

Ceph: A Distributed File System

Audience/Presented to LC Staff Meeting

August 10, 2017



M. Tran, M. Wan, L. Zhang



Introduction

- What is a Distributed File System (DFS)?
 - A file system that permits various hosts on separate machines to access and share files through a computer network.
 - Data may be distributed across many nodes, but users can access their files as though they were stored on one server.
- Why use distributed file systems?
 - High availability
 - Redundancy
 - Location-independent access
 - Scalability
- Why Ceph?
 - Provides block and file storage
 - Can handle large-scale file systems
 - Reduces traffic to metadata clusters using CRUSH algorithm
 - POSIX compliant



Source: http://3.bp.blogspot.com/-B_UA0D016xl/T2ycHjkPdvI/AAAAAAAAAIs/nIm3cjmTwk/s1600/dfs.jpg

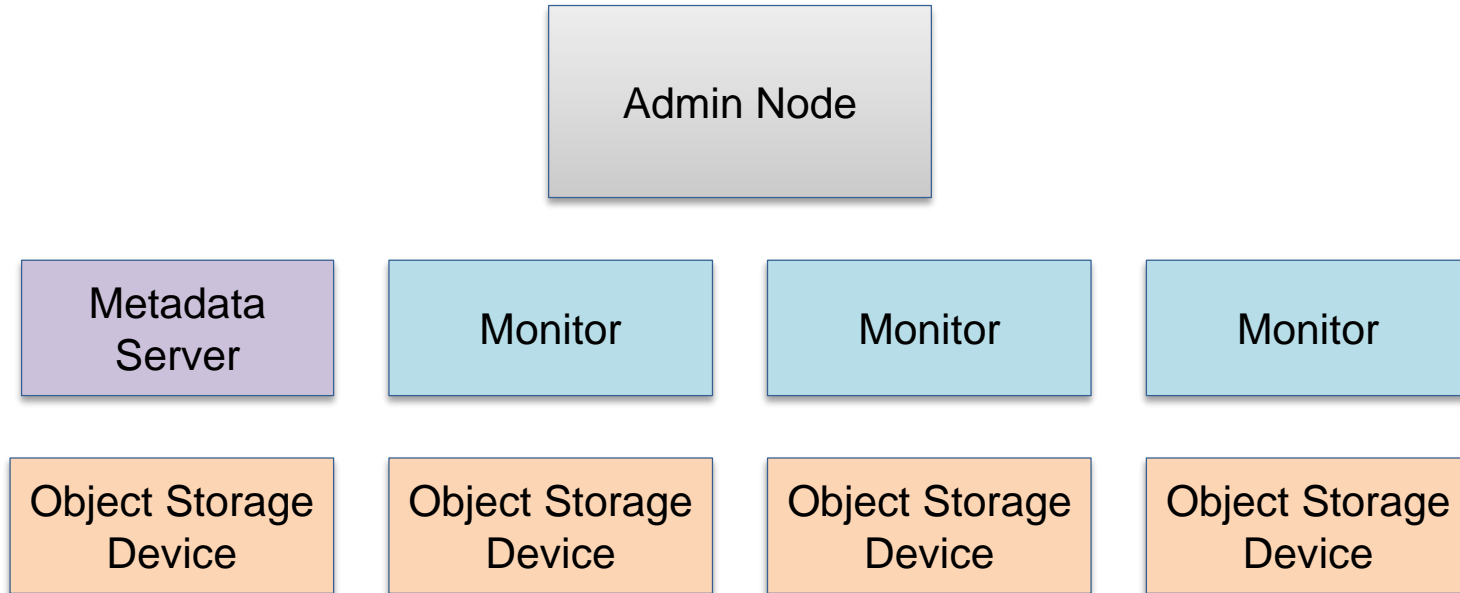
How does Ceph work?

- Components of the Ceph Storage Cluster
 - Monitors
 - Managers
 - Object Storage Daemons
 - Metadata Server (for use with Ceph File System)
- Stores data as objects within logical storage pools
- CRUSH algorithm
 - Controlled Replication Under Scalable Hashing
 - Determines which OSD stores the placement groups
 - Enables scaling, rebalancing, and recovery dynamically
- Ceph File System
 - POSIX-compliant interface
 - Files are mapped to objects and stored in the Ceph Storage Cluster.
 - Metadata Server prevents filesystem operations from consuming resources excessively

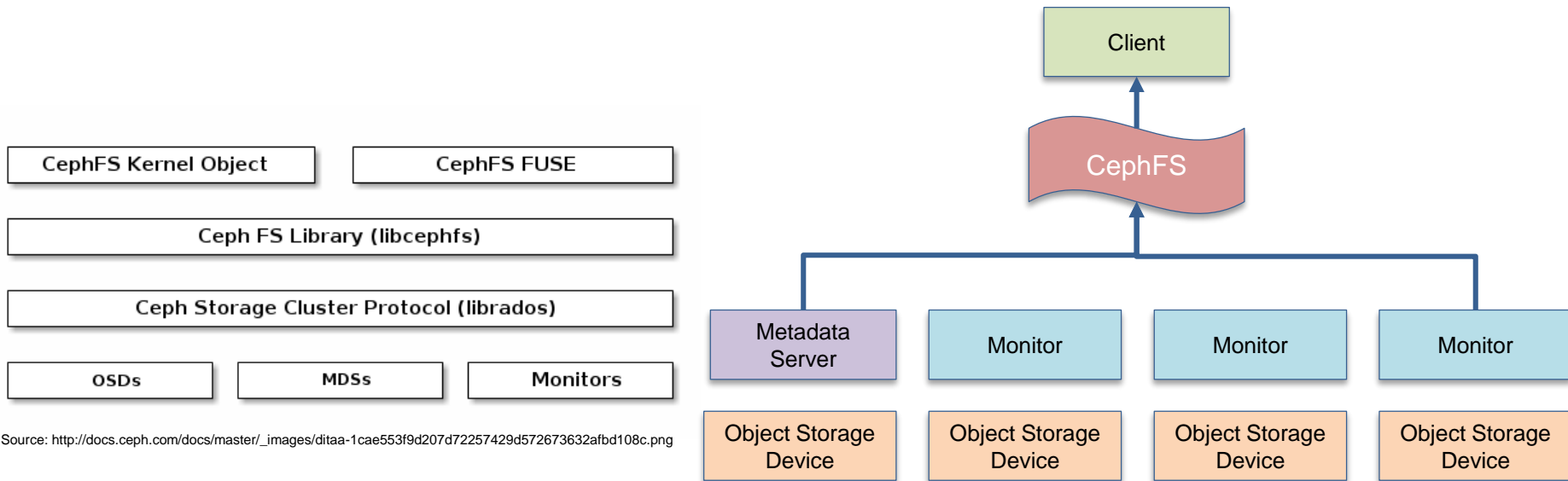


Source: http://ceph.com/wp-content/uploads/2016/07/Ceph_Logo_Stacked_RGB_120411_fa.png

