Tunable Durability

Andres Freund
PostgreSQL Developer & Committer
Citus Data – citusdata.com - @citusdata

http://anarazel.de/talks/pgconf-sv-2016-11-16/durability.pdf

citusdata

Atomicity

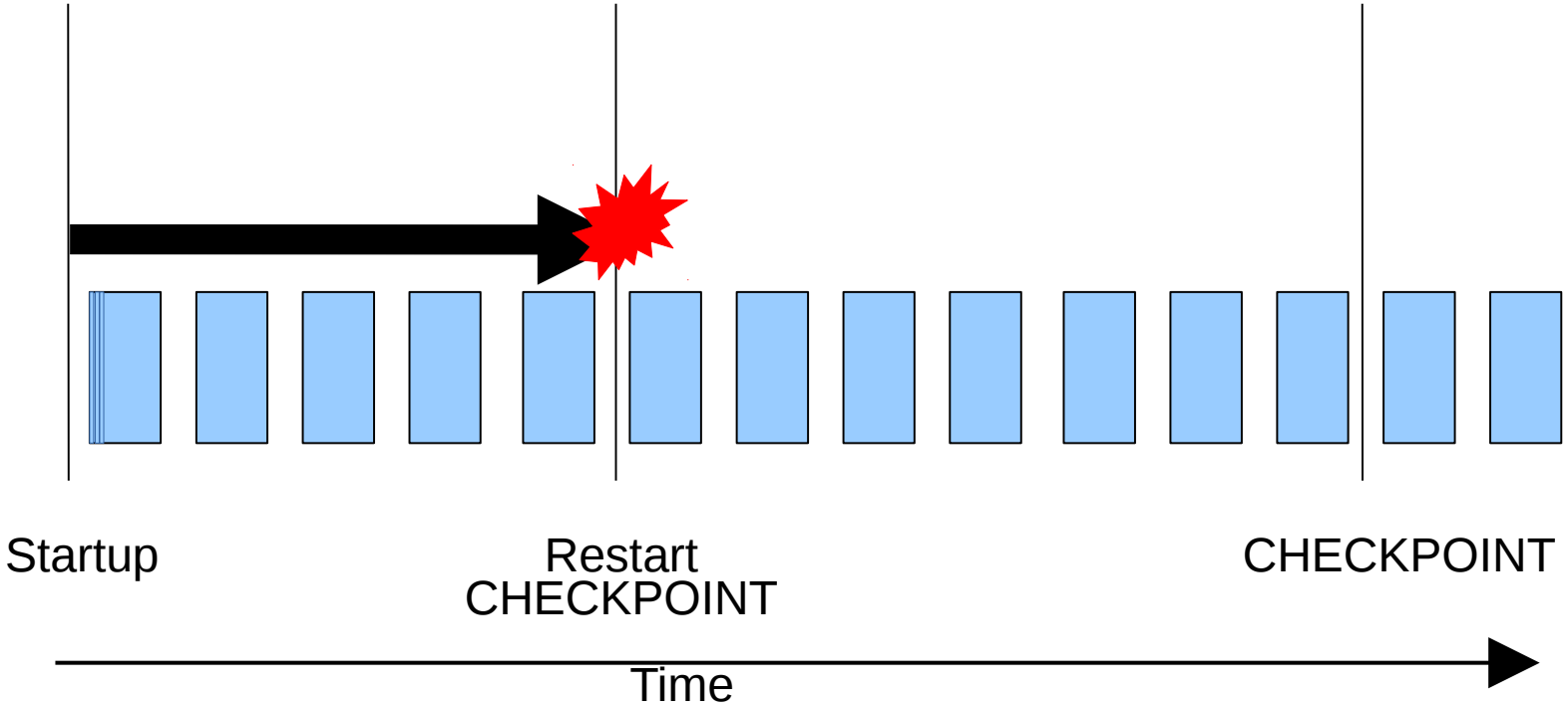
Consistency

Isolation

Durability

The durability property ensures that once a transaction has been committed, it will remain so, even in the event of power loss, crashes, or errors.

Recovery & Checkpoints



Triggering Checkpoints

- checkpoint_timeout = 5min
  - LOG: checkpoint starting: time
- checkpoint_segments = 3 / max_wal_size = 1GB
  - LOG: checkpoint starting: xlog
  - LOG: checkpoints are occurring too frequently (2 seconds apart)
- shutdown
  - LOG: checkpoint starting: shutdown immediate
- manually (CHECKPOINT;)
  - LOG: checkpoint starting: immediate force wait

WAL Tuning

- log_checkpoints = on
- Max checkpoint #writes: shared buffers
- Max checkpoints writes/sec w/ spreading:
  shared_buffers / (timeout * target)
- IO rate vs. recovery time
- Checkpoint Spreading (checkpoint_completion_target)
- Full Page Writes (pg_xlogdump --stats)
- Compression (9.5+) (wal_compression)

Architecting Durability

Costs of Journaling (Commits)

- Commit => synchronous write
- Synchronous write => roundtrip latency
- Latencies:
  - Rotational disks: high
  - Remote Disks (EBS!): low-medium + storage medium
  - Sata SSDs: very low
  - PCI-E SSDs: really really low
- Commit => One IO operation
- Write Amplification

