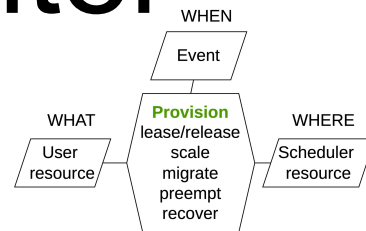
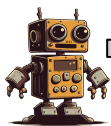


The Cost of Simplicity: Understanding Datacenter Scheduler Programming Abstractions



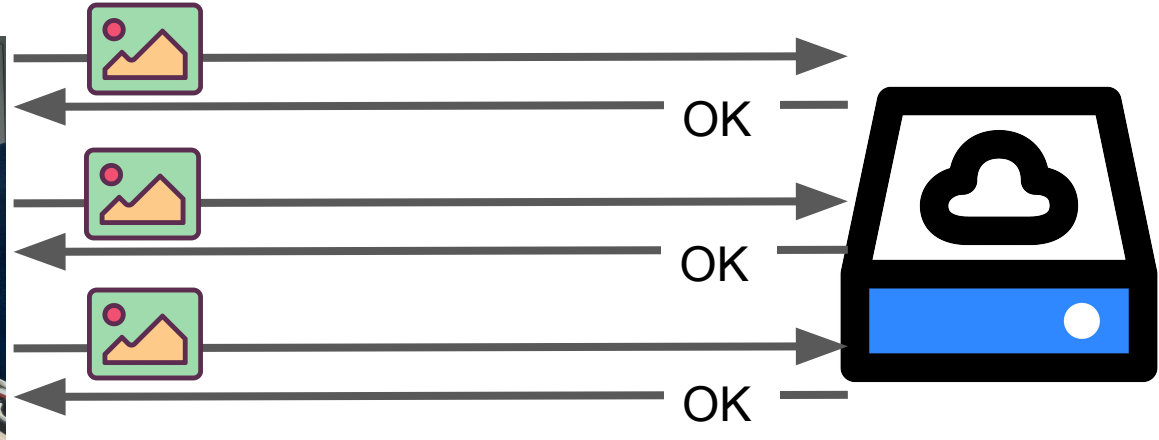
Aratz Manterola Lasa¹, **Sacheendra Talluri**²,
Tiziano De Matteis², Alexandru Iosup²

¹WarpStream Labs

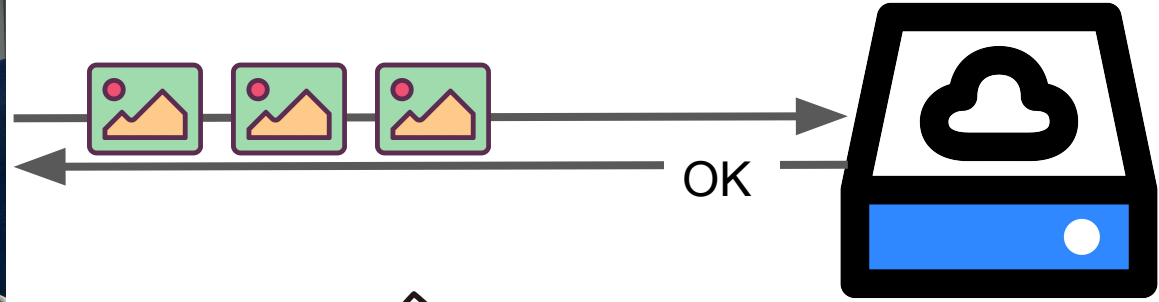
²Vrije Universiteit Amsterdam



Why APIs/abstractions?



Why APIs/abstractions?

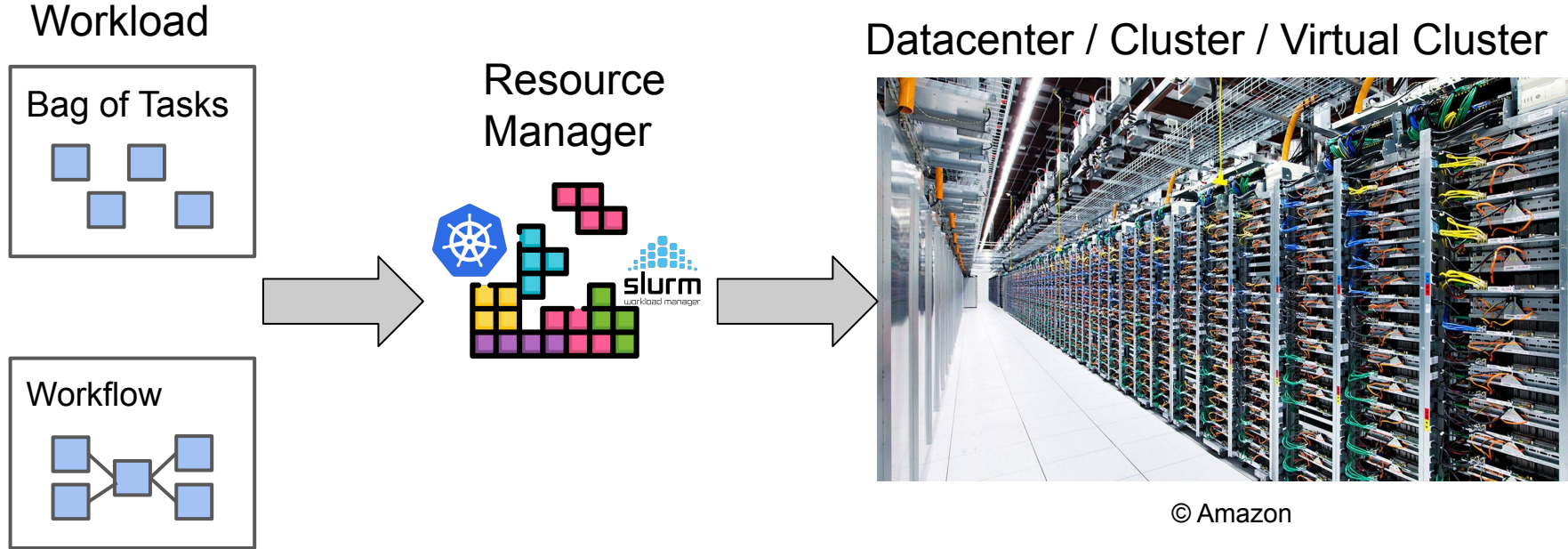


Less work for the user!

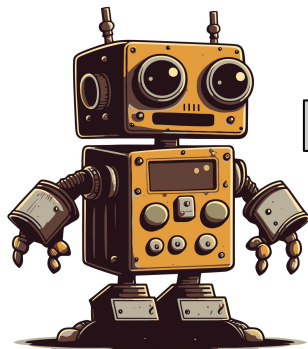


Better resource management
by the provider

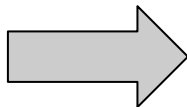
Datacenter System Model



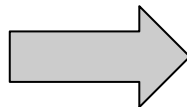
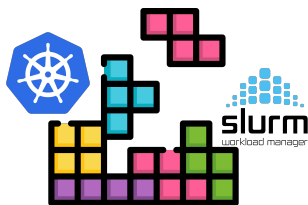
Runtimes Can Use Complex Resource Management APIs



CC Mehedi Hasan



Resource
Manager



Datacenter / Cluster / Virtual Cluster



© Amazon

Runtimes Can Use Complex Resource Management APIs

I will run jobs at 10:00

I'll be ready

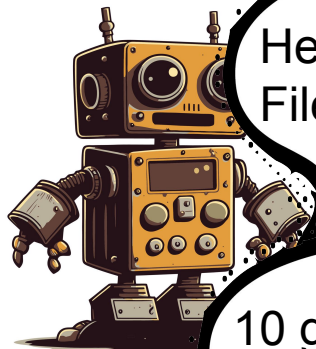
Here are jobs

Files read: 1

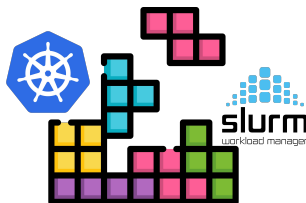
Network access pattern?

10 gathers, 1 all-reduce

Gather on n9-n19
Final reduce: n42



CC Mehedi



Datacenter /
Cluster /
Virtual Cluster



© Amazon

Realtime Can Use Complex Resource Management APIs

Data Grid

Decoupling Computation and Data Scheduling in Distributed Data-Intensive Applications

Kavitha Ranganathan¹ Ian Foster^{2*}

¹Department of Computer Science, University of Chicago, Chicago, IL 60637, USA
²Math and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA
{kranagan,ifoster}@cs.uchicago.edu

Abstract
In many energy physics, bioinformatics, and other disciplines, we encounter applications involving numerically heavy input jobs that both access and generate large data sets. Scheduling data-intensive work harnesses geographically distributed resources, large-scale data-intensive problems. Grid and cloud computing in such environments is challenging. A need to address a variety of requirements, such as resource utilization, response time, and best allocation policies while dealing with potentially independent sources of jobs and number of storage, compute, and network resources. We describe a scheduling framework that addresses these problems. Within this framework, data access, computation, and network resources are decoupled, autonomous resources are used for scheduling decisions on an alternative, periodic, observed data access patterns and load. We develop a family of job scheduling and data movement (exploration) algorithms and use simulation studies to evaluate various combinations. Our results suggest that while it is necessary to consider the impact of replication on the scheduling strategy, it is not always

critical. In this paper, we present a family of algorithms that can be used to schedule a wide variety of jobs, each accessing some subset of that data. Scheduling is a challenging task in this context. The

problem is to address this problem, we have defined a general and extensible scheduling framework within which we can instantiate a wide variety of scheduling algorithms, and then used simulation studies to explore the effectiveness of different algorithms within this framework. We assume a common model in which resource