Ceph Storage Cluster



Ceph File System



Benchmarking & Results

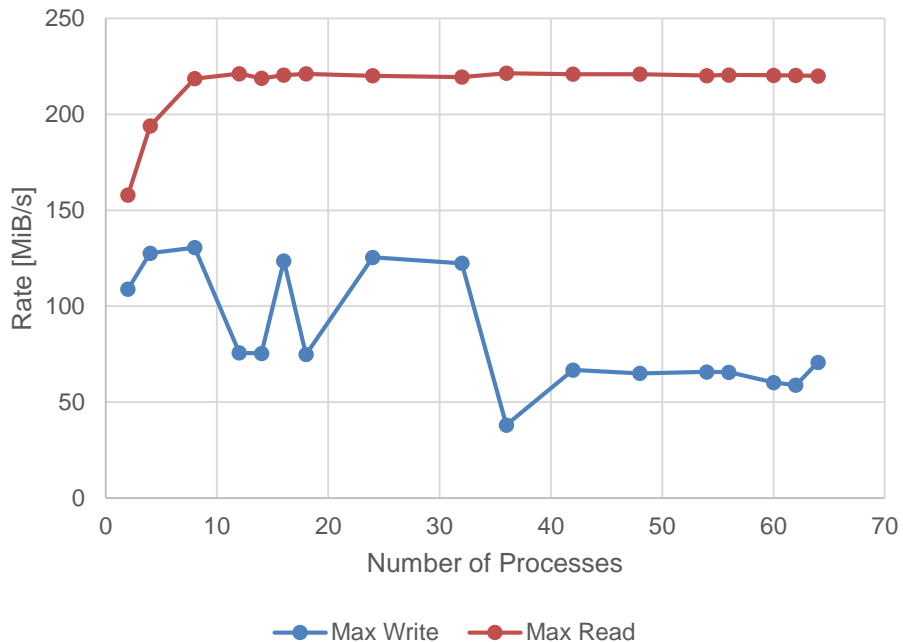
- Tested POSIX compliance using the POSIX Filesystem Test Suite
 - Passed 1951/1957 tests; failed 6/1957
 - Most UNIX systems aren't 100% POSIX compliant
- Tested read/write speeds using IOR
- Tested file creation/deletion speeds with mdtest



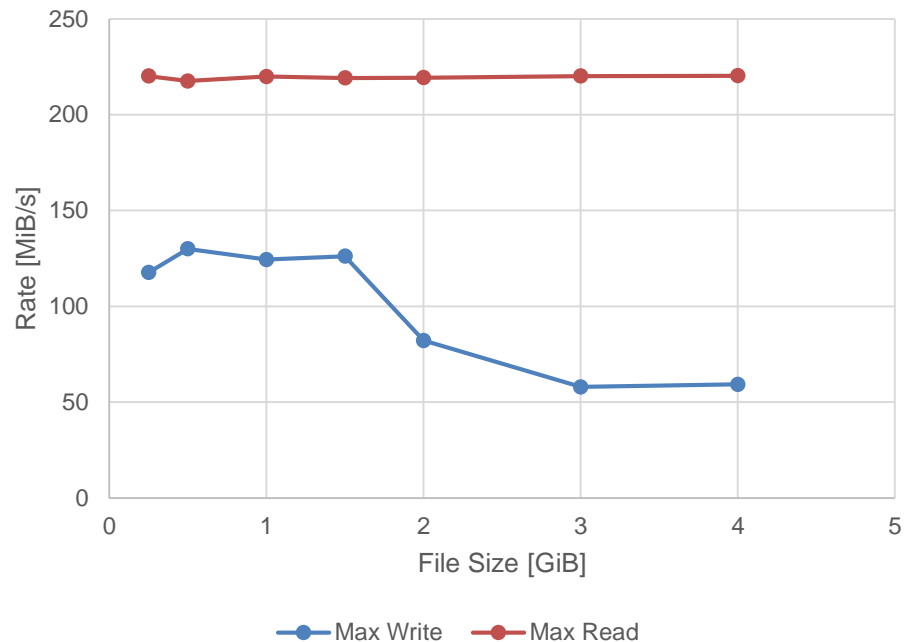
<https://www.iag.biz/wp-content/uploads/2016/08/223-Executive-Summary-Business-Analysis-Benchmark.jpg>

