CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms

Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya

Cloud Computing and Distributed Systems (CLOUDS) Laboratory
Department of Computer Science and Software Engineering
The University of Melbourne, Australia

Software: Practice and Experience (SPE), Volume 41, Number 1, Pages: 23-50,
ISSN: 0038-0644, Wiley Press, New York, USA, January, 2011

Outline

- Introduction
- Related Work
- CloudSim Architecture
- Design and Implementation
- Experiments and Evaluation
- Conclusions and Future Work

Introduction (1/2)

- "Cloud computing, the long-held dream of computing as a utility has the potential to transform a large part of the IT industry, making software even more attractive as a service." – University of Berkeley

- Clouds enable platform for dynamic and flexible application provisioning by exposing data center's capabilities as a network of virtual services
  - So users can access and deploy applications from anywhere in the Internet driven by demand and QoS requirements.

Introduction (2/2)

- It's not possible to perform benchmarking experiments in repeatable, dependable, and scalable environment using real-world Cloud.

- Considering that none of the current distributed system simulators offer the environment that can be used for modeling Cloud, we present CloudSim.

- Main contribution:
  - A holistic software framework for modeling Cloud computing environments and performance testing application services.

Related Work (1/2)

- Grid simulators:
  - GridSim
  - SimGrid
  - OptorSim
  - GangSim
- But none of them are able to isolate the multi-layer service abstractions(SaaS/PaaS/IaaS) differentiation and model the virtualized resources required by Cloud.

Related Work (2/2)

- HP, Intel and Yahoo established a global Cloud computing testbed, “Open Cirrus”.
  - Supporting a federation of data centers located in 10 organizations.
  - It’s expensive and hard to conduct repeatable experiments as resources vary from time to time due to its shared nature.
  - Access is limited to their members.
- Hence, simulation environments play an important role.

CloudSim Architecture (1/7)

- Modeling the Cloud

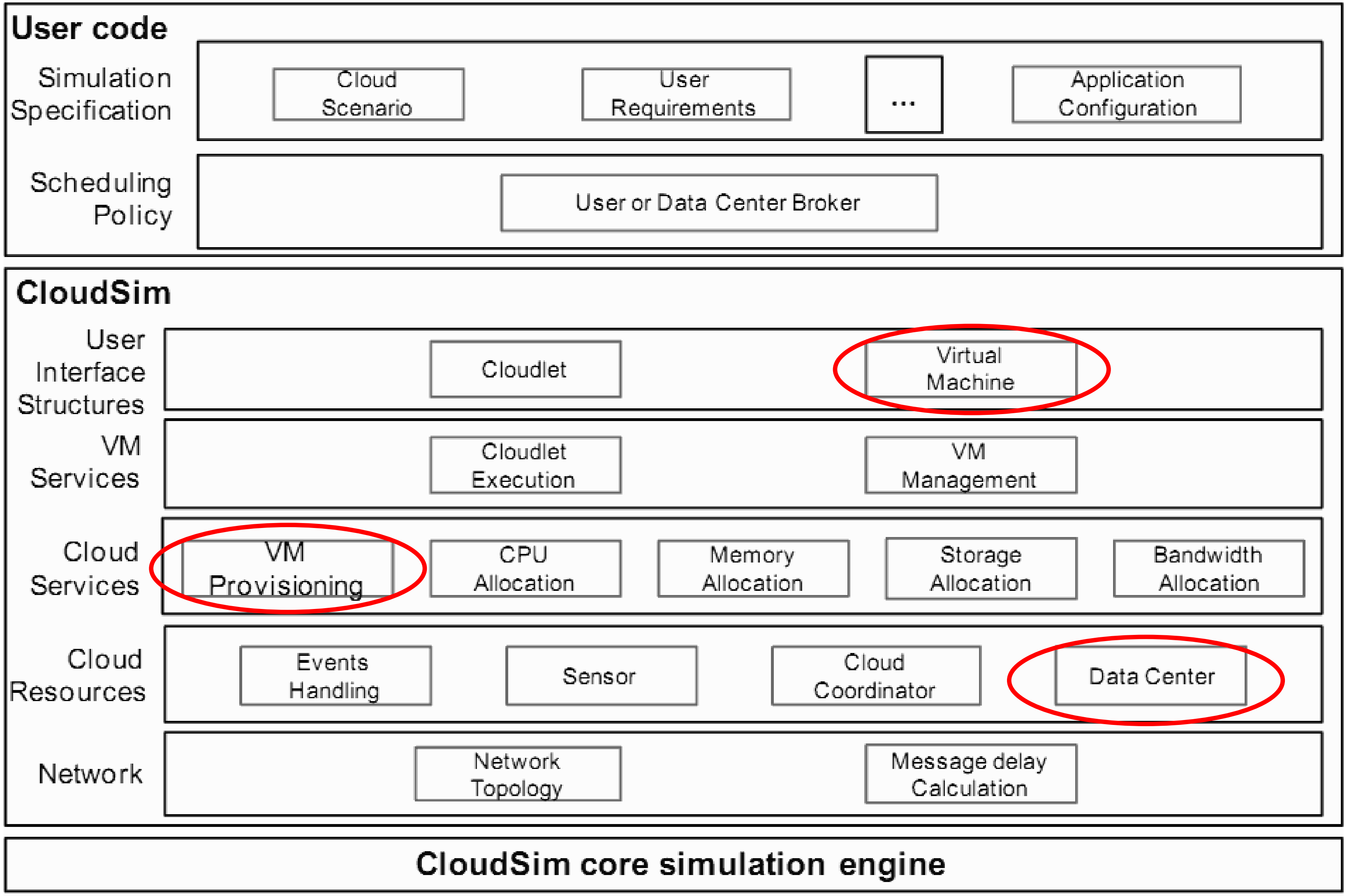


Figure 3. Layered CloudSim architecture.

