From Data Centers to Fog Computing: The Evaporating Cloud

Guillaume Pierre

RESCOM 2017 summer school
June 20th 2017

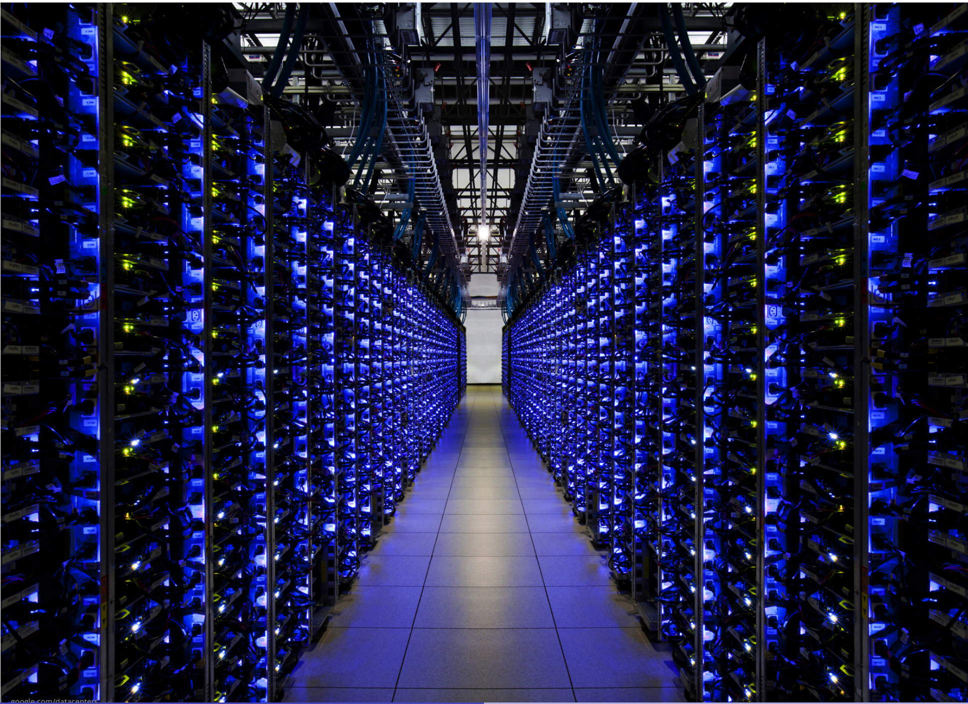

google.com/datacenters

Table of Contents

1 Cloud Data Centers

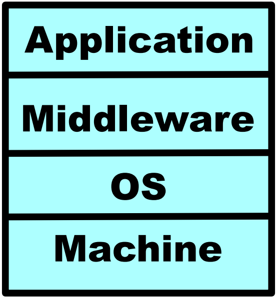
2 Mobile computing

3 Edge Computing

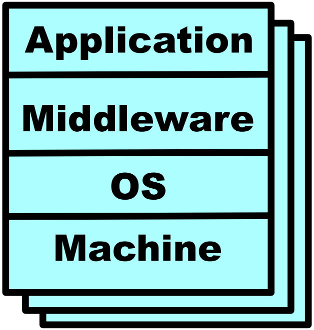
4 Fog Computing

5 Using a Raspberry Pi as a cloud node

6 Conclusion

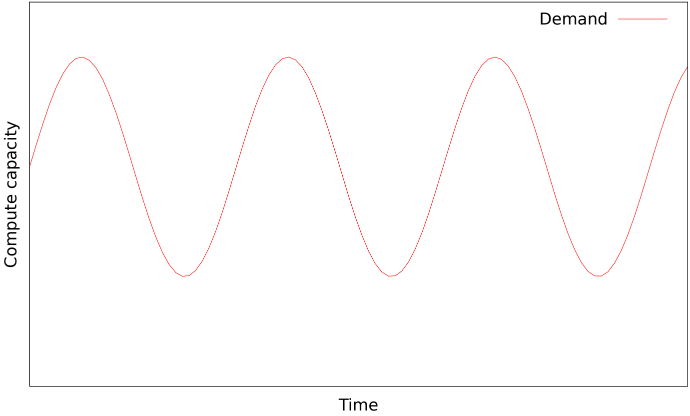


