**Topic: Above the clouds**

**Executive Summary:**

Cloud Computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services. The services themselves have long been referred to as Software as a Service (SaaS). The datacenter hardware and software is what we will call a Cloud. When a Cloud is made available in a pay-as-you-go manner to the general public, we call it a Public Cloud; the service being sold is Utility Computing. We use the term Private Cloud to refer to internal datacenters of a business or other organization, not made available to the general public. Thus, Cloud Computing is the sum of SaaS and Utility Computing, but does not include Private Clouds. People can be users or providers of SaaS, or users or providers of Utility Computing. We focus on SaaS Providers (Cloud Users) and Cloud Providers, which have received less attention than SaaS Users

New features in hardware involved in Cloud Computing:

* The illusion of infinite computing resources available on demand
* The elimination of an up-front commitment by Cloud users
* The ability to pay for use of computing resources on a short-term basis as needed

Any application needs a *model of computation, a model of storage, and a model of communication*. The statistical multiplexing necessary to achieve elasticity and the illusion of infinite capacity requires each of these resources to be virtualized to hide the implementation of how they are multiplexed and shared. Different utility computing offerings will be distinguished based on the level of abstraction presented to the programmer and the level of management of the resources

**Obstacles and Opportunities for Cloud Computing:**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Obstacle Category** | **Obstacle** | **Opportunity** |
| 1 | Adoption | Availability of Service | Use multiple cloud providers; use elasticity to prevent DDOS |
| 2 | Adoption | Data Lock -In | Standardize API; compatible SW to enable surge computing |
| 3 | Adoption | Data Confidentiality and Audibility | Deploy encryption; VLANs firewall; Geographical Data storage |
| 4 | Growth | Data Transfer bottleneck | Fedexing Disks: Data back-up/archival; Higher BW switches |
| 5 | Growth | Performance Unpredictability | Improve VM support; Flash memory; Gang Schedule VMs |
| 6 | Growth | Scalable Storage | Invent scalable store |
| 7 | Growth | Bugs in large distributed Systems | Invent Debugger that relies on distributed VMs |
| 8 | Growth | Scaling quickly | Invent auto scaler that relies on ML; snapshots for conversation |
| 9 | Policy and Business | Reputation Fate Sharing | Offer reputation guarding services like those for email |
| 10 | Policy and Business | Software-licensing | Pay for use licenses; bulk use sales |

**New technology and business trends:**

Emergence of web2.0 companies have changed the customer interaction significantly. Individuals can deal with cloud provider directly rather than choosing a content distribution provider. Amazon capitalized on this idea and has established a strong foothold in the cloud market.

**New Application Opportunities:**

Mobile Interactive applications, Parallel batch processing, rise of analytics, extension of compute intensive desktop applications, earthbound applications have a huge potential for growth if moved to cloud computing as they meet the required demands for resources and maintenance.

**Classes of Utility Computing:**

**Amazon EC2:** Hardware is provided; software stack is built by cloud users. Makes scalability difficult, as replication and state-management issues are dependent on applications.

**Google AppEngine:** Designed exclusively for web applications; has a stateless computation tier and stateful storage tier.

**Microsoft Azure:** Applications are written on Common Language Routine, a language-independent managed environment. Provide a mix of EC2 hardware and AppEngine application framework.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Amazon EC2** | **Microsoft Azure** | **Google App Engine** |
| **Computation Tier** | X86 Instruction set Architecture via Xen VM | Microsoft Common Language Routine VM | Predefined application structure and framework; programmer-provided “handlers” written in python, all in persistent state store in Megastore |
| **Storage Tier** | Block store(EBS), simple DB | SQL Data service, Azure Storage service | MegaStore/Big Table |
| **Network Tier** | Declarative specification of IP level technology | Automatic based on programmer’s declarative descriptions of app components(roles) | Fixed topology to accommodate 3-tier web app structure |

**Cloud Computing Economics:**

**Elasticity: Shifting the risk**

Elasticity is valuable to established companies as well as startups. For example, Target, the nation’s second largest retailer, uses AWS for the Target.com website. While other retailers had severe performance problems and intermittent unavailability on “Black Friday” (November 28), Target’s and Amazon’s sites were just slower by about 50% over the last 2.5 years. Even less-dramatic cases suffice to illustrate this key benefit of Cloud Computing: the risk of under-estimating workload is shifted from the service operator to the cloud vendor. The cloud vendor may charge a premium (reflected as a higher use cost per server-hour compared to the 3-year purchase cost) for assuming this risk.

Two Benefits from Elasticity:

* Unexpected scaling down carries a financial penalty for the users.
* Technology trend shows, prices will drop and new hardware and software technologies will be available. This gives an opportunity for cloud users to benefit from the reduced hardware pricing.

**Comparing costs:**

**Pay separately per resource:**  Most applications do not make equal use of storage, computation and network. Pay separately per resource, saves costs paid for minimum used resources.

**Power, cooling, physical plant costs:** Considering the operating costs for resources like electricity, plant set up, requires high investment and maintenance. Going cloud vendor saves investment charges significantly.

**Operation Costs:**

Hardware server reboots are easy and requires minimum staffing, however, moving to cloud saves costs spent on these factors. Although, migration of legacy enterprise application may require significant manual effort, it will be a one-time effort.

**Some Obstacles discussed:**

1. **Availability of Service:** Server outages cost business heavily. Users must plan to have multiple cloud services to ensure business is up and running continuously. Attacks from bots that cause Distributed Denial of Service, require extra system resources, a cloud vendor can manage these issues comfortably (using elasticity).
2. **Performance Unpredictability:**  Sharing I/O among virtual machines is actually a serious concern. Improved system architectures and use of flash memory, that use less I/O but provides high performance results are viable solutions to this problem. Today’s VMs and Oss do not provide an opportunity to ensure if all threads are running simultaneously. Hence a viable option would be to go for “gang scheduling” in cloud computing.
3. **Software Licensing:** Current Software licenses restrict the computers on which the software can run. Commercial software companies have to monitor their strategy with software license to better fit to cloud vendors. Since pay as you use software would be a difficult and tasking strategy for commercial companies, better option is to leave cloud vendors provide prepaid plans for bulk use that can be sold at discount.