

COMSM0010 Cloud Computing

Lecture 01

Introduction & Overview

Dave Cliff

Department of Computer Science
University of Bristol

csdtc@bristol.ac.uk



University of
BRISTOL

Housekeeping

Lectures:

- Tuesday 14:00-14:50 PHYS G12 MOTT
- Thursday 15:00-15:50 QUEENS QBI.15
- Unit Director: Dave Cliff
- M-level unit: you're expected to work largely independently

Housekeeping

Lectures:

- Tuesday 14:00-14:50 PHYS G12 MOTT
 - Thursday 15:00-15:50 QUEENS QBI.15
-
- Unit Director: Dave Cliff
 - Lecturers: Dave Cliff



- M-level unit: you're expected to work largely independently

Housekeeping

Lectures:

- Tuesday 14:00-14:50 PHYS G12 MOTT
 - Thursday 15:00-15:50 QUEENS QBI.15
-
- Unit Director: Dave Cliff
 - Lecturers: Dave Cliff & Dan Schien



- M-level unit: you're expected to work largely independently

Assessment

- 50% exam
 - Individual: you work alone
- 50% programming report: write a report of up to 10 pages (max)
 - Must be done individually (**no** group-work)
 - Prefer you to use free, open-source resources
 - NB Misuse of free cloud services can land you with credit-card bills!

If you are not yet comfortable with programming, this unit is definitely not yet a good choice for you

Deadline for coursework is Thursday 6th December 2019

Feedback/Formative Assessment deadlines (optional):

FA1 due Thursday 15th Oct 2018; FA2 (mock exam) Revision Week.

Time Budget

Lectures: 20 hours

Feedback Assessments: 2 hours

Programming Report: 48 hours

Independent Reading/Watching/Revising: 30 hours

TOTAL: 100 hours

NB There is way, way, **way** more reading material referenced than you could hope to read in 30 hours – the readings we reference are intended as a resource that will be useful in coming years; precisely what you choose to read will depend on your specific interests.

Time Management

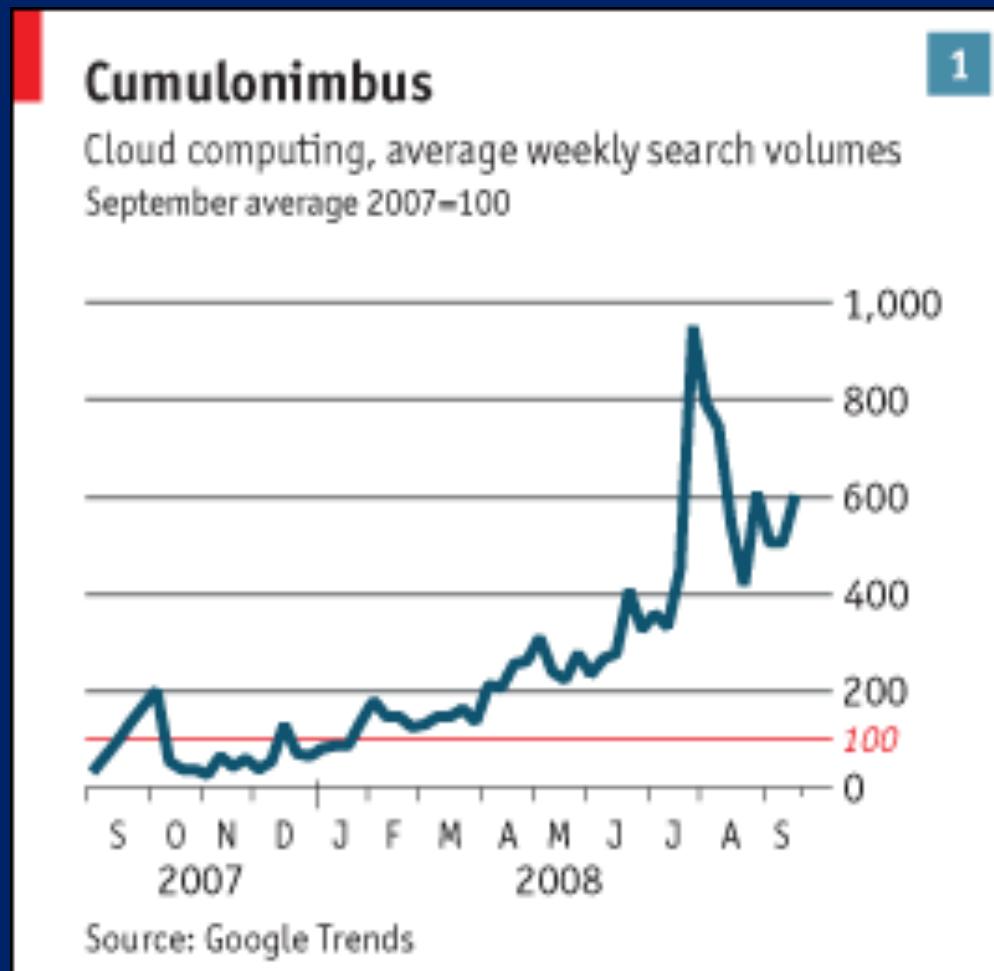
- Start work on the coursework assessment as soon as you can
- Use formative assessments FA1 & FA2 to get feedback on your work
- Be aware that the programming assignment allows you to score **some** points via attempting something that is deliberately specified as open-ended, and in principle you could spend several hundred hours extending and refining your system, but **that would simply not be a good use of your time**. You have only so much time to give to this unit (and that is true of all the other units that you are also studying), so **learn to recognize when it is time to stop**.
- Take care of yourself: your physical health **and** your mental health



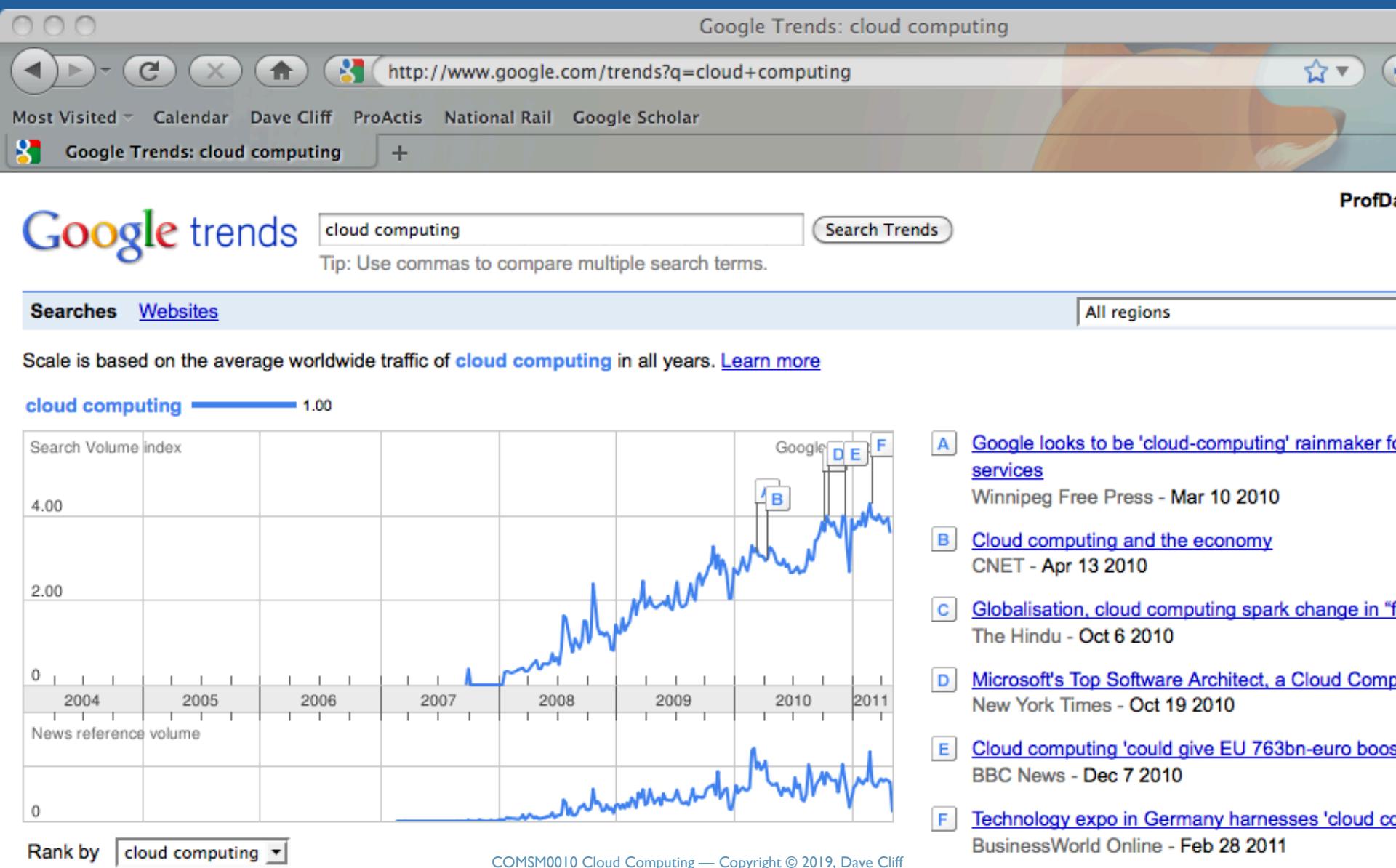
Bristol Futures: “The Challenge of Mental Health”
<https://www.youtube.com/watch?v=OtELvV8-OK0>

So, why all the fuss about cloud computing?

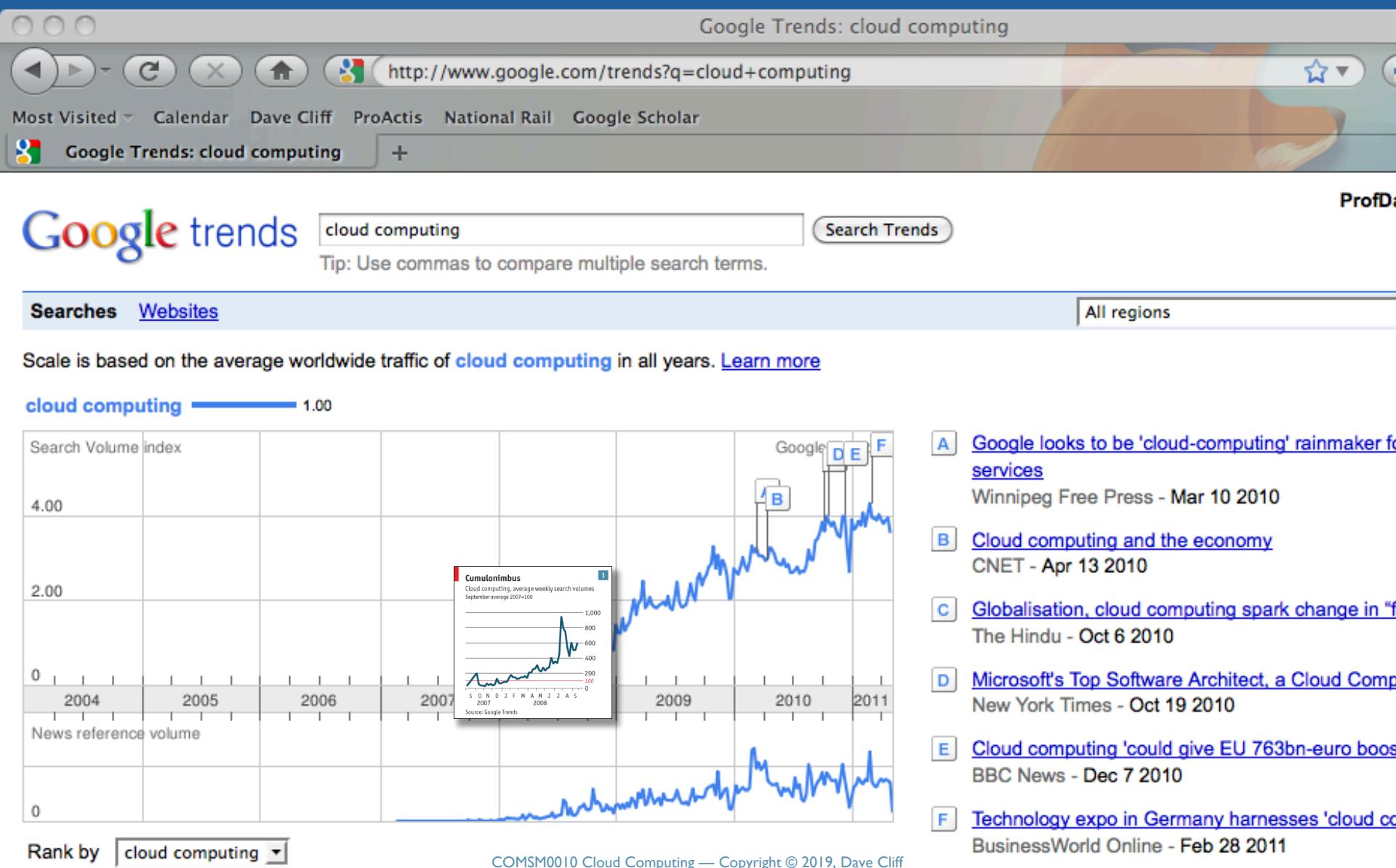
The Economist, Oct 23rd, 2008.



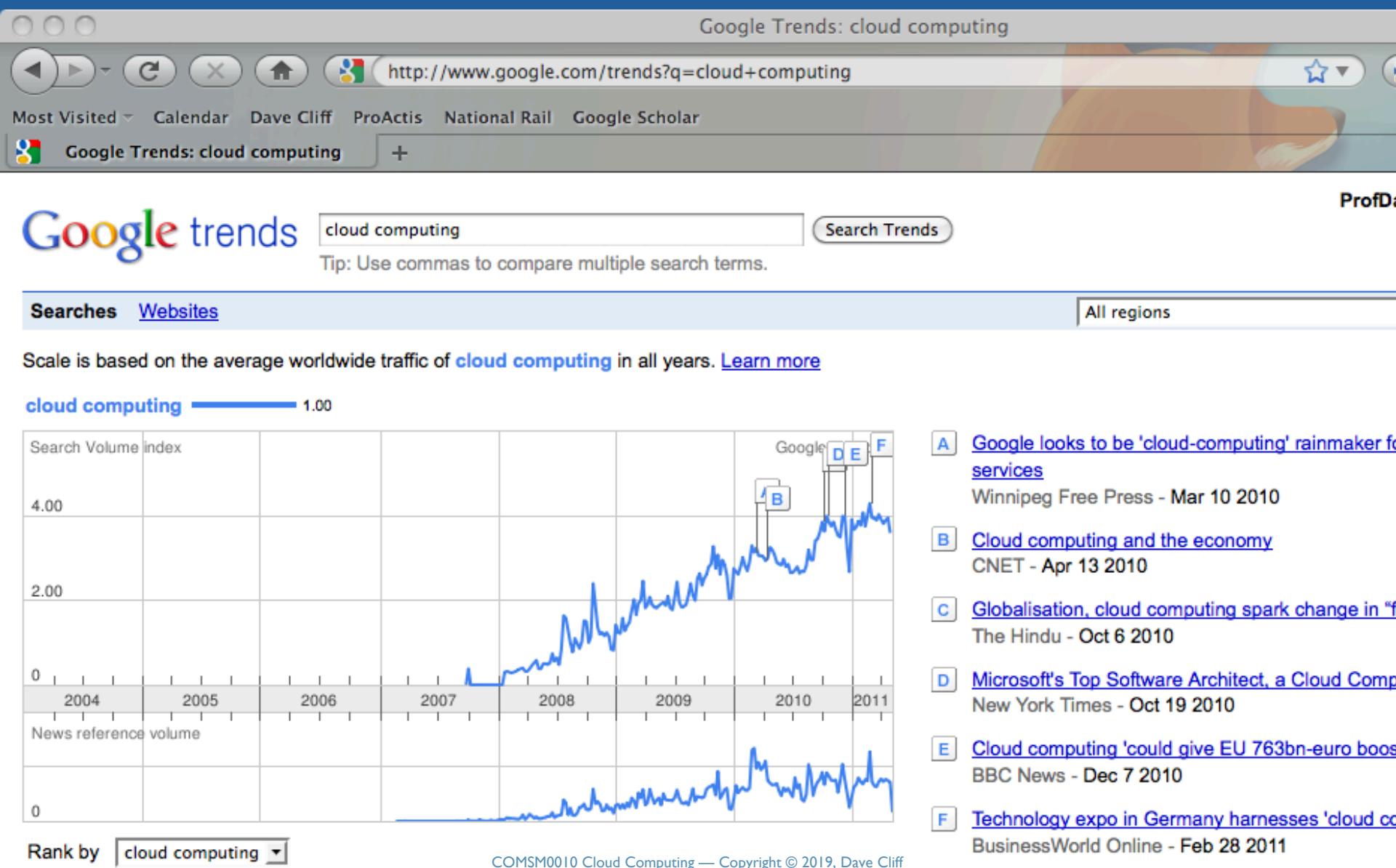
Dave's Mac, June 6th, 2011.



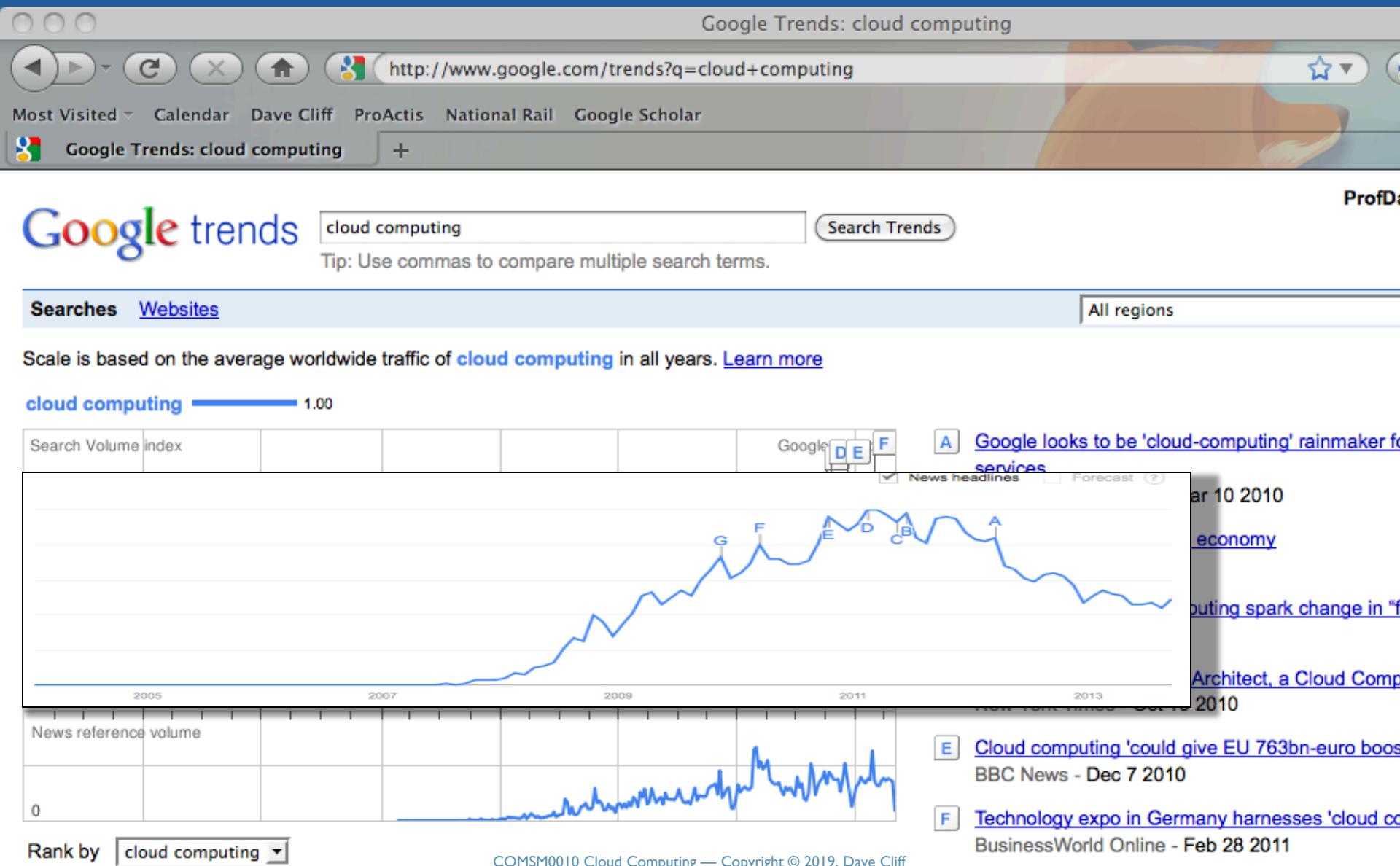
Dave's Mac, June 6th, 2011.



