VMPlaceS: A Generic Tool to Investigate and Compare VM Placement Algorithms

Adrien Lebre, Jonathan Pastor, Mario Südholt
ASCOLA Research Group
(Mines Nantes, Inria, LINA)

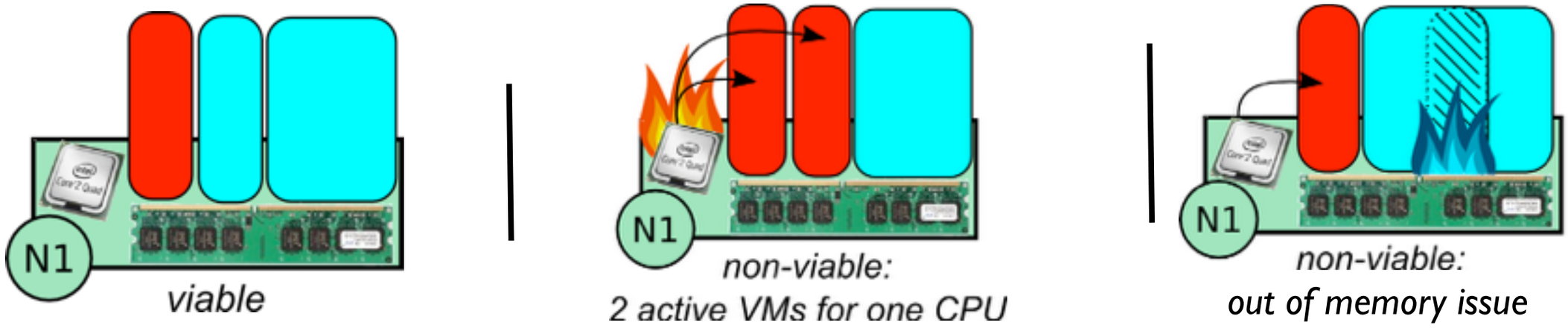
Agenda

- VM Placement Problem and challenges
- VMPlaces - a generic simulator to evaluate and compare VM placement strategies
- Use-case: Evaluation of three strategies aiming at mitigating CPU violations
- Conclusion/Future work

VM Placement Problem

- Fine management of resources: maximize the usage of CC resources while guaranteeing VM resource requirements (aka, SLAs)

- Find the ‘’right’’ mapping between needs of VMs and resources provided by PMs



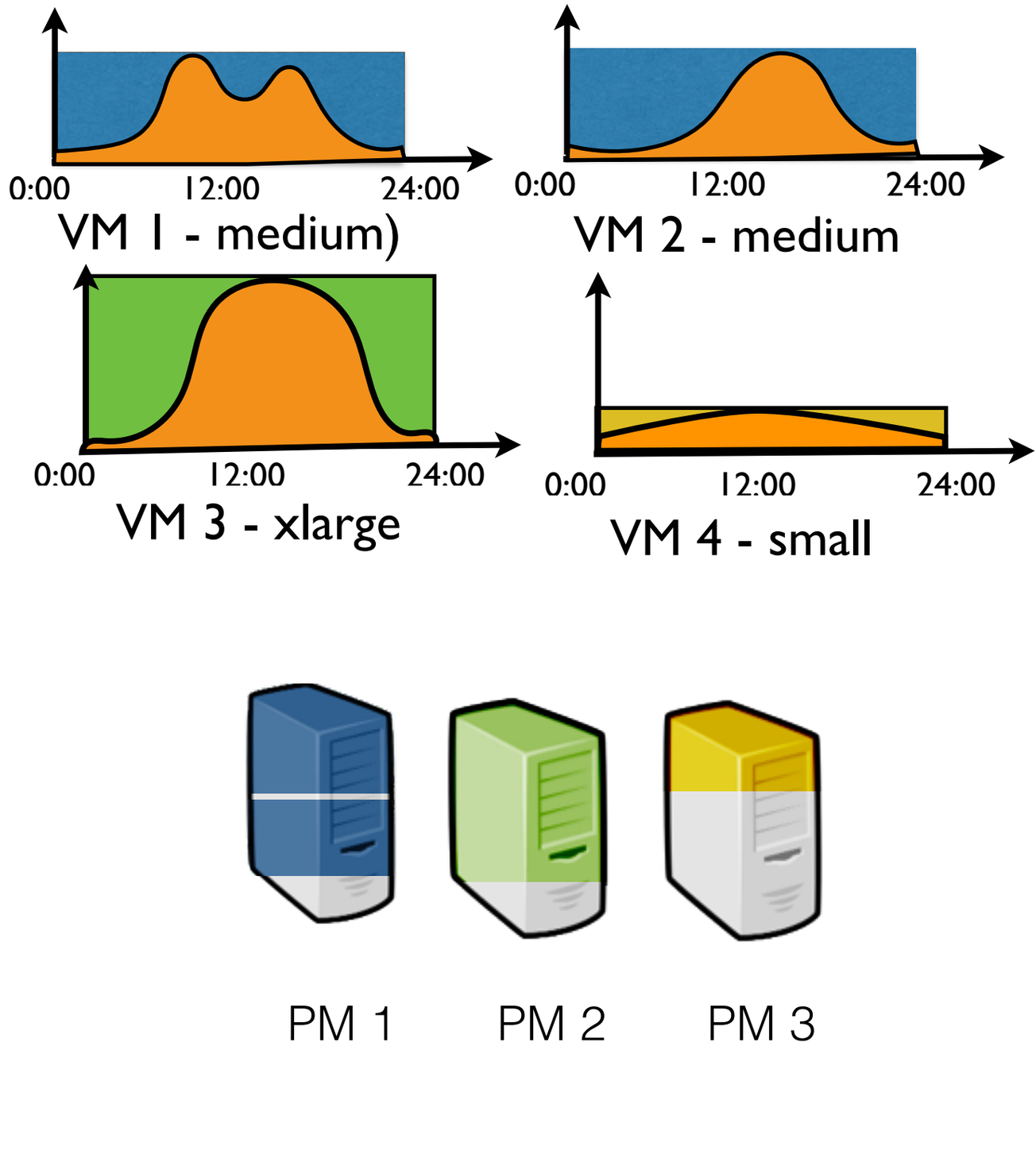
credits: F. Hermenier, OSDI poster session 2008

Challenge 1 - Fluctuations of VM Requirements

- Static placement policies (as delivered by most of the popular Cloud Computing management systems)

  ‘’Simple’’ but prevent CC providers to maximize the usage of CC resources (and thus their revenue)

- Advanced dynamic placement strategies to relocate VMs according to the scheduler objectives / available resources / waiting queue / ...

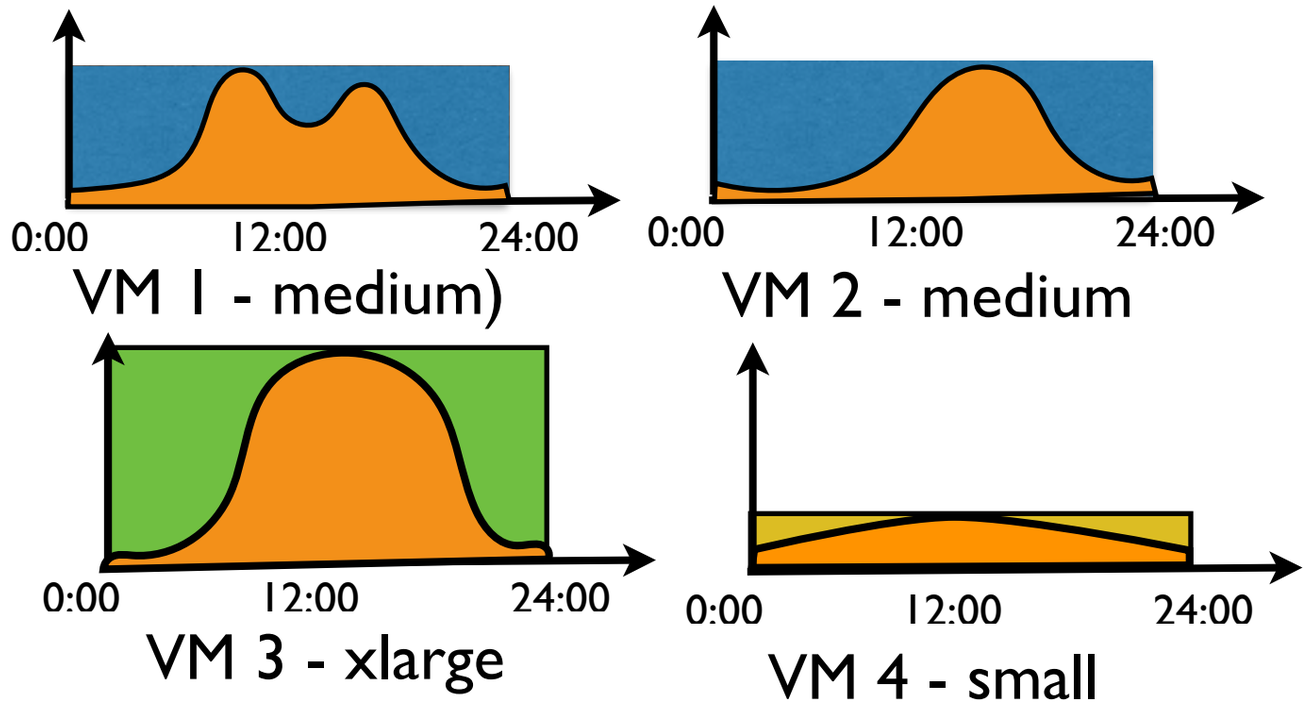


Challenge 1 - Fluctuations of VM Requirements

- Static placement policies (as delivered by most of the popular Cloud Computing management systems)

  ‘’Simple’’ but prevent CC providers to maximize the usage of CC resources (and thus their revenue)

- Advanced dynamic placement strategies to relocate VMs according to the scheduler objectives / available resources / waiting queue / ...



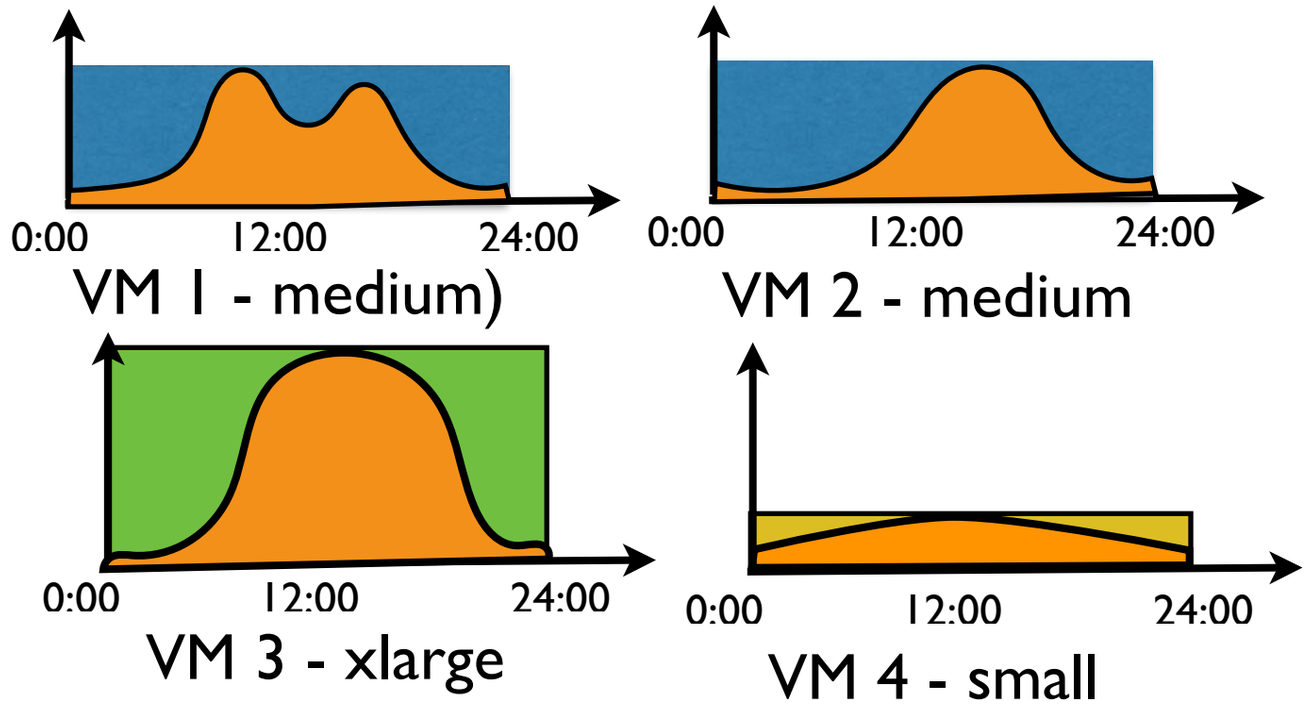
PM 1 PM 2

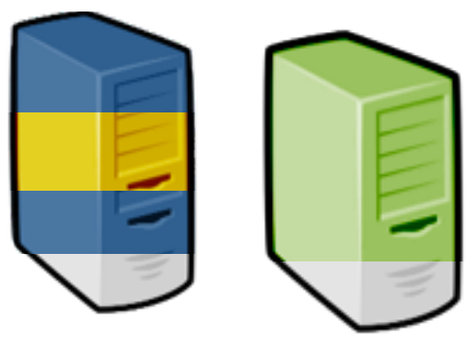
Challenge 1 - Fluctuations of VM Requirements

- Static placement policies (as delivered by most of the popular Cloud Computing management systems)

  ‘’Simple’’ but prevent CC providers to maximize the usage of CC resources (and thus their revenue)

- Advanced dynamic placement strategies to relocate VMs according to the scheduler objectives / available resources / waiting queue / ...





