Improving Resource Efficiency in Cloud Computing

Christina Delimitrou

Stanford University

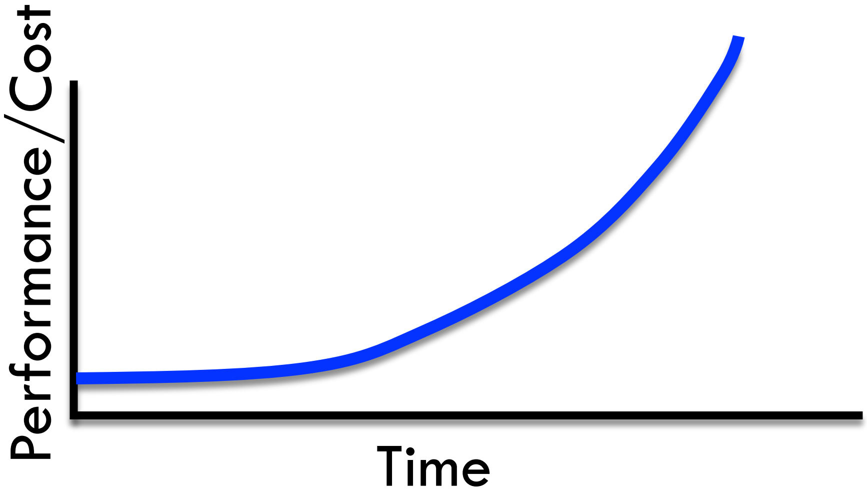
Defense – May 26th 2015

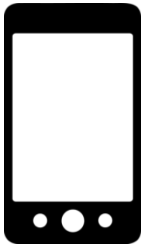
Resource efficiency is a first-order system constraint

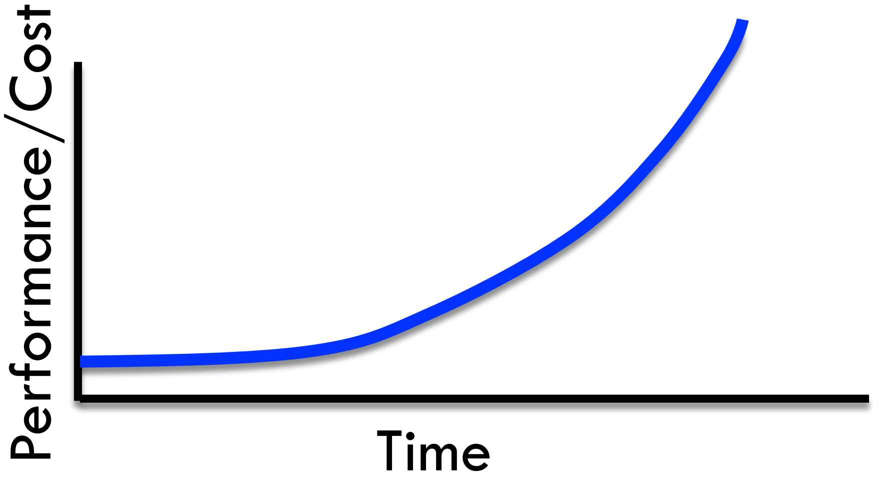
How efficiently do we utilize resources?

How efficiently do we design systems?

Why Care about Resource Efficiency?







~10K commodity servers

Sophisticated cluster managers

~10s MWatts

$100,000,000s

Private clouds:

- Google, Microsoft, Twitter, eBay

Public clouds:

- Amazon EC2, Windows Azure, GCE

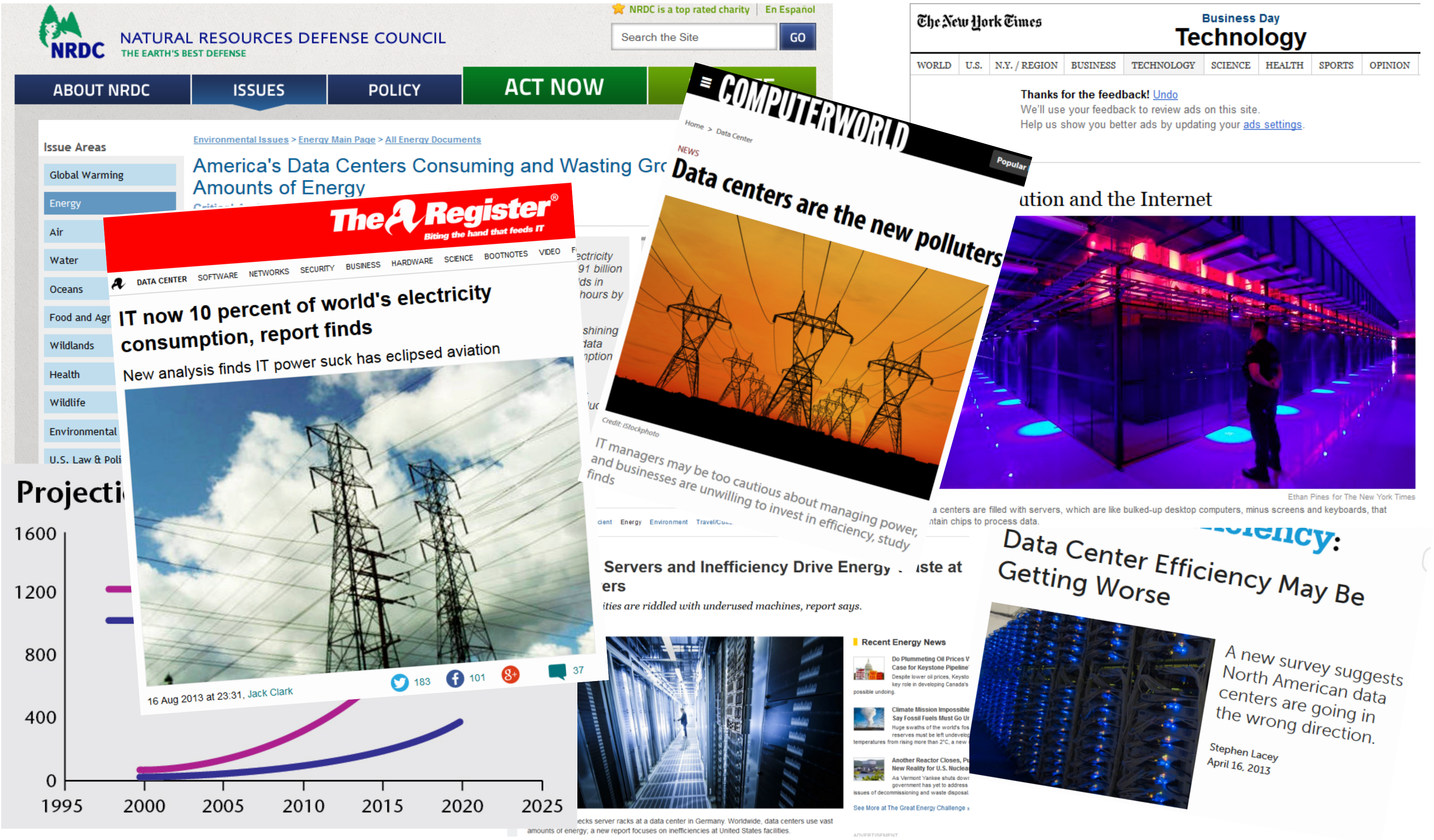
Google | google.com/datacenters

The Promise of Cloud Computing

- Flexibility
  - Provision and launch new services in seconds
- High performance
  - High throughput & low tail latency
- Cost effectiveness
  - Low capital & operational expenses

Cloud computing scalability:
high performance AND low cost

The Reality of Cloud Computing



NRDC NATURAL RESOURCES DEFENSE COUNCIL
THE EARTH'S BEST DEFENSE

NRDC is a top rated charity | En Español

Search the Site GO

ABOUT NRDC | ISSUES | POLICY | ACT NOW

Issue Areas

- Global Warming
- Energy
- Air
- Water
- Oceans
- Food and Agr
- Wildlands
- Health
- Wildlife
- Environmental
- U.S. Law & Poli

Environmental Issues > Energy Main Page > All Energy Documents

America's Data Centers Consuming and Wasting Gro
Amounts of Energy

ectricity
91 billion
ds in
hours by

shining
data
nption

The Register®
Biting the hand that feeds IT

DATA CENTER | SOFTWARE | NETWORKS | SECURITY | BUSINESS | HARDWARE | SCIENCE | BOOTNOTES | VIDEO

IT now 10 percent of world's electricity consumption, report finds

New analysis finds IT power suck has eclipsed aviation

16 Aug 2013 at 23:31, Jack Clark

183 | 101 | 37

Projecti

COMPUTERWORLD

Home > Data Center

Popular

NEWS

Data centers are the new polluters

Credit: iStockphoto

IT managers may be too cautious about managing power, and businesses are unwilling to invest in efficiency, study finds

The New York Times
Business Day
Technology

WORLD | U.S. | N.Y. / REGION | BUSINESS | TECHNOLOGY | SCIENCE | HEALTH | SPORTS | OPINION

Thanks for the feedback! Undo
We'll use your feedback to review ads on this site.
Help us show you better ads by updating your ads settings.

ution and the Internet

Ethan Pines for The New York Times

a centers are filled with servers, which are like bulked-up desktop computers, minus screens and keyboards, that ntain chips to process data.

cient | Energy | Environment | Travel/Cu

Servers and Inefficiency Drive Energy aste at ers

ities are riddled with underused machines, report says.

ecks server racks at a data center in Germany. Worldwide, data centers use vast nts of energy; a new report focuses on inefficiencies at United States facilities.

Recent Energy News

Do Plummeting Oil Prices V Case for Keystone Pipeline'
Despite lower oil prices, Keysto key role in developing Canada's possible undoing.

Climate Mission Impossible Say Fossil Fuels Must Go Ur
Huge swaths of the world's fos reserves must be left undevelop temperatures from rising more than 2°C, a new

Another Reactor Closes, Pu New Reality for U.S. Nuclea
As Vermont Yankee shuts dowr government has yet to address issues of decommissioning and waste disposal.

See More at The Great Energy Challenge »

ADVERTISEMENT

iciency:

Data Center Efficiency May Be Getting Worse

A new survey suggests North American data centers are going in the wrong direction.

Stephen Lacey
April 16, 2013

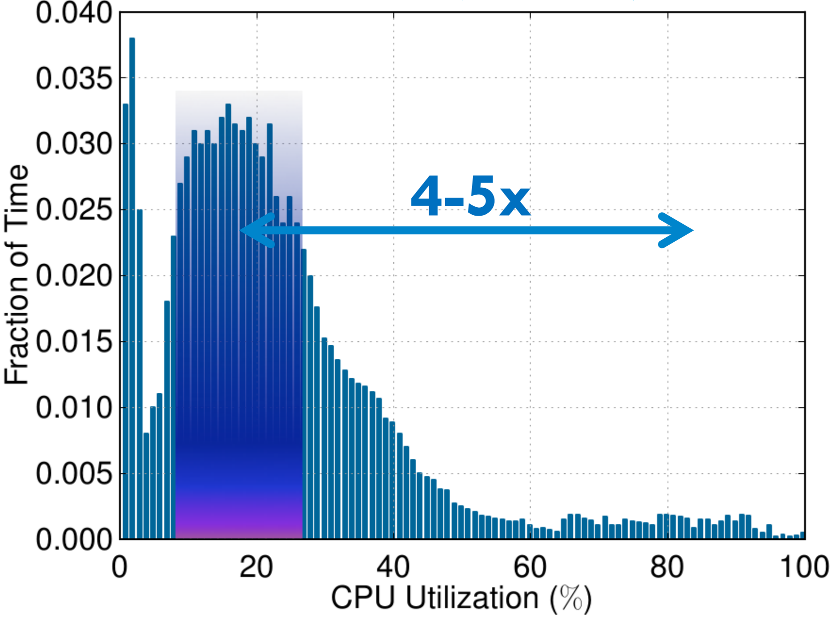
Scaling Datacenters

□ ~~Switch to commodity servers~~	One time trick
□ ~~Improve cooling/power distribution~~	< 10%
□ ~~Build more datacenters~~	>$300M per datacenter
□ ~~Add more servers~~	Power limit
□ ~~Rely on processor technology~~	End of voltage scaling

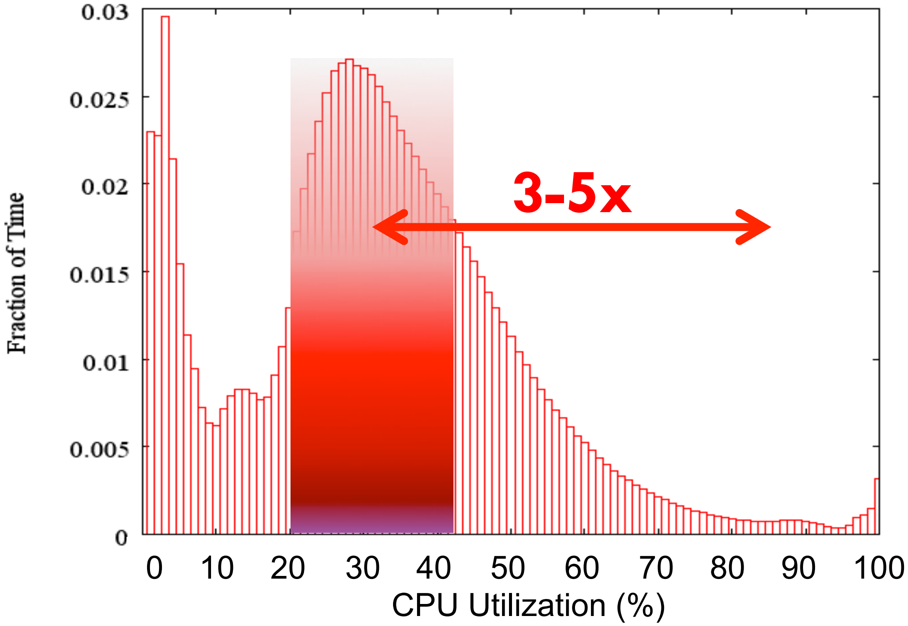
Use existing systems more efficiently

Datacenter Underutilization

Twitter (Mesos)[1]



Google (Borg)[2]



[1] C. Delimitrou and C. Kozyrakis. Quasar: Resource-Efficient and QoS-Aware Cluster Management, ASPLOS 2014.

