

Lecture 02: The Cluster Scaling Revolution, and Clusterjob

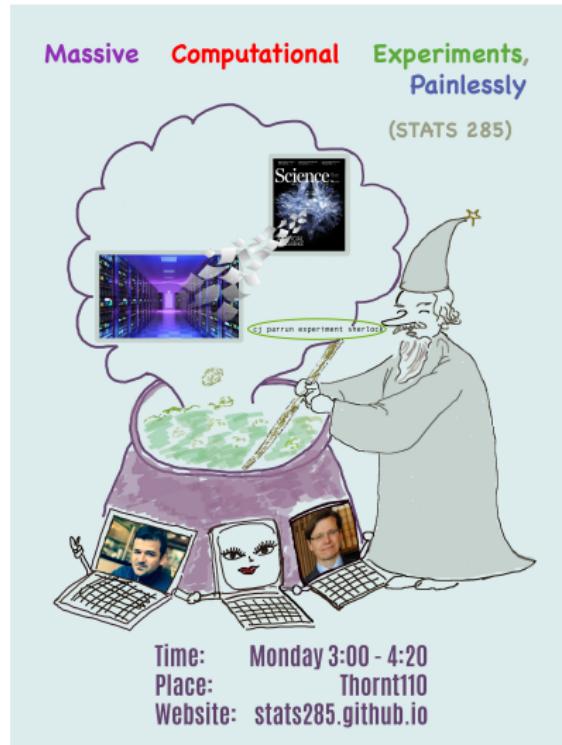
H. Monajemi/DL. Donoho

Stats285, Stanford

Oct/02/2017



Stanford University



Stanford University

*This document contains images obtained by routine Google Images searches. Some of these images may perhaps be copyright. They are included here for educational noncommercial purposes and are considered to be covered by the doctrine of **Fair Use**. In any event they are easily available from Google Images. It's not feasible to give full scholarly credit to the creators of these images. We hope they can be satisfied with the positive role they are playing in the educational process.*



Disclaimer, 2: Tweets

*This presentation contains several **fake tweets** that can be easily created online and are included solely as a parody.*

No-one should take anything in tweets seriously.



Stanford University

- 1 Computing discontinuity
- 2 The Cloud explosion
- 3 Cloud wars
- 4 Cloud is timely: Moore's law faltering
- 5 Computing change is real!
- 6 Why can clusters seem painful?
- 7 How we make clusters less painful?
- 8 Things you should know about clusters
- 9 All about CJ
- 10 Announcements



Stanford University

Global Economy → Computing → Science



Stanford University

Explosion of Computational Resources

Cloud Paradigm:

- Billions of smart devices each drive queries to cloud servers
- Millions of business relying on cloud for all needs

Symbiosis of cloud and economy is *lasting* and *disruptive*.

Cloud provides *any user same-day* delivery:

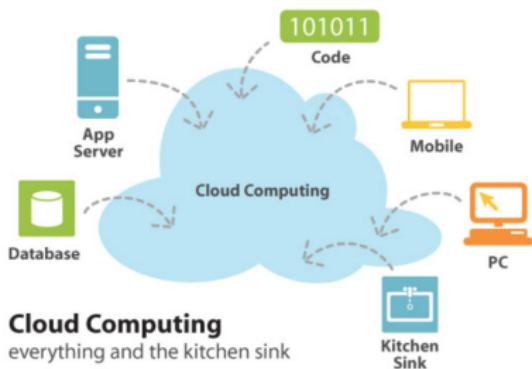
- Tens to hundreds of thousands of hours of CPU
- Pennies per CPU hour

Any user can consume *1 Million CPU hours* over a few days for a few \$10K's.



Stanford University

Cloud is all-purpose



Many Uses

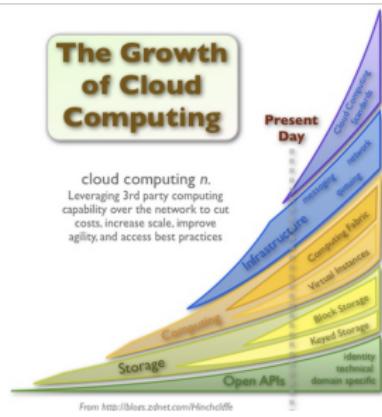
5 Reasons Businesses Use the Cloud

Every year, more and more businesses are adopting cloud computing. While it's usually thought of as an IT decision, it's now finally being considered a business decision to support business needs. Here are five reasons why more businesses are adding the cloud to their technology arsenals.



Widespread Advocacy

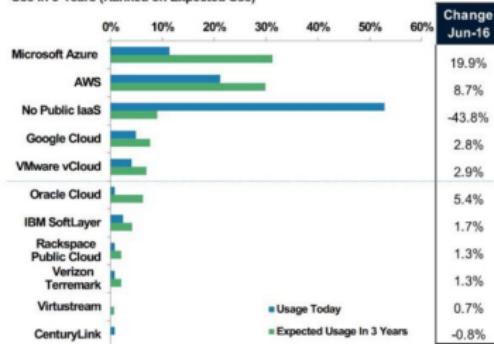
Services offered by Cloud Expanding



Proliferating Services

Exhibit 15: Cloud-based IaaS Use Today and Expected Use in Three Years

% of Respondents Using Cloud-based IaaS Today and Expected Use In 3-Years (Ranked on Expected Use)

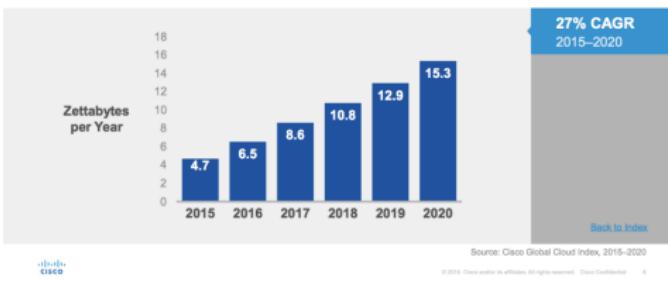


Businesses Expect Growth

Global Data Center Growth

Global Data Center Traffic Growth

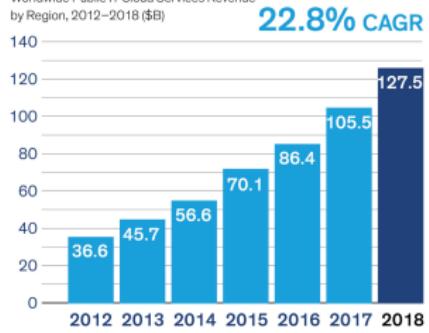
Data Center Traffic More Than Triples from 2015 to 2020



Traffic

The cloud market is growing rapidly.