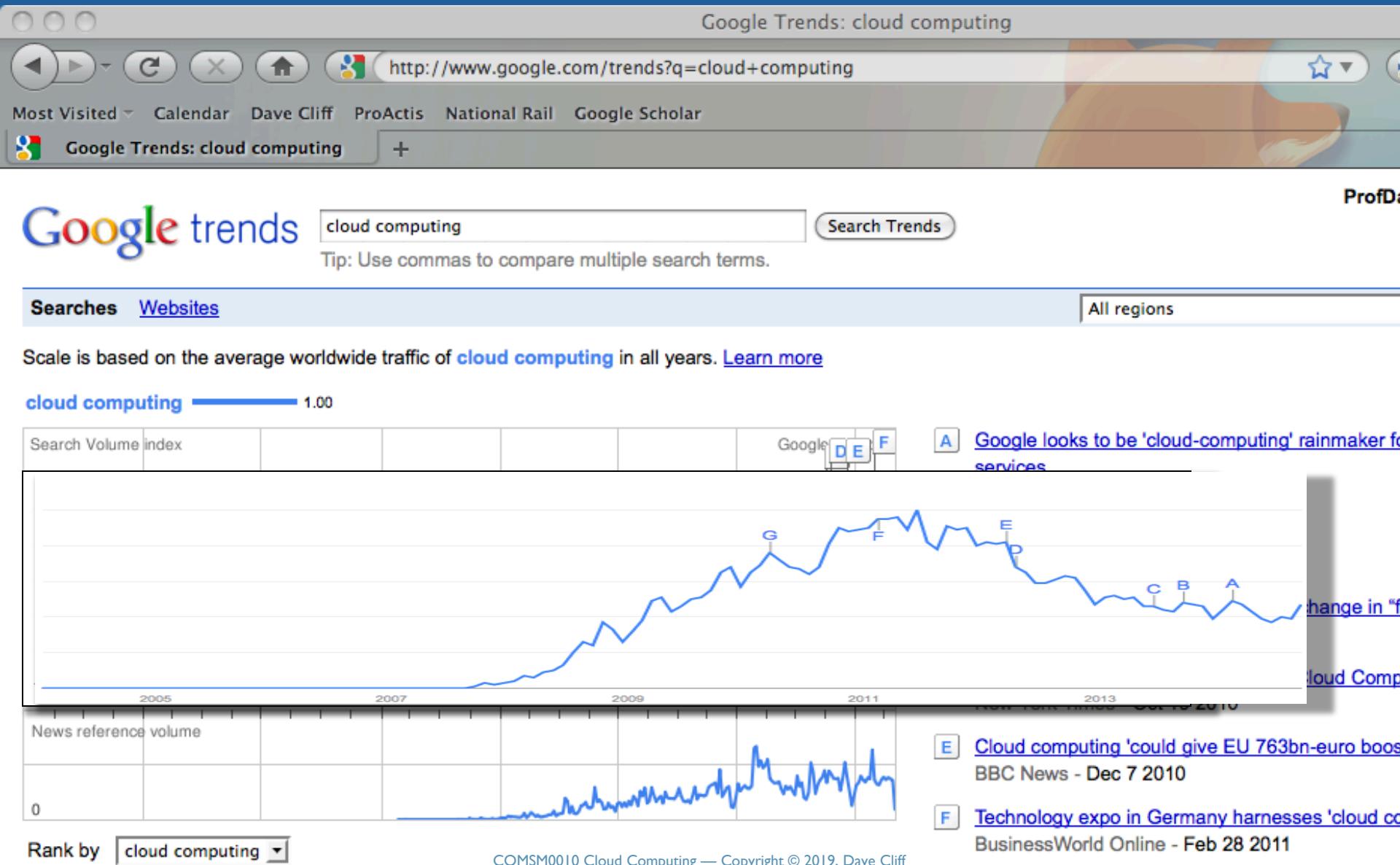
Dave's Mac, June 6th, 2011.



Dave's Mac, September 30th, 2013.



Dave's Mac, September 17th, 2014.



It's a once-a-decade kind of thing

- Mainframes
- Minicomputers
- Micros/PCs
- LAN/Distributed
- Internet/Web
- Utility/Service...



2008

For a non-technical introduction...



Remotely Hosted Services and "Cloud Computing"

Remotely Hosted Services
and "Cloud Computing"

Dave Cliff

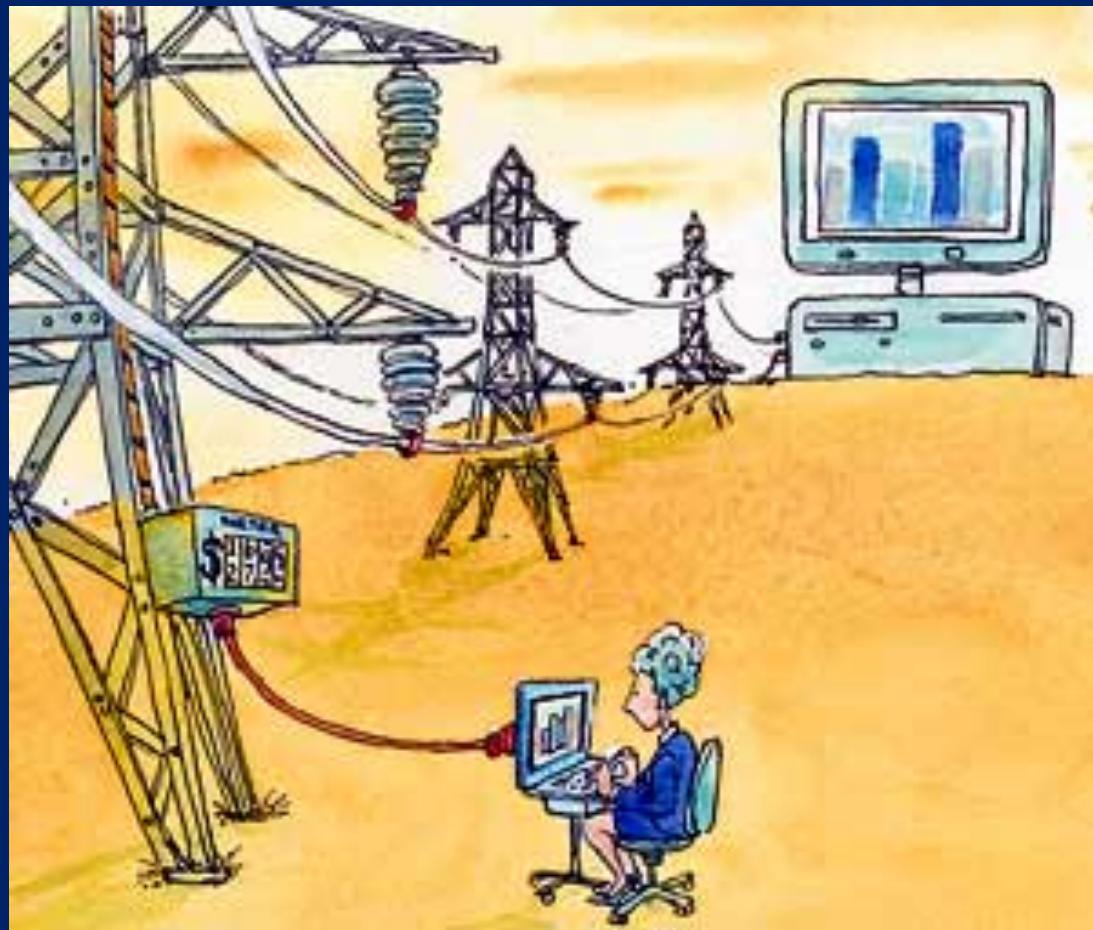
June 2010
© Becta 2010

<http://www.becta.org.uk>

NOT PROTECTIVELY MARKED

page 1 of 23

Let me tell you a story...



http://www.economist.com/displaystory.cfm?story_id=2352183

**A word about what goes on under the hood
(under the roof of the data centre)**

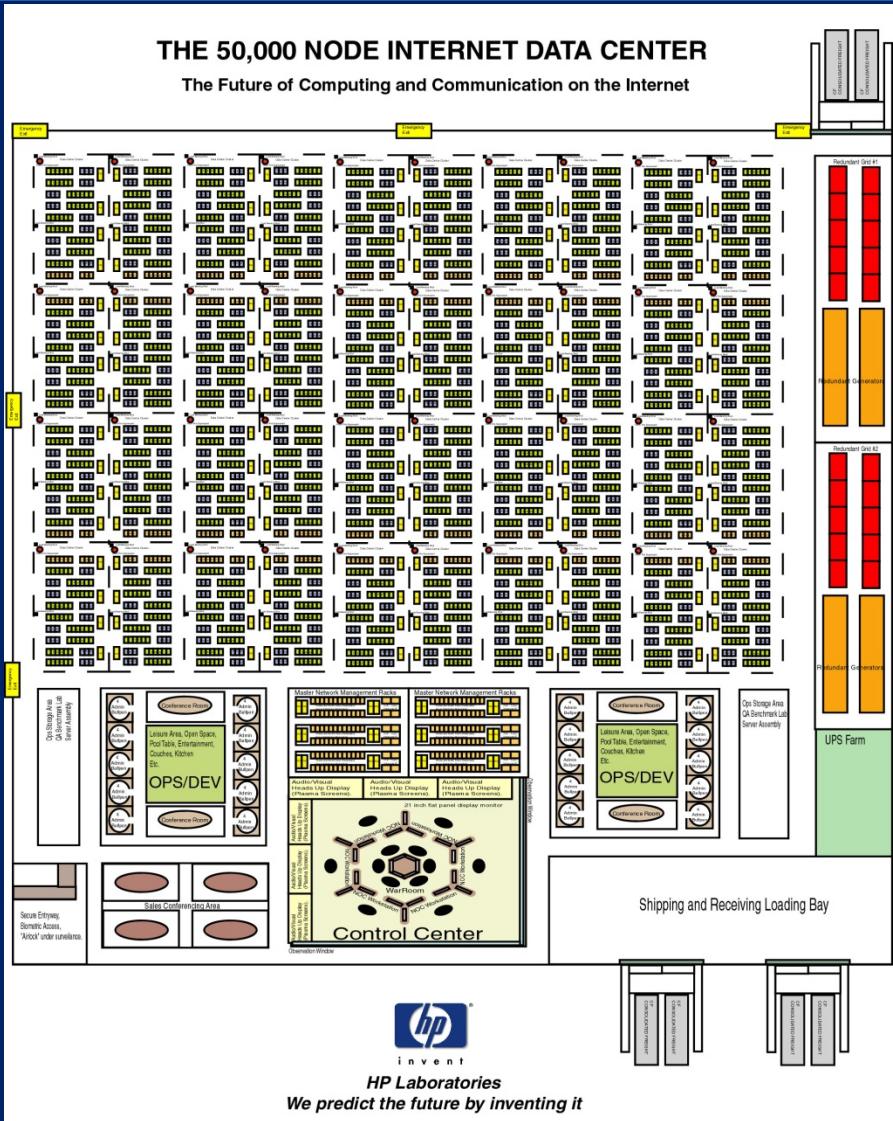
Normal Failure

Say that the servers you put in a cloud-data centre are guaranteed to have a 99.999% 3-year survival rate (“five-nines reliability”)

Some very rough maths:

- If you buy 10 servers, the probability that you have 100% of your servers still working 3 years later is $0.99999^{10}=99.990\%$.
- If you buy 50,000 servers, the probability that you have 100% of them still working 3 years later is $0.99999^{50000}=61\%$
- If you buy 500,000 servers, the probability that you have 100% of them still working 3 years later is $0.99999^{500000}=1\%$

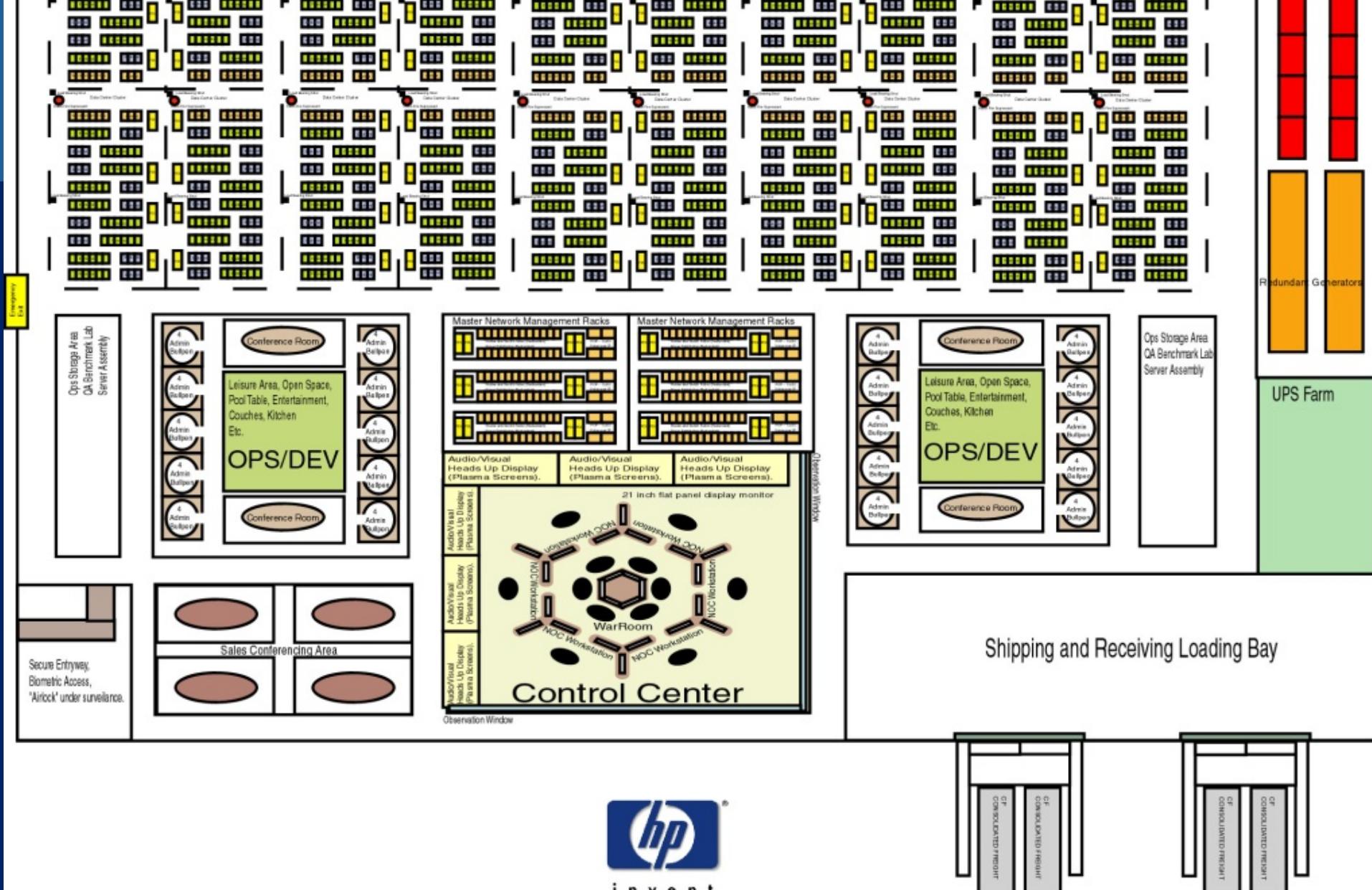
Example: HP Utility Data Centre, c.2001...



50,000 blade-servers
in a very big shed
with a very big air-conditioning system
Utility Data Center (UDC)

Just one shed in a global network...





HP Laboratories
We predict the future by inventing it

Modular Data Centres In Big Steel Boxes

The Register

Biting the hand that feeds IT

Cash 'n' Carrion Events Jobs

Hardware Software Music & Media Comms Security Management Science Odds & Sods

PCs & Chips Servers Storage Data Networking Semi-Coherent Computing Virtualization

crucial

The Register » Hardware »

Google nabs patent for Sun's Project Blackbox?

I own mobile data centers'

By Cade Metz in San Francisco → More by this author
Published Wednesday 10th October 2007 08:12 GMT
[Download free whitepaper - Effective collaboration for SMEs](#)

Google now owns a patent for data centers stuffed into shipping containers. You know, data centers like Sun's Project Blackbox.

The Mountain View outfit first filed for this "Modular Data Center" patent in December 2003, and today it was rubber stamped by the US Patent Office.

NEC. Empowering You Through Innovation

The patent describes a data center based on an "intermodal shipping container". That's a shipping container that's shipped by multiple carriers. "Rack on shelf mount"

Vendor Whitepapers: Free Download

- Step Towards the Future with Intel Enterprise IT Architecture Case Study: Melton Shire
- Virtualization eSymposium: Part 5 Robin Crewe on Desktop Virtualization
- The Rise of PDF Spam A MessageLabs Whitepaper: August 2007
- Using VMware VDI for thin-client desktops Case Study: Kane County
- Uncompromised Security in Virtual Machines

It's about scalability, not mobility

COMSM0010 Cloud Computing — Copyright © 2019, Dave Cliff

Page 23

And MSFT too (FT Feb 25, 2008)

FINANCIAL TIMES MONDAY FEBRUARY 25 2008

27

Companies | International

Microsoft predicts rise of the datacentre

Consolidation to blur distinctions

Central processing to replace desktops

By Richard Waters
in San Francisco

A handful of US companies is set to dominate the emerging market for "cloud computing", which will assume a central role in the information technology world, according to Steve Ballmer, chief executive of Microsoft.

His prediction – in an interview with the FT – points to a future in which the distinctions between software, hardware and internet companies fall away and a small number of big technology companies, each running a collection of vast datacentres, controls the IT landscape.

Cloud computing involves the centralised storage and processing of information – a shift that could reduce the role of desktop computers and the servers and other equipment run by many companies. This is forcing companies such as Microsoft to rethink their strategies.

Mr Ballmer predicted that a new super-group of tech companies would dominate the cloud computing market, each of them managing what amounts to a giant centralised computer made up of a number of big datacentres.

"Amazon has one. Rumours are Google will

have one. We've said we're going to have one," Mr Ballmer said.

The predicted emergence of super-powers in corporate computing points to a higher concentration of influence in the technology world. The winners would have the resources to operate at large scale, bringing down unit costs from competing to levels others cannot match.

Internet retailer Amazon was the first to start renting out capacity in its datacentres, selling storage and charging companies based on the number of transactions it processes. While Google has started offering online applications to companies, it has yet to start selling computing capacity in this way.

Mr Ballmer was speaking before the launch this week of one of Microsoft's most important new products this year, its latest software for servers known as Windows Server 2008. The software marks Microsoft's entry into the market for "virtualisation" technology.

The launch represents Microsoft's attempt to catch up with VMware, a software firm that has dominated the first phase of virtualisation.

Mr Ballmer has highlighted the initial battleground as a technology known as virtualisation. This involves separating computing tasks from the machines on which they run. Rather than having a single application running on a single operating system on each server, virtualisation makes it possible to run several applications and

Empire faces a next-generation adversary

FT INTERVIEW

Steve Ballmer is priming his forces to repel rival VMware in the battle to dominate virtualisation, says Richard Waters

Microsoft will this week fire the opening shot in a battle set to be every bit as important to its long-term future as its internet showdown with Google.

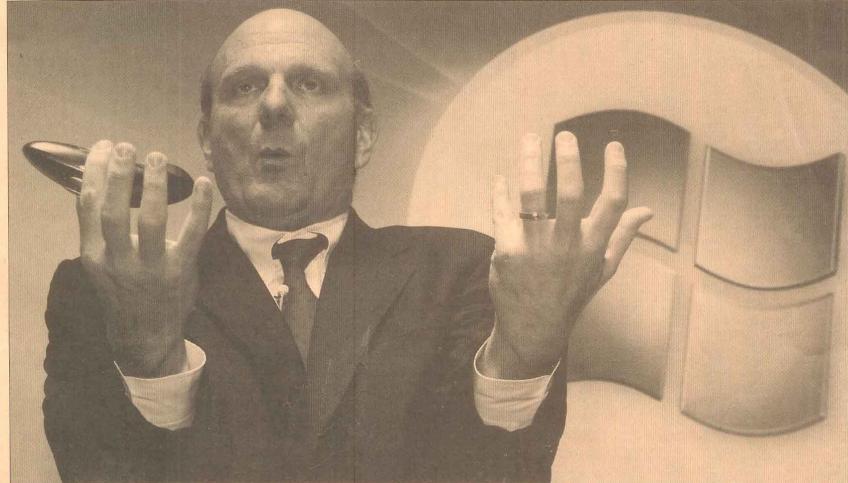
While the search war will help determine who rules the consumer web, this one will decide who reigns supreme in corporate datacentres and on office desktops – the heart of Microsoft's existing business.