CloudSim Architecture (2/7)

- Modeling the VM Allocation
  - Time-shared policy
  - Space-shared policy

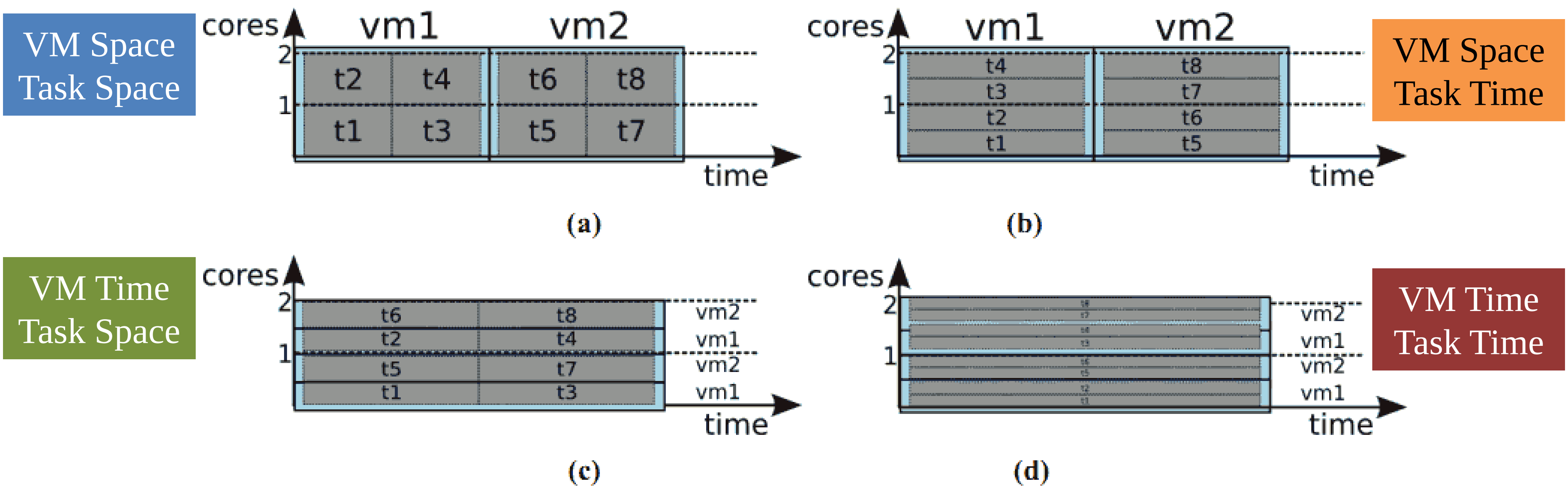


Figure 4. Effects of different provisioning policies on task unit execution: (a) Space-shared provisioning for VMs and tasks, (b) Space-shared provisioning for VMs and time-shared provisioning for tasks, (c) Time-shared provisioning for VMs, space-shared provisioning for tasks, and (d) Time-shared provisioning for VMs and tasks.

CloudSim Architecture (3/7)

- Modeling the Network Behavior (1/2)

Table 1. Latency Matrix.

$$\begin{bmatrix} 0 & 40 & 120 & 80 & 200 \\ 40 & 0 & 60 & 100 & 100 \\ 120 & 60 & 0 & 90 & 40 \\ 80 & 100 & 90 & 0 & 70 \\ 200 & 100 & 40 & 70 & 0 \end{bmatrix}$$

  - e_{ij} = Delay time from $entity_i$ to $entity_j$
  - It's much easier and cleaner to implement and manage.

CloudSim Architecture (4/7)

- Modeling the Network Behavior (2/2)

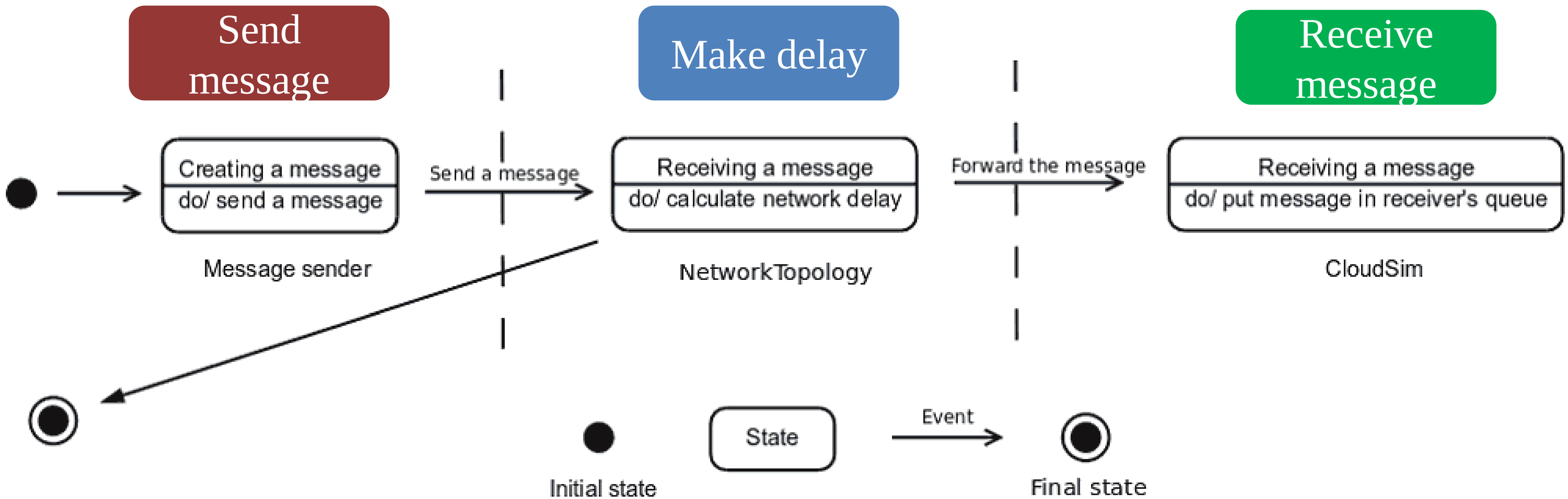


Figure 5. Network communication flow.

