Développement logiciel pour le Cloud (TLC)

Introduction

Quentin Dufour

Goals

- Understand functionalities offered by Cloud computing
- Understand which issues are solved (or not) by the cloud
- Understand how cloud computing platforms are organized internally
- Understand how software developers can make use of these offerings

Course and Instructors

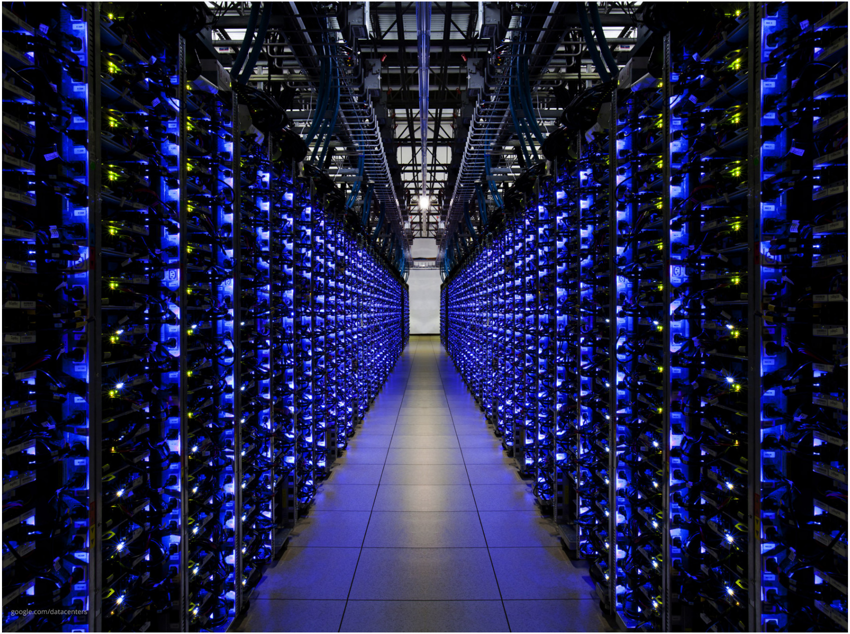
- CM course
  - Quentin Dufour quentin.dufour@inria.fr
- Practical Sessions
  - Quentin Dufour quentin.dufour@inria.fr

Introduction to Cloud Computing

Développement logiciel pour le Cloud (TLC)

Quentin Dufour

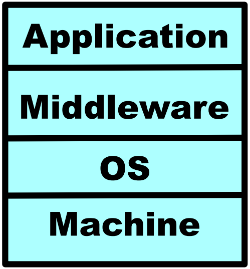
(credits: Davide Frey, François Taiani, Guillaume Pierre)



