

STC 2014

14th Annual International

Software Testing Conference 2014

December 4 - 5, 2014 | Bangalore

Organised by

QAI

In association with

ETI

Emerging Areas: Virtualization / Agile

Adopting Agile Environment : Utilizing Docker
Containers For Accelerating & Optimizing Testing



Authors - Khelender Sasan & Divya Saxena
(NEC Technologies India Ltd.)

Agenda

Background

- scenario of technology world

Technology's business impact...

- Agility
- Cost
- Newer Paradigms

Docker Technology

- Introduction
- Docker Architecture

Use Cases For Application Of Docker Containers

Technology usage demonstration

References

Background scenario of today's technology world !



Time to market is crucial !!



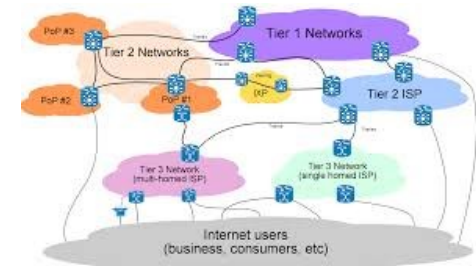
Resource cost optimization is also crucial (to thrive in competing world!)



Hardware optimization reached level of micro-servers (in parallel to HPC!)

PARADIGM SHIFT

Newer paradigm of software are evolving (that needs to be handled by new paradigm in testing!)

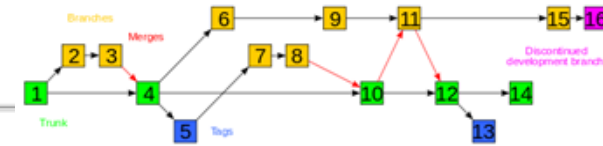


● Distributed computing (Docker container has started making some inroads here as well !)



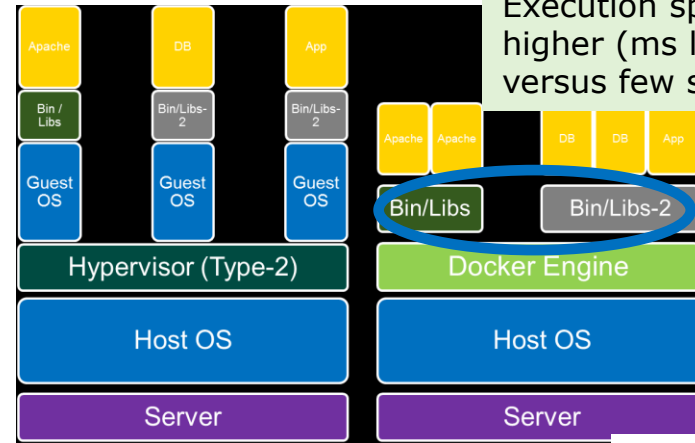
● Cloud (model) presence is ubiquitous

Agility...



Agility performance is paramount ! (Agile methodology & environment)

- Provides much better performance as compared to VMs
 - Launch time in sub-seconds: Reduced cycle time of test environment deployment and execution
 - Ease out configuration management of resource (Inherent CM)
 - Enhance CI / CD execution cases
 - **Decreased development and QA time (by ensuring replica environment with all dependencies across stages)**
 - Automation frameworks (orchid/fig)
 - Optimized server environments
- (shorten release cycles:
some orgs @ 3 release / day)



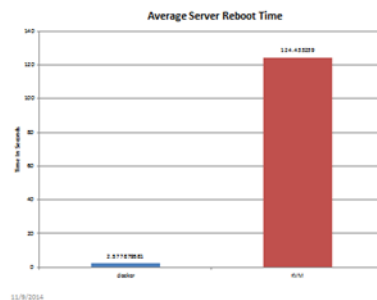
Execution speed is much higher (ms launches versus few seconds)

Sharing OS, Binaries & Libraries



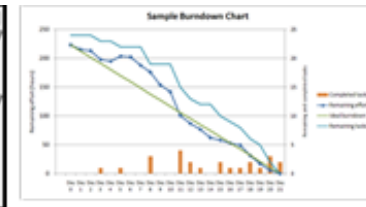
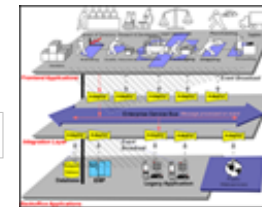
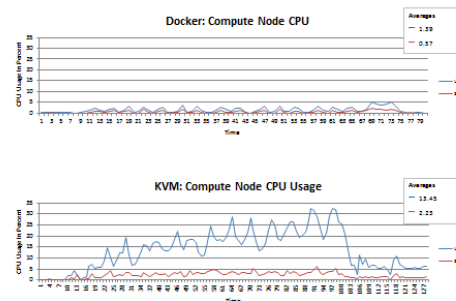
Cloudy Performance: Serial VM Reboot

IBM



Cloudy Performance: Serial VM Boot

IBM



Costing...



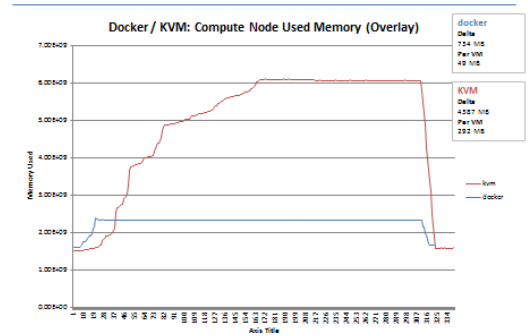
Work so far

- Cloud performance optimization !
(VM performance improvement
Several GB -> 100 MBs)

Docker container offer much better density than traditional VMs

- can scale ~10x – 100x
- Compute Resource Optimization**
- Support Cost Optimization**
- License Cost Optimization** (Pure OSS versus commercial VMs)

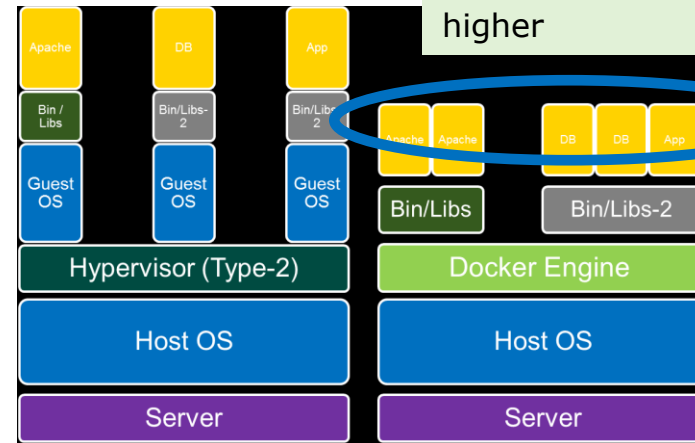
Cloudy Performance: Steady State Packing



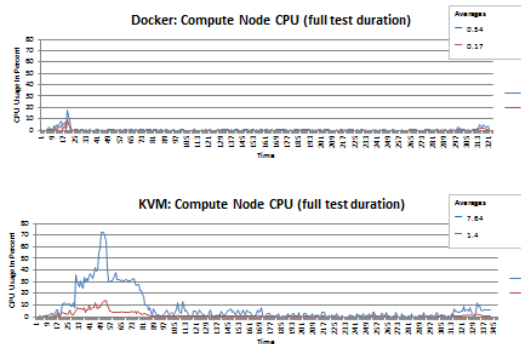
11/9/2014

14

Density of virtual entities is much higher



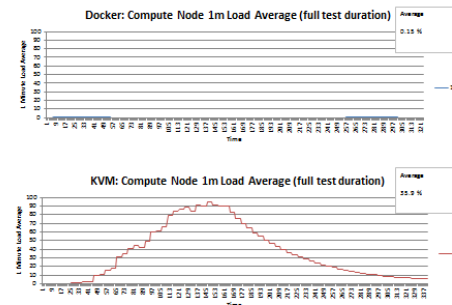
Cloudy Performance: Steady State Packing



11/9/2014

11

Cloudy Performance: Steady State Packing



11/9/2014

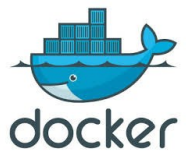
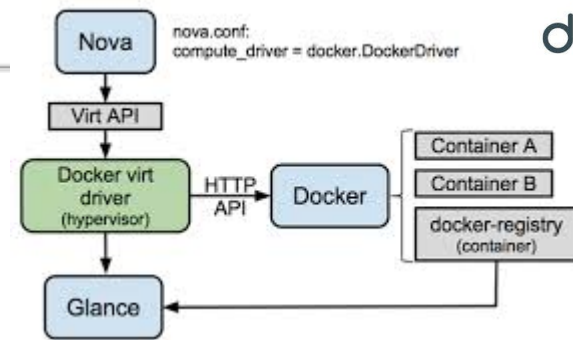
15



Strengthening new Paradigms!

