Cloud Computing

Jim Baker

jim.baker@{rackspace.com, python.org}

Overview

- What is this cloud computing thing? - one very big idea
- APIs
- SaaS, PaaS, Iaas, and other marketing terms

About me

- Architect at Rackspace, focused on platformization, cloud computing, and big data
- Once and future lecturer for CSCI 3155 Principles of Programming Languages
- Formerly on Ubuntu Server team at Canonical
- Formerly at Sauce Labs, supporting Selenium testing in the cloud
- Founding Juju team member, working on service orchestration, for the cloud
- Core developer of Jython and fellow of the Python Software Foundation
- Co-author of *Definitive Guide to Jython* from Apress
- Enjoy outdoor recreation and frequent travel!

The Economist, October 10 issue

- Demonstrates "cloud computing" is a popular term in the wider economy
- "Marketectures" (marketing + architecture) and ad bingo do make it more cloudy...
- Also cloud and the law, specifically issues around data sovereignty
- (anyone interested in Silicon Flatirons?)

One big idea

(blackboard)

One big idea

- Delegation of responsibility
- = the client does not care about servers, just services
- This can include name lookups - Domain Name Service (DNS) or service catalogs like Keystone
- Enables horizontal scaling, across possibly globally distributed data centers

Indirection and its power

> *All problems in computer science can be solved by another level of indirection.*
>
> — David Wheeler

Need to have a base case. . .

> *All problems in computer science can be solved by another level of indirection.*
>
> — David Wheeler
>
> *. . . except for the problem of too many layers of indirection*
>
> — Kevlin Henney

Completing indirection

?

Completing indirection of delegating responsibility

- DNS
- TCP/IP
- Certificate authorities for SSL/TLS to validate cert chain
- Schema catalogs in relational databases
- Content distribution networks (CDNs)
- etc etc

Another good idea: APIs

- Services should have APIs. . .
- which supports programmability
- Enables further scaling
- Check out DevOps (Developer/Operations) and similar terms

DevOps Boulder meetup

- DevOps Boulder
- Probably not OK if you crash this meeting tonight
- But do join and attend in the future!

SOA - Service Oriented Architecture

- Enables a base set of products to be extended via combination and further refinement
- Various implementation strategies - WSDL-based services, REST-based services
- But needs a common platform to combine together

Steve Yegge and The Google Platforms Rant

Or lessons learned on how Amazon learned to love platforms

- Summary
- Unfortunate public posting by Steve Yegge
- Steve was not fired after all...
- Steve is also an occasional user of and contributor to Jython, nice!

Being cloudy

Going up the stack:

- Infrastructure as a Service (IaaS)
- Platform as a Service (PaaS)
- Microservices (no, we don't call this services as a service!)
- Software as a Service (SaaS)

SaaS

- Started first - what if we took existing apps, made them available via a browser. . .
- Browser-native apps - Salesforce, moved to mobile
- Generally worked by sharding (by tenant, customer), lots of glueing
- Increasingly “cloud native” (do define!)

PaaS

- Heroku, Cloud 9 (Sauce. . .) - labs - great workflows, easy to try out ideas
- Google Cloud
- OpenStack Magnum shades into this, but less limitations

Microservices

Examples include:

- Mapping - including the original successful microservice, Google Maps
- Payment platforms
- Machine learning
- etc

Microservices

Examples include:

- Mapping - including the original successful microservice, Google Maps
- Payment platforms
- Machine learning
- etc

Note the analogue to business to business services, such as credit card processing, including new variants like Square

IaaS

- The data center has an API
- And we can connect to multiple data centers (DCs, aka regions) and availability zones (subdivided DCs)
- Instead of weeks, we get instances in seconds/minutes
- Generally using hypervisors, but also lighterweight containers (Docker/Kubernetes), bare metal (OpenStack Ironic)
- AWS, Azure, Google Cloud, OpenStack (such as provided by Rackspace or HP Public Cloud), or on your own DC
- Key terms include provisioning, discovery, ...