Traditional architecture

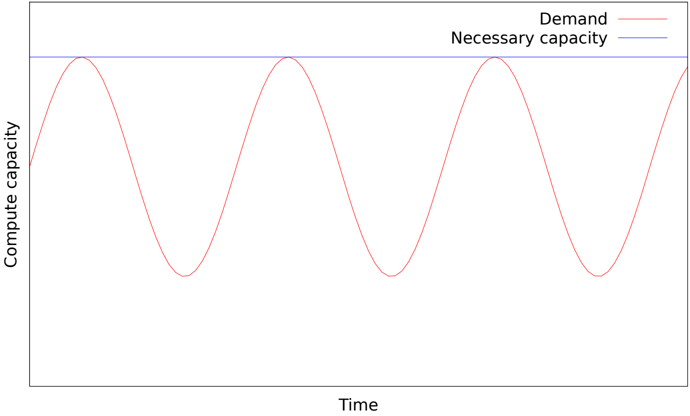


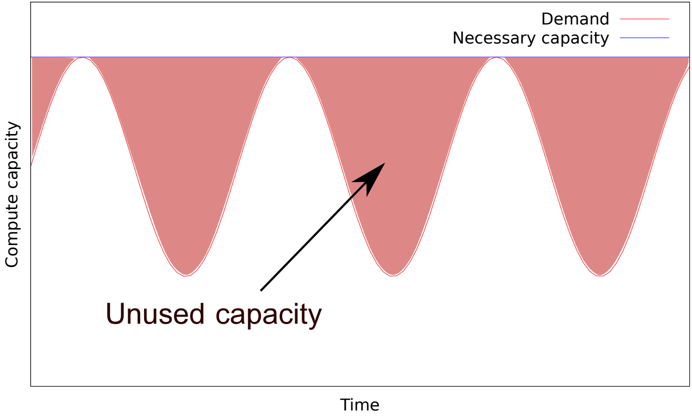
Traditional architecture

The varying capacity problem

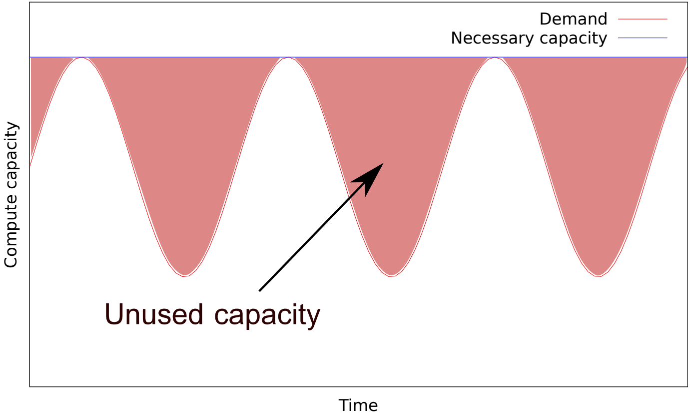


The varying capacity problem




Demand
Necessary capacity
Compute capacity
Unused capacity
Time

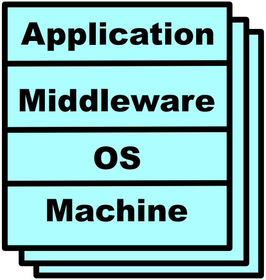
The varying capacity problem



What if demand increases beyond the capacity?

