COMPARATIVE EVALUATION OF CLOUD COMPUTING ENVIRONMENTS

Prof. Dr. Andreas Polze
Hasso-Plattner-Institute for Software Engineering at University Potsdam

Windows Core Workshop, April 15/16 2010, Beijing
Agenda

A Comparison of Cloud Computing Environments

- **Historical Perspective**
  - Grid Computing, Utility Computing, Autonomic Computing

- **A Classification of Cloud Implementations**
  - Infrastructure as a Service (IaaS)
    - Amazon Web Services, VMware vCloud
  - Platform as a Service (PaaS)
    - Google AppEngine, Windows Azure platform
  - Software as a Service (SaaS)
    - Salesforce.com, Microsoft Office Live, Google Apps

- Which Cloud to choose – Future Directions
Cloud Computing

- A new business model for IT services.
- Promises to cut operational and capital cost.
- Builds upon utility computing, grid computing, and autonomic computing.
- No longer assumes that developers and users are aware of the provisioning and management infrastructure.

Dynamic infrastructure, pay-as-you-go, may “shrink”
Grid Computing

- Pooling compute resources
  - For scientific computing and simulation
  - from multiple administrative domains

- Virtual Supercomputer, Virtual Organizations
  - SPMD - Same Program Multiple Data
  - coarse grained parallel applications

- Examples
  - SETI@HOME
  - Berkeley Open Infrastructure for Network Computing (BOINC)
  - Globus toolkit (GTK)

- Standardization efforts at Open Grid Forum (OGF)
Grid vs. Cloud

Grid never considered the interactive end user
- jobs are being submitted through portals
- results are being retrieved hours later in form of log files and collected program output.

Grid did not integrate administrative domains.
- install certificates, enable ssh login, and setting up mutual trust relationships across all partners

Creating a huge, distributed virtual supercomputer,
- which is managed in a batch-job-processing scheme.
Utility Computing

- Packaging of IT resources into a metered service
  - CPU, memory, network bandwidth, storage
  - similar to traditional utilities (such as the telephone network).
- Low initial costs,
  - pay-per-use billing model
  - quick reaction to changes in demand of IT services
- Not a new concept
  - rather long history in the world of ultra-reliable but expensive mainframe computers
  - a model for capacity-based charging
Autonomic Computing

- Operator does not control the system directly
  - policies and rules serve as input for self-management
  - Notion was coined by IBM

- Self-* functional areas:
  - Self-Configuration: Automatic configuration of components;
  - Self-Healing: Automatic discovery, and correction of faults;
  - Self-Optimization: Automatic monitoring and control of resources
  - Self-Protection: Protection from arbitrary attacks

- Cloud computing infrastructures (often called "the fabric") have to follow autonomic computing principles
Agenda

A Comparison of Cloud Computing Environments

- Historical Perspective
  - Grid Computing, Utility Computing, Autonomic Computing

- A Classification of Cloud Implementations
  - Infrastructure as a Services (IaaS)
    - Amazon Web Services, VMware vCloud
  - Platform as a Service (PaaS)
    - Google AppEngine, Windows Azure platform
  - Software as a Service (SaaS)
    - Salesforce.com, Microsoft Office Live, Google Apps

- Which Cloud to choose – Future Directions
Cloud Computing

...is a computing paradigm where the boundaries of computing will be determined by economic rationale rather than technical limits

The concept generally incorporates combinations of:

• infrastructure as a service (IaaS)
• platform as a service (PaaS)
• software as a service (SaaS)
Cloud Computing – the three layers

Challenges to the Cloud

- Has to abstract underlying hardware
- Be elastic in scaling to demand
- Build on a pay per use basis
Cloud Computing – status quo

Private vs. Public cloud – or both?
- Private: IBM, SAP
- Public: Amazon, Google, Microsoft
- Both: VMware, Salesforce's force.com

Unit of granularity
- Amazon Machine Image
- VMware Virtual Appliance .... Virtual Datacenter
- Web Role, Server Role (MS)
- Physical Machine (IBMs RC2)
Cloud Computing – status quo (contd.)