Cloud and the law

- Data sovereignty/data residency
- European Court of High Justice recent ruling that bilateral *safe harbor* agreement between US and EU is invalid
- HIPAA, PII, e-commerce considerations

Delegating the hard work to others

- Early microservices like PayPal - ecommerce is not just for Amazon and its affiliates
- Cloud services can be used to solve regulatory and legal compliance issues

Amazon Web Services - AWS

- EC2 - “elastic computing cloud” - buy computing by the minute
- S3 - “simple storage service” for *object* storage (does S3 support incremental patches, or only replacement?)
- Many other services - block storage (EBS), notification, stream processing (Kinesis), . . .
- Or set up your own EC2 - Eucalyptus (now part of HP)

OpenStack

Started as a collaboration between NASA and Rackspace, since has grown tremendously:

- Keystone - identity, service catalog
- Nova - compute
- Swift - storage
- Neutron - networking
- many, many other projects

Contrast: use your own servers

- You might just choose Docker or Vagrant
- You will see similar emphasis on programmability, even similar APIs
- Can control uniformly with cloud virtualization services like libcloud and JClouds

“docker docker docker” talk at the OpenStack Vancouver Summit

- Just add Docker to your talk title?
- Why not just add it multiple times? ;)
- There are always **hot**, must know technologies out there

Juju - a “canonical” example

Use a GUI or drive from the command line:

```
juju deploy mysql
juju deploy wordpress
juju add-relation wordpress mysql
juju expose wordpress
```

then scale up with

```
juju add-unit wordpress
```

Deploying Hadoop with Juju


```
$ juju deploy hadoop hdfs-datacluster-02
$ juju add-unit -n 2 hdfs-datacluster-02
$ juju add-relation hdfs-namenode:namenode \
    hdfs-datacluster-02:datanode
```

Implementing clouds

- Multitenancy
- Scaling via sharding/partitioning
- Immutability
- Shared nothing architectures
- Data - strong consistency vs eventual consistency
- SQL vs NoSQL
- Blue/green
- Virtualization - jails, ability to escape the jail
- Functional programming, referential transparency

Immutability

- Content distribution networks (CDNs)
- and the power of immutability! (in terms of being able to reason about it)
- Netflix

Relativity!

At scale, sequencing is expensive!

- Local sequencing is fairly cheap
- Maintaining order requires communication
- Communication proceeds no faster than the speed of light
- Unless we have ansibles ;)

Question

How far does light in a vacuum approximately travel in one **nanosecond**?

- A - 1 kilometer
- B - 1 meter
- C - 1 foot
- D - 1 cm
- E - 1 mm

An interesting unit: light-foot

- Useful unit: a *light-foot* $\approx$ 1.0167 nanoseconds
- Useful in the same way that units like tablespoons are useful - everyday intuitions
- Pioneering computer scientist Grace Hopper liked to talk about this unit
- Need to consider the **velocity factor**
- Consider a 1 foot USB cable:
  - No specifics about velocity factor on USB cables I could find
  - But gives some insight into what a nanosecond really is

Data center design

- It's all about the locality, to minimize communication hops and distance
- Same core, same chip, same board, same unit, same rack, same aisle, same data center. . .
- Design focused on communication latency as much as it's storage, computation

Data centers, illustrated

- Google streetview in the datacenter

Multidata center coordination

- Big problem because of communication bottlenecks
- Bigger problem because of data center connection reliability
- These issues are **related**!
- Datacenters are now distributed around the world
- Observations of ping time between cities by one network provider
- What could possibly go wrong?!!

Plug for CSCI 3155

- Why is PoPL - a theory course - one of the most pragmatic courses in the CS curriculum?
- A: functional programming

SQL vs NoSQL

It's not about "SQL" because many so-called NoSQL databases have a SQL-like query language. Instead it is about the cost of doing distributed operations:

- Transactions
- Joins

SQL vs NoSQL

(blackboard)

Chaos Monkey

Atomic OS

Blue/green

- Fedora Atomic
- CoreOS (Rackspace relationship. . .)

Cloud init and existentialism

Consider the whale in the Hitchhikers Guide to the Galaxy

- Who am i?
- What should i do?
- cloud init is the same idea - we need to assign identity to our servers so they can become part of the service, we can orchestrate them, etc