Introduction to Cloud Computing

Alexandru Iosup
Parallel and Distributed Systems Group
Delft University of Technology
The Netherlands

January 25, 2014
SPEC RG Cloud Meeting


* Subset.

“A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities [+ for] nontrivial QoS.” T. Foster, 1998 + 1999

January 25, 2014

What is Cloud Computing?
1. A Cloudy Buzzword

- 18 definitions in computer science (ECIS’10).
- NIST has one. Cal has one. We have one.
- “We have redefined cloud computing to include everything that we already do.” Larry Ellison, Oracle, 2009

January 25, 2014

What is Cloud Computing? 3. A Useful IT Service

“Use only when you want! Pay only for what you use!”

Q: What do you use?
Q: Why not this level?
Q: Why not this level?

January 25, 2014

Agenda

1. What is Cloud Computing?
2. IaaS Clouds, the Core Idea
3. The IaaS Owner Perspective
4. The IaaS User Perspective
5. Reality Check
6. Conclusion

January 25, 2014

IaaS Cloud Computing

VENI – @larGe: Massivizing Online Games using Cloud Computing
Joe Has an Idea ($$$)

Solution #1
Buy or Rent
- Big up-front commitment
- Load variability: NOT supported

Solution #2
Deploy on IaaS Cloud
- NO big up-front commitment
- Load variability: supported

Q: So are we just shifting the problem to somebody else, that is, the IaaS cloud owner?
- NO

Inside an IaaS Cloud Data Center

Time and Cost Sharing Among Users

Main Characteristics of IaaS Clouds
1. On-Demand Pay-per-Use
2. Elasticity (cloud concept of Scalability)
3. Resource Pooling
4. Fully automated IT services
5. Quality of Service

January 25, 2014
Agenda

1. What is Cloud Computing?
2. IaaS Clouds, the Core Idea
3. The IaaS Owner Perspective: How to Deploy a Cloud?
4. The IaaS User Perspective
5. Reality Check
6. Conclusion

IaaS Cloud Deployment Models

- Private: On-premises
- Public: Off-premises
- Hybrid

Resource Sharing Models

- Grids: Space-Sharing
- IaaS Clouds: Time-Sharing

Virtualization

- Applications
- Guest OS
- Virtual Resources

Virtualization and The Full IaaS Stack

- Virtual Machine Manager
- Virtual Infrastructure Manager

The Virtual Machine Lifecycle

1. Requested
2. Pending
3. Booting
4. Running
5. Terminated
6. Shutting-down
Use Case: Amazon Elastic Compute Cloud (EC2)

- Prominent IaaS provider
- Datacenters all over the world
- Many VM instance types
- Per-hour charging

<table>
<thead>
<tr>
<th>Instance</th>
<th>Capacity</th>
<th>US$/hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>m1.small</td>
<td></td>
<td>0.10</td>
</tr>
<tr>
<td>m1.large</td>
<td></td>
<td>0.38</td>
</tr>
<tr>
<td>c1.xlarge</td>
<td></td>
<td>0.76</td>
</tr>
</tbody>
</table>

Agenda

1. What is Cloud Computing?
2. IaaS Clouds, the Core Idea
3. The IaaS Owner Perspective
4. The IaaS User Perspective: How to Use Clouds? How to Choose Clouds?
5. Reality Check
6. Conclusion

Workload

MusicWave

OtherApp

OtherApp

OtherApp

OtherApp

Time

Load = 4

RunTime= 6

Provisioning and Allocation of Resources

Use Case: Workloads of Zynga (Massively Social Gaming)

Selling in-game virtual goods:
"Zynga made est. $270M in 2009 from."
http://techcrunch.com/2010/05/03/zynga-revenue/

"Zynga made more than $600M in 2010 from selling in-game virtual goods."
S. Greengard, CACM, Apr 2011

Sources: CNN, Zynga. 
Source: InsideSocialGames.com

Use Case: Workloads of Zynga (Massively Social Gaming)

- Load can grow very quickly
Provisioning and Allocation of Resources

Q: What is the interplay between provisioning and allocation?

Provisioning

Allocation

January 25, 2014

Provisioning and Allocation Policies

Q: How many policies exist? Q: How to select a policy?

Provisioning

Allocation


ETC.

