Large-scale Management of VMs

''de Flauncher à VM5k''

Hemera IPL - Final Review - Dec, the 17th

Adrien Lebre - Inria
Ascola Research Group

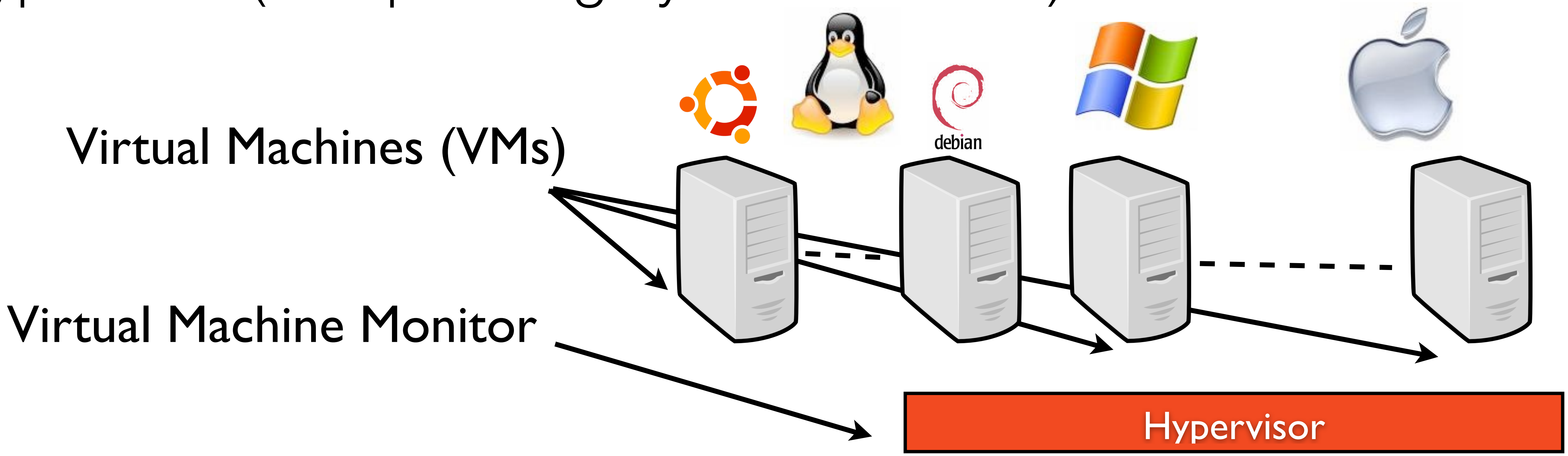
Grid'5000

Agenda

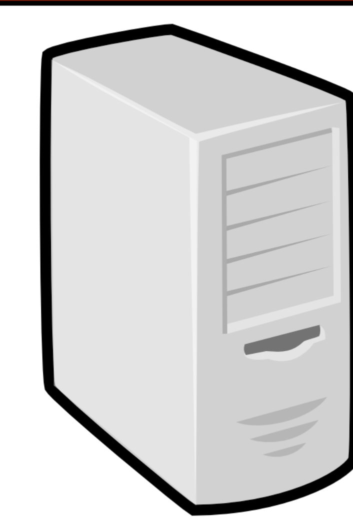
- An overview of how the way we addressed the VM placement problem throughout Hemera
- System Virtualization and VM capabilities
  - From a centralised prototype at small scale…
  - …to a large scale solution

System Virtualization

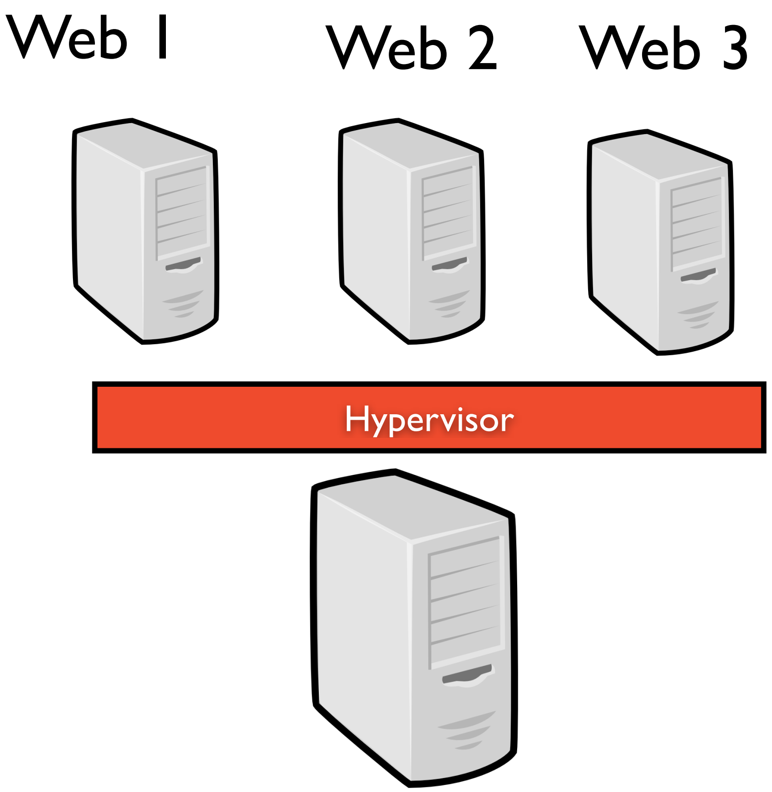
- One to multiple OSes on a physical node thanks to a hypervisor (an operating system of OSes)



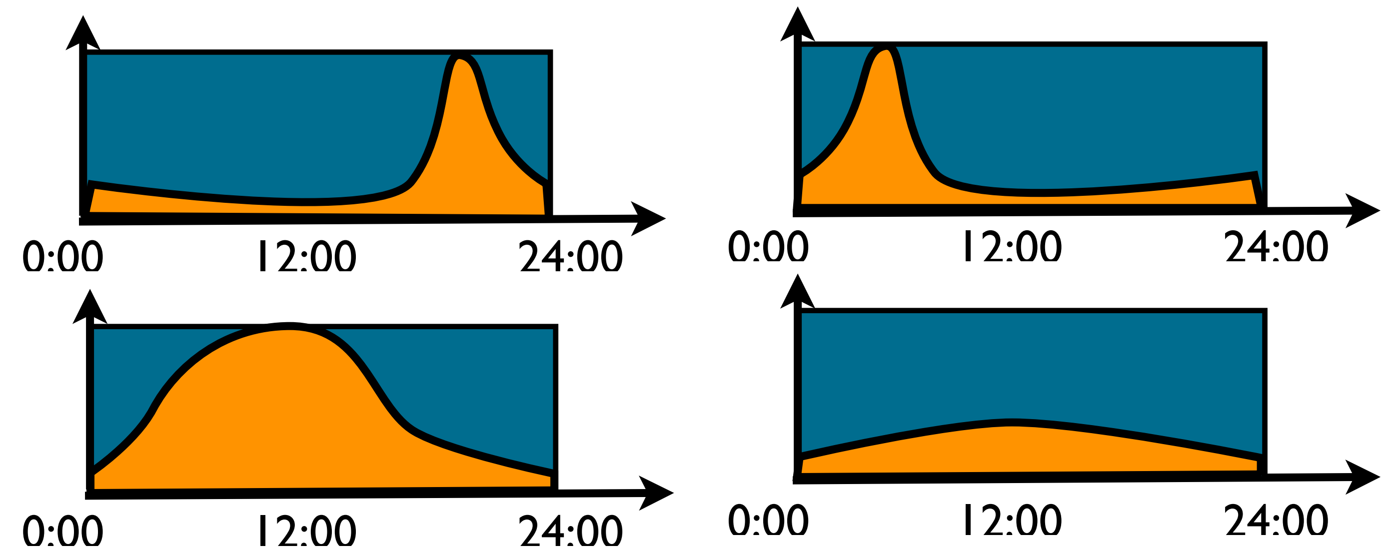
“A *virtual machine* (VM) provides a faithful implementation of a physical processor’s hardware running in a protected and isolated environment.
Virtual machines are created by a software layer called the *virtual machine monitor* (VMM) that runs as a privileged task on a physical processor.”



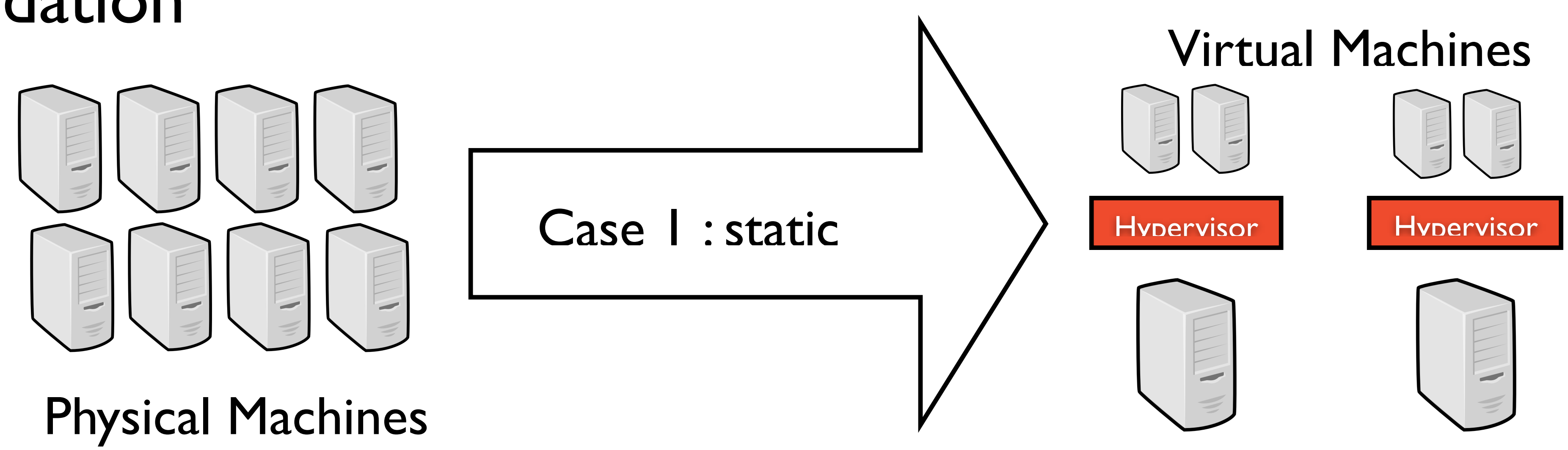
VM Capabilities



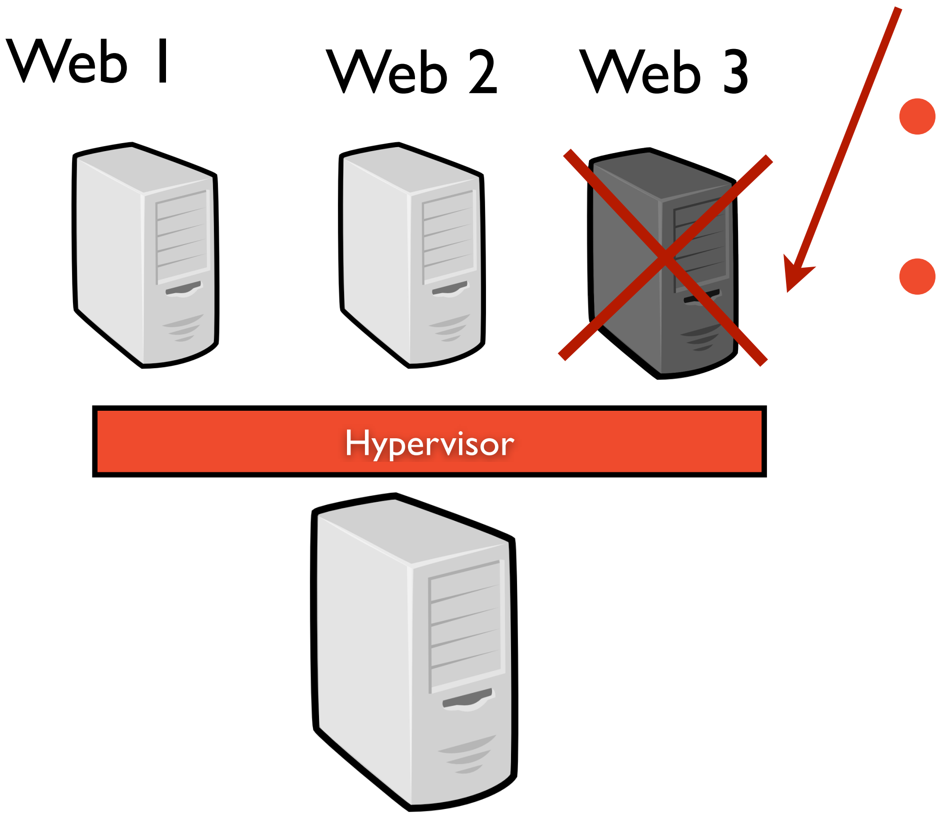
- Isolation (security between each VM)
- snapshot/suspend/resume/reboot (maintenance)



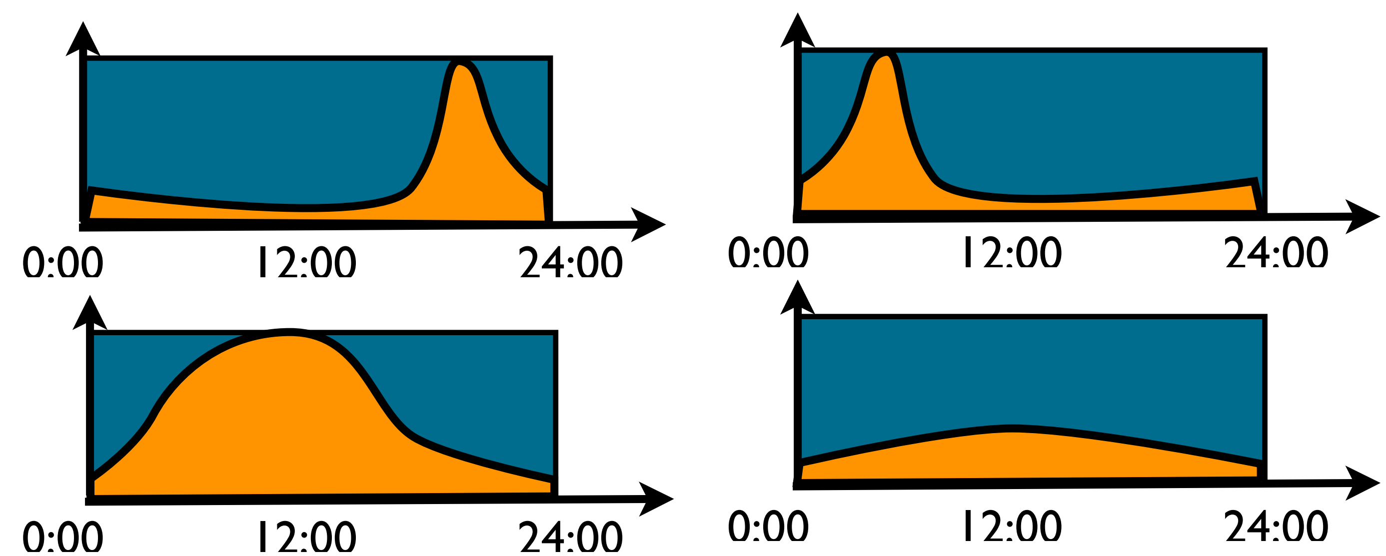
- Consolidation



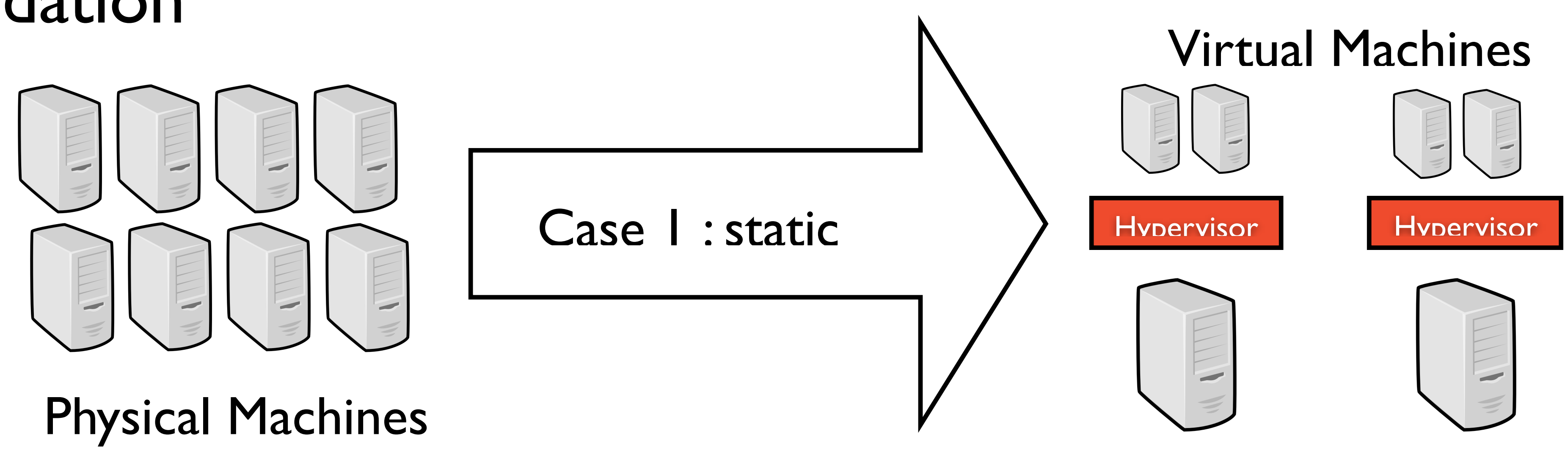
VM Capabilities



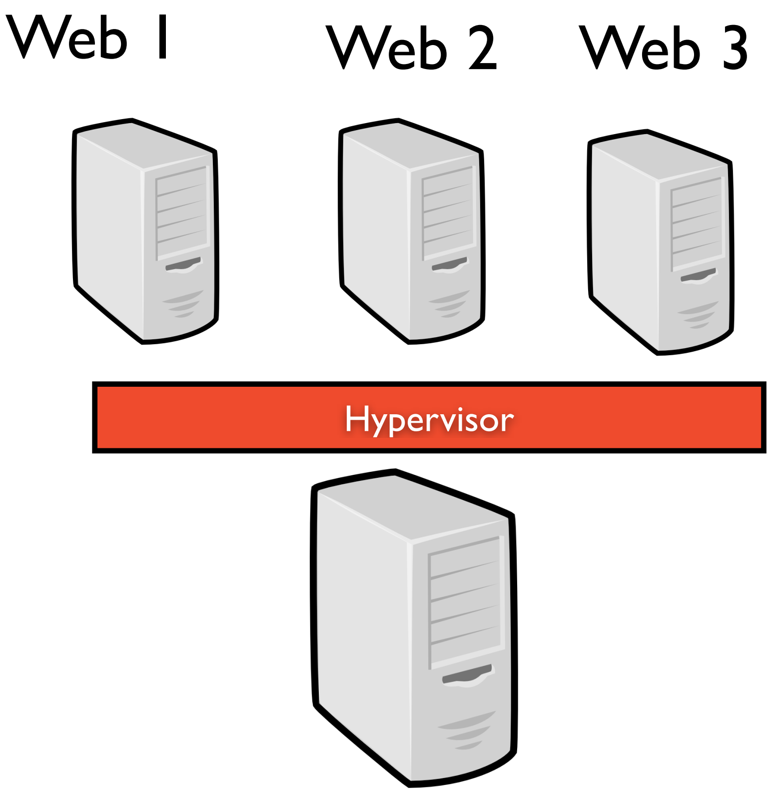
- Isolation (security between each VM)
- snapshot/suspend/resume/reboot (maintenance)