google.com/datacenters

https://lafibre.info/scaleway/dc5/
https://lafibre.info/scaleway/dc3-iliad/

Traditional system architectures



Application

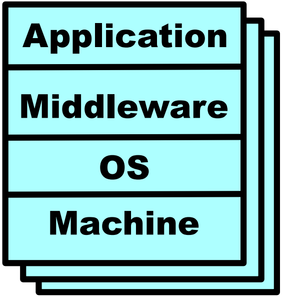
Middleware

OS

Machine

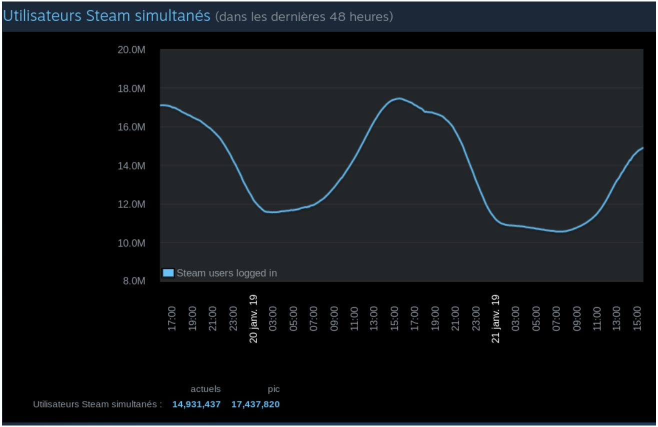
Traditional
architecture

Traditional system architectures



Traditional architecture

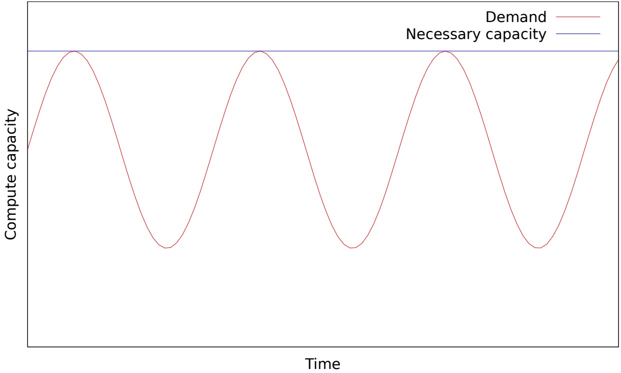
The varying capacity problem



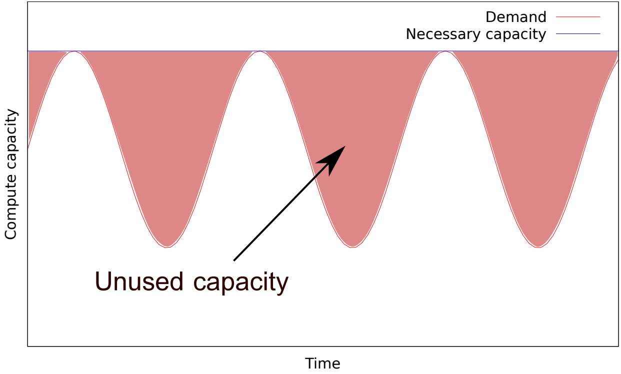
Utilisateurs Steam simultanés (dans les dernières 48 heures)

	actuels	pic
Utilisateurs Steam simultanés :	14,931,437	17,437,820

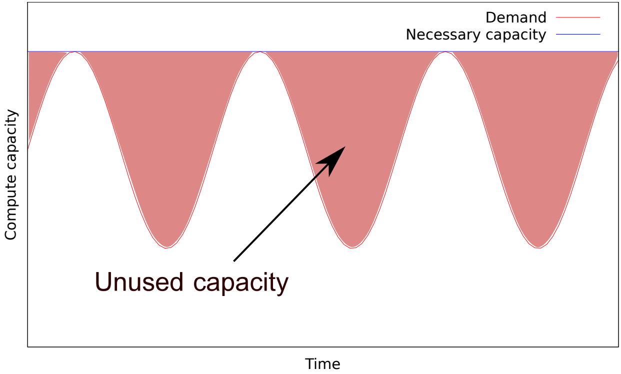
The varying capacity problem



The varying capacity problem



The varying capacity problem

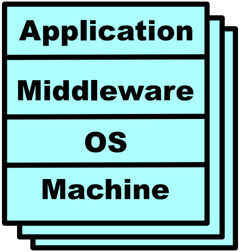


What if demand increases beyond the capacity?

Cloud Computing

- ☹ Difficult to vary capacity!
- ☹ Manual resource management

- ☺ Resources available on demand
- ☺ Resource management is fully automated
- ☺ Pay only for what you use