CloudSim Architecture (5/7)

- Modeling a Federation of Clouds
  - CloudCoordinator
  - Sensor

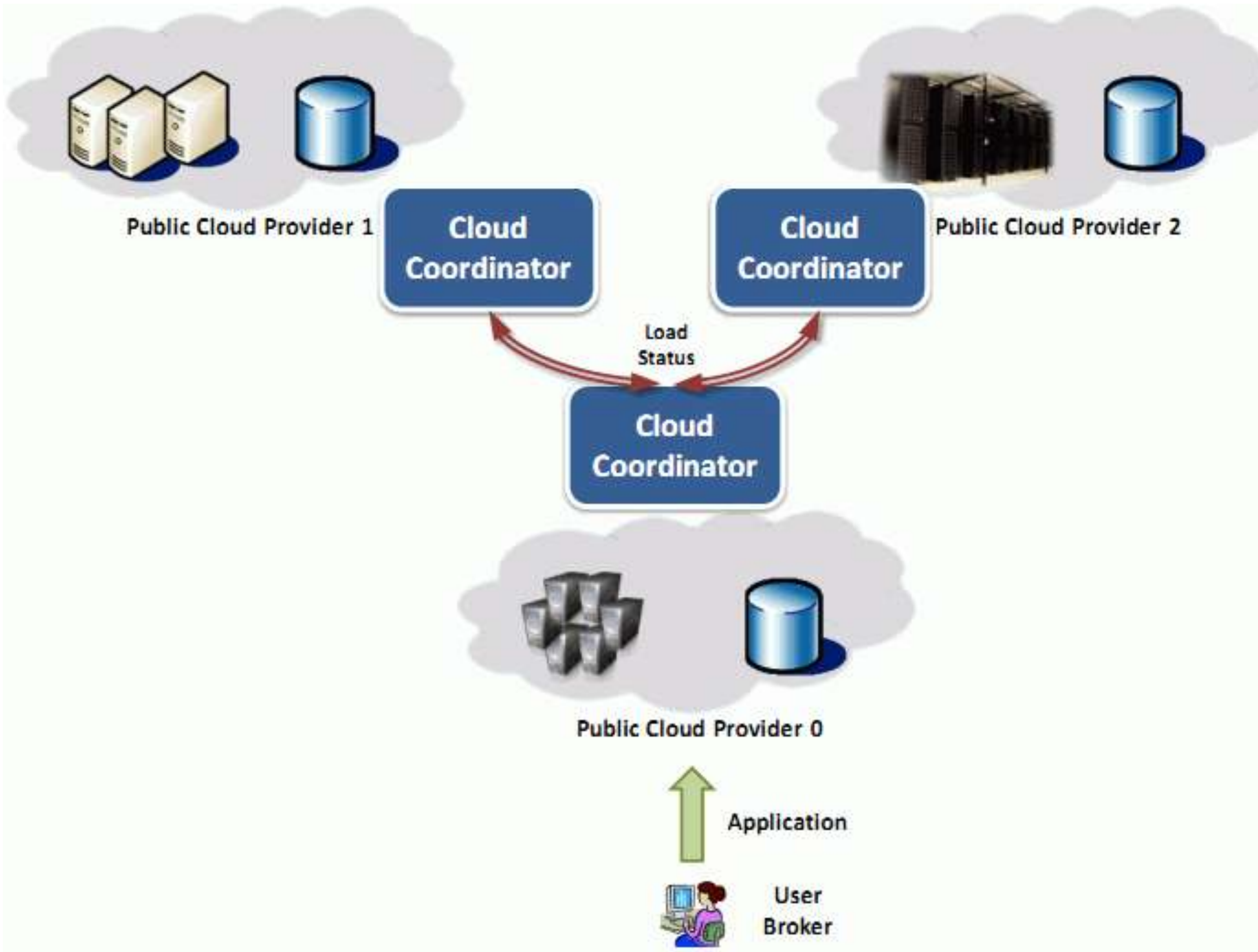


Figure 12. A network topology of federated Data Centers.

CloudSim Architecture (6/7)

- Modeling Dynamic Workloads
  - Cloudlet
  - UtilizationModel.getUtilization()
    - Input: Discrete time parameter
    - Return: Percentage of computational resource required
  - SLA model

CloudSim Architecture (7/7)

- Modeling Data Center Power Consumption
  - PowerModel.getPower()
    - Input: The current utilization for Cloud host
    - Return: The current power consumption value

Design and Implementation (1/4)