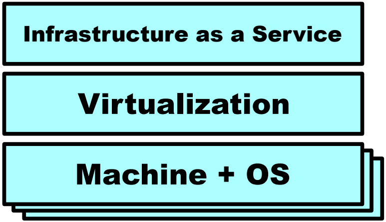
Cloud Computing

- Difficult to vary capacity!
- Manual resource management

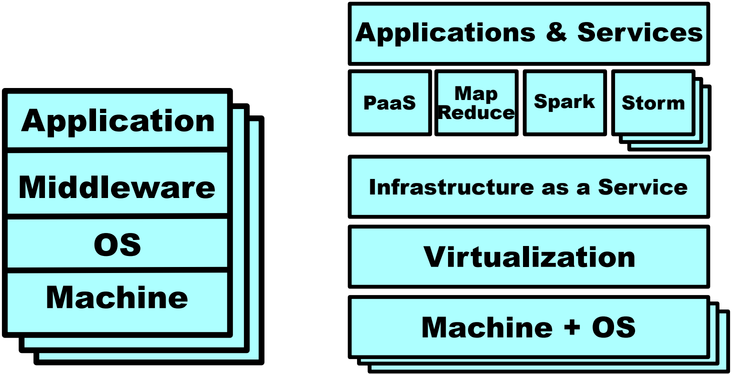


Traditional architecture

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



Cloud architecture



Traditional architecture

Cloud architecture

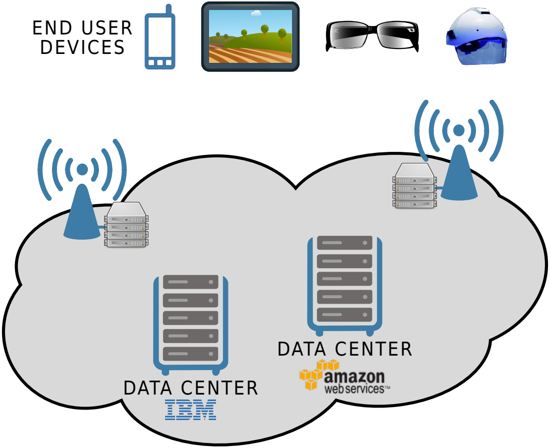
Table of Contents

1. Cloud Data Centers
2. Mobile computing
3. Edge Computing
4. Fog Computing
5. Using a Raspberry Pi as a cloud node
6. Conclusion

Since 2016: **mobile network traffic > fixed traffic**

END USER
DEVICES

Since 2016: **mobile network traffic > fixed traffic**

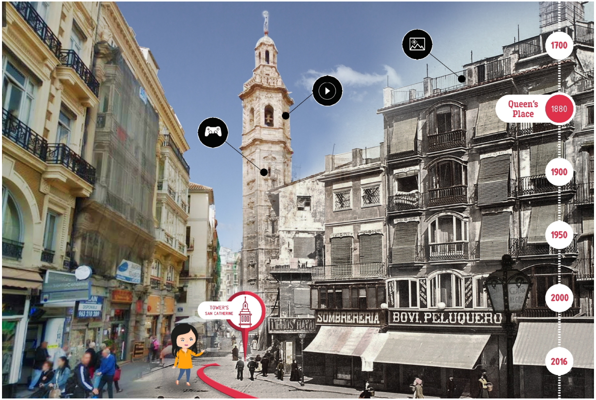


(example application)

New types of mobile applications

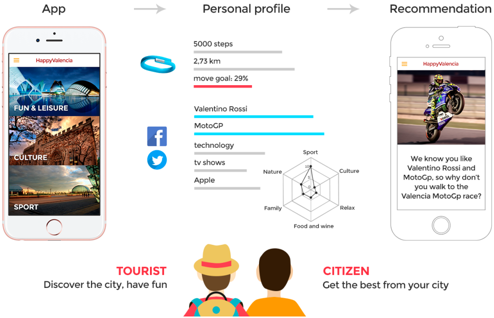
Interactive applications require ultra-low network latencies

- E.g., augmented reality require end-to-end delays under 20 ms
- But latencies to the closest data center are 20-30 ms using wired networks, up to 50-150 ms on 4G mobile networks!!!



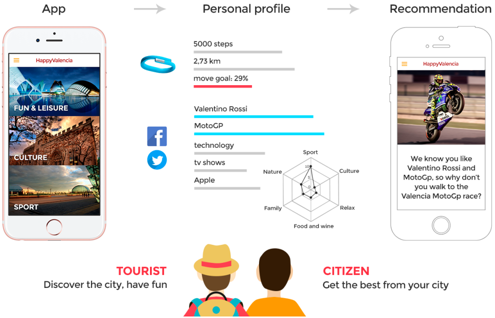
Throughput-oriented applications require local computations

- E.g., distributed videosurveillance is relevant only close to the cameras
- Why waste long-distance network resources?
- Fact: IoT-generated traffic grows faster than the Internet backbone capacity



