Autonomic and Energy-Efficient Management of Large-Scale Virtualized Data Centers

Eugen Feller

Advisor: Christine Morin
Inria Myriads Project-team, Rennes, France

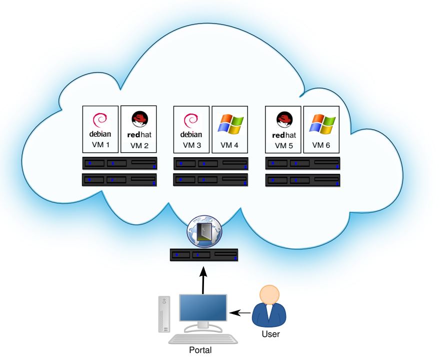
December 17, 2012

Cloud Computing

- On-demand self-service pay-as-you-go resource provisioning
- More and more applications are executed in large data centers

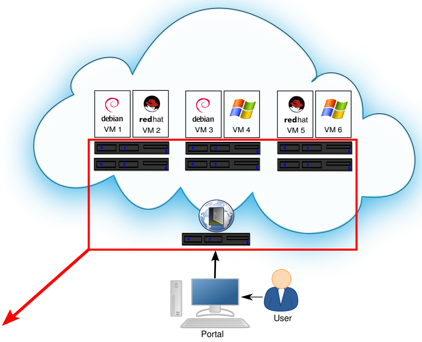
Infrastructure-as-a-Service (IaaS) Clouds

- Provide compute capacity in the form of **Virtual Machines (VMs)**
  - Illusion of a computer running its own operating system
- **Server virtualization**
  - Multiple VMs on a server
  - Live migration



Infrastructure-as-a-Service (IaaS) Clouds

- Provide compute capacity in the form of **Virtual Machines (VMs)**
  - Illusion of a computer running its own operating system
- **Server virtualization**
  - Multiple VMs on a server
  - Live migration

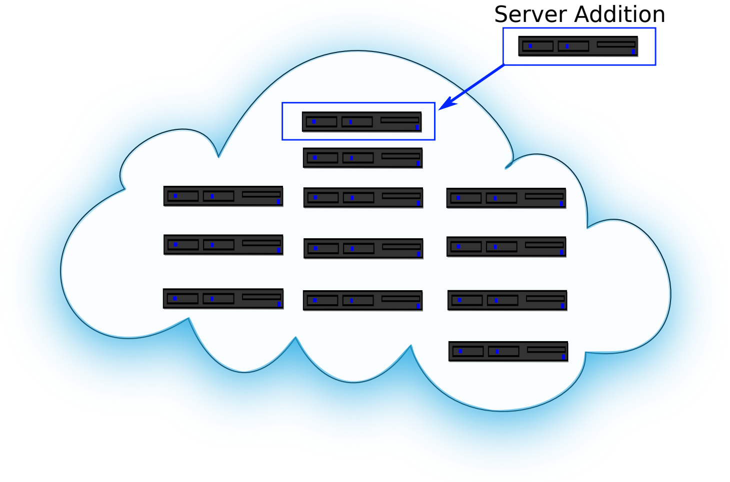


VM management system

- Controls the servers
- Accepts user requests
- Places VMs on the servers

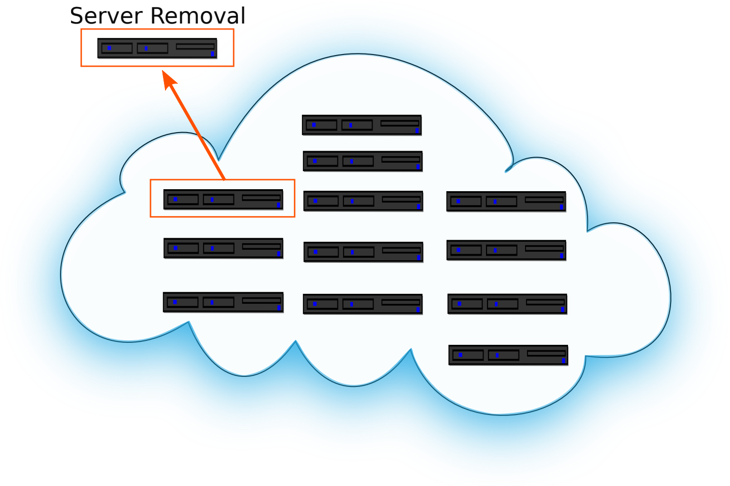
Challenge: Data Center Management

Challenge: Data Center Management



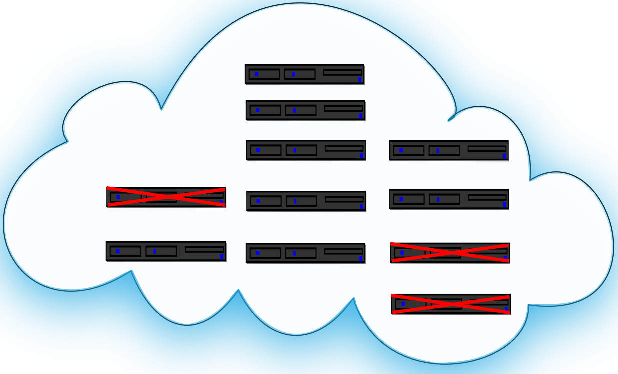
Challenge: Data Center Management

Challenge: Data Center Management



Challenge: Data Center Management

Challenge: Data Center Management



Manual management is impossible

Autonomic IaaS cloud management systems are desirable

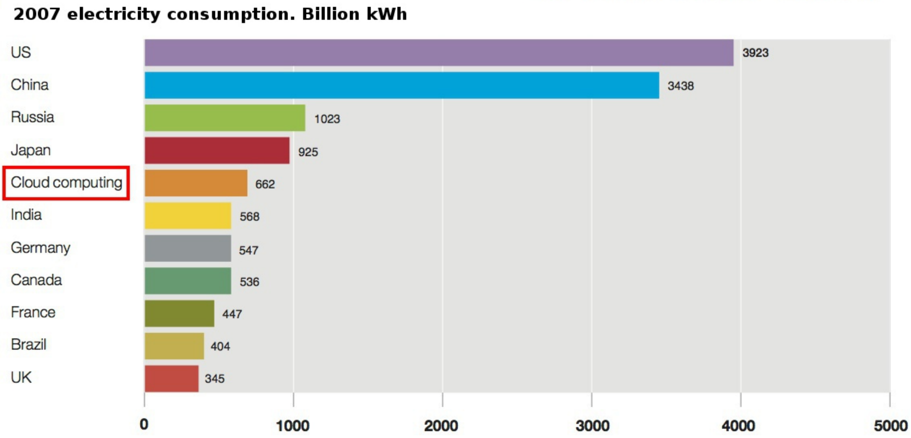
Autonomic System Management

How to achieve autonomic system management in IaaS clouds?

- **Self-configuration**
  - Support for dynamic server addition, removal
- **Self-healing**
  - Support for automated VM management system services fail-over

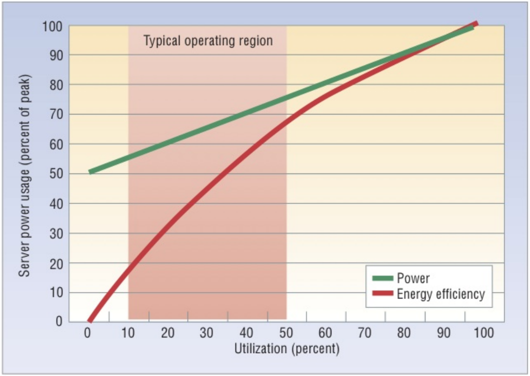
Challenge: Energy Saving

Huge energy amounts in large data centers



Energy Efficiency

- **Data centers are rarely fully utilized**
  - High fluctuating resource demands → Low utilization (10 to 50%)
- **Servers lack power proportionality**
  - High idle power consumption
  - Energy efficiency significantly drops under light loads



Energy Saving Approaches

- **Slow down the individual server components (e.g. CPU, memory)**
  - Becomes less attractive on modern hardware (Le Sueur et al. (2010))

Energy Saving Approaches

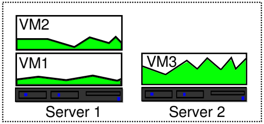
- **Slow down the individual server components (e.g. CPU, memory)**
  - Becomes less attractive on modern hardware (Le Sueur et al. (2010))
- **Transition parts of the server components into a sleep state**
  - Not always easy, as idle time is hard to achieve

Energy Saving Approaches

- **Slow down the individual server components (e.g. CPU, memory)**
  - Becomes less attractive on modern hardware (Le Sueur et al. (2010))
- **Transition parts of the server components into a sleep state**
  - Not always easy, as idle time is hard to achieve
- **Transition entire servers into a sleep state**
  - Entering sleep states can yield significant energy savings