- Programming models
  - WebRole, WebService in .NET
  - Native C++ program
  - J2EE (EJB)
  - vCloud API

- Interaction and Communication
  - Service bus
  - Message passing APIs
  - Virtual layer 2 network connectivity
  - Security / Trust
open cloud manifesto

dedicated to the belief that the cloud should be open

view the manifesto!

sign up to be added to the supporters list
Agenda

A Comparison of Cloud Computing Environments

- Historical Perspective
  - Grid Computing, Utility Computing, Autonomic Computing

- A Classification of Cloud Implementations
  - **Infrastructure as a Services (IaaS)**
    - Amazon Web Services, VMware vCloud
  - **Platform as a Service (PaaS)**
    - Google AppEngine, Windows Azure platform
  - **Software as a Service (SaaS)**
    - Salesforce.com, Microsoft Office Live, Google Apps

- Which Cloud to choose – Future Directions
Amazon Web Services (AWS)

- Since August 2006
- compute power, storage, and other services
- pay-per-use basis, no up-front expenses

Elastic Compute Cloud (Amazon EC2)
- Infrastructure-as-a-Service (IaaS)
- Amazon Machine Instance (AMI)
  - Linux, Windows, OpenSolaris…
  - Stored in S3 (Simple Storage System)
- Three standard types of compute instances
  - small (1 virtual core with 1 EC2 CU, 32 bit, 1.7 GB memory),
  - medium (2 virtual cores with 2 EC2 CU each, 64 bit, 7.5 GB memory)
  - extra large ((4 virtual cores with 2 EC2 CU each, 64 bit, 15 GB memory).

EC2 compute unit == 1.0-1.2 GHz 2007 Opteron or 2007 Xeon
S3 – Simple Storage Service

- Simple web services interface (REST, SOAP, BitTorrent)
- Data on S3 is unstructured (blobs)
  - no interface to run structured queries on S3 directly from client
  - write, read, and delete objects from 1 byte to 5 GB
    - number of objects a client can store is unlimited.
    - object is stored in a bucket and retrieved via a unique, developer-assigned key.
  - Bucket can be located in the United States or in Europe.
- Authentication mechanisms are provided
  - Objects can be made private or public,
  - rights can be granted to specific users.
Simple Queue Service (SQS)

- Reliable, highly scalable, hosted queue
- Move data between application components
  - Without losing messages or requiring components to be online
  - Foundation for building workflows
- A queue can be created in the US or in Europe.
  - Disjunct namespaces
  - Message body can contain up to 8 KB of text in any format.
  - Messages can be retained in queues for up to 4 days.
  - Messages can be sent and read simultaneously.
- On retrieval, messages are “locked” while being processed
- EC2 configuration rules for can be based on SQS attributes
Additional services

- SimpleDB adds additional layer on S3 to provide query interface
- CloudFront for hosting downloadable user interfaces
  - in form of a collection of JavaScript/Ajax code
- Virtual Private Cloud (Amazon VPC)
  - extension of the initial, purely public cloud
  - provides a secure and seamless bridge between a company’s existing IT infrastructure and the AWS cloud
- AWS tools
  - managing AMIs for EC2 and setting up SimpleDB configurations for S3
  - AWS Toolkit for Eclipse for common development workflows (Java)
  - Wrapper libraries for .NET, Perl, PHP, Ruby and Python
- AWS service level agreement (SLA)
  - Monthly Uptime of at least 99.9%, service credits as compensation
The Cloud – Amazon EC2, S3, SQS
Self-service, Prorated Super Computing Fun!

By DEREK GOTTFRID
TAGS: AWS, EC2, HADOOP, MAPREDUCE, S3

As part of eliminating TimeSelect, The New York Times has decided to make all the public domain articles from 1851-1922 available free of charge. These articles are all in the form of images scanned from the original paper. In fact from 1851-1980, all 11 million articles are available as images in PDF format. To generate a PDF version of the article takes quite a bit of work — each article is actually composed of numerous smaller TIFF images that...
*Loaded newspapers from 1851-1922 into S3*

*100 EC2 instances used to process the data*

*Created the full application in 5 days*

Source: C.Bell, Amazon.com, 2007
VMware’s vCloud initiative

- Provide SW so that anyone can easily create cloud:
  - e.g., SP, University, Enterprise
- Supports all applications
- Controlled programmatically:
  - From VMs, from capacity owner, from portal
  - Can be accessed from VI Client
- Can scale up / scale down
- Simple self service experience.
- Broad partner and research collaboration:
  - Researchers can replace any part of the service.
  - Researchers can replace the entire implementation and clone the API