Worldwide Public IT Cloud Services Revenue
by Region, 2012–2018 (\$B)



Revenue

Cloud Computing past 'Hype Hump'



Cloud Wars

The Telegraph

HOME NEWS SPORT BUSINESS ALL SECTIONS

Technology

News Reviews Opinion Internet security Social media Apple

More ▾

• Technology

Cloud wars: Google, Amazon and Microsoft battle to own the future of computing



Media

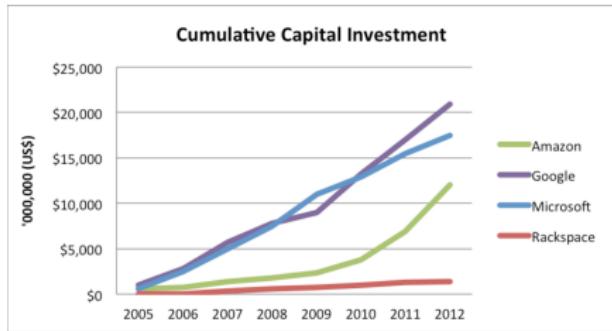


Symbols

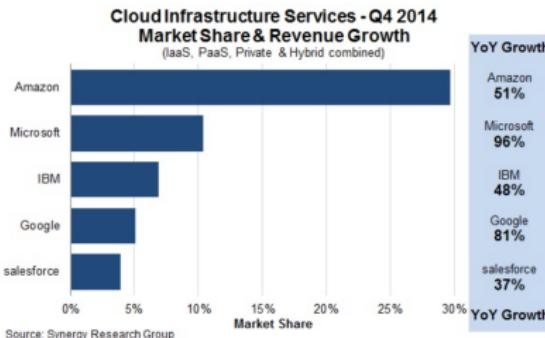


Stanford University

Big Three Invest and Profit



Traffic



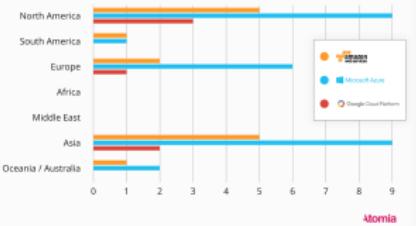
Revenue

Big Three Have Global Reach



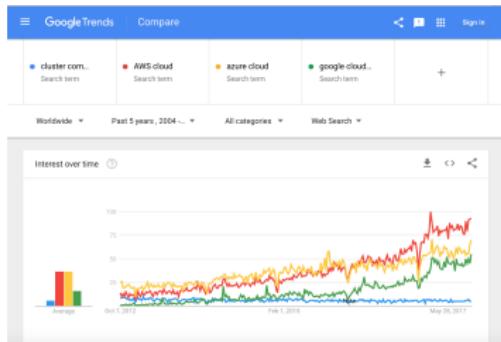
Map

Cloud provider data center locations worldwide



Breakdown

Big Three Have Buzz



Google Trends



Market Share Growth

AWS Solidly Leading

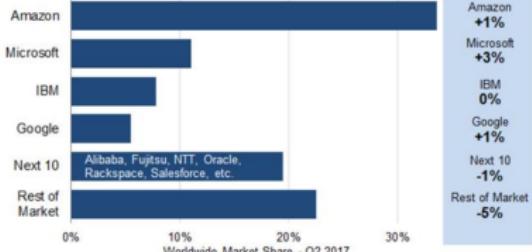
Figure 1. Magic Quadrant for Cloud Infrastructure as a Service



Gartner Magic Quadrant

Cloud Infrastructure Services - Q2 2017 Market Share & Revenue Growth

(IaaS, PaaS, Hosted Private Cloud)



Market Share

AWS is world's largest computer

AWS placed its [data centers](#) across 33 [availability zones](#) within [12 regions worldwide](#). Each availability zone has at least one data center (some have [as many as six](#)) that has [redundant power](#) for stability, networking and connectivity. In each data center, there are between [50,000 to 80,000 servers with up to 102 Tbps bandwidth](#).

If you assume an average of three data centers per zone and 65,000 servers per data center, you will end up having 6.4 million servers worldwide. For those of you who care about availability and performance of their applications in the cloud, the huge computing capacity of AWS ensures higher fault tolerance and low latency.



Stanford University

AWS vs. Azure vs. Google On-Demand Prices

Resource Type (us-east, Linux)	AWS Instance	Azure Instance	Google Instance	AWS OD Hourly	Azure OD Hourly	Google OD Hourly	AWS /GB RAM	Azure /GB RAM	Google /GB RAM
Standard 2 vCPU w SSD	m3.large	D2 v2	n1-standard-2	\$0.133	\$0.114	\$0.212	\$0.017	\$0.016	\$0.028
Highmem 2 vCPU w SSD	r3.large	D11 v2	n1-highmem-2	\$0.166	\$0.149	\$0.238	\$0.011	\$0.011	\$0.018
Highcpu 2 vCPU w SSD	c3.large	F2	n1-highcpu-2	\$0.105	\$0.099	\$0.188	\$0.028	\$0.025	\$0.104
Standard 2 vCPU no SSD	m4.large	D2 v2	n1-standard-2	\$0.108	\$0.114	\$0.100	\$0.014	\$0.016	\$0.013
Highmem 2 vCPU no SSD	r4.large	D11 v2	n1-highmem-2	\$0.133	\$0.149	\$0.126	\$0.009	\$0.011	\$0.010
Highcpu 2 vCPU no SSD	c4.large	F2	n1-highcpu-2	\$0.105	\$0.099	\$0.076	\$0.027	\$0.025	\$0.042

