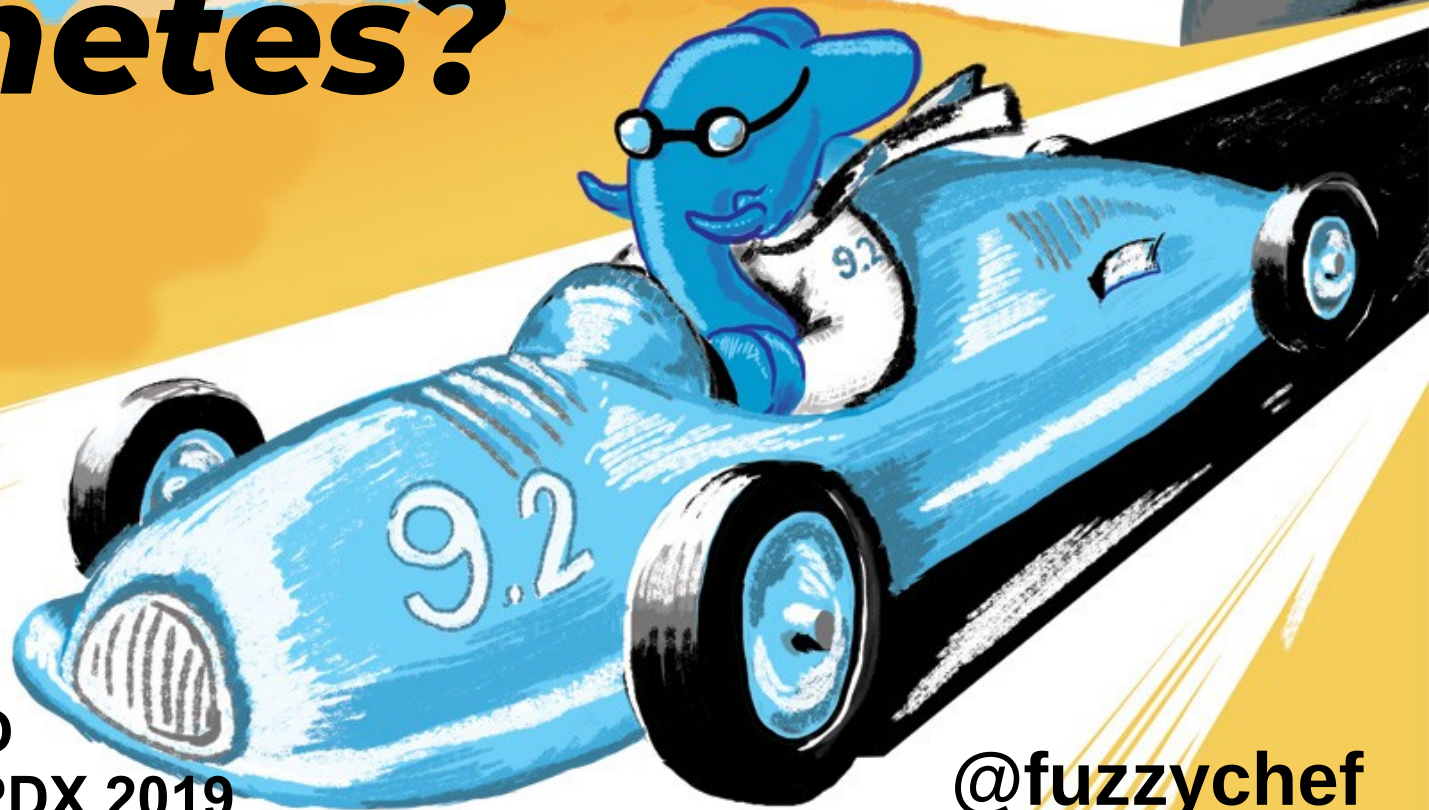


How Fast Was My Kubernetes?



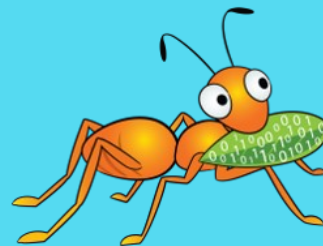
Red Hat

Josh Berkus
Red Hat OSPO
DevOpsDaysPDX 2019

@fuzzychef

Chris Johnson 2012

Kube Goes Stateful



STATEFUL APPLICATIONS

STATEFUL APPS EVERYWHERE

How Fast?