At stake, says Steve Ballmer, chief executive, is nothing less than "the way computers get managed and deployed". That shift, he tells the Financial Times, "is a bigger opportunity for us than anyone – though, any time you have a major shift in this industry, it can't move fast enough".

Others put it more bluntly. According to some, the big change in corporate computing could lead to a new technology architecture where Microsoft's traditional source of dominance – its position in desktop and server operating systems – carries far less weight.

"It's incredibly dangerous to Microsoft longer term," says Michael Cherry, an analyst at Directions on Microsoft, an independent research firm.

The initial battleground is a technology known as virtualisation. This involves separating computing tasks from the machines on which they run. Rather than having a single application running on a single operating system on each server, virtualisation makes it possible to run several applications and



Revolution is coming: Steve Ballmer, chief executive of Microsoft, has said that there is no stopping the arrival of virtualisation

Bloomberg

operating systems simultaneously – each is known as a "virtual machine". Those virtual machines can then be moved between actual, real world servers.

That creates a new efficiency since most corporate servers are estimated to run at only 15-20 per cent of capacity. It also greatly improves the manageability of computing tasks since they can easily be backed up or moved from one machine to another.

For a company that has made its money selling single instances of Windows each time a new computer is installed, this represents a potentially significant change in business practice.

So far, Microsoft has responded by dropping the price for multiple instances of Windows running on single machines. As this new

way of running computers continues to spread, it will have negative as well as positive impact on Microsoft's business model, says Mr Ballmer although he adds: "You can't fight City Hall – virtualisation is going to happen."

Yet the challenge to Microsoft extends beyond the adjustments needed to the way it charges for its products.

At the heart of virtualisation is a piece of software known as a hypervisor, which sits between the virtual machines and the server, making it possible for the multiple computing tasks to all draw on the resources of the same piece of hardware. That new base level of software could assume a more central role, taking on more of the functions currently undertaken by operating systems.

In much the same way, Microsoft's own operating system began as a small "microkernel" before expanding to become a dominant computing platform, says Mr Cherry. It is this potential that represents the biggest long-term challenge to Microsoft, he adds.

"Any piece of software can morph into any other piece of software," concedes Mr Ballmer, though he largely brushes off the comparison.

As in the fight with Google, this latest battle also involves an adversary whose early dominance of a new market has been turned heads in Silicon Valley. VMware was already gaining a reputation in corporate datacentres before its red-hot initial public offering last summer brought it to wider attention. The possibility that it could one day

assume a central role in the new computing architecture has made it the world's fourth most valuable independent software company.

Also echoing its fight with Google, Microsoft starts well behind as it takes on VMware – although in this case it stands a far better chance of catching up. Microsoft's response comes with the launch this week of Windows Server 2008. The company believes it has a big advantage in its favour as it squares up against VMware. One is the ability to bundle the software with the new Windows server so it gets wide distribution.

"It goes in Windows, so it makes it hard for a third party," says Mr Cherry.

Microsoft will add \$40 to the price of the new server software to include the virtu-

alisation feature. All in, the cost of Microsoft's software will be a third or less than that of VMware, says Mr Ballmer.

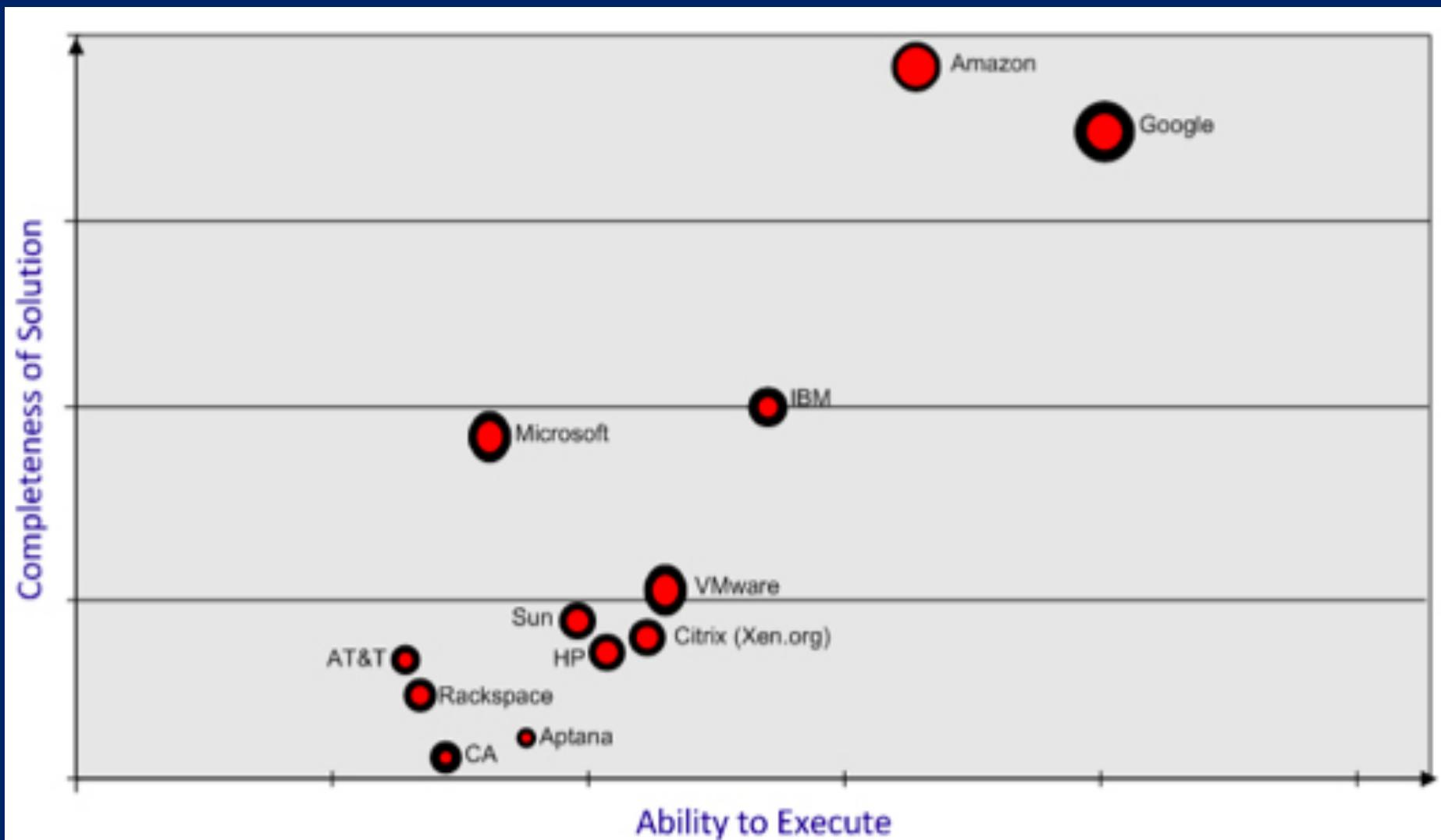
A second advantage for Microsoft is its existing position in corporate datacentres. Thanks to the market share that Windows has already built up on servers, its software tools are used by many IT staffs to manage their "real-world" machines. Those tools will now be able to manage the "virtual machines" that run in virtualised servers. Though VMware dominates the virtual world, it cannot match those real-world capabilities.

Yet with fewer than 10 per cent of corporate servers estimated to have been virtualised and with VMware moving fast to build on its early lead, the race is still at a very early stage.

See Editorial Comment

October 2009 survey data from Evans Data

http://www.theregister.co.uk/2009/10/06/evans_data_developer_cloud_perception_survey/



MSFT Chicago DataCentre

<http://www.silicon.com/technology/hardware/2009/11/04/photos-inside-a-microsoft-datacentre-39626370/2/#story>



MSFT Chicago DataCentre

<http://www.silicon.com/technology/hardware/2009/11/04/photos-inside-a-microsoft-datacentre-39626370/2/#story>



HP POD

Alternative Thinking for
the Data Center

Mix-and-match IT

Manage Growth

Delivery

Conventional IT

Deployment and Support



IBM Portable Modular Data Center



A Clever Backronym (*reverse-engineered acronym*)

CLOUD means:

- **C**ommon
- **L**ocation-independent
- **O**nline
- **U**tility-provisioned
- on-**D**emand
- (from Joe Weinman, AT&T, November 2008: <http://www.joeweinman.com/>)
- This neatly summarizes the official **NIST Definition of Cloud Computing**:
http://www.nist.gov/manuscript-publication-search.cfm?pub_id=909616

NIST's Essential Characteristics of Cloud Computing

- *On-demand self-service* – a consumer can unilaterally provision computing capabilities as needed automatically without requiring human interaction with each service provider.
- *Broad network access* – capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms.
- *Resource pooling* – provider's computing resources are pooled to serve multiple consumers, with different physical and virtual resources dynamically (re)assigned according to demand.
- *Rapid elasticity* – capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand.
- *Measured service* – cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service.

A common (and useful) initial stratification: *aaS

- **SaaS: Software as a Service**
 - End-user application software functionality remotely delivered over the internet/web
 - You used to buy this as a CD/DVD in a shrink-wrapped cardboard box
-
- **PaaS: Platform as a Service**
 - Developer application software (middleware) functionality, remotely accessible
 - E.g. provide a particular combination of OS, web-server, data-base, & scripting
 - Popular instance is the free “LAMP Stack” – Linux, Apache, MySQL, and PHP(/Perl/Python)
-
- **IaaS: Infrastructure as a Service**
 - IT infrastructure, almost always virtualized, remotely accessible, “bare metal”

A common (and useful) initial stratification: *aaS

- SaaS: Software as a Service
 - End-user application software functionality remotely delivered over the internet/web
 - You used to buy this as a CD/DVD in a shrink-wrapped cardboard box
-
- PaaS: Platform as a Service
 - Developer application software (middleware) functionality, remotely accessible
 - E.g. provide a particular combination of OS, web-server, database, & scripting
 - Popular instance is the free “LAMP Stack” – Linux, Apache, MySQL, and PHP(/Perl/Python)
-
- IaaS: Infrastructure as a Service
 - IT infrastructure, almost always virtualized, “bare metal”

A common (and useful) initial stratification: *aaS

- SaaS: Software as a Service
 - End-user application software functionality remotely delivered over the internet/web
 - You used to buy this as a CD/DVD in a shrink-wrapped cardboard box
-
- PaaS: Platform as a Service
 - Developer application software (middleware) functionality, remotely accessible
 - E.g. provide a particular combination of OS, web-server, database, etc.
 - Popular instance is the free “LAMP Stack” – Linux, Apache, MySQL, and PHP(/Perl/Python)
-
- IaaS: Infrastructure as a Service
 - IT infrastructure, almost always virtualized, “bare metal”



A common (and useful) initial stratification: *aaS

- SaaS: Software as a Service
 - End-user application software functionality remotely delivered over the internet/web
 - You used to buy this as a CD/DVD in a shrink-wrapped cardboard box
-
- PaaS: Platform as a Service
 - Developer application software (middleware) functionality, remotely accessible
 - E.g. provide a particular combination of OS, web-server, database, & scripting
 - Popular instance is the free “LAMP Stack” – Linux, Apache, MySQL, and PHP(/Perl/Python)
-
- IaaS: Infrastructure as a Service
 - IT infrastructure, almost always virtualized, “bare metal”

GOOGLE COMPUTE ENGINE

A common (and useful) initial stratification: *aaS

- SaaS: Software as a Service
 - End-user application software functionality remotely delivered over the internet/web
 - You used to buy this as a CD/DVD in a shrink-wrapped cardboard box
-
- PaaS: Platform as a Service
 - Developer application software (middleware) functionality, remotely accessible
 - E.g. provide a particular combination of OS, web-server, data-base, & scripting
 - But you still need to “fiddle” with Linux, Apache, MySQL, and PHP/Perl/Python)
- 
- IaaS: Infrastructure as a Service
 - Infrastructure, almost always virtualized, “bare metal”

A common (and useful) initial stratification: *aaS

- SaaS: Software as a Service
 - End-user application software functionality remotely delivered over the internet/web
 - You used to buy this as a CD/DVD in a shrink-wrapped cardboard box
-
- PaaS: Platform as a Service
 - Developer application software (middleware) functionality, remotely accessible
 - E.g. provide a particular combination of OS, web-server, data-base, & scripting
 - Popular examples include “Google Stack” (Linux, Apache, MySQL, and PHP/Perl/Python)
 - **GOOGLE APP ENGINE**
 - **AMAZON APPSTREAM, DYNAMODB, RDS**
 - IaaS: Infrastructure as a Service
 - IT infrastructure, almost always virtualized, modern accessible, “bare metal”
 - **AMAZON EC2, S3, SQS**
 - **GOOGLE COMPUTE ENGINE**

A common (and useful) initial stratification: *aaS

- SaaS: Software as a Service
 - End-user application software functionality, remotely delivered over the internet/web
 - You used to buy this as a CD/DVD in a shrink-wrapped cardboard box
- PaaS: Platform as a Service
 - Developer application software (middleware) functionality, remotely accessible
 - E.g. provide a particular combination of OS, web-server, data-base, & scripting
 - Popular examples include “Google Stack” (Linux, Apache, MySQL, and PHP/Perl/Python)
 - **GOOGLE APP ENGINE**
 - **AMAZON APPSTREAM, DYNAMODB, RDS**
 - Popular examples include “Amazon Stack” (Linux, Apache, MySQL, and PHP/Perl/Python)
 - **AMAZON EC2, S3, SQS**
- IaaS: Infrastructure as a Service
 - IT infrastructure, almost always virtualized, “bare metal”
 - **GOOGLE COMPUTE ENGINE**

A common (and useful) initial stratification: *aaS

- SaaS: Software as a Service
 - End-user application software functionality, remotely delivered over the internet/web
 - You used to buy this as a CD/DVD in a shrink-wrapped cardboard box
- PaaS: Platform as a Service
 - Developer application software (middleware) functionality, remotely accessible
 - E.g. provide a particular combination of OS, web-server, data-base, & scripting
 - Popular examples include “App Engine” (Google), “Lambda” (AWS), and PaaS (Perl/Python)
- IaaS: Infrastructure as a Service
 - IT infrastructure, almost always virtualized, “bare metal”

EVERYONE

REALLY, EVERYONE

GOOGLE APP ENGINE

AMAZON APPSTREAM, DYNAMODB, RDS

GOOGLE COMPUTE ENGINE

and then...

A common (and useful) initial stratification: *aaS

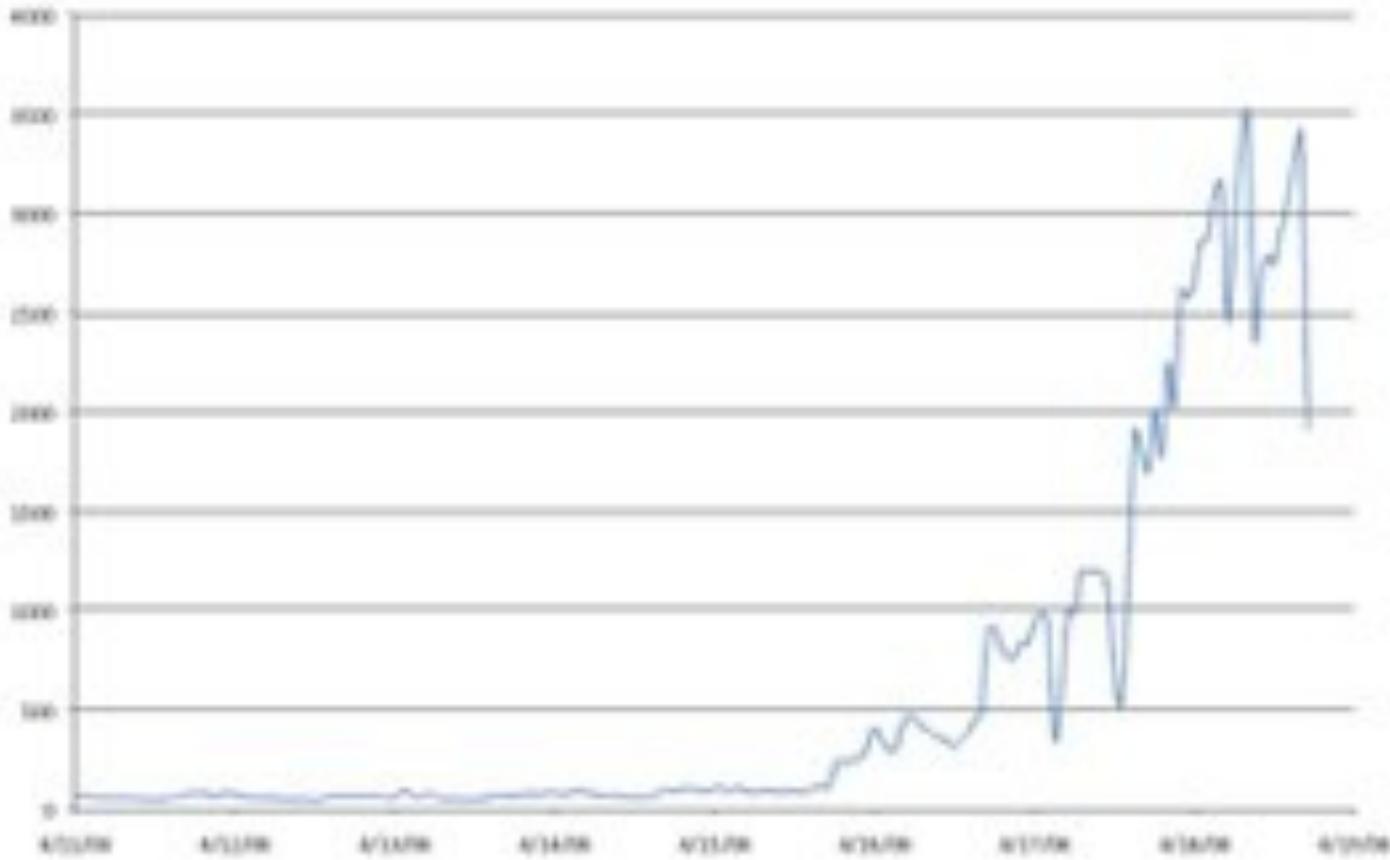
- **SaaS: Software as a Service**
 - End-user application software functionality delivered over the internet/web
 - You used to buy this as a CD/DVD in a shrink-wrapped cardboard box
- **PaaS: Platform as a Service**
 - Developer application software (middleware) functionality remotely accessible
 - E.g. provide a particular combination of OS, web-server, data-base, & scripting
 - Popular providers are Google App Engine (Java, Python), Amazon Lambda (Python), Microsoft Azure (C#, F#), and Parse (JavaScript/Python)
- **IaaS: Infrastructure as a Service**
 - IT infrastructure, almost always virtualized or more accurately, “bare metal”
 - Amazon EC2, S3, SQS
 - Google Compute Engine

A common (and useful) initial stratification: *aaS

- FaaS: Function as a Service (AKA “Serverless”)
- No server processes visibly running. Pay only for the time spent executing a function.
- Unlike PaaS, scale out without increasing number of servers.
- Amazon Lambda is the best-known example (cf Google Cloud Fns, MSFT Azure Fns).
- SaaS: Software as a Service
 - End-user application software functionality delivered over the internet/web
 - You used to buy this as a CD/DVD in a shrink-wrapped cardboard box
- PaaS: Platform as a Service
 - Developer application software (middleware) functionality remotely accessible
 - E.g. provide a particular combination of OS, web-server, data-base, & scripting
 - Popular providers include Google App Engine, Amazon AppStream, Amazon RDS, Amazon DynamoDB, Microsoft SQL Server, MySQL, PostgreSQL, Oracle Database, Microsoft Azure, IBM DB2, and Python (Django, Flask, etc.)
- IaaS: Infrastructure as a Service
 - IT infrastructure, almost always virtualized or more-or-less “bare metal”

Early examples

Animoto: This Week's EC2 Instance Usage



Animoto: This Week's EC2 Instance Usage



See Jeff Bezos (Amazon CEO) 2008 talk about this
<https://www.youtube.com/watch?v=ulc-VB-ke9o>

NewYorkTimes.com, Nov 1st, 2007: tiff-to-pdf

<http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/>

Self-service, Prorated Super Computing Fun! - Open Blog - NYTimes.com - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://open.blogs.nytimes.com/2007/11/01/self-service-prorated-super-computing-fun/ new york times" digitize ama

Most Visited Weather Multimap Calendar Milk: DC PhD reviews Elgg for STS

Google "new york times" digit

Welcome to TimesPeople Get Started TimesPeople Lets You Share and Discover the Best of NYTimes.com

HOME PAGE TODAY'S PAPER VIDEO MOST POPULAR TIMES TOPICS

The New York Times Monday, October 26, 2009

WORLD U.S. N.Y. / REGION BUSINESS TECHNOLOGY SCIENCE

IN THE EXCHANGING WORLD. YOUR EXCHANGE IS YOUR ALIANCE Learn more

OPEN

All the Code That's Fit to `printf()`

November 1, 2007, 5:30 PM

Self-service, Prorated Super Computing Fun!