January 25, 2014

Use Case: Two Provisioning Policies, Compared

Metrics for comparison

- Job Slowdown ($JSD$): Ratio of actual runtime in the cloud and the runtime in a dedicated non-virtualized environment

- Charged Cost ($C_C$)

- Utility ($U$)

January 25, 2014

Use Case: Two Provisioning Policies, Compared

Workloads

Uniform

Increasing

Bursty

January 25, 2014

Use Case: Two Provisioning Policies, Compared

Environments

<table>
<thead>
<tr>
<th>System</th>
<th>Hardware</th>
<th>VIM</th>
<th>Hypervisor</th>
<th>Max VMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>DASH/Delft</td>
<td>Dual quad-core 2.4 GHz 24 GB RAM 2x1 TB storage</td>
<td>OpenNebula</td>
<td>KVM</td>
<td>64</td>
</tr>
<tr>
<td>FIU</td>
<td>Pentium 4 3.0 GHz 5 GB RAM 340 GB Storage</td>
<td>OpenNebula</td>
<td>Xen</td>
<td>7</td>
</tr>
<tr>
<td>Amazon EC2</td>
<td>unknown/variuous</td>
<td>-</td>
<td>Xen</td>
<td>20</td>
</tr>
</tbody>
</table>

January 25, 2014

### Use Case: Many Provisioning Policies, Compared

**Job Slowdown (JSD)**

- When is OnDemand worse than Startup?
  - Waiting for machines to boot

**Charged Cost ($C$)**

- Why is OnDemand worse than Startup?
  - VM thrashing

**Utility ($U$)**

- Why no OnDemand on Amazon EC2?

### Agenda

1. What is Cloud Computing?
2. IaaS Clouds, the Core Idea
3. The IaaS Owner Perspective
4. The IaaS User Perspective
5. Reality Check: Who Uses Public Commercial Clouds?
6. Conclusion

### The Real IaaS Cloud

- "The path to abundance"
- On-demand capacity
- Cheap for short-term tasks
- Great for web apps (EIP, web crawl, DB ops, I/O)

- "The killer cyclone"
- Not so great performance for scientific applications (compute- or data-intensive)
Zynga zCloud: Hybrid Self-Hosted/EC2

- After Zynga had large scale
- More efficient self-hosted servers
  - Run at high utilization
- Use EC2 for unexpected demand

Other Cloud Customers

- 218 virtual CPUs
- 9TB/2TB block/S3 storage
- 6.5TB/2TB I/O per month

January 25, 2014

Agenda

1. What is Cloud Computing?
2. IaaS Clouds, the Core Idea
3. The IaaS Owner Perspective
4. The IaaS User Perspective
5. Reality Check
6. Conclusion

January 25, 2014

Agenda Take-Home Message

- Cloud Computing = IaaS + PaaS + SaaS
- Core idea = lease vs self-own
  - On-Demand, Pay-per-Use, Elastic, Pooled, Automated, QoS
- The Owner Perspective
  - Time-Sharing
  - Virtualization
- The User Perspective
  - Variable workloads
  - Provisioning and Allocation policies
- Reality Check: 100s of users

Thank you for your attention!
Questions? Suggestions? Observations?

More Info:
- http://www.st.ewi.tudelft.nl/~iosup/research.html
- http://www.st.ewi.tudelft.nl/~iosup/research_cloud.html
- http://www.pds.ewi.tudelft.nl/

Alexandru Iosup
A_Iosup@tudelft.nl
http://www.pds.ewi.tudelft.nl/~iosup/ (or google “iosup”)
Parallel and Distributed Systems Group
Delft University of Technology

January 25, 2014

... And Now For Something Different

- Ongoing projects in Cloud Computing at TUD
Cloudification: PaaS for MSGs

(Platform Challenge) Build Social Gaming platform that uses (mostly) cloud resources

• Close to players
• No upfront costs, no maintenance
• Compute platforms: multi-cores, GPUs, clusters, all-in-one!
• Performance guarantees
• Hybrid deployment model
• Code for various compute platforms—platform profiling
• Load prediction miscalculation costs real money
• What are the services?
• Vendor lock-in?
• My data

Proposed hosting model: dynamic

• Using data centers for dynamic resource allocation

Main advantages:
1. Significantly lower over-provisioning
2. Efficient coverage of the world is possible

Resource Provisioning and Allocation

Static vs. Dynamic Provisioning

Why Portfolio Scheduling?

• Old scheduling aspects
  • Workloads evolve over time and exhibit periods of distinct characteristics
  • No one-size-fits-all policy: hundreds exist, each good for specific conditions
• Data centers increasingly popular (also not new)
  • Constant deployment since mid-1990s
  • Users moving their computation to IaaS-cloud data centers
  • Consolidation efforts in mid- and large-scale companies
• New scheduling aspects
  • New workloads
  • New data center architectures
  • New cost models
  • Developing a scheduling policy is risky and ephemeral
  • Selecting a scheduling policy for your data center is difficult
  • Combining the strengths of multiple scheduling policies is ...

What is Portfolio Scheduling?