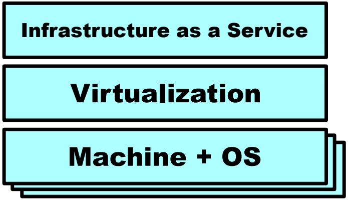

Application

Middleware

OS

Machine

Traditional architecture

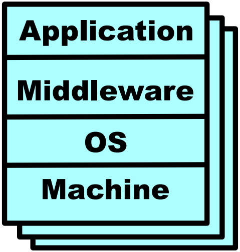

Infrastructure as a Service

Virtualization

Machine + OS

Cloud architecture

Cloud Computing



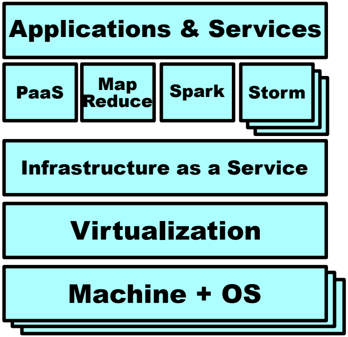
Application

Middleware

OS

Machine

Traditional architecture



Applications & Services

PaaS

Map Reduce

Spark

Storm

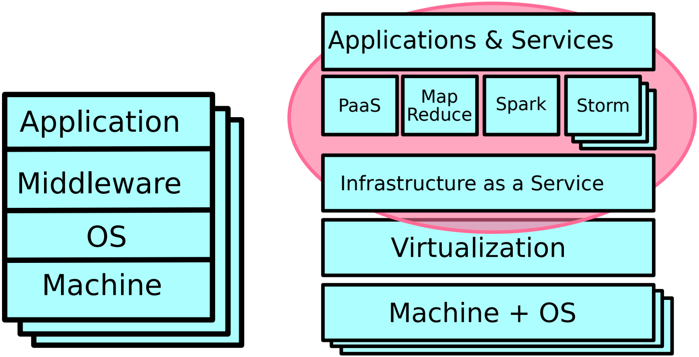
Infrastructure as a Service

Virtualization

Machine + OS

Cloud architecture

Cloud Computing



Traditional architecture

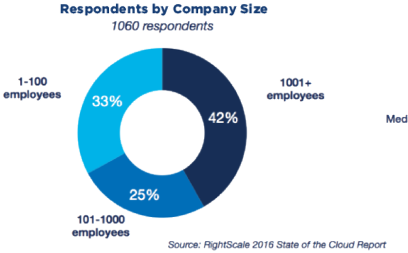
Cloud architecture

What defines Cloud computing exactly?

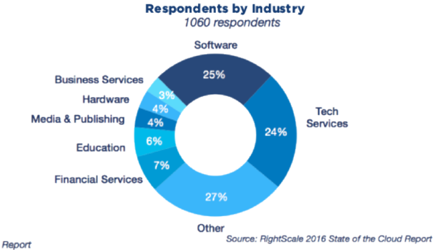
1. Computing is considered as a service customers use, not as hardware they own
   - This is called utility computing
2. Cloud providers offer a collection of compute/storage/network services via the Internet
   - Customers cannot get physical access to the devices
   - The actual location of devices is (almost) irrelevant
3. The cloud hides the complexity and details of the physical infrastructure from its users
   - Users only see a simple API + a graphical interface
4. Services are available on demand
   - Always available, anywhere, anytime
5. Pay-per-use
   - Pay only for the resources you actually use. You can release resources any time and stop paying immediately.

Who uses the cloud?

Everybody USES THE CLOUD. . .

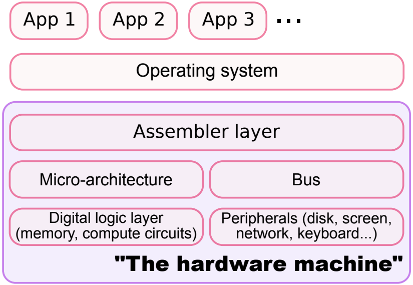


Source: RightScale 2016 State of the Cloud Report



Source: RightScale 2016 State of the Cloud Report

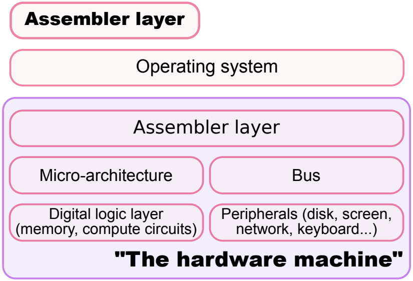
Virtualization


App 1
App 2
App 3
...
Operating system
Assembler layer
Micro-architecture
Bus
Digital logic layer (memory, compute circuits)
Peripherals (disk, screen, network, keyboard...)
"The hardware machine"

Traditional machine architecture:

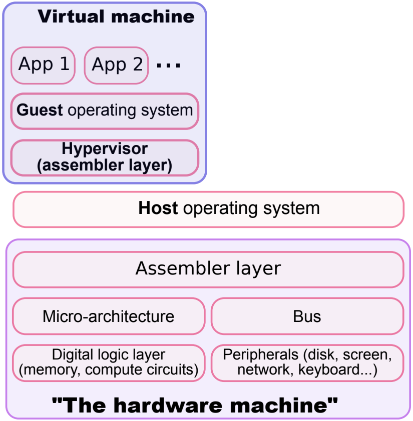
- Applications
- Operating system
- Hardware

Virtualization


Assembler layer
Operating system
Assembler layer
Micro-architecture
Bus
Digital logic layer (memory, compute circuits)
Peripherals (disk, screen, network, keyboard...)
"The hardware machine"

Let's create a "special application" which behaves exactly the same as the assembler layer. . .

