International Advanced Research Workshop
on High Performance Computing, Grids and Clouds
27th June 2011, Cetraro, Italy

A Multi-Cloud Management Architecture and Early Experiences

Jose Luis Lucas Simarro

Rafael Moreno-Vozmediano, Ruben S. Montero and Ignacio M. Llorente

dsa-research.org

Distributed Systems Architecture Research Group
Universidad Complutense de Madrid

Acknowledgments

The research leading to these results has received funding from the Ministerio de Ciencia e Innovación of Spain through research grant TIN2009-07146.

dsa-research.org

Contents

Introduction

A Multi-cloud Management Architecture and Early Experiences

- The Cloud computing market is growing. Different cloud providers exhibit their own features, which have some similarities but also differences between them.
  - Pricing models: pay-as-you-go or long time reservation models. But different prices or reservation periods.
  - Interfaces: Create, monitor, terminate VMs. But proprietary interfaces.
  - or Image types and its performance.

  That's very much information for cloud users.

- The use of federated clouds and multi-cloud deployments is also increasing:
  - Integrate resources for different providers.
  - Increase reliability and scalability.
  - Reduce cost.

Introduction

- Cloud broker.

  Useful tool to deal with so many cloud providers.

- Benefits:
  - Usability: Uniform interface to manage user infrastructure.
  - Optimization: Scheduling policies and algorithms to optimize service parameters (infrastructure cost, service performance, etc.).
  - Advanced Service Management: Auto-scaling service, etc.
  - Adaptability: Different services can be adapted to the broker.

- We propose a cloud brokering architecture oriented to manage the deployment of multi-tier and elastic services among available clouds.

Challenges

- Cloud interface: Different interfaces depending on the provider.

  OCCI or DeltaCloud: Make uniform the interface to any cloud provider.

- Instance types: Standard instances, High-CPU instances, High-memory instances, configurable and live-changeable instances, etc.

  Compare and characterize instances.

- Pricing schemes: Fixed prices (rarely change) or dynamic prices.

  Cloud manager: collects information & works as a pricing interface.

- Image types: Different image formats (e.g. AMI in Amazon EC2, etc.)

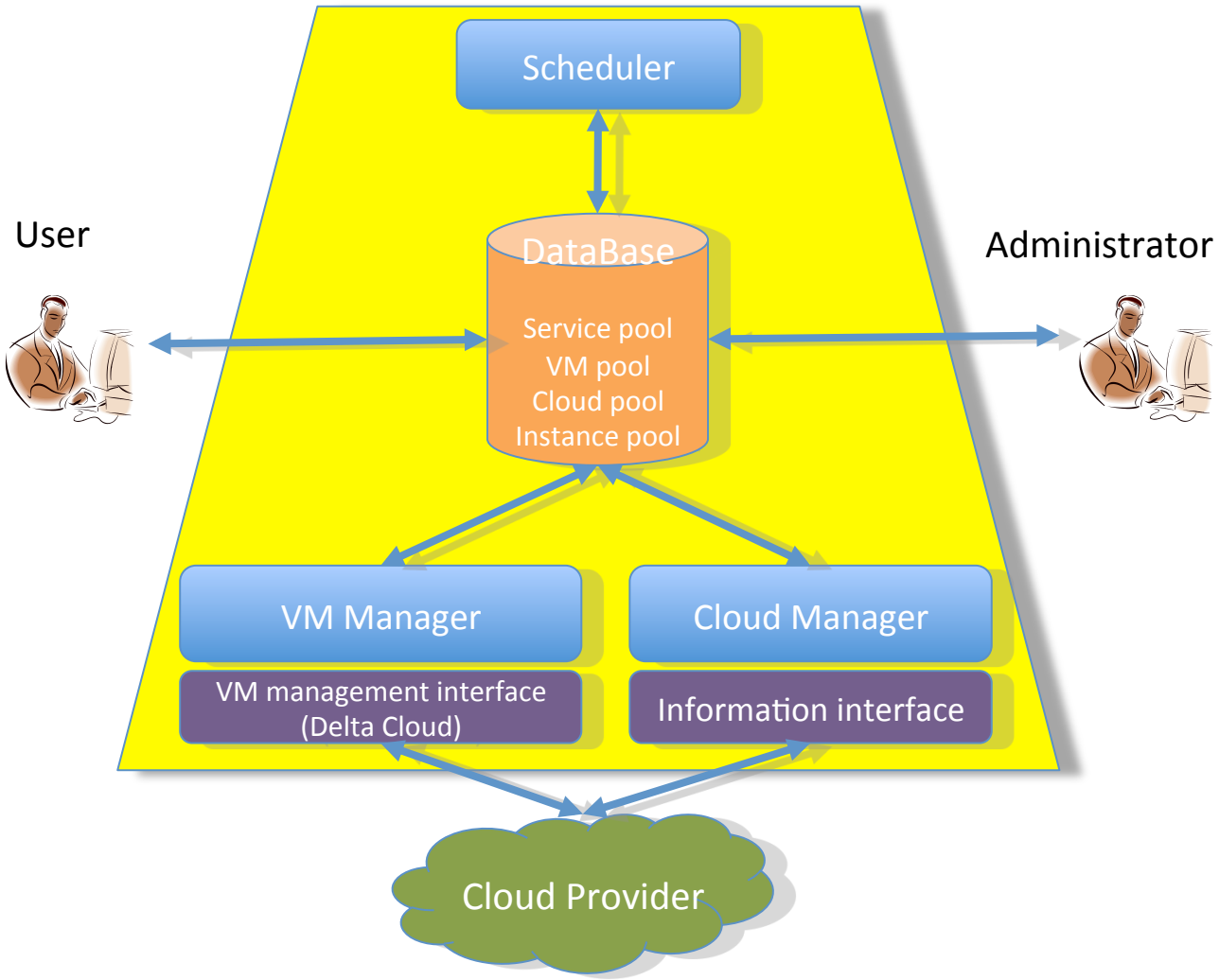
  Contextualize a default image.

