HCLOUD: RESOURCE-EFFICIENT PROVISIONING IN SHARED CLOUD SYSTEMS

Christina Delimitrou[1] and Christos Kozyrakis[2]

[1]Stanford/Cornell University, [2]Stanford University/EPFL

http://mast.stanford.edu

ASPLOS – April 5th 2016

Executive Summary

- Problem: cloud provisioning is difficult
  - Many resource offerings → resource/cost inefficiencies
  - Interference from other users → performance jitter

- HCloud: resource-efficient public cloud provisioning
  - User provides ~~resource reservations~~ performance goals
  - Automatic selection of instance type and configuration
  - Adjust allocations between reserved and on-demand
  - High utilization and low performance jitter

Motivation

Google Cloud Platform

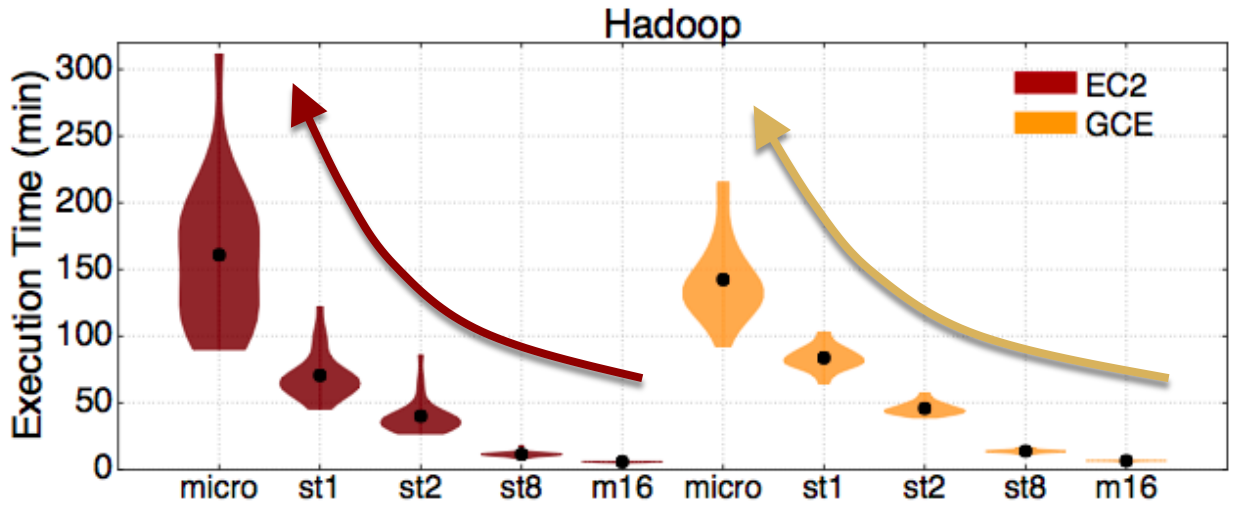
t2.nano	n1-standard-1	A0
t2.micro	n1-standard-2	A1
t2.small	n1-standard-4	A2
t2.medium	n1-standard-8	A3
t2.large	n1-standard-16	A4
m4.large	n1-standard-32	A5
m4.xlarge	n1-highmem-2	A6
m4.2xlarge	n1-highmem-4	A7
m4.4xlarge	n1-highmem-8	A8
m4.10xlarge	n1-highmem-16	A9
m3.medium	n1-highmem-32	A10
m3.large	n1-highcpu-2	A11
m3.xlarge	n1-highcpu-4	D1
m3.2xlarge	n1-highcpu-8	D2
c4.large	n1-highcpu-16	D3
c4.xlarge	n1-highcpu-32	D4
c4.2xlarge	f1-micro	D11
c4.4xlarge	g1-small	D12

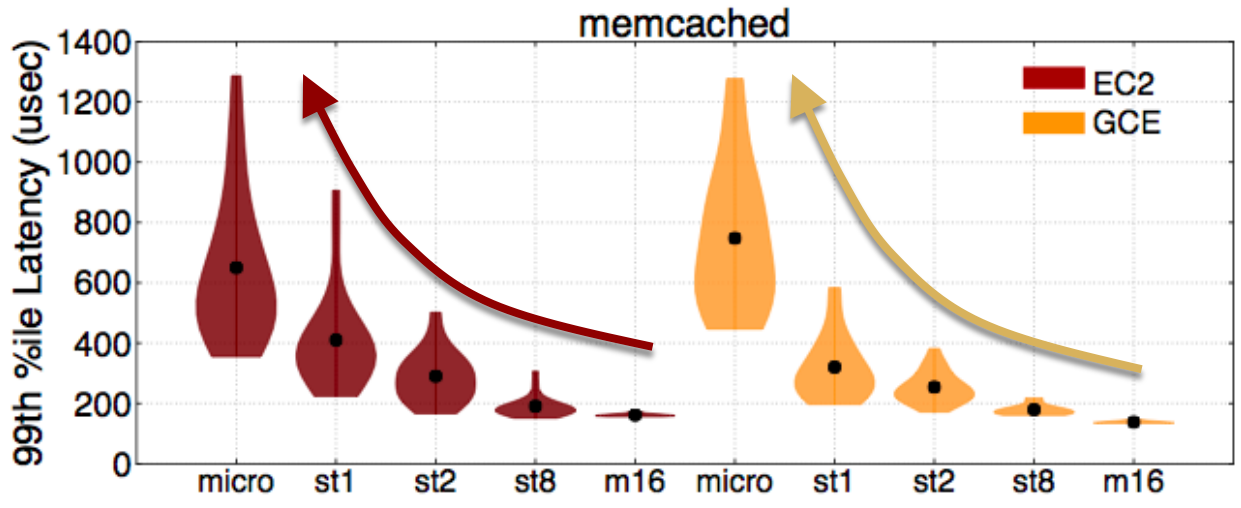
amazon web services™

Google Cloud Platform

Windows Azure™

t2.no
t2.mi
t2.sm
t2.me
t2.la
m4.lo
m4.x
m4.2
m4.4
m4.1
m3.m
m3.lo
m3.x
m3.2
c4.la
c4.xl
c4.2x





A0
A1
A2
A3
A4
A5
A6
A7
A8
A9
A10
A11
D1
D2
D3
D4
D11

Google Cloud Platform

Reserved	On-demand	Spot	On-demand	Discounted	Reserved	On-demand
t2.nano	t2.nano	t2.nano	n1-standard-1	n1-standard-1	A0	A0
t2.micro	t2.micro	t2.micro	n1-standard-2	n1-standard-2	A1	A1
t2.small	t2.small	t2.small	n1-standard-4	n1-standard-4	A2	A2
t2.medium	t2.medium	t2.medium	n1-standard-8	n1-standard-8	A3	A3
t2.large	t2.large	t2.large	n1-standard-16	n1-standard-16	A4	A4
m4.large	m4.large	m4.large	n1-standard-32	n1-standard-32	A5	A5
m4.xlarge	m4.xlarge	m4.xlarge	n1-highmem-2	n1-highmem-2	A6	A6
m4.2xlarge	m4.2xlarge	m4.2xlarge	n1-highmem-4	n1-highmem-4	A7	A7
m4.4xlarge	m4.4xlarge	m4.4xlarge	n1-highmem-8	n1-highmem-8	A8	A8
m4.10xlarge	m4.16xlarge	m4.10xlarge	n1-highmem-16	n1-highmem-16	A9	A9
m3.medium	m3.medium	m3.medium	n1-highmem-32	n1-highmem-32	A10	A10
m3.large	m3.large	m3.large	n1-highcpu-2	n1-highcpu-2	A11	A11
m3.xlarge	m3.xlarge	m3.xlarge	n1-highcpu-4	n1-highcpu-4	D1	D1
m3.2xlarge	m3.2xlarge	m3.2xlarge	n1-highcpu-8	n1-highcpu-8	D2	D2
c4.large	c4.large	c4.large	n1-highcpu-16	n1-highcpu-16	D3	D3
c4.xlarge	c4.xlarge	c4.xlarge	n1-highcpu-32	n1-highcpu-32	D4	D4
c4.2xlarge	c4.2xlarge	c4.2xlarge	f1-micro	f1-micro	D11	D11
c4.4xlarge	c4.4xlarge	c4.4xlarge	g1-small	g1-small	D12	D12
c4.8xlarge	c4.8xlarge	c4.8xlarge			D13	D13
c3.large	c3.large	c3.large			D14	D14
c3.xlarge	c3.xlarge	c3.xlarge			D1 v2	D1 v2
c3.2xlarge	c3.2xlarge	c3.2xlarge			D2 v2	D2 v2
c3.4xlarge	c3.4xlarge	c3.4xlarge			D3 v2	D3 v2
c3.8xlarge	c3.8xlarge	c3.8xlarge			D4 v2	D4 v2
r3.large	r3.large	r3.large			D5 v2	D5 v2

Google Cloud Platform

t2.nano	t2.nano	t2.nano	n1-standard-1	n1-standard-1	A0	A0
t2.micro	t2.micro	t2.micro	n1-standard-2	n1-standard-2	A1	A1
t2.small	t2.small	t2.small	n1-standard-4	n1-standard-4	A2	A2
t2.medium	t2.medium	t2.medium	n1-standard-8	n1-standard-8	A3	A3
t2.large	t2.large	t2.large	n1-standard-16	n1-standard-16	A4	A4
m4.large	m4.large	m4.large	n1-standard-32	n1-standard-32	A5	A5
m4.xlarge	m4.xlarge	m4.xlarge	n1-highmem-2	n1-highmem-2	A6	A6
m4.2xlarge	m4.2xlarge				A7	A7
m4.4xlarge	m4.4xlarge				A8	A8
m4.10xlarge	m4.10xlarge				A9	A9
m3.medium	m3.medium				A10	A10
m3.large	m3.large				A11	A11
m3.xlarge	m3.xlarge				D1	D1
m3.2xlarge	m3.2xlarge				D2	D2
c4.large	c4.large				D3	D3
c4.xlarge	c4.xlarge	c4.xlarge	n1-highcpu-32	n1-highcpu-32	D4	D4
c4.2xlarge	c4.2xlarge	c4.2xlarge	f1-micro	f1-micro	D11	D11
c4.4xlarge	c4.4xlarge	c4.4xlarge	g1-small	g1-small	D12	D12
c4.8xlarge	c4.8xlarge	c4.8xlarge			D13	D13
c3.large	c3.large	c3.large			D14	D14
c3.xlarge	c3.xlarge	c3.xlarge			D1 v2	D1 v2
c3.2xlarge	c3.2xlarge	c3.2xlarge			D2 v2	D2 v2
c3.4xlarge	c3.4xlarge	c3.4xlarge			D3 v2	D3 v2
c3.8xlarge	c3.8xlarge	c3.8xlarge			D4 v2	D4 v2
r3.large	r3.large	r3.large			D5 v2	D5 v2

Reserved

On-demand

Reserved

On-demand

Cost, predictability, availability,
flexibility, spin-up overheads,
retention time, load fluctuation,
external load, sensitivity to interference, …