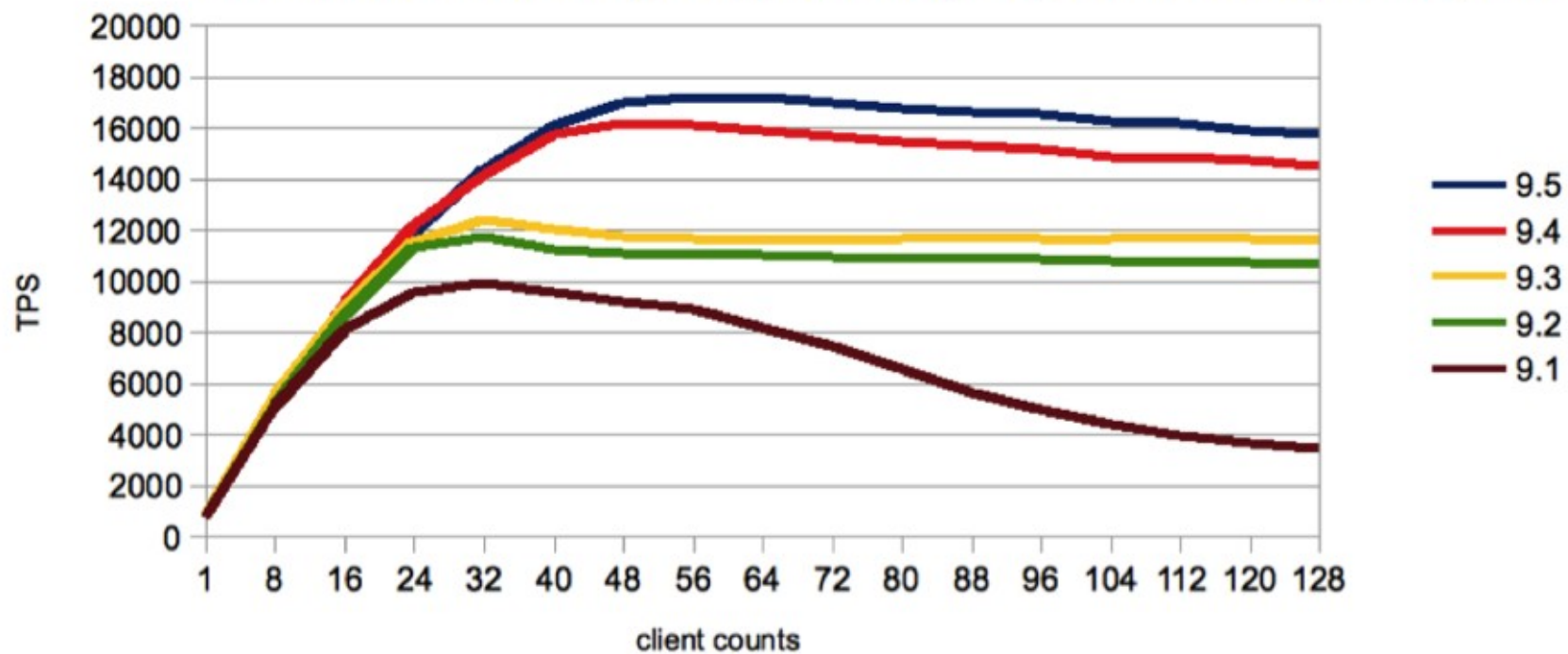
Uses kubetest to run k8s.io/perf-tests/run-e2e.sh against a 5000-node cluster created with `cluster/kube-up.sh`