- Network: Network communications, network latencies, Virtual Private Networks, etc.

System Architecture

A Multi-cloud Management Architecture and Early Experiences

Cloud Brokering



System Architecture

A Multi-cloud Management Architecture and Early Experiences

Roles

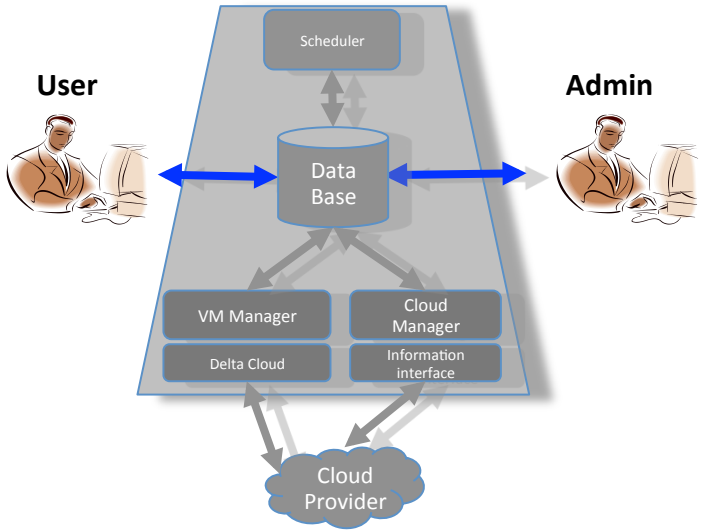
- User:
  - Receives information from the broker.
  - Specifies a new service, descibing it through a service description file.

- Administrator:
  - Configures the broker before the start of its execution:

    available cloud providers,

    available instance types,

    etc…



System Architecture

A Multi-cloud Management Architecture and Early Experiences

Components

- Database:

  Central component of the architecture.

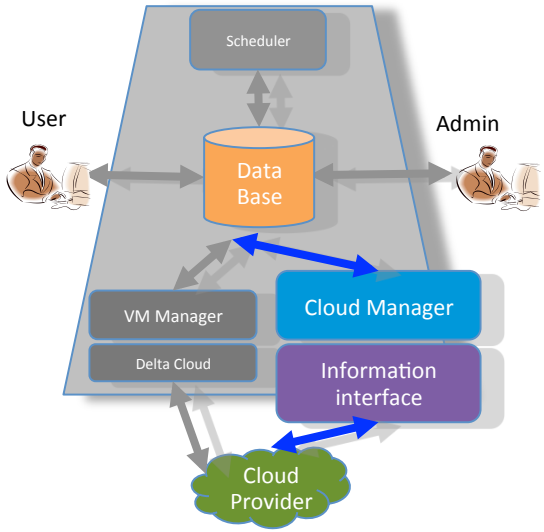
  Stores the Service, VM, Cloud and Instance lists.