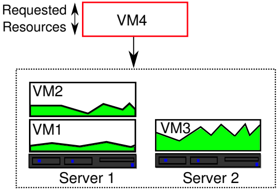
Idle Time Creation

- **Three methods**
  - Energy-efficient VM placement



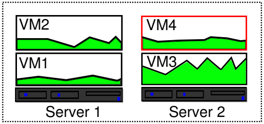
Idle Time Creation

- **Three methods**
  - Energy-efficient VM placement



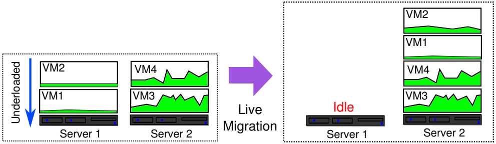
Idle Time Creation

- **Three methods**
  - Energy-efficient VM placement



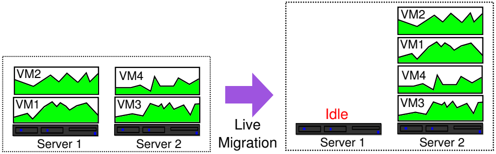
Idle Time Creation

- **Three methods**
  - Energy-efficient VM placement
  - Server underload detection and mitigation



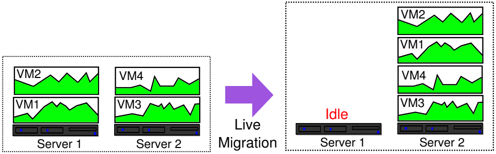
Idle Time Creation

- **Three methods**
  - Energy-efficient VM placement
  - Server underload detection and mitigation
  - Periodic VM consolidation



Idle Time Creation

- **Three methods**
  - Energy-efficient VM placement
  - Server underload detection and mitigation
  - Periodic VM consolidation



Self-optimization for energy efficiency

Objective

Design and implement an autonomic VM management system for large-scale IaaS clouds

- Ease of management
- High availability
- Energy efficiency

Contributions

- **Snooze: autonomic and energy-efficient VM management system**
  - Self-configuring and self-healing VM management system
  - Self-optimizing integrated energy management approach

- **Energy-efficient VM management algorithms**
  - VM placement
  - VM consolidation

Contributions

- **Snooze: autonomic and energy-efficient VM management system**
  - Self-configuring and self-healing VM management system
  - Self-optimizing integrated energy management approach

- **Energy-efficient VM management algorithms**
  - VM placement
  - **VM consolidation**

First Contribution Presented

- **Self-configuring and self-healing VM management system**
- Self-optimizing integrated energy management approach

Existing VM Management Systems

System	Architecture	Self-configuration	Self-healing	Evaluation

Existing VM Management Systems

System	Architecture	Self-configuration	Self-healing	Evaluation
OpenNebula, OpenStack, Nimbus, Entropy	Centralized	No	No	Real system
CloudStack, VMware DRS	Centralized	No	Yes (Replicated servers)	Real system

Existing VM Management Systems

System	Architecture	Self-configuration	Self-healing	Evaluation
OpenNebula, OpenStack, Nimbus, Entropy	Centralized	No	No	Real system
CloudStack, VMware DRS	Centralized	No	Yes (Replicated servers)	Real system
Eucalyptus	Static Hierarchy	No	No	Real system

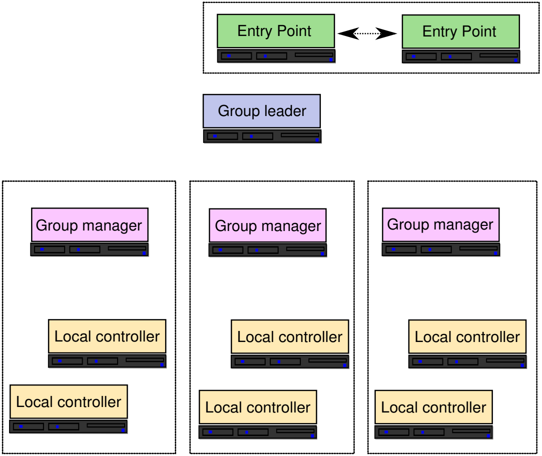
Existing VM Management Systems

System	Architecture	Self-configuration	Self-healing	Evaluation
OpenNebula, OpenStack, Nimbus, Entropy	Centralized	No	No	Real system
CloudStack, VMware DRS	Centralized	No	Yes (Replicated servers)	Real system
Eucalyptus	Static Hierarchy	No	No	Real system
Rouzaud-Cornabas, J. (2010), DVMS, V-MAN	P2P	No	No	Simulator

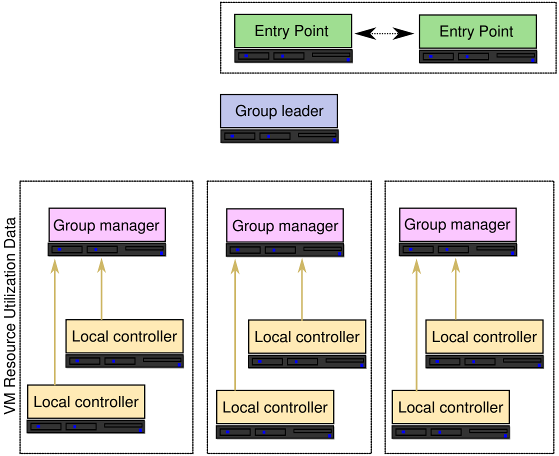
Existing VM Management Systems

System	Architecture	Self-configuration	Self-healing	Evaluation
OpenNebula, OpenStack, Nimbus, Entropy	Centralized	No	No	Real system
CloudStack, VMware DRS	Centralized	No	Yes (Replicated servers)	Real system
Eucalyptus	Static Hierarchy	No	No	Real system
Rouzaud-Cornabas, J. (2010), DVMS, V-MAN	P2P	No	No	Simulator
Snooze	**Dynamic Hierarchy**	**Yes**	**Yes (No dedicated servers)**	**Real system**