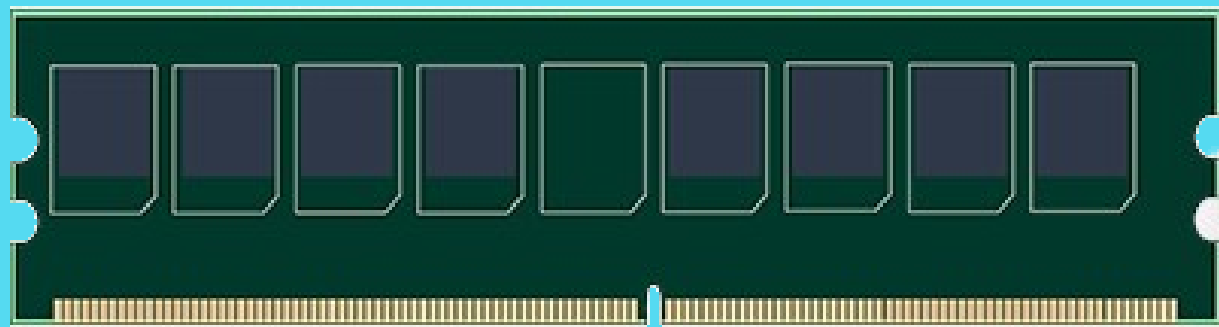
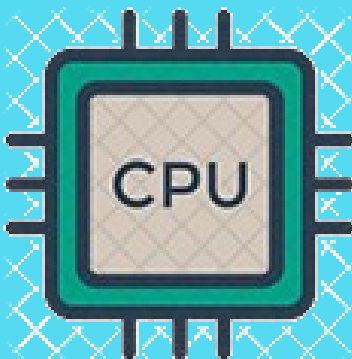
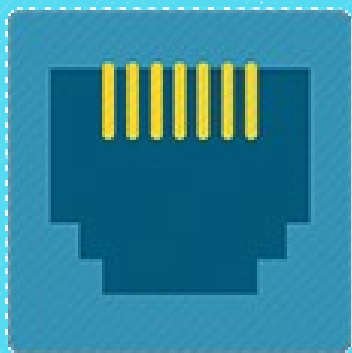
09-06 10:02 PDT @1170019332915204099 -- 07-09 10:02 PDT @1148638438308712451; Served from cache in 0.

About ▾	Size ▾	Options ▾	09-06	09-05	09-04	09-03	09-02	09-01	08-31
Graph ▾	Local Time: OFF		10:02 PDT	10:01 PDT	10:02 PDT				
Display Clustered Failures List			1170019332911	169656830511	169294583111	1689321896311	1685698191311	1682074298511	16784505802
			38752f7f9	100608f44	84fe3db5c	975d0736b	c7c89f8c6	11678fb1c	3d17fd5c4
				00fe6c3b6	7e3e6d3fa		fa46df691	b4aa999c2	
Show 1 stale tests (no results in 10m)									
ClusterLoaderV2									
Overall			R			R			
testing/density/config.yaml									
testing/load/config.yaml									
Up									
Check APIReachability									
Deferred TearDown									
DumpClusterLogs									
Extract									

pgbench -M prepared

median of 3 30-minute runs, scale_factor=1000, max_connection=200, shared_buffer=8GB.





Database Tests

- MySQL Sysbench:
basic IO stats
- Postgres pgBench:
load time & random writes
- CockroachDB TPCC:
lock-bound complex txns

6 blade cluster

20 cores ea.

128 GB RAM

2 SSDs w/ 200GB ea.

shared network

The Stats

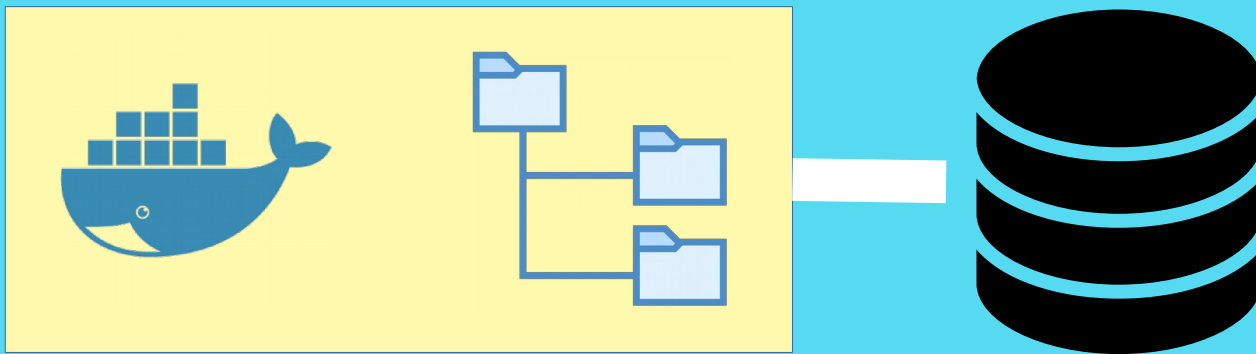
sysbench	seq. writes	random reads/s	random writes/s
pgbench	db load time	txns /sec	avg latency
tpcc	N/A	new orders /sec	95% latency

Bare Metal (control)

sysbench	88.4_{mb}/s	10725/s	7160/s
pgbench	404s	11282/s	2.8ms
tpcc	N/A	1290_{/m}	36.7ms