IOR Performance Testing

File Size = 512 MiB

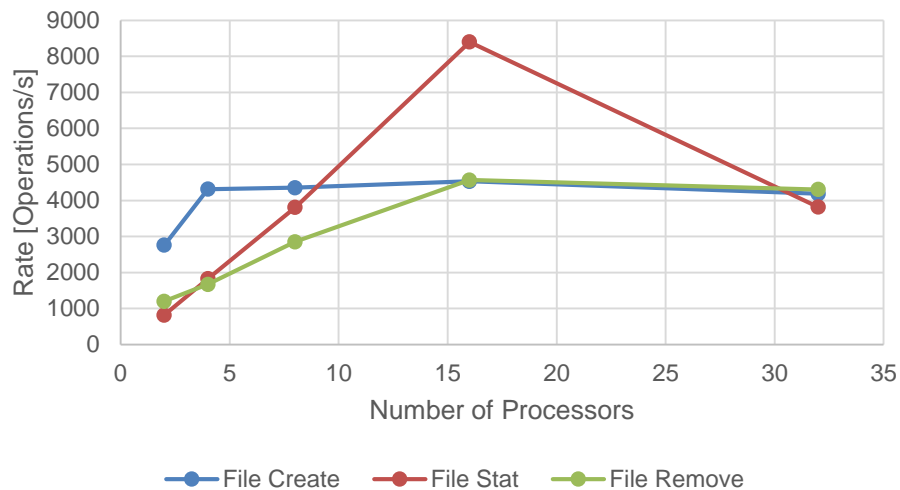


Processes = 8

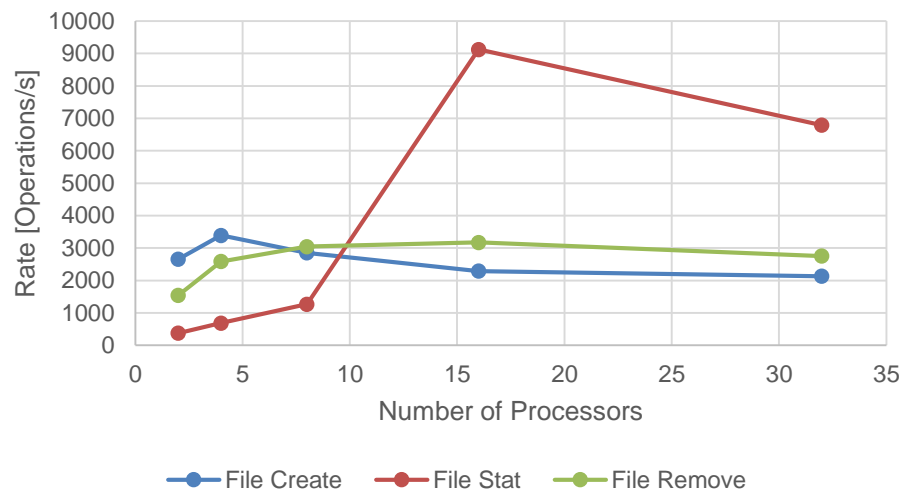


mdtest

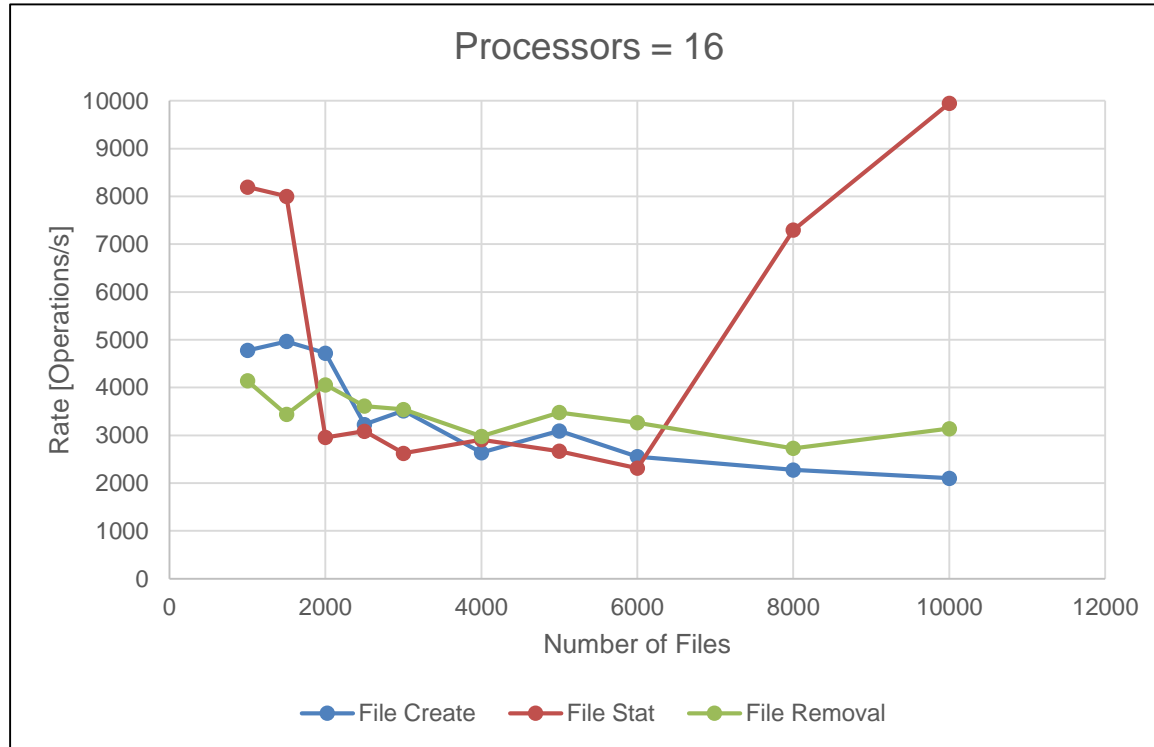
Files = 1,000



Files = 10,000



mdtest



Failover Testing

- Demonstrate that our implementation of CephFS can survive a failure of up to $\frac{1}{4}$ of the total system.
- Explore how the system responds when an OSD, a monitor, or a manager fails.



Source:
<http://www.istockphoto.com/illustrations/failover?excludenudity=true&sort=mostpopular&mediatype=illustration&phrase=failover>

Initial State: Healthy

```
[root@enickel9 ~]# ceph -s
cluster:
  id:          c7c85f67-7991-45c1-92b5-ace7f7b6344e
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum enickel5,enickel6,enickel7
  mgr: enickel7(active)
  mds: 1/1/1 up {0=enickel4=up:active}
  osd: 4 osds: 4 up, 4 in

data:
  pools:      2 pools, 256 pgs
  objects:    43 objects, 33109 kB
  usage:      21409 MB used, 2331 GB / 2351 GB avail
  pgs:        256 active+clean
```

After Taking Down an OSD

```
[root@enickel8 ~]# ceph -s
cluster:
  id:      c7c85f67-7991-45c1-92b5-ace7f7b6344e
  health: HEALTH_WARN
          1 osds down
          1 host (1 osds) down
          Degraded data redundancy: 31/129 objects degraded (24.031%),
199 pgs unclean, 199 pgs degraded, 199 pgs undersized

services:
  mon: 3 daemons, quorum enickel5,enickel6,enickel7
  mgr: enickel7(active)
  mds: 1/1/1 up {0=enickel4=up:active}
  osd: 4 osds: 3 up, 4 in; 199 remapped pgs

data:
  pools: 2 pools, 256 pgs
  objects: 43 objects, 33109 kB
  usage: 21441 MB used, 2330 GB / 2351 GB avail
  pgs: 31/129 objects degraded (24.031%)
      199 active+undersized+degraded
      57 active+clean
```

Recovered State

```
[root@nickeli ~]# ceph -s
cluster:
  id:          c7c85f67-7991-45c1-92b5-ace7f7b6344e
  health: HEALTH_OK

services:
  mon: 3 daemons, quorum enickel5,enickel6,enickel7
  mgr: enickel6(active)
  mds: 1/1/1 up {0=enickel4=up:active}
  osd: 4 osds: 3 up, 3 in

data:
  pools:      2 pools, 256 pgs
  objects:    43 objects, 33112 kB
  usage:      16164 MB used, 1748 GB / 1763 GB avail
  pgs:        256 active+clean
```

After Taking down a Monitor

```
[root@nickeli ~]# ceph -s
cluster:
  id:          c7c85f67-7991-45c1-92b5-ace7f7b6344e
  health: HEALTH_WARN
             no active mgr
             1/3 mons down, quorum enickel5,enickel6

services:
  mon: 3 daemons, quorum enickel5,enickel6, out of quorum: enickel7
  mgr: no daemons active
  mds: 1/1/1 up {0=enickel4=up:active}
  osd: 4 osds: 3 up, 3 in

data:
  pools:      2 pools, 256 pgs
  objects:    43 objects, 33109 kB
  usage:      16164 MB used, 1748 GB / 1763 GB avail
  pgs:        256 active+clean
```

Partially Recovered

```
[root@nickeli ~]# ceph -s
cluster:
  id:      c7c85f67-7991-45c1-92b5-ace7f7b6344e
  health: HEALTH_WARN
          1/3 mons down, quorum enickel5,enickel6

services:
  mon: 3 daemons, quorum enickel5,enickel6, out of quorum: enickel7
  mgr: enickel6(active)
  mds: 1/1/1 up {0=enickel4=up:active}
  osd: 4 osds: 3 up, 3 in

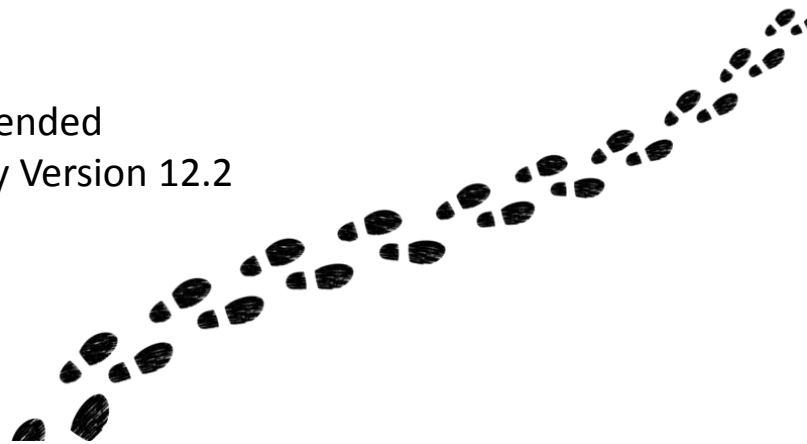
data:
  pools: 2 pools, 256 pgs
  objects: 43 objects, 33109 kB
  usage: 16164 MB used, 1748 GB / 1763 GB avail
  pgs: 256 active+clean
```

Discussion

Challenges	Solutions
LVMs already installed	Installed OSDs manually
Benchmarking: Tests read from cache	Re-ran tests using 2 clients
New version released halfway through	Updated all of our nodes to Version 12.1.2
Not enough troubleshooting documentation	Trial and error; reinstalling Ceph

Next Steps

- Integrate Ceph with NFS
 - We would like to mount CephFS on clients that don't have Ceph installed.
 - Currently, we do this by having one node of the cluster act as a NFS server.
 - This methods is flawed: if the NFS server goes down, clients lose access to the file system.
- Improve performance, particularly write speeds
- Incorporate additional metadata servers
 - Multiple metadata servers is not currently recommended
 - Ceph plans to support multiple metadata servers by Version 12.2



Thank you for your help and support!

Thomas Bennett

Elsa Gonsiorowski

Dave Fox

Geoff Cleary

Bryan Dixon

Pam Hamilton

Go Team Cephalopod!