- Cloud Manager:

  Collects information (instance availability, instance prices, etc.) for all the instances listed in the Instance list (via RSS, Web, etc.).

  Updates the database.

  Specially useful in dynamic prices case.



System Architecture

A Multi-cloud Management Architecture and Early Experiences

Components

- Scheduler:

  Reads the service description file provided by the user.

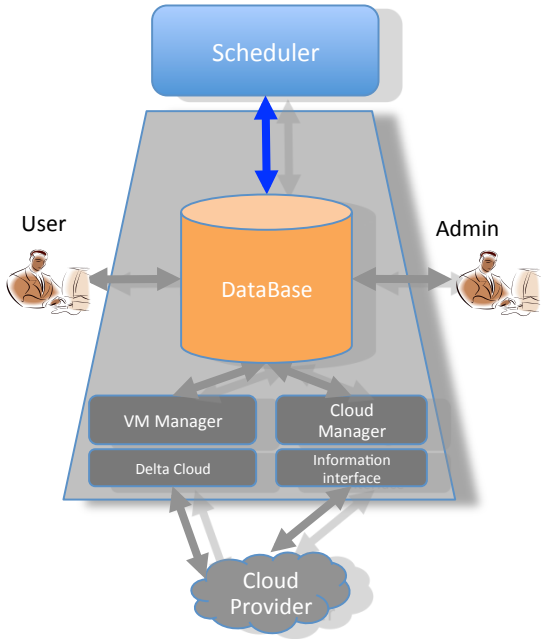
  Reads updated information from available instances and clouds.

  Invokes the scheduling module.

  Decides the best deployment option.

  Updates the VM pool.

  Mathematical Programming & Solvers.



System Architecture

A Multi-cloud Management Architecture and Early Experiences

Components

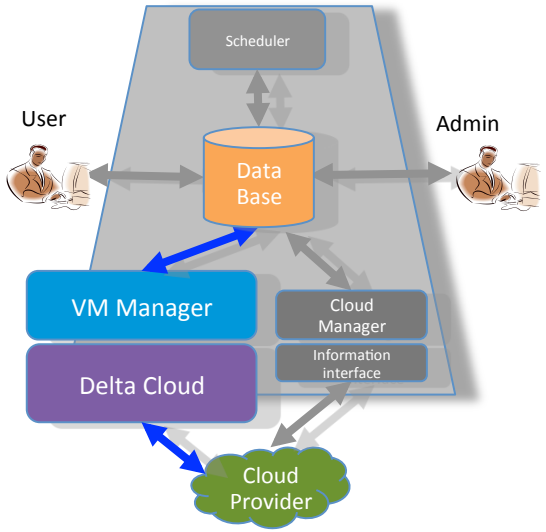
- VM Manager:

  Reads the VM list for new deployments.

  Control the VM life cycle: submits the VMs in pending state, shutdowns the VMs in terminated state.

  Monitors periodically the deployed VMs (CPU, Mem, etc.).

  Updates VMs information in the VM list.



System Architecture

A Multi-cloud Management Architecture and Early Experiences

Service description

- Defined by the user, contains detailed information about one service.

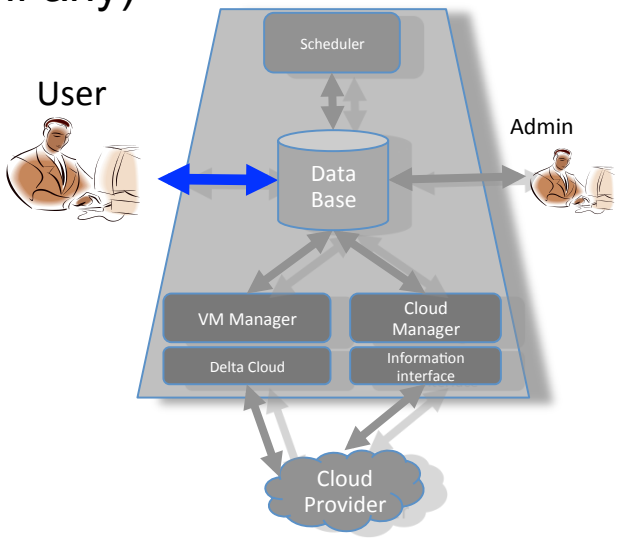
- A service is composed by a set of components (e.g., component1 (c1): Front-end (PBS, SGE, etc.); c2: NFS server; c3: worker nodes; etc.) with a scheduling module and start/end timing information.
- Each component:
  - Group of VMs (+ post-configuration files, if any)