Local Volumes

Kubernetes Node



Local Volumes

sysbench	88.1_{mb/s} - 0.4%	10720/s -0.01%	7157/s -0.01%
pgbench	446s +10.4%	9657/s -14.5%	3.3ms +17%
tpcc	N/A	1290_{/m} 0%	52.4ms +12.2%

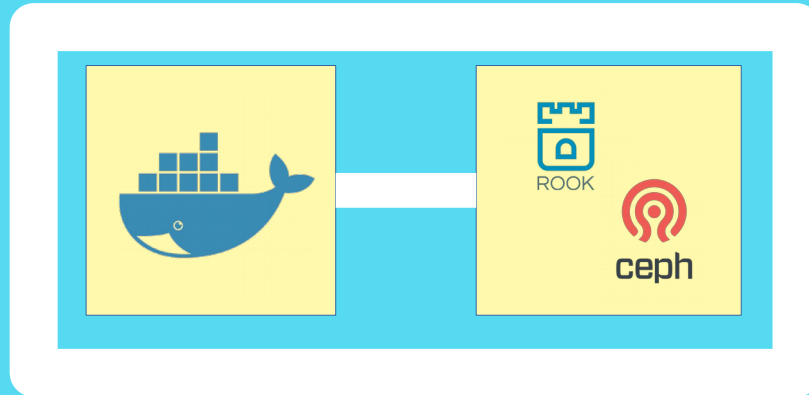
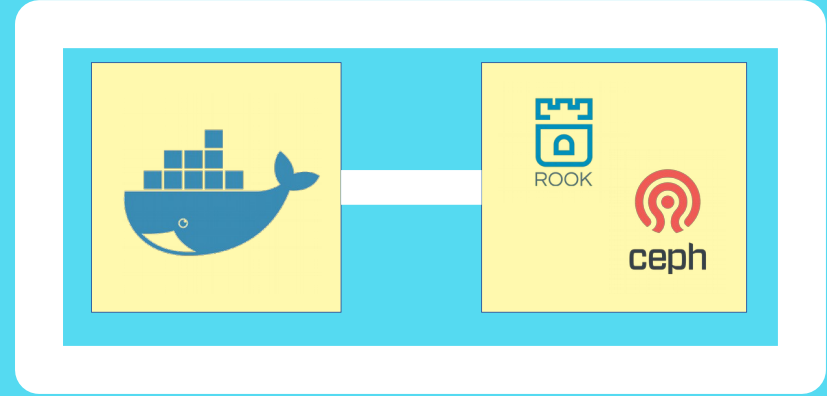
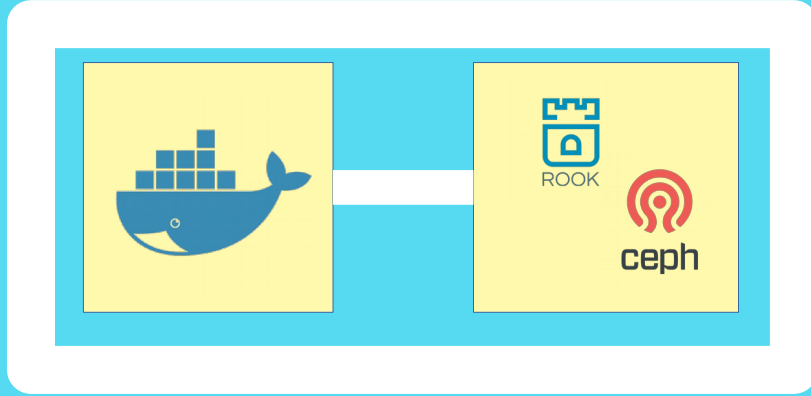
Local Volumes

sysbench	88.1mb/s	10720/s	7157/s
	- 0.4%	-0.01%	-0.01%
pgbench	446s	9657/s	3.3ms
	+10.4%	-14.5%	+17%
tpcc	N/A	1290/m	52.4ms
		0%	+12.2%

