The Raspberry Pi Cloud

Richard Cziva

13 August, 2014
Morgan Stanley, Glasgow

University of Glasgow

Agenda

- Introduction
- The Raspberry Pi project
- Cloud Computing and IaaS
- Building a Cloud from Raspberry Pi devices
- Future work, conclusion

Agenda

- *Introduction*
- The Raspberry Pi project
- Cloud Computing and IaaS
- Building a Cloud from Raspberry Pi devices
- Future work, conclusion

About me

- Worked at Morgan Stanley in Budapest
  - Enterprise Infrastructure (EI)
    - Virtualization Engineering (VirtEng)
  - Developed tools for the VMware infrastructure
    - Perl, Python
  - Worked on the OpenStack proof of concept evaluation

About me

- Started my PhD in Computing Science a year ago at the University of Glasgow
- Currently working as a Research Assistant
- Research interests:
  - Cloud Computing
  - Network-Aware Virtual Machine Management
  - Software Defined Networking in Cloud DCs

Agenda

- Introduction
    - *The Raspberry Pi project*
- Cloud Computing and IaaS
- Building a Cloud from Raspberry Pi devices
- Future work, conclusion

The Raspberry Pi

- Credit card sized, low cost computer
- Found by the Raspberry Pi Foundation
- Aiming student experiments with low investment
- The computer uses less power than a light bulb
- System on chip, a computer on a single low voltage chip

Background

- Research around the device started at 2006
- Computing Lab (CL) at University of Cambridge
- First model was based on Atmel ATmega644 microcontroller with USB and HDMI

Features

- Broadcom System on a Chip (SoC)
- x86 not used -> ARM architecture
  - Operating systems should be ARM compatible
    - Raspbian (Debian clone)
    - Fedora Pi
    - Arch Linux
- Boots from an SD card (class 10)

Features

- GPIO to interconnect with other circuit boards (sensors, LEDs, etc)
- Main programming language is Python, C, Perl
- Power supply 700mA at 5V
  - can operate from 4xAA batteries

Models

- Model A
  - 1 USB port and no Ethernet
- Model B
  - 256 MB RAM (rev1) / 512 MB RAM (rev2)
  - 2 USB ports and 10/100 Ethernet
- Model B+

Specifications

- SoC BCM2835
  - CPU: ARM 700 MHz (armv6l)
  - GPU: Broadcom Videocore IV
  - RAM: 256 MB SDRAM

Implemented ideas

- Home automation systems
- Security webcam
- MP3 player
- Game server (Minecraft)
- Alarm system
- Brewery controller (beer or wine)

Example projects - BrewPi

Example projects - Supercomputer

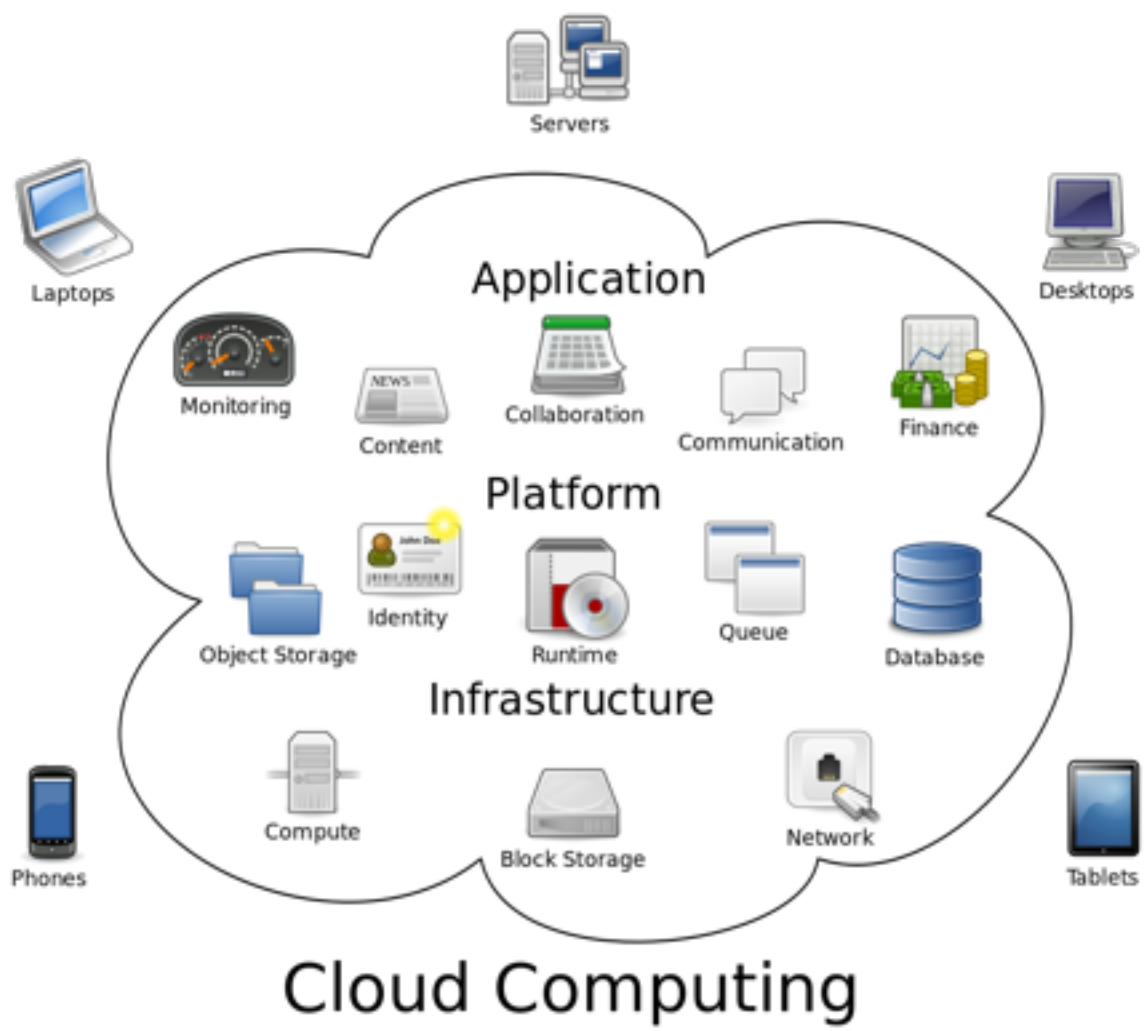
Implemented ideas - WiFi Internet Radio

Agenda

- Introduction
- The Raspberry Pi project
    - *Cloud Computing and IaaS*
- Building a Cloud from Raspberry Pi devices
- Future work, conclusion

Cloud Computing

- IT services, delivered over the Internet



Cloud Clients
web browser, mobile app, thin client

↕

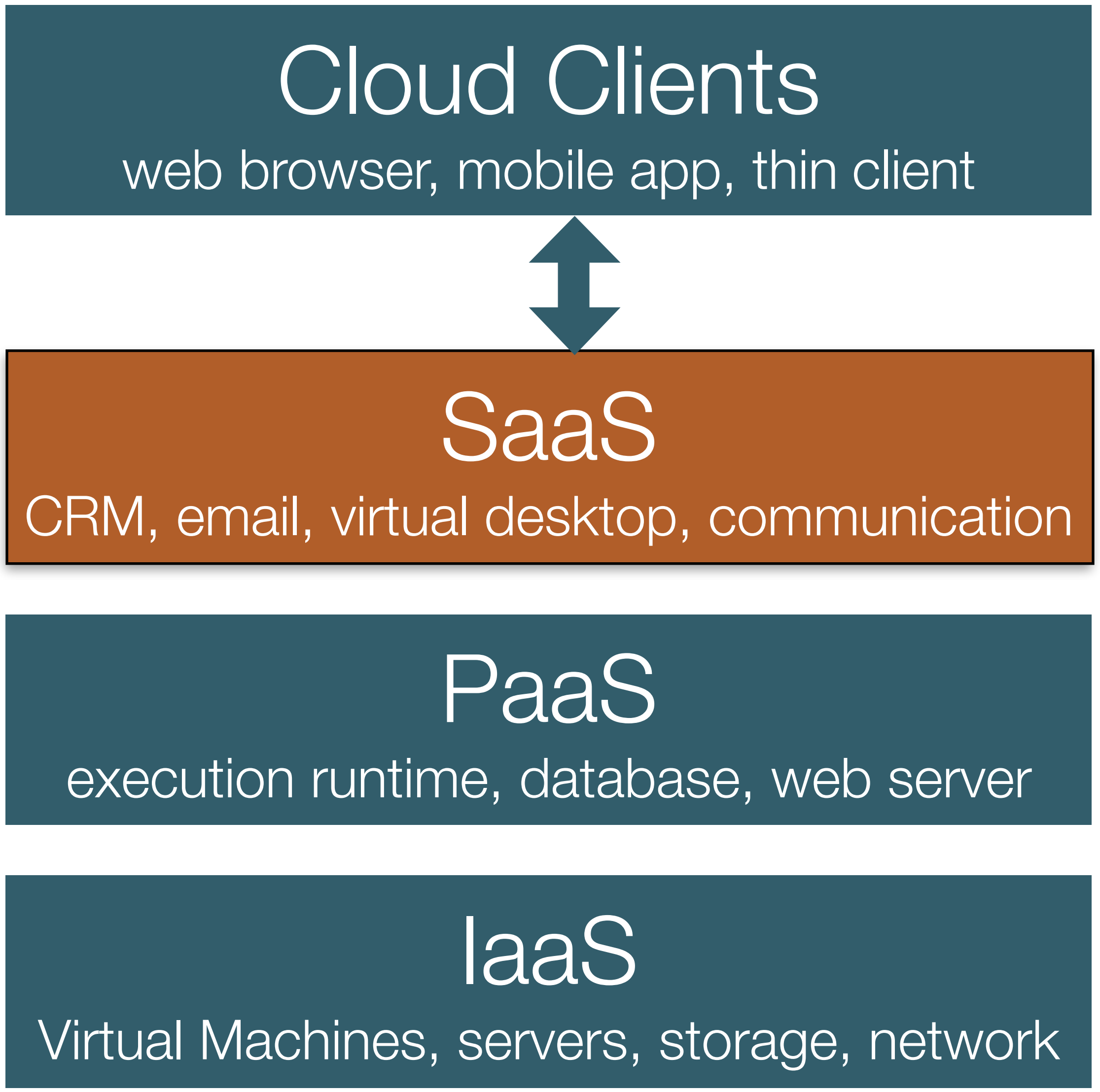
SaaS
CRM, email, virtual desktop, communication

