ZFS and MySQL on Linux, the Sweet Spots

ZFS User Conference 2018

Jervin Real

PERCONA

MySQL

The World's Most Popular Open Source Database

ZFS

Is MySQL for storage.

ZFS + MySQL

MySQL Needs

- A reliable, durable, performant storage

ZFS Provides

- A reliable, durable, performant(?) storage

ZFS + MySQL

MySQL Needs

- A reliable, durable, performant storage
- At the same time, users demand:
  - At rest encryption (number of choices available)
  - Compression (InnoDB compression is a bit complex for many)
  - Reliable and sane backups (except when you leave your dataset to grow)

ZFS Provides

- A reliable, durable, performant(?) storage

ZFS + MySQL

MySQL Needs

- A reliable, durable, performant storage
- At the same time, users demand:
  - At rest encryption (number of choices available)
  - Compression (InnoDB compression is a bit complex for many)
  - Reliable and sane backups (except when you leave your dataset to grow)

ZFS Provides

- A reliable, durable, performant(?) storage
- Compression options, encryption, sane backups

ZFS + MySQL

Its a compromise and they should meet somewhere in between ...

WARNING: You will see graphs ...

Use Case: Large MySQL Dataset

Large MySQL Dataset

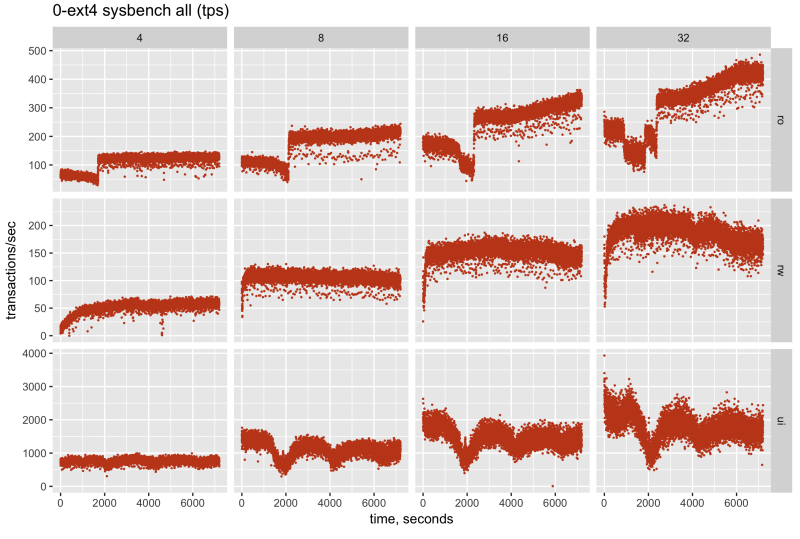
Probably not what you should be doing, but:

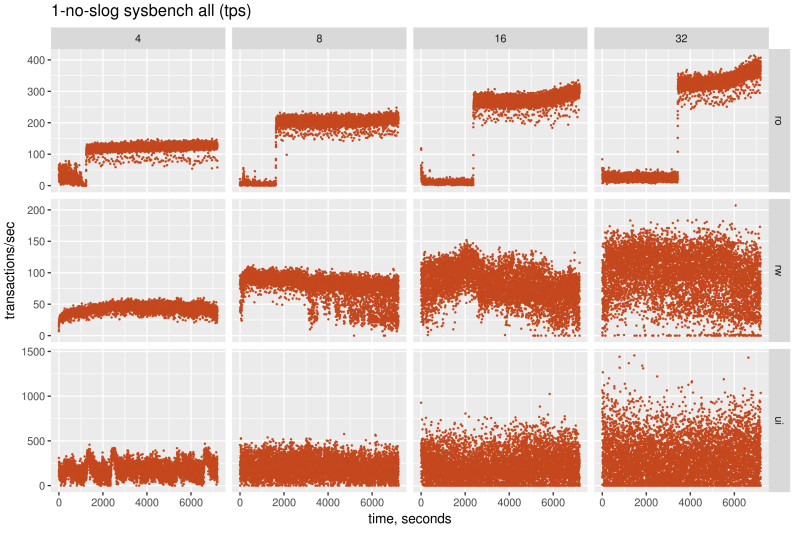
- Single large dataset, in the TBs range
  - Backup and recovery challenges
  - Storage space constraints
  - Long data retention periods (fintech/healthcare)
- Mixed storage engines (yes MySQL allows it)

Large MySQL Dataset

Switching is not as straighforward however:

- Thick performance bands on ZFS
- TPS down to 0 on write heavy tests



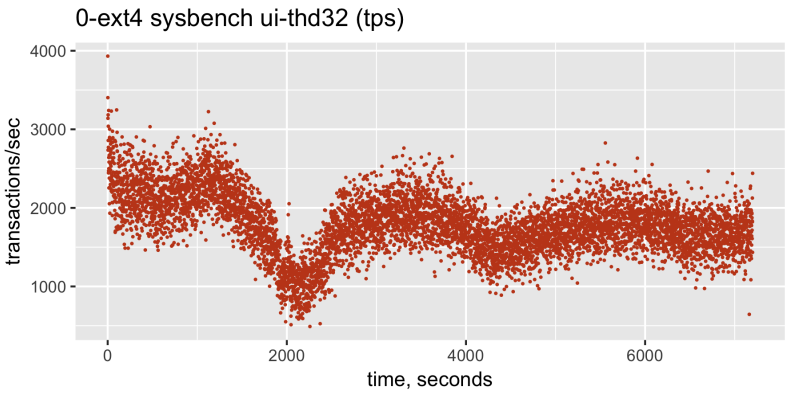


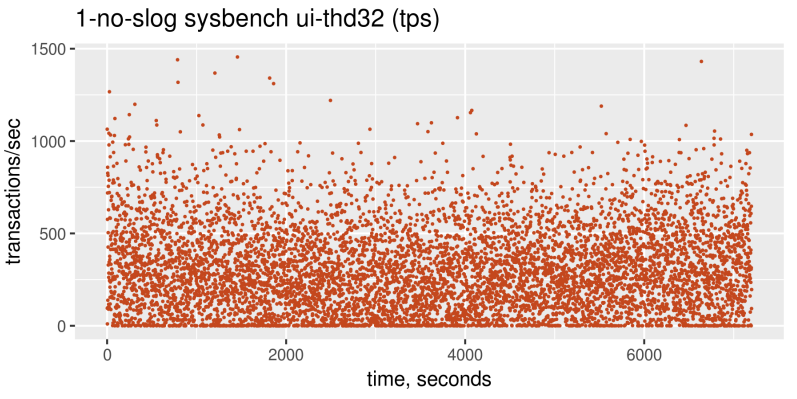
PERCONA

Large MySQL Dataset

Switching is not as straighforward however:

- Gets worse on 32 threads, write-only workload

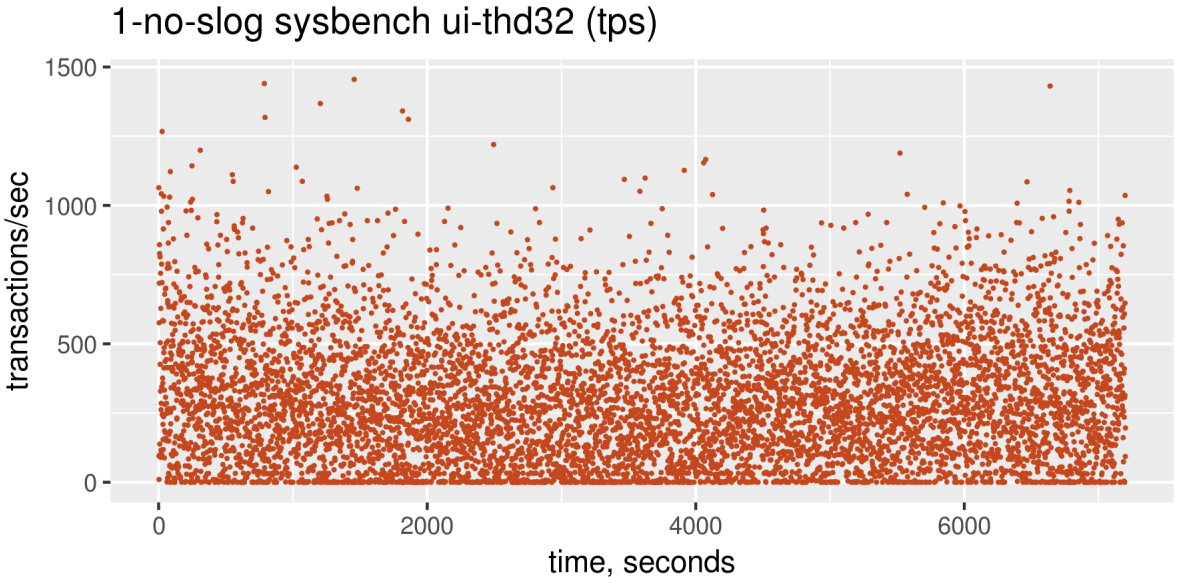




PERCONA

Large MySQL Dataset

Switching is not as straighforward however:



- Transactions/sec drops to zero a lot of times!

Large MySQL Dataset

Switching is not as straighforward however:

- HP DL380G7 6xSAS disks, P410 (with battery backed cache)
- Disks configured as JBOD, ZFS pool 2x3 mirrored stripes

ZFS

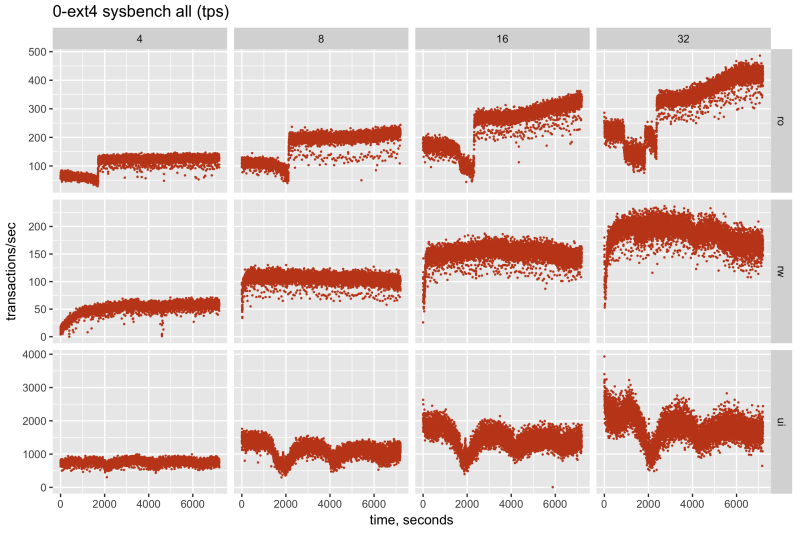
```
logbias=throughput
zfs_arc_max=1073741824
zfs_prefetch_disable=1
atime=12
recordsize=16k/128k
```

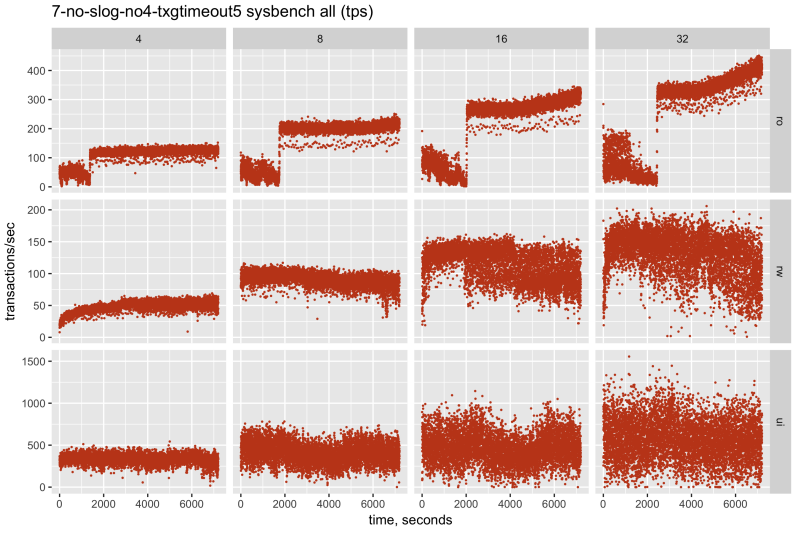
MySQL

```
sync_binlog=1
innodb_flush_log_at_trx_commit=1
innodb_doublewrite=0
```