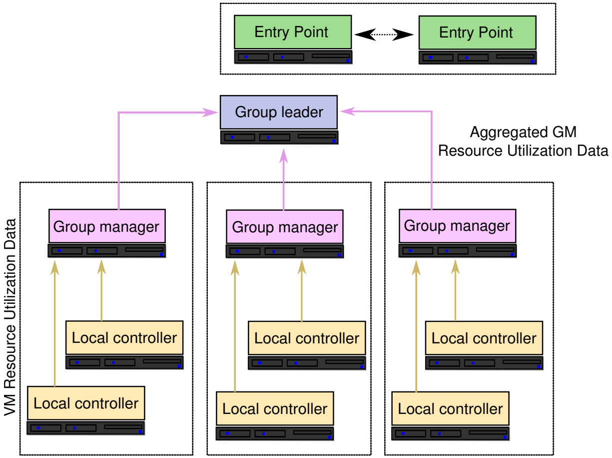
System Architecture



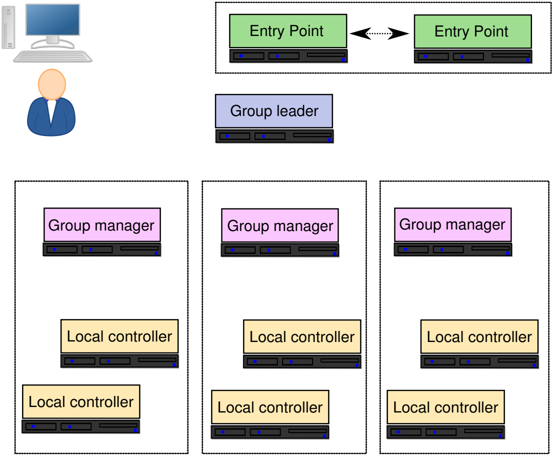
System Architecture



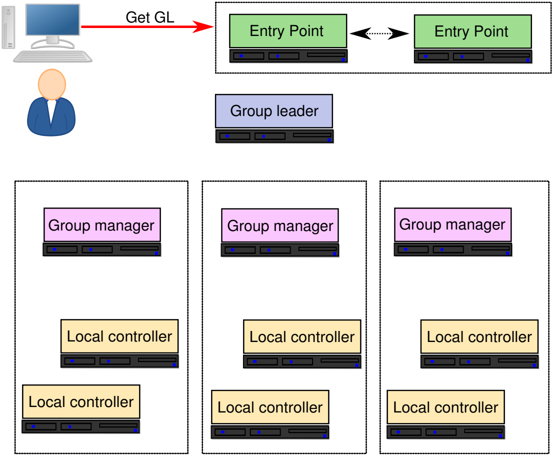
System Architecture



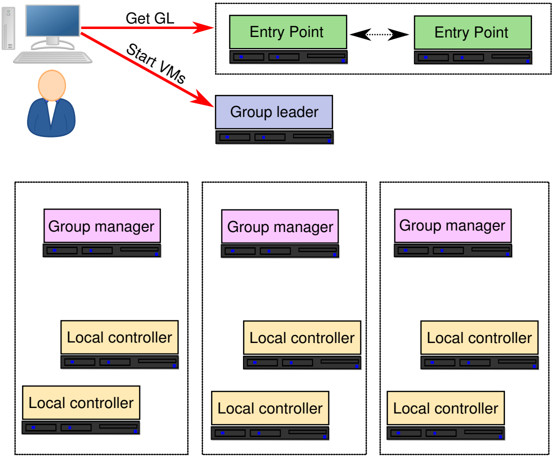
VM Submission Example



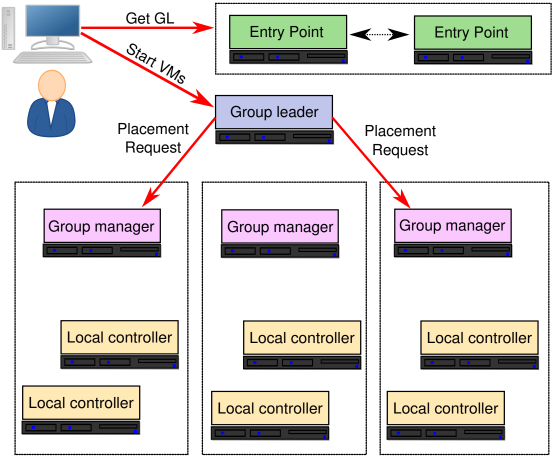
VM Submission Example



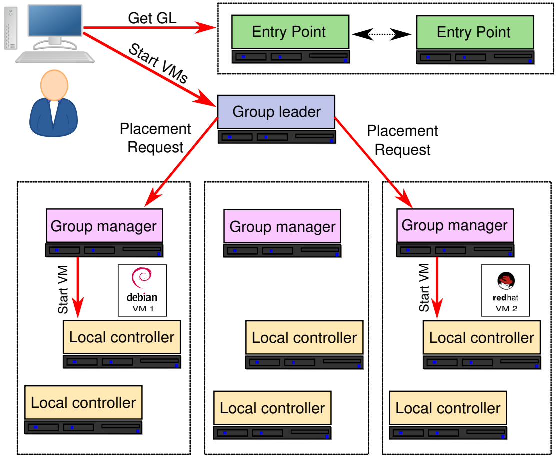
VM Submission Example



VM Submission Example



VM Submission Example



Hierarchy Construction and Maintenance

- How to build the hierarchy?
- How to add/remove servers?
- How to deal with server failures?

Hierarchy Construction and Maintenance

- How to build the hierarchy?
- How to add/remove servers?
- How to deal with server failures?

Self-configuration and self-healing mechanisms

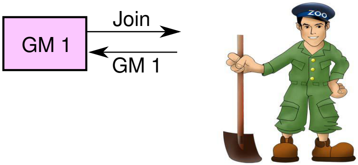
Hierarchy Construction Protocols

- **Three steps**
  - Group leader election
  - Group manager join
  - Local controller join

Group Leader Election

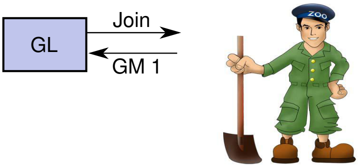
- **Group leader election algorithm exploiting Apache ZooKeeper**
  - Scalable and fault-tolerant coordination framework

- **Group leader election algorithm exploiting Apache ZooKeeper**
  - Scalable and fault-tolerant coordination framework



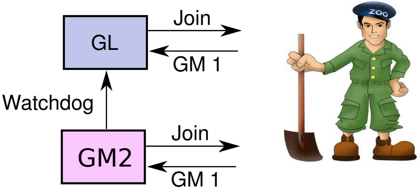
Group Leader Election

- **Group leader election algorithm exploiting Apache ZooKeeper**
  - Scalable and fault-tolerant coordination framework



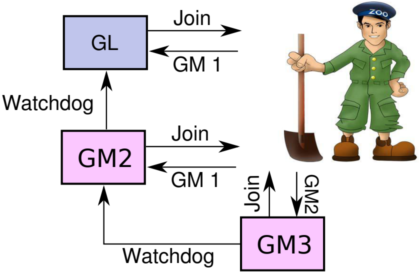
Group Leader Election

- **Group leader election algorithm exploiting Apache ZooKeeper**
  - Scalable and fault-tolerant coordination framework



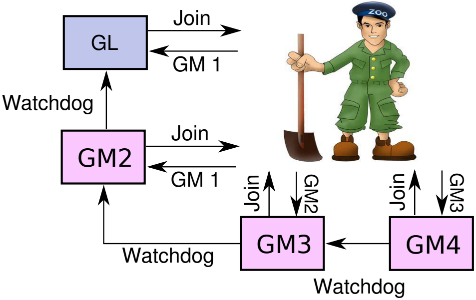
Group Leader Election

- **Group leader election algorithm exploiting Apache ZooKeeper**
  - Scalable and fault-tolerant coordination framework



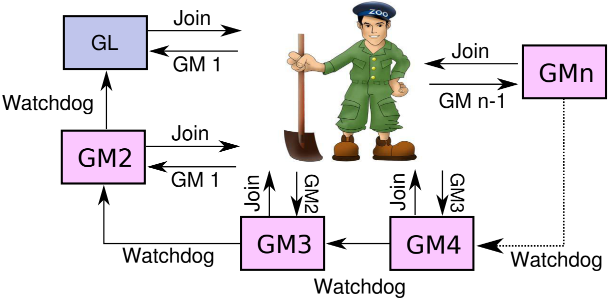
Group Leader Election

- **Group leader election algorithm exploiting Apache ZooKeeper**
  - Scalable and fault-tolerant coordination framework

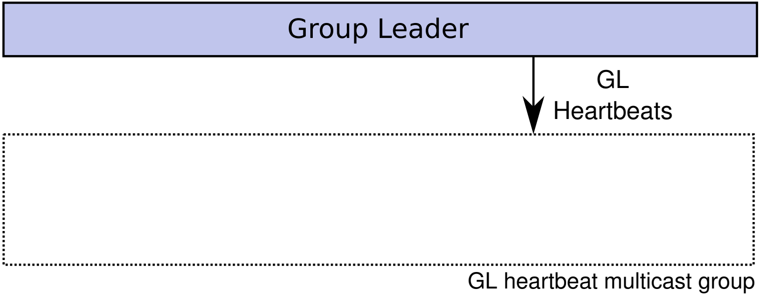


Group Leader Election

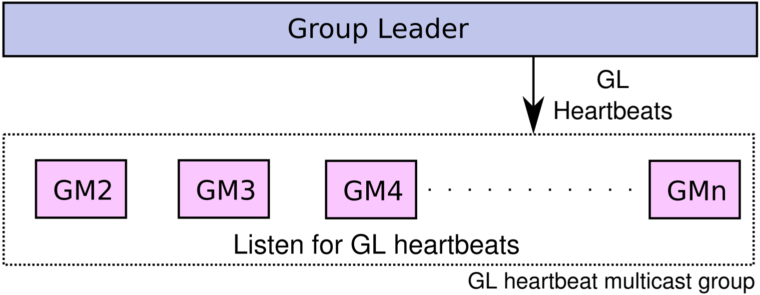
- **Group leader election algorithm exploiting Apache ZooKeeper**
  - Scalable and fault-tolerant coordination framework



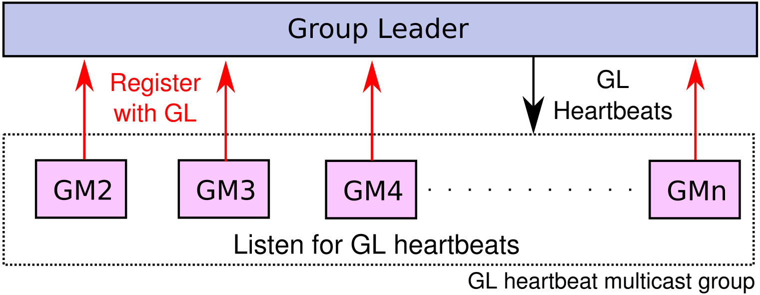
Group Manager Join



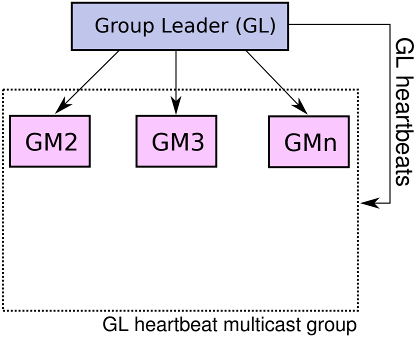
Group Manager Join



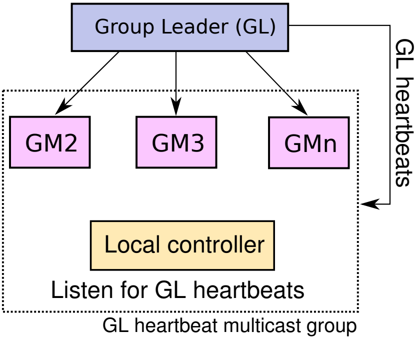
Group Manager Join



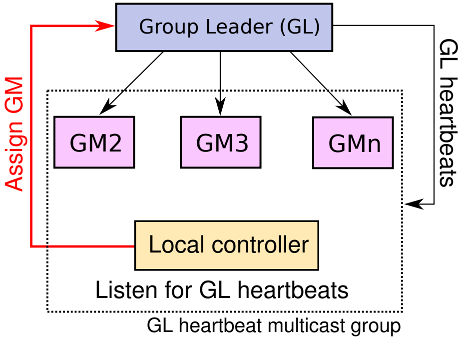
Local Controller Join



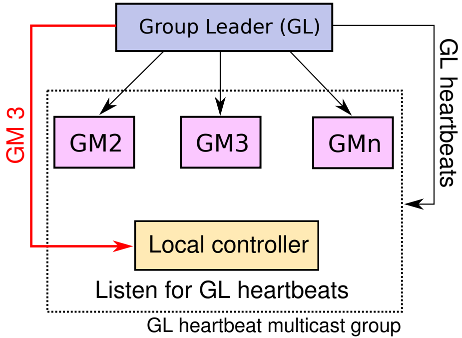
Local Controller Join



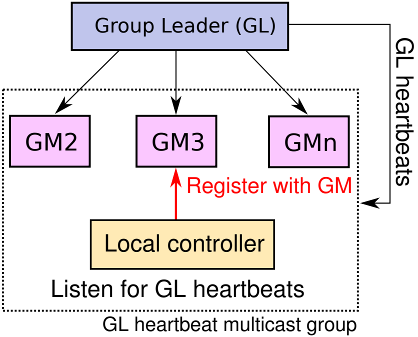
Local Controller Join



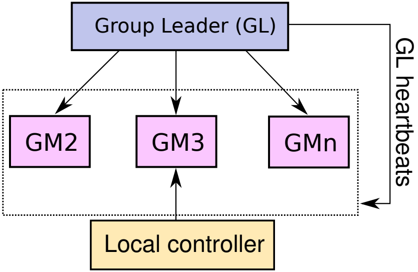
Local Controller Join



Local Controller Join



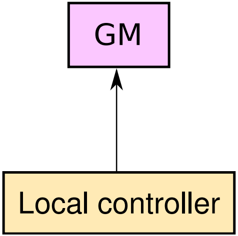
Local Controller Join



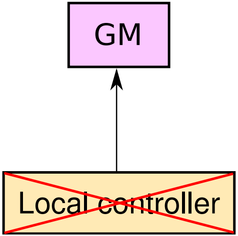
Hierarchy Reconstruction and Maintenance