Virtualization

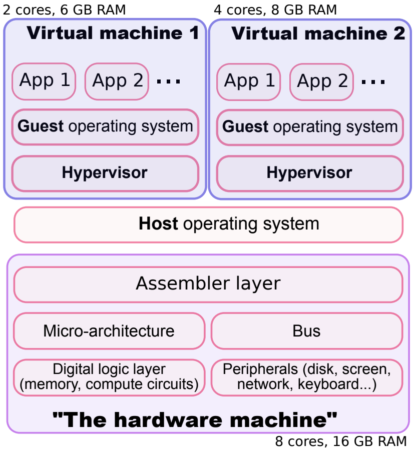

Virtual machine
App 1
App 2
...
Guest operating system
Hypervisor (assembler layer)
Host operating system
Assembler layer
Micro-architecture
Bus
Digital logic layer (memory, compute circuits)
Peripherals (disk, screen, network, keyboard...)
"The hardware machine"

We can execute any operating system on top of it. . .

. . . and any application over the guest operating system

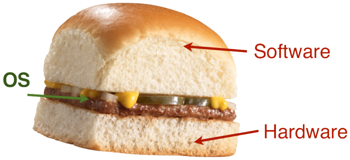
⇒ We have a virtual machine

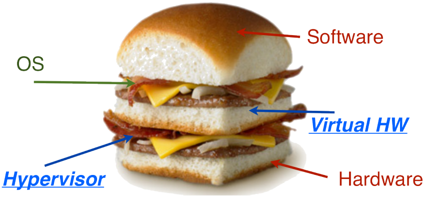
Virtualization



We can run multiple virtual machines on the same physical machine:

- Each virtual machine runs in full isolation from the other VMs
- Each virtual machine owns a subset of the hardware resources of the physical machine


Software
OS
Hardware


Software
OS
Virtual HW
Hypervisor
Hardware

Why is virtualization interesting for cloud providers?

Isolation: I can create multiple VMs on the same machine and give each VM to a different user (they will not see nor interfere with each other)

Customization: Each user can customize their VMs according to their own requirements.

Consolidation: Few applications can really exploit a large server machine to its maximum capacity. With virtualization I can split this capacity in smaller units and thereby increase my resource utilization.

Management: Virtualization simplifies resource management: I can measure how many resources each user is using, migrate VMs from one host to another, etc.

Virtualization technologies

Virtualization technologies are now totally mainstream:

- Commercial: VMware, Microsoft App-V, . . .
- Open-Source: KVM, VirtualBox, Xen, . . .

Paravirtualization vs. full virtualization:

- Paravirtualization works on any hardware platform but it requires special support in the guest OS.
- Full virtualization does not require special support in the guest OS.
  - Originally done through binary translation of system calls
  - Today it exploits special features of modern CPUs,

Virtualization technologies

Virtualization technologies are now totally mainstream:

- Commercial: VMware, Microsoft App-V, . . .
- Open-Source: KVM, VirtualBox, Xen, . . .

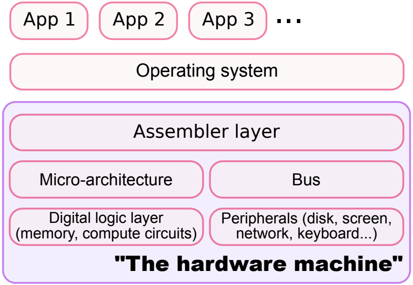
Paravirtualization vs. full virtualization:

- Paravirtualization works on any hardware platform but it requires special support in the guest OS.
- Full virtualization does not require special support in the guest OS.
  - Originally done through binary translation of system calls
  - Today it exploits special features of modern CPUs,

Virtual machines are often considered too heavyweight ☹

- Startup time $\sim$ 2–5 minutes
- Each guest OS needs lots of memory
- Each OS needs to execute lots of background stuff
- Impossible to run 100+ VMs on a single machine. . .

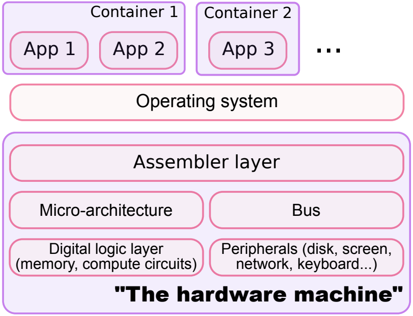
Containers (zones, jails, etc.)


