Cloud Computing and SBSE

David R. White
SSBSE 2013 Tutorial

david.r.white@glasgow.ac.uk

ssbse

University of Glasgow

Our Paper

Cloud Engineering is
Search Based Software Engineering too

Mark Harman[a], Kiran Lakhotia[a], Jeremy Singer[b],
David R. White[b], Shin Yoo[a]

[a] *University College London, Gower Street, London, WC1E 6BT*
[b] *School of Computing Science, University of Glasgow, G12 8QQ*

Abstract

Many of the problems posed by the migration of computation to cloud platforms can be formulated and solved using techniques associated with Search Based Software Engineering (SBSE). Much of cloud software engineering involves problems of optimisation: performance, allocation, assignment and the dynamic balancing of resources to achieve pragmatic trade-offs between many competing technical and business objectives. SBSE is concerned with the application of computational search and optimisation to solve precisely these kinds of software engineering challenges. Interest in both cloud computing and SBSE has grown rapidly in the past five years, yet there has been little work on SBSE as a means of addressing cloud computing challenges.

Journal of Systems & Software. Volume 86, Issue 9, September 2013, Pages 2225–2241.

Structure

1. Cloud Computing
2. Cloud Infrastructure

---

3. Opportunities for SBSE
4. Challenges

Cloud Computing

Google

"cloud computing" hype

Web | Images | Maps | Shopping | News | More | Search tools

About 2,020,000 results (0.32 seconds)

[IT Analyst Dan Kusnetzky Talks about **Cloud Computing** and Cloud ...]

slashdot.org/.../it-analyst-dan-kusnetzky-talks-about...

11 Jul 2013

Dan Kusnetzky and I started out talking about **cloud computing**; what it is and isn't, how "cloud" is often ...

[Up in the Cloud: **Hype** and High Expectations for **Cloud Computing** ...]

knowledge.wharton.upenn.edu/article.cfm?articleid=3171

16 Jan 2013 - Up in the Cloud: **Hype** and High Expectations for **Cloud Computing** by Knowledge@Wharton, the online business journal of the Wharton ...

[Cutting Through the **Hype** on **Cloud Computing** | Doug Hornig ...]

www.financialsense.com/.../doug.../cutting-through-**hype-cloud-computi**...

6 Sep 2012 - How many times have you heard something like this? "The Cloud is going to change the way we ______." Fill in the blank with any phrase from ...

[Is **Cloud Computing** just **Hype**? - Geekswithblogs]

geekswithblogs.net/hroggero/.../08/.../is-**cloud-computing**-just-**hype**.aspx

9 Aug 2013 - So.. is **Cloud Computing** really just **hype**? I thought I would touch on a topic that seems to come up frequently from the business community; ...

[What's wrong with the Gartner **Hype** Cycle for **Cloud Computing** ...]

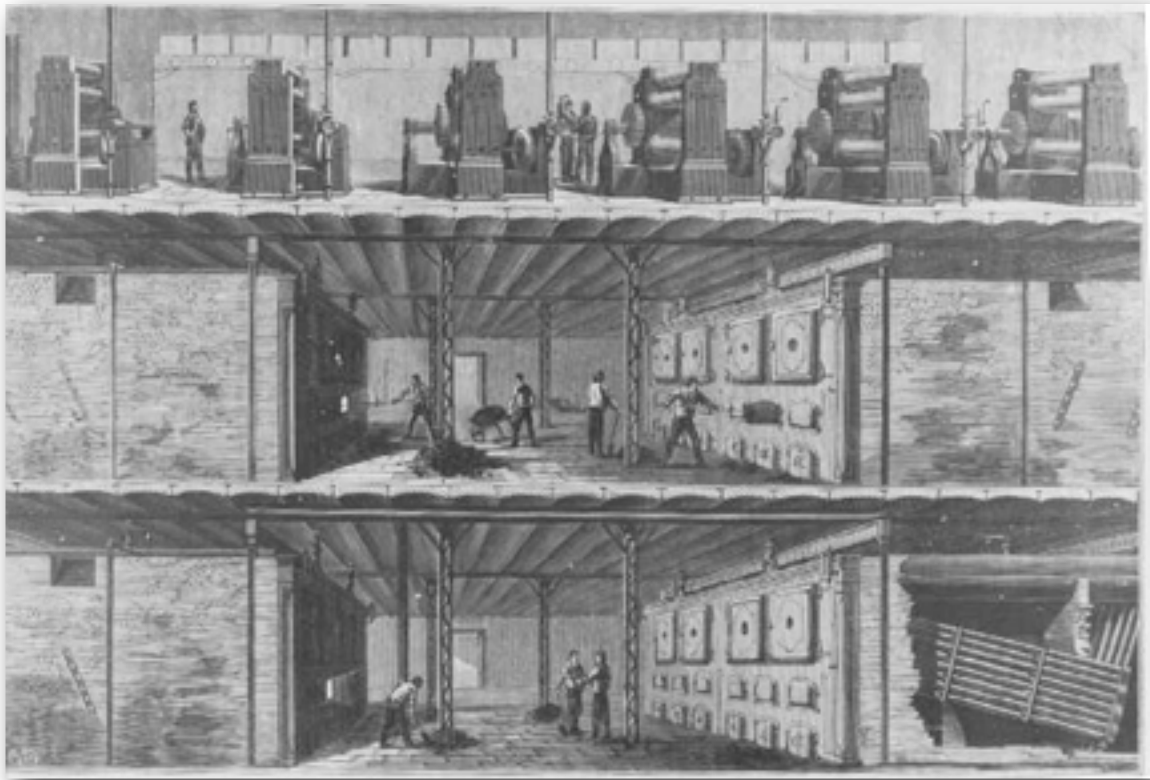
search**cloudcomputing**.techtarget.com/.../Whats-wrong-with-the-Gartner-...

Cloud computing is in various stages of adoption, says research that appears in a Gartner **Hype** Cycle for **Cloud Computing** report, though some argue its terms ...

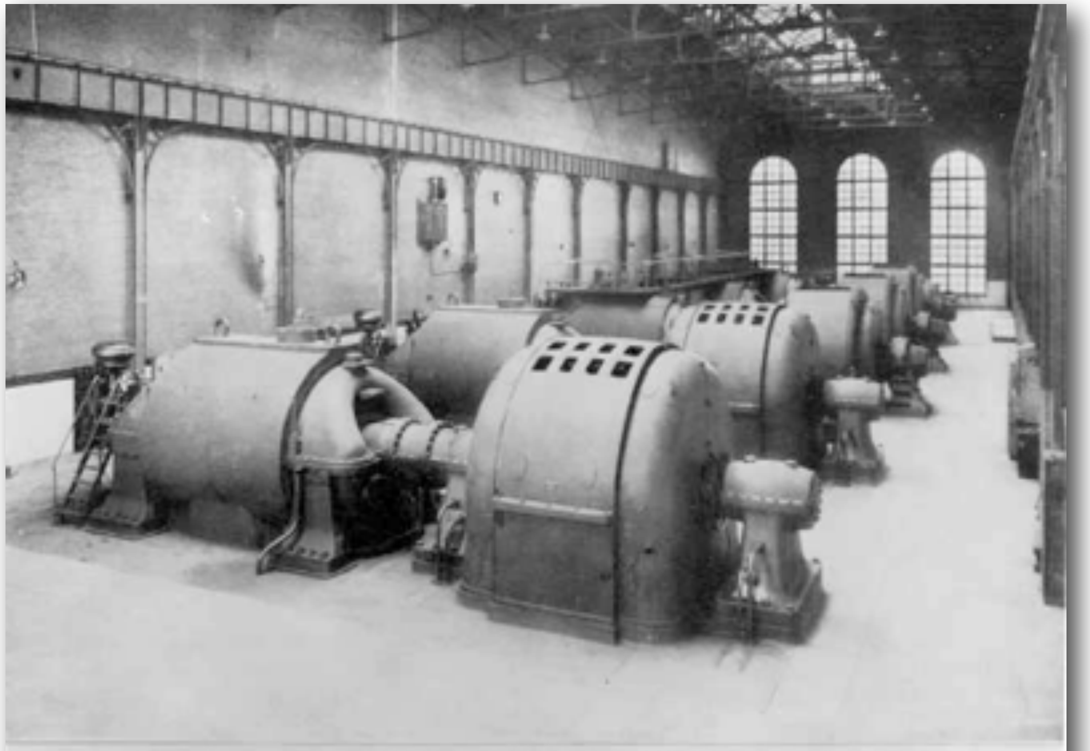
© Sandy Fleischmann / CC-BY-2.0

"The Big Switch"
Nicholas Carr

http://www.nicholasgcarr.com/bigswitch/



Edison's Pearl Street Station
c. 1883



2. STEAM TURBO-GENERATORS, LONG ISLAND RAILWAYS, LONG ISLAND CITY, N. Y.

Steam generators,
Long Island Railways
c.1907

Economies of Scale

Consolidation on a massive scale.

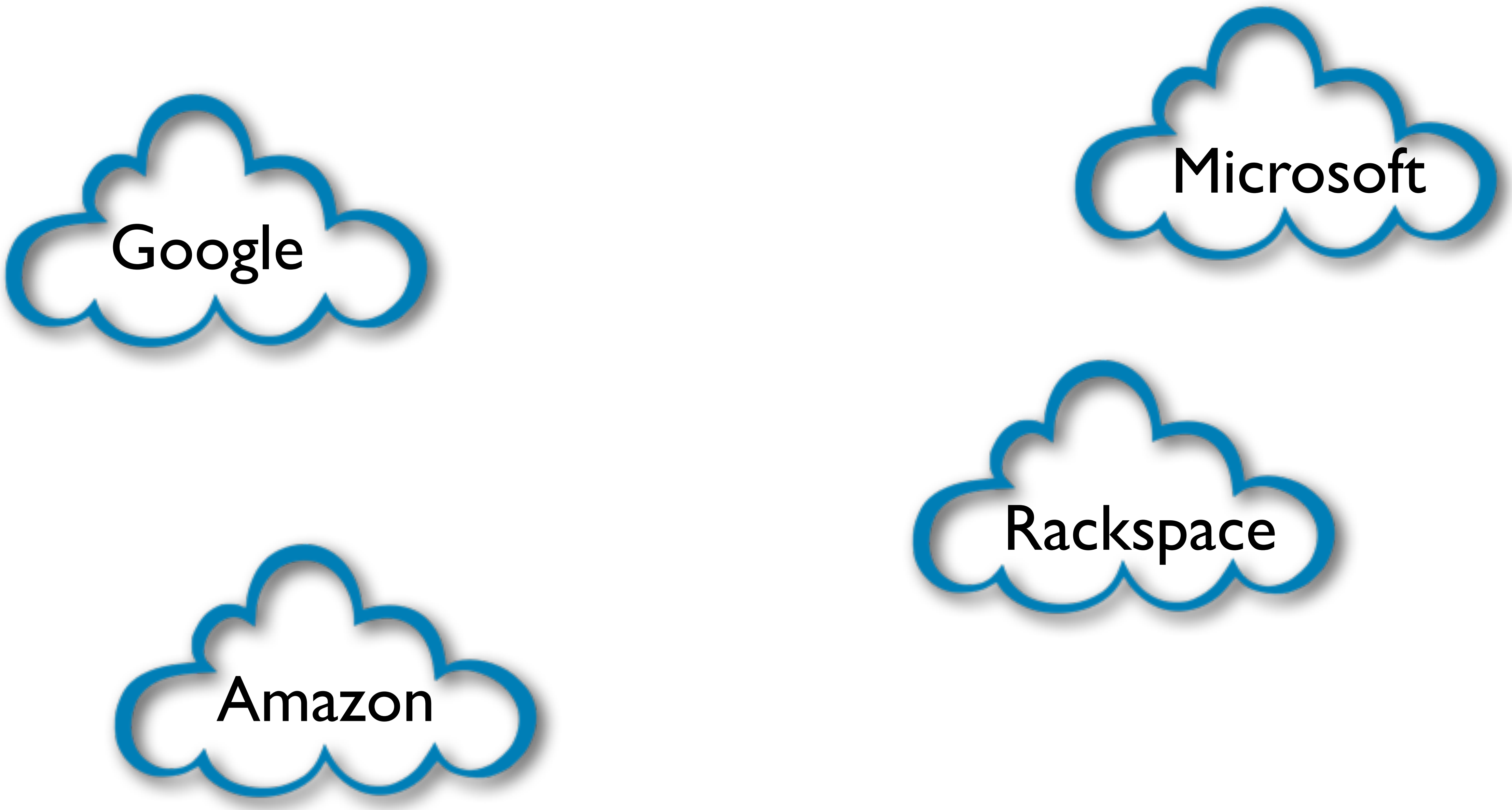
Improved efficiency through centralisation of *innovation*.