By DEREK GOTTFRID

As part of [eliminating TimeSelect](#), [The New York Times](#) has decided to make all the public domain articles from 1851-1922 available free of charge. These articles are all in the form of images scanned from the original paper. In fact from 1851-1980, all 11 million articles are available as images in PDF format. To generate a PDF version of the article takes quite a bit of work — each article is actually composed of numerous smaller TIFF images that need to be scaled and glued together in a coherent fashion.

COMSM0010 Cloud Computing — Copyright © 2019, Dave Cliff

Previously we had generated all the PDFs dynamically. This approach had

...if I used only four machines, it could take some time to generate all 11 million article PDFs. But thanks to the swell people at Amazon, I got access to a few more machines and churned through all 11 million articles in just under 24 hours using 100 EC2 instances, and generated another 1.5TB of data to store in S3. (In fact, it work so well that we ran it twice, since after we were done we noticed an error in the PDFs.)

CO
CHANNEL
NOW PLAYING

AUTOS

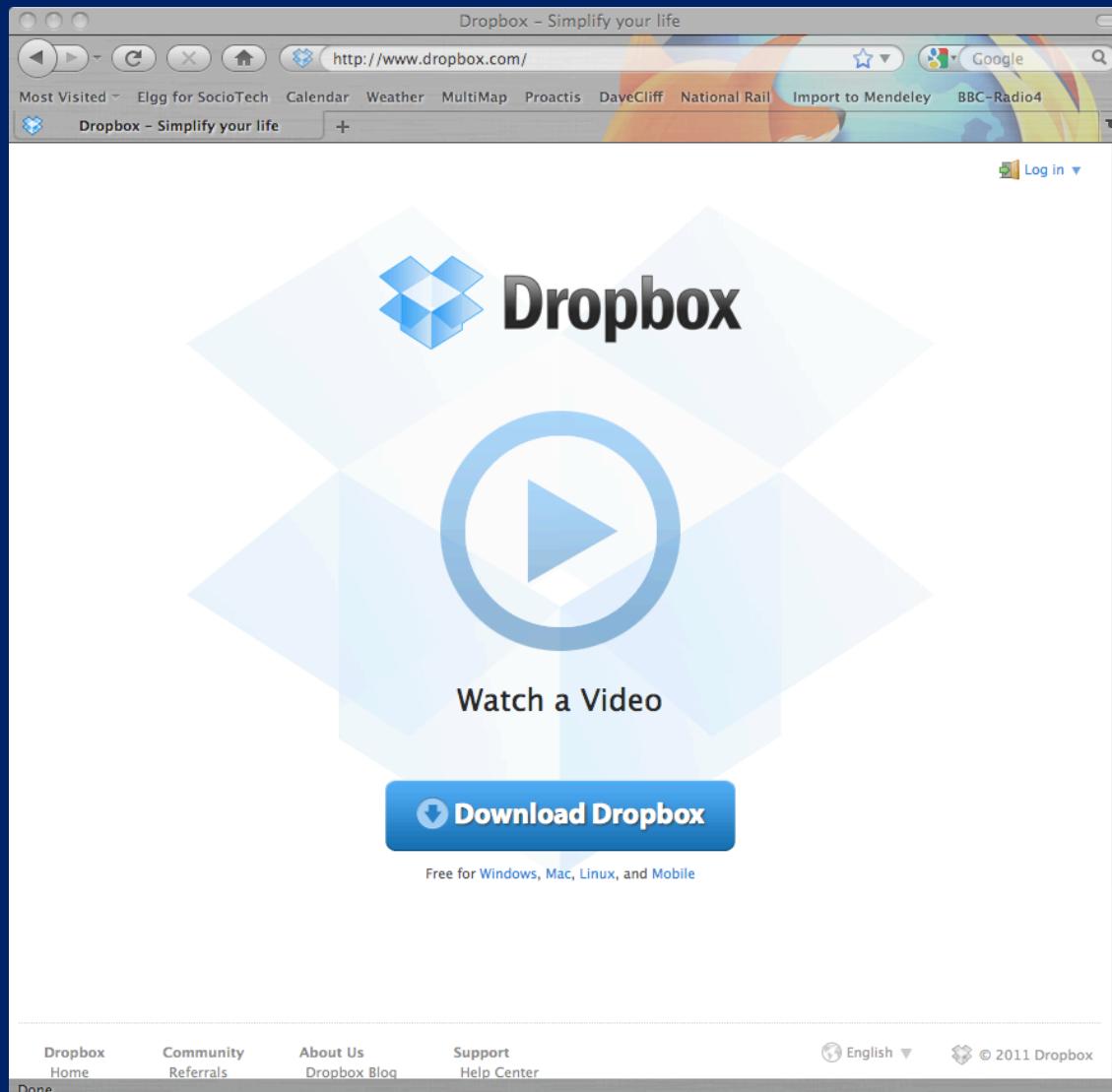
mes,
ur
ource

Times Developer Network

Why just read the news when you can hack it? Visit developer.nytimes.com to learn more about Times APIs, get API keys and participate in developer forums.

Universit
BRIST

dropbox.com



Sign In

See what's next.

WATCH ANYWHERE. CANCEL AT ANY TIME.

JOIN FREE FOR A MONTH >

Animoto, NYT, Dropbox, & Netflix
all courtesy of AWS...

AWS: Amazon Web Services



The screenshot shows the AWS Free Tier landing page. At the top, there's a dark navigation bar with the AWS logo, a search bar, and links for Contact Sales, Support, English, My Account, and a prominent orange "Create an AWS Account" button. Below the bar, a large banner features a colorful background of floating 3D cubes and the text "AWS Free Tier". A sub-section of the banner describes the service: "The AWS Free Tier enables you to gain free, hands-on experience with the AWS platform, products, and services." Below this, a yellow "Create a Free Account" button is centered. The main content area below the banner is white and contains three sections: "Free Tier Details", "Get Started", and "Free Tier Software". Each section has a small icon and some descriptive text. At the bottom, a purple header reads "AWS Free Tier Details" and includes filters for "FEATURED", "12 MONTHS FREE", "ALWAYS FREE", "TRIALS", "PRODUCT CATEGORIES", and "ALL".

AWS Free Tier

The AWS Free Tier enables you to gain free, hands-on experience with the AWS platform, products, and services.

Create a Free Account

Free Tier Details

Get Started

Free Tier Software

AWS Free Tier Details

★ FEATURED 📅 12 MONTHS FREE 🔒 ALWAYS FREE ⏲ TRIALS ▾ PRODUCT CATEGORIES ✓ ALL

...we will talk more about AWS in later lectures...

Amazon are the biggest, but not the only player...

Google Cloud Platform

[Why Google](#)[Products](#)[Solutions](#)[Pricing](#)[Security](#)[Documentation](#)[Console](#)[Contact sales](#)[Try free](#)

Build What's Next Better software. Faster.

- ✓ Use Google's core infrastructure, data analytics and machine learning.
- ✓ Secure and fully featured for all enterprises.
- ✓ Committed to open source and industry leading price-performance.

[TRY IT FREE](#)[CONTACT SALES](#)

Forrester Research

Google Cloud is named the Insight PaaS Leader by Forrester.

GCP Region Expansion

Run workloads in even more locations around the world. Our newest regions: Finland and Los Angeles.

Response to CPU Vulnerabilities

Information and steps you may take to protect your organization from Spectre and Meltdown.

Build & Run Scalable Web Apps on Google's Infrastructure

Programming

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Programming

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	xiii
1. Introducing Google App Engine	1
The Runtime Environment	2
The Static File Servers	4
The Datastore	4
Entities and Properties	5
Queries and Indexes	6
Transactions	6
The Services	8
Google Accounts	9
Task Queues and Cron Jobs	9
Developer Tools	10
The Administration Console	11
Things App Engine Doesn't Do...Yet	12
Getting Started	13
2. Creating an Application	15
Setting Up the SDK	15
Installing the Python SDK	16
Installing the Java SDK	20
Developing the Application	24
The User Preferences Pattern	24
Developing a Python App	25
Developing a Java App	39
The Development Console	54
Registering the Application	55
The Application ID and Title	57
Setting Up a Domain Name	58
Google Apps and Authentication	59
Uploading the Application	60

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	xi
1. Introducing Google App Engine	1
The Runtime Environment	1
The Frontend	4
Deployment Properties	4
Queries and Indexes	5
Transactions	6
The Services	8
Scaling	9
Task Queues and Cron Jobs	9
Datastore API	10
The Administration Console	11
Debugging and Profiling	12
Getting Started	13
2. Creating an Application	15
Setting Up Your Environment	15
Installing the Python SDK	16
Configuring the Application	16
The User Preference Router	21
Developing a Python App	22
Developing a Java App	29
Deploying to the App Engine	34
Registering the Application	35
The Application Configuration File	36
Setting Up a Domain Name	38
Google AppEngine Authentication	39
Upgrading the Application	40

Introducing the Administration Console	61
3. Handling Web Requests	63
The App Engine Architecture	64
Configuring the Frontend	66
Configuring a Python App	66
Configuring a Java App	68
Domain Names	69
App IDs and Versions	70
Request Handlers	72
Static Files and Resource Files	75
Secure Connections	81
Authorization with Google Accounts	83
How the App Is Run	85
The Python Runtime Environment	86
The Java Runtime Environment	87
The Sandbox	88
App Caching	89
Logging	93
Quotas and Limits	96
Request Limits	96
CPU Limits	97
Service Limits	98
Deployment Limits	98
Billable Quotas	100
Resource Usage Headers	101
4. Datastore Entities	103
Entities, Keys, and Properties	104
Introducing the Python Datastore API	105
Introducing the Java Datastore API	108
Property Values	110
Strings, Text, and Blobs	112
Unset Versus the Null Value	112
Multivalued Properties	113
Keys and Key Objects	114
Using Entities	116
Getting Entities Using Keys	116
Inspecting Entity Objects	117
Saving Entities	118
Deleting Entities	119

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	xii
1. Introducing Google App Engine	1
The Runtime Environment	1
The Frontend	4
Deployment Properties	4
Queries and Indexes	5
Transactions	6
The Services	8
Task Queues	9
Datastore API	9
The Administration Console	11
Debugging and Profiling	12
Getting Started	13
2. Creating an Application	15
Setting Up Your Environment	15
Installing the Python SDK	16
Configuring the Application	20
The User Preference Router	24
Developing a Python App	25
Developing a Java App	29
The Datastore API	34
Registering the Application	35
The Application Configuration File	36
Setting Up a Domain Name	38
Google AppEngine Authentication	39
Upgrading the Application	40

Introducing the Administration Console	61
3. Handling Web Requests	63
The App Engine Architecture	64
Configuring the Frontend	66
Configuring a Python App	66
Configuring a Java App	68
Domain Names	69
App IDs and Versions	70
Request Handlers	72
Static Files and Resource Files	75
Secure Connections	81
Authorization with Google Accounts	83
How the App Is Run	85
The Python Runtime Environment	86
The Java Runtime Environment	87
The Sandbox	88
App Caching	89
Logging	93
Quotas and Limits	96
Request Limits	96
CPU Limits	97
Service Limits	98
Deployment Limits	98
Billable Quotas	100
Resource Usage Headers	101
4. Datastore Entities	103
Entities, Keys, and Properties	104
Introducing the Python Datastore API	105
Introducing the Java Datastore API	108
Property Values	110
Strings, Text, and Blobs	112
Unset Versus the Null Value	112
Multivalued Properties	113
Keys and Key Objects	114
Using Entities	116
Getting Entities Using Keys	116
Inspecting Entity Objects	117
Saving Entities	118
Deleting Entities	119

Programming

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	... iii
1. Introducing Google App Engine	1
The Runtime Environment	4
The Datastore	4
Datastore Properties	5
Queries and Indexes	6
Transactions	6
The Services	8
Queues and Tasks	9
Task Queues and Cron Jobs	9
Distributed Cache	10
The Administration Console	11
Using Google's Tools	12
Getting Started	13
2. Creating an Application	15
Setting Up Your Environment	15
Installing the Python SDK	16
Installing the Java SDK	20
Configuring the Application	24
The User Preference Store	25
Developing a Java App	25
Developing a Python App	26
Deploying the Application	27
Registering the Application	28
The Application Configuration File	29
Setting Up a Domain Name	29
Google AppEngine Authentication	30
Upgrading the Application	30

4

Introducing the Administration Console	61
3. Handling Web Requests	63
The Application Console	64
Configuring the Frontend	66
Configuring a Python App	66
Configuring a Java App	67
Tomcat Namespaces	69
Artifactory Versions	70
Request Headers and Resources	72
Static Content and Resource Files	73
Secure Connections	74
Authenticating with Google Accounts	74
How the App Is Run	75
The Python Runtime Environment	76
The Java Runtime Environment	77
The Admin Environment	78
App Caching	79
Logs	79
Quotas and Limits	80
Request Rate Limits	80
CPU Limits	81
Services	81
Deployment Limits	82
Billing	83
Resource Usage Readers	83
4. Database Entities	101
Entities, Keys, and Properties	101
Introducing the Python Datastore API	102
Introducing the Java Datastore API	108
Properties	110
String, Text, and Blob	112
List and Map Values	112
Multivalued Properties	113
Key and Value	114
Using Entity	116
Creating Entities Using Keys	116
Inspecting Entity Objects	117
Setting Entity Properties	118
Deleting Entities	119

40 | Table of Contents

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	xii
1. Introducing Google App Engine	1
The Runtime Environment	1
The Python API	4
Datastore Properties	4
Queries and Indexes	5
Transactions	6
The Services	8
Deployment	9
Task Queues and Cron Jobs	9
Background Tasks	10
The Administration Console	11
Debugging Static Content	12
Getting Started	13
2. Creating an Application	15
Setting Up Your Environment	15
Introducing the Python SDK	16
Developing an Application	20
The User Preference Filter	24
Developing a Java App	25
The Java Persistence API	26
Deploying the Application	28
The Administration Console	29
Registering the Application	30
The Application Configuration File	30
Setting Up a Domain Name	38
Google AppEngine Authentication	39
Upgrading the Application	40

5. Datastore Queries	121
Queries and Kinds	122
Query Results and Keys	122
GQL	123
The Python Query API	126
The Query Class	127
GQL in Python	128
Retrieving Results	129
Keys-Only Queries	131
The Java Query API	132
Keys-Only Queries in Java	133
Introducing Indexes	134
Automatic Indexes and Simple Queries	136
All Entities of a Kind	137
One Equality Filter	137
Greater-Than and Less-Than Filters	138
One Sort Order	139
Queries on Keys	141
Kindless Queries	142
Custom Indexes and Complex Queries	143
Multiple Sort Orders	143
Filters on Multiple Properties	144
Multiple Equality Filters	147
Not-Equal and IN Filters	150
Unset and Nonindexed Properties	150
Sort Orders and Value Types	152
Queries and Multivalued Properties	153
A Simple Example	153
MVPs in Python	154
MVPs and Equality Filters	155
MVPs and Inequality Filters	156
MVPs and Sort Orders	157
Exploding Indexes	159
Configuring Indexes	159
Index Configuration for Python	160
Index Configuration for Java	161
6. Datastore Transactions	163
Entities and Entity Groups	165
Keys, Paths, and Ancestors	166
Ancestor Queries	167
What Can Happen in a Transaction	168
Transactional Reads	169

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	... iii
1. Introducing Google App Engine	1
The Runtime Environment	1
The Python API	4
The Java API	4
Datastore Properties	5
Queries and Indexes	5
Transactions	6
The Services	8
Scaling	9
Task Queues and Cron Jobs	9
Diving In	10
The Administration Console	11
Using Google Books Do... Yet	12
Getting Started	13
2. Creating an Application	15
Setting Up Your Environment	15
Installing the Python SDK	15
Installing the Java SDK	16
Configuring the Application	16
The User Preference Router	17
Developing a Python App	24
Developing a Java App	25
The Configuration File	26
Deploying the Application	27
Testing the Application and Tracing	28
Setting Up a Domain Name	29
Google AppEngine Authentication	30
Upgrading the Application	30

5. Datastore Queries	121
Queries and Kinds	122
Query Results and Keys	122
GQL	123
The Python Query API	126
The Query Class	127
GQL in Python	128
Retrieving Results	129
Keys-Only Queries	131
The Java Query API	132
Keys-Only Queries in Java	133
Introducing Indexes	134
Automatic Indexes and Simple Queries	136
All Entities of a Kind	137
One Equality Filter	137
Greater-Than and Less-Than Filters	138
One Sort Order	139
Queries on Keys	141
Kindless Queries	142
Custom Indexes and Complex Queries	143
Multiple Sort Orders	143
Filters on Multiple Properties	144
Multiple Equality Filters	147
Not-Equal and IN Filters	150
Unset and Nonindexed Properties	150
Sort Orders and Value Types	152
Queries and Multivalued Properties	153
A Simple Example	153
MVPs in Python	154
MVPs and Equality Filters	155
MVPs and Inequality Filters	156
MVPs and Sort Orders	157
Exploding Indexes	159
Configuring Indexes	159
Index Configuration for Python	160
Index Configuration for Java	161
6. Datastore Transactions	163
Entities and Entity Groups	165
Keys, Paths, and Ancestors	166
Ancestor Queries	167
What Can Happen in a Transaction	168
Transactional Reads	169

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	... xii
1. Introducing Google App Engine	1
The Runtime Environment	1
The Services	4
The Features	4
Deployment Properties	5
Queries and Indexes	6
Transactions	6
The Services	8
Security	9
Task Queues and Cron Jobs	9
The Admin Console	10
The Administration Console	11
Using Google App Engine	12
Getting Started	13
2. Creating an Application	15
Setting Up Your Environment	15
Installing the Python SDK	16
Configuring the Python SDK	20
Developing an Application	24
The User Preference Router	25
Developing a Java App	25
Developing a Python App	26
Developing a Go App	27
Developing a .NET App	28
Developing a Node.js App	29
Developing a Ruby App	30
Developing a Scala App	31
Developing a Clojure App	32
Developing a C++ App	33
Developing a PHP App	34
Developing a Perl App	35
Developing a Visual Basic App	36
Developing a VBScript App	37
Developing a JSP App	38
Developing a JBoss Seam App	39
Developing a Spring MVC App	40
Developing a Play Framework App	41
Developing a Grails App	42
Developing a Django App	43
Developing a Ruby on Rails App	44
Developing a Node.js App	45
Developing a Clojure App	46
Developing a C++ App	47
Developing a PHP App	48
Developing a Perl App	49
Developing a JSP App	50
Developing a JBoss Seam App	51
Developing a Spring MVC App	52
Developing a Play Framework App	53
Developing a Grails App	54
Developing a Ruby on Rails App	55
Developing a Node.js App	56
Developing a Clojure App	57
Developing a C++ App	58
Developing a PHP App	59
Developing a Perl App	60

46

Introducing the Administration Console	61
3. Handling Web Requests	63
The Application Console	64
Configuring by Firewall	66
Configuring a Python App	66
Configuring a Java App	67
Configuring a Go App	68
Configuring a .NET App	69
Configuring a Node.js App	70
Configuring a Clojure App	71
Configuring a C++ App	72
Configuring a PHP App	73
Configuring a Perl App	74
Configuring a JSP App	75
Configuring a JBoss Seam App	76
Configuring a Spring MVC App	77
Configuring a Play Framework App	78
Configuring a Grails App	79
Configuring a Ruby on Rails App	80
Configuring a Node.js App	81
Configuring a Clojure App	82
Configuring a C++ App	83
Configuring a PHP App	84
Configuring a Perl App	85
Configuring a JSP App	86
Configuring a JBoss Seam App	87
Configuring a Spring MVC App	88
Configuring a Play Framework App	89
Configuring a Grails App	90
Configuring a Ruby on Rails App	91
Configuring a Node.js App	92
Configuring a Clojure App	93
Configuring a C++ App	94
Configuring a PHP App	95
Configuring a Perl App	96
Configuring a JSP App	97
Configuring a JBoss Seam App	98
Configuring a Spring MVC App	99
Configuring a Play Framework App	100
Configuring a Grails App	101
4. Database Entities	103
Entities, Keys, and Properties	104
Introducing the Python API	105
Introducing the Java Datastore API	106
Introducing the Python Datastore API	107
Properties	108
String Properties	109
String, Text, and Blob Properties	110
List and Set Properties	111
Multi-valued Properties	112
Local and Global Properties	113
Using Properties	114
Using Filters	115
Comparing Using Keys	116
Comparing Entity Objects	117
Comparing Datastore	118
Comparing Deleting Entities	119