  (e.g. c3: 10 worker nodes).
  - Scheduling strategy:

  Static or dynamic.
  - Optimization criteria

  (e.g. cost optimization).
  - User restrictions

  (e.g. only use 'small' instances).



More about scheduling

- Scheduling strategy (per component):
  - Static: Cloud providers conditions and user requirements do not change. The resource selection is done once, before service deployment.
  - Dynamic: Variable size services or changing cloud conditions (resource prices, resource availability, etc.).
    - Scheduling period: Interval between consecutive scheduling decisions.
    - Prediction algorithms: Since prices can vary along time depending on providers demand.

More about scheduling

- Optimization Criteria:

  The criterion to optimize is an unique function. This function is composed by one or more service variables. For example:

  - Infrastructure cost: Prices change even inside the same cloud provider (see different locations of Amazon EC2). Changing the placement of the VMs to the cheapest clouds would decrease user investment.

  - Service performance: Users can measure the performance of the available instances (hosting user application) in two ways; by service metrics (e.g. req/seg) or generic metrics (e.g. MFLOPS).

  - Others: Number of Instances, Number of CPU cores, Amount of RAM, Energy consumption, CPU consumption, Memory consumption, etc.

  - Or *combined criteria*, e.g. based on cost/performance ratio, or assigning a weight to each component.

