



HiPIC

Introduction to Cloud Computing

2009 KOCSEA Symposium

Jongwook Woo, PhD

jwoo5@calstatela.edu

High-Performance Internet Computing Center (HiPIC)

Computer Information Systems Department

California State University, Los Angeles

Copyrights © Jongwook Woo



CSULA



HiPIC

Contents

- **New York Times case**
- **Map/Reduce**
- **Cloud Computing**
- **Why now?**
- **Models**
- **Running your Biz with Cloud Computing**

Copyrights © Jongwook Woo

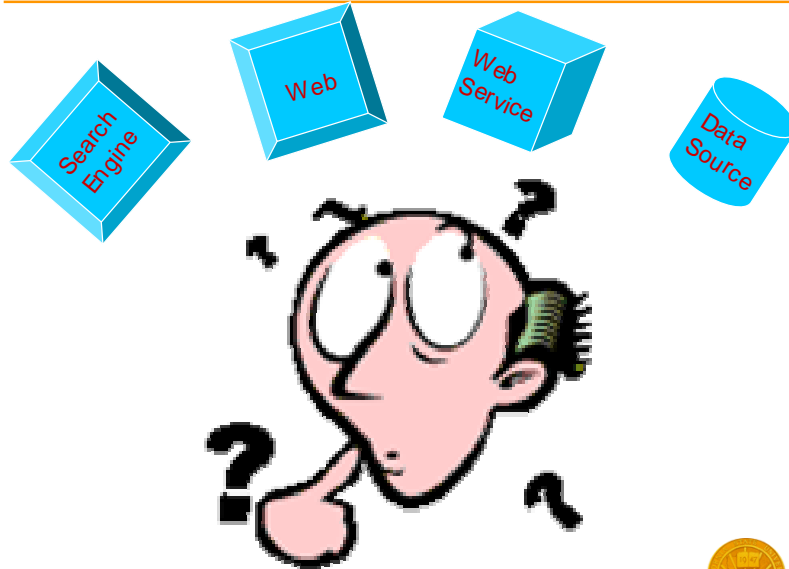


CSULA



HIPIC

What is Cloud Computing



Copyrights © Jongwook Woo



CSULA



HIPIC

Have you heard about Cloud Computing?

■ Heard about it

- but not clear what it is

■ First Impression

- In late 2007, the New York Times wanted to make available over the web its entire archive of articles,
 - 11 million in all, dating back to 1851.
 - four-terabyte pile of images in TIFF format.
 - needed to translate that four-terabyte pile of TIFFs into more web-friendly PDF files.
 - not a particularly complicated but large computing chore,
 - requiring a whole lot of computer processing time.
 - a software programmer at the Times, Derek Gottfrid,
 - playing around with Amazon Web Services, Elastic Compute Cloud (EC2),
 - uploaded the four terabytes of TIFF data into Amazon's Simple Storage System (S3)
 - In less than 24 hours, 11,000 PDFs, all stored neatly in S3 and ready to be served up to visitors to the Times site.
 - The total cost for the computing job? \$240
 - 10 cents per computer-hour times 100 computers times 24 hours

Copyrights © Jongwook Woo



CSULA



HPIC

Have you heard about Cloud Computing? (Cont'd)

■ Second Impression (Motivation) in parallel

- Consulted companies: CitySearch.com
 - Search Engines: FAST, Apache Solr (Lucene) and Nutch
- Need to analyze User's behavior based on users' log information
 - Data Mining needed for peta-bytes log file
 - With Map/Reduce in Apache Hadoop or Amazon EC2
 - Google implemented in its file systems
 - To analyze users' behaviors

Copyrights © Jongwook Woo



CSULA



HPIC

What is MapReduce

■ Provides Restricted parallel programming model

- User implements Map() and Reduce()
- Libraries take care of EVERYTHING else
 - Parallelization
 - Fault Tolerance
 - Data Distribution
 - Load Balancing