PaaS
execution runtime, database, web server

IaaS
Virtual Machines, servers, storage, network

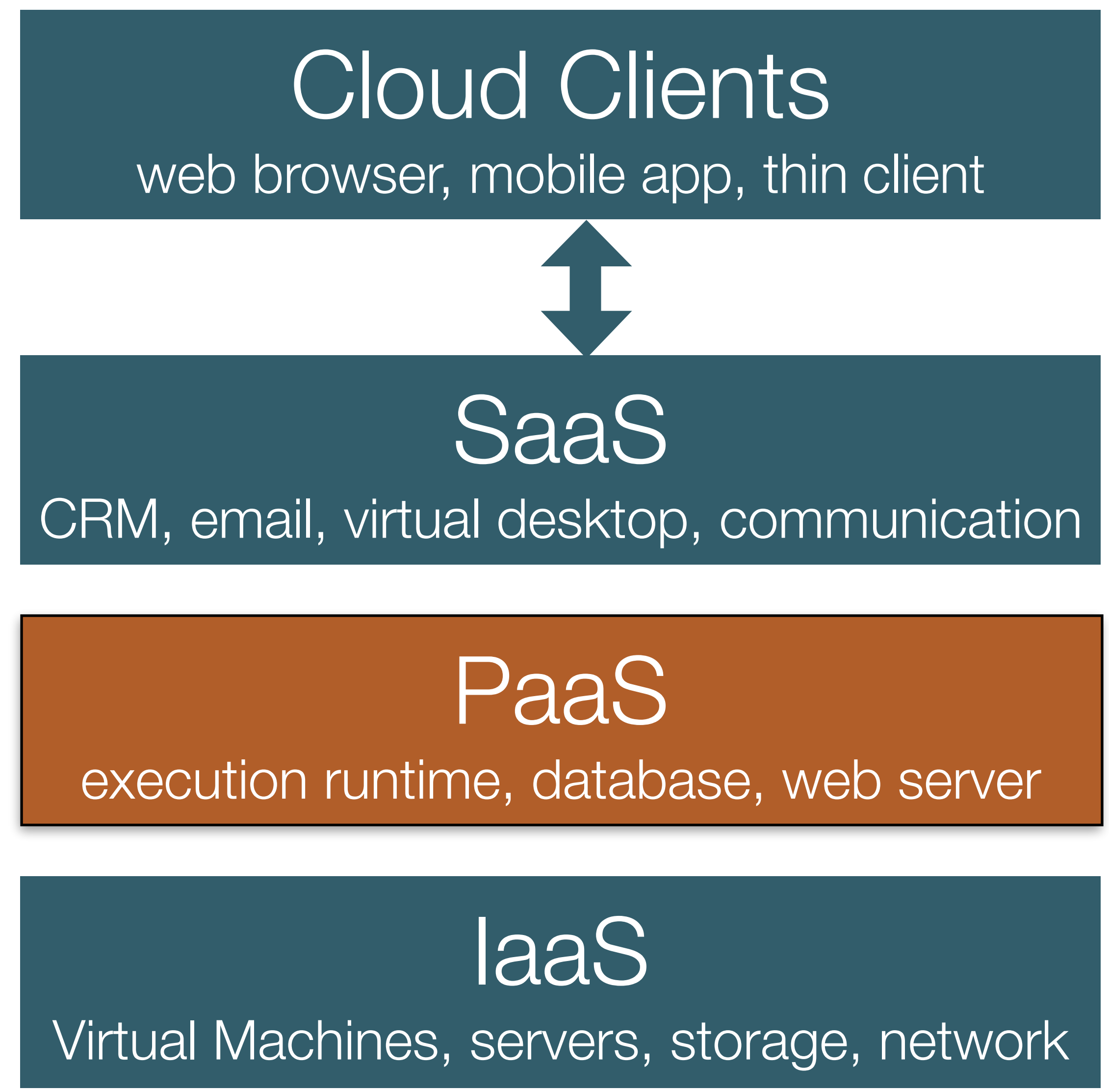
Software as a Service (SaaS)

- “On demand software”
- Examples:
  - Messaging, collaboration systems
  - ERP, CRM and HRM systems
- Priced after number of users



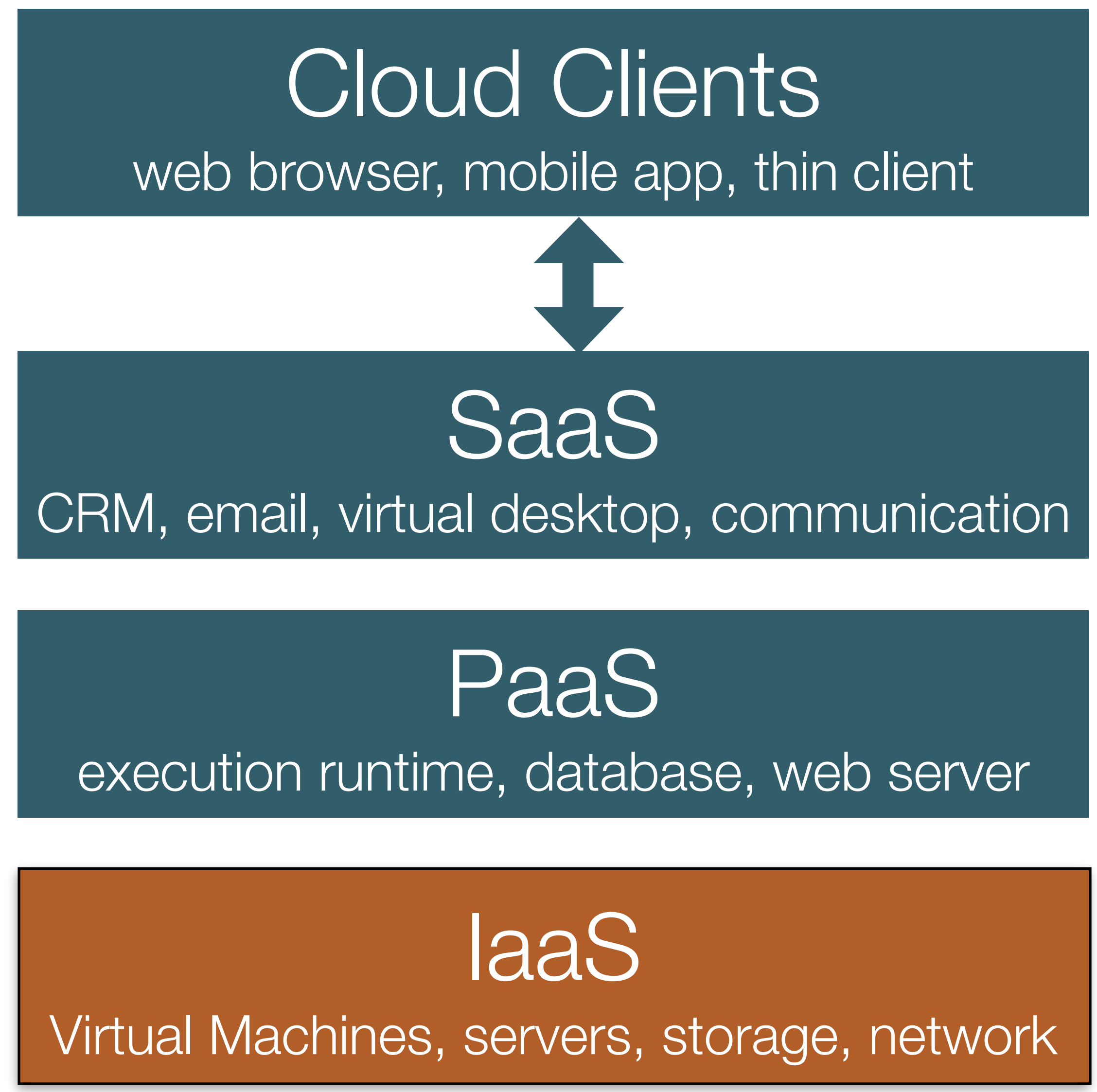
Platform as a Service (PaaS)

- A computing platform with a solution stack
- Examples:
  - Google App Engine
  - Heroku
  - Windows Azure Websites



