Next Generation Clouds

... and beyond ...

Rolando Martins

CRACS/INESC TEC

February 8, 2017

About Me - I

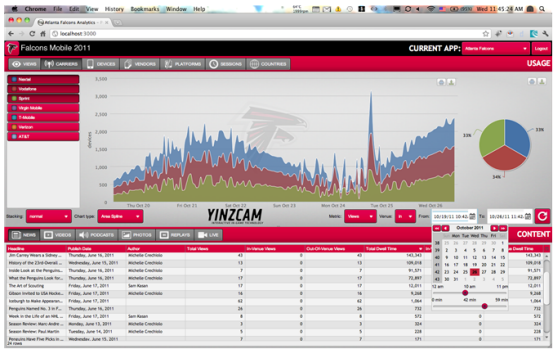
efacec

FACULDADE DE CIÊNCIAS
UNIVERSIDADE DO PORTO

About Me - II

Carnegie Mellon University

YINZCAM



Talk Outline

1. Part I - Classical Cloud-Computing
   - What is "the Cloud"?
   - Why should you care?
   - Types of Clouds
   - Private Cloud - OpenStack & Containers
2. Some Cloud Components That You Will Need
   - Distributed Storage
   - Middleware systems
   - Computational Frameworks
   - Failure Diagnosis
3. Part II - Hands on Cloud Computing
   - Stories from a SysOps
4. Part III - Edge Computing
   - Hyrax
   - Cloudlets

What is "the Cloud"?

- Is it Clustering with a different name?
  - Not really! Althought it has common ground...
  - Neither is a Cluster++. It is meant to support multi-resource sharing across multi-tenants.
- Where, Whom, When and Why?
  - 'Merica, Amazon, 2006, sell over-capacity to make $$$.
  - Currently approaching a $10 billion-a-year business...

Why should you care?

- For entrepreneurs:
  - Access to cheap and almost limitless resources
  - Minimal ramp-up time
- For companies:
  - Cut operational costs
  - leverage transient peak seasons (X-Mas,...)
- **For researchers**:
  - Study problems at scale
  - Immediate social and economical impact

One Cloud abstraction to rule them all?

- Multiple abstractions are presently offered (*aaS), namely:
  - PaaS (Platform as a Service):
    - ★ Appscale, Google App Engine
  - SaaS (Software as a Service):
    - ★ Gmail, OneDrive
  - **IaaS (Infrastructure as a Service)**:
    - ★ Amazon AWS, Google Cloud, Azure, Rackspace, OpenStack

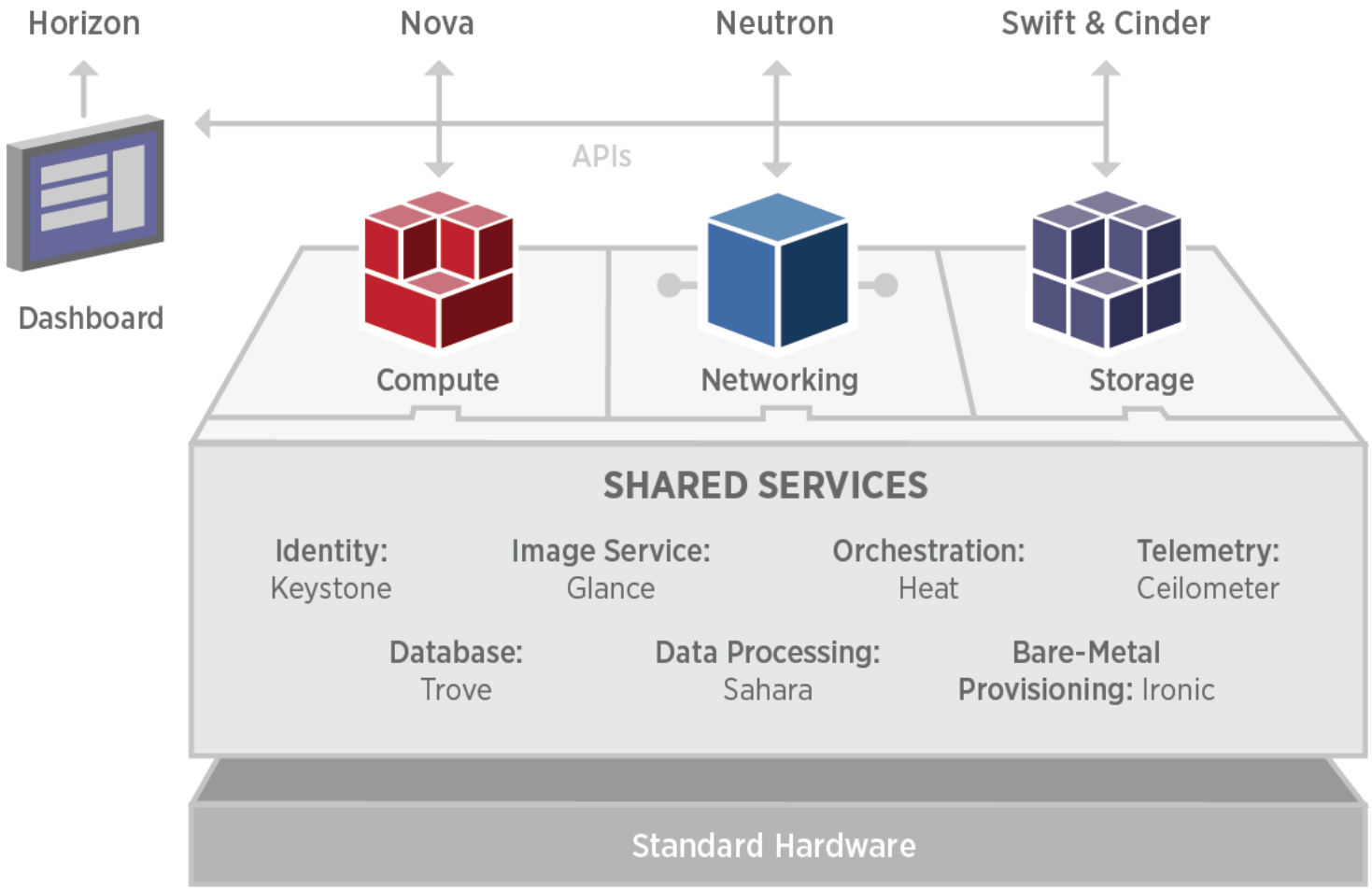
Basic resources of an IaaS

- Computation provided by instances, e.g., EC2 instances.
  - Segregated by regions, normally backed by a distinct data-center
- Storage provided in different models:
  - Volume service that provides strongly consistency, e.g., EBS (Amazon Block Store)
  - NoSQL backed service with eventual consistency (optional but useful), e.g., S3
- DNS management (Route 53)
- SSL termination + load balancers