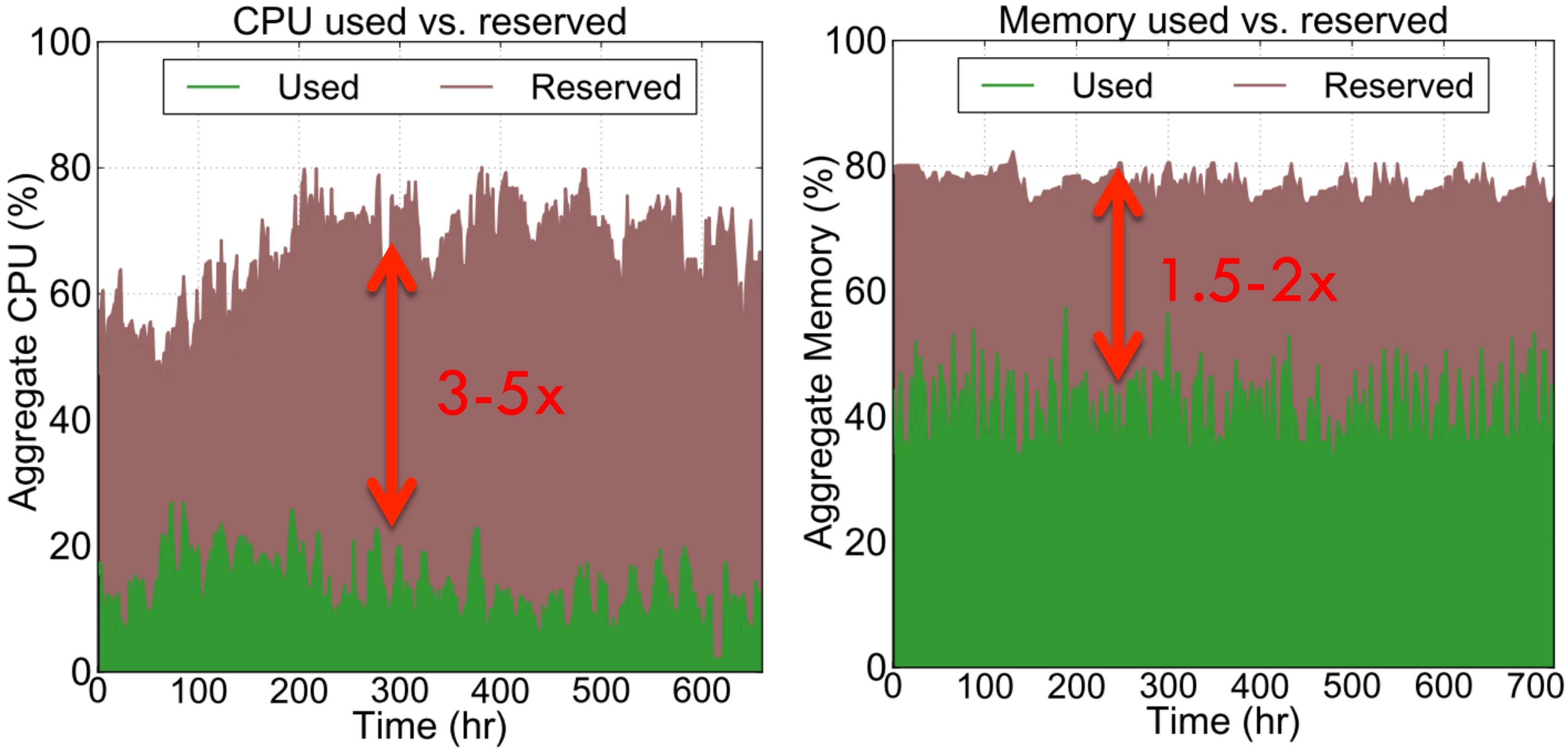
[2] L. A. Barroso, U. Holzle. The Datacenter as a Computer, 2013.

Datacenter Underutilization…

~~Is the cluster manager's fault~~

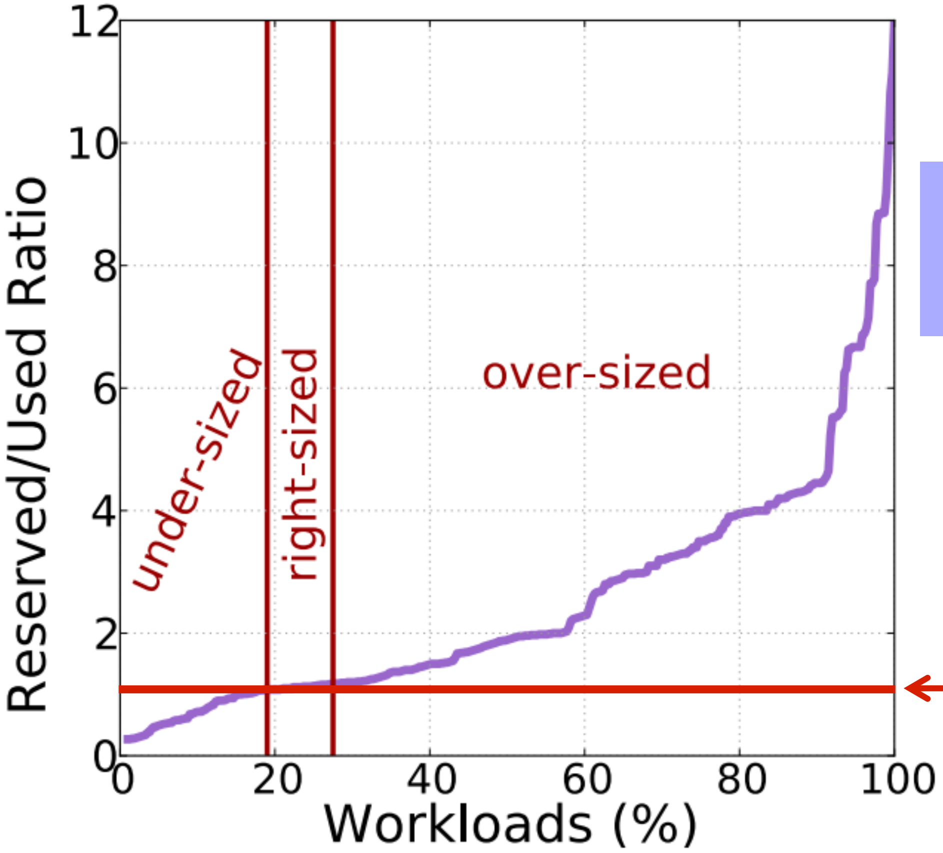
Is the user's fault!

Reserved vs. Used Resources



- □ Twitter: up to 5x CPU & up to 2x memory overprovisioning

Reserved vs. Used Resources



~25,000 jobs
936 distinct users

[ASPLOS'14]

Reservation=Usage

□ 20% of job under-sized, **~70% of jobs over-sized**

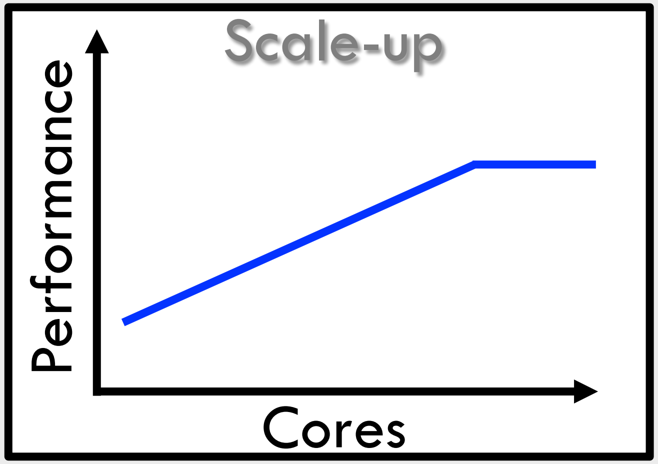
Datacenter Underutilization…

Is the user’s fault!

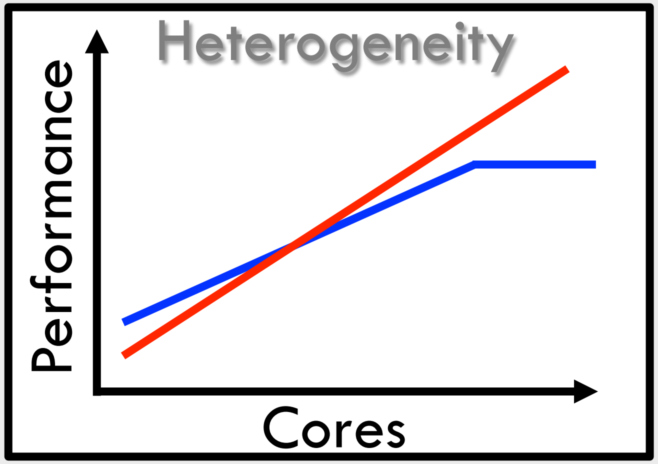
(not really…)

Resource Management is Hard

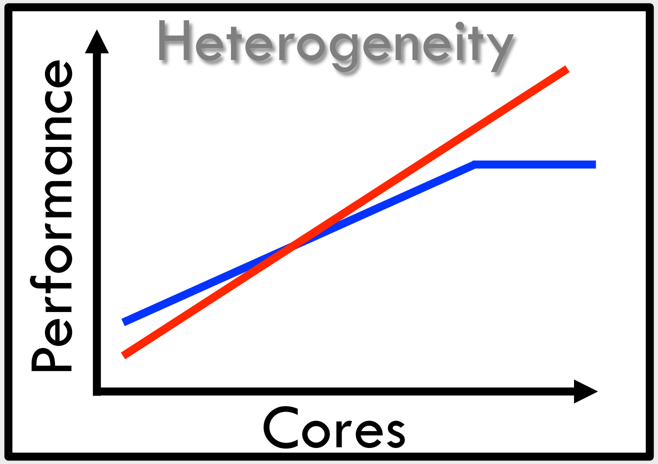
Performance Depends on



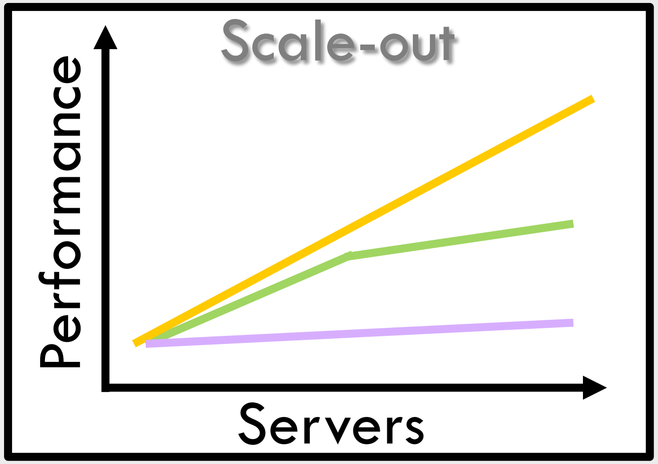
Performance Depends on



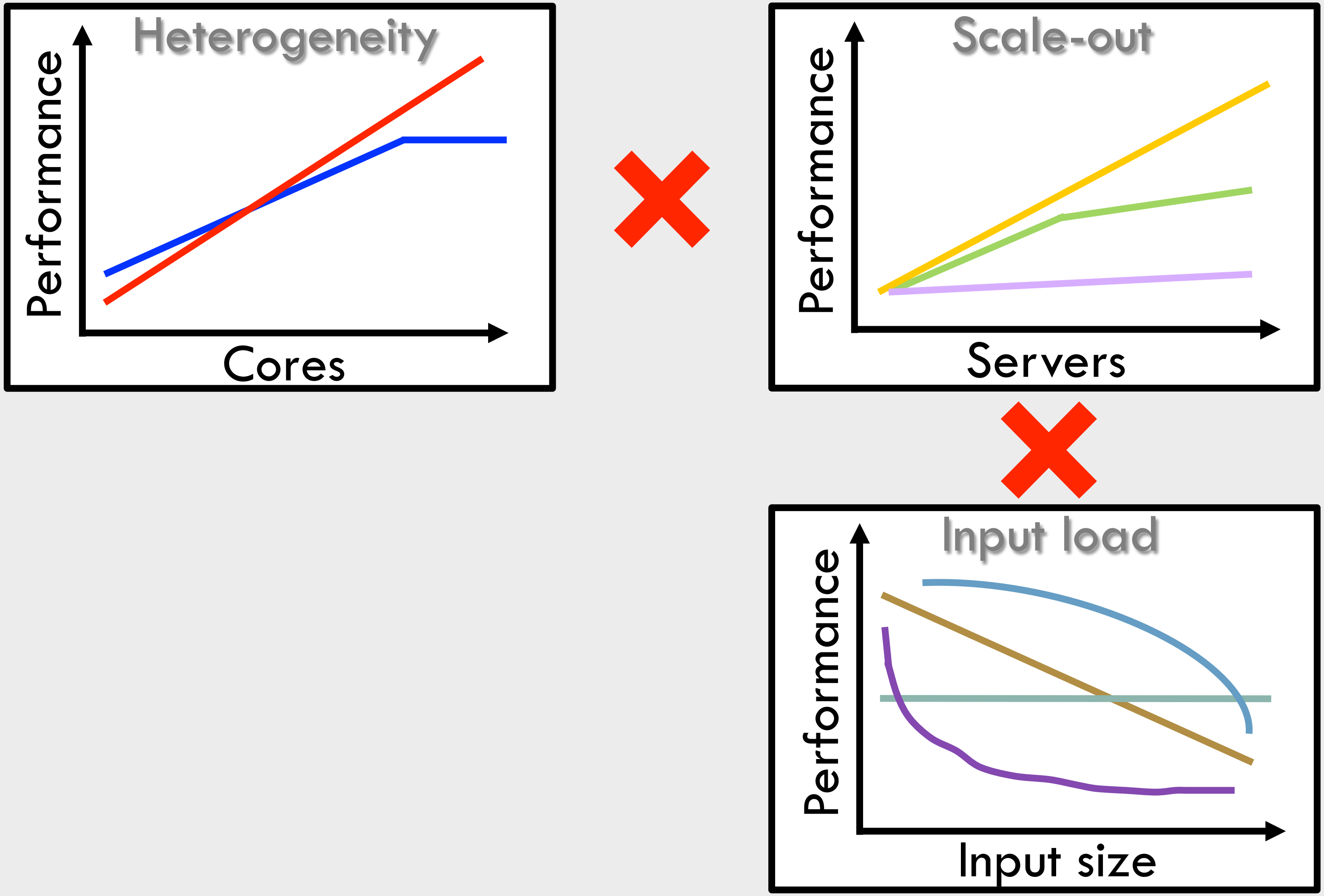
Performance Depends on



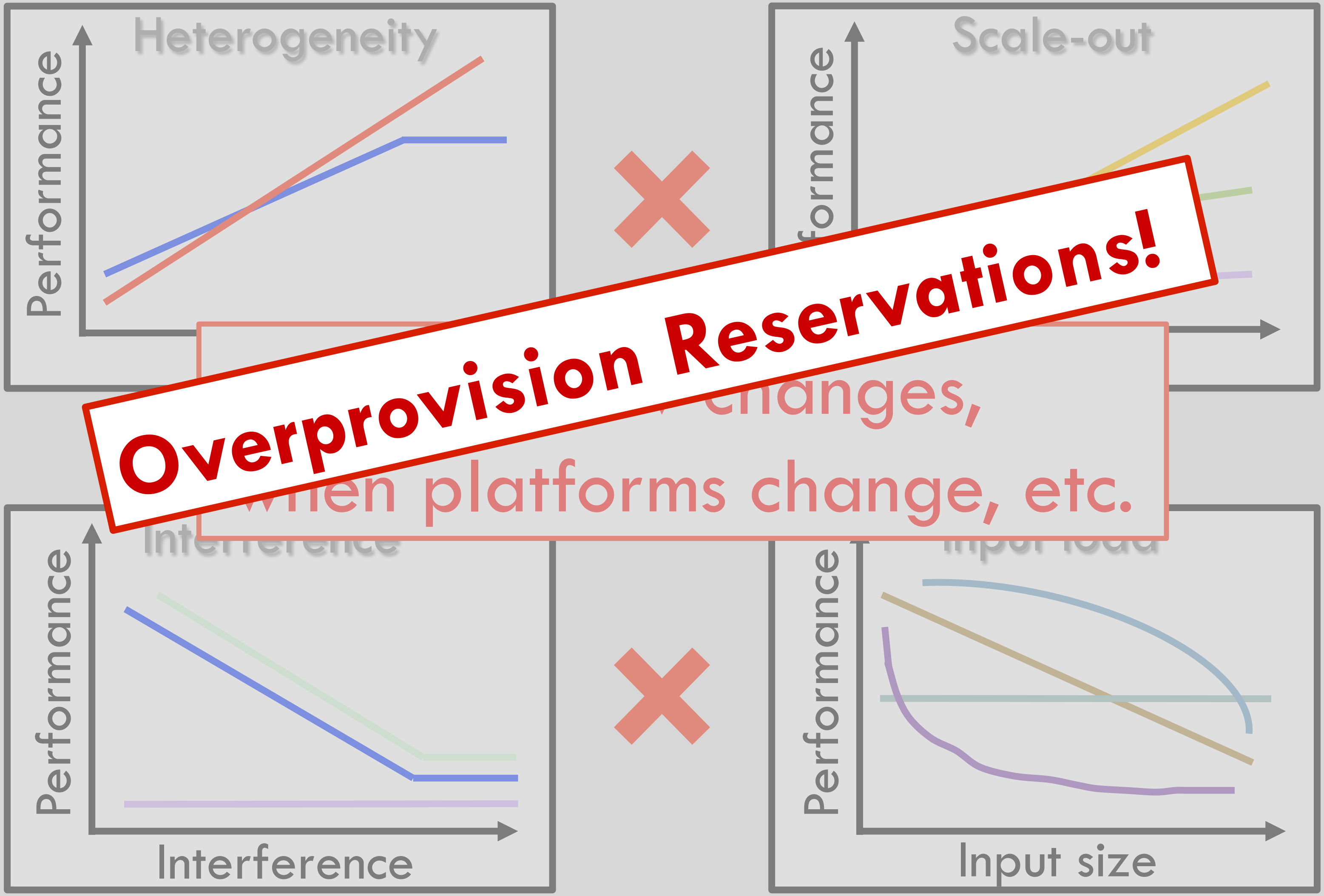
×



Performance Depends on



Performance Depends on



changes,
when platforms change, etc.