Infrastructure as a Service (IaaS)

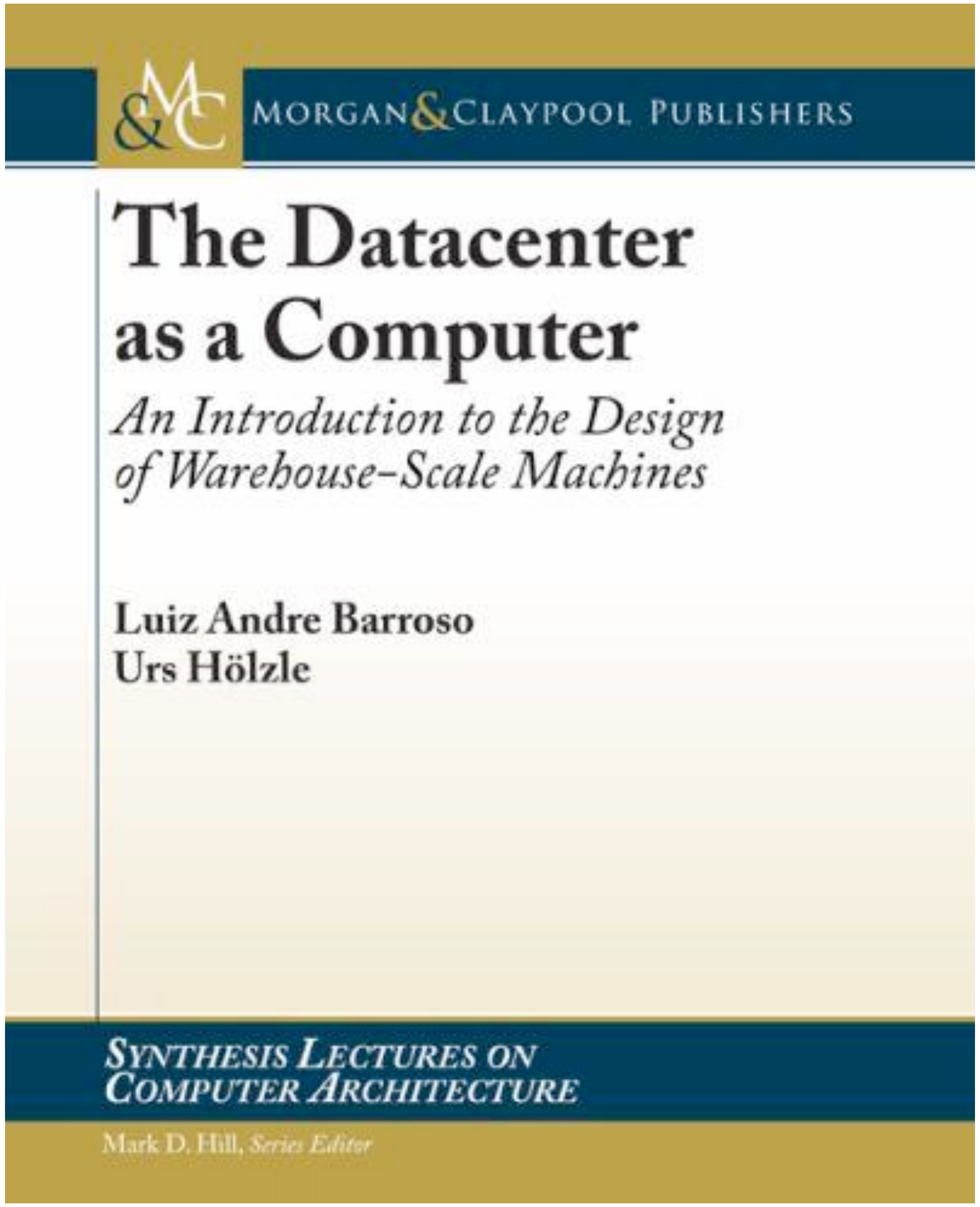
- Virtual Machines
- Priced after resources (CPU, memory, network, etc) used
- Examples:
  - Amazon EC2
  - Google Compute Service
  - VMware vCloud

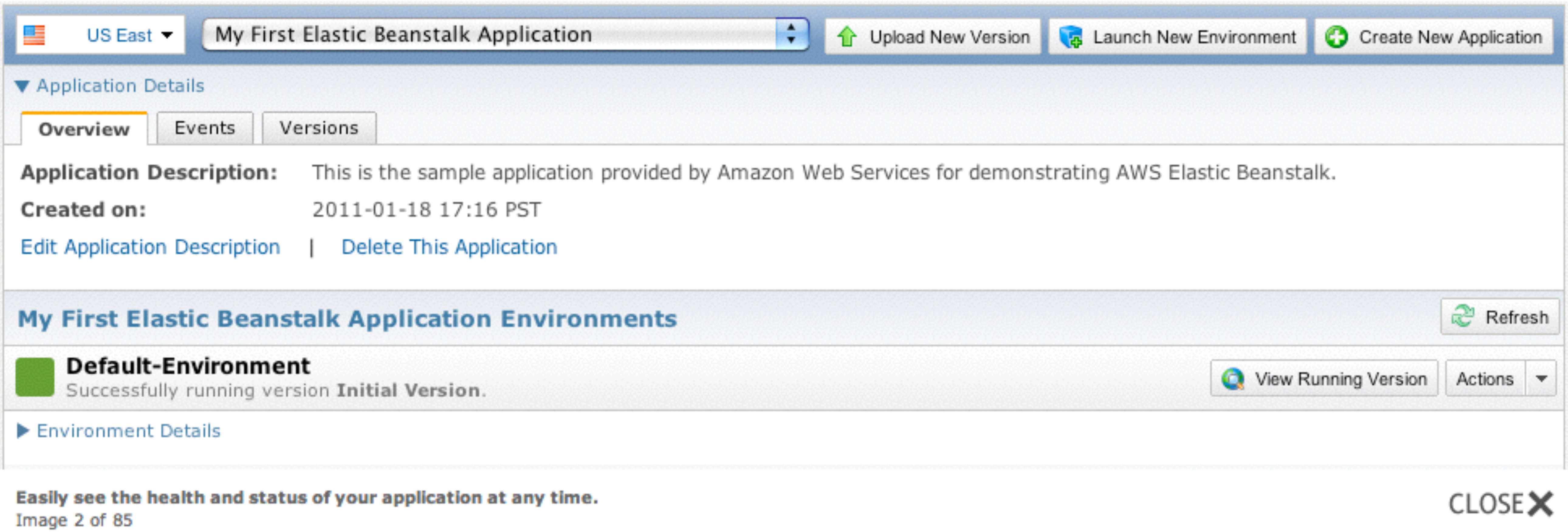


Agenda

- Introduction
- The Raspberry Pi project
- Cloud Computing and IaaS
    - *Building a Cloud from Raspberry Pi devices*
- Future work, conclusion

A new type of Software Engineering





Black Box Cloud Computing

Microsoft, Dublin. $500 million.

Facebook, N.C. $606 million.

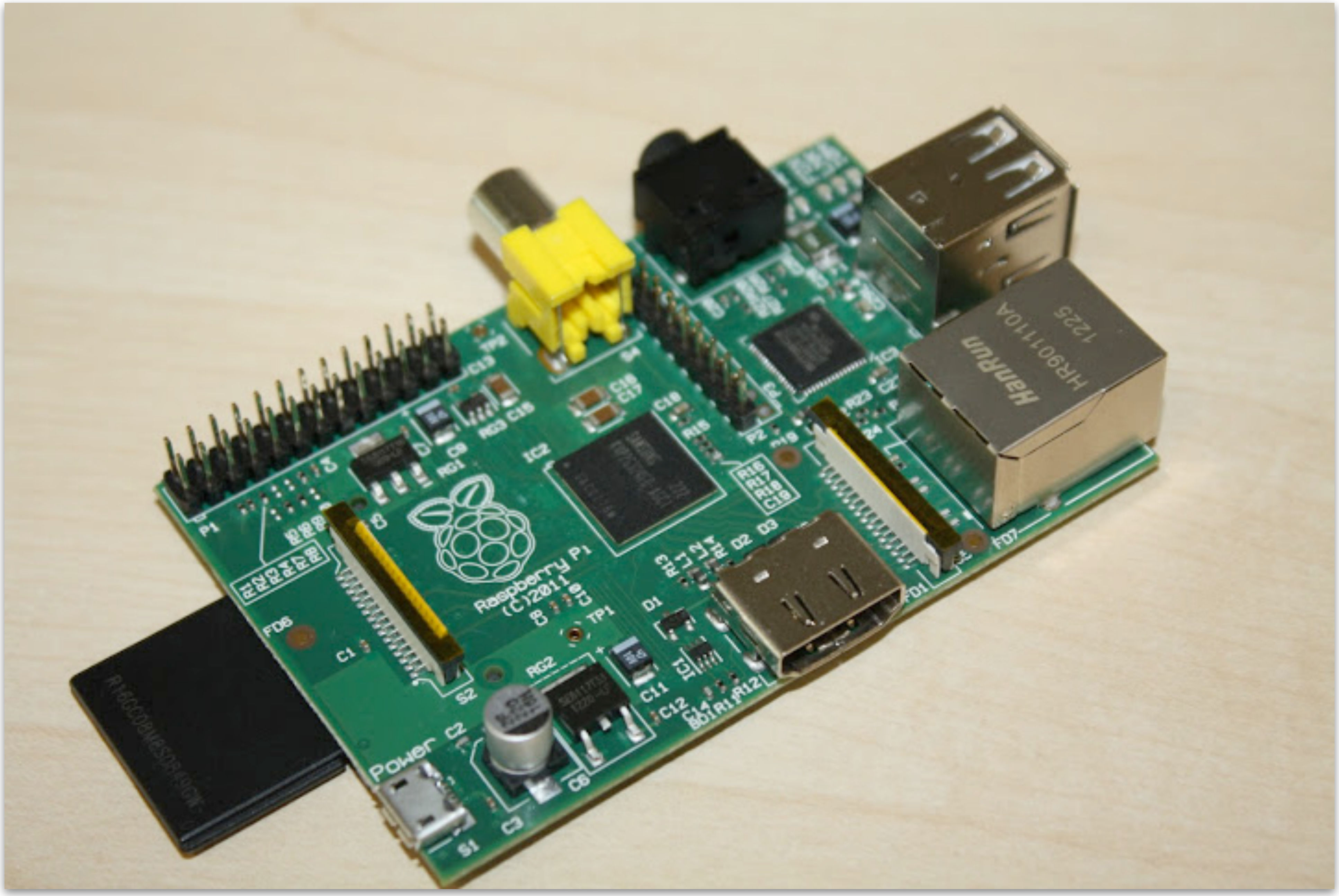
Apple, Maiden, DC. $1 billion.

