Architectural Implications of Cloud Computing

Grace A. Lewis
Research, Technology and Systems Solutions (RTSS) Program

SATURN 2011
May 18, 2011
Agenda

Basic Cloud Computing Concepts

Architectural Implications of Cloud Computing

Final Thoughts
Cloud Computing

“A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstracted, virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.”

Cloud Computing Types

Based on Type of Capability

- Software-as-a-Service (SaaS)
- Platform-as-a-Service (PaaS)
- Infrastructure-as-a-Service (IaaS)

Based on Who Can Access Resources

- Public Cloud
- Private Cloud

© 2011 Carnegie Mellon University
Infrastructure-as-a-Service (IaaS)

Mainly computational infrastructure available over the internet, such as compute cycles and storage

Allows organizations and developers to extend their IT infrastructure on an on-demand basis

Examples of IaaS Providers

- **Amazon Elastic Compute Cloud (EC2)**
  - Provides users a special virtual machine (AMI) that can be deployed and run on the EC2 infrastructure

- **Amazon Simple Storage Solution (S3)**
  - Provides users access to dynamically scalable storage resources

- **IBM Computing on Demand (CoD)**
  - Provides users access to highly configurable servers plus value-added services such as data storage

- **Microsoft Live Mesh**
  - Provides users access to a distributed file system; targeted at individual use
Platform-as-a-Service (PaaS)

Application development platforms that allow the usage of external resources to create and host applications of a larger scale than an individual or small organization would be able to handle.

Examples of PaaS providers:

- **Akamai EdgePlatform**
  - Large distributed computing platform for web application deployment (focus on analysis and monitoring of resources)

- **Force.com**
  - Platform to build and run applications and components bought from AppExchange or custom applications

- **Google App Engine**
  - Platform to develop and run applications on Google’s infrastructure

- **Microsoft Azure Services Platform**
  - On-demand compute and storage services as well as a development platform based on Windows Azure

- **Yahoo! Open Strategy (Y!OS)**
  - Platform to develop and web applications on top of the existing Yahoo! Platform (focus on social applications)
Software-as-a-Service (SaaS)

Model of software deployment in which a third-party provider licenses an application to customers for use as a service on demand

Examples

• Google Apps
  – Web-based office tools such as e-mail, calendar and document management tools

• Salesforce.com
  – Full customer relationship management (CRM) application

• Zoho
  – Large suite of web-based applications, mostly for enterprise use
Cloud Computing Types — Based on Access

Public
- Offered as a service, usually over an Internet connection
- Typically charge a pay-per-use fee
- Users can scale on-demand and do not need to purchase hardware
- Cloud providers manage the infrastructure and pool resources into capacity required by consumers

Private
- Deployed inside the firewall and managed by the user organization
- User organization owns the software and hardware running in the cloud
- User organization manages the cloud and provides cloud resources
- Resources typically not shared outside the organization and full control is retained by the organization
Drivers for Cloud Computing Adoption

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalability</td>
<td>Organizations have access to a large amount of resources that scale based on user demand</td>
</tr>
<tr>
<td>Elasticity</td>
<td>Organization’s can manually or dynamically decide on resource utilization based on changing needs</td>
</tr>
<tr>
<td>Virtualization</td>
<td>Each user has a single view of the available resources, independently of how they are arranged in terms of physical devices</td>
</tr>
<tr>
<td>Lower Infrastructure Costs</td>
<td>The pay-per-use model allows an organization to only pay for the resources they need with basically no investment in the physical resources available in the cloud. There are no infrastructure maintenance or upgrade costs</td>
</tr>
<tr>
<td>Availability</td>
<td>Organizations have the ability for the user to access data and applications from around the globe</td>
</tr>
<tr>
<td>Collaboration</td>
<td>Organizations are starting to see the cloud as a way to work simultaneously on common data and information</td>
</tr>
<tr>
<td>Risk Reduction</td>
<td>Organizations can use the cloud to test ideas and concepts before making major investments in technology</td>
</tr>
<tr>
<td>Reliability</td>
<td>In order to support SLAs (service-level agreements), cloud providers have reliability mechanisms that are much more robust than those that could be cost-effectively provided by a single organization</td>
</tr>
</tbody>
</table>
### Barriers for Cloud Computing Adoption

<table>
<thead>
<tr>
<th>Barriers</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>The key concern is data privacy: organizations do not have control of or know where their data is being stored</td>
</tr>
<tr>
<td>Interoperability</td>
<td>A universal set of standards and/or interfaces has not yet been defined, resulting in a significant risk of vendor lock-in</td>
</tr>
<tr>
<td>Resource Control</td>
<td>The amount of control that the organization has over the cloud environment varies greatly</td>
</tr>
<tr>
<td>Latency</td>
<td>All access to the cloud is done via the internet, introducing latency into every communication between the user and the environment</td>
</tr>
<tr>
<td>Platform or Language Constraints</td>
<td>Some cloud environments provide support for specific platforms and languages only</td>
</tr>
<tr>
<td>Regulation</td>
<td>There are concerns in the cloud computing community over jurisdiction, data protection, fair information practices, and international data transfer</td>
</tr>
</tbody>
</table>
Agenda

Basic Cloud Computing Concepts

Architectural Implications of Cloud Computing

Final Thoughts
IaaS: Examples of Architecture and Design Questions

What computation is performed in the cloud? A full application? Only certain functionality?

