SCALABLE WEB PROGRAMMING

CS193S - Jan Jannink - 2/04/10

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. Publish APIs: *(Mar.)**
10. Future

* assignment due

Project Management

* We will score your work on the basis of git checkins
    * the more the better
    * to get credit your checkins must be under your id
* There is a small portion of your score relating to your team
    * a bonus for every person who contributes to your project
* Bonus for code that helps or integrates teammates code

Project Schedule & Due Dates

* Initial due date, Thursday, February 18
    * don't wait until then to start checking in code
        * we'll provide comments and redirects to guide further work
* Polished projects, Friday, March 5
* Demo Day, Wednesday, March 10
    * teams + apps + investors + lunch 11:30-2, 204 Packard Bldg.

A Few Notes

* It appears there will be one PHP based project
    * serve as control group to other projects
* It will have at least one GWT integration piece
    * GWT embeddable widget
* We will compare its scalability more carefully
    * the project advocate has a greater responsibility to the team

Q & A

* Is one exception fair?
* Team questions
* Project questions
* Timeline questions
    * choose & start a sub project asap, make sure team knows
* Demo questions

Googol : 10^{100}

* Scalability is the art of abstraction/pattern matching
  * estimated number of atoms in the universe (~10^{80})
  * possible configurations of smallest viroid genome (~10^{1000})
* New perspective on current state of the universe
  * based on a search algorithm within possible configurations

Google

* Collect and connect all of the world's data
* 1997: Backrub demo, terabyte disk array, 3 crawlers,
    * monthly data refresh
* 2000: 4 x 486, 8 x HDD per 1U slot, 160 computers per rack
    * terabyte RAM purchase from pricewatch.com
* 2010: ~10^6 servers, 10^{18} bytes, continuous data refresh

Top 500 Supercomputers

* Jaguar 225K cores, ~20K nodes, ~400W / node, Cray, Linux
* Top 5 all run Linux
* Little difference compared to Google except
  * two orders of magnitude smaller
  * order of magnitude more expensive
  * less configurable, growable, upgradable, etc.

Back to Software

* Transform manual repetitive tasks into automated ones
* A lot of IT work starts out fully manual
* command line sequences
* write scripts based on command history
* develop dashboard to manage scripts
* iterate

Server Automation Workshop

* Amazon EC2, S3
    * create your own server image (AMI)
        * adapt an existing one manually (windows is possible)
        * store image to S3
        * migrate image back & forth with local virtual server
        * run multiple instances

Automatic Server Provisioning

* Dashboard, pay as you go model much easier than hosting
* Experiment with webservers
    * try load balancing
* Set up a cache server, mySQL server
* Develop separate AMIs for each
* Script the launching / retiring of server instances

Hosting, Redundancy

* Rackspace model
    * provisioning can take days vs. minutes
* Colocation model
    * many options in Bay Area
* Combine with AWS for flexibility
    * load balance from your hosting provider

Maintaining IT

* Scalability in IT is indistinguishable from elsewhere
* Revision control for scripts, logs, etc.
* Splunk search engine for IT
* Use Lucene for a cheap internal version of Splunk
* Patterns in data guide subsequent work
* Maximize the communication value of all stored content