Asynchronous Commits

- Flush primarily in background
- Checkpointing unaffected
- Configure according to requirements
  - synchronous_commit = off / local
  - per transaction / session / role / database / instance

Unlogged Tables

- CREATE UNLOGGED TABLE …
- ALTER TABLE … SET [UN]LOGGED;
- Durability: No crash safety
- Replication: Trigger based
- Write overhead: minimal

fsync = off

citusdata

Streaming Replication

Asynchronous Replication

- synchronous_commit = off / local
- synchronous_standby_names = ‘’
- Fast
- Few guarantees
- Highest Availability

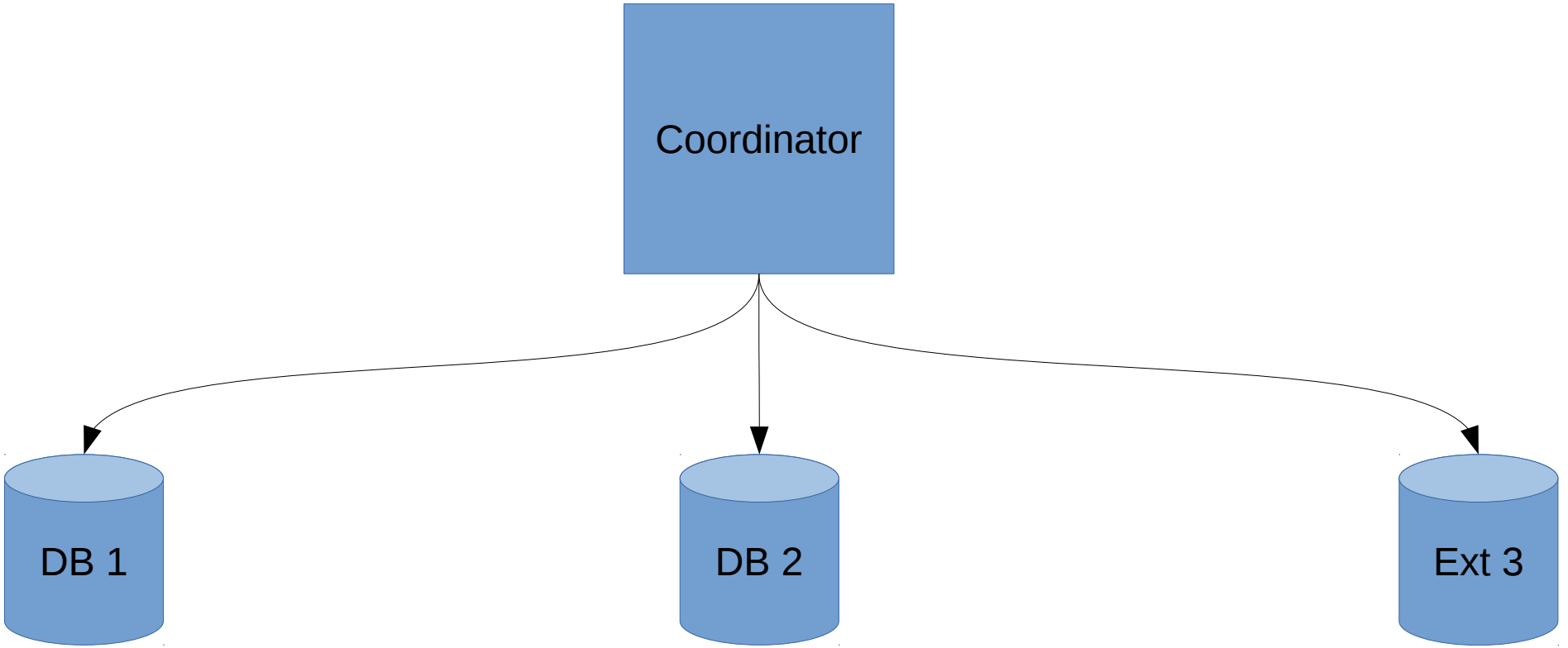
Synchronous Replication

- synchronous_standby_names = ‘a, b, ...’
- synchronous_standby_names = ‘k (a, b, …)’
- primary_conninfo = ‘… application_name = a’
- synchronous_commit =
  - off
  - local / on (synchronous_standby_names = ‘’)
  - remote_write
  - on (synchronous_standby_names = ‘...’)
  - remote_apply
- wal_sender_timeout

Synchronous Replication

- Not magic
  - Use selectively
- New failure modes
  - Use selectively
- Not necessarily consistent (CAP, not ACID)!
- Apply not synchronized (< 9.6)
- No quorum commit (10?)

Two-Phase Commit



BEGIN

INSERT

PREPARE TRANSACTION 'andres_1';

COMMIT PREPARED 'andres_1';

ROLLBACK PREPARED 'andres_1';

BEGIN

UPDATE

PREPARE

COMMIT PREPARED

BEGIN

FOO

PREPARE

COMMIT PREPARED

citusdata

Tunable Durability

Andres Freund
PostgreSQL Developer & Committer
Citus Data – citusdata.com - @citusdata

http://anarazel.de/talks/pgconf-sv-2016-11-16/durability.pdf

citusdata