App 1
App 2
App 3
...
Operating system
Assembler layer
Micro-architecture
Bus
Digital logic layer (memory, compute circuits)
Peripherals (disk, screen, network, keyboard...)
"The hardware machine"

Traditional machine architecture:

- Applications
- Operating system
- Hardware

Containers (zones, jails, etc.)



Let's create groups of processes which belong together:

- Process groups are totally isolated from each other
- Each process group has its own hardware resource limits (CPU, RAM, . . .)
- Each process group has its own network access policy

⇒ We have containers

Container technologies

- Containers are an operating system feature
  - No need for special CPU support
  - Fully supported in Linux, Windows, BSD, Solaris...
  - Built with composable primitives on Linux (namespaces, cgroups, etc.) so add an extra software layer to simplify management like Docker, LXC, rkt
- Containers are less customizable than VMs
  - Container owners cannot choose their OS
  - But was that really necessary in the first place? Not always.
- Containers are much more lightweight than VMs
  - No need to run lots of (mostly identical) operating systems next to each other
  - Containers often start in less than 1 second
  - Can easily run hundreds of containers on a mid-sized machine

Containers provide indispensable flexibility for micro-service architectures

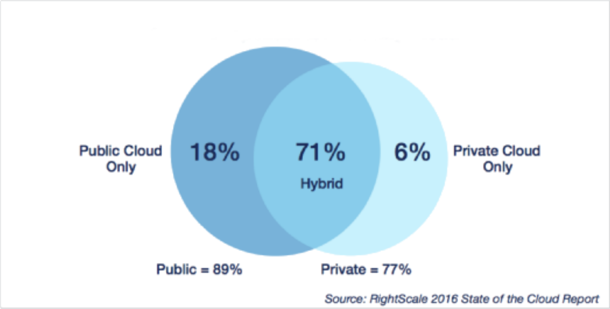
Public clouds vs. private clouds

Public clouds:

- Resources are owned by an external company (Amazon, Microsoft, Google, . . .)
- Pay per use (credit card)

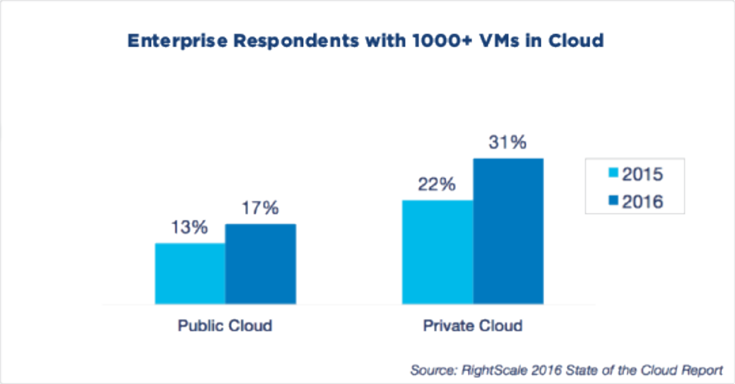
Private clouds:

- Resources are owned by your own company
- Internal resource accounting



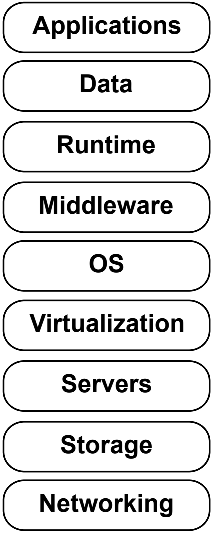
Source: RightScale 2016 State of the Cloud Report