Cloud

- Enhanced performance of OpenStack through docker support in Nova (Compute engine)

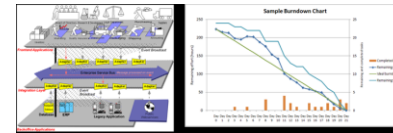


CI-CD & DevOps

- Thousand of off-the shelf docker images for quick launch without repeated installs !
- Accelerated environment replication using CM facility!
- Computing Resource treatment as pet versus cattle



docker registry

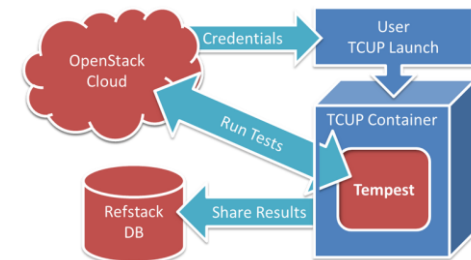


Cloud deployments Interoperability testing (OpenStack Cloud framework)

- Refstack using docker

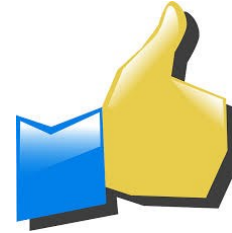


Pets
Versus
Cattle!



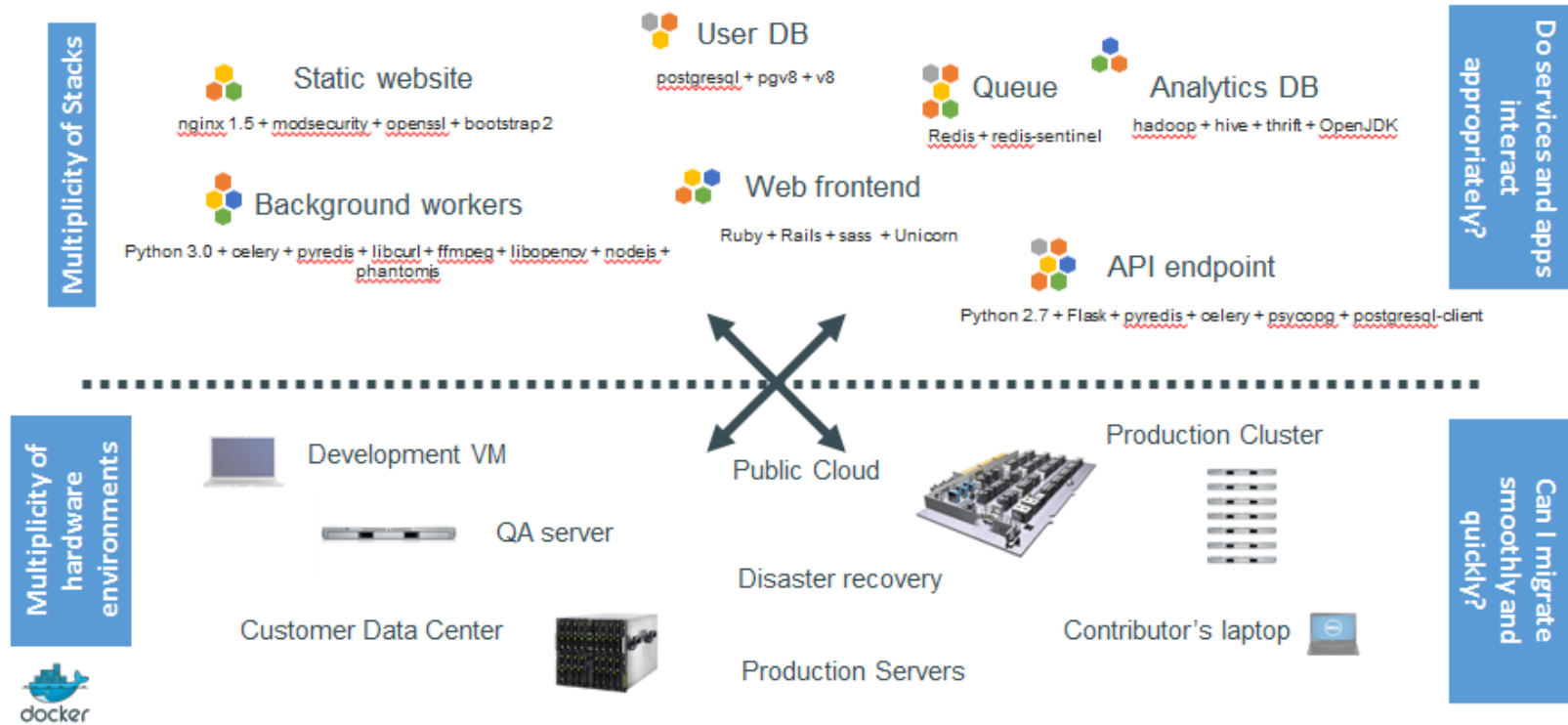
Solomon Hykes: person behind Docker Container

- DotCloud : Provides Cloud solutions
- Brain behind Docker : Solomon Hykes
- Theme / Motto :
 - "The real value of Docker is not technology, **it's getting people to agree on something.**"