Simulation

- Simulate a Cloud Datacentre
- e.g. NS/3 for networking, CloudSim, Mininet etc
- Fidelity? Convincing?
- Intangible; doesn't model everything; the details matter

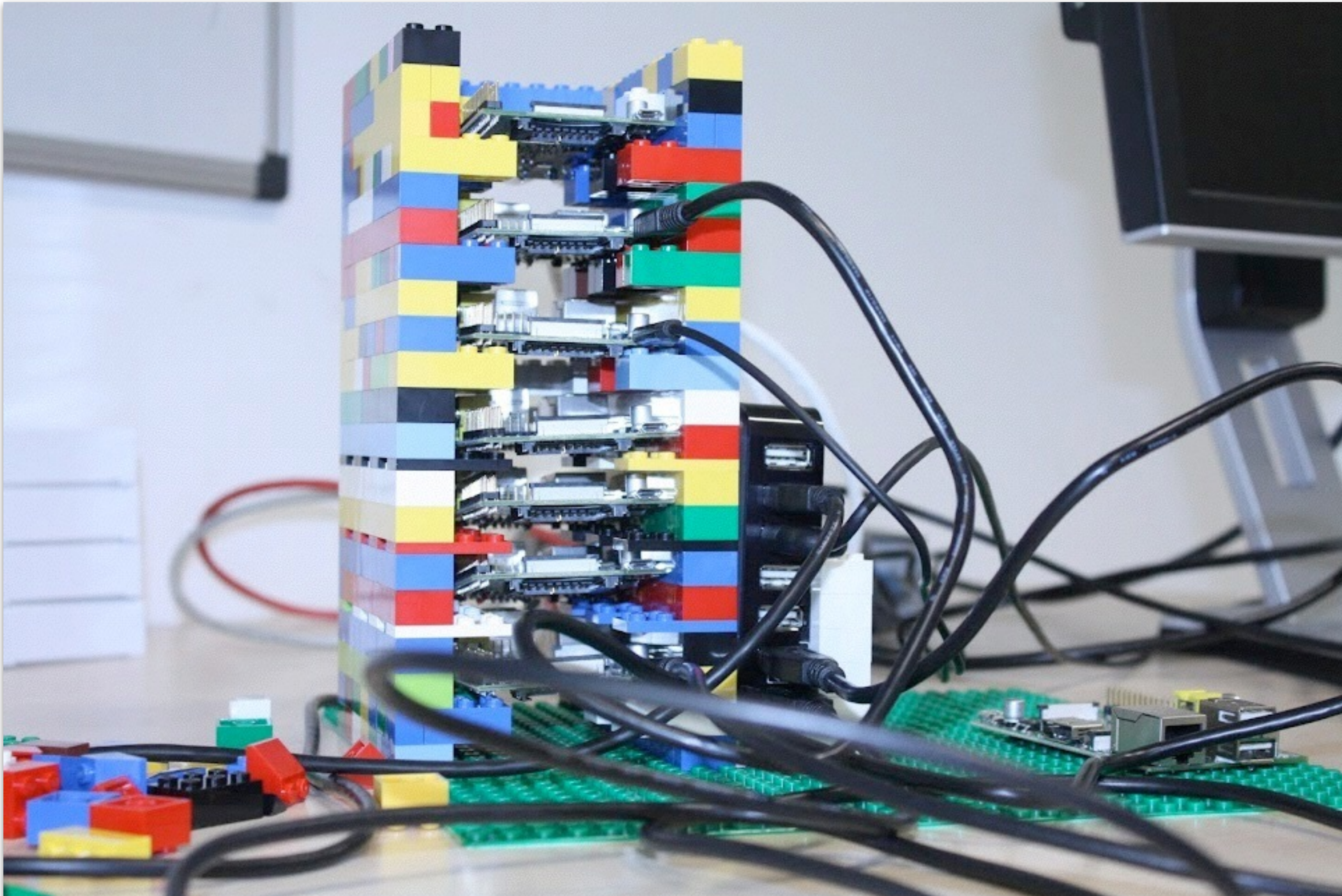
Build Something

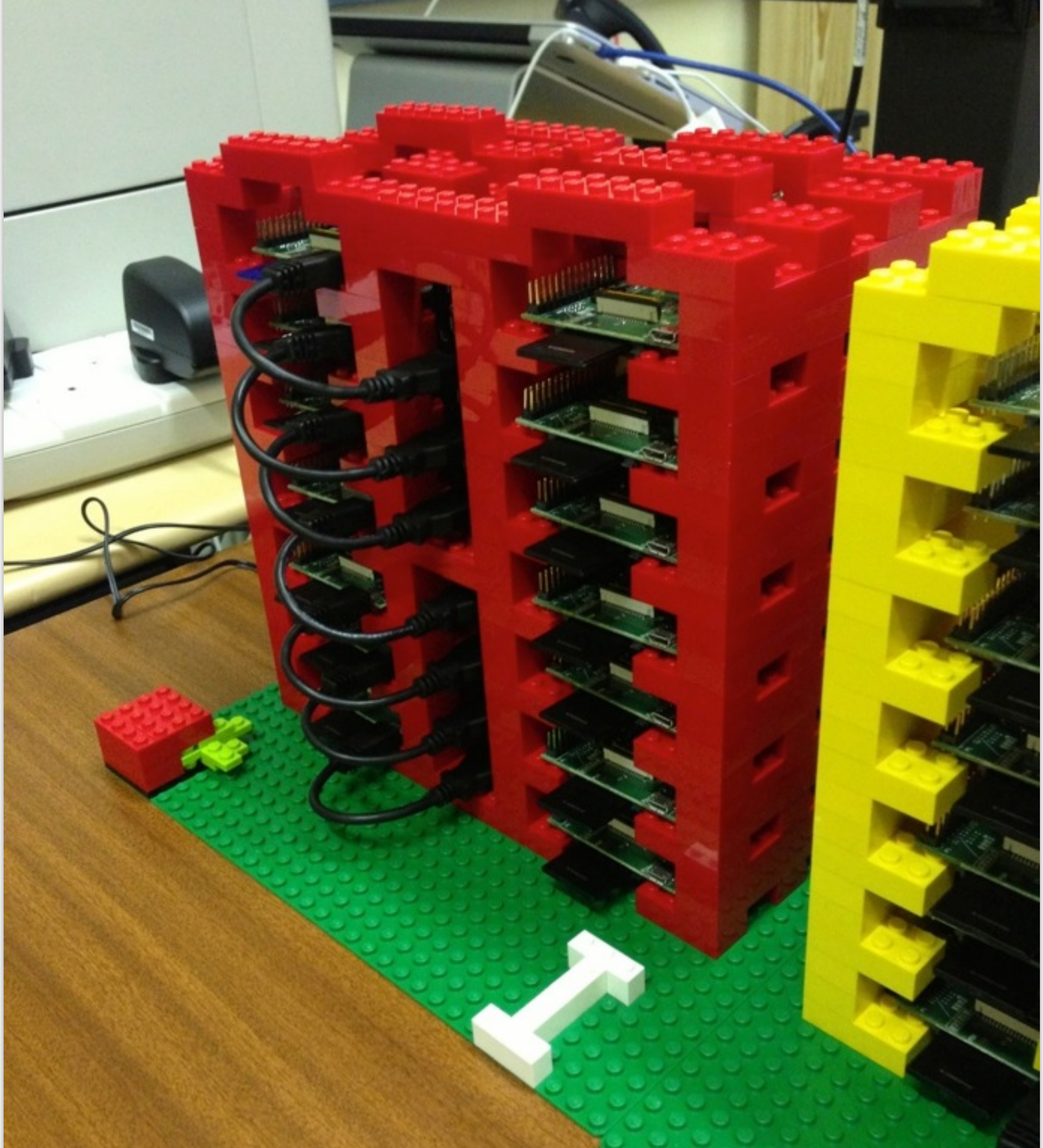
What About…
A Scale Model?



