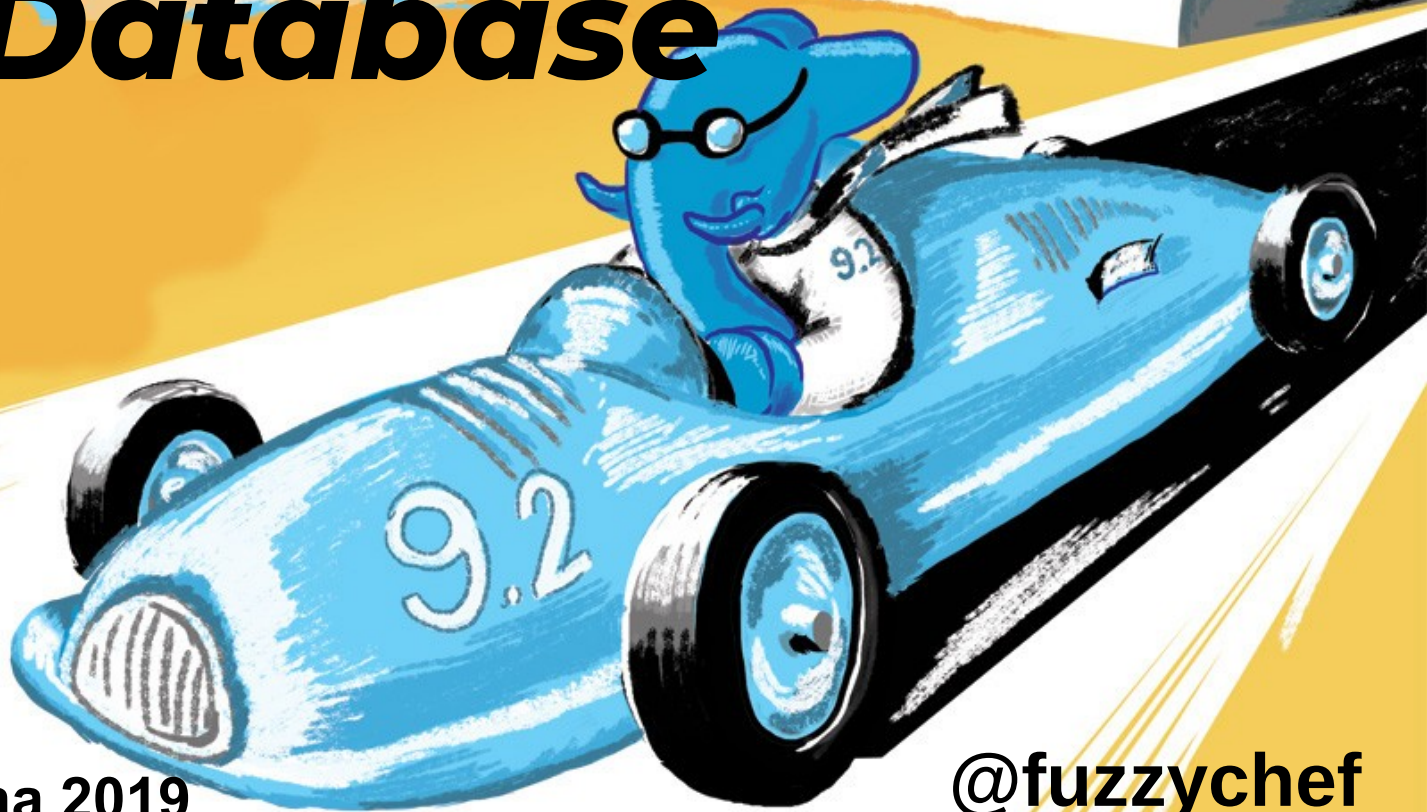


Benchmark Your Cloud Native Database

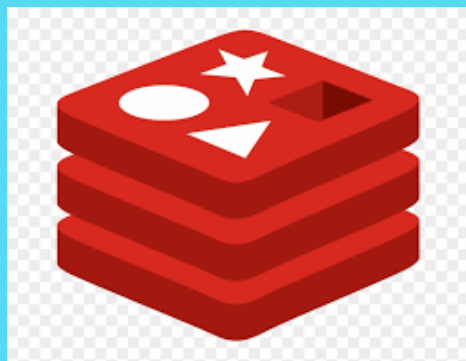
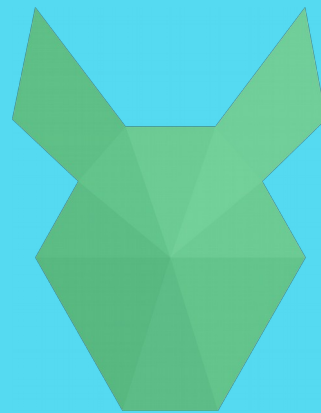
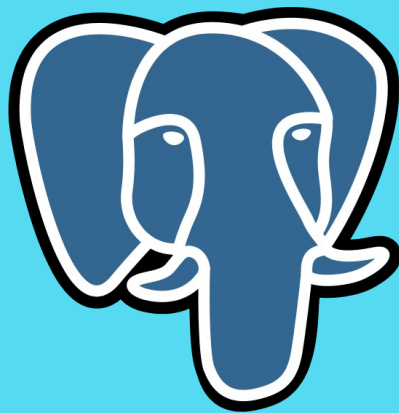