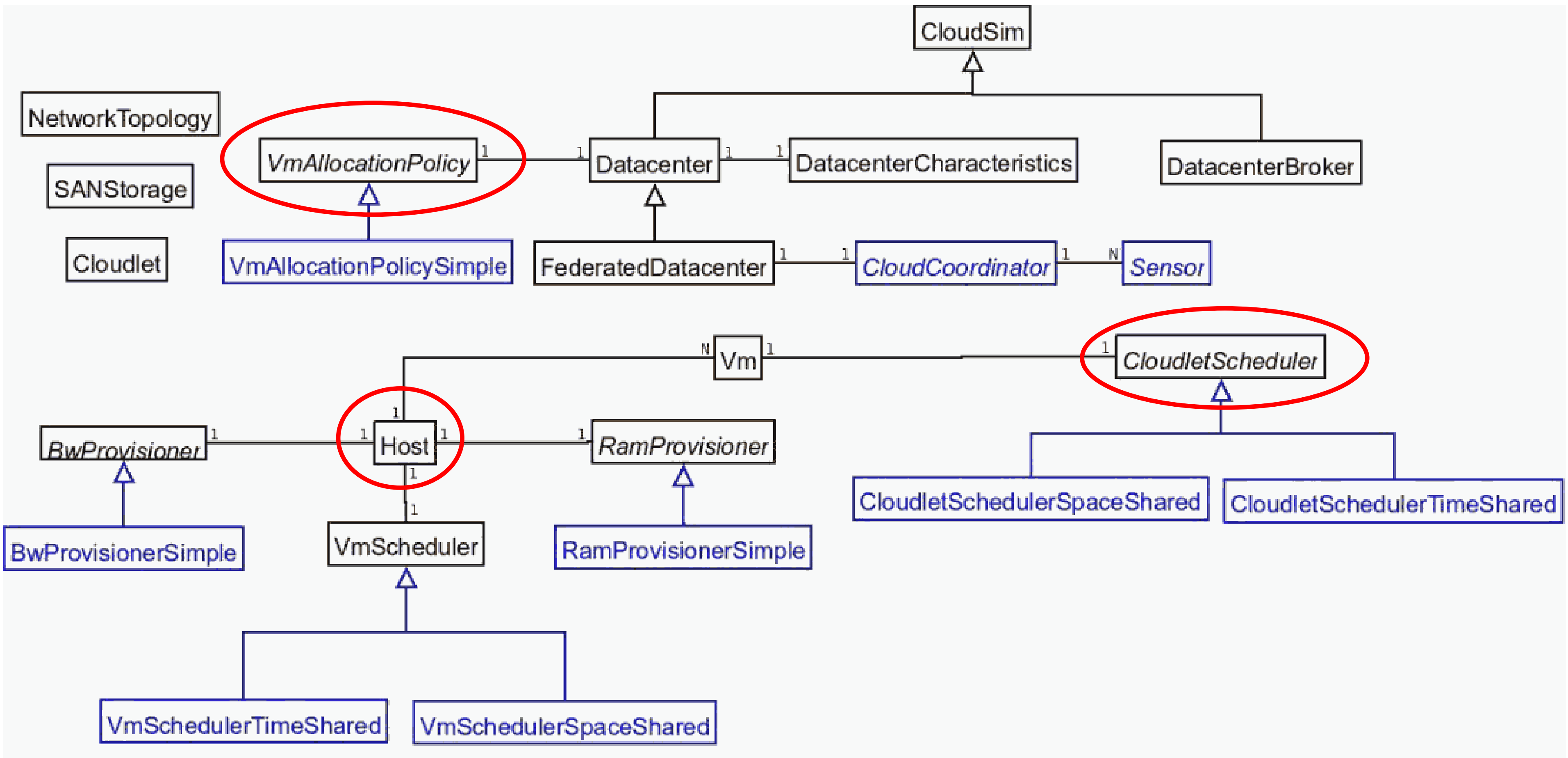


Figure 6. CloudSim class design diagram.

Design and Implementation (2/4)

- CloudSim Core Simulation Framework

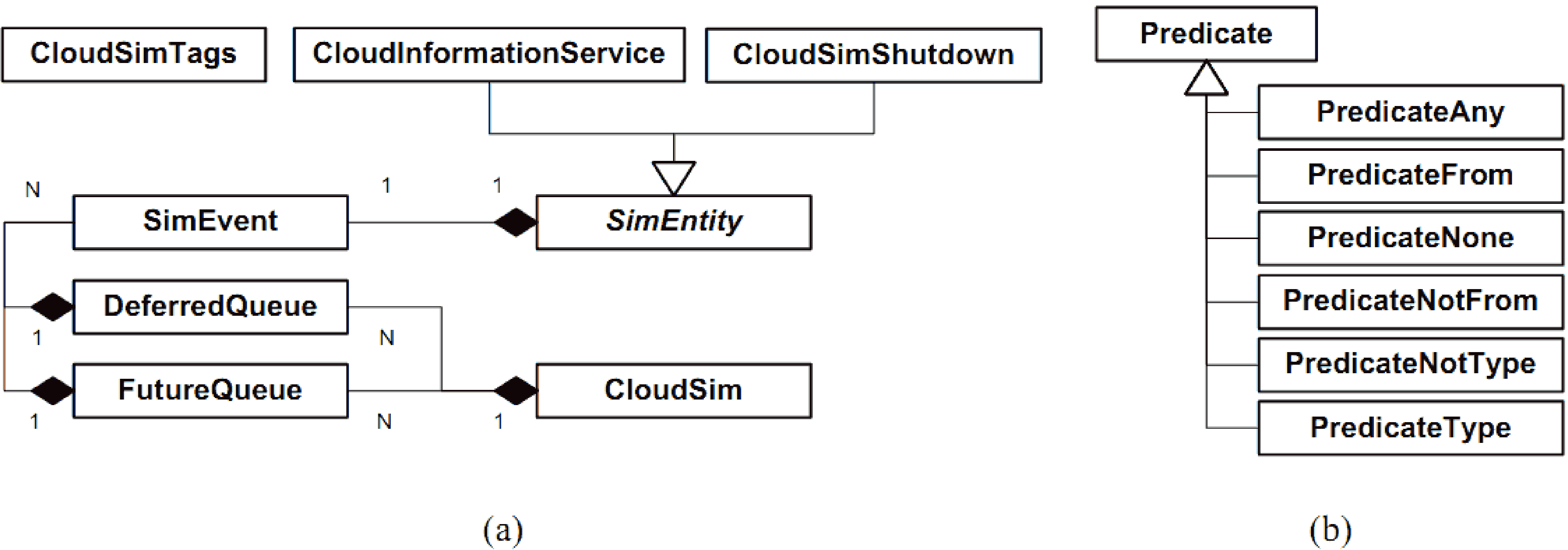


Figure 7. CloudSim core simulation framework class diagram: (a) main classes, (b) predicates.