As of Dec 2, 2016

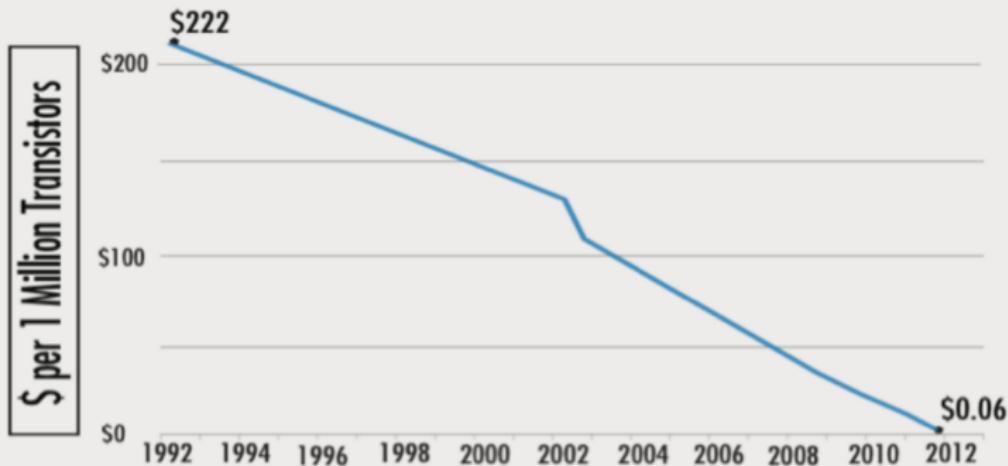
Source: RightScale



Stanford University

Traditional Story of Moore's Law: Inexorable Progress

Computing Cost-Performance (1992 - 2012)

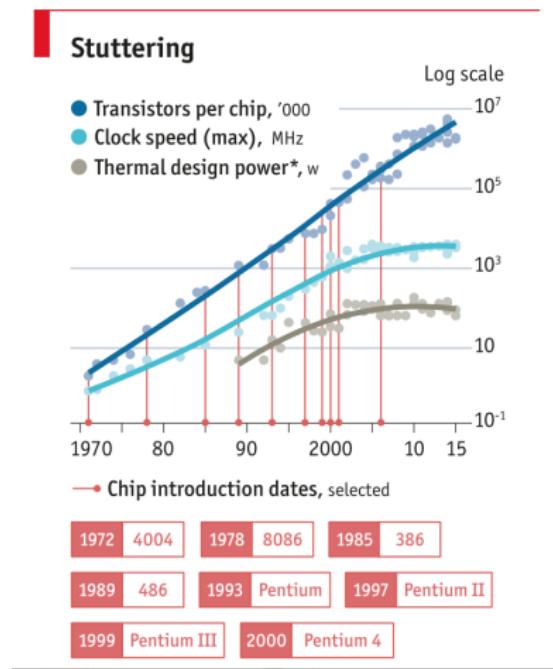


Source: Deloitte University Press



Stanford University

Recent Story of Moore's Law: Stagnation



*The result is a consensus among Silicon Valley experts that **Moore's law is near its end**. "From an economic standpoint, **Moores law is dead**", says Linley Gwennap, who runs a Silicon Valley analysis firm. Dario Gil, IBM's head of research and development, is similarly frank: "I would say categorically that **the future of computing cannot just be Moores law any more.**" Bob Colwell, a former chip designer at Intel, thinks the industry may be able to get down to chips whose components are just five nanometres apart by the early 2020s but "you'll struggle to persuade me that they'll get much further than that.*

Guardian

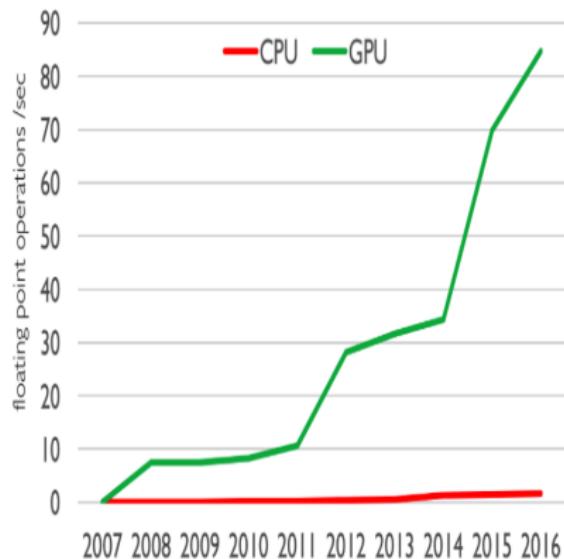
<https://www.theguardian.com/technology/2017/jan/26/vanishing-point-rise-invisible-computer>



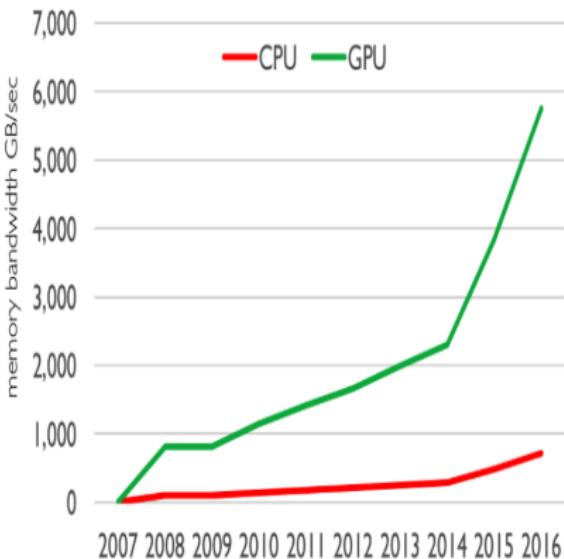
Stanford University

GPU's may give temporary respite

TeraFLOPS, sp



Memory Bandwidth



GPU performance forecast to saturate by 2020.
AWS is offering GPU's

Conclusions

- Over the last ten years, massive computational resources have been created
- Publicly available to anyone for price 3 cents per GB per CPU hour
- Near-unlimited quantities (for a price)
- Expansion by factors of 1000's in immediate computing capacity when job is *trivially parallelizable*
- Traditional routes to enhanced performance are blocked.
- Welcome to the era of **Computing Change**



Stanford University

Computing Change Era!