Google Cloud Platform

Reserved	On-demand	Spot	On-demand	Reserved	On-demand
t2.nano	t2.nano	t2.nano	n1-standard-1	n1-standard-1	A0	A0
t2.micro	t2.micro	t2.micro	n1-standard-2	n1-standard-2	A1	A1
t2.small	t2.small	t2.small	n1-standard-4	n1-standard-4	A2	A2
t2.medium	t2.medium	t2.medium	n1-standard-8	n1-standard-8	A3	A3
t2.large	t2.large	t2.large	n1-standard-16	n1-standard-16	A4	A4
m4.large	m4.large	m4.large	n1-standard-32	n1-standard-32	A5	A5
m4.xlarge	m4.xlarge	m4.xlarge	n1-highmem-2	n1-highmem-2	A6	A6
m4.2xlarge	m4.2xlarge	m4.2xlarge	n1-highmem-4	n1-highmem-4	A7	A7
m4.4xlarge	m4.4xlarge	m4.4xlarge	n1-highmem-8	n1-highmem-8	A8	A8
m4.10xlarge	m4.16xlarge	m4.10xlarge	n1-highmem-16	n1-highmem-16	A9	A9
m3.medium	m3.medium	m3.medium	n1-highmem-32	n1-highmem-32	A10	A10
m3.large	m3.large	m3.large	n1-highcpu-2	n1-highcpu-2	A11	A11
m3.xlarge	m3.xlarge	m3.xlarge	n1-highcpu-4	n1-highcpu-4	D1	D1
m3.2xlarge	m3.2xlarge	m3.2xlarge	n1-highcpu-8	n1-highcpu-8	D2	D2
c4.large	c4.large		n1-highcpu-16	n1-highcpu-16	D3	D3
c4.xlarge	c4.xlarge		n1-highcpu-32	n1-highcpu-32	D4	D4
c4.2xlarge	c4.2xlarge		f1-micro	f1-micro	D11	D11
c4.4xlarge	c4.4xlarge		g1-small	g1-small	D12	D12
c4.8xlarge	c4.8xlarge				D13	D13
c3.large	c3.large	c3.large			D14	D14
c3.xlarge	c3.xlarge	c3.xlarge			D1 v2	D1 v2
c3.2xlarge	c3.2xlarge	c3.2xlarge			D2 v2	D2 v2
c3.4xlarge	c3.4xlarge	c3.4xlarge			D3 v2	D3 v2
c3.8xlarge	c3.8xlarge	c3.8xlarge			D4 v2	D4 v2
r3.large	r3.large	r3.large			D5 v2	D5 v2

Overprovisioning
High cost &
Unpredictable performance

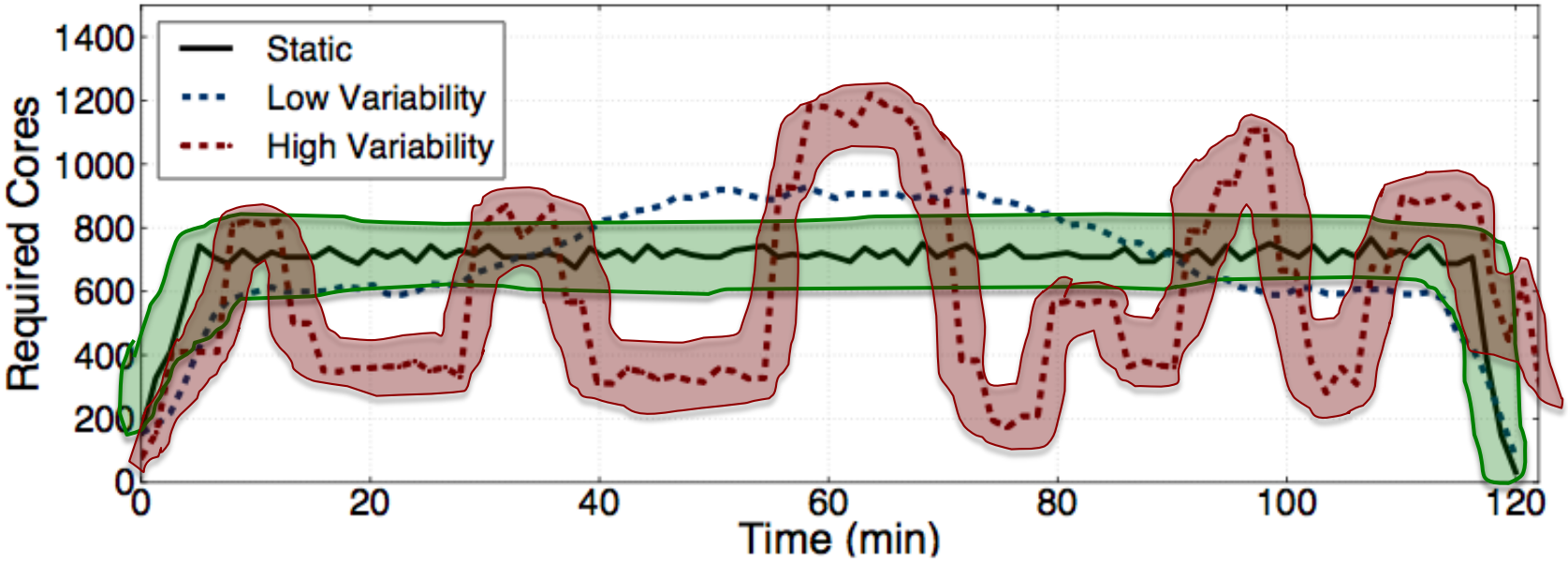
Cloud Provisioning Goals

✗ Determine appropriate instance size/type

✗ Determine appropriate instance configuration

✗ Dynamically adjust allocation decisions at runtime

Cloud Provisioning Scenarios



- Batch analytics, latency-critical applications, and scientific workloads in all scenarios
  - Batch: 50%
  - Latency: 40%
  - Scientific: 10%

Provisioning Baselines

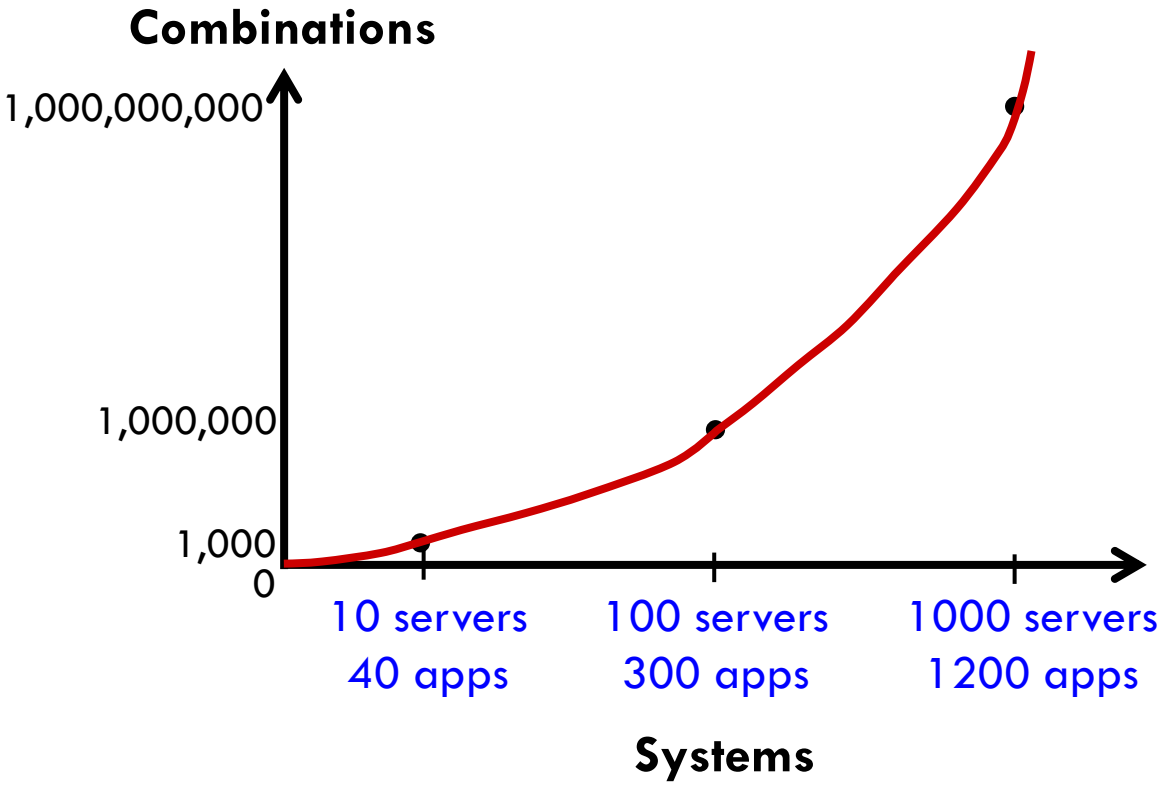
- All reserved (SR):
  - 16vCPU instances on GCE, no external load
  - Predictable performance, high cost, low flexibility
  - Applications provisioned for peak requirements

- All on-demand:
  - Largest instances (OdF): only 16vCPU instances
  - Mixed instances (OdM): mix of large and smaller instances
  - Interference, low cost, high flexibility

- Resource management:
  - ~~Least-loaded scheduler~~

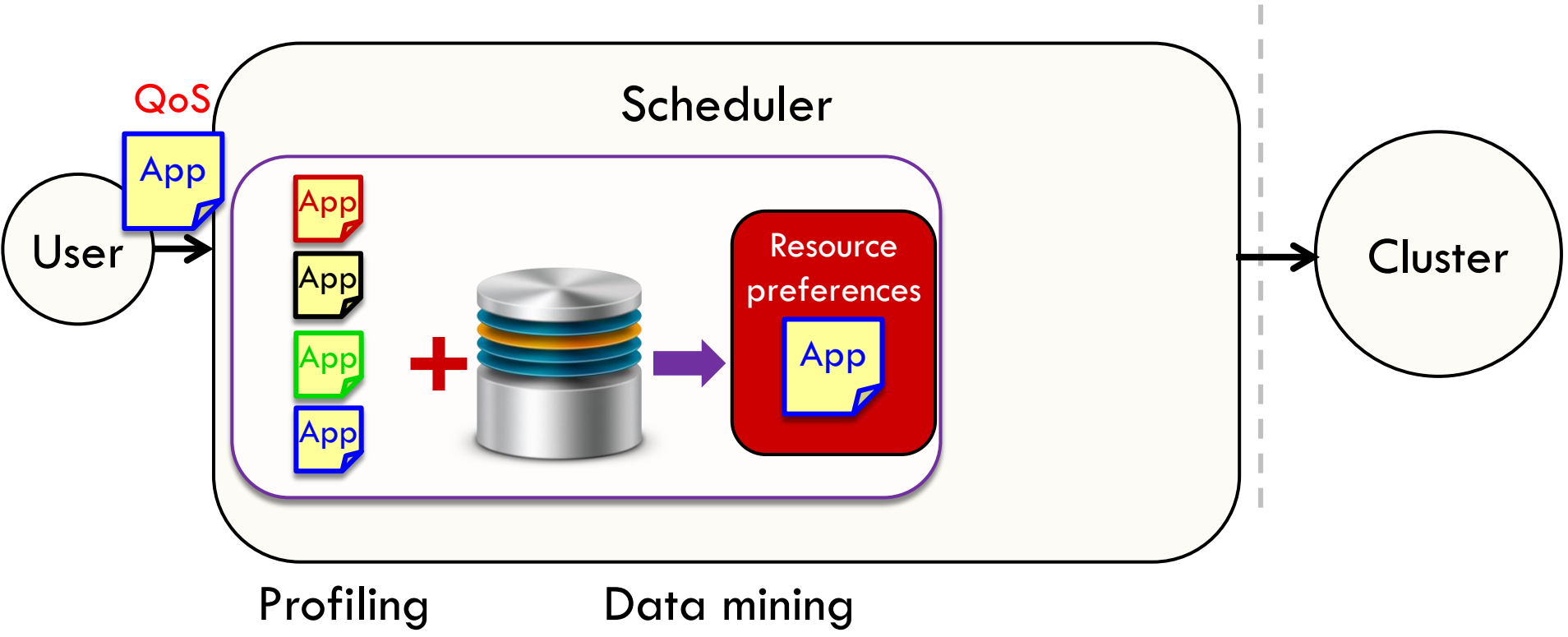
Extracting Resource Preferences

- Heterogeneity
- Interference
- Resources per server
- Resource ratio
- Number of servers
- Application params