- Consolidation

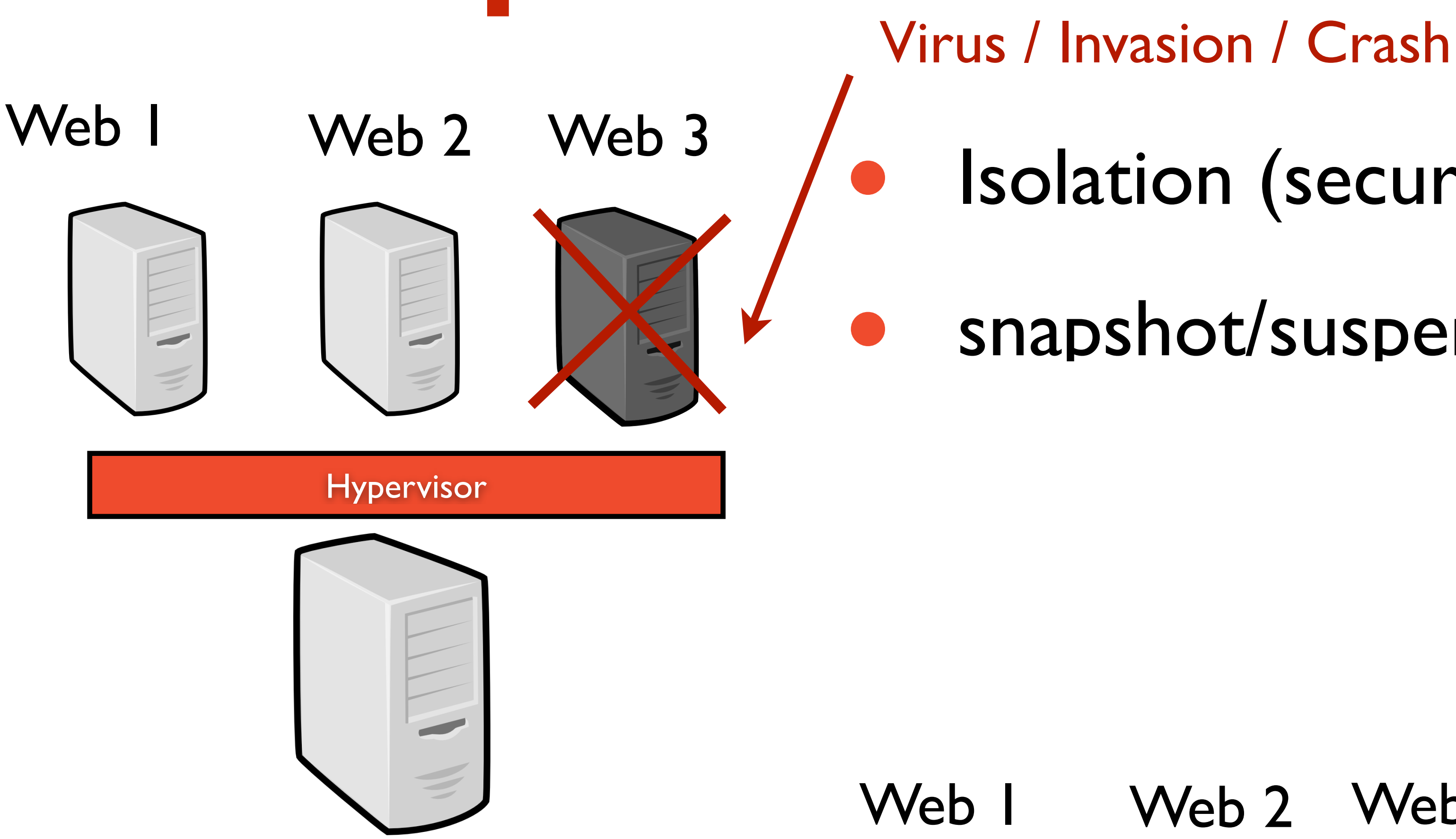


VM Capabilities



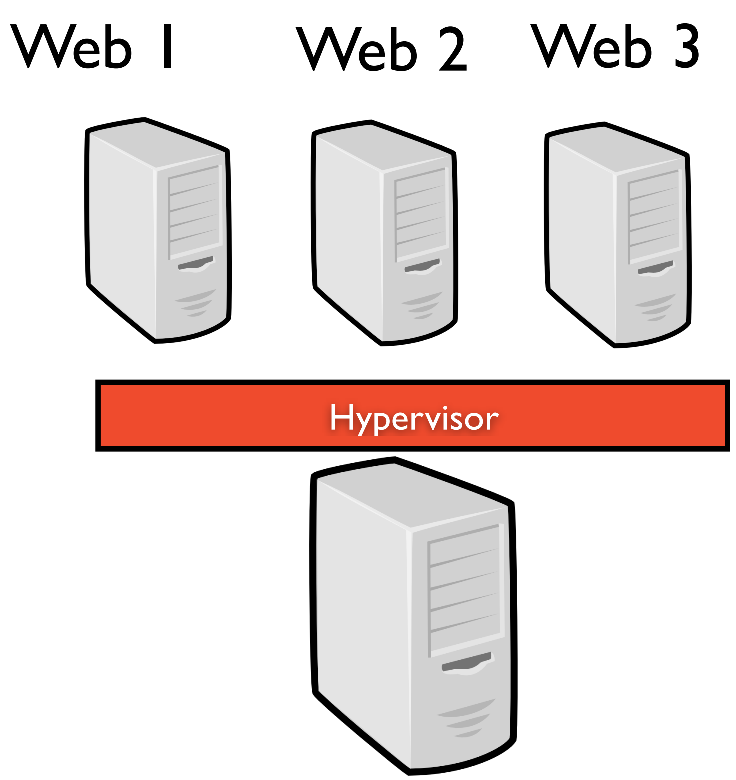
- Isolation (security between each VM)
- snapshot/suspend/resume/reboot (maintenance)

VM Capabilities

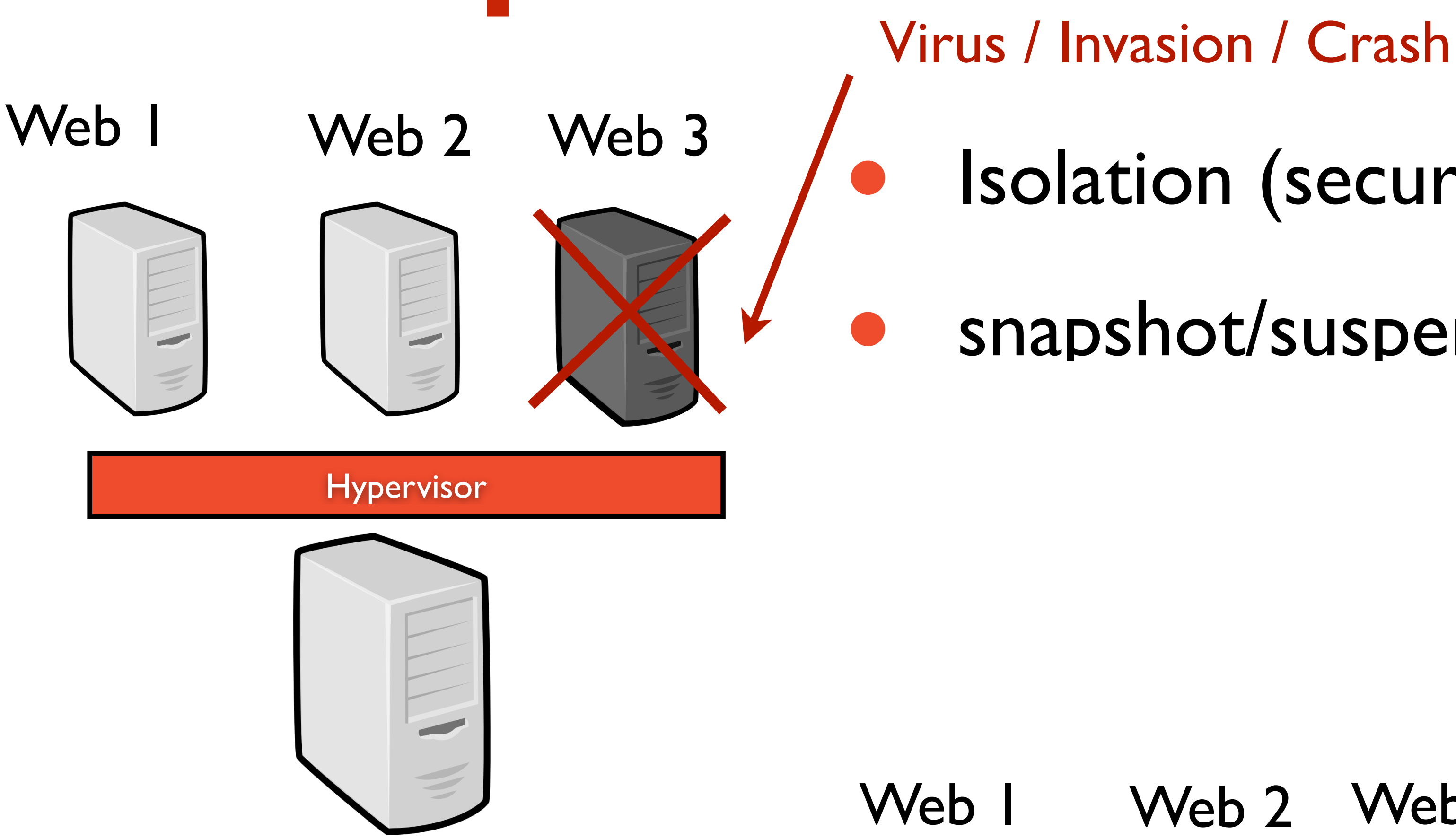


- Isolation (security between each VM)
- snapshot/suspend/resume/reboot (maintenance)

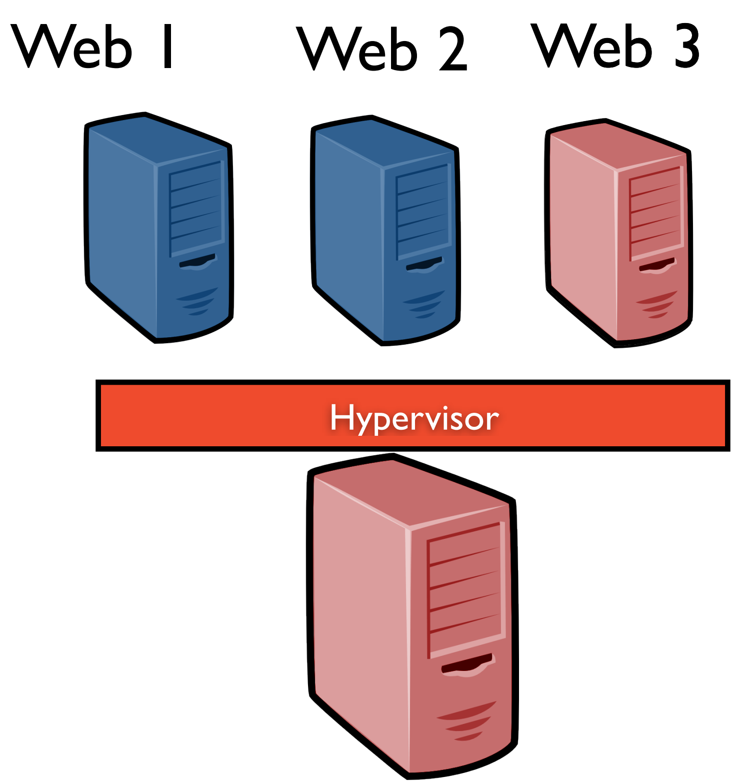
- Consolidation (load-balancing)
- Negligible downtime (~ 60 ms)



VM Capabilities

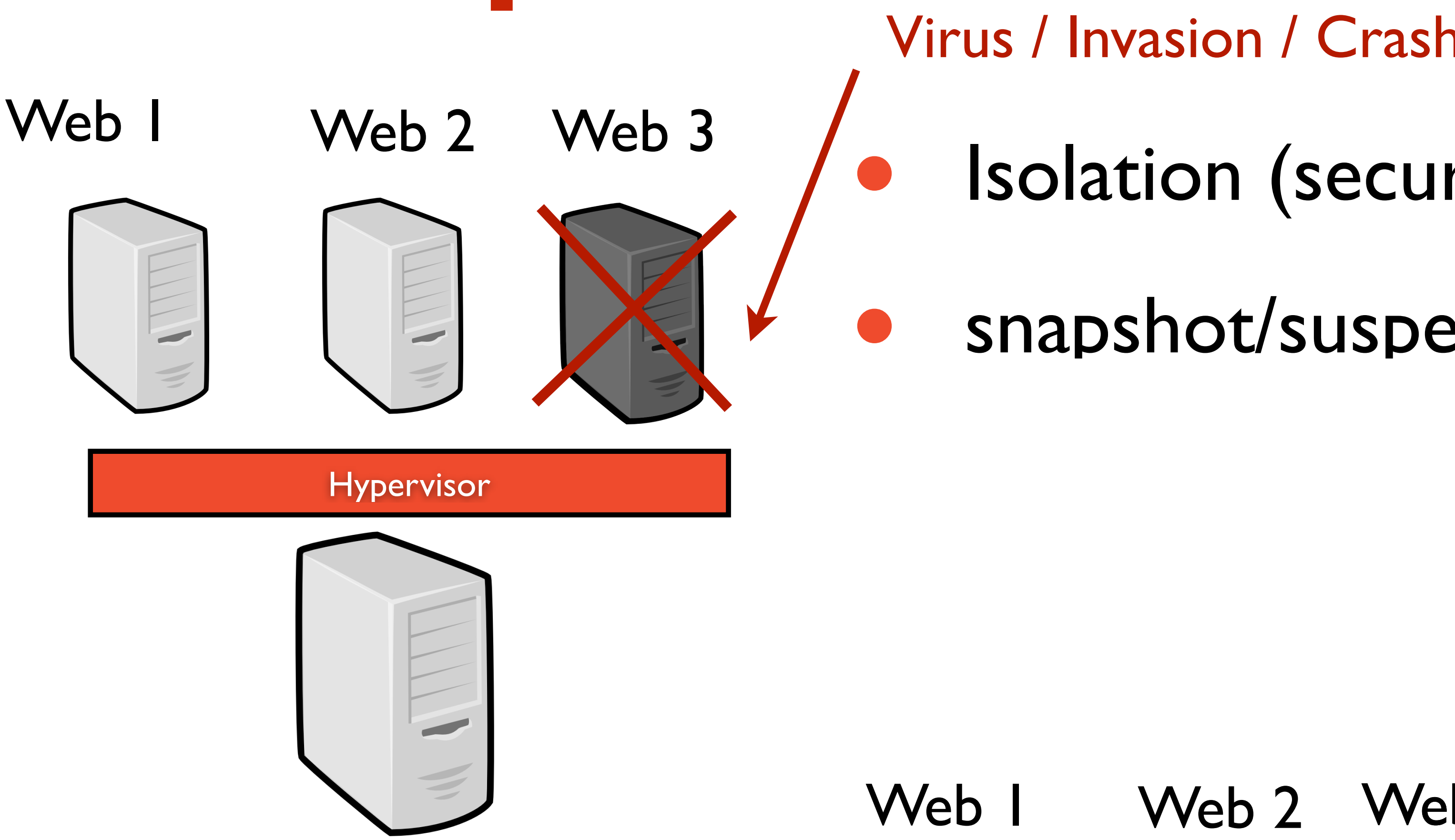


- Isolation (security between each VM)
- snapshot/suspend/resume/reboot (maintenance)



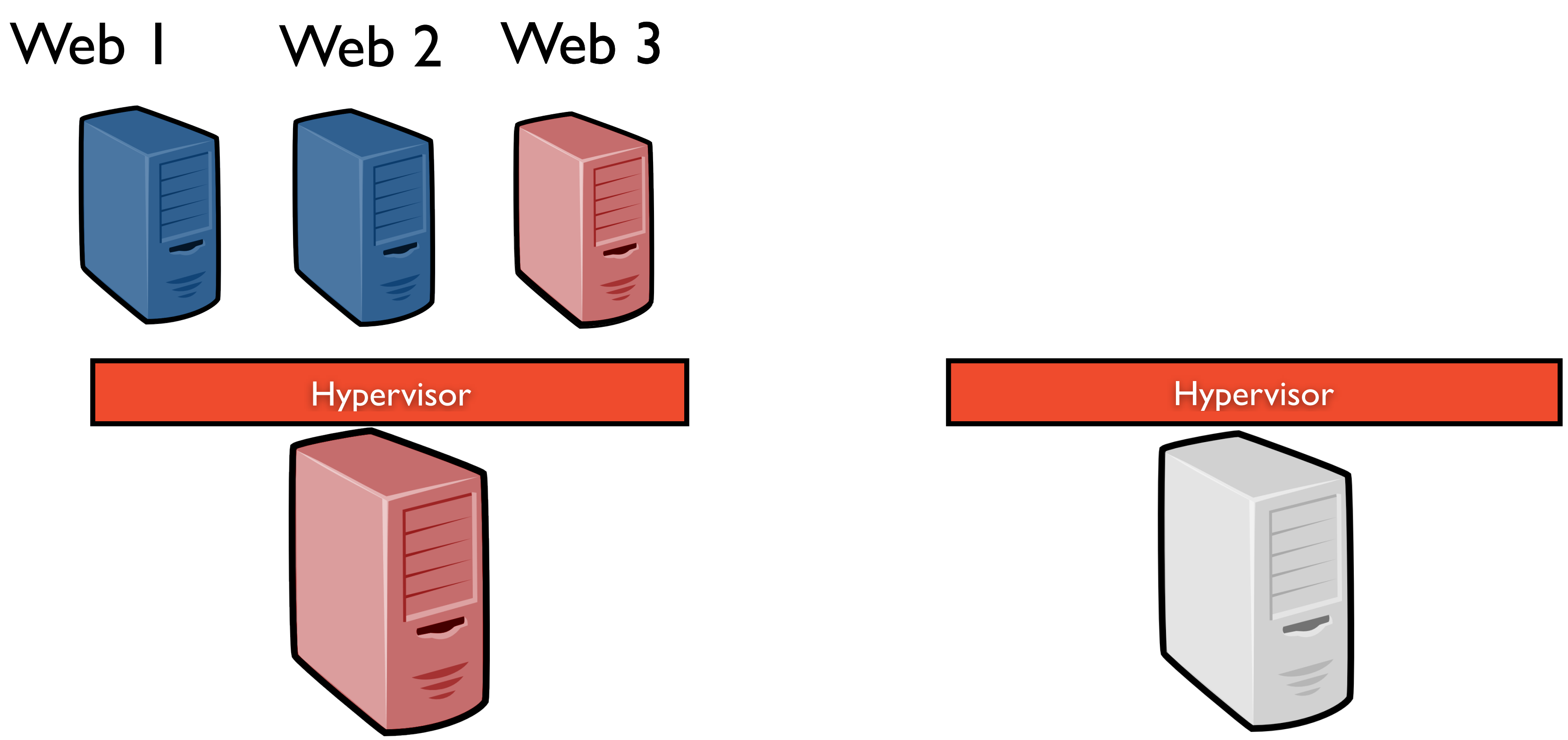
- Consolidation (load-balancing)
- Negligible downtime (~ 60 ms)