Source: <https://www.pinterest.com/pin/445504588117025745>





Elastic Stack Installation & Configuration

Anna Gassen, Ciara Goetze, James Gadson III
Team G Code



Objective

- Install and configure Elastic Stack on the Academy clusters
- Gather logs from all nodes
- Develop some insightful searches
- Research data analysis concepts

Elastic Stack

You know, for search

- Our clusters produce more than 1500 log messages per minute
- Comprised of six open-source tools: Elasticsearch, Logstash, Kibana, Beats, X-Pack, Elastic Cloud
- Allows quick analyzation, visualization, and mining of millions of log files
- Identify trends, statistics, and abnormalities



Logstash



- Collects data from many different sources at the same time
- Filters and parses each message, converts it into a common format for easier analysis
- Aggregates and transports data to Elasticsearch (or the software of your choice)



Filebeat



- A lightweight log file shipping agent
- Part of the Beats family of data shippers
- Communicates directly with Logstash or Elasticsearch
- Easily forwards and centralizes log files

Elasticsearch

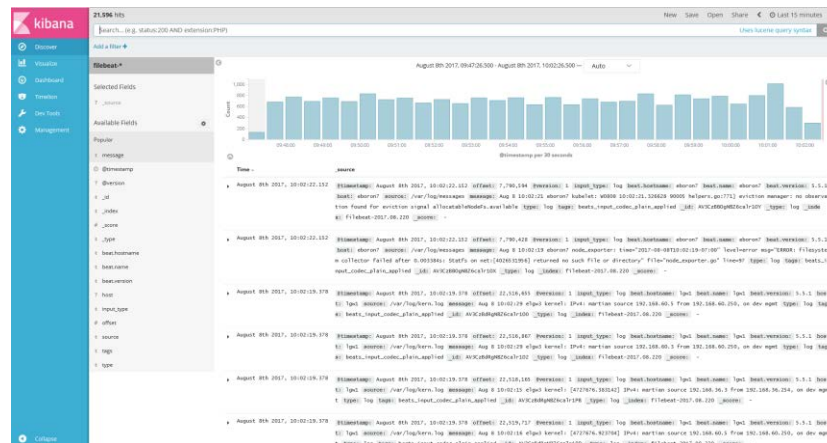


- Full-text search engine that searches and centrally stores data
- Quickly find, retrieve, and analyze big volumes of data
- Distributed and highly scalable
- Near real time search
- Uses RESTful API, JSON, and Lucene

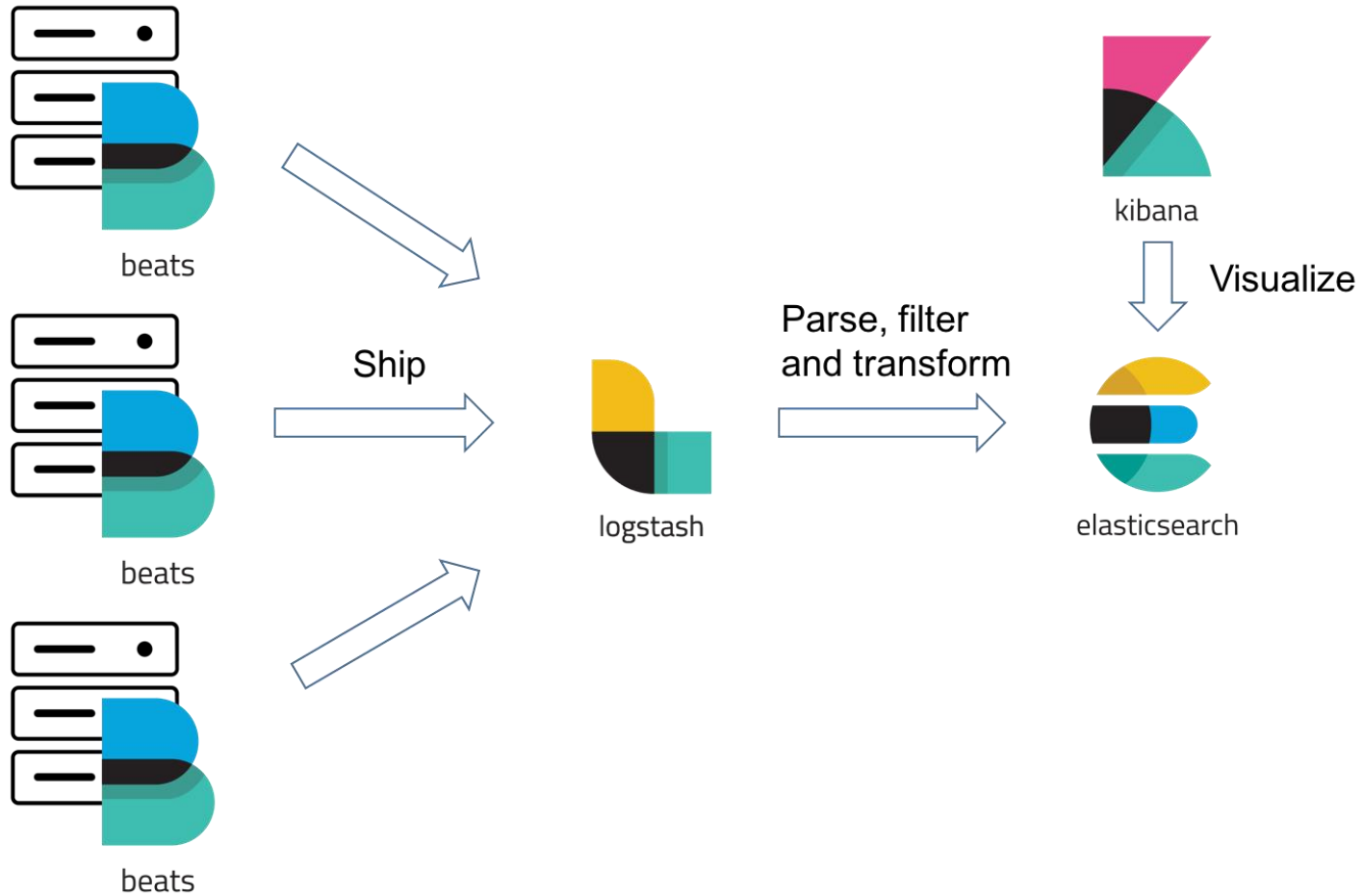
Kibana



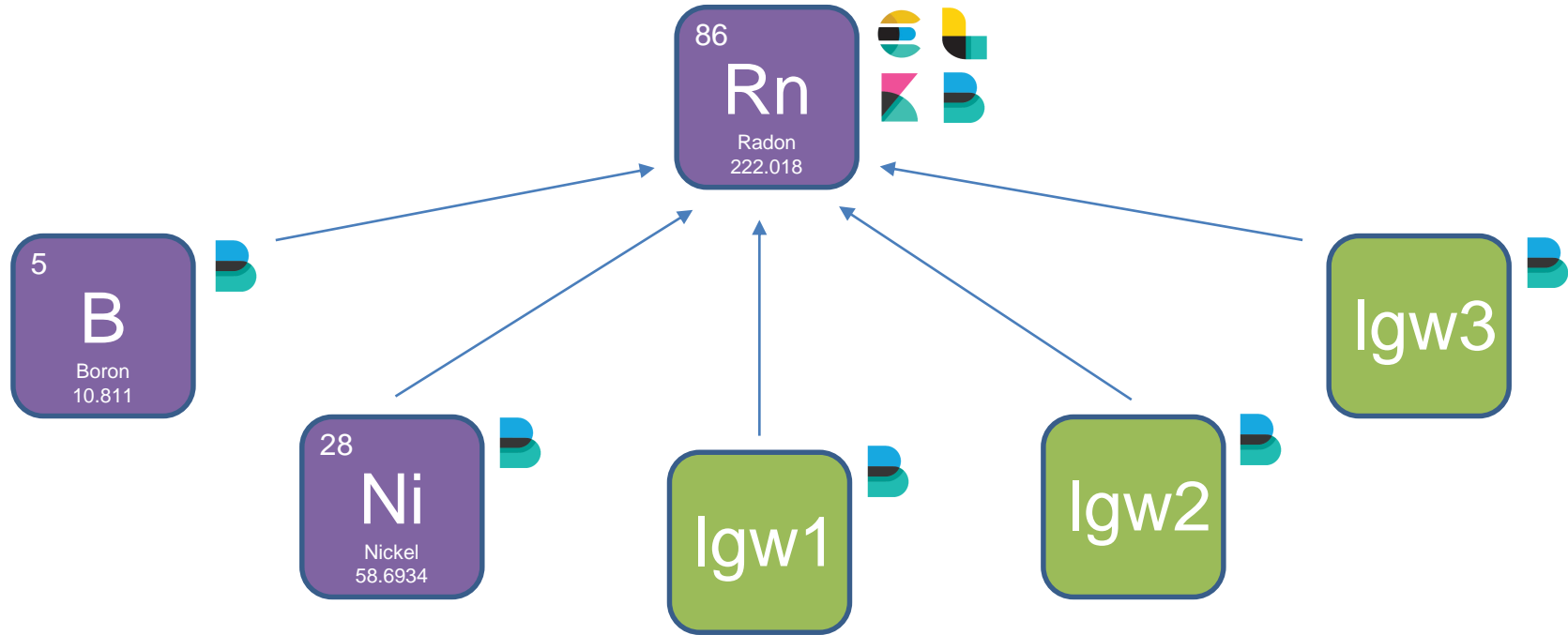
- Data visualization tool for log and time series analytics
- Makes navigation and monitoring of logs more intuitive
- Provides numerous graph and dashboard options to display information