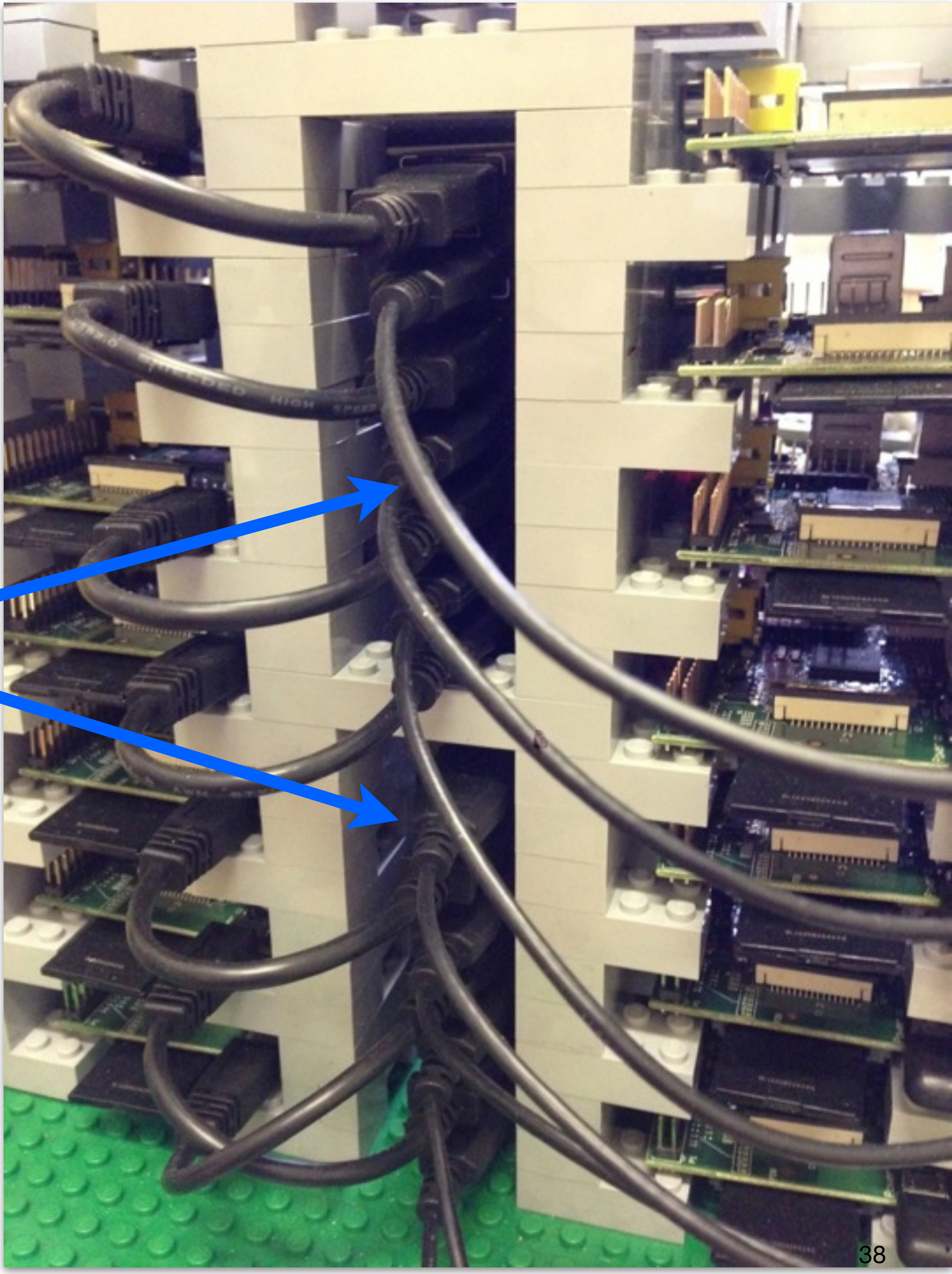
A Datacentre for £10k?

Designing Racks

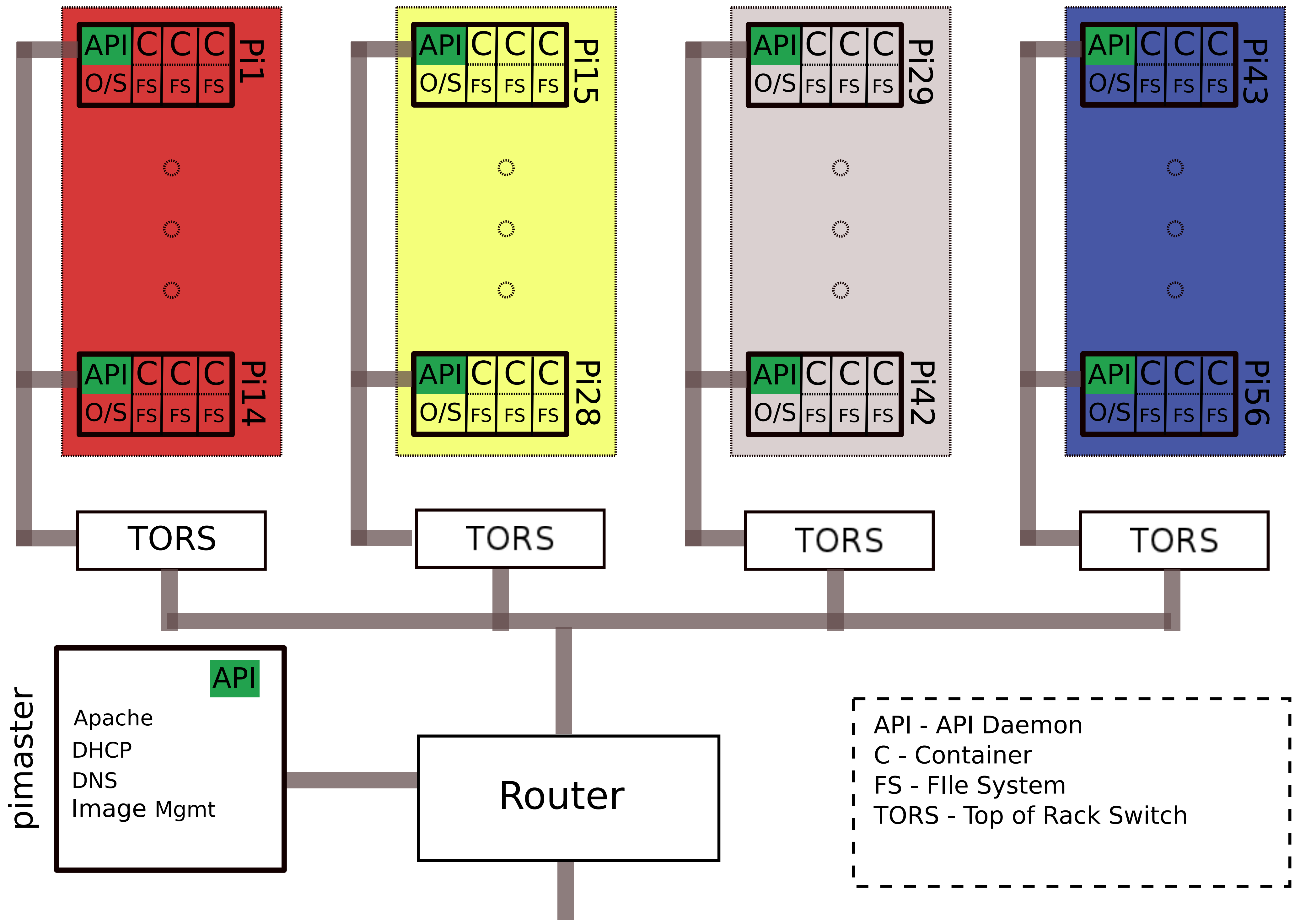


USB Hubs



What about the Software?

Software Architecture



Virtualisation is Heavy

Xen and the Art of Virtualization

Paul Barham*, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris,
Alex Ho, Rolf Neugebauer†, Ian Pratt, Andrew Warfield

University of Cambridge Computer Laboratory
15 JJ Thomson Avenue, Cambridge, UK, CB3 0FD
{firstname.lastname}@cl.cam.ac.uk

ABSTRACT

Numerous systems have been designed which use virtualization to subdivide the ample resources of a modern computer. Some require specialized hardware, or cannot support commodity operating systems. Some target 100% binary compatibility at the expense of performance. Others sacrifice security or functionality for speed. Few offer resource isolation or performance guarantees; most provide only best-effort provisioning, risking denial of service.

This paper presents Xen, an x86 virtual machine monitor which allows multiple commodity operating systems to share conventional hardware in a safe and resource managed fashion, but without sacrificing either performance or functionality. This is achieved by providing an idealized virtual machine abstraction to which operating systems such as Linux, BSD and Windows XP, can be *ported* with minimal effort.

Our design is targeted at hosting up to 100 virtual machine instances simultaneously on a modern server. The virtualization approach taken by Xen is extremely efficient: we allow operating systems such as Linux and Windows XP to be hosted simultaneously for a negligible performance overhead — at most a few percent compared with the unvirtualized case. We considerably outperform competing commercial and freely available solutions in a range of microbenchmarks and system-wide tests.