Large MySQL Dataset

Trying to optimize without shelling out extra cash:





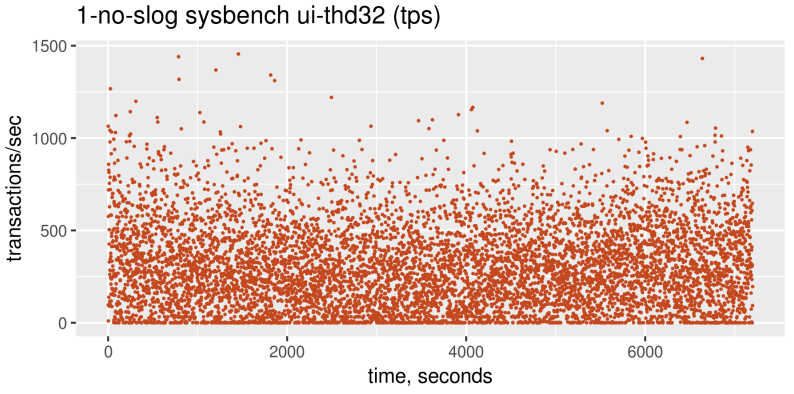
Minimum TPS drops just by tuning `zfs_dirty_data_max=128M`:

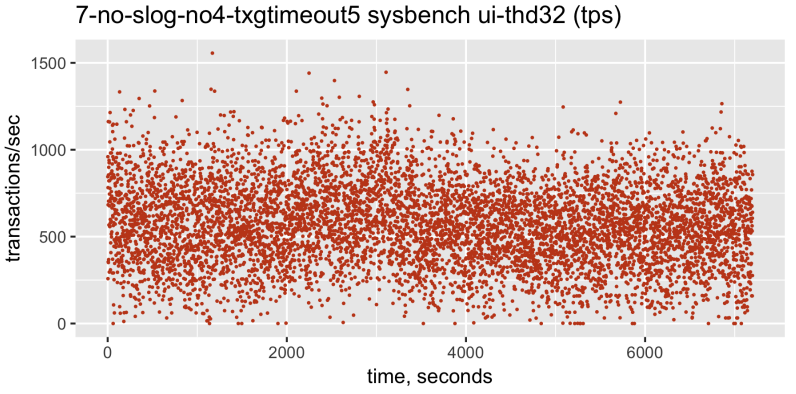
```
./1-no-slog/sb-no-slog-ui32.csv                                  383
./7-no-slog-no4-txgtimeout5/sb-no-slog-no4-txgtimeout5-ui32.csv   18
```

PERCONA

Large MySQL Dataset

Trying to optimize without shelling out extra cash:

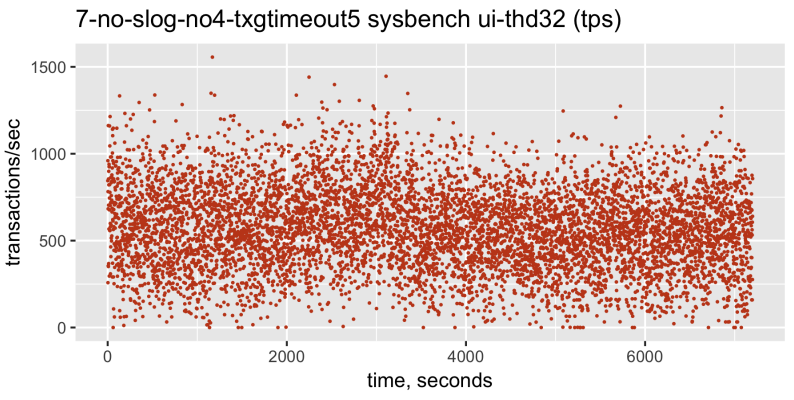


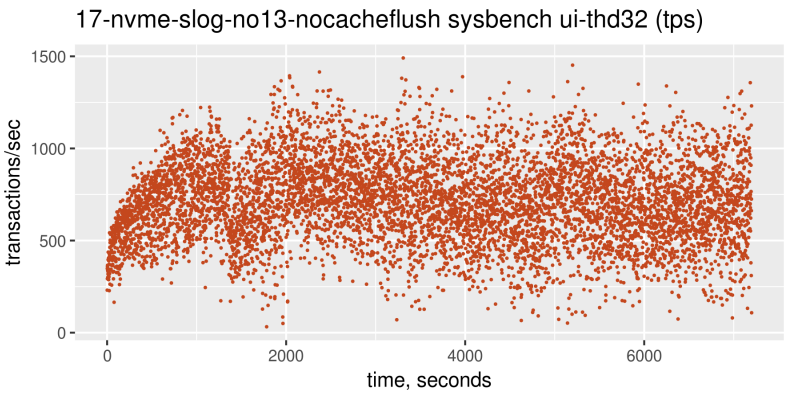


- Still quite far off from EXT4 numbers, but we know our ceiling

Large MySQL Dataset

Biting the bullet, adding an NVMe SLOG:

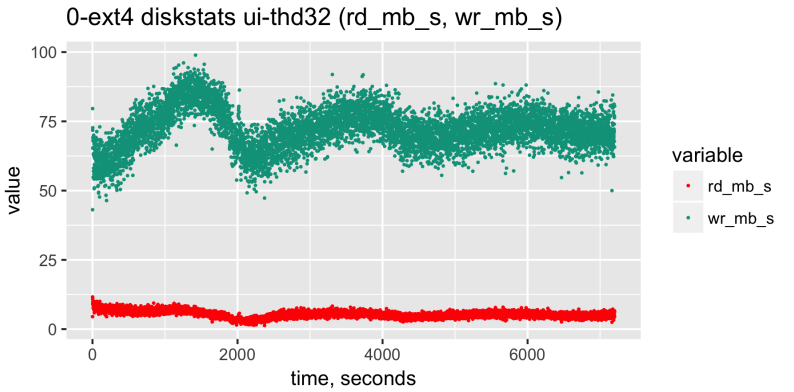


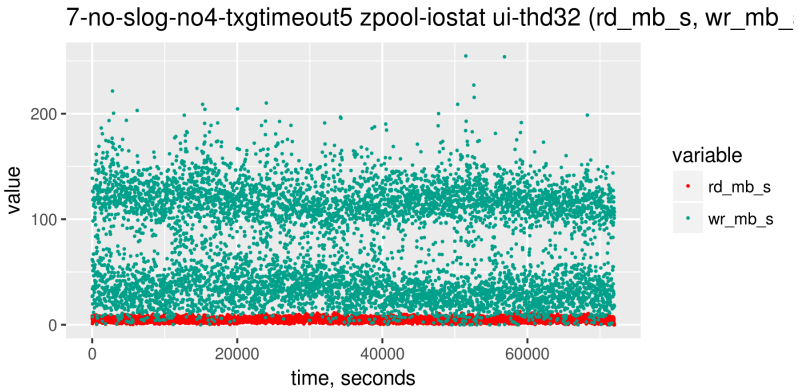


- Raised further, but not as far as we expected
- `zfs_nocacheflush=1` helps too, we are using RAID controller with battery backed cache

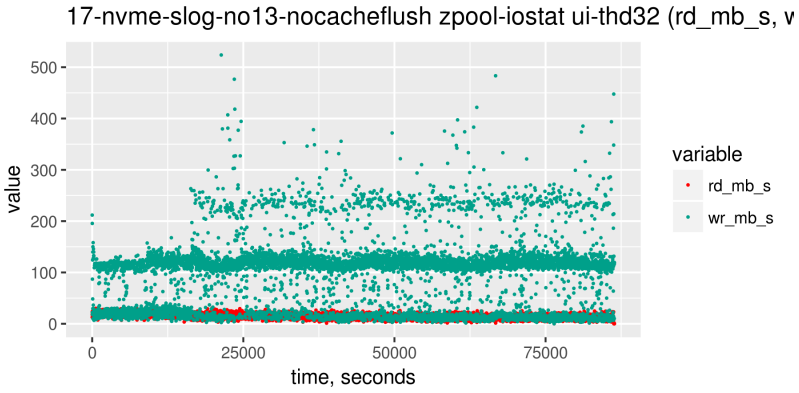
Large MySQL Dataset

Biting the bullet, adding an NVMe SLOG:





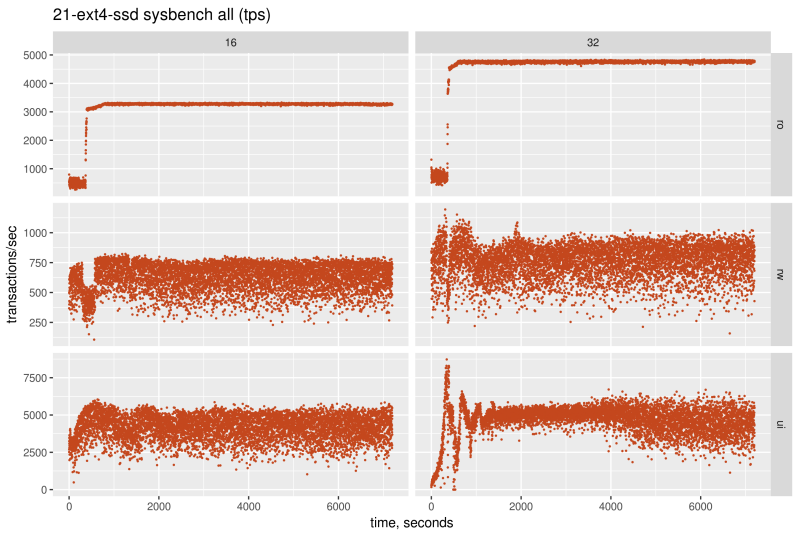
- Limited by how much throughput and IOPs combined our main pool disks can deliver.

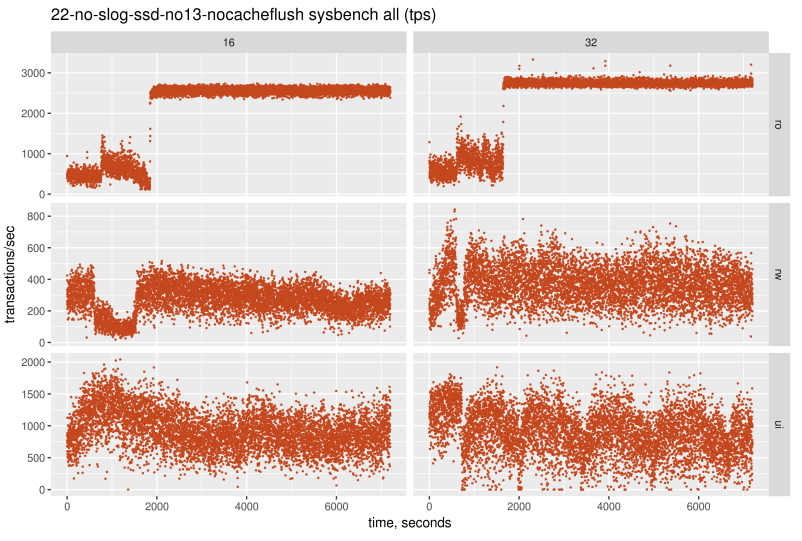


PERCONA

Large MySQL Dataset

Well let's test with SSD as well:





- Same hardware, 6x Samsung 860 SSD drives
- Same ZFS config without NVMe SLOG
- Big gap from EXT4, just a matter of tuning for capacity with SSD