Enterprises are increasingly relying on the cloud



Source: RightScale 2016 State of the Cloud Report

Cloud service layers



Applications

Data

Runtime

Middleware

OS

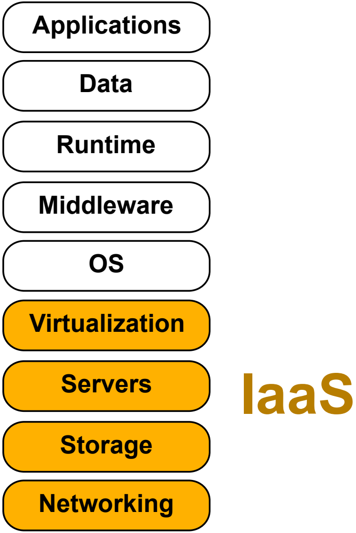
Virtualization

Servers

Storage

Networking

Cloud service layers



Applications

Data

Runtime

Middleware

OS

Virtualization

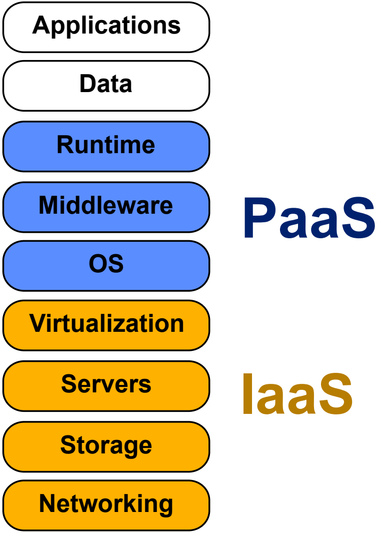
Servers

Storage

Networking

IaaS

Cloud service layers



Applications

Data

Runtime

Middleware

OS

Virtualization

Servers

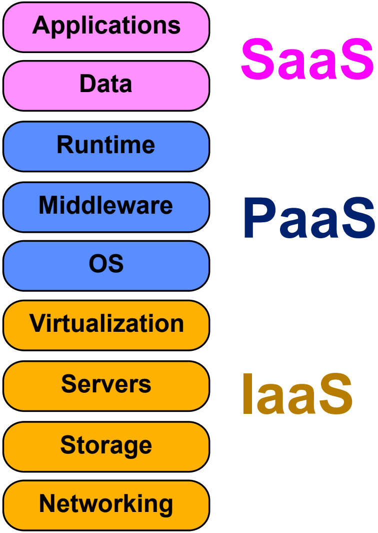
Storage

Networking

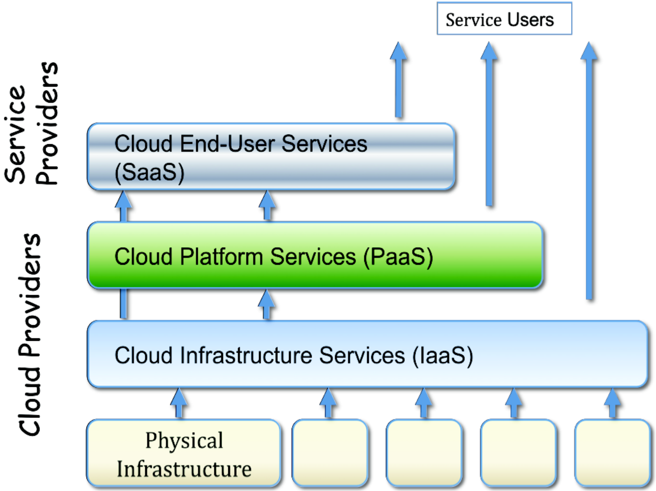
PaaS

IaaS

Cloud service layers



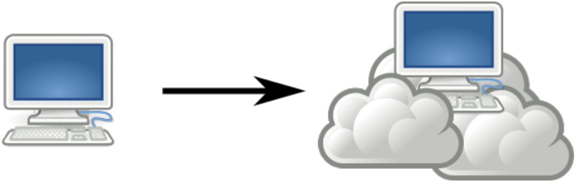
Cloud service layers


Service Users
Service Providers
Cloud Providers
Cloud End-User Services (SaaS)
Cloud Platform Services (PaaS)
Cloud Infrastructure Services (IaaS)
Physical Infrastructure

The main Cloud service layers

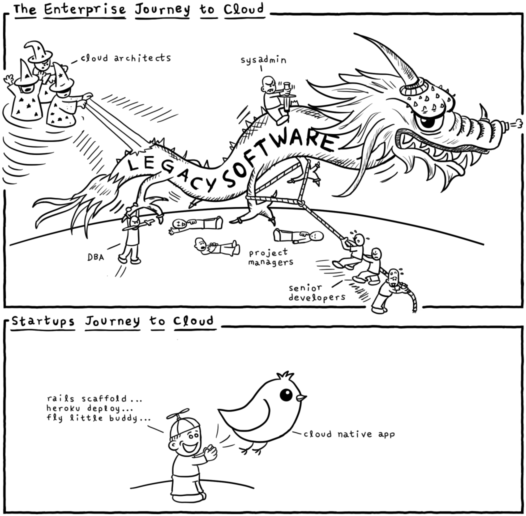
- Infrastructure-as-a-Service (IaaS) offers basic computing services
  - Computing: “Create a new machine for me”
  - Data storage: “Store/retrieve this data for me”
  - Communication
- Platform-as-a-Service (PaaS) offers high-level services for developers
  - Runtime environments: “Here is my Web application, run it for me”
  - Big data frameworks: “parallelize this program for me”
  - Databases: “I need a SQL database”
  - Development tools: “Give me a git repository”
- Software-as-a-Service (SaaS) offers high-level services for end users
  - Mail: Gmail
  - Office applications: Google docs
  - Enterprise applications: human resource applications, finance…
  - Payment services: Paypal
  - Data management: Dropbox, iCloud
  - Music on demand: iTunes, Spotify