1. INTRODUCTION

Modern computers are sufficiently powerful to use virtualization to present the illusion of many smaller *virtual machines* (VMs), each running a separate operating system instance. This has led to a resurgence of interest in VM technology. In this paper we present Xen, a high performance resource-managed virtual machine monitor (VMM) which enables applications such as server consolidation [42, 8], co-located hosting facilities [14], distributed web services [43], secure computing platforms [12, 16] and application mobility [26, 37].

Successful partitioning of a machine to support the concurrent execution of multiple operating systems poses several challenges. Firstly, virtual machines must be isolated from one another: it is not acceptable for the execution of one to adversely affect the performance of another. This is particularly true when virtual machines are owned by mutually untrusting users. Secondly, it is necessary to support a variety of different operating systems to accommodate the heterogeneity of popular applications. Thirdly, the performance overhead introduced by virtualization should be small.

Xen hosts commodity operating systems, albeit with some source modifications. The prototype described and evaluated in this paper can support multiple concurrent instances of our XenoLinux guest operating system; each instance exports an application binary interface identical to a non-virtualized Linux 2.4. Our port of Windows

Virtualisation is Heavy

"dom0 uses at least 200MB"

Pi had 256MB, now 512MB

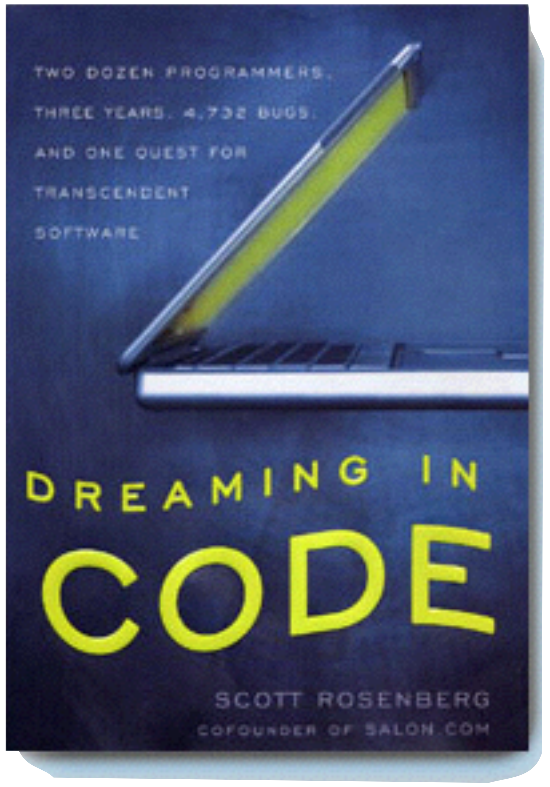
http://docs.vmd.citrix.com/XenServer/4.0.1/installation/apd.html

Containers are Light

- Using Linux Containers (LXC)
- Based on “cgroups” Linux kernel functionality:
    - process-space separation
    - “chroot on steroids” - separate file systems.
- After much kernel recompilation… it worked!

Containers are Light

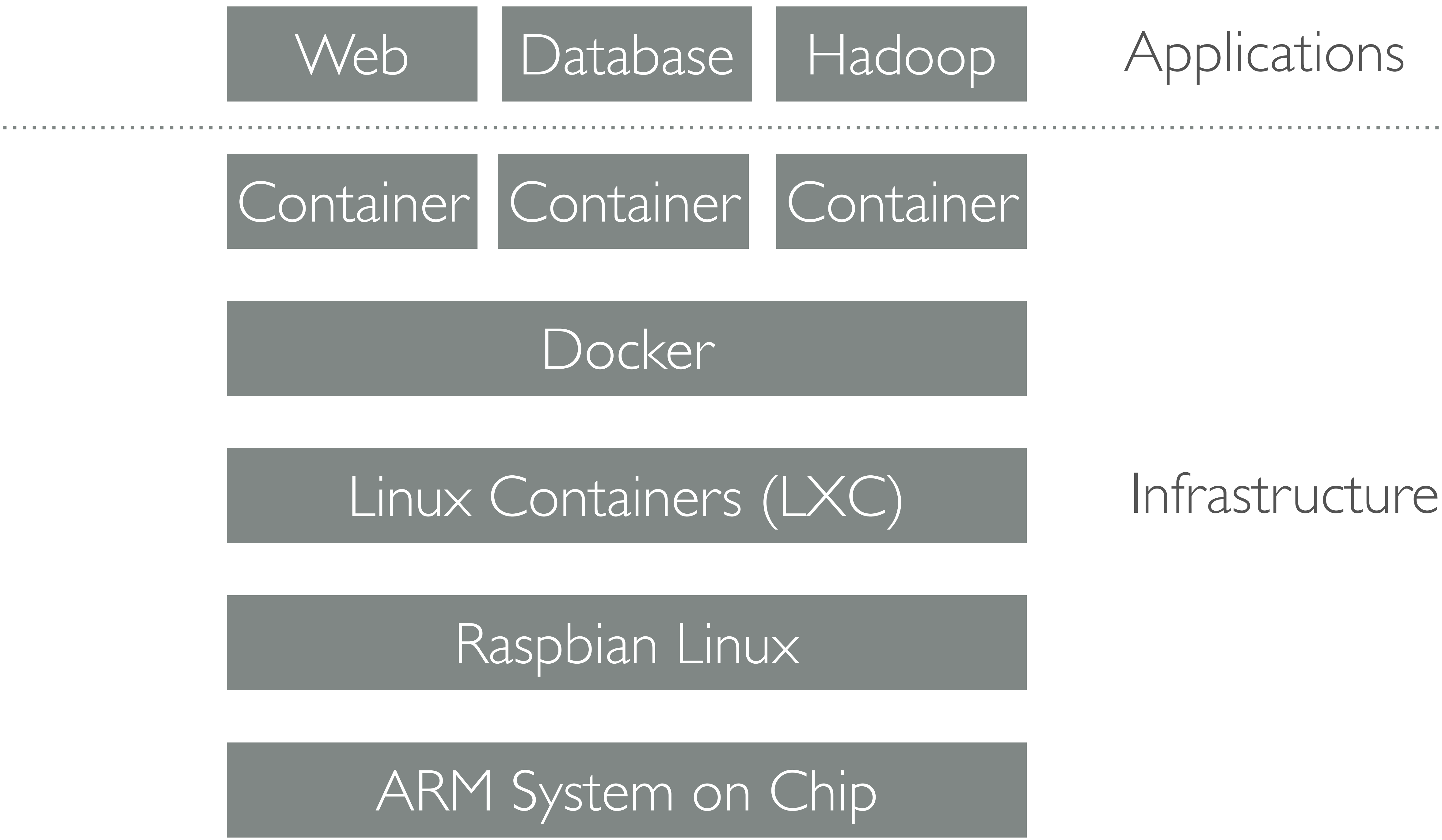
- To get LXC working — recompile kernel and configure Raspbian
- To get Libvirt working — get SELinux working on the Pi, then get all the other dependencies working. Then patch Libvirt
- To get ovirt working…
- “yak shaving”



docker

+ Shipyard

Software Stack



Applications

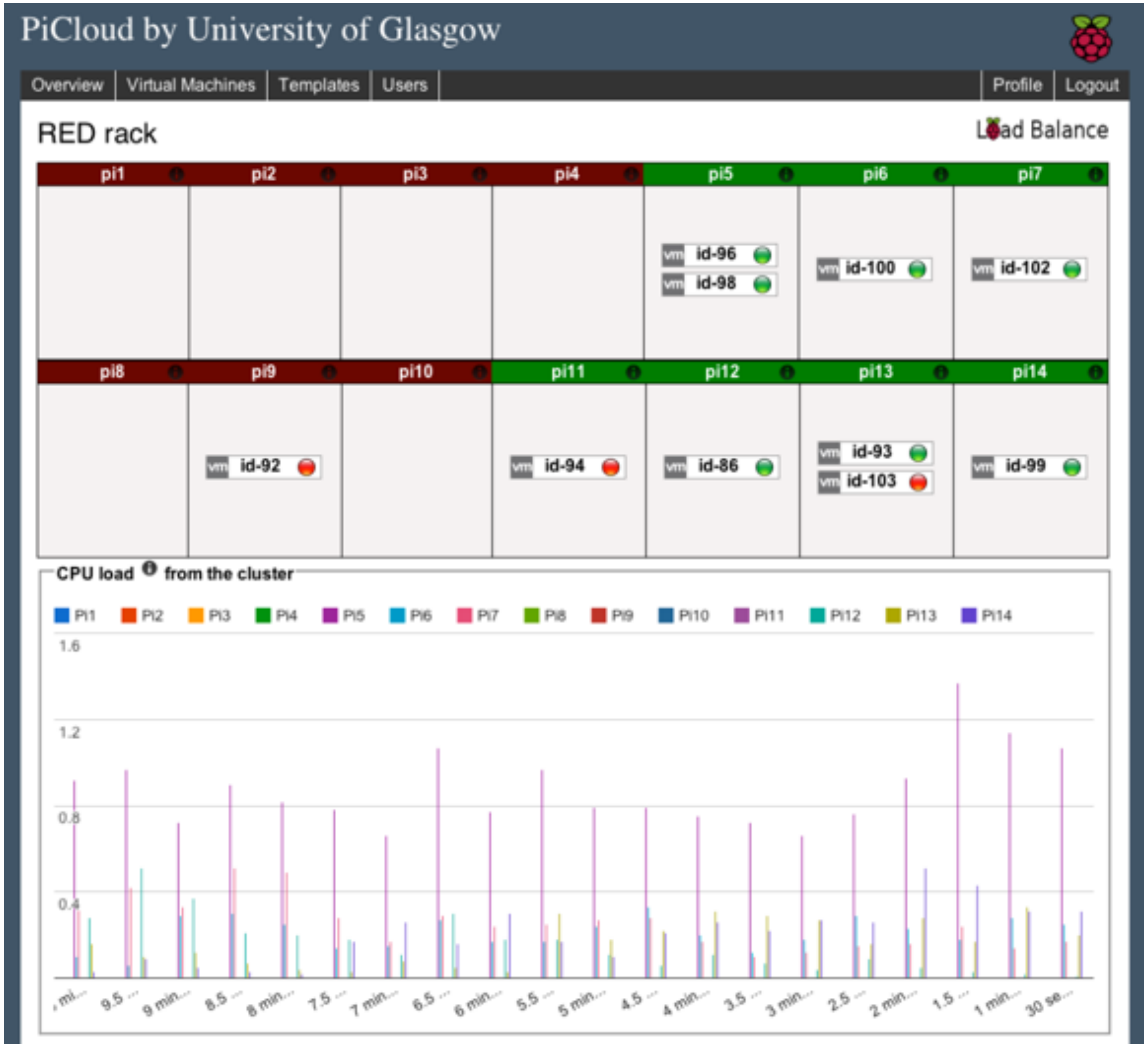
LIGHTTPD fly light.

