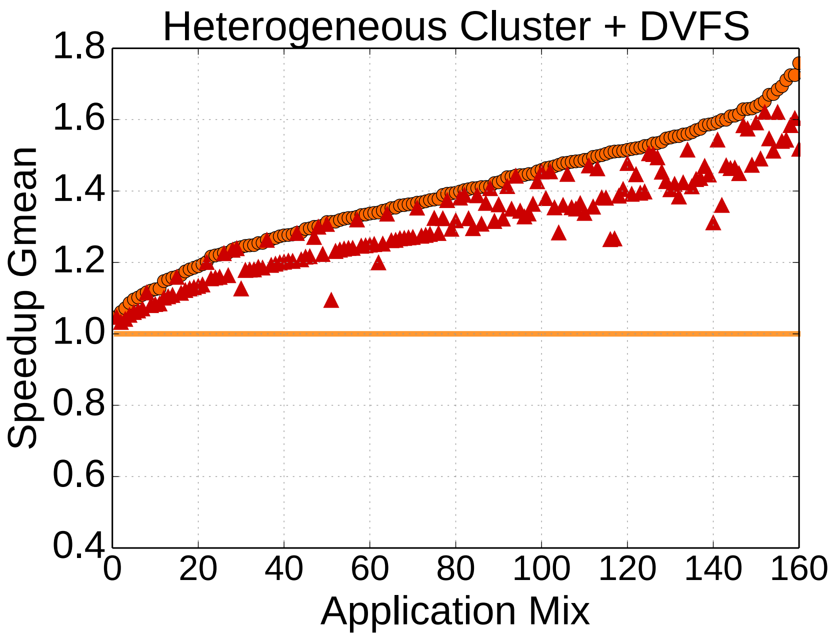
Versus Mage-Static



Mage outperforms *Mage-Static* by rescheduling applications that were mispredicted or that exhibit diurnal patterns

