Resource provisioning requirements are very diverse

Existing resource provisioning solutions tend to be specialized for specific scenarios.

There is no resource provisioning model that supports all of these usage scenarios simultaneously and efficiently.

The job abstraction is widely used, but has some drawbacks.

Leases are a more general abstraction, but pose challenges

Resource leasing doesn't refer just to the hardware.

A lease should also include the software environment installed on the hardware.

The availability of the lease can be complex, and multiple types must co-exist efficiently.

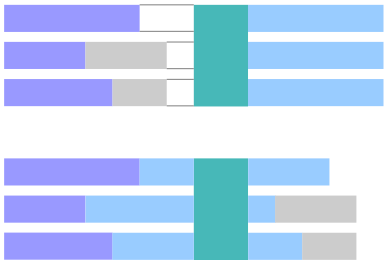
Leases still need to play nice with jobs.

Combining Batch Execution and Leasing Using Virtual Machines

Borja Sotomayor
University of Chicago

Kate Keahey
Argonne National Laboratory

Ian Foster
Argonne National Laboratory
University of Chicago



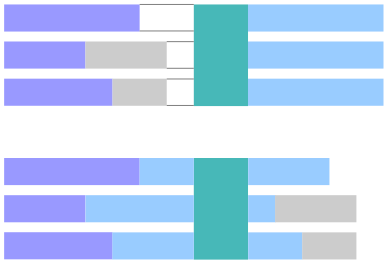
- Resource leases
- Design and Implementation
- Experimental Results
- Conclusions

Combining Batch Execution and Leasing Using Virtual Machines

Borja Sotomayor
University of Chicago

Kate Keahey
Argonne National Laboratory

Ian Foster
Argonne National Laboratory
University of Chicago



- Resource leases
- Design and Implementation
- Experimental Results
- Conclusions

A lease is an agreement with terms encompassing hardware, software and availability.

A lease can request n nodes. Each node requires a (p, m, d, b) tuple of resources.

A lease includes a description of the required software environment.

We use simple availability terms to express **best-effort** and **AR** leases.

- Start time
- Maximum duration
- Preemptability

Resource leases are mapped onto hardware resources

A site has W nodes.

Each node has P CPUs, M Megabytes of memory, and D MB of local storage.

The read/write transfer rate of the disk is H_r and H_w MB/s, respectively.

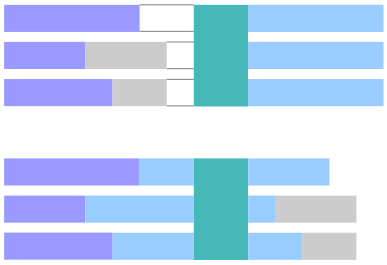
The nodes are connected by a switched network with a bandwidth of B MB/s

Combining Batch Execution and Leasing Using Virtual Machines

Borja Sotomayor
University of Chicago

Kate Keahey
Argonne National Laboratory
University of Chicago

Ian Foster
University of Chicago



Resource leases

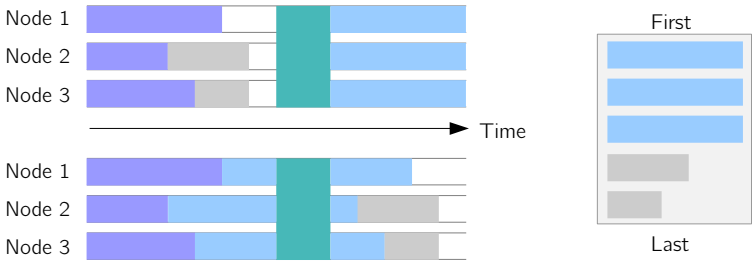
Design and Implementation

Experimental Results

Conclusions

We have designed a lease management architecture called Haizea

Leases are implemented as virtual machines (VMs). Each VM is allocated a (p, m, d, b) tuple. A VM requires a disk image of size S_i



We leverage the suspend/resume/migrate capability of VMs.

Using suspend/resume for scheduling is not a new idea

Preempting schedulers already do this using checkpointing. However, they can't assume that everything is checkpointable.

With VMs, you can suspend any computation transparently without making the software checkpointing-aware...

... but we have to deal with other scheduling challenges.

Haizea schedules overhead instead of deducting it from the user's allocation.

Disk image transfers are scheduled separately from the VMs.