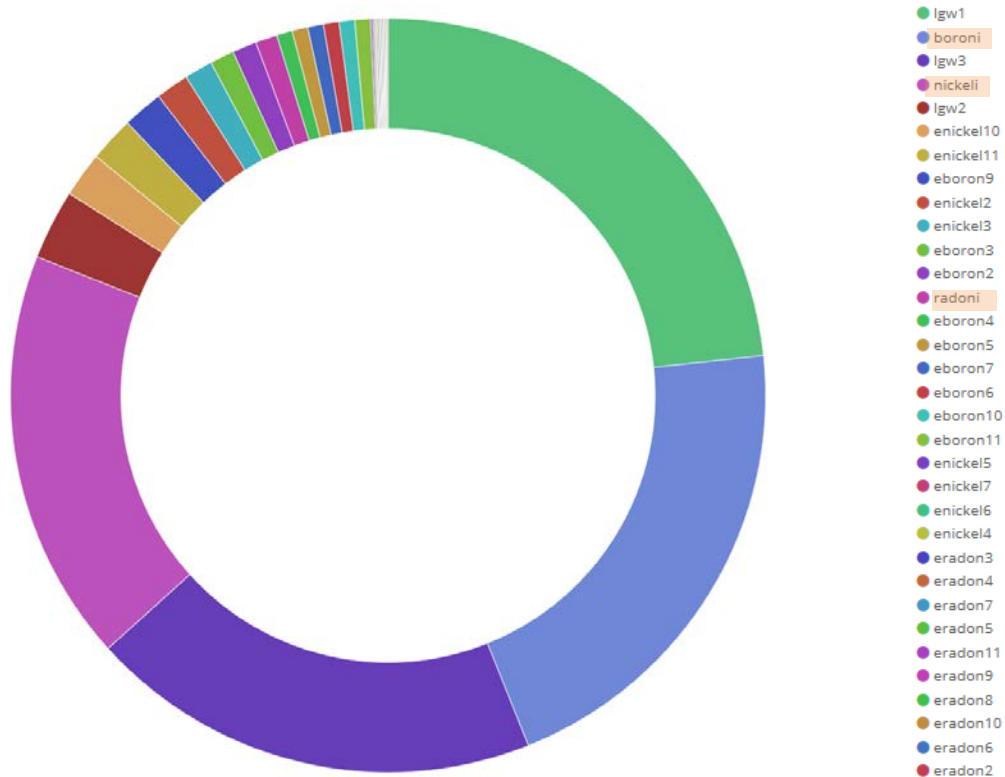
```
1 {
2   "_index": "filebeat-2017.08.220",
3   "_type": "log",
4   "_id": "AV3CzBB0gNBZ6calr10Y",
5   "_score": 1,
6   "_source": {
7     "@timestamp": "2017-08-08T17:02:22.152Z",
8     "offset": 7790594,
9     "@version": "1",
10    "input_type": "log",
11    "beat": {
12      "hostname": "eboron7",
13      "name": "eboron7",
14      "version": "5.5.1"
15    },
16    "host": "eboron7",
17    "source": "/var/log/messages",
18    "message": "Aug  8 10:02:21 eboron7 kubelet: W0808 10:02:21.326628  90005 helpers.go:771] eviction manager: no observation found for eviction signal allocatableNodeFs.available",
19    "type": "log",
20    "tags": [
21      "beats_input_codec_plain_applied"
22    ]
23  },
24  "fields": {
25    "@timestamp": [
26      1502211742152
27    ]
28  }
29 }
```