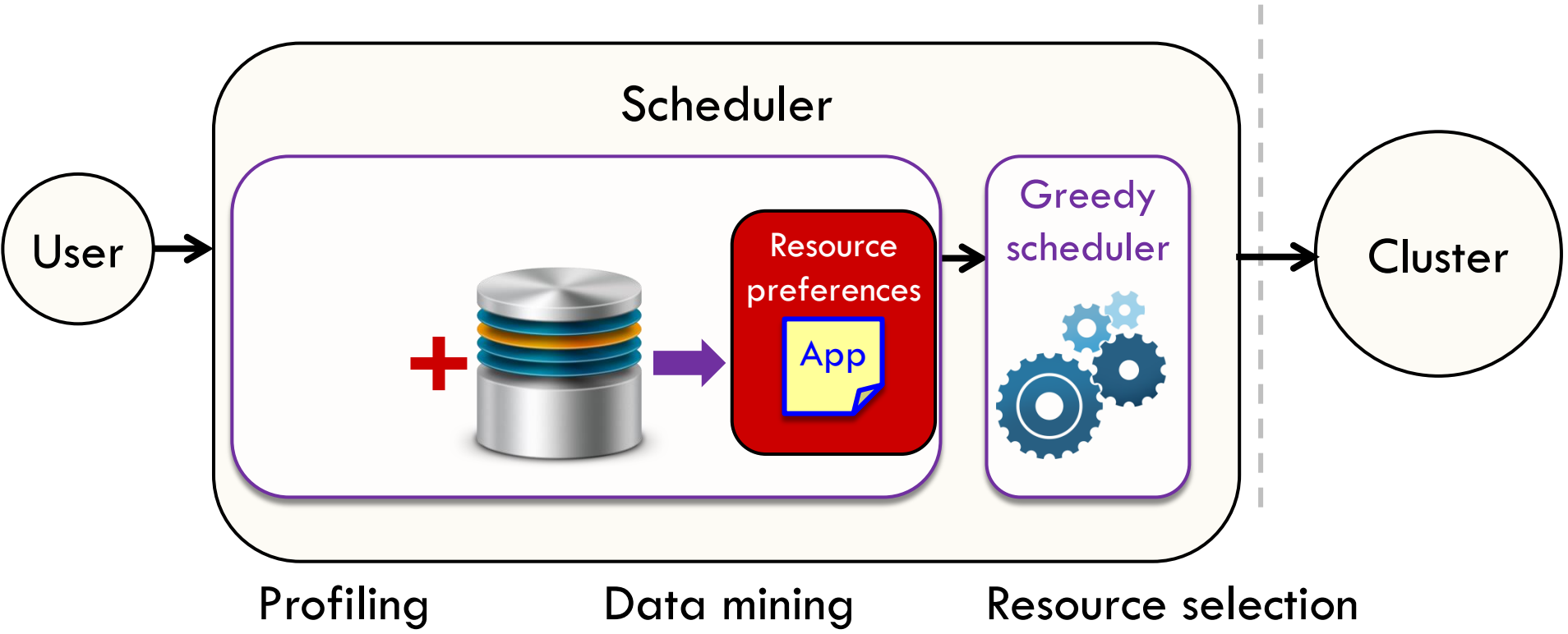


- Exhaustive characterization is infeasible

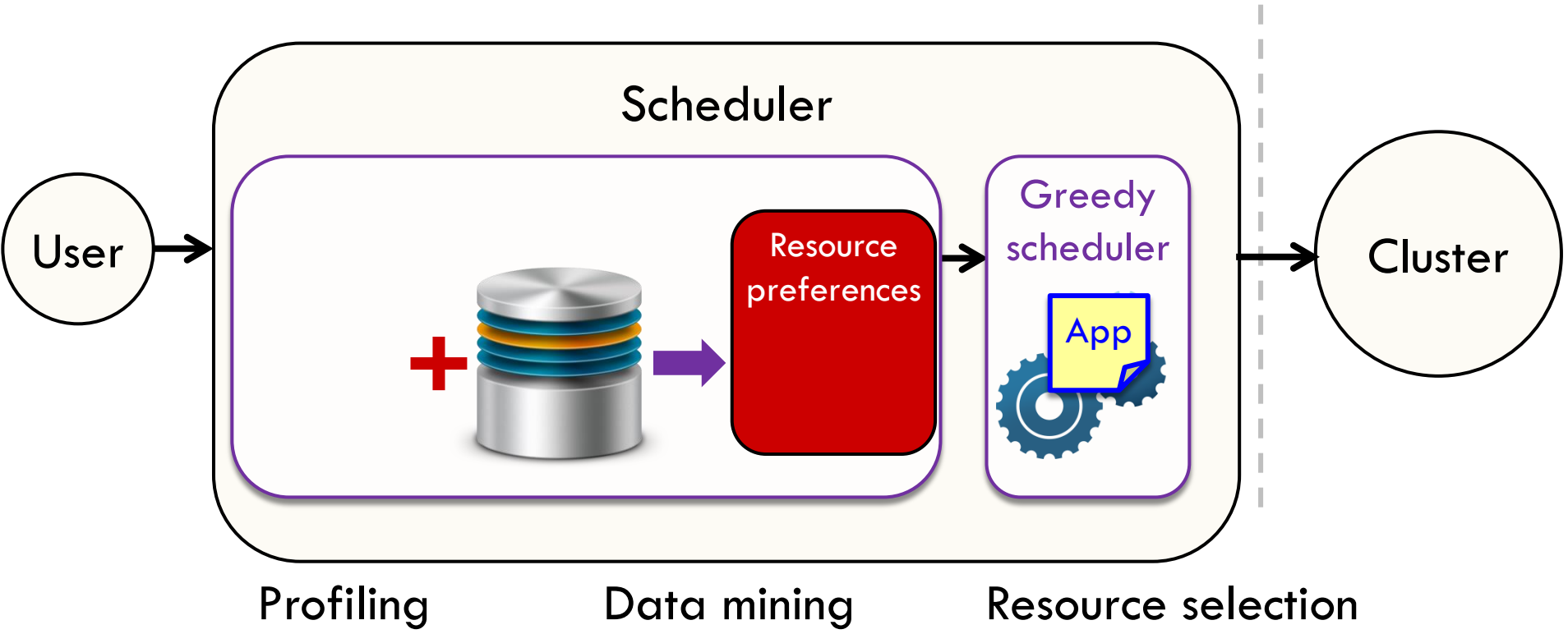
Quasar Overview [ASPLOS'14]



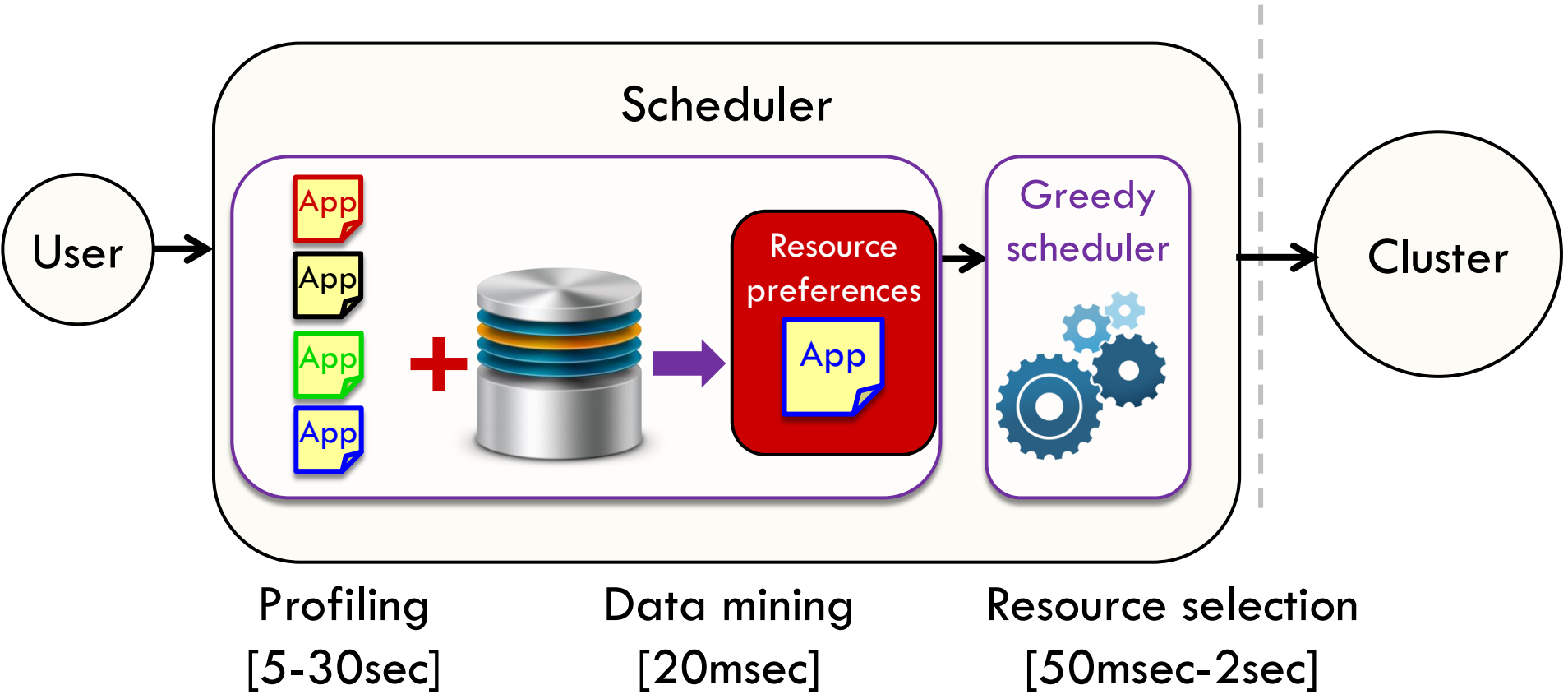
Quasar Overview [ASPLOS'14]



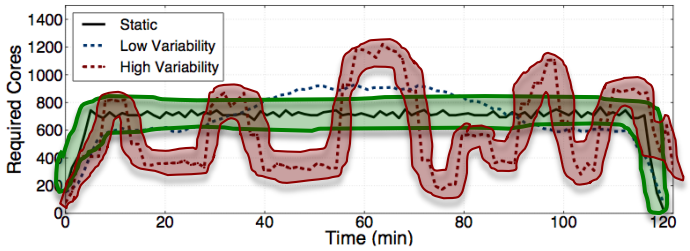
Quasar Overview [ASPLOS'14]