Stanford University

Computing change skeptics



Donald J. Trump

@realDonaldTrump

Professor Donoho was terrible on @stats285 today. He said COMPUTING CHANGE is the most important thing, not all of the current disasters!

RETWEETS
10,803

LIKES
21,017



12:16 AM -



206



Donald J. Trump

@realDonaldTrump

The concept of 'computing change' was created by and for the Chinese in order to make U.S. scientists non-competitive.

RETWEETS
14,861

LIKES
24,807



12:23 AM - 30 Sep 2017



266



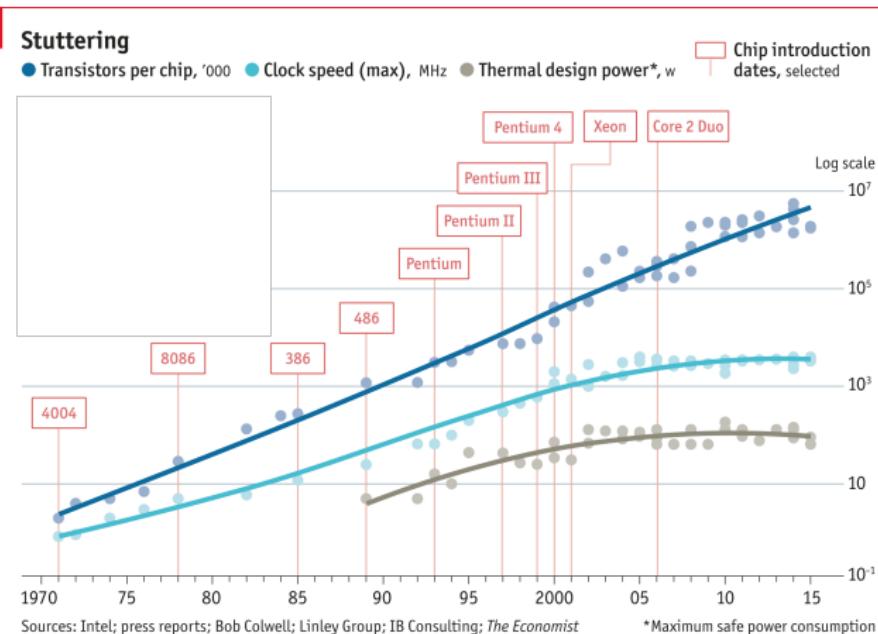
15K



25K

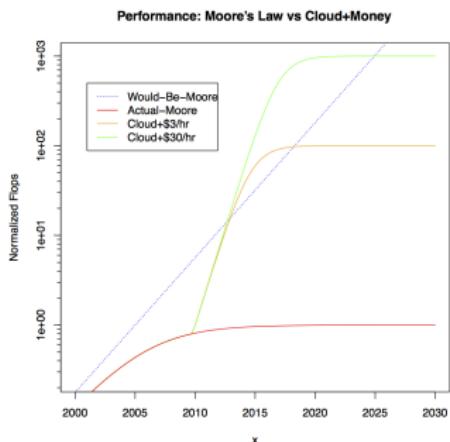
Computing Phase Transition - I

- Classic Moore's law predicts 32x increase from 2010-2020
- Moore's law no longer possible *physically and economically*

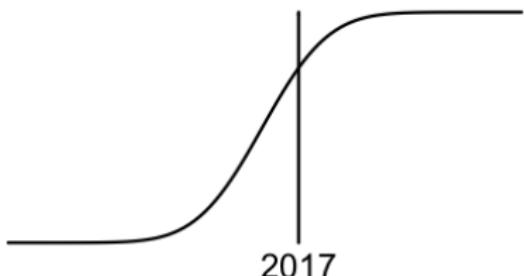


Computing Phase Transition - II

- Classic Moore's law predicts 32x increase from 2010-2020
- Moore's law no longer possible *physically* and *economically*
- ... Today, computing power *easily* accessible \uparrow 10,000x
 - thanks to the symbiosis of cloud and economy!

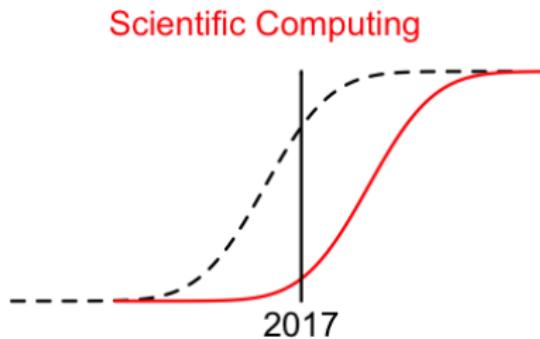


Computing Power



Computing Phase Transition - III

- Classic Moore's law predicts 32x increase from 2010-2020
- Moore's law no longer possible *physically* and *economically*
- ... Today, computing power *easily* accessible \uparrow 1000x
- ... Traditional laptop/desktop scientific computing will lag behind



Stanford University

Computing change real, Hot science emerging

- Consumer desire to be online → IT enrichment
- Better IT → new ways to discover how things work (new science)

“There is good authority for the prediction that within ten years a digital computer will be the world’s chess champion, and that another will discover and prove an important new mathematical theorem” -- Harold J. Leavitt, Management in the 1980’s (1958 article)

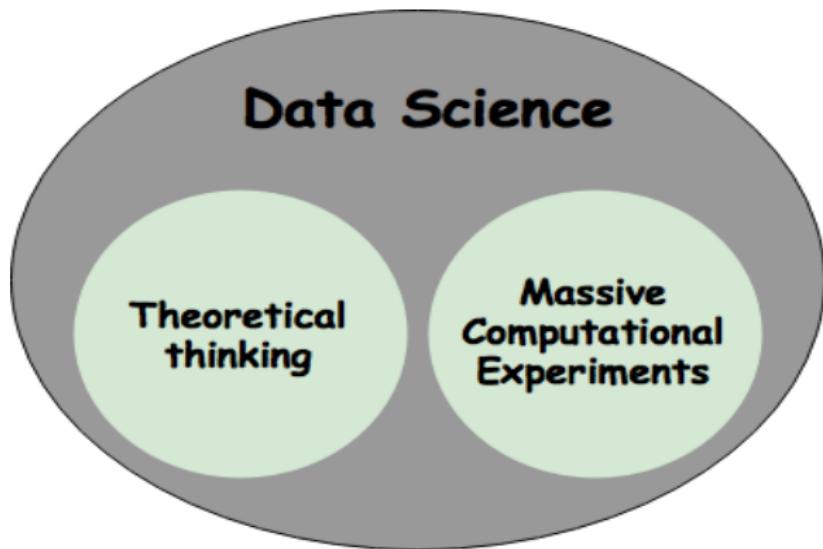
- The new “hot science” can discover new things beyond the reach of theory!