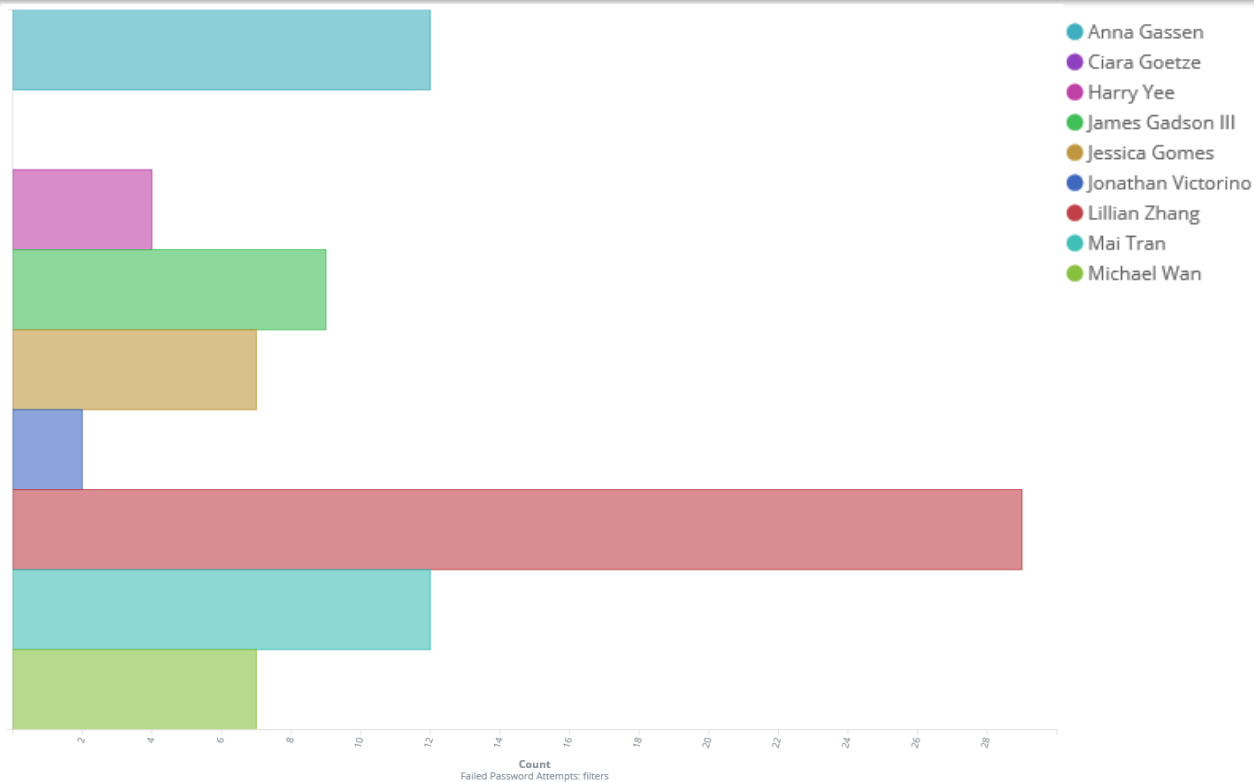
Approach



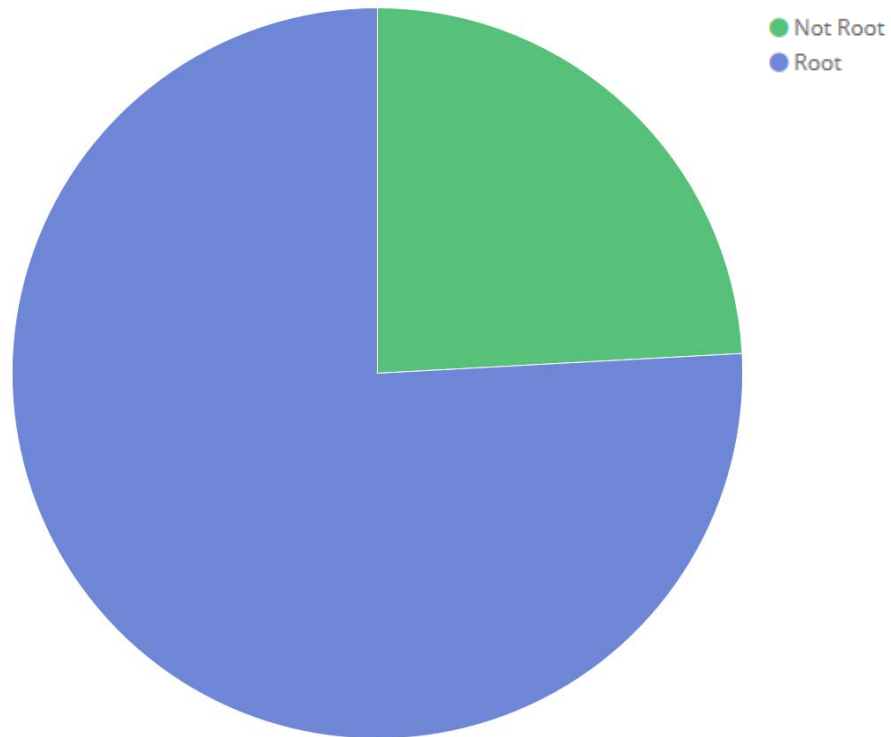
Number of Documents per Node



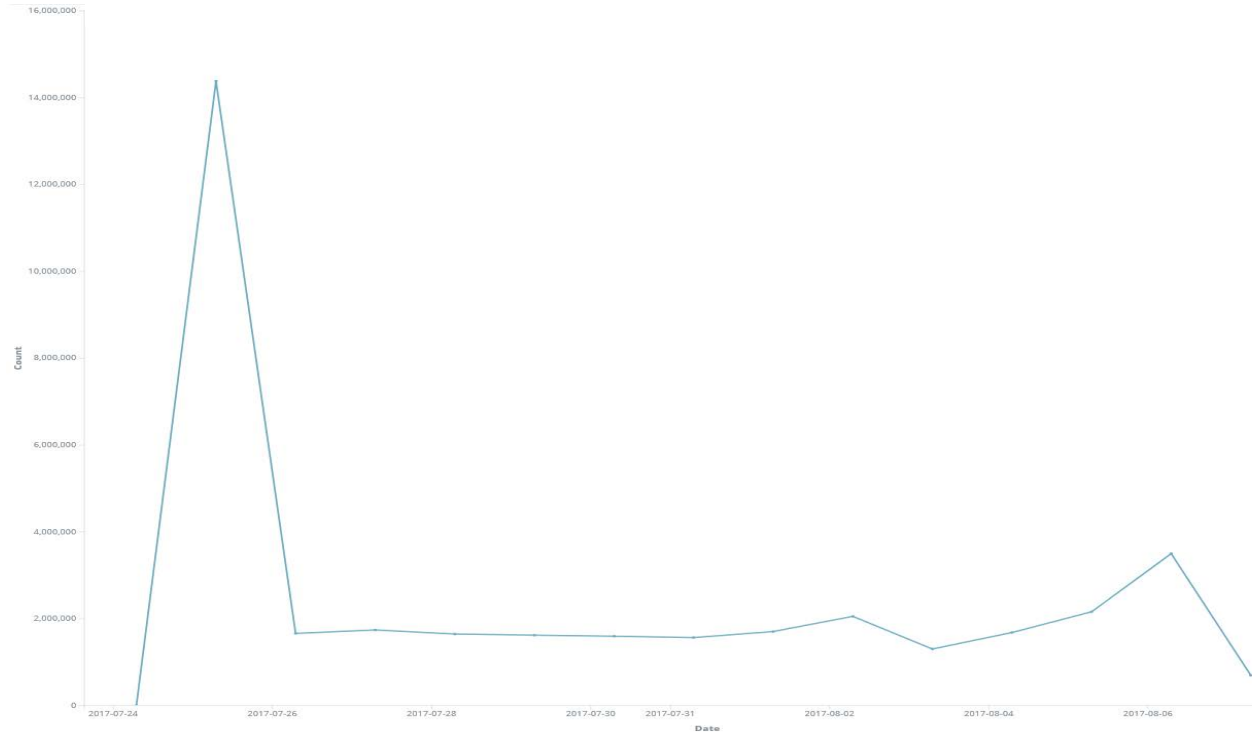
Failed Login Attempts



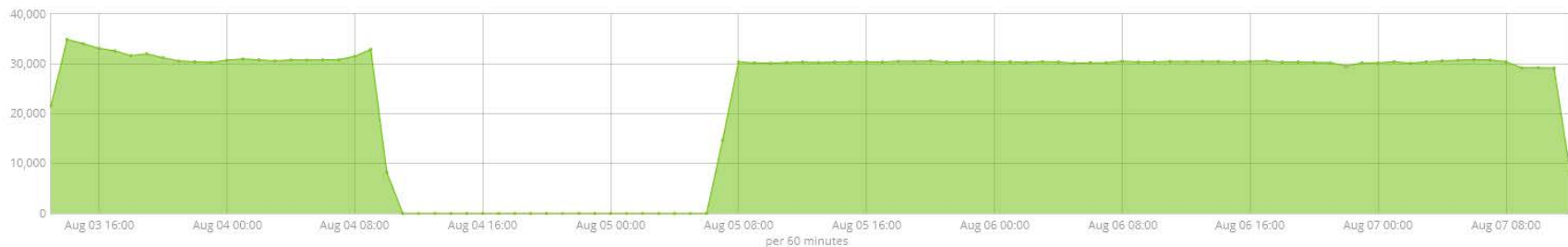
Root vs Non-Root Logins



Number of Documents per Day



Martian Source Warnings



```
August 7th 2017, 13:22:09.461 @timestamp: August 7th 2017, 13:22:09.461 offset: 35,971,095 @version: 1 beat.hostname: lgw1 beat.name: lgw1 beat.version: 5.5.1 input_type: log host
t: lgw1 source: /var/log/kern.log message: Aug 7 13:22:23 elgw3 kernel: IPv4: martian source 192.168.36.2 from 192.168.36.254, on dev mgmt type: log tag
s: beats_input_codec_plain_applied _id: AV2-XXtGgNBZ6calhzUk _type: log _index: filebeat-2017.08.219 _score: -
```

Future work

- Research Logstash pipeline configuration options
- Utilize Beats and X-Pack
- Perform more complex Elasticsearch queries
- Configuring Elastic Stack to be useful to future Academy interns