Creates new possibilities.

Is there one Cloud?

Is there one Cloud?

Is there one Cloud?



Main Cloud Providers

Amazon Web Services

Google

Microsoft Azure

GoGrid, Rackspace.

How Big is the Cloud?

Company	Number of Servers	Date
Rackspace	94122	March 2013
Facebook	hundreds of thousands	June 2013
Microsoft	> 1 million	July 2013
Google	> Microsoft	July 2013

Single datacentre 10000 - 50000 servers.

http://www.datacenterknowledge.com/archives/2009/05/14/whos-got-the-most-web-servers/

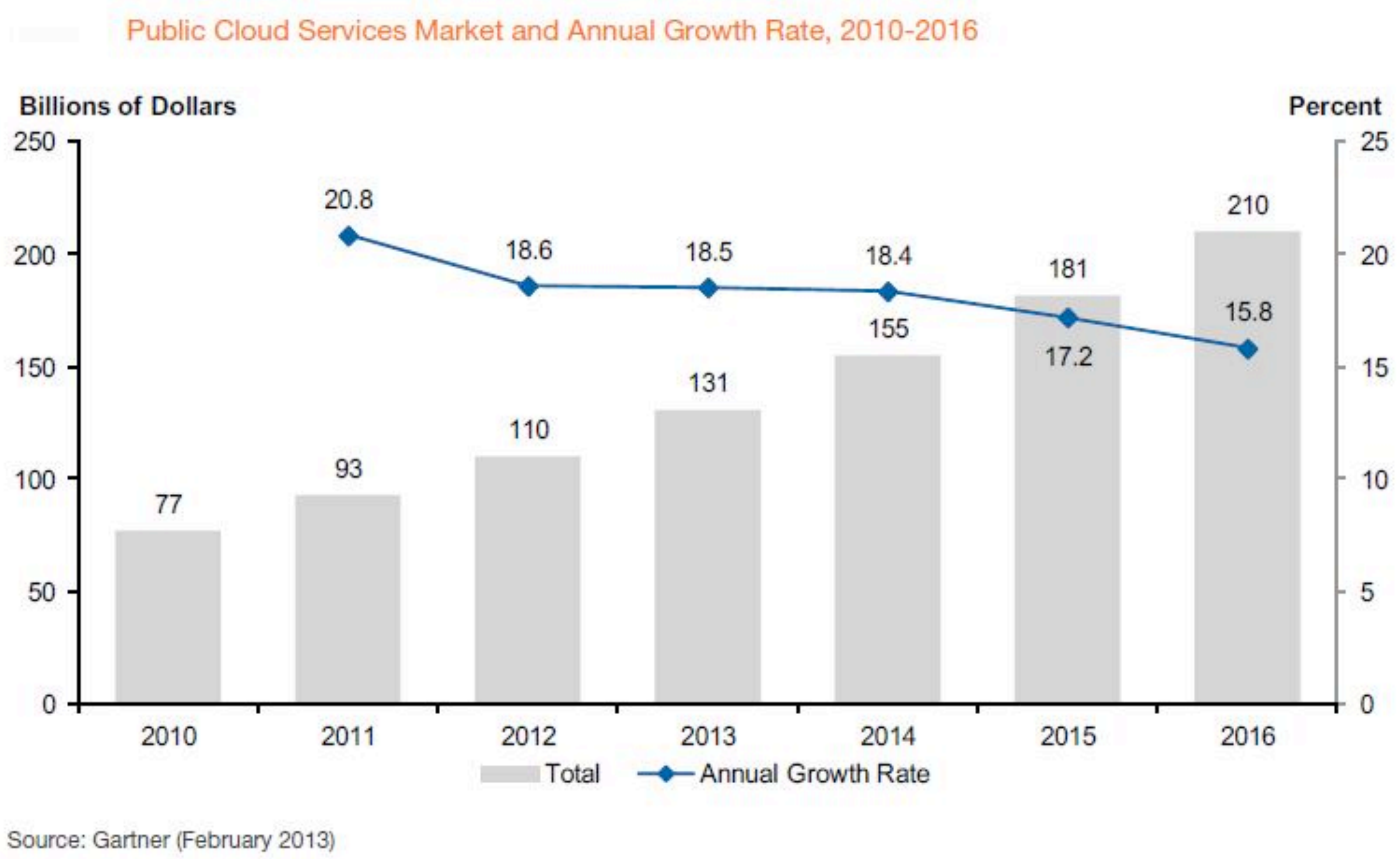
http://www.extremetech.com/extreme/161772-microsoft-now-has-one-million-servers-less-than-google-but-more-than-amazon-says-ballmer

“**Each day** Amazon Web Services adds enough new capacity to support all of Amazon.com’s global **infrastructure** through the company’s **first 5 years**, when it was a $2.76B annual revenue enterprise."

James Hamilton, Amazon Web Services Team.

http://mvdirona.com/jrh/work/

Cloud Uptake



http://www.forbes.com/sites/louiscolumbus/2013/02/19/gartner-predicts-infrastructure-services-will-accelerate-cloud-computing-growth/

A Client's Viewpoint

Pay for server hours, not for servers.

Elastic scaling allows adaptive supply.

Massively parallel processing for everyone (even GPGPU).

Vast data storage and processing.

Incentives to Adopt

Switch to operating, not capital, budget.

Focus on processors, memory, power, network.

Lower management overheads.

Disadvantages for Clients

Vendor lock-in.

Reliance: Reputation wins over SLAs.

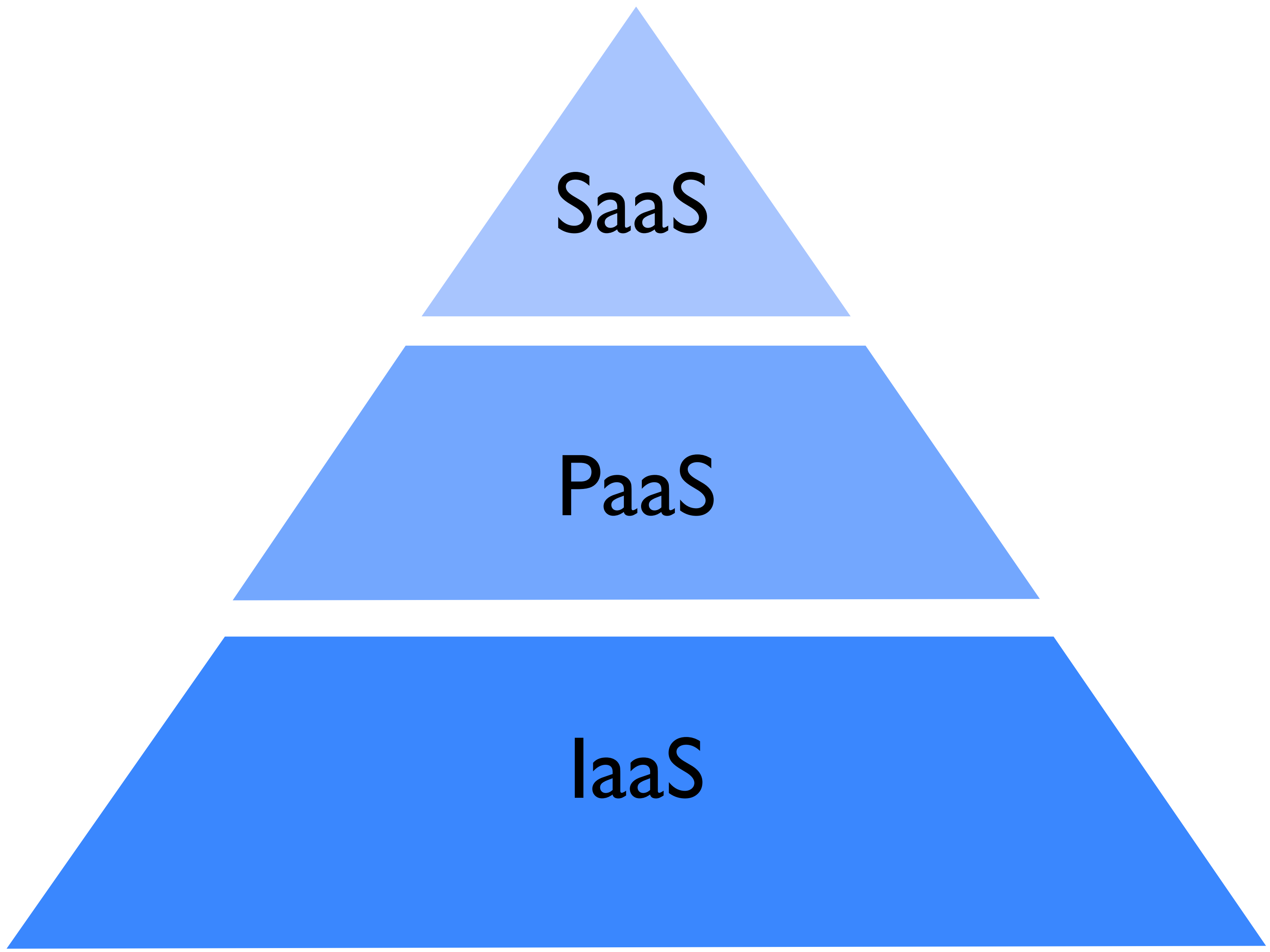
Data security (hackers / governments).

Provenance issues.

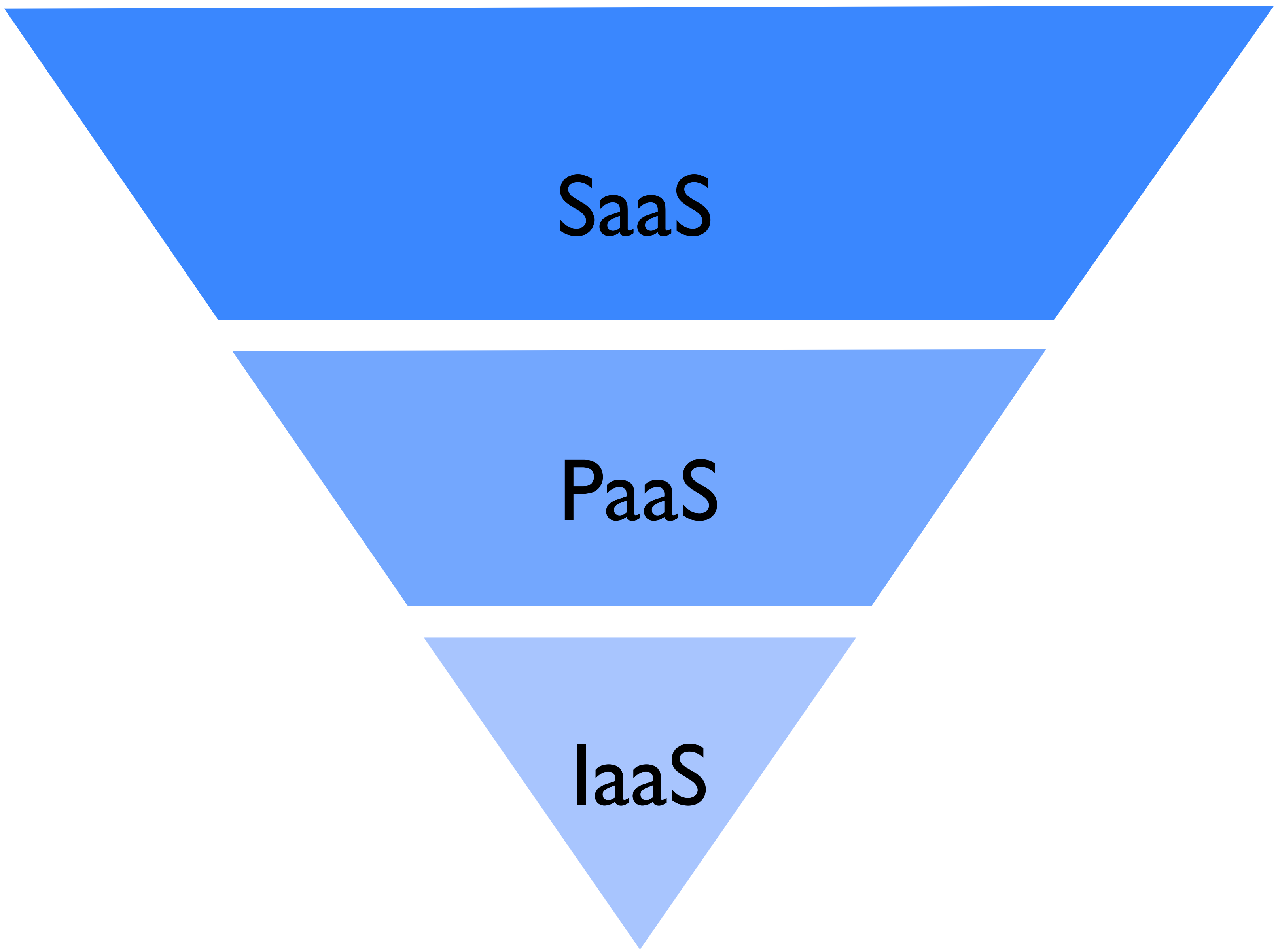
Migration costs.