Throughput-oriented applications require local computations

- E.g., distributed videosurveillance is relevant only close to the cameras
- Why waste long-distance network resources?
- Fact: IoT-generated traffic grows faster than the Internet backbone capacity



Question: **who owns computing resources located closest to the mobile end users?**

Table of Contents

1. Cloud Data Centers
2. Mobile computing
3. Edge Computing
4. Fog Computing
5. Using a Raspberry Pi as a cloud node
6. Conclusion

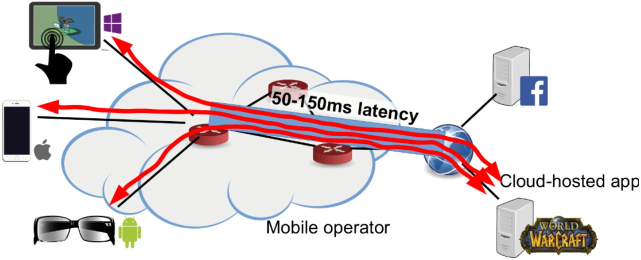
Mobile network operator's revenues

- **Where does mobile network operator's revenues come from?**
  - 1990's: Voice (not any more)
  - 2000's: Text/SMS (not any more)
  - 2010's: Data (not for very long. . .)
  - What's next?

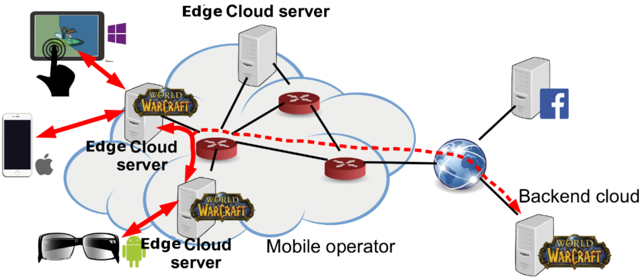
Mobile network operator's revenues

- **Where does mobile network operator's revenues come from?**
  - 1990's: Voice (not any more)
  - 2000's: Text/SMS (not any more)
  - 2010's: Data (not for very long. . .)
  - What's next? **Services!**
- Let's steal part of the cloud computing business. . .
  - No cloud data center can be closer to the users than us! 🙂

Before:



After:



Standardization efforts

- European Telecommunications Standards Institute: ~~Mobile~~ Multi-Access Edge Computing
  - Fujitsu, Hewlett-Packard, Huawei, Intel, Juniper, Motorola, NEC,Nokia, Orange, Samsung, Sony, Vodafone, ...
  - Focus: integration within mobile phone networks

Standardization efforts

- European Telecommunications Standards Institute:
  ~~Mobile~~ Multi-Access Edge Computing
  - Fujitsu, Hewlett-Packard, Huawei, Intel, Juniper, Motorola, NEC,Nokia, Orange, Samsung, Sony, Vodafone, ...
  - Focus: integration within mobile phone networks
- Open Edge Computing Initiative
  - Focus: opennes
  - Intel, Huawei, Vodafone, Carnegie-Mellon university, ...

Standardization efforts

- European Telecommunications Standards Institute:
  ~~Mobile~~ Multi-Access Edge Computing
  - Fujitsu, Hewlett-Packard, Huawei, Intel, Juniper, Motorola, NEC,Nokia, Orange, Samsung, Sony, Vodafone, . . .
  - Focus: integration within mobile phone networks
- Open Edge Computing Initiative
  - Focus: opennes
  - Intel, Huawei, Vodafone, Carnegie-Mellon university, . . .
- Open Fog Consortium
  - ARM, Cisco, Dell, Intel, Microsoft, . . .

Standardization efforts

- European Telecommunications Standards Institute:
  ~~Mobile~~ Multi-Access Edge Computing
  - Fujitsu, Hewlett-Packard, Huawei, Intel, Juniper, Motorola, NEC,Nokia, Orange, Samsung, Sony, Vodafone, ...
  - Focus: integration within mobile phone networks
- Open Edge Computing Initiative
  - Focus: opennes
  - Intel, Huawei, Vodafone, Carnegie-Mellon university, ...
- Open Fog Consortium
  - ARM, Cisco, Dell, Intel, Microsoft, ...
- Fog/Edge/Massively Distributed Clouds WG at OpenStack

Table of Contents

1. Cloud Data Centers
2. Mobile computing
3. Edge Computing
4. Fog Computing
5. Using a Raspberry Pi as a cloud node
6. Conclusion

Fog computing = cloud + edge + end-user devices
as a single execution platform

- Low latency
- Localized traffic (privacy, less global traffic...)