Acknowledgements

- David Fox
- Geoff Cleary
- Pam Hamilton
- Bryan Dixon
- Richard Randall



Kubernetes Implementation into HPC

Livermore Computing

August 10, 2017

Jessica Gomes, Jonathan Victorino, Harry Yee
HPC CEA Interns



Meet Team Kubes



Jessica Gomes
University of Illinois at
Urbana-Champaign

Harry Yee
UCLA

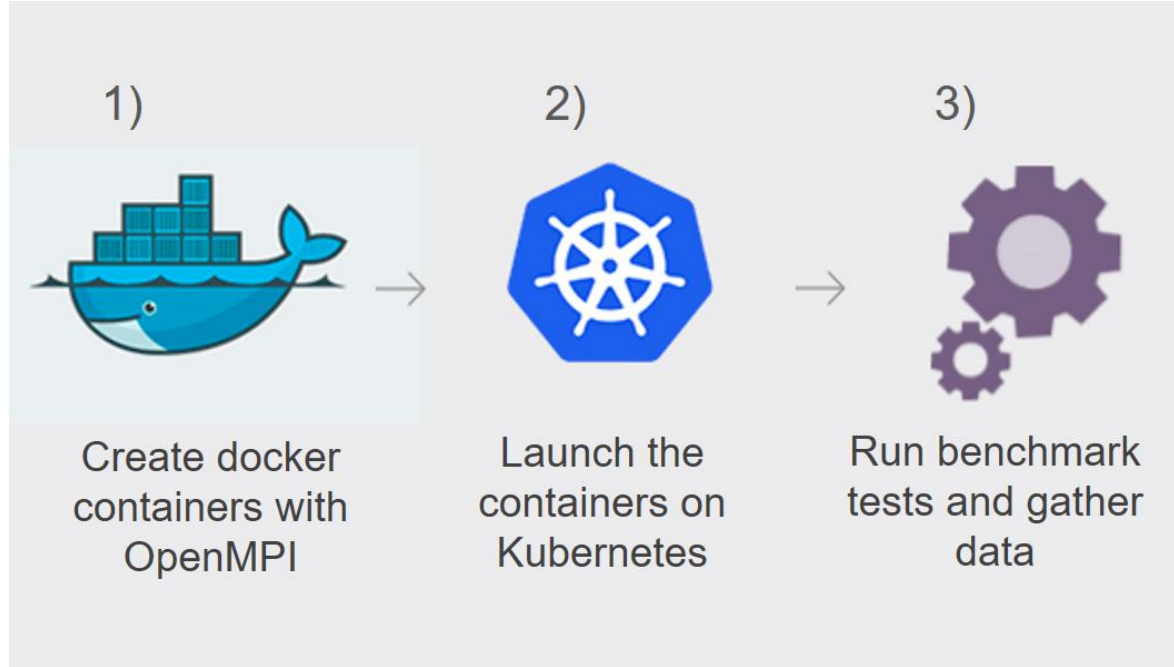
Jonathan Victorino
Stanford University

Objectives

- I. Setup a Kubernetes cluster using Docker containers
- II. Run LINPACK benchmark tests comparing Kubernetes and bare metal cluster
- III. Determine performance overhead of running the cluster in Kubernetes
- IV. Automate Kubernetes builds and deployments using Puppet

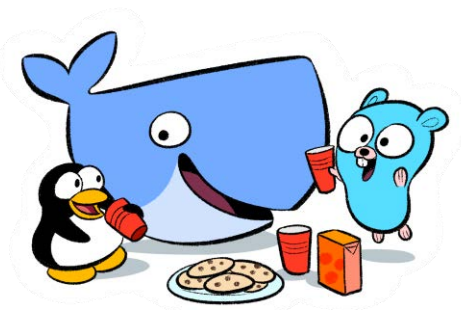


Our Project





- World's leading software container platform
- **Container**: stand-alone package that includes everything needed to run it
- Easily create images for containers using Dockerfiles
- Use Docker Hub to automatically update the containers



What is Kubernetes?

- Open-source system for automating deployment, scaling, and management of containerized applications
- Schedule and deploy any number of Docker container replicas onto a node cluster



The New York Times

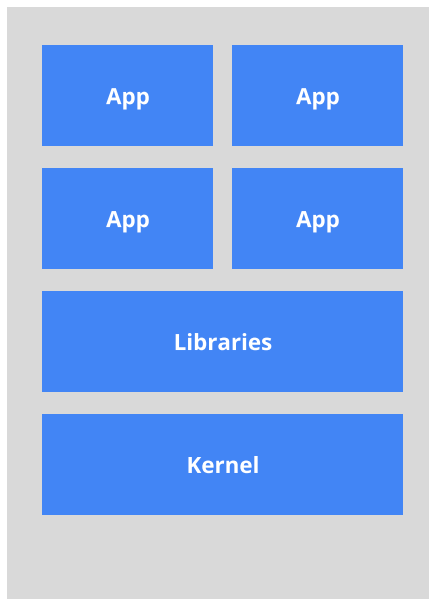


kubernetes



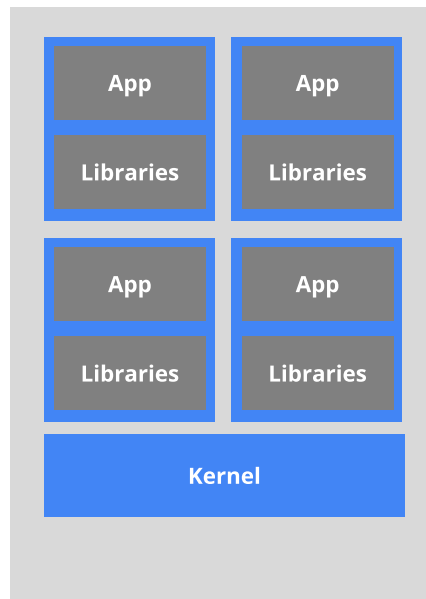
Kubernetes Container Visualization

The old way: Applications on host



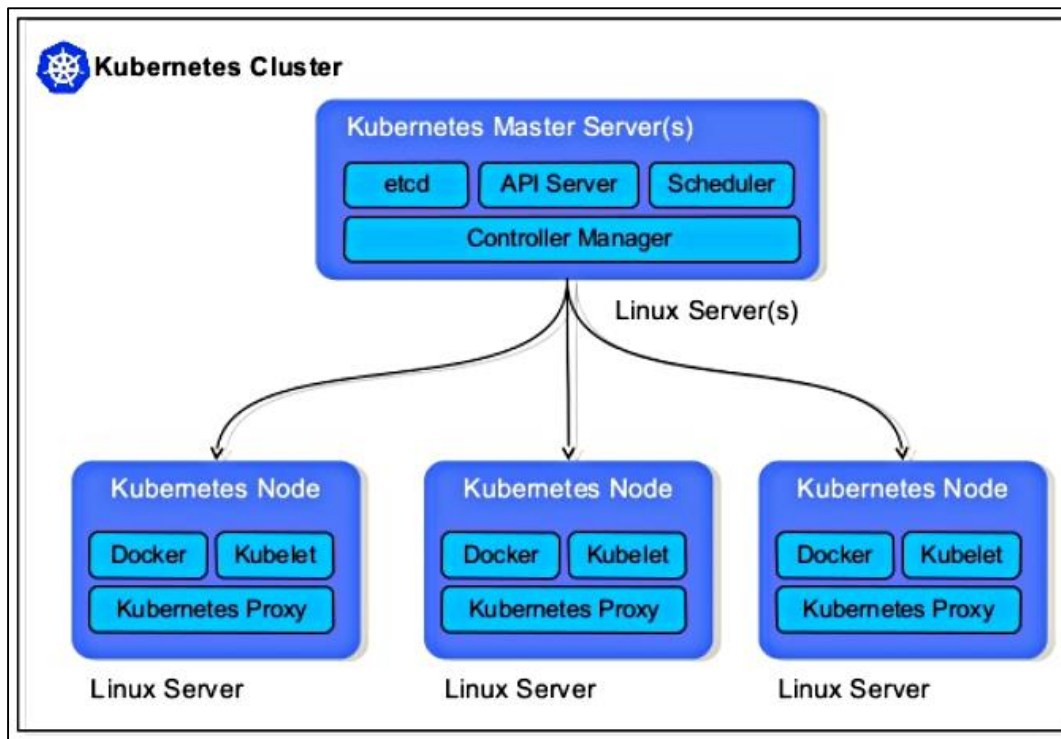
*Heavyweight, non-portable
Relies on OS package manager*