Availability Demand

Paragon: QoS-Aware Scheduling for Heterogeneous Datacenters

Sijj Shen¹, Alexandru Iosup¹,
Delft University of Technology,
The Netherlands,
Google, Mountain View, CA, USA

Abstract—Datacenters are at the core of a wide variety of data ICT services, ranging from scientific computing to online gaming. Due to the scale of today's datacenters, the failure of computing resources is a common occurrence that may disrupt the availability of ICT services, leading to revenue loss. Although many high availability (HA) techniques have been proposed to mask resource failures, datacenter users—who rent datacenter

effectively instead of HA concepts, analyzing the impact of resource failures on user experience, and reducing the overall 15-hour

However, all open-source resource managers offer only simple APIs

NetHint: White-Box Networking for Multi-Tenant Data Centers

Jingrong Chen Hong Zhang¹ Wei Zhang Liang Luo² Jeffrey Chase Ion Stoica¹ Danyang Zhao

¹Duke University ²UC Berkeley ³University of Washington

Abstract

A cloud provider today provides its network resources to its tenants as a black box, such that cloud tenants have little knowledge of the underlying network characteristics. Meanwhile, data-intensive applications are increasingly migrated to the cloud, and these applications have both the ability and the incentive to adapt their data transfer schedules based on the cloud network characteristics. We find that the black-box networking abstraction and the adaptiveness of data-intensive applications together create a mismatch, leading to sub-optimal application performance.

This paper explores a white-box approach to resolving this mismatch. We propose NetHint, an interactive mechanism between a cloud tenant and a cloud provider to jointly enable application performance. With NetHint, the provider provides a hint — an indirect indication of the underlying network characteristics (e.g., link-layer network topologies for a tenant's virtual machines, number of co-locating tenants, network bandwidth utilizations), and the tenant's applications then adapt their transfer schedules accordingly. The NetHint design provides abundant network information for cloud tenants to compute their optimal transfer schedules, while introducing little overhead for the cloud provider to collect and expose this information. Evaluation results show that NetHint improves the average performance of all traffic completion time, broadcast completion time, and MapReduce shuffle completion time by 2.7x, 1.5x, and 1.2x, respectively.

1 Introduction

Data-intensive applications (e.g., network functions, data

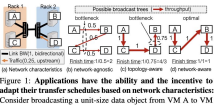


Figure 1: Applications have the ability and the incentive to adapt their transfer schedules based on network characteristics. Consider broadcasting a unit-size data object from VM A to VM B, C, and D. (a) Shows the network characteristics, all links have bidirectional bandwidth of 1. VM D has upstream background traffic of 0.25. (b) to (d) show possible broadcast trees and their corresponding broadcast finish time. The arrows represent traffic flows and the numbers represent the throughput.

The black-box model has worked well for decades due to its simplicity. However, with the emergence of popular data-intensive applications (e.g., data analytics, distributed deep learning, and distributed reinforcement learning) in the cloud, we observe that such a black-box model is no longer efficient (82). The crux is that many of these emerging applications have both the ability and the incentive to adapt their transfer schedules based on the underlying network characteristics, but it is difficult to do so with a black-box network.

Consider broadcast, an important communication primitive in reinforcement learning and ensemble model serving. Figure 1 shows an example that VM A broadcasts to VM B to VM D. Figure 1(b) shows a possible broadcast tree constructed under the black-box model. Without the underlying network characteristics, the broadcast tree is network-agnostic, which introduces link stress on the cross-rack link. Figure 1 shows

NetHint: White-Box Networking for Multi-Tenant Data Centers



Figure 1: Applications have the ability and the incentive to adapt their transfer schedules based on network characteristics. Consider broadcasting a unit-size data object from VM A to VM B, C, and D. (a) Shows the network characteristics, all links have bidirectional bandwidth of 1. VM D has upstream background traffic of 0.25. (b) to (d) show possible broadcast trees and their corresponding broadcast finish time. The arrows represent traffic flows and the numbers represent the throughput.

The black-box model has worked well for decades due to its simplicity. However, with the emergence of popular data-intensive applications (e.g., data analytics, distributed deep learning, and distributed reinforcement learning) in the cloud, we observe that such a black-box model is no longer efficient (82). The crux is that many of these emerging applications have both the ability and the incentive to adapt their transfer schedules based on the underlying network characteristics, but it is difficult to do so with a black-box network.

Consider broadcast, an important communication primitive in reinforcement learning and ensemble model serving. Figure 1 shows an example that VM A broadcasts to VM B to VM D. Figure 1(b) shows a possible broadcast tree constructed under the black-box model. Without the underlying network characteristics, the broadcast tree is network-agnostic, which introduces link stress on the cross-rack link. Figure 1 shows

Paragon

Paragon: QoS-Aware Scheduling for Heterogeneous Datacenters

Sijj Shen¹, Alexandru Iosup¹,
Delft University of Technology,
The Netherlands,
Google, Mountain View, CA, USA

Abstract

Large-scale data-intensive applications such as scientific computing, online gaming, and video streaming are increasingly migrating to the cloud. Datacenters are at the core of a wide variety of data ICT services, ranging from scientific computing to online gaming. Due to the scale of today's datacenters, the failure of computing resources is a common occurrence that may disrupt the availability of ICT services, leading to revenue loss. Although many high availability (HA) techniques have been proposed to mask resource failures, datacenter users—who rent datacenter

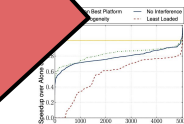


Figure 1: Performance degradation for 5,000 applications on 1000 EC2 servers with heterogeneous configurations. Interference-avoidance and deadline-tightness scheduling approaches to deal with scheduling problems result in less failed jobs. Results are ordered from worst to best performance workloads.

Energy-credit scheduling for energy-aware virtual machine scheduler for cloud systems

Nakki Kim¹, Jungwook Cho², Euisong Seo^{3*}

¹School of ECE, Chon National University of Science and Technology, Gyeongsang, 630-747, Republic of Korea
²Department of Computer Science, KAIST, Daejeon, 305-390, Korea
³Applied Informatics Research Institute (ARI), Spanish National Research Council (CSIC), Spain

Abstract

Virtualization facilitates the provision of flexible resources and improves energy efficiency through the consolidation of virtualized events into a smaller number of physical servers. As an increasing number of virtualized events are processed through the number of virtual machine instances, however, increasing load on the deployment of server hardware is not sufficient because the cooling and energy costs for data centers will increase the overhead costs for hardware. This paper proposes a model for estimating the energy consumption of each virtual machine without additional measurement hardware. Our model estimates the energy consumption of a virtual machine based on its processor energy generation by the virtual core power consumption and the data transfer rate. The proposed scheme can provide energy consumption according to the energy behavior of each virtual machine. The suggested scheme was implemented in the IaaS virtualization system, and an evaluation shows that the suggested scheme estimates and provides energy consumption with errors of less than 3% of the total energy consumption.

Energy-credit scheduling for energy-aware virtual machine scheduler for cloud systems

Nakki Kim¹, Jungwook Cho², Euisong Seo^{3*}

¹School of ECE, Chon National University of Science and Technology, Gyeongsang, 630-747, Republic of Korea
²Department of Computer Science, KAIST, Daejeon, 305-390, Korea
³Applied Informatics Research Institute (ARI), Spanish National Research Council (CSIC), Spain

Energy-credit scheduling for energy-aware virtual machine scheduler for cloud systems

Nakki Kim¹, Jungwook Cho², Euisong Seo^{3*}

¹School of ECE, Chon National University of Science and Technology, Gyeongsang, 630-747, Republic of Korea
²Department of Computer Science, KAIST, Daejeon, 305-390, Korea
³Applied Informatics Research Institute (ARI), Spanish National Research Council (CSIC), Spain

Energy-credit scheduling for energy-aware virtual machine scheduler for cloud systems

Nakki Kim¹, Jungwook Cho², Euisong Seo^{3*}

¹School of ECE, Chon National University of Science and Technology, Gyeongsang, 630-747, Republic of Korea
²Department of Computer Science, KAIST, Daejeon, 305-390, Korea
³Applied Informatics Research Institute (ARI), Spanish National Research Council (CSIC), Spain

Energy-credit scheduling for energy-aware virtual machine scheduler for cloud systems

Nakki Kim¹, Jungwook Cho², Euisong Seo^{3*}

¹School of ECE, Chon National University of Science and Technology, Gyeongsang, 630-747, Republic of Korea
²Department of Computer Science, KAIST, Daejeon, 305-390, Korea
³Applied Informatics Research Institute (ARI), Spanish National Research Council (CSIC), Spain

Energy-credit scheduling for energy-aware virtual machine scheduler for cloud systems

Nakki Kim¹, Jungwook Cho², Euisong Seo^{3*}

¹School of ECE, Chon National University of Science and Technology, Gyeongsang, 630-747, Republic of Korea
²Department of Computer Science, KAIST, Daejeon, 305-390, Korea
³Applied Informatics Research Institute (ARI), Spanish National Research Council (CSIC), Spain

Energy-credit scheduling for energy-aware virtual machine scheduler for cloud systems

Nakki Kim¹, Jungwook Cho², Euisong Seo^{3*}

¹School of ECE, Chon National University of Science and Technology, Gyeongsang, 630-747, Republic of Korea
²Department of Computer Science, KAIST, Daejeon, 305-390, Korea
³Applied Informatics Research Institute (ARI), Spanish National Research Council (CSIC), Spain

Reservation-Based Scheduling: You're Late Don't Blame Us!