- **Three kinds of failures**
  - Local controller
  - Group manager
  - Group leader
- **Two steps to tolerate failures**
  1. Error detection
  2. Recovery

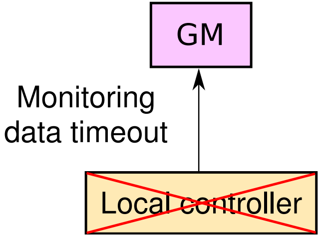
Local Controller Failure Handling



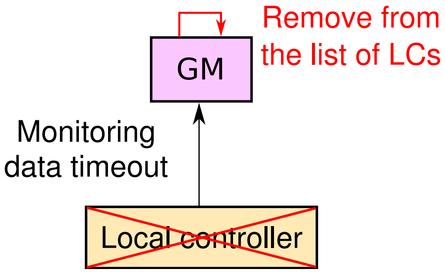
Local Controller Failure Handling



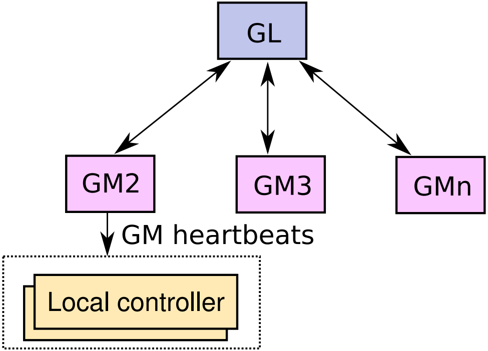
Local Controller Failure Handling



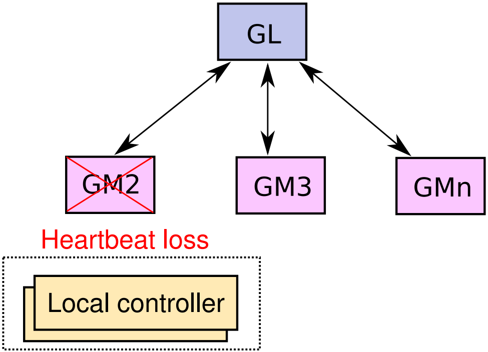
Local Controller Failure Handling



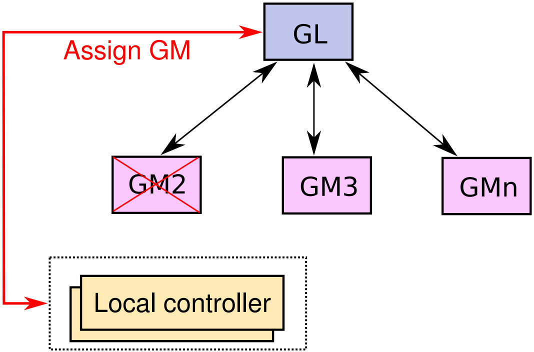
Group Manager Failure Handling



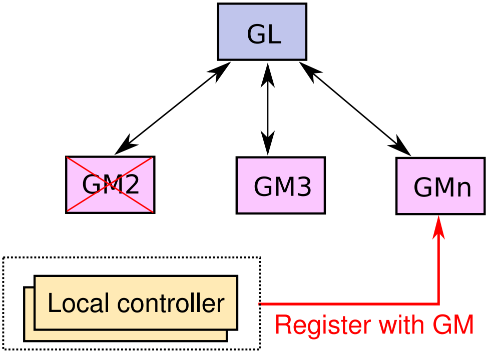
Group Manager Failure Handling



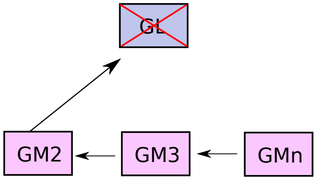
Group Manager Failure Handling



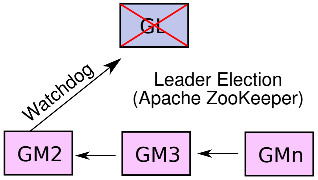
Group Manager Failure Handling



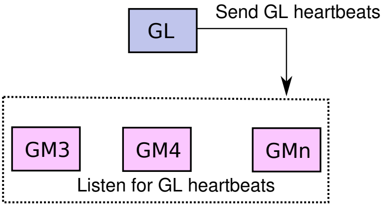
Group Leader Failure Handling



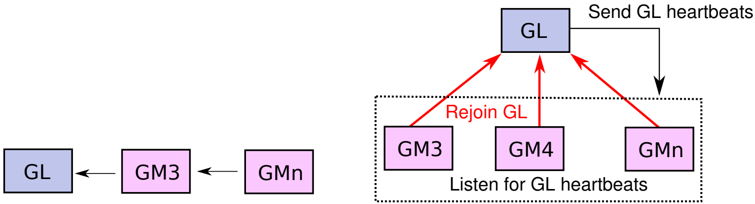
Group Leader Failure Handling



Group Leader Failure Handling



Group Leader Failure Handling



Hierarchy Management Evaluation

- **Scalability and self-healing**
  - Number of LC servers managed by a GM
  - Number of GM servers managed by a GL
  - Cost of the heartbeat mechanisms
  - Cost of the self-healing mechanisms

Prototype implementation deployed on the Grid'5000 testbed

E. Feller, L. Rilling, and C. Morin. Snooze: A Scalable and Autonomic Virtual Machine Management Framework for Private Clouds. In the *12th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid)*, May 2012.

Hierarchy Management Evaluation

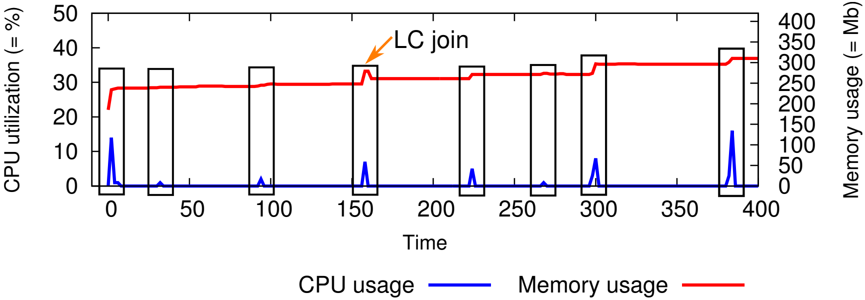
- **Scalability and self-healing**
  - Number of LC servers managed by a GM
  - Number of GM servers managed by a GL
  - Cost of the heartbeat mechanisms
  - Cost of the self-healing mechanisms

Prototype implementation deployed on the Grid'5000 testbed

E. Feller, L. Rilling, and C. Morin. Snooze: A Scalable and Autonomic Virtual Machine Management Framework for Private Clouds. In the *12th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid)*, May 2012.

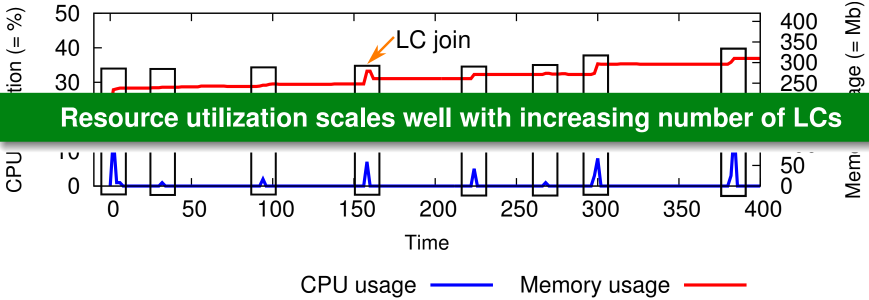
Group Manager Scalability

How does the GM server CPU and memory utilization scale with increasing number of LC servers?



Group Manager Scalability

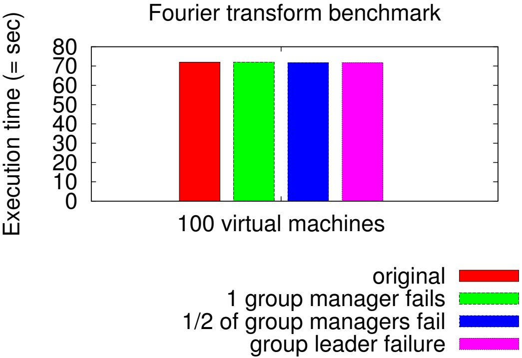
How does the GM server CPU and memory utilization scale with increasing number of LC servers?