Challenge of Software Industry

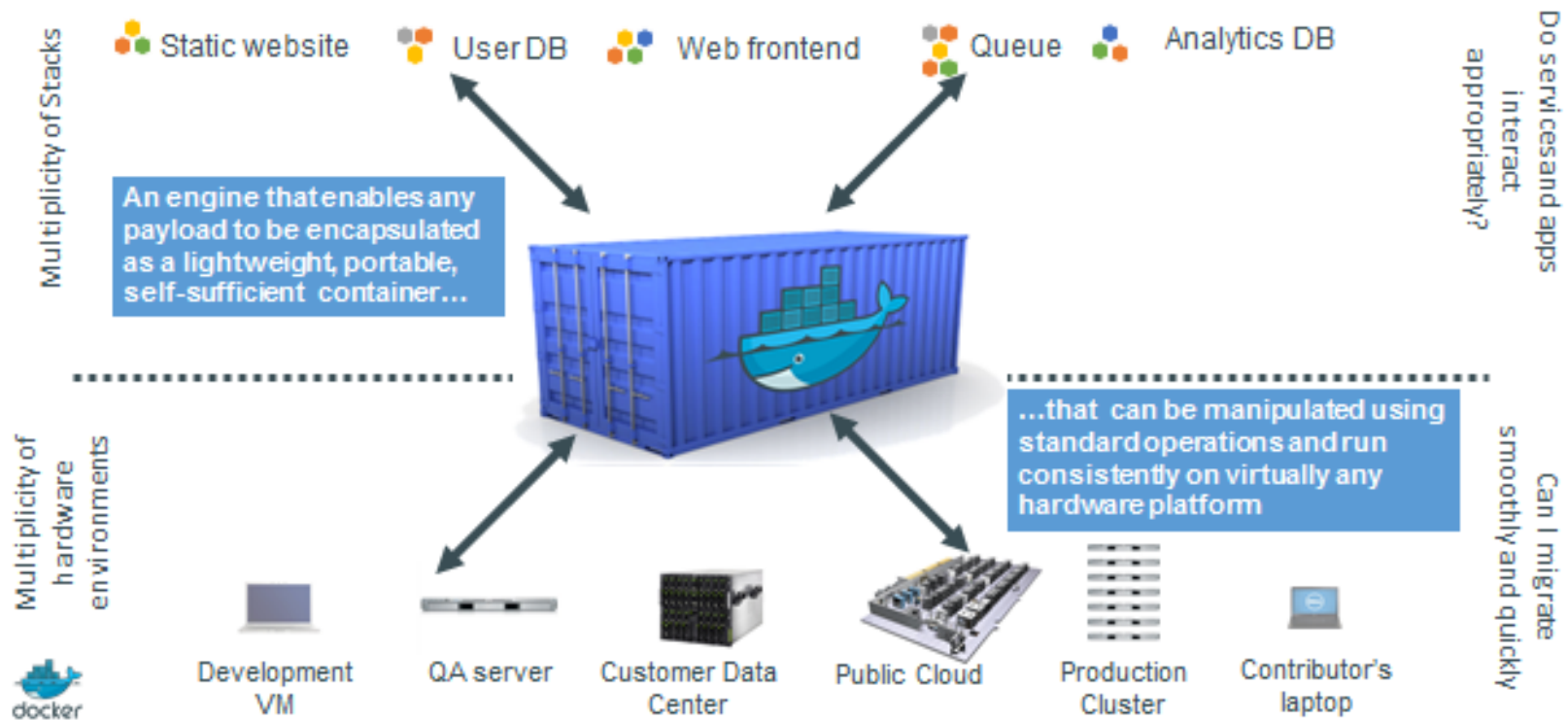
The Challenge



Source: <http://www.slideshare.net/Docker/dockerintronovember-131125185628phpapp02-37588934>

Solution

Docker is a shipping container system for code



Source: <http://www.slideshare.net/Docker/dockerintronovember-131125185628phpapp02-37588934>

Solomon Hykes: person behind Docker Container

DotCloud : Provides Cloud solutions

Brain behind Docker : Solomon Hykes

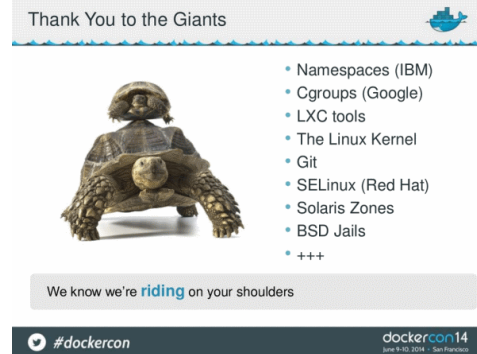
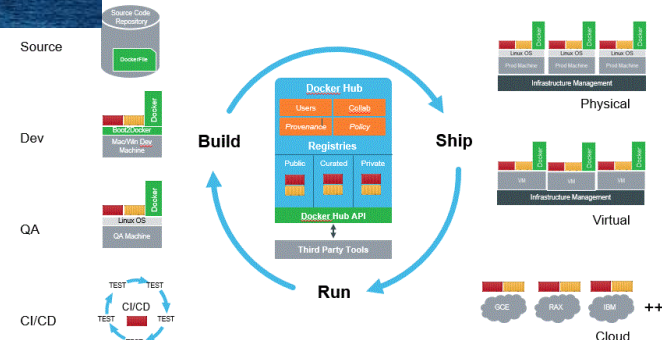
Theme / Motto :

- “The real value of Docker is not technology, **it's getting people to agree on something.**”

Got inspiration of solution for software troubles from Shipping Industry

Nothing New => Just recipe was required !

- Acknowledged all recipe ingredients in Docker Conf'14



Docker Containers and technology business ecosystem

Microsoft & Docker

- Docker and Microsoft partner to bring container applications across platforms (15-Oct-2014)

Google & Docker

- Google Launches Managed Service For Running Docker-Based Applications On Its Cloud Platform (4-Nov-2014)
- Google Container Engine to use Kubernetes for providing "Docker as a Service" ("Cluster As a Service"?)

Linux Distros & Docker

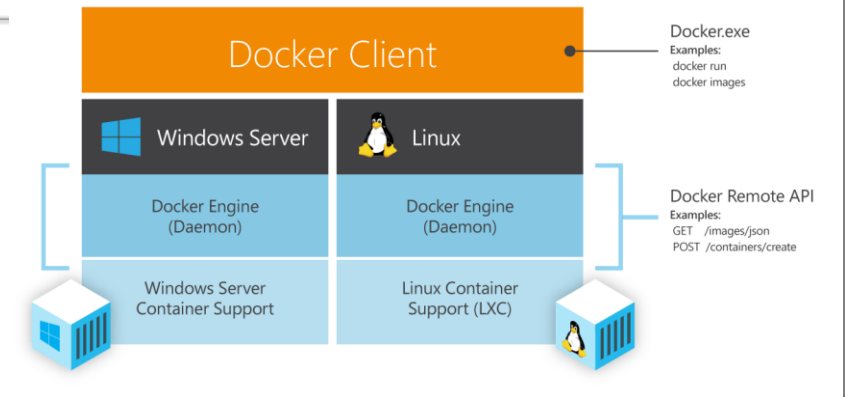
- Canonical Releases Ubuntu 14.10 With Tighter Docker and Cloud Foundry Integrations (24-Oct-2014)
- SUSE bundles Docker and processor optimizations into major new Linux distro (27-Oct-2014)
- Red Hat To Bring Docker Support To Enterprise Linux And OpenShift (15-Apr-2014)

Docker acquisitions

- Docker acquires Orchard (for Orchestration & integration platforms)
- Docker Acquires Koality In Engineering Talent Grab (7-Oct-2014)

Server Environment Optimization (CoreOS) : Planning to serve a diet Linux Platform

- Every App / Service will be served using docker container !



Creating newer solutions & Paradigms !



Distributed Computing

- Clustering automation & Orchestration (Kubernetes)
- Server Optimization (CoreOS)
- Library for running docker container in cloud (fig, orchard)
- Libraries / solutions for automation in distributed space
 - Libchan, libswarm



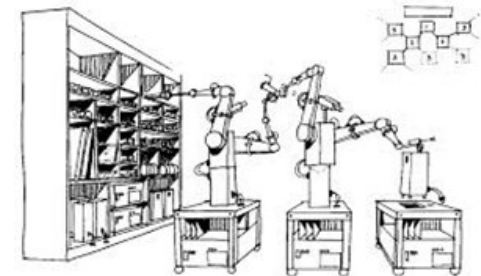
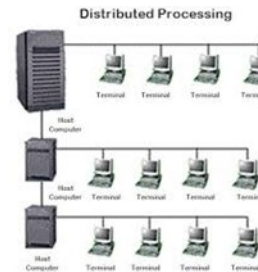
Kubernetes

κυβερνήτης: Greek for "pilot" or "helmsman of a ship"
the open source cluster manager from Google



Challenging environment for QA:

- **Hardware : Micro-servers (form factor based)**
- **Software : Distributed / Cloud based**
- Testing (system and overall) going to be challenging, where components can be present across physical deployments of units
- Need to work in optimal space without disturbing development environment
 - Non-invasive testing can be enabled by Docker (remember OpenFlow in SDN)



More possibilities

- Can easily support integration of Virtual environment in Automation Frameworks!
- More Testing Scenarios
 - Interoperability
 - Network space
- Supports nested virtualization (and inner processes running in native environment only!)
- Acceleration of software setups on Fresh Hardware setups



Hypervisor VM versus Docker Containers

Virtual Machine (Hardware virtualization)

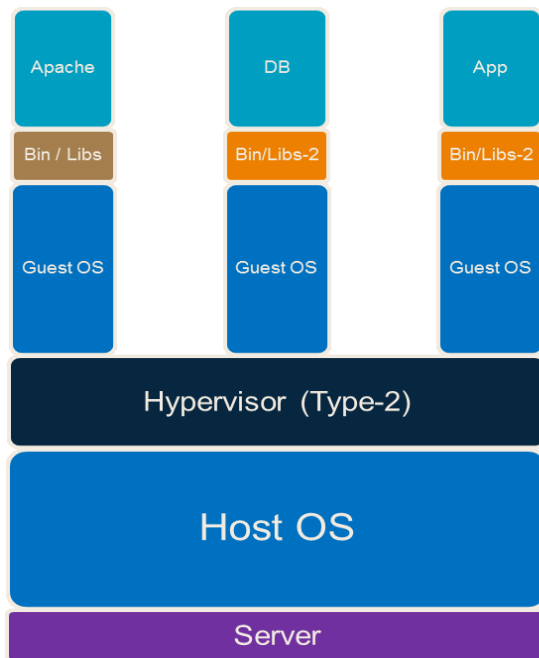
- Includes hardware simulation & OS execution
- Requires order of GB – 100s of MBs of memory for each instance (Heavy Weight)
- Can simulate few VMs per server (Expensive)
- Instance launch complete takes several seconds minute (slower)