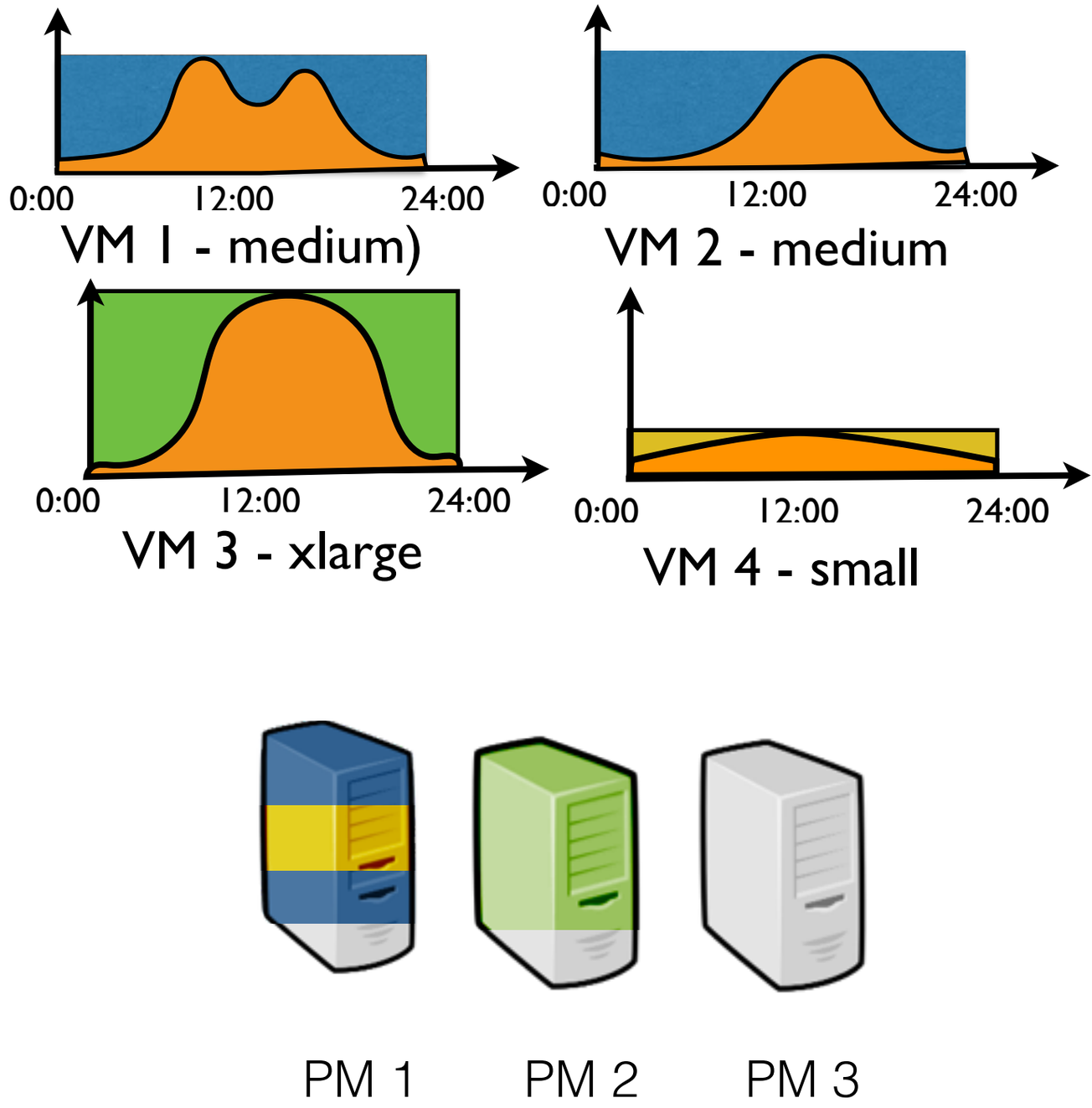
PM 1 PM 2

Challenge 1 - Fluctuations of VM Requirements

- Static placement policies (as delivered by most of the popular Cloud Computing management systems)

  ‘’Simple’’ but prevent CC providers to maximize the usage of CC resources (and thus their revenue)

- Advanced dynamic placement strategies to relocate VMs according to the scheduler objectives / available resources / waiting queue / ...

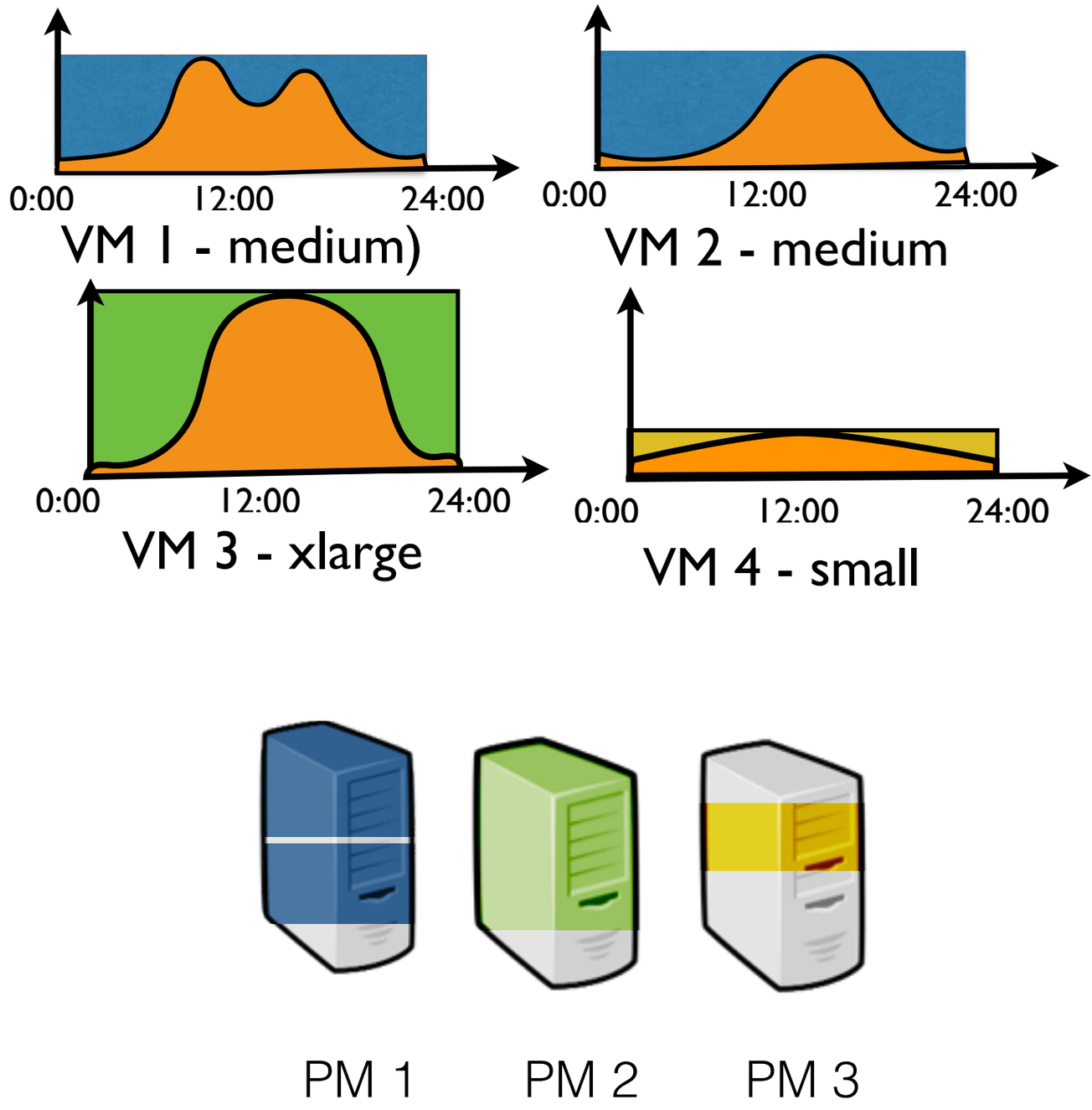


Challenge 1 - Fluctuations of VM Requirements

- Static placement policies (as delivered by most of the popular Cloud Computing management systems)

  ‘’Simple’’ but prevent CC providers to maximize the usage of CC resources (and thus their revenue)

- Advanced dynamic placement strategies to relocate VMs according to the scheduler objectives / available resources / waiting queue / ...

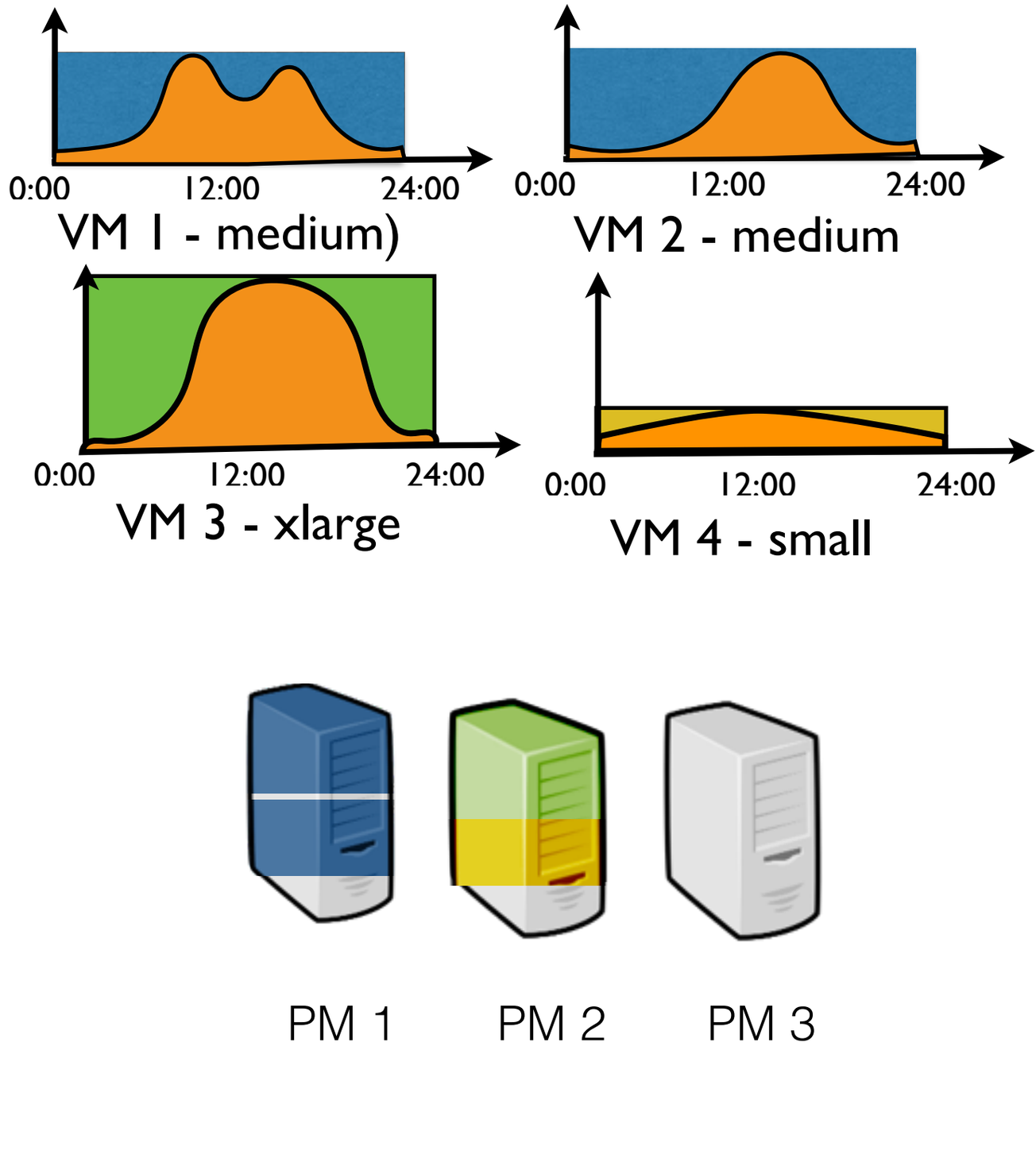


Challenge 1 - Fluctuations of VM Requirements

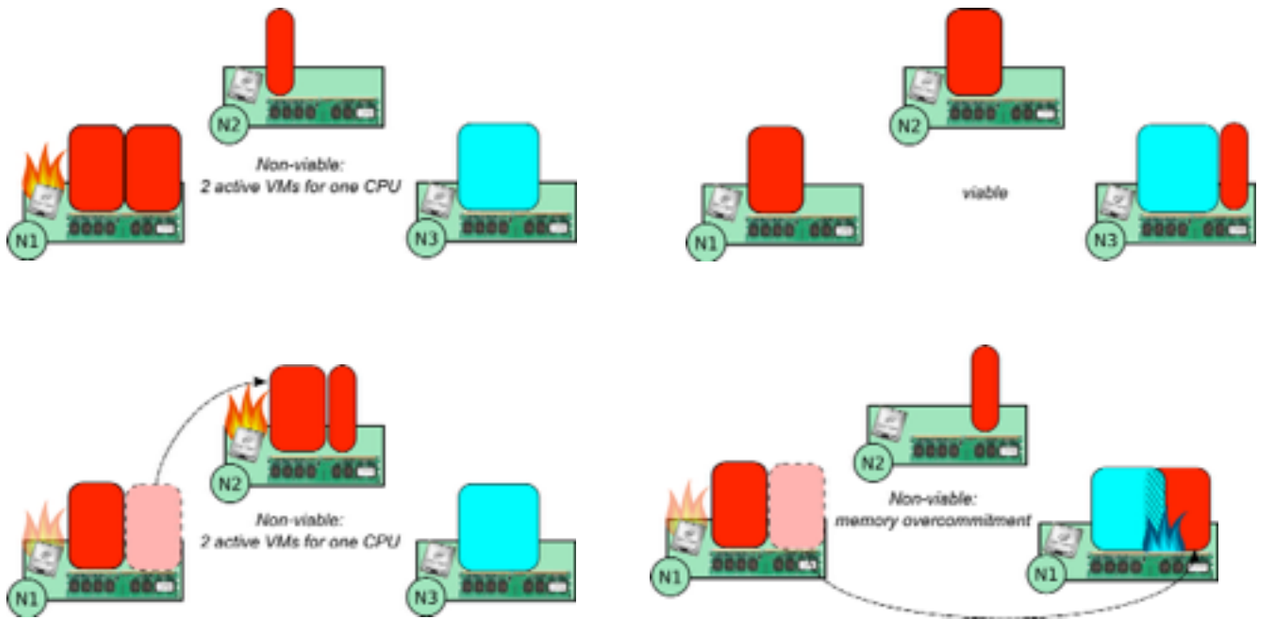
- Static placement policies (as delivered by most of the popular Cloud Computing management systems)

  ‘’Simple’’ but prevent CC providers to maximize the usage of CC resources (and thus their revenue)

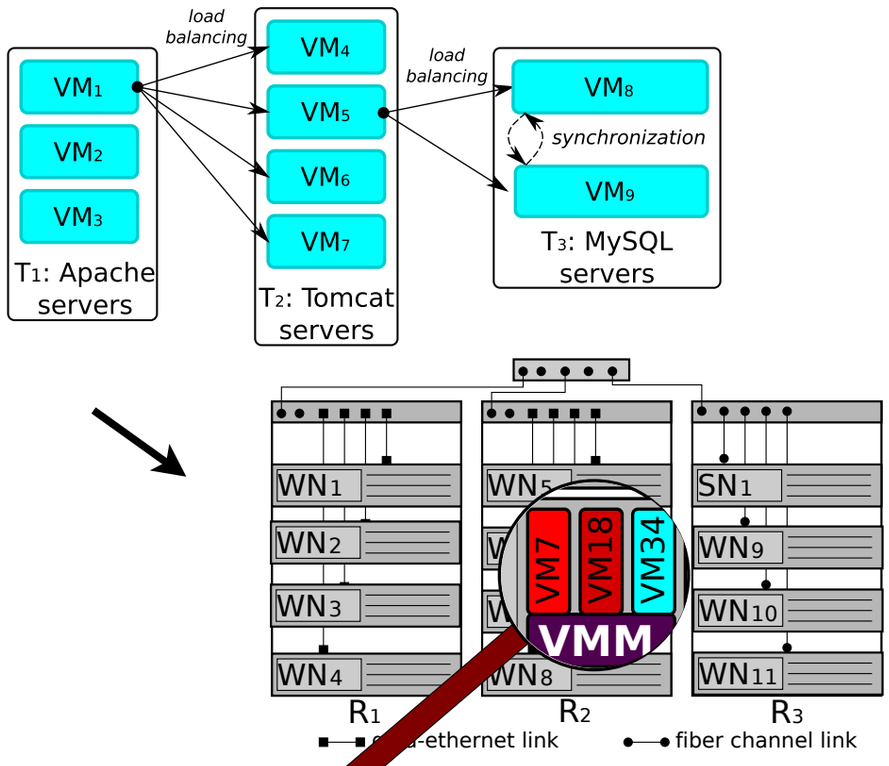
- Advanced dynamic placement strategies to relocate VMs according to the scheduler objectives / available resources / waiting queue / ...