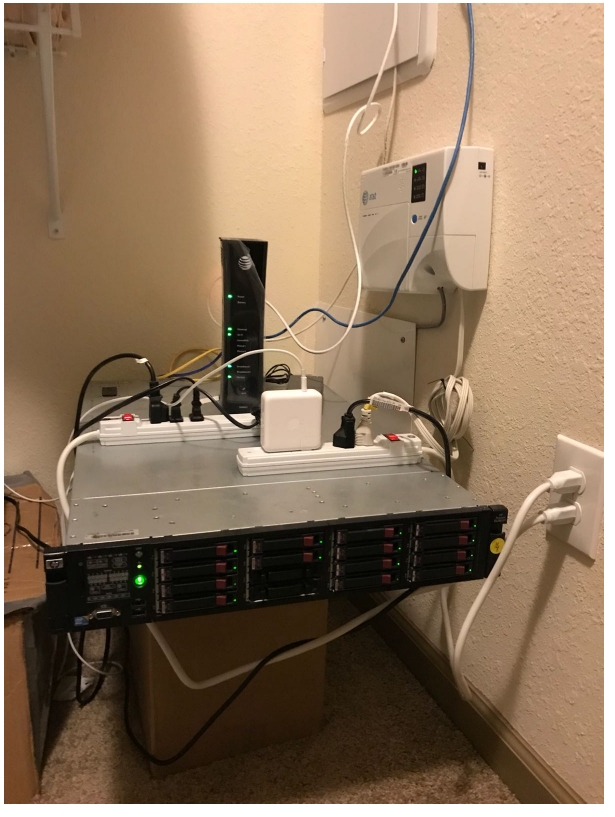
PERCONA

Large MySQL Dataset

Unfortunately, just as we were having fun:

¯_(ツ)_/¯

Watch this space:
https://github.com/dotmanila/zfs-mysql



Large MySQL Dataset

What we learned so far:

- Know your pool capability and capacity
- Performance depends on the slowest component (pool disks vs slog)

Use Case: Percona XtraDB Cluster

Percona XtraDB Cluster

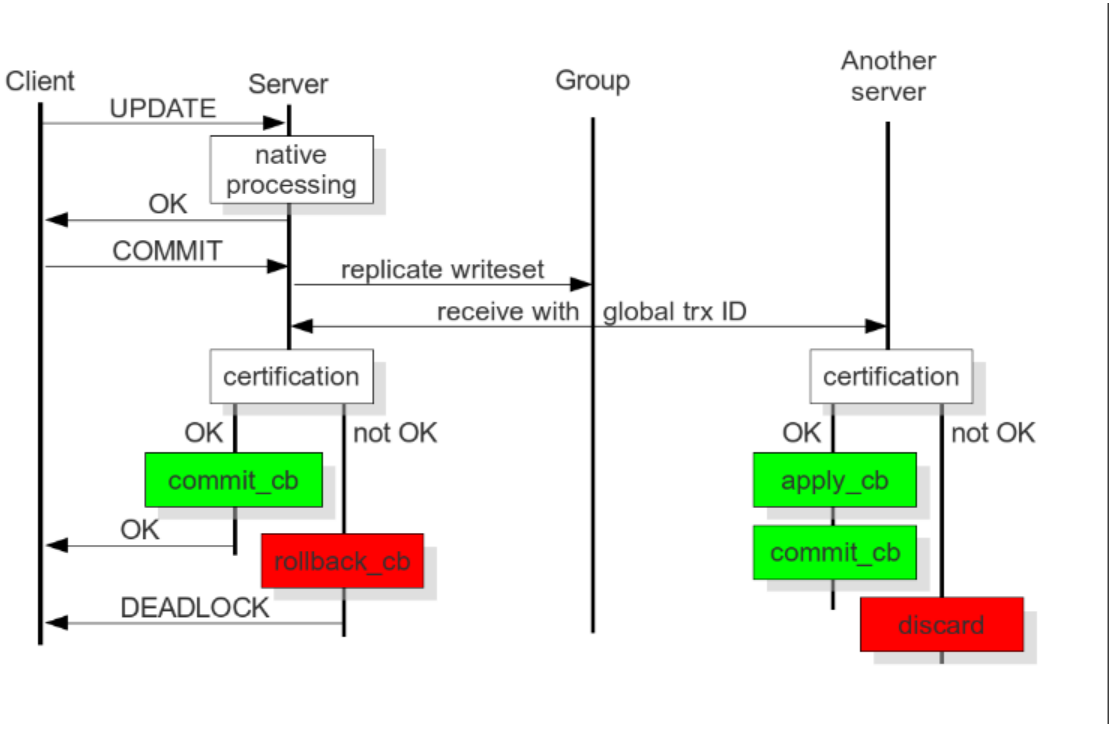
A group of MySQL servers with:

- Synchronous Replication
- Multi Master*, True Parallel Replication
- Automatic node provisioning

- *Requires compatible workload type

PERCONA

How Galera replication works:



- Image Credit: http://galeracluster.com/documentation-webpages/certificationbasedreplication.html#how-certification-based-replication-works

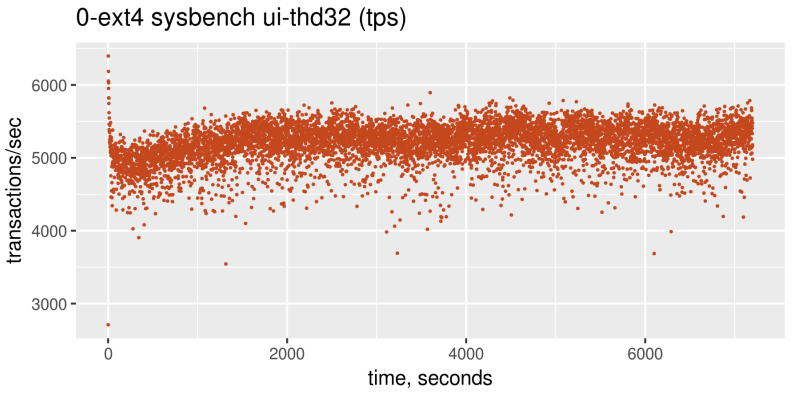
Why ZFS fits:

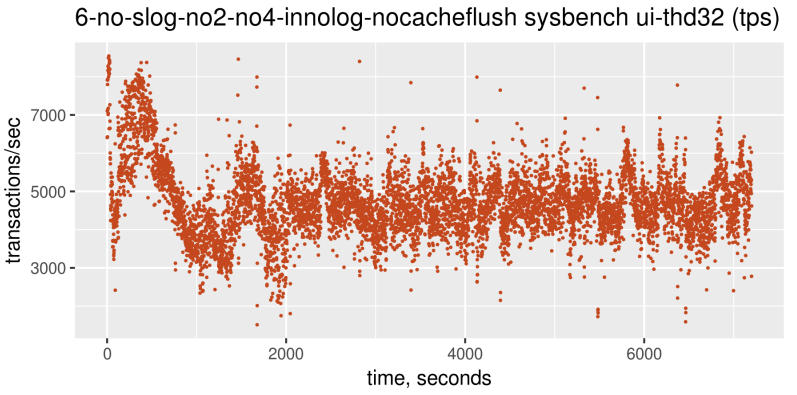
- Writeset certifications within the cluster allows least per node durability.
  - `sync_binlog = 0`
  - `innodb_flush_log_at_trx_commit = 0`