Stanford University

What is this *New Hot Science*?

- it is *Data Science*!



- Data Science will transform Mathematics, Engineering, Medicine, Finance, Security, ... “literally all human life!”

- Just as Climate Change demands adaptation,
- Computing Change demands adaptation:
 - **Psychological change** and rethinking of scientific values
 - **Pose** bold research **hypotheses** to settle computationally
 - **Design massive computing experiments**
 - **Adopt** painless computing **frameworks**
 - **Raise money** to pay for cloud-based computing
 - *Push Button*
- We describe one such framework today: **CJ**
 - In daily use at Stanford
 - Developed by Yours Truly.



Stanford University

Cluster Facilities at Stanford and Elsewhere

- Stanford offers cluster access through
 - Sherlock – 727 servers: 127 shared, 600 owned by faculty
 - Sherlock2 – new generation of Sherlock cluster
 - FarmShare – mostly for coursework and unsponsored research
- To gain access
 - Sponsoring faculty must email
`research-computing-support@stanford.edu`
- Resources where you can learn more about clusters
 - Sherlock: <http://sherlock.stanford.edu>
 - FarmShare: <https://web.stanford.edu/group/farmshare/cgi-bin/wiki/>
 - XSEDE: <https://www.xsede.org>
 - TACC: <https://www.tacc.utexas.edu>
 - PSC: <https://www.psc.edu>
 - OSG: <https://www.opensciencegrid.org>



Stanford University

Outline

- 1 Computing discontinuity
- 2 The Cloud explosion
- 3 Cloud wars
- 4 Cloud is timely: Moore's law faltering
- 5 Computing change is real!
- 6 Why can clusters seem painful?
- 7 How we make clusters less painful?
- 8 Things you should know about clusters
- 9 All about CJ
- 10 Announcements



Stanford University

Many clusters, many systems, many policies!

The image shows a man sitting at a desk, looking overwhelmed with his hands on his head. He is surrounded by several screenshots of different job submission and management interfaces:

- Using the OSG**: A screenshot of the OSG Client Software interface, showing sections for "Running Jobs" and "OSG Client Software". It includes a list of steps for finding OSG sites and running jobs.
- TACC User Portal**: A screenshot of the TACC User Portal homepage, featuring a "Guide" section and links for Home, News, Resources, Allocations, Documentation, Training, Consulting, and About.
- AWS Batch**: A screenshot of the AWS Batch User Guide, showing sections for "What is AWS Batch?", "Setting Up", "Getting Started", "Jobs", "Submitting a Job", "Job States", "Automated Job Retries", and "Job Definitions".
- Google Cloud Platform**: A screenshot of the Google Cloud Platform homepage, showing sections for Why Google, Products, Solutions, and Launcher.
- STANFORD UNIVERSITY | FARMSHARE**: A screenshot of the Stanford FarmShare GridEngine interface, showing sections for "Submit a job", "Using the command-line tool", "Run one of the following commands", "Submit a job", "Manage a cluster", and "APIs & Reference".
- SLURMSubmit**: A screenshot of the SLURMSubmit interface, showing sections for "SEARCH THIS WIKI", "NAVIGATION", "TOOLS", and "CONTENTS".



Stanford University

Changing working scripts, more work!

SUBMIT MULTIPLE JOBS AT ONCE WITH WRAP

The wrap feature of sbatch is very powerful. With it you can send any arguments run are inside the quotation marks after --wrap, for example, mod to create multi-line sbatch submissions based on a directory contents or any st matching to do this.

For example, lets say you want to do something to all fastq files in a directory, matching the string pattern *.fastq. Then we toss that as an argument to sbatch

Create a shell script called wrap.sh:

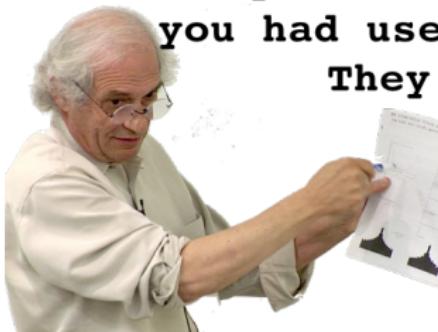
```
#!/bin/sh
for FILE in *.fastq;
do
    sbatch -p normal -t 10 --mem=200 --wrap="gzip ${FILE}"
    sleep 1 # pause for 1 second so we don't overload the scheduler
done
```

My script runs just fine on my laptop. To run it in parallel on cluster, they say I have to change it and give parameters as command line args!!!



Manual tracking, irreproducibility and error!

Can you send me the code and parameters
you had used to produce these results?
They do not seem correct!



Oh, God! That was like 3 month ago.
Since then, I ran a million more
jobs. I can't seem to find it!



Stanford University

Outline

- 1 Computing discontinuity
- 2 The Cloud explosion
- 3 Cloud wars
- 4 Cloud is timely: Moore's law faltering
- 5 Computing change is real!
- 6 Why can clusters seem painful?
- 7 How we make clusters less painful?**
- 8 Things you should know about clusters
- 9 All about CJ
- 10 Announcements



Stanford University

We need to rethink the way we do computational experiments



Stanford University

What does an experiment involve?

In our telling, a computational experiment involves:

- ① **Precise Specification** (define metric and parameters)
- ② **Execution and management** of all the jobs
- ③ **Harvesting** of all the data generated by all the jobs
- ④ **Analysis** of the data
- ⑤ **Reporting** of results.