VM Capabilities



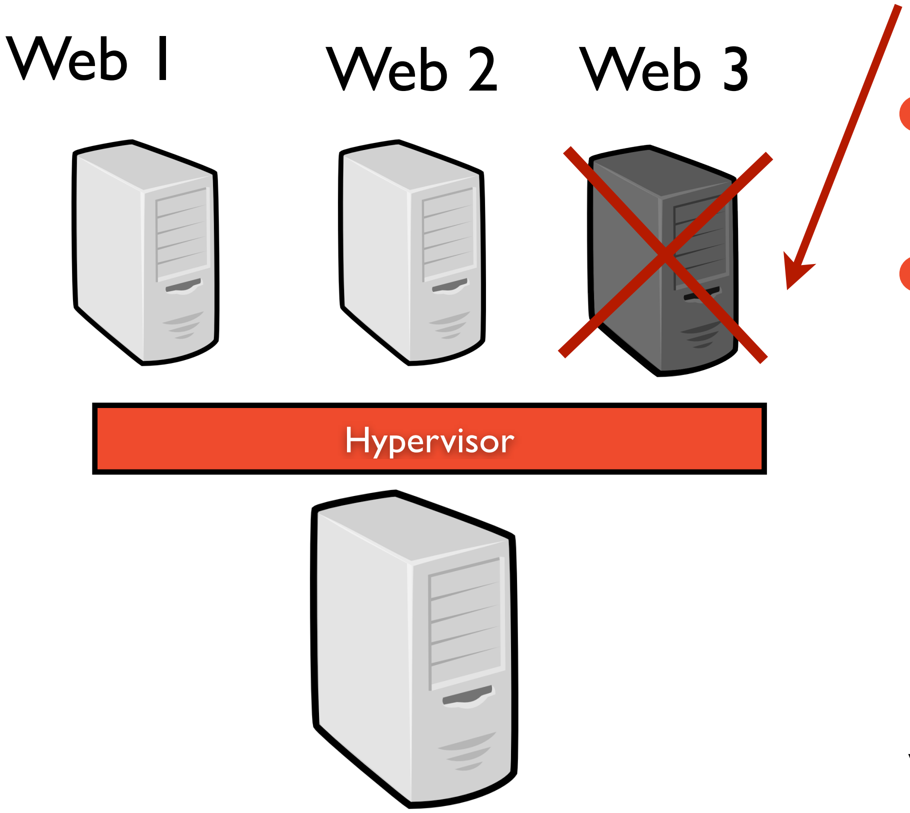
- Isolation (security between each VM)
- snapshot/suspend/resume/reboot (maintenance)

- Consolidation (load-balancing)
- Negligible downtime (~ 60 ms)

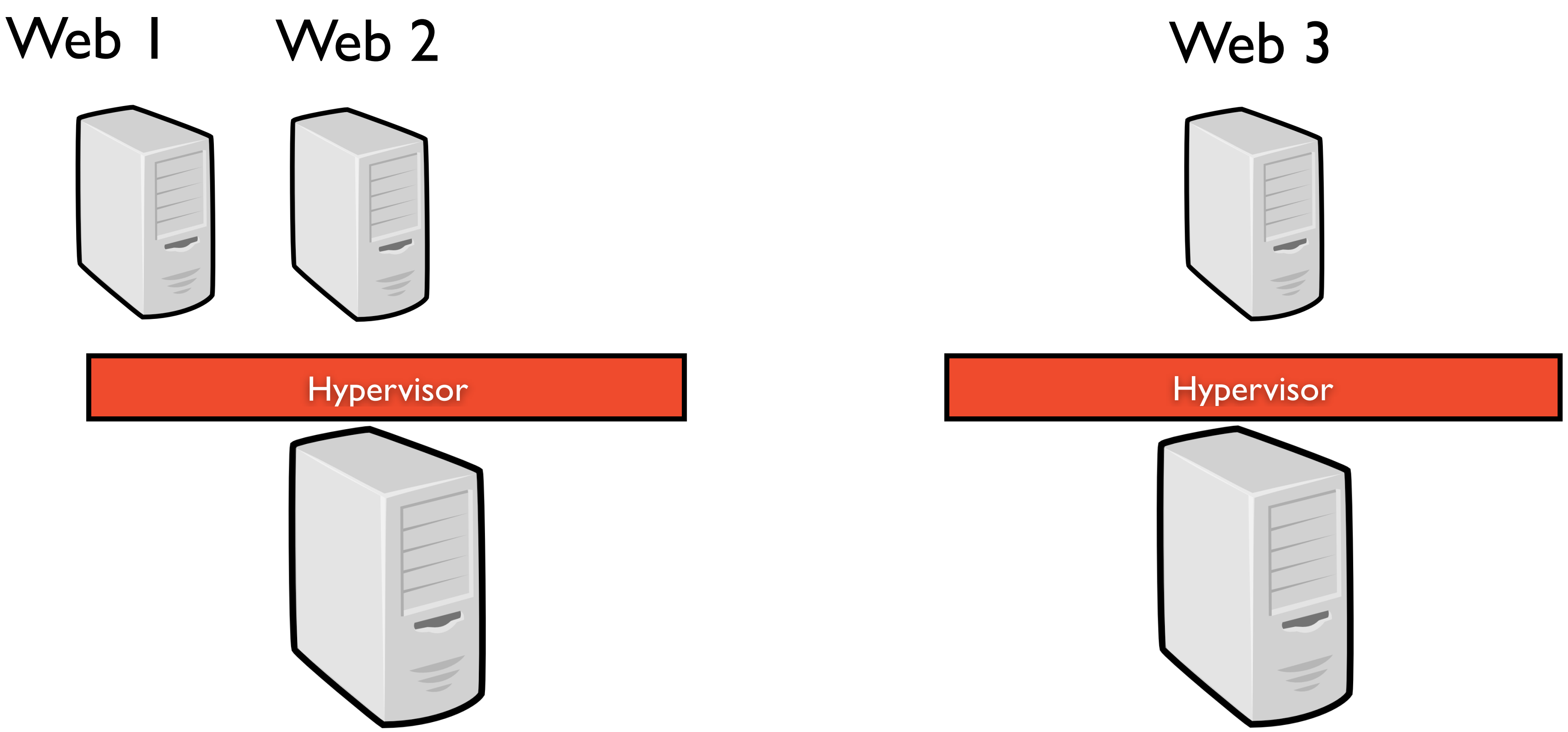


VM Capabilities

Virus / Invasion / Crash

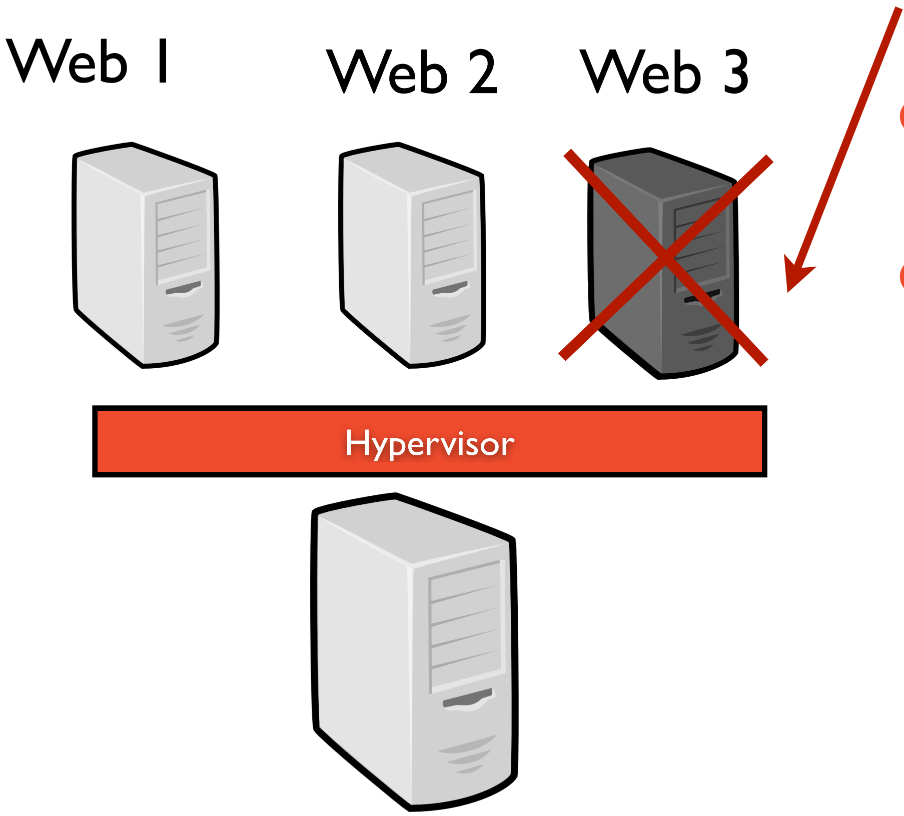


- Isolation (security between each VM)
- snapshot/suspend/resume/reboot (maintenance)

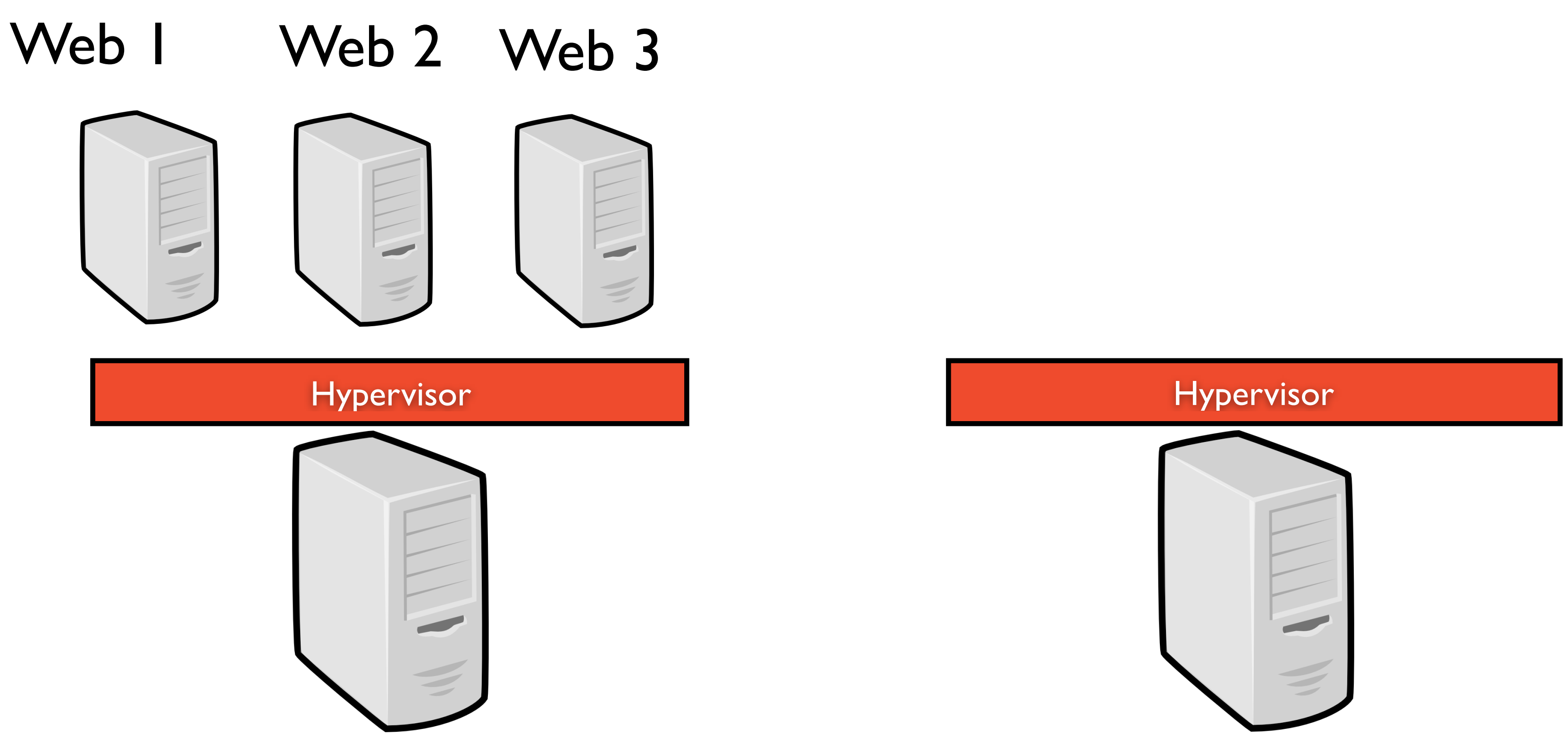


- Consolidation (load-balancing)
- Negligible downtime (~ 60 ms)

VM Capabilities

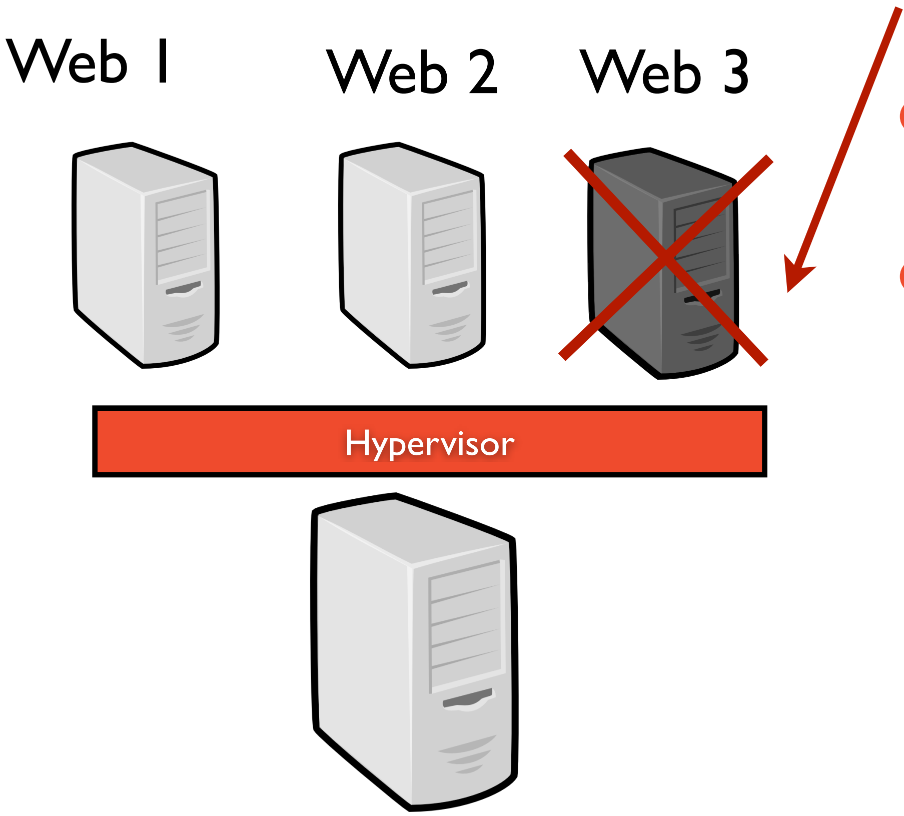


- Isolation (security between each VM)
- snapshot/suspend/resume/reboot (maintenance)

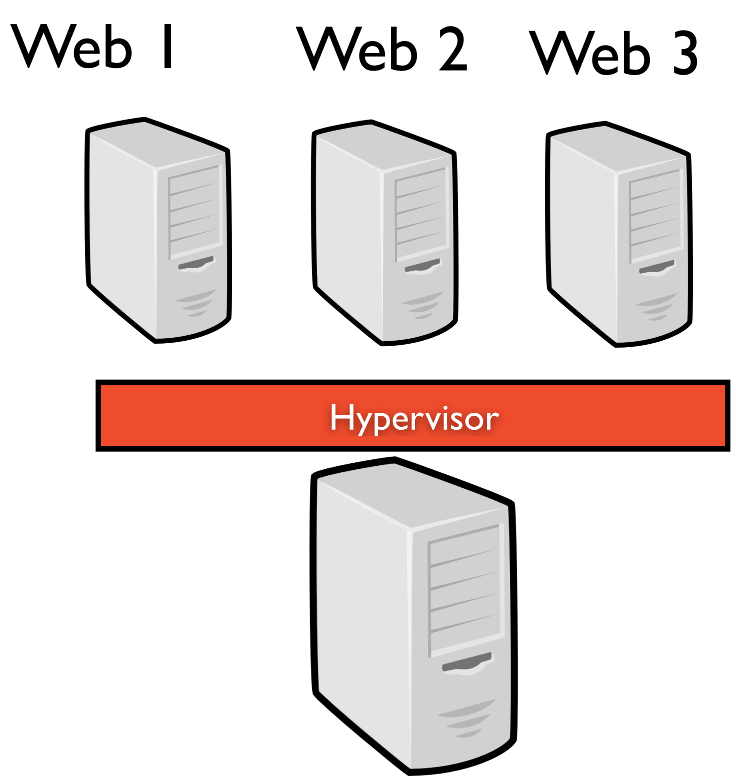


- Consolidation (load-balancing)
- Negligible downtime (~ 60 ms)

VM Capabilities



- Isolation (security between each VM)
- snapshot/suspend/resume/reboot (maintenance)

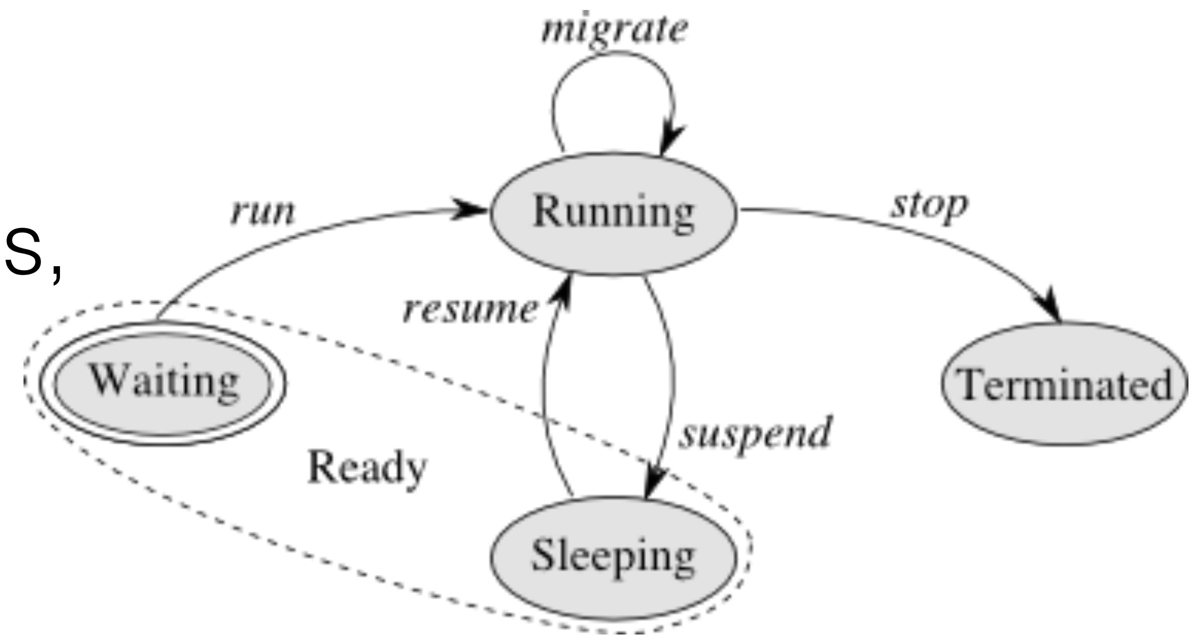


- Consolidation (load-balancing)
- Negligible downtime (~ 60 ms)

A VM-based Operating System ?