Carlo Curino¹, Djellel E. Difallah¹, Chris Douglas², Subru Krishnan¹,
Raghu Ramakrishnan¹, Sriram Rao¹
¹Cloud Information Services Lab (CISL) — Microsoft Corp.
²{curino, difallo, subru, sriram}@microsoft.com, {djellel, rama}@mit.edu

Abstract
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Introduction
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

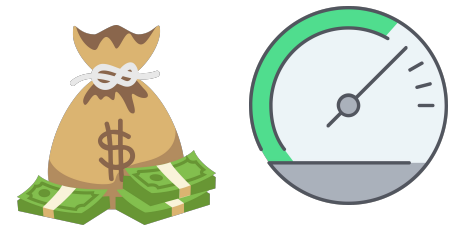
Reservation-based Scheduling
This paper introduces reservation-based scheduling, a novel approach to scheduling in multi-tenant datacenters. We develop our solution as a set of four key contributions: (1) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (2) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (3) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs; (4) we propose a reservation-based scheduling approach that allows users to declaratively reserve resources for their jobs.

Research Questions

RQ1. What API features are missing from commercial open-source resource managers?

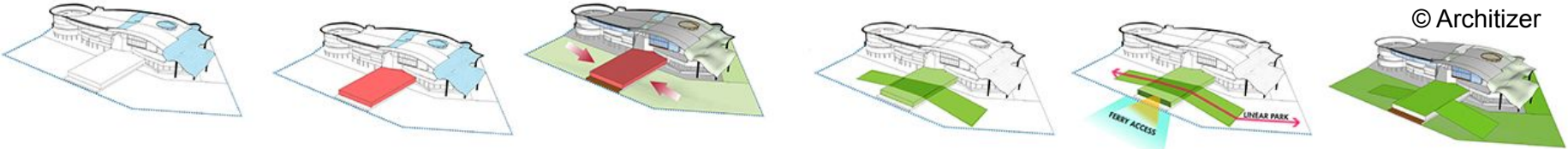
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

RQ2. What is the performance cost of these missing features?



Research Questions

RQ0. What is the reference architecture for resource manager APIs?



RQ1. What API features are missing from commercial open-source resource managers?

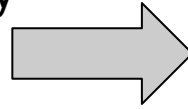
<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

RQ2. What is the performance cost of these missing features?

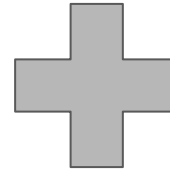


Resource Manager API Ref. Arch. Design Process

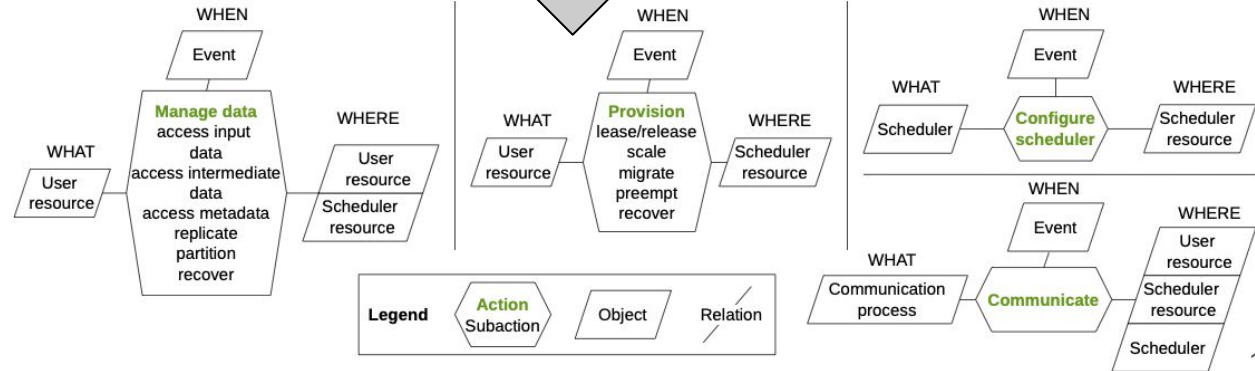
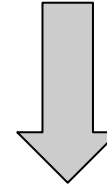
Systematic Literature Survey
 Keywords eg.: scheduler
 AND (datacenter OR API)
 Sorted by: normalized citations