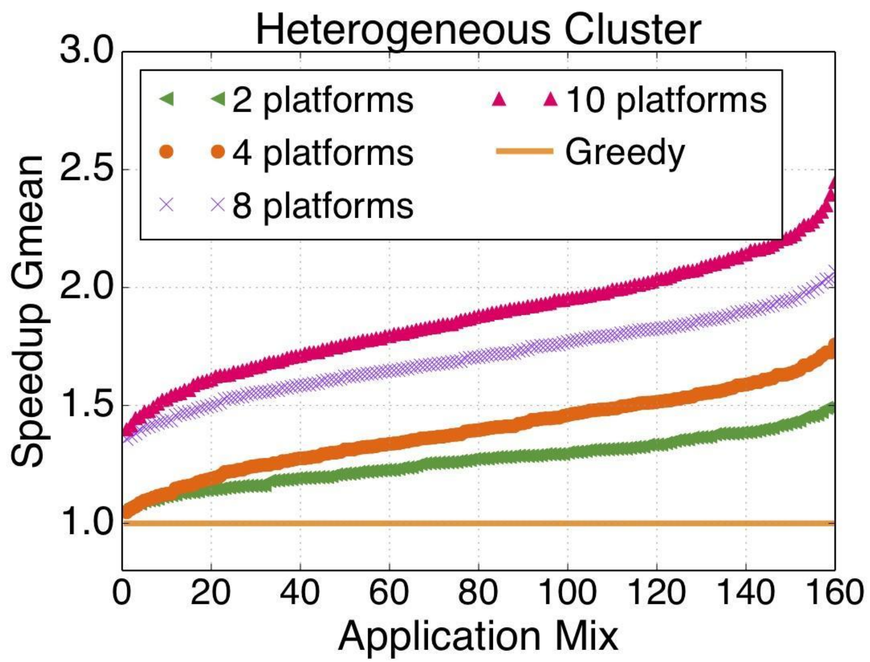
Versus Paragon+PIE and Paragon+Paragon

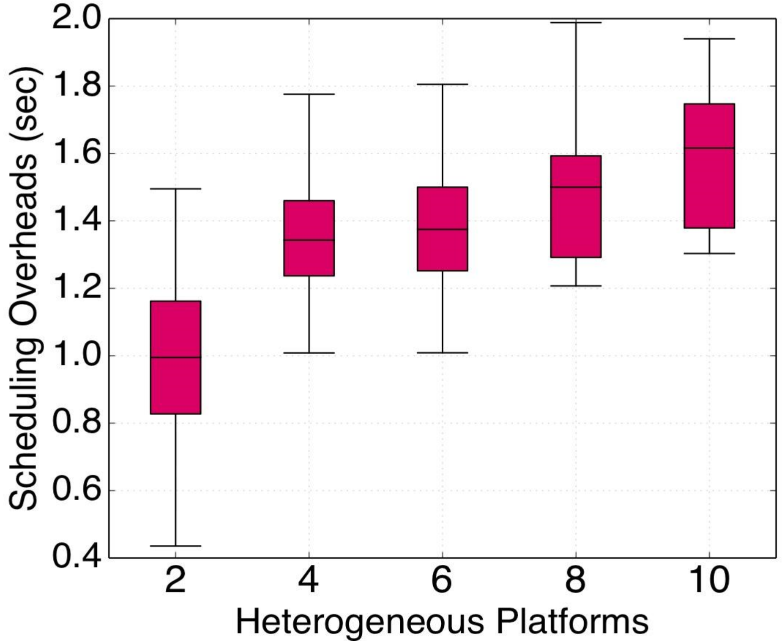




Mage outperforms Paragon+PIE and Paragon+Paragon by having a global view of resource availability and per-application resource requirements

Sensitivity to Heterogeneity Increase





- As degree of heterogeneity increases, the benefits of using Mage also increases
  - Results are also consistent for heterogeneous CMPs
- Minimal scheduling overhead as degree of heterogeneity increases

Conclusion

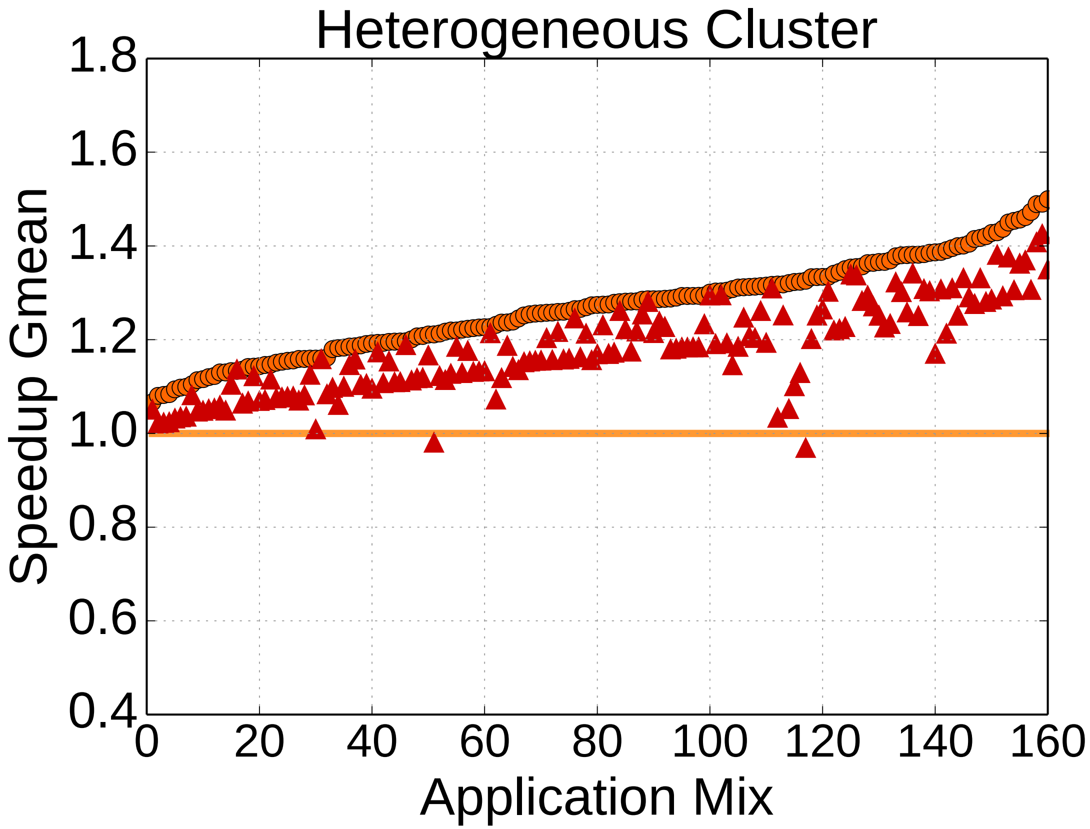
- Heterogeneity is becoming more prevalent; need a scheduler that can match applications to their resource needs
- Mage is a *tiered* scheduler that bridges the gap between CMP- and cluster-level heterogeneous scheduling
- Mage leverages a novel *staged*, parallel SGD algorithm to quickly and accurately classify applications
- Mage is lightweight and scalable
- Mage outperforms heterogeneity-agnostic *and* the sum of CMP- and cluster-level schedulers

