Introduction to Cloud Computing

CERN Cloud 50-60K Xen and KVM, Open Nebula MetaScheduler

Large Hadron Collider at CERN Maximillien Brice
Slide Credits

http://net.pku.edu.cn/~course/cs402/2009/
彭波
pb@net.pku.edu.cn
北京大学信息科学技术学院
6/30/2009

Steve Hanna, Juniper Networks, 2009
Foster, Zhao, Raicu, Lu (this week’s paper)
Cloud Computing is...

- Dynamically scalable shared resources
- Accessed everywhere via the Internet
- Large-scale data processing
- Shared resources with other customers: storage, computing, services, data
- Appeal for startups
  - Cost efficiency: pay for what you use
  - Software as platform
Cloud Computing is...

“...characterized by economies of scale, in which a pool of abstracted, virtualized, dynamically scalable, manage computing power, storage, platforms, and services are delivered on demand to external customers over the Internet.”

{Foster, Zhao, Raicu, Lu, U. Chicago, Argonne National Labs}
Clouds v. Grids

Figure 1: Grids and Clouds Overview
Key Characteristics of Clouds

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

- very large datacenters
- large-scale software infrastructure
- operational expertise
“pay-as-you-go” 好比让用户把电源插头插在墙上，你得到的电压和Microsoft得到的一样，只是你用得少，pay less；utility computing的目标就是让计算资源也具有这样的服务能力，用户可以使用500强公司所拥有的计算资源，只是use less pay less。这是cloud computing的一个重要方面。
Flavors of Cloud Computing

- **SaaS – Software as a Service**
  - Network-hosted application

- **DaaS – Data as a Service**
  - Customer queries against provider’s database

- **PaaS – Platform as a Service**
  - Network-hosted software development platform

- **IaaS – Infrastructure as a Service**
  - Provider hosts customer VMs or provides network storage

- **IPMaaS – Identity and Policy Management as a Service**
  - Provider manages identity and/or access control policy for customer

- **NaaS – Network as a Service**
  - Provider offers virtualized networks (e.g. VPNs)
Cloud Computing Providers

**DaaS**
- DataDirect
- STRIKEiron

**SaaS**
- Cisco Webex
- Salesforce on demand applications
- Microsoft Office Live Meeting
- Workday
- Google Apps
- Microsoft Office Live

**PaaS**
- force.com platform as a service
- Amazon Web Services
- IBM
- GridGain
- Windows Azure
- Facebook Developers

**IPMaaS**
- PingIdentity
- TRICIPHER

**IaaS (DC/server)**
- Verizon
- Telefonica
- Akamai
- AT&T
- Amazon Web Services
- Google
- CloudWorks
- Windows Azure

**IPM**
- Software & Data
- Infrastructure

**NaaS**
- Verizon Business
- Limelight Networks
- Verizon
- BT

Copyright © 2009 Juniper Networks, Inc.
Key Players

The Next Gen = Cloud Computing

- Amazon Web Services
- Google App Engine
- Microsoft Windows Azure
Amazon Web Services

- Amazon’s infrastructure (auto scaling, load balancing)
- Elastic Compute Cloud (EC2) – scalable virtual private server instances
- Simple Storage Service (S3)
- Simple Queue Service (SQS) – messaging
- SimpleDB - database
- Flexible Payments Service, Mechanical Turk, CloudFront, etc.
# Price of Amazon EC2

<table>
<thead>
<tr>
<th></th>
<th><strong>United States</strong></th>
<th><strong>Europe</strong></th>
<th><strong>Linux/UNIX Usage</strong></th>
<th><strong>Data Transfer In</strong></th>
<th><strong>Data Transfer Out</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standard On-Demand Instances</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Small (Default)</td>
<td></td>
<td></td>
<td>$0.10 per hour</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Large</td>
<td></td>
<td></td>
<td>$0.40 per hour</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extra Large</td>
<td></td>
<td></td>
<td>$0.80 per hour</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>High CPU On-Demand Instances</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td></td>
<td></td>
<td>$0.20 per hour</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extra Large</td>
<td></td>
<td></td>
<td>$0.80 per hour</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| **Data Transfer In** |                      |                      |
|----------------------|----------------------|
| All Data Transfer    | $0.10 per GB         |

| **Data Transfer Out** |                      |                      |
|-----------------------|----------------------|
| First 10 TB per Month | $0.17 per GB         |
| Next 40 TB per Month  | $0.13 per GB         |
| Next 100TB per Month  | $0.11 per GB         |
| Over 150 TB per Month | $0.10 per GB         |