47

Table of Contents

5. Database Queries	121
Entity Selection	122
Query Entities and Keys	122
QL	123
The Python Query API	126
The Query Class	127
QL2	128
Retrieving Results	129
Entity Selection	131
The Java Query API	132
Comparing Selectors in Java	133
Introducing Indexes	134
Annotating Entities and Simple Queries	135
All Entities of a Kind	137
One Entity	137
Greater Than and Less Than Filters	138
��	138
Querries on Keys	140
Equality	142
Custom Indexes and Complex Queries	143
Using Indexes	144
Filtering Multiple Properties	144
Using Equality	145
Not Equal and IN Filters	146
Less Than or Equal and Greater Than	148
Sun Order and Value Properties	150
Querries and Multi-valued Properties	152
Using Indexes	153
GQL in Python	154
GQL and Equality Filters	155
GQL and Equality Filters	156
GQL and Equality Filters	157
Exploring Indexes	159
Comparing GQL and QL	159
Index Configuration for Python	160
More on Indexes for the Java	163
Entity Selection	163
6. Database Transactions	163
Entities and Entity Groups	164
Transactions	165
Entity Groups	166
ancestor Queries	167
What Can Happen in a Transaction	168
Transactions in Java	169

48

Table of Contents

Programming

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	xi
1. Introducing Google App Engine	1
The Runtime Environment	1
The Java Environment	4
The Python Environment	4
Deployment Properties	5
Queries and Indexes	6
Transactions	6
The Services	8
Metrics and Monitoring	9
Task Queues and Cron Jobs	9
The Logging API	9
The Administration Console	11
Using Google's Pre-built Tools	12
Getting Started	13
2. Creating an Application	15
Setting Up Your Application	15
Introducing the Python SDK	16
Introducing the Java SDK	16
Configuring the Application	20
The User Preference Editor	24
Developing a Java App	25
Developing a Python App	25
Deploying the Application	26
Running the Application	26
Testing the Application	27
Deployment Triggers	27
Setting up a Domain Name	28
Google App Engine Authentication	29
Upgrading the Application	30

46

Introducing the Administration Console	61
3. Handling Web Requests	63
The Python Request Handler	64
Configuring the Python Handler	66
Configuring a Python App	66
Customizing the Python Handler	69
Dynamically Naming the Python Handler	70
Request Headers and Response Headers	72
Setting Headers and Response Files	73
Secure Connections	74
Authenticating with Google Accounts	75
How the App Is Run	75
The Python Runtime Environment	76
The Java Runtime Environment	76
The Admin Console	76
App Coding	78
Logging	79
Quotas and Limits	80
Request and Response	80
CPU Limits	80
Service Limits	80
Deployment Limits	80
Task Queue Limits	80
Resource Usage	80
Resource Usage Headers	81
4. Database Entities	103
Entities, Keys, and Properties	104
Introducing the Python Datastore API	105
Introducing the Java Datastore API	108
Properties	110
String, Text, and Blob	112
List and String Value	112
Multi-valued Properties	113
Long and Double	114
Using Enums	114
Using Ids	115
Inspecting Entity Objects	117
Serializing Entities	118
Deleting Entities	119

48 | Table of Contents

5. Database Queries	121
Introducing Queries	122
Query Results and Keys	122
QL	122
The Python Query API	126
The Query Class	127
QL	128
Retrieving Results	129
Keyless Queries	131
The Java Query API	132
Introducing Indexes	132
Annotating Entities and Simple Queries	132
All Entities of a Kind	137
Order	137
Greater Than and Less Than Filters	138
Equality	139
Ranges on Keys	140
Equality	142
Custom Indexes and Complex Queries	143
Indexing	143
Timestamp Equality Properties	144
Timestamp Range	145
Not Equal and IN Filters	146
Searched and Unsearched Properties	147
Searched Order and Value	148
Queried and Influenced Properties	149
Mutual Exclusivity	150
GQL in Python	151
GQL Equality Filters	151
GQL AND Equality Filters	152
Exploring Indexes	157
Configuring Indexes	159
Index Configuration for Python	160
Index Configuration for Java	163
6. Database Transactions	163
Entities and Entity Groups	165
Introducing Transactions	166
ancestor Transactions	167
What Happens in a Transaction	168
Transactions Inside a Transaction	169

50 | Table of Contents

Transactions in Python	169
Transactions in Java	172
How Transactions Are Used	172
How Entities Are Read	178
Batch Updates	179
How Entities Are Updated	180
7. Data Modeling with Python	181
Models and Properties	184
Properties	185
Property Value Types	186
Properties	187
Non-inherited Properties	188
Inheritance	189
Modeling Inheritance	190
Modeling Inheritance via Aggregation	191
Modeling Relationships	191
One-to-Many Relationships	192
One-to-One Relationships	193
Many-to-Many Relationships	194
Model Inheritance	196
Queries and Model Inheritance	199
Creating and Using Property Classes	200
Validating Property Values	201
Handling Validation Errors	202
Customizing Default Values	204
Modeling Values	205
For More Information	223
8. The Java Persistence API	207
Setting Up JPA	208
Entity Classes	209
Entity Properties	212
Embeddable Classes	213
Saving, Fetching, and Deleting Objects	214
Transactions and JPA	216
Queries and JPQL	219
Annotations	220
For More Information	223
9. The Memory Cache	227
The Python Memcache API	228
Getting and Setting Values in Python	229
Setting and Getting Multiple Values	230

52 | Table of Contents

Programming

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	xi
1. Introducing Google App Engine	1
The Runtime Environment	1
The Java Environment	4
The Python Environment	4
Datastore Properties	5
Queries and Indexes	6
Transactions	6
The Services	8
Metrics and Metrics	9
Task Queues and Cron Jobs	9
Diving In	9
The Administration Console	11
Using Google's Default Do... Yet	12
Getting Started	13
2. Creating an Application	15
Setting Up Your Application	15
Introducing the Python Side	16
Installing the Java Side	20
Configuring the Application	24
The User Preference Router	25
Developing a Java App	25
Developing a Python App	26
Deploying the Application	26
Running the Application	27
Testing the Application and Trac...	27
Setting Up a Domain Name	28
Google AppEngine Authentication	29
Upgrading the Application	30

46

Introducing the Administration Console

3. Handling Web Requests	63
The Python Request Handler	64
Configuring the Python App	66
Controlling the Python App	66
Domain Names	69
API and Java Versions	70
Request Headers	72
Status Codes and Response Files	73
Secure Connections	74
Authenticating with Google Accounts	75
How the App Is Run	76
The Python Runtime Environment	76
The Java Runtime Environment	78
The Admin Console	78
App Caching	79
Logging	79
Quotas and Limits	80
Request Throttling	80
CPU Limits	81
Service Limits	81
Task Queues and Cron Jobs	82
The Administration Console	82
Using Google's Default Do... Yet	83
Getting Started	83

4. Database Entities	103
Entities, Keys, and Properties	104
Introducing the Python Datastore API	105
Introducing the Java Datastore API	108
Properties	109
String, Text, and Blob	112
List and Map Values	112
Multivalued Properties	113
Keyed Properties	114
Using Entities	116
Creating Entities Using Keys	117
Inspecting Entity Objects	118
Updating Entities	119
Deleting Entities	119

47

Table of Contents

5. Database Queries

5.1. Querying Entities and Keys	121
Query Bindings and Keys	122
QL	122
The Python Query API	126
The Query Class	127
QL2	128
Retrieving Entities	129
Entity Properties	131
The Java Query API	132
Accessing Entities in Java	133
Introducing Indexes	135
Annotating Entities and Simple Queries	136
All Entities of a Kind	137
One Entity at a Time	137
Greater Than and Less Than Filters	138
Querying	139
Querries on Keys	140
Indexing	142
Custom Indexes and Complex Queries	143
Indexing	144
Timestamp-Order Properties	144
Introducing Indexes	145
Non-Equal and IN Filters	146
Searched Order and IN Properties	147
Searched Order and Multi-valued Properties	148
Multiple Indexes	149
QL3	150
QL4	151
MQL	152
MQL and Equality Filters	153
MQL and Equality Filters	154
MQL and Equality Filters	155
Exploring Indexes	156
Configuring Indexes	157
Index Configuration for Python	158
Index Configuration for Java	159
Index Configuration	160
6. Database Transactions	163
Entities and Transaction Groups	165
Transactions and Transaction Groups	166
ancestor Queries	167
What Can Happen in a Transaction	168
Transactions in Python	169
Transactions in Java	172
How Transactions Are Used	173
Basic Examples	178
How Transactions Are Updated	180

48

Table of Contents

7. Data Modeling with Python

7.1. Models and Properties	181
Properties	184
Boolean Properties	185
Frequency Value Types	186
Timestamp Properties	187
Nonmultivalued Properties	188
Associations	189
List Properties	190
Modeling Associations	191
Modeling Relationships	192
One-to-Many Relationships	193
One-to-One Relationships	194
Many-to-Many Relationships	195
Model Inheritance	196
Queries and Property Classes	199
Comparing Open Property Classes	200
Validating Property Values	201
Managing Object State	202
Customizing Default Values	204
Associations	205
Saving, Fetching, and Deleting Objects	214
Associations	215
Saving, Fetching, and Deleting Objects	216
Transactions and the Python API	218
Querries and SQL	219
Associations	220
For More Information	223
8. The Java Persistence API	207
Saving (ip PA	208
Entity Properties	209
Embedding Entities	210
Saving, Fetching, and Deleting Objects	214
Transactions and the Java API	215
Querries and SQL	217
Associations	218
For More Information	223
9. The Memory Cache	227
The Python Memcache API	228
Setting and Getting Multiple Values	229
Setting and Getting Single Values	230

x | Table of Contents

10. Sending and Receiving Mail and Instant Messages

10.1. Sending Mail	231
Email and the Mail Service	231
Sending Email	234
Receiving Addresses	235
Receiving	236
Attachments	237
Sending Email in Python	238
Sending Email in Java	261
Receiving Email in Python	263
Receiving Email in Java	264
Processing Results with Callbacks	266
Processing Results	267
Sending and Receiving	269
Sending a Chat Message	270
Checking for Incoming User Status	271
Receiving XMPP Messages	272
Receiving XMPP Messages in Python	273
Receiving XMPP Messages in Java	273
10.2. Bulk Data Operations and Remote Access	277
Setting Up the Remote API for Python	278
Setting Up the Remote API for Java	279
Using the Bulk Loader Tool	280
Backing up and Restoring	281
Backing up and Restoring	282
Downloading Data	286
Copying Data with the Bulk Loader	289
Using the Remote Shell Tool	290

Table of Contents | x

Programming

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	xii
1. Introducing Google App Engine	1
The Runtime Environment	1
The Java Environment	4
The Python Environment	4
Datastore Properties	5
Queries and Indexes	6
Transactions	6
The Services	8
Scaling	9
Task Queues and Cron Jobs	9
Django Models	9
The Administration Console	11
Using Google's Toolkit Do... Yet	12
Getting Started	13
2. Creating an Application	15
Setting Up Your Application	15
Introducing the Python SDK	16
Installing the Java SDK	20
Configuring the Application	24
The User Preference Router	25
Developing a Java App	25
Developing a Python App	26
Deploying the Application	27
Running the Application and Test	27
Setting Up a Domain Name	28
Google App Engine Authentication	29
Upgrading the Application	30

46

Introducing the Administration Console

Introducing the Administration Console	61
The Java Admin Console	63
The Python Admin Console	64
Configuring a Python App	66
Configuring a Java App	66
Domain Names	69
Artificial URLs	70
Request Handler	72
Static Handler	73
Resource Handler	73
Secure Connection	81
Authenticating with Google Accounts	81
How the App Is Run	83
The Python Runtime Environment	84
The Java Runtime Environment	84
The Admin Environment	84
App Coding	86
Logging	89
Quotas and Limits	90
Request Handler	90
CPU Limits	91
Memory Limits	91
Deployment Limits	92
Logs	93
Resource Usage Reader	93
Resource Usage Reader	93

3. Handling Web Requests

Introducing the Java Environment	103
Entities, Keys, and Properties	104
Introducing the Java Persistence API	107
Introducing the Java Datastore API	108
Properties in Java	109
Strings, Text, and Blob	112
List and Map Values	112
Multivalued Properties	113
Entity and Object	114
Using Entities	116
Creating Entities Using Keys	117
Inspecting Entity Objects	118
Deleting Entities	119

4. Database Entities

Entities and Entity Groups	143
Introducing Entities	144
Annotations	144
Annotations for Java	145
Annotations for Python	145
MPTP and Equality Filters	146
MPTP and Equality Filters for Python	147
Exploiting Indices	149
Comparing Entities	150
Index Configuration for Python	160
Migrating Entities from Java	163
Using Entities	163
Using Keys	164
Inspecting Entity Objects	165
Deleting Entities	169

5. Database Queries

6. Database Transactions

7. Data Modeling with Python

Models and Properties	183
Properties	184
Frequency Value Types	186
Entity Properties	186
Nonstandard Properties	188
List Properties	190
Modeling Relationships	192
One-to-One Relationships	193
One-to-Many Relationships	193
Many-to-Many Relationships	196
Model Inheritance	198
Queries and Entity Classes	199
Customizing Property Values	201
Handling Validation	202
Customizing Default Values	204
Saving and Deleting Objects	205
Saving, Fetching, and Deleting Objects	214
Entity Properties	212
Entity Methods	213
Saving, Fetching, and Deleting Objects	214
Transactions	215
Querries and SQL	217
Indexes	220
For More Information	223

8. The Java Persistence API

Saving (p PA)	208
Entity Persistence	208
Entry Properties	210
Entity Methods	211
Saving, Fetching, and Deleting Objects	214
Transactions	215
Querries and SQL	217
Indexes	220
For More Information	223
9. The Memory Cache	227
The Python Memcache API	228
Getting and Setting Values in Python	229
Setting and Getting Multiple Values	230

10. Fetching URLs and Web Resources

Memcache Namespaces	231
Cache Expression	231
Deleting	232
Memcache Counters	233
Cache Javadoc	233
The Java Jcache API	234
Fetching URLs in Python	240
Fetching URLs in Java	242
Asynchronous Requests in Python	244
Asynchronous Requests in Java	246
Processing Results with Callbacks	247

11. Sending and Receiving Mail and Instant Messages

Mailgun Mailgun Service	251
Sending Email in Python	251
Sending Email in Python	251
Sending Email in Java	253
Receiving Email	256
Anymail	257
Sending Email in Python	258
Sending Email in Java	261
Receiving Email in Python	263
Receiving Email in Java	264
Forwarding Email	266
Sending XMPP Messages	267
Receiving XMPP Messages	269
Sending and Receiving Instant Messages	270
Receiving Instant Messages	271
Receiving Chat Messages	272
Receiving and Publishing User Status	273
Receiving XMPP Messages in Python	273
Receiving XMPP Messages in Java	273

12. Bulk Data Operations and Remote Access

13. Task Queue and Scheduled Tasks

14. The Django Web Application Framework

15. Deploying and Managing Applications

Index

x | Table of Contents

Programming

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	... iii
1. Introducing Google App Engine	1
The Runtime Environment	1
The Datastore	4
Datastore Properties	4
Queries and Indexes	5
Transactions	6
The Services	8
Security	9
Task Queues and Cron Jobs	9
Logs and Metrics	9
The Administration Console	11
Using Google Books Do... Yet	12
Getting Started	13

4

viii

Introducing the Administration Console	61
3. Handling Web Requests	63
The Admin Console	64
Configuring to Proceed	66
Configuring a Python App	66
Configuring Java	68
Domain Names	69
Artifacts and Versions	70
Request Headers	72
Setting Headers	73
Secure Connections	81
Authenticating with Google Accounts	81
How the App Is Run	83
The Python Runtime Environment	84
The Java Runtime Environment	84
The Admin Console	84
App Configuration	86
Loggings	87
Quotas and Limits	90
Reporting Metrics	90
CPU Limits	91
Memory Limits	91
Deployment Limits	92
Task Queues	93
Resource Usage	93
Request Usage Reader	93
5. Database Queries	121
Entity, Keys, and Properties	122
Query Bindings and Keys	122
QL	122
The Python Query API	126
The Query Class	126
QL	127
Retrieving Entities	128
Entity Properties	129
The Java Query API	131
Using the Java Persistence API	131
Introducing Indexes	132
Annotating Entities and Simple Queries	132
All Entities of a Kind	137
One Entity	137
Greater Than and Less Than Filters	138
Greater Than or Equal To	138
Querries on Keys	141
String Filters	142
Custom Indexes and Complex Queries	143
Using the Java Persistence API	143
Filtering Multiple Properties	144
Introducing Entity Groups	145
Not Equal and IN Filters	146
Joining Entities with Properties	147
Sub Orders and Value Properties	148
Querries and Multi-threaded Properties	149
Locking Entities	150
QL in Python	151
MQL and Equality Filters	151
MQL and Equality Filters	156
MQL and Equality Filters	157
Exploring Indexes	159
Configuring Indexes	160
Index Configuration for Python	160
Index Configuration for Java	163
6. Database Entities	163
Entities and Entity Groups	165
Introducing Entities	166
Ancestor Queries	167
What Can Happen in a Transaction	168
Deleting Entities	169

viii

Table of Contents

7. Data Modeling with Python	181
Models and Properties	184
Properties	184
Frequency Value Types	186
Properties	186
Nonstandard Properties	188
Associations	190
Modeling One-to-One Aggregation	192
Modeling Relationships	193
One-to-Many Relationships	193
Many-to-Many Relationships	194
Model Inheritance	196
Queries and Query Classes	199
Customizing Property Values	201
Handling Large Data	203
Customizing Default Values	204
Deleting Entities	205
8. The Java Persistence API	207
Setting Up JPA	208
Entity Persistence	209
Entity Properties	212
Associations	212
Saving, Fetching, and Deleting Objects	214
Transactions and JPA	214
Querries and JPQL	217
Relationships	218
For More Information	223
9. The Memory Cache	227
The Python Memcache API	228
Setting and Getting Single Values	229
Setting and Getting Multiple Values	230

ix

Table of Contents

>40% of this book
is all about
“databases”

Transactions in Python	169
Transactions in Java	172
How Transactions Are Used	175
How Entities Are Read	178
Batch Updates	179
How Entities Are Updated	180
7. Data Modeling with Python	181
Models and Properties	184
Properties	184
Frequency Value Types	186
Properties	186
Nonstandard Properties	188
Associations	190
Modeling One-to-One Aggregation	192
Modeling Relationships	193
One-to-Many Relationships	193
Many-to-Many Relationships	194
Model Inheritance	196
Queries and Query Classes	199
Customizing Property Values	201
Handling Large Data	203
Customizing Default Values	204
Deleting Entities	205
8. The Java Persistence API	207
Saving Up JPA	208
Entity Persistence	209
Entity Properties	212
Associations	212
Saving, Fetching, and Deleting Objects	214
Transactions and JPA	214
Querries and JPQL	217
Relationships	218
For More Information	223
9. The Memory Cache	227
The Python Memcache API	228
Setting and Getting Single Values	229
Setting and Getting Multiple Values	230

x

Table of Contents

Memcache Namespace	231
Cache Expression	231
Deleting Items	232
Memcache Counter	233
Cache Counter	233
The Java Jcache API	234
10. Fetching URLs and Web Resources	239
Fetching URLs in Python	240
Fetching URLs in Java	242
Asynchronous Requests in Python	244
Asynchronous Requests in Java	246
Processing Results with Callbacks	247
11. Sending and Receiving Mail and Instant Messages	251
Handling the Mail Service	253
Sending Email Instances	254
Sending Text Instances	255
Receiving Instances	256
Archetypes	257
Archetypes in Python	258
Sending Email in Java	261
Receiving Email in Python	263
Receiving Email in Java	264
Decomposing Mails	266
Decomposing Mails in Java	269
Sending XMPP Messages	267
Sending XMPP Instances	268
Archetypes	269
Sending XMPP Instances	270
Archetypes in Java	271
Receiving XMPP Instances	272
Receiving XMPP Instances in Python	273
Receiving XMPP Instances in Java	273
12. Bulk Data Operations and Remote Access	277
Sending Up the Remote API for Python	278
Setting Up the Remote API for Java	279
Using the Bulk Loader Tool	280
Backing Up and Restoring Data	281
Backing Up and Restoring Data	282
Downloading Data	286
Copying Data with the Bulk Loader	287
Using the Remote Shell Tool	288

xi

Table of Contents

Using the Remote API from a Script	291
13. Task Queues and Scheduled Tasks	293
Task Queues	294
Pushing Tasks and Worker Buckets	295
Task Sharding and Routes	297
Task Queues and Cron Triggers	299
Using Task Queues in Python	301
Using Task Queues in Java	307
Automated Task Processing	308
Scheduled Tasks	308
14. The Django Web Application Framework	311
Introducing Django	314
Creating a Project	315
Creating a Model	316
The Request Handler Script	317
The Django App Model Object	320
Creating Model Instances	320
Using App Engine Models With Django	321
Using Django Models With Python	321
Using Django Models With Java	324
15. Deploying and Managing Applications	333
Upgrading Python	334
Upgrading Java	335
User Versions	335
Managing Configuration	337
Managing Indices	337
Backing Up and Restoring Logs	339
Backing Up the Database	342
Backing Up and Restoring Data	342
Managing Developers	343
Quarantine Testing	343
Generating Help	343
Index	347

xi

Table of Contents

Programming

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	xii
1. Introducing Google App Engine	1
The Runtime Environment	1
The Datastore	4
Databases	4
Queries and Indexes	5
Transactions	6
The Services	8
Metrics	9
Logs	9
Task Queues and Cron Jobs	9
The DevConsole	10
The Administration Console	11
Using Google Cloud Endpoints	12
Getting Started	13
2. Creating an Application	15
Setting Up Your Application	15
Introducing the Python SDK	16
Installing the Java SDK	20
Developing an Application	24
The User Preference Reader	24
Developing a Java App	25
Developing a Python App	26
Developing a Go App	27
Developing a Node.js App	28
Developing an App with Tasks	29
Setting up a Domain Name	30
Upgrading the Application	30