Copyrights © Jongwook Woo



CSULA



HIPIC

Map and Reduce

- **Functions borrowed from functional programming languages (eg. Lisp)**
- **Useful model for many practical tasks, especially for huge (peta- or Terra-bytes) but non-complicated data**
 - New York Times case
 - Log file for web companies

Copyrights © Jongwook Woo



CSULA



HIPIC

Map

- **Convert data to (key, value) pairs**
- **map() functions run in parallel,**
 - creating different intermediate values from different input data sets

Copyrights © Jongwook Woo

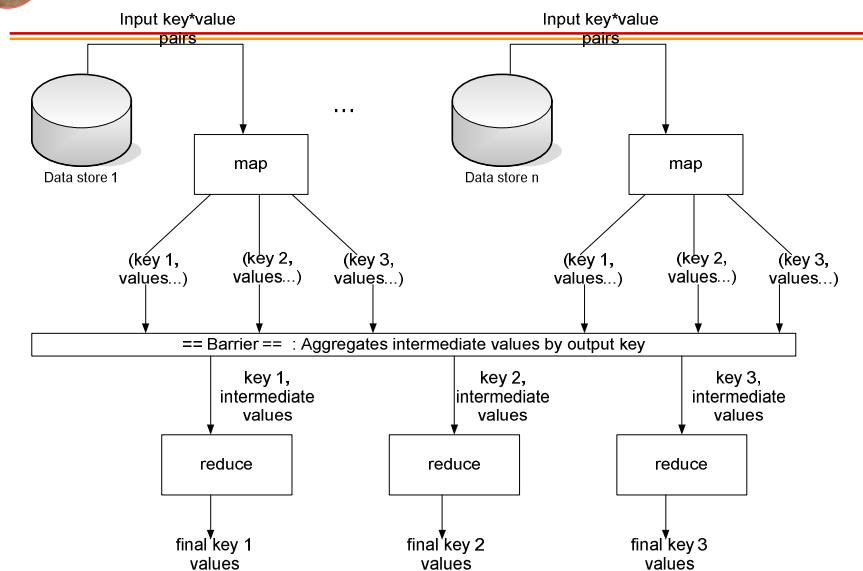


CSULA



Reduce

- **reduce()** combines those intermediate values into one or more *final values* for that same output key
- **reduce()** functions also run in parallel,
 - each working on a different output key
- **Bottleneck:**
 - reduce phase can't start until map phase is completely finished.





HPIC

Example: Sort URLs in the largest hit order

■ Map()

- Input <logFilename, file text>
- Parses file and emits <url, hit counts> pairs
 - eg. <http://hello.com, 1>

■ Reduce()

- Sums all values for the same key and emits <url, TotalCount>
 - eg. <http://hello.com, (3 5 2 7)> => <http://hello.com, 17>

Copyrights © Jongwook Woo



CSULA



HPIC

Cloud Computing

■ So, is Map/Reduce Cloud Computing?

■ Yes, but it is not all

- NIST: Cloud computing is 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.
- We could simply say it is something that provides S/W and H/W services to the users and companies

Copyrights © Jongwook Woo



CSULA



HPIC

Why Cloud Computing now?

■ Growth of the Internet usage

- Broadband networking
- Mobile, location-aware services

■ Massive data – horizontal scale

- User-generated content, digital media
- Even more data ahead with web
 - Past: need data
 - Present / Future: too many data
 - For example, Think scientific data at JPL

Copyrights © Jongwook Woo



CSULA



HPIC

Cloud Computing Models

■ Software as a Service

- Applications on-demand

■ Platform as a Service

- Developer platform for creating applications

■ Infrastructure as a Service

- Storage and compute capabilities offered as a service

Copyrights © Jongwook Woo



CSULA



HPIC

Software as a Service (SaaS)

■ Applications on demand:

- Subscription-based, multi-tenant, nothing to download or manage

■ Google Apps (docs, email), Microsoft Exchange Online, Yahoo Mail, TurboTax Online, YouTube, Twitter, Flickr, Salesforce.com, Redmine, Assembla

Copyrights © Jongwook Woo



CSULA



HPIC

Platform as a Service (PaaS)