- Small Instance (Default) 1.7 GB of memory, 1 EC2 Compute Unit (1 virtual core with 1 EC2 Compute Unit), 160 GB of instance storage, 32-bit platform
- Large Instance 7.5 GB of memory, 4 EC2 Compute Units (2 virtual cores with 2 EC2 Compute Units each), 8 GB of instance storage, 64-bit platform
- Extra Large Instance 15 GB of memory, 8 EC2 Compute Units (4 virtual cores with 2 EC2 Compute Units each), 1690 GB of instance storage, 64-bit platform
Example: Animoto

- **Animoto** is a video rendering & production house with service available over the Internet *(cloud customer)*
  - With their patent-pending technology and high-end motion design, each video is a fully customized orchestration of user-selected images and music in several formats, including DVD.

- **Animoto** is entirely hosted on cloud *(cloud provider)*

- **Released Facebook App**: users were able to easily render their photos into MTV like videos
  - Ramped from 25,000 users to 250,000 users in three days
  - Signing up 20,000 new users per hour at peak
  - Went from 50 to 3500 servers in 5 days
  - Two weeks later scaled back to 100 servers
  - [http://www.animoto.com](http://www.animoto.com)
Example: Eli Lilly

- Eli Lilly is the 10th largest pharmaceutical company in the world *(cloud customer)*

- Moved entire R&D environment to cloud *(cloud provider)*

- Results:
  - Reduced costs
  - Global access to R&D applications
  - Rapid transition due to VM hosting

  - Time to deliver new services greatly reduced:
    - New server: 7.5 weeks down to 3 minutes
    - New collaboration: 8 weeks down to 5 minutes
    - 64 node linux cluster: 12 weeks down to 5 minutes
Cloud Computing PROS and CONS

PROS
• Reduced costs, consumption-based costs
• Efficient resource sharing
• Management taken care of by cloud provider
• Fast time to roll out new services
• Dynamic resource availability for crunch periods

CONS
• Security and privacy
• Latency and BW considerations
• Absence of robust SLAs
• Compliance/regulatory laws mandate on-site data ownership
• Inter-operability, portability, lock-in
• Availability and reliability
# Challenges

Table 6: Top 10 Obstacles to and Opportunities for Adoption and Growth of Cloud Computing.

<table>
<thead>
<tr>
<th>Obstacle</th>
<th>Opportunity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Availability of Service</td>
<td>Use Multiple Cloud Providers to provide Business Continuity; Use Elasticity to Defend Against DDOS attacks</td>
</tr>
<tr>
<td>2 Data Lock-In</td>
<td>Standardize APIs; Make compatible software available to enable Surge Computing</td>
</tr>
<tr>
<td>3 Data Confidentiality and Auditability</td>
<td>Deploy Encryption, VLANs, and Firewalls; Accommodate National Laws via Geographical Data Storage</td>
</tr>
<tr>
<td>4 Data Transfer Bottlenecks</td>
<td>FedExing Disks; Data Backup/Archival; Lower WAN Router Costs; Higher Bandwidth LAN Switches</td>
</tr>
<tr>
<td>5 Performance Unpredictability</td>
<td>Improved Virtual Machine Support; Flash Memory; Gang Scheduling VMs for HPC apps</td>
</tr>
<tr>
<td>6 Scalable Storage</td>
<td>Invent Scalable Store</td>
</tr>
<tr>
<td>7 Bugs in Large-Scale Distributed Systems</td>
<td>Invent Debugger that relies on Distributed VMs</td>
</tr>
<tr>
<td>8 Scaling Quickly</td>
<td>Invent Auto-Scaler that relies on Machine Learning; Snapshots to encourage Cloud Computing Conservationism</td>
</tr>
<tr>
<td>9 Reputation Fate Sharing</td>
<td>Offer reputation-guarding services like those for email</td>
</tr>
<tr>
<td>10 Software Licensing</td>
<td>Pay-for-use licenses; Bulk use sales</td>
</tr>
</tbody>
</table>
The interesting thing about Cloud Computing is that we’ve redefined Cloud Computing to include everything that we already do. . . . I don’t understand what we would do differently in the light of Cloud Computing other than change the wording of some of our ads.

