

# Computing Systems for Metabolomics

Sean Wilkinson

University of Alabama at Birmingham

# My Personal Objectives

- Review basic concepts in computing systems
- Explain what the cloud “is” (and is not!)
- Cover practical issues your lab may face
- Entertain you while you nom on delicious food
- Plug my own research if time allows

# Simple Models of Computing

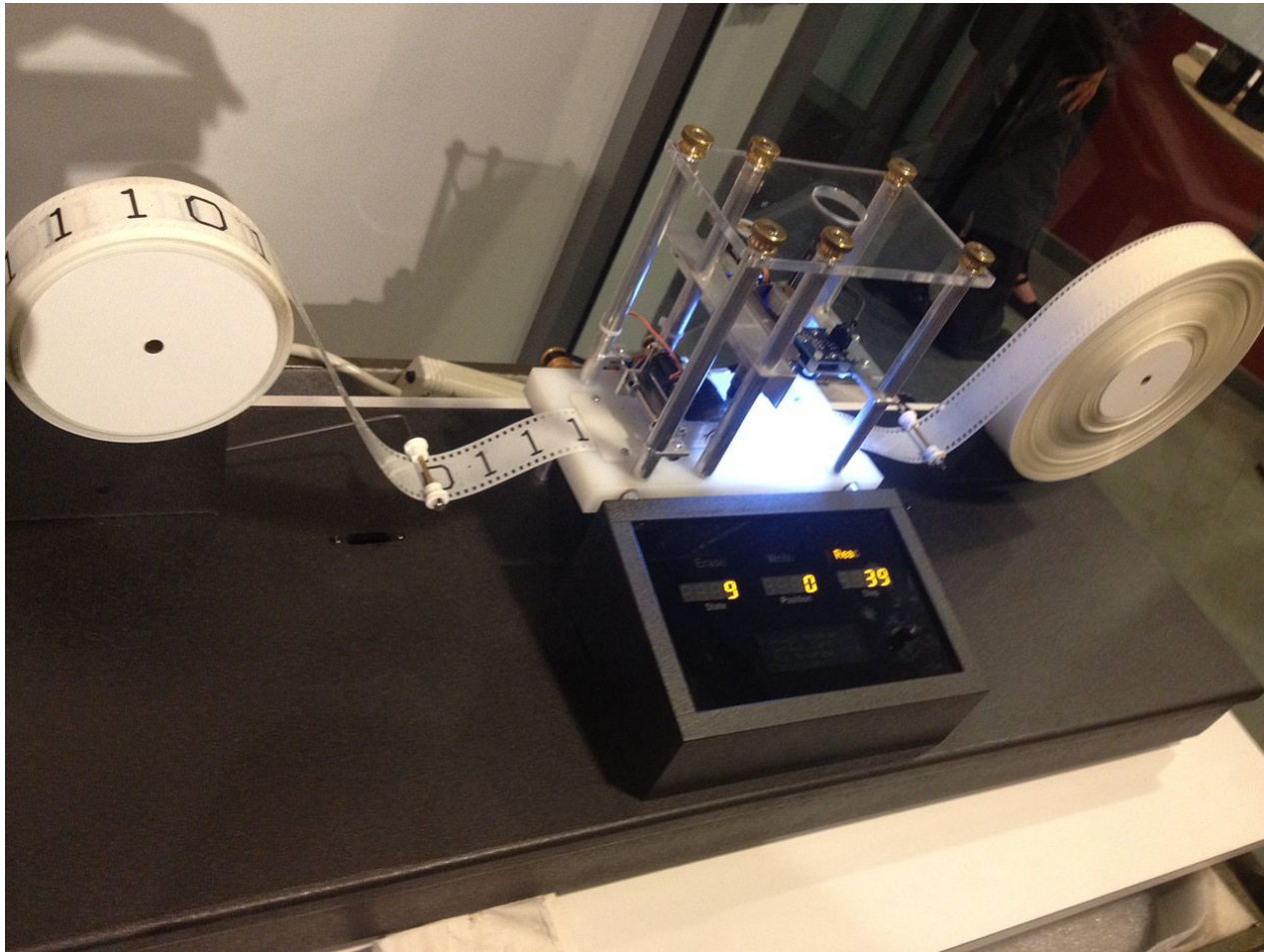
- Turing Machine
- Von Neumann Machine
- Wilkinson Machine (hehe):