http://blogs.gartner.com/lydia_leong/2012/12/05/cloud-iaas-slas-can-be-meaningless/

* as a Service



* as a Service



* as a Service

Continued

Business Process as a Service could be more significant.

- Payments, Billing.
- Credit checking.
- Automated Telephony.
- Printing.

Meta-Service aaS

Meta-services have arisen to broker and resell cloud services.

Example Pricing

Provider	VCPUs	RAM	Disk	Per hour
Google	1	3.75GB	420GB	$0.145
Google	1 Shared	0.6 GB	0	$0.021
Amazon	1	3.7GB	410GB	$0.12
Amazon	1	0.6 GB	0	$0.02

Assuming EU Hosting.

http://calculator.s3.amazonaws.com/calc5.html

https://cloud.google.com/pricing/compute-engine

A Contrarian Viewpoint

Lanier warns of the rise of "Siren Servers" and their potential to distort the distribution of power and wealth in society.

NIST Definition

"Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

Criticised for being too narrow.

http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

Demo...

Google Compute Engine

https://cloud.google.com/products/compute-engine

2. Cloud Infrastructure

Google's datacentre in Council Bluffs, Iowa

http://www.google.co.uk/about/datacenters/


The Hot Aisle
http://www.google.co.uk/about/datacenters/


Microsoft, Dublin.
$500 million.

Facebook, N.C.
$606 million.

Apple, Maiden, DC.
$1 billion.

Servers

Tens of thousands of individual x86 servers.

Arranged in racks of anywhere between 20 and 64 servers, subdivided into chassis.

Carry multiple network ports, may use multihoming.

Batteries on the rack - avoid UPS.

Servers

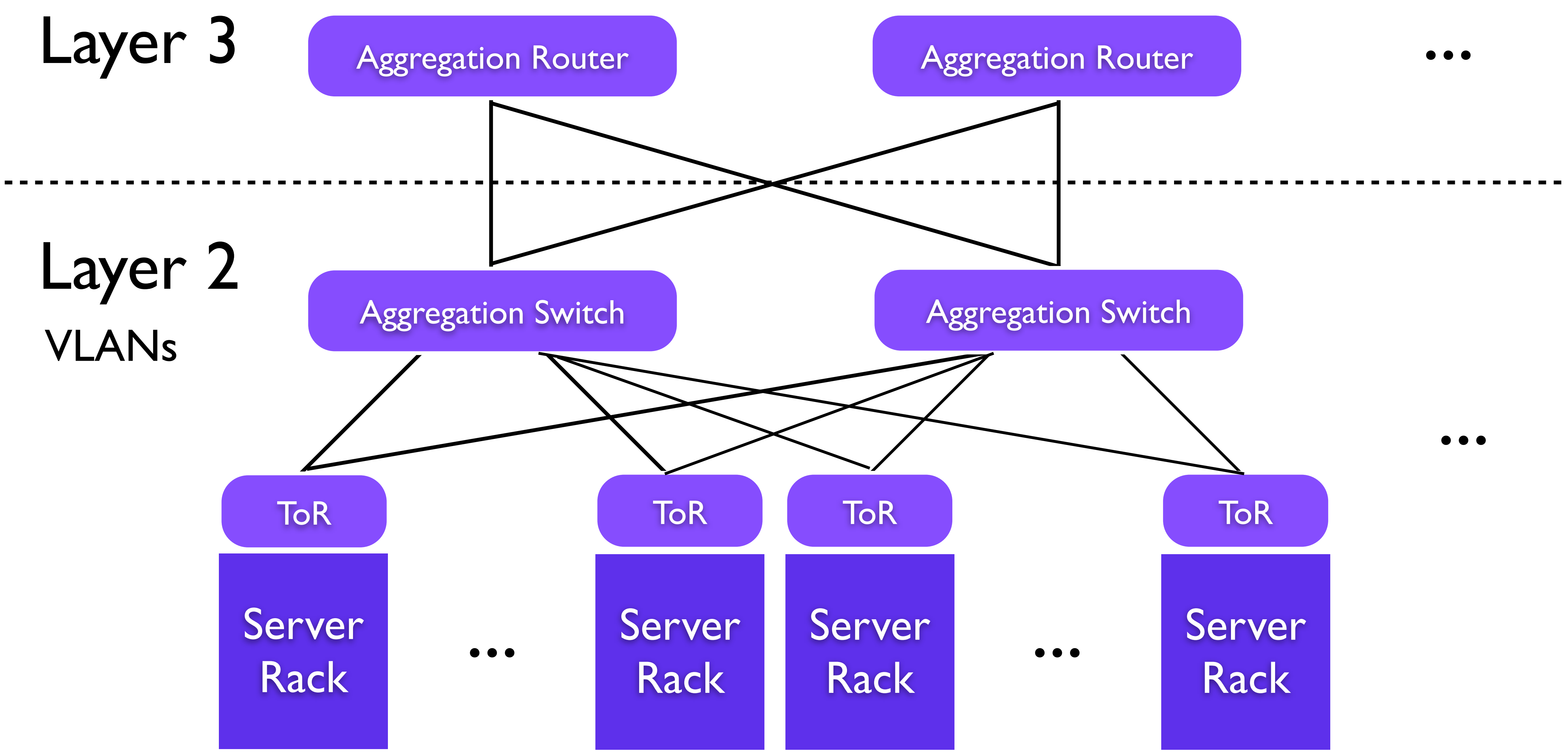
Not composed of off-the-shelf hardware.

Large providers buy components (e.g. CPU, RAM) directly, e.g. from Intel, and use customised boards to omit unneeded components.

Blade and non-blade designs.

http://www.opencompute.org/wp/wp-content/uploads/2013/02/Open_Compute_Project_Intel_Server_Open_Rack_v0.3.pdf

Network Topology



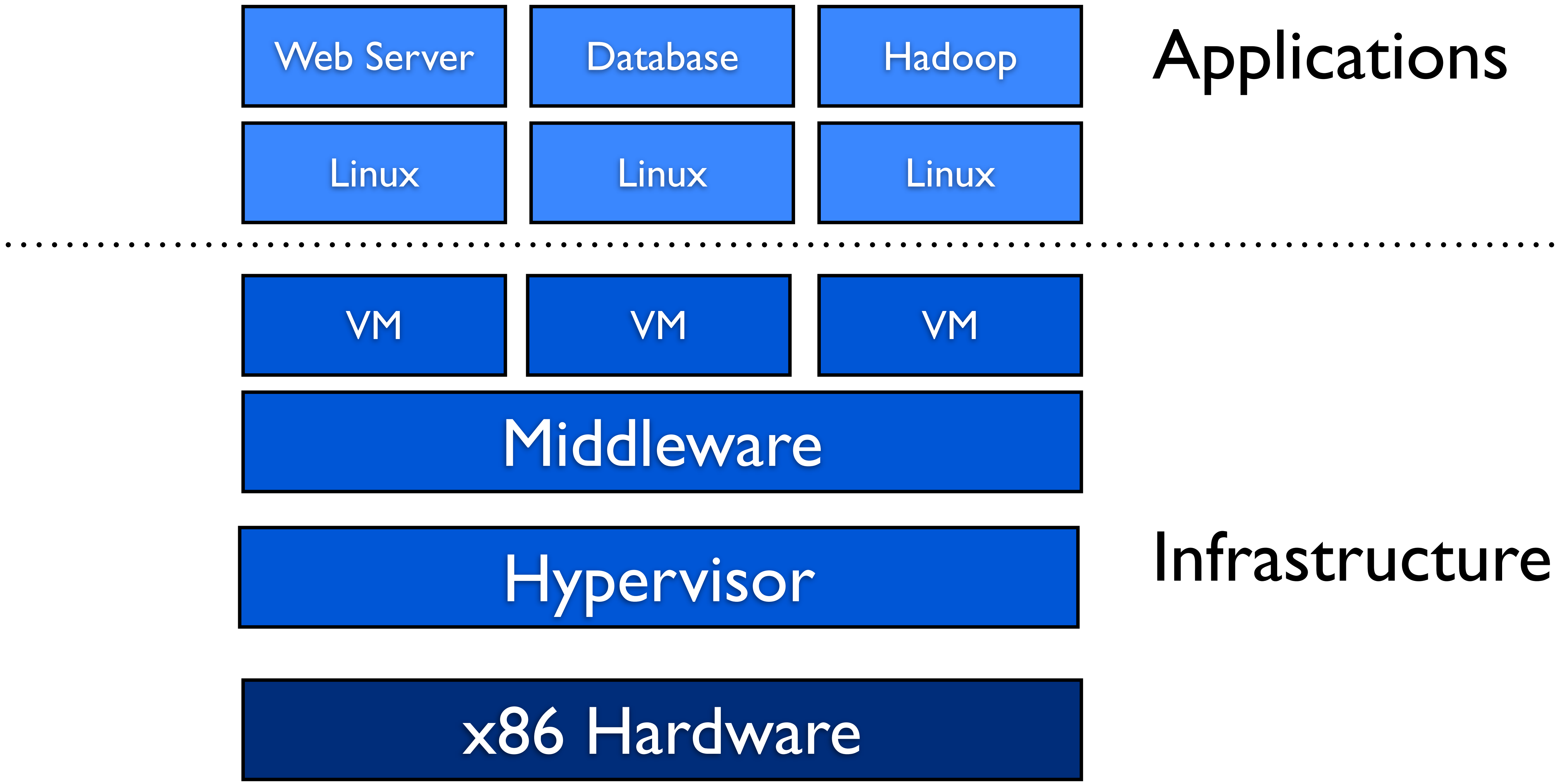
VL2: A Scalable and Flexible Data Center Network. Greenberg et al. , SIGCOMM 2009.