Josh Berkus
Red Hat
KubeCon China 2019

@fuzzychef

Chris Johnson 2012







Kelsey Hightower ✓

@kelseyhightower

Following



Kubernetes has made huge improvements in the ability to run stateful workloads including databases and message queues, but I still prefer not to run them on Kubernetes.

6:04 AM - 13 Feb 2018

306 Retweets **712** Likes



why not?

1. management

Sorry!

2. storage setup complexity



3. performance

why performance?

- DBAs and SAs care
- ease-of-use vs. speed
- migration roadblock



Benchmarking

benchmarking = comparing

- to other types of storage
- to previous releases
- to other configurations
- to spec requirements

types of storage

1. bare metal
2. node local storage
3. network storage
4. cloud-native distributed storage

types of storage

1. bare metal (no K8S)
2. node local storage (hostPath)
3. network storage (EBS)
4. cloud-native distributed storage (Rook/Ceph)

types of storage

1. bare metal (no K8S)
2. node local storage (hostPath)
3. ~~network storage (EBS)~~
4. cloud-native distributed storage (Rook/Ceph)

**Random
Reads**

**Random
Writes**

**Sequential
Reads**

**Sequential
Writes**

**Random
Reads**

&

**Random
Writes**

**Sequential
Reads**

**Sequential
Writes**

Latency

*How long it takes for
each request to
complete*

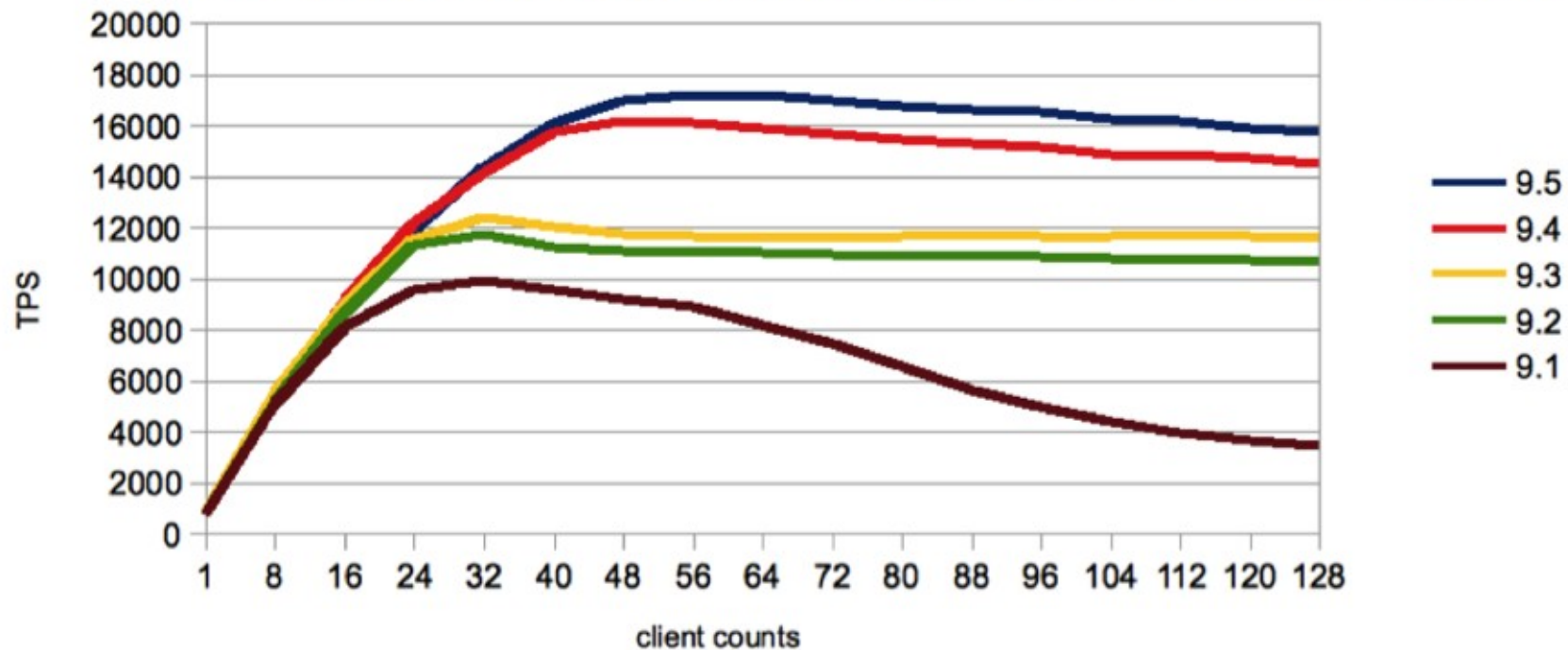
Throughput

*How many requests/
how much data we can
handle in a period*

3 x 3 x 2


pgbench -M prepared

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



DB (micro) benchmarks

- Sysbench
- PostgreSQL pg_bench
- CockroachDB workloads



***No longer
open source!***

sysbench

- created by MySQL team
- many system tests (CPU, mem, DB)
- we use it to check file IO
 - random RW, seq R, seq W

postgres pg_bench

- ships with postgres
- DB micro-benchmark
- measures:
 - random transactional reads/writes
 - load & index times (ETL) (seq)

cdb workloads

- suite of DB micro-benchmarks
- Bank
 - random RW bench, like pg_bench
 - throughput
- TPCC
 - more complex, lock-bound, write-heavy workload
 - latency

microbenchmarking DOs

- many runs
- long runs
- multiple file/DB sizes*
- multiple threads/clients
- use bare metal

why bare metal?

- no noisy neighbors
- larger sizes
- fewer runs
- higher consistency



the numbers

caveats

- **not comparable** btw. tests/databases
- DBs minimally tuned
 - mostly “out of box”
- Your Mileage May Vary
 - my HW & SW is different from yours

6 blade cluster

20 cores ea.