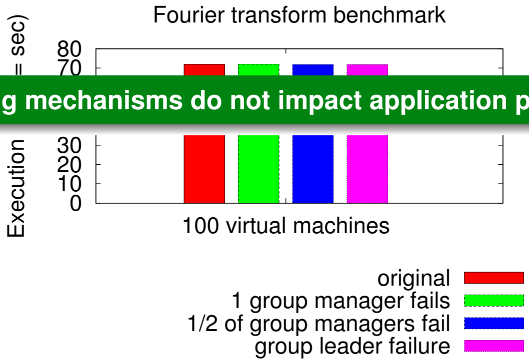
Resource utilization scales well with increasing number of LCs

Cost of the Self-healing Mechanisms

What is the impact of the self-healing mechanisms on the application performance?



What is the impact of the self-healing mechanisms on the application performance?



Self-healing mechanisms do not impact application performance

Outline

- Self-configuring and self-healing VM management system
- **Self-optimizing integrated energy management approach**

Mechanisms and Algorithms for Energy Efficiency

- **How to favour idle times**
  - Energy-efficient VM placement
  - Underload server detection and mitigation
  - Periodic VM consolidation
- **Server overload detection and mitigation**
- **Power management**
  - Automatic detection and power cycling of idle servers
  - Server wakeup when not enough resources are available

Existing Energy Management Approaches in IaaS Clouds

Approach	Algorithm	Resources	Placement	Underload, Overload Mitigation	VM Consolidation	Power management	Evaluation

Existing Energy Management Approaches in IaaS Clouds

Approach	Algorithm	Resources	Placement	Underload, Overload Mitigation	VM Consolidation	Power management	Evaluation
Entropy	Constraint programming	CPU, RAM	Yes (Consolidation)	Overload (Consolidation)	Yes	Server off/on	Real system and simulations

Existing Energy Management Approaches in IaaS Clouds

Approach	Algorithm	Resources	Placement	Underload, Overload Mitigation	VM Consolidation	Power management	Evaluation
Entropy	Constraint programming	CPU, RAM	Yes (Consolidation)	Overload (Consolidation)	Yes	Server off/on	Real system and simulations
Sandpiper	Greedy	CPU, RAM, Network	No	Overload	No	No	Real system
V-MAN	Greedy	Number of VMs	No	No	Yes	No	Simulation
Sercon	Greedy	CPU, RAM	No	No	Yes	No	Simulation
Borgetto et al. (2012)	Greedy	CPU, RAM	Yes	Underload, Overload	Yes	Server off/on	Simulation
VMware DRM	Greedy (Proprietary)	CPU, RAM	Yes	Underload, Overload	No	Server off/on	Real system

Existing Energy Management Approaches in IaaS Clouds

Approach	Algorithm	Resources	Placement	Underload, Overload Mitigation	VM Consolidation	Power management	Evaluation
Entropy	Constraint programming	CPU, RAM	Yes (Consolidation)	Overload (Consolidation)	Yes	Server off/on	Real system and simulations
Sandpiper	Greedy	CPU, RAM, Network	No	Overload	No	No	Real system
V-MAN	Greedy	Number of VMs	No	No	Yes	No	Simulation
Sercon	Greedy	CPU, RAM	No	No	Yes	No	Simulation
Borgetto et al. (2012)	Greedy	CPU, RAM	Yes	Underload, Overload	Yes	Server off/on	Simulation
VMware DRM	Greedy (Proprietary)	CPU, RAM	Yes	Underload, Overload	No	Server off/on	Real system
Snooze	**Greedy (extensible)**	**CPU, RAM, Network**	**Yes**	**Underload, Overload**	**Yes (modified Sercon)**	**Server off/on**	**Real system**

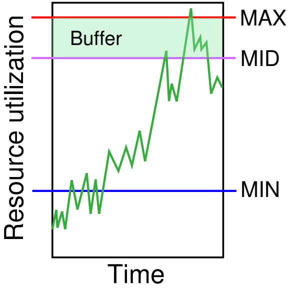
Underload and Overload Mechanism

- **How to deal with underload and overload situations?**
  - Detection of server underload/overload situations
  - Relocation of VMs from underloaded/overloaded servers

Underload and Overload Detection Approach

Local controllers periodically estimate their resource utilization based on locally aggregated VM resource utilization data

- Multi-dimensional
    - CPU
    - RAM
    - Network Rx
    - Network Tx



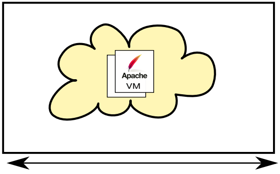
Underload Relocation Algorithm

Triggered by the GM in the event of server underload

- Key ideas
  - Move VMs from underloaded LC to LCs with enough spare capacity
  - **All-or-nothing approach:** Either migrate all VMs or none
- Description
  - Sort VMs from underloaded LC in decreasing order of estimated utilization
  - Sort destination LCs in decreasing order of estimated utilization
  - Attempt to assign the VMs to the destination LCs starting from the first one
  - If some VM could not be assigned abort the algorithm
  - . . . otherwise perform live migrations

Integrated Energy Management Evaluation

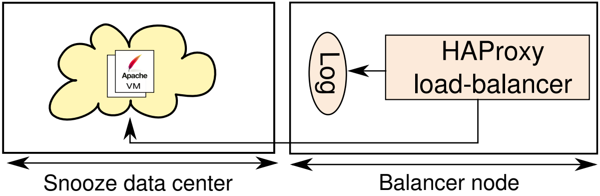
Evaluation with an elastic web service



Deployed on 34 power-metered servers of the Grid'5000 testbed

E. Feller, C. Rohr, D. Margery, and C. Morin. Energy Management in IaaS Clouds: A Holistic Approach. In the *5th IEEE International Conference on Cloud Computing (CLOUD)*, May 2012.

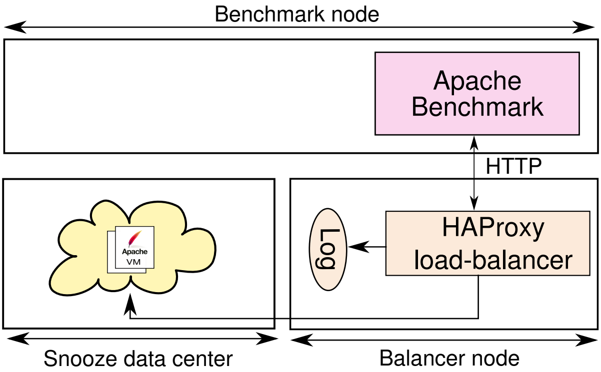
Evaluation with an elastic web service



Deployed on 34 power-metered servers of the Grid'5000 testbed

E. Feller, C. Rohr, D. Margery, and C. Morin. Energy Management in IaaS Clouds: A Holistic Approach. In the *5th IEEE International Conference on Cloud Computing (CLOUD)*, May 2012.

Evaluation with an elastic web service

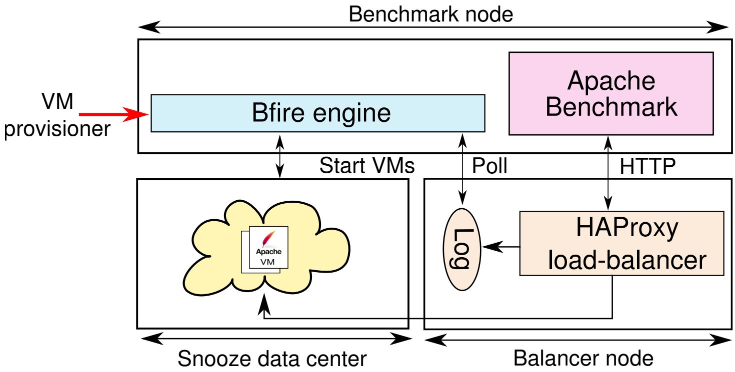


Deployed on 34 power-metered servers of the Grid'5000 testbed

E. Feller, C. Rohr, D. Margery, and C. Morin. Energy Management in IaaS Clouds: A Holistic Approach. In the *5th IEEE International Conference on Cloud Computing (CLOUD)*, May 2012.

Integrated Energy Management Evaluation

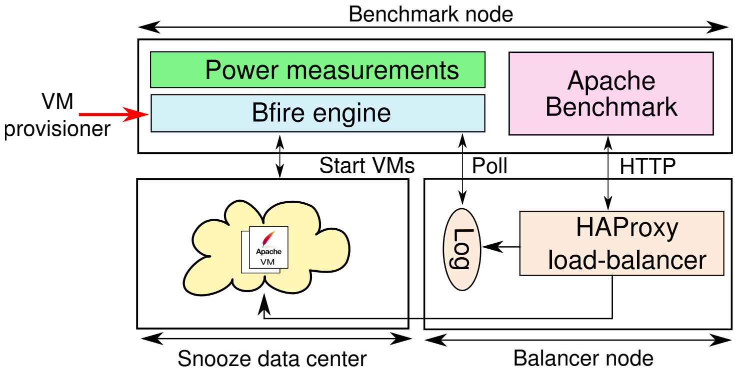
Evaluation with an elastic web service



Deployed on 34 power-metered servers of the Grid'5000 testbed

E. Feller, C. Rohr, D. Margery, and C. Morin. Energy Management in IaaS Clouds: A Holistic Approach. In the *5th IEEE International Conference on Cloud Computing (CLOUD)*, May 2012.