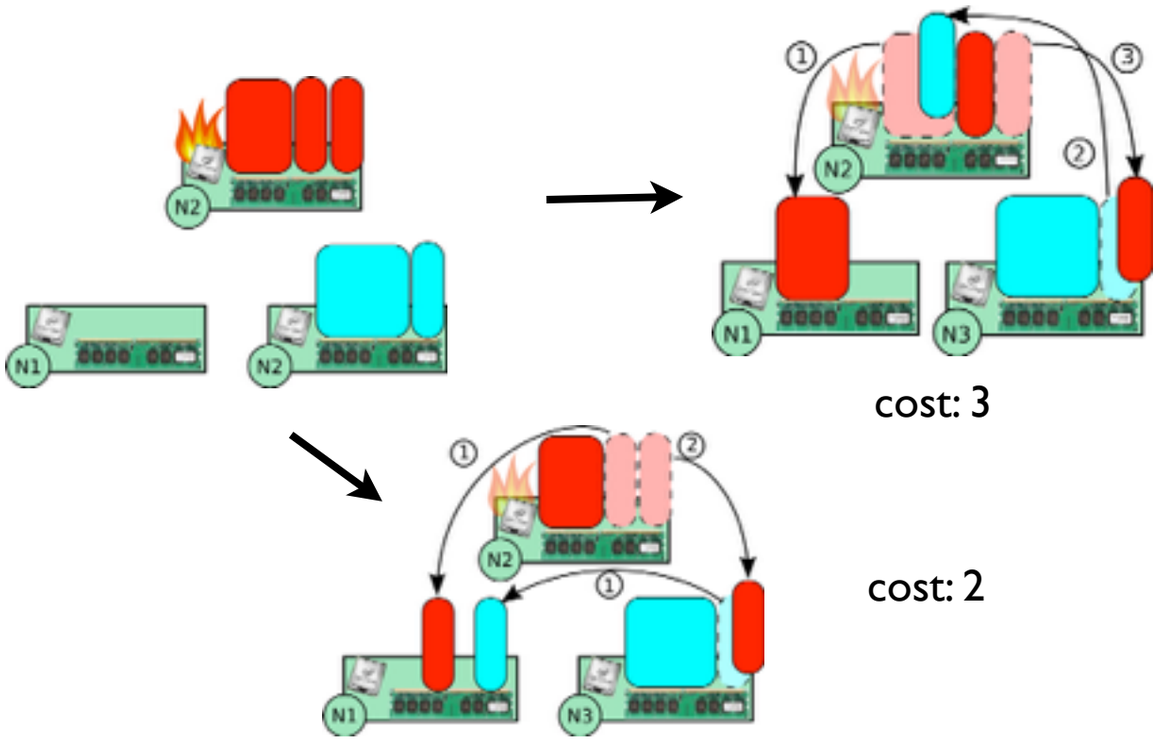
Challenge 2 - Manipulation constraints



Non-viable manipulations - SLA violations

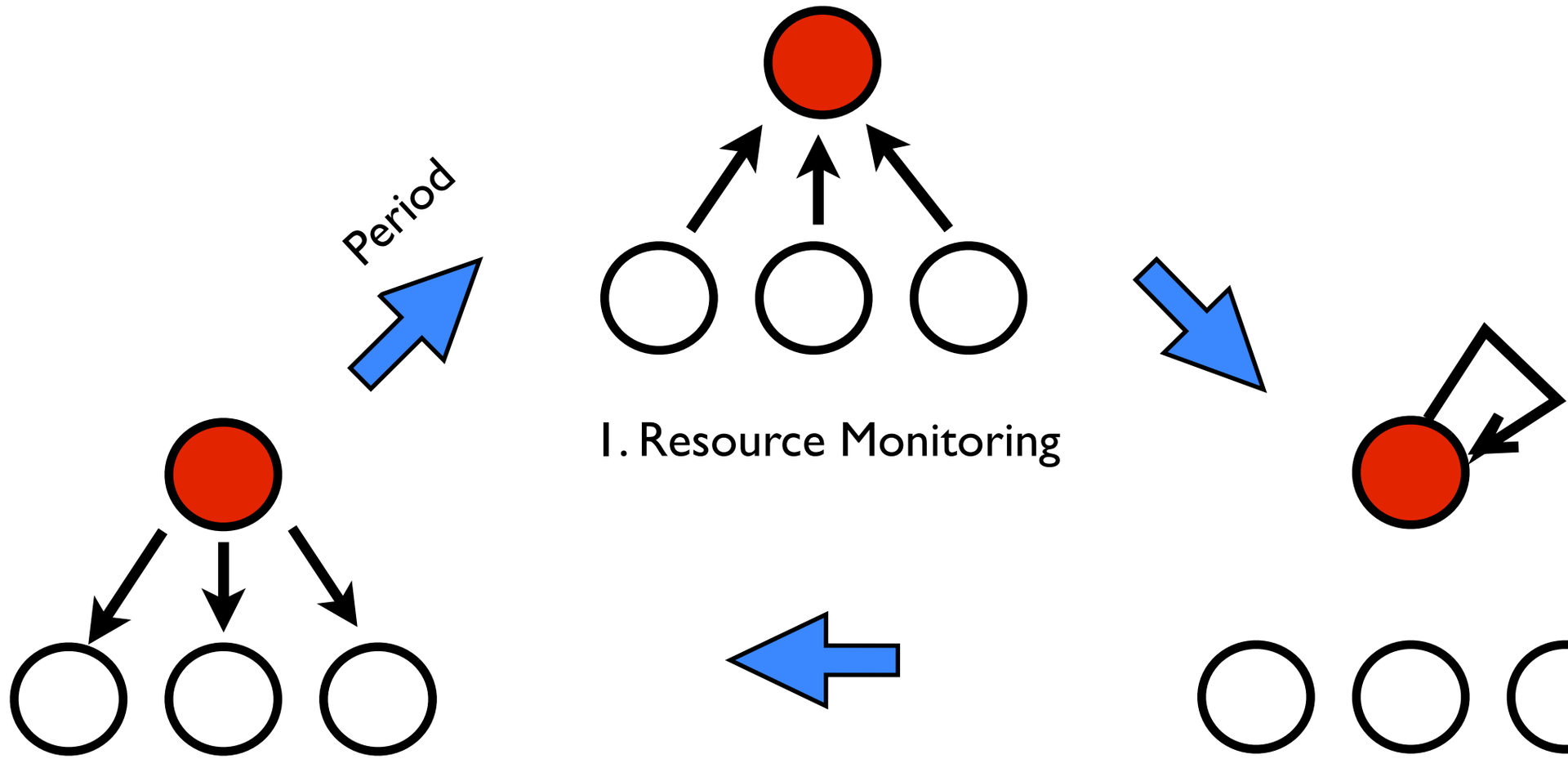


- Optimize the mapping according the needs of VMs, **their placement constraints** and the resources provided by PMs

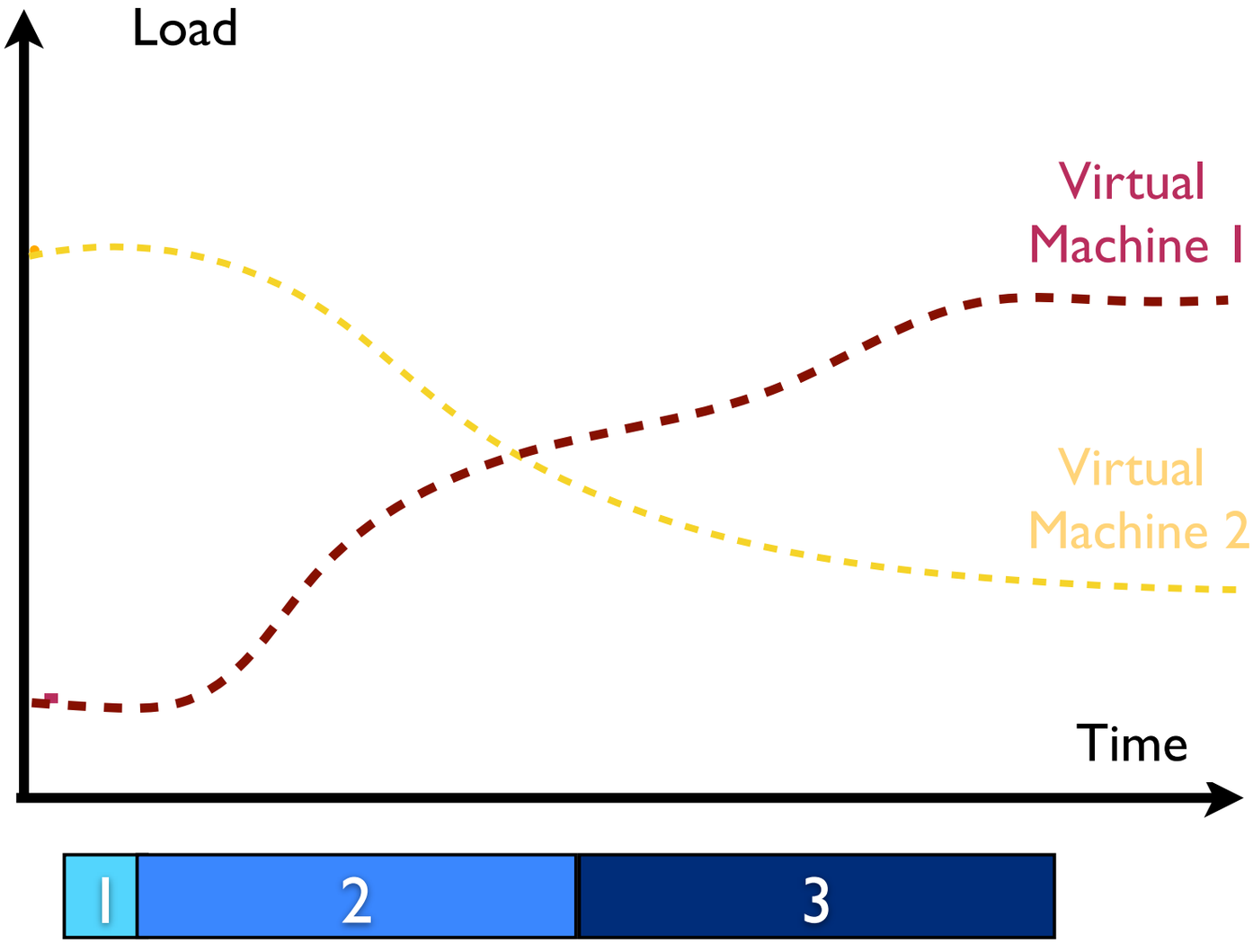


credits: F. Hermenier, Sophia Antipolis University, www.btrplace.org

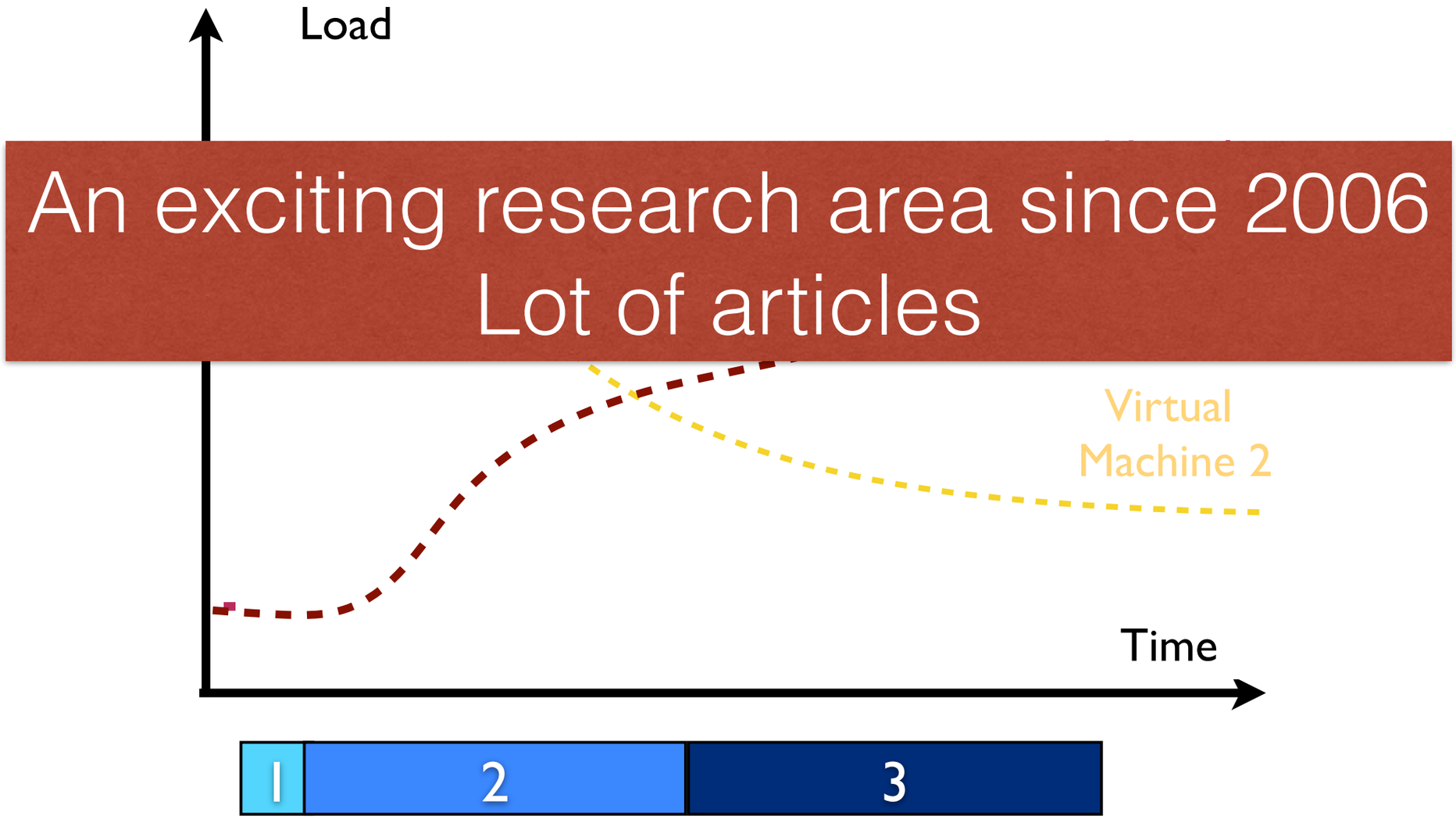
Challenge 3 - Reactivity/Scalability Criteria



Challenge 3 - Reactivity/Scalability Criteria



Challenge 3 - Reactivity/Scalability Criteria



VM Placement Algorithms

Web Images More... Sign in

Google allintitle: placement OR consolidation OR scheduler "virtual machine"

Scholar About 138 results (0.02 sec) My Citations

Articles
Case law
My library

Any time
Since 2015
Since 2014
Since 2011
Custom range...
2008 —
Search

Sort by relevance
Sort by date

include patents
include citations

Publication: **IEEE OR ACM OR SPRINGER**

Improving the scalability of data center networks with traffic-aware **virtual machine placement** [PDF] from ntu.edu.tw
X Meng, V Pappas, L Zhang - ... , 2010 Proceedings **IEEE**, 2010 - ieeexplore.**ieee**.org
Abstract—The scalability of modern data centers has become a practical concern and has attracted significant attention in recent years. In contrast to existing solutions that require changes in the network architecture and the routing protocols, this paper proposes using ...
Cited by 440 Related articles All 15 versions Cite Save

Optimal **virtual machine placement** across multiple cloud providers
S Chaisiri, BS Lee, D Niyato - ... , 2009. APSCC 2009. **IEEE** Asia- ..., 2009 - ieeexplore.**ieee**.org
Abstract—Cloud computing provides users an efficient way to dynamically allocate computing resources to meet demands. Cloud providers can offer users two payment plans, ie, reservation and on-demand plans for resource provisioning. Price of resources in ...
Cited by 172 Related articles All 3 versions Cite Save

Multi-objective **virtual machine placement** in virtualized data center environments
J Xu, JAB Fortes - ...), 2010 **IEEE/ACM** Int'l Conference on & Int'l ..., 2010 - ieeexplore.**ieee**.org
Abstract—Server **consolidation** using virtualization technology has become increasingly important for improving data center efficiency. It enables one physical server to host multiple independent virtual machines (VMs), and the transparent movement of workloads from ...
Cited by 161 Related articles All 6 versions Cite Save

A network-aware **virtual machine placement** and migration approach in cloud computing [PDF] from uow.edu.au
JT Piao, J Yan - Grid and Cooperative Computing (GCC), 2010 ..., 2010 - ieeexplore.**ieee**.org
Abstract—Cloud computing represents a major step up in computing whereby shared computation resources are provided on demand. In such a scenario, applications and data thereof can be hosted by various networked virtual machines (VMs). As applications, ...
Cited by 80 Related articles All 10 versions Cite Save

Reducing electricity cost through **virtual machine placement** in high performance computing clouds [PDF] from unam.mx
K Le, R Bianchini, J Zhang, Y Jaluria, J Meng... - Proceedings of 2011 ..., 2011 - dl.**acm**.org
Abstract In this paper, we first study the impact of load **placement** policies on cooling and

VM Placement Algorithms

Web Images More... Sign In

Google allintitle: placement OR consolidation OR scheduler "virtual machine"

Scholar About 138 results (0.02 sec) My Citations

Web Images More...