$$y = f(x),$$

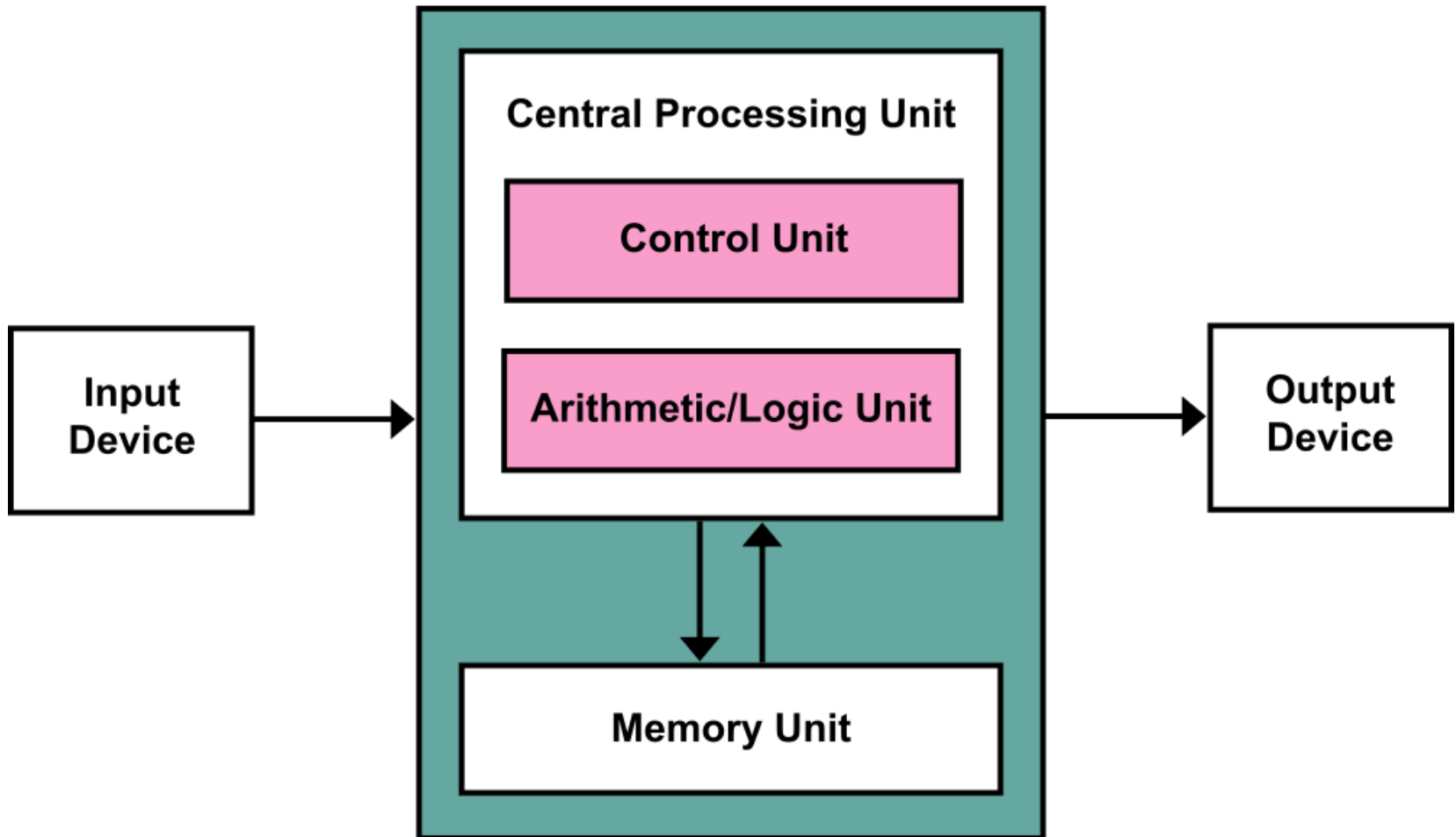
where

$f$  is an algorithm,  $x$  is input, and  $y$  is output.