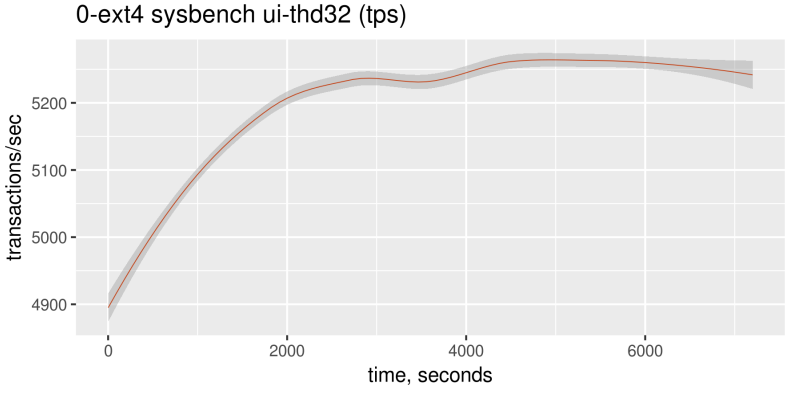
Why ZFS fits:

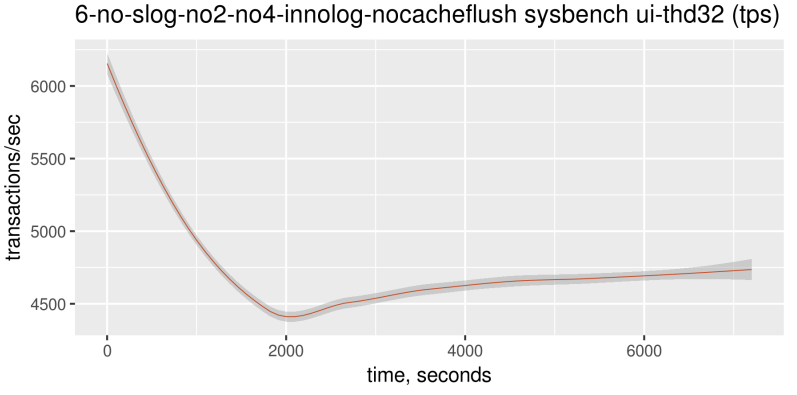
- We should be able to tune ZFS with least durability as well.
  - `sync=disabled`
  - `zfs_nocacheflush=1`
- Plus PXC settings:
  - `innodb_doublewrite=0`
  - `innodb_log_checksums=OFF`
  - `innodb_checksum_algorithm=none`

Exploring ZFS on PXC, i3.2xlarge









Exploring ZFS on PXC, i3.2xlarge

```
[mysqld]
server-id=1
datadir=/mysql/data
wsrep_provider=/usr/lib/galera3/libgalera_smm.so
wsrep_cluster_address=gcomm://
binlog_format=ROW
default_storage_engine=InnoDB
innodb_doublewrite=0
innodb_log_group_home_dir=/mysql/logs
innodb_io_capacity=5000
innodb_autoinc_lock_mode=2
innodb_flush_log_at_trx_commit=0
innodb_buffer_pool_size=48G
innodb_log_file_size=8G
innodb_log_checksums=OFF
innodb_checksum_algorithm=none
pxc_strict_mode=ENFORCING
wsrep_node_name=zfs01
wsrep_slave_threads=8
wsrep_node_address=10.1.2.117
wsrep_cluster_name=zfs
wsrep_sst_method=xtrabackup-v2
wsrep_sst_auth="msandbox:msandbox"
```

```
options zfs zfs_arc_max=1073741824
options zfs zfs_prefetch_disable=1
options zfs zfs_nocacheflush=1
```

```
sudo zpool create -o ashift=12 -f mysql /dev/nvme0n1
sudo zfs set recordsize=16k mysql
sudo zfs set atime=off mysql
sudo zfs set logbias=latency mysql
sudo zfs set primarycache=metadata mysql
sudo zfs set compression=lz4 mysql
sudo zfs set sync=disabled mysql

sudo zfs create -o recordsize=128K mysql/logs
sudo zfs create -o recordsize=16K mysql/data
```

Why ZFS fits:

- And still take advantage of ZFS features:
  - ZFS snapshots for SST
    - Reading records/blocks from source multithreaded
    - Writing records to destination is also multithreaded
  - Encryption
  - Compression

State Snapshot Transfer:

- XtraBackup
  - Donor becomes unavailable, on a 3 node cluster, you lose 2/3 when one needs SST.
  - Potentially slow for large uncompressed datasets
    - On the fly compression and decompression
    - It is however parallel
    - Depends on compressability
- ZFS Snapshot
  - Donor remains available
  - Raw SEND/REC

State Snapshot Transfer:

- Taking ZFS Snapshot with Percona Server/Percona XtraDB Cluster

```
mysql> LOCK TABLES FOR BACKUP;
mysql> LOCK BINLOG;
mysql> SHOW MASTER STATUS;
mysql> -- take ZFS snapshot here
mysql> UNLOCK TABLES;
mysql> UNLOCK BINLOG;
```

State Snapshot Transfer:

- Streaming encrypted and compressed:

```
ubuntu@ip-10-1-2-117:~$ time sudo zfs send -wcR mysql@201804192210 | mbuffer -s 128
in @ 22.2 MiB/s, out @ 76.4 MiB/s,  156 GiB total, buffer   0% full
summary:  156 GiByte in 23min 22.1sec - average of  114 MiB/s

real  23m24.961s
user  0m9.052s
sys 6m39.036s
```

```
ubuntu@ip-10-1-2-126:~$ mbuffer -s 128k -m 1G -I 9999 | sudo zfs recv -F mysql
in @ 77.9 MiB/s, out @ 77.9 MiB/s,  156 GiB total, buffer   0% full
summary:  156 GiByte in 23min 24.7sec - average of  114 MiB/s

ubuntu@ip-10-1-2-126:~$ sudo zfs load-key -a
2 / 2 key(s) successfully loaded

ubuntu@ip-10-1-2-126:~$ sudo zfs mount -a
```