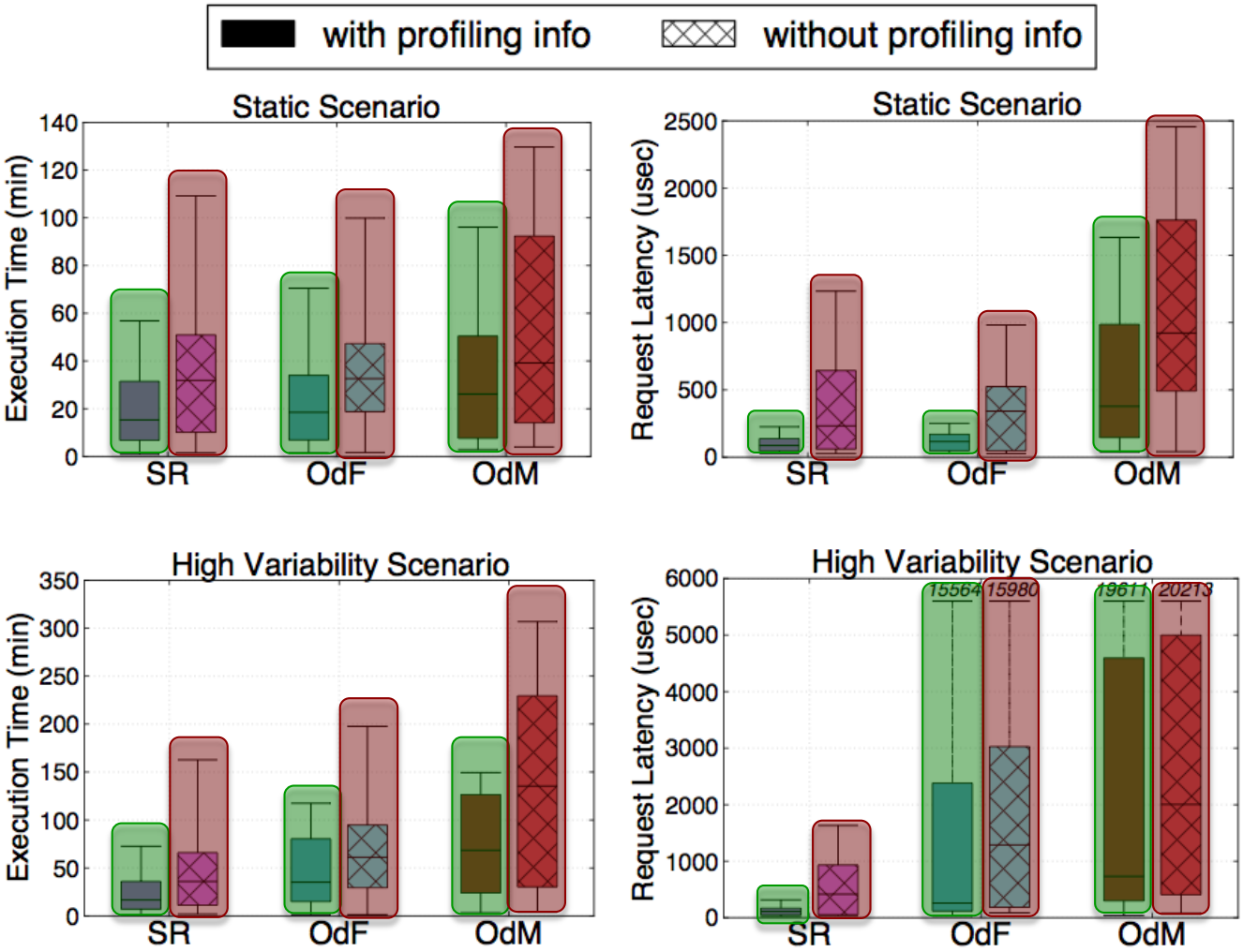


Quasar Overview [ASPLOS'14]

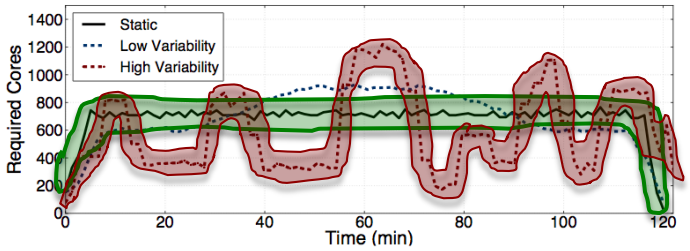


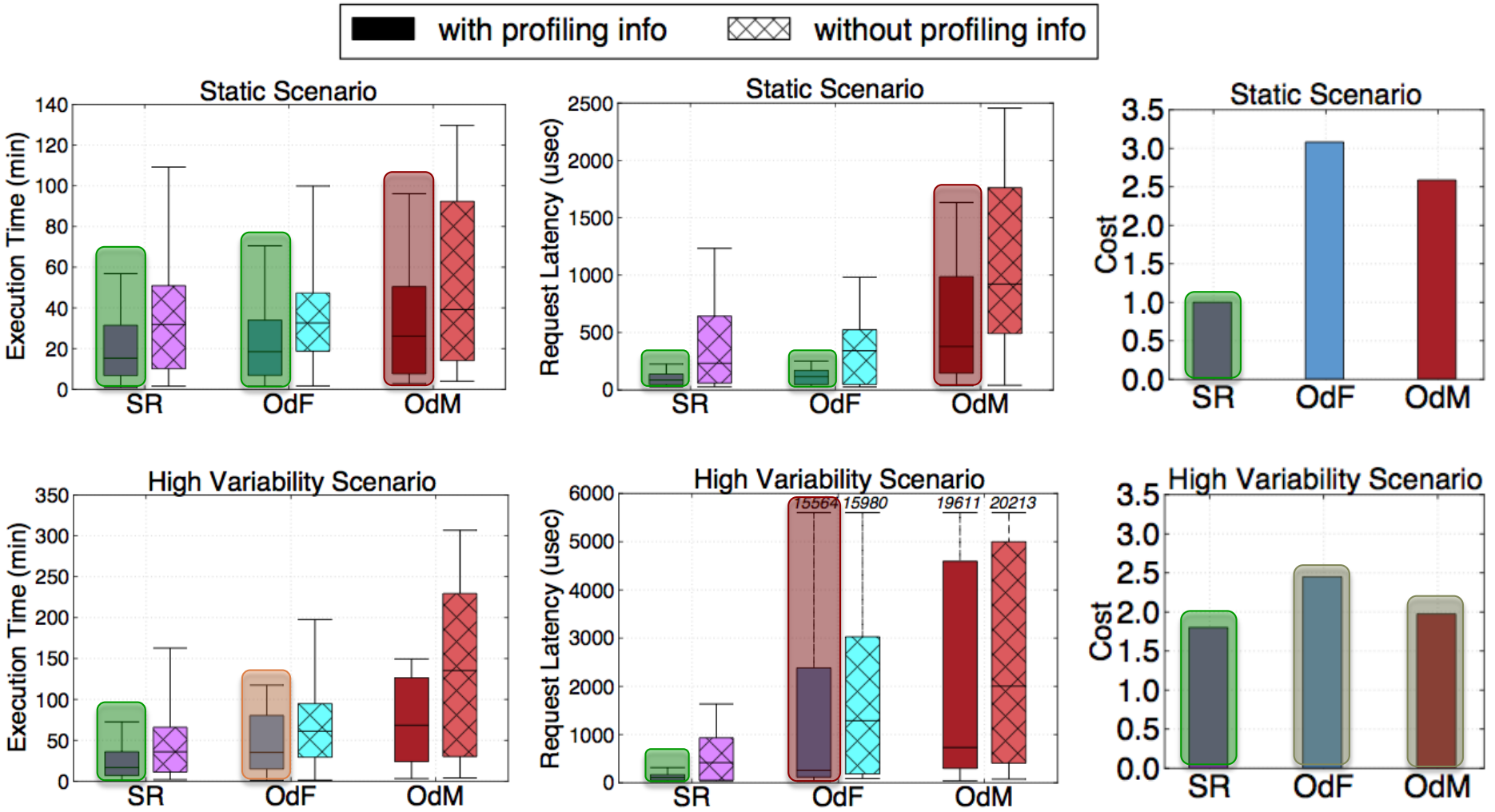
Evaluation


Static
Low Variability
High Variability
Required Cores
Time (min)


with profiling info
without profiling info
Static Scenario
Execution Time (min)
Request Latency (usec)
High Variability Scenario
SR
OdF
OdM
15564
15980
19611
20213

Evaluation


Static
Low Variability
High Variability
Required Cores
Time (min)


with profiling info
without profiling info
Static Scenario
Execution Time (min)
Request Latency (usec)
Cost
High Variability Scenario
15564
15980
19611
20213
SR
OdF
OdM

Cloud Provisioning Goals

✔ Determine appropriate instance size/type

✗ Determine appropriate instance configuration

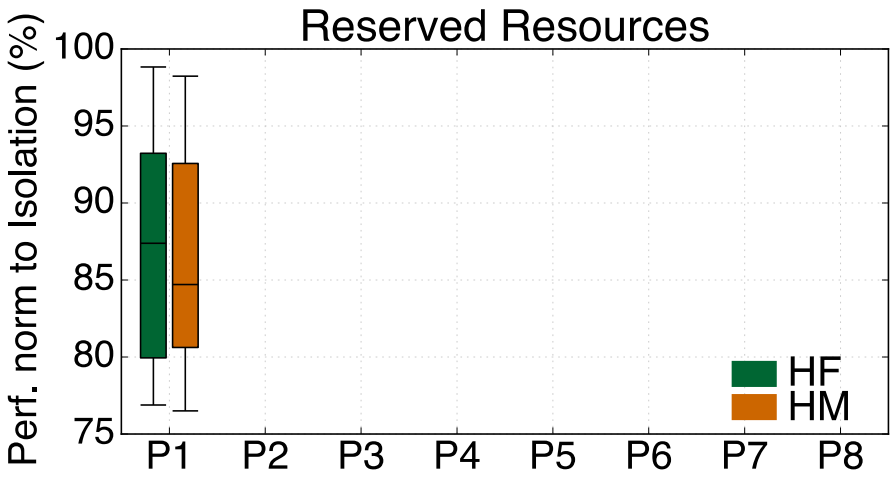
✗ Dynamically adjust allocation decisions at runtime

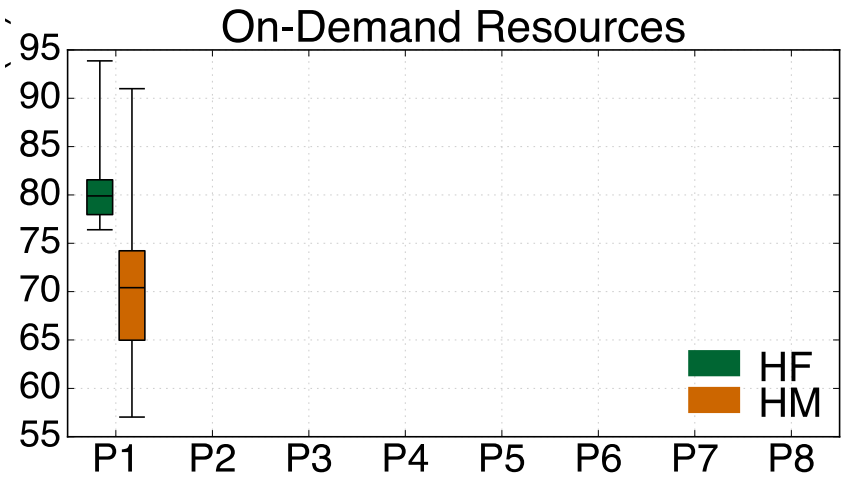
Hybrid Cloud Resource Allocation

- Insight:
  - Combine reserved and on-demand resources

- Challenge:
  - Separate applications between reserved (~private) and on-demand (public) resources