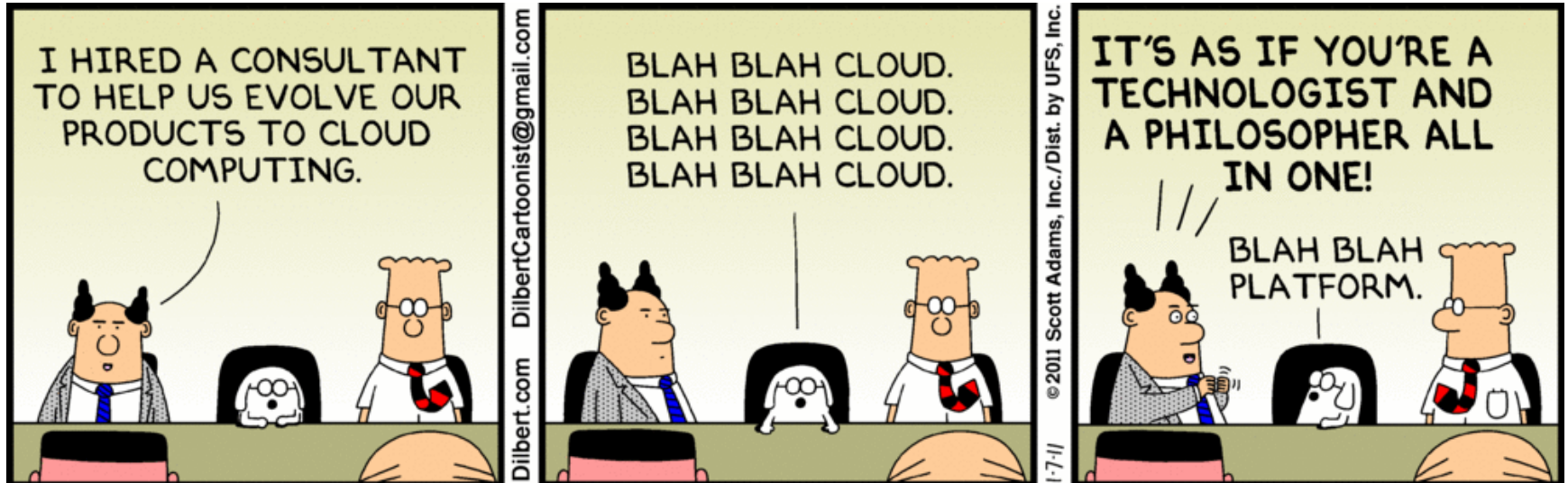
# Turing Machine



# Von Neumann Machine



# What is "The Cloud"?



# Kelvin-Helmholtz Clouds

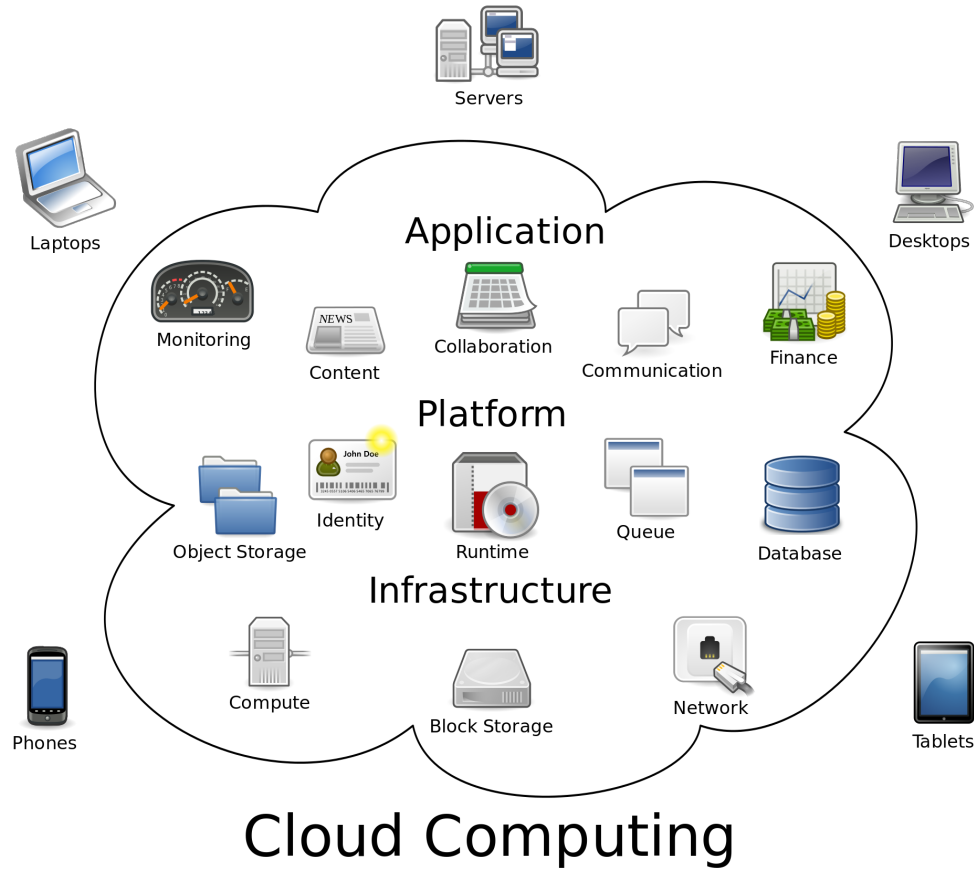


# Kelvin-Helmholtz Clouds

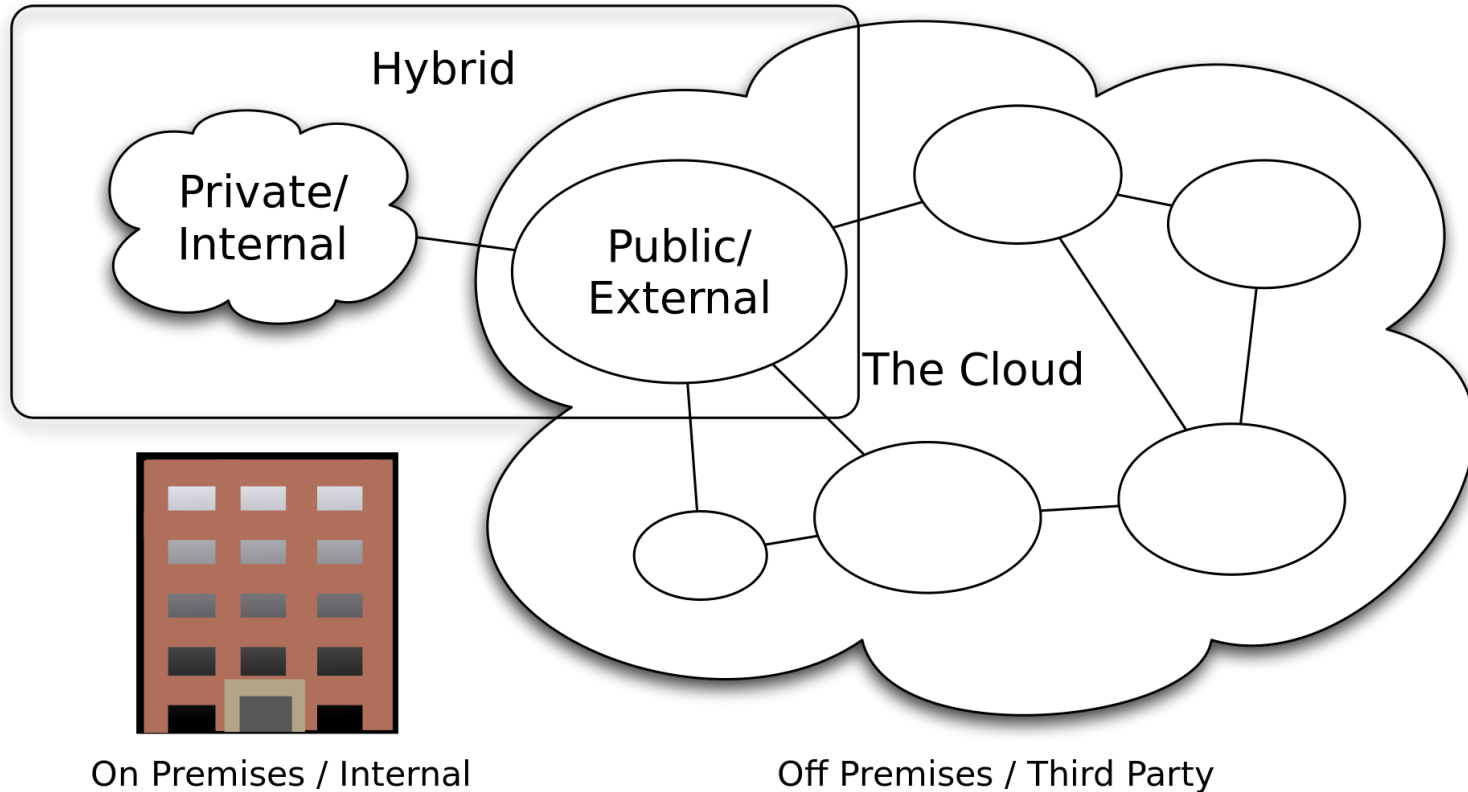




# According to Wikipedia ;-)



# Hybrid Cloud



## Cloud Computing Types

# Practical Problem

You know Big Data need Big Computers, but now you are asking yourself questions such as

How do I get one?

How do I use it?

How can I afford it?

# How do I get one?

- Buy a “real” workstation
  - Pay up front
  - Full control over every nitty, gritty detail
  - Full responsibility if the machine dies
- Lease “cloud” resources
  - Pay as you go
  - Significantly less control (sometimes zero)
  - Guaranteed performance and availability

# Should I use the cloud?

Probably!

Unless you are using your workstation at full capacity for more than 4 hours per day, you will save money by using the cloud.

Thus, we'll focus on cloud computing 😊

# WARNING

The cloud *may* lets you satisfy constraints in

- Governance
- Convenience
- Cost
- Performance

NOTE: These are arranged roughly in order of importance!

# Which cloud should I use?

The major players right now are