State Snapshot Transfer: ZFS POC Tests

- r4.4xlarge, single EBS 4000 PIOPs
- PXC 5.7.19
- Idle cluster, no traffic during snapshot transfers

Config	Event	Time
ext4	prepare	21m31.831s
	sst	3m5s
	sst (threads=8)	3m5s
zfs (xenial)	prepare	40m53.066s
	sst	9m37s (same for 8thd)
	send/recv	10min 48.9sec
	send -c/recv	NA
zfs (git)	prepare	33m56.747s
	sst	4m35s
	sst (threads=8)	3m22s
	send/recv	4min 03.6sec
	send -c/recv	4min 02.3sec
zfs (git, gzip)	prepapre	45m23.769s
	sst	3m40s
	send/recv	6min 39.0sec
	send -c/recv	2min 55.3sec

Use Case: MySQL Dedicated Backup Nodes

MySQL Dedicated Backup Nodes

Backups are fun:

- Performance
- Restore Time Objective
- Restore Point Objective
- Security

MySQL Dedicated Backup Nodes

Backup options for InnoDB:

- Percona XtraBackup
- Delayed Replica

Backups are fun: XtraBackup (1/2)

- Performance
  - Fast parallel copy, fast on the fly compression with qpress
  - Decompress in parallel too
- Restore Time Objective
  - Depends on where backups are stored relative to restore target
  - How fast to transfer backups from stored state to usable state
  - PITR also adds up and depends on how frequent backups are made
  - Full + incremental backups are possible

Backups are fun: XtraBackup (2/2)

- Restore Point Objective
  - PITR with full + incremental + binary logs roll forward
  - Gap between incremental and binary log target is crucial
- Security
  - Encryption on the fly, decryption separate process
  - Decompression and decryption can be done at the same time in parallel with some shell-fu
  - `cat table.ibd.xbcrypt.qp | xbcrypt -d | qpress -di table.ibd`

Backups are fun: Delayed MySQL Replica (1/2)

- Performance
  - Full dataset immediately available
  - PITR recovery speed depends on configured delay and SQL thread speed
- Restore Time Objective
  - Depends on configured delay and SQL thread speed

Backups are fun: Delayed MySQL Replica (2/2)

- Restore Point Objective
  - PITR range depends on configured delay (1hr vs 1day)
- Security
  - Scoped within instance security (at rest, in transit)

Testing Snapshots

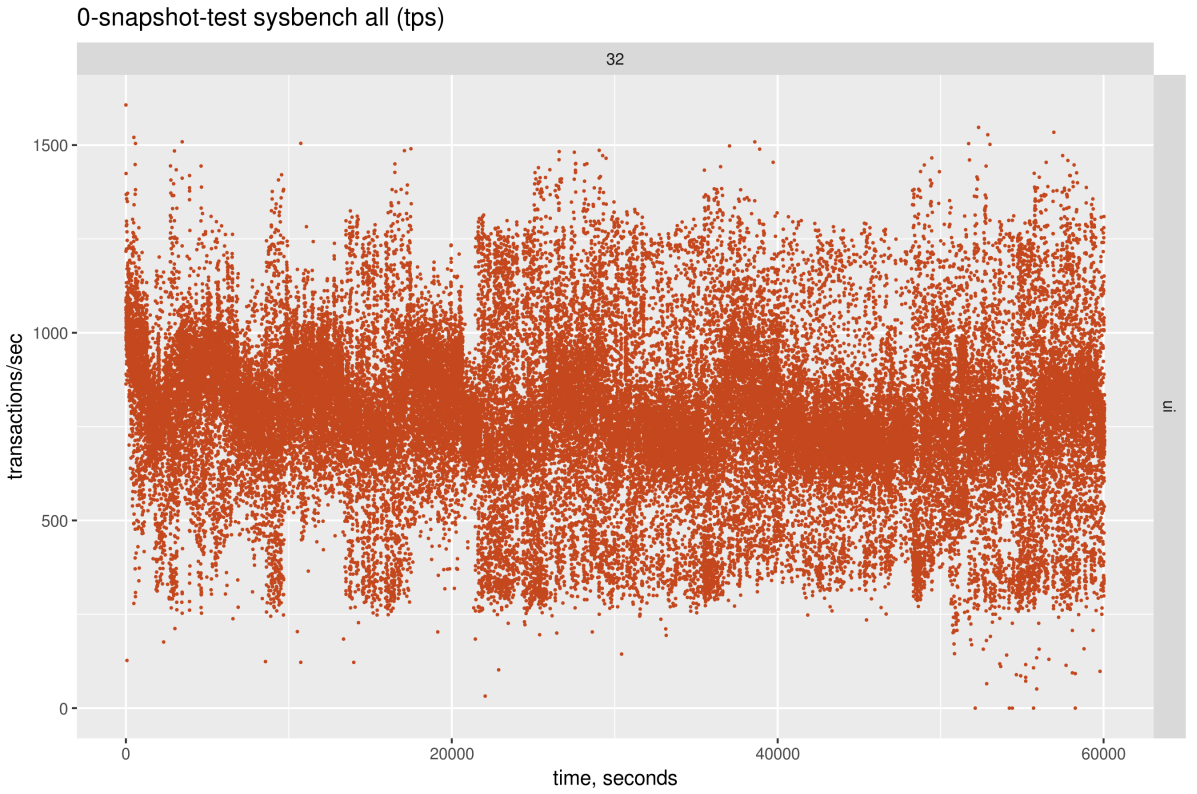
- Run sysbench oltp_update_index test, 32 threads on an i3.2xlarge
- Take snapshots every 5mins, copy the last snapshot every hour (send to `/dev/null`)

```
while true; do
  echo "$(date +%Y-%m-%d_%H:%M) sleeping ...";
  sleep 300;
  sudo zfs snapshot -r mysql@$(date +%Y%m%d%H%M);
done
```

```
while true; do
  snap=$(zfs list -t snap | egrep 'mysql@' | tail -n1 | awk '{print $1}');
  time sudo zfs send -R $snap | cat - > /dev/null;
  sleep 3600;
done
```