CherryPy

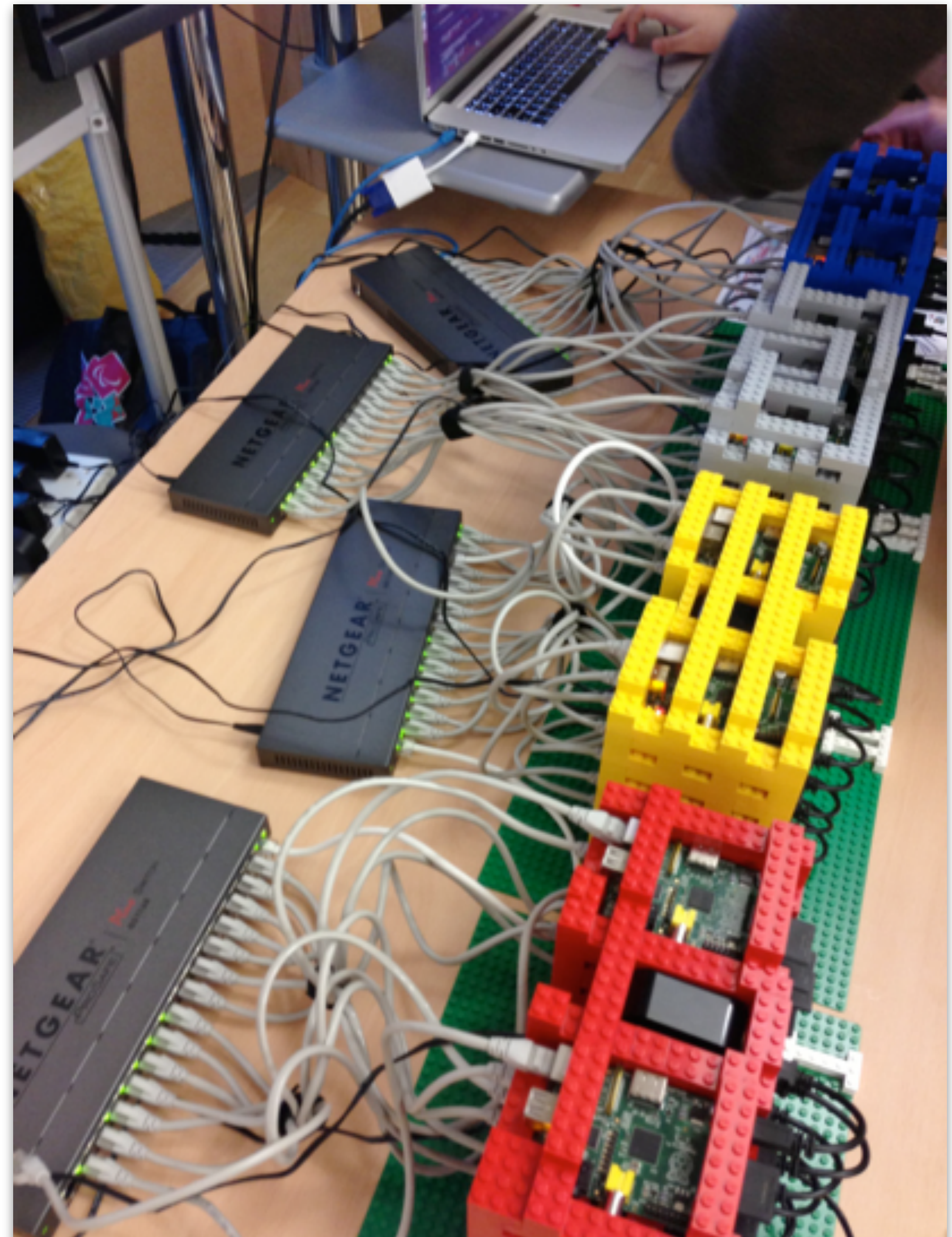
hadoop

This is **not** a compute cluster.

The Management Interface



Total Cost: £2000



DEMO

Teaching and Research Applications

Student Projects

- Hardware Design
- Web Management
- Container middleware
- Network instrumentation
- IP-less networking
- Teaching tools
- Genetic Programming in the Cloud

Side Projects

- Vector Pascal Compiler for the Pi
- Jikes RVM port to the Pi
- Robot Operating System (ROS) in the Cloud

Lectures

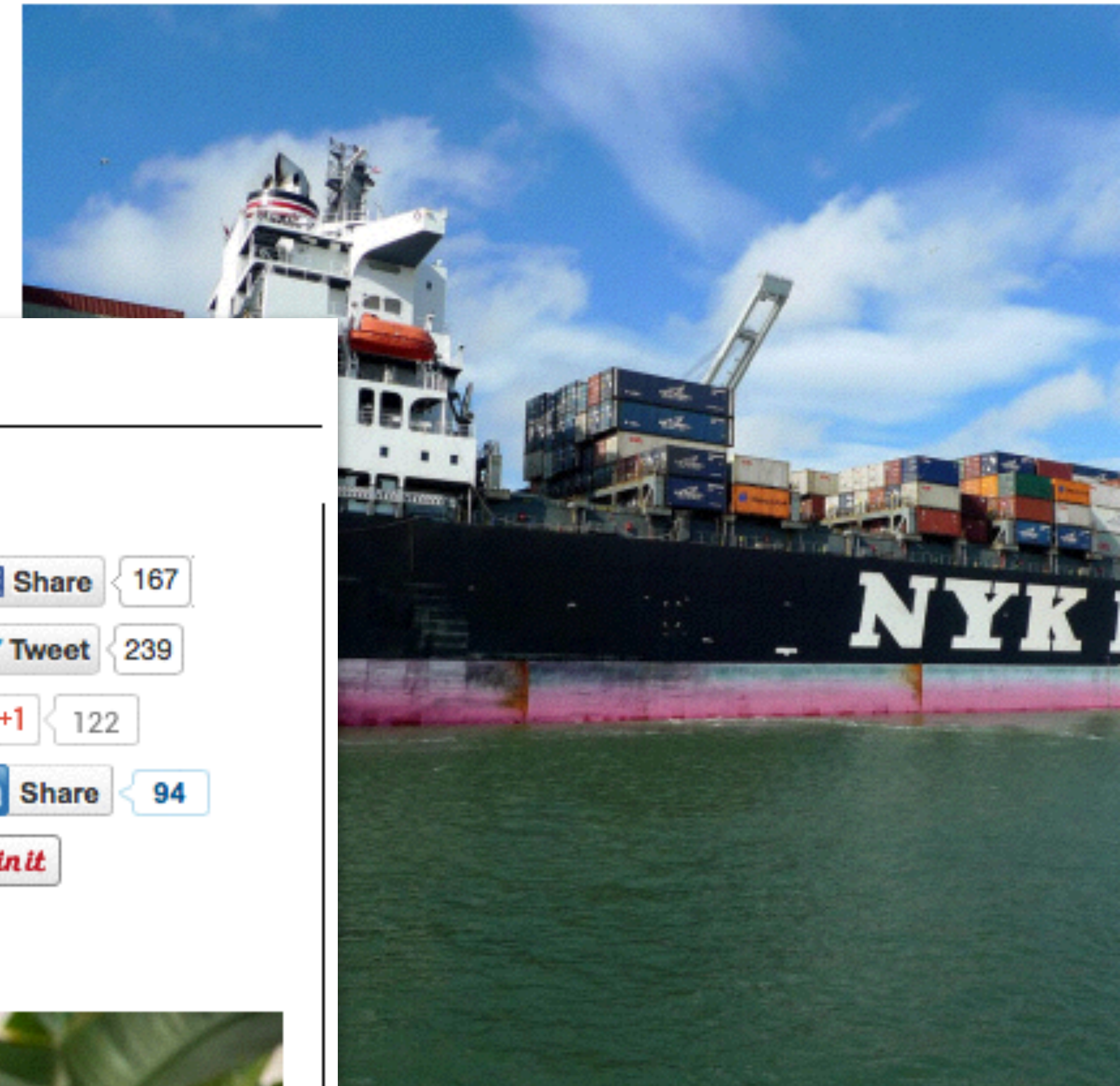
- Software distribution, version management
- Distributed programming
- Long term goal: Cloud Computing Module

Infrastructure Research

- Is the scale model faithful?
- Can lightweight containers replace virtualisation?
- Is it possible to build a robust, responsive and secure ‘peerless cloud’ ?
- Are there alternative networking paradigms that suit cloud computing better?
- Can we use dynamic optimisation algorithms to improve cloud efficiency?