- Amazon
- Google
- Microsoft

Currently, I recommend Amazon, but keep an eye on IBM and Joyent. (Rackspace is toast.)

# How do I use it?

1. Get an Amazon account.
2. Log in to <https://console.aws.amazon.com/>.
3. Select the service you want to use 😊

Or, if you're a programmer, Amazon has really good Software Development Kits for most common languages. (They're really convenient!)



# Relevant Services

- Elastic Compute Cloud (EC2)
  - Create and destroy virtual machines instantly
  - Customize machines if you like that sort of thing, or use images from Amazon Marketplace if you don't (<https://aws.amazon.com/marketplace/>)
  - Pay only for what you use
  - Buy a Top500 machine for twenty minutes, then throw it away – EC2 makes HPC disposable!  
(<http://goo.gl/KggCa>)

# Relevant Services

- Elastic Block Store (EBS)
  - Create and destroy virtual hard drives instantly
  - Configurable from 1 gigabyte to 1 terabyte
  - EC2 instances can mount these as hard drives, but they perform more like network drives
  - Pay for what you use, plus the size of the provisioned storage (you pay until you destroy it)
  - Tricky to upload data directly to it

# Relevant Services

- Simple Storage Service (S3)
  - Create “buckets” for files and folders with full access control and web publishing
  - Files can be up to 5 terabytes each, and there is no limit to the number of objects you can store
  - 99.999999999% durability over a given year!
  - Everything can be set up using only a web browser, but you can also automate using Amazon’s SDK (<http://aws.amazon.com/tools/>)

# Relevant Services

- Glacier
  - Create “vaults” for immutable “archives” with full access control and web publishing
  - Archives are often TAR or ZIP files to save money
  - Archives can be up to 40 terabytes each, and there is no limit to the number of archives you can store within a vault
  - Think of this as the cloud analog of a tape drive
  - Same durability but *really* slow retrieval vs. S3

# Data Transfer

- You have three options:
  - For “small” files and fast networks, you can upload data directly from your web browser or use Amazon’s SDK to upload to S3 or Glacier
  - The SDK’s Multipart Upload API can be used for parallel and streaming uploads to S3 and Glacier
  - Physically mail your hard drive(s) to Amazon to upload to EBS, S3, or Glacier. Avoid this if you can!  
(<http://aws.amazon.com/importexport/>)

