UNIVERSITEIT TWENTE.

Cloud Computing: Should we care?

Luís Ferreira Pires

TTT presentation, 25 March 2010

Personal motivation

- A lot of people have been talking about Cloud Computing and I was curious about it
- I learned something about it I want to share this knowledge with you

Questions at the end

- Should we start taking cloud computing into account in our research or just ignore it as yet another deployment strategy?
- If we should consider cloud computing then, how should we do it?

Contents

Observation

- A lot of big people are putting big money in Cloud Computing
  → this should be hot, although no guarantee after the subprime crisis!

Definition of Cloud computing

Informal definition

- A model for computing in which something is done 'in the cloud'

What is 'the cloud' then?

- Iconic representation of the Internet!

Definition of Cloud Computing

Official definition

- A model for on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction

National Institute of Standards and Technology, August 2009

But what is the 'problem'?

'IT is often a showstopper'

Example: two companies merge and all organisational problems are solved, but the IT people of the new company says it is impossible to merge the 'old' IT systems in a reasonable timescale

- IT is often guided by trends, not real solutions
- Failure to do anything, fear of change

forces in different directions!

- IT people look for trends and are distracted from looking for real solutions to problems (support to business goals)
- IT architectures became too complex (often layered) and costly to maintain

Solution?

- Service-oriented architecture (SOA) has been introduced as an architectural solution for these problems → it is not a product, but rather a discipline or strategy (you can't buy it)
- Cloud computing helps when used in combination with SOA because it allows resources to be leveraged over the Internet as services in order to control IT costs and make IT more agile → companies concentrate on their core businesses

Man lifting a stone with a lever

Term most often used in the cloud computing literature meaning (probably) 'To move or lift with or as if with a lever' or 'To improve or enhance' (you choose!)

Types of cloud computing

Roughly speaking, resources ' in the cloud' can be software or hardware

Software resources

- Typically software applications that used to be downloaded and installed on the end users machine, and are now offered as web applications (or through web services) on the Internet

Hardware resources

- Virtual hardware platforms that are being offered to users on which they can execute their software
- More difficult to grasp (the 'new stuff'?)

Examples

Software resources

- ConfMan is one of the pioneering conference management systems → you can still download and install it to run on your servers
- Nowadays people prefer to use conference management systems that run in the cloud, accessed through a web interface (EasyChair, EDAS, etc.)

- Often called Software-as-a-Service (SaaS)
- Typical examples: Gmail, Google Docs, salesforce.com (CRM)

Hardware resources

- Amazon S3 is a storage resource in the cloud available for any user (who pays for it!)

Computing as commodity

People often compare the emergence of Cloud Computing with the Electrical Power Network

- In the beginning of the industrial revolution each factory had its own power generation system
- Later on it became evident that it was much more efficient to build an Electrical Power Network and let users (factories but also citizens) connect to it → Electricity became commodity
- Something similar could also happen with computing → computing 'out of the wall'!

Cloud computing characteristics

- On demand self service
  Users determine on demand the resources to be deployed
- Ubiquitous network access
  Resources access anywhere through the Internet
- Location-independent resource polling (multitenant model)
  Many users sharing the actual physical resources from different locations
- Rapid elasticity
  Rapidly increase or decrease amount of resources depending on demand
- Pay per use

Cloud Computing technologies

- Virtualisation at different levels (processing and storage, server, applications, etc.) to map efficiently these virtual resources to the physical resources of the cloud infrastructure
- Powerful service management platform in order to manage all resources that can be virtualised in the cloud
  → normally provided via a web interface
- Policy-based automation
- Resources monitoring infrastructure

- Enabled by the commoditisation of bandwidth (and increased transmission speeds)

Some potential benefits

- Relieves the users for the burden of buying, installing, managing and maintaining hardware
- Allows a better (more efficient) use of resources, including space (real estate) and electrical power
  → Cloud computing is often presented in connection with green computing!

- Allows services to scale up and down quickly in order to follow the demand
  → typical example is a service that gets demand peaks in which case more resources should be allocated to this service, and these resources are released when the demand peaks are over

Some potential benefits

- Allows services to be offered more quickly to end-users
  → spare the time to buy, install and configure hardware systems (often two to three months)
- Self-service activities (managing servers and software, allocating processing power and memory, change software versions, stop and restart servers, etc.) can be performed in a couple of minutes and 24 hours a day

Is Cloud Computing really new?

- Nothing is really new nowadays...
- Cloud computing is based on the principles of time-sharing that have been used since the beginning of computing
  - In the beginning of computing we didn't have personal computers
    → first they didn't exist and later we couldn't afford them!
  - The solution was to share mainframes!
- Cloud computing relies on good old virtualisation techniques, but they are being used in scales and with an elasticity never used before!
  - Now we are sharing physical resources (storages and processors) by means of virtualisation!