Overprovision Reservations!

Can we improve resource efficiency while preserving application QoS guarantees?

Potential: 3-5x efficiency; $10Ms in cost savings

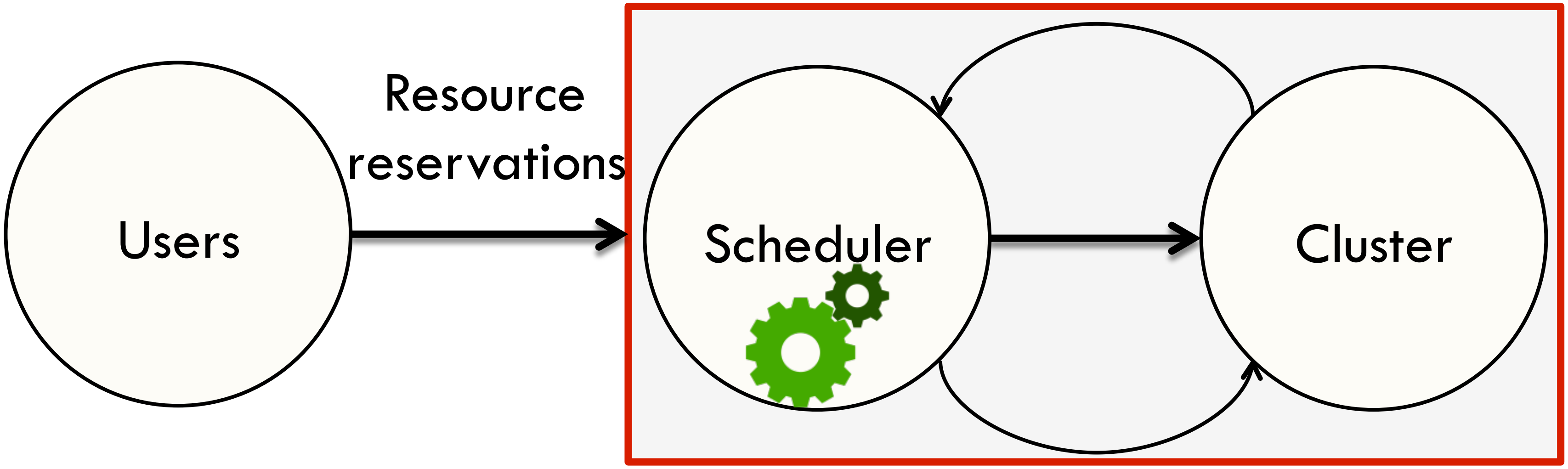
Requirements

- Automate resource management
  - Large, multi-dimensional space → Leverage big data
- General solution
  - Different application types (batch, latency-critical)
  - Different types of hardware
- Cross-layer design
  - Architecture → OS → Scheduler → Application design

Contributions

Contributions

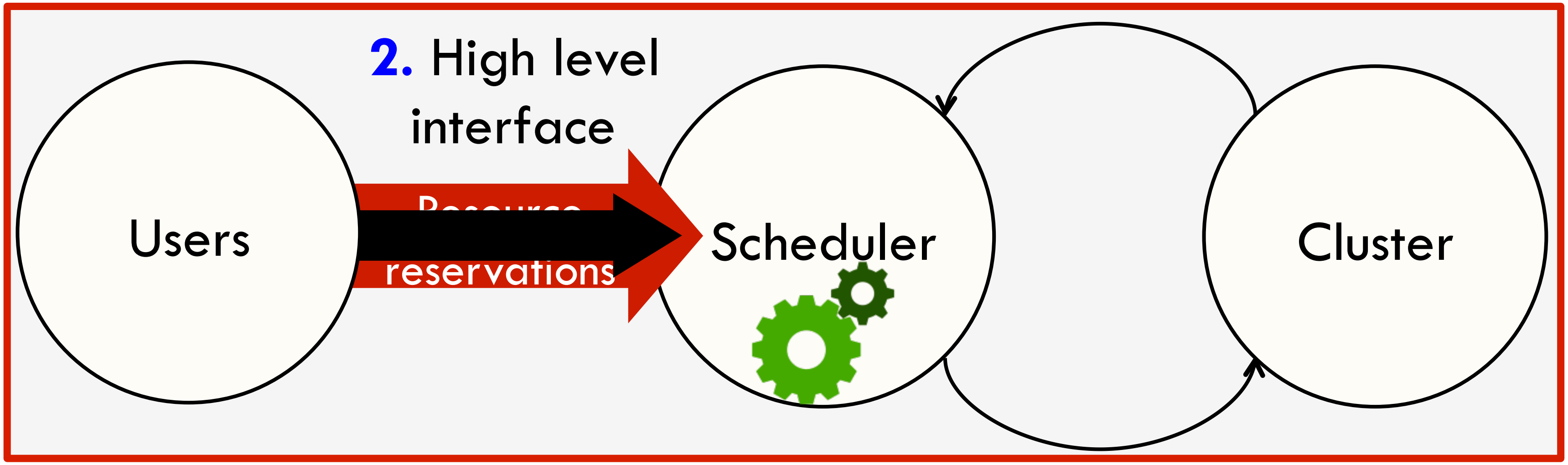
Paragon [ASPLOS'13, TopPicks'14] [IISWC'13]



1. Practical data mining

Contributions

Quasar [ASPLOS'14]



2. High level interface

1. Practical data mining

Contributions

Systems:

Application assignment: **Paragon** [ASPLOS'13, TopPicks'14, CAL'13, IISWC'13]

Cluster management: **Quasar** [ASPLOS'14]

Contributions

Systems:

Application assignment:	**Paragon** [ASPLOS'13, TopPicks'14], **iBench** [IISWC'13]
Cluster management:	**Quasar** [ASPLOS'14]
Scalable scheduling:	**Tarcil** [SOCC'15]

Contributions

Systems:

Application assignment:	**Paragon** [ASPLOS'13, TopPicks'14], **iBench** [IISWC'13]
Cluster management:	**Quasar** [ASPLOS'14]
Scalable scheduling:	**Tarcil** [SOCC'15]
Cloud provisioning:	**Hybrid Cloud** [in submission]

Contributions

Systems:

Application assignment:	**Paragon** [ASPLOS'13, TopPicks'14], **iBench** [IISWC'13]
Cluster management:	**Quasar** [ASPLOS'14]
Scalable scheduling:	**Tarcil** [SOCC'15]
Cloud provisioning:	**Hybrid Cloud** [in submission]
Admission control:	**ARQ** [ICAC'13]

Contributions

Systems:

Application assignment: **Paragon** [ASPLOS'13, TopPicks'14], **iBench** [IISWC'13]

Cluster management: **Quasar** [ASPLOS'14]

Scalable scheduling: **Tarcil** [SOCC'15]

Cloud provisioning: **Hybrid Cloud** [in submission]

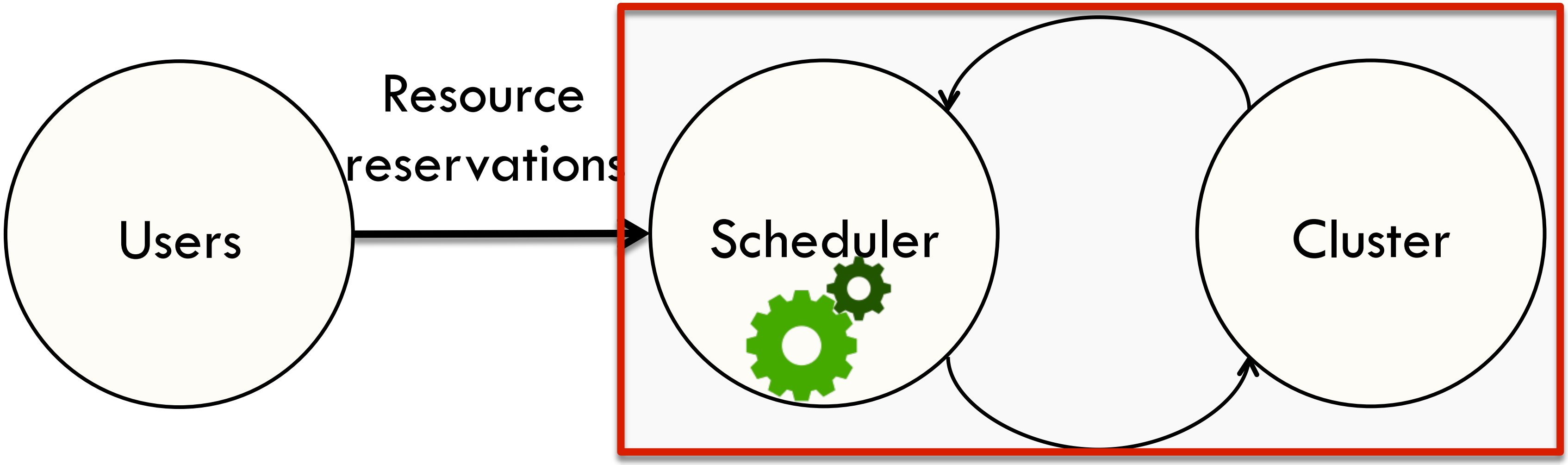
Admission control: **ARQ** [ICAC'13]

Datacenter application modeling:

ECHO [IISWC'12], **Storage application modeling** [CAL'12, IISWC'11, ISPASS'11]

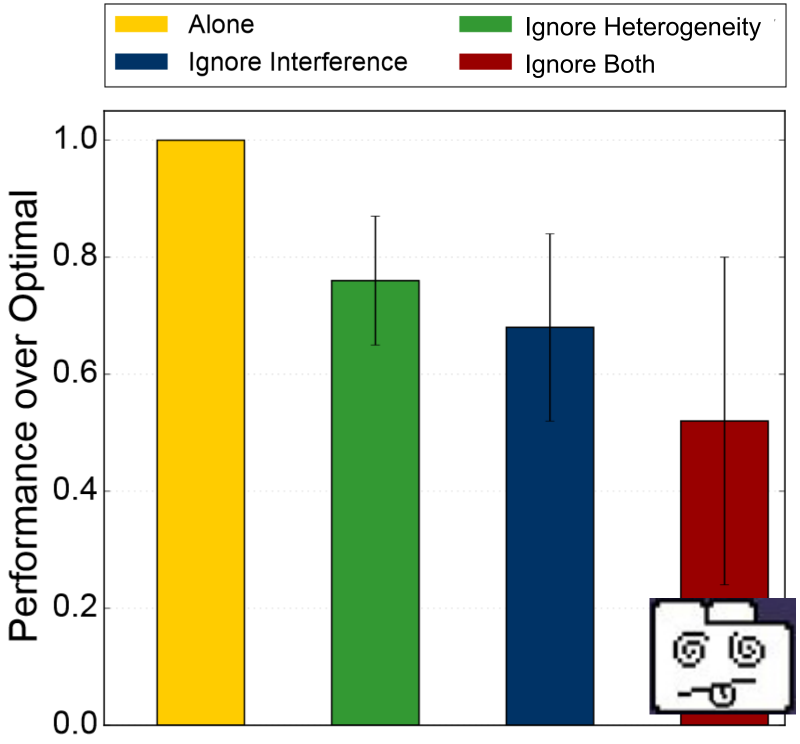
Paragon

[ASPLOS'13, TopPicks'14]



Practical data mining techniques

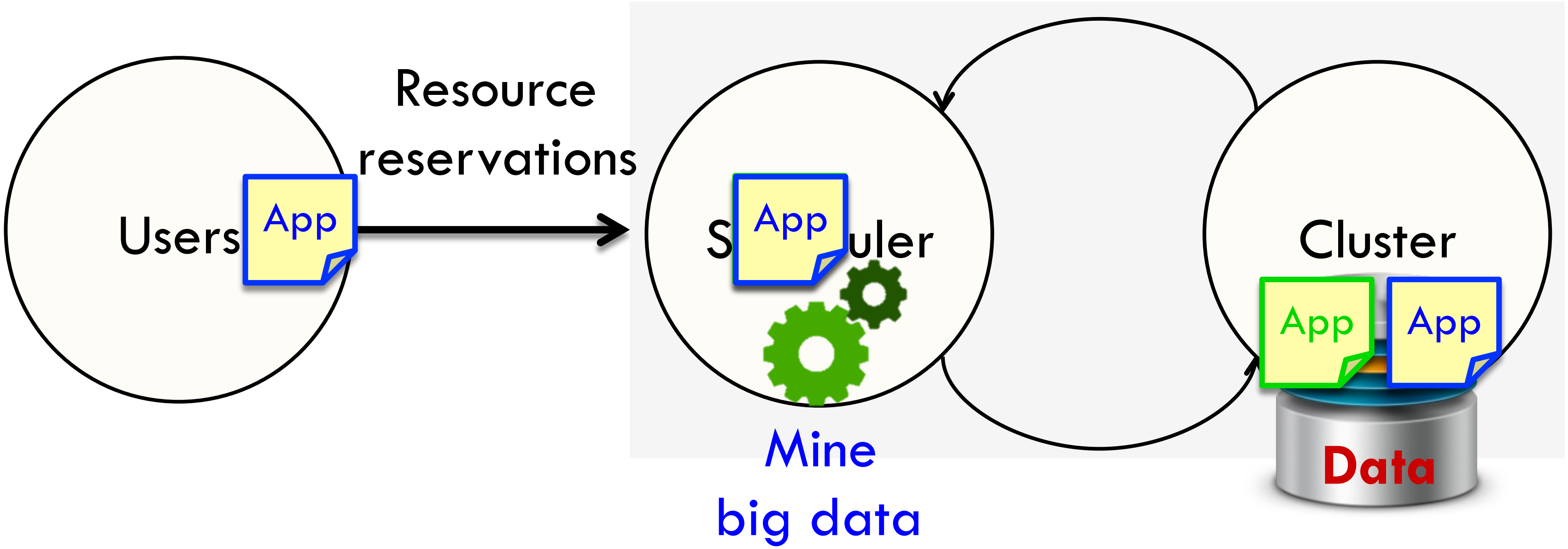
Heterogeneity & Interference Matter



- Heterogeneity
  - DCs provisioned over 15 years
  - Multiple server generations & configurations
- Interference
  - Apps contend on shared resources
    - CPU & cache hierarchy
    - Memory system
    - Storage & network I/O

Extracting Resource Preferences

- Naïve: exhaustive characterization
  - ~10-20 platforms x 1,000 apps



- Looks like a recommendation problem

Recommendation Systems

- Content-based systems:
  - Description of items (keywords, feature vector, etc.)
  - Profile of user preferences (history, model, user-system interaction, etc.)