What data is stored in the cloud? Is it synchronized with other sets of data? Are there data privacy concerns?

What communication mechanisms exist between the consumer and the cloud resource?

How are resource failures detected and communicated? How are SLA commitments maintained and monitored?

What security mechanisms are provided by the cloud resource?

Systems residing in the cloud or using resources from the cloud will have to be designed and architected to account for lack of full control over important quality attributes.
PaaS: Examples of Architecture and Design Questions

What data is stored in the cloud? Is it possible for the system to run in the cloud and the data to remain local?

Will cloud resources always be active? Is a cloud bursting strategy appropriate?

Where do external users authenticate?

Are all system elements compatible with the cloud platform? Are adapters necessary?

What data is stored in the cloud? Is it possible for the system to run in the cloud and the data to remain local?
SaaS: Examples of Architecture and Design Questions

What type of client is used to interact with the SaaS resource?

How does the cloud system fit with the existing infrastructure?

What data adapters and transformers are necessary to interoperate with other systems?

What additional mechanisms need to be put in place to monitor system performance and usage?

Is the SaaS security architecture compatible with the organization’s security architecture?
Cloud Consumer Example Decision # 1: Data Model

Typical decisions of a distributed environment

• Local vs. Remote
• Total vs. Partitioned
• Distributed vs. Centralized
• Active Replication vs. Passive Replication
• Data Security Model

Challenges

• Data privacy
• Data synchronization
• Performance
Cloud Consumer
Example Decision # 2: User Authentication Model

Authentication is the mechanism by which consumers and providers prove to one another that they are acting on behalf of specific users or systems

Typical decisions of a distributed, multi-organizational environment

• Local vs. Remote Authentication
• Single Sign-On or Separate Authentication
• Local or Remote Identity Data
• Authentication Method

Challenges

• Incompatible authentication methods
• Physical security of identity data
• Synchronization of identity data
• Auditing
Cloud Consumer
Example Decision # 3: Allocation of Functionality

Decisions depend on the type of cloud implementation

• What functionality to deploy in the cloud?
• What functionality has to be implemented in addition to the functionality offered by the cloud provider?
  – Security
  – Management
  – Abstraction layers, e.g. data access, transformations, adapters
Cloud Consumer Example Decision # 4: Cloud Bursting

Refers to a system that is designed for average load, but is capable of load balancing to a cloud when it reaches its full capacity

Decisions

• Activation, initialization and de-activation of the cloud resource
• State and data synchronization
• Computational elements to determine full capacity
• Computational elements for monitoring load and usage
Cloud Consumer Example Decision # 5: Cloud Resource Management

Decisions

- Elements for failure detection and communication
- Elements for SLA monitoring
- Logging: where, what and when
Cloud Provider Example Decision #1: Multi-Tenancy

Mainly in SaaS implementations, a tenant is an organization that makes use of cloud resources.

Multi-tenancy requires

- Awareness of tenant context: the capability of recognizing the identity of the tenant requesting the resources based on message information as well as configuration data.
- Data isolation: tenants should only have access to their own data.
- Performance isolation: resource performance should conform to service-level agreements, regardless of the load on the system.
Cloud Provider
Example Decision #1: Multi-Tenancy

Option 1: One option is to have one instance per tenant.

Option 2: Another option is to have multiple identical instances that are managed by a load balancer.

Option 3: Another option is to have a single instance that uses tenant configuration data to provide tenant context.

Hybrid options are also possible.
Cloud Provider Example Decision #2: Virtualization Strategy

Virtualization in general is the abstraction of computing resources, e.g.