46

Introducing the Administration Console

3. Handling Web Requests	63
The Python Request Handler	64
Configuring a Python App	66
Controlling Requests	68
Domain Names	69
API and Libraries	70
Request Headers	72
Server Headers and Resource Files	73
Secure Connections	74
Authenticating with Google Accounts	75
How the App Is Run	76
The Python Runtime Environment	77
The Java Runtime Environment	78
API Caching	79
Logging	80
Quotas and Limits	80
Request Logging	80
CPU Limits	81
Memory Limits	81
Service Limits	81
Deployment Limits	81
Task Queues and Cron Jobs	82
The Administration Console	82
Using Google Cloud Endpoints	83
Getting Started	83

401 | Table of Contents

4. Database Entities	103
Entities, Keys, and Properties	104
Introducing the Python Datastore API	105
Introducing the Java Datastore API	108
Properties	109
Strings, Text, and Blob	112
List and String Values	112
Multivalued Properties	113
Keys and References	114
Using Entities	116
Creating Entities Using Keys	117
Inspecting Entity Objects	118
Deleting Entities	119

401 | Table of Contents

5. Database Queries	121
Query Results and Keys	122
QL	122
The Python Query API	126
The Query Class	126
QL2	127
Retrieving Results	128
Entity Keys	129
The Java Query API	131
Accessing the Datastore in Java	131
Introducing Indexes	132
Anatomy of a Query and Simple Queries	132
All Entities of a Kind	137
One Entity	137
Greater Than and Less Than Filters	138
Greater Than or Equal To	138
Querries on Keys	140
Indexing	141
Custom Indexes and Complex Queries	142
Index Configuration	143
Timestamped Properties	144
Using Indexes	145
Non Equal and IN Filters	146
Less Than or Equal To Properties	146
Scan Order and Value Properties	147
Querries and Multidimensional Properties	148
Using Keys	149
GQL in Python	150
GQL Query Filters	150
GQL and Equality Filters	150
GQL and Inequality Filters	150
GQL and Range Filters	150
Exploring Indexes	150
Configuring Indexes	151
Index Configuration for Python	152
Index Configuration for Java	153

401 | Table of Contents

6. Database Transactions	143
Entities and Entity Groups	165
Transactions and Transactional Entities	166
ancestor Transactions	167
What Can Happen in a Transaction	168
Transactional Fetches	169

401 | Table of Contents

7. Data Modeling with Python	181
Models and Properties	184
Properties	185
Frequency Value Types	186
Properties and References	187
Nonstandard Properties	188
List Properties	189
Modeling Relationships	190
Modeling Relationships	191
One-to-One Relationships	192
One-to-Many Relationships	193
Many-to-Many Relationships	194
Model Inheritance	196
Queries and Entity Classes	199
Comparing Entity Classes	200
Validating Property Values	201
Handling Validation Errors	202
Customizing Default Values	204
Setting Entity Values	205
B. The Java Persistence API	207
Setting Up JPA	208
Entity Classes	209
Entity Properties	212
Embedding Entities	213
Saving, Fetching, and Deleting Objects	214
Transactions and JPA	215
Querries and JPQL	217
Relationships	218
For More Information	223
9. The Memory Cache	227
The Python Memcache API	228
Caching Data in Python	229
Setting and Getting Multiple Values	230

401 | Table of Contents

>40% of this book
is all about
“data X’s”

10. Fetching URLs and Web Resources	239
Fetching URLs in Python	240
Fetching URLs in Java	242
Asynchronous Requests in Python	244
Asynchronous Requests in Java	246
Processing Results with Callbacks	247
11. Sending and Receiving Mail and Instant Messages	251
Email and IMAP Services	253
Sending Email	254
Receiving Email	255
Reading Addresses	256
Addressing	257
Sending Email in Python	258
Sending Email in Java	261
Receiving Email in Python	263
Receiving Email in Java	264
Sending XMPP Messages	267
Receiving XMPP Messages	269
Setting Up XMPP	270
Configuring XMPP	271
Sending XMPP Messages	272
Receiving XMPP Messages in Python	273
Receiving XMPP Messages in Java	273
12. Bulk Data Operations and Remote Access	277
Setting Up the Remote API for Python	278
Setting Up the Remote API for Java	279
Using the Bulk Loader Tool	280
Backing up Data	281
Upgrading Data	282
Downloading Data	286
Configuring the Bulk Loader	289
Using the Remote Shell Tool	290

401 | Table of Contents

401 | Table of Contents

Programming

Google App Engine



O'REILLY® | Google Press

Dan Sanderson

Table of Contents

Preface	xi
1. Introducing Google App Engine	1
The Runtime Environment	4
The Datastore	4
Datastore Properties	5
Queries and Indexes	6
Transactions	6
The Services	8
Scaling	9
Task Queues and Cron Jobs	9
Django Support	9
The Administration Console	11
Using Google's Pre-built Tools	12
Getting Started	13
2. Creating an Application	15
Setting Up Your Application	15
Introducing the Python SDK	16
Installing the Java SDK	20
Configuring the Application	24
The User Preference Router	25
Developing a Java App	25
Developing a Python App	26
Developing a Shared Java/Python Application	26
Developing a Shared Java/Python Application	27
Deploying the Application	35
The Administration Console and Task Queue API	36
Setting up a Domain Name	38
Google App Engine Authentication	39
Upgrading the Application	40

46

Introducing the Administration Console	61
3. Handling Web Requests	63
The Python Request Handler	64
Configuring the Python Request Handler	66
Configuring a Python App	66
Controlling Requests	67
Timeout Names	69
API and Versions	70
Request Headers	72
Setting Headers and Response Files	73
Secure Connections	74
Authenticating with Google Accounts	75
How the App Is Run	75
The Python Runtime Environment	76
The Java Runtime Environment	76
The Admin Console	77
API Configuration	78
Logging	79
Quotas and Limits	80
Request Headers	80
CPU Limits	81
Memory Limits	81
Deployment Limits	82
Task Queues	83
Request Headers	84
Resource Headers	85
Request Headers	86
Request Headers	87
Request Headers	88
Request Headers	89
Request Headers	90
Request Headers	91
Request Headers	92
Request Headers	93
Request Headers	94
Request Headers	95
Request Headers	96
Request Headers	97
Request Headers	98
Request Headers	99
Request Headers	100
Request Headers	101
4. Database Entities	103
Entities, Keys, and Properties	104
Introducing the Python Datastore API	105
Introducing the Java Datastore API	108
Properties	110
String, Text, and Blob	112
List and Map Properties	112
Multi-valued Properties	113
Lazy Properties	114
Using Entities	116
Creating Entities Using Keys	117
Inspecting Entity Objects	118
Setting Properties	119
Deleting Entities	120

47

5. Database Queries	121
Introducing Queries	122
Query Results and Keys	122
QL	122
The Python Query API	126
The Query Class	126
QL Syntax	127
Retrieving Results	128
Entity Results	129
The Java Query API	131
Using the Query API in Java	131
Introducing Indexes	132
Annotating Entities and Simple Queries	132
All Entities of a Kind	132
One Entity at a Time	132
Greater Than and Less Than Filters	133
Greater Than or Equal To	133
Querries on Keys	134
Querying on Indexes	134
Custom Indexes and Complex Queries	135
Using Indexes	135
Filtering Multiple Properties	136
Introducing Entity Groups	136
Using Entity Groups	137
Non-Equal and IN Filters	138
Using Order By Properties	138
Sort Order and Value Properties	139
Querries and Multi-valued Properties	139
Using Keys	140
QL in Python	140
QL and Equality Filters	141
QL and Inequality Filters	141
QL and Range Filters	142
QL and Entity Groups	142
Exploring Indexes	143
Configuring Indexes	143
Index Configuration for Python	143
Index Configuration for Java	143
6. Database Transactions	143
Entities and Groups	145
Introducing Transactions	146
Ancestor Queries	147
What Can Happen in a Transaction	148
Transactions in Python	149
Transactions in Java	150
How Entities Are Read	152
Batch Updates	152
How Entities Are Updated	153

48

>40% of this book
is all about
“dataSTORES”

Transactions in Python	169
Transactions in Java	172
How Entities Are Read	173
Batch Updates	174
How Entities Are Updated	180
7. Data Modeling with Python	181
Models and Properties	184
Properties	184
Frequency Value Types	186
Properties and Persistence	186
Nonstandard Properties	188
List Properties	189
Modeling Relationships	190
Modeling One-to-Many Aggregation	191
Modeling Relationships	192
One-to-Many Relationships	193
One-to-One Relationships	193
Many-to-Many Relationships	194
Model Inheritance	196
Queries and Entity Classes	199
Comparing Entity Classes	200
Validating Property Values	201
Handling Validation Errors	202
Customizing Default Values	204
Setting Entity Values	205
8. The Java Persistence API	207
Setting Up JPA	208
Entity Classes	209
Entity Properties	212
Embedding Entities	213
Saving, Fetching, and Deleting Objects	214
Transactions and JPA	215
Querries and JPQL	217
Relationships	220
For More Information	223
9. The Memcache	227
The Python Memcache API	228
Setting and Getting Values in Python	229
Setting and Getting Multiple Values	230

49

Memcache Namespace	231
Cache Expression	231
Deleting Items	232
Memcache Counters	233
Cache Services	233
The Java Memcache API	234
10. Fetching URLs and Web Resources	239
Fetching URLs in Python	240
Fetching URLs in Java	242
Asynchronous Requests in Python	244
Asynchronous Requests in Java	246
Processing Results with Callbacks	247
11. Sending and Receiving Mail and Instant Messages	251
Email and the Mail Service	251
Sending Email in Python	252
Sending Addresses	255
Receiving Mail	256
Asynchronous	257
Sending Email in Java	261
Receiving Email in Java	264
Sending and Receiving IMs	265
Receiving IMs in Python	266
Sending XMPP Messages	267
XMPP Configuration	268
Sending XMPP Messages in Java	273
12. Bulk Data Operations and Remote Access	277
Setting Up the Remote API for Python	278
Setting Up the Bulk Loader Tool	279
Using the Bulk Loader	280
Backing up Data	281
Upgrading Data	282
Downloading Data	286
Copying Data with the Bulk Loader	289
Using the Remote Shell Tool	290

50

Using the Remote API from a Script	291
13. Task Queues and Scheduled Tasks	293
Task Queues	294
Defining Task and Token Buckets	294
Task Handlers and Routes	295
Task Queues and Cron Triggers	296
Using Task Queues in Python	299
Using Task Queues in Java	303
Scheduled Task Expiring	307
Scheduling a Task	308
14. The Django Web Application Framework	311
Introducing Django	314
Configuring Django	315
The Request Handler Script	316
The Django Application Object	317
Configuring Application Settings	320
Using App Engine Models With Django	322
Using Django's Model Forms	324
Working with the Django Test Client	327
15. Deploying and Managing Applications	333
Upgrading Versions	333
Managing Application Configuration	335
Managing Indexes	337
Managing Indexes Using the Indexing Log	339
Managing the Datastore	342
Managing Datastores	343
Managing Developers	344
Quarantine Testing	345
Getting Help	345
Index	347