- Dat

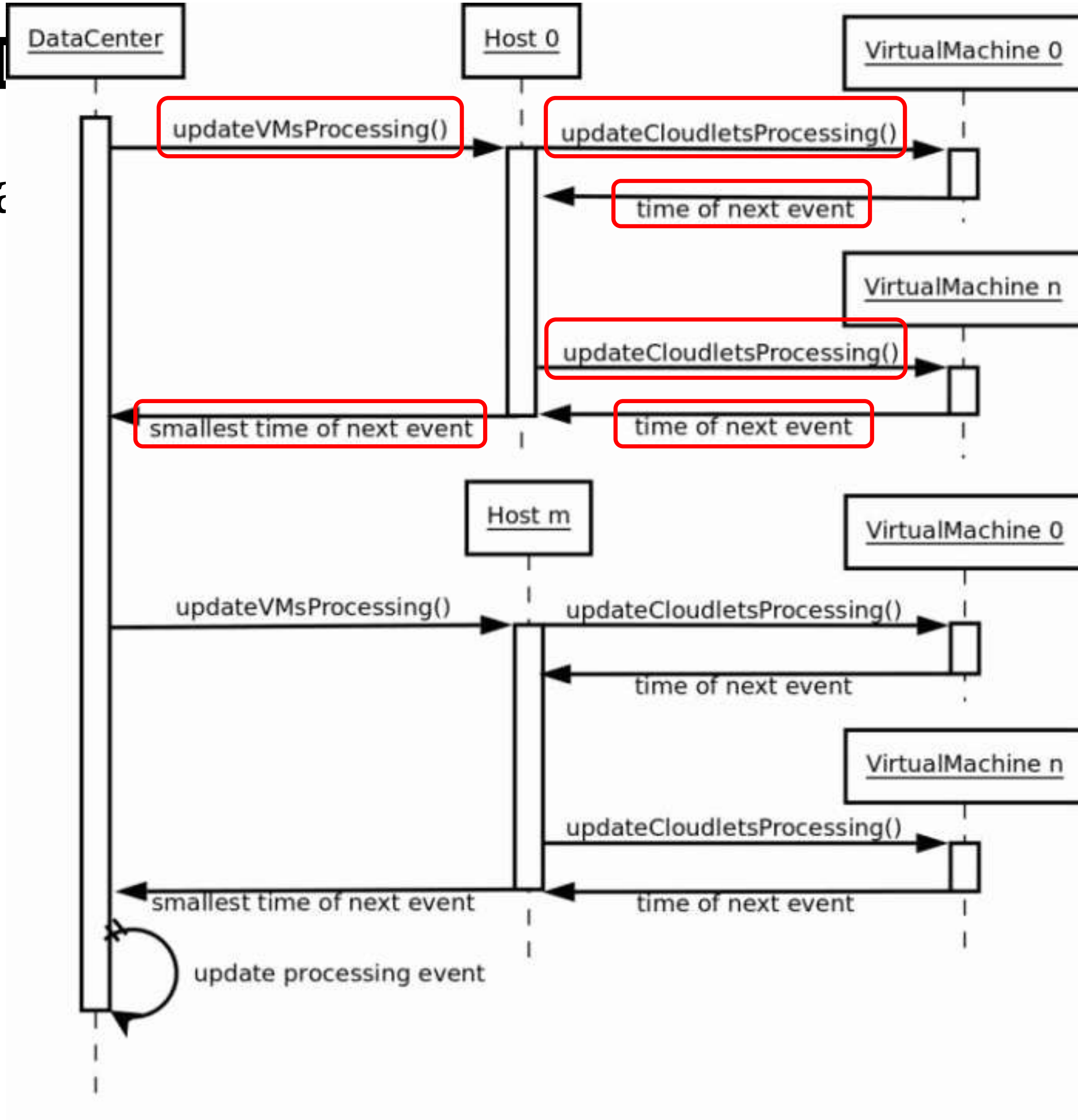


Figure 8. Cloudlet processing update process.

Design and Implementation (4/4)

- Communication among Entities

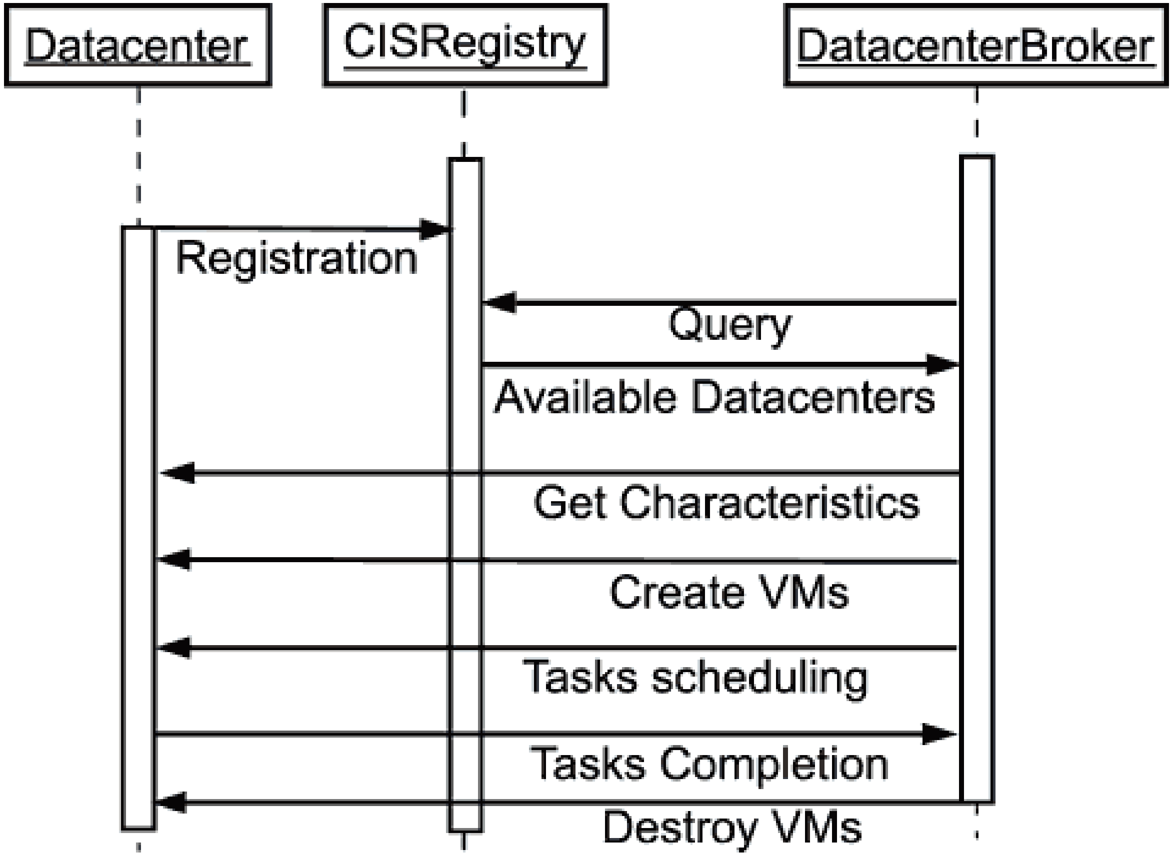


Figure 9. Simulation data flow.