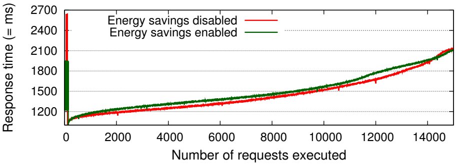
Evaluation with an elastic web service



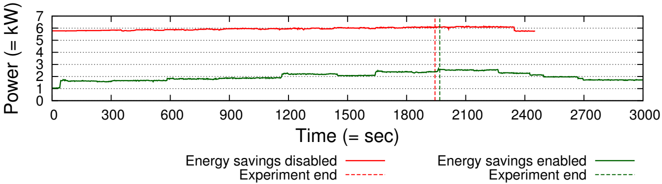
Deployed on 34 power-metered servers of the Grid'5000 testbed

E. Feller, C. Rohr, D. Margery, and C. Morin. Energy Management in IaaS Clouds: A Holistic Approach. In the *5th IEEE International Conference on Cloud Computing (CLOUD)*, May 2012.

- **Apache Benchmark Performance**



- **Data Center Power Consumption**

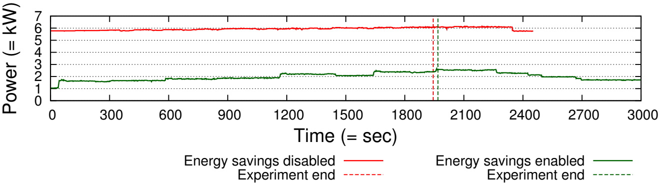


Energy Saving Evaluation

- **Apache Benchmark Performance**

Limited performance degradation
Up to 67% energy savings for the evaluated application

- **Data Center Power Consumption**



First Contribution Summary

- Self-configuring and healing hierarchical architecture
- Integrated energy management approach
  - VM placement and consolidation, server underload/overload mitigation, power management
  - Four-dimensional aggregation-based underload/overload mitigation
  - First implementation of the Sercon algorithm in a real system
- A robust prototype
- Experimentally validated on the Grid'5000 testbed

Virtual machine consolidation

Existing VM Consolidation Algorithms

Approach	Algorithms	Worst-case Complexity	Solution	Parallelization
Greedy	Sercon	Polynomial	Close to optimal	No

Existing VM Consolidation Algorithms

Approach	Algorithms	Worst-case Complexity	Solution	Parallelization
Greedy	Sercon	Polynomial	Close to optimal	No
Mathematical programming	Constraint programming	Exponential	Optimal	Yes

Existing VM Consolidation Algorithms

Approach	**Algorithms**	**Worst-case Complexity**	**Solution**	**Parallelization**
Greedy	Sercon	Polynomial	Close to optimal	No
Mathematical programming	Constraint programming	Exponential	Optimal	Yes
Metaheuristics	Genetic algorithms, **Ant Colony Optimization**	**Polynomial**	**Close to optimal**	**Yes**

Existing VM Consolidation Algorithms

Approach	Algorithms	Worst-case Complexity	Solution	Parallelization
Greedy	Sercon	Polynomial	Close to optimal	No
Mathematical programming	Constraint programming	Exponential	Optimal	Yes
Metaheuristics	Genetic algorithms, **Ant Colony Optimization**	**Polynomial**	**Close to optimal**	**Yes**

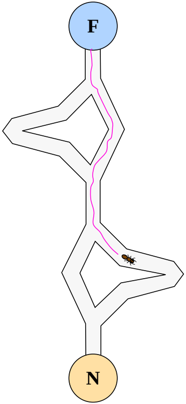
First attempt to apply Ant Colony Optimization on VM consolidation

Ant Colony Optimization

- Ants work independently
- Indirect communication using pheromone in the environment
- Decisions are taken probabilistically

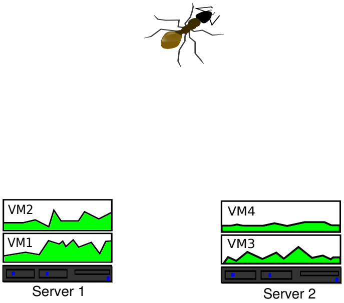
Ant Colony Optimization

- Ants work independently
- Indirect communication using pheromone in the environment
- Decisions are taken probabilistically



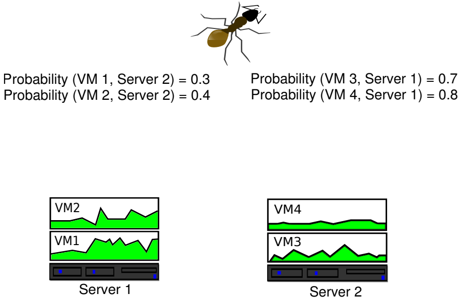
Our Mapping from Paths to VMs and Servers

- **Design principles**
  - Ants compute solutions concurrently
  - Best solution is preserved
  - Pheromone on VM-server pairs
  - Probabilistic pair choice



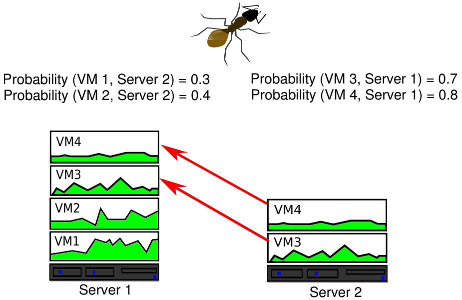
Our Mapping from Paths to VMs and Servers

- **Design principles**
  - Ants compute solutions concurrently
  - Best solution is preserved
  - Pheromone on VM-server pairs
  - Probabilistic pair choice



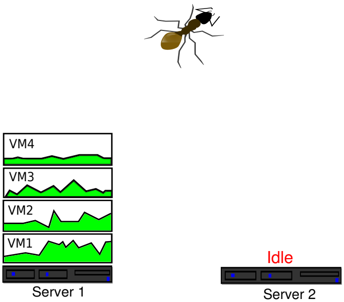
Our Mapping from Paths to VMs and Servers

- **Design principles**
  - Ants compute solutions concurrently
  - Best solution is preserved
  - Pheromone on VM-server pairs
  - Probabilistic pair choice



Our Mapping from Paths to VMs and Servers

- **Design principles**
  - Ants compute solutions concurrently
  - Best solution is preserved
  - Pheromone on VM-server pairs
  - Probabilistic pair choice



- **Algorithm components**
  - Objective function
  - Probabilistic pair selection rule
  - Pair pheromone update rule

VM Consolidation Scalability Issues

- **VM consolidation by nature is not scalable**
  - Computing optimal solutions is exponential in time and space
  - Solution quality degrades at scale

VM Consolidation Scalability Issues

- **VM consolidation by nature is not scalable**
  - Computing optimal solutions is exponential in time and space
  - Solution quality degrades at scale
- **Desirable properties**
  - Scalability with increasing number of servers and VMs
  - High packing efficiency (PE)

$$PE := \frac{\text{Number of released servers}}{\text{Total number of servers}} \times 100$$

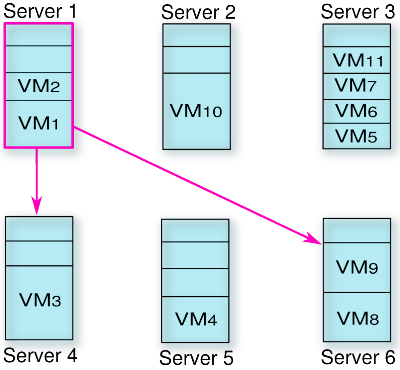
  - Minimize the number of migrations

Fully Decentralized VM Consolidation System

- Servers maintain only a **partial system view**
- **VM consolidation is applied within these partial views**
- Partial views are modified **periodically and randomly**

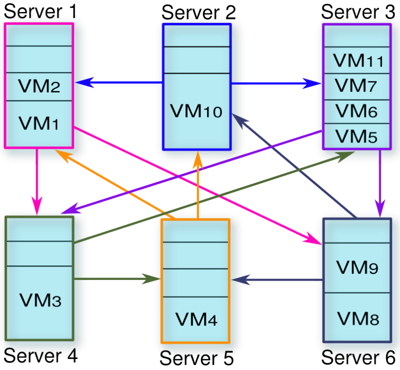
Fully Decentralized VM Consolidation System

- Servers maintain only a **partial system view**
- **VM consolidation is applied within these partial views**
- Partial views are modified **periodically and randomly**



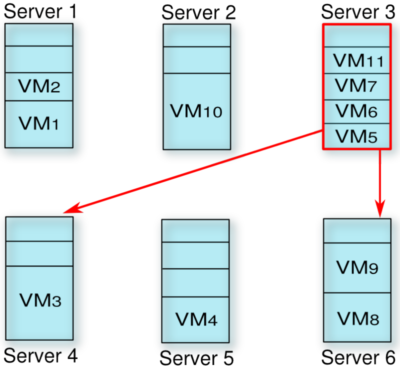
Fully Decentralized VM Consolidation System

- Servers maintain only a **partial system view**
- **VM consolidation is applied within these partial views**
- Partial views are modified **periodically and randomly**