Docker containers (OS level virtualization)

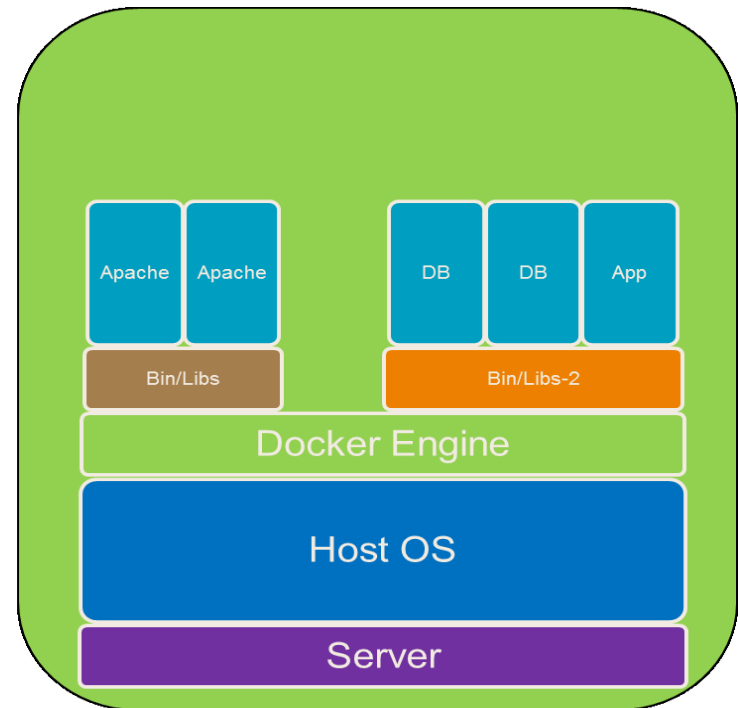
- Based on Linux Containers and Union File System.
- Requires order of few MBs of memory for each instance (Light Weight)
- Can simulate much higher containers per server (Cheaper)
- Instance launched in sub-second time-frame (much faster)

Docker Technology : Introduction

Framework built on top of Linux containers that can package an application and its dependencies => Can be launched / deployed in form of **software container** on any Linux server (providing **portability** of underlying platform: **cloud, bare-metal, server, desktop or laptop**)



Traditional Hypervisor based VMs



Container based Application execution

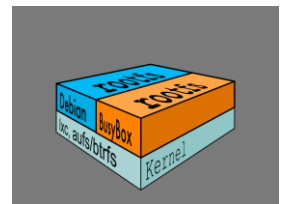
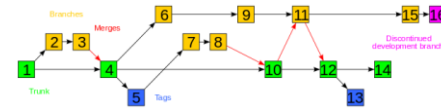
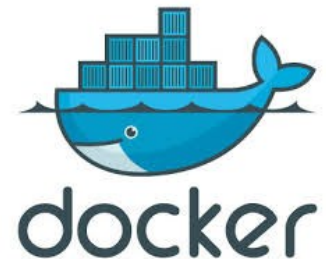
Docker Technology Introduction...continued...

Container relies on Linux kernel facilities: **chroot**, **cgroups** and **name-space** facilities that provide resource isolation (CPU, memory, block I/O, network, etc.) => Doesn't require a separate OS instance.

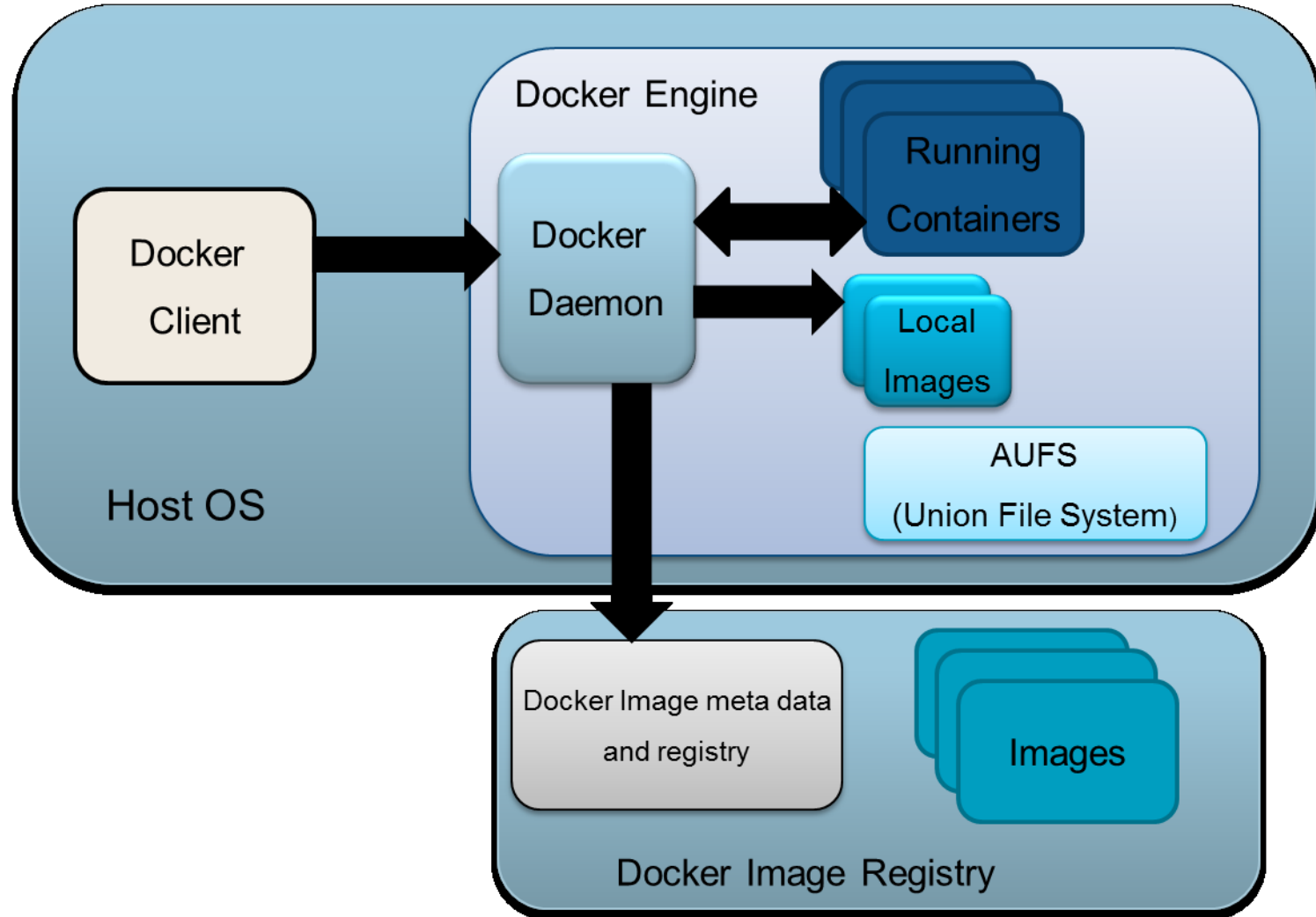
Docker provides **framework** & a **high level API** over Linux containers for portable deployment of applications across machines

Docker provides **versioning capability** to track successive versions of a container