The painless computing paradigm should seamlessly integrate and automate all these 5 tasks

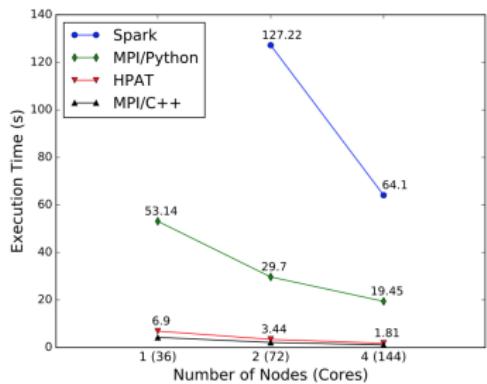


Stanford University

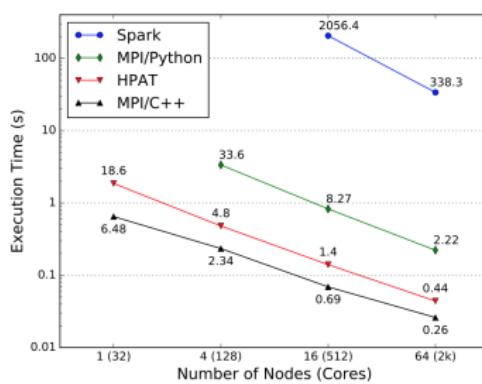
Desired features of a painless paradigm

- **Simple:** the *right* level of abstraction!

Good example: Popularity of Spark though 59x slower than MPI!



(a) Scaling on Amazon AWS cloud (c4.8xlarge instances, 256M 10-feature samples, 20 iterations).



(b) Scaling on Cori supercomputer (1B 10-feature samples, 20 iterations). Please note the logarithmic scale.

Totoni et al. 2017, “A Case Against Tiny Tasks in Iterative Analytics”

Desired features of a painless computing system

- **Simple:** the *right* level of abstraction!
- **Scalable:** push-button massive scaling-up of experiments
- **Reproducible:** all the tasks done in a reproducible way
- **Transparent:** easily be understood post facto

We will see later how we can build such a system



Stanford University

Outline

- 1 Computing discontinuity
- 2 The Cloud explosion
- 3 Cloud wars
- 4 Cloud is timely: Moore's law faltering
- 5 Computing change is real!
- 6 Why can clusters seem painful?
- 7 How we make clusters less painful?
- 8 Things you should know about clusters
- 9 All about CJ
- 10 Announcements



Stanford University

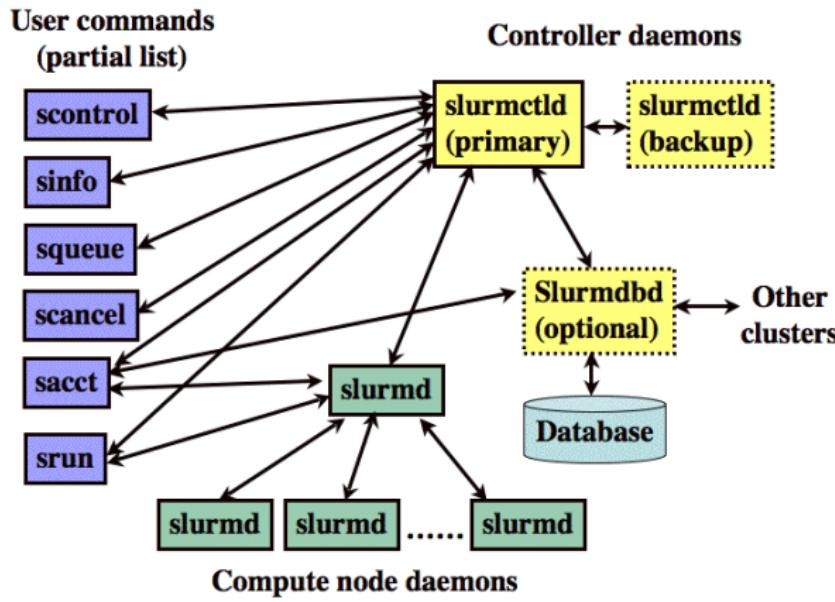
Some cluster terminology

- **cluster:** A collection of compute nodes (servers)
 - *node* (IP address)
 - *sockets* (typically 2-4)
 - *cores* (10 core/chip on Sherlock)
- **job :** a unit of work/execution comprised of **tasks/steps**
 - a job can use one or several cores (CPUs)
- **job scheduler:** application that controls execution of jobs
 - + a.k.a. batch scheduling, cluster management system, workload automation, batch queue system (BQS)
 - + examples: Portable Batch System(PBS), Sun Grid Engine (SGE), HTCondor, SLURM Workload Manager, Apache Mesos
- **job queue:** a data structure of jobs to run used by BQS



Stanford University

- Simple Linux Utility Resource Management
- Used by Sherlock and Farmshare clusters at Stanford



Stanford University

Typical submission script

```
$sbatch bashMain.sh  
Submitted batch job 15831
```

```
#!/bin/bash  
#SBATCH --mem=8G  
#SBATCH --time=48:00:00  
#SBATCH --partition donoho  
  
echo starting job $SLURM_JOBID  
  
module load matlab/R2016b  
  
matlab -nosplash -nodisplay <<HERE  
run('universality.m');  
quit;  
HERE  
  
echo ended job $SLURM_JOBID
```



Stanford University

Bad habits:

- Repetitive interactive logging on to the cluster
- Manual copy of your codes and script
- Manually using \$batch each time

Good habits:

- Automating your activities
- Occasional logging on to the cluster

Let's see next how we can build an automation system



Stanford University

Outline

- 1 Computing discontinuity
- 2 The Cloud explosion
- 3 Cloud wars
- 4 Cloud is timely: Moore's law faltering
- 5 Computing change is real!
- 6 Why can clusters seem painful?
- 7 How we make clusters less painful?
- 8 Things you should know about clusters
- 9 All about CJ
- 10 Announcements



Stanford University

CJ stands for ClusterJob



Documentation People Support

[Sign in](#)

[Sign up](#)

A photograph of a Stanford University building, likely the Hoover Tower, featuring a large mural on its facade depicting a scene from the Bible. The building has red-tiled roofs and arched windows. In the foreground, there is a dark overlay containing text and a call-to-action button.