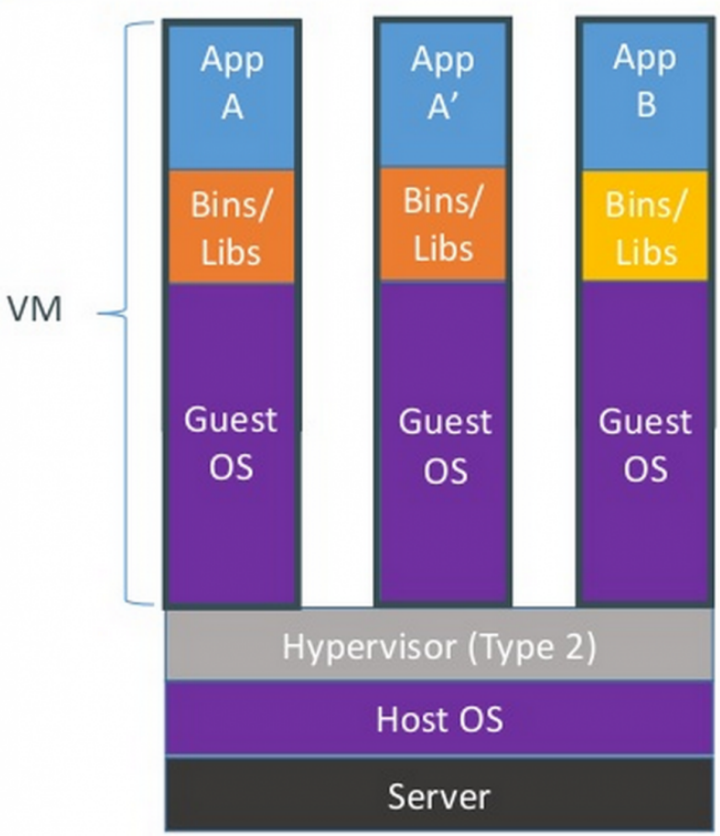
Private, Public and Hybrid Clouds

- Public Clouds:
  - Amazon AWS is the most used cloud provider
- Private Clouds:
  - OpenStack is the most widely deployed and known
- Hybrid Clouds = *Private* + *Public*
  - Typically an institution has its core services and data in-house and uses the public cloud for long running batch jobs

Deploying Private Clouds - OpenStack Architecture

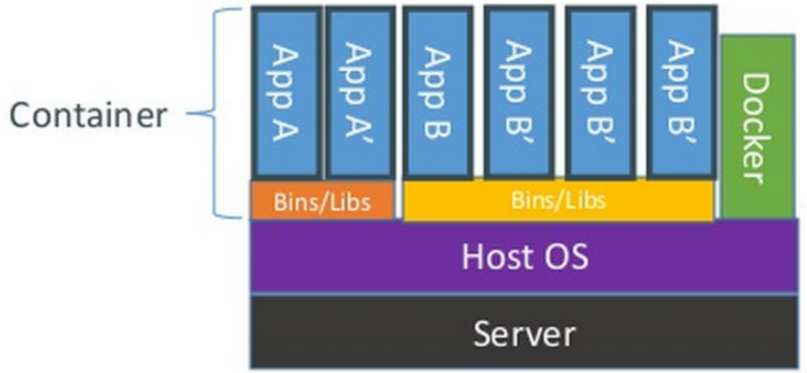


VMs vs Containers

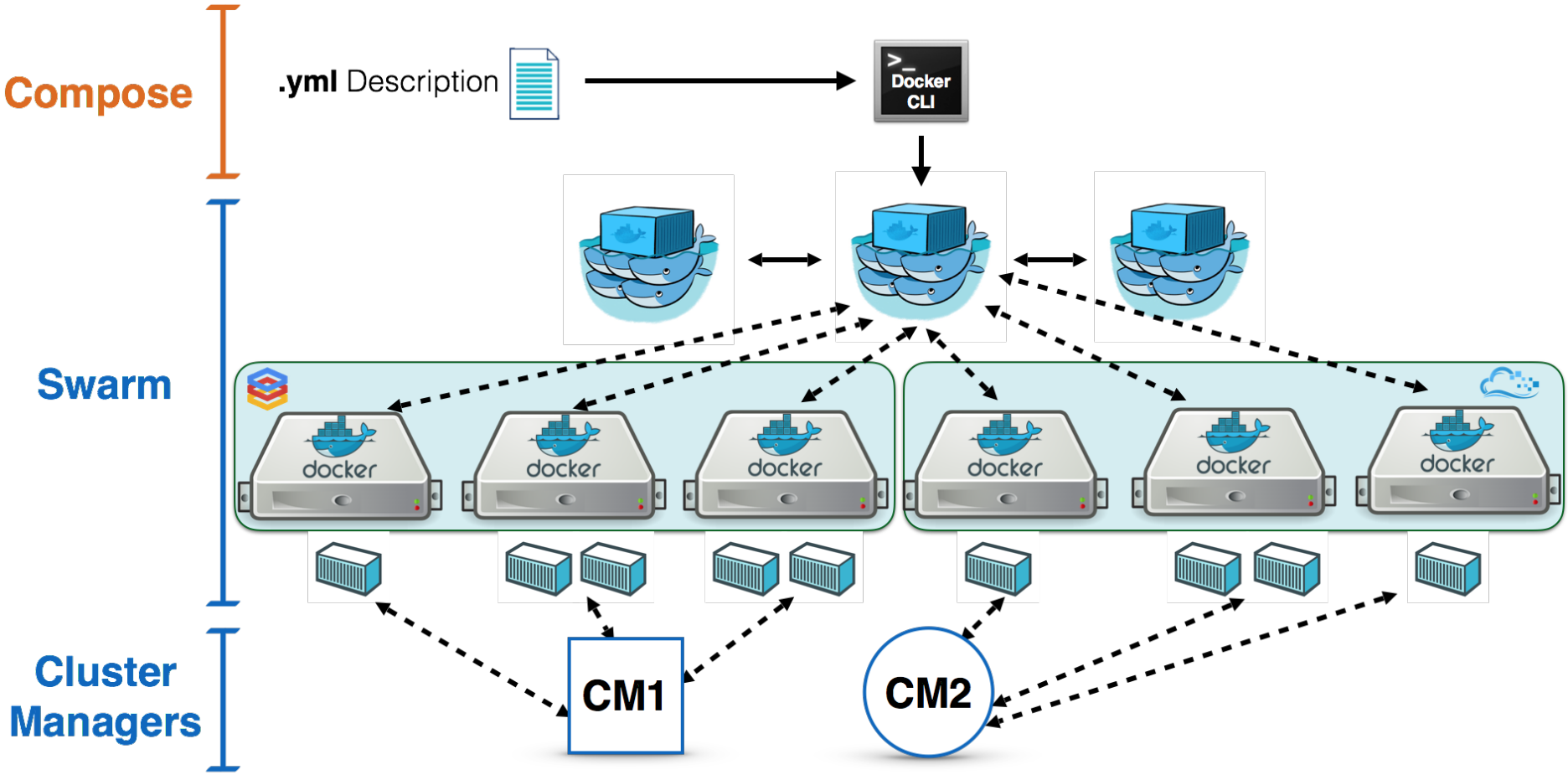


Containers are isolated, but share OS and, where appropriate, bins/libraries

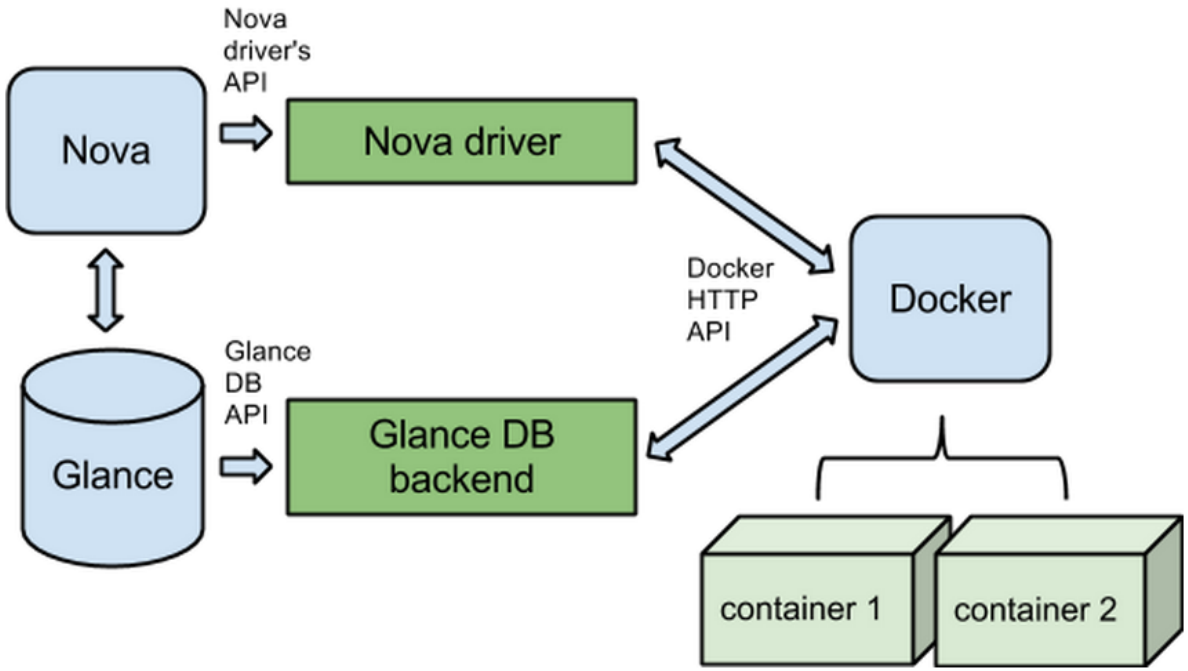
...result is significantly faster deployment, much less overhead, easier migration, faster restart