- General idea: manipulate vjobs instead of jobs (by encapsulating each submitted job in one or several VMs)
[Hermenier et al., VTDC 2010]

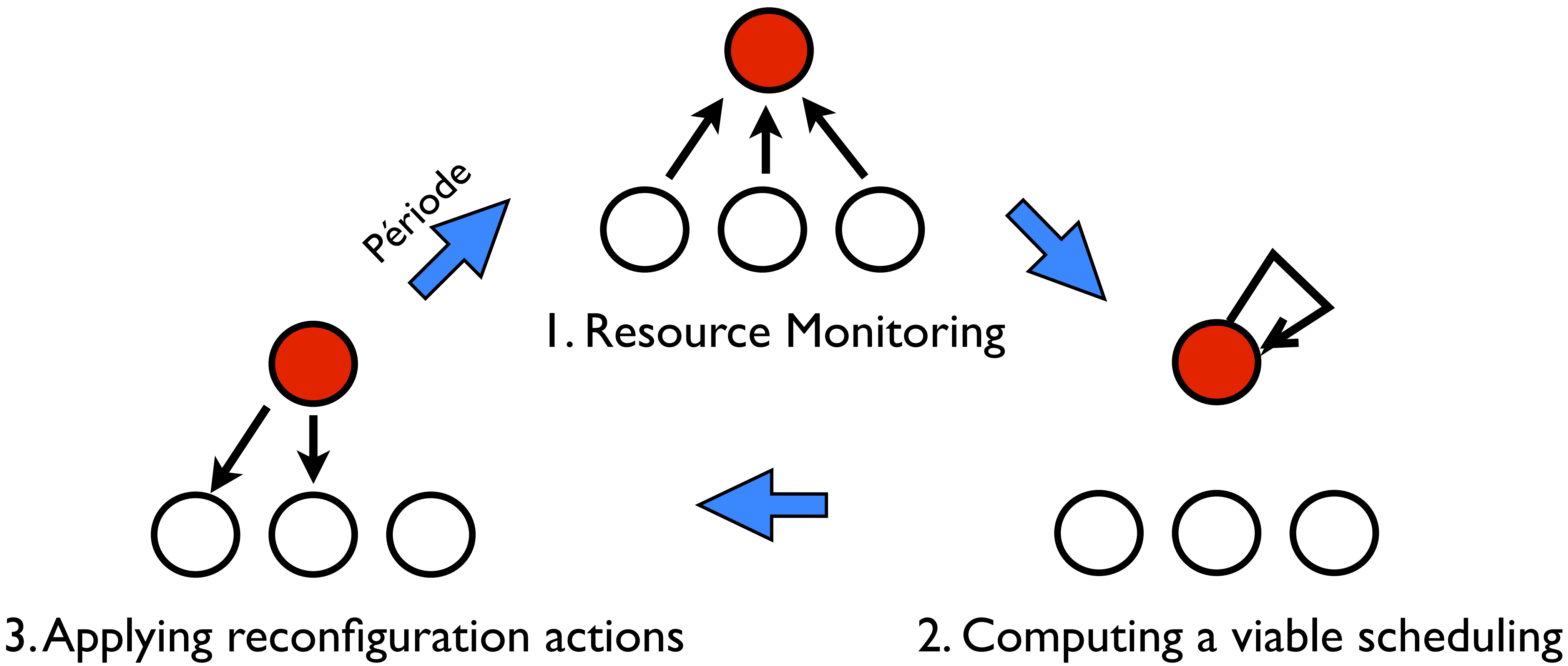
- In a similar way of usual processes, each vjob is in a particular state:



- A vjob context switch (a set of VM context switches) enables to efficiently rebalance the distributed infrastructures according to the scheduler objectives / available resources / waiting vjobs queue

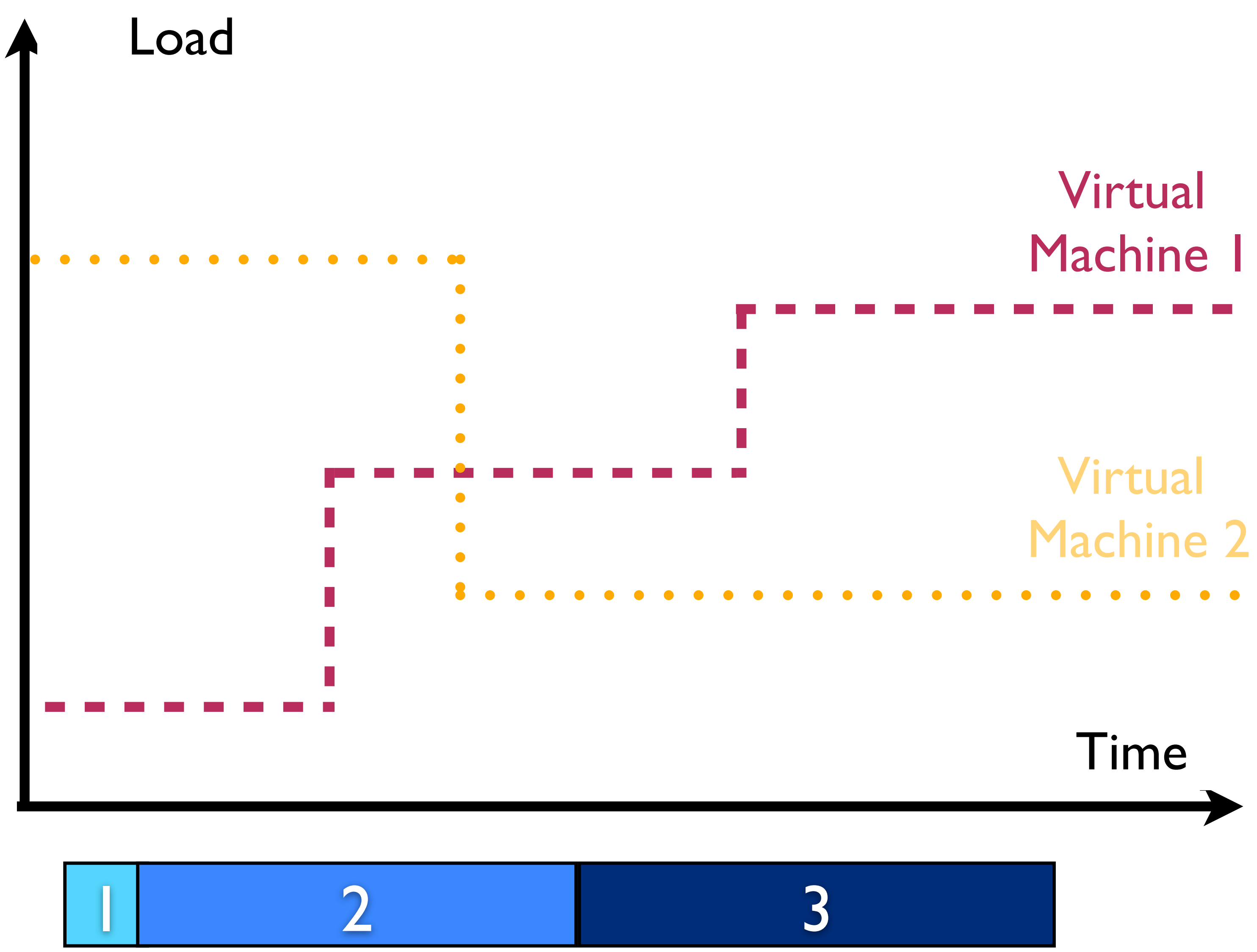
Back to 2009

- Centralized approach: the Entropy proposal [Hermenier et al., VEE 2009], a success story !



Back to 2009

- Centralized approach: the Entropy proposal [Hermenier et al., VEE 2009], a success story !

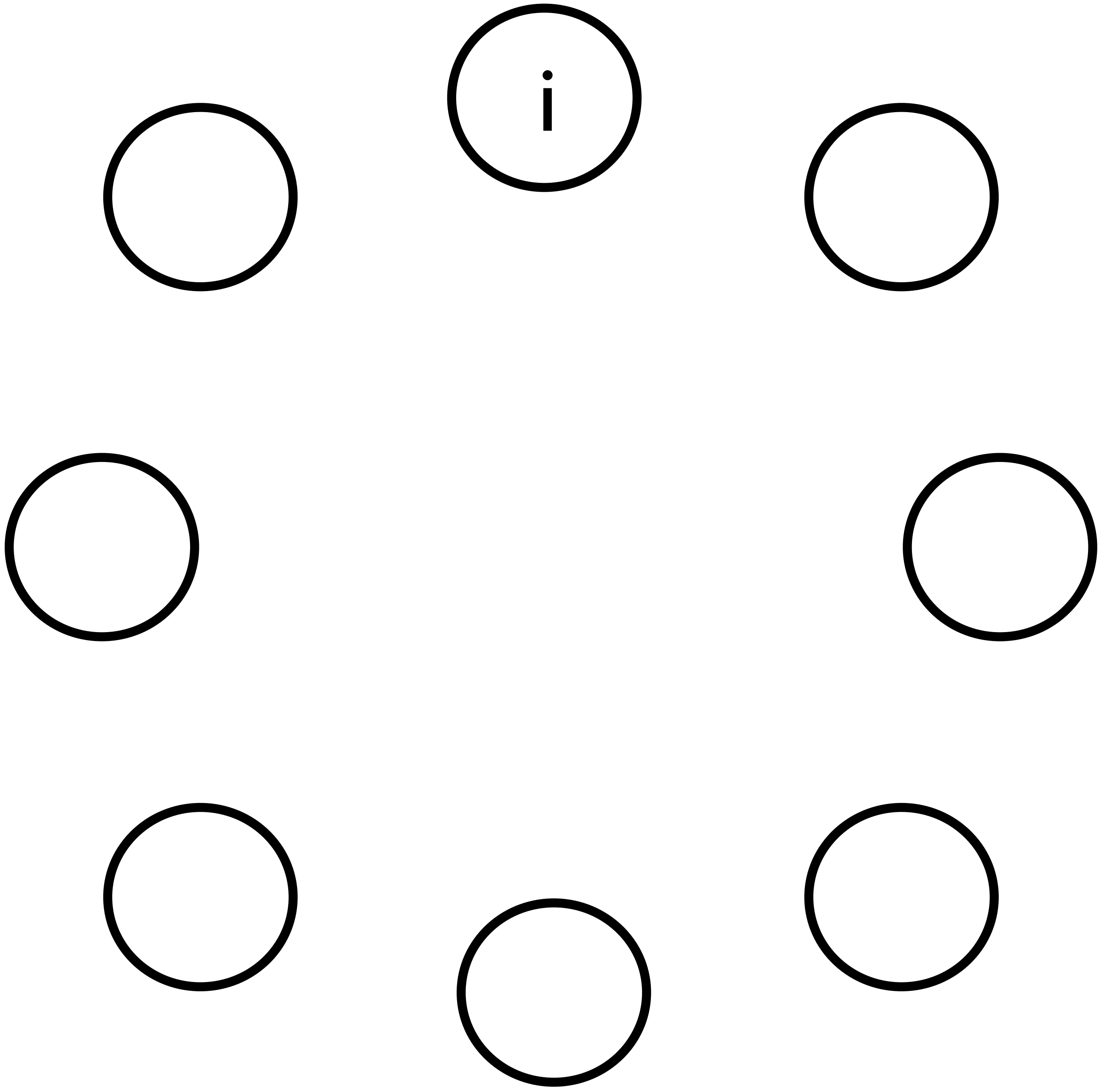


⇒ Scalability/Reactivity concerns

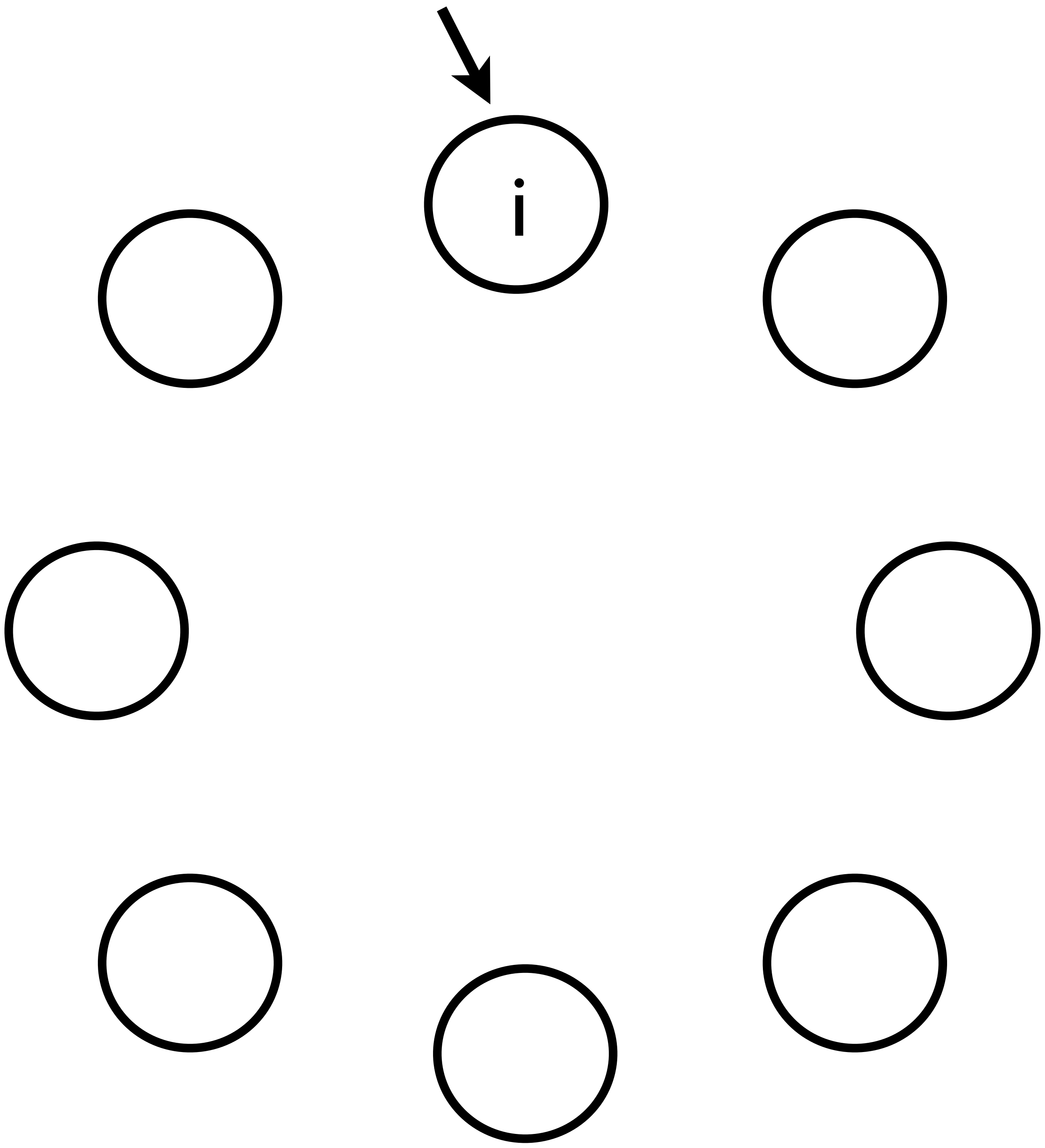
Distributed VM Scheduler

- Cooperation between direct neighbours to solve events
  - Event driven
  - Peer to peer, no service node
  - Local interactions between nodes
  - Monitoring
  - Scheduling