Fog > Cloud

- We need cloud servers close to the users, but the users are everywhere (and they are mobile)
  - Let's place one cloud server within 1-hop WiFi range of any end-user device
  - ⇒ Fog computing resources will need to be distributed in thousands of locations

Fog > Cloud

- We need cloud servers close to the users, but the users are everywhere (and they are mobile)
  - Let's place one cloud server within 1-hop WiFi range of any end-user device

  ⇒ Fog computing resources will need to be distributed in thousands of locations

- Fog nodes will be connected with commodity networks
  - Forget your multi-Gbps data center networks!

Fog > Cloud

- We need cloud servers close to the users, but the users are everywhere (and they are mobile)
  - Let's place one cloud server within 1-hop WiFi range of any end-user device
  - ⇒ Fog computing resources will need to be distributed in thousands of locations
- Fog nodes will be connected with commodity networks
  - Forget your multi-Gbps data center networks!
- Fog nodes must be small, cheap, easy to deploy and to replace
  - We started using single-board computers as our (powerful!) cloud servers

Fog > Cloud

- We need cloud servers close to the users, but the users are everywhere (and they are mobile)
  - Let's place one cloud server within 1-hop WiFi range of any end-user device
  - ⇒ Fog computing resources will need to be distributed in thousands of locations
- Fog nodes will be connected with commodity networks
  - Forget your multi-Gbps data center networks!
- Fog nodes must be small, cheap, easy to deploy and to replace
  - We started using single-board computers as our (powerful!) cloud servers

Raspberry PIs are more powerful than you may think

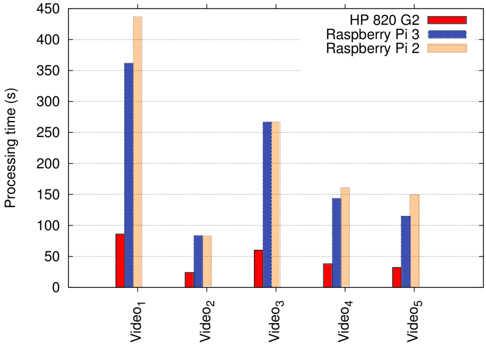
	RPi3	Pine A64+	HP 820 G2
CPU (s)	46.5	4.1	2.8
Memory (MB/s)	933	1098	5658
Network (Mb/s)	94.2	922	935
Storage (MB/s)	2.53	2.42	23.9
Power when idle (W)	2	2	15
Power under load (W)	4.4	4.1	24.5
Price	∼ 92 €	∼ 74 €	∼ 1600 €

We chose RPi3 as our base platform for the time being

- Mostly because of easy purchase options and community support. . .

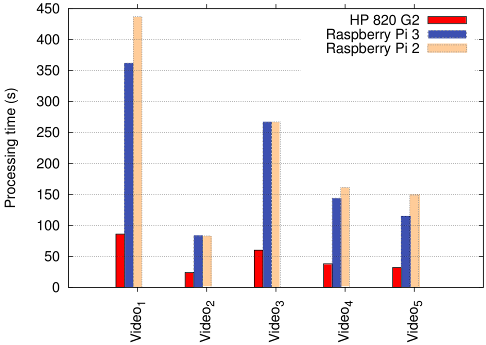
What about a real application?

- Input: live video stream
- Processing: face recognition algorithm running inside Apache Flink



What about a real application?