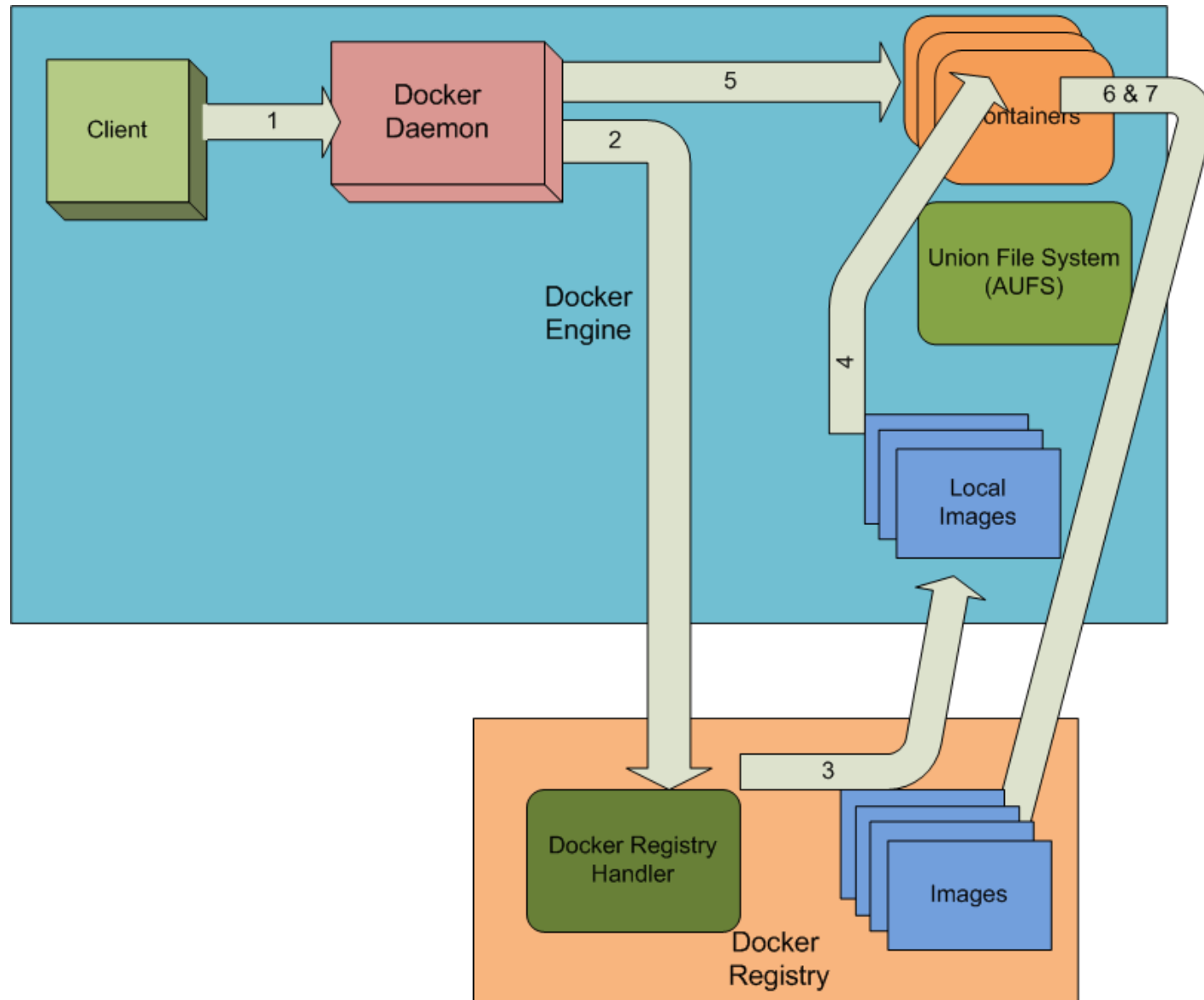
Docker uses **AUFS** (Advanced multi-layered Unification File System) – It greatly enhances the **performance** of docker containers and provides for **optimized** usage of underlying disk resources



Docker Architecture



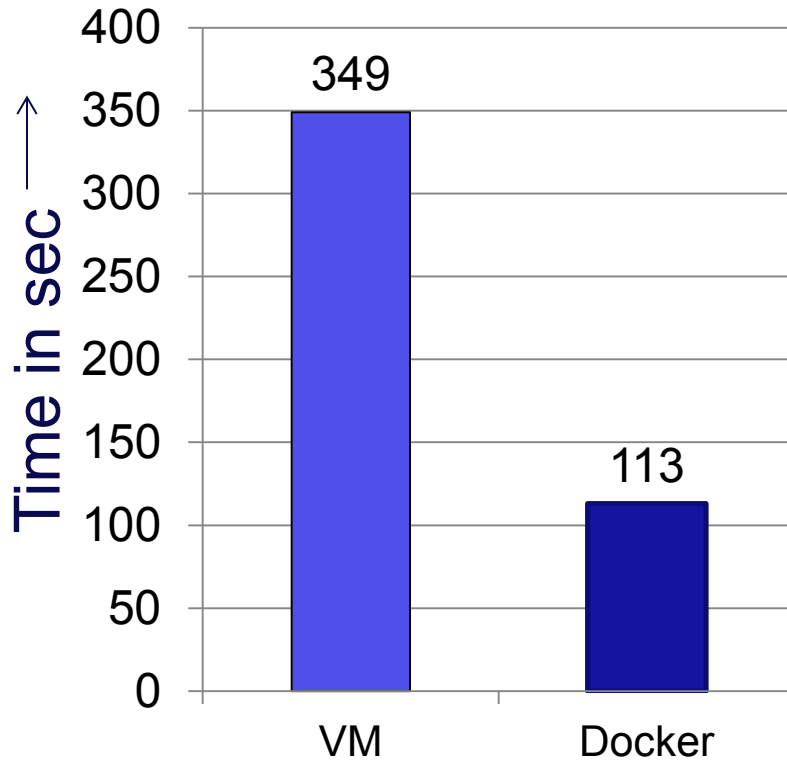
Usage Flow



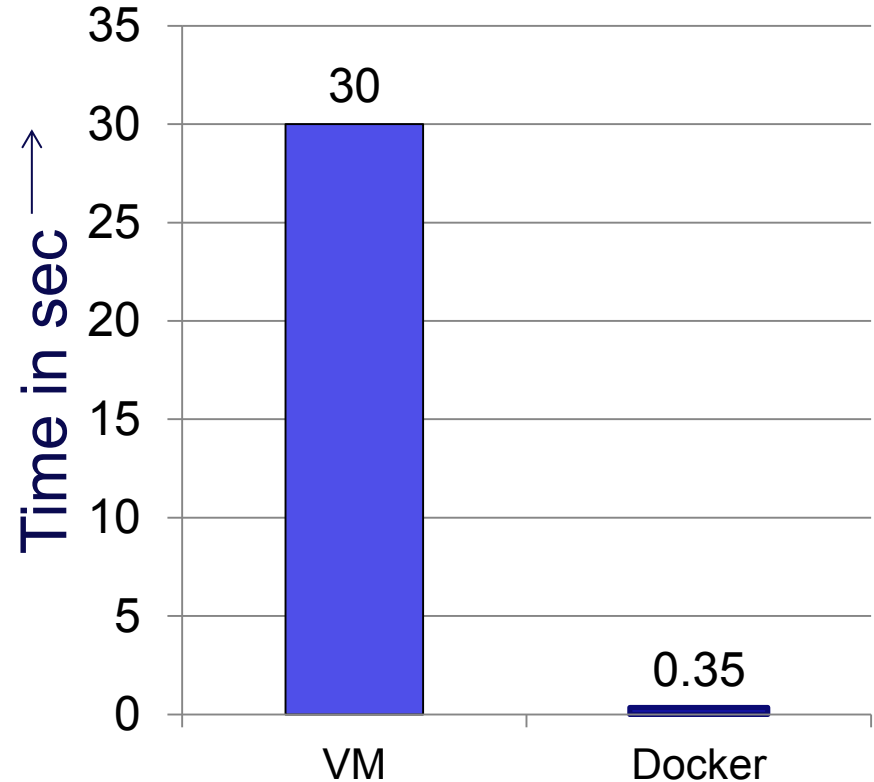
Initial Benchmarking Results (1/3)