When a lease must start at a specific time, the image transfer is scheduled to arrive before that time.

Images are reused on the nodes using a caching algorithm.

Transfers are integrated into scheduling decisions, with the goal of minimizing image transfers.

We make some assumptions in our resource model

One VM per processor

No contention between application network traffic (generated "inside" the lease) and network traffic from image transfers.

We will relax these in future work

Implementation

We have implemented a prototype of the Haizea lease management architecture in Python.

Currently it runs in simulation (using a discrete event simulation methodology)

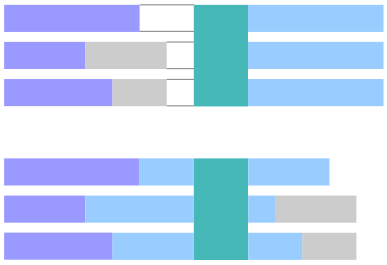
Accepts a workload of best-effort and AR leases as input and produces the schedule as output.

Combining Batch Execution and Leasing Using Virtual Machines

Borja Sotomayor
University of Chicago

Kate Keahey
Argonne National Laboratory

Ian Foster
Argonne National Laboratory
University of Chicago



Our experiments used workloads combining best-effort and AR leases

The base workload is an SDSC Blue Horizon job trace taken from the Parallel Workloads Archive. Each job becomes a best-effort lease with p=1 and m=1024.

We generate 72 workloads by artificially injecting AR leases according to three parameters.

ρ : aggregate duration of AR leases
5% 10% 15%
20% 25% 30%

δ : average duration of AR leases
1h 2h 3h 4h

ν : number of nodes in AR leases
1-24 (small) 15-48 (medium)
49-72 (large)

Inter-lease interval computed based on these parameters

AR leases are requested 24 hours in advance

We measure three metrics

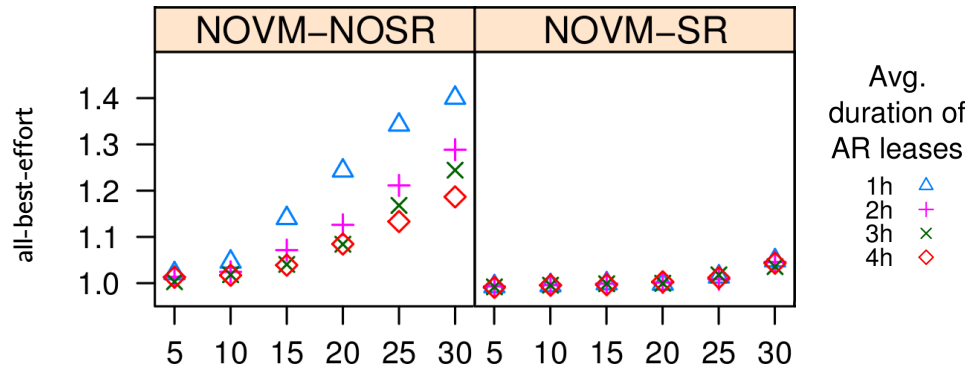
One per workload and configuration. Smaller is better.

all-best-effort: Time from the start of experiment to the moment the last best-effort lease is completed. Normalized relative to the time to run all best-effort leases without any injected ARs.