The new way: Deploy containers



*Small and fast, portable
Uses OS-level virtualization*

Kubernetes Architectural Overview



Installation Process



```
root@boroni:~/boron_puppet/manifests
ensure => "file",
owner => "root",
group => "root",
mode => "644",
source => "/root/boron_puppet/tftpboot/pxelinux.cfg/default",
}

file { ['/etc/dhcp/dhcpd.conf']:
  ensure => "file",
  owner => "root",
  group => "root",
  mode => "644",
  source => "/root/boron_puppet/etc/dhcp/dhcpd.conf",
}

$packages = ['ntp', 'openmpi', 'openmpi-devel', 'slurm', 'munge',
  'nfs-utils', 'sudo', 'yum-utils', 'device-mapper-persistent-data', 'lvm2']
package { $packages:
  ensure => installed,
}

exec {'dockerInstall':
  command => "/root/boron_puppet/scripts/dockerInstallScript.sh",
  #unless => 'test -e /usr/bin/docker',
}

package {'docker-ce':
  ensure => installed,
}

exec {'kubeInstall':
  command => "/root/boron_puppet/scripts/kubeInstallScript.sh",
  #unless => 'test -e /usr/bin/kubelet && test -e /usr/bin/kubeadm',
}

$kubepackages = ['kubelet', 'kubeadm']
package { $kubepackages:
  ensure => installed,
}
```

Puppet Manifest File

```
root@boroni:~/boron_puppet/scripts
#!/bin/bash

# Docker installation
sudo yum-config-manager --add-repo
https://download.docker.com/linux/
centos/docker-ce.repo

sudo yum makecache fast
sudo yum -y install docker-ce
sudo systemctl start docker
sudo systemctl enable docker
```

Docker Installation Script

```
root@boroni:~/boron_puppet/scripts
#!/bin/bash

# Kubectl Installation
curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s
https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl

# Change Permissions
chmod +x ./kubectl
sudo mv ./kubectl /usr/local/bin/kubectl

# Install Packages Kubelet and Kubeadm
touch /etc/yum.repos.d/kubernetes.repo
sudo echo "[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg
https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg" >
/etc/yum.repos.d/kubernetes.repo

# Disable SELinux Enforcement
setenforce 0
```

Kubernetes Packages Installation Script

.yaml Files

Branch: master MPI-Kubernetes-Cluster / master.yaml

Find file Copy path

U-THE-LAB/dixon30 Reduced CPU amount to 15 vs 16

9603fcf 23 hours ago

2 contributors @h @j

Executable File 65 lines (64 sloc) 1.59 KB

Raw Blame History

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: mpi-master
5   labels:
6     service: mpi-master
7 spec:
8   type: NodePort
9   ports:
10  - port: 22
11    nodePort: 30001
12    protocol: TCP
13  selector:
14    app: mpi-master-deployment
15 ---
16 apiVersion: apps/v1beta1
17 kind: Deployment
18 metadata:
19   # Unique key of the Deployment instance
20   name: mpi-master-deployment
21 spec:
22   # 3 Pods should exist at all times.
23   replicas: 1
24   template:
25     metadata:
26       labels:
27         # Apply this label to pods and default
28         # the Deployment label selector to this value
29         app: mpi-master-deployment
```

Branch: master MPI-Kubernetes-Cluster / worker.yaml

Find file Copy path

U-THE-LAB/dixon30 Reduced CPU amount to 15 vs 16

9603fcf 23 hours ago

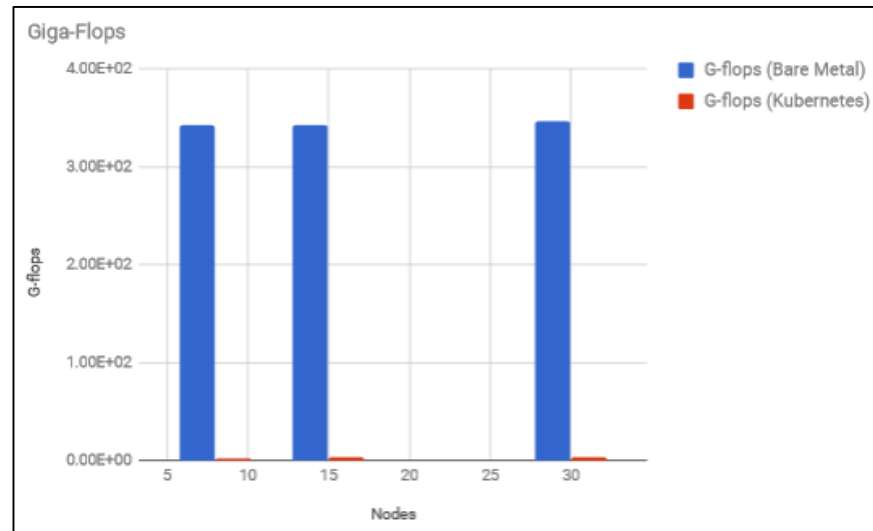
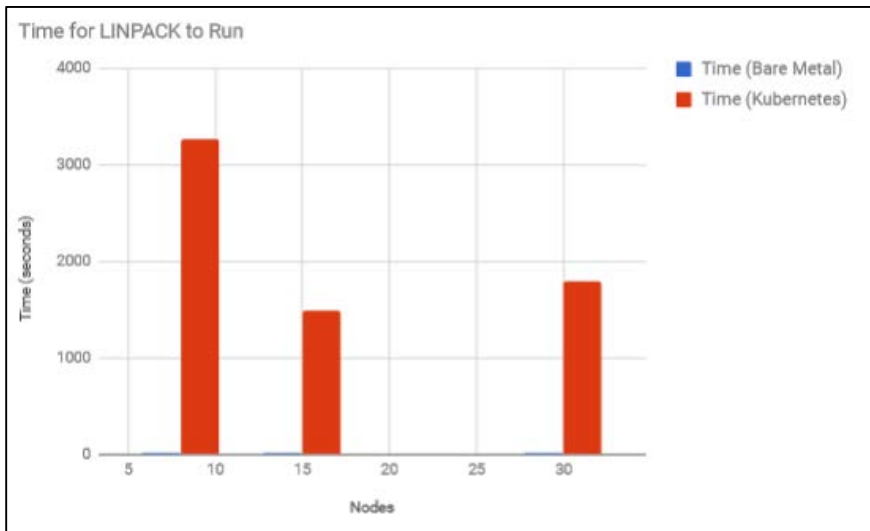
2 contributors @h @j

Executable File 39 lines (38 sloc) 988 Bytes

Raw Blame History

```
1 apiVersion: apps/v1beta1
2 kind: Deployment
3 metadata:
4   # Unique key of the Deployment instance
5   name: mpi-worker
6 spec:
7   # 3 Pods should exist at all times.
8   replicas: 5
9   template:
10    metadata:
11      labels:
12        # Apply this label to pods and default
13        # the Deployment label selector to this value
14        app: mpi-worker
15    spec:
16      volumes:
17        - name: ssh-keys
18          secret:
19            secretName: ssh-key-secret
20        # - name: test-volume
21        # persistentVolumeClaim:
22        #   claimName: test-pvc
23    containers:
24      - name: kube-mpi
25        # Run this image
26        image: javawolfpack/base-centos-mpi
27        imagePullPolicy: Always
28        ports:
29          - containerPort: 22
```


LINPACK Benchmarks



	Time (seconds)	G-flops
Bare Metal	15.39	3.47E+02
Kubernetes	1796.654	2.99E+00

Conclusion

- Kubernetes
 - Large performance overhead on clusters
 - Extra setup steps
 - Not ideal for HPC
- Use bare metal cluster
- After further testing, the bottleneck was found to be due to a networking issue
- Time and G-flops were comparable for containers running on the same node to the bare metal configuration



Future Plans

- Use Puppet to automate builds and deployments
- Automatically ssh from worker to master
- Run more Benchmark tests on Kubernetes cluster
- Find out how to fix the bottleneck and improve communication between containers