128 GB RAM

2 SSDs w/ 200GB ea.

shared network

6 blade cluster

20 cores ea.

128 GB RAM

2 SSDs w/ 200GB ea.

shared network

***measuring
file sync IO
more than raw writes***

6 blade cluster

20 cores ea.

128 GB RAM

2 SSDs w/ 200GB ea.

shared network

host filesystem

- run tests using a host install, no Kubernetes
- gives reference numbers
- using xfs & lvm

sysbench

**Random
Reads/s**

&

**Random
Writes/s**

**Sequential
Reads**

**Sequential
Writes**

sysbench

10725

rnd r/s

&

7160

rnd w/s

22.6

gb/s read

88.4

mb/s write

pgbench	db load time	txns /sec	avg latency
bank	N/A	txns /sec	95% latency
tpcc	N/A	new orders /sec	95% latency

pgbench	404s	11282 txns	2.8ms
bank	N/A		
tpcc	N/A		

pgbench	404s	11282 txns	2.8ms
bank	N/A	?	?
tpcc	N/A	?	?

local volumes test

- uses hostPath (or local PV) volumes
- basically just local storage via a container

```
storageClassName:  
manual
```

```
persistentVolumeReclaim  
Policy: Recycle  
  capacity:  
    storage: 100Gi  
  accessModes:  
    - ReadWriteOnce  
  hostPath:  
    path: "/localpv/pv/  
pg"
```

sysbench

10720

- 0.01%

&

7157

- 0.01%

22.4

-0.9%

88.1

- 0.4%

pgbench	446s +10.4%	9657 -14.5%	3.3ms +17%
bank	N/A	4717 ops/s	16.8ms
tpcc	N/A	1290 notpm	52.4ms

pgbench	446s +10.4%	9657 -14.5%	3.3ms +17%
bank	N/A	4717 ops/s	16.8ms
tpcc	N/A	1290 notpm	52.4ms

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

rook storage

- 5-node rook+ceph cluster
- only 2 replicas (small cluster)
- some default tweaks for performance
- CockroachDB-on-Ceph, not Rook CDB

sysbench

9363

- 17%

&

6252

- 13%

22.5

+0.2%

111.3

+ 25%

pgbench	611s +28%	4466 -54%	7.1ms +115%
bank	N/A	1546 -57%	37.6ms +123%
tpcc	N/A	1290 +/- 0%	117ms +103%

improving CNDB performance

- better network support (non-shared)
 - try other overlay networks (Weave, Calico)
- multiple SSDs
- distribute workload over CDB better
- Ceph tuning

conclusions

- Benchmark your own hardware with simple DB benchmarks to test your performance
- Local Volume performance is equivalent to bare metal
- Rook/Ceph has good throughput, but about double the latency for random writes

conclusions

- *Beware secondary issues that look like performance differences*
- On public cloud, cloud latency effects mask a lot of performance differences

contact/copyright

- Rook questions? Visit the Rook booth or the Red Hat Booth
- Josh Berkus:
 - jberkus@redhat.com
 - [@fuzzychef](#) on Twitter
 - [@jberkus](#) on Slack