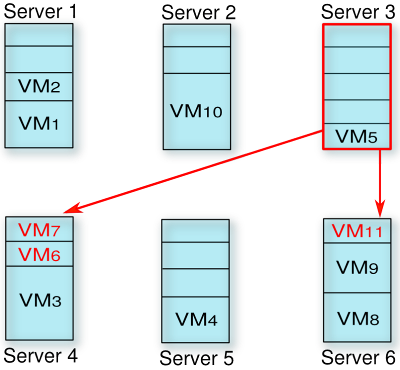
Fully Decentralized VM Consolidation System

- Servers maintain only a **partial system view**
- **VM consolidation is applied within these partial views**
- Partial views are modified **periodically and randomly**



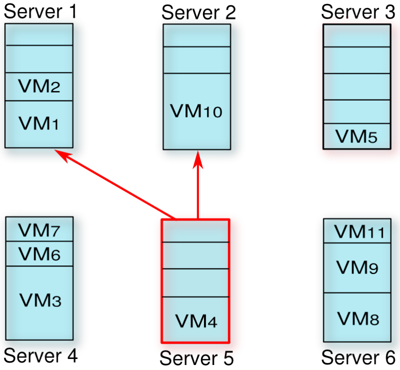
Fully Decentralized VM Consolidation System

- Servers maintain only a **partial system view**
- **VM consolidation is applied within these partial views**
- Partial views are modified **periodically and randomly**



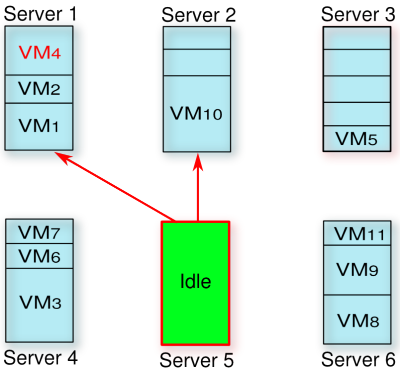
Fully Decentralized VM Consolidation System

- Servers maintain only a **partial system view**
- **VM consolidation is applied within these partial views**
- Partial views are modified **periodically and randomly**



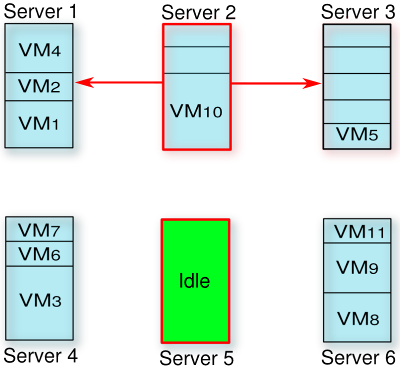
Fully Decentralized VM Consolidation System

- Servers maintain only a **partial system view**
- **VM consolidation is applied within these partial views**
- Partial views are modified **periodically and randomly**



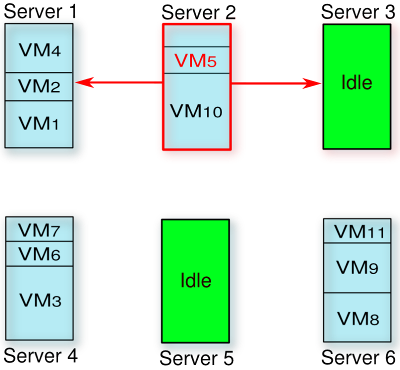
Fully Decentralized VM Consolidation System

- Servers maintain only a **partial system view**
- **VM consolidation is applied within these partial views**
- Partial views are modified **periodically and randomly**



Fully Decentralized VM Consolidation System

- Servers maintain only a **partial system view**
- **VM consolidation is applied within these partial views**
- Partial views are modified **periodically and randomly**



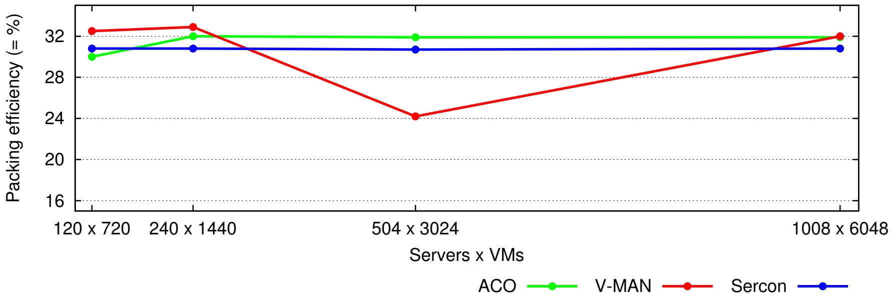
Fully Decentralized VM Consolidation System Evaluation

- **Criteria**
  - Scalability
  - Packing efficiency
  - Number of migrations
- **Experiments**
  - Comparison of different VM consolidation algorithms
    - Sercon
    - V-MAN
    - Our ACO-based VM consolidation algorithm
  - Comparison with a centralized system
- **Evaluated by emulation**

E. Feller, C. Morin, and A. Esnault. A Case for Fully Decentralized Dynamic VM Consolidation in Clouds. In the *4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom) (Best Paper Finalist)*, December 2012.

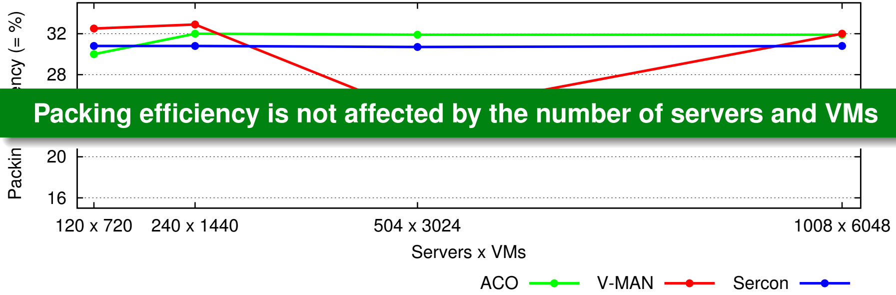
Scalability

How does the system scale in terms of its packing efficiency with increasing number of servers and VMs?



Scalability

How does the system scale in terms of its packing efficiency with increasing number of servers and VMs?



Packing efficiency is not affected by the number of servers and VMs

Comparison With a Centralized System Topology

What is the packing efficiency and number of migrations compared to a centralized system?

Topology	Algorithm	Migrations	Packing Efficiency (%)
Centralized	**Sercon**	**1920**	**31.7**
	ACO	Failed	Failed
P2P	V-MAN	4189	32.0
	ACO	**4015**	**31.9**
	Sercon	1872	30.8

Experiments with 1008 servers and 6048 VMs

Comparison With a Centralized System Topology

What is the packing efficiency and number of migrations compared to a centralized system?

Topology	Algorithm	Migrations	Packing Efficiency (%)
Centralized	**Sercon**	**1920**	**31.7**
	ACO	Failed	Failed
	ACO	**4015**	**31.9**
	Sercon	1872	30.8

Packing efficiency and number of migrations close to a centralized system

Experiments with 1008 servers and 6048 VMs

Second Contribution Summary

- ACO-based VM consolidation algorithm
- Fully decentralized VM consolidation system
- Validated on the Grid'5000 experimentation testbed
  - Scalable with increasing numbers of servers and VMs
  - Packing efficiency close to a centralized system

Criteria	Best algorithm	2nd	3rd
#Migrations	Sercon	ACO	V-MAN
Packing efficiency	V-MAN	ACO	Sercon

Conclusion

- **Snooze: autonomic and energy-efficient VM management system for large-scale IaaS clouds**
  - Self-configuring and healing hierarchical architecture
  - Platform to evaluate VM management algorithms in a real system
  - Open-source software (http://snooze.inria.fr)
    - External users: IRIT Toulouse, EDF R&D, LIFL, LBNL, and Medion Seattle
    - Support: Inria technological action
- **Algorithms for energy efficiency**
  - Evaluation of an integrated approach
    - First implementation of Sercon consolidation algorithm in a real system
    - Novel approach for underload/overload management
    - Up to 64% of energy savings
  - First ACO-based placement and consolidation algorithms
    - Viable approach in a fully decentralized system

Short-term Perspectives

- Further evaluate the Snooze system
  - Larger-scale experiments
  - Real-world workloads
  - Hierarchy energy overheads
- Exploit Snooze to experimentally compare state of the art VM management algorithms
- Further increase the Snooze hierarchy autonomy and energy-efficiency
  - Re-balance the hierarchy dynamically
  - Remove local controller/group manager distinction
  - Power-cycle idle GMs

Long-term Perspectives

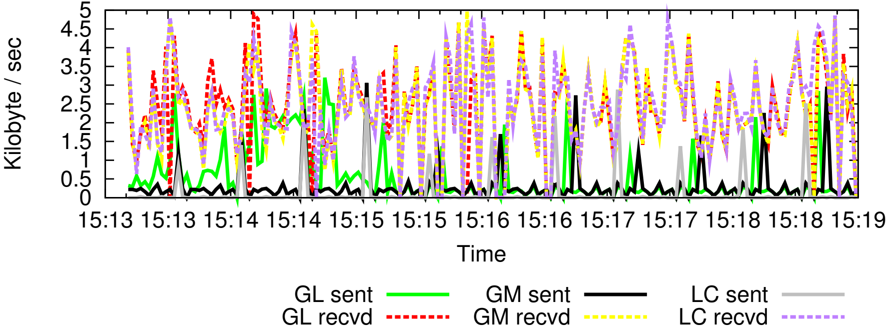
- Metrics for better capturing aggregated resource utilization data
- Improving consolidation
  - Co-location and anti-colocation constraints
  - Consider VM resource demand complementarities
  - Data center network topology aware consolidation
  - Consolidation interval predictions
- Thermal management

Thank you for your attention!