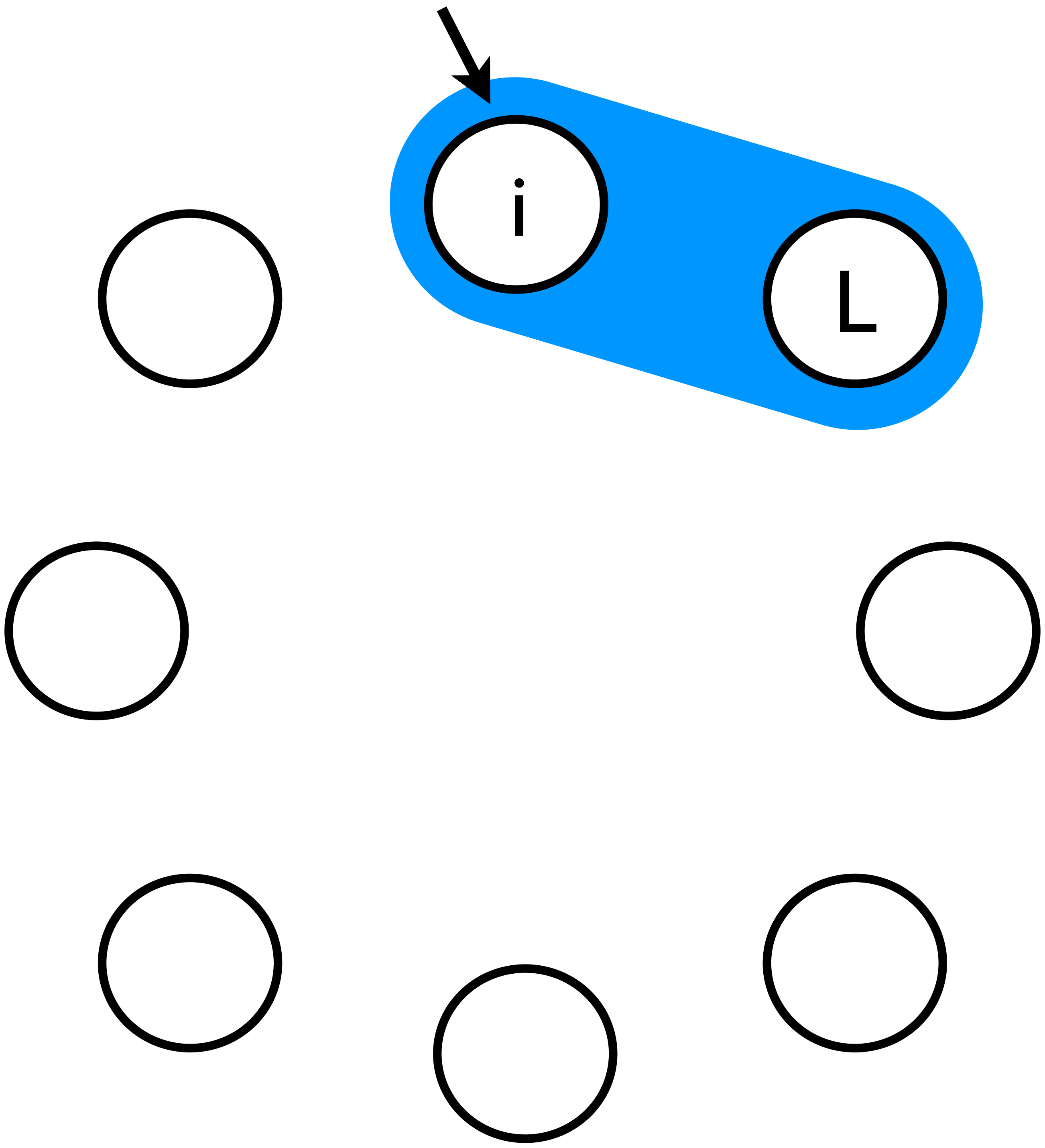
Distributed VM Scheduler



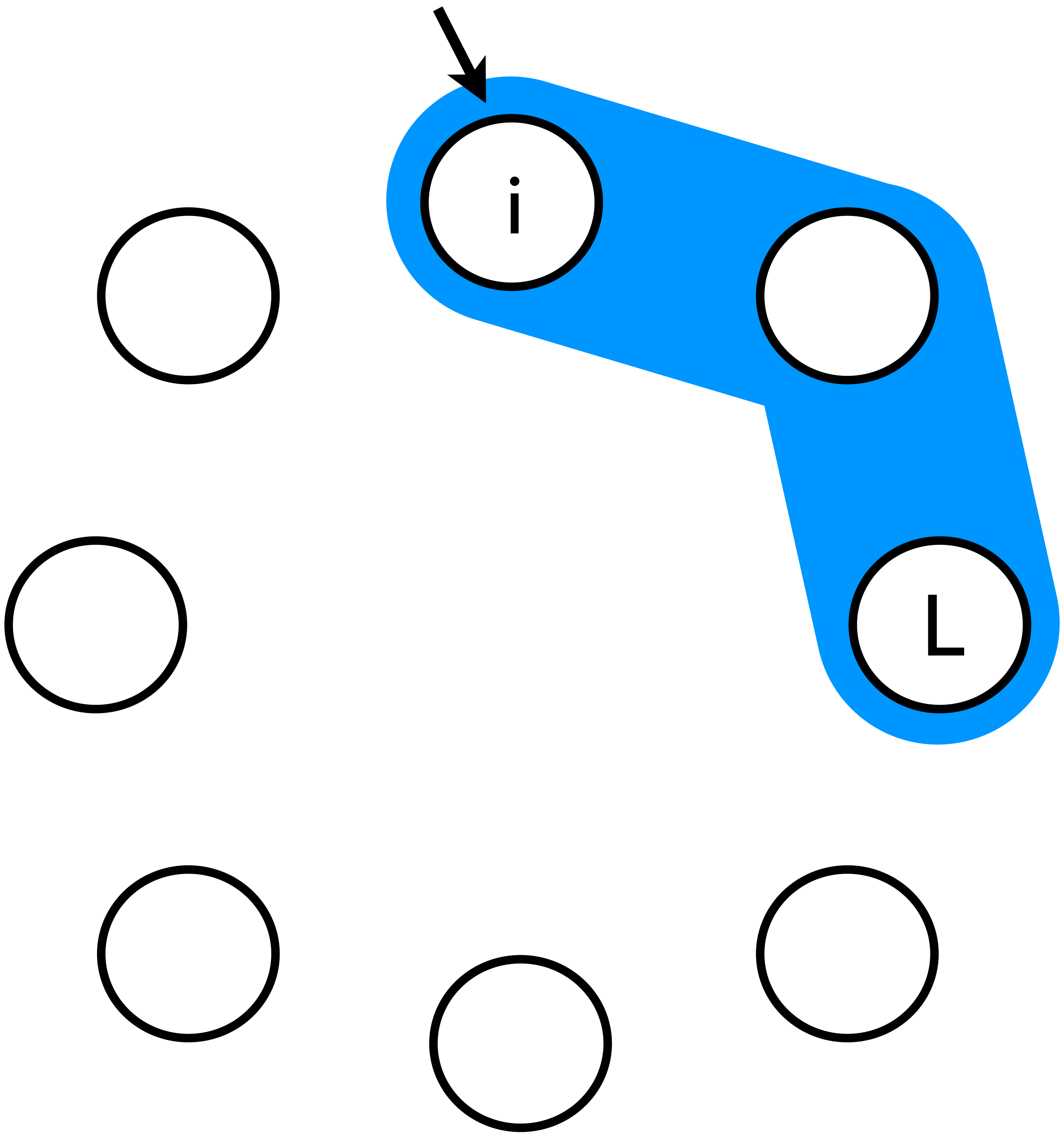
Distributed VM Scheduler



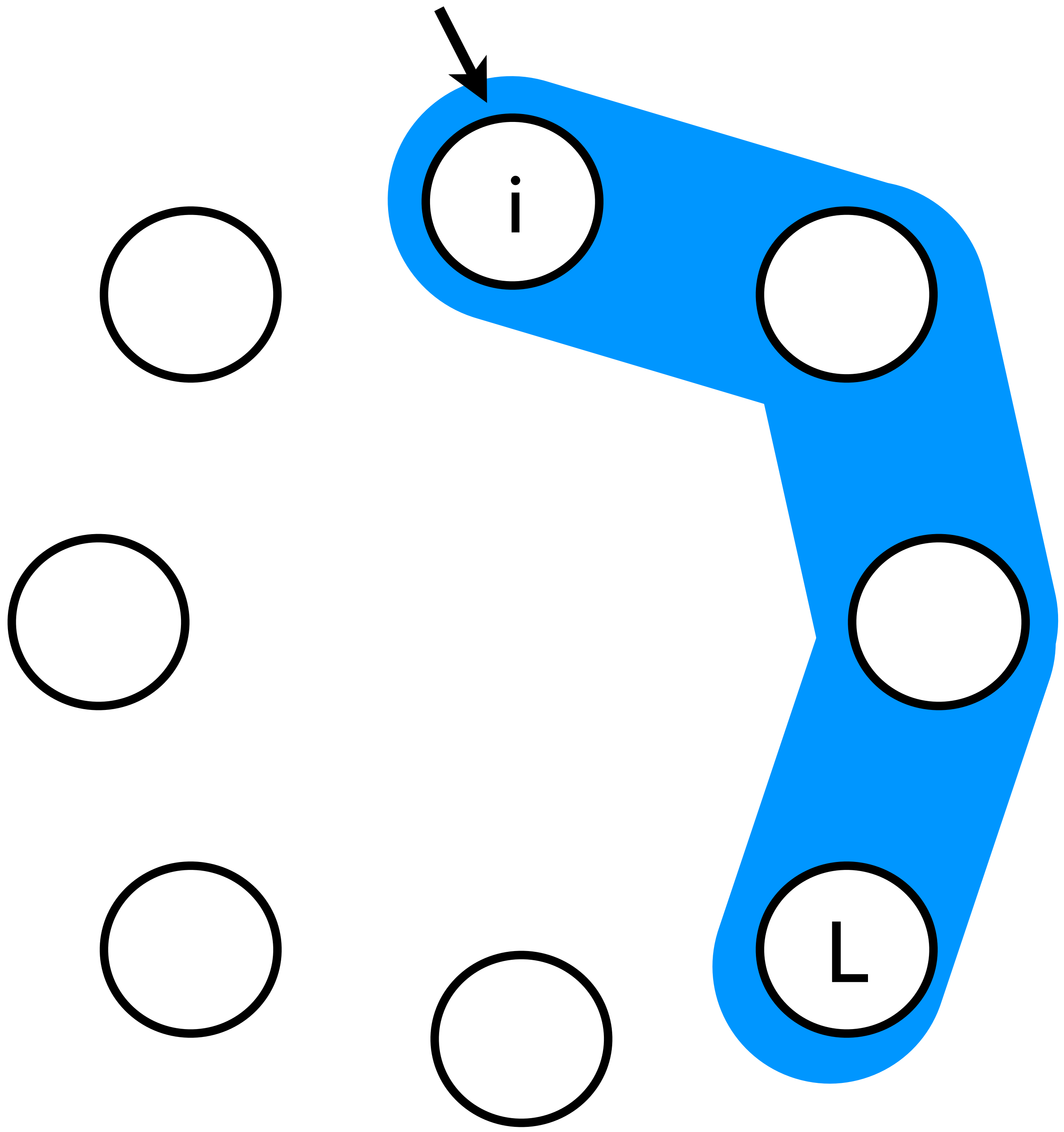
Distributed VM Scheduler



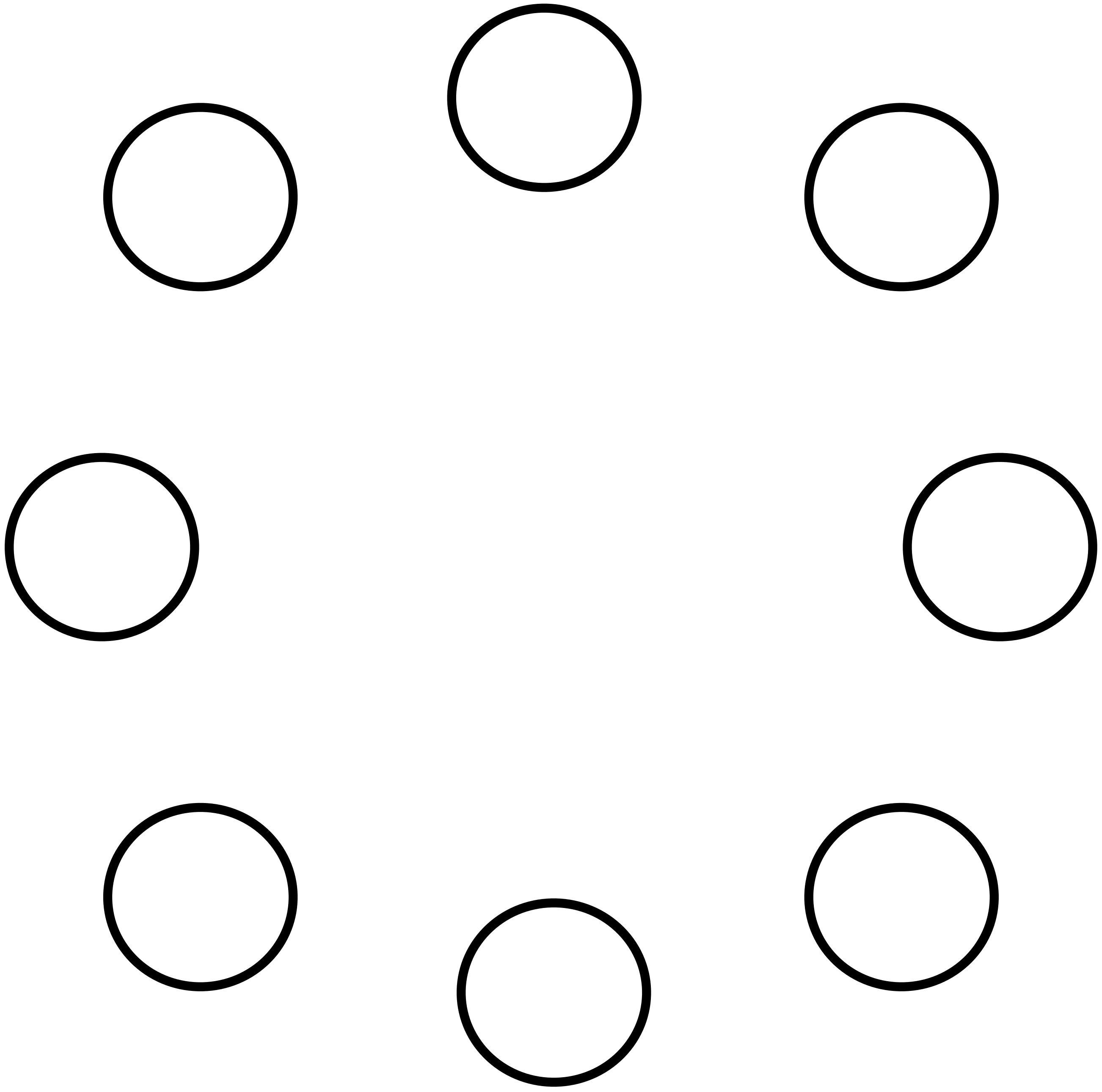
Distributed VM Scheduler



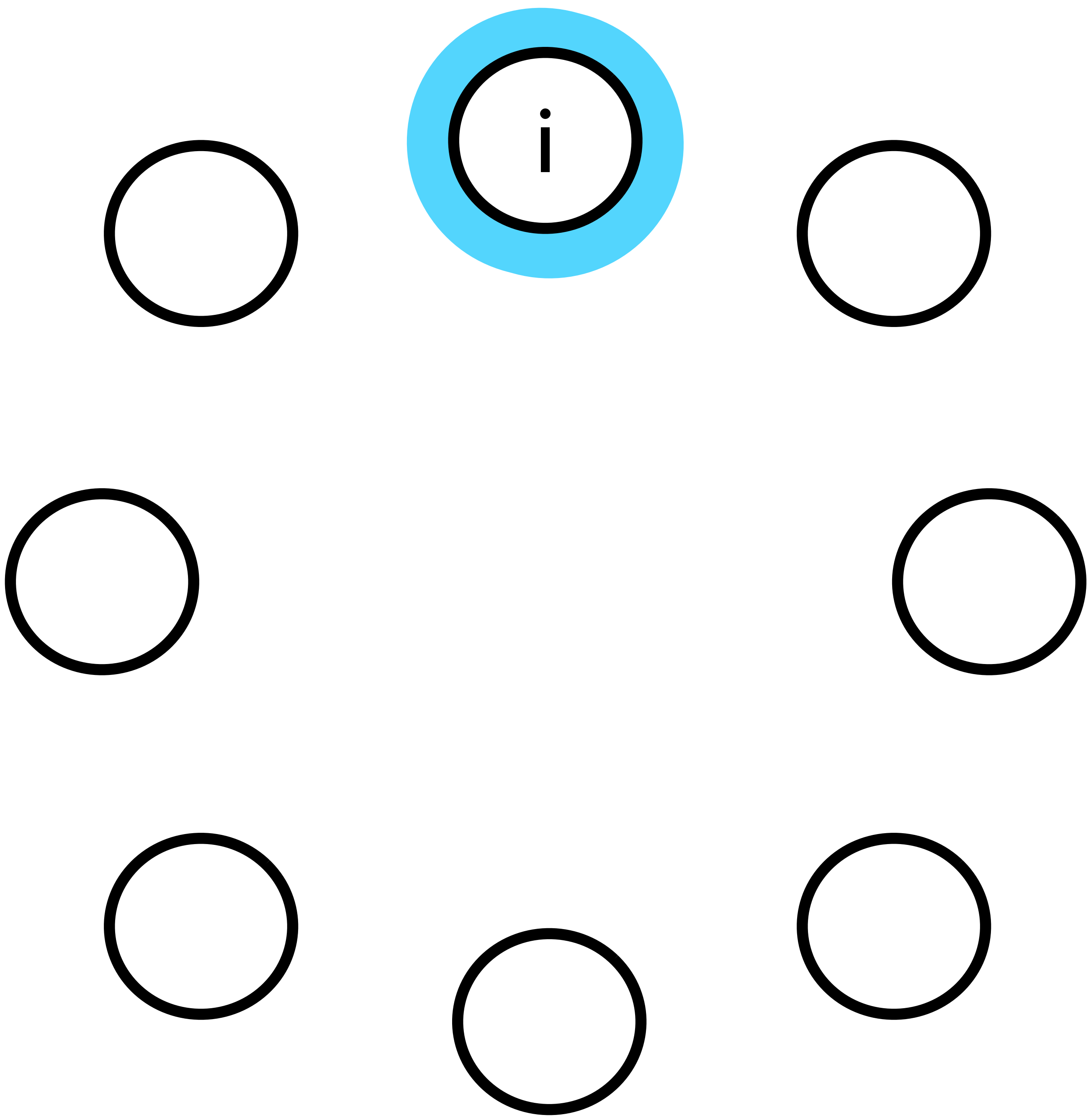
Distributed VM Scheduler



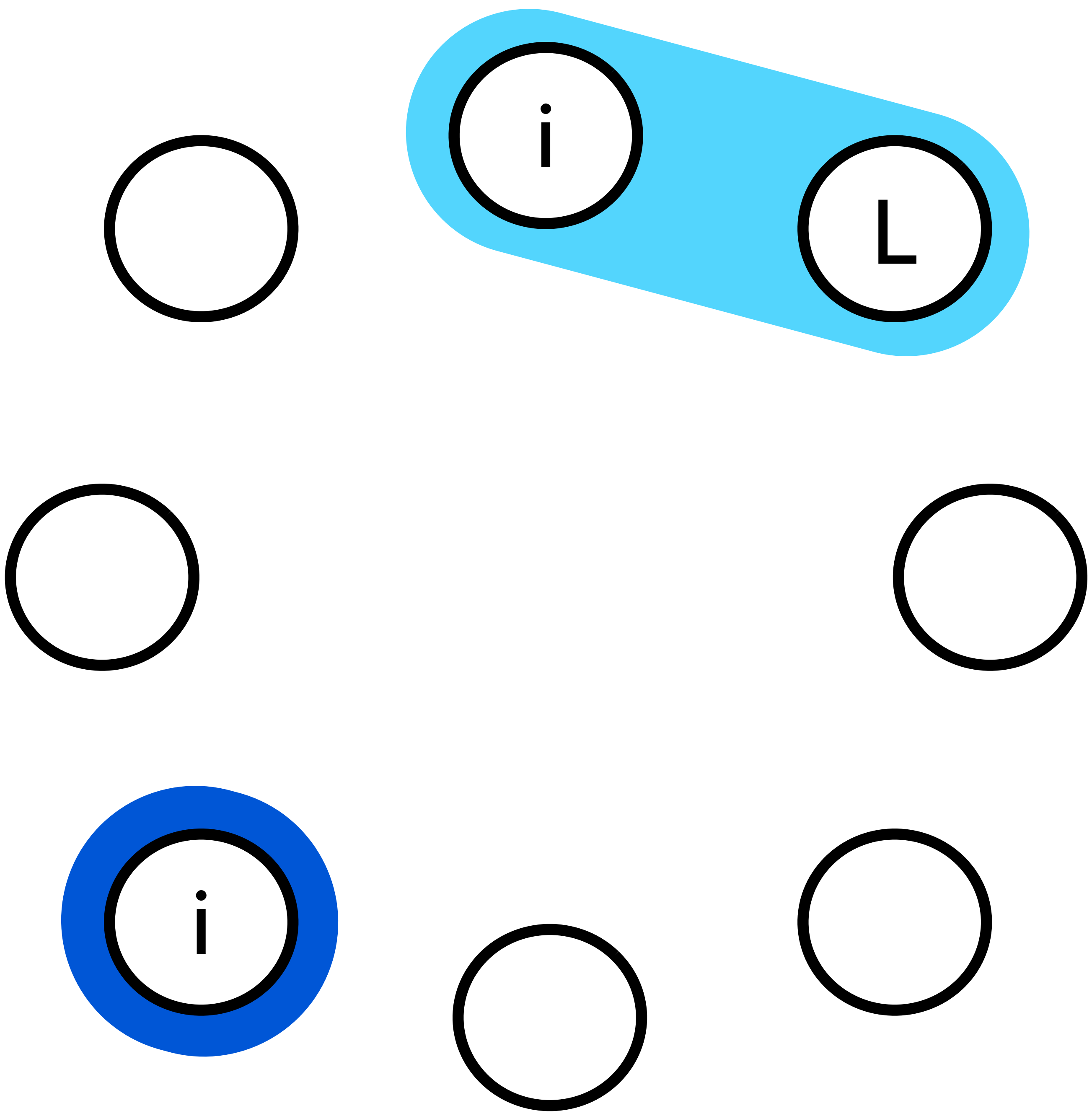
Distributed VM Scheduler



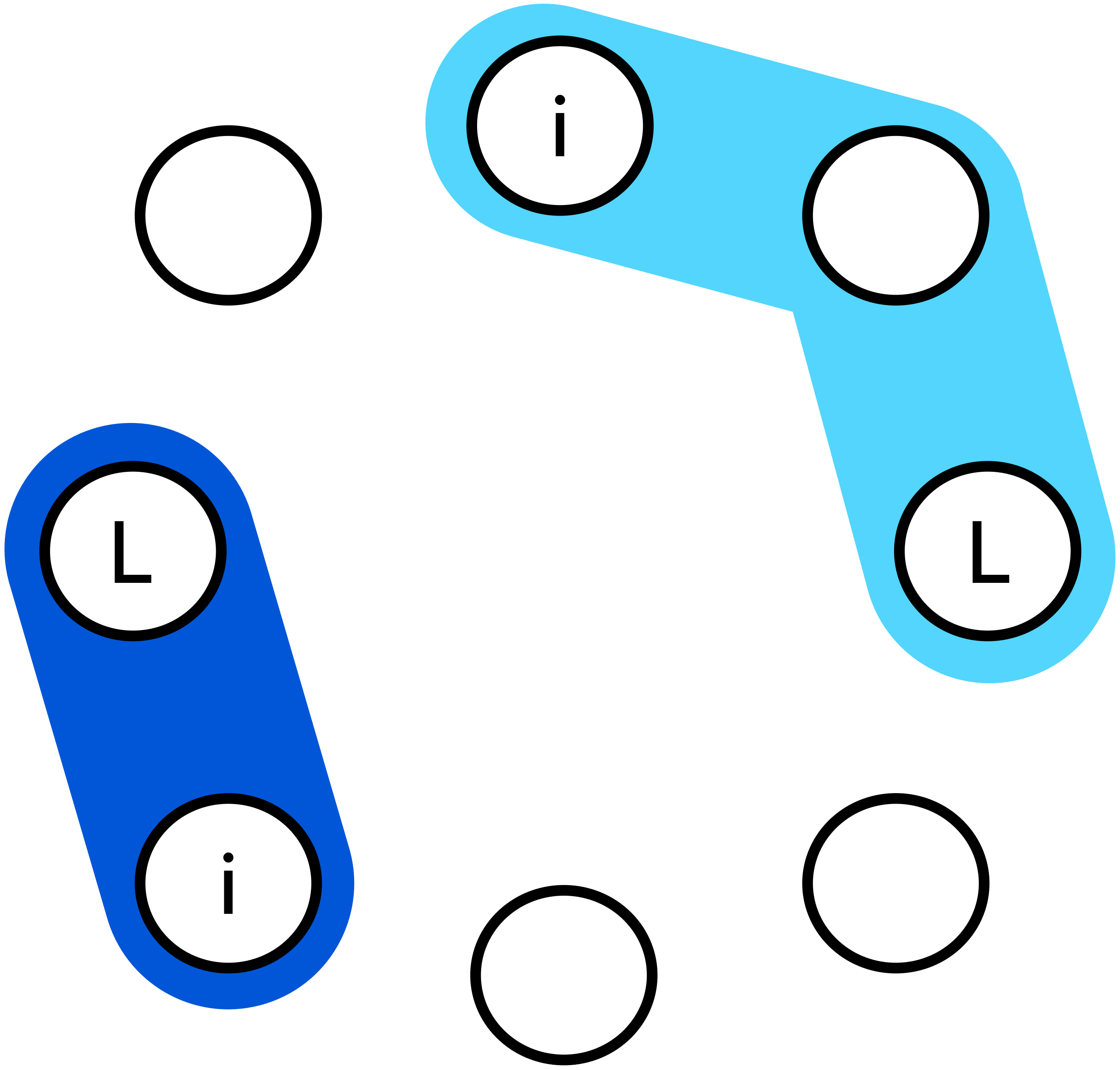
Distributed VM Scheduler



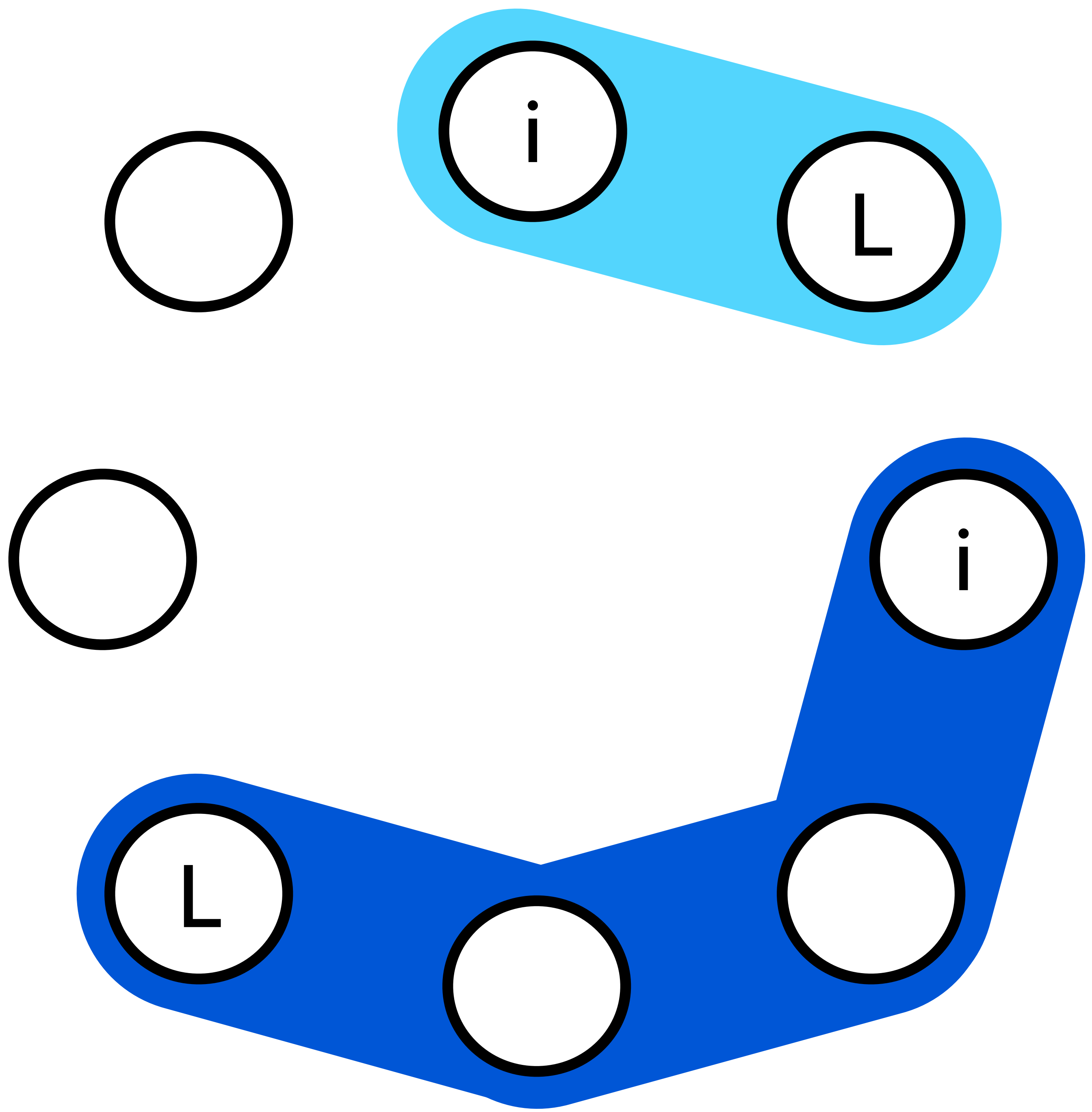
Distributed VM Scheduler



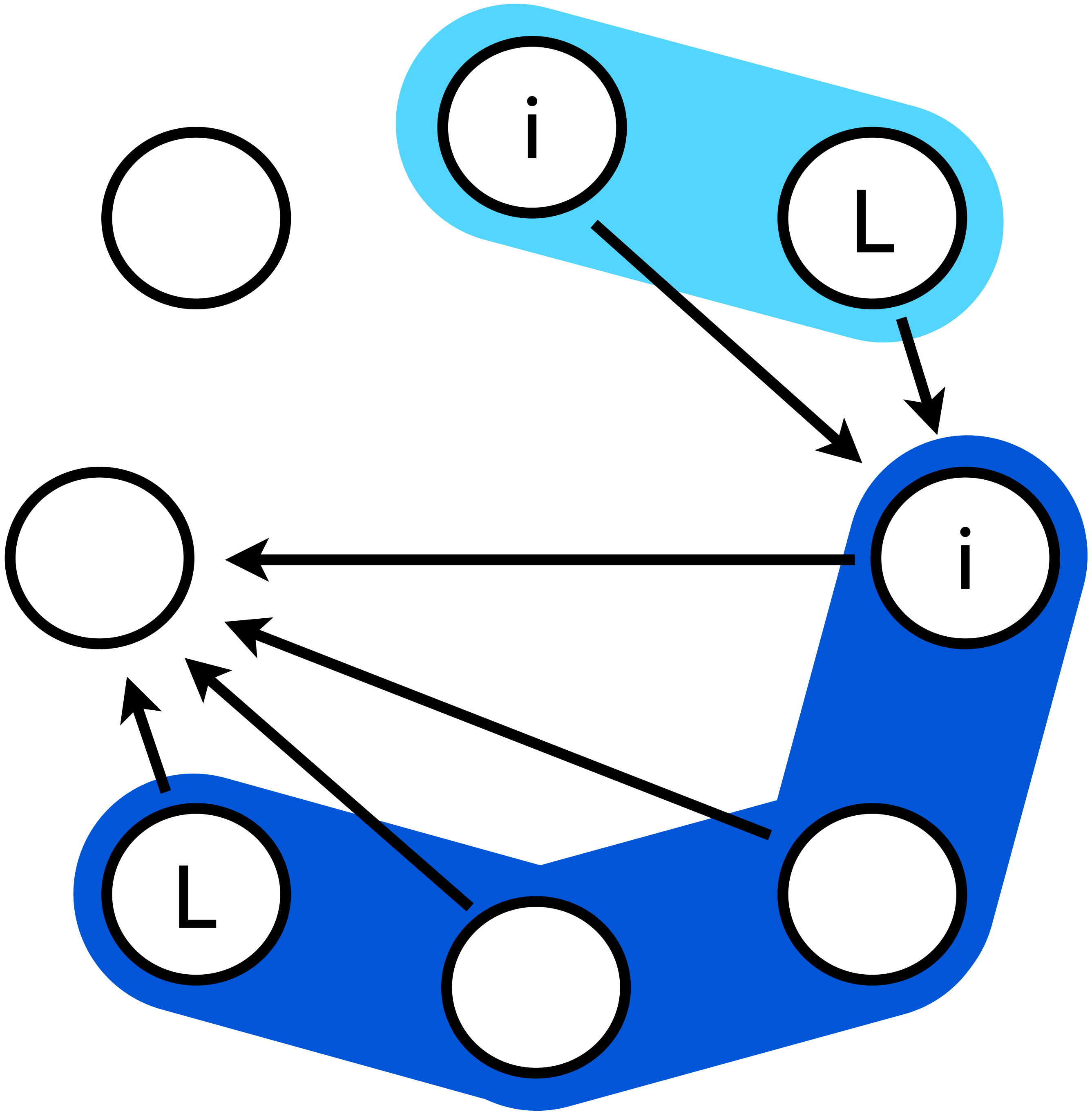
Distributed VM Scheduler



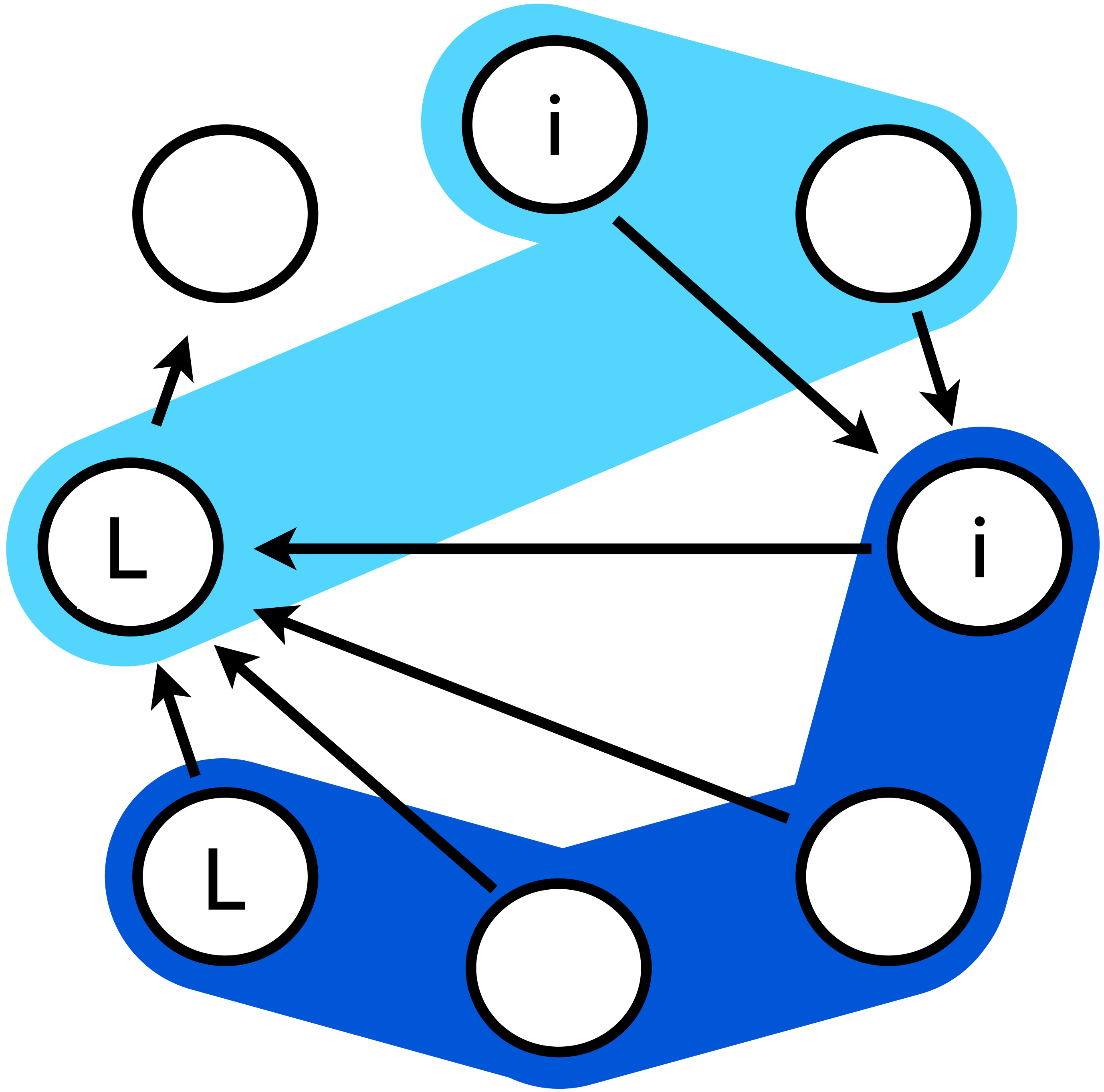
DVMS - Shortcuts



DVMS - Shortcuts

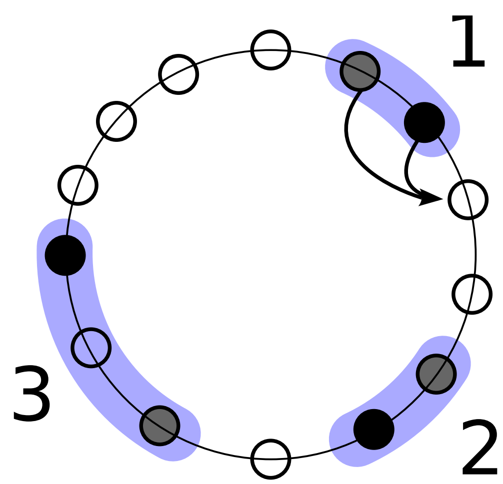