- Input: live video stream
- Processing: face recognition algorithm running inside Apache Flink



- The RPIs are "only" 3-5 times slower than my laptop
- But they are 17 times cheaper
- If my applications scale horizontally I can use as many RPIs as necessary

Table of Contents

1 Cloud Data Centers

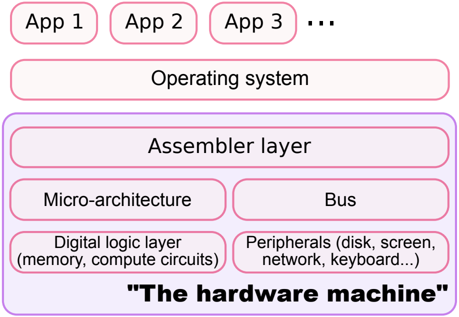
2 Mobile computing

3 Edge Computing

4 Fog Computing

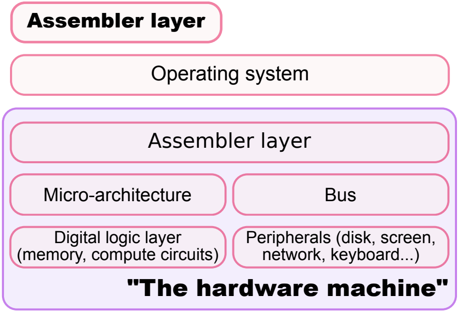
5 Using a Raspberry Pi as a cloud node

6 Conclusion

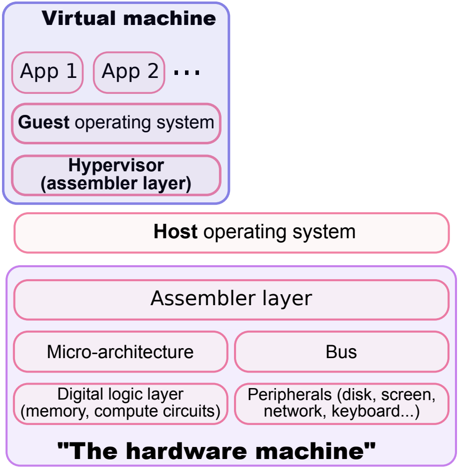


Traditional machine architecture:

- Applications
- Operating system
- Hardware



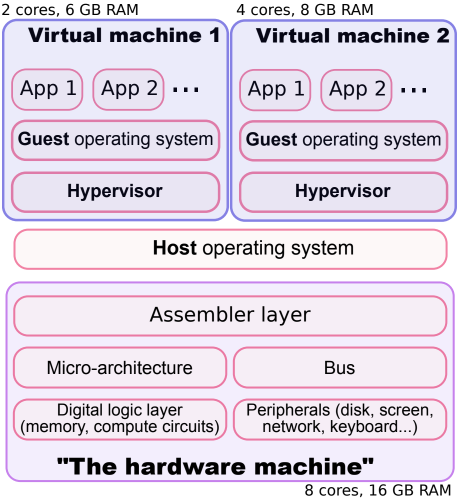
Let's create a "special application" which behaves exactly the same as the assembler layer...



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

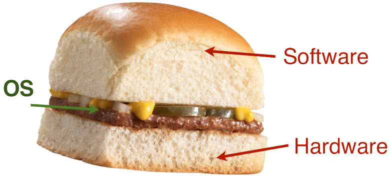
... and any application over the guest operating system

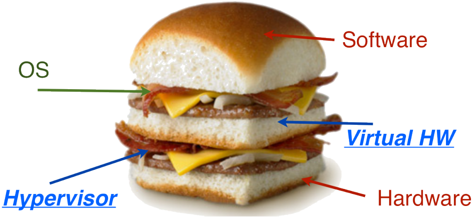
$\Rightarrow$ We have a virtual machine



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. Slow!
- Full virtualization exploits special features of modern CPUs, does not require special support in the guest OS. Faster!

Virtualization technologies

Virtualization technologies are now totally mainstream:

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

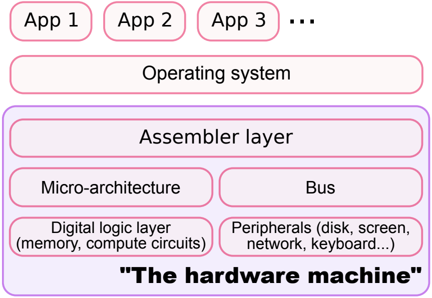
Paravirtualization vs. full virtualization:

- Paravirtualization works on any hardware platform but it requires special support in the guest OS. Slow!
- Full virtualization exploits special features of modern CPUs, does not require special support in the guest OS. Faster!

Virtual machines are waaaaaayyyyyy too heavyweight for a RPI

- 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...