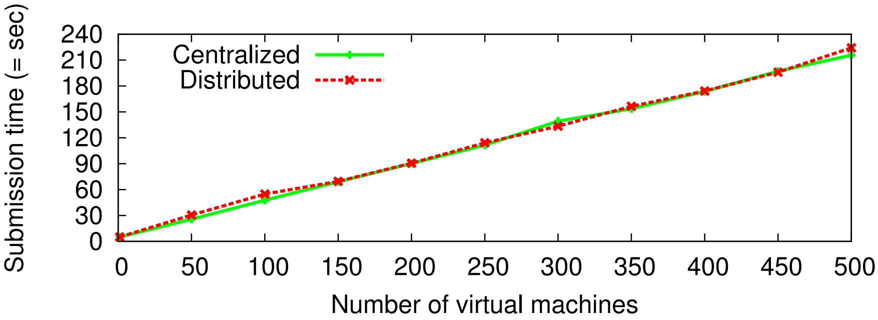
Discussion

Backup slides

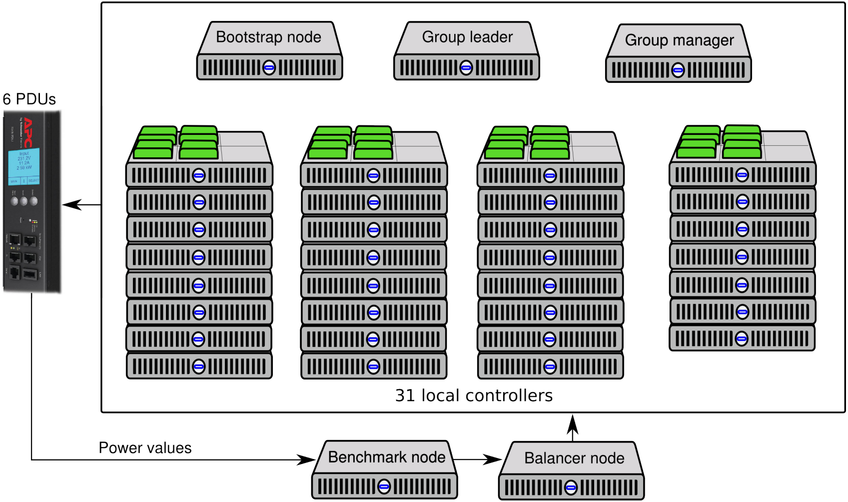
Snooze Heartbeat Overhead



Snooze Submission Time



Energy Management Data Center



Energy Management Parameters

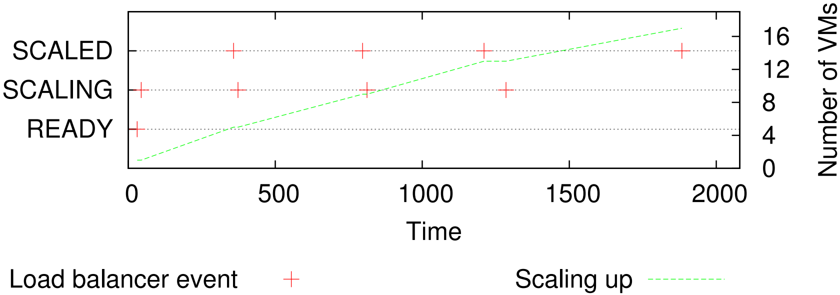
Resource	MIN, MID, MAX
CPU,	0.2, 0.9, 1
Memory	0.2, 0.9, 1
Network	0.2, 0.9, 1

Policy	Algorithm
Dispatching	RoundRobin
Placement	FirstFit
Overload	Greedy
Underload	Greedy
Consolidation	Sercon

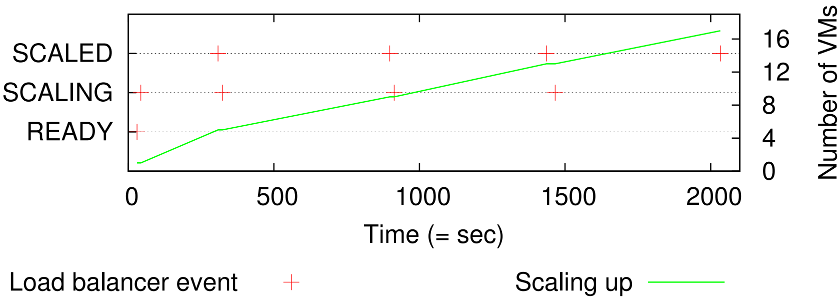
Parameter	Value
Packing density	0.9
Monitoring backlog	15
Resource estimators	average
Consolidation interval	10 min

Parameter	Value
Idle time threshold	2 min
Wakeup threshold	3 min
Power saving action	shutdown
Shutdown driver	system
Wakeup driver	IPMI

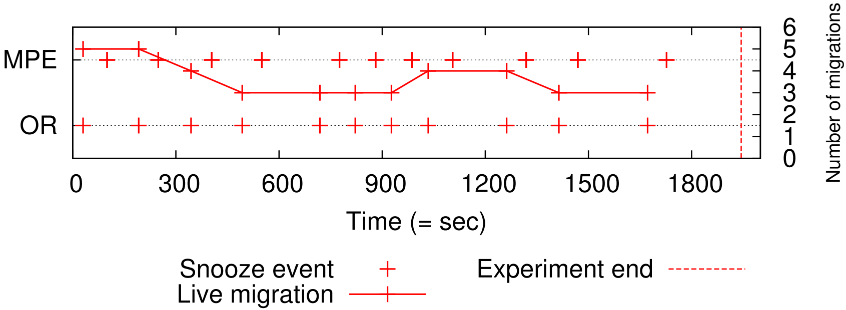
Bfire Events With Energy Savings Disabled



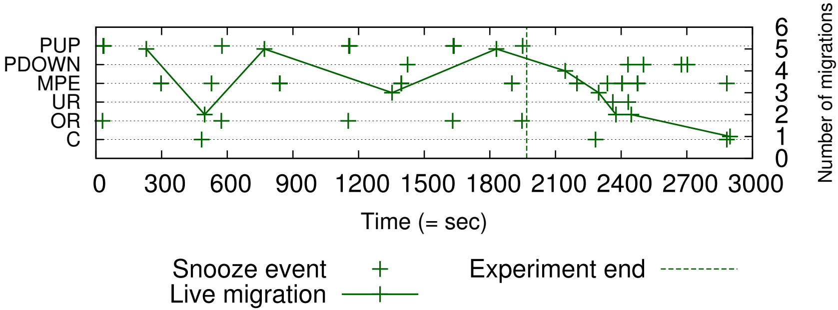
Bfire Events With Energy Savings Enabled



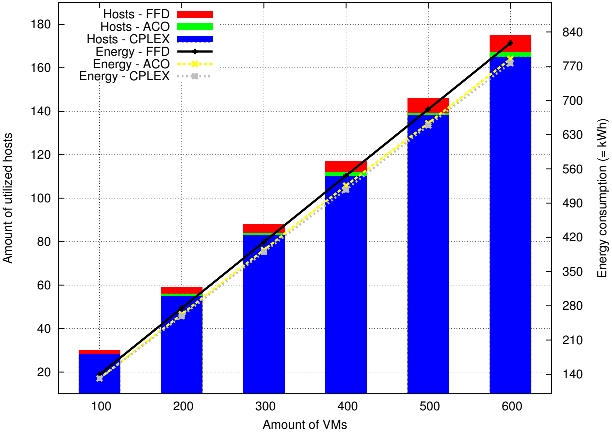
Snooze Events No Energy Savings Disabled



Snooze Events With Energy Savings Enabled



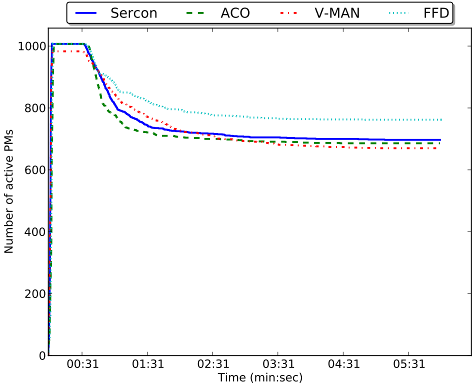
VM Placement Evaluation



Fully Decentralized VM Consolidation - Emulator Parameters

Parameter	Value
Number of PMs and VMs	1008 (resp. 6048)
Experiment duration	360s
Consolidation interval	30s
Shuffling interval	10s
Neighbourhood size	16 PMs
Considered resources	CPU, memory and network
PM total capacity vector	(48, 26, 20)
VM requested capacity vectors	(0.2, 0.5, 0.1), (1, 1, 1), (2, 1, 1), (4, 2, 2), (8, 4, 4), (16, 8, 4)

Fully Decentralized VM Consolidation - Number of Active Servers



Fully Decentralized VM Consolidation - Number of Migrations