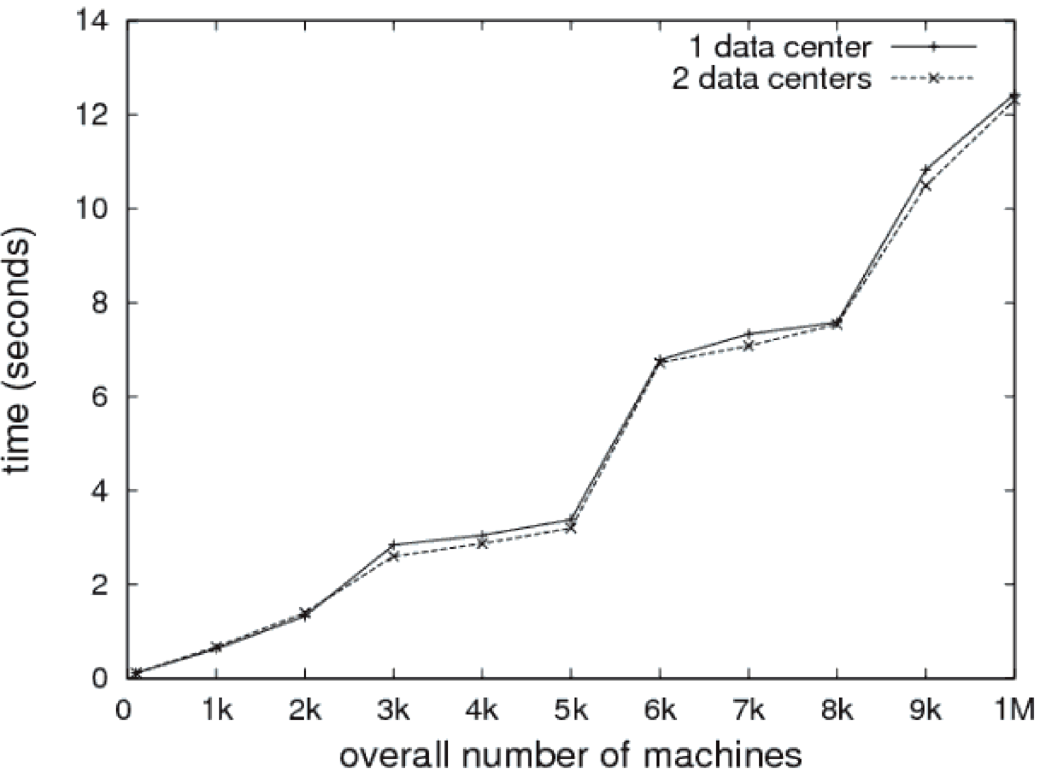
Experiments and Evaluation (1/9)

Scalability and Overhead Evaluation (1/3)

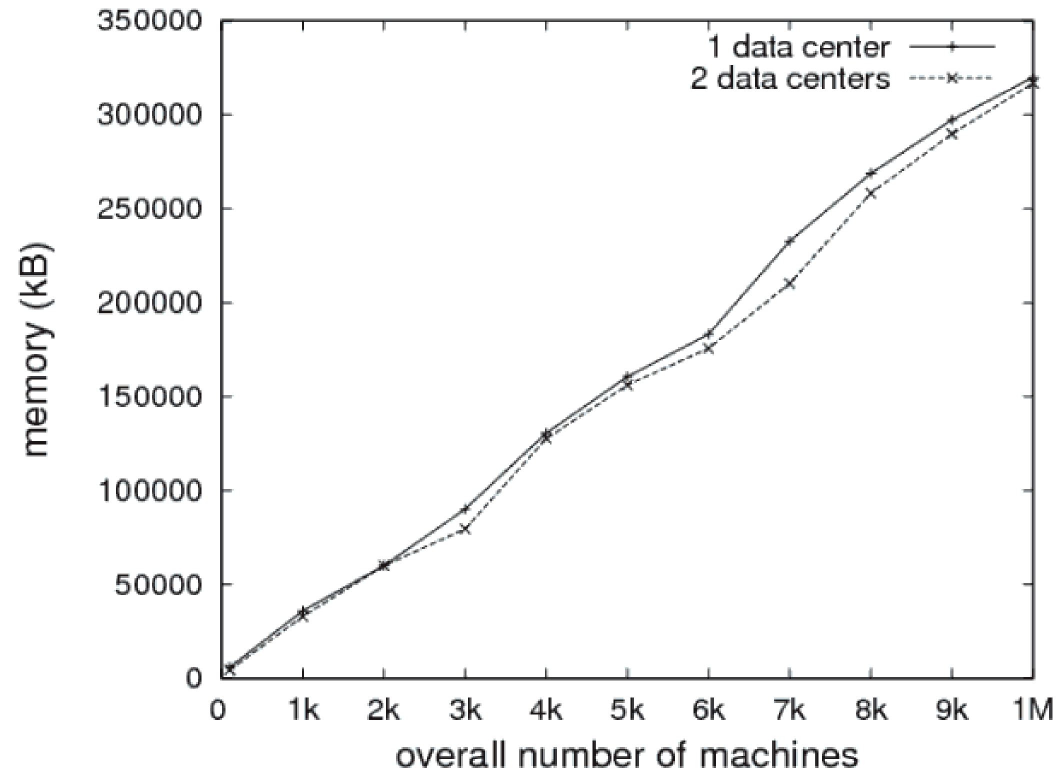
- The first tests that we present here are aimed at analyzing overhead and scalability of memory usage, and overall efficiency of CloudSim.
- The tests were conducted on a machine:
  - 2 Intel Xeon Quad-core 2.27 GHz
  - 16 GB of RAM memory
  - Created a single VM Ubuntu 8.04
- Two setup:
  1. All the machines were hosted within a single data center.
  2. The machines were symmetrically distributed across two data centers.
- Number of hosts in both the experiments varied from 1000 to 1,000,000. Each experiment was repeated 30 times.