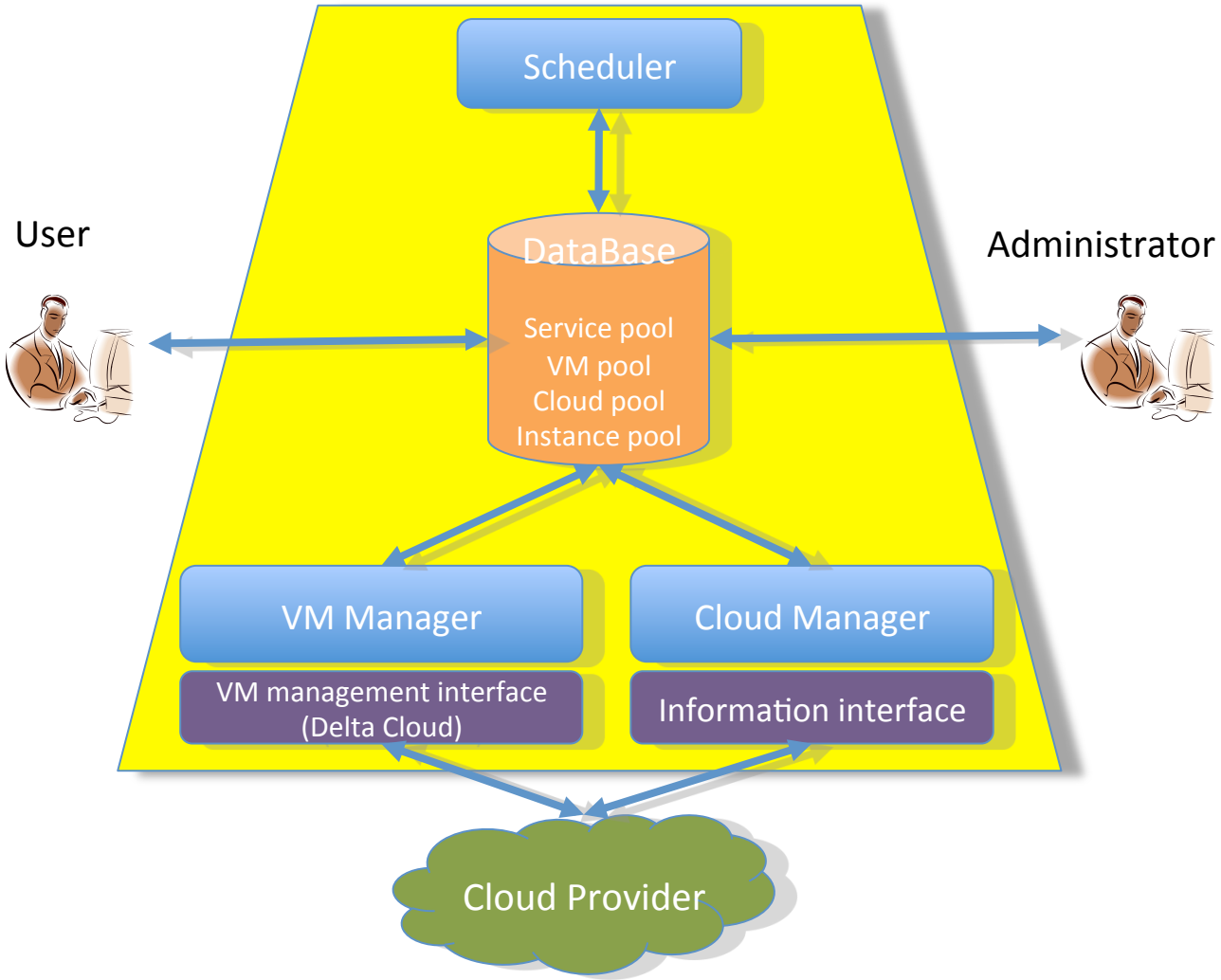
More about scheduling

- User constraints:
  - Instance types (IT): Some types of instance can work better than others with a particular application (e.g. only small and medium ITs; at least 20% of each IT; etc.).
  - Reallocation constraint: Some applications need to have several VMs online in every moment (typically web servers). The user can indicate that only 50% of the VMs can be reallocated in each scheduling decision.
  - Cloud constraint: The user can select his favorite clouds to deploy the VMs in (e.g. only Amazon EC2 to have all the VMs in the same cloud).
  - Others:
    - Maximum cost.
    - Minimum performance.
    - Max/Min number of Instances, Resource consumption, etc...

System Architecture

A Multi-cloud Management Architecture and Early Experiences

Recap



Deployment Case 1: Static scheduling.

- Use case description:

  "It's needed 16 virtual machines for a computation cluster. The objective is to maximize the VM computing capacity with a budget of X€ per hour."

- Scheduling strategy: static scheduling (scheduled only once).
- Optimization criterion: maximize VMs computing capacity.

Instance type	small	medium	large	xlarge
CPU (# cores)	1	1	2	4
CPU (Ghz/core)	1	2	2	2
Memory (GB)	1.7	3.5	7.5	15
Disk (GB)	160	300	850	1700
Computing capacity	1	2	4	8

Tab. 1 – Instance type features.

- Restrictions:
  - Price: different budgets.
  - Pricing scheme: On-demand prices.
  - Cloud allocation: Amazon EC2 (US and EU) or ElasticHost.
  - Instance type: small, medium, large or extra large instances.

Early Experiences

A Multi-cloud Management Architecture and Early Experiences

Deployment Case 1: Static scheduling.