Thank you!
Questions?

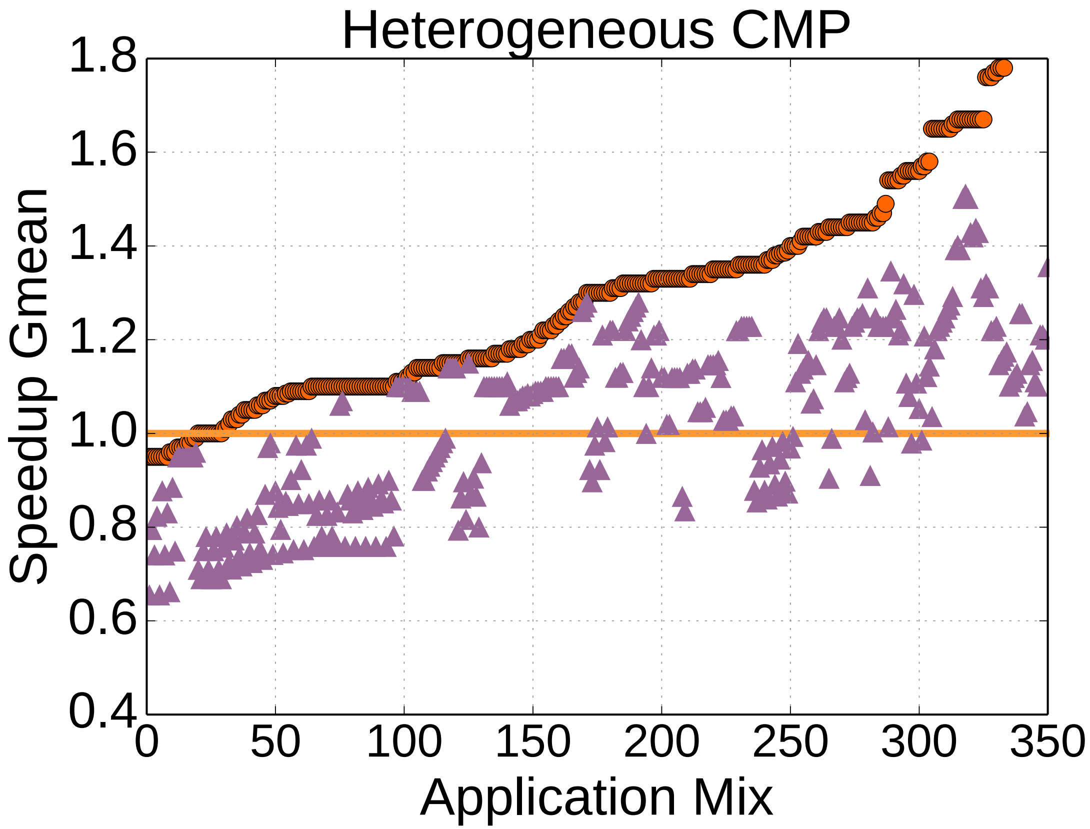
faromero@stanford.edu

Backup

Versus Paragon



Versus PIE


Heterogeneous CMP
Speedup Gmean
Application Mix
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0
50
100
150
200
250
300
350