Convincing (?) example: the New York Times archives

See http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/

The New York Times

Problem

- The New York Times decided to make all the public domain articles from 1851-1922 available free of charge → 11 million articles as images in PDF format scanned from the original paper
- Generating a PDF version of an article takes quite a bit of work → scaling and gluing together bits of TIFF images
- PDFs could be generated dynamically on demand, but they decided to pre-generate all the articles and statically serve them

Convincing (?) example: the New York Times archives

Solution

- Upload 4TB of source data into the Amazon S3 storage, write some code that would run on numerous EC2 instances to read the source data, create PDFs, and store the results back into S3
- Use S3 to serve the PDFs to the general public

Result

- PDF generation algorithm executed in parallel in multiple machines
- 11 million articles were all processed in just under 24 hours using 100 Amazon Elastic Cloud (EC2) instances (actually done twice!)
- It would have been nearly impossible (or extremely expensive) to do this on a single data center!

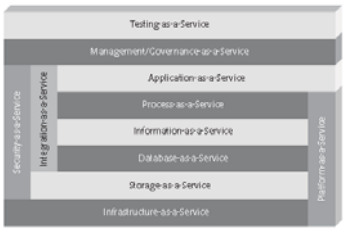
Some drawbacks of cloud computing

- Security (always security to spoil the fun...)
  How to be sure evil people will not mess up your system while it is running in the cloud?
- Control
  Putting stuff in the cloud means that you have to trust the cloud provider. What if it goes bankrupt, is taken over or goes mad?
- Cost
  Mainly porting costs, but anyone considering using cloud computing should ' do the math'
- Openness
  Danger of lock-in to a cloud solution

Drawbacks are also opportunities!

Cloud Computing 'architecture'

Organised in terms of 'layers' of X-as-a-service [Linthicum2009]



X-as-a-Service

Storage-as-a-Service
Rudimentary support for storing data in a storage in the network (example: Amazon S3)

Database-as-a-Service
Cloud provider supports a database and cloud user defines the tables and stores the data accordingly

Information-as-a-Service
Information is provided to the cloud user via APIs (normally via web services) to be used in the user's application (examples: geonames.org, freedb)

X-as-a-Service

Process-as-a-Service
Cloud provider allows the user to define and deploy a (workflow) process in which different services are coordinated to support some business process; process runs in the systems of the cloud provider

Application-as-a-Service
Same as Software-as-a-Service; cloud provider supports some application that can be accessed by the user via a web browser; APIs may also be defined to access the application e.g. through web services (examples: Gmail, Google Docs, salesforce.com)

X-as-a-Service

Platform-as-a-Service
Cloud provider offers a development and deployment platform that can be used to develop and deploy application (examples: Google Apps and Force.com)

Security-as-a-Service
Core security services on demand; typically identity management (for single sign-on, like, for example, openID)

Integration-as-a-Service
Complete enterprise architecture integration stack running from the cloud (data transformation, routing, interfacing and logging)

X-as-a-Service

Management/Governance-as-a-Service
Management of cloud services (topology, resource utilisation, virtualisation, uptime)
Enforcement of policies on data and services

Testing-as-a-Service
Cloud providers host test systems that test services or web applications remotely

Infrastructure-as-a-Service
Complete data centre facilities, combining storage, database, governance, management and platform-as-a-service in a single abstraction (example: Amazon EC2)

Cloud Computing 'architecture'

- Different cloud computing developments are often difficult to compare and combine
  - For example, how to compare or combine Application-as-a-Service (like Gmail) with Storage-as-a-Service (like Amazon S3)?
- What appears to be a reference model (or an architecture) is actually an attempt to classify and relate these forms of cloud computing!
  - Cloud computing initiatives in adjacent layers do not necessarily interoperate!

Cloud types

Public clouds
Clouds open to different enterprises in a pay-per-use model

Private clouds
Clouds operated in an enterprise datacenter or in a separate location, but for the benefit and under control of a single enterprise

Hybrid clouds
Combinations of public and private clouds
→ model being used for example by banks nowadays!

Amazon web services (AWS)

Collection of web services that offer elastic virtualisation of resources (storage and processing) on demand

- Amazon Simple Storage Service (S3)
- Amazon Elastic Cloud Compute (EC2)
- Amazon Simple Queue Service (SQS)
- Amazon CloudFront
- Amazon Simple DB
- ...

Amazon S3

- Storage service that can be accessed real-time via web services
- Allows one to store objects from 1 byte to 5 GB
- Flat namespace → set of buckets with objects in it
- Particularly useful for backup
- Not comparable to a remote drive → no directory hierarchies!