(Unmodified) enterprise applications



Not extremely exciting. . . ☹

(Unmodified) enterprise applications (bis)



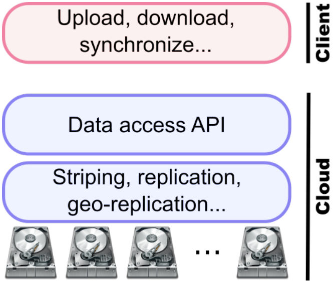
Daniel Stori {turnoff.us}
Thanks to Michael Tharrington

Data storage services

Dropbox

iCloud

Amazon S3

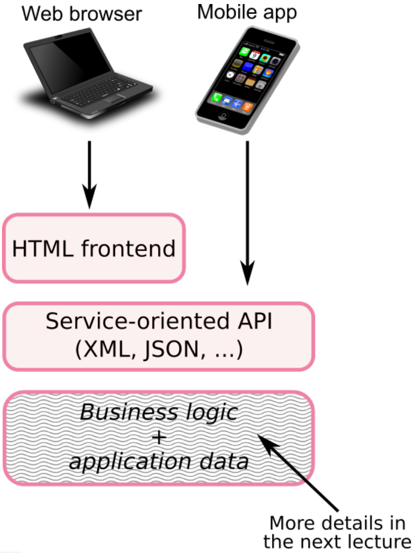


Upload, download, synchronize...

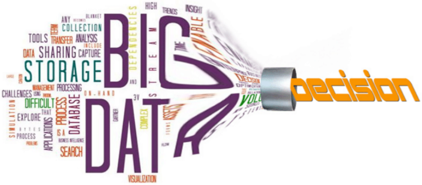
Data access API

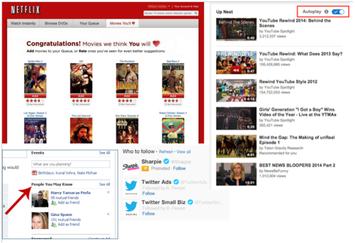
Striping, replication, geo-replication...

Web/mobile applications



Data analytics




NETFLIX

Watch Instantly · Browse DVDs · Your Queue · Movies You'll ♥

Congratulations! Movies we think You will ♥

Add movies to your Queue, or Rate ones you've seen for even better suggestions.

Events · See All

People You May Know · See All

Harry Tamacas Perla · 85 mutual friends · Add as friend

Gina Spano · 101 mutual friends · Add as friend

Who to follow · Refresh · View all

Sharpie @Sharpie · Promoted · Follow

Twitter Ads @TwitterAds · Follow

Twitter Small Biz @TwitterSmallBiz · Follow

Up Next · Autoplay

YouTube Rewind 2014: Behind the Scenes · by YouTube Spotlight · 3,212,337 views

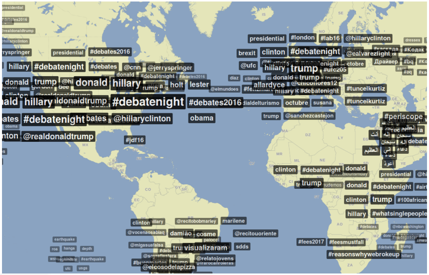
YouTube Rewind: What Does 2013 Say? · by YouTube Spotlight · 106,428,985 views

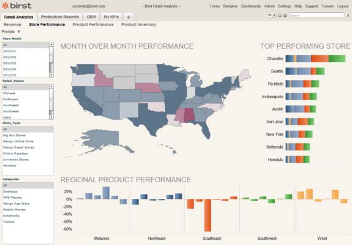
Rewind YouTube Style 2012 · by YouTube Spotlight · 154,722,053 views