Hybrid Allocation Policies

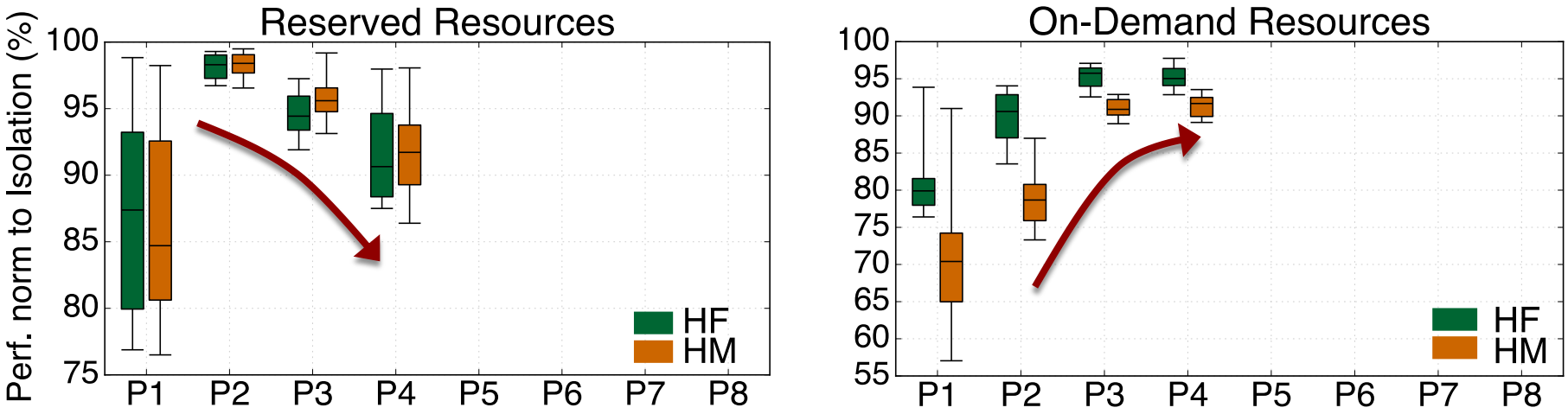




✗ Account for application interference sensitivity

P1: Random
P2: Q > 80% to reserved
P3: Q > 50% to reserved
P4: Q > 20% to reserved
P5: Reserved load < 50%
P6: Reserved load < 70%
P7: Reserved load < 90%
P8: Dynamic Policy

Hybrid Allocation Policies

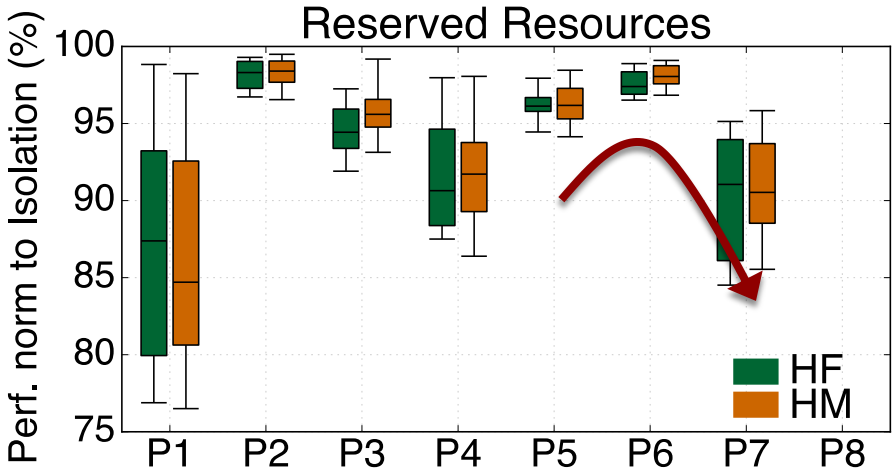


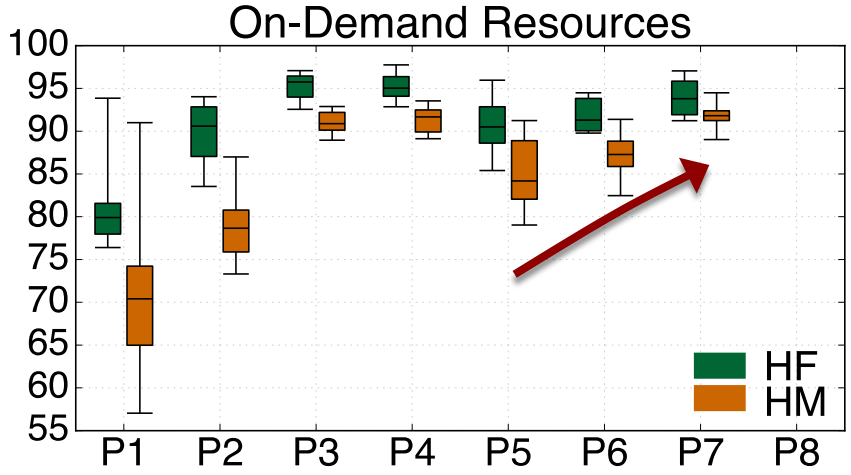
✓ Account for application interference sensitivity

✗ Do not overload reserved resources

P1: Random
P2: Q > 80% to reserved
P3: Q > 50% to reserved
P4: Q > 20% to reserved
P5: Reserved load < 50%
P6: Reserved load < 70%
P7: Reserved load < 90%
P8: Dynamic Policy

Hybrid Allocation Policies





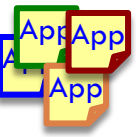
✓ Account for application interference sensitivity

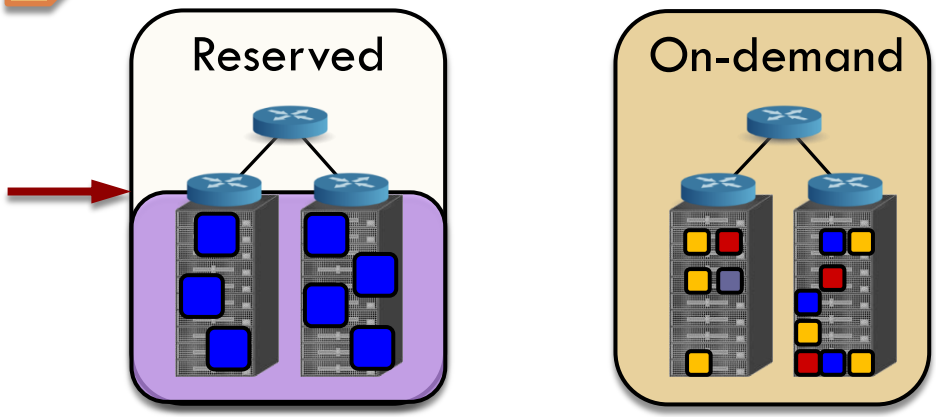
✓ Do not overload reserved resources

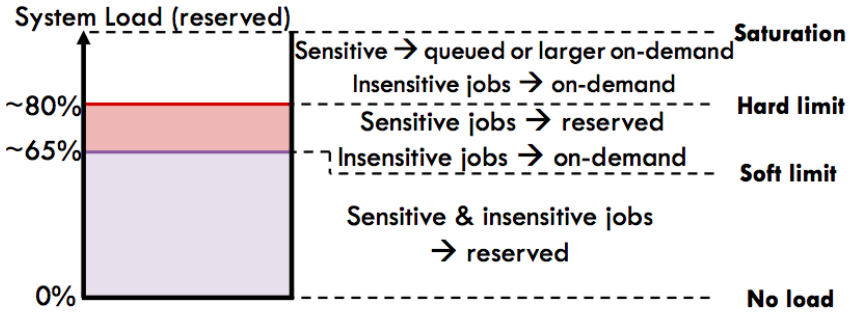
✗ Dynamic decisions (e.g., utilization limits)

P1: Random
P2: Q > 80% to reserved
P3: Q > 50% to reserved
P4: Q > 20% to reserved
P5: Reserved load < 50%
P6: Reserved load < 70%
P7: Reserved load < 90%
P8: Dynamic Policy

HCloud

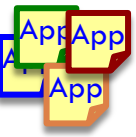


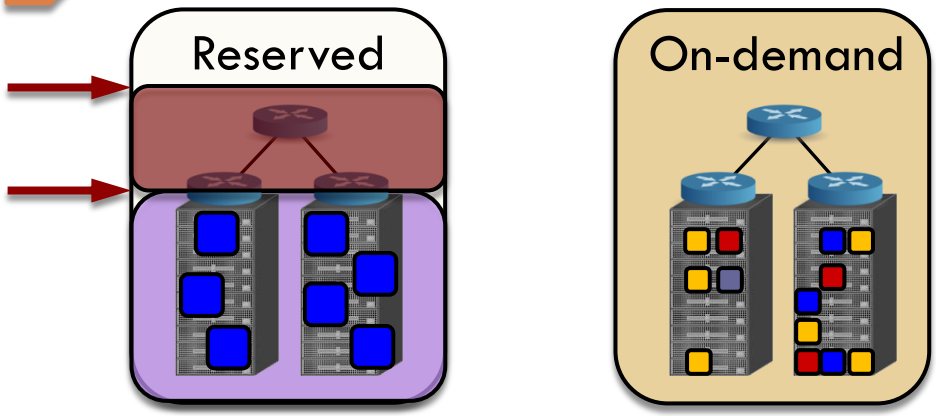


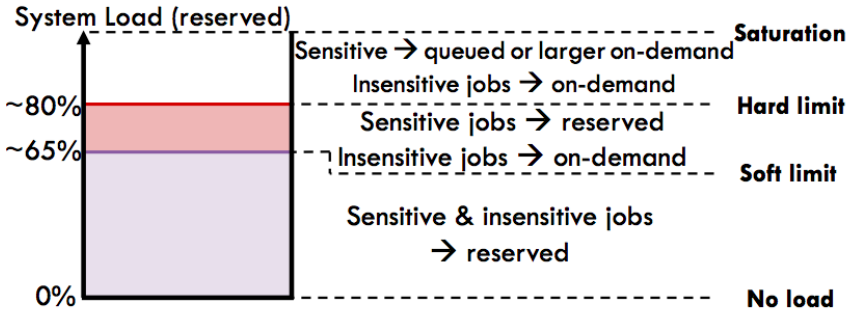


- Insights:
  - Account for interference sensitivity
  - Set load limits dynamically