Google allintitle: server consolidation

Scholar About 50 results (0.02 sec) My Citations

Articles
Case law
My library

Any time
Since 2015
Since 2014
Since 2011
Custom range...

Sort by relevance
Sort by date

include patents
include citations

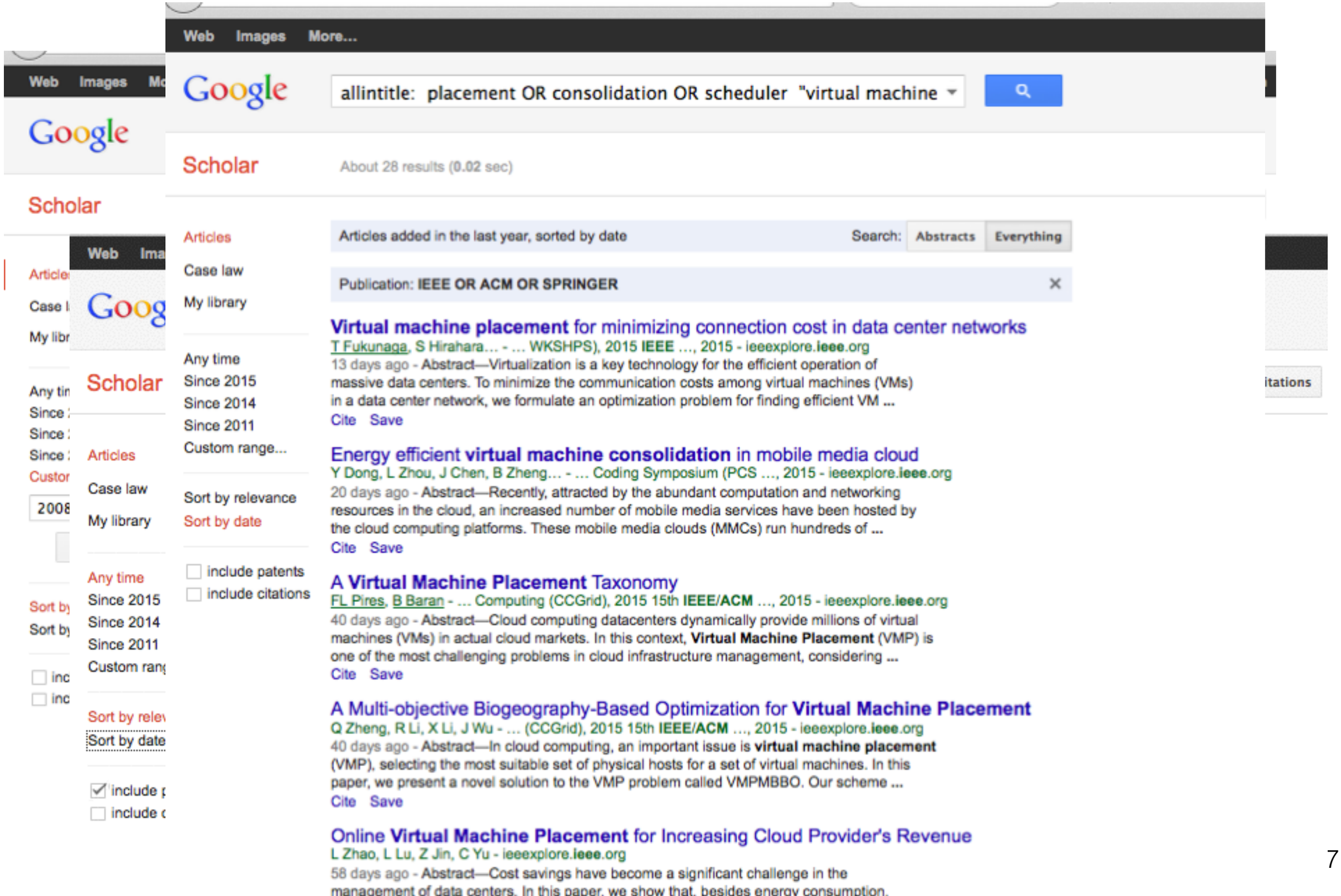
Publication: IEEE OR ACM OR SPRINGER

Virtual hierarchies to support server consolidation [PDF] from wisc.edu
MR Marty, MD Hill - ACM SIGARCH Computer Architecture News, 2007 - dl.acm.org
Abstract **Server consolidation** is becoming an increasingly popular technique to manage and utilize systems. This paper develops CMP memory systems for **server consolidation** where most sharing occurs within Virtual Machines (VMs). Our memory systems maximize ...
Cited by 185 Related articles All 8 versions Cite Save

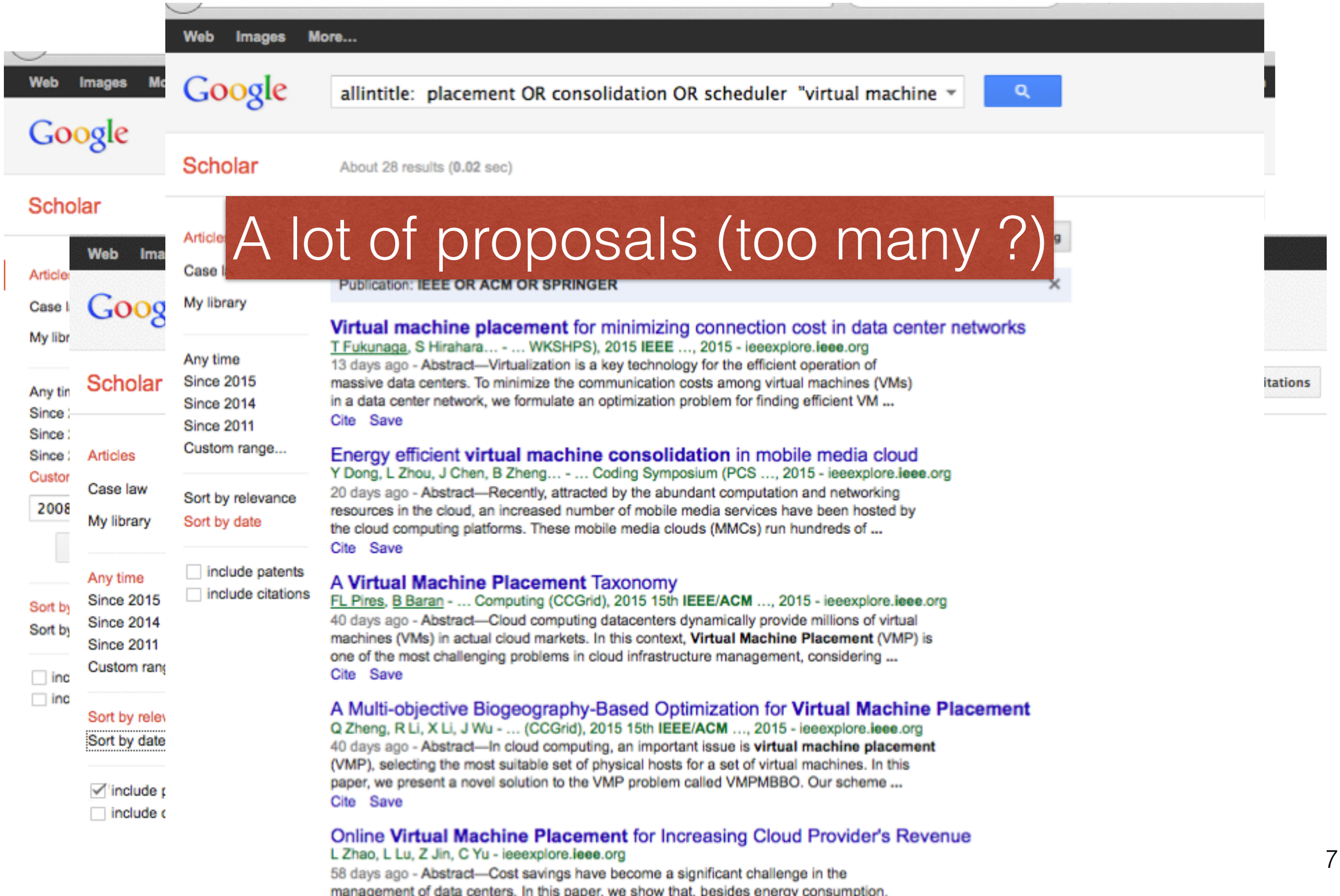
Beyond server consolidation [HTML] from acm.org
W Vogels - Queue, 2008 - dl.acm.org
Virtualization technology was developed in the late 1960s to make more efficient use of hardware. Hardware was expensive, and there was not that much available. Processing was largely outsourced to the few places that did have computers. On a single IBM ...
Cited by 151 Related articles All 2 versions Cite Save

A mathematical programming approach for server consolidation problems in virtualized data centers [PDF] from tum.de
B Speitkamp, M Bichler - Services Computing, IEEE ..., 2010 - ieeexplore.ieee.org
Abstract—Today's data centers offer IT services mostly hosted on dedicated physical servers. **Server** virtualization provides a technical means for **server consolidation**. Thus, multiple virtual servers can be hosted on a single **server**. **Server consolidation** describes ...
Cited by 143 Related articles All 8 versions Web of Science: 39 Cite Save

VM Placement Algorithms

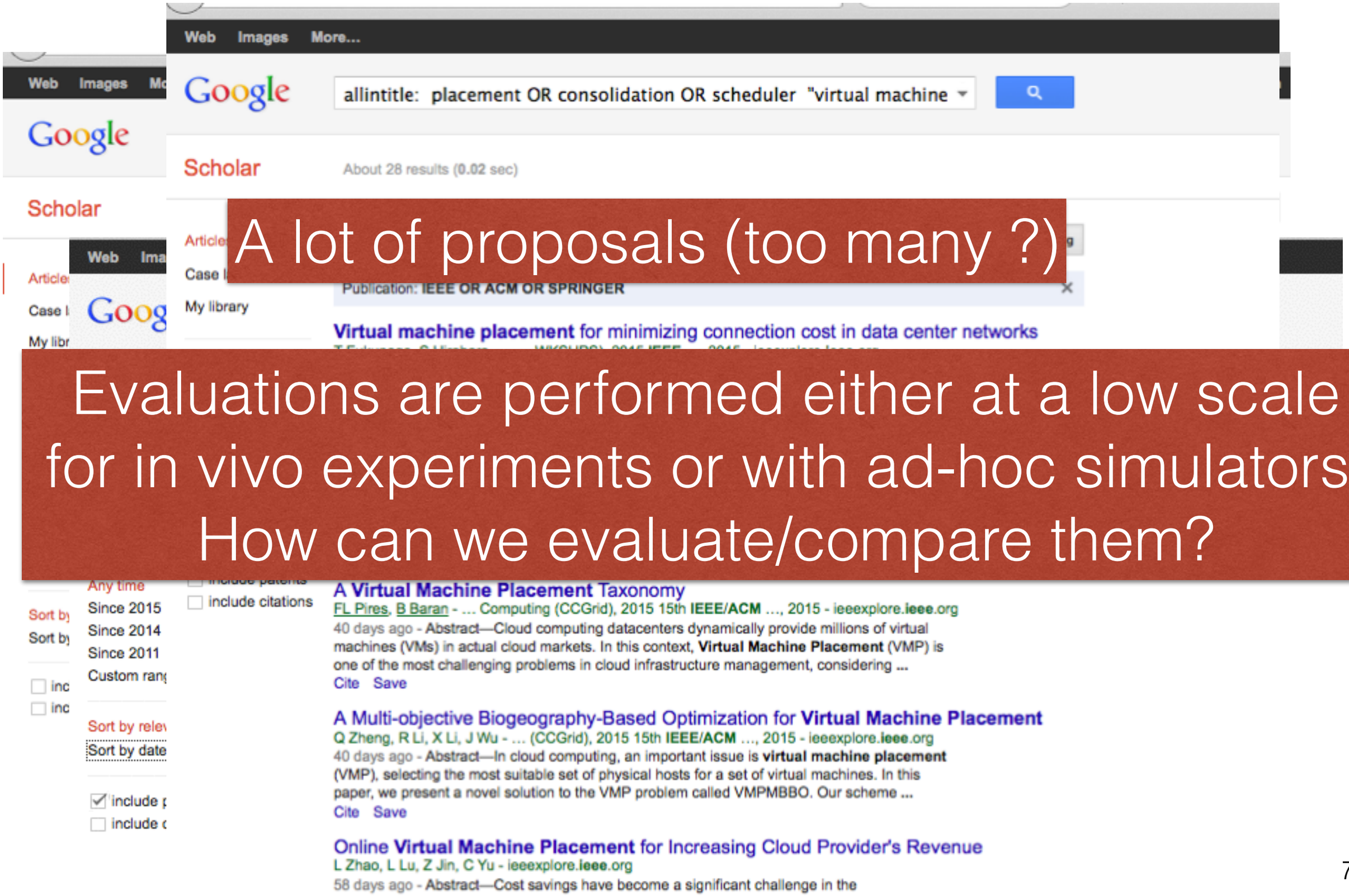


VM Placement Algorithms



A lot of proposals (too many ?)