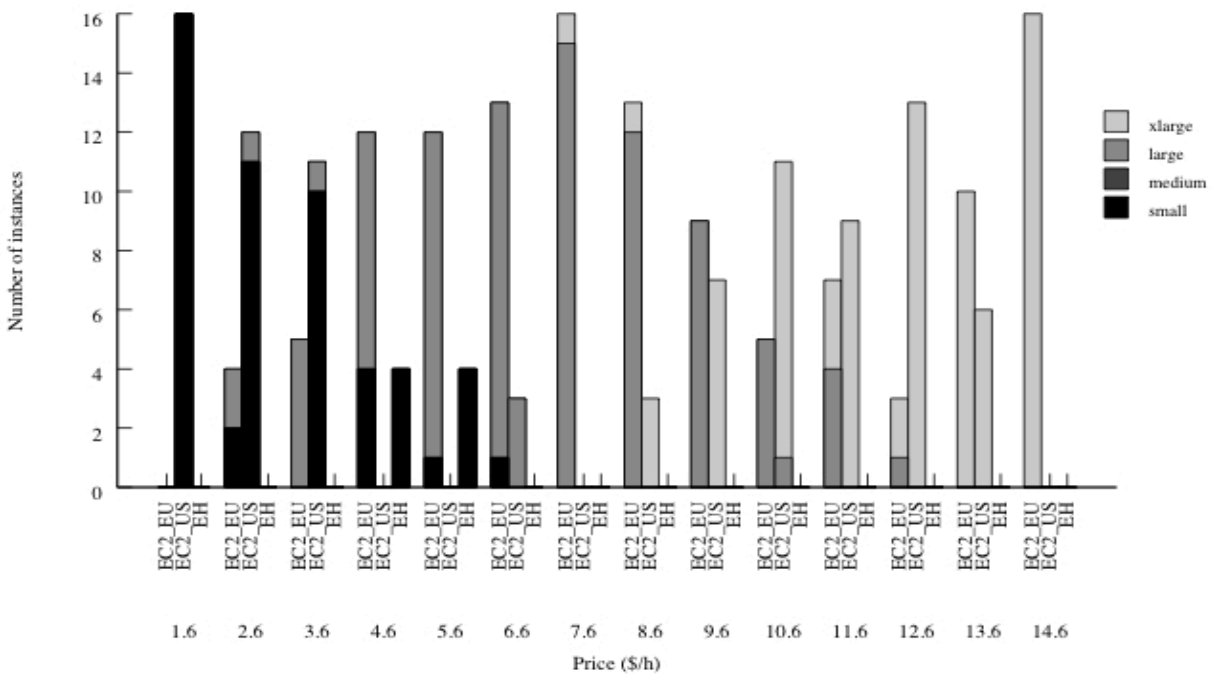


Fig. 1 – Optimal deployment of the required infrastructure.

Max_price	total_price	Max_price	total_price
1.6	1.6	8.6	8.56
2.6	2.6	9.6	9.56
3.6	3.6	10.6	10.6
4.6	4.592	11.6	11.6
5.6	5.582	12.6	12.6
6.6	5.69	13.6	13.6
7.6	7.48	14.6	14.08

Tab. 2 – Total cost of the infrastructure in each case.

Deployment Case 2: Dynamic scheduling.

- Use case description:

  "It's needed 11 'XL' virtual machines for a computation cluster to be used in a virtual classroom (8h to 22h). The objective is to minimize the total cost of the infrastructure."

- Scheduling strategy: dynamic scheduling.

- Optimization criterion: minimize total cost.

- Restrictions:
  - Pricing scheme: dynamic prices.
  - Cloud allocation: Amazon EC2 (USW, EU, AS)
  - Instance type: Extra large instances.

- Scheduling period: 1h.

Deployment Case 2: Dynamic scheduling.

Fig. 3 – Pricing evolution in some Amazon EC2 locations.