DVMS - Shortcuts



Distributed VM Scheduler

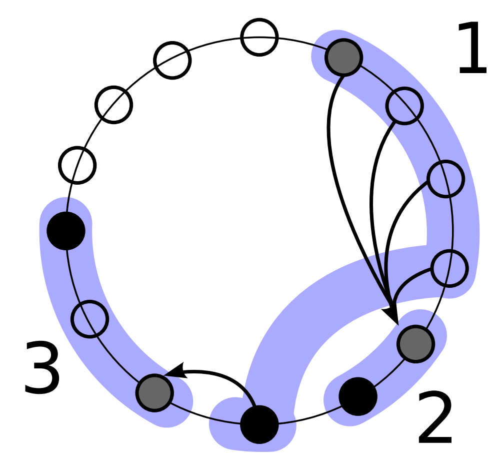
- Cooperation between direct neighbours to solve events
  - Nodes have a local view of the system / Local invocation of the resolution algorithm
  - Simulation (using Simgrid)10K PMs, 80K VMs
    No live-migration model, in vivo experiments are definitely mandatory
  - **Flauncher, deploying a large number of VM on top of Grid5000**
    (With the support of Hemera, 6 FTE months)
    From days to two/three hours to deploy such a testbed
    Finalist **IEEE Scale challenge 2013 (500 PMs, 4500 VMs)**
    [Quesnel et al., CPE 2013]



credits: F. Quesnel et al., DVMS April 2012

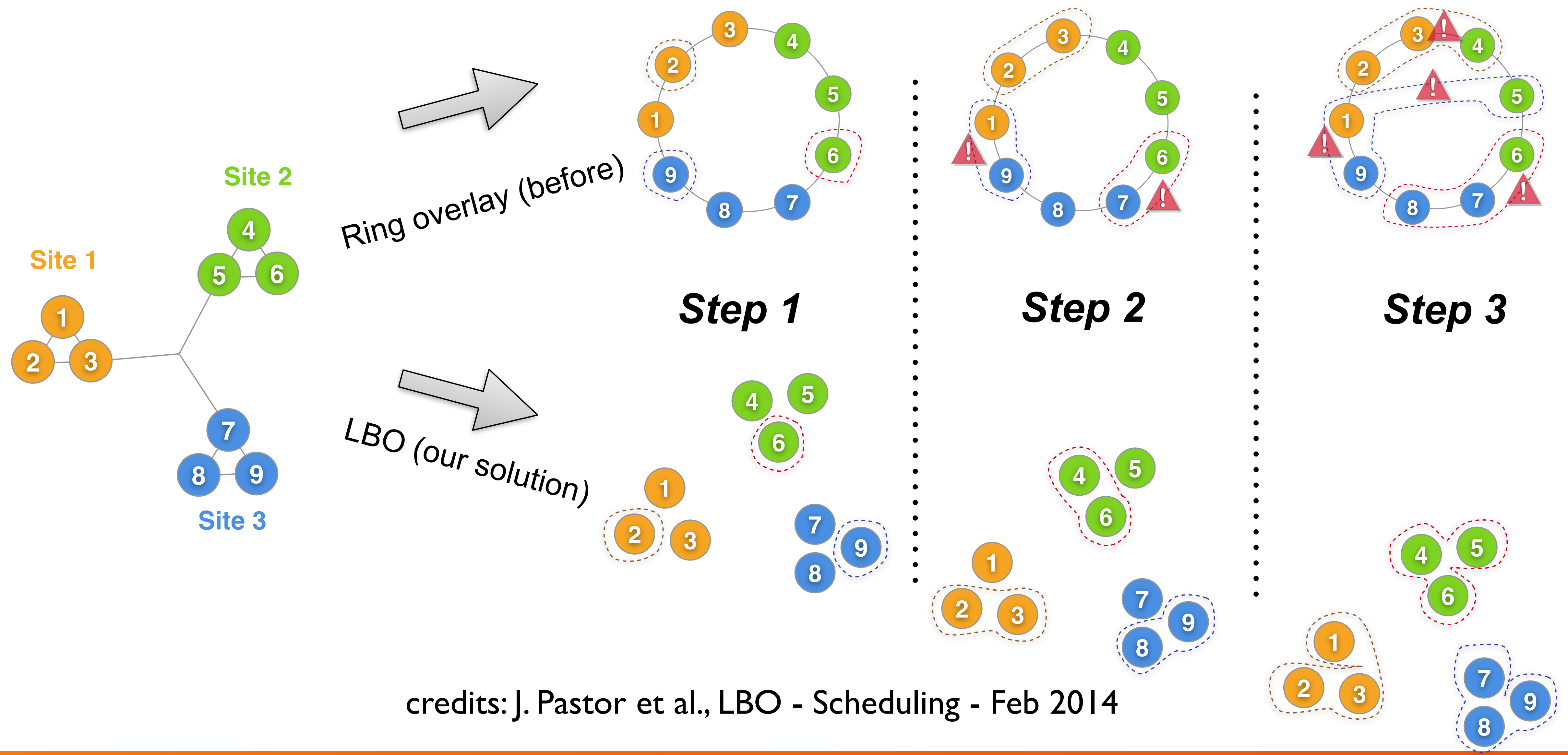

Scalability/reactivity but….