Reproducible
High-throughput
Computational Research

[Sign up for Clusterjob](#)

Sign up to access your computations
anytime, anywhere.



“This is how it [computation] should be done.” – V. Morgenshtern



“Your software has made my life much easier.” – C. Chang



Stanford University

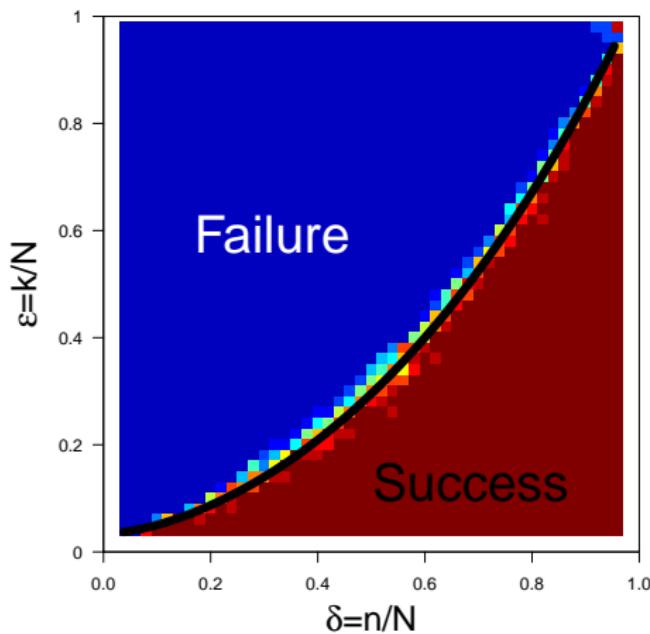
Our vision for CJ

- *push a button, fire and forget*
- *harvest, analyze and publish discovery*



How does CJ work, an example

Compressed Sensing Phase Transition Experiments:



Stanford University

How does CJ work, an example

Write a *simple and decipherable* MATLAB script:

```
% test.m
% This test code calculates the
% probability of successful
% reconstruction in compressed sensing.
% Author: Hatef Monajemi Nov 1 2016

file = 'results.txt';
delta    = 0.1:.1:.9;
epsilon = 0.02:0.02:0.98;
for i = 1:length(delta)
for j = 1:length(epsilon)
    pr = computeProb(delta, epsilon);
    fid = fopen(file,'at');
    fprintf(fid, '%3.2f,%3.2f,%3.2f\n', ...
              delta,epsilon,pr);
    fclose(fid)
end
end
```



Stanford University

How does CJ work, an example

Let CJ handle the rest.

- Submit 441 separate jobs by a simple command

```
$ cj parrun test.m corn -dep bin -m "Test PT"
```



Stanford University

How does CJ work, an example

Let CJ handle the rest.

- Check status of jobs

```
$ cj state 8ab7a5aa

pid 8ab7a5aaafab8232cc3da05a7814bed1d21dd0aa
remote_account: monajemi@sherlock.stanford.edu
1      10097772      COMPLETED
2      10097773      COMPLETED
3      10097774      COMPLETED
.
.
.
441    10097786      RUNNING
```



Stanford University

How does CJ work, an example

Let CJ handle the rest.

- Retrieve information

```
$ cj log

pid 8ab7a5aaafa1b8232cc3da05a7814bed1d21dd0aa
date: 2016-Oct-08 11:47:37 (GMT -07:00:00)
user: monajemi
agent: 2DCA5476-8197-11E6-B8C8-3A835C8A0BAC
account: monajemi@corn.stanford.edu
script: test.m
initial_flag: parrun

Test PT
```



Stanford University

How does CJ work, an example

Let CJ handle the rest.

- Easily harvest results

```
$ cj reduce results.txt 8ab7a5aa
```



Stanford University

How does CJ work, an example

Let CJ handle the rest.

- ... and many more functionalities

```
$ cj help
```



Stanford University

CJ demo



Stanford University

CJ project in more detail

Open source project:

<https://github.com/monajemi/clusterjob>

The screenshot shows the GitHub repository page for `monajemi/clusterjob`. The top navigation bar includes links for Features, Business, Explore, Marketplace, Pricing, and a sign-in/sign-up button. The repository name is displayed in the header, along with a 'Watch' button (1), a 'Star' button (3), and a 'Fork' button (1). Below the header, there are tabs for Code, Issues (8), Pull requests (0), Projects (0), Wiki, and Insights. A brief description of the project is provided: "ClusterJob: An automated system for painless and reproducible massive computational experiments" with a link to <http://clusterjob.org>. Key statistics are listed: 348 commits, 4 branches, 4 releases, 1 contributor, and BSD-3-Clause license. A dropdown menu shows the branch is set to master. Buttons for "New pull request", "Find file", and "Clone or download" are present. The commit history is shown in a table with columns for author, message, date, and file changes. The most recent commit is by `monajemi` on Feb 26, 2015, with the message "Initial commit".

Author	Message	Date
monajemi	fix	Latest commit 937757c 9 days ago
example	fix	9 days ago
src	&& added instaedof :	10 days ago
Initial commit	monajemi committed on Feb 26, 2015	ab21b80