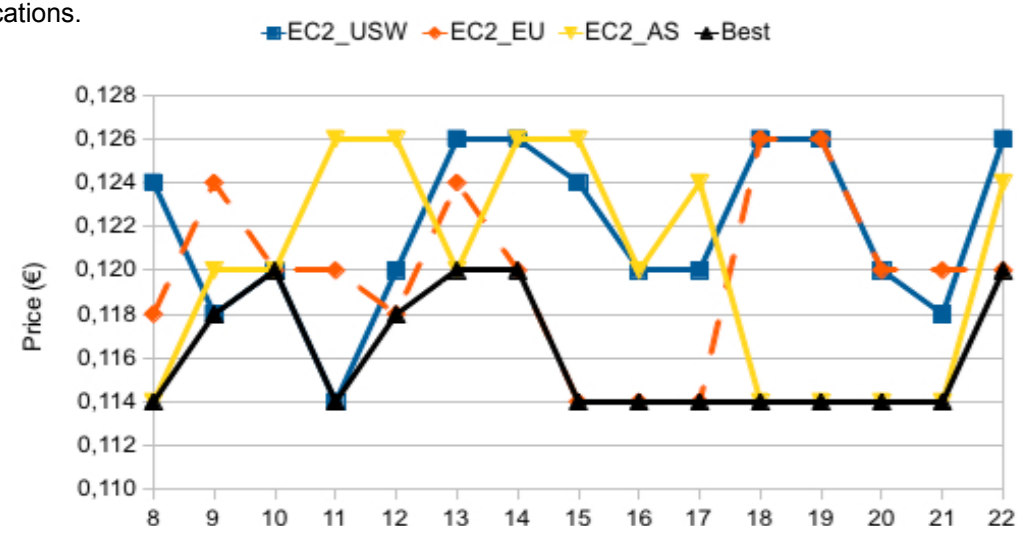
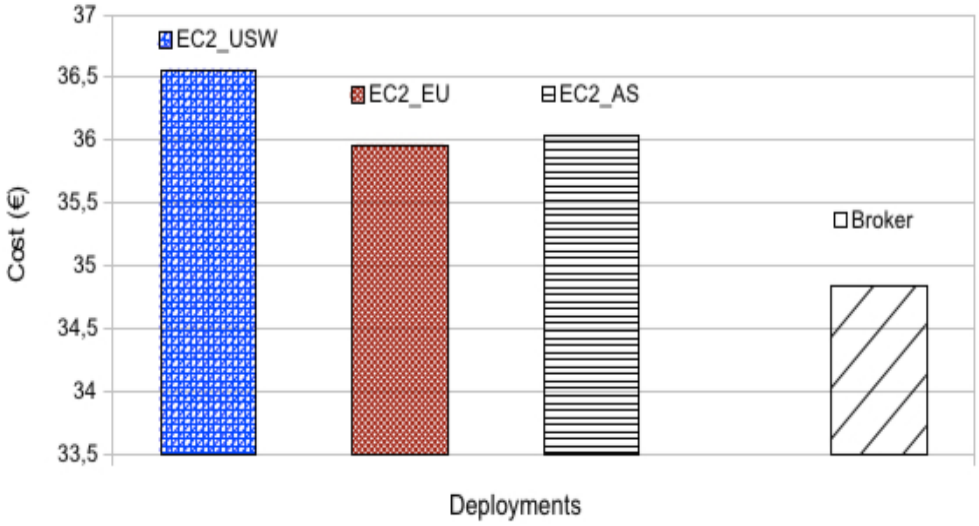


Fig. 4 - Cost comparison between using the broker or not.



Deployment Case 3: Dynamic scheduling.

- Use case description:

  "It's needed 11 'XL' virtual machines to be used as Web server workers. Some of them can be reconfigured. The objective is to minimize the total cost of the infrastructure."

- Scheduling strategy: dynamic scheduling.

- Optimization criterion: minimize total cost.

- Restrictions:
  - Pricing scheme: dynamic prices.
  - Cloud allocation: Amazon EC2 (USW, EU, AS).
  - Instance type: Extra large instances.
  - Reallocation: Different reallocation rates.

- Prediction algorithm.

Deployment Case 3: Dynamic scheduling.

Fig. 5 – Cost comparison between static and dynamic deployments.