Source: O. Krieger, VMware, 2009
vCloud API

- Build internal clouds based on VMware’s technology stack
  - Service providers can build **Infrastructure-as-a-Service** portals
  - ISVs can package existing or new applications, for internal or external clouds

vCloud workloads are virtual appliances (vApps)

- pre-built software solutions optimized for the cloud,
  - consisting of multiple virtual machines that are bundled
- upload, download, instantiation, deployment and operation of vApps.

vApps can be comprised of multiple virtual machine images

- can be combined with virtual networking resources, effectively enabling layer-2 connectivity across virtual machines.
Agenda

A Comparison of Cloud Computing Environments

- Historical Perspective
  - Grid Computing, Utility Computing, Autonomic Computing

- A Classification of Cloud Implementations
  - Infrastructure as a Services (IaaS)
    - Amazon Web Services, VMware vCloud
  - Platform as a Service (PaaS)
    - Google AppEngine, Windows Azure platform
  - Software as a Service (SaaS)
    - Salesforce.com, Microsoft Office Live, Google Apps

- Which Cloud to choose – Future Directions
Google AppEngine

- Run web applications on Google's infrastructure
  - Build your own SaaS offerings

AppEngine features:
- Dynamic web serving
- Persistent storage with queries, sorting and transactions
- Automatic scaling and load balancing
- APIs for authenticating users and sending email
- A fully featured local development environment that simulates Google App Engine on a client computer
- Task queues for work outside of the scope of a web request
- Scheduled tasks for triggering events
Two runtime environments...

- Java and Python environments
- Apps accessible through
  - client's domain name (using Google Apps)
  - Google's reserved domain called.appspot.com
- Multiple languages and technologies
  - Java (JVM, Java servlets), JavaScript and Ruby
  - Fast Python interpreter and the Python standard library
- Applications run in a secure environment that provides limited access to the underlying operating system.
AppEngine sandboxing:

- Connection to other computers on the Internet:
  - URL fetch and email services.
  - HTTP (or HTTPS) requests on the standard ports.
- An application cannot write to the file system.
  - An app can read files (application code)
  - Must use App Engine datastore, memcache for persistency
- Application code only runs …
  - in response to web request, queued task, or scheduled task,
  - must return response data within 30 seconds in any case.
  - cannot spawn a sub-process or execute after response was sent
- App Engine costs nothing to get started
AppEngine Development Workflow

- Java and Python SDKs…
  - Include emulator of App Engine services on local computer
  - SDK includes all APIs and libraries available on App Engine
  - Simulates secure sandbox environment
- SDK also includes tool to upload apps to AppEngine.
- Java SDK for Java 5 or Java 6
- Google Plugin for Eclipse
  - can be used to create, test and upload App Engine applications
  - command-line tools for running local server and uploading apps
- Python SDK is implemented in pure Python 2.5
The Internet “Cloud”

Windows Azure

Windows Live
Microsoft Office Live
XBOX LIVE
Other Experiences from Microsoft & Others

Windows
Microsoft Office
XBOX

Microsoft-based Devices
Other Devices

© Andreas Polze - HPI / Univ. Potsdam, Germany, April 2010
Windows Azure components

- **Windows Azure** is a Windows-based scalable environment for compute, storage, hosting, and management capabilities.
  - links to on-premises applications with secure connectivity, messaging, and identity management.

- **SQL Azure** is a full relational database in the cloud.
  - Implementation is based on SQL server.

- **AppFabric** provides Network Services for the Cloud.
  - Offers identity management and firewall friendly messaging.
  - Messaging between on-premises IT applications and cloud-based services.
  - AppFabric integrates the previously known .NET services.
Windows Azure Platform
(Azure Services Platform)

Cloud Applications

SQL Azure

.NET Services

On-Premises Applications

Windows

Others

Windows Azure

© Andreas Polze - HPI / Univ. Potsdam, Germany, April 2010
Windows Azure

- Scalable, virtualized hosting environment
- Flexible storage with blobs, tables, and queues
- Model-driven service lifecycle management
- Rich local & offline developer experience
- Seamless programming interface in Visual Studio / .NET

Azure Services Platform

- SQL Services
- .NET Services
- Live Services
- Microsoft SharePoint Services
- Microsoft Dynamics CRM Services
Azure Compute Services

- Internet
- Load Balancer
- Web Site (ASPX, WCF)
- Storage
  - Queues
  - Tables
  - Blobs
  - Windows Azure
- Worker Service

© Andreas Polze - HPI / Univ. Potsdam, Germany, April 2010
Windows Azure Storage Service
A closer look

HTTP/HTTPS

Blobs

Tables

Queues
SQL Azure

(SQL Services)
.NET Services

.NET Service Bus

.NET Access Control Service

© Andreas Polze - HPI / Univ. Potsdam, Germany, April 2010
Simulates Windows Azure on the user's local computer.