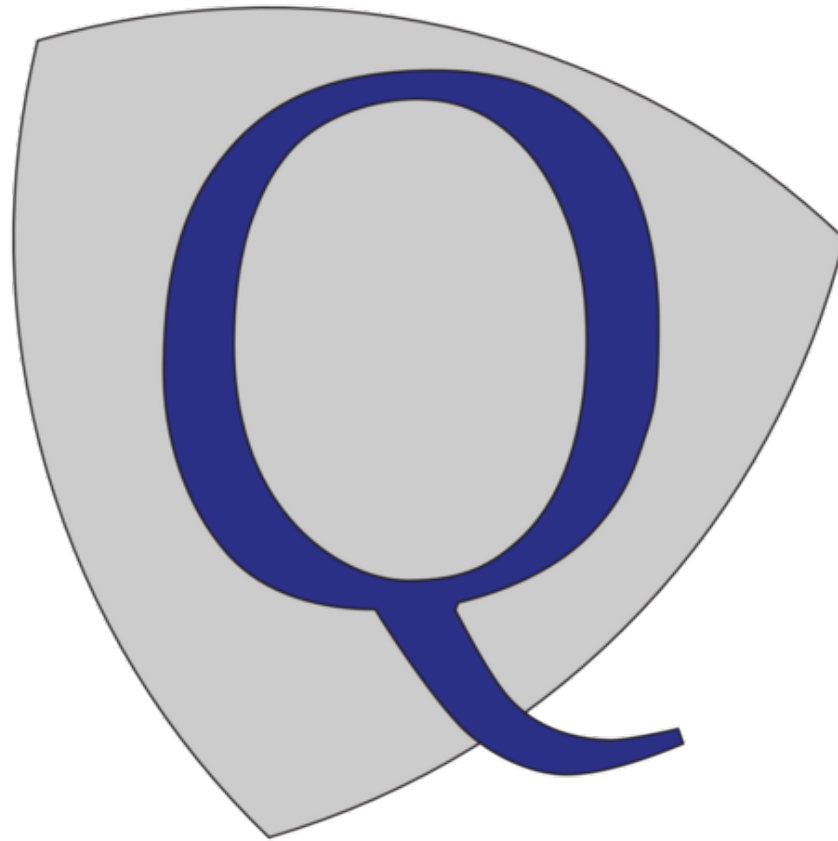
# Proposed (Untested) Solution

1. Upload data to S3 using the Multipart Upload API in Amazon's SDK.
2. Create an EC2 instance from a community-provided metabolomics image from Amazon Marketplace.
3. Process data on EC2 and save results to S3.
4. Archive data and/or results into Glacier.
5. Destroy unnecessary EC2 and S3 resources.

# Other clouds

- Google's offerings *may* outperform Amazon's, but you have to apply for an account. (Rant.)
- Microsoft Azure is strong but relatively new.
- Joyent's Manta is cool – for programmers.
- OpenStack is a huge pain in the butt. It hasn't lived up to expectations yet. Avoid it for now.
- Google Genomics and IBM Watson expose APIs usable by a World Wide Computer...

# Shameless Plug: QMachine





# World Wide Computing

The image shows a browser window displaying the QMachine website. The browser's address bar shows the URL `https://www.qmachine.org`. The website has a navigation menu with links for "QMachine", "Code", "Papers", "Status", "Version", and "Wiki". A yellow notification banner at the top states: "On May 27, QMachine: Commodity Supercomputing in Web Browsers was accepted for publication in BMC Bioinformatics. A link to the open-access paper will be available shortly." The main content area features the equation  $HPC + WWW = QM$  in large, bold letters. Below this, a paragraph explains: "Mix High Performance Computing with the World Wide Web and you'll get QMachine, a web service that can incorporate ordinary browsers into a World Wide Computer — without installing anything." A blue button labeled "Learn more »" is positioned below the text. The page is divided into two columns. The left column is titled "Submitter Example" and contains the instruction: "Open your browser's built-in console and enter the following:" followed by a code block: 

```
QM.submit(2, function (x) { return x + 2; }).print()
```

 and the note "You may need to volunteer to run it, though!". The right column is titled "Volunteer Example" and contains the instruction: "Check the box below to volunteer your machine's resources as part of QM's crowdsourced supercomputer. To specify a 'box', edit the text." Below this is a text input field containing the name "sean" and a checkbox.