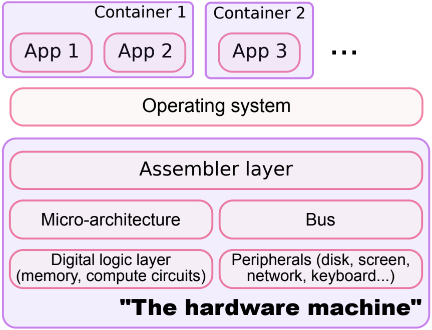
Containers



Traditional machine architecture:

- Applications
- Operating system
- Hardware

Containers



Let's create groups of processes which belong together:

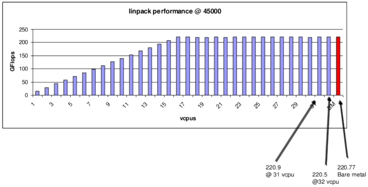
- Process groups are totally isolated from each other
- Each process group belongs to a single user
- Each process group has its own hardware resource limits (CPU, RAM, ...)
- Each process group has its own network access policy
- Etc.

$\Rightarrow$ We have containers

Container technologies

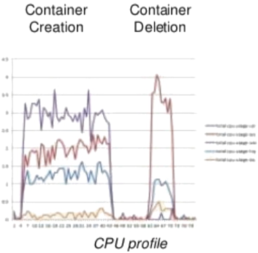
- Containers are an operating system feature
  - No need for special CPU support
  - Fully supported in Linux, Windows, ...
  - We usually add an extra software layer to simplify management: Docker
- Containers are less customizable than VMs
  - Container owners cannot choose their OS
  - But was that 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
  - We can easily run hundreds of containers on a mid-sized machine

Performance



Performance is extremely close to bare-metal

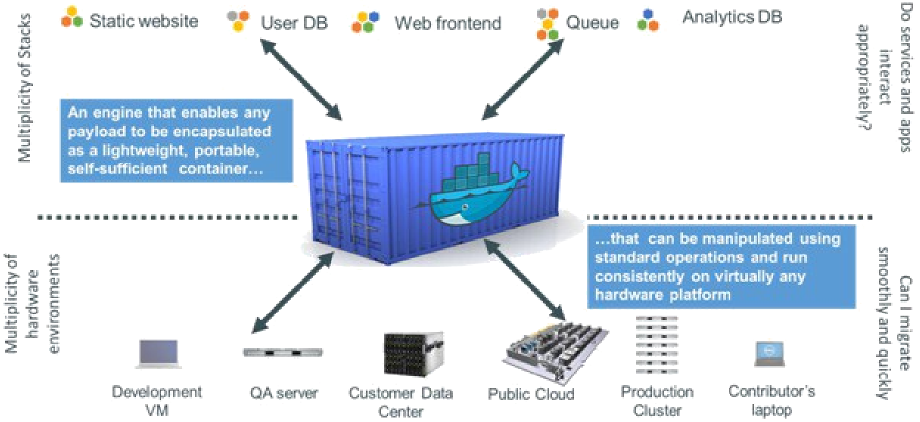
Agility



- Starting 150 containers w/ Apache
  - Total time: 36 s (240 ms/container)
  - Consumes about 2% of CPU
  - Memory usage: ∼10 MB/container
- Stopping 150 containers:
  - Total time: 9 seconds