Testing Snapshots

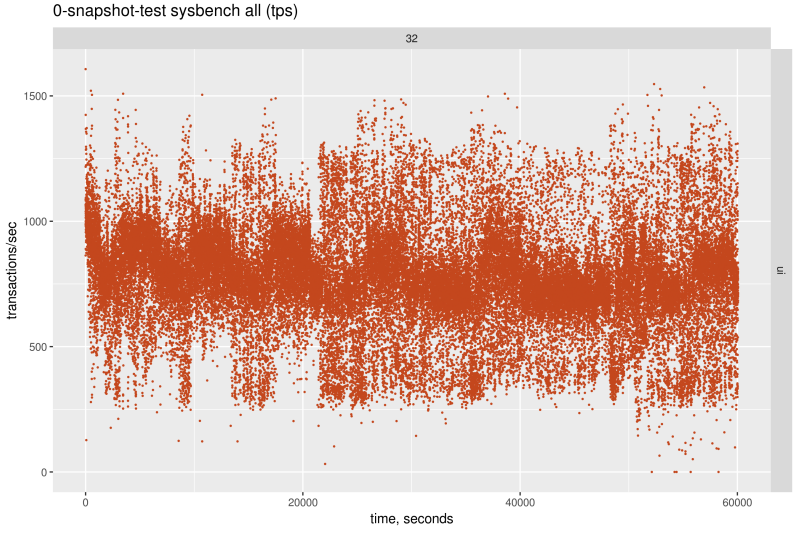
- Transactions/sec looks to be even and non-degrading

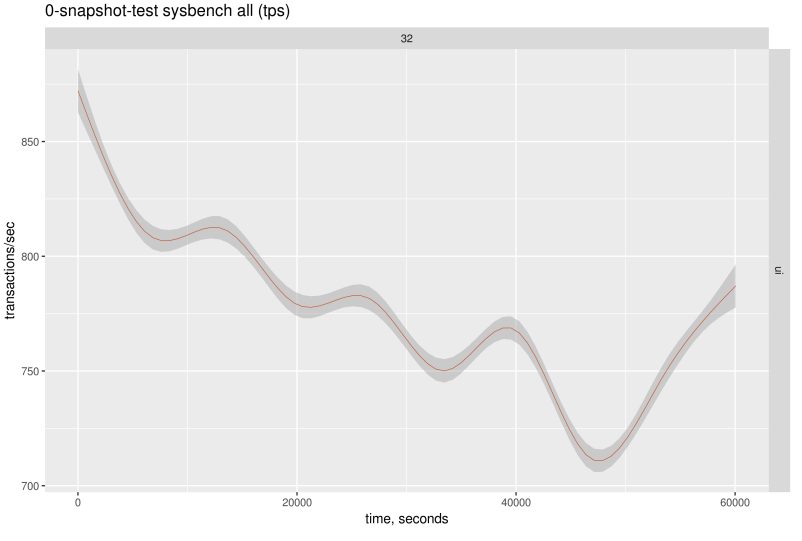


PERCONA

Testing Snapshots

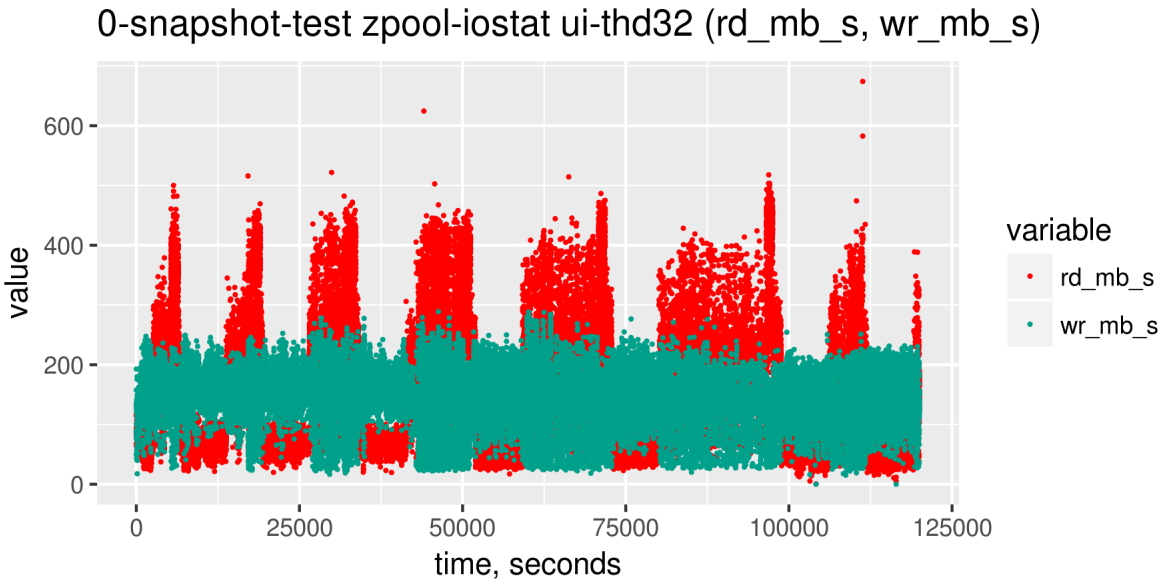
- Running the graph on Loess smoothing, reveals something interesing
- Transactions/sec degrades overtime





Testing Snapshots

- Reads saturating the disk
  - Binary logs size increased



Backups are fun: ZFS Snapshots (1/2)

- Performance
  - Compressed optimized SEND/RECV
- Restore Time Objective
  - Same as Xtrabackup (full + incremental + binlogs speed)
  - In place snapshots allows near instantaneous rollback!

Backups are fun: ZFS Snapshots (2/2)

- Restore Point Objective
  - In place snapshots allows rollback as far as space allows
  - Combined with replication, essentially also a delayed replica but faster!
- Security
  - Encrypted datasets can remain encrypted in transit and to destination

Looking Forward

Looking Forward

Exciting things up ahead:

- ZSTD compression!
- More MySQL related potential tuning:
  - Separate undo log directory - InnoDB undo log are best suited for SSDs
- PXC state snapshot transfer (automatic provisioning) with ZFS snapshots

Questions!

Percona Live Call 2018 is Next Week!

https://www.percona.com/live/18/