HCloud

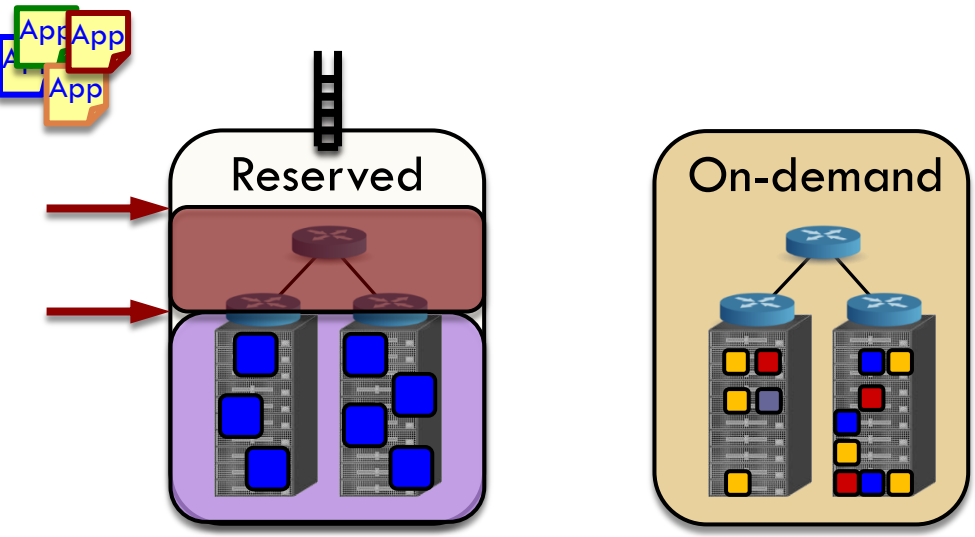


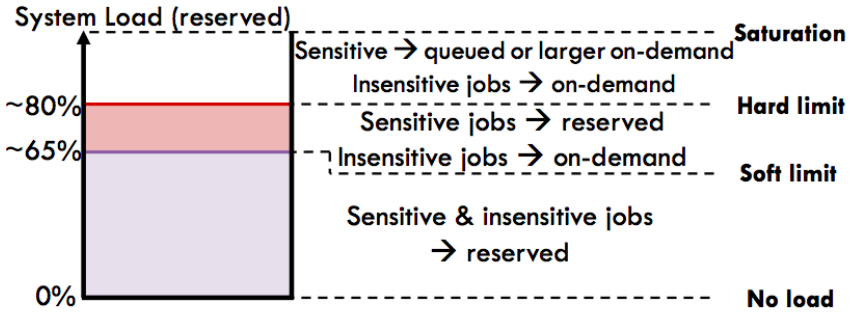




- Insights:
  - Account for interference sensitivity
  - Set load limits dynamically

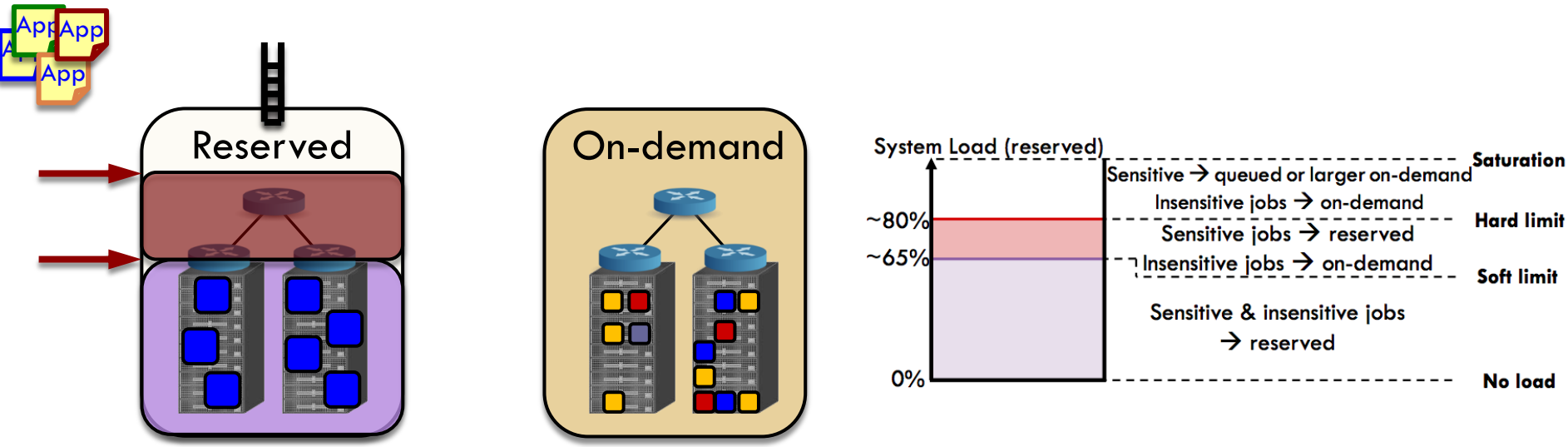
HCloud





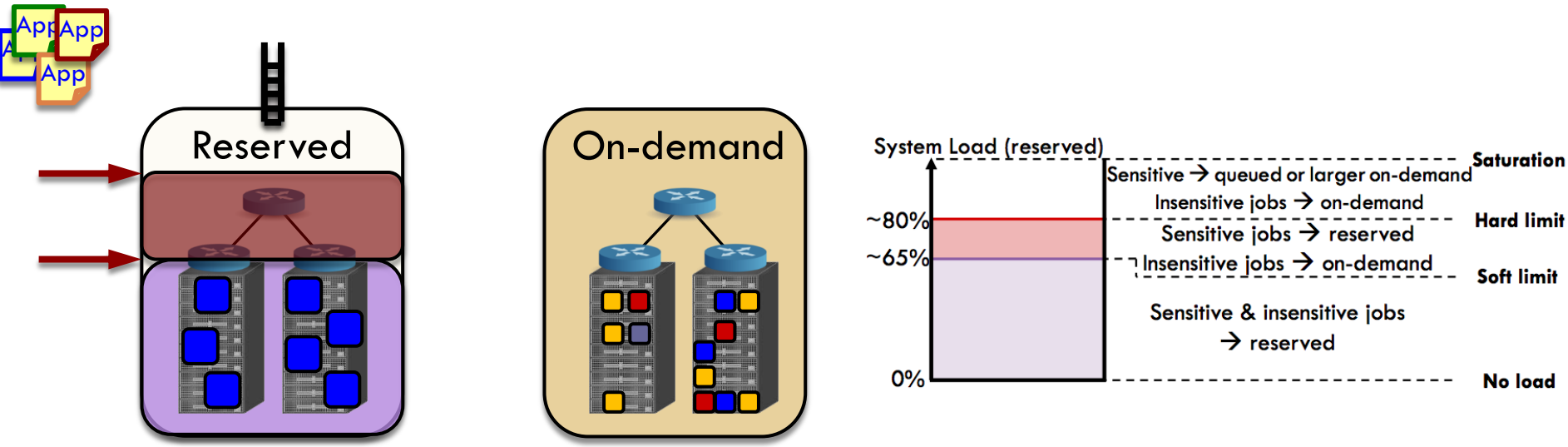
- Insights:
  - Account for interference sensitivity
  - Set load limits dynamically

HCloud



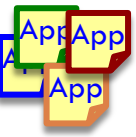
- Insights:
  - Account for interference sensitivity
  - Set load limits dynamically (feedback loop on queue length)

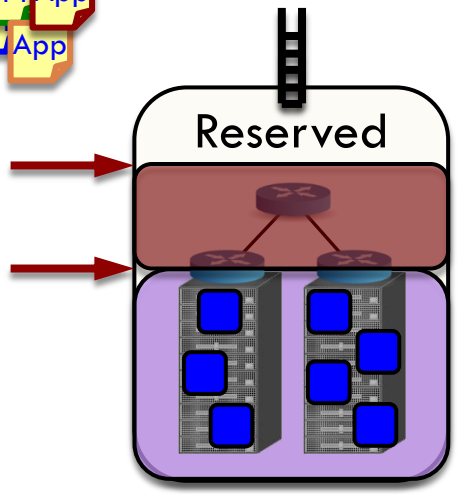
HCloud

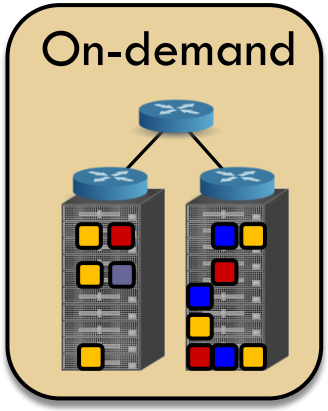


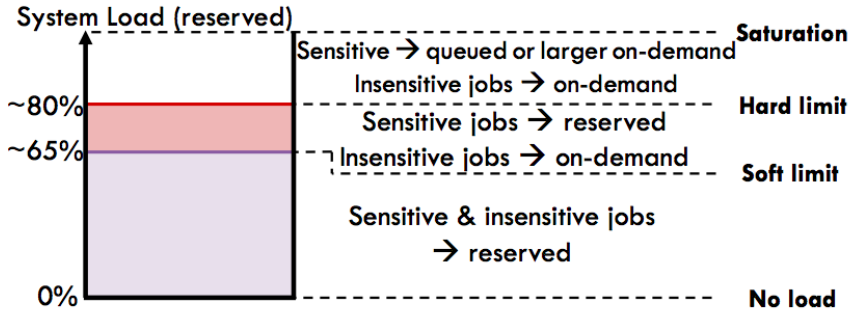
- Insights:
  - Account for interference sensitivity
  - Set load limits dynamically (feedback loop on queue length)
- What signals a sensitive job?

HCloud

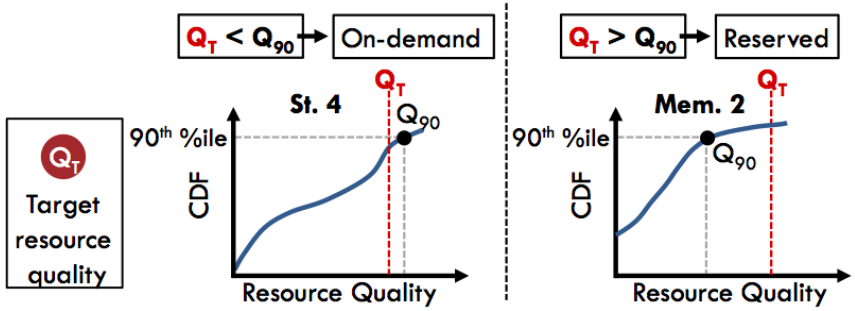




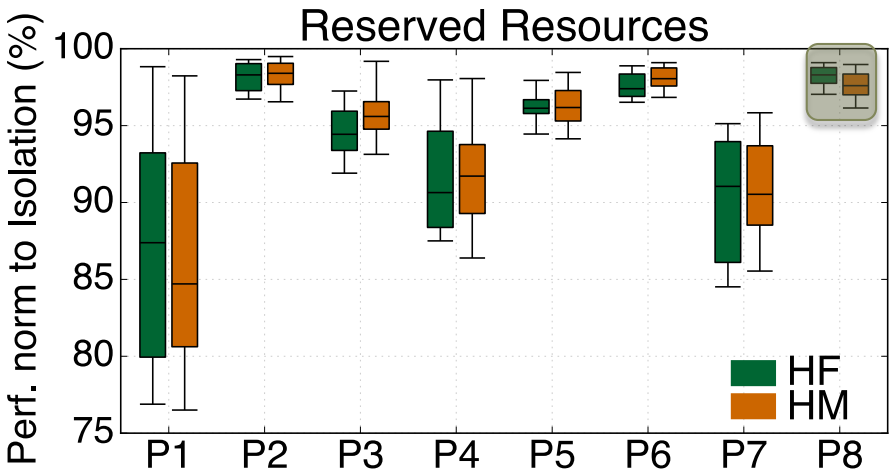


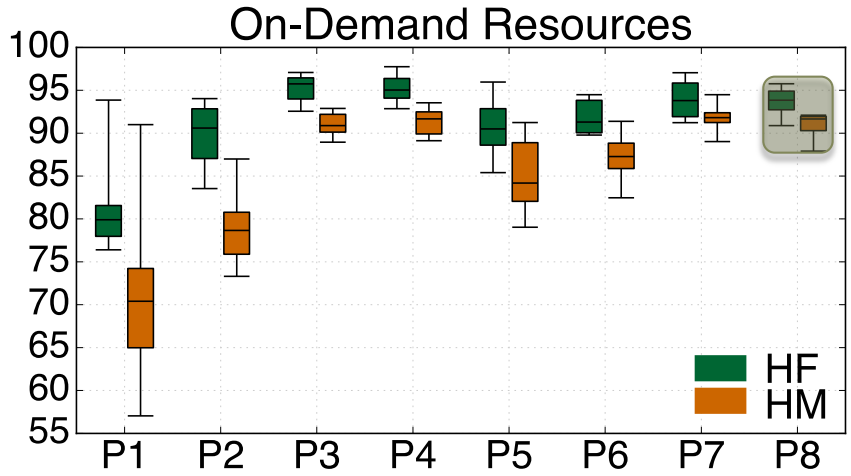


- Insights:
  - Account for interference sensitivity
  - Set load limits dynamically
- What signals a sensitive job?