Docker Architecture



Meshing OpenStack and Docker



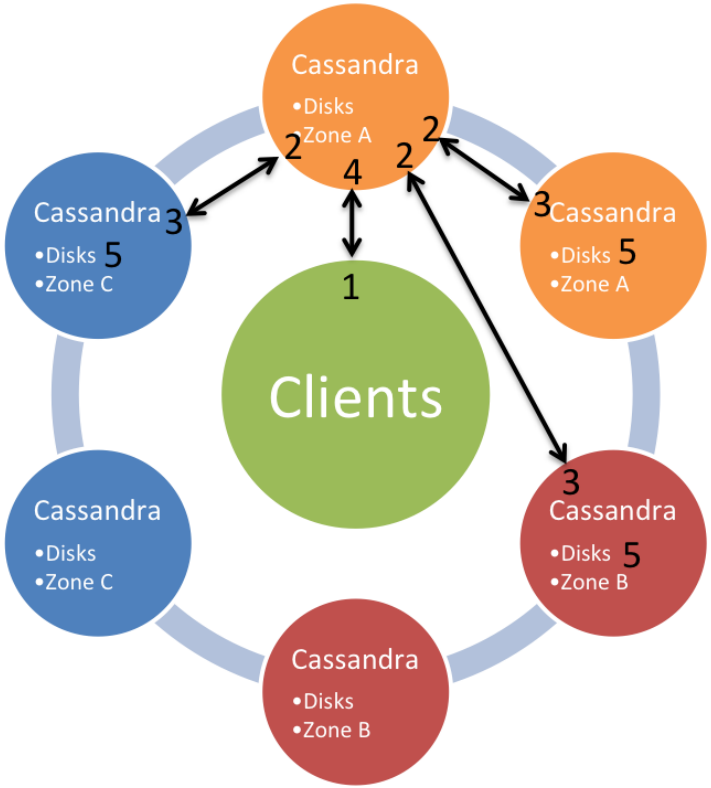
Cloud Service/Components That you will Need

Distributed Storage - DHT

Cassandra Write Data Flows

Single Region, Multiple Availability Zone

1. Client Writes to any Cassandra Node
2. Coordinator Node replicates to nodes and Zones
3. Nodes return ack to coordinator
4. Coordinator returns ack to client
5. Data written to internal commit log disk



If a node goes offline, hinted handoff completes the write when the node comes back up.

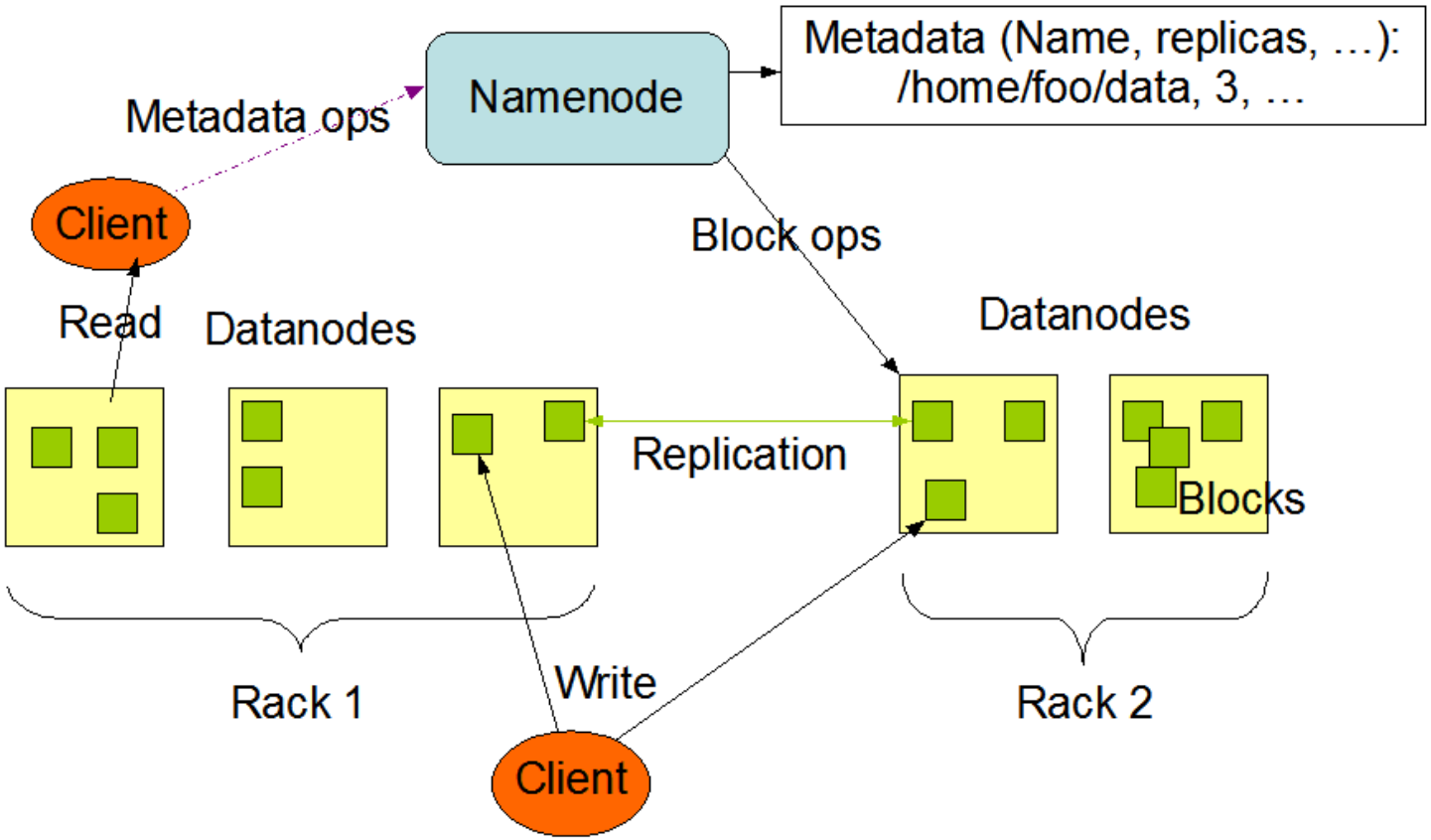
Requests can choose to wait for one node, a quorum, or all nodes to ack the write

