

Scalable Web Programming

CS193S - Jan Jannink - 2/23/10

Weekly Syllabus

1. Scalability: *(Jan.)*

2. Agile Practices

3. Ecology/Mashups

4. Browser/Client

5. Data/Server: *(Feb.)*

6. Security/Privacy

7. Analytics

8. Cloud/Map-Reduce

9. Published APIs: *(Mar.)**

10. Future

*** PROJECT DUE DATE**

Administrative Stuff

- * Grading change for project submissions
 - * better of project submissions = 50% of grade
 - * weaker of project submissions = 30% of grade
- * 720 checkins to the projects by Thursday evening
- * Good feedback loop with project advocates
 - * 1½ PHP projects, 4½ GWT projects

Project Thoughts

- * Use git with some care
 - * no executables, temp files etc.
 - * review commits the way you review homework
- * Surprising how UI polish counts, even to techies
- * Button mashing stress tests are kind of fun
- * At least 1 map/reduce opportunity in the bunch

Cloud Computing

- * Internet always diagrammed as a cloud
- * 'Always' essentially equals 20 years
- * Commoditization of networked compute cycles
 - * SaaS (Salesforce)
 - * PaaS (AppEngine)
 - * IaaS (AWS)

Roll your Own

- * Build a custom datacenter, or collocate
- * Do a ton of IT setup, server provisioning, etc.
- * Set up high speed link to Internet exchange
- * Negotiate peering (data swap) with others

- * Hmm, there's got to be something easier

Choose your level of IT

- * Infrastructure as a Service (IaaS)
 - * virtual hardware, many management tools
- * Platform as a Service (PaaS)
 - * virtual scalability, focus on app
- * Software as a Service (SaaS)
 - * a la carte virtual app, focus on customization

Innovation Competency

- * If IT innovation is part of your DNA, roll your own
 - * bootstrap with AWT
- * If app innovation is your thing, scalability may not
 - * AppEngine takes much scalability burden away
- * If product innovation is critical
 - * build around a fully customizable core service

Implications

- ✱ Lots of things are clouds if you think about it
 - ✱ WWW
 - ✱ Botnets
 - ✱ Folding@home
 - ✱ Etc.

Dark Clouds - Botnets

- * Self propagating networks of hacked computers
- * Compute time is sold as in other distributed nets
- * In 2009 an estimated 200 billion spam sent daily
 - * Estimated 90% of email is spam
 - * 90% of spam is untargeted

Distributed Computing

- * About 8 petaFLOPS contributed to major efforts
 - * protein folding
 - * SETI @ home, etc.
- * Over 1,000,000 operations per human per second
 - * many peer reviewed publications from results

Opinion - Cyborg

C'MON IN A CLOUD LECTURE?

Opinion - Cyborg



C'MON IN A CLOUD LECTURE?

Opinion - Cyborg



Opinion - Cyborg



C'MON IN A CLOUD LECTURE?

Fatdoor Example

- * Intense Mapping, White Pages, Wiki mashup
- * Data on S3, Google Base
 - * 180 Million U.S. IDs
 - * 30 Million U.S. Businesses
- * Map/reduce MySQL query framework on EC2
- * Not launched as planned due to privacy concerns

Some Lessons Learned

- * Bucketize data at a reasonable density on S3
 - * 10K items per bucket max
 - * preserves performance of most operations
- * EC2 server management transforms sys admin job
- * Tons of great public data on Google Base
 - * Ex. Yelp info

Cool Features

- * Easy to restart crashed servers remotely
 - * physical servers need specialized hardware
- * EC2 server instances start fast (a couple minutes)
 - * feels just like working with virtual servers
 - * remote desktop or VNC is also available
- * Full suite of scalability services in place

Virtualization, Emulation

- * Complementary technology to the cloud, enabling
 - * multiplexing of single system resources
 - * checkpointing/restarting of servers
 - * migration/duplication of servers
- * Emulation provides
 - * sandboxing, anti-obsolescence, portability

Virtualization Directions

- * More compact server state representations
- * Transparent migration of server images
- * Implementation of server forking
 - * equivalent in concept to Unix `fork()` call
 - * provides fault tolerance, parallelism
- * Just in time server provisioning

Cloudy Forecast

- * Colocation not going away, but
- * Startup timescales are shortening
 - * investor impatience, competition, funding
- * Fixing mistakes needs to be accordingly faster
 - * no more learning scalability on the job
- * SaaS business opportunities will grow fastest

Worth Checking Out

- ✱ Local colocation

- ✱ <http://www.layer42.net/>

- ✱ Honeynet project, distributed computing

- ✱ <http://www.honeynet.org/papers/bots/>

- ✱ <http://www.hyper.net/dc-howto.html>

- ✱ Amazon (again)

- ✱ <http://aws.amazon.com/>

Q & A Topics

- ✱ Future cloud services
- ✱ Cloud security
- ✱ Programming environments for clouds

Scalable Web Programming

CS193S - Jan Jannink - 2/23/10