…matching a ring on a real network backbone



Distributed and Locality-aware

- Leverage a locality based overlay (vivaldi) + a shortest path algorithm to favour cooperations between close nodes



credits: J. Pastor et al., LBO - Scheduling - Feb 2014

Distributed and Locality-aware

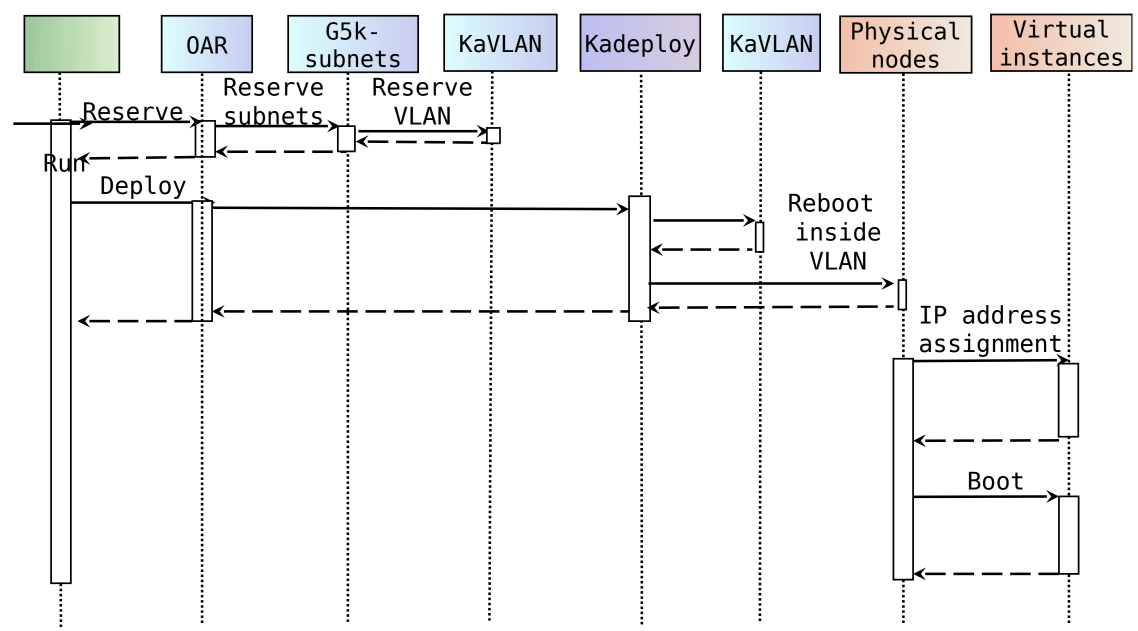
- Leverage a locality based overlay (vivaldi) + a shortest path algorithm to favour cooperations between close nodes
- A collaboration between ASAP, ASCOLA and MYRIADS [pastor et al., Europar 2014]
- Leveraging vm5k to validate the prototype
  - vm5k: a Flauncher production ready system
    Completely rebuilt on top of the Execo framework (Python)
  - Winner ex aequo of the Grid'5000 challenge

Next step: storage dimension

Conclusion

- System virtualisation changed the distributed computing landscape (from the process to the container granularity: xen, KVM, dockers,…)

- Investigating ``containerization'' concerns implies to …
  Deploy the template
  Configure/Start each instance
  Control the execution
  ... before conducting experiments



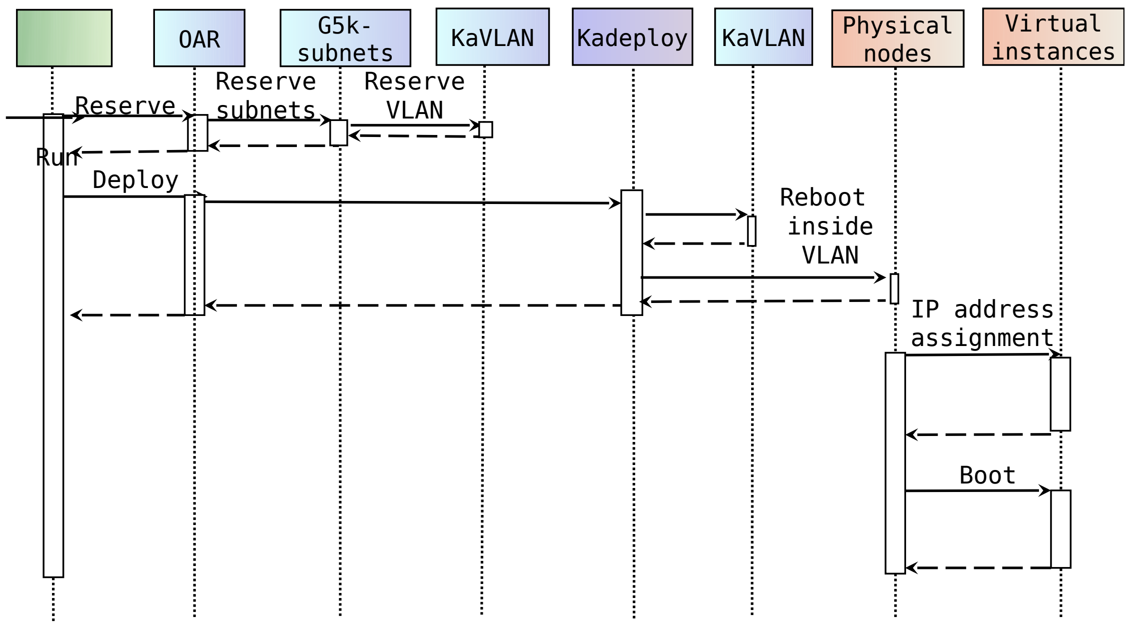
- Performing such a task on
  - Few VMs on one node 
  - Hundred of VMs on one site 
  - Thousands of VMs on distinct sites 

- HEMERA contribution: designing/implementing tools to make the study and the investigation of such concerns at large scale easier

Conclusion

- System virtualisation changed the distributed computing landscape (from the process to the container granularity: xen, KVM, dockers,…)

- Investigating ``containerization'' concerns implies to …
  Deploy the template
  Configure/Start each instance
  Control the execution
  ... before conducting experiments



- Performing such a task on
  - Few VMs on one node
  - Hundred of VMs on one site
  - Thousands of VMs on distinct sites

- HEMERA contribution: designing/implementing tools to make the study and the investigation of such concerns at relevant scale easier

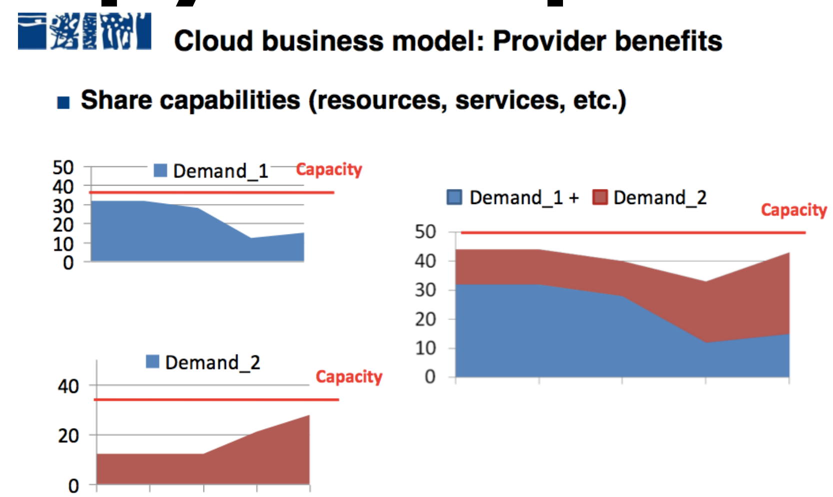
Questions

- Hemera Virtualization related activities
  - Two national meetings (2011,2014 - 25 attendees)
    One internal one (2012 - 10 attendees)
    Organization of the ACM VTDC workshop collocated with the HPDC conference (2011, 2012, 2013)
  - Two challenges (large scale deployment and Virtual Machine Performance)
  - 2012/2013, deploy major toolkits (OpenNebula, Nimbus, CloudStack) with the financial support of the EIT ICT lab.
  - vm5k (1 year FTE taking into account previous development of Flauncher)
  - Several publications, twice IEEE finalists (second prize in 2013 with the Snooze proposal)
  - Five on-going activities leveraging the vm5k/Execo framework (from G5K to SimGrid and beyond)
    VM Booting time, multi-core and virtualisation concerns (collocation/migration), HDD I/O competition
  - A springboard or a rather a launch pad for the Discovery IPL ;)

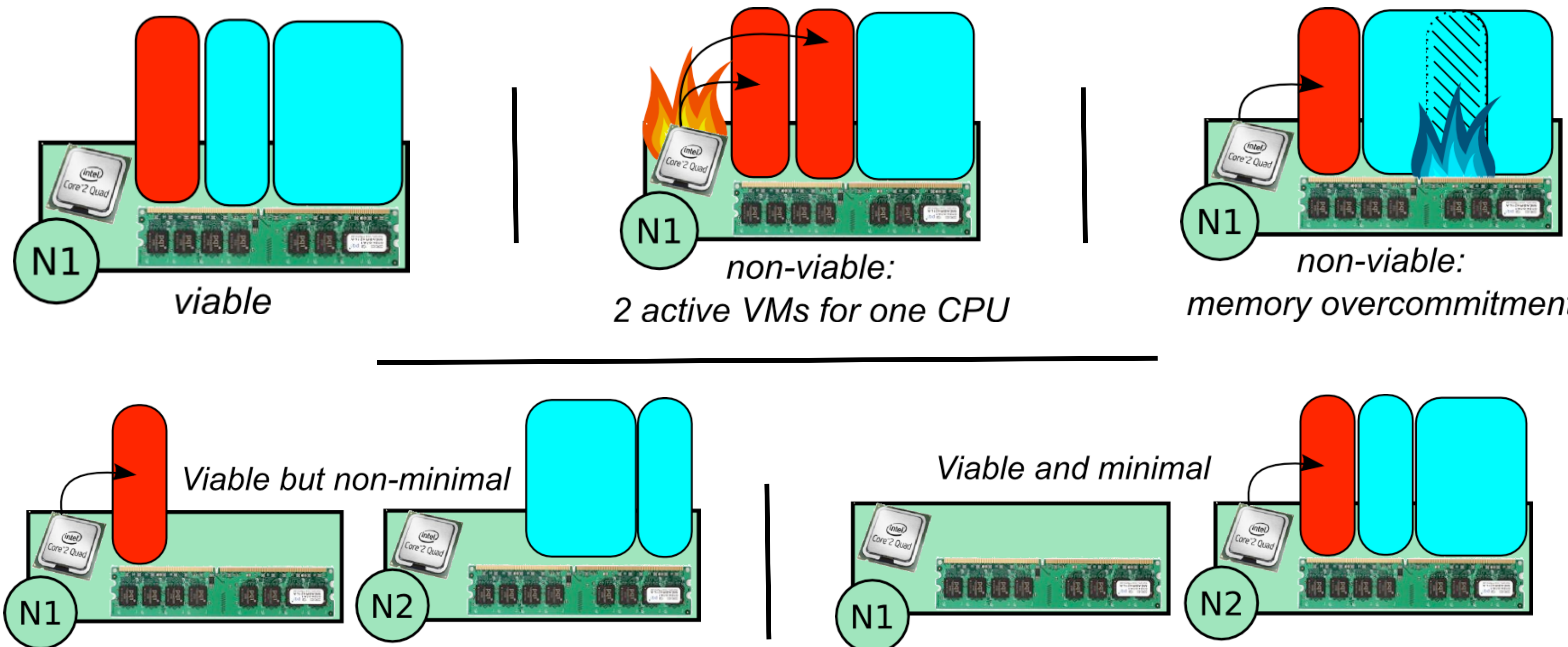
Background

Background - The Entropy Proposal

- Fine management of resources (efficiency and energy constraints)
- Find the "right" mapping between needs of VMs and resources provided by PMs

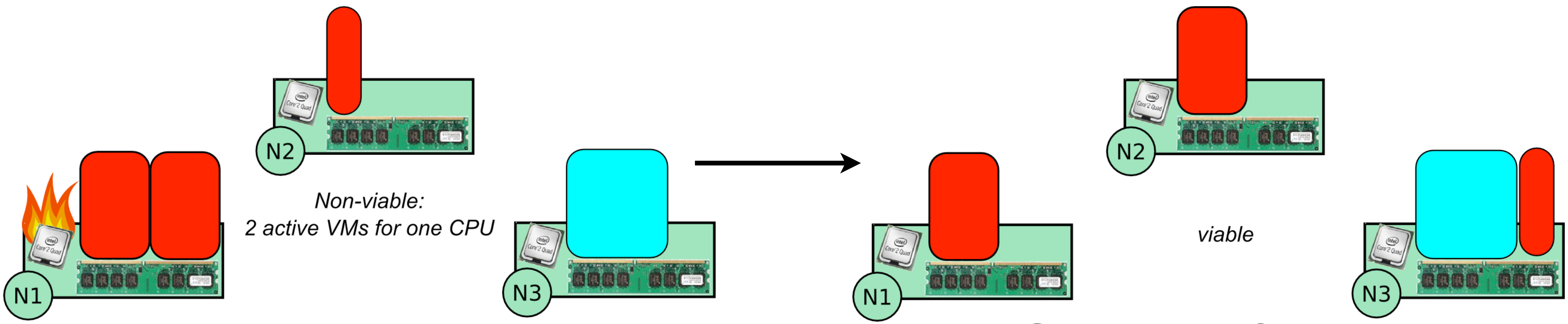


credits: S. Tata, Telecom Summer School 2013



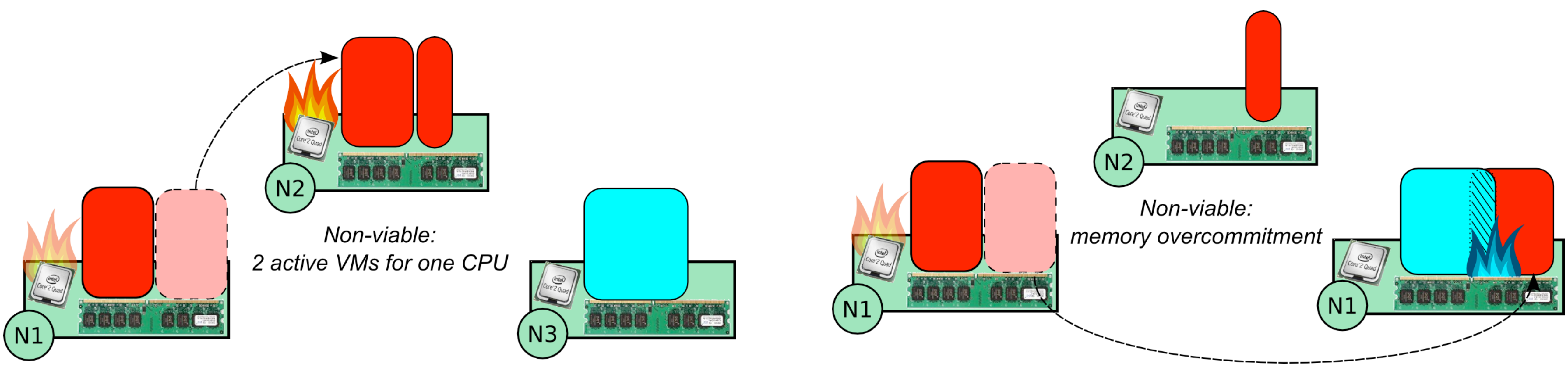
credits: F. Hermenier, OSDI poster session 2008