App Execution Time

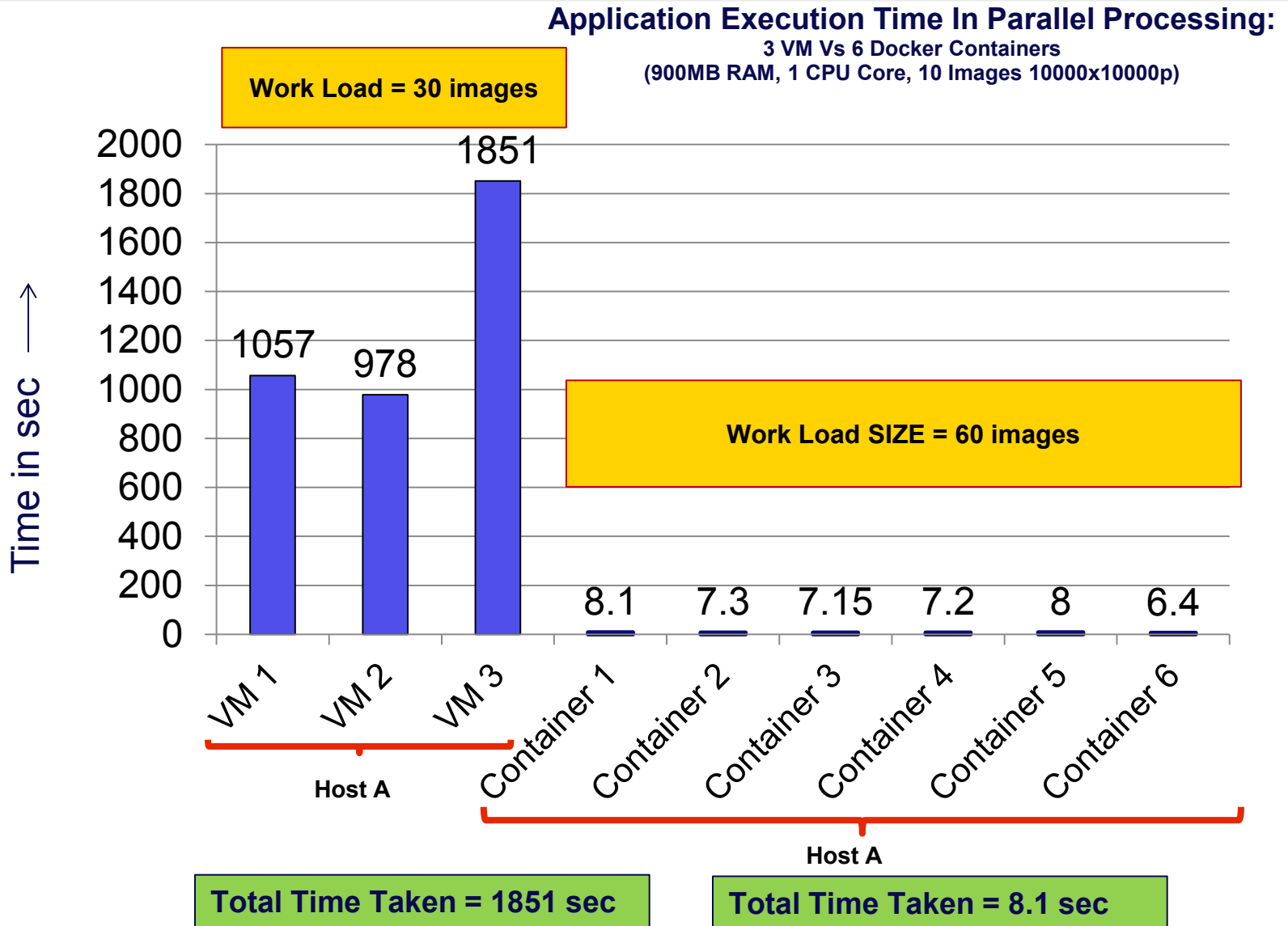
(1024MB RAM, 2 CPU Core,
10 images 10000x10000p)



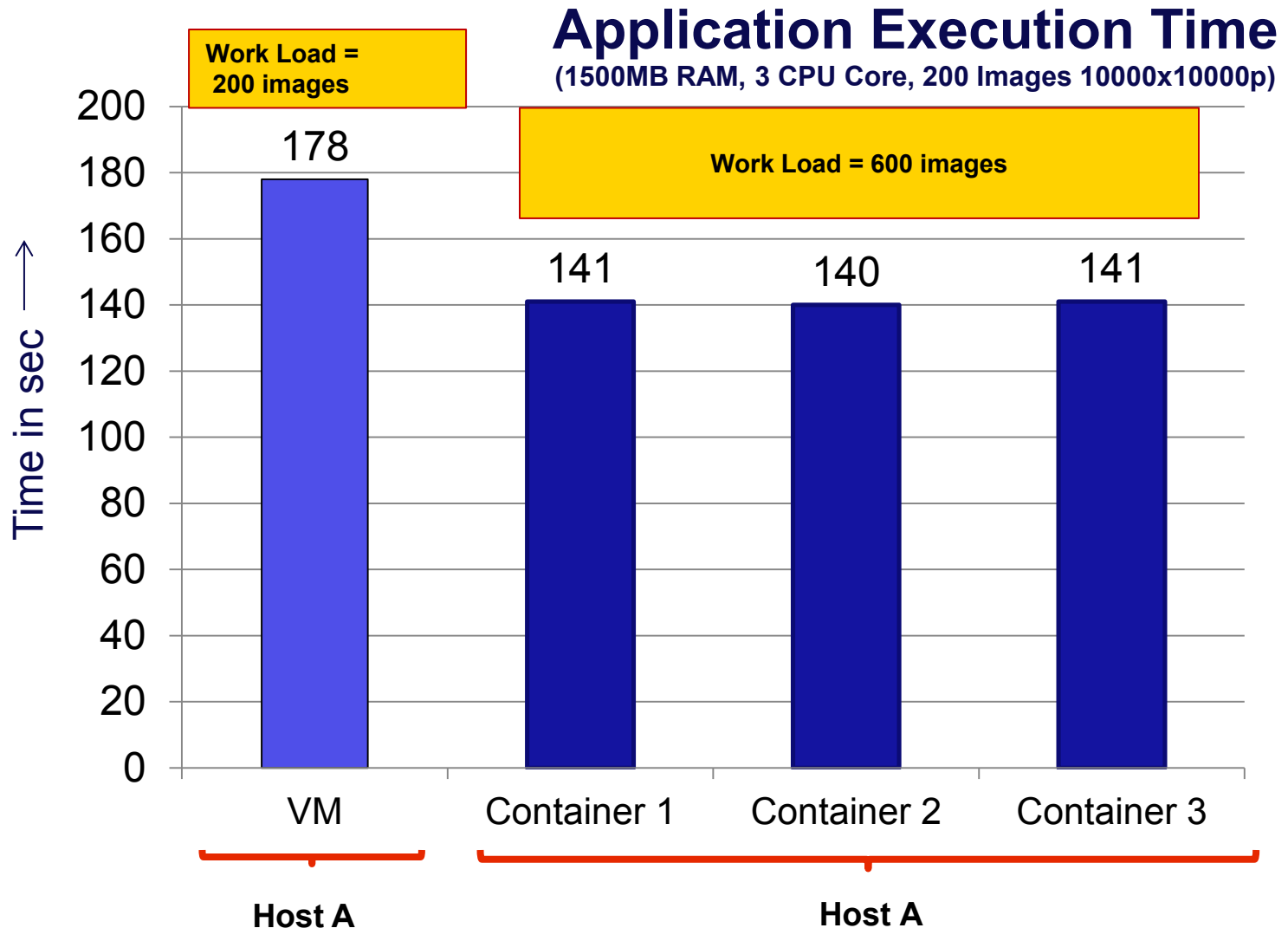
Launch Time



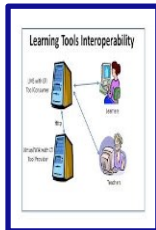
Initial Benchmarking Results (2/3)



Initial Benchmarking Results (3/3)