- It has high durability (data does not get lost), but relatively low speed and medium availability / reliability (it goes out of the air sometimes!)

Amazon S3

Functionality
- Find buckets and objects
- Discover their metadata
- Create new buckets
- Upload new objects
- Delete existing buckets and objects

Example of access through the `s3cmd` command line tool

```
s3cmd mb s3://com.imaginary.movies
s3cmd put home_movie.mp4
  s3://com.imaginary.movies/home_movie.mp4
```

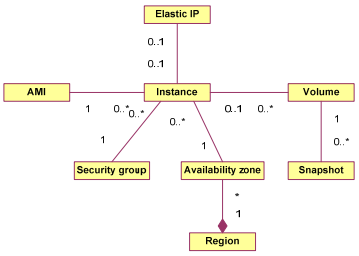

Amazon EC2

- Allows its users to create, deploy, manage and destroy servers ('instances')
- New instance can be created (launched) from a so called Amazon Machine Image (AMI)
- Offers ephemeral and persistent storages, but instances may also use S3 for persistent storage
- Allows the control of firewalls and execution of rules based on security groups

Amazon EC2 concepts

- AMI: image file containing the binary code to be deployed
- Instance: virtual server running at some point in time
- Elastic IP address: static IP address assigned to access the instance
- Region: group of availability zones
  Three regions are currently available US East, US West and Europe
- Availability zones: zones in the regions that do not share points of failure (necessary for robustness!)
- Security group: looks like a network segment governed by a firewall
- Block storage volume: block-level storage that can be mounted by an instance (similar to a Storage Area Network)
- Snapshot (of a volume): copy of the volume contents for backup

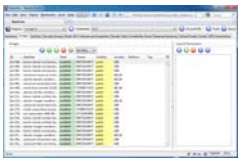
Amazon EC2 concepts

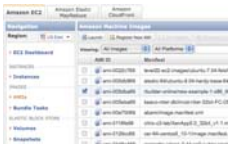


Amazon EC2 access

EC2 can be accessed in three different ways, through

- Amazon web services console
- Elasticfox Firefox plugin
- Amazon command line tools





Amazon EC2 usage example

- In order to launch an instance first an AMI needs to be chosen
- There are many AMIs available, containing different software configurations
- Chosen AMI has to be stored in S3 first
- Instance is launched from an AMI stored in S3

Example: AMI available from Amazon

```
IMAGE ami-225fba4b ec2-public-images/fedora-core4-apache-
   mysqlv1.07.manifest.xml amazon available public i386
   machine
```

image state
image kind
image id
image owner
target architecture
image manifest file (metadata)

Amazon EC2 usage example

- Whenever an instance is running an account in the instance is necessary in order to access it
- This is prepared beforehand to avoid having the password of the account literally written in a file inside the instance (schema with private and public keys)
- Once the user has an account in the instance he can manage it, for example, by using SSH to login to the instance

Amazon EC2 instance

Example: Instance created from the AMI discussed before

```
INSTANCE i-b1a21bd8 ami-1fd73376 ec2-75-101-201-11.compute-
   1.amazonaws.com domU-12-31-38-00-9D-44.compute-1.internal
   running 0 m1.small 2008-08-11T14:39:09+0000 us-east-1c
   aki-a72cf9ce ari-a52cf9cc
```

instance id
image id
cloud internal IP address (URL)
instance type: defines RAM, disk space and CPU of the virtual server
elastic IP address (actually URL)
region

Amazon EC2 data storage

Types of storage

- Persistent cloud storage → supported by S3
- Ephemeral instance storage → lifespan of the instance
- Elastic Block Storage (EBS)→ allows one to mount from 1GB to 1TB from a single instance

Once an instance is running you can create an EBS and mount it to be used by this instance (as an ext3 drive)

Anti-climax

- Larry Ellisson's (Oracle's CEO) view on Cloud computing

  http://www.youtube.com/watch?v=8UYa6gQC14o

Cloud Computing challenges

- Interoperability at the different layers between different providers
- Portability of deployed services
- Composition support
  - For example, impossible to have single sign-on accross Amazon, Google and Apple Store
- Security (safety, trust, etc.) issues

- Consequences of lack of standards!

Conclusions and questions

- We have at least to know what it is about
- How does Cloud Computing influence the development process of software applications?
  - It influences for sure deployment, but also testing and mainly the monitoring of non-functional properties
- What is the research agenda for Cloud Computing?

References

- D.S. Linthicum. Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-Step Guide
- G. Reese. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud
- IBM. Seeding the Clouds: Key Infrastructure Elements for Cloud Computing
- M. Naghshineh. Cloud Computing: Incremental or Transformational? Keynote presentation at ECOWS 2009
- T. Singh. Keynote on Cloud computing at ICSOC 2009