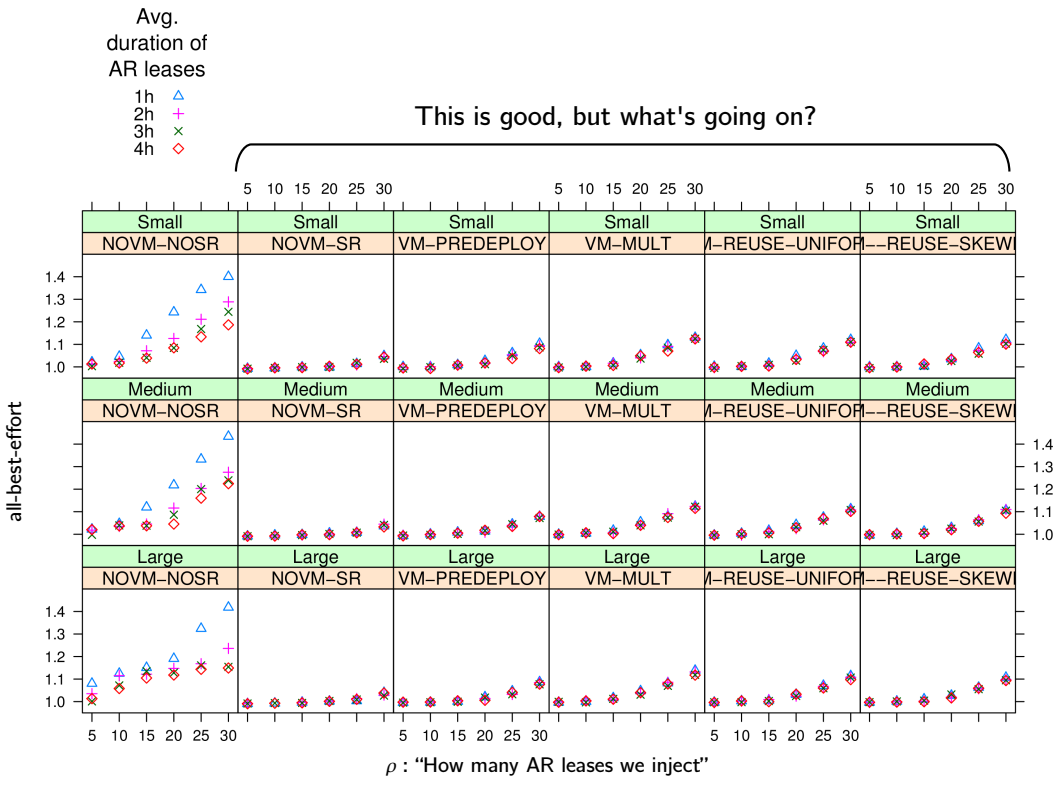
δ = 4h

No VMs, no suspend/resume
(NO-VM-NOSR)

ν = small



ρ : "How many AR leases we inject"



ρ : "How many AR leases we inject"

We measure three metrics

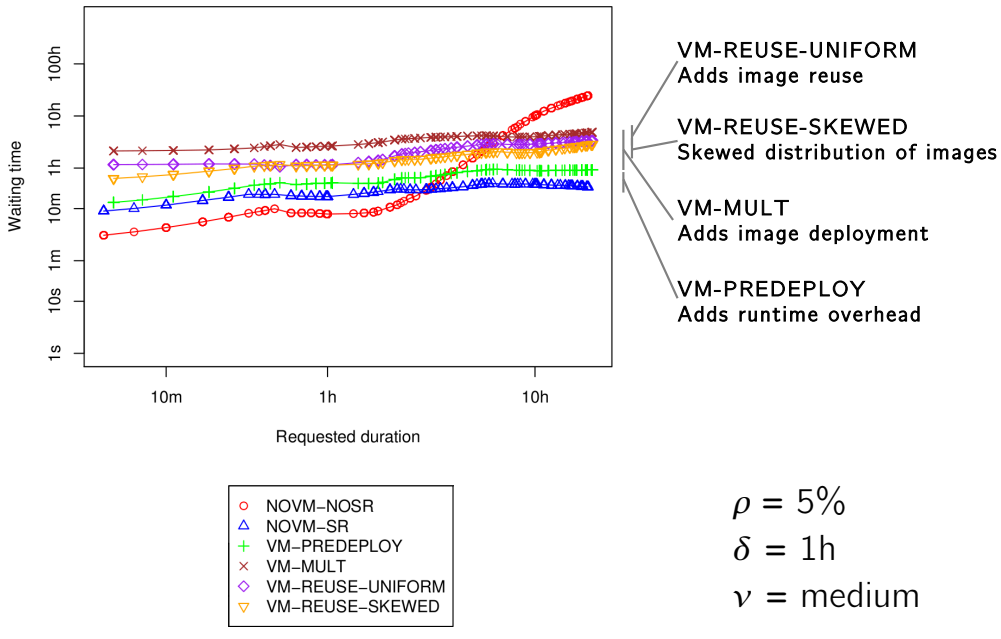
One per workload and configuration. Smaller is better.

all-best-effort: Time from the start of experiment to the moment the last best-effort lease is completed. Normalized relative to the time to run all best-effort leases without any injected ARs.