Open Compute

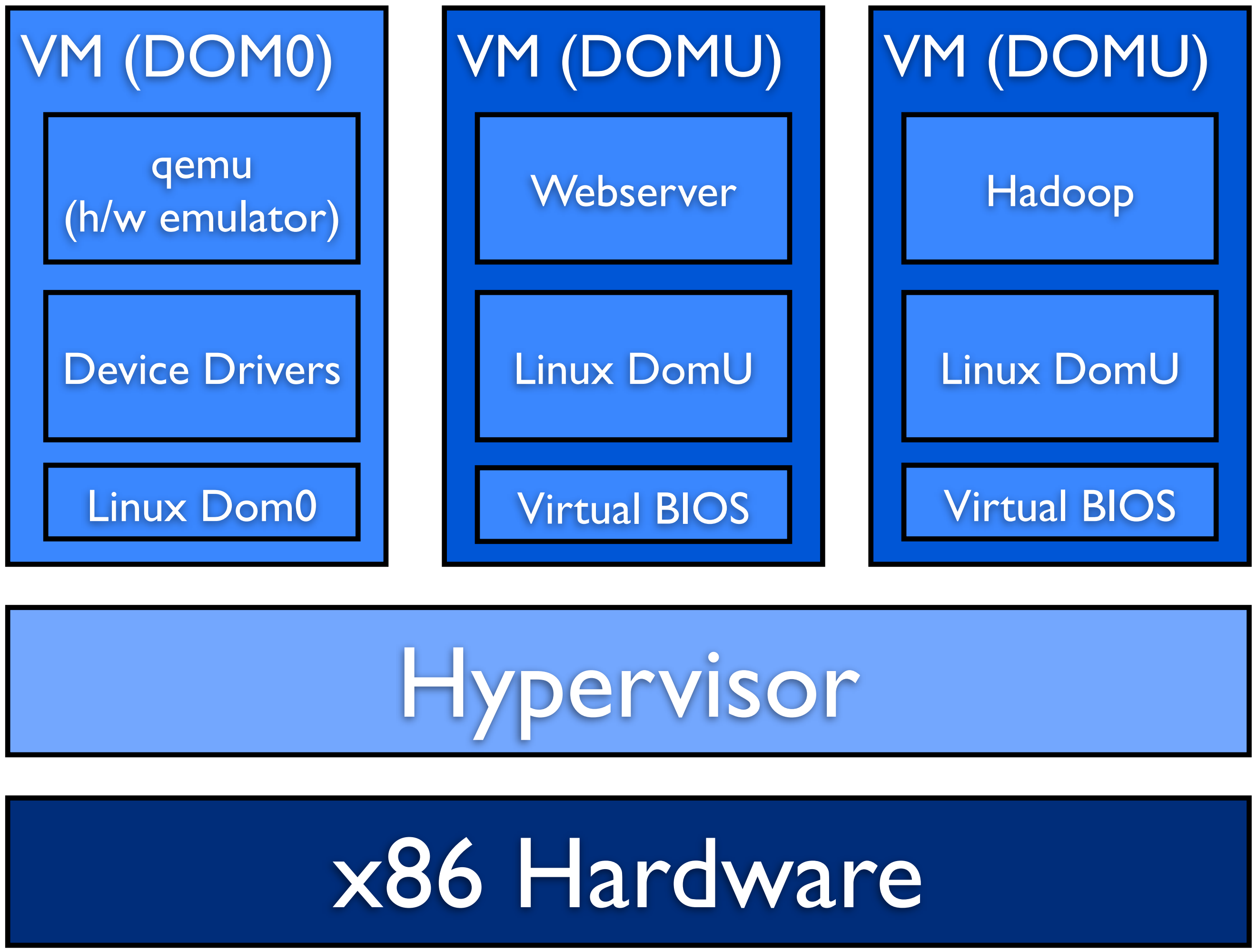
Facebook have open-sourced their designs:

http://www.opencompute.org/

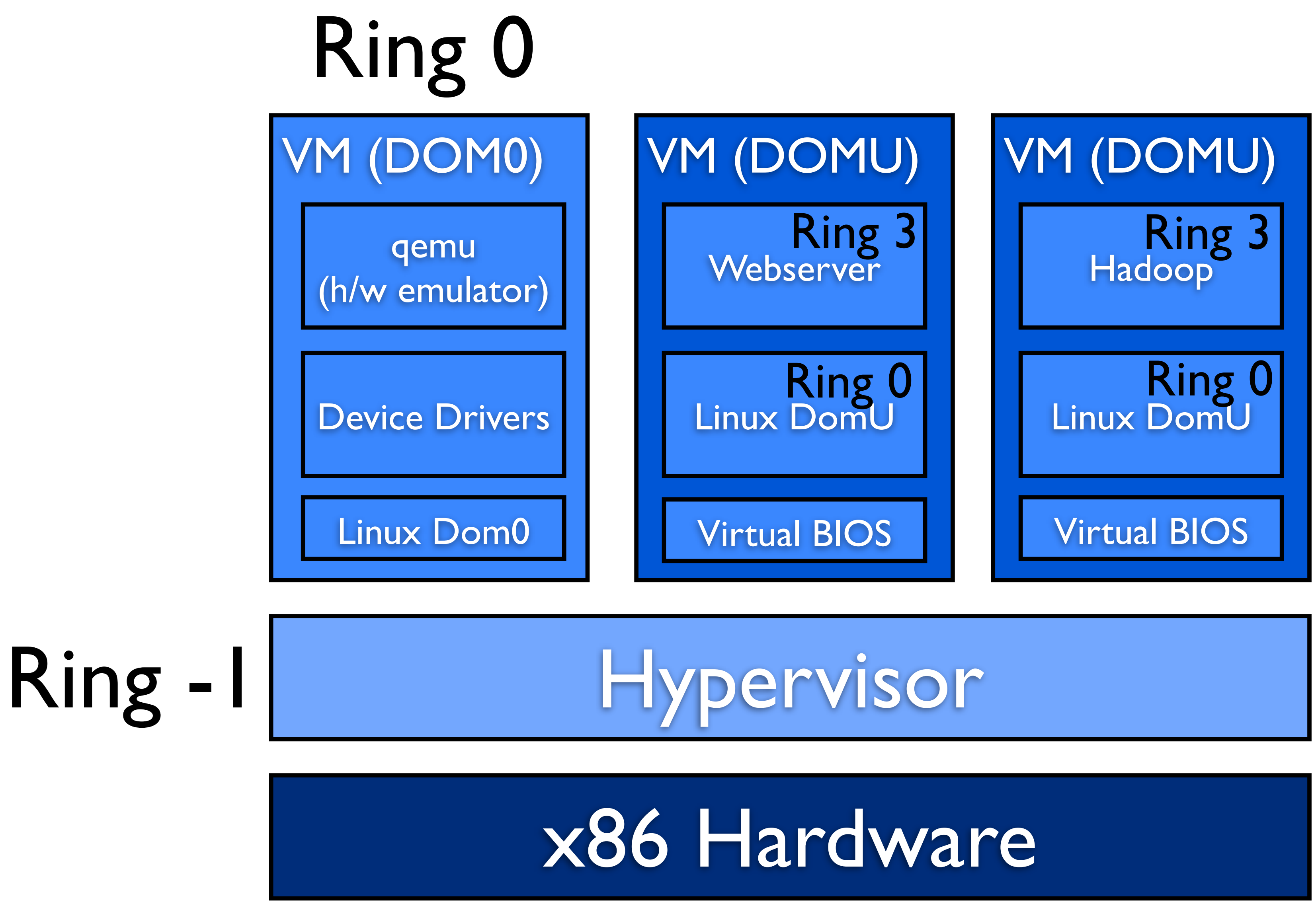
Typical Cloud Stack



Virtualisation



Virtualisation



Virtualisation

Full (hardware assisted) virtualisation - runs unmodified guests.

Paravirtualisation - cooperation of host operating system to improve performance. Operations involving privileged instructions are replaced with calls to the hypervisor.

Hypervisors

Hypervisor	Users	Model	Type
Xen	Amazon, Rackspace	Open Source	Full/Para.
VMWare ESX	Most	Proprietary	Full
Hyper-V (MS)	Microsoft	Proprietary	Full
KVM	Google	Open Source	Full

Xen Hypervisor

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

Xen and the Art of Virtualisation. Barham et al. SOSP 2003

Checkpointing & Snapshotting

Snapshotting - quickly backup a machine to disk.

Live migration - move VMs between machines for consolidation and maintenance. Very short downtimes.

Software Suites

OpenStack.

Eucalyptus.

http://www.openstack.org/

http://www.eucalyptus.com/

Linux Containers

"chroot on steroids"

Alternative to full virtualisation.

Uses Linux kernel control groups and namespaces functionality to isolate processes.

Lower overhead than virtualisation.

Not entirely secure (yet), so not suited for multi-client cotenancy.

https://www.docker.io/

Data Storage

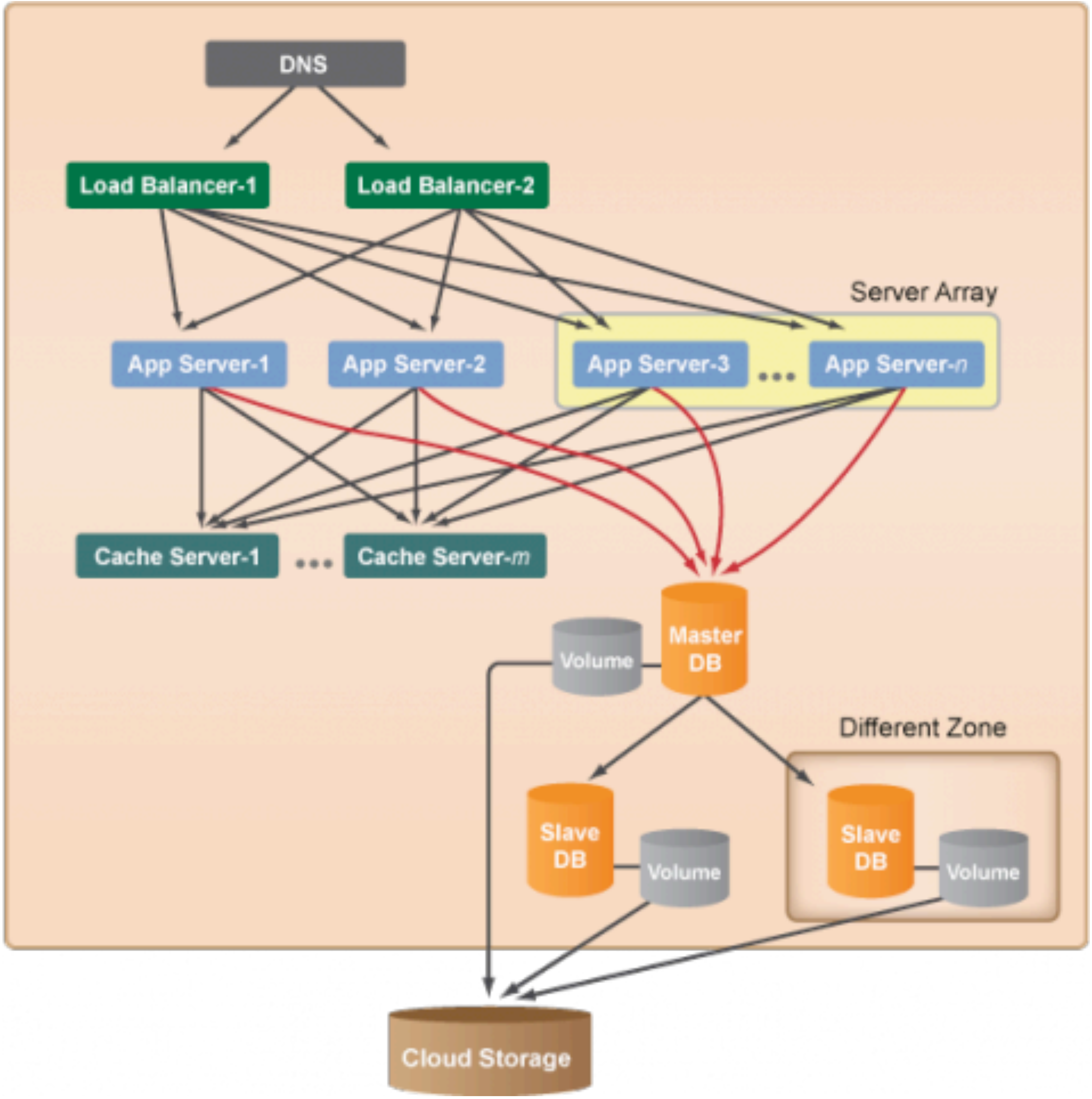
Hard disks mounted with servers; data sharded and distributed.

Two main storage types:

1. Key-Value stores (Amazon S3)

   See Megastore, Spanner, NoSQL DBs.