Human scalability

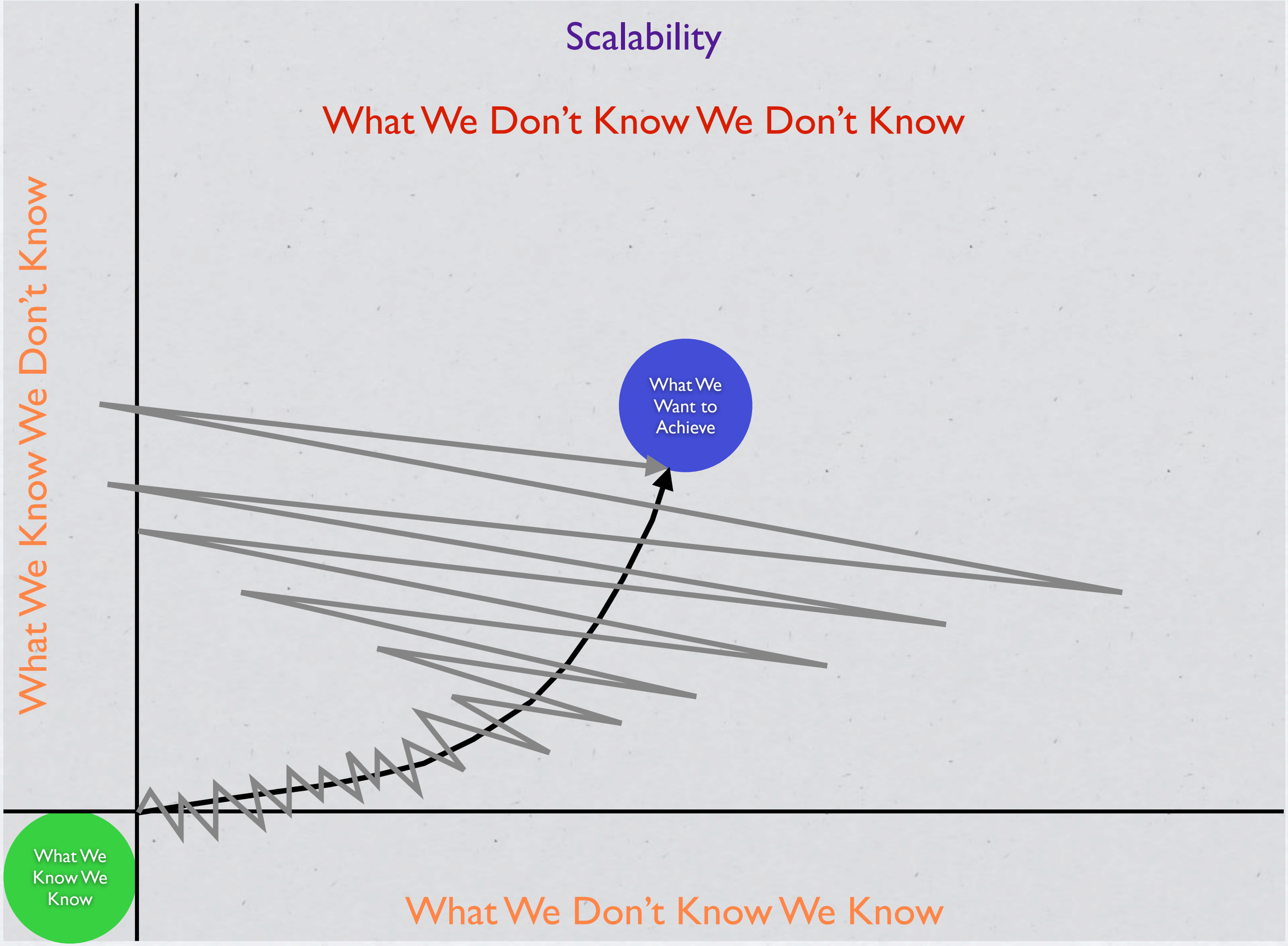
* Take repetitive manual process
* Find simple automation technique
* Implement, Improve
* Take next repetitive manual process…
    * apply at work, at home, wherever possible
    * if I could only write a diapering script that works on twins…

Scalability as a Process

* Solutions from nature
    * spiral sea shells could technically grow without end
    * self similar growth patterns also transcend size limitations
* Generally our environment is much more granular in nature
    * scaling is similar to finding stepping stones
    * each stepping stone is a technology to allow the next growth

Scalability

What We Don't Know We Don't Know



Worth Checking Out

* Search engine auto completion
  * http://www.predictablyirrational.com/?p=704
* Top 500 Supercomputers, Amazon AWS, Splunk, Lucene
  * http://www.top500.org/
  * http://aws.amazon.com/
  * http://www.splunk.com/
  * http://lucene.apache.org/java/docs/

Q & A Topics

* Left over project questions
* Micromanaging vs. Scalability
    * where is the cutoff?
    * how do you get buy in?
* Does it matter if the universe is a computer?