SSTable disk writes and compactions occur asynchronously

NETFLIX

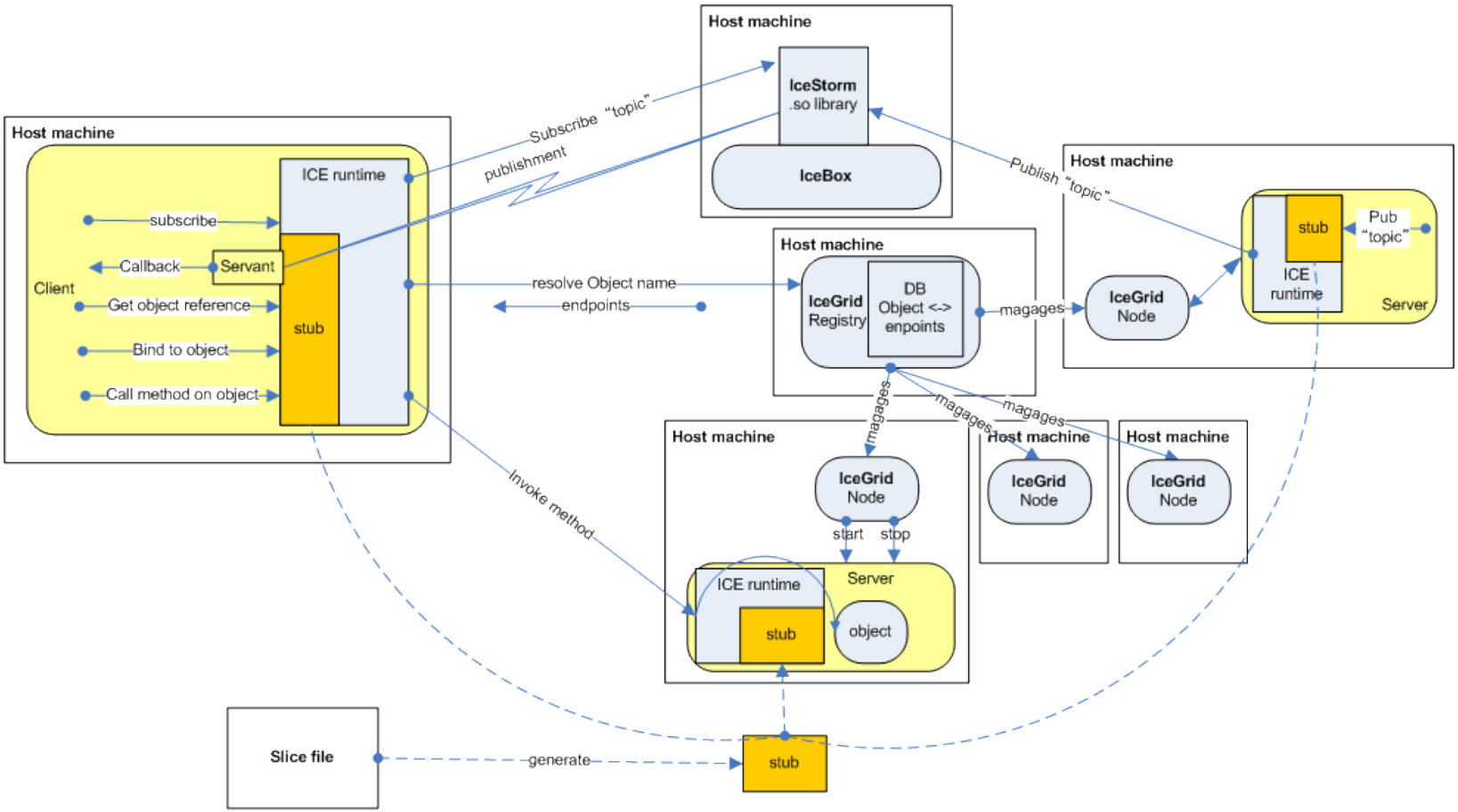
Distributed Storage - Distributed FileSystem

- HDFS (inspired by Google File System (GFS))