Experiments and Evaluation (2/9)

Scalability and Overhead Evaluation (2/3)



(a)



(b)

Figure 10. CloudSim evaluation: (a) overhead (b) memory consumption.

Experiments and Evaluation (3/9)

Scalability and Overhead Evaluation (3/3)

- The results showed that overhead does not grow linearly with system size. Instead, it grows in steps when a specific number of hosts were used in the experiment.

- The obtained results showed that the time to instantiate an experiment setup with 1 million hosts is around 12 seconds.

- Hence, CloudSim is capable for supporting a large scale simulation environment with little or no overhead as regards to initialization time and memory consumption.

- “2 data centers are better” can be explained with the help of an efficient use of a multicore machine by Java.

Experiments and Evaluation (4/9)

Evaluating Federated Cloud Computing Components (1/4)

- The next set of experiments aimed at testing CloudSim's components that form the basis for modeling and simulation of a federated network of clouds (private, public, or both).

- We evaluated a straightforward load-migration policy that performed online migration of VMs across federated cloud providers in case the origin provider did not have the requested number of free VM slots available.
    1. Create a VM instance that had the same configuration as the original VM and which was also compliant with the destination provider configurations.
    2. Migrate the Cloudlets assigned to the original VM to the newly instantiated VM.

Experiments and Evaluation (5/9)

Evaluating Federated Cloud Computing Components (2/4)

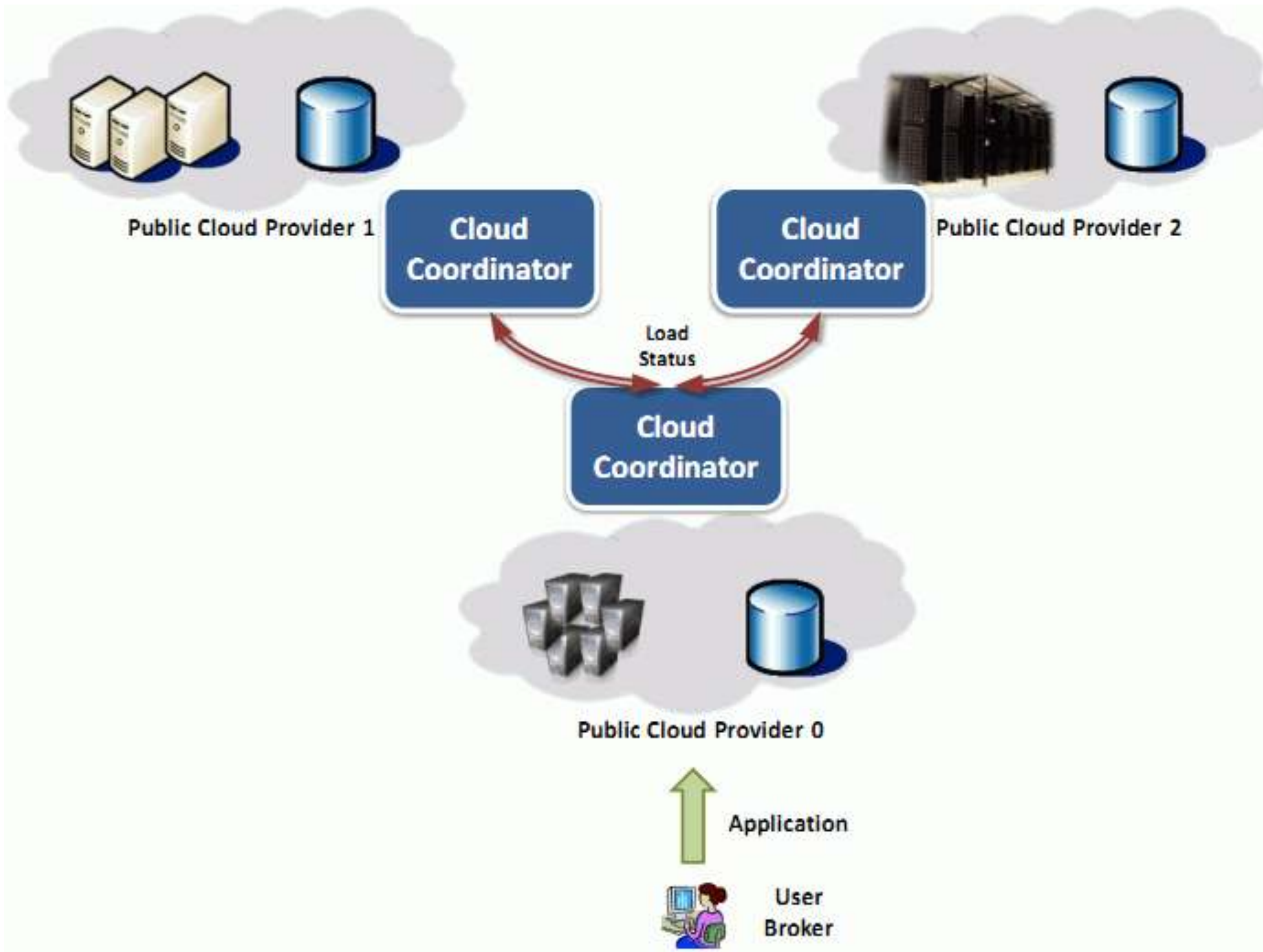


Figure 12. A network topology of federated Data Centers.

Experiments and Evaluation (6/9)

Evaluating Federated Cloud Computing Components (3/4)

- Data center in the federated network:
  - 50 hosts, 10GB RAM, 2TB storage, 1 processor with 1000MIPS
  - Time-shared VM scheduler
- User requested 25 VMs:
  - 256MB of memory, 1GB of storage, 1 CPU
  - Time-shared Cloudlet scheduler
  - Length of Cloudlets: 1,800,000 MIs
- Two setup:
  - With federated network
  - Without federated network

Experiments and Evaluation (7/9)

Evaluating Federated Cloud Computing Components (4/4)

Table 2. Performance Results.