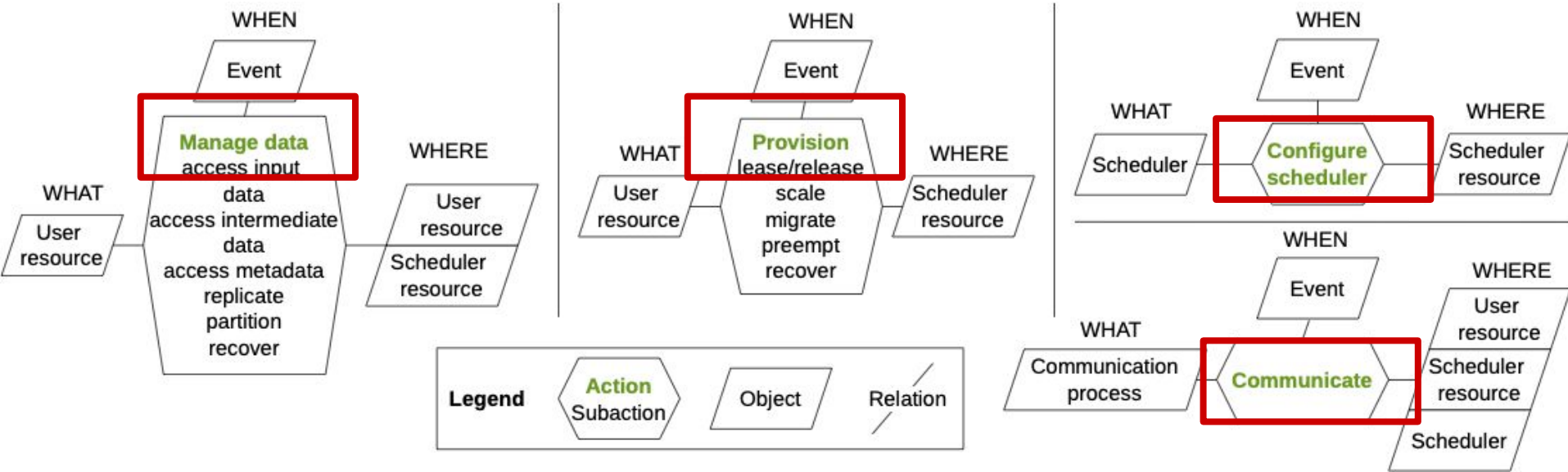
First 15 papers



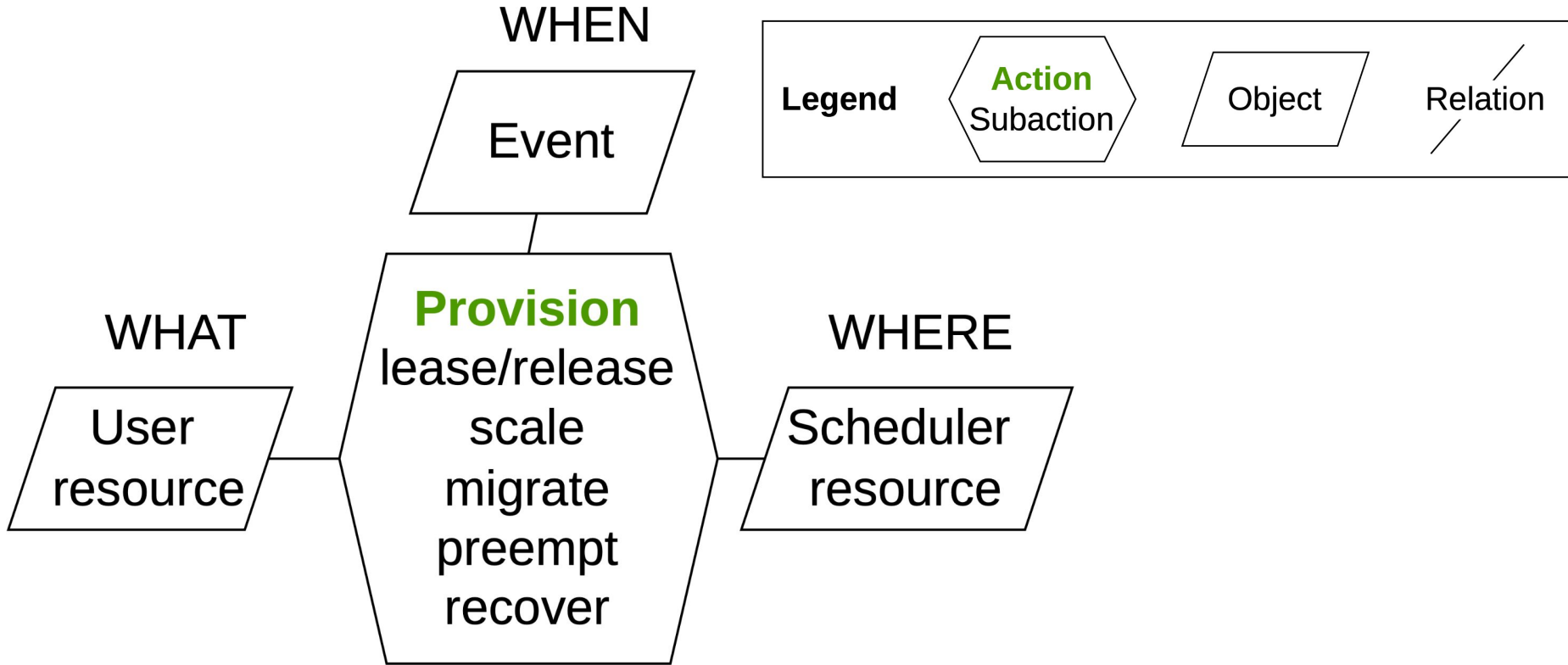
Features from
 5 industrial
 schedulers



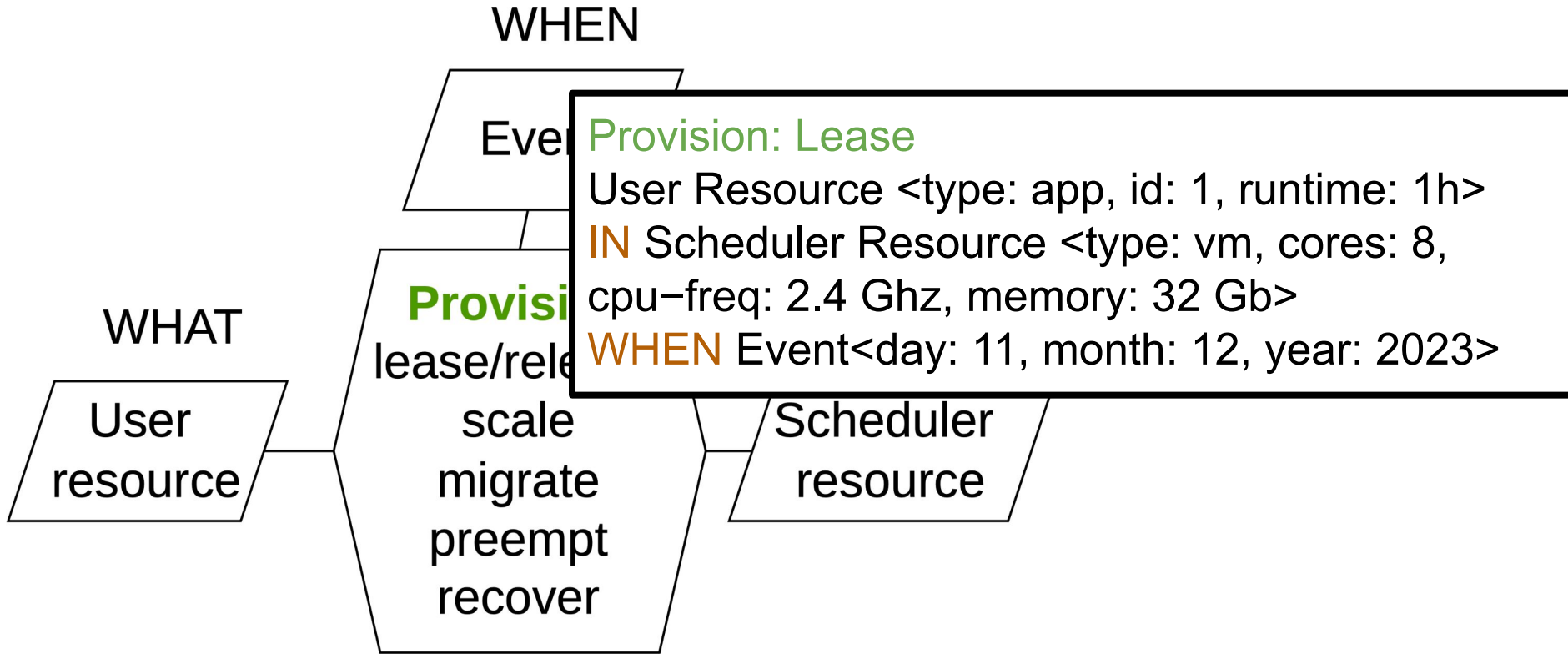
Resource Manager API Reference Architecture



Resource Manager API Reference Architecture - Provision

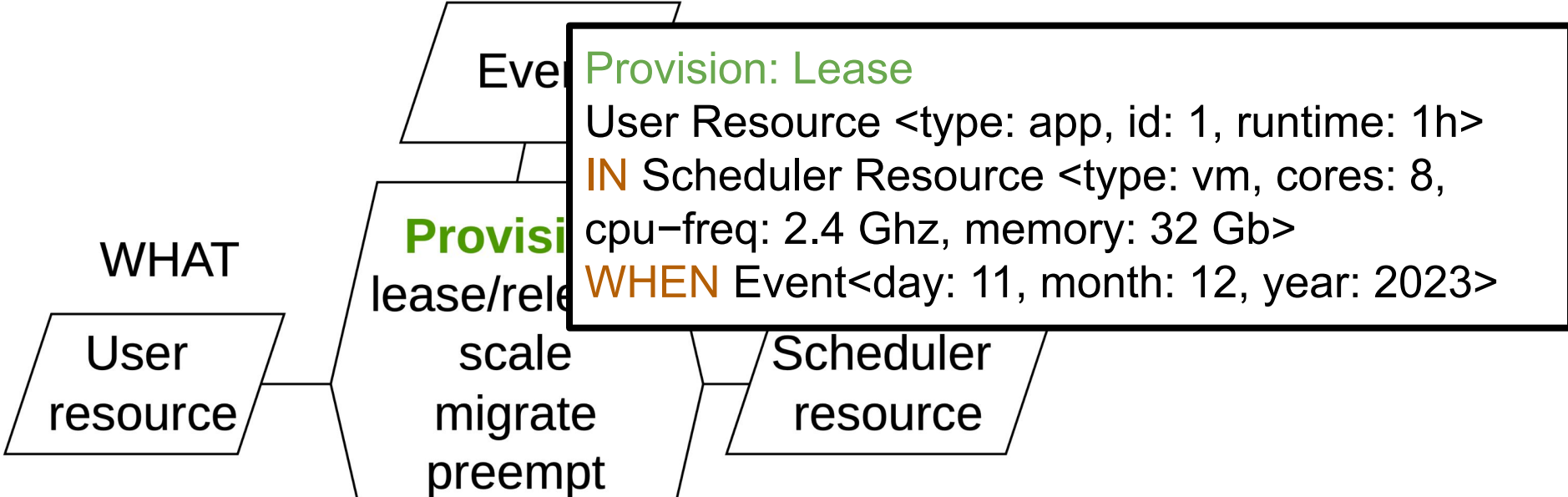


Resource Manager API Reference Architecture - Syntax



Resource Manager API Reference Architecture - Syntax

WHEN



RQ0. What is the reference architecture for resource manager APIs?

Mapping Industrial Scheduler API **Actions** to Ref. Arch.

Action	Sub-Action	Schedulers				
		Ku	Sl	Sp	Co	Ai
Provision	Lease / release	●	●	●	●	●
	Scale	●	○	◐	○	○
	Migrate	○	○	○	○	○
	Preempt	◐	●	○	●	○
	Recover	◐	◐	●	◐	◐
Configure scheduler		●	◐	●	●	●
Manage data	Access input	●	◐	●	●	●
	Access interm.	○	○	◐	○	○
	Access metadata	○	○	○	○	○
	Replicate	○	○	●	○	○
	Partition	○	○	●	○	○
	Recover	◐	○	●	●	○
Communicate		◐	●	◐	◐	◐

Ku = Kubernetes

Sl = SLURM

Sp = Spark

Co = HTCondor

Ai = Airflow

● = Full support

◐ = Partial support

○ = No support

APIs Missing in Industrial Schedulers

Action	Sub-Action	Schedulers				
		Ku	Sl	Sp	Co	Ai
Provision	Lease / release	●	●	●	●	●
	Scale	●	○	◐	○	○
	Migrate	○	○	○	○	○
	Preempt	◐	●	○	●	○
	Recover	◐	◐	●	◐	◐
Configure scheduler	Little Support for Data Management					
Manage data	Access input	●	◐	●	●	●
	Access interm.	○	○	◐	○	○
	Access metadata	○	○	○	○	○
	Replicate	○	○	●	○	○
	Partition	○	○	●	○	○
Communicate	Recover	◐	○	●	●	○
		◐	●	◐	◐	◐

Ku = Kubernetes

Sl = SLURM

Sp = Spark

Co = HTCondor

Ai = Airflow

● = Full support

◐ = Partial support

○ = No support

APIs Missing in Industrial Schedulers

Action	Sub-Action	Schedulers				
		Ku	Sl	Sp	Co	Ai
Provision	Lease / release	●	●	●	●	●
	Scale	●	○	◐	○	○
	Migrate	○	○	○	○	○
	Preempt	◐	●	○	●	○
	Recover	No Support for Migration				
Configure scheduler	No Support for Migration					
Manage data	Access input	●	◐	●	●	●
	Access interm.	○	○	◐	○	○
	Access metadata	○	○	○	○	○
	Replicate	○	○	●	○	○
	Partition	○	○	●	○	○
	Recover	◐	○	●	●	○
Communicate		◐	●	◐	◐	◐

Ku = Kubernetes

Sl = SLURM

Sp = Spark

Co = HTCondor

Ai = Airflow

● = Full support

◐ = Partial support

○ = No support

APIs Missing in Industrial Schedulers

Action	Sub-Action	Schedulers				
		Ku	Sl	Sp	Co	Ai
Provision	Lease / release	●	●	●	●	●
	Scale	●	○	◐	○	○
	Migrate	○	○	○	○	○
	Preempt	◐	●	○	●	○
	Recover	No Support for Migration				
Configure scheduler	No Support for Migration					
Manage data	Access input	●	◐	●	●	●
	Access interm.	○	○	◐	○	○
	Access metadata	○	○	○	○	○
	Replicate	○	○	●	○	○

Ku = Kubernetes

Sl = SLURM

Sp = Spark

Co = HTCondor

Ai = Airflow

● = Full support

◐ = Partial support

○ = No support

RQ1. What API features are missing from commercial open-source resource managers?