**Larry Ellison**, quoted in the Wall Street Journal, September 26, 2008

It’s stupidity. It’s worse than stupidity: it’s a marketing hype campaign. Somebody is saying this is inevitable — and whenever you hear somebody saying that, it’s very likely to be a set of businesses campaigning to make it true.

Cloud is here…

Google alone has 450,000 systems running across 20 datacenters, and Microsoft's Windows Live team is doubling the number of servers it uses every 14 months, which is faster than Moore's Law.

“Data Center is a Computer”
Parallelism everywhere
Massive Scalable Reliable Resource Management
Data Management
Programming Model & Tools
<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Particle Physics Large Hadron Collider (15PB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Human Genomics (7000PB)</td>
<td>1GB / person 200PB+ captured 200% CAGR</td>
<td></td>
</tr>
<tr>
<td>World Wide Web (~1PB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wikipedia (10GB)</td>
<td>100% CAGR</td>
<td></td>
</tr>
<tr>
<td>Annual Email Traffic, no spam (300PB+)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Internet Archive (1PB+)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Estimated On-line RAM in Google (8PB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Personal Digital Photos (1000PB+)</td>
<td>100% CAGR</td>
<td></td>
</tr>
<tr>
<td>200 of London’s Traffic Cams (8TB/day)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2004 Walmart Transaction DB (500TB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Typical Oil Company (350TB+)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Merck Bio Research DB (1.5TB/qtr)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UPMC Hospitals Imaging Data (500TB/yr)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MIT Babytalk Speech Experiment (1.4PB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Terashake Earthquake Model of LA Basin (1PB)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>One Day of Instant Messaging in 2002 (750GB)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Total digital data to be created this year **270,000PB** (IDC)
How much data?

- Internet archive has 2 PB of data + 20 TB/month
- Google processes 20 PB a day (2008)
- “all words ever spoken by human beings” ~ 5 EB
- CERN’s LHC will generate 10-15 PB a year
- Sanger anticipates 6 PB of data in 2009

640K ought to be enough for anybody.
NERSC User George Smoot wins 2006 Nobel Prize in Physics

Cosmic Microwave Background Radiation (CMB): an image of the universe at 400,000 years

Smoot and Mather 1992
COBE Experiment showed anisotropy of CMB
The Current CMB Map

- Unique imprint of primordial physics through the tiny anisotropies in temperature and polarization.
- Extracting these $\mu$Kelvin fluctuations from inherently noisy data is a serious computational challenge.
Example: Wikipedia Anthropology

Kittur, Suh, Pendleton (UCLA, PARC), “He Says, She Says: Conflict and Coordination in Wikipedia”
CHI, 2007

Increasing fraction of edits are for work indirectly related to articles

---

**Experiment**

- Download entire revision history of Wikipedia
- 4.7 M pages, 58 M revisions, 800 GB
- Analyze editing patterns & trends

**Computation**

- Hadoop on 20-machine cluster

---

Figure 4. Changing percentage of edits over time showing that decreasing direct work (article) and increasing indirect work (article talk, user, user talk, other, and maintenance).
Example: Web Page Analysis


Experiment

- Use web crawler to gather 151M HTML pages weekly 11 times
  - Generated 1.2 TB log information
- Analyze page statistics and change frequencies

Systems Challenge

“Moreover, we experienced a catastrophic disk failure during the third crawl, causing us to lose a quarter of the logs of that crawl.”