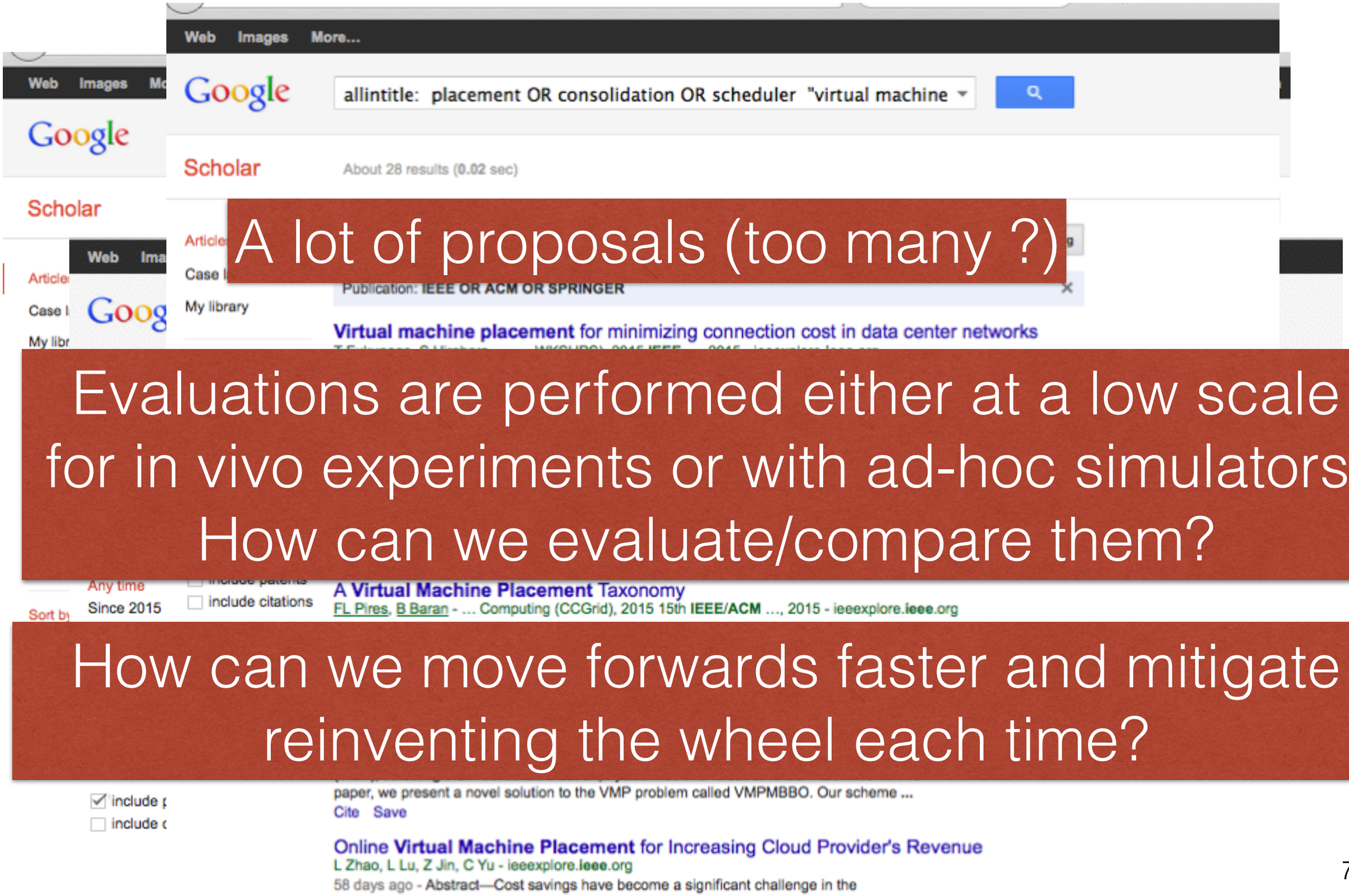
VM Placement Algorithms



A lot of proposals (too many ?)

Evaluations are performed either at a low scale for in vivo experiments or with ad-hoc simulators. How can we evaluate/compare them?

VM Placement Algorithms



A lot of proposals (too many ?)

Evaluations are performed either at a low scale for in vivo experiments or with ad-hoc simulators. How can we evaluate/compare them?

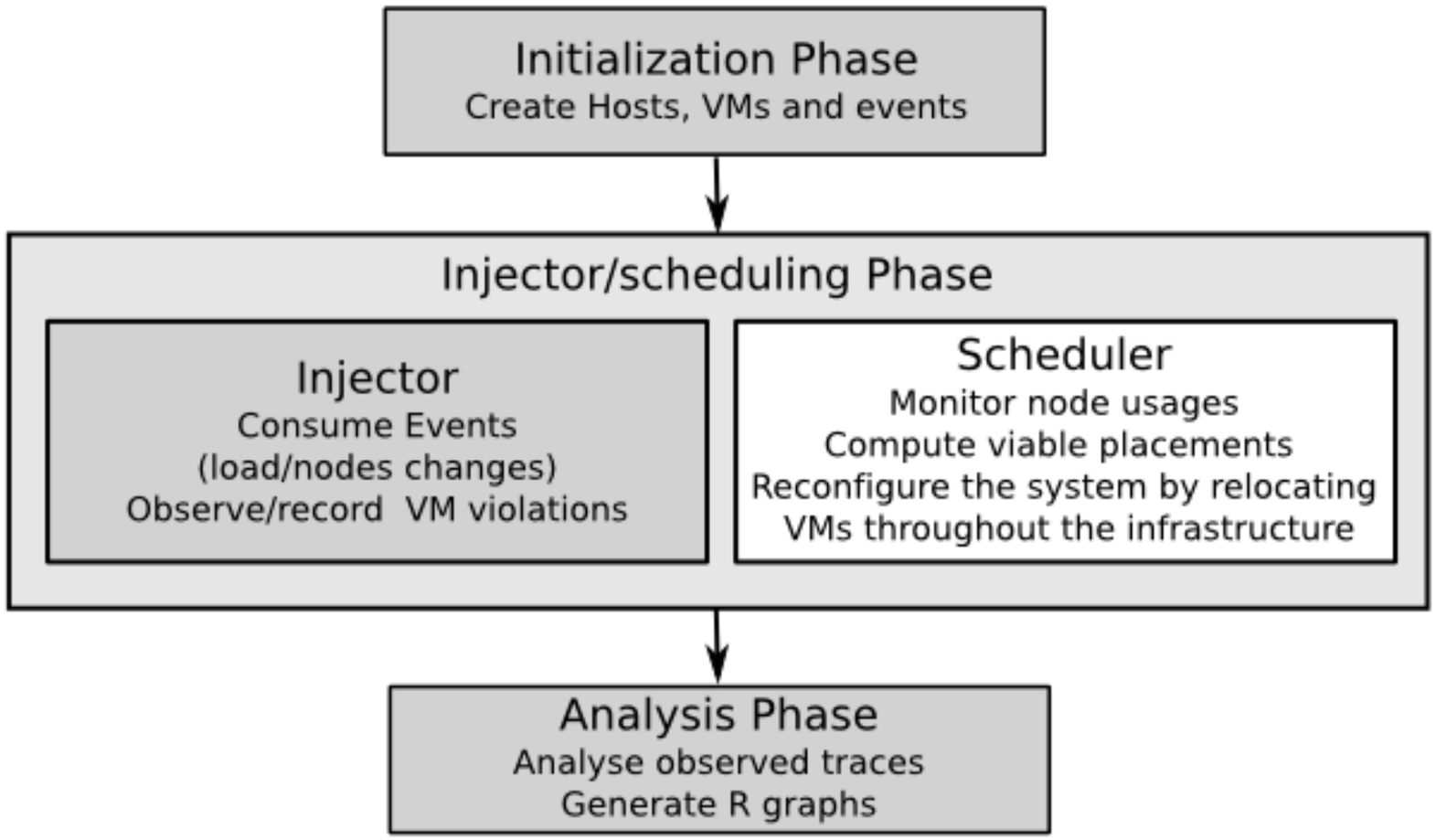
How can we move forwards faster and mitigate reinventing the wheel each time?

VM PLACEment Simulator

- A dedicated simulator to
  - Evaluate/compare VM placement policies at large-scale
  - Relieve researches of the burden of dealing with VM creations and workload fluctuations
- Leveraging latest SimGrid VM extensions
  - Enabling researchers to control VMs in the same manner as in the real world (*e.g.*, create/destroy, start/shutdown, suspend/resume and migrate)
  - Accurate live migration model (implementing the pre-copy strategy) The time and the resulting traffic of a migration is computed by taking into account competition arising in the presence of resource sharing and the memory refresh rate [CloudCom2013].

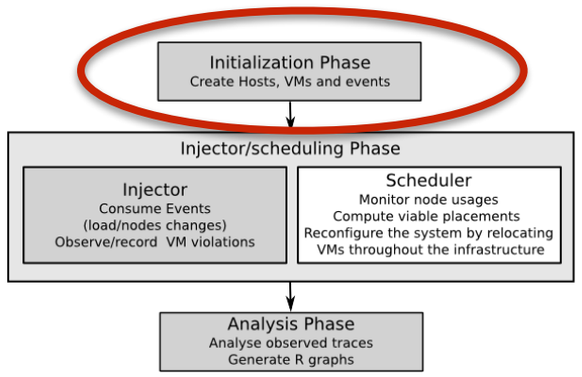
VMPlaceS - Overview

- A three steps simulation



- Researchers develop their scheduling algorithm in JAVA (or SCALA) using the SimGrid MSG API and a more abstract interface (XHost, XVM and SimulatorManager)

VMPlaceS - Initialization Phase



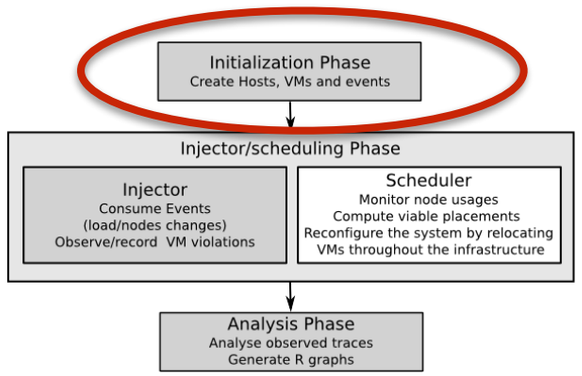
- Simulated infrastructure
  - Defined as input parameters : p hosts, s host(s), n VMs
  - Hosts are organized in forms of topologies (clusters being the most common ones but more complex structures can be defined to investigate federated scenarios for instance)

- VMs
  - Defined as input parameters through classes: **`nb_cpus:ramsize:net_bw:mig_speed:mem_speed`**
  - Assigned in a round robin fashion to the p hosts
  - VMs starts with a CPU consumption of 0 evolving during the simulation according to the events injection

VM Classes definitions

small-1:1:1024:125:125:80
medium-1:2:4096:125:125:40
medium-2:2:4096:125:125:60
medium-3:2:4096:125:125:80
large:4:8192:125:125:60
xlarge:8:16384:125:125:50

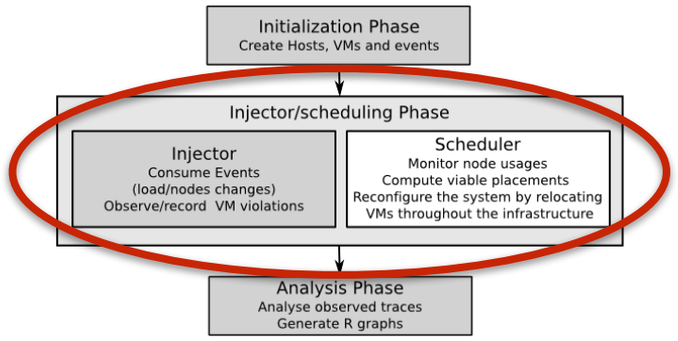
VMPlaceS - Events Generation



- Generation of the events queue at the end of the initialization phase
  - CPU load change events
    - Generated in order to change the load of each VM every `t` seconds on average (`t` being a random variable following an exponential distribution with rate λt)
    - The CPU load evolves according to a Gaussian distribution (defined by μ and σ)
  - Node apparition/removal events have been added since the submission (other events can be easily added for instance to stress other dimensions).
- Once the global event queue is ready and the VMs have been started, the evaluation of the scheduling mechanism can start for the given duration.

VMPlaceS - Injector/Scheduling Phase

- Injector



  - It consumes the events

  - Changes on the cpu load lead to SLA violations but also modify the time to migrate VMs from one PM to another one.

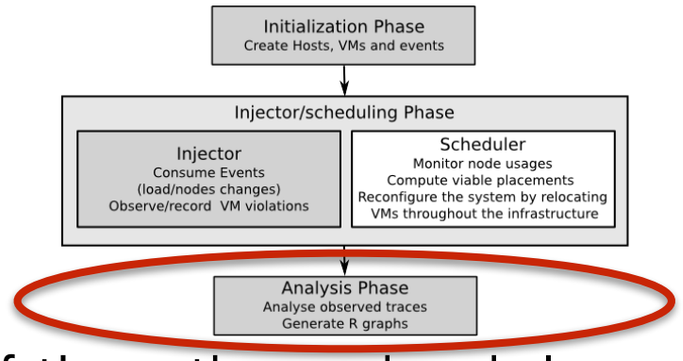
- Scheduler

  - Implemented by researchers/end-users

  - The computation phase of the VMPP is really performed (i.e., VMPlaces really invokes the scheduler)

  - Apply reconfiguration plans by invoking resume/suspend/ migrate operations of SimGrid.