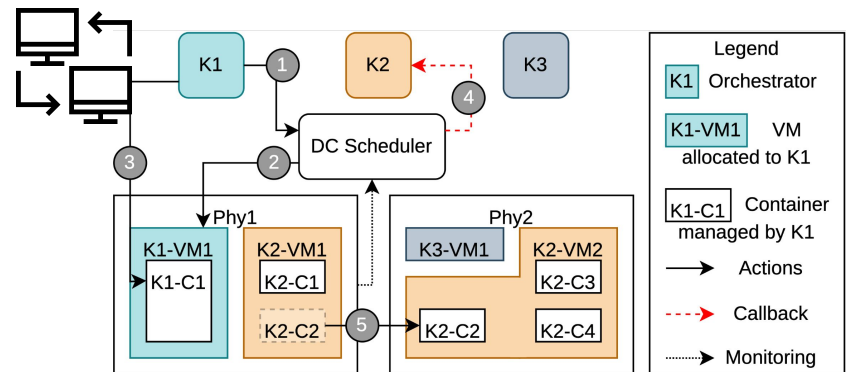
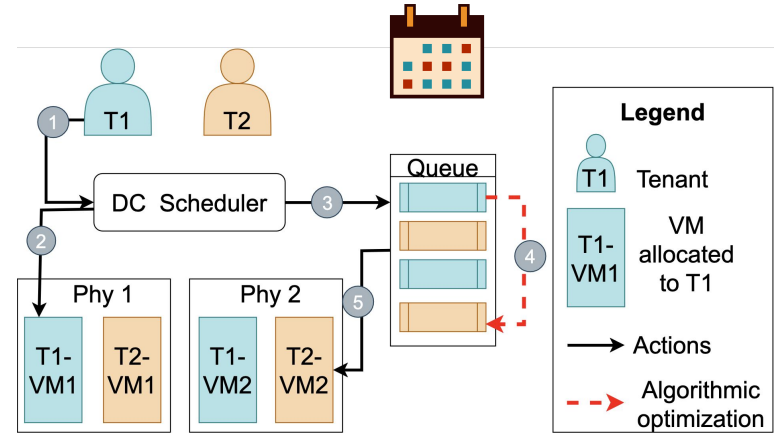
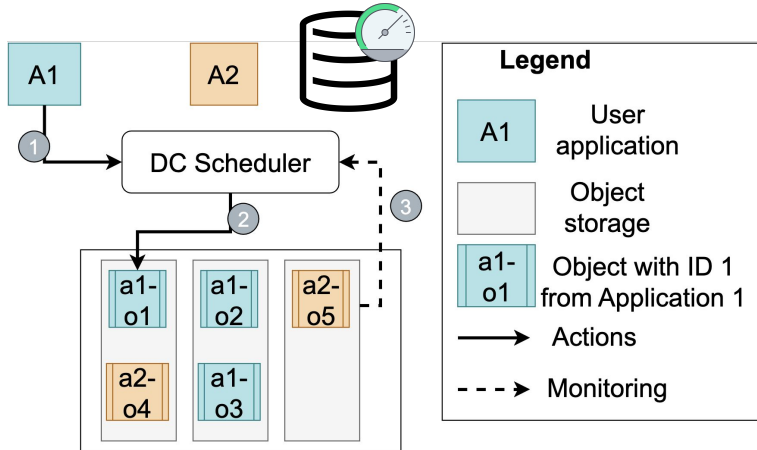
Communicate

◐ ● ◐ ◐ ◐

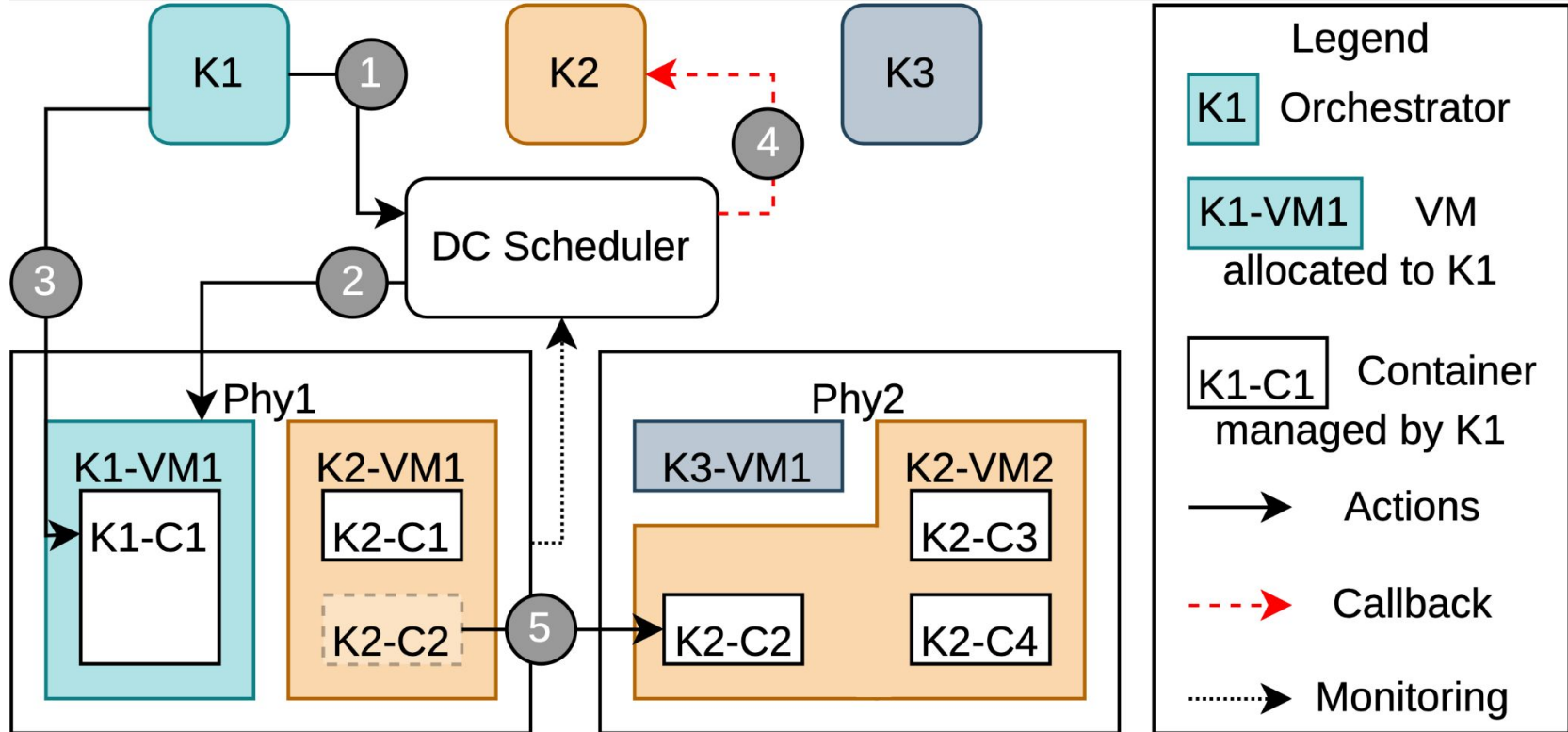
Cost of Missing APIs - Experiments

3 Experiments Using the OpenDC Simulator

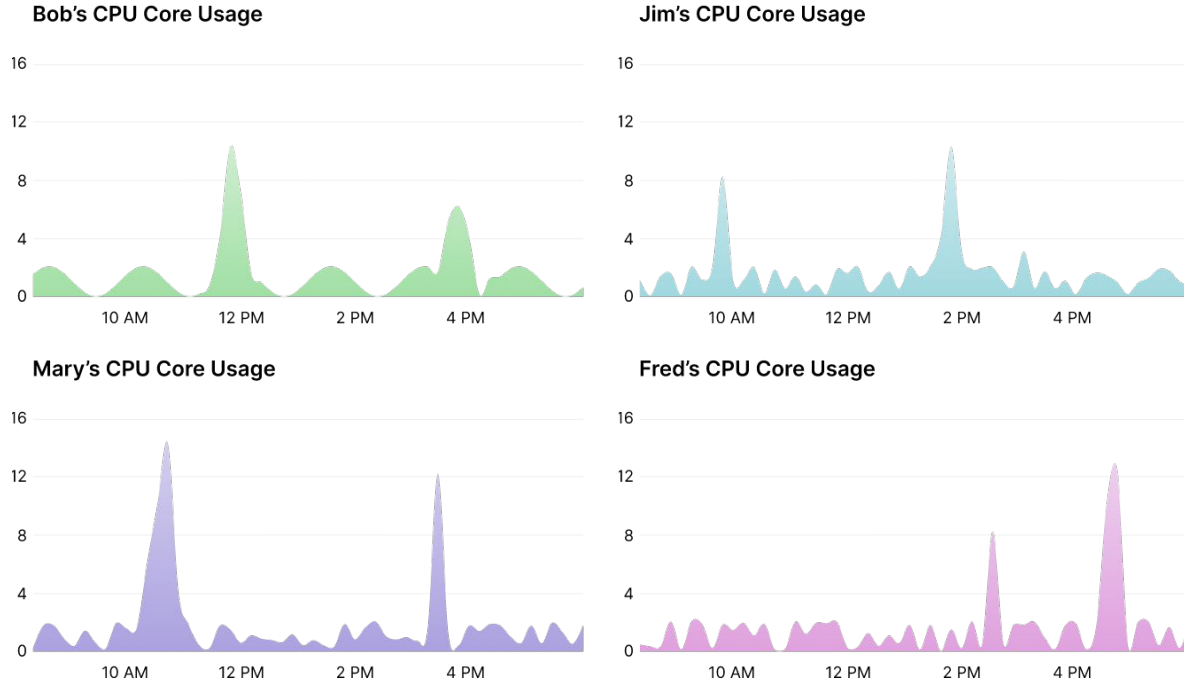
1. Resource Reservation
2. Container Migration
3. Storage Metadata