Partial Interference Sensitivity – SGD Step 2

	App1:Core1 App2:Core2 App3:Core3	App1:Core1 App2:Core3 App3:Core2	App1:Core2 App2:Core1 App3:Core3	App1:Core3 App2:Core2 App3:Core1
App1	$MIPS_{1,1}$	$MIPS_{1,2}$	?	?
App2	$MIPS_{2,1}$	?	?	$MIPS_{2,6}$
App3	$MIPS_{3,1}$	?	$MIPS_{3,3}$	?

Solution: Run SGD *without* those columns, and add them in afterwards

Partial Interference Sensitivity – SGD Step 2

$\left[A_{SGD1} \right] \frown$

	App1:Core1 App2:Core2 App3:Core3	App1:Core1 App2:Core3 App3:Core2	App1:Core2 App2:Core1 App3:Core3	App1:Core3 App2:Core2 App3:Core1
App1	$MIPS_{1,1}$	$MIPS_{1,2}$	?	?
App2	$MIPS_{2,1}$	?	?	$MIPS_{2,6}$
App3	$MIPS_{3,1}$	?	$MIPS_{3,3}$	?

Solution: Run SGD *without* those columns, and add them in afterwards

Partial Interference Sensitivity – SGD Step 2

$$\underbrace{\begin{bmatrix} A_{SGD1} \end{bmatrix} \frown \begin{array}{r|c|c|c|c|} & \begin{matrix}\text{App1:Core1}\\ \text{App2:Core2}\\ \text{App3:Core3}\end{matrix} & \begin{matrix}\text{App1:Core1}\\ \text{App2:Core3}\\ \text{App3:Core2}\end{matrix} & \begin{matrix}\text{App1:Core2}\\ \text{App2:Core1}\\ \text{App3:Core3}\end{matrix} & \begin{matrix}\text{App1:Core3}\\ \text{App2:Core2}\\ \text{App3:Core1}\end{matrix} \\ \hline \text{App1} & MIPS_{1,1} & MIPS_{1,2} & MIPS_{1,3} & MIPS_{1,6} \\ \hline \text{App2} & MIPS_{2,1} & MIPS_{2,2} & MIPS_{2,3} & MIPS_{2,6} \\ \hline \text{App3} & MIPS_{3,1} & MIPS_{3,2} & MIPS_{3,3} & MIPS_{3,6} \\ \hline \end{array}}_{A_{SGD2}}$$

Solution: Run SGD *without* those columns, and add them in afterwards

Complete Placements – SGD Step 3

$$\left[\, A_{SGD2} \,\right] \cap$$

	App1:Core2 App2:Core3 App3:Core1	App1:Core3 App2:Core1 App3:Core2
App1	[min_{ASGD2}, max_{ASGD2}]	[min_{ASGD2}, max_{ASGD2}]
App2	[min_{ASGD2}, max_{ASGD2}]	[min_{ASGD2}, max_{ASGD2}]
App3	[min_{ASGD2}, max_{ASGD2}]	[min_{ASGD2}, max_{ASGD2}]

Populate remaining columns with results from Partial Placements

Complete Placements – SGD Step 3

$$\underbrace{\begin{bmatrix} A_{SGD2} \end{bmatrix} \frown \begin{array}{r|c|c|} & \begin{array}{c}\text{App1:Core2}\\ \text{App2:Core3}\\ \text{App3:Core1}\end{array} & \begin{array}{c}\text{App1:Core3}\\ \text{App2:Core1}\\ \text{App3:Core2}\end{array} \\ \hline \text{App1} & MIPS_{1,4} & MIPS_{1,5} \\ \hline \text{App2} & MIPS_{2,4} & MIPS_{2,5} \\ \hline \text{App3} & MIPS_{3,4} & MIPS_{3,5} \\ \hline \end{array}}_{A_{SGD3}}$$

Select column from A_{SGD3} with highest geometric mean for scheduling