ARM-based OpenStack: it's like an army of cell phones powering cloud-based computing

How Docker turned intricate Linux developer pixie dust



ENTERPRISE | cloud | open source

The Future of Cloud Computing Now Runs on All Versions of Linux

BY KLINT FINLEY 11.27.13 1:41 PM

Follow @klintron

Share 167
Tweet 239
+1 122
Share 94
Pin it

Agenda

- Introduction
- The Raspberry Pi project
- Cloud Computing and IaaS
- Building a Cloud from Raspberry Pi devices
    - *Future work, conclusion*

Future work

- More students involved
- Load balancer
- Docker integration
- Mobile Cloud management
- ARM Cloud
- and many more…

Summary

We have built a Raspberry Pi Cloud!

This is an ongoing project

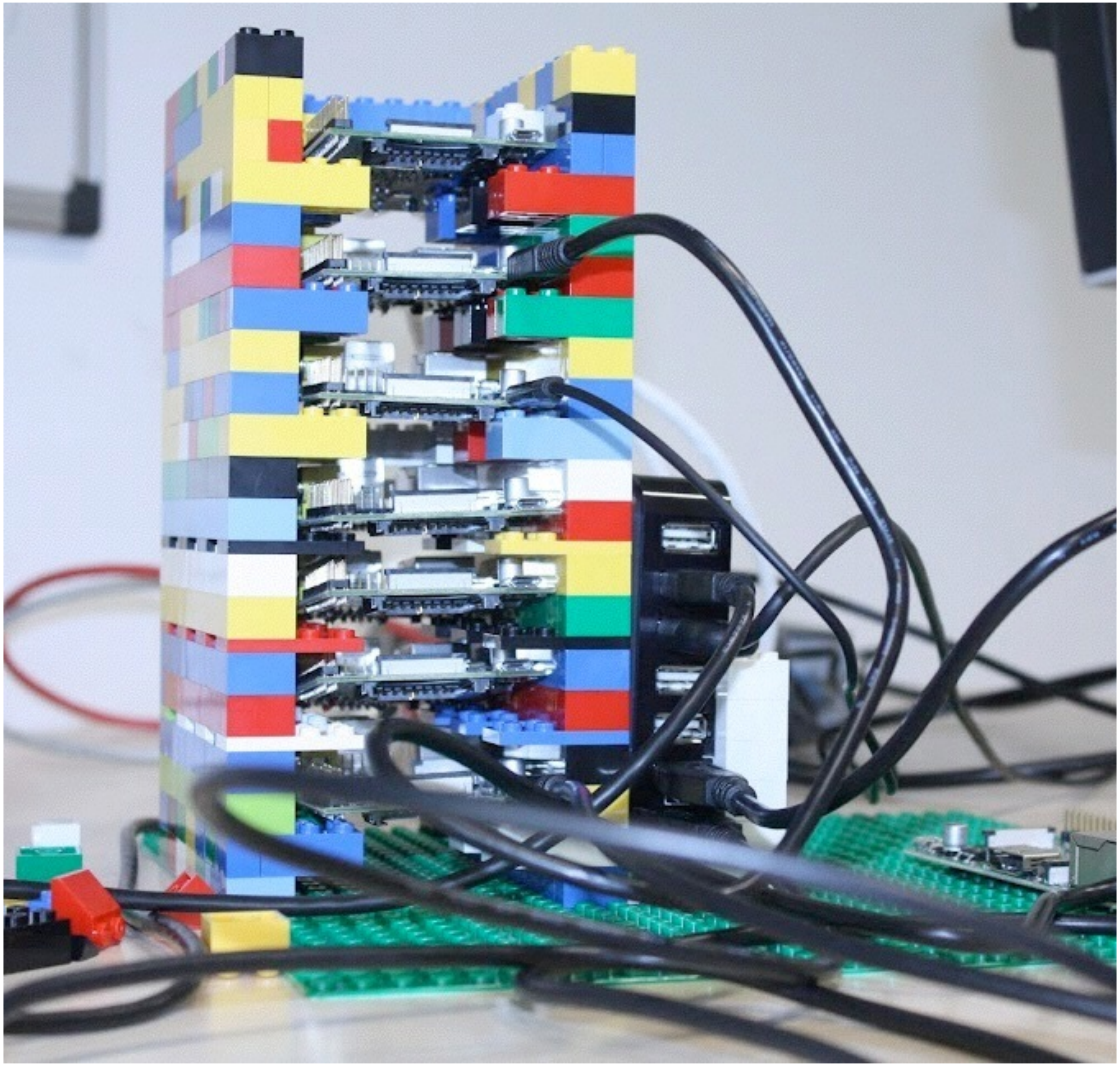
First: replicate. Then: improve.

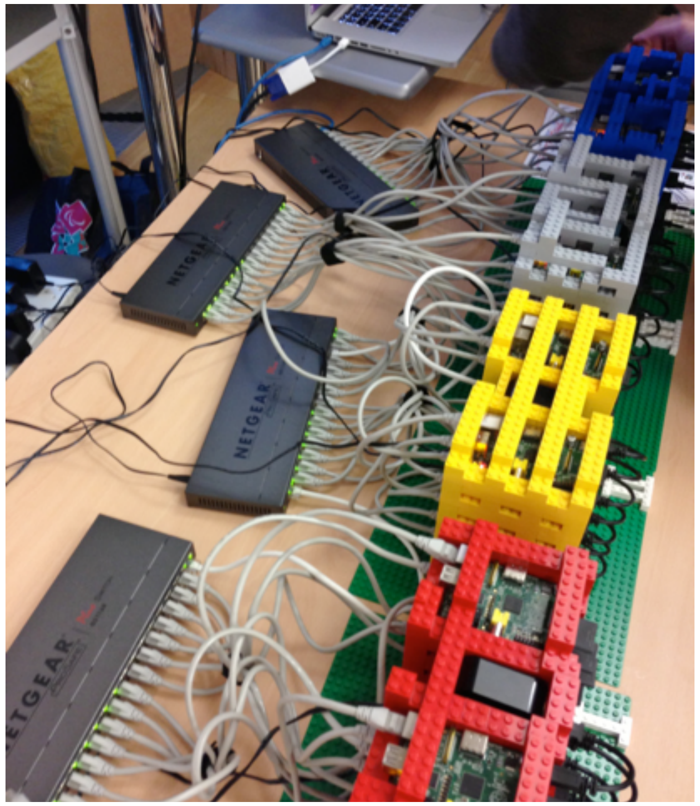
Acknowledgements

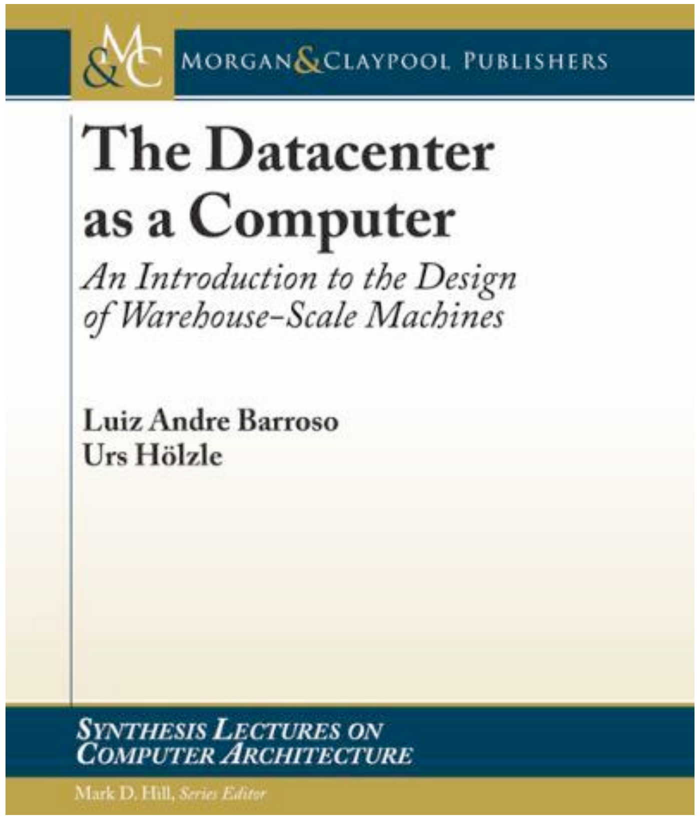
The University of Glasgow Chancellor's Fund for Learning and Teaching

Dr David R. White (Investigator)
Dr Posco Tso (Co-investigator)
Dr Jeremy Singer (Lecturer)
Dr Dimitrios Pezaros (Lecturer)
Simon Jouet (Electronics, hacker)
Robbie Simpson (Summer Intern, original architect)
Richard Cziva (Cloud Management, hacker)

http://raspberrypicloud.wordpress.com/







Thank you for your attention!
Richard Cziva