Container Migration Experiment - Setup



Virtual Machines - Resource Underutilization

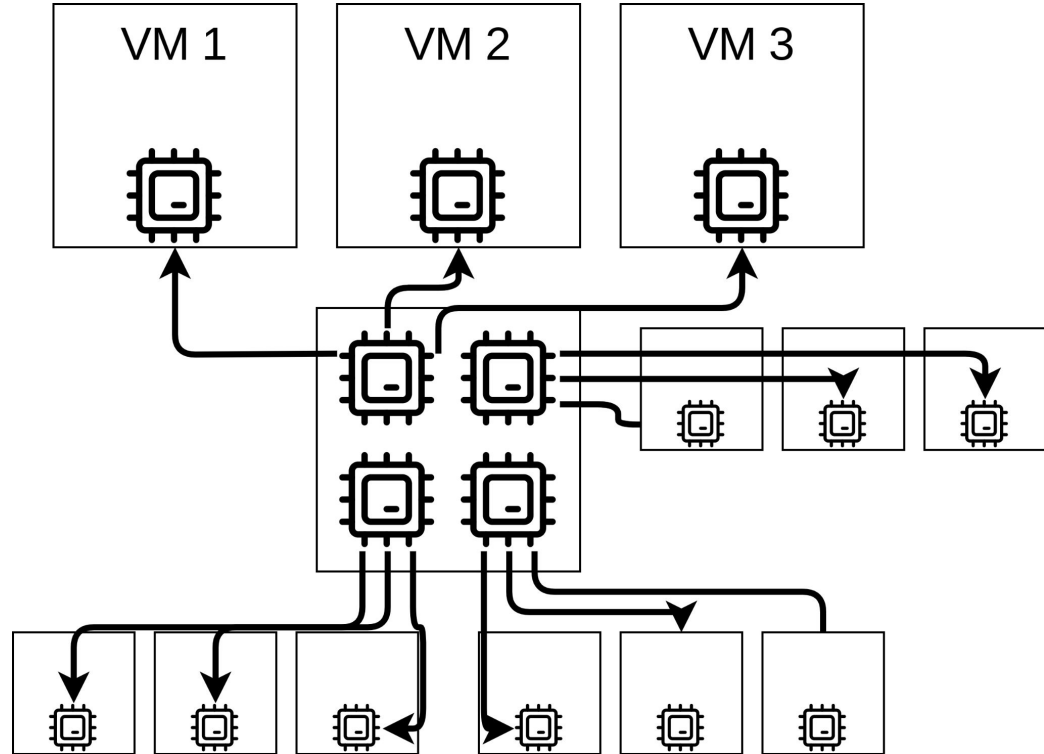


Oversubscription

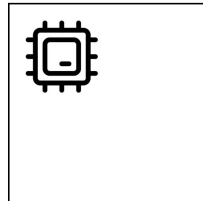
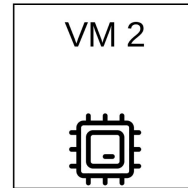
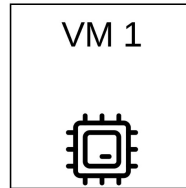
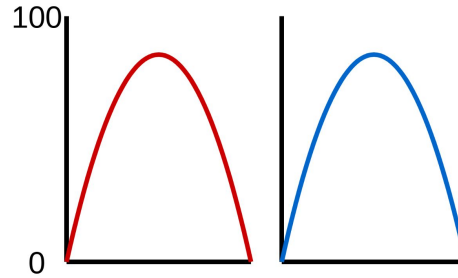
1 CPU for 3 VMs

Oversubscription ratio = 3

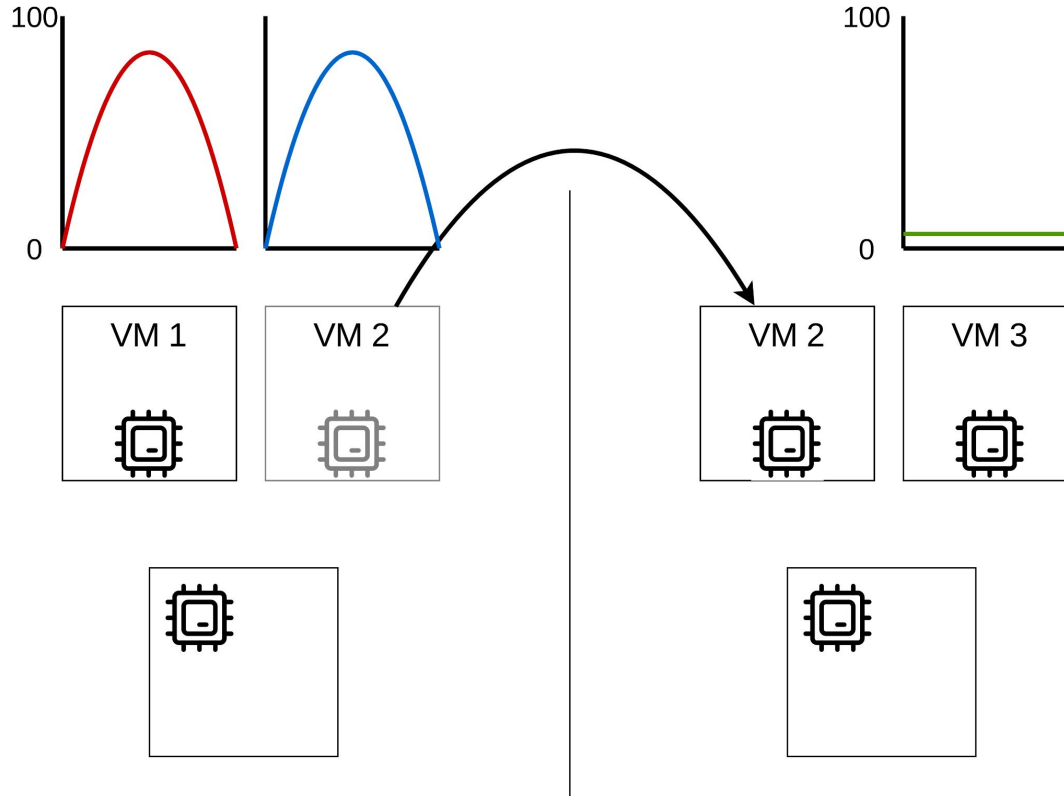
Time sharing



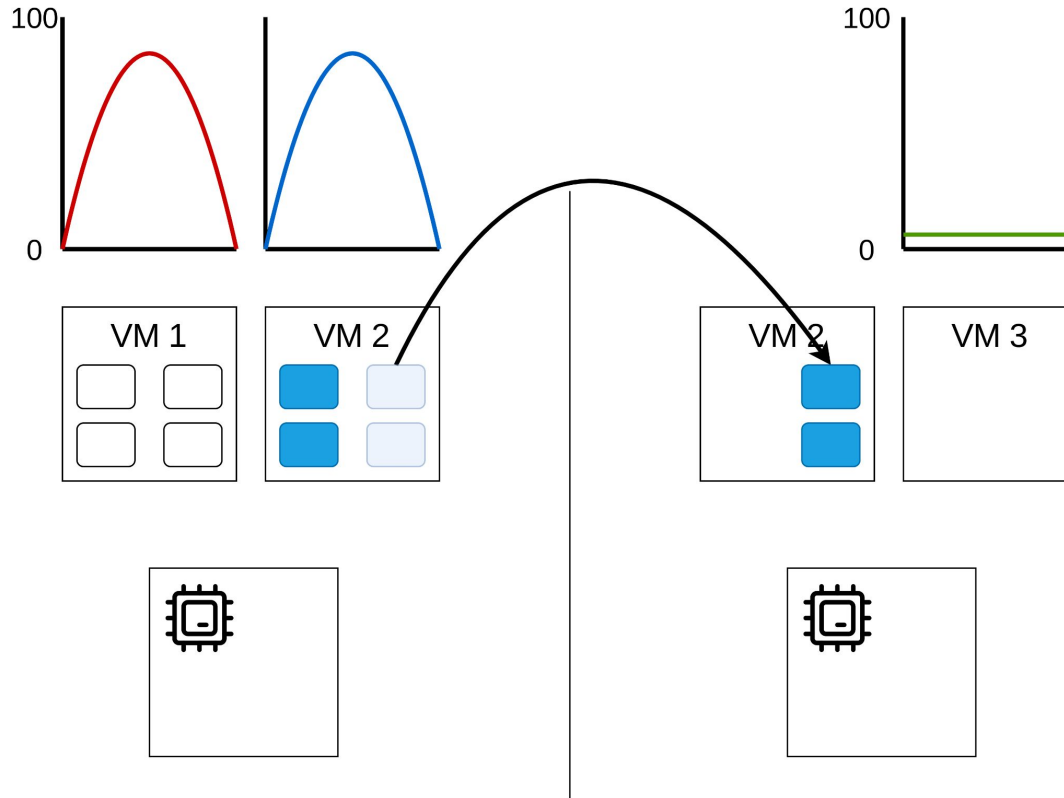
Oversubscription - Resource Contention



Migration



Container Migration



Container Migration Experiment - Setup

Cluster Setup:

1 Physical Cluster

5 Kubernetes Virtual Clusters

512 Mbps migration speed

80% average CPU utilization target

Physical cluster size calibrated for each trace

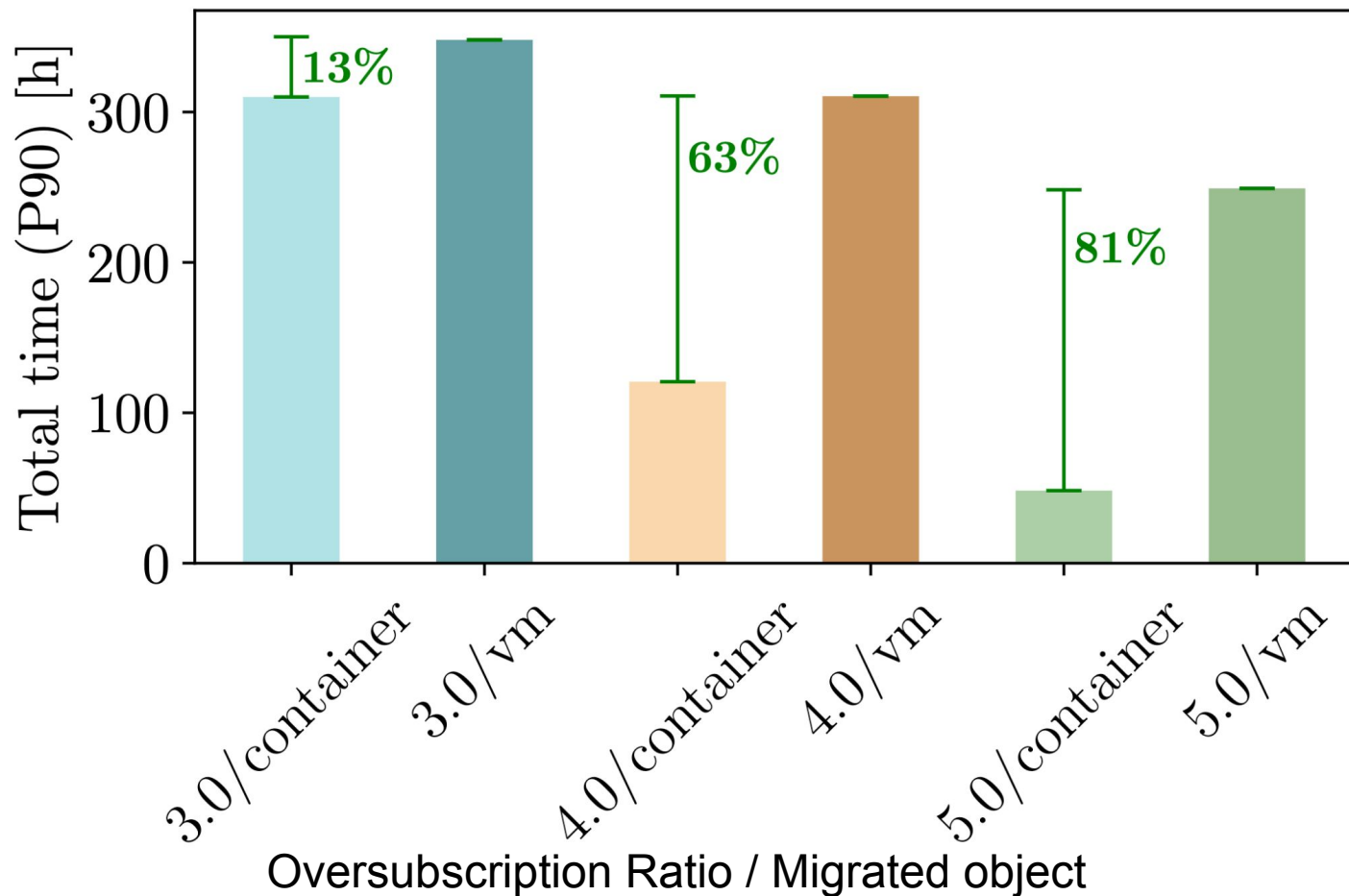
3 Traces:

Google 2011 (~25 machines)

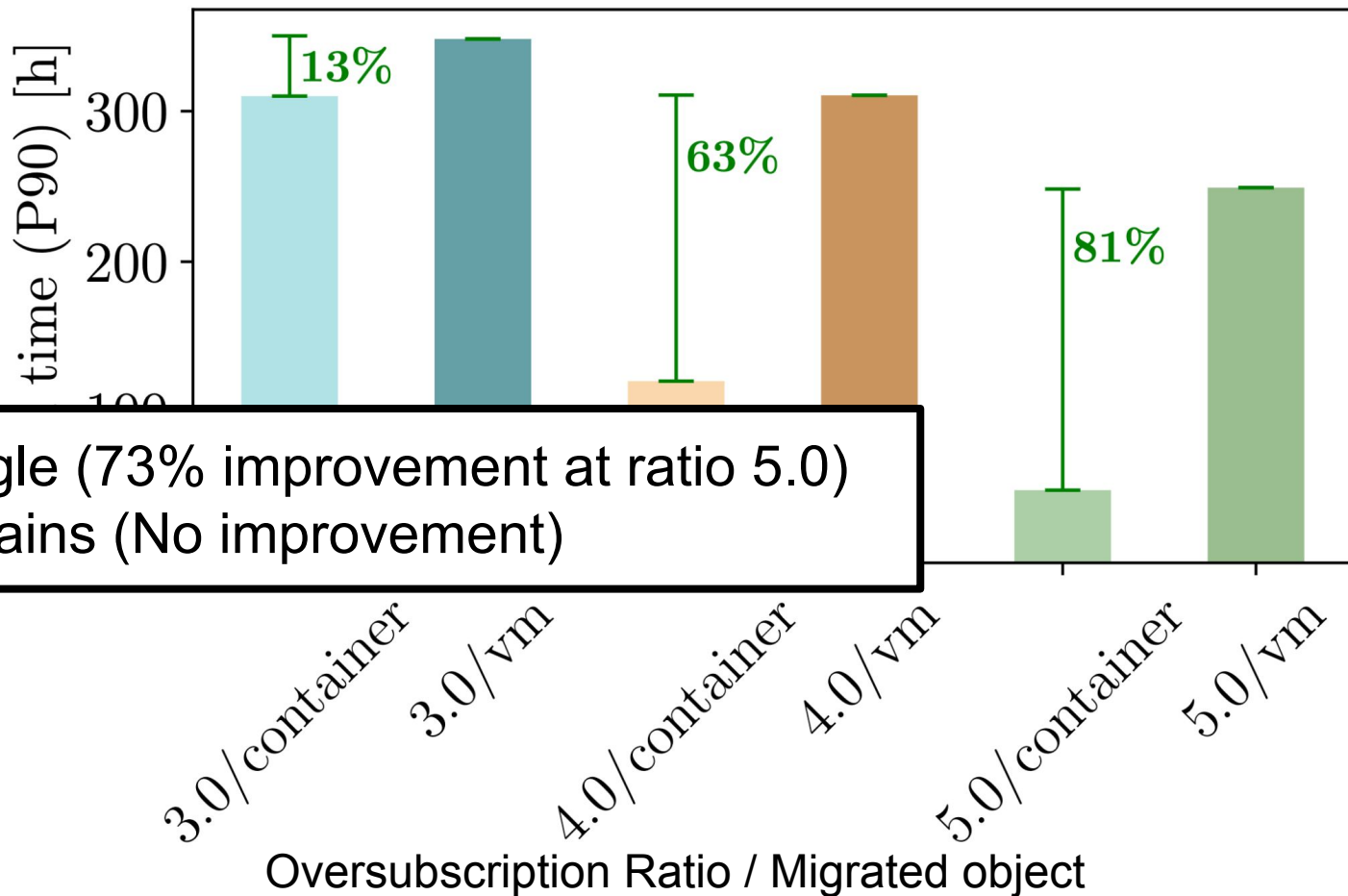
Azure 2017 (~65 machines)

BitBrains 2015 (~100 machines)

Container Migration - Results - Azure P90 App Runtime



Container Migration - Results - Azure P90 App Runtime



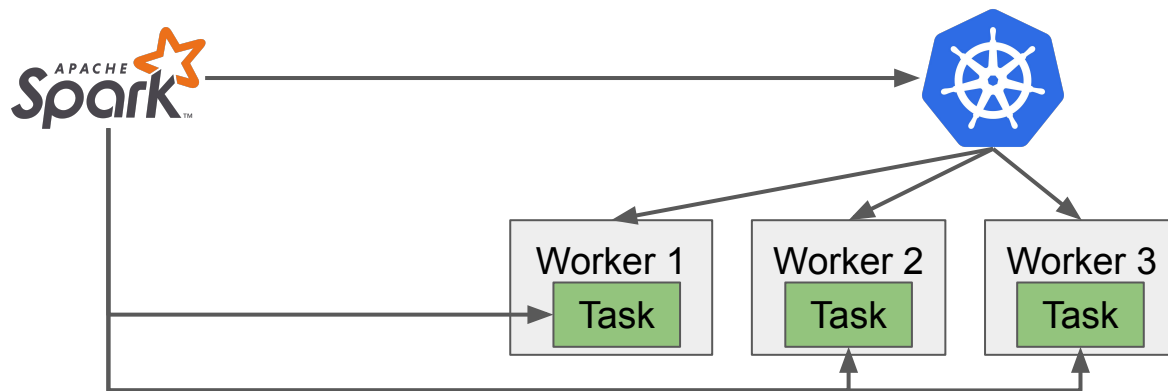
Key Takeaways

- Distributed Systems Runtimes can take advantage of complex APIs for resource management
- Current open-source resource managers are missing complex resource management abstractions
 - Migrations, Data Management
- Complex abstractions offer performance benefits for some workloads, but not all
 - 17% to 81% improve in 90th percentile app runtime using container migration callbacks for Azure trace, but not the Bitbrains trace