Use Cases For Application Of Docker Containers



Interoperability testing
across Linux Distributions
and application packages



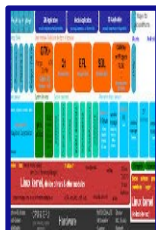
Implementation
optimizations



Networking scenarios



Scalability testing



Non-invasive testing



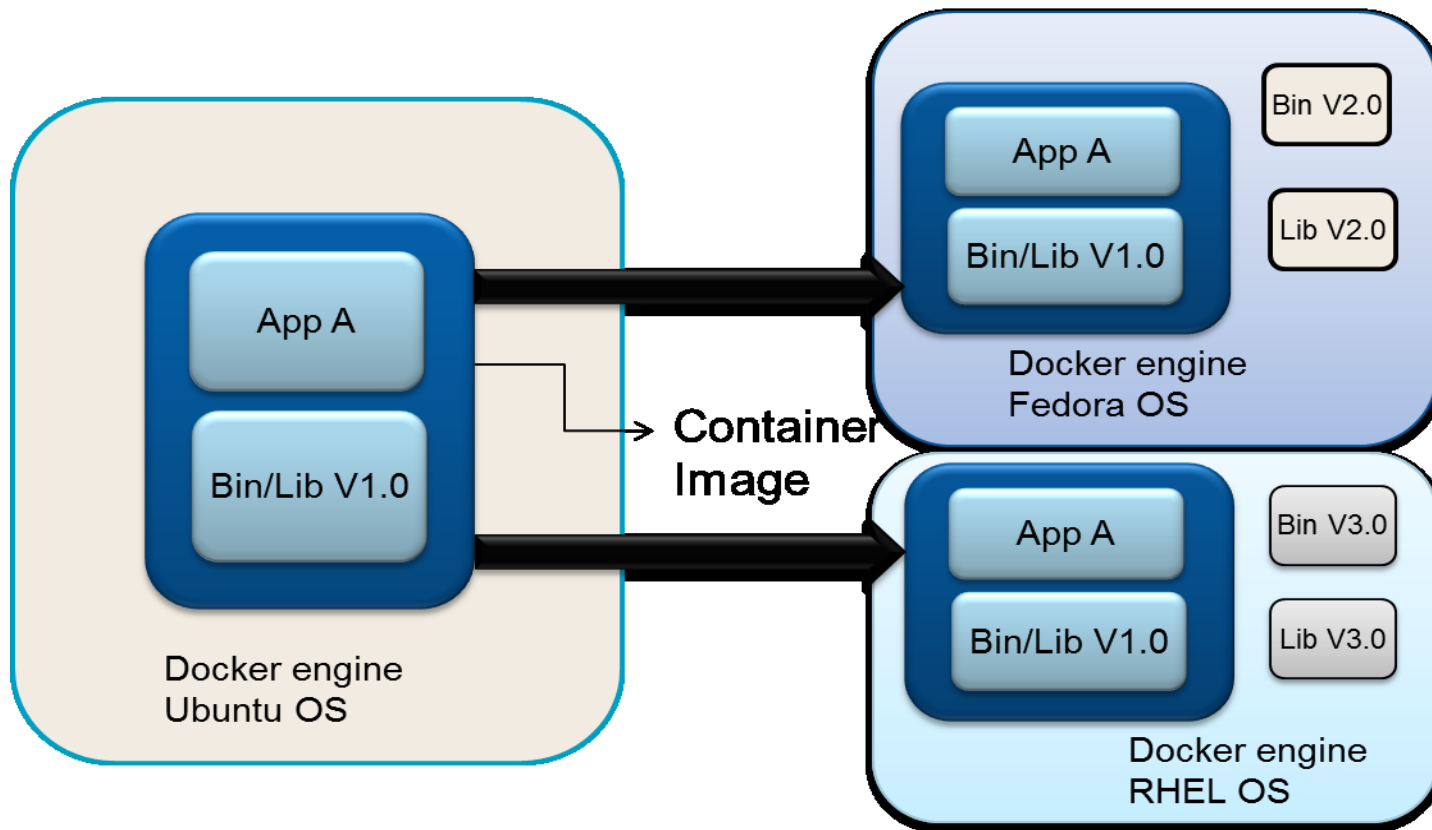
Application Stress testing:
behavior validation



Optimize frequent installs &
setups during fresh
hardware bring-up

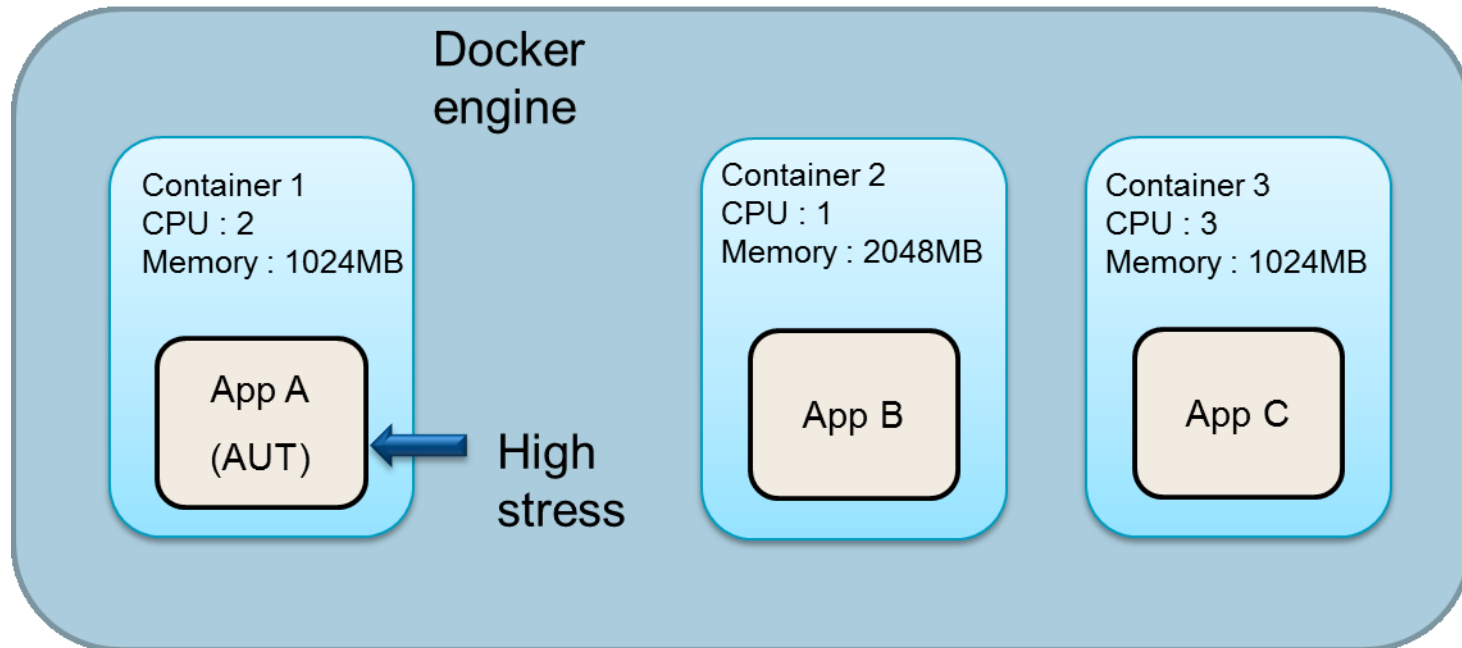
Distributions / Interoperability testing (1/3)

Testing an application along with required library versions (using bundled images) across various different Linux systems



Stress Testing Validation

Application Stress testing: behavior validation



Technology Usage Demonstration (1/5)

Docker installation

```
divya@ubuntu64-1:~$ sudo apt-get install docker.io
divya@ubuntu64-1:~$ ps -aef | grep docker.io | grep -v grep
root      491   484   5 14:45 ?        00:01:17 /usr/bin/docker.io -d
divya@ubuntu64-1:~$
```

Download an existing container image from public docker registry

```
divya@ubuntu64-1:~$ sudo docker pull ubuntu:14.04
Pulling repository ubuntu
53bf7a53e890: Download complete
511136ea3c5a: Download complete
134b5dc84bc7: Download complete
692254366b1a: Download complete
ed98671f0531: Download complete
bffdbd3bc4b2: Download complete
964692831e07: Download complete
divya@ubuntu64-1:~$
```

Technology Usage Demonstration (2/5)