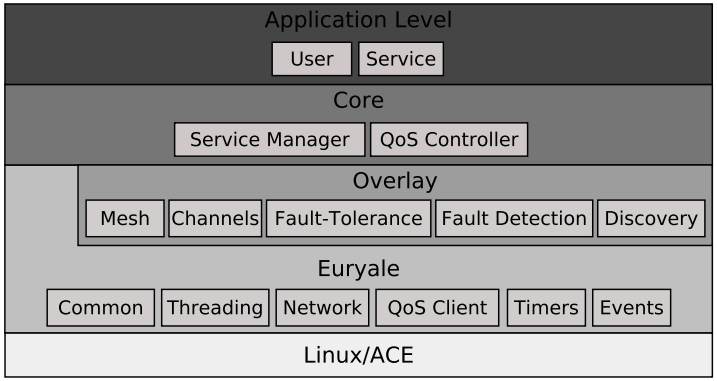
General Purpose Middleware

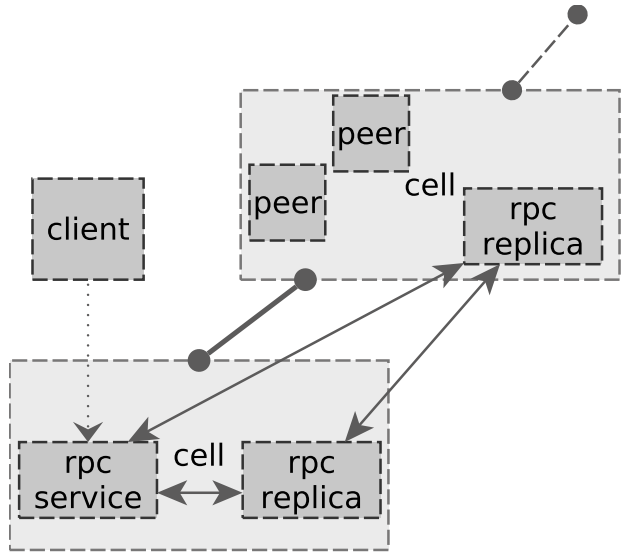
- ICE

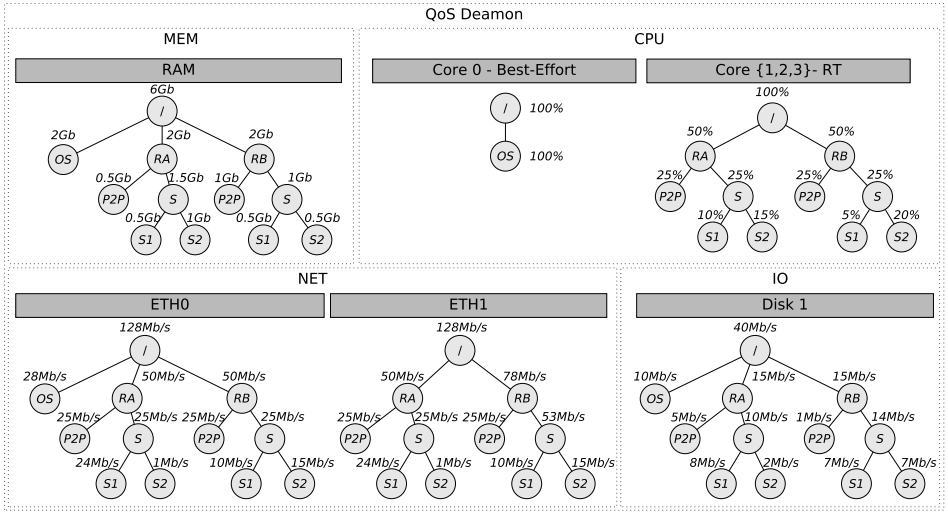


Real-time Fault-tolerant Middleware

- Stheno







Stheno - RPC Service Example

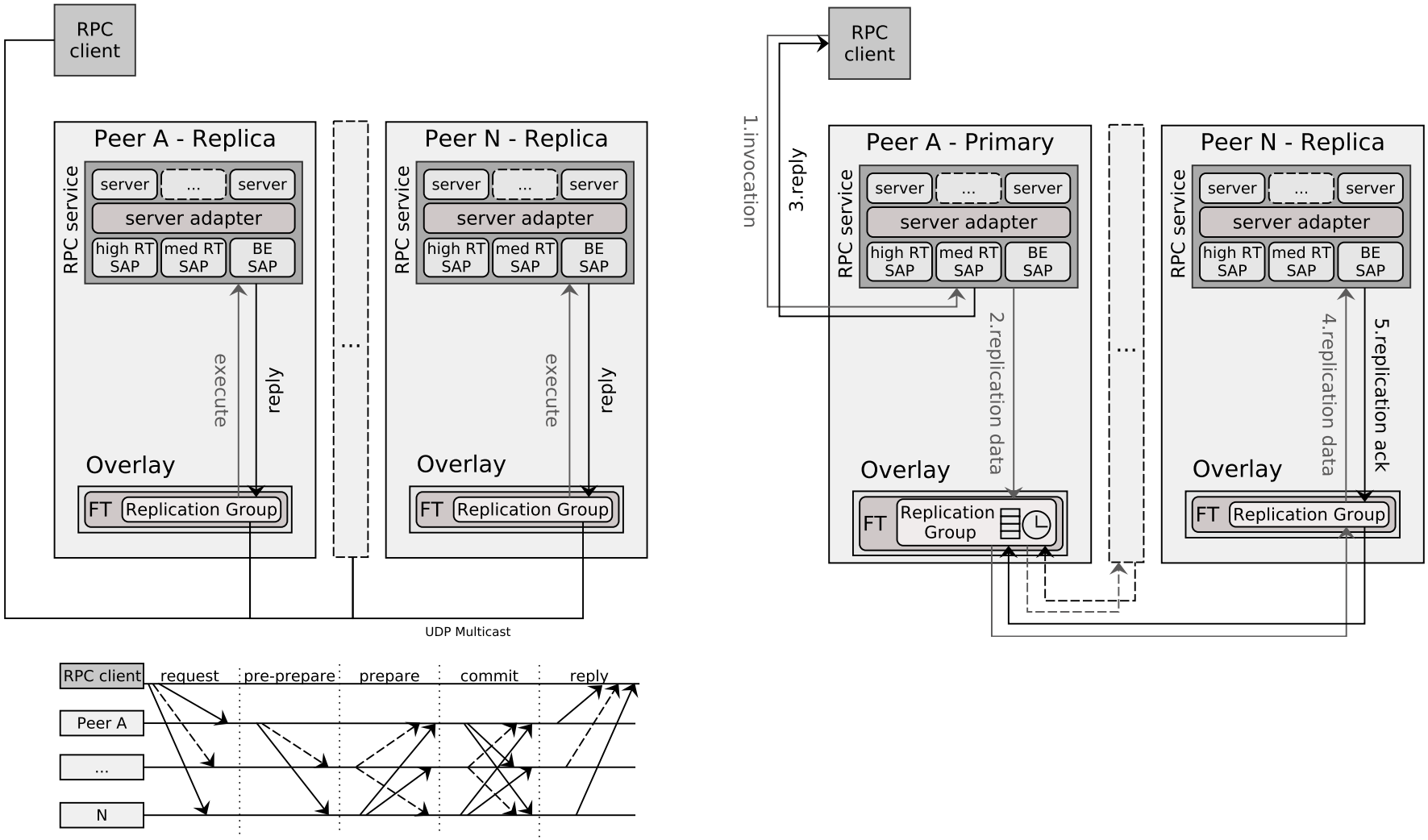
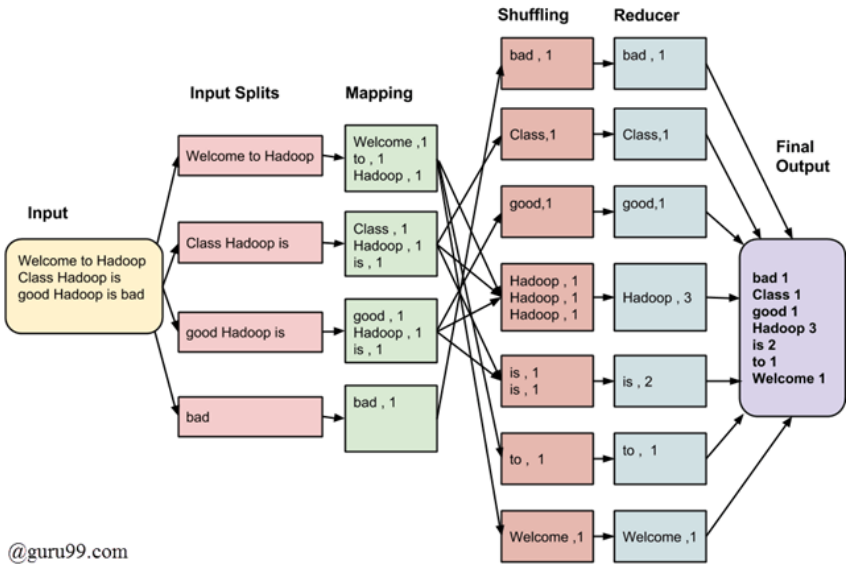
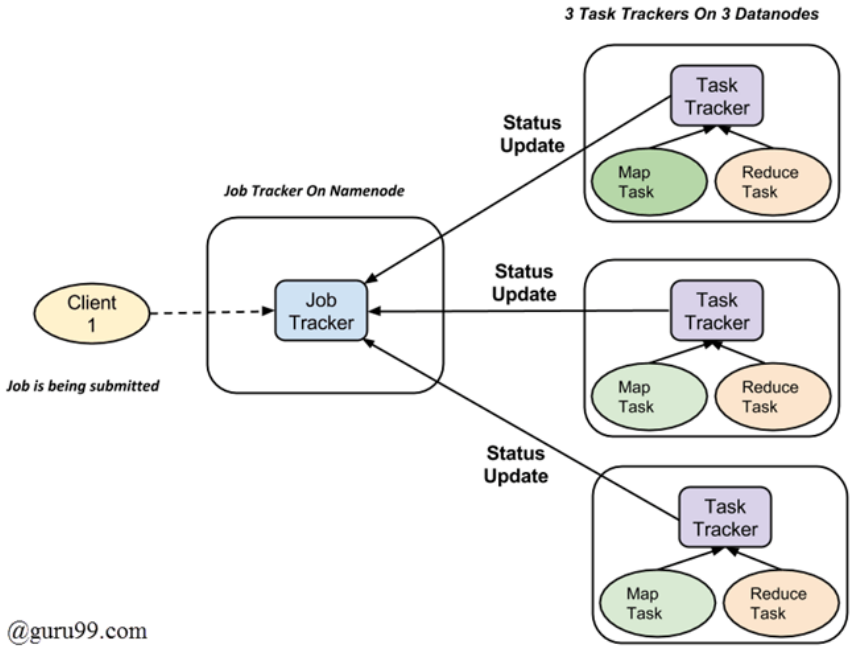


Figure: RPC implementations using Paxos(left) and Passive Replication(right)

Batch Computation Frameworks

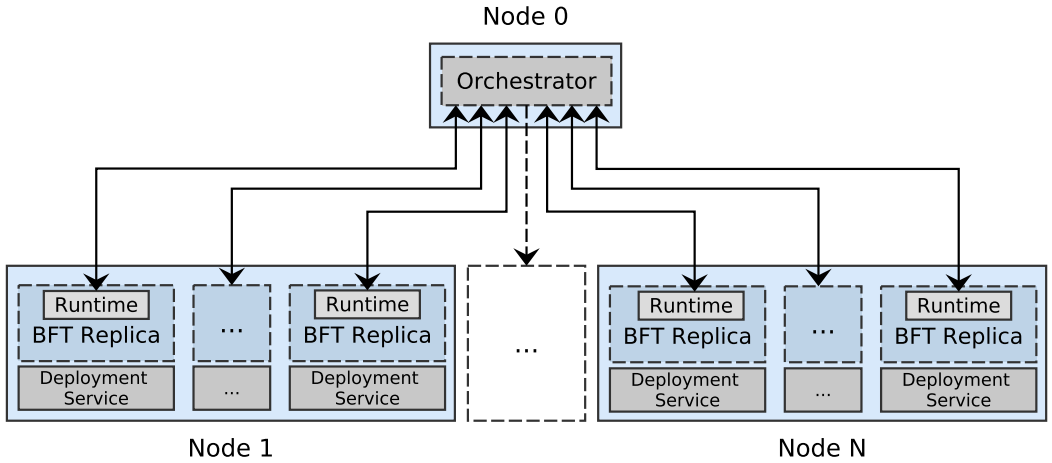
- MapReduce: Simplified Data Processing on Large Clusters

