Outline

- Introduction
  - Related Work
  - Simulating Apps With Communicating Tasks
  - Implementation
  - Performance Evaluation
  - Conclusions
Cloud Computing

- Became popular amongst various communities
- Elastic and scalable on-demand resources
- Users want efficient and cost-effective execution to their applications
- Cloud providers’ configuration and policies should be chosen wisely
- Service-oriented features
- Clouds deploy wide variety of applications
- Both industry and scientific applications vary from massively parallel applications to message passing applications
- Cloud users want to test their applications and make sure they perform well on the cloud
Why Simulator?

- Cloud testbeds - variable demands, supply patterns, system sizes and resources. Multiple test runs can be very time consuming and sometimes impossible.
- Simulator - controlled experimentation, reproducible results and comparison of different solutions in similar environments.
Simulator Configuration

- Simulator should allow the user to configure:
  - Application model- in terms of computational tasks and communication between tasks
  - The Cloud platform- available resources and their connection
Before NetworkCloudSim

- Simulators view datacenter resources as a collection of VMs
- Integrate with very simplistic application models
- Result in non-realistic or inaccurate solutions
NetworkCloudSim Contributions

- More realistic application models than any other available Cloud simulation framework
- Support communicating applications such as MPI
- Scalable and fast simulations
- Configurable - allow simulating variety of network topologies
Message Passing Applications

- **message passing** sends a message to a process and relies on the process and the supporting infrastructure to select and invoke the actual code to run.

- For example, MPI - Message Passing Interface whose interface includes MPI_SEND and MPI_RECV, lets processes communicate and share data with each other.
Outline

• Introduction
• Related Work
• Simulating Apps With Communicating Tasks
• Implementation
• Performance Evaluation
• Conclusions
Grid Computing Simulators

- Can simulate parallel applications
- But no support for modelling virtualized resources, as those in the Cloud
Cloud Computing Simulators

- **Green Cloud**: Implements TCP/IP in the data center network, has large simulation time and high memory requirements.

- **CloudSim**: Scales well with low simulation overhead. No modelling of internal data center network.

- Both support only simplistic applications.
How Can We Improve Simulators?

- We want the simulated components to have more details about their physical analogs.
- In NetworkCloudSim, adding generic application model and data center network model.
- NetworkCloudSim uses the CloudSim simulator.
Outline

- Introduction
- Related Work
  - Simulating Apps With Communicating Tasks
- Implementation
- Performance Evaluation
- Conclusions
CloudSim Architecture

User code
Simulation Specification
Cloud Scenario  User Requirements
Scheduling Policy
User or Data Center Broker

CloudSim
User Interface Structures
Cloudlet  Virtual Machine
VM Services
Cloudlet Execution  VM Management
Application Cloudlet  Application
Cloud Services
VM Provisioning  CPU Allocation
Memory Allocation  Storage Allocation
Bandwidth Allocation
Cloud Resources
Events Handling  Sensor  Cloud Coordinator
Data Center  Networked Data Center
Network
Network Topology  Message delay Calculation

CloudSim Discrete Event Simulation Core
Design Issues- Application Model

- Currently no appropriate simulation of parallel and distributed applications
- We want to be able to simulate communicating tasks- consist of some computation and communication phases
- Decided to add the NetworkCloudlet class that represents a task that computes and communicates
Design Issues- Application Model

- Application Model- AppCloudlet object that consists of several NetworkCloudlet elements
- Each element runs in one VM (on the simulator)
- Element computation stage- number of MIPS or seconds
- Element communication stage- amount of transferred data
Design Issues- Network Modelling

- CloudSim supports network between data centers but not inside them.
- Sometimes there is significant network overhead within a data center, VM migration for example.
- Network Model: decided on flow network model which has less overhead than packet network model.
- ‘Packet’ requires sending more data between entities because it has congestion control modelling.
Design Issues- Network Modelling

- VM interconnections- in CloudSim each VM is connected to all other VMs
- We want a more realistic data center modelling- fat-tree type of network
- Added 3 levels of switches: root, aggregate and edge level
Network flow Model Design