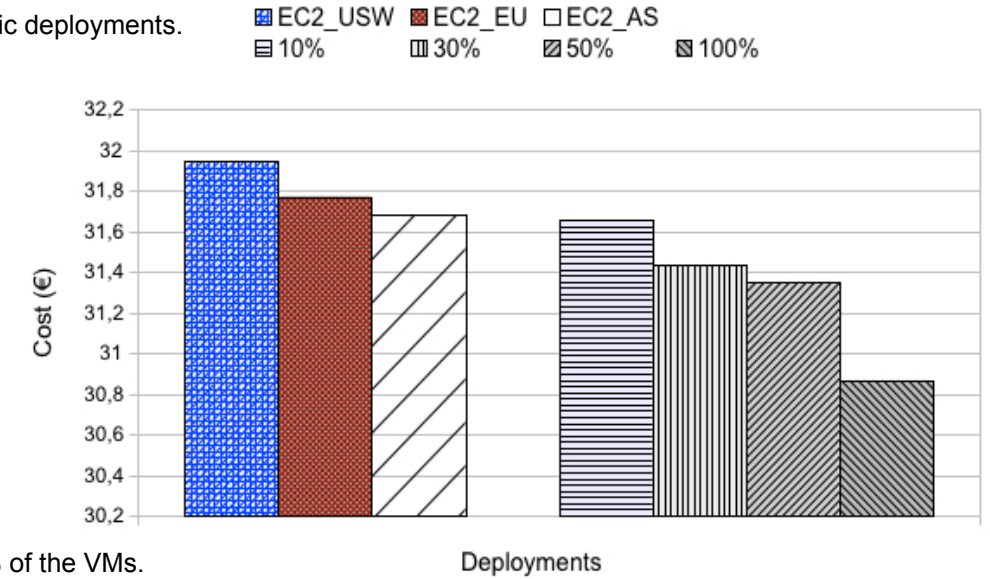
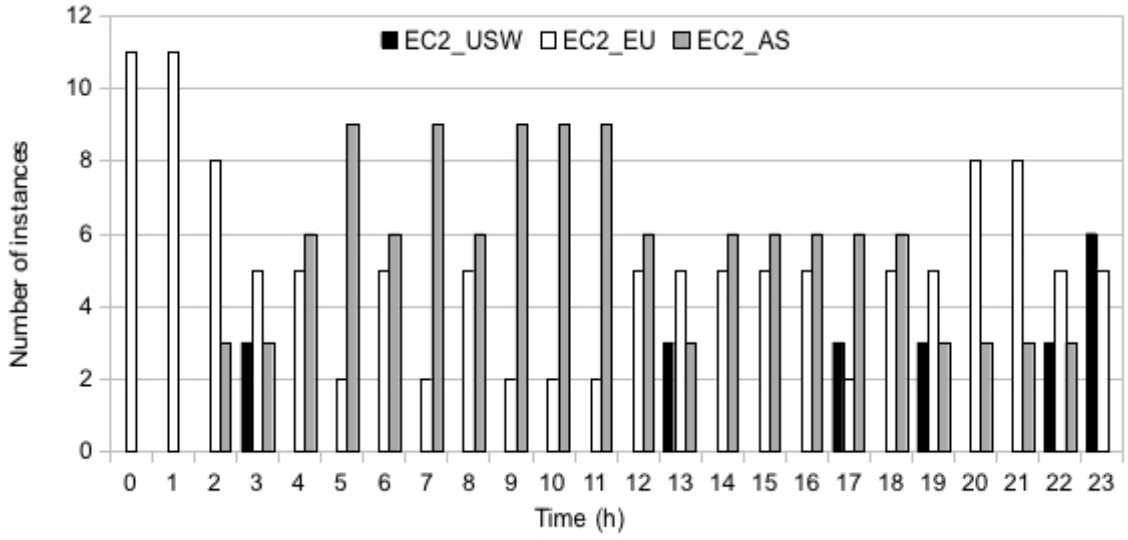


Fig. 6 - Dynamic deployment reallocating up to 30% of the VMs.



Conclusions and future work

Conclusions

- Broker architecture: Deploy services across multiple clouds.
- Optimize service parameters (cost, performance, etc.)
- Different scheduling policies, optimization criteria and user restrictions.
- Take advantage of the best features of each cloud for deploying an infrastructure.

Future work

- Characterize new applications and test their performance under optimal deployments in a multi cloud environment.
- Research in prediction algorithms.
- Research in scheduling strategies.

dsa-research.org

Other works

A Multi-cloud Management Architecture and Early Experiences

Publications

- R. Moreno-Vozmediano, R.S. Montero, I.M. Llorente. *Multi-Cloud Deployment of Computing Clusters for Loosely-Coupled MTC Applications*. **IEEE Transactions on Parallel and Distributed Systems**. Special Issue on Many Task Computing, 22(6):924-930, April 2011.

- R. Moreno-Vozmediano, R.S. Montero, I.M. Llorente. *Elastic management of web server clusters on distributed virtual infrastructures*. **Concurrency and Computation: Practice and Experience**, Article first published online, February 2011.

- R.S. Montero, R. Moreno-Vozmediano, I.M. Llorente. *An Elasticity Model for High Throughput Computing Clusters*. **Journal of Parallel and Distributed Computing**, 71(6):750-757, June 2010.

- I.M. Llorente, R. Moreno-Vozmediano, R.S. Montero. *Cloud Computing for On-Demand Grid Resource Provisioning*. **In Advances in Parallel Computing, Proceedings of HPC 2009**, Volume 18, Pages 177-191, 2009.

Questions

A Multi-cloud Management Architecture and Early Experiences

The research leading to these results has received funding from the Ministerio de Ciencia e Innovación of Spain through research grant TIN2009-07146.