- includes local storage services that simulate the Blob, Queue, and Table services available in Windows Azure.
- development storage UI provides a means to view the status of the local storage services and to start, stop, and reset them.

The development fabric UI allows to

- view service deployments and role instances,
- start and stop a service, and
- test logging levels.

CSPack tool prepares a service for deployment,
- either to the development fabric or to the Windows Azure fabric.

CSRun runs a service in the development fabric.
Agenda

A Comparison of Cloud Computing Environments

- Historical Perspective
  - Grid Computing, Utility Computing, Autonomic Computing

- A Classification of Cloud Implementations
  - Infrastructure as a Services (IaaS)
    - Amazon Web Services, VMware vCloud
  - Platform as a Service (PaaS)
    - Google AppEngine, Windows Azure platform
  - Software as a Service (SaaS)
    - Salesforce.com, Microsoft Office Live, Google Apps

- Which Cloud to choose – Future Directions
Salesforce.com – SaaS pioneer

- Customer Relationship Management (CRM)
  - availability of 99.9% for cloud infrastructure
  - Sales, Service & Support, Partner Relationship Management, Marketing, Content, Ideas and Analytics modules
  - Customization on the platform- or the tab-level

- AppExchange
  - Directory of third-party apps for Salesforce
  - over 800 apps from 450 ISV

- Web Services API
  - enables integration with other systems
  - wrapper libraries for Java, and .NET
Force.com – PaaS

- Opening SalesForce.com infrastructure for third-party
- Apex and Visualforce
  - Force.com IDE with testrunner – local fabric
    - proprietary Java-like language for Force.com
  - Adobe Flash Builder for Force.com
    - user interfaces in HTML, AJAX or Flex
  - Visualforce relies on the concept of mashups for integration of user-interface components on the client side
- Force.com persistency layer (owns the data…)
  - Object-based data store
  - ERP connectors for Oracle, SAP
Microsoft Office Live - SaaS

- Internet-based services for consumers and SMEs
- Office Live Workspace and Office Live Small Business
  - "Software + Services" philosophy coined by Microsoft
  - relies on applications installed locally on client computers
  - Silverlight plugin for rich user-interface
- No online collaboration
  - documents are "checked out" and "checked in"
  - Office Live integration with SharedView for for real-time screen sharing as a means of online collaboration
Synchronizing life

Sync, share, and access the information you care about—wherever you happen to be.

Sign in

What is Live Mesh?
Discover how Live Mesh brings your digital world together.

What's inside?
Take a look at the features in Live Mesh.

Get started
A quick guide to help you get your mesh rolling.
Google Apps - SaaS

Web applications with office functionality
- Accessible as docs.google.com for a single user
- Configurable to use client’s domain name with the service

Tool categories:
- communication tools
  - Gmail, Google Talk, and Google Calendar
- productivity tools
  - Google Docs: text files, spreadsheets, and presentations
- Frontend
  - customizable start page (iGoogle),
  - Google Sites (to develop web pages)

Access rights based on Google Accounts
Agenda

A Comparison of Cloud Computing Environments

- Historical Perspective
  - Grid Computing, Utility Computing, Autonomic Computing

- A Classification of Cloud Implementations
  - Infrastructure as a Services (IaaS)
    - Amazon Web Services, VMware vCloud
  - Platform as a Service (PaaS)
    - Google AppEngine, Windows Azure platform
  - Software as a Service (SaaS)
    - Salesforce.com, Microsoft Office Live, Google Apps

- Which Cloud to choose – Future Directions
Cloud Computing Building Blocks

Virtualization
- dynamically increase and decrease computing capacity
- can be achieved on many different layers of abstraction
  - application server, operating system processes,
  - virtual machines, logical partitions of physical hardware
- For cloud computing - mostly logical machine level of abstraction