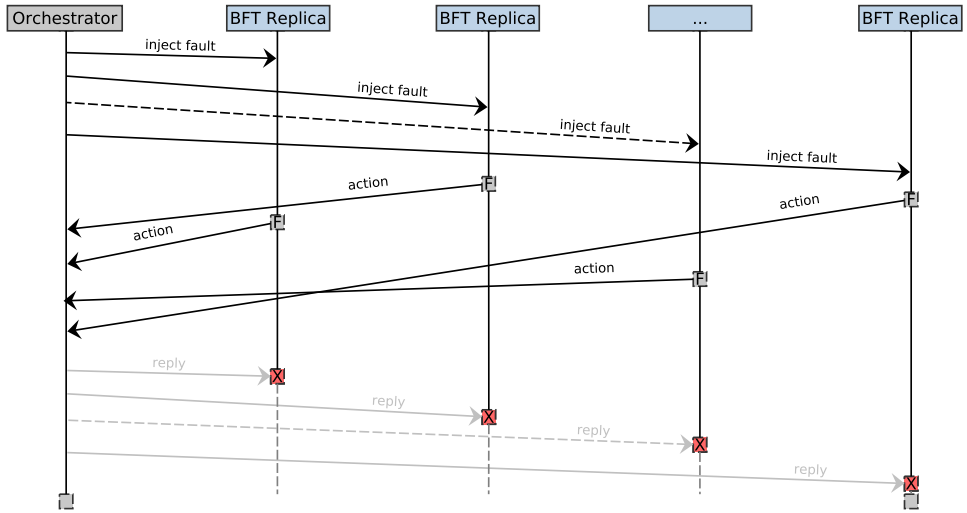




Fault-Injection

Hermes Fault Injection Framework





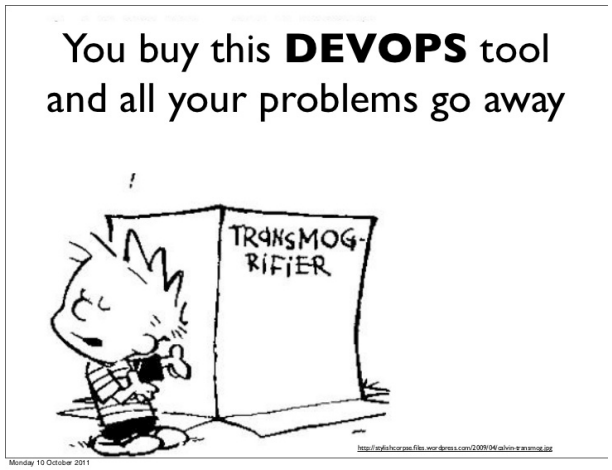
Part II - Hands on Cloud Computing

My 2 years on the Grinder

- YinzCam is the leading sports app provider in the US
- Apps with millions of downloads, tens of thousands of very active users
- Clients across the NBA, NHL, NFL, MLS (US) and AFL (Rugby - Australia)
- Features include live streaming, on-the-fly replays, seat upgrades, loyalty, social networks, ads
- All done with a very small team

DevOps Rulebook

- Log everything, literally everything!
  - You will need it to fast debug your system on-the-fly
- Monitor, Monitor, Monitor!
  - Don't rely on vendor's monitoring data!
  - Use your own! e.g., collectd
- Runaway from vendor lock-in and single vendor syndrome
  - Design your systems to account for multiple vendor, e.g., OpenStack with offloading to multiple vendors
  - Changing a production system may be prohibitive!
- Automate everything



Bob soon discovered that external factors affected his performance

Elasticity

- Boostrapping a new instance may take up to 5 minutes (including DNS's name propagation)
- For large events, e.g., conference news, this can lead to unacceptable levels of performance
- Know your workloads and provision accordingly
- A new recent approach is to provision large instances and then use containers on them, e.g. Docker

Monitoring

- Hardware malfunctioning still happens on the cloud!
- Complex subsystem, such as MySQL manage instances, are prone to performance issues
- Service architecture should handle service migration from scratch:
  - Handle maintenance periods
  - and, host crashes

Vendor Lock-In

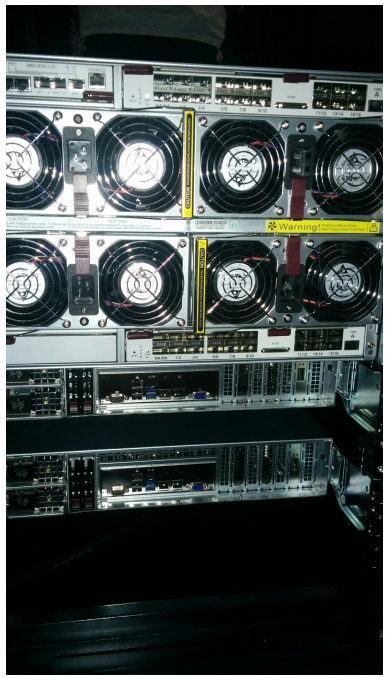
- Avoid managed services, e.g., MySQL instances, to avoid black boxes
- This will also allow you to chose between vendors
- That will save you headaches and money!

Security and Privacy

- You are relying on the vendor's perimeter security
- But you will also have to enforce security inside your system
  - Turn off HTTP, and only use HTTPS
  - Install an IDS and SIEM (e.g., community version of OSSIM by AlienVault)
  - Turn off password login, only use key-pair for SSH
  - Change periodically the passwords on more "legacy" services, e.g., MySQL
- If you have SENSITIVE information, DON'T use solely public clouds
- Anonimize your data and store it, but better is just to encrypt it and spread it among several vendors

Current Cloud Project

- INESC TEC shared cloud computing facility
- Dual Infiniband + dual ethernet ports per node (98Gbit/s)
- 2 Storages + Blade:
  - 232 cores + 3TB RAM
  - 32 TB shared storage



Part III - Edge Computing

What are Edge Clouds?

- "a computational storage cloud comprised solely of a collection of nearby wireless edge devices, with the purpose of pooling these devices data and processing power to support a new class of proximity-aware applications that benefit the owners of these devices."

Whats required to support such applications?

A: Communication infrastructure (traditionally)

- Expensive (3G/4G or WiFi coverage)
- Low user capacity (25-50 users / AP)

- **Alternative**: Can we take advantage of D2D communication? (WiFi-Direct, TDLS, Bluetooth)

Whats required to support such applications? (cont.)

B: Networking services for mobile devices

- w/ infrastructure:
  - Costly and QoE degrades rapidly in user-dense/emergency scenarios
- wo/ infrastructure:
  - Cheap but complex and cannot guarantee availability (delay-tolerant)



- How to use D2D protocols to create overlays networks and communicate between any two devices within?

Whats required to support such applications? (cont.)