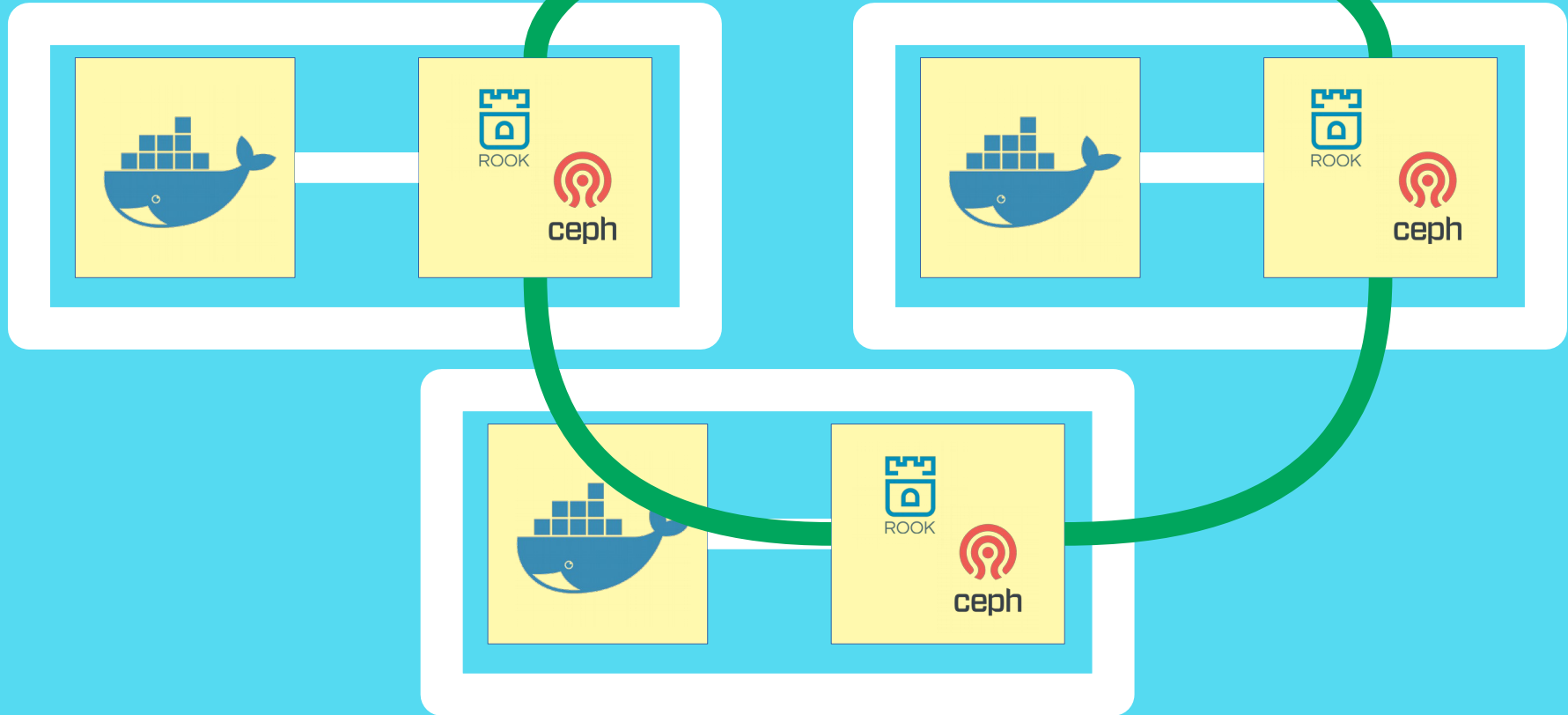
network latency

- (1) used NodePort in order to run pgbench client on bare metal
- (2) extra network hops added command latency
- (3) pgbench sends a lot of short commands, with no batching

Cloud-Native Storage



Cloud-Native Storage



Cloud-Native Storage

sysbench	111mb/s + 25%	9363/s -17%	6252/s -13%
pgbench	611s +28%	4466/s -54%	7.1ms +115%
tpcc	N/A	1290/m 0%	117ms +150%

Cloud-Native Storage

sysbench	111_{mb/s} + 25%	9363/s -17%	6252/s -13%
pgbench	611s +28%	4466/s -54%	7.1ms +115%
tpcc	N/A	1290_{/m} 0%	117ms +150%

Conclusions

- Need performance? Use Local Volumes
 - (but be prepared to handle your own failover)
- Cloud-Native storage doubles latency in exchange for redundancy/availability.
- Network configuration can add as much latency as storage.

contact/copyright

- Come see us in the Red Hat table!
- Josh Berkus:
 - www.databasesoup.com
 - jberkus@redhat.com
 - @fuzzychef on Twitter
 - @jberkus on Slack



This presentation is copyright 2019 Josh Berkus and Red Hat Inc. It is licensed Creative Commons Share Alike 4.0. The Racecar image is property PostgreSQL Project, and is licensed Creative Commons Attribution. Buzz Lightyear is property of the Disney corporation and is used here as parody fair use.