- Collaborative filtering:
  - Uncover similarities between users and items
  - No need to know item features or explicit user preferences in advance

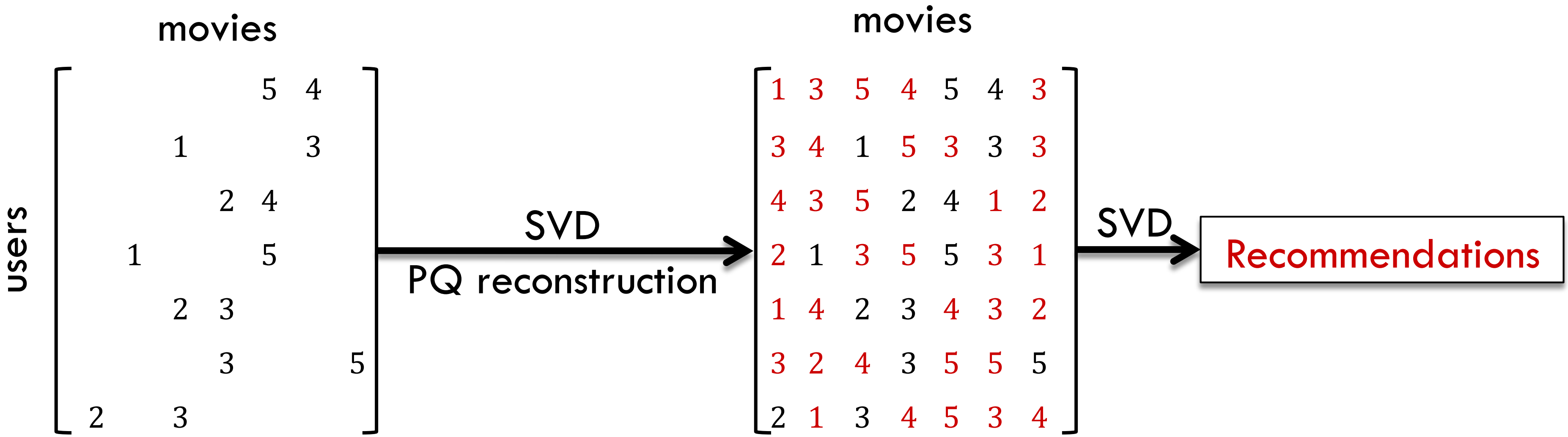
Recommendation Systems

- Content-based systems:
  - Description of items (keywords, feature vector, etc.)
  - Profile of user preferences (history, model, user-system interaction, etc.)

- Collaborative filtering:
  - Uncover similarities between users and items
  - No need to know item features or explicit user preferences in advance

Something familiar…

- Collaborative filtering – similar to Netflix Challenge system
  - Singular Value Decomposition (SVD) + PQ reconstruction (SGD)



movies

users

$$\begin{bmatrix} & & & & 5 & 4 & \\ & & 1 & & & 3 & \\ & & & 2 & 4 & & \\ & 1 & & & 5 & & \\ & & 2 & 3 & & & \\ & & & 3 & & & 5 \\ 2 & & 3 & & & & \end{bmatrix} \xrightarrow[\text{PQ reconstruction}]{\text{SVD}} \begin{bmatrix} 1 & 3 & 5 & 4 & 5 & 4 & 3 \\ 3 & 4 & 1 & 5 & 3 & 3 & 3 \\ 4 & 3 & 5 & 2 & 4 & 1 & 2 \\ 2 & 1 & 3 & 5 & 5 & 3 & 1 \\ 1 & 4 & 2 & 3 & 4 & 3 & 2 \\ 3 & 2 & 4 & 3 & 5 & 5 & 5 \\ 2 & 1 & 3 & 4 & 5 & 3 & 4 \end{bmatrix} \xrightarrow{\text{SVD}} \boxed{\text{Recommendations}}$$

movies

Sparse utility matrix

Dense utility matrix

SVD

user

movie

rating (e.g., ★★★★)

$$
\begin{array}{c|cccc|}
 & m_1 & m_2 & \dots & m_n \\
\hline
u_1 & a_{11} & a_{12} & \dots & a_{1n} \\
u_2 & a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
u_m & a_{m1} & a_{m2} & \dots & a_{mn} \\
\hline
\end{array}
$$

$$=$$

$$
\begin{array}{c}
u_1 \\ \\ \dots \\ u_m
\end{array}
\begin{bmatrix}
u_{11} & \dots & u_{1r} \\
\vdots & \ddots & \vdots \\
u_{m1} & \dots & u_{mr}
\end{bmatrix}
\times
\begin{bmatrix}
\sigma_1 & \dots & 0 \\
\vdots & \ddots & \vdots \\
0 & \dots & \sigma_r
\end{bmatrix}
\times
\begin{array}{c}
\begin{array}{ccc} m_1 & \dots & m_n \end{array} \\
\begin{bmatrix}
v_{11} & \dots & v_{1r} \\
\vdots & \ddots & \vdots \\
v_{n1} & \dots & v_{nr}
\end{bmatrix}
\end{array}
$$

SVD

movie

user

rating (e.g.,★★★★)

$$
\begin{array}{c|cccc}
 & \mathtt{m_1} & \mathtt{m_2} & \dots & \mathtt{m_n} \\
\hline
\mathtt{u_1} & a_{11} & a_{12} & \dots & a_{1n} \\
\mathtt{u_2} & a_{21} & a_{22} & \dots & a_{2n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathtt{u_m} & a_{m1} & a_{m2} & \dots & a_{mn}
\end{array}
$$

$$=$$

$$
\begin{array}{c|ccc}
\mathtt{u_1} & u_{11} & \dots & u_{1r} \\
 & \vdots & \ddots & \vdots \\
\dots & & & \\
\mathtt{u_m} & u_{m1} & \dots & u_{mr}
\end{array}
\times
\begin{bmatrix}
\sigma_1 & \dots & 0 \\
\vdots & \ddots & \vdots \\
0 & \dots & \sigma_r
\end{bmatrix}
\times
\begin{array}{ccc}
\mathtt{m_1} & \dots & \mathtt{m_n} \\
\hline
v_{11} & \dots & v_{1r} \\
\vdots & \ddots & \vdots \\
v_{n1} & \dots & v_{nr}
\end{array}
$$

correlation of user to similarity concept

similarity concept

correlation of movie to similarity concept

Heterogeneity Classification

$$\begin{array}{c|ccccccc|}
 & \text{Movie 1} & \text{Movie 2} & \text{Movie 3} & \text{Movie 4} & \text{Movie 5} & \cdots & \text{Movie M} \\
\hline
\text{User A} & \star\star\star\star & & & \star\star\star\star\star & & & \\
\text{User B} & & & \star\star & & \star\star\star & & \\
\vdots & & & & & & & \\
\text{User N} & & \star\star\star\star & & & & & \star \\
\hline
\end{array}$$

Heterogeneity Classification

$$\begin{array}{lccccccc}
 & \text{Platform 1} & \text{Platform 2} & \text{Platform 3} & \text{Platform 4} & \text{Platform 5} & \cdots & \text{Platform M} \\
\text{User A} & \star\star\star\star & & & \star\star\star\star\star & & & \\
\text{User B} & & & \star\star & & \star\star\star & & \\
\vdots & & & & & & & \\
\text{User N} & & \star\star\star\star & & & & & \star
\end{array}$$

Heterogeneity Classification

$$\begin{array}{c|ccccccc|}
 & \text{Platform 1} & \text{Platform 2} & \text{Platform 3} & \text{Platform 4} & \text{Platform 5} & \cdots & \text{Platform M} \\
\text{App A} & \bigstar\bigstar\bigstar\bigstar & & & \bigstar\bigstar\bigstar\bigstar\bigstar & & & \\
\text{App B} & & & \bigstar\bigstar & & \bigstar\bigstar\bigstar & & \\
\vdots & & & & & & & \\
\text{App N} & & \bigstar\bigstar\bigstar\bigstar & & & & & \bigstar
\end{array}$$

Heterogeneity Classification

$$\begin{array}{c|ccccccc|} & \text{Platform 1} & \text{Platform 2} & \text{Platform 3} & \text{Platform 4} & \text{Platform 5} & \ldots & \text{Platform M} \\ \text{App A} & \text{1,500QPS} & & & \text{843QPS} & & & \\ \text{App B} & & & \text{458QPS} & & \text{946QPS} & & \\ \vdots & & & & & & & \\ \text{App N} & & \text{1,016QPS} & & & & & \text{186QPS} \end{array}$$

App performance

Heterogeneity Classification

$$\begin{array}{l|ccccccc|}
 & \text{Platform 1} & \text{Platform 2} & \text{Platform 3} & \text{Platform 4} & \text{Platform 5} & \ldots & \text{Platform M} \\
\hline
\text{App A} & \text{1,500QPS} & & & \text{843QPS} & & & \\
\text{App B} & & & & & & & \\
\vdots & & & & & & & \\
\text{App N} & & & & & & & \\
\hline
\end{array}$$

Profiled Performance

Inferred Performance

Heterogeneity Classification

$$
\begin{array}{c|ccccccc|}
 & \text{Platform 1} & \text{Platform 2} & \text{Platform 3} & \text{Platform 4} & \text{Platform 5} & \ldots & \text{Platform M} \\
\text{App A} & \text{1,500QPS} & \text{843QPS} & \text{675QPS} & \text{843QPS} & \text{1,786QPS} & \ldots & \text{8,675QPS} \\
\text{App B} & & & & & & & \\
\vdots & & & & & & & \\
\text{App N} & & & & & & &
\end{array}
$$

Profiled Performance

Inferred Performance

Heterogeneity Classification

	Platform 1	Platform 2	Platform 3	Platform 4	Platform 5	. . .	Platform M
App A	1,500QPS	843QPS	675QPS	843QPS	1,786QPS	. . .	8,675QPS
App B	987QPS	458QPS	773QPS	1,073QPS	986QPS	. . .	1,836QPS
⋮							
App N							

Profiled Performance

Inferred Performance

Heterogeneity Classification

	Platform 1	Platform 2	Platform 3	Platform 4	Platform 5	. . .	Platform M
App A	1,500QPS	843QPS	675QPS	843QPS	1,786QPS	. . .	8,675QPS
App B	987QPS	458QPS	773QPS	1,073QPS	986QPS	. . .	1,836QPS
⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮
App N	9,893QPS	7,686QPS	786QPS	1,118QPS	997QPS	. . .	1,354QPS

Performance depends on app type:
QPS, completion time, IPC, …

Profiled Performance

Inferred Performance

Interference Classification

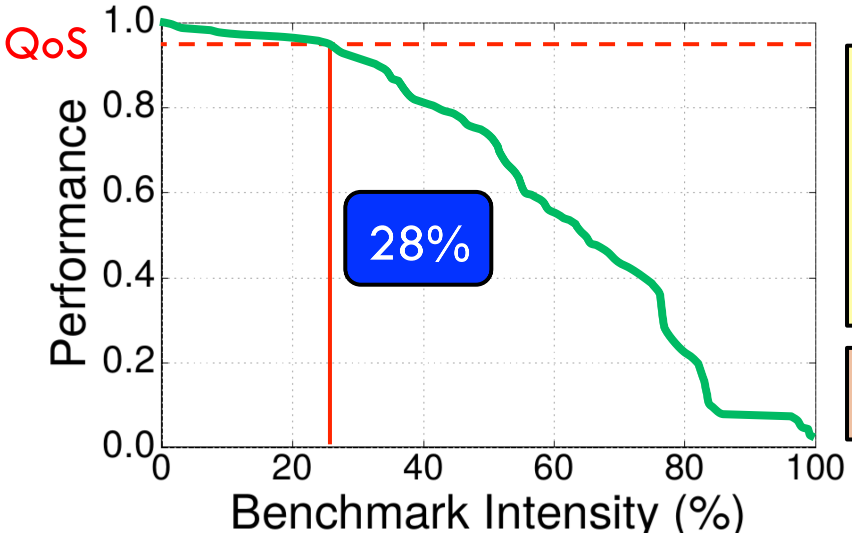
$$\begin{array}{c|ccccccc|}
 & \text{L1-i \$} & \text{LLC} & \text{Mem bw} & \text{CPU Int} & \text{I/O bw} & \cdots & \text{Net bw} \\
\text{App A} & 95 & 81 & 7 & 56 & 43 & \cdots & 100 \\
\text{App B} & 92 & 4 & 14 & 18 & 81 & \cdots & 78 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
\text{App N} & 45 & 49 & 56 & 11 & 99 & \cdots & 54
\end{array}$$

Profiled Sensitivity

Inferred Sensitivity

Measuring Interference Sensitivity

- □ Cross-application profiling: infeasible
- □ Measuring in hardware: platform-dependent & inaccurate
- □ **iBench**[1]: set of microbenchmarks of tunable intensity



Increase intensity until the application violates QoS (tolerated interference)

Generated interference?

[1] C. Delimitrou and C. Kozyrakis. "iBench: Quantifying Interference for Datacenter Applications" [IISWC'13]

Why SVD?