Background - The Entropy Proposal



Current Status

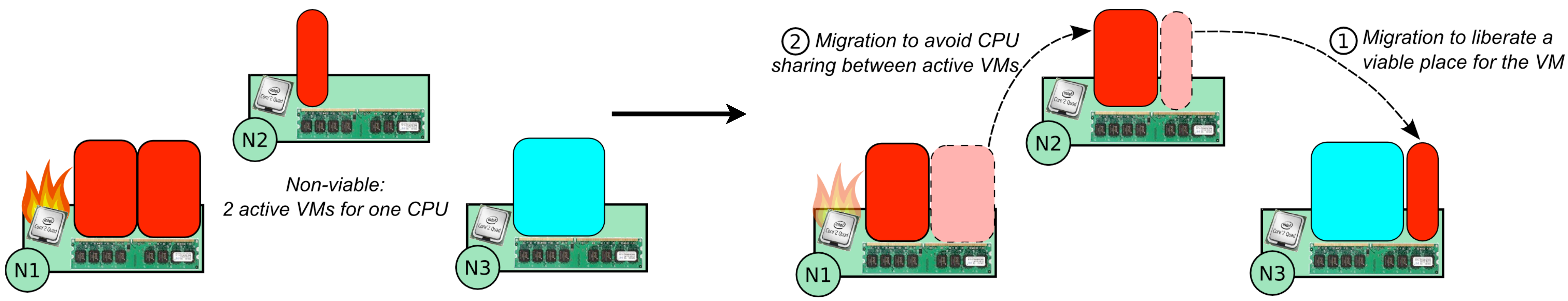
Correct Status



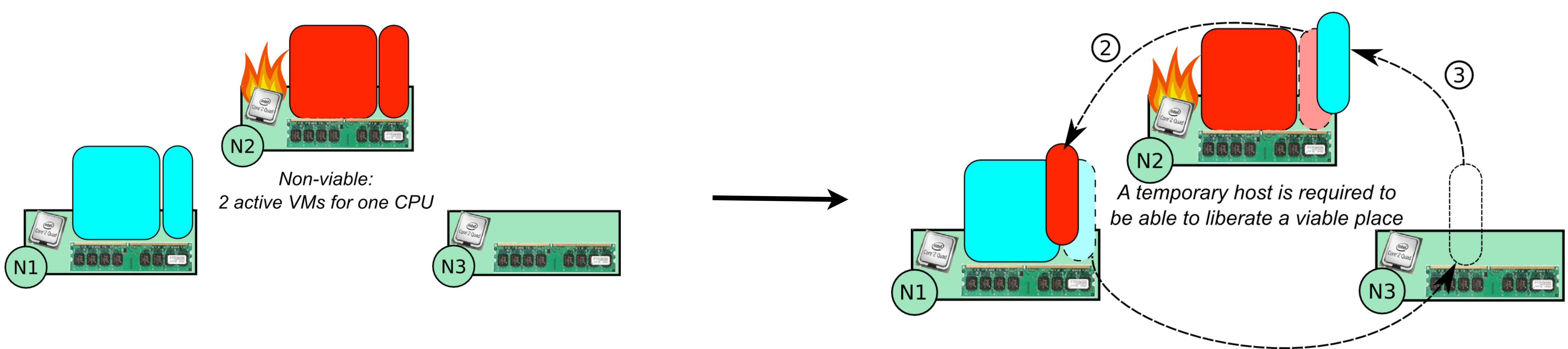
Non-viable manipulations

Background - The Entropy Proposal

- Order VM Operations



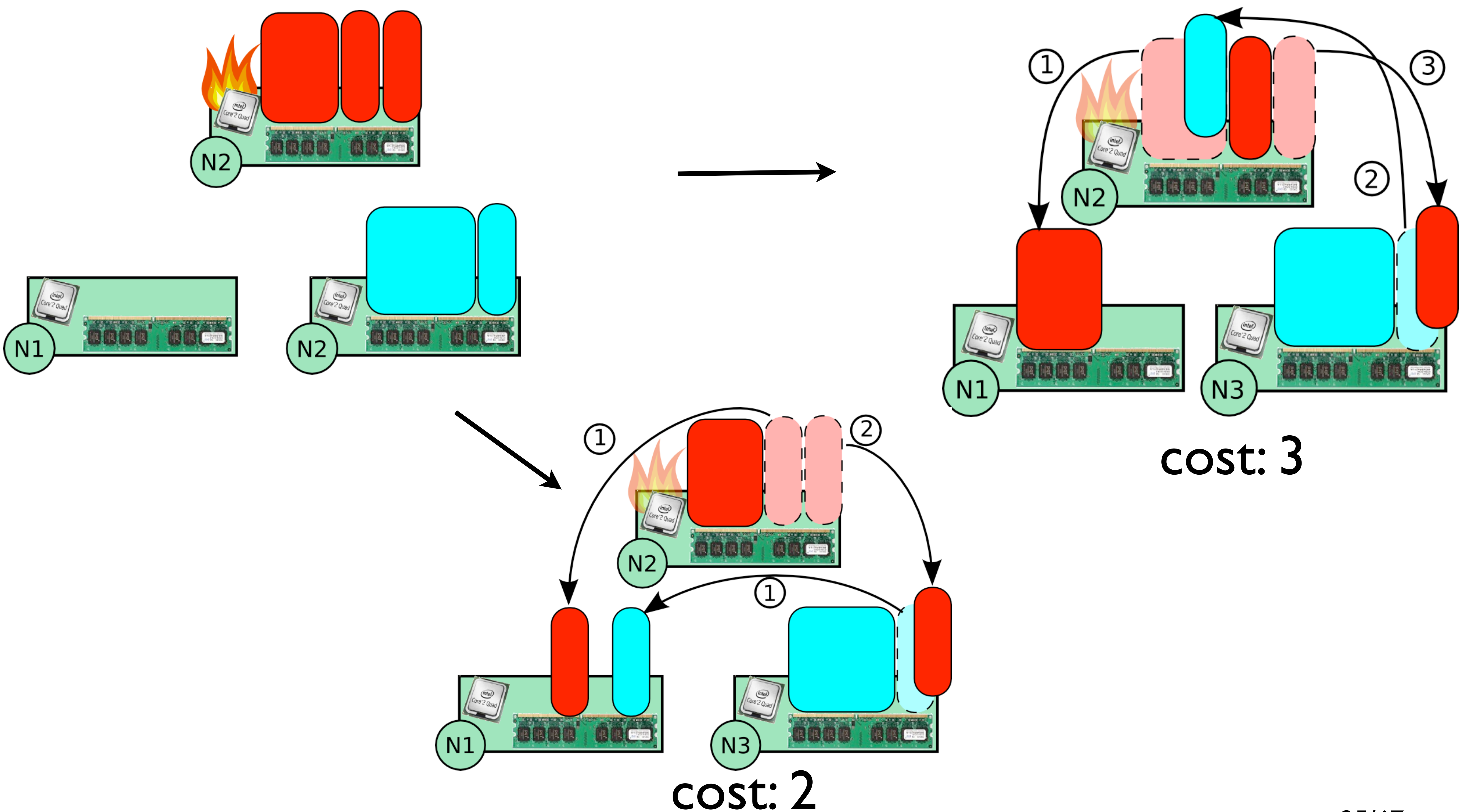
Sequential dependency



Cyclic dependency

Background - The Entropy Proposal

- Optimizing the reconfiguration process



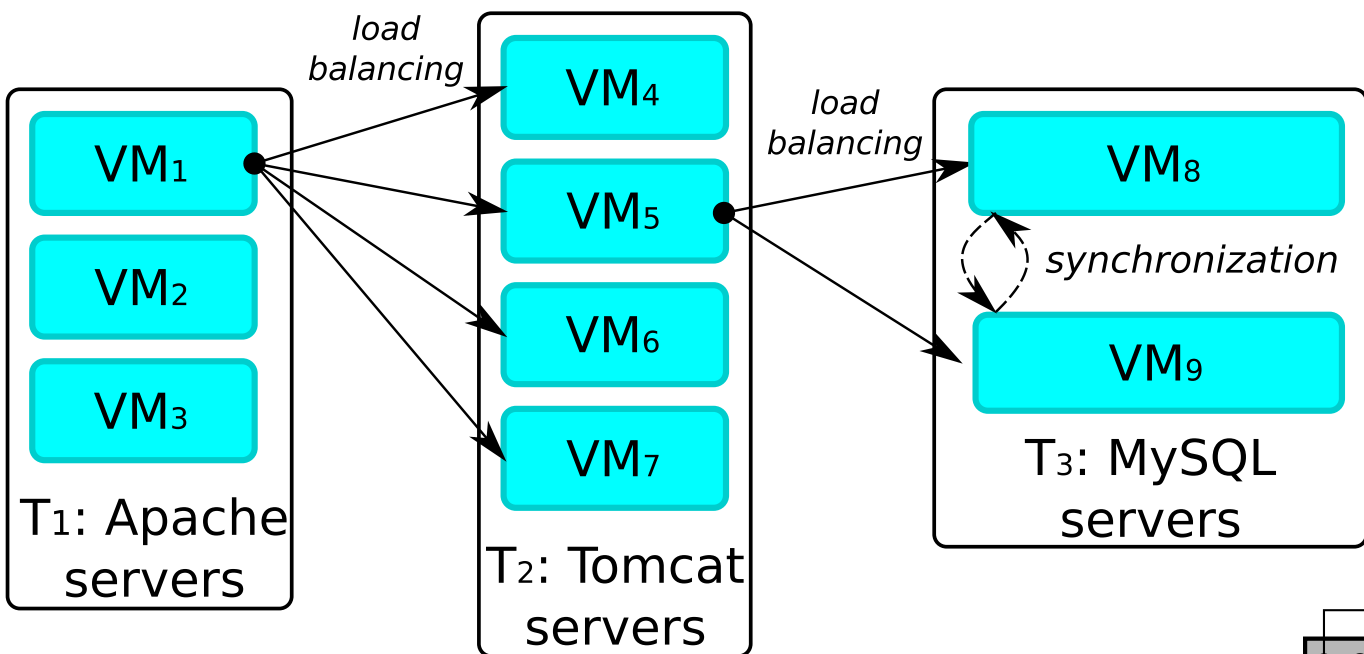
Background - The Entropy Proposal
More Constraints

- Manipulate VEs dynamically can lead to non desired configurations
- Additional constraints should be considered

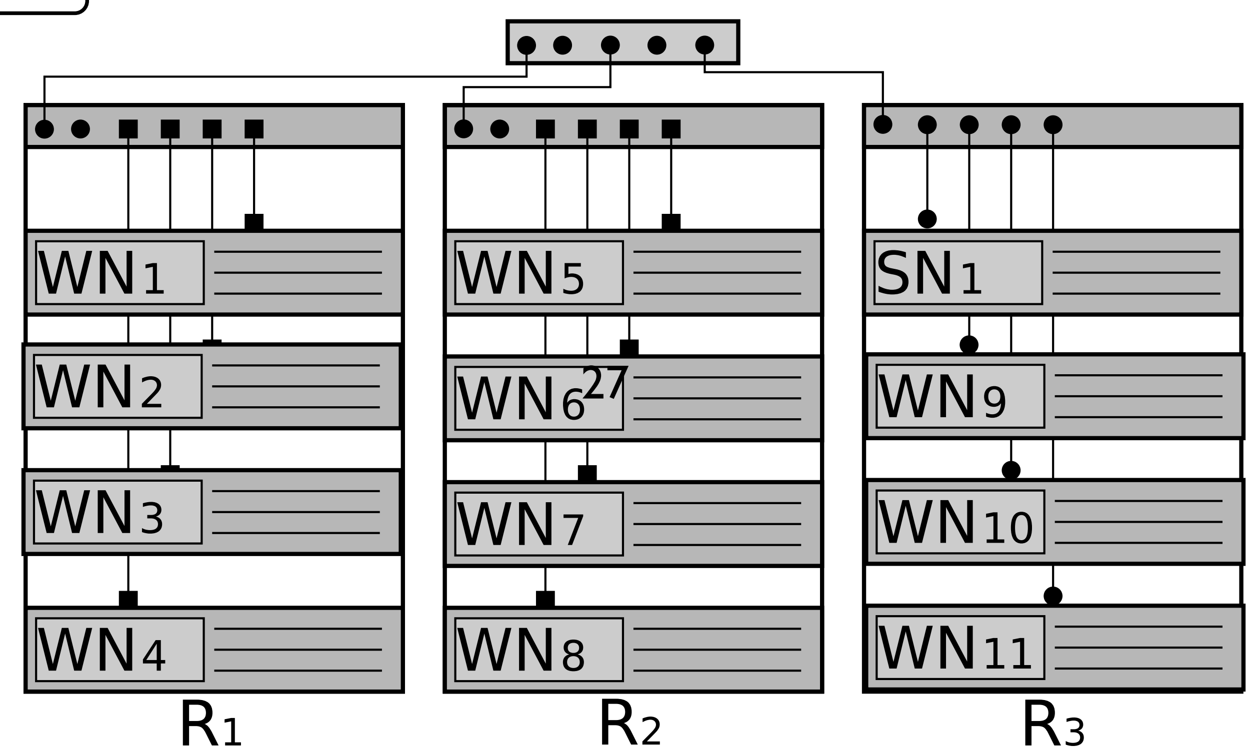
  To take into account particular requirements according to the infrastructure (performance, HA, maintenance operations....)

  To maintain VE "consistency" during reconfigurations

Background - Plasma and Entropy

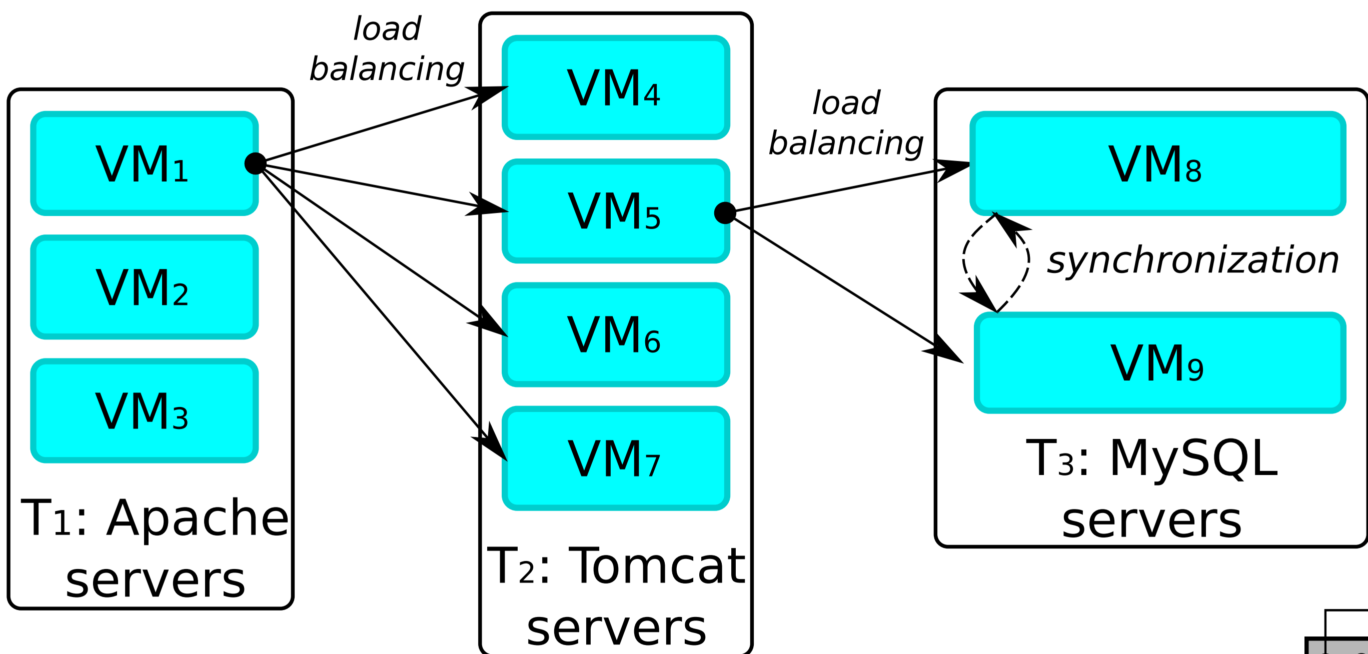


Virtualized HA application



Credits: Hermenier et al., RR-7545 INRIA

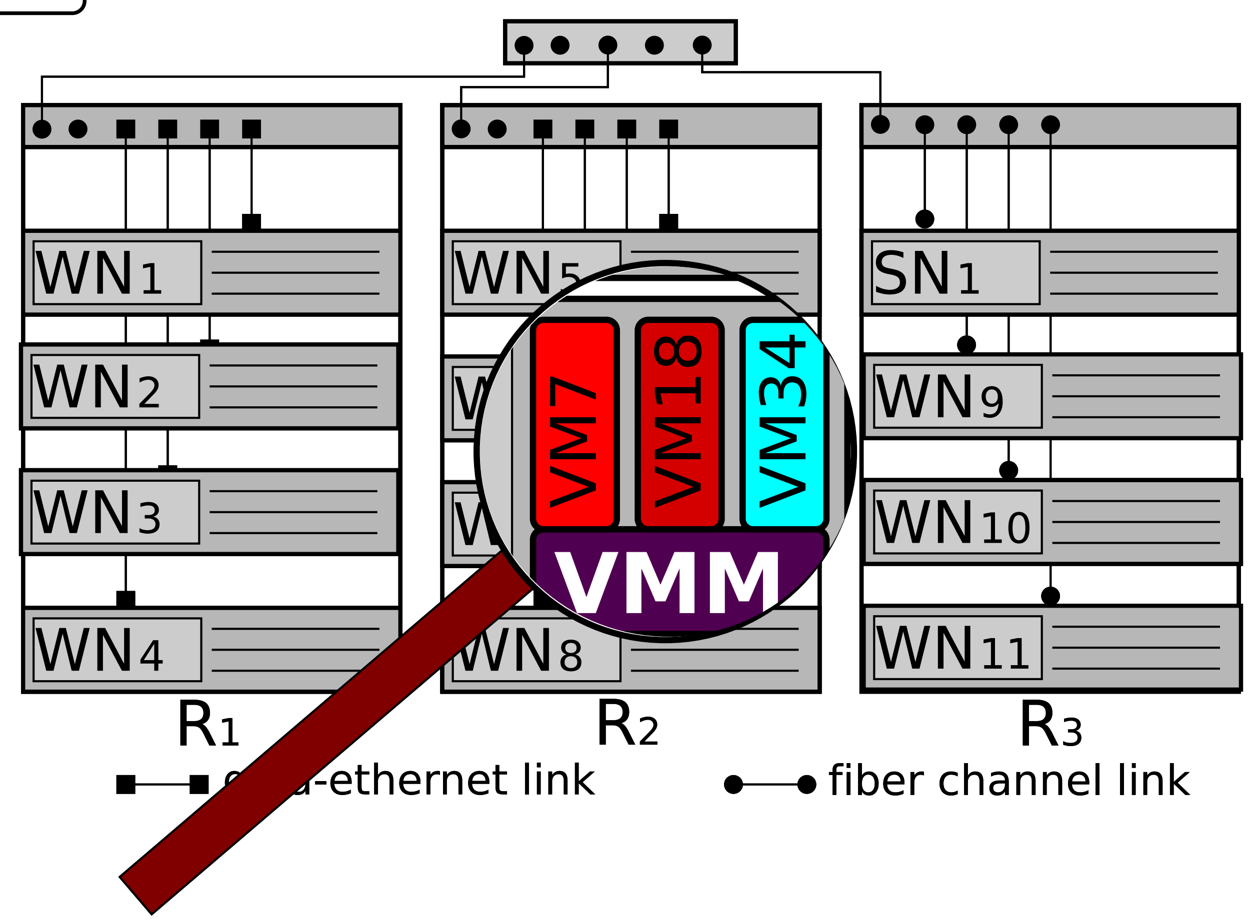
Background - Plasma and Entropy



Virtualized HA application

Plasma, a DSL to describe:

- the infrastructure
- the VEs and their placement constraints



Credits: Hermenier et al., RR-7545 INRIA

Background - Plasma and Entropy

- ban({VM1,VM2}, {N1, N2})

  Prevents a set of VMs from being hosted on a given set of nodes

- fence({VM1,VM2}, {N1, N2})

  Forces a set of VMs to be hosted on a set of nodes

- spread({VM1,VM2})

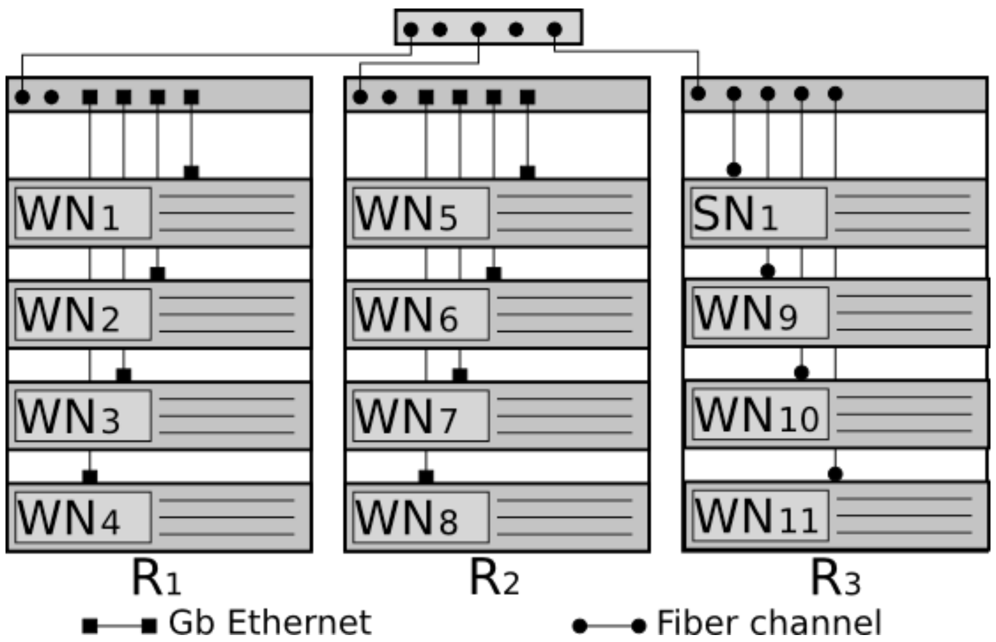
  Ensures that the specified VMs are never hosted on the same node at the same time

- latency({VM1,VM2}, {{N1, N2}, {N3, N4}})

  Forces a set of VMs to be hosted on a single group of nodes

- See more on **http://btrp.inria.fr/**

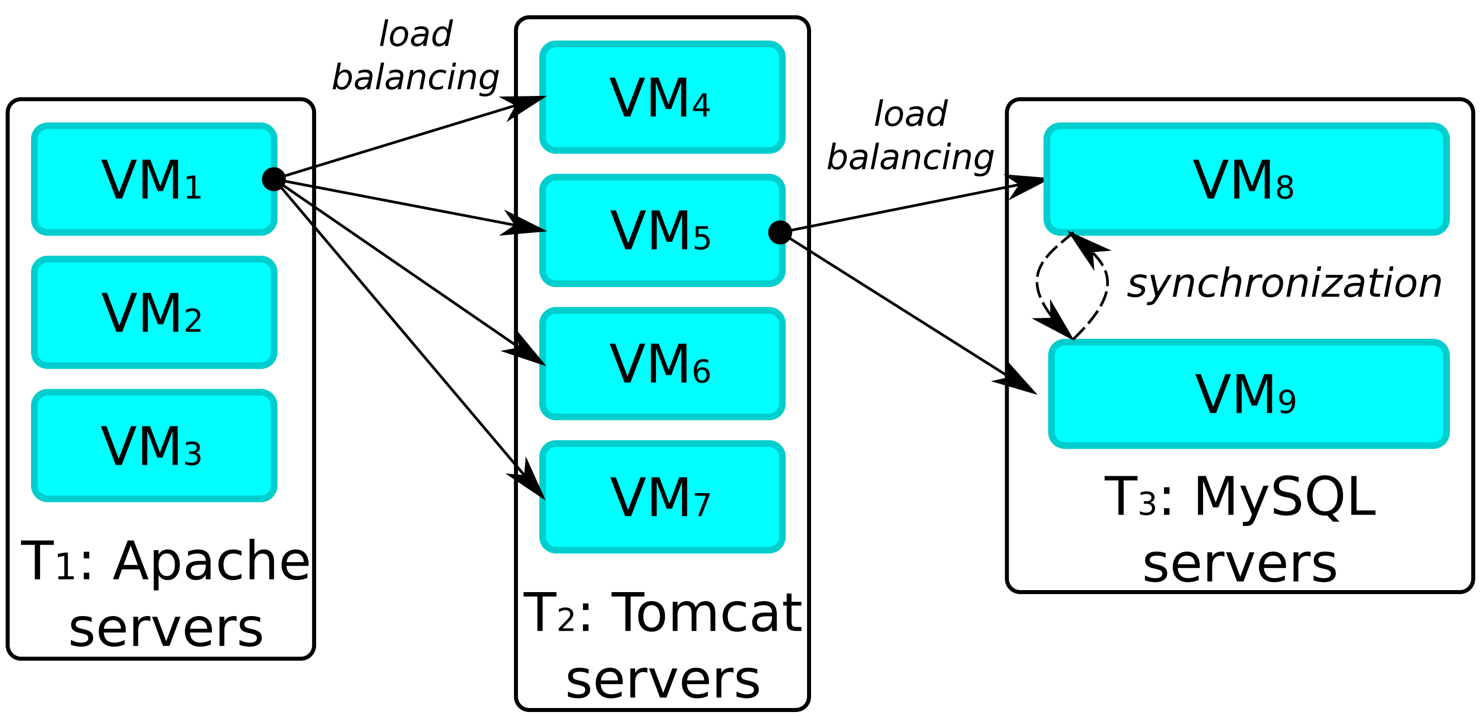
Infrastructure/Application Description




```
// Infrastructure
$R1 = {WN1 ,WN2 ,WN3 ,WN4 };
$R2 = WN [5..8];
$R3 = WN [9..11] + {SN1 };

// Classes of latency
$small = {$R3 };
$medium = $R [1..3];

// Constraints
ban ( $ALL_VMS , {SN1 });
ban ( $ALL_VMS , {WN5 });
fence ($A1 , $R2 + $R3 );
```




```
// The 3 tiers
$T1 = {VM1 ,VM2 ,VM3 };
$T2 = VM [4..7];
$T3 = VM [8..9];

// Fault tolerance to hw. failures
spread($T1);
spread($T2);
spread($T3);

// Efficient synchronization
latency ($T3 , $small );
```