VMPlaceS - Trace Analysis

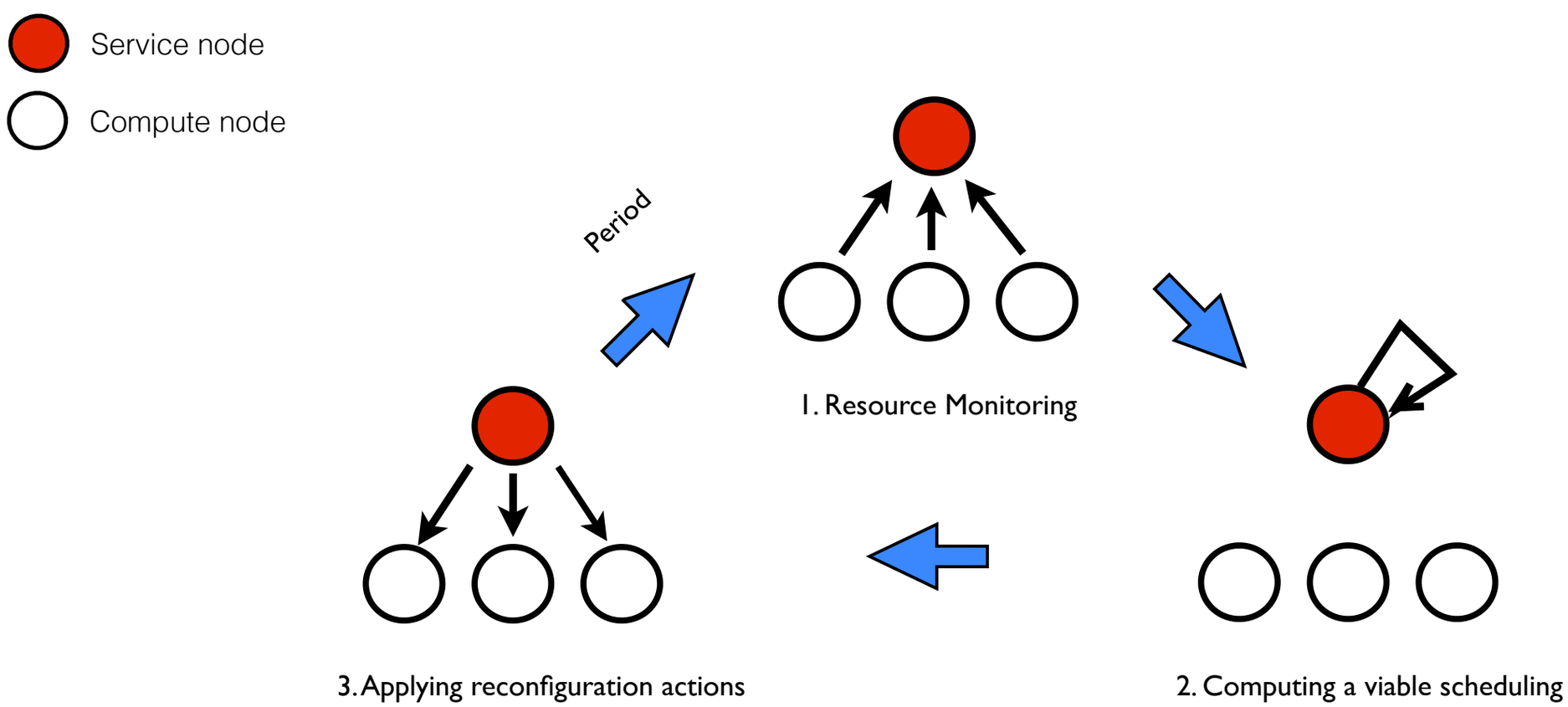


- VMPlaces records several metrics
  - load of VMs/hosts, number of migrations, number of time the scheduler has been invoked, number of times it succeeds or fails to resolve non viable violations, …

- Using an extending version of SimGrid's TRACE module
- Delivering a JSON trace file which is then consumed by R to deliver tables/graphs
  - Duration of violations per VMs, PMs and overall (**μ** and **σ**), Duration of scheduling computation/reconfiguration phases (**μ** and **σ**), …

VMPlaceS Accuracy

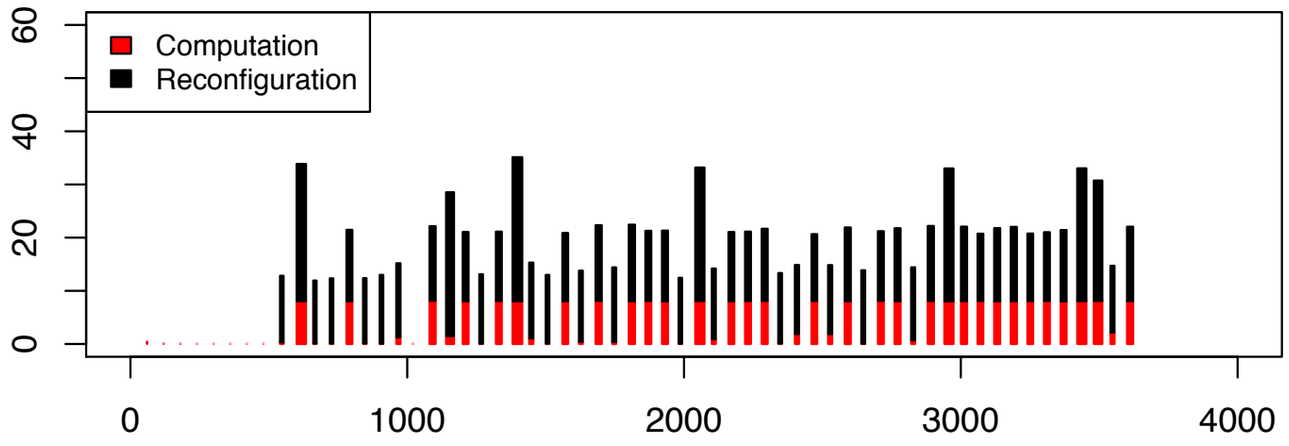
- Implementation of a dedicated version of VMPlaceS on Grid'5000
- Implementation of the Entropy proposal [VEE'09] in both systems
  - Scheduling algorithm based on the Choco solver (Constraint Programming)



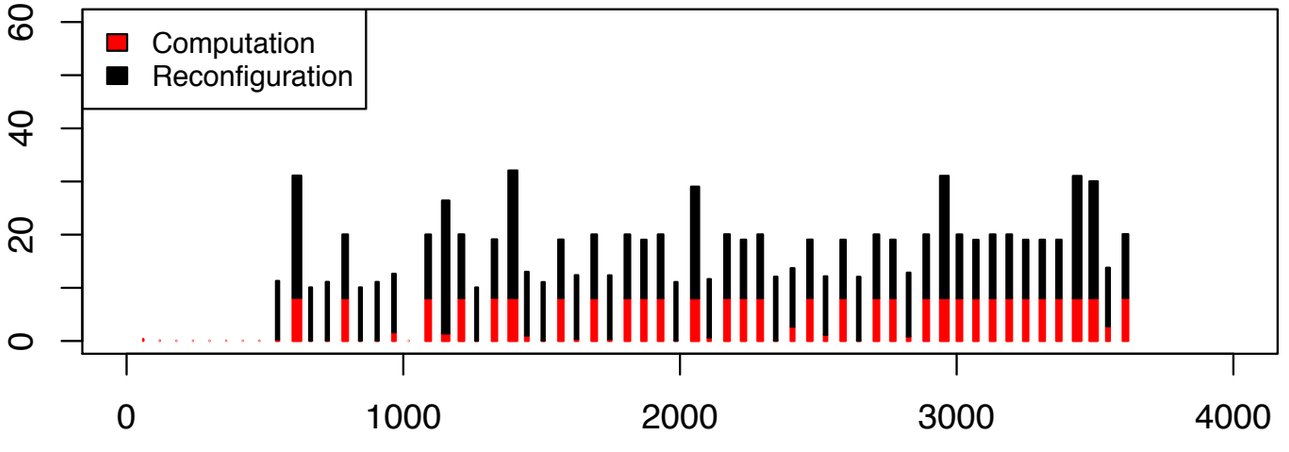
VMPlaceS Accuracy

- Comparison of the executions in both worlds (in vivo / simulated)
  - 32PMs, 4 cores/16GB/1Gbps per PM
  - VMs based on classes: 1:1GB:1Gbps:1Gbps:x with x varies between 0 and 80%
  - 192 VMs (6 per node) . $\lambda t=\#VMs/300$ $\mu=60$ and $\sigma=20$
  - The scheduling algorithm has been invoked every 60 seconds over a 3600 seconds execution
  - a dedicated tool to inject the load in each VM running on top of G5K according to the events consumed by the injector

Grid'5000



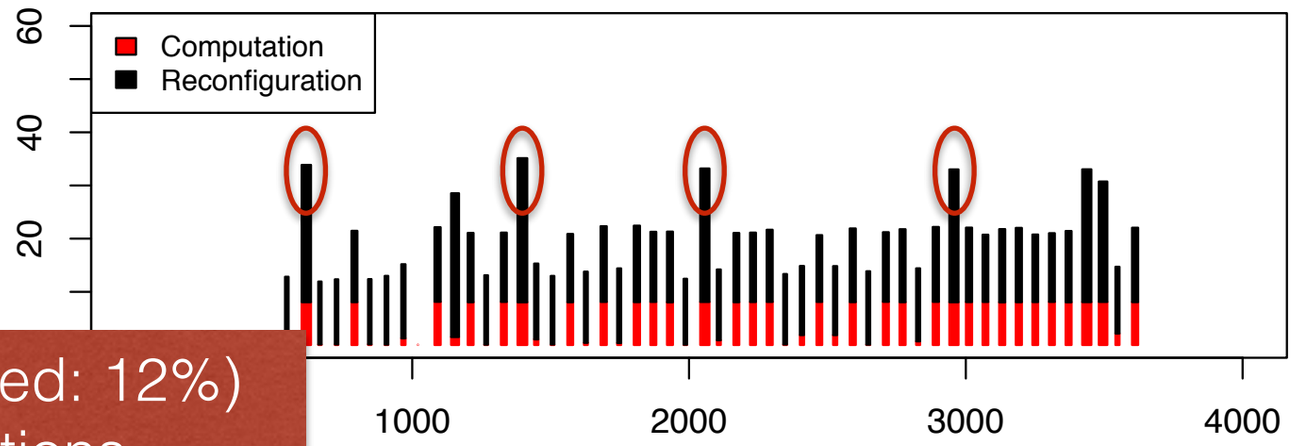
VMPlaceS



VMPlaceS Accuracy

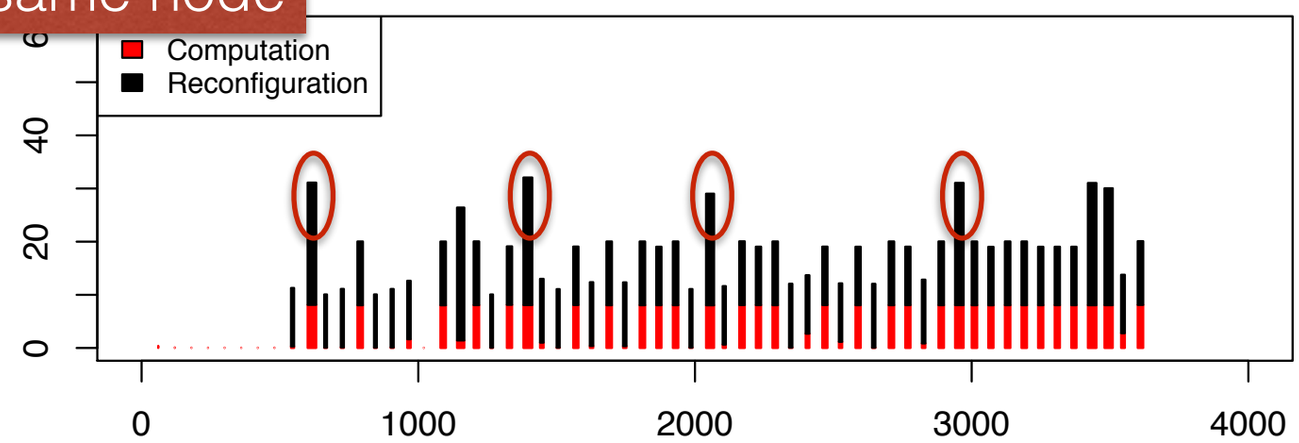
- Comparison of the executions in both worlds (in vivo / simulated)
  - 32PMs, 4 cores/16GB/1Gbps per PM
  - VMs based on classes: 1:1GB:1Gbps:1Gbps:x with x varies between 0 and 80%
  - 192 VMs (6 per node) . $\lambda t=\#VMs/300$ $\mu=60$ and $\sigma=20$
  - 3600 seconds execution
  - a dedicated tool to inject the load in each VM running on top of G5K according to the events consumed by the injector

Grid'5000



Difference between 6% and 18% (med: 12%)
Worst case when multiple migrations
are performed simultaneously to the same node

VMPlaceS

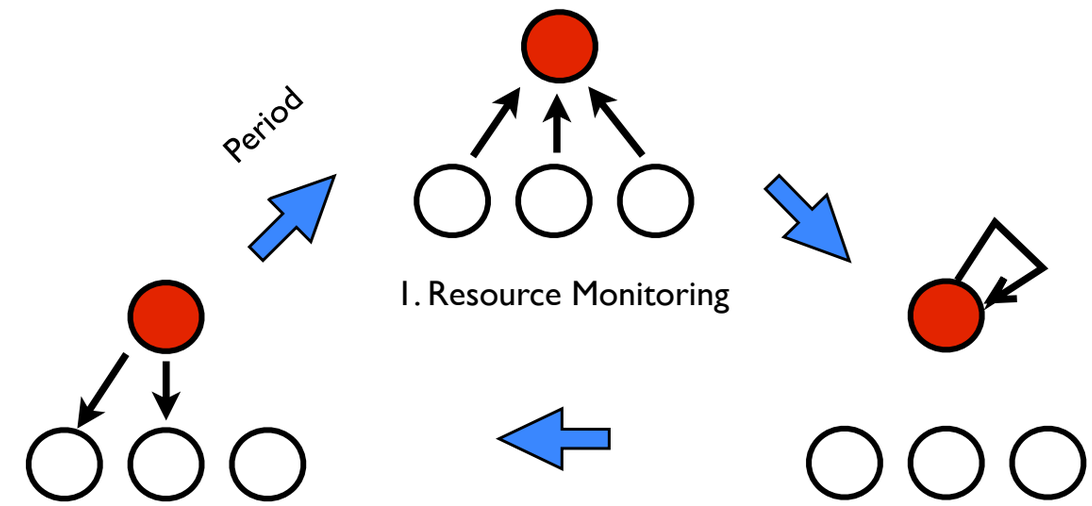


A First Use-Case

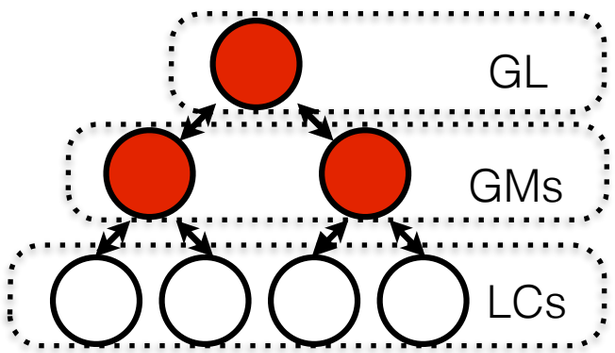
- To illustrate how different strategies can be evaluated/compared
- Three strategies aiming at mitigating CPU violations
  - Centralized (the Entropy proposal, [VEE'09])
  - Hierarchical (The Snooze proposal, [CCGRID'12])
  - Distributed (The DVMS proposal, [CCPE'12])
- For the sake of simplicity, the same scheduling algorithm has been used in each policy [VEE'09]