- Network virtualization: division of available bandwidth into channels that can be assigned to a particular resource in real time
- Storage virtualization: combination of physical storage devices into what appears to be a single storage device, e.g. SAN (storage area network)
- Server virtualization: hiding of server resources (number and identity of individual physical servers, processors, and operating systems) from server users, e.g. VMs (virtual machines)

Server Virtualization Example

[Image: news.cnet.com/i/bto/20090528/Virtualization_stack_270x258.jpg]
Cloud Provider Example Decision #2: Virtualization Strategy

Server Virtualization Example

Full Virtualization

- Requires dedicated machine
- OS can execute other applications, e.g. custom monitoring applications

Host-Based Virtualization

- Hypervisor
- Host OS
- Hardware
- OS can execute other applications, e.g. custom monitoring applications
- However, there is a performance penalty

Main question is “How and when are virtual machines deployed, started, initialized, deactivated, replaced, managed and terminated?”
Cloud Provider
Example Decision #3: Resource Interfaces

Cloud APIs are not yet standardized, so each cloud provider has its own specific APIs for managing its services. Currently, most Cloud APIs are SOAP- or REST-based.
Cloud Provider Example Decision #3: Resource Interfaces

Supported Protocols
- Functionality
- Configuration
- Management

QoS Support
- Security
- Usability
- Configurability

Sample Amazon EC2 Operations (IaaS)
- Create Image
- Stop Instances
- Create Security Group
- Monitor Instances

Sample Google App Engine Operations (PaaS)
- Upload Application Code
- Authenticate User
- Send E-mail

Sample Zoho.com Operations (SaaS)
- Set Up Application
- View Application Usage Data
- Embed in “X”
Agenda

Basic Cloud Computing Concepts
Architectural Implications of Cloud Computing
Final Thoughts
Cloud Computing is at the “Peak of Inflated Expectations”

Source: Gartner, Hype Cycle for Emerging Technologies, 2009

- Cloud Computing
- E-Book Readers
- Social Software Suites
- Microblogging
- Green IT
- Video Telepresence
- Mesh Networks: Sensor
- Online Video
- Corporate Blogging
- Wikis
- Electronic Paper
- Tablet PC
- Idea Management
- Web 2.0
- Social Network Analysis
- Over-the-Air Mobile Phone Payment Systems, Developed Markets
- RFID (Case/Pallet)
- SCA
- Location-Aware Applications
- Speech Recognition
- As of July 2009

Years to mainstream adoption:
- less than 2 years
- 2 to 5 years
- 5 to 10 years
- more than 10 years
- obsolete
- before plateau

Source: Gartner, Hype Cycle for Emerging Technologies, 2009
The Concept of Private Clouds is Starting to Appear

Source: Gartner, Hype Cycle for Emerging Technologies, 2010
Final Thoughts

Cloud Computing is in essence an economic model
  • It is a different way to acquire and manage IT resources
There are multiple cloud providers—the cloud is real
  • Currently most cloud consumers are small enterprises
  • Large enterprises are exploring private clouds
  • The number of providers will most probably grow as people start seeing greater savings and improvements to reduce adoption barriers

Cloud Computing adoption requires cost/benefit/risk analysis to determine
  • What resources to move to the cloud (if any)
  • What situations warrant use of cloud resources, even for one-time situations
  • Implementation of private clouds vs. usage of public clouds
  • What risks are associated with using resources on the cloud
  • What risks are associated to providing resources in the cloud
Final Thoughts

Decisions from a cloud consumer perspective depend on
  • Required control level
  • Required security level
  • Compatibility with local infrastructure

Decisions from a cloud provider perspective depend on
  • Market/user characteristics
  • Established SLAs
  • Available technology

In general, these are not fully technical decisions
  • Processes — especially engineering practices
  • Governance
  • Cost/Benefit analysis
Cloud Provider and Tool References

• 3tera: http://www.3tera.com/
• Amazon Elastic Compute Cloud (EC2): http://aws.amazon.com/ec2/
• Amazon Simple Storage Solution (S3): http://aws.amazon.com/s3/
• Eucalyptus Systems: http://www.eucalyptus.com/
• Force.com: http://www.salesforce.com/platform/
• Google App Engine: http://code.google.com/appengine/
• IBM Computing On Demand: http://www-03.ibm.com/systems/deepcomputing/cod/
• Microsoft Azure Services Platform: http://www.microsoft.com/azure/
• Microsoft Live Mesh: http://www.mesh.com/
• Salesforce.com: http://www.salesforce.com/crm/products.jsp
• Ubuntu: http://www.ubuntu.com/cloud
• Yahoo! Open Strategy (Y!OS): http://developer.yahoo.com/yos/intro/
• Zoho: http://www.zoho.com/
Contact Information

Grace A. Lewis
Research, Technology and Systems Solutions (RTSS) Program
System of Systems Practice (SoSP) Initiative

Software Engineering Institute
4500 Fifth Avenue
Pittsburgh, PA 15213-2612
USA

Phone: +1 412-268-5851
Email: glewis@sei.cmu.edu
WWW: http://www.sei.cmu.edu/about/people/glewis.cfm
NO WARRANTY

THIS MATERIAL OF CARNEGIE MELLON UNIVERSITY AND ITS SOFTWARE ENGINEERING INSTITUTE IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

Use of any trademarks in this presentation is not intended in any way to infringe on the rights of the trademark holder.

This Presentation may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

This work was created in the performance of Federal Government Contract Number FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.