Performance Metrics	With Federation	Without Federation
Average Turn Around Time (Secs)	2221.13	4700.1
Makespan (Secs)	6613.1	8405

- The simulation results revealed that the availability of federated infrastructure of clouds reduces the average turn-around time by more than 50%, while improving the makespan by 20%.

Experiments and Evaluation (8/9)

Case study: Energy-Conscious Management of Data Center (1/2)

- Test the capability of CloudSim for modeling energy-conscious VM provisioning.
  - 1 data center, 100 hosts
  - 1 CPU (1000 MIPS), 2GB RAM, 1TB storage
- Request for 400 VMs
  - 1 CPU (250 MIPS), 256MB RAM, 1GB storage
  - Each VM required 150,000 MIs (10 minutes)
- 3 techniques
  1. Benchmark trivial (Maximum frequency)
  2. DVFS enabled
  3. DVFS enabled + VM live migration if host utilization above threshold

Experiments and Evaluation (9/9)

Case study: Energy-Conscious Management of Data Center (2/2)

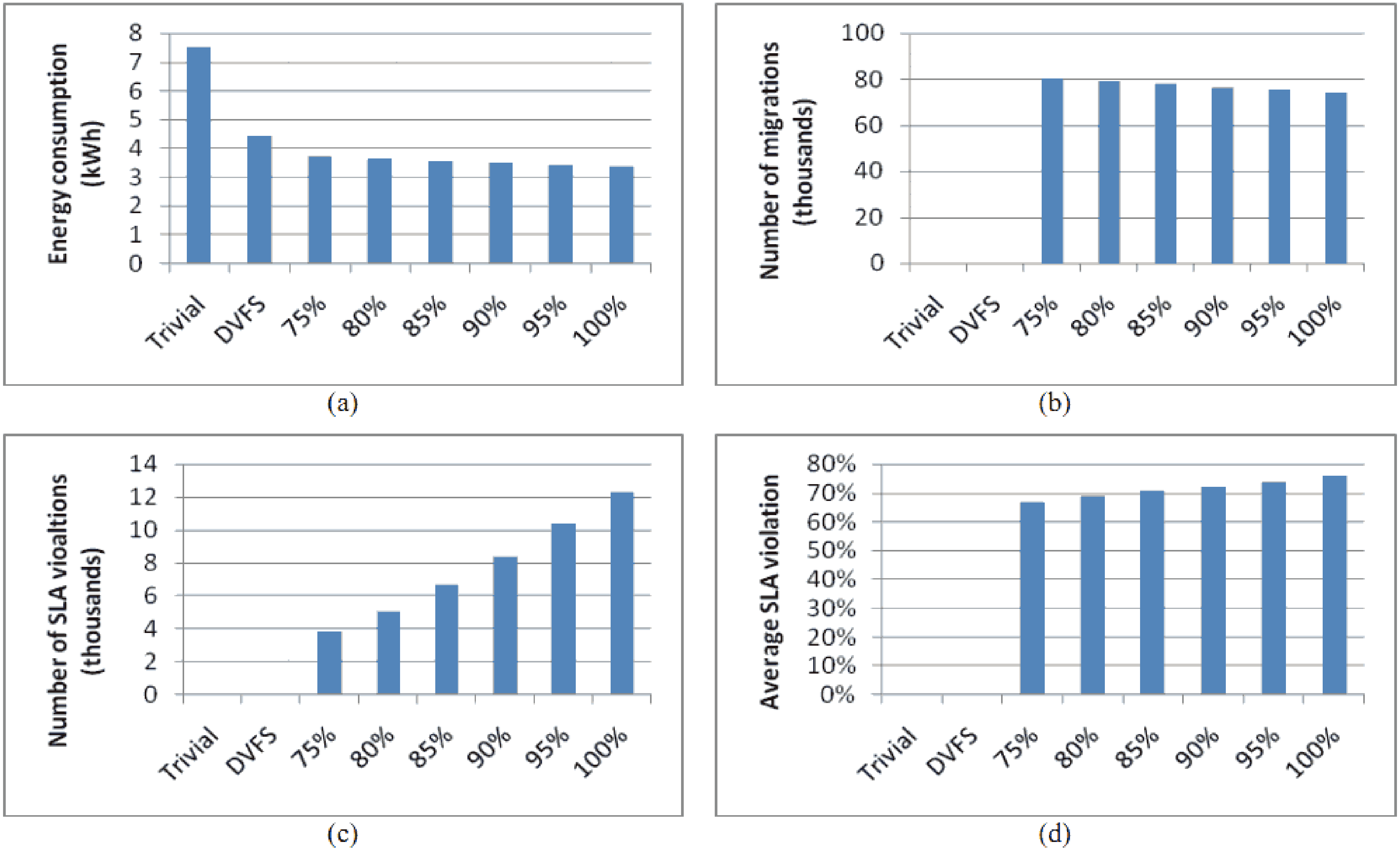


Figure 14. Experimental results: (a) total energy consumption by the system, (b) number of VM migrations, (c) number of SLA violations, and (d) average SLA violation.

Conclusions and Future Work (1/2)

- CloudSim is a simulation tool that allows Cloud developers:
  - Test performance of their provisioning policies in a repeatable and controllable environment free of cost.
  - Tune the bottlenecks before real-world deployment.
- By evaluations, we found CloudSim:
  - It supports a large scale simulation environment with little overhead.
  - It exposes powerful features that could easily be extended for modeling custom Cloud environments.

Conclusions and Future Work (2/2)

- We plan to add new pricing and provisioning policies to CloudSim. Others like:
  a) Workload models
  b) Models for database services such as blob, SQL
  c) QoS monitoring capability at VM
  d) Pricing models for public Clouds
- Furthermore, the geographical region where the data center is hosted influences total power bills.
  – Like locations where power cost is low and has less hostile weather conditions.
  – So, our future work will investigate new models and techniques for allocation of services to applications depending on energy efficiency and expenditure of service providers.