51



Data Serving in the Cloud

Raghu Ramakrishnan

Chief Scientist, Audience and Cloud Computing

Brian Cooper

Adam Silberstein

Utkarsh Srivastava

Yahoo! Research

Joint work with the Sherpa team in Cloud Computing

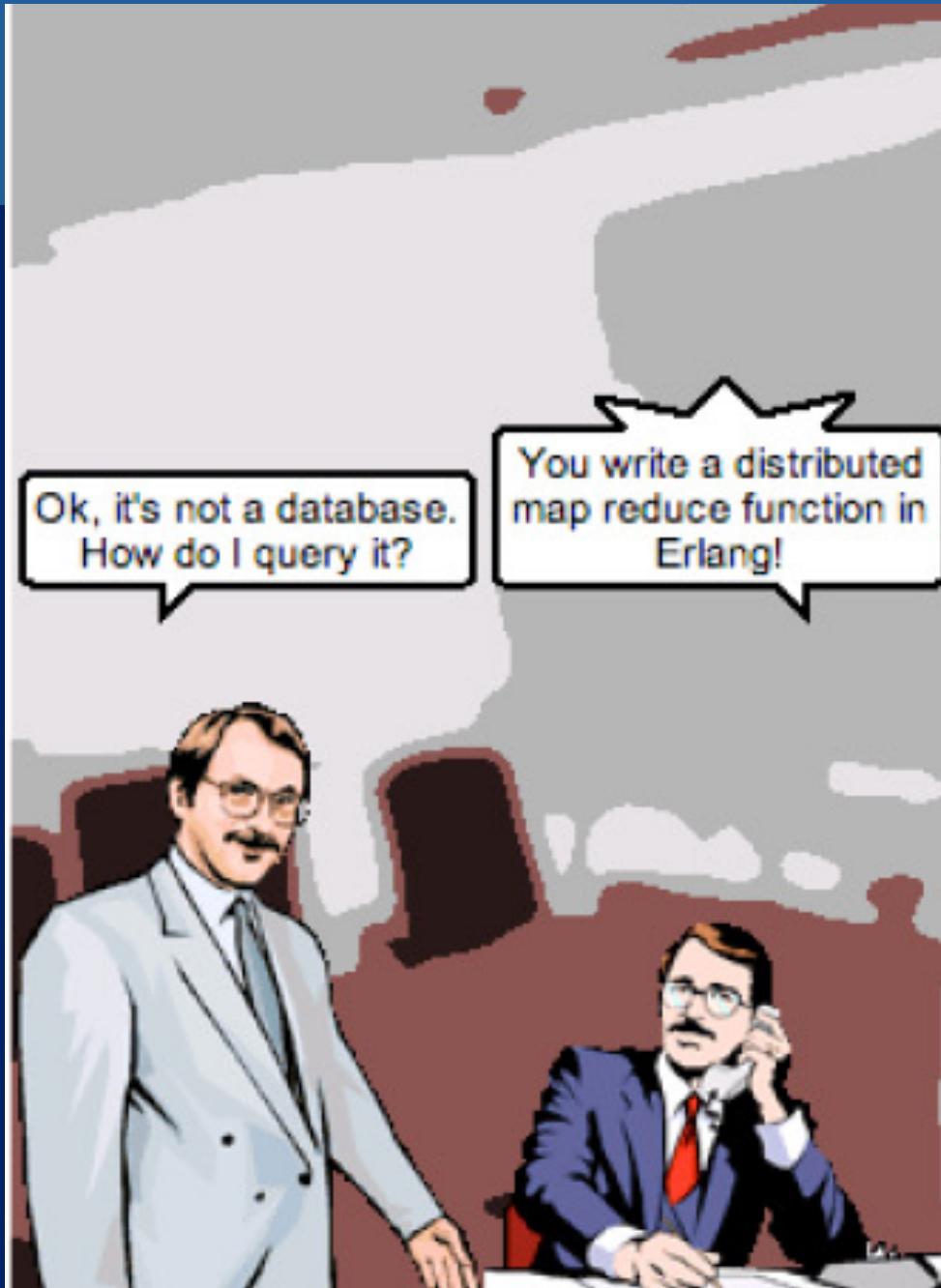
Y! Databases and Key-Value Stores

Fault-tolerance



<http://browsertoolkit.com/fault-tolerance.png>

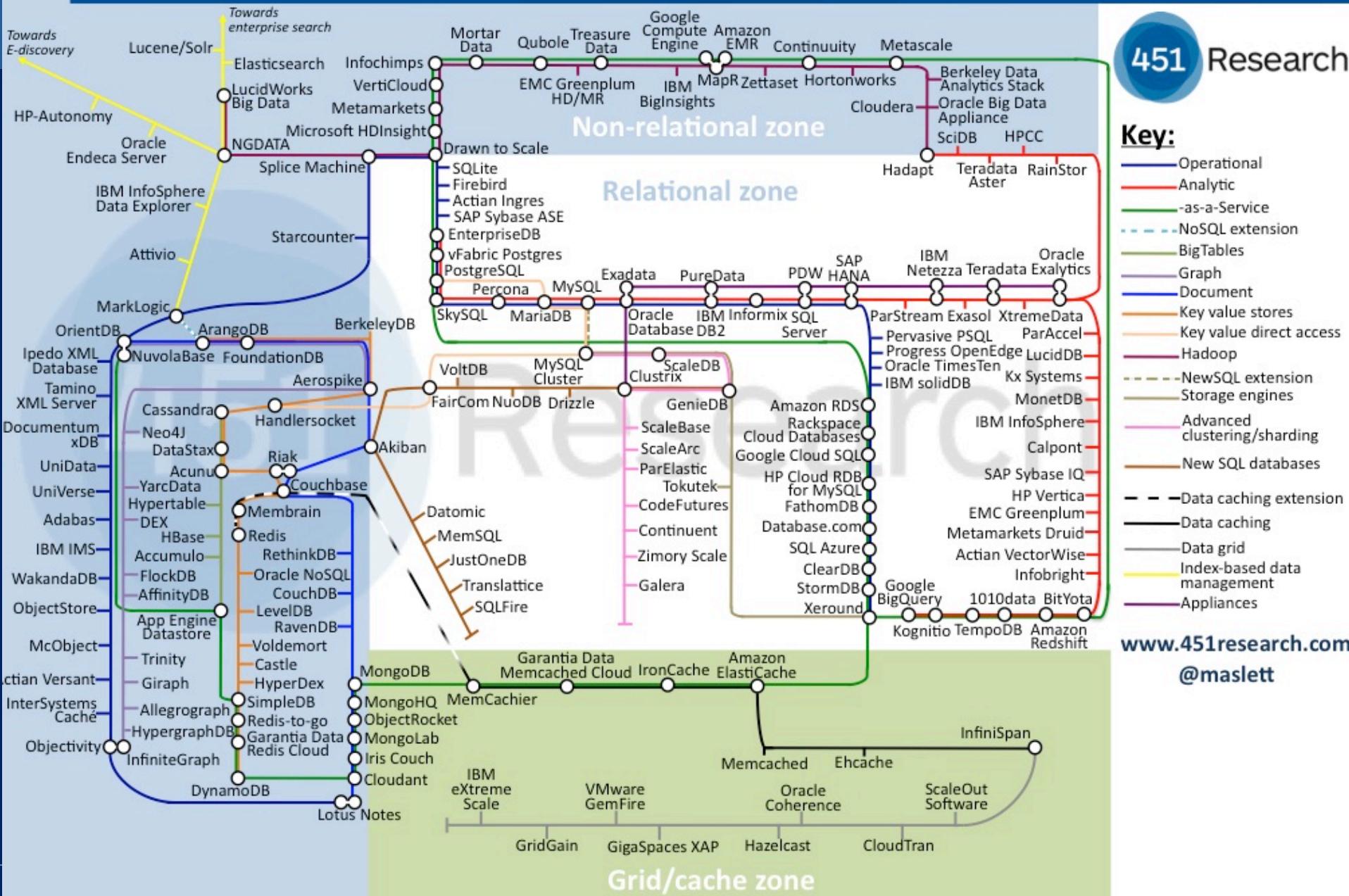






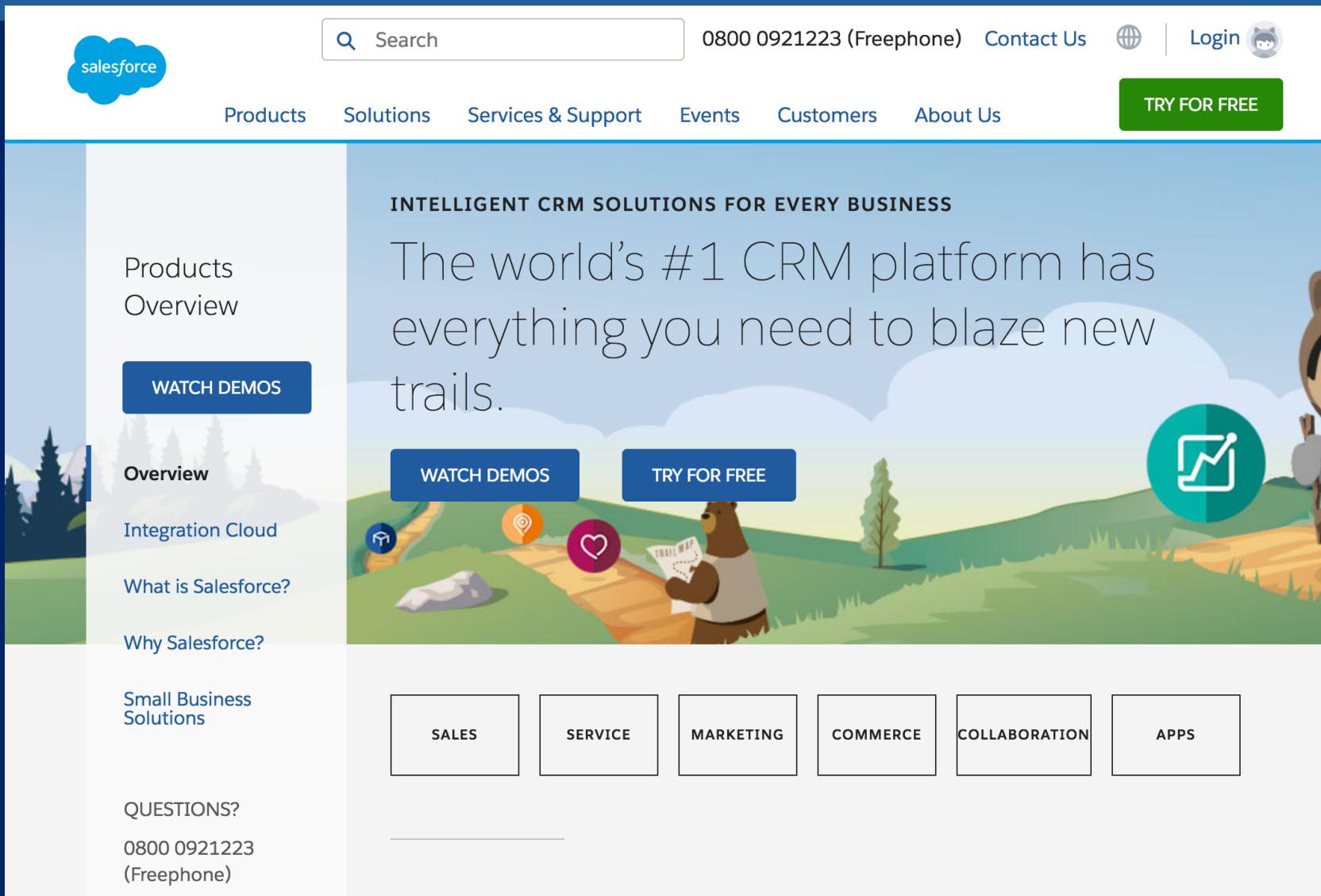
Database Landscape Map – February 2013

451 Research



Other cloud providers are available...

Salesforce.com: pioneers of SaaS



The screenshot shows the homepage of Salesforce.com. At the top, there is a navigation bar with the Salesforce logo, a search bar, phone number (0800 0921223), contact us link, and login options. Below the navigation is a main banner with the heading "INTELLIGENT CRM SOLUTIONS FOR EVERY BUSINESS" and a subtext: "The world's #1 CRM platform has everything you need to blaze new trails." It features a "WATCH DEMOS" button and a "TRY FOR FREE" button. To the left, a sidebar provides links to "Products Overview", "Integration Cloud", "What is Salesforce?", "Why Salesforce?", "Small Business Solutions", and a "QUESTIONS?" section with a phone number. At the bottom, there are six boxes labeled "SALES", "SERVICE", "MARKETING", "COMMERCE", "COLLABORATION", and "APPS".

salesforce

Search 0800 0921223 (Freephone) Contact Us Login

Products Solutions Services & Support Events Customers About Us TRY FOR FREE

INTELLIGENT CRM SOLUTIONS FOR EVERY BUSINESS

The world's #1 CRM platform has everything you need to blaze new trails.

WATCH DEMOS TRY FOR FREE

Products Overview

Integration Cloud

What is Salesforce?

Why Salesforce?

Small Business Solutions

QUESTIONS?
0800 0921223
(Freephone)

SALES SERVICE MARKETING COMMERCE COLLABORATION APPS

Heroku.com: PaaS, acquired by Salesforce in 2010

The screenshot shows the Heroku website homepage. At the top, there's a navigation bar with links for 'Products', 'Elements', 'Pricing', 'Documentation', 'Support', and 'More'. On the right side of the header are a search bar, a 'Log in or' link, and a 'Sign up' button. Below the header is a large, stylized illustration of three mobile devices (a smartphone, a tablet, and a laptop) floating in a cloud-like environment. To the right of the illustration, the word 'ENTERPRISE' is written in capital letters. Below it, a purple text box contains the headline 'Turn your company into an apps company'. Underneath the headline is a paragraph of text: 'Today every company needs apps to engage their customers and run their businesses. Step up your ability to build, manage, and deploy great apps at scale with Heroku.' Below this text is a purple 'SIGN UP FOR FREE' button and a blue 'Explore Heroku Enterprise' link. At the bottom of the page, there's a dark purple footer bar with the text 'Get straight to building apps'.

The page features a prominent headline: "This is not just any cloud. This is the IBM Cloud." Below it, a sub-headline reads: "Built for all your applications. AI ready. Secure to the core." A "Create an account" button is visible. On the right, there's a "Let's talk" button next to a yellow speech bubble icon.

Cloud Why IBM Products Solutions Garage Pricing Blog Docs Support My cloud console [Sign up](#)

Search

This is not just any cloud.
This is the IBM Cloud.

Built for all your applications. AI ready. Secure to the core.

Create an account

Let's talk

Latest news and announcements

IBM expands collaboration with Cloudflare through the [Call for Code](https://developer.ibm.com/callforcode/)

Call for Code asks how the cloud can help us prepare for natural disasters

MSFT: Windows Azure

 Microsoft Azure

Contact Sales: 0800-026-0797  Search  My account Portal  Dave 

Overview  Solutions Products  Documentation Pricing Training Marketplace  Partners  Support  Blog More  Free account 

Your vision. Your cloud.

Turn your ideas into solutions faster using a trusted cloud that has been designed for you.

Azure. Cloud for all.

[Start for free >](#)



Microsoft Ignite

Watch keynotes and sessions



Microsoft Ignite highlights

VMWare: the virtualization foundation

The screenshot shows the VMware website homepage. On the left, a vertical sidebar lists navigation options: VMWARE CLOUD, PRODUCTS, SUPPORT, SOLUTIONS, PROFESSIONAL SERVICES, DOWNLOADS, PARTNER PROGRAMS, and COMPANY. The main content area features the VMware logo and the tagline "Possible Begins with You". It encourages users to "Explore the content catalog, and create your own agenda." A green button labeled "Schedule Your Sessions" is visible. In the bottom right corner, there's a blue button for "Sales Questions? Chat with our sales team". The background features a large, abstract graphic of a person's face composed of dots.

VMWARE CLOUD

PRODUCTS

SUPPORT

SOLUTIONS

PROFESSIONAL SERVICES

DOWNLOADS

PARTNER PROGRAMS

COMPANY

United Kingdom Login Training Community Store 1-877-486-9273

Search

vmware®

Possible Begins with You

Explore the content catalog, and create your own agenda.

Schedule Your Sessions

vmworld™

Possible Begins with You

Now Available: Fusion 11 and Workstation 15

Introd Platine

Sales Questions?
Chat with our sales team

Is that everyone?

Is that everyone?



Is that everyone?

The screenshot shows the Oracle Cloud homepage. At the top left is the "ORACLE Cloud" logo. To the right are several navigation links: "Sign In" (with a user icon), "Contact" (with a phone icon), "Chat" (with a speech bubble icon), "English" (with a globe icon), "Estimate" (in a blue button), and "Buy" (in a blue button). Below these are four main menu items with dropdown arrows: "Applications", "Platform", "Infrastructure", and "Resources". A search bar is located at the bottom right of the header.

Get Started with Oracle Cloud Platform for
Free

[Create your free account >](#)

Up to 3,500 free hours

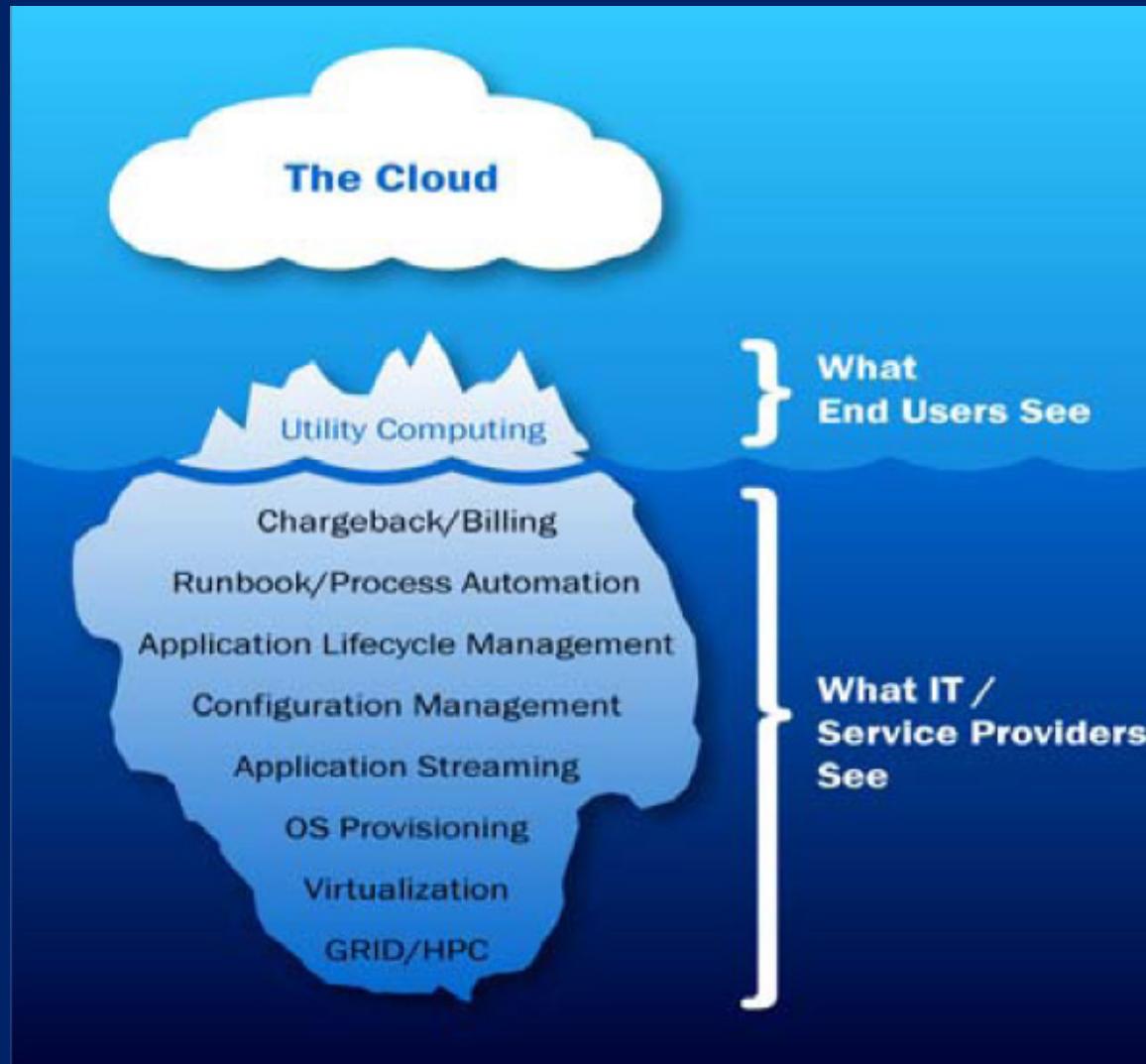
USD300 Free Credits*
available in select
countries and valid for up
to 30 days

Available only on Oracle Cloud



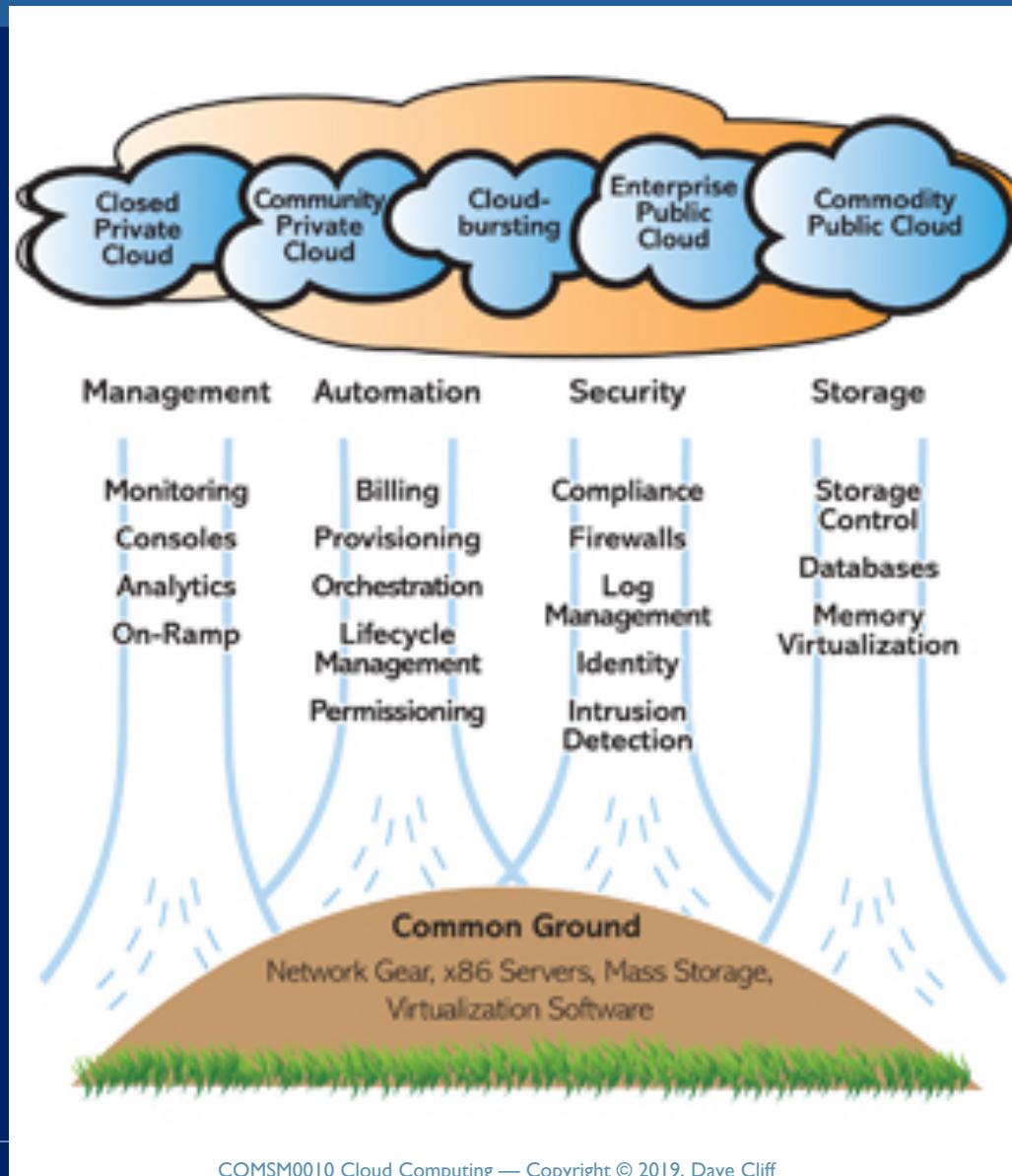
451 Group's “Cloudscape” (W. Fellows, Oct 2009)

<http://www.451group.com/cloudscape/>



451 Group's “Cloudscape” (W. Fellows, Oct 2009)

<http://www.451group.com/cloudscape/>



It ain't easy...

25th Sept 2015

http://www.theregister.co.uk/2015/09/23/aws_outage_explained/

The screenshot shows a web browser displaying an article from The Register. The page has a red header with the site's logo and slogan "Biting the hand that feeds IT". Below the header is a navigation menu with links to various sections like Data Centre, Software, Networks, Security, etc. The main content area features a large headline: "Revealed: Why Amazon, Netflix, Tinder, Airbnb and co plunged offline". Below the headline is a subtext: "And the dodgy database at the heart of the crash is suffering again right now". To the right of the main content is a sidebar titled "More like this" with links to "Amazon" and "Aws". At the bottom of the page is a large image showing two white humanoid figures carrying a large pink rectangular object with the "amazon web services™" logo on it.

DATA CENTRE SOFTWARE NETWORKS SECURITY INFRASTRUCTURE BUSINESS HARDWARE SCIENCE BOOTNOTES FORUMS

Data Centre ▶ Cloud

Revealed: Why Amazon, Netflix, Tinder, Airbnb and co plunged offline

And the dodgy database at the heart of the crash is suffering again right now

More like this

Amazon Aws

IBM White

Learn how to Bu and secure hy

A cloud a developm Build and de applicatio

Best pra a cloud-e Building a se applications

8th Nov 2017

www.theregister.co.uk/2017/11/08/google_memcache_outage_root_cause/



Data Centre ▶ Cloud

Google broke its own cloud, again, with dud DB config change

Memcache was gone in 20 seconds and down for nearly two hours

By [Simon Sharwood](#) 8 Nov 2017 at 08:26

14

SHARE ▼

Google's again 'fessed up to cooking its own cloud.

This time the mess was brief – just under two hours last Monday – and took down its Memcache service. The result was “Managed VMs experienced failures of all HTTP requests and App Engine API calls during this incident.”

There's a little upside in the fact that Google now offers a replacement for Managed VMs, the “App Engine Flexible Environment”, and it stayed up

1st Jun 2018

https://www.theregister.co.uk/2018/06/01/aws_outage/

The Register®
Biting the hand that feeds IT

DATA CENTRE SOFTWARE SECURITY DEVOPS BUSINESS PERSONAL TECH SCIENCE

Use discount code **THEREGISTER** and save €625

Gartner SYMPOS 4 – 8 November, Barcelona,

Data Centre ▶ Cloud

AWS outage killed some cloudy servers, recovery time is uncertain

‘Power event’ blamed, hit subset of kit in US-EAST-1

By [Simon Sharwood](#) 1 Jun 2018 at 00:48

16 SHARE ▼

Updated Parts of Amazon Web Services' US-East-1 region have experienced about half an hour of downtime, but some customers' instances and data can't be restored because the hardware running them appears to have experienced complete failure

Interoperability is an issue...

Open Cloud Manifesto

Dedicated to the belief that the cloud should be open



This work is licensed under a [Creative Commons Attribution-Share Alike 3.0 Unported License](#)

www.opencloudmanifesto.org

Now defunct???

Open Cloud Manifesto

Dedicated to the belief that the cloud should be open



This work is licensed under a [Creative Commons Attribution-Share Alike 3.0 Unported License](#)

www.opencloudmanifesto.org

Open-source software helps...

Openstack: www.openstack.org

The screenshot shows the official OpenStack website at www.openstack.org/software/. The header includes the OpenStack logo, search bar, and navigation links for Software, Users, Community, Marketplace, Events, Blog, Docs, Join, and Log In. A banner at the top encourages exploring containers within an OpenStack infrastructure, with a 'READ NOW' button. Below the banner, the main title 'OpenStack: The Open Source Cloud Operating System' is displayed. The central feature is a diagram illustrating the architecture. It shows 'Your Applications' at the top, connected via 'APIs' to three main components: 'Compute' (yellow cube), 'Networking' (purple cube), and 'Storage' (blue cube). These three components are built on 'OpenStack Shared Services' (light blue layer) which sits atop 'Standard Hardware'. On the left, there's an icon for the 'OpenStack Dashboard'. On the right, the text 'OPENSTACK CLOUD OPERATING SYSTEM' is visible.

www.eucalyptus.com (acquired by HP in Sep 2014)

The screenshot shows a web browser window with the URL www8.hp.com/us/en/cloud/helion-eucalyptus. The page header includes the HP logo and links for 'For Home', 'For Work', and 'Support'. A search bar is at the top right. Below the header, the text 'Solutions' is visible. The main section features the heading 'HP Helion Eucalyptus' and the subtext 'Open source hybrid cloud software for AWS users'. It includes navigation links for 'Overview' (underlined), 'HP Helion Eucalyptus', 'Downloads', and 'Support and Documentation'. A red button labeled 'Contact a cloud expert' is present. The background image shows a close-up of a person's face with a blue diamond graphic overlay containing a quote.