Stanford University

A look inside, core modules

CJ is written in Perl

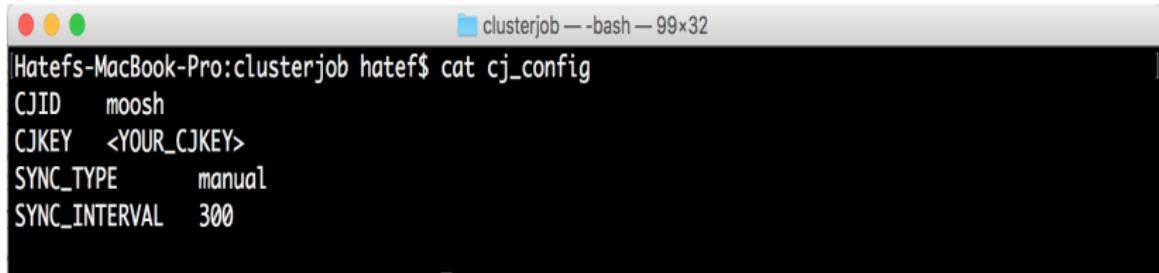
```
Hatefs-MacBook-Pro:clusterjob hatef$ ls .
CJlog      LICENSE    cj_config  example    ssh_config
INSTALL    README.md  dep.pl     src        todo
Hatefs-MacBook-Pro:clusterjob hatef$
Hatefs-MacBook-Pro:clusterjob hatef$
Hatefs-MacBook-Pro:clusterjob hatef$ ls src/
CJ          CJ.pl      CJ.pm      external    sanity_checks tmp
Hatefs-MacBook-Pro:clusterjob hatef$
Hatefs-MacBook-Pro:clusterjob hatef$
Hatefs-MacBook-Pro:clusterjob hatef$ ls -1 src/CJ.* ; ls -1 src/CJ/*
src/CJ.pl
src/CJ.pm
src/CJ/CJVars.pm
src/CJ/CJ_reduce.m
src/CJ/Get.pm
src/CJ/Install.pm
src/CJ/Matlab.pm
src/CJ/Python.pm
src/CJ/R.pm
src/CJ/Run.pm
src/CJ/Scripts.pm
src/CJ/Sync.pm
Hatefs-MacBook-Pro:clusterjob hatef$
```



Stanford University

Configuring CJ - I

- Your Cjid is unique
- Keep your CJkey **private** (used for Firebase DB).



```
Hatefs-MacBook-Pro:clusterjob hatef$ cat cj_config
Cjid      moosh
CJkey    <YOUR_CJKEY>
SYNC_TYPE    manual
SYNC_INTERVAL 300
```



Stanford University

Configuring CJ - II

Info of Clusters

```
clusterjob — bash — 96x35
[Hatefs-MacBook-Pro:clusterjob hatef$ cat ssh_config
[sherlock2]
Host      login.sherlock.stanford.edu
User      monajemi
Bqs       SLURM
Repo      /scratch/users/monajemi/CJRepo_Remote
MAT       matlab/R2017a
MATlib    ~/BPDN/CVX/cvx:~/mosek/7/toolbox/r2013a
Python    python/3.6
Pythonlib pytorch:torchvision:cuda80:scipy:matplotlib:torchvision:-c soumith
[sherlock2]

[corn]
Host      corn.stanford.edu
User      monajemi
Bqs       SGE
Repo      /farmshare/user_data/monajemi/CJRepo_Remote
MAT       matlab/r2016b
MATlib    ~/BPDN/CVX/cvx:~/mosek/7/toolbox/r2013a
Python    python/3.4.3
Pythonlib scipy
[corn]

[rice]
Host      rice.stanford.edu
User      monajemi
```



Stanford University

What happens when you issue parrun?

● Pseudo code of PARRUN... Part I: Preparation

```
# build job and directory info
my ($date,$ssh,$pid,$program_type,$localDir,$remoteDir) = run_common($self);

# setup conda env for python
$self->setup_conda_venv($ssh) if($program_type eq 'python');

# parse script out, find the loops, tags and ranges of indices
my $codeobj          = &CJ::CodeObj($self->{path},$self->{program},$self->{dep_folder});
my $parser           = $codeobj->parse();
my ($idx_tags,$ranges) = $codeobj->findIdxTagRange($parser,$self->{verbose});

# Check job is feasible
my $max_jobs = &CJ::max_jobs_allowed(...);
&CJ::err("Maximum jobs exceeded ...") unless ($max_jobs >= $totalJobs);

# build necessary submission scripts and reproducible code
$count = 0;
foreach my $loop (0..$nloops) {
    foreach my $i (0..$#idx_set) {
        $count++;
        &CJ::CodeObj("$localDir/$count",$program)->build_reproducible_script($runflag);
        &CJ::Scripts::make_par_shell_script($count,...);
        $master_script = &CJ::Scripts::make_master_script($master_script, $count, ...);
    }
}

# Compress and archive package
&CJ::system("tar -czf $Starfile $pid/");
```



Stanford University

What happens when you issue parrun?

- Pseudo code of PARRUN... Part II: Firing up

```
# send package to cluster
&CJ::system("rsync -arvz ${localDir}/${tarfile} $ssh->{account}:$remoteDir/");

# submit jobs
&CJ::system("ssh $ssh->{account} 'bash -l master.sh > $remoteDir/qsub.info");

# bring back submission info
&CJ::system("rsync -avz $ssh->{account}:$qsubfilepath $info_dir");

# parse submission info
($job_ids,$errors) = &CJ::read_qsub($local_qsub_info_file);
$self->_checkSubmitSuccess($job_ids,$errors, ...);

# record run info
my $runinfo={
    pid          => ${pid},
    user         => ${CJID},
    ...
};
# save record locally and remote DB
&CJ::add_record($runinfo);
&CJ::write2firebase($pid,$runinfo, ...);
```



Stamford University

What about the data?

- There is a number of applications for data transfer:
 - scp
 - rsync (used by CJ)
 - Globus
 - bbcp (from SLAC)
- ‘Comment-CJ’ directive for data already on the cluster:
%CJ -s ‘local-path’ ‘cluster-path’
#CJ -s ‘local-path’ ‘cluster-path’



Stanford University

- We are experiencing a *computing phase transition*
- Scientists need to adapt to this change by rethinking computing habits!
- CJ is an open-source software for *painless massive computing*
- You can use CJ for Matlab and Python jobs in your research.
- You can contribute to CJ project by forking it on GitHub.



Stanford University

Outline

- 1 Computing discontinuity
- 2 The Cloud explosion
- 3 Cloud wars
- 4 Cloud is timely: Moore's law faltering
- 5 Computing change is real!
- 6 Why can clusters seem painful?
- 7 How we make clusters less painful?
- 8 Things you should know about clusters
- 9 All about CJ
- 10 Announcements



Stanford University

- ❶ 2017 Gear Up for Social Science Data Extravaganza
Friday, October 27, 2017
Get the full program at: bit.ly/slgearup
- ❷ Homework



Stanford University

- *push a button, fire and forget*
- *harvest, analyze and publish breakthrough discovery*



Stanford University