Spawning container from downloaded container image

```
divya@ubuntu64-1:~$ sudo docker run -i -t ubuntu:14.04 /bin/bash
root@9f87c3b3a066:/#
```

OS level virtualization demonstration

```
root@9f87c3b3a066:/# sleep 100 &
[1] 8
root@9f87c3b3a066:/# ps -aef
UID          PID    PPID  C STIME TTY          TIME CMD
root         1      0  0  09:41 ?           00:00:00 /bin/bash
root         8      1  0  09:42 ?           00:00:00 sleep 100
root         9      1  0  09:42 ?           00:00:00 ps -aef
root@9f87c3b3a066:/#
```

```
divya@ubuntu64-1:~$ ps -aef | grep sleep
root         2800   2724  0 15:12 pts/11    00:00:00 sleep 100
divya        2803   2752  0 15:13 pts/24    00:00:00 grep --color=auto sleep
divya@ubuntu64-1:~$ ps -aef | grep /bin/bash
root         2716   2365  0 15:11 pts/4     00:00:00 sudo docker run -i -t ubuntu:14.04 /bin/bash
root         2717   2716  0 15:11 pts/4     00:00:00 docker run -i -t ubuntu:14.04 /bin/bash
root         2724   491   0 15:11 pts/11    00:00:00 /bin/bash
divya        2805   2752  0 15:13 pts/24    00:00:00 grep --color=auto /bin/bash
divya@ubuntu64-1:~$
```

Technology Usage Demonstration (3/5)

Concept of chroot and aufs

```
root@9f87c3b3a066:/# touch sampleFile
root@9f87c3b3a066:/# ls -l sampleFile
-rw-r--r-- 1 root root 0 Sep 25 09:45 sampleFile
root@9f87c3b3a066:/#
```

```
divya@ubuntu64-1:~$ sudo find /var/lib/docker -name "sampleFile"
/var/lib/docker/containers/9f87c3b3a066bf1131a49d57213874c2e12abd428eb97f87944f47e0d528397c/root/sampleFile
/var/lib/docker/aufs/diff/9f87c3b3a066bf1131a49d57213874c2e12abd428eb97f87944f47e0d528397c/sampleFile
/var/lib/docker/aufs/mnt/9f87c3b3a066bf1131a49d57213874c2e12abd428eb97f87944f47e0d528397c/sampleFile
divya@ubuntu64-1:~$
```

Setting up local private docker registry

```
divya@ubuntu64-1:~$ sudo docker pull samalba/docker-registry:latest
Pulling repository samalba/docker-registry
6b86e5be37f9: Download complete
511136ea3c5a: Download complete
b3553b91f79f: Download complete
ca63a3899a99: Download complete
ff01d67c9471: Download complete
7428bd008763: Download complete
c7c7108e0ad8: Download complete
826544226fdc: Download complete
2e2525381d8a: Download complete
ac45f5b4c074: Download complete
86be4d0e9e36: Download complete
ab76794dacab: Download complete
6802e92c2da8: Download complete
5f3425169d60: Download complete
d4cf7d2a4a02: Download complete
c57ea96fb80a: Download complete
a1123aa3c2a1: Download complete
3249ad30604a: Download complete
0f729653c534: Download complete
divya@ubuntu64-1:~$
```

Technology Usage Demonstration (4/5)

```
divya@ubuntu64-1:~$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
ubuntu	14.04	c4ff7513909d	6 weeks ago	225.4 MB
samalba/docker-registry	latest	baf4b735e5c9	9 weeks ago	421 MB

```
divya@ubuntu64-1:~$
```

```
divya@ubuntu64-1:~$ sudo docker run -d -p 5000:5000 samalba/docker-registry:latest
695e3f28361c3149b2e13aa862d4ce6c217654b2405706a81efbe58b3a7531b5
```

Uploading images in local private docker registry

```
divya@ubuntu64-1:~$ sudo docker tag ubuntu:14.04 localhost:5000/ubuntu_local
divya@ubuntu64-1:~$ sudo docker push localhost:5000/ubuntu_local
The push refers to a repository [localhost:5000/ubuntu_local] (len: 1)
Sending image list
Pushing repository localhost:5000/ubuntu_local (1 tags)
511136ea3c5a: Image successfully pushed
1c9383292a8f: Image successfully pushed
9942dd43ff21: Image successfully pushed
d92c3c92fa73: Image successfully pushed
0ea0d582fd90: Image successfully pushed
cc58e55aa5a5: Image successfully pushed
c4ff7513909d: Image successfully pushed
Pushing tag for rev [c4ff7513909d] on {http://localhost:5000/v1/repositories/ubuntu_local/tags/latest}
divya@ubuntu64-1:~$
```

Technology Usage Demonstration (5/5)

Versioning (Committing changes as new image)

```
divya@ubuntu64-1:~$ sudo docker run -i -t localhost:5000/ubuntu_local touch file1
```

```
divya@ubuntu64-1:~$ sudo docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
TUS	PORTS	NAMES		
99ebaa84bc0a	localhost:5000/ubuntu_local:latest	touch file1	About a minute ago	Exited
t 0		happy_tesla		
695e3f28361c	samalba/docker-registry:latest	/bin/sh -c exec dock	7 minutes ago	Up
7 minutes	0.0.0.0:5000->5000/tcp	stupefied_nobel		

```
divya@ubuntu64-1:~$
```

```
divya@ubuntu64-1:~$ sudo docker commit 99ebaa84bc0a localhost:5000/ubuntu_update
550c086e7245bc9d0effad72cc4a8d295b726e55ec053cb01bca709871c39245
divya@ubuntu64-1:~$
```

```
divya@ubuntu64-1:~$ sudo docker push localhost:5000/ubuntu_update
The push refers to a repository [localhost:5000/ubuntu_update] (len: 1)
Sending image list
Pushing repository localhost:5000/ubuntu_update (1 tags)
Image 511f136ea3c5a already pushed, skipping
Image 1c9383292a8f already pushed, skipping
Image 9942dd43ff21 already pushed, skipping
Image d92c3c92fa73 already pushed, skipping
Image 0ea0d582fd90 already pushed, skipping
Image cc58e55aa5a5 already pushed, skipping
Image c4ff7513909d already pushed, skipping
550c086e7245: Image successfully pushed
Pushing tag for rev [550c086e7245] on [http://localhost:5000/v1/repositories/ubuntu_update/tags/latest]
divya@ubuntu64-1:~$
```


References

Docker (<http://www.docker.com/>)

- Introduction to Docker (November-2013) (<http://www.slideshare.net/Docker/dockerintronovember-131125185628phpapp02-37588934>)
- Docker: Automated and Consistent Software Deployments (<http://www.infoq.com/news/2013/03/Docker>)
- Running private repository (<https://blog.codecentric.de/en/2014/02/docker-registry-run-private-docker-image-repository/>)

Docker performance characteristics

- <http://bodenr.blogspot.in/2014/05/kvm-and-docker-lxc-benchmarking-with.html>

Budge around Docker

- Docker and Microsoft partner to bring container applications across platforms (<http://news.microsoft.com/2014/10/15/dockerpr/>)
- Google to offer Docker as a Service (using Kubernetes) (<http://siliconangle.com/blog/2014/11/06/with-docker-as-a-service-google-wants-developers-to-think-beyond-cloud/>)
- CoreOS (based on docker) as a lean and mean virtualization machine (<http://www.networkworld.com/article/2840343/opensource-subnet/coreos-a-lean-mean-virtualization-machine.html>)

Docker Acquisitions

- Docker acquires Koality (<http://techcrunch.com/2014/10/07/docker-acquires-koality-in-engineering-talent-grab/>)
- Docker acquires Orchard (<http://www.zdnet.com/docker-acquires-london-startup-orchard-laboratories-7000031921/>)

Miscellaneous

- Ubuntu server 14.10 adds new features for docker containers (<http://siliconangle.com/blog/2014/10/23/ubuntu-server-14-10-adds-new-features-for-openstack-containers/>)
- Automated Testing of Hardware Appliances with Docker (<http://www.appneta.com/blog/automated-testing-with-docker/>)



Questions?



Thanks

Empowered by Innovation

NEC