Implemented Strategies - Overview

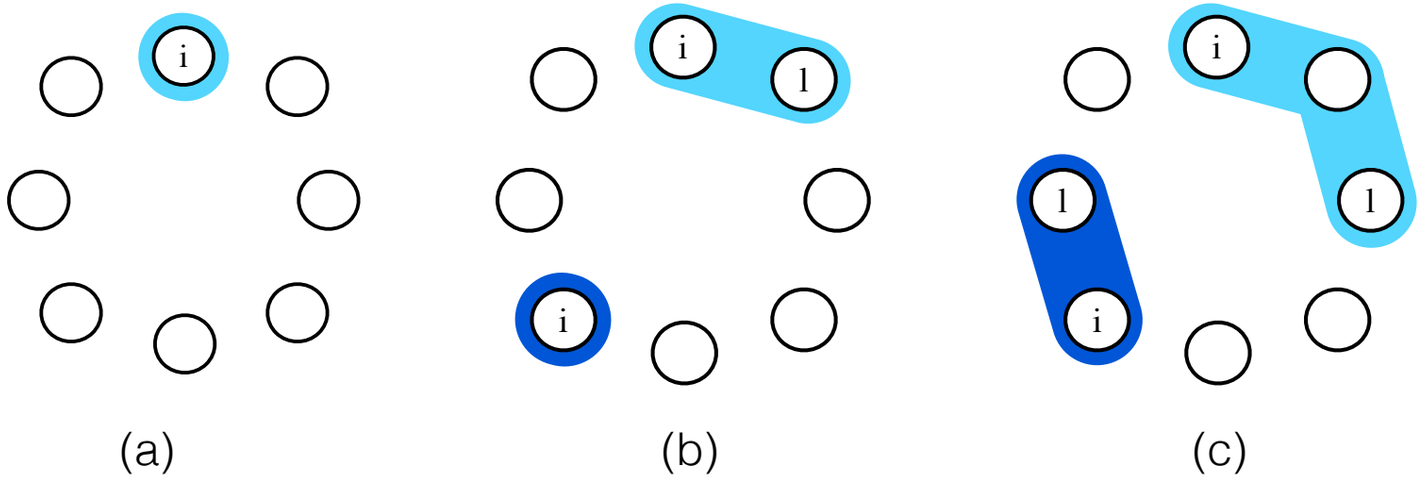
- Centralized

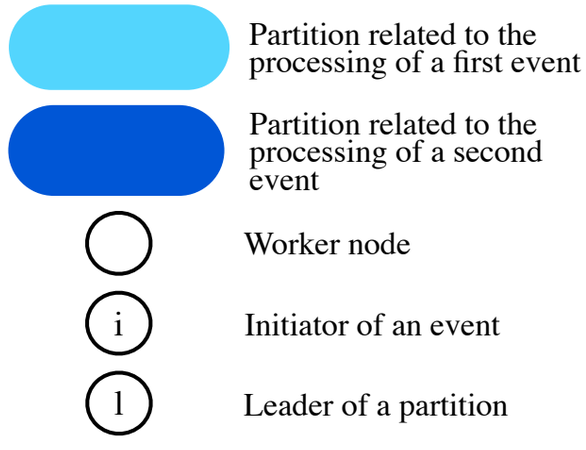


- Hierarchical



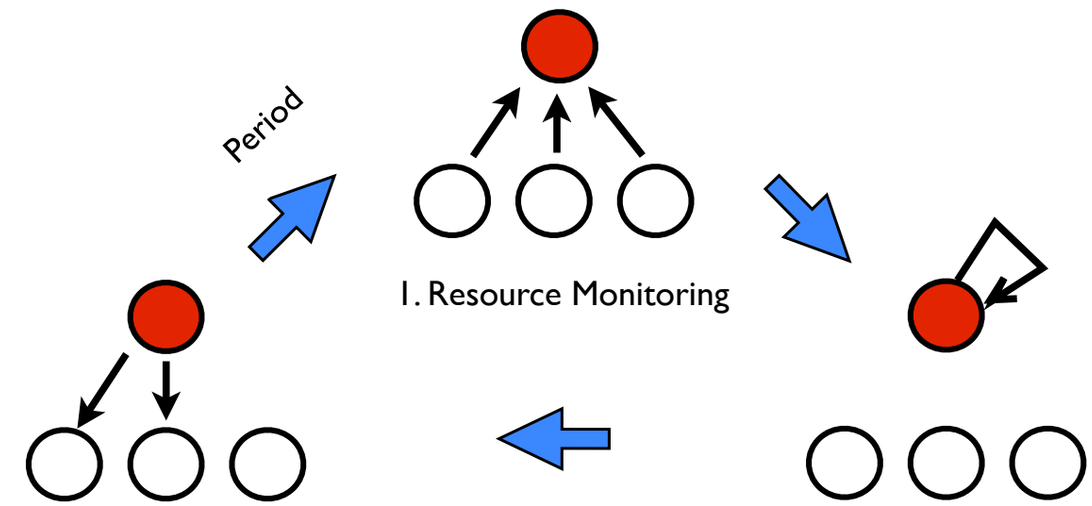
- Distributed



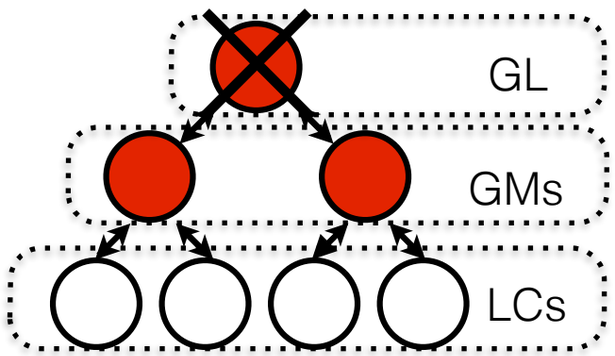


Implemented Strategies - Overview

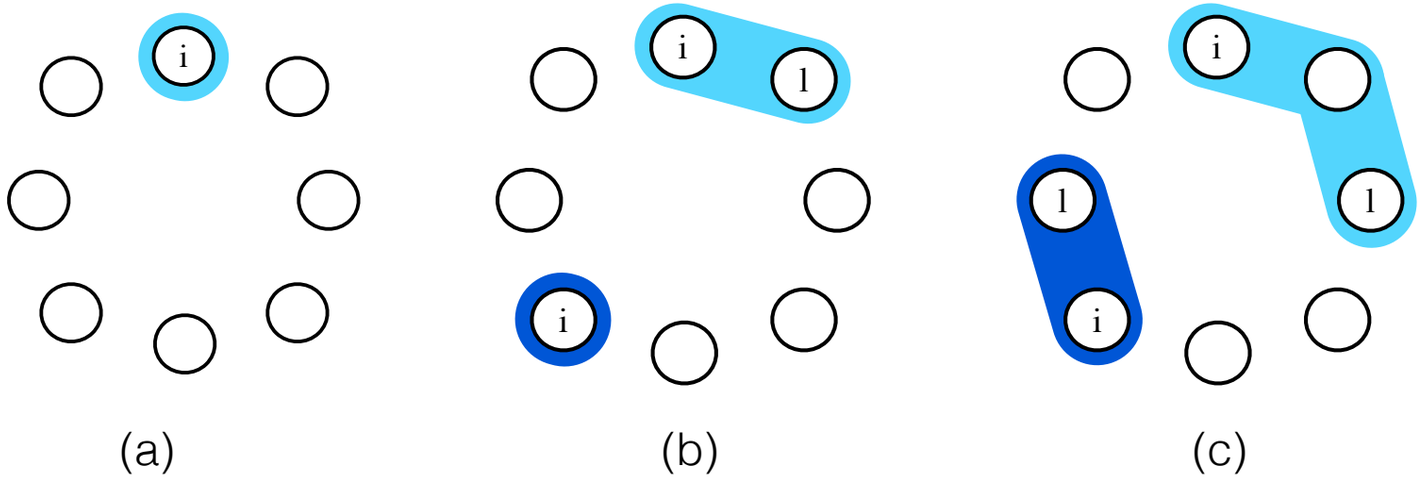
- Centralized

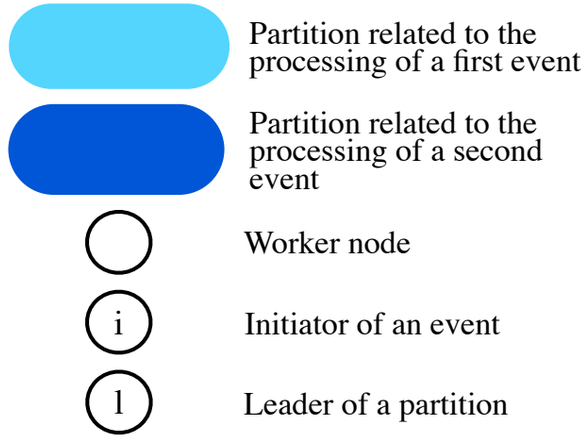


- Hierarchical



- Distributed





Entropy/Snooze/DVMS Analysis

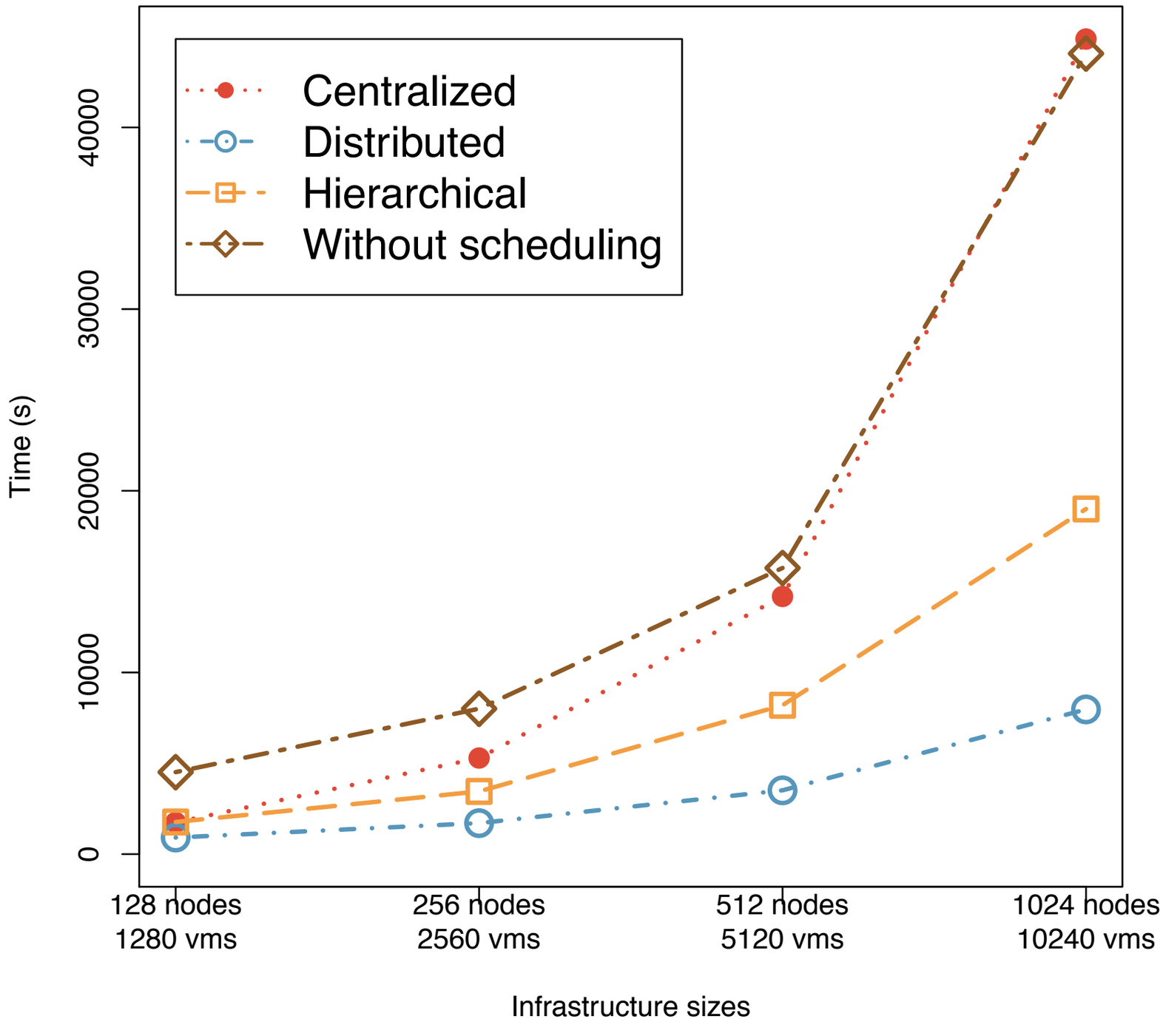
- Simulation Input parameters

  - PMs: 8 cores, 32GB, 1Gbps, 7 cores are considered.
    VMs: 1:1GB:1Gbps:1Gbps:x with x varies between 0 and 80%
    10 VMs per PM
    Cluster infrastructure composed of 128, 256, 512, 1024 PMs

    λt=#VMs/300 μ=60 and σ=20
    Duration: 1800 seconds
    (stationary state reached after 20 min, global cluster load 85%)

  - For Entropy and Snooze additional simulated PMs have been provided (one GM per 32 PMs)
    period 30 seconds.

Entropy/Snooze/DVMS Analysis



Cumulated violation time

Entropy/Snooze/DVMS Analysis

Infrastructure size	Duration of **violations** ($\mu \pm \sigma$)		
	Centralized	Hierarchica	
128 nodes	21.26 ± 13.55	21.07 ± 12.32	9.55 ± 2.57
256 nodes	40.09 ± 24.15	21.45 ± 12.10	9.58 ± 2.5
512 nodes	55.63 ± 42.26	24.54 ± 16.95	9.57
1024 nodes	81.57 ± 86.59	29.01 ± 38.14	

DVMS outperforms the others !?

Infrastructure size	Duratio		Distributed
128 n		± 4.63	0.29 ± 0.03
		2.65 ± 4.69	0.25 ± 0.02
	29.14	2.83 ± 4.98	0.21 ± 0.01
	.41 ± 50.35	2.69 ± 4.92	0.14 ± 0.01

1./ Can we find a good partitioning size for Snooze ?

2./ What would be the benefit for Snooze of a reactive approach?

Infrastructure size	Duration of **reconfigurations** ($\mu \pm \sigma$)		
	Centralized	Hierarchical	Distributed
128 nodes	10.34 ± 1.70	10.02 ± 0.14	10.01 ± 0.11
256 nodes	10.26 ± 1.45	10.11 ± 0.83	10.01 ± 0.08
512 nodes	11.11 ± 3.23	10.28 ± 1.50	10.08 ± 0.82
1024 nodes	18.90 ± 7.57	10.30 ± 1.60	10.04 ± 0.63

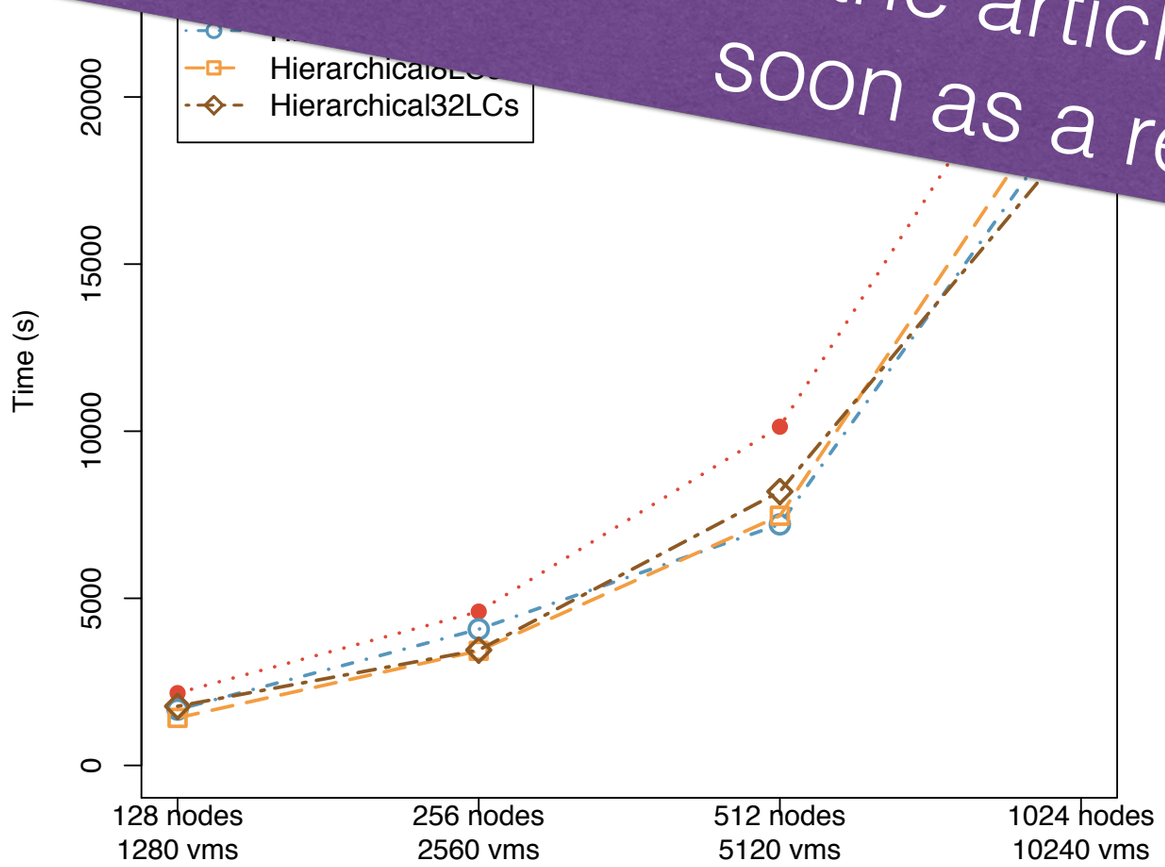
While the centralized approach does not scale, both phases are constant from the time viewpoint for the two other approaches