Figure 2. Distribution of document lengths overall and for selected top-level domains.
DNA Sequencing

- Genome of an organism encodes genetic information in long sequence of 4 DNA nucleotides: ATCG
  - Bacteria: ~5 million bp
  - Humans: ~3 billion bp
- Current DNA sequencing machines can generate 1-2 Gbp of sequence per day, in millions of short reads (25-300bp)
  - Shorter reads, but much higher throughput
  - Per-base error rate estimated at 1-2% (Simpson, et al, 2009)
- Recent studies of entire human genomes have used 3.3 (Wang, et al., 2008) & 4.0 (Bentley, et al., 2008) billion 36bp reads
  - ~144 GB of compressed sequence data
Divide and Conquer

Partition

Combine

"Work"

w₁

"worker"

r₁

w₂

"worker"

r₂

w₃

"worker"

r₃

"Result"
Difficult because

(Often) don’t know the order in which workers run
(Often) don’t know where the workers are running
(Often) don’t know when workers interrupt each other

Thus, we need:

Semaphores (lock, unlock)
Conditional variables (wait, notify, broadcast)
Barriers

Still, lots of problems:

Deadlock, livelock, race conditions, ...

Moral of the story: be careful!

Even trickier when the workers are on different machines
What’s MapReduce

Parallel/Distributed Computing Programming Model

- **Input split**
- **Map tasks**
- **Reduce tasks**
- **Output**

Diagram:
- **Input split**: split 0, split 1, split 2, split 3, split 4
- **Map tasks**: map(), map(), map(), map(), map()
- **Reduce tasks**: reduce(), reduce(), reduce(), reduce()
- **Output**: part 0, part 1, part 2

Diagram elements:
- Input split
- Shuffle
- Output
Word Frequencies in Web pages

- 输入：one document per record
- 用户实现map function，输入为
  - key = document URL
  - value = document contents
- map输出(potentially many) key/value pairs.
  - 对document中每一个出现的词，输出一个记录<word, “1”>

```
“document1”, “to be or not to be”
```

```
“to”, “1”
“be”, “1”
“or”, “1”
...
```
Example continued:

- MapReduce 运行系统(库)把所有相同 key 的记录收集到一起 (shuffle/sort)
- 用户实现 reduce function 对一个 key 对应的 values 计算
  - 求和 sum

```
key = “be”
values = “1”, “1”

“2”
```

```
key = “not”
values = “1”

“1”
```

```
key = “or”
values = “1”

“1”
```

```
key = “to”
values = “1”, “1”

“2”
```

- Reduce 输出 <key, sum>

```
“be”, “2”
“not”, “1”
“or”, “1”
“to”, “2”
```
MapReduce Runtime System
History of Hadoop

- 2004 - Initial versions of what is now Hadoop Distributed File System and Map-Reduce implemented by Doug Cutting & Mike Cafarella
- December 2005 - Nutch ported to the new framework. Hadoop runs reliably on 20 nodes.
- January 2006 - Doug Cutting joins Yahoo!
- February 2006 - Apache Hadoop project official started to support the standalone development of Map-Reduce and HDFS.
- March 2006 - Formation of the Yahoo! Hadoop team
- May 2006 - Yahoo sets up a Hadoop research cluster - 300 nodes
- April 2006 - Sort benchmark run on 188 nodes in 47.9 hours
- May 2006 - Sort benchmark run on 500 nodes in 42 hours (better hardware than April benchmark)
- October 2006 - Research cluster reaches 600 Nodes
- December 2006 - Sort times 20 nodes in 1.8 hrs, 100 nodes in 3.3 hrs, 500 nodes in 5.2 hrs, 900 nodes in 7.8
- January 2006 - Research cluster reaches 900 node
- April 2007 - Research clusters - 2 clusters of 1000 nodes
- Sep 2008 - **Scaling Hadoop to 4000 nodes at Yahoo!**
Resource Mgmt and Scheduling in Clouds

- Market-based Resource Management
- SLAs brokered between customer and provider
- Reservation systems
- Deadline and budget-constrained rm/scheduling
- Profit-driven scheduling models
- Scheduling of interactive applications, QoS
- Scheduling to meet data needs (move computation to the data)
- Scheduling for high throughput computing
The CIS 410/510 Debate (Tue 5/3)

- **Team 1**: Scheduling for Clouds requires major innovation in scientific and economic models, policies, algorithms.
- **Team 2**: Scheduling for Clouds is simply Grid scheduling on steroids (no fundamental new insights needed, just need to address pragmatic and technological issues of scale and commerce).
- **Judges** will decide based on technical merit and presentation
- 10 minutes each team.
- Winning team gets 10 pts and free lunch, losing team gets 8 pts, judges get 9 pts