WebServices
- access to cloud resources via WebService APIs
- APIs typically are hidden behind wrapper libraries in
  - Java, C#, C++ (and other .NET languages) for the client
  - "web languages" such as Python, Ruby, and PHP.
Orchestration / Service Bus
- Compute services, data stores, and messaging middleware.
- Platforms differ with respect to their support for integrating external services
  - hosted on premise or at partner sites

Clients-side user interfaces
- Cloud computing apps need to expose a client-side UI
- Portals integrate mashups in a web browser
  - Simple ser interfaces often based on Ajax and JavaScript.
  - Trend to fully fledged component model such as JavaBeans/Applets or Silverlight/.NET
  - Downloadable and can be executed on the client's computer
Why is virtualization important? (Problems with Cloud Computing...)

- Server consolidation
  - Virtualizing a number of under-utilized systems on a single server may yield distinct savings in power, space, cooling, and administration
  - Live migration may deal with varying server utilization

- Virtualization for developers
  - Run multiple OS instances, crash independently
  - Ease kernel debugging (similar to debugging user-space apps)
Virtualization problems

- Security: extended attack surface
- Virtualization-based malware
- Must trust hypervisor

Intel VT-x, AMD Pacifica
Blue Pill Idea (simplified)

- Native Operating System
  - CALL bluepill
  - Native Operating System continues to execute, but inside Virtual Machine this time...
  - RET

- PROC bluepill
  - enable SVM
  - prepare VMCB
  - VMRUN
    - check VMCB.exitcode
    - Blue Pill Hypervisor
      - only during first call
      - RET from bluepill PROC, never reached in host mode, only executed once in guest mode
      - VMCB
        - RIP

source: J. Rutkowska, Black Hat USA 2006, © Black Hat
Various VMM bugs

Tavis Ormandy, Google:

Presents methodology used to find multiple bugs in several various VMMs:
- VMWare, XEN, Bochs, MS Virtual PC, Parallels

mostly fuzzing-based methods used to test
- Instruction parsing
- I/O Device emulation

Most of the bugs found classified as DoS
Programming Models #1: OpenCL, CUDA

OpenCL – Open Computing Language
CUDA – Compute Unified Device Architecture

Open standard for portable, parallel programming of heterogeneous parallel computing CPUs, GPUs, and other processors
OpenCL Design Goals

- Use all computational resources in system
  - Program GPUs, CPUs, and other processors as peers
  - Support both data- and task-parallel compute models
- Efficient C-based parallel programming model
  - Abstract the specifics of underlying hardware
- Abstraction is low-level, high-performance but device-portable
  - Approachable – but primarily targeted at expert developers
  - Ecosystem foundation – no middleware or “convenience” functions
- Implementable on a range of embedded, desktop, and server systems
  - HPC, desktop, and handheld profiles in one specification
- Drive future hardware requirements
  - Floating point precision requirements
  - Applicable to both consumer and HPC applications
OpenCL Platform Model

- One Host + one or more Compute Devices
  - Each Compute Device is composed of one or more Compute Units
  - Each Compute Unit is further divided into one or more Processing Elements
Programming Models #2: Intel’s Ct technology

- Ct adds new data types (parallel vectors) & operators to C++
  - Library-like interface and is fully ANSI/ISO-compliant
- Ct abstracts away architectural details
  - Vector ISA width / Core count / Memory model / Cache sizes
- Fully leverage deterministic parallel programming models
  - I.e. Make data races impossible
- Expresses complex behaviors through simple operators
- Presents a simple and predictable performance model
- Provides a forward-scaling programming model

- Is a project being worked on at Intel since some time already
Ct: Nested Data Parallelism in C/C++

Physics, Image, Video, Signal Processing, ...

C++

Ct-based Parallel Data Types

1

Compile once and forward scale

C/C++ Compiler

C/C++ libs

Ct Runtime

Multi-Core-Multi_Threading Challenge
Conclusions

- New programming models on the horizon – think cloud
  - Hybrid Clouds will be the future

- The MultiThreading MultiCore Challenge
  - Must learn to write parallel programs (OpenMP, MPI, etc.)
  - Parallel computing on a chip

- Monitoring the cloud
  - With respect to resource usage, security, SLA
  - Must include the virtualization layer
  - Green Cloud