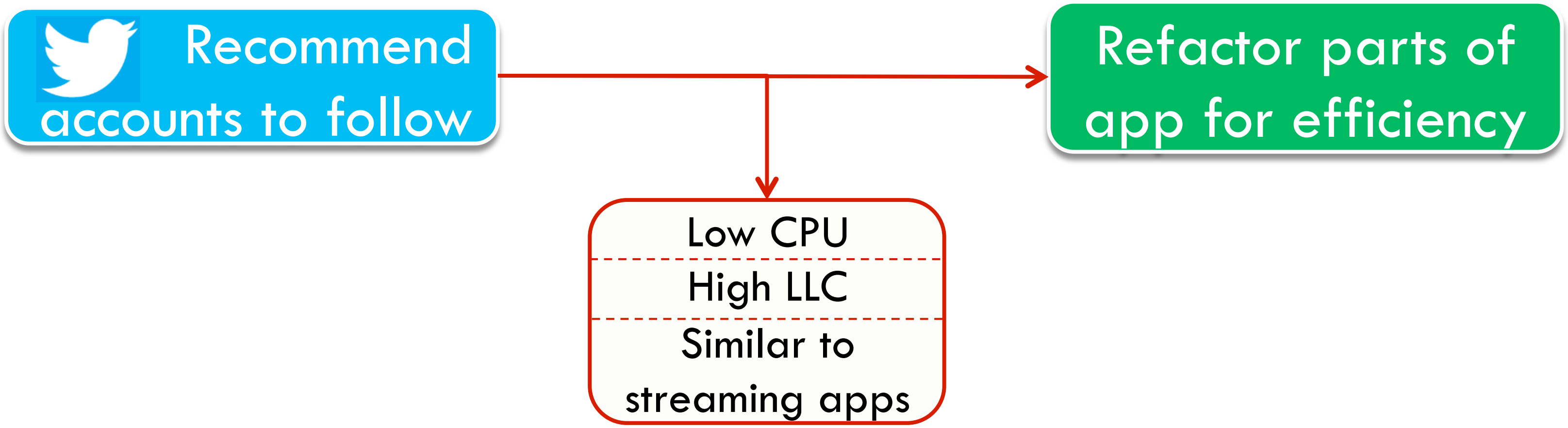
SVD+SGD: Low reconstruction error

Simple, fast, scalable ($O(\min(m^2n, n^2m))$)

Offer insight on similarities

Apps that benefit from high CPU frequency

Apps similar in I-cache are also similar in branch behavior



Recommend accounts to follow

Refactor parts of app for efficiency

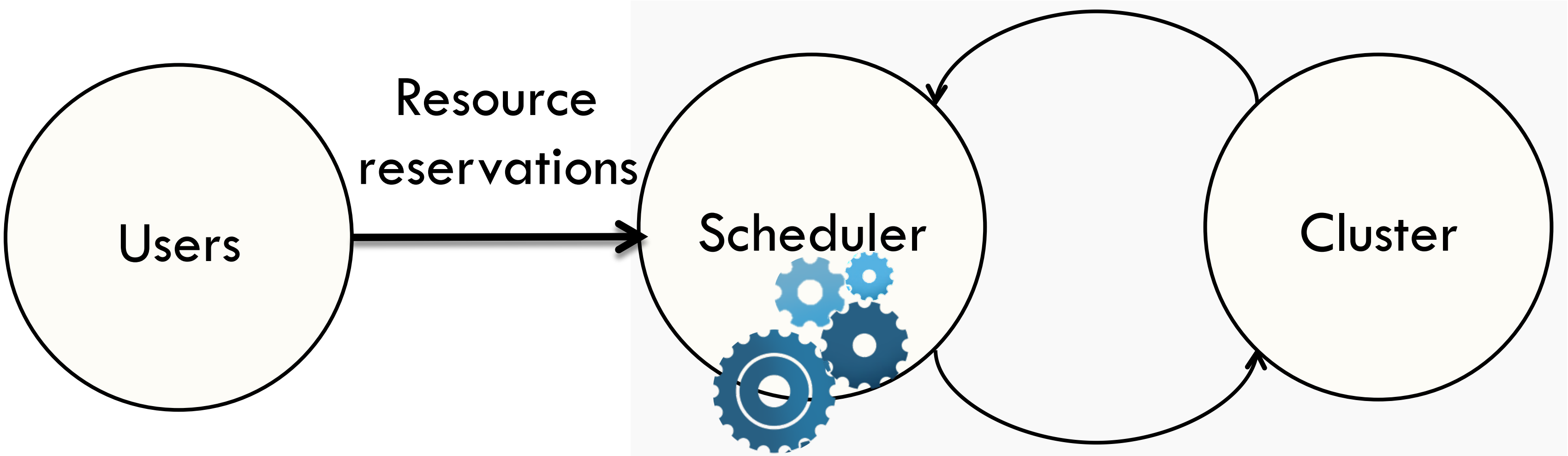
Low CPU

High LLC

Similar to streaming apps

Greedy Resource Selection

- Select servers that:
  - Can tolerate the interference of new application
  - Generate interference the new application can tolerate
  - Have appropriate platform configuration



Evaluation

- 1,000 EC2 servers
  - 14 different server configurations
  - 2 vCPU to 16 vCPU instances

- 5,000 applications
  - SPEC, PARSEC, SPLASH-2, BioParallel, Minebench, SpecWeb, Hadoop benchmarks

- Objectives:
  - High application performance
  - High resource utilization

Validation

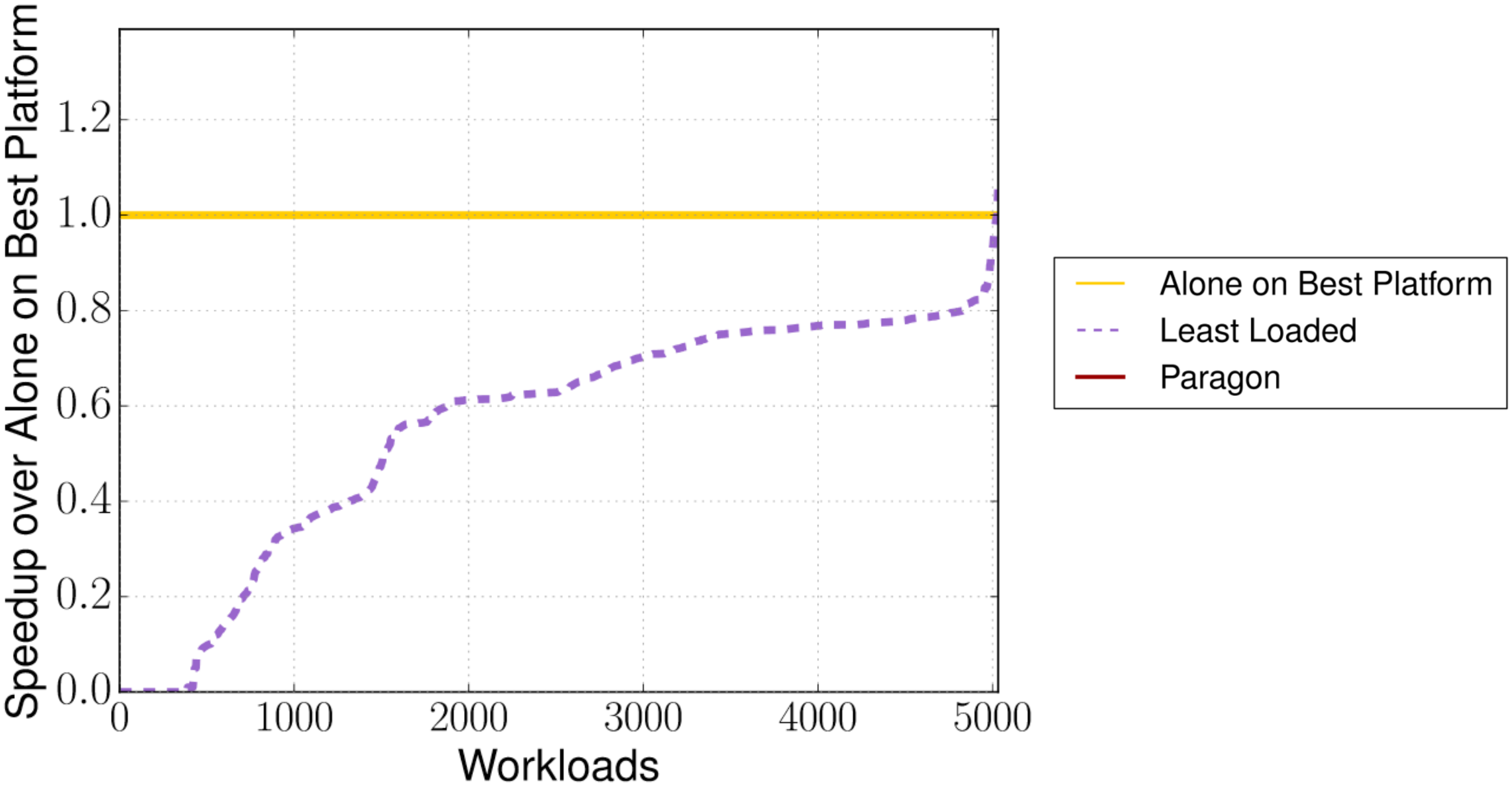
- 1,000 servers
- 5,000 applications
- Start with zero knowledge

3.5%

Classification Engine	Metric	Applications (%)		
		CPU-bound	Memory-bound	I/O-bound
Heterogeneity	Avg estimation error	3.1%	3.6%	4.1%
Interference	Avg estimation error	3.7%	3.5%	5.1%

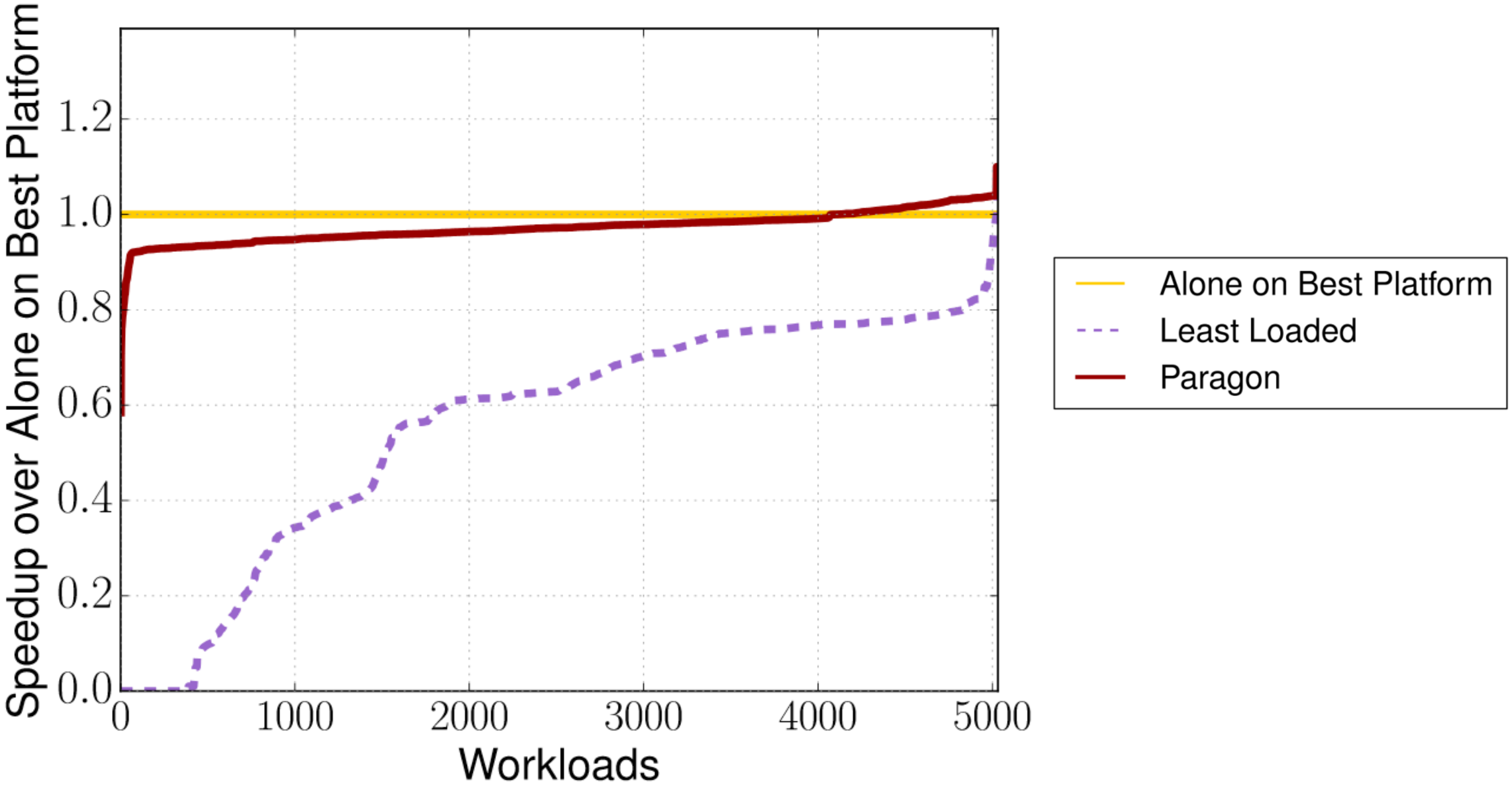
3.8%

Evaluation: Performance



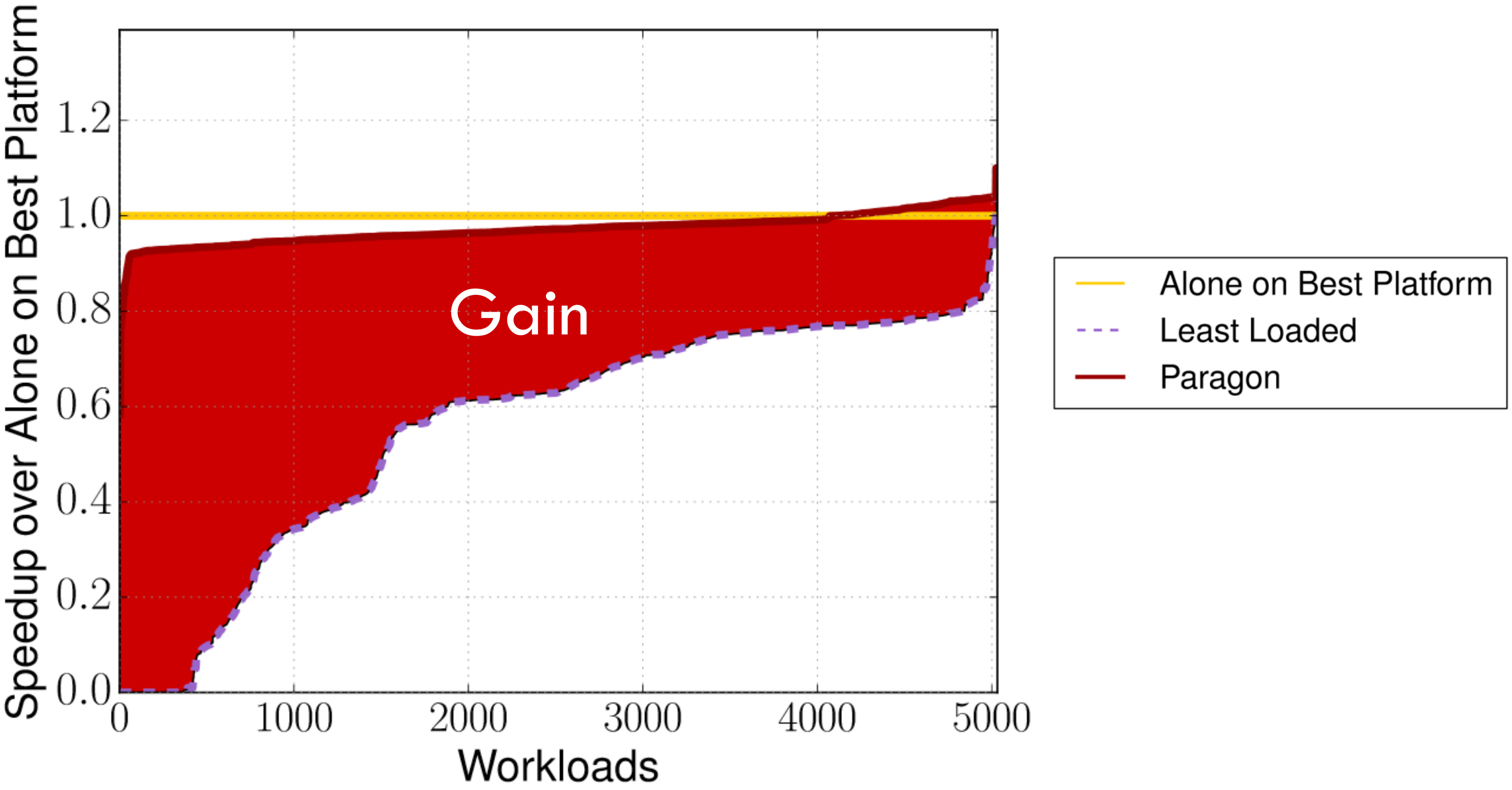
- Least loaded scheduler (common practice today)
  - Violates QoS for 97% of workloads