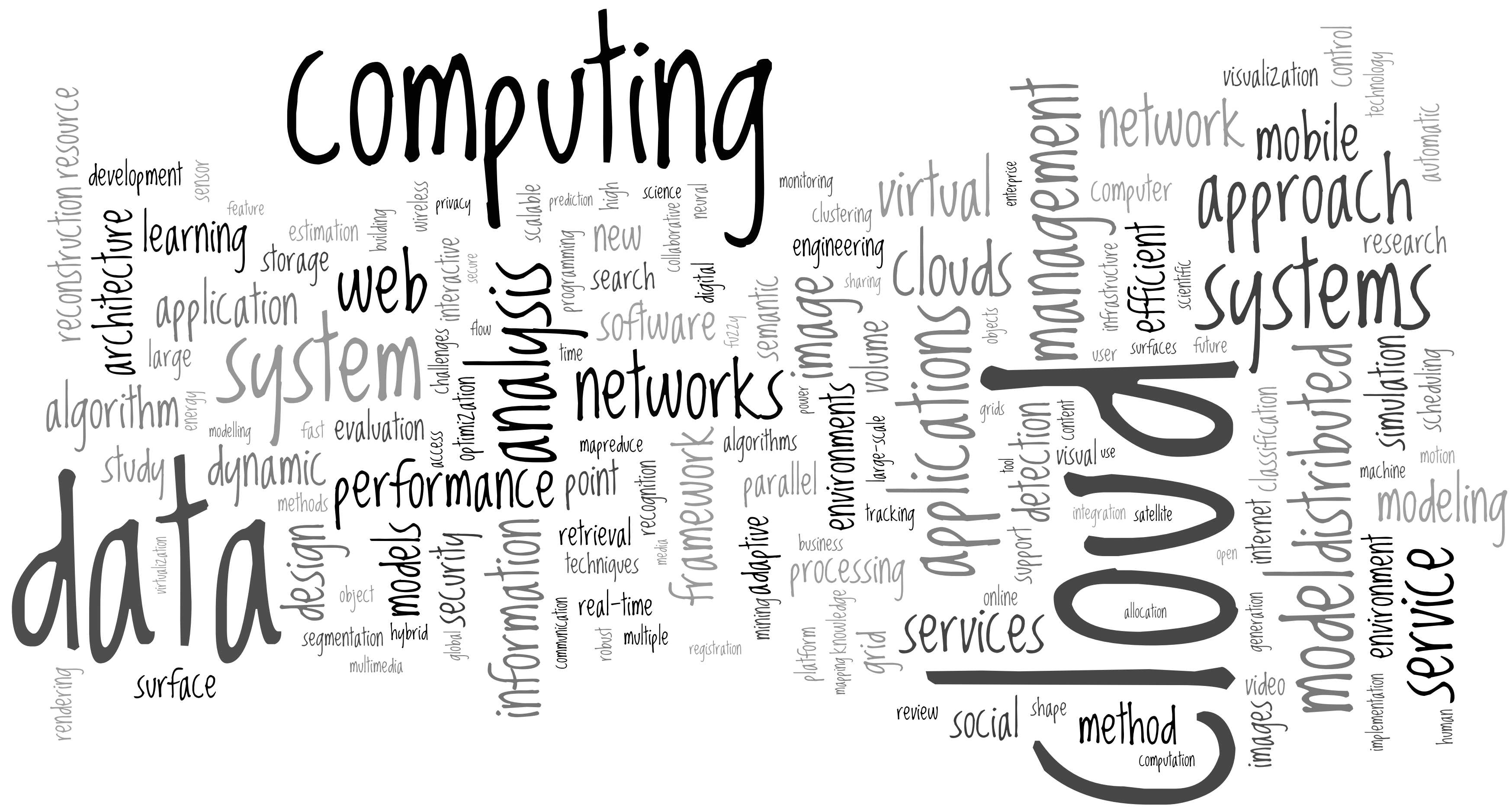
2. File systems (Amazon EBS).



http://www.rightscale.com/blog/enterprise-cloud-strategies/architecting-scalable-applications-cloud

3. Opportunities for SBSE

Cloud Word Cloud



Cloud Word Cloud

Cloud Word Cloud

Search - Optimisation - Adaptation
Scheduling - Control - Classification - Estimation
Clustering - Mining - Features
Efficiency - Power - Time

Consolidation

One of the original motivations for VMs.

Cloud computing is consolidating *everything*.

Many problems in the Cloud are concerned with how to *optimise* this consolidation.

Provider Objectives

Fewer servers.

Energy efficiency.

Increase bandwidth, lower latency.

Managing oversubscription.

Reliability.

Client Objectives

Anticipate demand.

Scalable systems: refactoring, parallelisation, management of scaling.

Resource efficiency - lower outgoings.

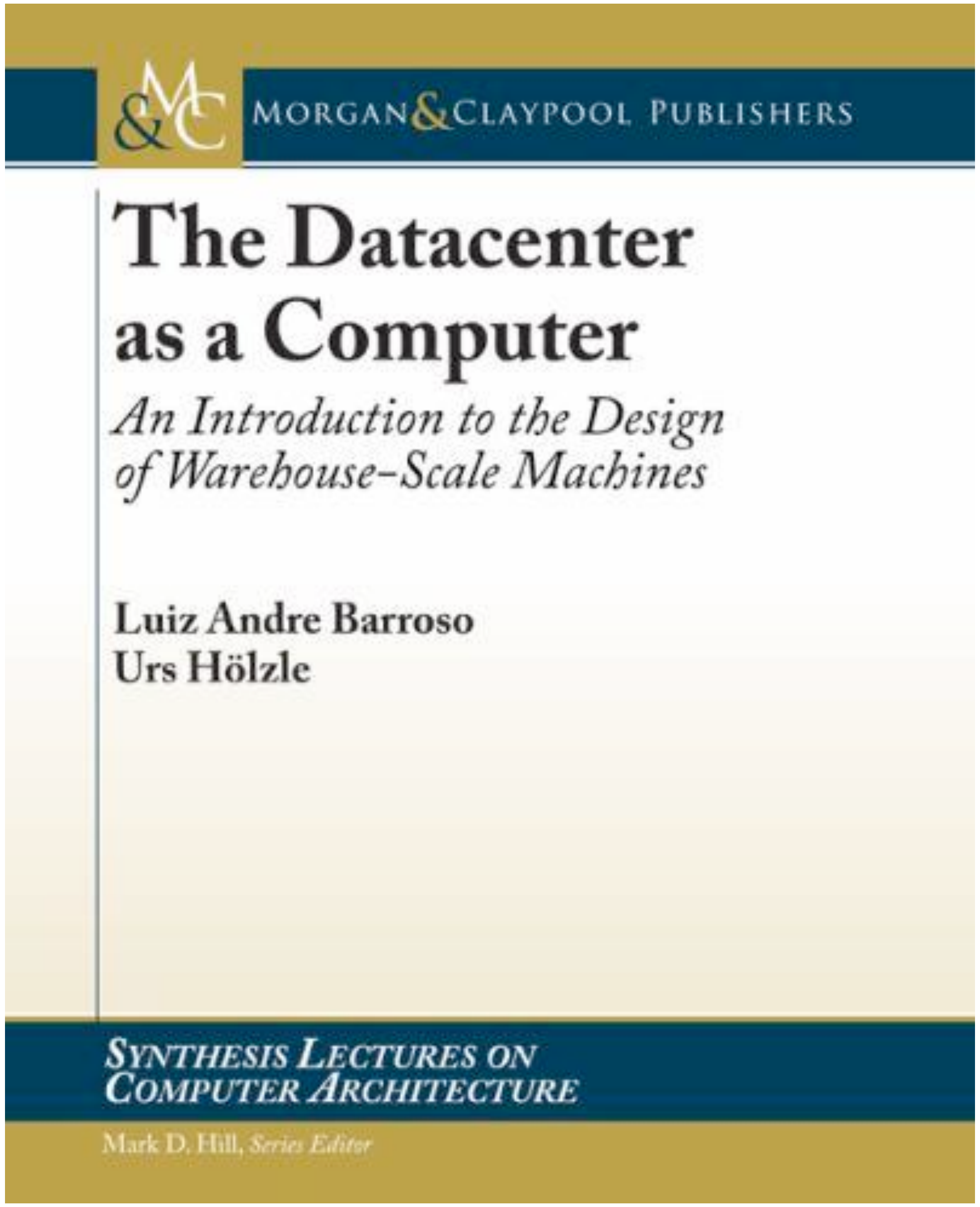
Fault tolerance.

Security.

Efficient business decisions.

SSBSE-Friendly Features of the Cloud

A Different Kind of Software Engineering



Accelerated Development

Deployment on a daily or weekly basis.

New deployment models:

- Canarying.
- A/B testing.
- Blue-green deployment.

http://martinfowler.com/bliki/BlueGreenDeployment.html

Multiplicity Computing

Deployment models and duplication provide great opportunities for SBSE:

Multiple oracles.

Real-time feedback.

Multiplicity Computing: A Vision of Software Engineering for Next-Generation Computing Platform Applications. Cadar et al. FoSER 2010.

Big SBSE Data

Possible to gather large amounts of data not previously available.

Usage profiles without deploying probes.

Quantifying Costs

A Systematic Study of Automated Program Repair: Fixing 55 out of 105 Bugs for $8 Each

Claire Le Goues Michael Dewey-Vogt
Computer Science Department
University of Virginia
Charlottesville, VA
{legoues,mkd5m}@cs.virginia.edu

Stephanie Forrest
Computer Science Department
University of New Mexico
Albuquerque, NM
forrest@cs.unm.edu

Westley Weimer
Computer Science Department
University of Virginia
Charlottesville, VA
weimer@cs.virginia.edu

***Abstract*—There are more bugs in real-world programs than human programmers can realistically address. This paper evaluates two research questions: "What fraction of bugs can be repaired automatically?" and "How much does it cost to repair a bug automatically?" In previous work, we presented *GenProg*, which uses genetic programming to repair defects in off-the-shelf C programs. To answer these questions, we: (1) propose novel algorithmic improvements to GenProg that allow it to scale to large programs and find repairs 68% more often, (2) exploit GenProg's inherent parallelism using cloud computing resources to provide grounded, human-competitive cost measurements, and (3) generate a large, indicative benchmark set to use for systematic evaluations. We evaluate GenProg on 105 defects from 8 open-source programs totaling 5.1 million lines of code and involving 10,193 test cases.**

patch overflow and illegal control-flow transfer vulnerabilities; AutoFix-E [9], which can repair programs annotated with design-by-contract pre- and post-conditions; and AFix [10], which can repair single-variable atomicity violations. In previous work, we introduced *GenProg* [11], [12], [13], [14], a general method that uses genetic programming (GP) to repair a wide range of defect types in legacy software (e.g., infinite loops, buffer overruns, segfaults, integer overflows, incorrect output, format string attacks) without requiring *a priori* knowledge, specialization, or specifications. GenProg searches for a repair that retains required functionality by constructing variant programs through computational analogs of biological processes.

A Systematic Study of Automated Program Repair: Fixing 55 out of 105 Bugs for $8 Each. Le Goues et al. ICSE 2012.

Snapshots

VM state can be saved with little overhead.

- Recreation of bugs.
- Useful for testing.
- Automated bug-fixing?

High Impact

Small improvements matter.

Gains can be rolled out across large numbers of systems.

Potential SBSE Applications

Amazon Machine Images (AMIs)

An Amazon Machine Image (AMI) is a special type of pre-configured operating system and virtual application software which is used to create a virtual machine within the Amazon Elastic Compute Cloud (EC2). It serves as the basic unit of deployment for services delivered using EC2.

Read the Amazon EC2 Developer Guide for information on safely using shared AMIs.

awsmarketplace

AWS Marketplace

The AWS Marketplace is the premier source for AMIs with more than 500 popular free and commercial AMIs in 25 categories. AWS Marketplace AMIs have been pre-configured to launch on AWS with 1-Click® and have been screened for security.