Hybrid Allocation Policies





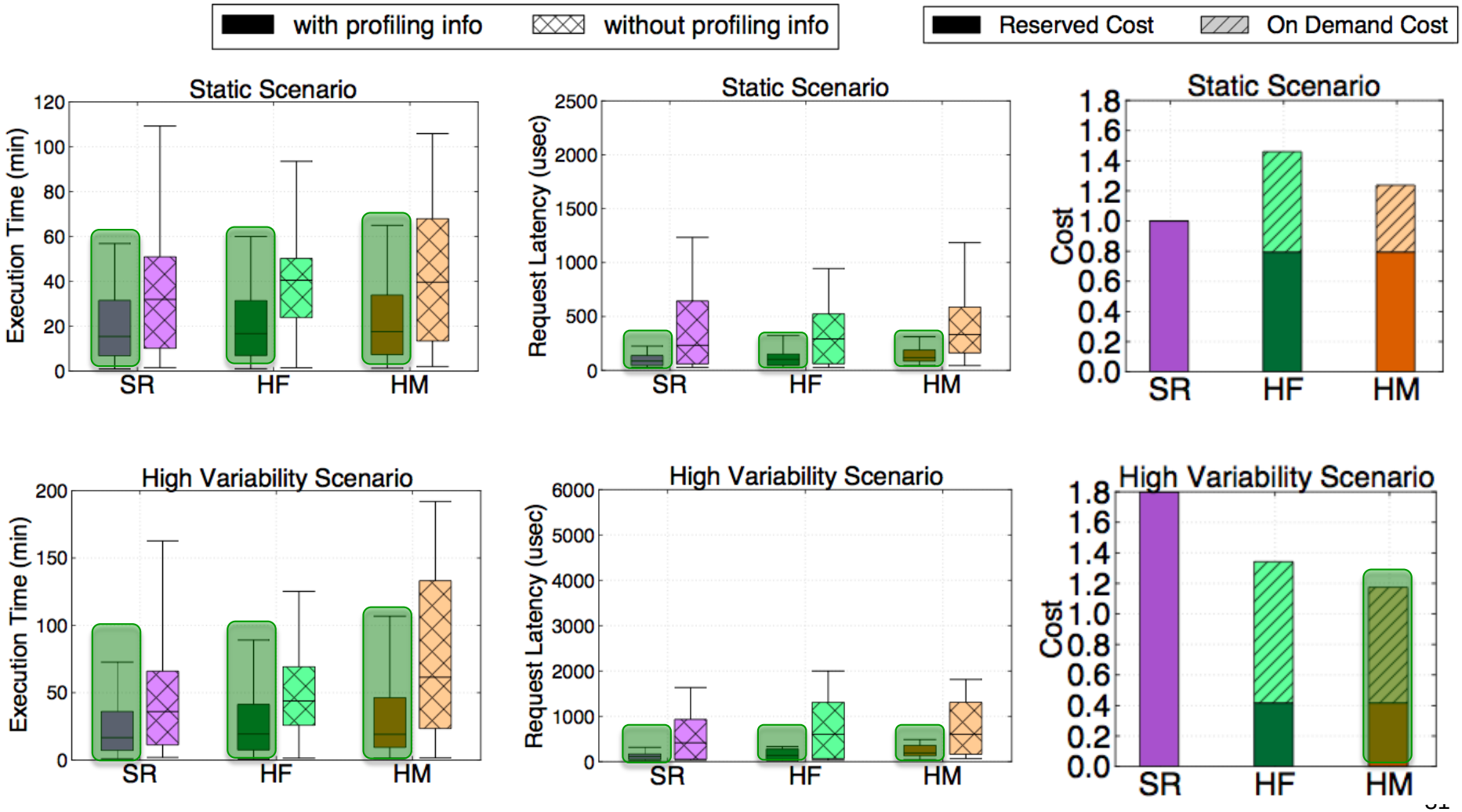
✓ Account for application interference sensitivity

✓ Do not overload reserved resources

✓ Dynamic decisions (e.g., utilization limits)

P1: Random
P2: Q > 80% to reserved
P3: Q > 50% to reserved
P4: Q > 20% to reserved
P5: Reserved load < 50%
P6: Reserved load < 70%
P7: Reserved load < 90%
P8: Dynamic Policy

Evaluation


with profiling info
without profiling info
Reserved Cost
On Demand Cost
Static Scenario
Execution Time (min)
Request Latency (usec)
Cost
High Variability Scenario
SR
HF
HM

Cloud Provisioning Goals

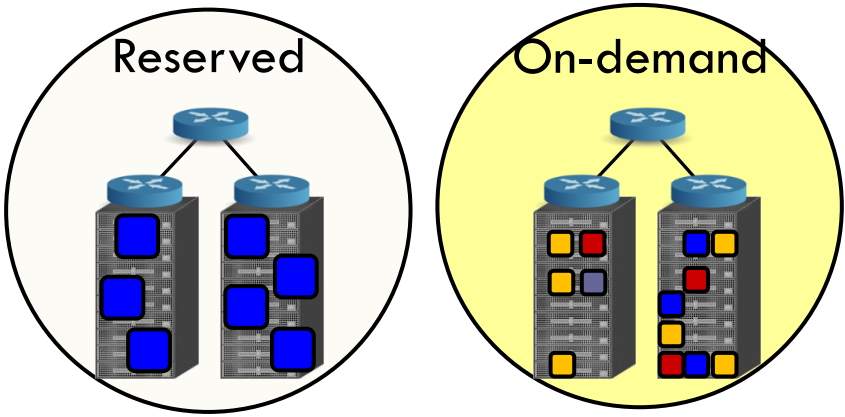
✔ Determine appropriate instance size/type

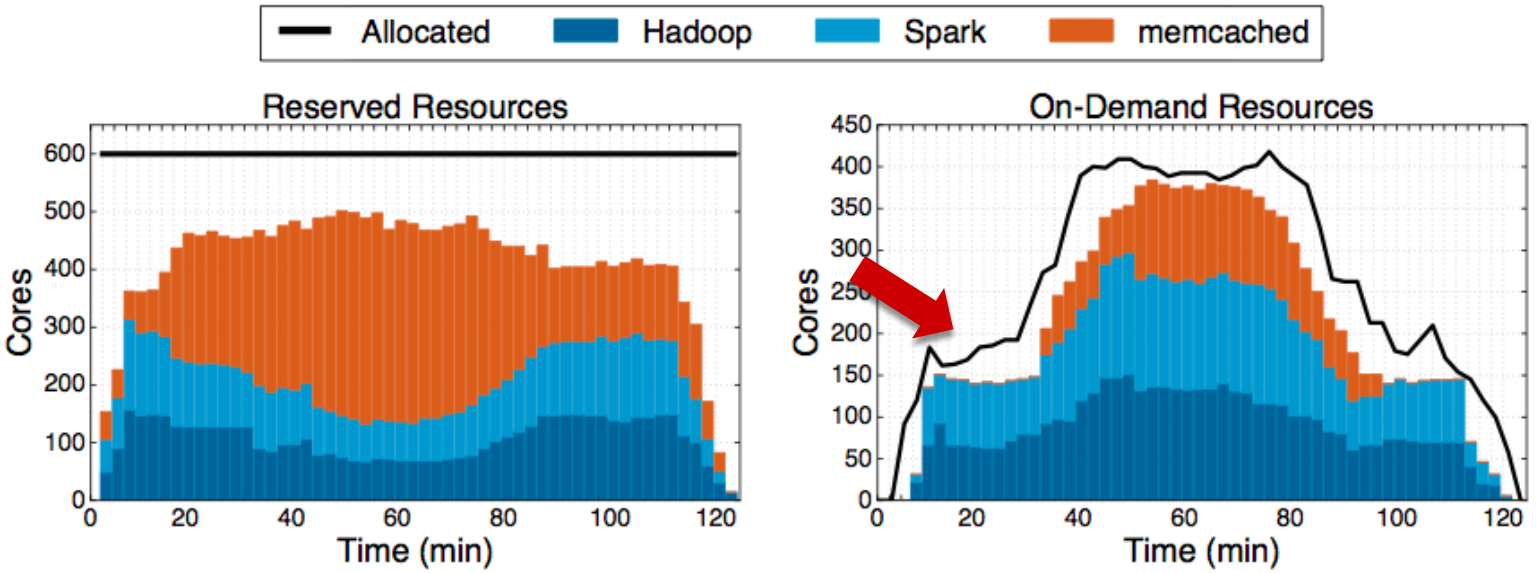
✔ Determine appropriate instance configuration

✗ Dynamically adjust allocation decisions at runtime

Adjusting Allocations at Runtime

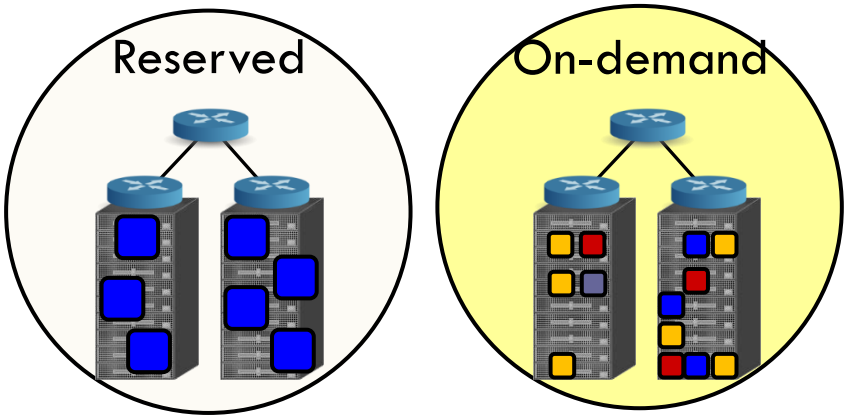
- Runtime: monitor app performance
  - Remove co-scheduled apps
  - Move to larger instance within cluster
  - Move to reserved cluster