Evaluation: Performance



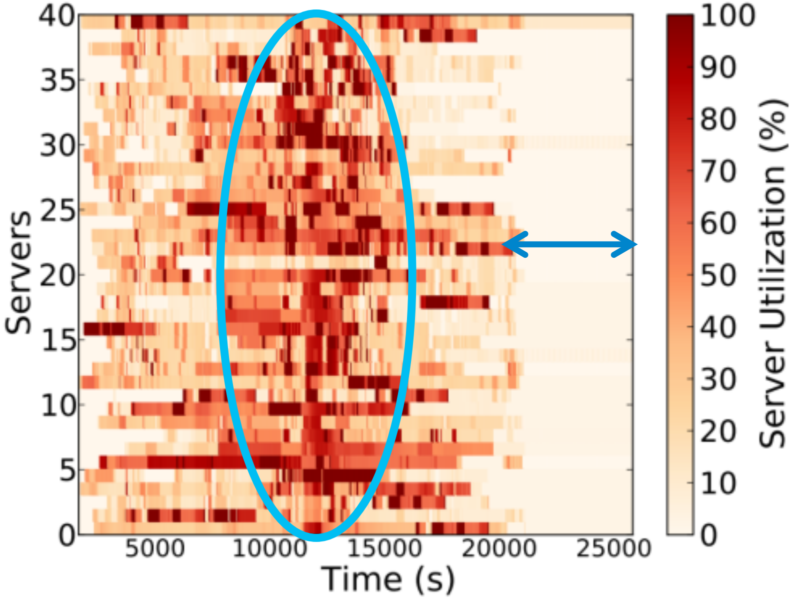
- Paragon preserves QoS for 71% of workloads
- Bounds degradation to less than 10% for 90% of workloads

Evaluation: Performance

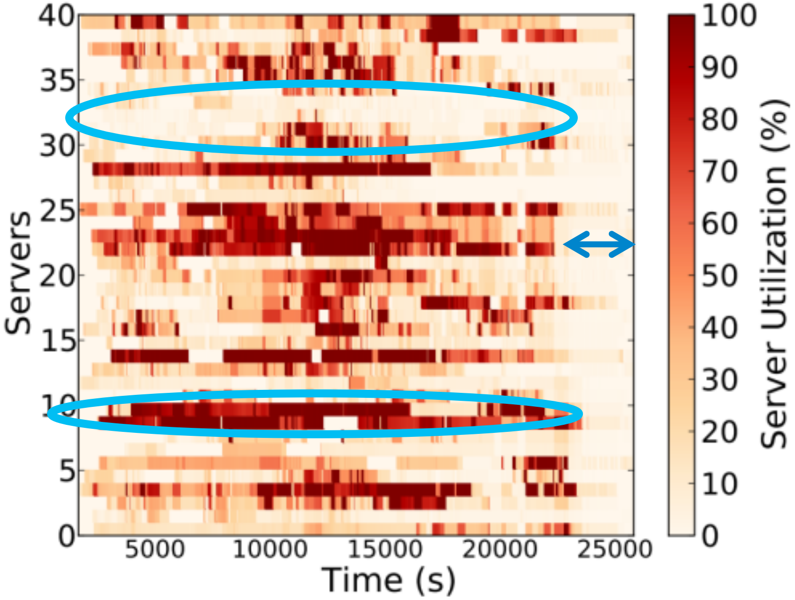


- □ Paragon preserves QoS for 71% of workloads
- □ Bounds degradation to less than 10% for 90% of workloads

Evaluation: System Utilization



Paragon



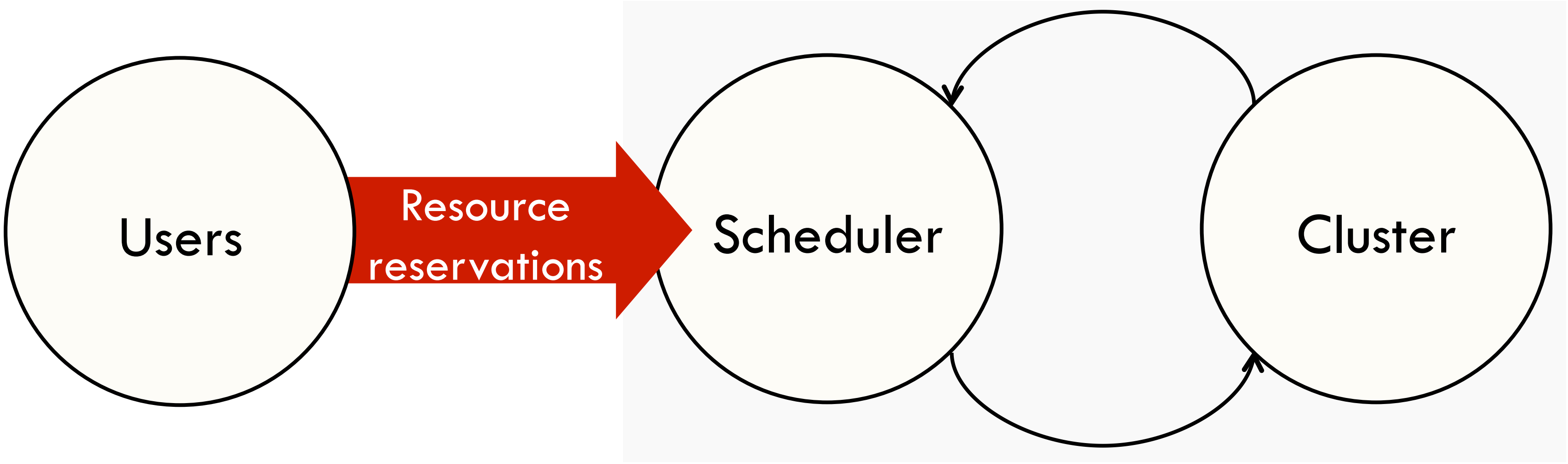
Least-Loaded (LL)

- Utilization increases from 19% to 58%

Are We Done?

A Larger Problem

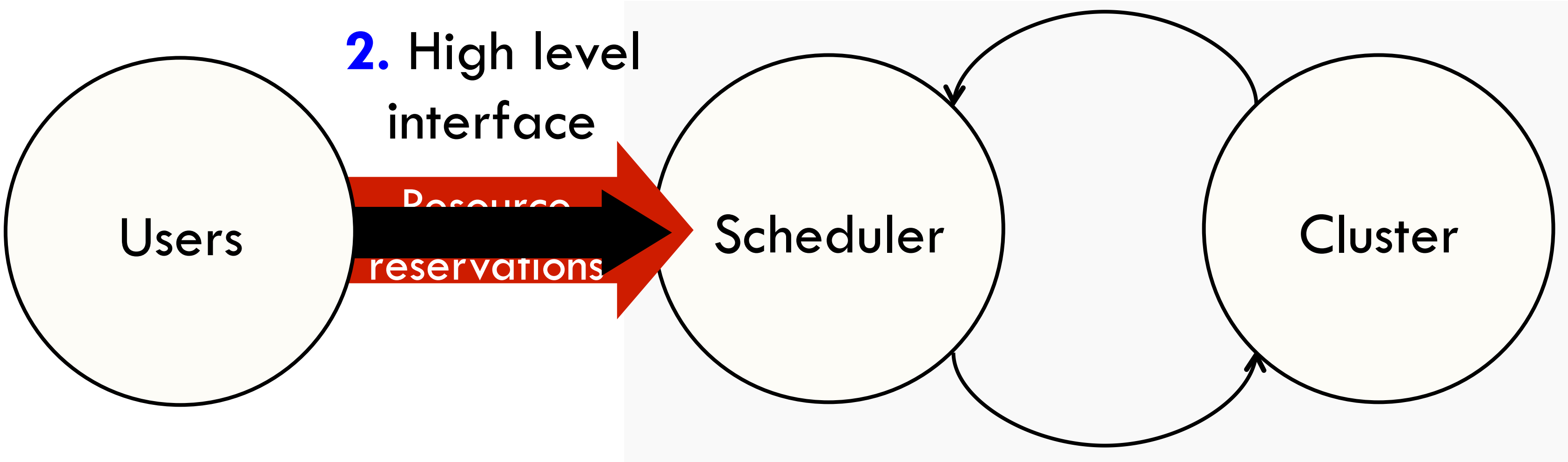
The *user* specifies resource reservations → **overprovisioning**



1. Practical data mining techniques

Quasar

[ASPLOS'14]



1. Practical data mining techniques

High-Level Interfaces

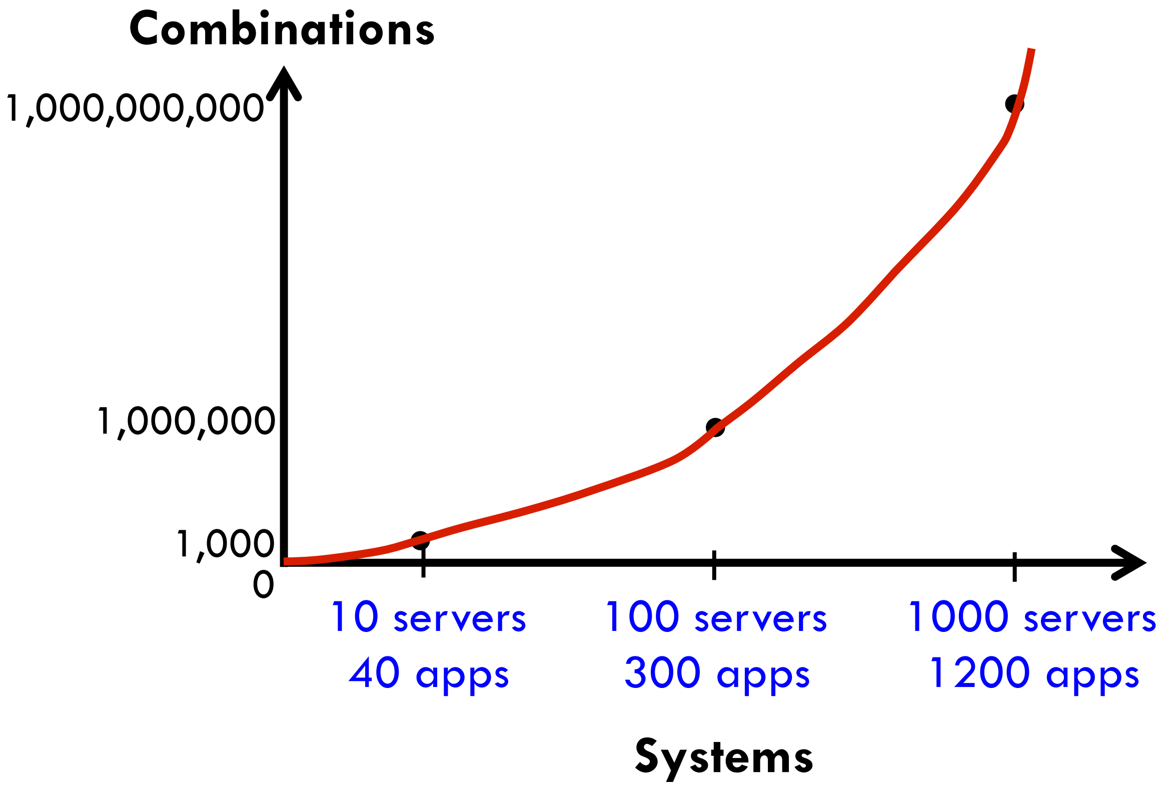
Focus on **what** performance is needed, not on **how** to achieve it

- Declarative interfaces:
  - SQL → describe the queries, not how they should be executed
  - DSLs → user describes program, language/compiler optimize
- Performance targets:
  - Batch: completion time, deadline
  - Interactive: throughput, tail latency

Extracting Resource Preferences

- Need to translate performance to resources

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



- Exhaustive characterization is infeasible

Applying Data Mining

> 100,000,000

$$
\begin{array}{c|cccc|}
 & \text{Platform 1 \& L-i \$} & \text{Platform 1 \& LLC} & \cdots & \text{Pl[illegible] M \& Net bw} \\
 & \text{\& 2 CPU/64GB RAM} & \text{\& 2 CPU/64GB RAM} & & \text{[illegible]/48GB RAM} \\
 & \text{\& 2 servers} & \text{\& 1 server} & & \text{\& 1 server} \\
\text{App A} & \text{1,500QPS} & \text{843QPS} & & \text{10,456QPS} \\
\text{App B} & \text{987QPS} & \text{458QPS} & \cdots & \text{1,836QPS} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\text{App N} & \text{10,893QPS} & \text{7,686QPS} & \cdots & \text{1,354QPS}
\end{array}
$$

Not practical!