Showing 1-25 of **1675 AMIs** Sort by: Date – newest first

ubuntu

PostgreSQL8.4 PostGIS 1.5 Ubuntu 12.04

PostgreSQL8.4 PostGIS 1.5 Ubuntu 12.04

Listed on Aug 20, 2013 23:19 GMT

Launch AMI

› More Info

ubuntu

BOINC Server and Climate at Home Ubuntu 12.04

BOINC Server and Climate at Home, Ubuntu

Listed on Aug 20, 2013 23:18 GMT

Launch AMI

› More Info

ubuntu

ubuntu-precise-docker

Ubuntu 12.04 LTS preinstalled with Docker (docker.io)

Listed on Aug 20, 2013 23:18 GMT

Launch AMI

› More Info

DPS Notification Station

http://aws.amazon.com/amis

Shrinking VM Size

Reduce VM Size: shorter boot, copy, migration times. Smaller instances.

Can we automate VM Image specialisation?

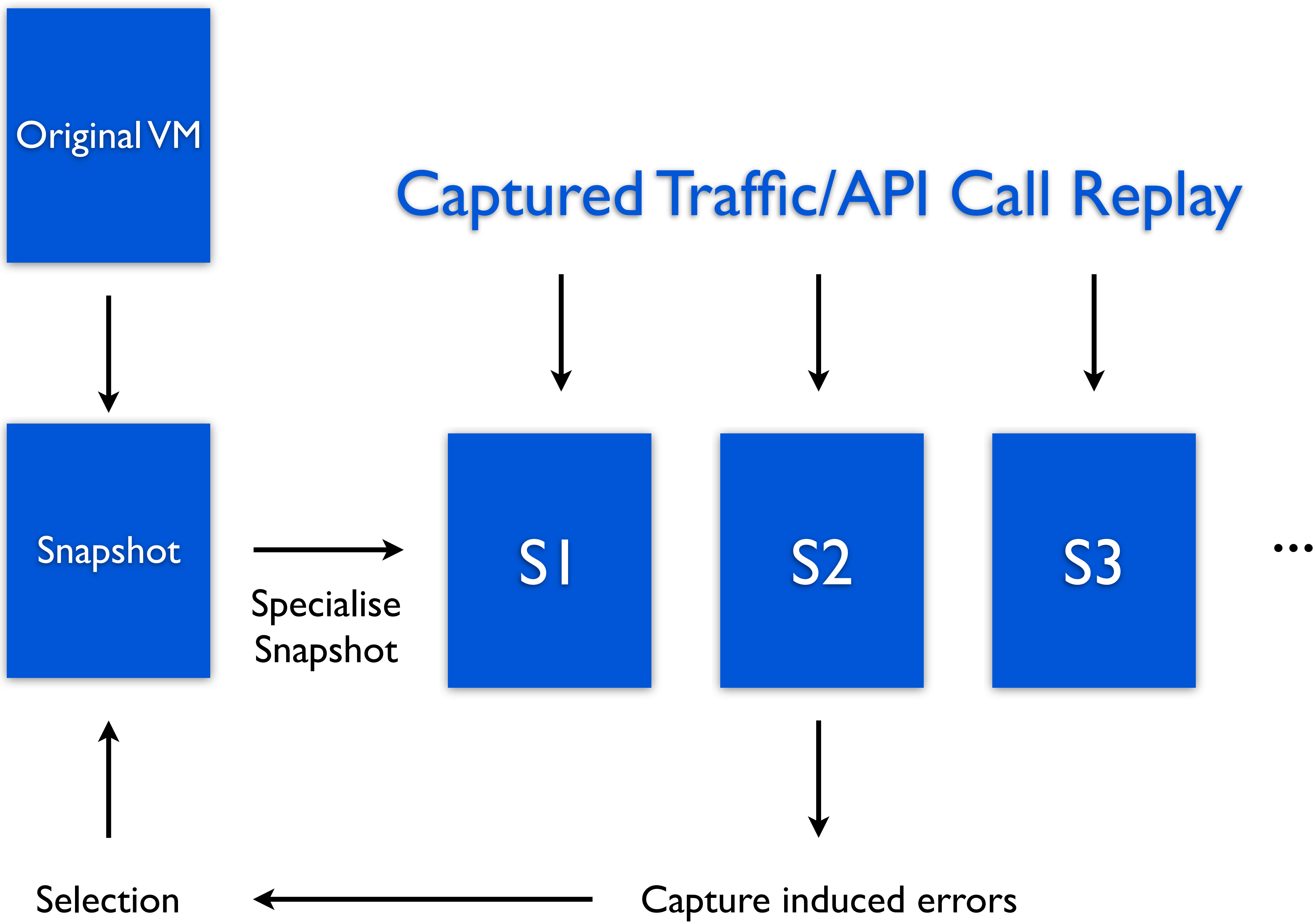
Cumulative benefits.

Package Removal

Search package manager dependency graph to find potential removals.

Essentially a subset selection problem.

Objectives: image size, memory usage.


Original VM
Captured Traffic/API Call Replay
Snapshot
Specialise Snapshot
S1
S2
S3
...
Selection
Capture induced errors

Optimise Scale Management

Reduce instance usage; improve responsiveness.

How do I anticipate and react to demand?

Scale Management - MS Azure

```
<constraintRules>
    <rule name=”Peak” enabled=”true” rank=”100”>
        <actions>
            <range min=”4” max=”4” target=”RoleA”/>
        </actions>
        <timetable startTime=”08:00:00” duration=”02:00:00”>
            <daily/>
        </timetable>
    </rule>
</constraintRules>
```

Scheduled Scaling

Scale Management - MS Azure

```
<constraintRules>
    <rule name=”Peak” enabled=”true” rank=”100”>
        <actions>
            <range min=”4” max=”4” target=”RoleA”/>
        </actions>
        <timetable startTime=”08:00:00” duration=”02:00:00”>
            <daily/>
        </timetable>
    </rule>
</constraintRules>
```

Scheduled Scaling

```
<rule name=”Example Scaling Rule” rank=”100”>
    <when>
        <greater operand=”CPU_RoleA” than=”80” />
    </when>
    <actions>
        <scale target=”WorkerRoleA” by=”2”/>
    </actions>
</rule>
```

Response to fluctuations

Tuning Scaling

Search for scaling policies using Genetic Programming or Grammatical Evolution.

Objectives: reduce expenditure, improve responsiveness.

Search-Based Genetic Optimization for Deployment and Reconfiguration of Software in the Cloud

Sören Frey, Florian Fittkau, and Wilhelm Hasselbring
Software Engineering Group
Kiel University
24118 Kiel, Germany
{sfr, ffi, wha}@informatik.uni-kiel.de

***Abstract*—Migrating existing enterprise software to cloud platforms involves the comparison of competing cloud deployment options (CDOs). A CDO comprises a combination of a specific cloud environment, deployment architecture, and runtime reconfiguration rules for dynamic resource scaling. Our simulator CDOSim can evaluate CDOs, e.g., regarding response times and costs. However, the design space to be searched for well-suited solutions is extremely huge. In this paper, we approach this optimization problem with the novel genetic algorithm CDOXplorer. It uses techniques of the search-based software engineering field and CDOSim to assess the fitness of CDOs. An experimental evaluation that employs, among others, the cloud environments Amazon EC2 and Microsoft Windows Azure, shows that CDOXplorer can find solutions that surpass those of other state-of-the-art techniques by up to 60%. Our experiment code and data and an implementation of CDOXplorer are available as open source software.**

***Index Terms*—Cloud computing, Search-based software engineering, Deployment optimization**

Furthermore, as we also intend to support distributed systems, we consider the QoS-aware composition of software components that run on one node as a single service that is provided to an arbitrary number of components on other nodes. Such deployment optimization problems are intractable as they are known to be NP-hard [4]. In our previous work, we introduced the simulation tool CDOSim [5] that implements a phase of our cloud migration approach CloudMIG [6, 7]. CDOSim facilitates the simulation of CDOs for determining their respective response times, costs, and SLA violations.

We integrated CDOSim into our tool CloudMIG Xpress that provides support for CloudMIG. With CloudMIG Xpress, CDOs can be manually configured and simulated on the basis of a reverse-engineered architectural system model with monitored or synthetic workload. However, the design space that spans for all possible CDOs is huge, the elements of a single CDO exhibit complex non-linear interdependencies, and

Search-Based Genetic Optimization for Deployment and Reconfiguration of Software in the Cloud. Frey et al. ICSE 2013.

Virtual Machine Placement

Much work done on this already, but still early: e.g. considering network traffic patterns.

Implementing Scalable, Network-Aware Virtual Machine Migration for Cloud Data Centers

Fung Po Tso*, Gregg Hamilton*, Konstantinos Oikonomou†, and Dimitrios P. Pezaros*
*School of Computing Science, University of Glasgow, G12 8QQ, UK
†Department of Informatics, Ionian University, 49100 Corfu, Greece
Email: {posco.tso, dimitrios.pezaros}@glasgow.ac.uk, okon@ionio.gr, g.hamilton.3@research.gla.ac.uk

***Abstract*—Virtualization has been key to the success of Cloud Computing through the on-demand allocation of shared hardware resources to Virtual Machines (VM)s. However, the network-agnostic placement of VMs over the underlying network**

Experiments over Amazon's EC2 revealed that a marginal 100 msec additional latency resulted in 1% drop in sales, while Google's revenues dropped by 20% due to a 500 msec increase

Implementing Scalable, Network-Aware Virtual Machine Migration for Cloud Data Centers. Tso et al. IEEE Cloud 2013.

Memory Optimisation

Managing the use of RAM, paging and network memory between competing processes.

Overdriver: Handling Memory Overload in an Oversubscribed Cloud

Dan Williams * Hakim Weatherspoon
Dept. of Computer Science
Cornell University, Ithaca, NY
{djwill, hweather}@cs.cornell.edu

Hani Jamjoom Yew-Huey Liu
IBM T. J. Watson Research Center, Hawthorne, NY
{jamjoom,yhliu}@us.ibm.com

Abstract

With the intense competition between cloud providers, oversubscription is increasingly important to maintain profitability. Oversubscribing physical resources is not without consequences: it increases the likelihood of overload. Memory overload is particularly damaging. Contrary to traditional views, we analyze current

in the form of virtual machines. With this trend, effective usage of data center resources is becoming increasingly important. A cloud provider using the classical model of overprovisioning each VM with enough physical resources to support relatively rare peak load conditions will have trouble competing with one that can provide similar service guarantees using less resources. This suggests an opportunity for cloud providers to oversubscribe data center

Overdriver: handling memory overload in an oversubscribed cloud. Williams et al. VEE 2011.

Network Specialisation

Network no longer needs to be generic: design for datacentre needs.

e.g. modifying TCP/IP parameters (minor), multipath routing (major).

Ultimately: Software Defined Networking.

Measurement-Based TCP Parameter Tuning in Cloud Data Centers

Simon Jouet and Dimitrios P. Pezaros
School of Computing Science, University of Glasgow
s.jouet.1@research.gla.ac.uk, dimitrios.pezaros@glasgow.ac.uk

***Abstract*—TCP congestion control has been a native part of all modern Operating System implementations where parameters are initialized assuming an underlying high Bandwidth Delay Product (BDP) environment. However, the significantly lower BDP in Data Centre (DC) networks makes such conservative transport-layer parameters together with deep-buffered switches and bursty traffic a factor of performance degradation, eventually leading to throughput incast collapse.**

In this paper, we propose a Software Defined Networking (SDN) approach to tune TCP initial window and retransmission timers for newly created flows based on a network-wide view created by aggregating known characteristics and temporal measurements at a central controller. Through simulation, we show the detrimental effect static TCP parameters have on mice flows and demonstrate the benefits of network-aware per-flow tuning. We show that the average latency under bursty traffic can be improved by a factor of eight, and that flow start and completion times can be improved by a factor of two and five, respectively.