Investigate Variants

- Evaluate the impact of having smaller partitions in Snooze

but partitions grow from 2 LCs to 32 LCs per GM

Other variants and possible improvements (not addressed in the article but available on demand and soon as a research report)



Cumulated violation time

512	
1024	173

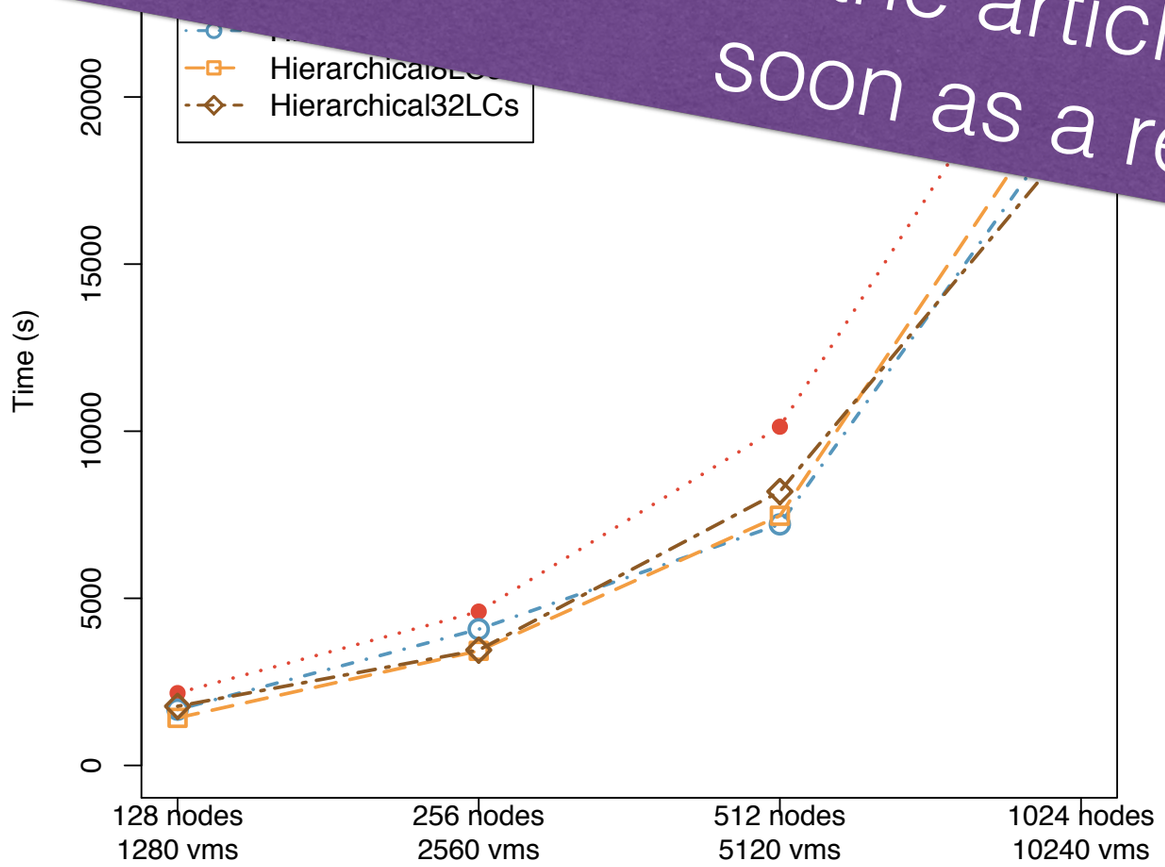
Infra. Size	Duration of the **computations** ($\mu \pm \sigma$)			
	2 LCs	4 LCs	8 LCs	32 LCs
128	0.16 ± 1.23	0.34 ± 1.81	0.58 ± 2.40	2.53 ± 4.62
256	0.18 ± 1.31	0.42 ± 1.99	0.66 ± 2.50	2.65 ± 4.69
512	0.15 ± 1.20	0.33 ± 1.78	0.67 ± 2.54	2.83 ± 4.98
1024	0.19 ± 1.37	0.42 ± 2.02	0.89 ± 2.90	2.69 ± 4.91

Investigate Variants

- Evaluate the impact of having smaller partitions in Snooze

but partitions grow from 2 LCs to 32 LCs per GM

Other variants and possible improvements (not addressed in the article but available on demand and soon as a research report)



Cumulated violation time

512	
1024	173

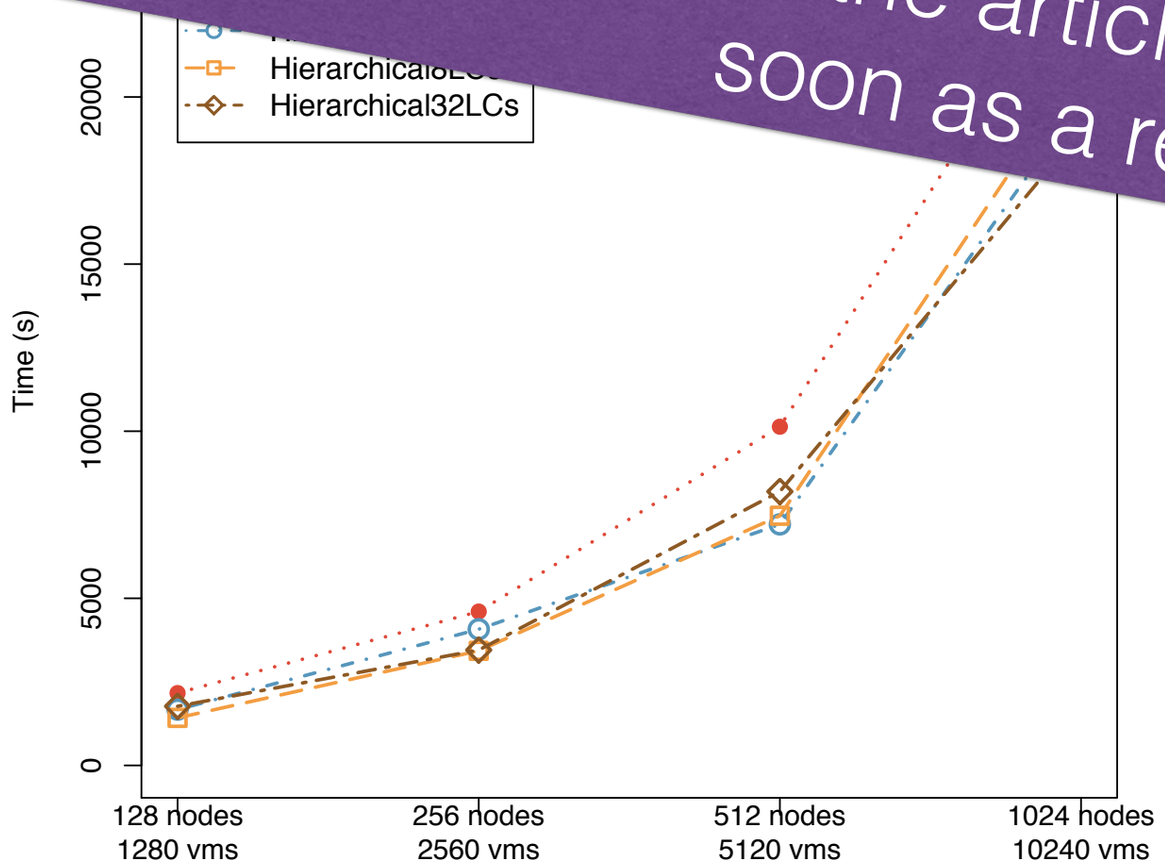
Infra. Size	Duration of the **computations** ($\mu \pm \sigma$)			
	2 LCs	4 LCs	8 LCs	32 LCs
128	0.16 ± 1.23	0.34 ± 1.81	0.58 ± 2.40	2.53 ± 4.62
256	0.18 ± 1.31	0.42 ± 1.99	0.66 ± 2.50	2.65 ± 4.69
512	0.15 ± 1.20	0.33 ± 1.78	0.67 ± 2.54	2.83 ± 4.98
1024	0.19 ± 1.37	0.42 ± 2.02	0.89 ± 2.90	2.69 ± 4.91

Investigate Variants

- Evaluate the impact of having smaller partitions in Snooze

but partitions grow from 2 LCs to 32 LCs per GM

Other variants and possible improvements (not addressed in the article but available on demand and soon as a research report)



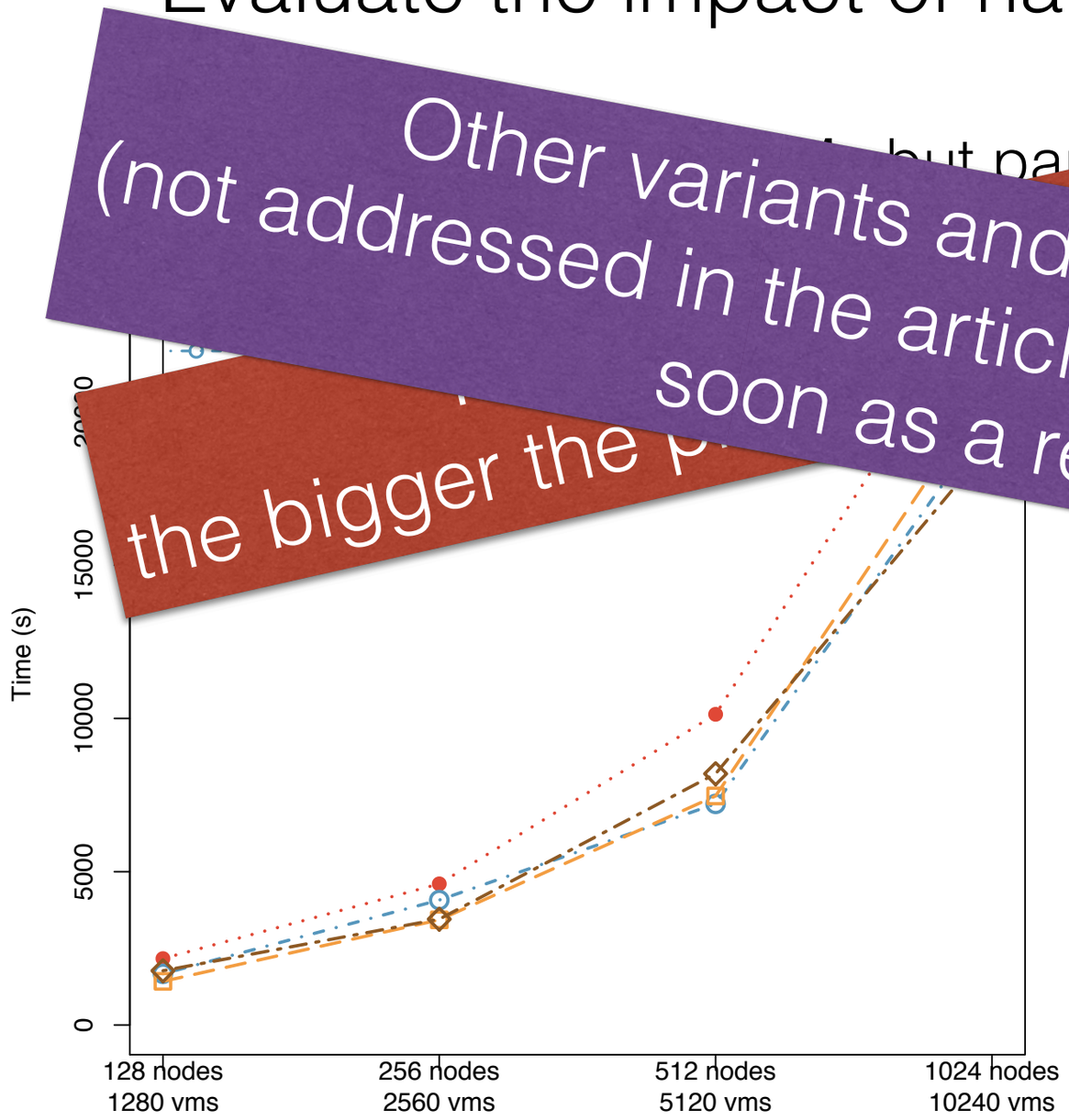
Cumulated violation time

512	
1024	173

Infra. Size	Duration of the computations ($\mu \pm \sigma$)			
	2 LCs	4 LCs	8 LCs	32 LCs
128	0.16 ± 1.23	0.34 ± 1.81	0.58 ± 2.40	2.53 ± 4.62
256	0.18 ± 1.31	0.42 ± 1.99	0.66 ± 2.50	2.65 ± 4.69
512	0.15 ± 1.20	0.33 ± 1.78	0.67 ± 2.54	2.83 ± 4.98
1024	0.19 ± 1.37	0.42 ± 2.02	0.89 ± 2.90	2.69 ± 4.91

Investigate Variants

- Evaluate the impact of having smaller partitions in S ze



Cumulated violation time

512	
1024	173

Infra. Size	Duration of the computations ($\mu \pm \sigma$)			
	2 LCs	4 LCs	8 LCs	32 LCs
128	0.16 ± 1.23	0.34 ± 1.81	0.58 ± 2.40	2.53 ± 4.62
256	0.18 ± 1.31	0.42 ± 1.99	0.66 ± 2.50	2.65 ± 4.69
512	0.15 ± 1.20	0.33 ± 1.78	0.67 ± 2.54	2.83 ± 4.98
1024	0.19 ± 1.37	0.42 ± 2.02	0.89 ± 2.90	2.69 ± 4.91

Other variants and possible improvements (not addressed in the article but available on demand and soon as a research report)

ge of the partition, find a viable solution

the bigger the p

VMPlaceS - Conclusion

- Difficulties to conduct relevant evaluation of VM placement strategies (in vivo conditions, lot of metrics to monitor, scalability/reactivity, …)

- VMPlaceS, a framework providing

  - Programming support for the definition of new VM placement strategies
    Execution support for their accurate simulation at large scale
    Means to analyze the collected traces

  - Available online (with the three aforementioned strategies)
    http://beyondtheclouds.github.io/VMPlaceS/

- On-going and future work

  - New events: node apparition/removal (memory/network load changes soon)
    VM boot time, VM image migrations
    Injection of addition VMs on-the-fly

Thanks

adrien.lebre@inria.fr

http://beyondtheclouds.github.io