- Exhaustive classification is impractical

Practical Resource Recommendations

Classification Engine	Goal
Heterogeneity	Determine suitable server platform
Interference	Determine sensitivity to resource interference

Practical Resource Recommendations

Classification Engine	Goal
Heterogeneity	Determine suitable server platform
Interference	Determine sensitivity to resource interference
Scale-up	Determine amount/ratio of resources per server

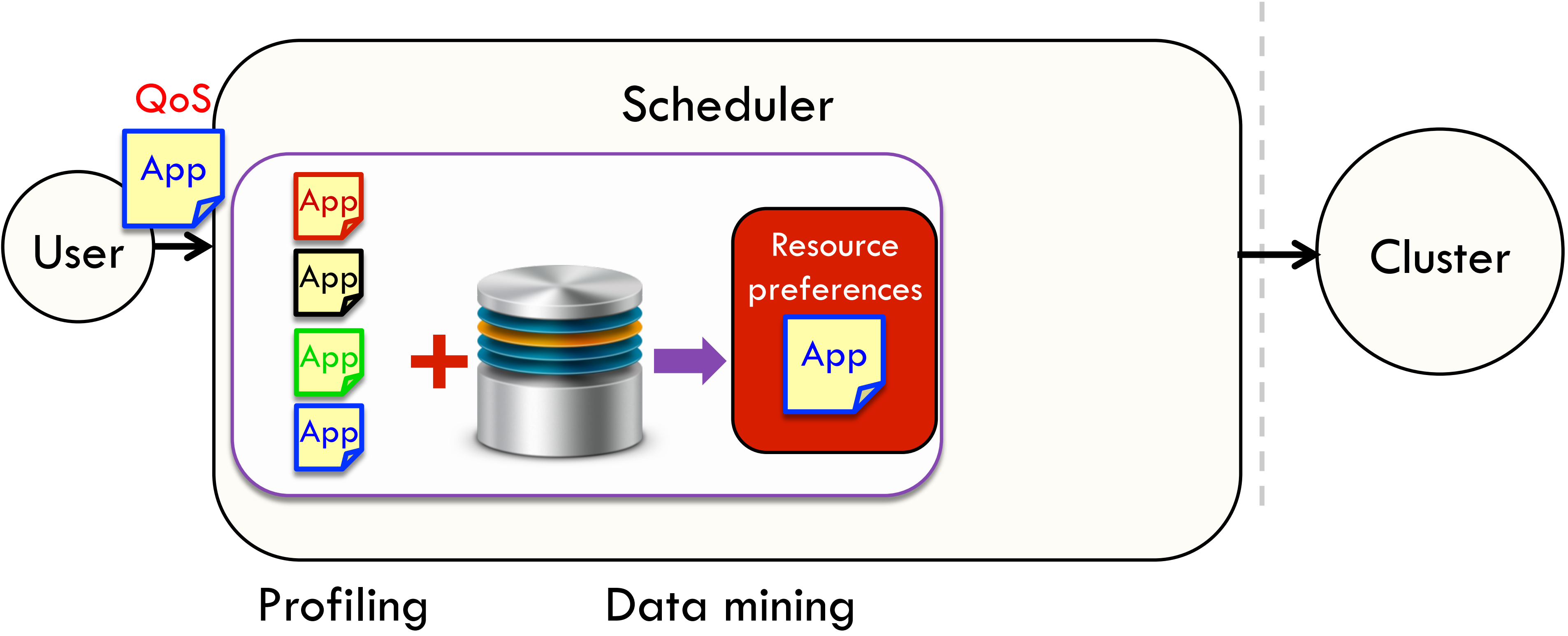
Practical Resource Recommendations

Classification Engine	Goal
Heterogeneity	Determine suitable server platform
Interference	Determine sensitivity to resource interference
Scale-up	Determine amount/ratio of resources per server
Scale-out	Determine appropriate number of servers

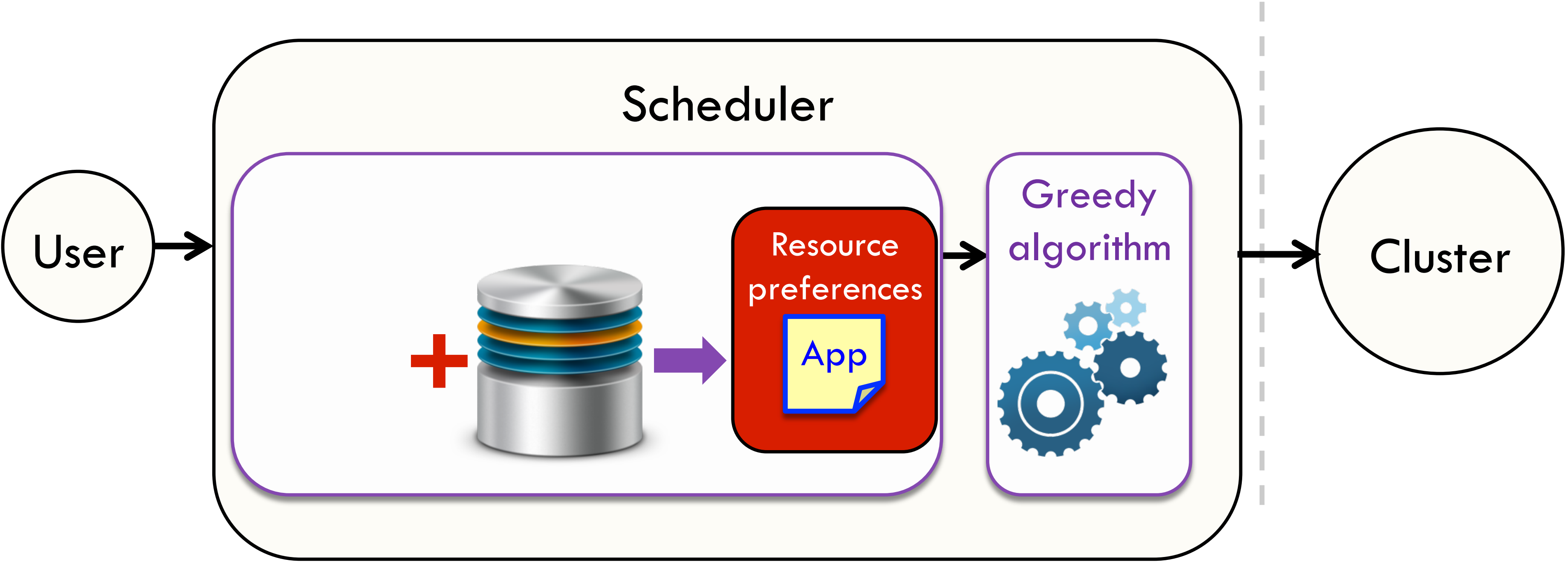
Practical Resource Recommendations

Classification Engine	Goal
Heterogeneity	Determine suitable server platform
Interference	Determine sensitivity to resource interference
Scale-up	Determine amount/ratio of resources per server
Scale-out	Determine appropriate number of servers
Application params	Determine appropriate settings for Hadoop, Spark, …

Quasar Overview

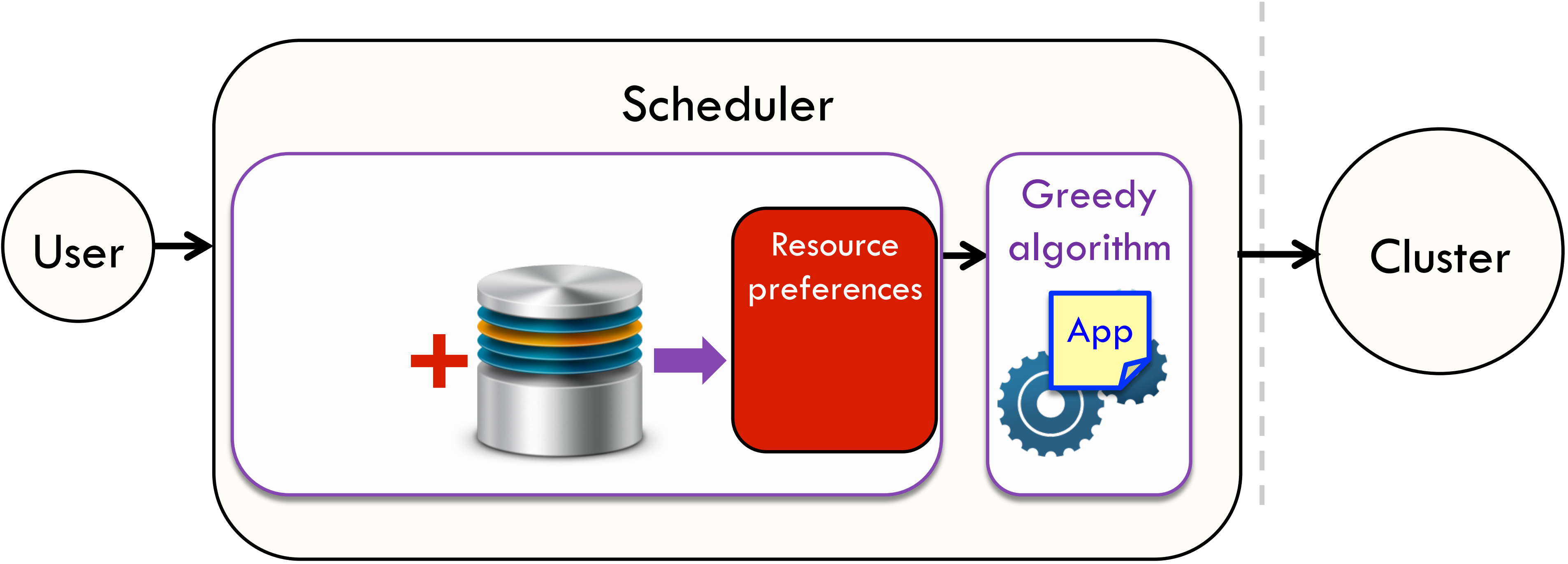


Quasar Overview



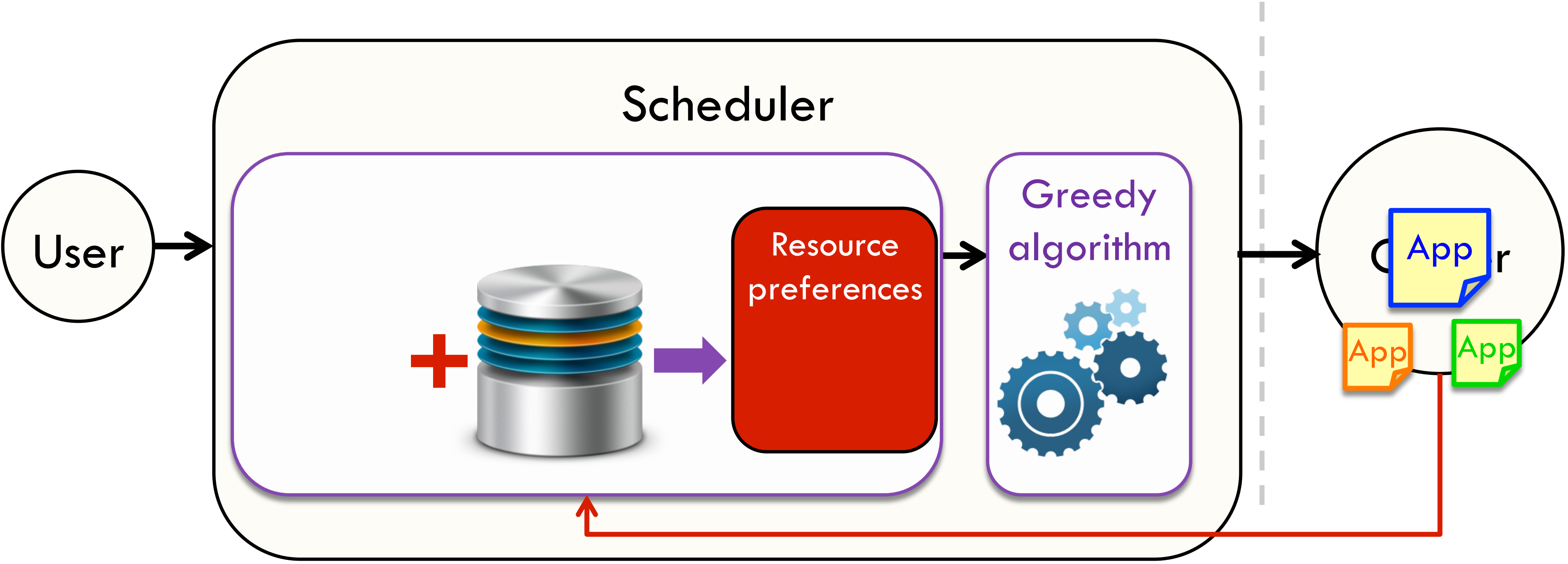
Profiling   Data mining   Resource selection

Quasar Overview



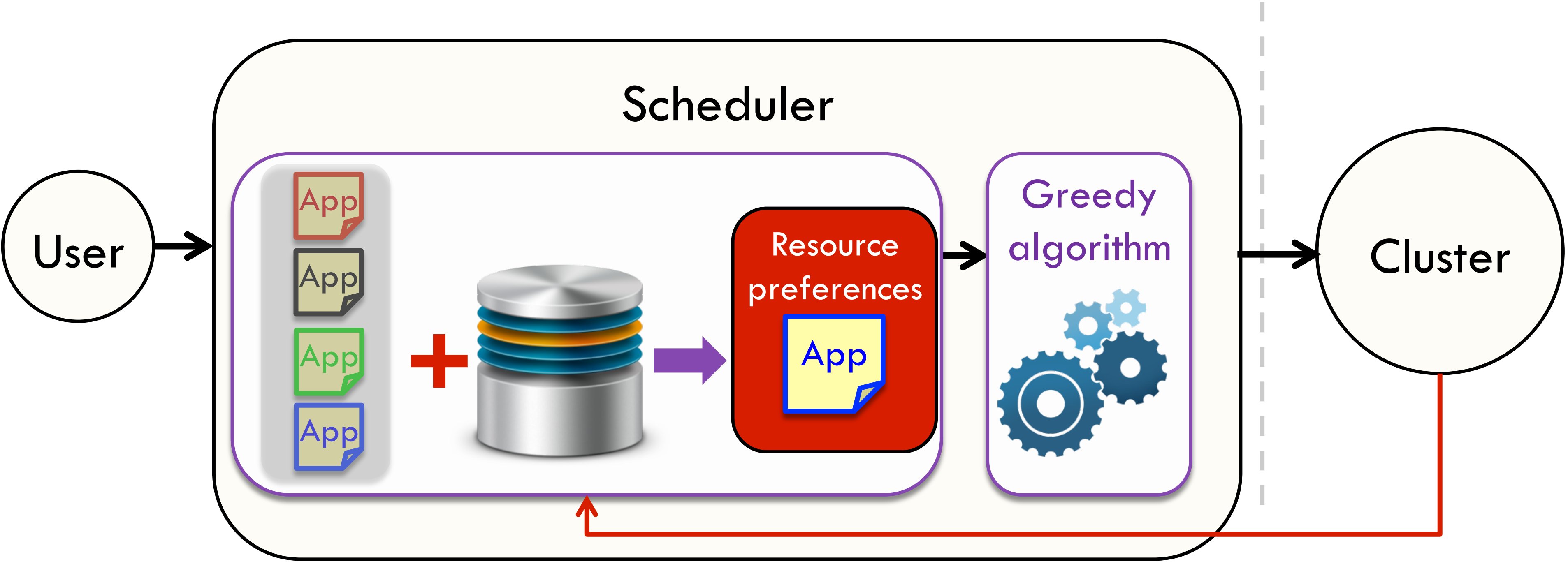
Profiling Data mining Resource selection

Quasar Overview



Profiling Data mining Resource selection

Quasar Overview



Profiling
[10-60sec]

Data mining
[20msec]

Resource selection
[50msec-2sec]

One-time for repetitive apps

Quasar Implementation

- 10,000 loc of C++ and Python

- Runs on Linux and OS X

- Supports frameworks in C/C++, Java, Scala and Python
  - ~100-600 loc for framework-specific code