http://www.slideshare.net/BodenRussell/realizing-linux-containerslxc

Docker is a shipping container system for code

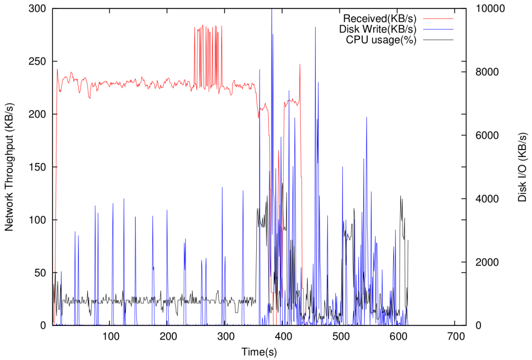


https://impythonist.wordpress.com/2015/06/21/

docker-the-future-of-virtualization-for-your-django-web-development/

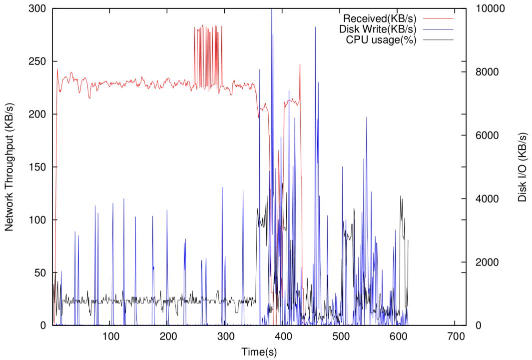
Container deployment in a RPI

- Let's deploy a very simple Docker container on the RPI3
  - Standard ubuntu container (∼45 MB) + one extra 51-MB layer



Container deployment in a RPI

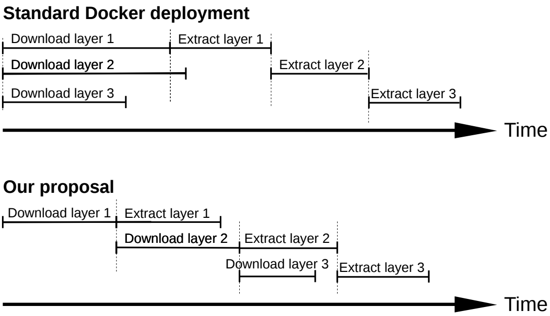
- Let's deploy a very simple Docker container on the RPI3
  - Standard ubuntu container (∼45 MB) + one extra 51-MB layer



- Total deployment time: **over 10 minutes!!!**
- Docker downloads all layers then decompresses then writes to disk

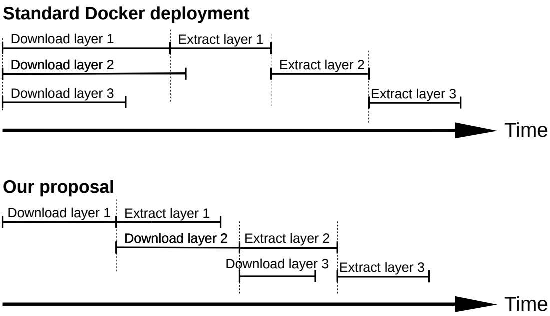
Very preliminary results

- **Idea:** let's parallelize the deployment process to use the bandwidth and disk I/O simultaneously



Very preliminary results

- **Idea:** let's parallelize the deployment process to use the bandwidth and disk I/O simultaneously



- Current improvement ∼10%
- Greater improvements over slow networks
- We still have lots of performance improvement potential. . .

This is only the beginning

Lots of remaining research problems:

- How do we develop a fog computing application?
  - One VM per user: easy but (mostly) useless
  - Multiuser applications: much harder!!! We need fog-aware middlewares
- How do we manage applications?
  - How do we express the resource/location/performance requirements for each application component?
  - When should we migrate/replicate/delete components to maintain performance?
  - Etc.
- How do we manage resources?
  - Assign specific resources to each container (and manage conflicts)
  - Monitoring / anomaly detection
  - System upgrades
  - Etc.

Table of Contents

1. Cloud Data Centers
2. Mobile computing
3. Edge Computing
4. Fog Computing
5. Using a Raspberry Pi as a cloud node
6. Conclusion

Conclusion

- Cloud data centers are very powerful and flexible
  - But not all applications can use them (latency, traffic locality)
- If we evaporate a cloud, then we get a fog
  - Extremely distributed infrastructure: there must be a server node close to every end user
    - ★ Server nodes must be small, cheap, easy to add and replace
    - ★ Server nodes are very far from each other
- This is only the beginning
  - No satisfactory edge/fog platforms are available today (we are not even close)
  - There remains thousands of potential PhD research topics in this domain

Shameless announcement

A European H2020 project named FogGuru will start soon on similar issues: France (PI), Germany, Italy, Sweden, Spain

- Application and resource management in scalable fog platforms
- Stream processing middleware systems for fog applications
- Blueprints for innovative fog applications
- Full-scale experimental deployment in Valencia (Spain)

We are looking for:

- 8 ambitious and talented PhD students
- 1 project manager / postdoc researcher