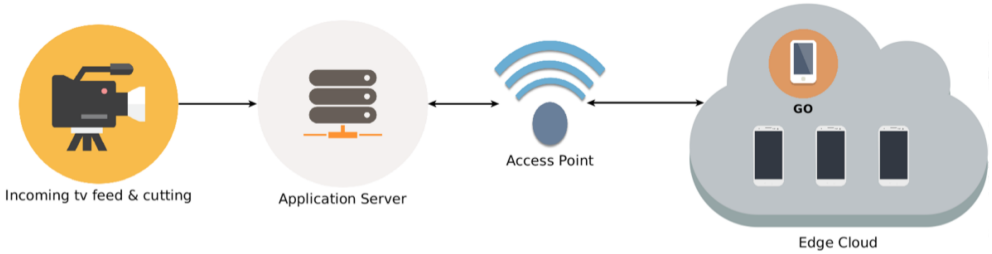
C: Storage, Computation and Security Services

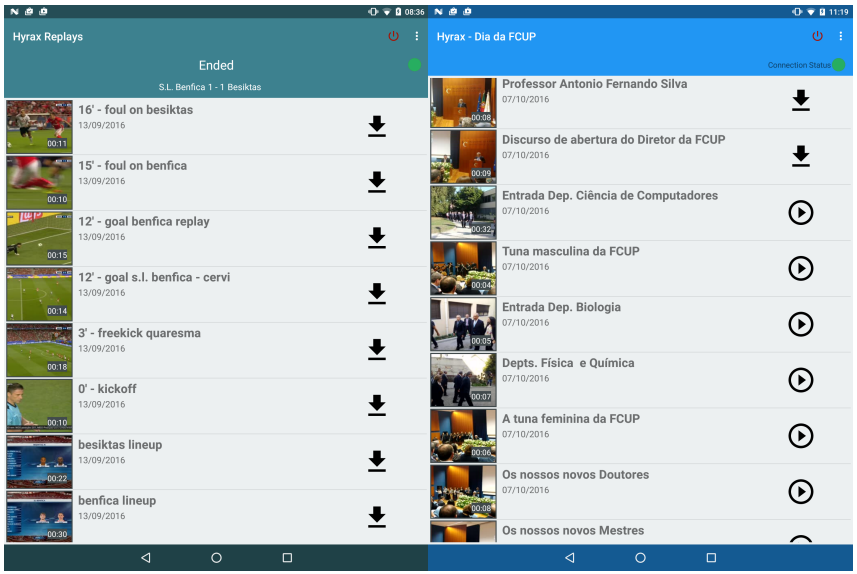
- Data sharing among devices requires an internet broker (4G / WiFi)
- Easier to enforce security and control mechanisms

- How to securely share computational resources using D2D networks?

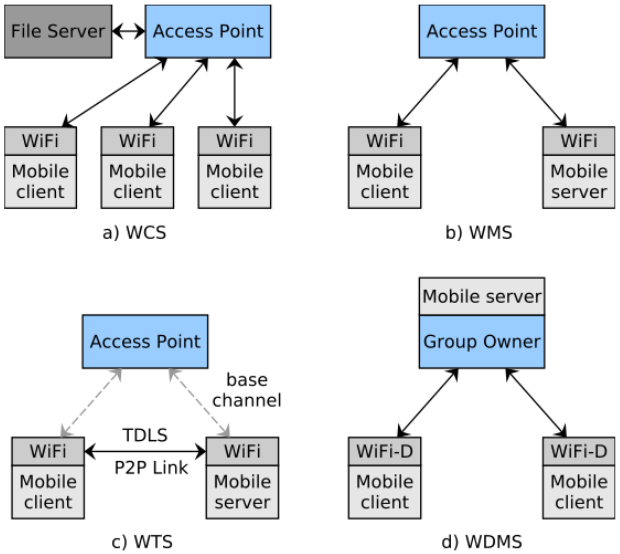
Hyrax in Action!

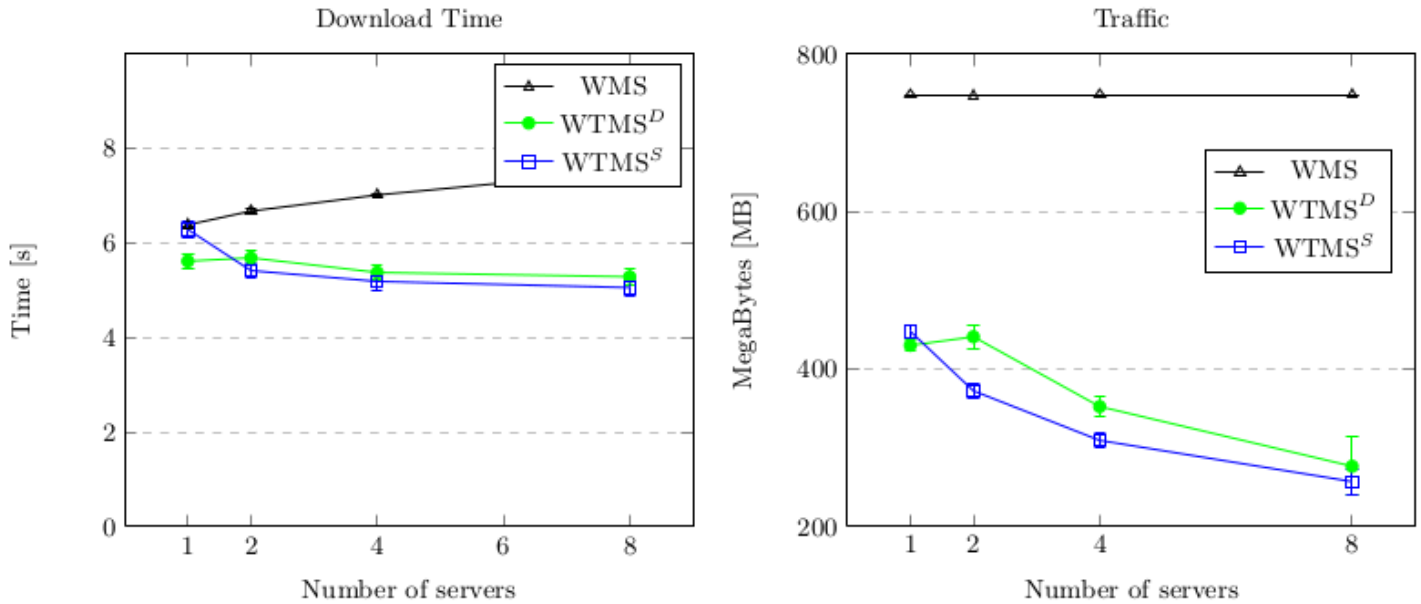
- A CMU-Portugal project: CRACS (FCUP), CMU and NOVA-LINCS





Some Results from Hyrax





Edge Clouds: Current Status

- Churn, Churn and Churn!
  - Depends on the application case, but it normally ranges from medium to highly dynamical systems
- Technological limitations
  - Lack of mobile OS interoperability, .e.g. TDLS only works for AndroidOS
  - Cripple features to safeguard cloud-based business models
- (Lack of) Security
  - All mobile platforms have bad track records
  - Workarounds are possible with customizable roms or through the use of secure tokens
- (Absent) Privacy
  - Probably the biggest issue we face today
  - Tagging, meta-data extraction, ...

What Communication Technologies are available?