In a Nutshell, for Data Centers

• Create a set of scheduling policies
  • Resource provisioning and allocation policies, in this work
  • Online selection of the active policy, at important moments
  • Periodic selection, in this work
  • Same principle for other changes: pricing model, system
**Performance Evaluation**

1) Effect of Portfolio Scheduling (1)

A portfolio scheduler can be better than any of its constituent policies.

Q: What can prevent a portfolio scheduler from being better than any of its constituent policies?

- Portfolio scheduling is 8%, 11%, 45%, and 30% better than the best constituent policy.

**Performance Evaluation**

1) Effect of Portfolio Scheduling (2)

Q: How well do you think a single (provisioning, job selection, VM selection) policy would perform? Will it be dominant? (Rhetorical)

**Mobile Social Gaming and the SuperServer**

(Platform Challenge) Support Social Gaming on mobiles

- Mobiles everywhere (2bn+ users)
- Gaming industry for mobiles is new Growing Market
- SuperServer to generate content for low-capability devices?
- Battery for 3D/Networked games?
- Where is my server? (Ad-hoc mobile gaming networks?)
- Security, cheat-prevention

**Extending the Capabilities of Mobile Devices through Cloud Offloading ... with Application to Online Social Games**

Design & Implementation:

- based on OpenTTD
- repeatability
- offloading mechanisms
- instrumentation for metrics

Metrics:

- processing time
- packet size
- inter-arrival rate

**Social Everything! So Analytics**

- Social Network=undirected graph, relationship=edge
- Community=sub-graph, density of edges between its nodes higher than density of edges outside sub-graph

(Analytics Challenge)

Improve gaming experience

- Ranking / Rating
- Matchmaking / Recommendations
- Play Style/Tutoring

Self-Organizing Gaming Communities

- Player Behavior

Adapted from: Dagstuhl Seminar on Information Management in the Cloud, http://www.dagstuhl.de/program/calendar/partlist/?semnr=11321&SUOG
Benchmarking suite
Platforms and Process

- Platforms
  - Hadoop
  - Giraph
  - GraphLab
  - Stratosphere
  - YARN

- Process
  - Evaluate baseline (out of the box) and tuned performance
  - Evaluate performance on fixed-size system
  - Future: evaluate performance on elastic-size system
  - Evaluate scalability

Giraph: results for all algorithms, all data sets

- Storing the whole graph in memory helps Giraph perform well
- Giraph may crash when graphs or messages become larger

BFS: results for all platforms, all data sets

- No platform can run fastest of every graph
- Not all platforms can process all graphs
- Hadoop is the worst performer


Conclusion and ongoing work

- Performance is f(Data set, Algorithm, Platform, Deployment)
- Cannot tell yet which of (Data set, Algorithm, Platform) the most important (also depends on Platform)
- Platforms have their own drawbacks
- Some platforms can scale up reasonably with cluster size (horizontally) or number of cores (vertically)

- Ongoing work
  - Benchmarking suite
  - Build a performance boundary model
  - Explore performance variability

DotA communities

- Players are loosely organised in communities
  - Operate game servers
  - Maintain lists of tournaments and results
  - Publish statistics and rankings on websites
  - Dota-League: players join a queue and matchmaking forms teams
  - DotAlicious: players can choose which match/team to join

R. van de Bovenkamp, S. Shen, A. Iosup, F. A. Kuipers: Understanding and recommending play relationships in online social gaming. COMSNETS 2013: 1-10

Relationships in the gaming graph

- Players who regularly play together in DotAlicious do so in more diverse combinations than in Dota-League
- Contrary to Dota-League, DotAlicious players tend to play on the same side: playing together intensifies the social bond
- Winning together increases friendship relationships, while losing together weakens friendship relationships
- Small clusters of friends with very strong social ties exist

R. van de Bovenkamp, S. Shen, A. Iosup, F. A. Kuipers: Understanding and recommending play relationships in online social gaming. COMSNETS 2013: 1-10
**Matchmaking application**

- Replay match list, but also consider clusters in gaming graph
- Scoring methodology:
  - Points per cluster: Number of players in the match that are part of the same cluster
  - Excluding largest cluster of the network and clusters of size 1

<table>
<thead>
<tr>
<th>Player</th>
<th>Cluster</th>
<th>Player</th>
<th>Cluster</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>f</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>b</td>
<td>2</td>
<td>g</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>c</td>
<td>3</td>
<td>h</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>d</td>
<td>3</td>
<td>i</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>e</td>
<td>4</td>
<td>j</td>
<td>3</td>
<td>5</td>
</tr>
</tbody>
</table>

- Team 1
- Team 2
- Total of 7 points for this match

---

**Results matchmaking**

**Can do much better than random matchmaking**

**Can already improve original matchmaking algorithm for all gaming graphs!**