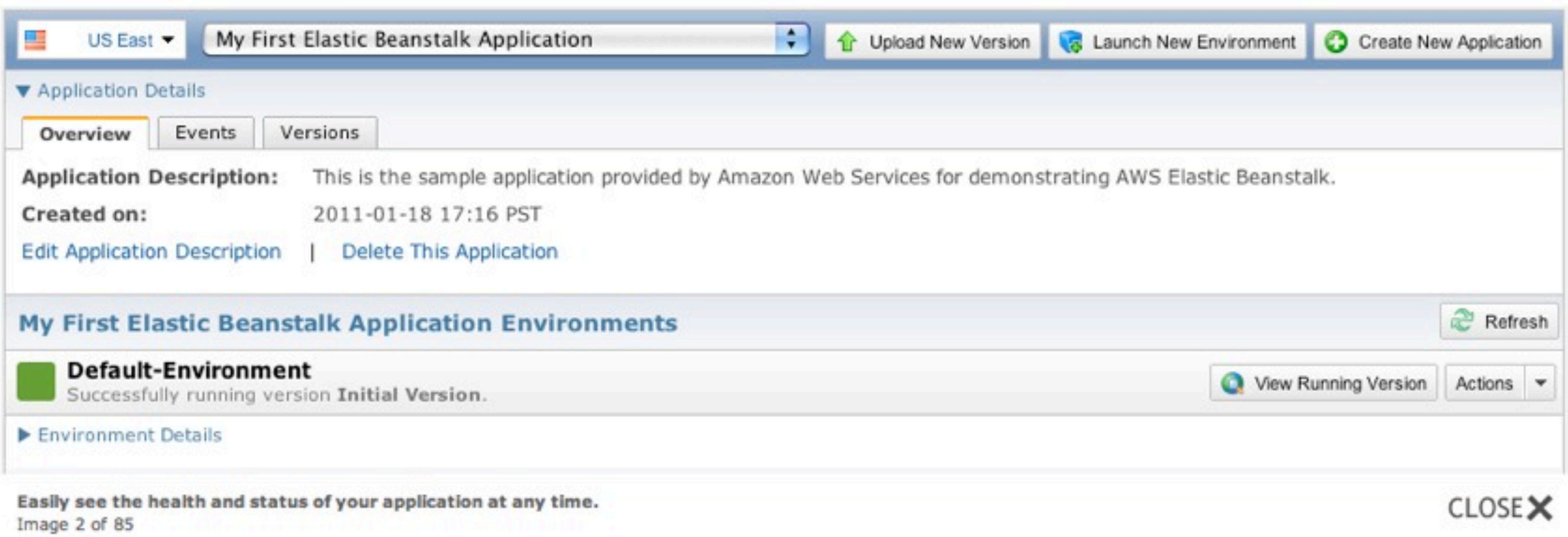
are active as the window size should reflect the Bandwidth Delay Product (BDP) of the network. This discrepancy in the parameters relates to the fact that Internet mechanisms have been designed for Long Fat Pipes (LFPs), network links with long latency and high bandwidth, while in DCs the sub-millisecond link latencies results in much smaller BDP. With 80% of the traffic staying inside cloud DCs, it becomes evident that such traffic should be optimized accordingly [5].

The initial parameters of TCP connections are fixed as part of the OS implementation regardless of the destination. This approach has worked over the Internet due to two factors: the parameters are conservative and the slow-start phase is used to detect the throughput of the bottleneck link. However, DCs have different requirements, the traffic is dominated by mice flows, small in size and duration which requires fast response times. Amazon, Google and Microsoft showed a loss

Measurement-Based TCP Parameter Tuning in Cloud Data Centres.
Jouet and Pezaros. To appear in ICNP 2013.

4. Challenges

Black Box Computing



Replicability

Noisy neighbours.

Heterogeneous hardware; hardware faults.

Empirical Evaluation

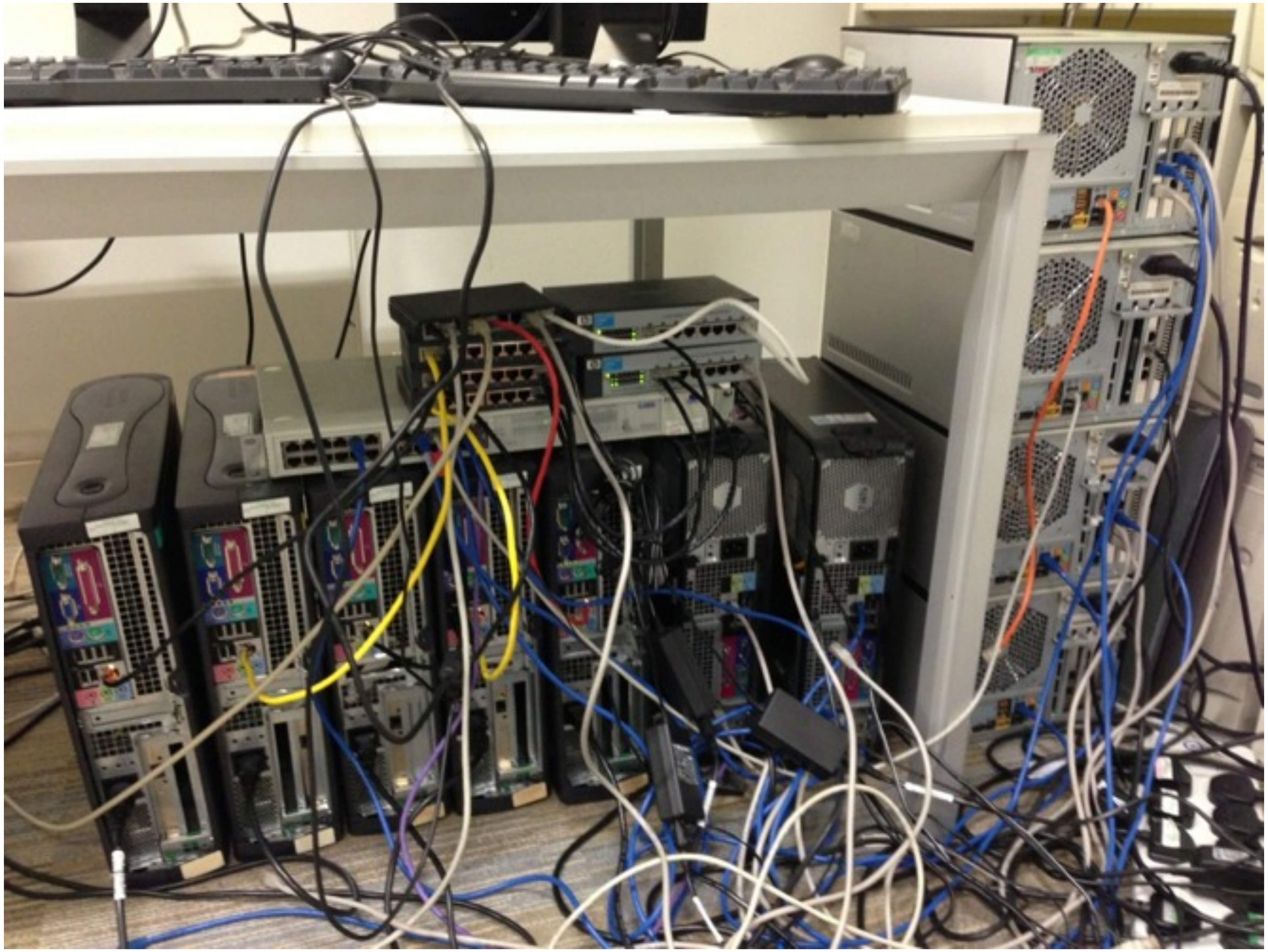
Simulation:

- Cloud Simulators, CloudSim (Java), BigHouse
- Networks - NS3.

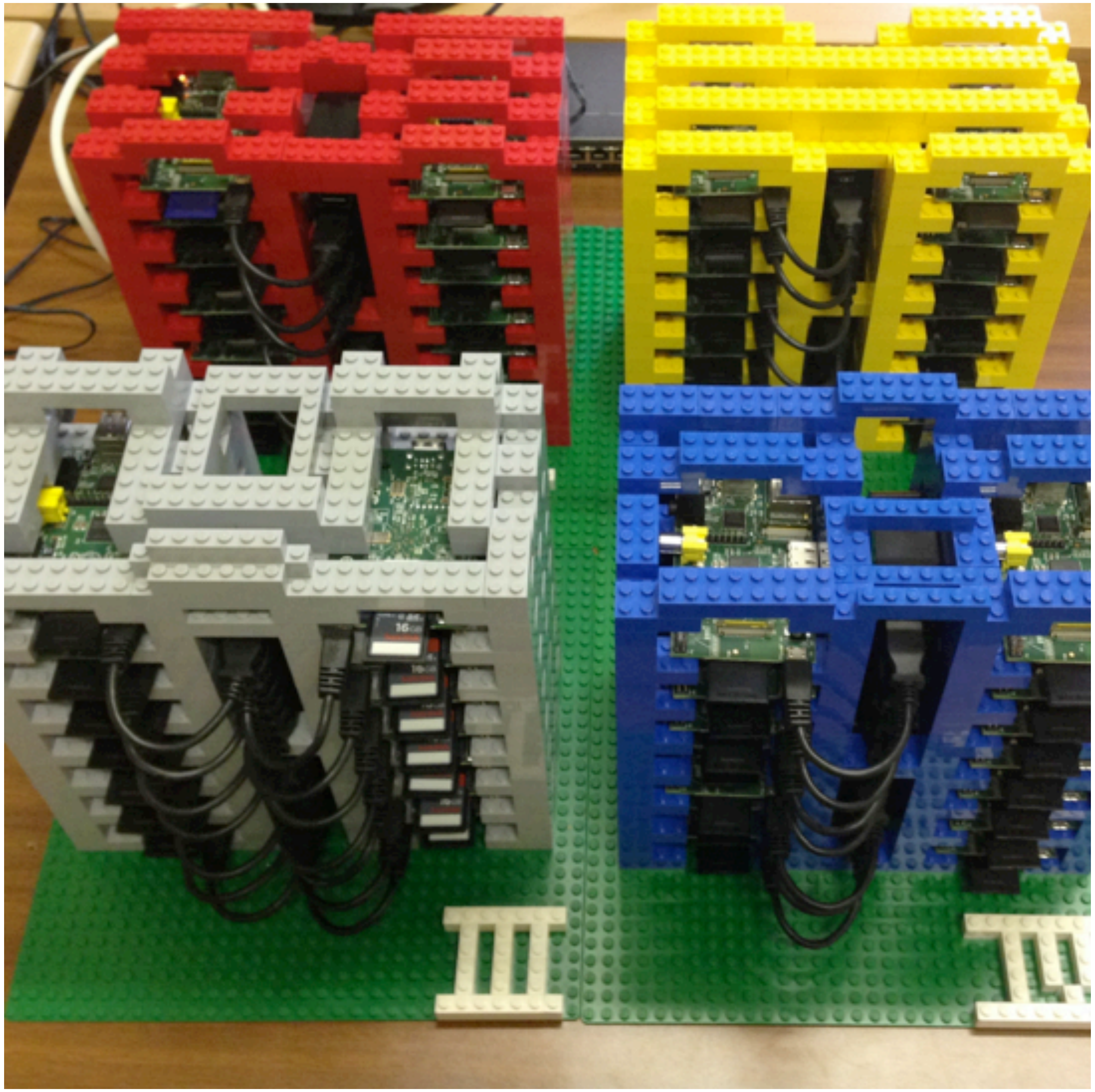
Testbeds:

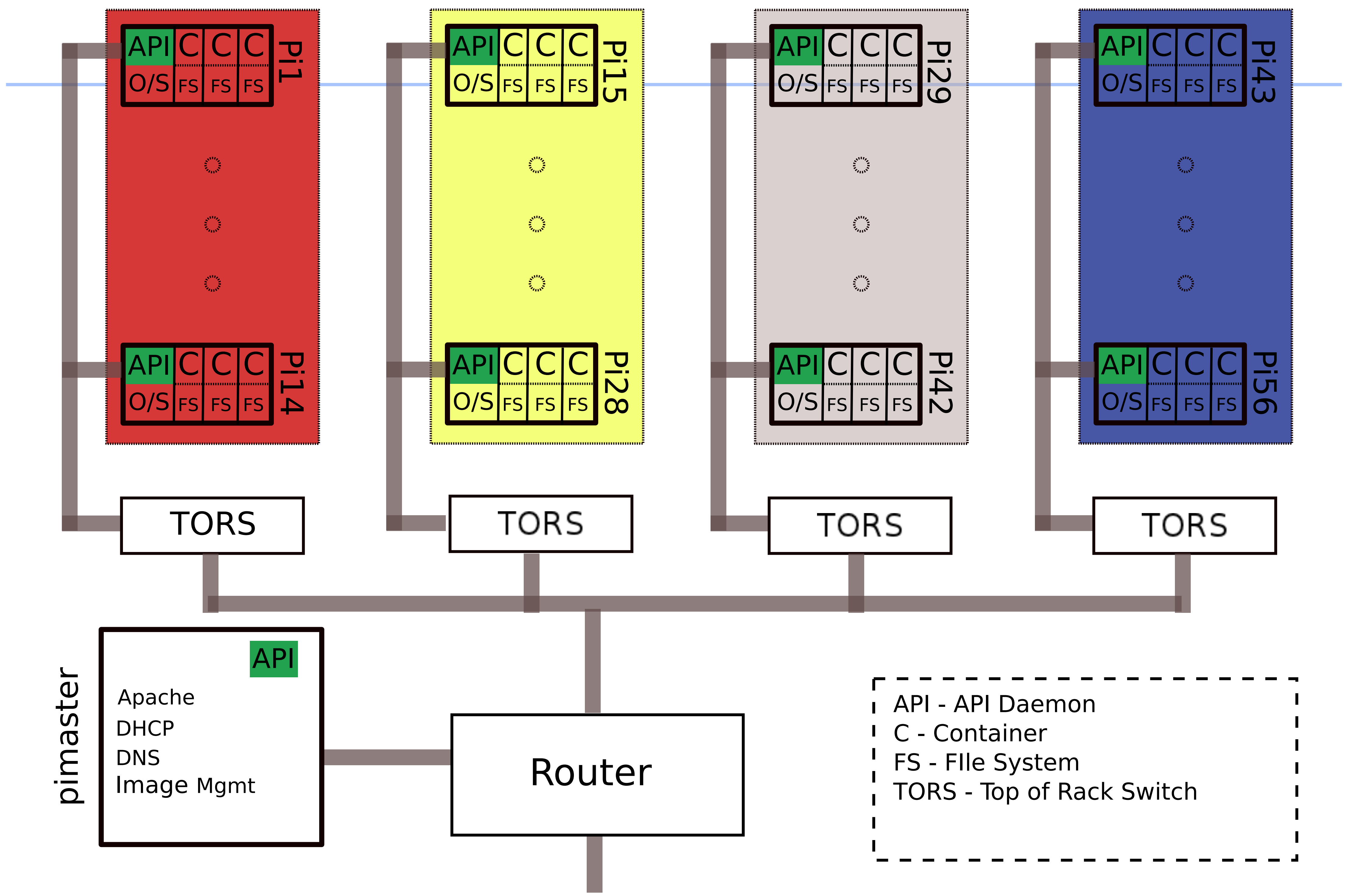
- x86 testbeds.
- New approach at Glasgow: 'scale model' testbeds.

x86 Testbeds

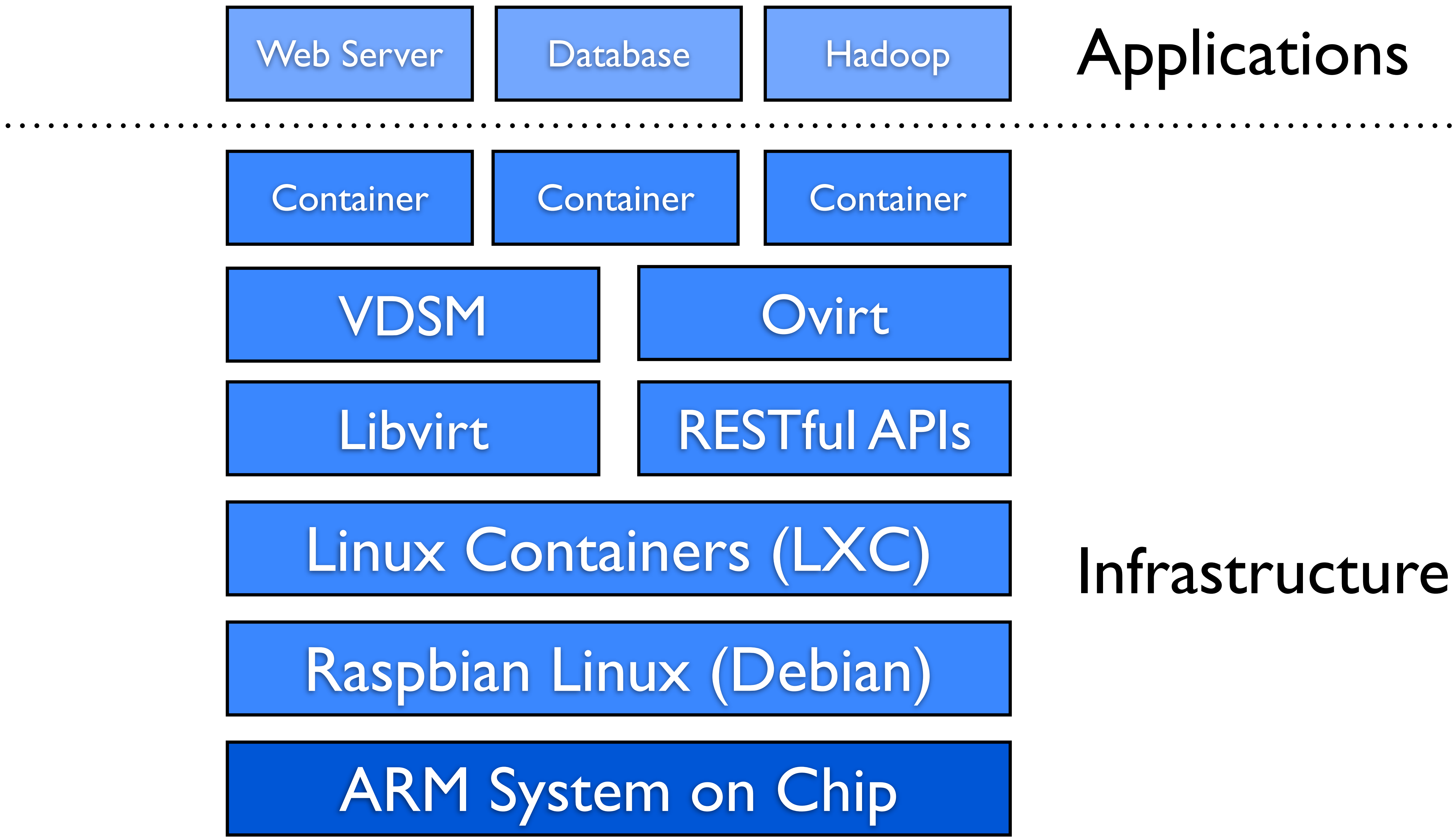


Raspberry Pi Cloud




API
C
C
C
O/S
FS
FS
FS
Pi1
Pi14
Pi15
Pi28
Pi29
Pi42
Pi43
Pi56
TORS
TORS
TORS
TORS
pimaster
API
Apache
DHCP
DNS
Image Mgmt
Router
API - API Daemon
C - Container
FS - FIle System
TORS - Top of Rack Switch

Raspberry Pi Cloud Software Stack



Raspberry Pi Cloud

http://raspberrypicloud.wordpress.com/

The Glasgow Raspberry Pi Cloud: A Scale Model for Cloud Computing Infrastructures.
Tso et al.
International Workshop on Resource Management of Cloud Computing 2013.

Workloads and Data

Cloud providers are notoriously secretive about their infrastructure and workloads.

Possibly the biggest challenge to Cloud empirical work.

There is some data available.

Workloads and Data

2013 IEEE Seventh International Symposium on Service-Oriented System Engineering

An Approach for Characterizing Workloads in Google Cloud to Derive Realistic Resource Utilization Models

Ismael Solis Moreno, Peter Garraghan, Paul Townend, Jie Xu
School of Computing
University of Leeds
Leeds, UK
{scism, scpmg, p.m.townend, j.xu} @ leeds.ac.uk

***Abstract*- Analyzing behavioral patterns of workloads is critical to understanding Cloud computing environments. However, until now only a limited number of real-world Cloud datacenter tracelogs have been available for analysis. This has led to a lack of methodologies to capture the diversity of patterns that exist in such datasets. This paper presents the first large-scale analysis of real-world Cloud data, using a recently released dataset that features traces from over 12,000 servers over the period of a month. Based on this analysis, we develop a novel approach for characterizing workloads that for the first time considers Cloud workload in the context of both user and task in order to derive a model to capture resource estimation and utilization patterns. The derived model assists in understanding the relationship between users and tasks within workload, and enables further work such as resource optimization, energy-efficiency improvements, and failure correlation. Additionally, it provides a mechanism to create patterns that randomly fluctuate based on realistic parameters. This is critical to emulating dynamic environments instead of statically replaying records in the tracelog. Our approach is evaluated by contrasting the logged data against simulation experiments, and our results show that the derived model parameters correctly describe the operational environment within a 5% of error margin, confirming the great variability of patterns that exist in Cloud computing.**

As a result of business and confidentiality concerns, there has been a lack of available data from real Cloud operational environments to analyze. Recently, due to the publication of limited traces from Google [2] and Yahoo! [3], there has been an increasing effort to provide mechanisms to characterize workload dynamicity. However first efforts were strongly constrained by traces with very short observational periods [4]. Analyses and methodologies derived from just a few hours of production data are diminished by the uncertainty generated from the lack of realistic scenarios. Others that have had access to private large datasets introduce methodologies of analysis based on coarse-grain statistics [4, 5], which are appropriate to reveal general characteristics of the operational environment but not sufficient to describe and characterize the workload diversity that is generated in Cloud environments. Finally, more recent approaches [6, 7] have attempted to capture this diversity by classifying the different types of tasks discovered in the data.

Presently there is a lack of research that deeply analyses and models the relationship between users, tasks and their associated characteristics as an integrated concept of a workload in the Cloud environment. This is an extremely important factor to consider, as the volume and the behavior of tasks that exist within Cloud environments are driven by

An Approach for Characterizing Workloads in Google Cloud to Derive Realistic Resource Utilization Models. Moreno et al. SOSE 2013.

Online Optimisation

Testing, debugging, fixing, improving live software is an obvious SBSE application.

Much more dynamic environment than previous SBSE applications.

Dynamic Adaptive Search Based Software Engineering. Harman et al. ESEM 2012.

Research Note

RN/12/13

NIA^{3}CIN: Non-Invasive Autonomous and Amortised Adaptivity Code Injection

November, 2012

Shin Yoo

Telephone: +44 (0)20 3108 5032
Fax: +44 (0)171 387 1397
Electronic Mail: s.yoo@cs.ucl.ac.uk
URL: http://www.cs.ucl.ac.uk/staff/S.Yoo

Abstract

Search Based Software Engineering has high potential for optimising non-functional properties such as execution time or power consumption. However, many non-functional properties are dependent not only on the software system under consideration but also the environment that surrounds the system. This results in two problems. First, systems optimised in offline environment may not perform as well online. Second, the system needs to be taken offline and re-optimised when the environment changes. This paper introduces the novel concept of amortised optimisation to solve both problems, and presents an open source implementation. We evaluate the framework to optimise block matrix multiplication algorithm.

https://bitbucket.org/ntrolls/niacin

Summary

An introduction to Cloud Computing and the underlying infrastructure.

Many problems in engineering the cloud can be expressed as optimisation problems.

There are exciting possibilities (and challenges!) for SBSE.

“Testing software online is a major problem.”

Jim Larus
Principal Researcher, MS Research.

PLDI Keynote.
June 2013.

Repairing
Refactoring software online is a major problem."
Optimising

Jim Larus
Principal Researcher, MS Research.

PLDI Keynote.
June 2013.

Acknowledgements

Mark Harman, Jeremy Singer, Kiran Lakhotia, Shin Yoo.

Posco Tso, Dimitrios Pezaros, Simon Jouet.

david.r.white@glasgow.ac.uk