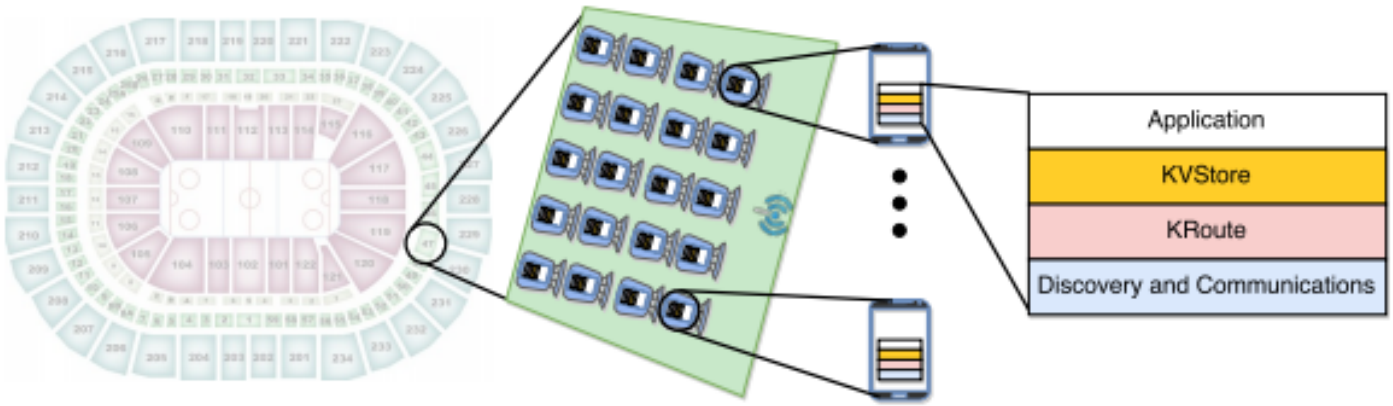
- Using non-rooted devices (Rodrigues'16):
  - Bluetooth (slow, slow and slow)
    - ★ Limited to 7 connections per device
  - WiFi TDLS
    - ★ Only available on Android
    - ★ AndroidOS makes its use very opaque, .e.g, no information on the state of a link
    - ★ Capabilities very dependent on device
  - WiFi Direct
    - ★ Only available on Android
    - ★ AndroidOS makes its cumbersome to use
    - ★ Very rigid and buggy implementation, .e.g, limited to 5 clients per soft-AP

Batch Processing on the Edge

- Possible but not suitable (Marinelli'09):
  - Jobtracker and NameNode only available on a central server
  - Assumed that nodes were static
  - Only used infrastructure WiFi
- Open problems:
  - Who watches the watchers?
  - Coordination is a nightmare
  - Job handover cannot rely on a central entity
  - Transient failures, i.e. churn, make it very difficult to have complete results

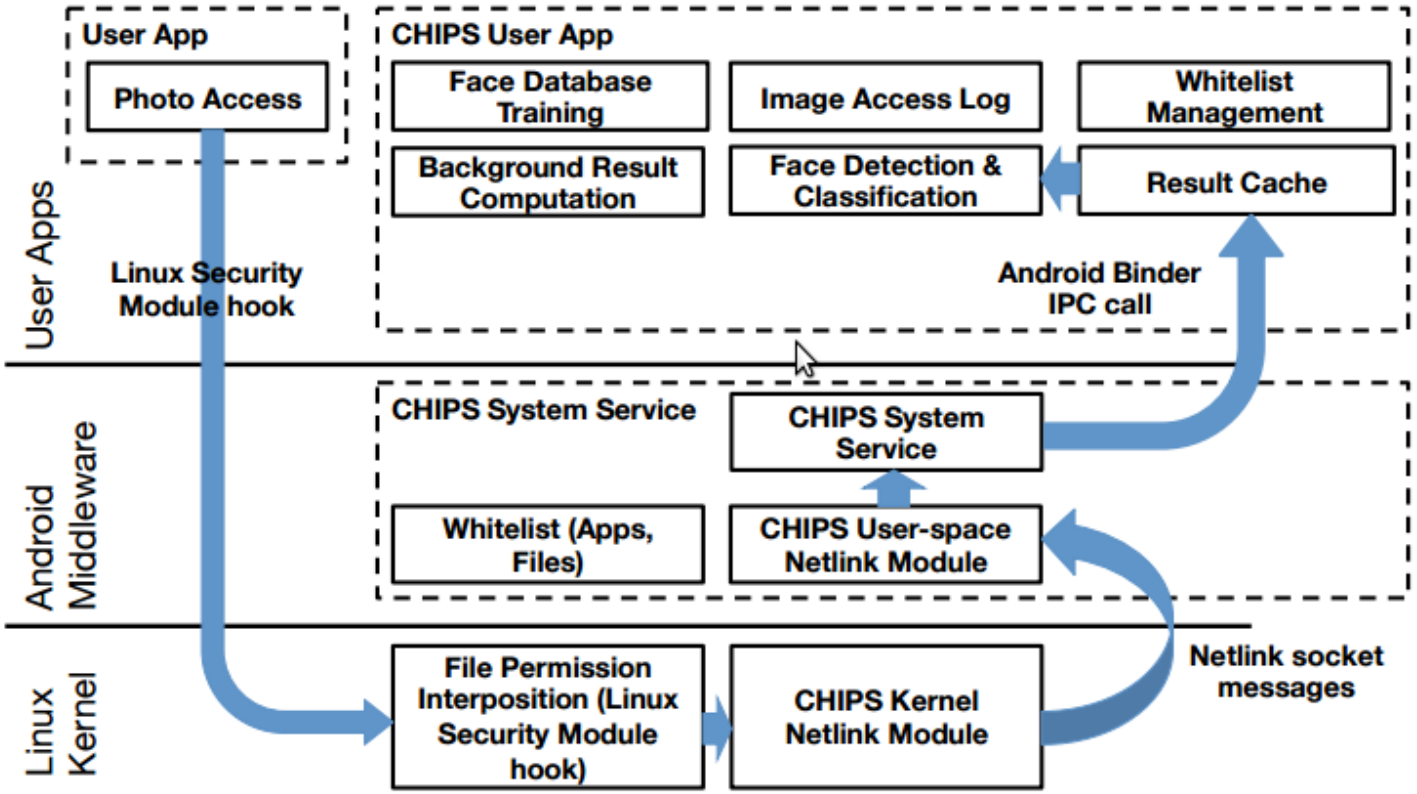
Key-Value Storage (Krowd'15)

- Elegant abstraction but not quite there yet:
  - Assumptions are weak, assume peers are rather stationary
  - Built on top of consistent hashing
  - Repair costs still not accounted



Privacy and Security in Edge Clouds

- CHIPS: Content-based Heuristics for Improving Photo Privacy for Smartphones (Jiaqi Tan'14)



And what about computation?

- Single-domain trust can achieve it, i.e., verified apps are digitally signed by a trusted provider
- True P2P computation still not possible:
  - Homomorphic encryption still not mature and slow
  - Proof carrying code still limited

What are Cloudlets?

- Satyanarayanan (CMU) defines Cloudlet as "A new architectural element that arises from the convergence of mobile computing and cloud computing. It represents the middle tier of a 3-tier hierarchy: mobile device - cloudlet - cloud. It can be viewed as a data center in a box whose goal is to bring the cloud closer"
- Seminal work from Satya @CMU:
  - Virtual Machine Introspection
  - Openstack++
  - Delta upgrades
- Current Status:
  - All things virtualized
  - Viewed as an utility on every city block

Cloudlets, who would use them?

- Museums and hospitals:
  - Enhancing low-latency services (e.g., VR, context aware services)
  - Complementing or replacing WiFi infrastructure

Cloudlets: Road ahead

- Containers for the rescue
- Middleware for context-aware services
- Support for *ad-hoc* networking
- Ongoing MSc

Cloudlets as a Middle Tier

- Scalable access to more powerful resources
- Provide wireless connectivity
- Low-latency services
- Self-repair layer
- Indoor localization
- Context-aware as a Service (CaaS)
  - Lightweight Data mining
  - Support for dynamical and fast reconfiguration for mobile applications

What about security and privacy with Cloudlets?

- Cloudlets suffer from the same issue as infrastructure clouds, you lose privacy if you use them
- As of now, security is based on trust on the provider

Another Application Case for Hyrax: Amber Alert

- Amber alert is used to find missing children
- Computer vision algorithms are needed to implement a system
- CV algorithms can be split between feature detection and matching
- We can use this knowledge and split work among edge and cloudlets
- This approach enhances privacy by avoid uploading content to a public cloud
- Energy usage patterns are being researched (Miguel Coimbra@FCUP)
- Ongoing MSc

Thanks!

- Thinking of doing a PhD or MSc on Cloud Computing?
- Reach me at **rmartins@dcc.fc.up.pt**