Girls' Generation "I Got a Boy" Wins Video of the Year - Live at the YTMAs · by YouTube Spotlight · 350,421 views

Mind the Gap: The Making of unReal Episode 1 · by Teton Gravity Research · Recommended for you

BEST NEWS BLOOPERS 2014 Part 2 · by NewsBeFunny · 1,010,904 views

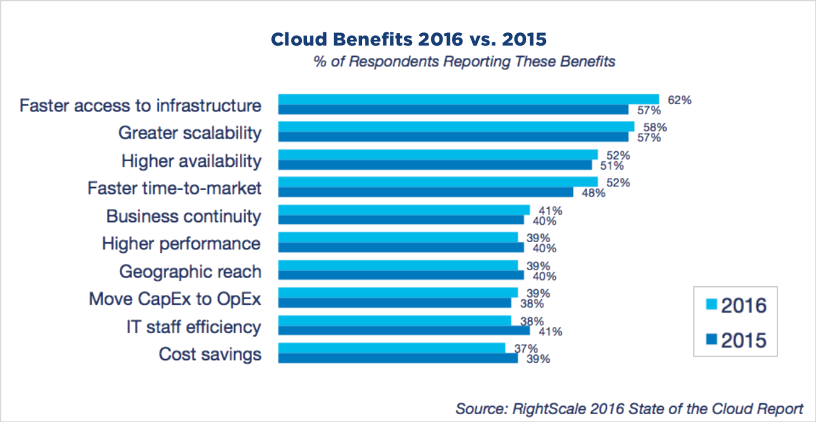



birst

Retail Analytics · Production Reports · OEM · My KPIs

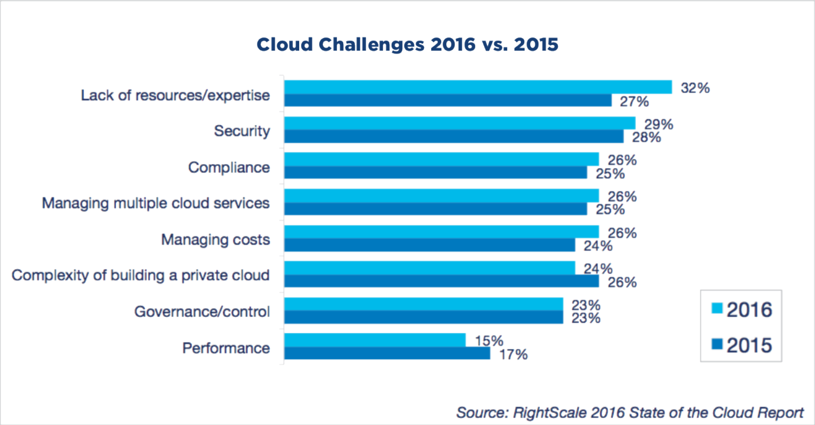
Revenue · Store Performance · Product Performance · Product Inventory

Cloud benefits



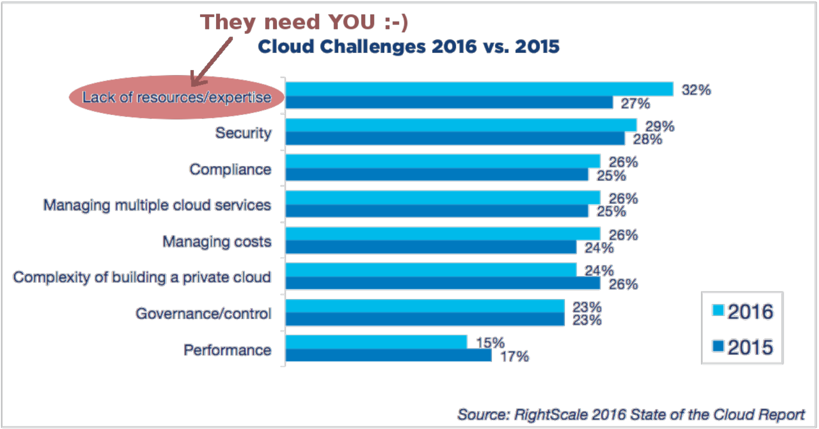
Source: RightScale 2016 State of the Cloud Report

Cloud challenges



Source: RightScale 2016 State of the Cloud Report

Cloud challenges



Source: RightScale 2016 State of the Cloud Report