- \( f = \text{size}_f, u, v \) is a flow where \( \text{size}_f \) is number of bytes in the flow
- \( bw \)- bandwidth, \( lat \)- network latency
- \( delay = lat + \frac{\text{size}_f}{bw} \) is the duration of a single work flow
Outline

- Introduction
- Related Work
- Simulating Apps With Communicating Tasks
- Implementation
  - Performance Evaluation
  - Conclusions
Design of NetworkCloudSim
Classes to model a network topology

- **Switch**: models delay in forwarding data
  - 3 types: root, aggregate, edge
- **NetworkPacket** and **HostPacket**
Classes for Application Modelling

- NetworkCloudlet
- AppCloudlet
- Extension of all scheduling classes
In two levels
  - Host level
  - VM level

For scheduling any application, NetworkCloudlets are scheduled on different VMs by the NetworkDatacenterBroker.

Scheduling within a VM is done by the NetworkCloudletScheduler.
The scheduler iterates over all the NetworkCloudlets in a VM.

For each NetworkCloudlet the scheduler checks for its current stage and operates as detailed in the algorithm.

When the current stage number reaches the total stages number, the NetworkCloudlet is finished.
VM Level Scheduling

14: if cl.currstage = 'Recv' then
15:    check the packet in packet_recv_queue
16:    cl.currnumstage ++
17:    update the current stage
18: end if
19: if cl.currstage = cl.totalNumStages then
20:    cl.currstage = Finished
21: end if
22: if cl.currstage = 'Finished' then
23:    update the total networkcloudlet execution time
24:    remove the networkcloudlet from current_execution_queue
25:    insert another networkcloudlet from waiting_queue to current_execution_queue
26:    notify the NetworkDatacenterBroker about the completion of networkcloudlet.
27: end if
28: end for
VM Level Scheduling Example
Outline

• Introduction
• Related Work
• Simulating Apps With Communicating Tasks
• Implementation
• Performance Evaluation
• Conclusions
First Experiment

- Compare real execution time of a controlled MPI program with the simulated execution time from the NetworkCloudSim
First Experiment

- Data Center Infrastructure
Two Experimental Scenarios

- (a) varying the amount of data transferred from the main process to other 7 processes
- (b) varying number of communicating processes with 1000000 MPI INT elements transferred from one process to another
Results

(a) Effect of Data Sent

(b) Effect of Communicating Processes
Second Experiment

- Internal simulator comparisons - no physical system
- Goal: Evaluation of task assignment and scheduling policies
- Submit a mixture of applications (parallel and parameter sweep) to the data center in 200 app/sec rate
- Same topology as in first experiment
Part 1

- Test the effect of resource allocation to each task of the application
- Compared 2 scheduling policies:
  - Random-NonOverlap - No stopping of execution of current network cloudlet
  - Random-Overlap - In case that the current network cloudlet is waiting for a data packet, we start executing the next network cloudlet in the queue
- Random signifies that allocation of a VM to a task is done randomly
Results

![Graph showing average response time for different mixture of applications]

- **X-axis**: Mixture of Applications (10%:90%, 30%:70%, 50%:50%, 70%:30%, 90%:10%)
- **Y-axis**: Average Response Time (secs/cloudlet)
- **Legend**:
  - Random-Overlap
  - Random-NonOverlap
Part 2

- Compare the impact of resource allocation policy
- Two policies:
  - Random-Overlap
  - RoundRobin-Overlap
Results
Experiments Conclusions

- First Experiment- Network CloudSim is effective in modelling the execution of parallel applications

- Second Experiment- Modelling of network is an essential part of the Cloud simulations
Outline

- Introduction
- Related Work
- Simulating Apps With Communicating Tasks
- Implementation
- Performance Evaluation
- Conclusions
Conclusions

- Use of simulation frameworks is becoming increasingly popular in Cloud computing community.
- These frameworks allow fast evaluation of scheduling and resource allocation mechanisms within Cloud data centers.
- NetworkCloudSim supports modelling of different network topologies and wide variety application models.
- Built on top of widely used simulator, CloudSim.
Additional Conclusions

- The evaluations results show high accuracy of the simulator.
- The simulator can help in building advance scheduling and resource allocation mechanisms for Clouds.
- In the future, integration of packet level network model.
Questions?