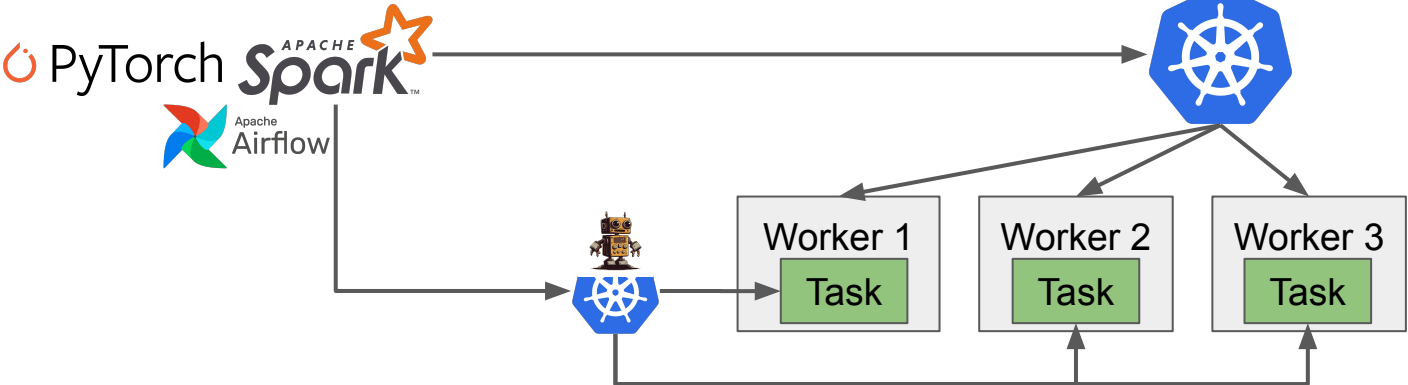
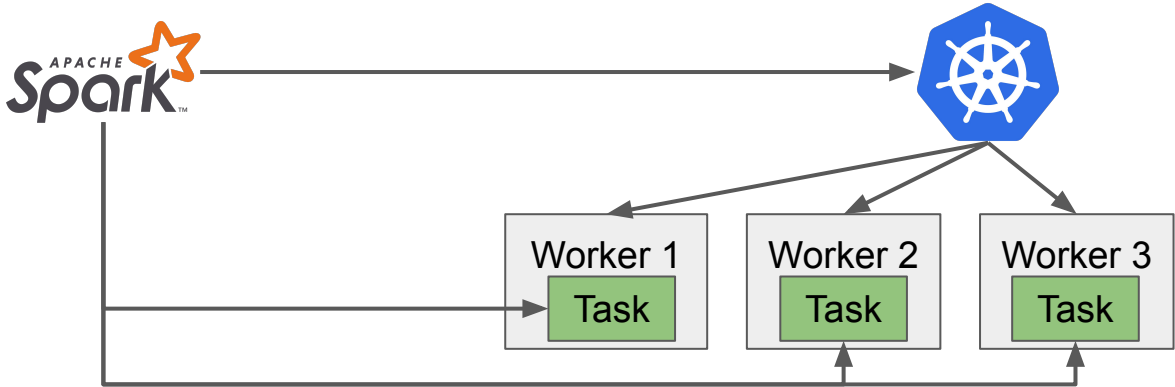
Why didn't these APIs make it into industrial schedulers?

1. New APIs/scheduling techniques only work for specific workloads
 - a. Widely-deployed schedulers need generality
2. Need to implement separately for different systems and system versions
 - a. Can't use Spark 3.2 scheduler with Spark 2.4
 - b. Cannot use a SLURM scheduler with Kubernetes

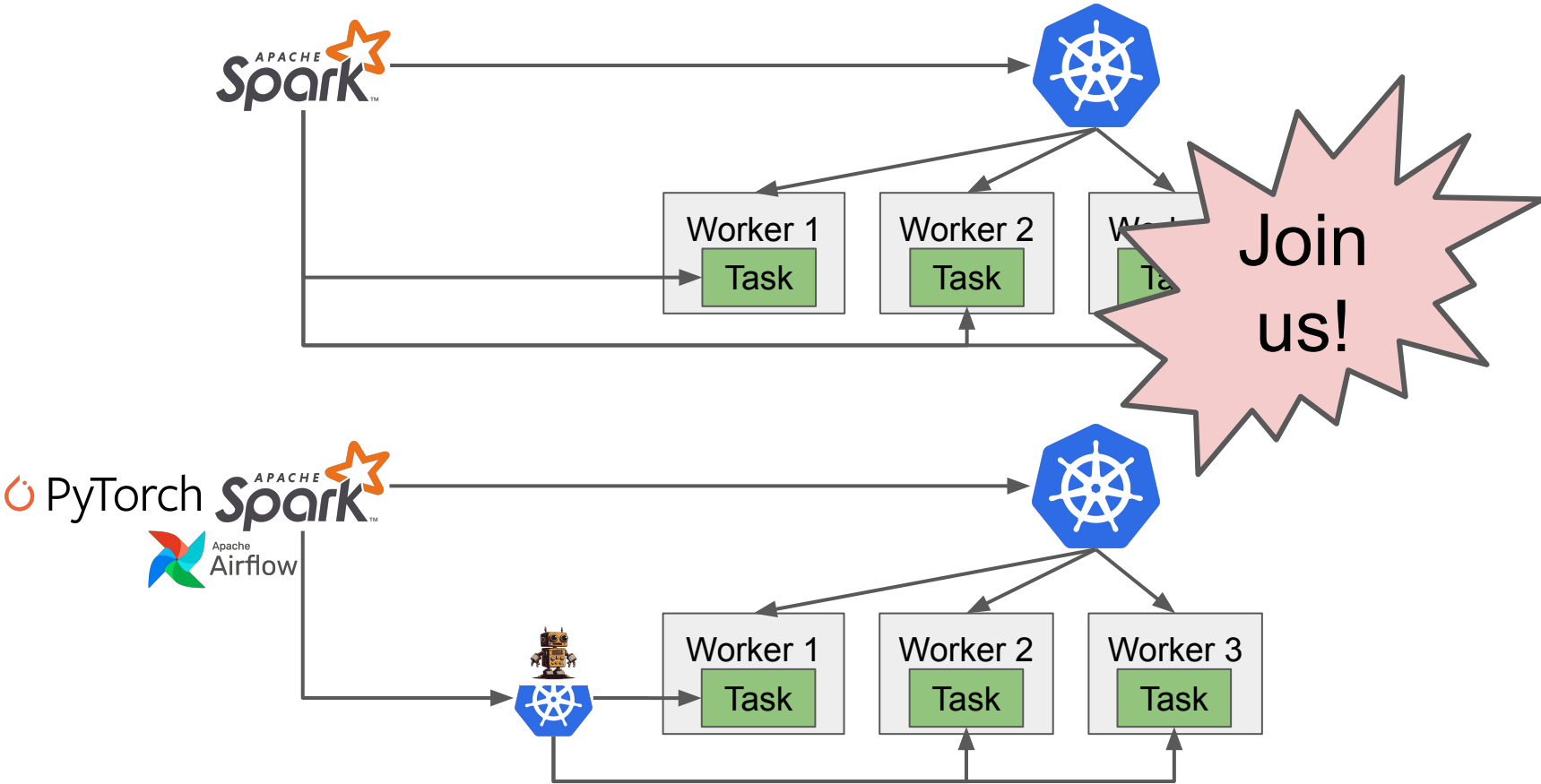
Future Work - Kubernetes-based Embeddable Scheduler



Future Work - Kubernetes-based Embeddable Scheduler



Future Work - Kubernetes-based Embeddable Scheduler



Key Takeaways

- Distributed Systems Runtimes can take advantage of complex APIs for resource management
- Current open-source resource managers are missing complex resource management abstractions
 - Migrations, Data Management
- Complex abstractions offer performance benefits for some workloads, but not all
 - 17% to 81% improve in 90th percentile app runtime using container migration callbacks for Azure trace, but not the Bitbrains trace

Further Reading

[this work] Talluri, S., Herbst, N., Abad, C., De Matteis, T., & Iosup, A. (2024). ExDe: Design space exploration of scheduler architectures and mechanisms for serverless data-processing. *Future Generation Computer Systems*, 153, 84-96.

[related work on scheduler APIs] Manterola Lasa, A., Talluri, S., De Matteis, T., & Iosup, A. (2024, May). The Cost of Simplicity: Understanding Datacenter Scheduler Programming Abstractions. In *Proceedings of the 15th ACM/SPEC International Conference on Performance Engineering* (pp. 166-177).

[using the tools] Talluri, S., Herbst, N., Abad, C., Trivedi, A., & Iosup, A. (2023, May). A Trace-driven Performance Evaluation of Hash-based Task Placement Algorithms for Cache-enabled Serverless Computing. In *Proceedings of the 20th ACM International Conference on Computing Frontiers* (pp. 164-175).

[reference architecture] Andreadis, G., Versluis, L., Mastenbroek, F., & Iosup, A. (2018, November). A reference architecture for datacenter scheduling: design, validation, and experiments. In *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 478-492). IEEE.