QMachine Code Papers Status Version Wiki

On May 27, **QMachine: Commodity Supercomputing in Web Browsers** was accepted for publication in *BMC Bioinformatics*. A link to the open-access paper will be available shortly.

## HPC + WWW = QM

Mix High Performance Computing with the World Wide Web and you'll get QMachine, a web service that can incorporate ordinary browsers into a World Wide Computer — without installing anything.

[Learn more »](#)

### Submitter Example

Open your browser's built-in console and enter the following:

```
QM.submit(2, function (x) { return x + 2; }).print()
```

You may need to volunteer to run it, though!

### Volunteer Example

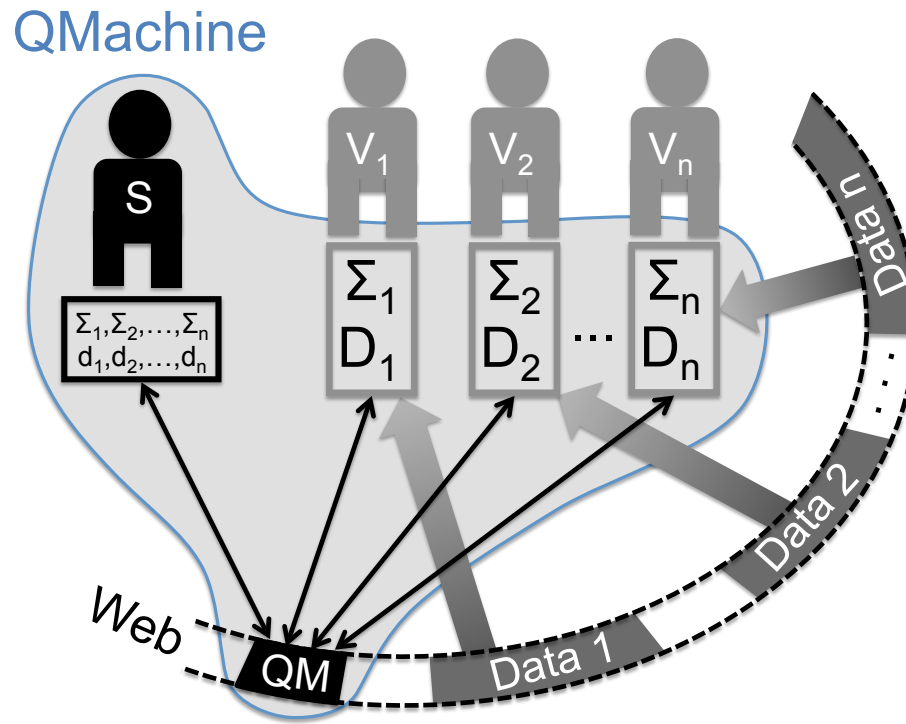
Check the box below to volunteer your machine's resources as part of QM's crowdsourced supercomputer. To specify a "box", edit the text.

# HPC + WWW = QM

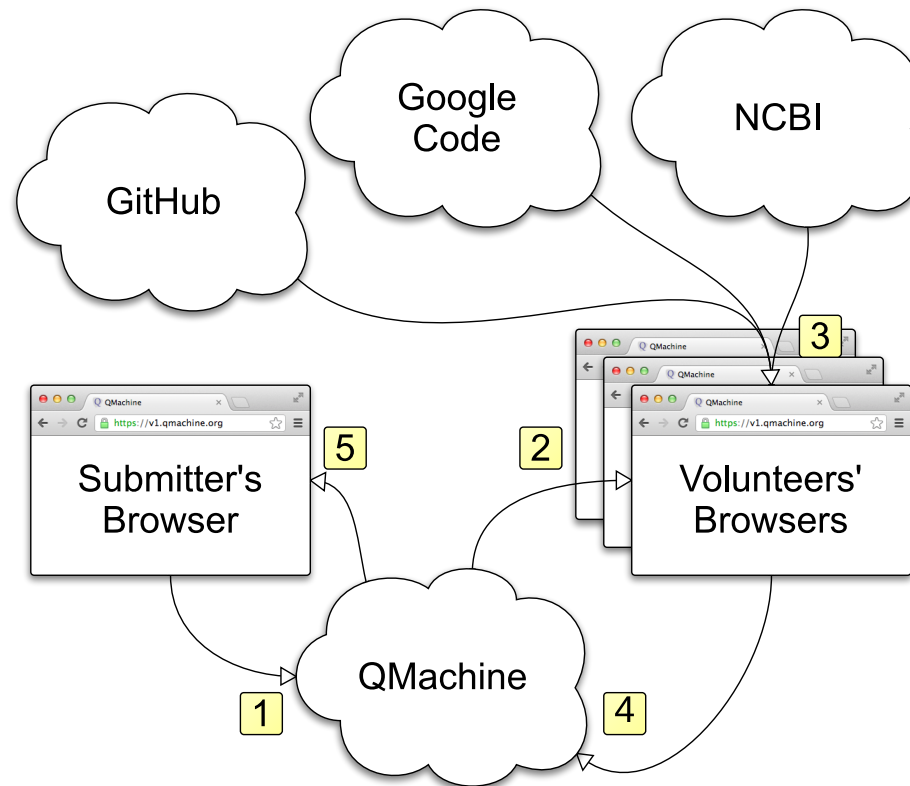
Mix High Performance Computing with the World Wide Web and you'll get QMachine, a web service that can incorporate ordinary browsers into a World Wide Computer – without installing anything.