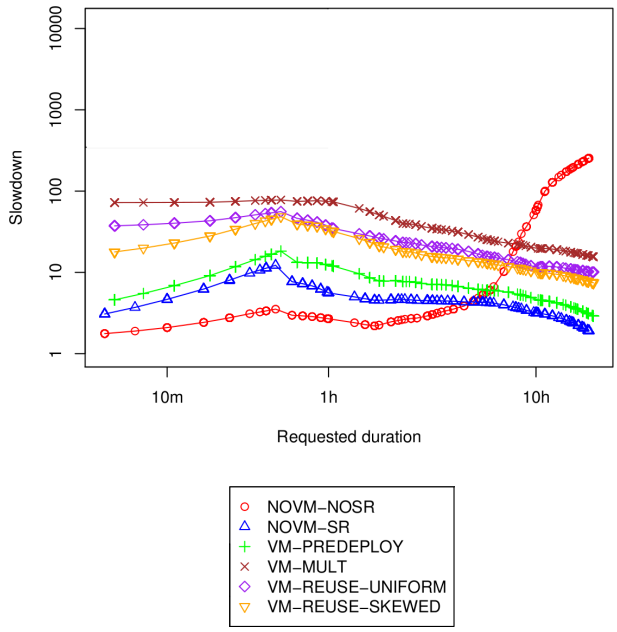
One per lease. Smaller is better.

Wait time: Time from submission of a best-effort lease to the moment it starts running.

Bounded slowdown: Time from submission of a best-effort lease to the moment it finishes, divided by the time it would take to run on an unloaded system.



ρ = 5%
δ = 1h
ν = medium

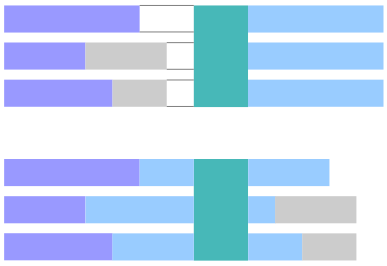


Combining Batch Execution and Leasing Using Virtual Machines

Borja Sotomayor
University of Chicago

Kate Keahey
Argonne National Laboratory
University of Chicago

Ian Foster
University of Chicago



Resource leases

Design and Implementation

Experimental Results

Conclusions

VMs can implement leases more efficiently than traditional schedulers, but several issues must be dealt with

Haizea allows users to request resource leases that encompass hardware resources, software environments, and availability.

Our experimental results show that a VM-based approach can overcome the utilization problems typically associated with ARs.

Deployment overhead must be managed, as it can have a considerable impact on performance.

Our results can be improved

The all-best-effort metric gives an intuitive understanding of utilization, but we need more concrete data to understand the trade-offs.

We need to find out what happens when we alter some of the parameters, such as disk image size, memory size, network bandwidth, advance notice, etc.

We need to run this on real hardware, which will require dealing with hardware failures.

Haizea now manages real hardware, and we are close to a public release

Haizea has been integrated with the OpenNebula virtual infrastructure manager (http://www.opennebula.org/).

OpenNebula+Haizea will be used to manage virtual infrastructure in the EU Reservoir project (http://www.reservoir-fp7.eu/)

An Apache2-licensed release is forthcoming.

http://haizea.cs.uchicago.edu/

OpenNEbula.org HAIZEA

The next step is improving our model and researching policies

We consider just two types of leases: preemptible best-effort and non-preemptible AR.

- What other leases are there in nature, and how will they affect our model?

The model assumes that lease preparation only requires transferring a VM image.

- This is a realistic use case, but there are *a lot* of other ways of preparing a lease.

We consider a trivial all-accept policy.

- How will non-trivial policies affect performance? What lease terms should affect those policies?

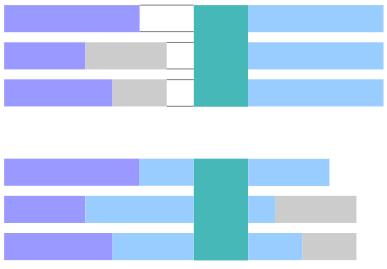
Acknowledgements

Anne Rogers (University of Chicago)

Work funded by NSF grant #509408 "Virtual Playgrounds"

Originally developed in the Globus Virtual Workspaces group at ANL/UC (http://workspace.globus.org/)

Combining Batch Execution and Leasing Using Virtual Machines



Borja Sotomayor
borja@cs.uchicago.edu
University of Chicago

Kate Keahey Ian Foster
{keahey,foster}@mcs.anl.gov
Argonne National Laboratory
University of Chicago