HP Helion Eucalyptus

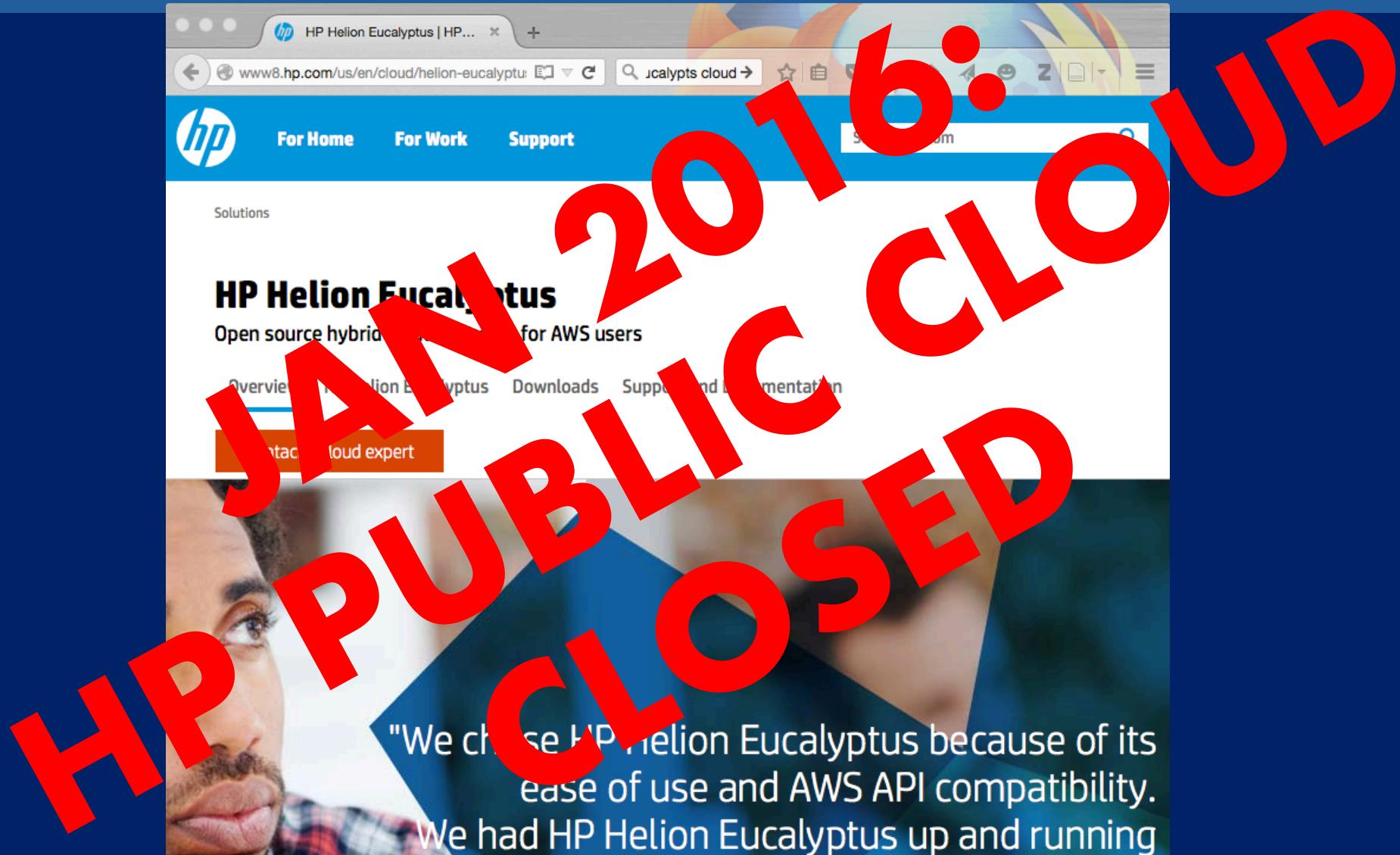
Open source hybrid cloud software for AWS users

Overview HP Helion Eucalyptus Downloads Support and Documentation

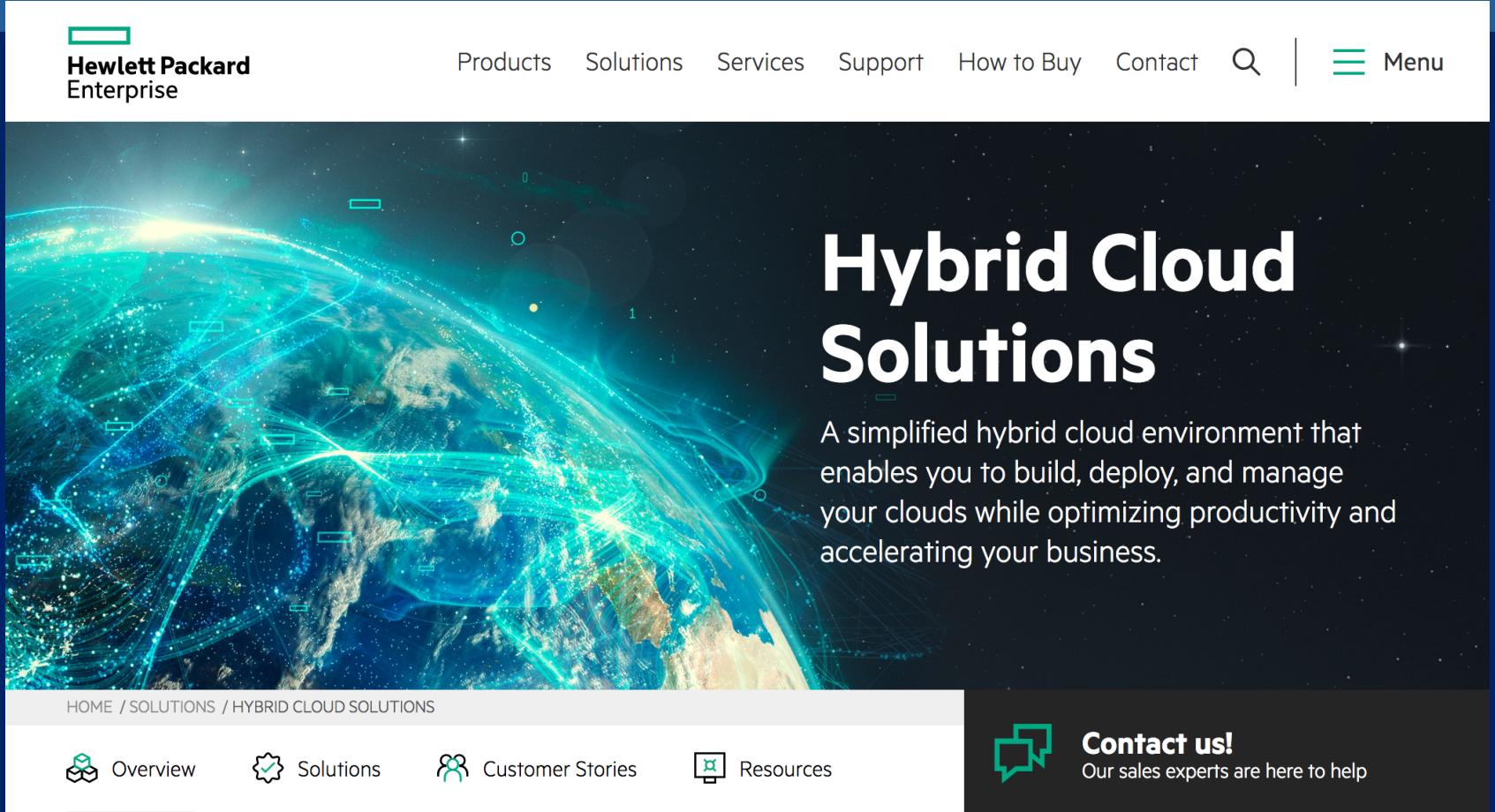
Contact a cloud expert

"We chose HP Helion Eucalyptus because of its ease of use and AWS API compatibility. We had HP Helion Eucalyptus up and running

www.eucalyptus.com (acquired by HP in Sep 2014)



HPE now focused on providing hybrid cloud systems



The image shows the HPE website's "Hybrid Cloud Solutions" page. The header features the HPE logo and navigation links for Products, Solutions, Services, Support, How to Buy, Contact, and a search bar. A large, stylized image of Earth with glowing blue network connections represents a global hybrid cloud environment. The main title "Hybrid Cloud Solutions" is prominently displayed in large white font. Below it is a descriptive paragraph: "A simplified hybrid cloud environment that enables you to build, deploy, and manage your clouds while optimizing productivity and accelerating your business." At the bottom left, there are four menu items: "Overview" (selected), "Solutions", "Customer Stories", and "Resources". On the right, a "Contact us!" button with a speech bubble icon and the text "Our sales experts are here to help" is visible. A circular orange button with a white speech bubble icon is located in the bottom right corner. The footer contains the text "Harness the benefits of private and public clouds" and a paragraph about the advantages of cloud computing.

Hewlett Packard Enterprise

Products Solutions Services Support How to Buy Contact   Menu

Hybrid Cloud Solutions

A simplified hybrid cloud environment that enables you to build, deploy, and manage your clouds while optimizing productivity and accelerating your business.

HOME / SOLUTIONS / HYBRID CLOUD SOLUTIONS

 Overview  Solutions  Customer Stories  Resources

 **Contact us!**
Our sales experts are here to help

Harness the benefits of private and public clouds

With cloud, it's never been easier—or faster—to deploy applications. Without cross-cloud visibility to manage hybrid cloud resources and applications, the result may be unpredictable spending and inefficiency that thwarts productivity and agility.

Summary So Far

- You now know key terminology and the basic concepts
- A small number of big companies jockeying to be IaaS/PaaS providers
- A much large number of SaaS providers are enabled by that IaaS/PaaS
- A variety of types/styles of cloud computing (private / hybrid / public)
- Convergence to an oligopoly of <5 providers seems certain (almost)
- In the past 15 years cloud computing has gone from a wild idea to a central aspect of enterprise IT; lowers barriers to entry; “democratised” computing.

Coming up...

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

- W01 L01 01/10/19 Introduction & Overview (DC)
L02 03/10/19 **9 The Programming Project (DC)**
- W02 L03 08/10/19 Anatomy of a Cloud App (DS)
L04 11/10/19 Cloud Economics (DC)
- W03 L05 15/10/19 *SaaS (DC)
L06 18/10/19 Virtualization (DS)
- W04 L07 22/10/19 Containerization (DS)
L08 25/10/19 Using Docker (DS)
- W05 L09 29/10/19 Serverless (DS)
L10 01/11/19 Using Kubernetes (DS)
- W06 L11 05/11/19 Scalable Software Architectures (DS)
L12 08/11/19 Metrics and Autoscaling (DS)
- W07 L13 12/11/19 Distributed File Systems (DC)
L14 15/11/19 DevOps (DS)
- W08 EXPLORE WEEK
- W09 L15 26/11/19 Hadoop (DC)
L16 29/11/19 Spark (DC)
- W10 L17 03/12/19 Cloud Databases (DC)
L18 06/12/19 Graph Databases (DS)
- W11 L19 10/12/19 Stream Processing (DC)
L20 13/12/19 Cloud Security (DC)
- W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *aaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 **9 Containerization (DS)**

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 **Using Docker (DS)**

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 9 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Auto-scaling (DS)

W07 L13 12/11/19 **9 Distributed File Systems (DC)**

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK



Professor Dave Cliff
Professor of Computer Science

W01 L01 01/10/19 Introduction & Overview (DC)

L02 03/10/19 The Programming Project (DC)

W02 L03 08/10/19 Anatomy of a Cloud App (DS)

L04 11/10/19 Cloud Economics (DC)

W03 L05 15/10/19 *SaaS (DC)

L06 18/10/19 Virtualization (DS)

W04 L07 22/10/19 Containerization (DS)

L08 25/10/19 Using Docker (DS)

W05 L09 29/10/19 Serverless (DS)

L10 01/11/19 Using Kubernetes (DS)

W06 L11 05/11/19 Scalable Software Architectures (DS)

L12 08/11/19 Metrics and Autoscaling (DS)

W07 L13 12/11/19 Distributed File Systems (DC)

L14 15/11/19 DevOps (DS)

W08 EXPLORE WEEK

W09 L15 26/11/19 Hadoop (DC)

L16 29/11/19 Spark (DC)

W10 L17 03/12/19 Cloud Databases (DC)

L18 06/12/19 Graph Databases (DS)

W11 L19 10/12/19 Stream Processing (DC)

L20 13/12/19 Cloud Security (DC)

W12 REVISION WEEK



- W01 L01 01/10/19 Introduction & Overview (DC) FA1 set: deadline 15/10/19
- L02 03/10/19 The Programming Project (DC)
- W02 L03 08/10/19 Anatomy of a Cloud (DC)
- L04 11/10/19 Cloud Economics (DC)
- W03 L05 **15/10/19** * (DC)
- L06 18/10/19 Virtualization (DS)
- W04 L07 22/10/19 Containerization (DS)
- L08 25/10/19 Using Docker (DS)
- W05 L09 29/10/19 Serverless (DS)
- L10 01/11/19 Using Kubernetes (DS)
- W06 L11 05/11/19 Scalable Software Architectures (DS)
- L12 08/11/19 Metrics and Autoscaling (DS)
- W07 L13 12/11/19 Distributed File Systems (DC)
- L14 15/11/19 DevOps (DS)
- W08 EXPLORE WEEK
- W09 L15 26/11/19 Hadoop (DC)
- L16 29/11/19 Spark (DC)
- W10 L17 03/12/19 Cloud Databases (DC)
- L18 06/12/19 Graph Databases (DS)
- W11 L19 10/12/19 Stream Processing (DC)
- L20 13/12/19 Cloud Security (DC)
- W12 REVISION WEEK

- W01 L01 01/10/19 Introduction & Overview (DC) FA1 set: deadline 15/10/19
- L02 03/10/19 The Programming Project (DC) CW1 set: deadline 06/12/19
- W02 L03 08/10/19 Anatomy of a Cloud App (DS)
- L04 11/10/19 Cloud Economics (DC)
- W03 L05 15/10/19 *SaaS (DC)
- L06 18/10/19 Virtualization (DS)
- W04 L07 22/10/19 Containerization (DS)
- L08 25/10/19 Using Docker (DS)
- W05 L09 29/10/19 Serverless (DS)
- L10 01/11/19 Using Kubernetes (DS)
- W06 L11 05/11/19 Scalable Software Architectures (DS)
- L12 08/11/19 Metrics and Auto-scaling (DS)
- W07 L13 12/11/19 Distributed File Systems (DC)
- L14 15/11/19 DevOps (DS)
- W08 EXPLORE WEEK
- W09 L15 26/11/19 Hadoop (DC)
- L16 29/11/19 Spark (DC)
- W10 L17 03/12/19 Cloud Databases (DC)
- L18 06/12/19 Graph Databases (DS)
- W11 L19 10/12/19 Stream Processing (DC)
- L20 13/12/19 Cloud Security (DC)
- W12 REVISION WEEK

FAI: Feedback Assessment #1 (optional)

- **FAI:** write a one-page summary (no more than one page) of what cloud computing is, and why it is significant. It should be possible to write this summary from the content of this Lecture 1 (and Lecture 3), and Dave Cliff's BECTA article (cited above).
- You are encouraged to include additional material not covered in the lectures if/when you feel it is appropriate to do so.
- The aim of this assessment is to give you practice/feedback at writing to a word-limit, and to identify any problems or issues with your writing style.
- Format: A4 paper-size, single column, one-inch margins all round, and 11pt text. In a standard font (e.g. like Times New Roman) you should easily be able to fit 500 words on a single page.
- Give your file a UNIQUE filename – use your student/candidate number
 - e.g.: dc12345 in the filename: dc12345FAI.pdf (please submit either .pdf or .docx)
- The deadline for submission of FAI is **Tuesday 15th Oct, 2019**.

Cloud Computing L01: sample exam questions

Q1. For each of the following acronyms, state what phrase the acronym abbreviates, explain what the phrase means, and name a commercial example (i.e., one provided by a private company): IaaS; PaaS; SaaS. (9 marks).

Q2. The USA's National Institute of Standards and Technology (NIST) established a de facto standard definition of cloud computing, involving five key concepts: on-demand self-service; broad network access; resource pooling; rapid elasticity; and measured service. Explain what each of these concepts mean. (10 marks).

Q3. What is meant by “normal failure” in the context of managing cloud-computing data-centres? (3 marks).

Masters Thesis projects

- If you want to do your Masters Thesis project with me possibly/probably working with major industry partner, please email me ASAP.
- I have loads of ideas
- Lots of times I have helped students' project work be published as papers in international peer-review conferences: good for CV!
- E.g. today, on [www.arxiv.org...](http://www.arxiv.org)

Masters Thesis projects

arXiv.org > cs > arXiv:1909.12926

Computer Science > Computers and Society

A Cloud-Native Globally Distributed Financial Exchange Simulator for Studying Real-World Trading—Latency Issues at Planetary Scale

Bradley Miles, Dave Cliff

(Submitted on 18 Sep 2019)

We describe a new public-domain open-source simulator of an electronic financial exchange, and of the traders that interact with the exchange, which is a truly distributed and cloud-native system that has been designed to run on widely available commercial cloud-computing services, and in which various components can be placed in specified geographic regions around the world, thereby enabling the study of planetary-scale latencies in contemporary automated trading systems. Our simulator allows an exchange server to be launched in the cloud, specifying a particular geographic zone for the cloud hosting service; automated-trading clients which attach to the exchange can then also be launched in the cloud, in the same geographic zone and/or in different zones anywhere else on the planet, and those clients are then subject to the real-world latencies introduced by planetary-scale cloud communication interconnections. In this paper we describe the design and implementation of our simulator, called DBSE, which is based on a previous public-domain simulator, extended in ways that are inspired by the architecture of the real-world Jane Street Exchange. DBSE relies on TCP network communications protocols and implements a subset of the information exchange. We show results from an experiment in which a client connects to a cloud facility located in

Search...

All fields

Search

Help | Advanced Search

Download:

- PDF only
(license)

Current browse context:
cs.CY
< prev | next >
new | recent | 1909

Change to browse by:

cs
q-fin
q-fin.TR

References & Citations

- NASA ADS

Export citation
Google Scholar

Bookmark



Science
WISE

To be presented at the European Modelling and Simulation Symposium (EMSS2019).
Lisbon, Portugal, 18th-20th September 2019.

A CLOUD-NATIVE GLOBALLY DISTRIBUTED FINANCIAL EXCHANGE SIMULATOR FOR STUDYING REAL-WORLD TRADING-LATENCY ISSUES AT PLANETARY SCALE

Bradley Miles^(a) and Dave Cliff^(b)

Department of Computer Science
University of Bristol, Bristol, BS8 1UB, U.K.

^(a)bm15731.2015@my.bristol.ac.uk, ^(b)csdtc@bristol.ac.uk,

ABSTRACT

We describe a new public-domain open-source simulator of an electronic financial exchange, and of the traders that interact with the exchange, which is a truly distributed and cloud-native system that been designed to run on widely available commercial cloud-computing services, and in which various components can be placed in specified geographic regions around the world, thereby enabling the study of planetary-scale latencies in contemporary automated trading systems. The speed at which a trader can react to changes in the market is a key concern in current financial markets but is difficult to study latency issues using conventional market simulators, and is extremely difficult to study "in the wild" because of the financial and regulatory barriers to entry in conducting experimental work on real financial exchanges. Our simulator allows an exchange server to be launched in the cloud, specifying a particular geographic zone for the cloud hosting service; automated-trading clients which attach to the exchange can then also be launched in the same geographic zone as the exchange.

highly-paid individuals on the floors of major financial exchanges. By today's standards, these interactions between skilled human traders were slow, inefficient, and often error prone. Consequently, as soon as the relevant technologies were available, the buying and selling of financial products became a digital interaction and the traditional trading floors in national financial exchanges around the world were gradually closed. Modern-day electronic financial exchanges are highly sophisticated and complicated distributed computational systems that enable institutions, such as investment banks, fund managers, brokers, and insurance companies, to remotely connect and trade on the world's open markets. Although the transition from physical to electronic markets was largely completed (or at least inevitable) by the end of the twentieth century, the