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.
why not?

1. management
2. storage setup complexity
3. performance

Sorry!
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
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

Sequential Reads

&

Random Writes/s

Sequential Writes
**sysbench**

10725 & 7160

rnd r/s & rnd w/s

22.6 & 88.4

gb/s read & mb/s write
<table>
<thead>
<tr>
<th></th>
<th>pgbench</th>
<th>db load time</th>
<th>txns/sec</th>
<th>avg latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>bank</td>
<td></td>
<td>N/A</td>
<td></td>
<td>95% latency</td>
</tr>
<tr>
<td>tpcc</td>
<td></td>
<td>N/A</td>
<td>new orders/sec</td>
<td>95% latency</td>
</tr>
<tr>
<td>pgbench</td>
<td>404s</td>
<td>11282 txns</td>
<td>2.8ms</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>bank</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tpcc</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>pgbench</td>
<td>bank</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>------</td>
<td>-------</td>
<td>-----</td>
</tr>
<tr>
<td>404s</td>
<td>11282</td>
<td>2.8ms</td>
<td>N/A</td>
<td>?</td>
</tr>
<tr>
<td>11282 txns</td>
<td>N/A</td>
<td>?</td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>2.8ms</td>
<td>?</td>
<td>?</td>
<td></td>
<td>?</td>
</tr>
</tbody>
</table>
local volumes test

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

storageClassName: manual

persistentVolumeReclaimPolicy: Recycle
capacity:
  storage: 100Gi
accessModes:
  - ReadWriteOnce
hostPath:
  path: "/localpv/pv/pg"
sysbench

10720 & 7157
- 0.01%  - 0.01%

22.4 & 88.1
-0.9%  - 0.4%
<table>
<thead>
<tr>
<th></th>
<th><strong>pgbench</strong></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>446s</strong></td>
<td><strong>9657</strong></td>
<td><strong>3.3ms</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>+10.4%</strong></td>
<td><strong>-14.5%</strong></td>
<td><strong>+17%</strong></td>
</tr>
<tr>
<td><strong>bank</strong></td>
<td>N/A</td>
<td></td>
<td><strong>4717 ops/s</strong></td>
<td><strong>16.8ms</strong></td>
</tr>
<tr>
<td><strong>tpcc</strong></td>
<td>N/A</td>
<td></td>
<td><strong>1290 notpm</strong></td>
<td><strong>52.4ms</strong></td>
</tr>
<tr>
<td>Benchmark</td>
<td>Time</td>
<td>Operations</td>
<td>Latency</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
<td>------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>pgbench</td>
<td>446s</td>
<td>9657</td>
<td>3.3ms</td>
<td></td>
</tr>
<tr>
<td></td>
<td>+10.4%</td>
<td>-14.5%</td>
<td>+17%</td>
<td></td>
</tr>
<tr>
<td>bank</td>
<td>N/A</td>
<td>4717</td>
<td>16.8ms</td>
<td></td>
</tr>
<tr>
<td>tpcc</td>
<td>N/A</td>
<td>1290</td>
<td>52.4ms</td>
<td></td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>Test</th>
<th>Time</th>
<th>Diff</th>
<th>Time</th>
<th>Diff</th>
</tr>
</thead>
<tbody>
<tr>
<td>pgbench</td>
<td>611s</td>
<td>+28%</td>
<td>4466s</td>
<td>-54%</td>
</tr>
<tr>
<td>bank</td>
<td>N/A</td>
<td></td>
<td>1546s</td>
<td>-57%</td>
</tr>
<tr>
<td>tpcc</td>
<td>N/A</td>
<td></td>
<td>1290s</td>
<td>+/- 0%</td>
</tr>
</tbody>
</table>
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

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.