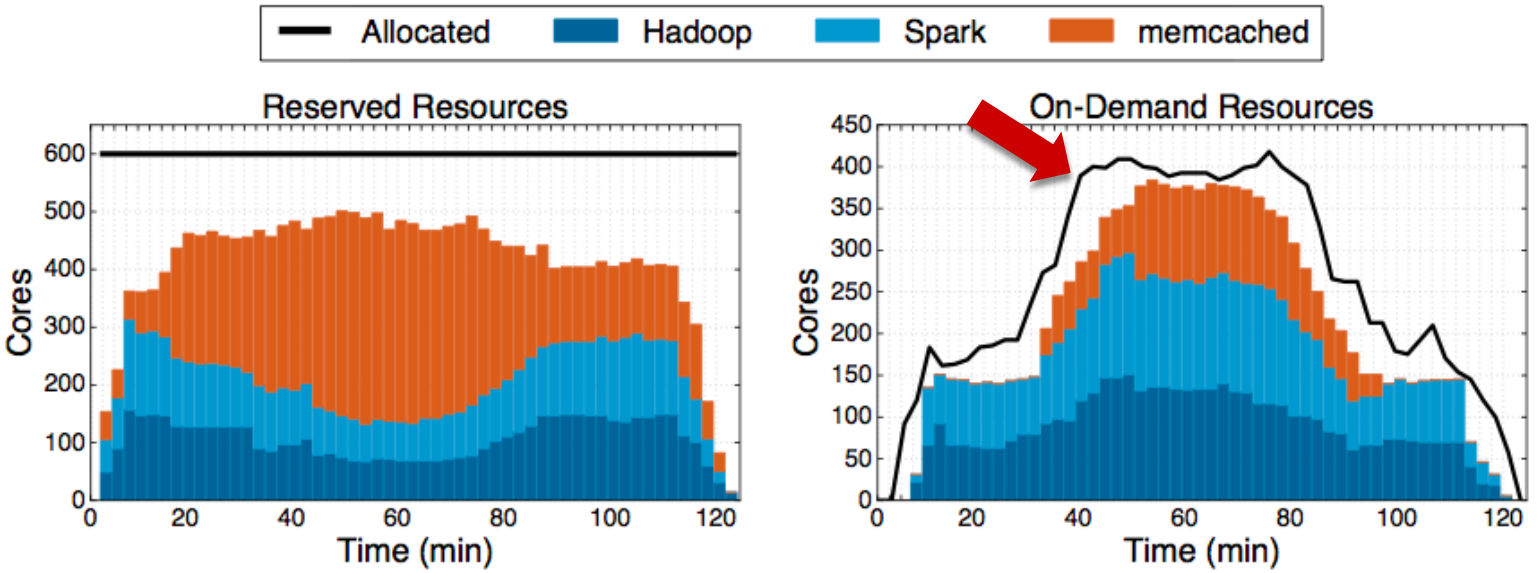




Adjusting Allocations at Runtime

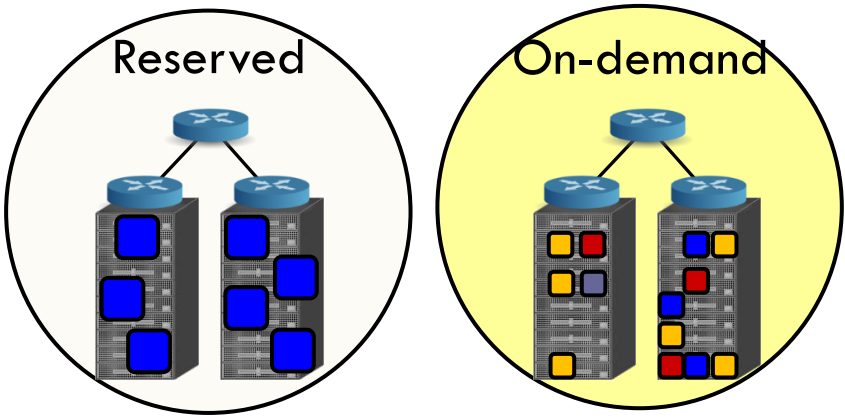
- Runtime: monitor app performance
  - Remove co-scheduled apps
  - Move to larger instance within cluster
  - Move to reserved cluster

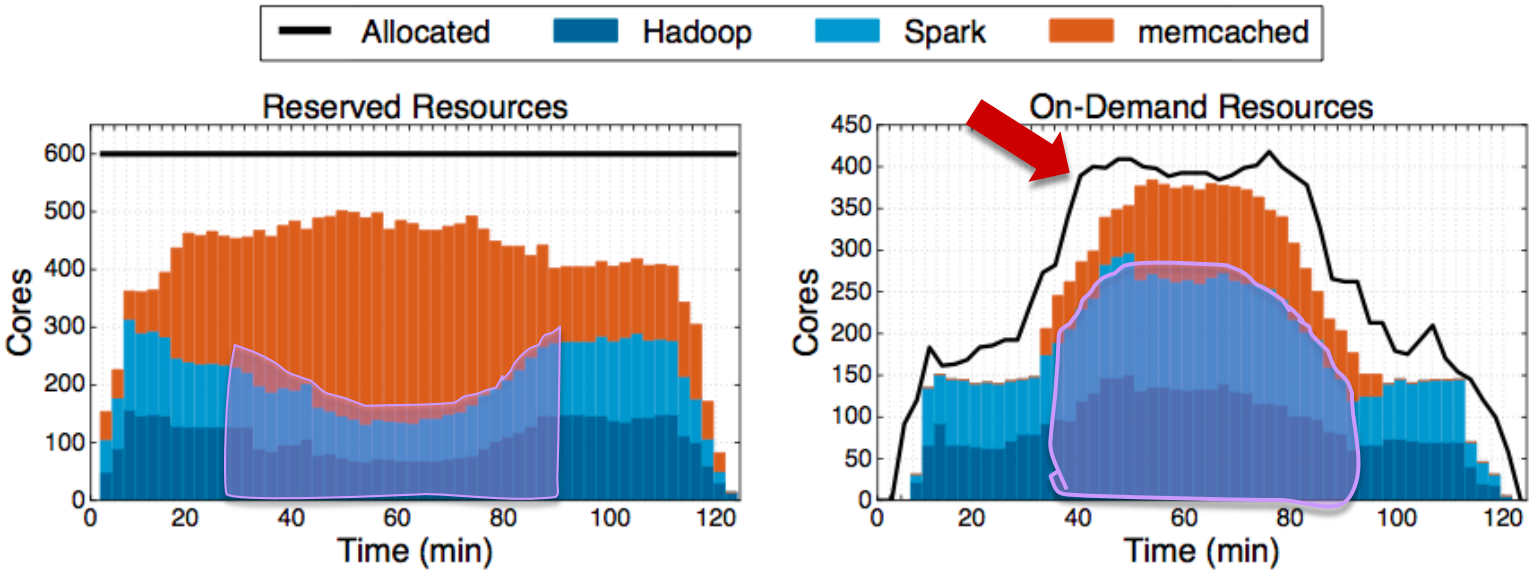




Adjusting Allocations at Runtime

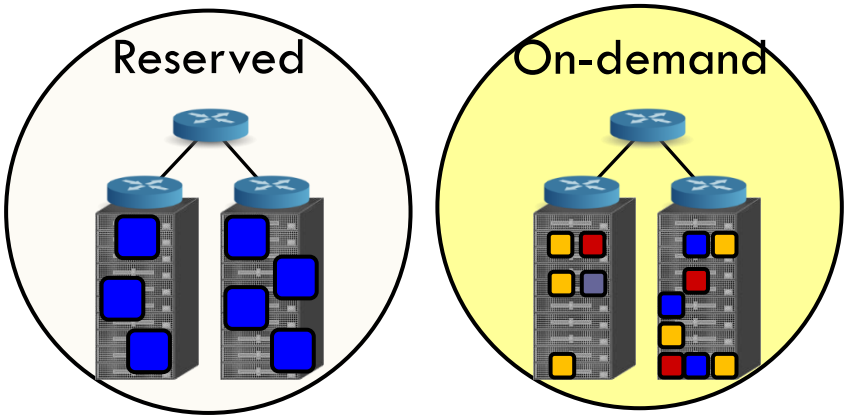
- Runtime: monitor app performance
  - Remove co-scheduled apps
  - Move to larger instance within cluster
  - Move to reserved cluster

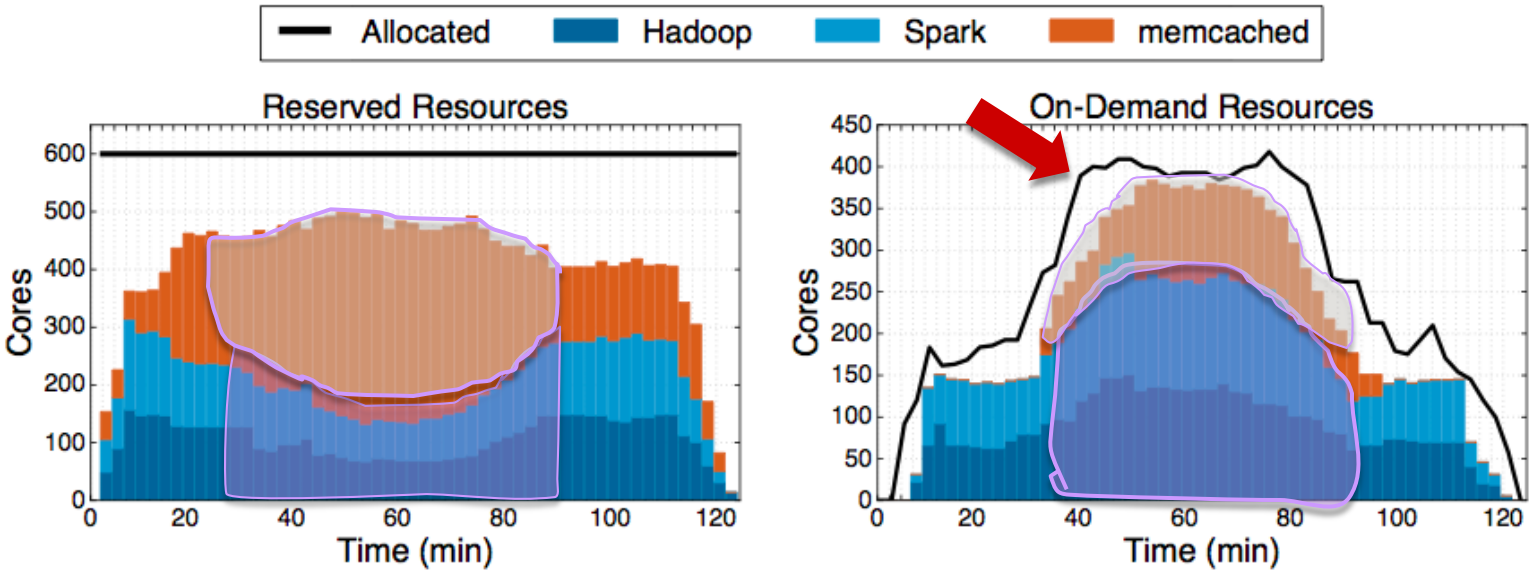




Adjusting Allocations at Runtime

- Runtime: monitor app performance
  - Remove co-scheduled apps
  - Move to larger instance within cluster
  - Move to reserved cluster





Conclusions

- HCloud: hybrid cloud provisioning (reserved & on-demand)
  - Account for interference sensitivity
  - Account for performance-cost tradeoffs
  - Adjust to load fluctuations
  - 2.1x better performance than on-demand, 50% lower cost than reserved
- See paper for:
  - Performance unpredictability analysis on EC2 and GCE
  - Sensitivity studies on system & app parameters
    - Spin-up overheads, cost, retention time, load, app characteristics, external load, …
  - Further workload scenarios

Questions??

- HCloud: hybrid cloud provisioning (reserved & on-demand)
  - Account for interference sensitivity
  - Account for performance-cost tradeoffs
  - Adjust to load fluctuations
  - 2.1x better performance than on-demand, 50% lower cost than reserved
- See paper for:
  - Performance unpredictability analysis on EC2 and GCE
  - Sensitivity studies on system & app parameters
    - Spin-up overheads, cost, retention time, load, app characteristics, external load, …
  - Further workload scenarios

Questions??

Thank you

Resource Efficiency