Righting Your Writes

Greg Smith

2ndQuadrant US

11/04/2010

2ndQuadrant
Professional PostgreSQL

About this presentation

- The master source for these slides is `http://projects.2ndquadrant.com`

The buffer cache

- shared_buffers sets size
- 256MB - 8GB is typical
- Traditional tuning suggests around 25% of total RAM

Checkpoints

- All dirty data in buffer cache must be flushed to disk eventually
- WAL segments are 16MB
- Checkpoint requested after every checkpoint_segments worth of writes
- Timed checkpoint every checkpoint_timeout (5 minute default)
- Traditional tuning sets checkpoint_segments 32-128

Checkpoint spikes

- Before 8.3, all dirty data written in one burst
- 8.3 added Spread Checkpoints
- Defaults aim to finish 50% of the way through next checkpoint
- Fsync flush to disk happens at end of checkpoint
- Optimal behavior: OS already wrote data out before fsync call
- Attempts to spread the sync out didn't work usefully
- Spikes still happen

Linux filesystem trivia

- Checkpoint rewrite tests all on Linux
- Default and only stable Linux filesystem then was ext3
- ext3 handles fsync by writing all cached data to disk
- Spread sync can't help if every fsync writes all data out
- WAL writes do fsync too
- One reason why separating WAL and database disks helps so much
- XFS and ext4 allow granular sync

Linux write caching

- dirty_ratio and dirty_backround_ratio control % of RAM to allow dirty
- More aggressive writing happens when thresholds crossed
- Writes can become blocked
- Ideally, dirty RAM fits in battery backed cache size
- Kernel before 2.6.22: 10%/40% of RAM are thresholds
- Kernel 2.6.22 and later: 5%/10% are defaults

Having a bad day on purpose with ext3

- log_checkpoints shows sync time
- 8GB of RAM in server
- 5% dirty=400MB
- 10% dirty=800MB
- 256MB of battery-backed cache
- Standard pgbench test dirties data very fast

2ndQuadrant
Professional PostgreSQL

Checkpoint Goggles, ext3

```
Sync #1 time=0.001000 msec
Sync #2 time=0.001000 msec
Sync #3 time=63331.980000 msec
Sync #4 time=0.003000 msec
Sync #5 time=0.001000 msec
...
Sync #30 time=4.388000 msec
Sync #31 time=6.798000 msec
Sync 31 files longest=63331.98 msec average=2046.65 msec
checkpoint complete: wrote 67378 buffers (25.7%);
write=218.740s, sync=63.447s, total=282.391s
```

2ndQuadrant
Professional PostgreSQL

Checkpoint Goggles, XFS

```
Sync #1 time=0.004000
Sync #2 time=171.304000 gap=0.000000 msec
Sync #3 time=1002.743000 gap=0.041000 msec
Sync #4 time=0.004000 gap=0.037000 msec
...
Sync #12 time=14071.240000 gap=0.036000 msec
Sync #13 time=0.003000 gap=0.052000 msec
Sync 13 files longest=14071.24 msec average=829.14 msec
```

2ndQuadrant
Professional PostgreSQL

A really bad day on a popular web site

- XFS
- Lots of RAM
- shared_buffers=512MB, typically under 200MB dirty at checkpoint time
- Often gigabytes of write cache dirty with random writes
- Still well under 10%, Linux is unfortunately not too concerned
- sync= time 50 minutes?!
- Not even 1MB/second into a medium sized disk array

Writes in PostgreSQL

- Checkpoint write: most efficient
- Background writer write: still good
- Backend write, fsync aborbed by background writer: fine if OS caches
- Backend write, BGW queue filled, backend does fsync itself: bad

Backend sync counts

```
$ psql -x -c "select * from pg_stat_bgwriter"
checkpoints_timed | 0
checkpoints_req | 4
buffers_checkpoint | 6
buffers_clean | 0
maxwritten_clean | 0
buffers_backend | 654685
buffers_backend_sync | 84
buffers_alloc | 1225
```

2ndQuadrant
Professional PostgreSQL

The root problem

- Background writer stop working normally while running sync
- Never pauses to fully consume the fsync queues backends fill
- Once filled, all backend writes do their own fsync
- Serious competition for the checkpoint writes

Solutions

- Introduce a pause to spread out writes after each file sync
- During the pause time, continue running regular background writer work
- Improve general fsync queue management

2ndQuadrant
Professional PostgreSQL

Production results

- On high traffic site for over two months
- Checkpoints timing back to normal again
- Serious site outages at checkpoint time eliminated

Patch status

- Planned for submission to PostgreSQL 9.1
- Checkpoint Goggles logging level needs fine tuning
- Value of exposing buffers_backend_sync debatable
- Sync spread not as fully auto-tuning as write spread yet
- Haven't evaluated against ext4 yet
- No testing outside of Linux

Planning impact

- Be careful using large settings for shared_buffers with heavy writes
- Monitor size of OS cache dirty data to measure problems here
- grep "Dirty:" /proc/meminfo
- ext3 increasingly bad as total system memory continues to increase
- Revival of XFS popularity for over 16TB filesystems makes it more viable now