<https://www.qmachine.org/>

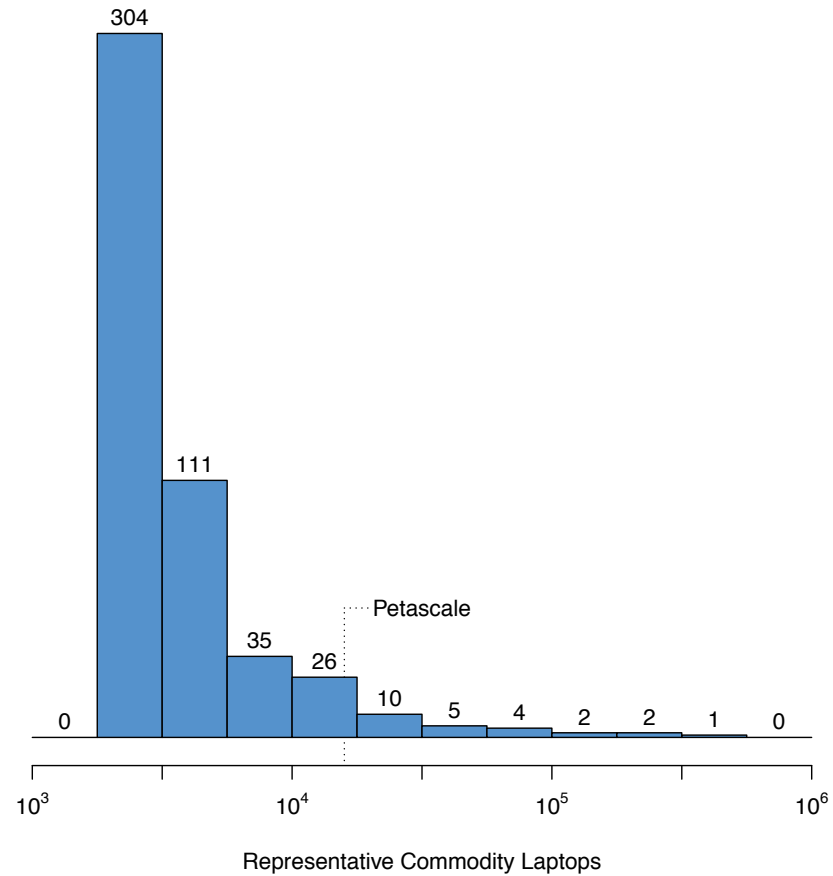
# QMachine's Architecture



# Supercomputing in Web Browsers



# Commodity Supercomputing



# Summary

- Computers just do what humans would do, but they do it faster and without complaining.
- Computation is now a commodity.
- Computers are everywhere, and someday, you won't have to install anything to get your work done.

The image features a background of vertical green lines of varying lengths, resembling the digital rain effect from the movie The Matrix. The word "THANKS" is written in a large, white, serif font with a bright green glow around it, centered horizontally across the middle of the image.

THANKS