■ On-demand develop and deploy apps

- Unique programming model, auto-scaling
- Often both a platform and a channel

■ Google AppEngine, Engine Yard

Copyrights © Jongwook Woo



CSULA



HPIC

Infrastructure as a Service (IaaS)

■ On-demand virtual infrastructure

- Lowest level, most general, self-provisioning
- Unlimited managed resources

■ Amazon AWS (EC2, S3, SQS), Microsoft Azure, RackSpace Cloud, Savis, Terremark, Joyent

Copyrights © Jongwook Woo



CSULA



HPIC

Benefits

■ Efficiency

- Pay As-You-Go
- Op-ex vs. Cap-ex
- Virtualization

■ Flexibility

- Demand Scalable Services

■ Speed

- Rapid, Self Provisioning
- Faster Deployment
- API-Driven

Copyrights © Jongwook Woo



CSULA



Issues

■ Security

- Security is still your responsibility
- Learn everything you can (attackers will)
- Ease of use often comes with greater risk
- Monitor – don't assume your provider will alert you



Public vs Private Clouds

■ Public

- Pay as you go,
- Multitenant Applications and services
- Access virtually unlimited resources

■ Private

- Cloud Computing model in a company's own datacenter
- Resources directly owned
 - but therefore constrained

■ Hybrid

- Mixed usage of both public and private clouds,
 - Often integrated into the same application





HPIC

Example Cloud APIs

- Amazon's AWS
 - EC2, S3, SQS, SDB.
- ServePath
 - GoGrid API
- Google Map
 - Map API
- Sun Microsystems
 - Open Cloud API
- Vmware
 - vCloud API

Copyrights © Jongwook Woo



CSULA



HPIC

Economics of Cloud Computing

- Easy to run your start-up company
- Pay as you go reduces startup costs and risk for the investor
 - Capex (capital expense)
 - Typically large upfront cost of purchasing equipment
 - Opex (operating expense)
 - Monthly cost of renting equipment
 - You don't need to spend money for maintaining servers

Copyrights © Jongwook Woo



CSULA



HPIC

Running the company (Example)

- Setup DB server, Web/App server to the given OS at godaddy.com
 - and deploy your codes to EngineYard
- Setup company email at Google mail
- Setup project management tool at Assembla.com
- Use laptops and mobile phones to implement your products etc for everything else

Copyrights © Jongwook Woo



CSULA



HPIC

Conclusion

- **Peta-, or Tera-bytes of data to analyze**
 - Use Map/Reduce approach
- **IT has been changed to “Purchase services”**
 - Don't need to purchase equipments
 - You can easily run your start-up company in cloud computing world

Copyrights © Jongwook Woo



CSULA



HPIC



CSULA

Copyrights © Jongwook Woo



HPIC

References

- **“Introduction to Cloud Computing - for Startups and Developers”,** Lew Tucker, Ph.D. CTO, Cloud Computing, Sun Microsystems, Inc.
- **“Introduction to Cloud Computing - for Enterprise Users”,** Lew Tucker, Ph.D. CTO, Cloud Computing, Sun Microsystems, Inc.
- **“Google’s Parallel Programming Model and Implementation MapReduce”,** Klara Nahrstedt and Sam King, UIUC
- **“How to painlessly process terabytes of data”,** John R. Gilbert, UCSB
- **“MapReduce Theory and Implementation”,** Christophe Bisciglia, Aaron Kimball, & Sierra Michels-Slettvet, University of Washington and Google



CSULA

Copyrights © Jongwook Woo



HPIIC

Example: Count word occurrences of each word in a large collection of documents

```
map(String input_key, String input_value):
    // input_key: document name
    // input_value: document contents
    for each word w in input_value:
        EmitIntermediate(w, "1");

reduce(String output_key, Iterator
intermediate_values):
    // output_key: a word
    // output_values: a list of counts
    int result = 0;
    for each v in intermediate_values:
        result += ParseInt(v);
    Emit(AsString(result));
```

Copyrights © Jongwook Woo



CSULA