- Side-effect free profiling runs with sealed containers

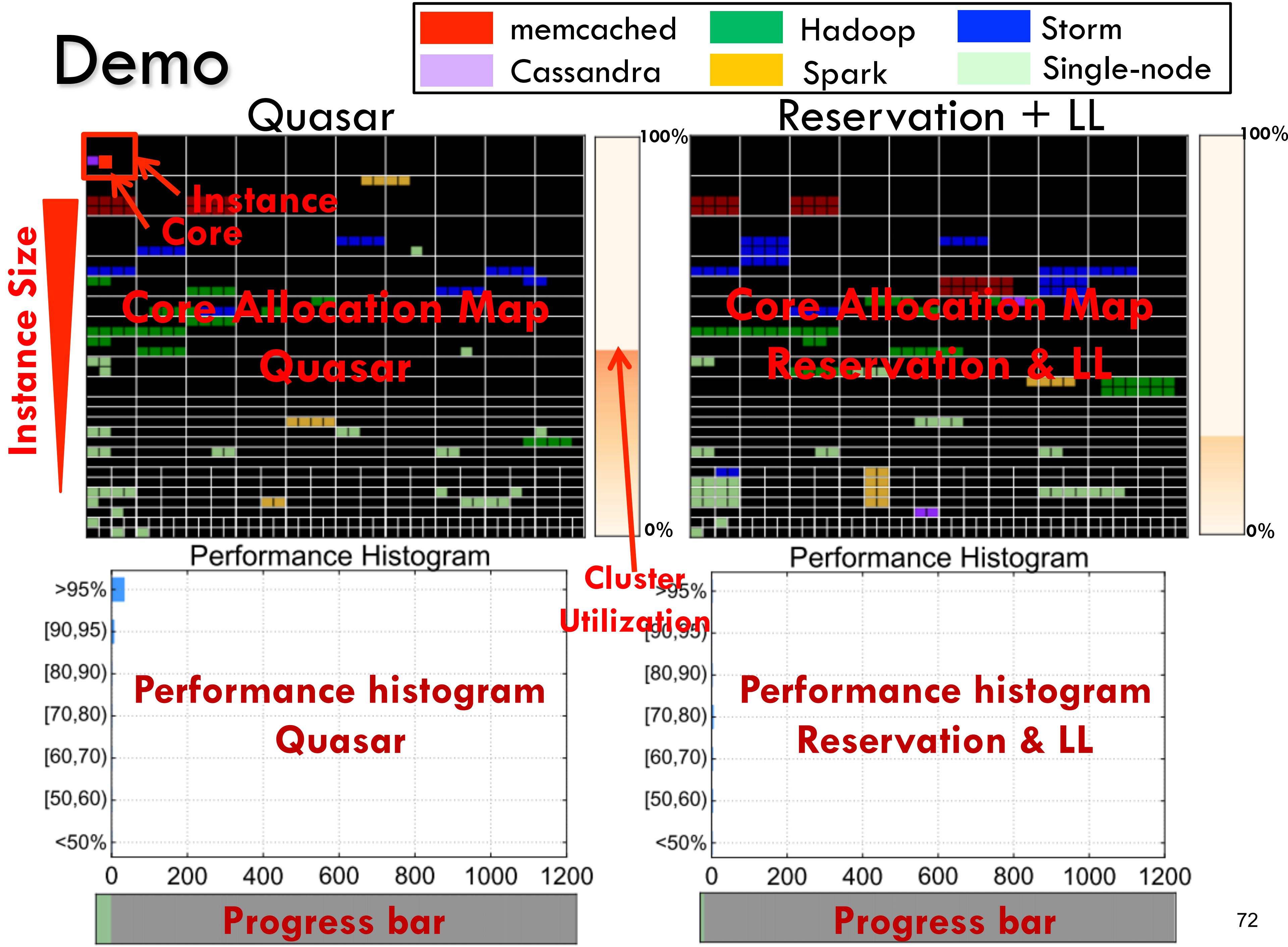
Evaluation: Cloud Scenario

- Cluster
  - 200 EC2 servers, 14 different server types

- Workloads: 1,200 apps with 1sec inter-arrival rate
  - Analytics: Hadoop, Spark, Storm
  - Latency-critical: Memcached, HotCrp, Cassandra
  - Single-threaded: SPEC CPU2006
  - Multi-threaded: PARSEC, SPLASH-2, BioParallel, Specjbb
  - Multiprogrammed: 4-app mixes of SPEC CPU2006

- Objectives: high cluster utilization and good app QoS

Demo



Instance Size

Instance

Core

Core Allocation Map Quasar

Core Allocation Map Reservation & LL

Cluster Utilization

Performance histogram Quasar

Performance histogram Reservation & LL

Progress bar

Progress bar

Demo

Cloud Scenario Summary

Quasar achieves:

- 91% of applications meet QoS
- ~10% overprovisioning as opposed to up to 5x
- Up to 70% cluster utilization at steady-state
- 23% shorter scenario completion time

Early Adoption

twitter

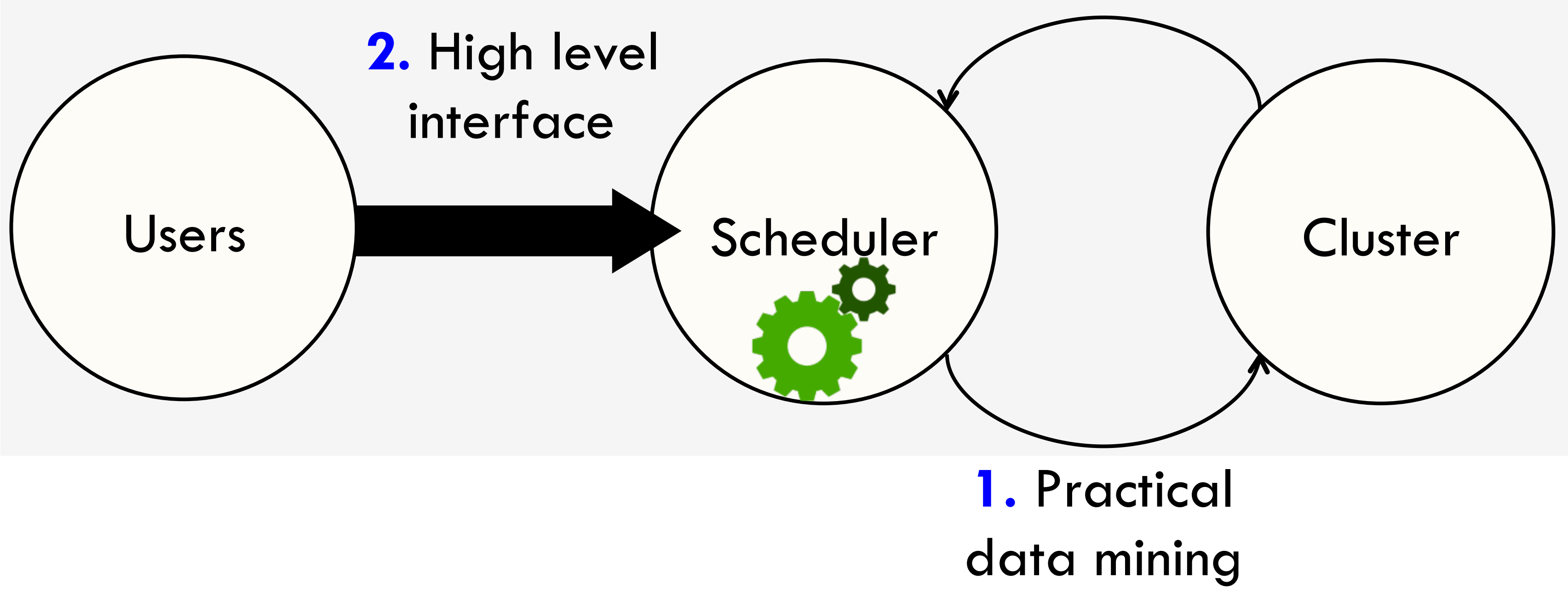
at&t

https://github.com/att-innovate/charmander

MESOS

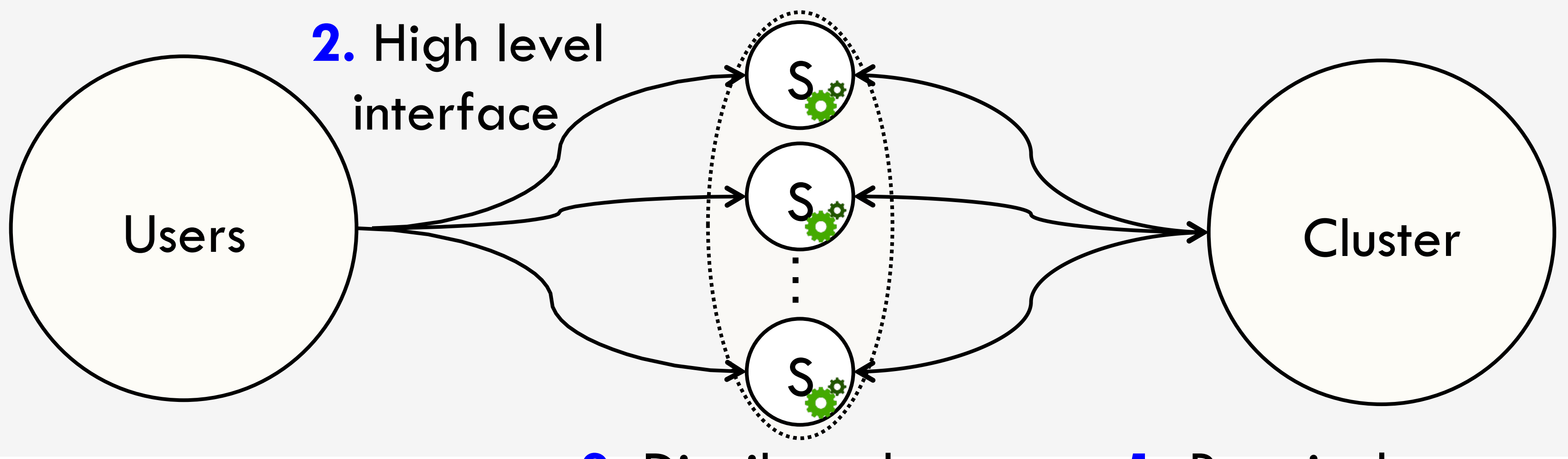
Contributions

Quasar [ASPLOS'14]



Contributions

Tarcil [SOCC'15]



2. High level interface

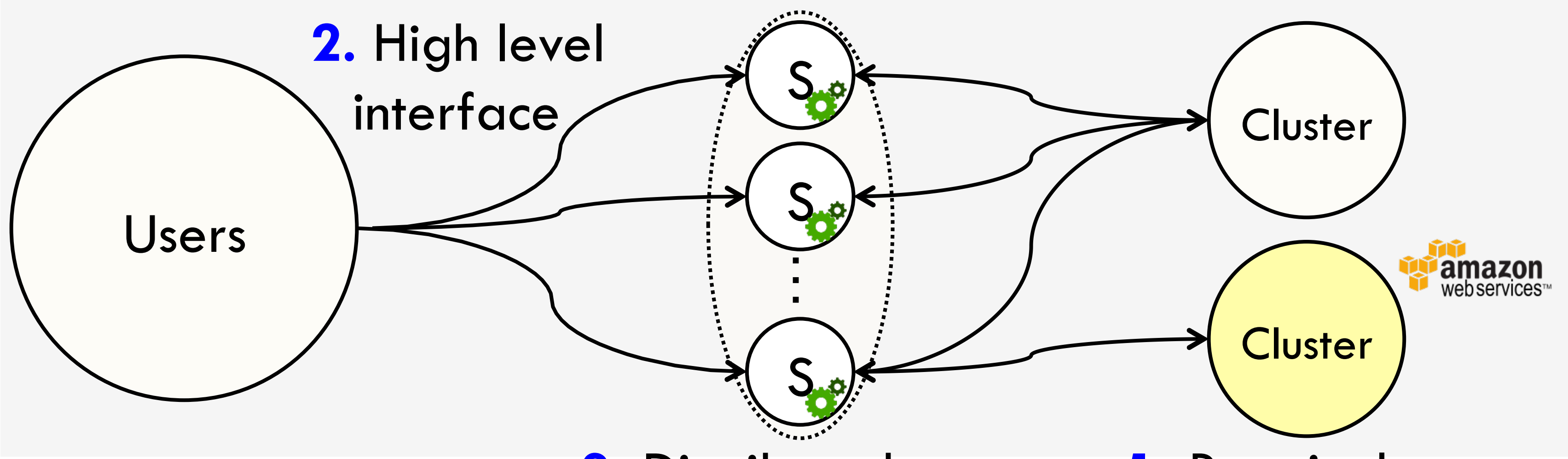
3. Distributed, sampling-based scheduling

1. Practical data mining

Contributions

Hybrid Cloud [in submission]

4. Private vs. public resources



2. High level interface

3. Distributed, sampling-based scheduling

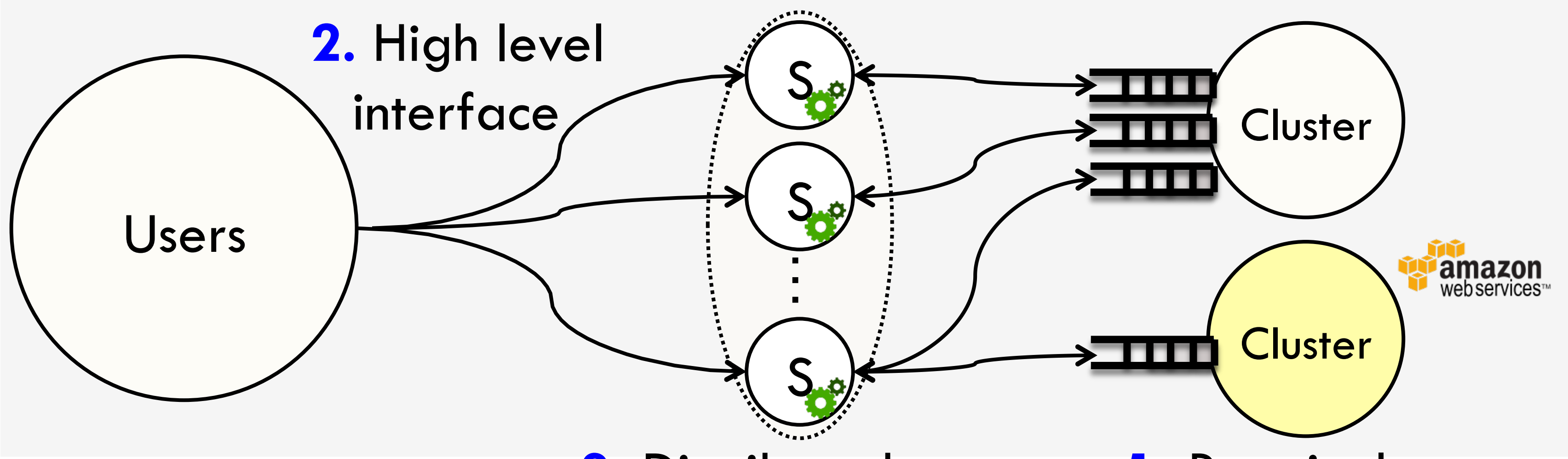
1. Practical data mining

Contributions

ARQ [ICAC'13]

5. Admission control

4. Private vs. public resources



2. High level interface

3. Distributed, sampling-based scheduling

1. Practical data mining

Conclusions

- Resource efficiency: significant challenge in systems of all scales
  - Focus on scalability of large-scale datacenters

- Cluster management: high utilization & high app performance
  - High-level declarative interface
  - Practical data mining techniques
  - Cross-layer design

Questions??

- Resource efficiency: significant challenge in systems of all scales
  - Focus on scalability of large-scale datacenters

- Cluster management: high utilization & high app performance
  - High-level declarative interface
  - Practical data mining techniques
  - Cross-layer design

Thank you!