The implementation of MariaDB parallel replication

Kristian Nielsen

MariaDB Foundation

TeqHub November 2024

TeqHub

Thanks to Nina and Jens for making these TeqHub happen!

About me

- Kristian Nielsen <knielsen@knielsen-hq.org>
- Chief Architect Replication, MariaDB Foundation
- Author of MariaDB group commit, Global Transaction ID (GTID) and parallel replication
- MySQL and MariaDB developer since 2005
- Free Software developer since 1990(ish)

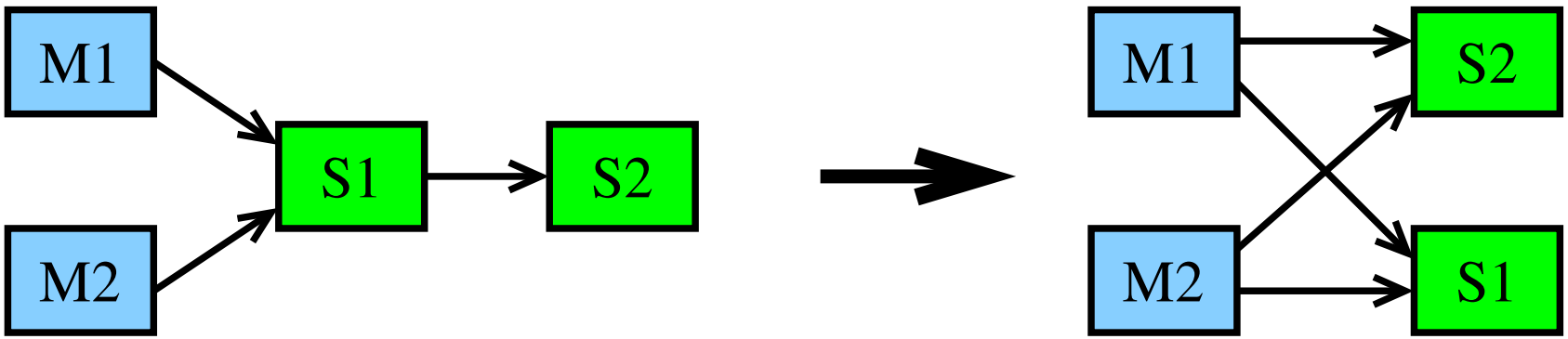
Replication background 1

Context: SQL database, application changing data:

- `INSERT INTO t1 VALUES (10, 100, "FooBar");`
- `UPDATE t2 SET a=a+1 WHERE b=5;`
- `DELETE FROM t3 WHERE pk1=10 and pk2="Knob";`
- `DROP TABLE t4;`
- `...`

Replication background 2

Master-slave replication:



Replication is **asynchronous**.

Replication background 3

Transactions on master run in **parallel**.
Transactions on slave replicate in **commit order**

T1	T2	T3
	BEGIN	
		BEGIN
		DELETE 3
	INSERT 1	
BEGIN		
INSERT 2		
		UPDATE 2 <wait>
	UPDATE 4	
COMMIT		<wakeup>
	COMMIT	
		COMMIT

The need for concurrency

Can replicate transactions one-by-one.

Careful row-level locking on master ensures **identical** result on slave.

BUT! One by one will be **too slow**, slave will not be able to keep up with master.

Need to replicate transactions **in parallel**.

Parallel replication – the challenge

T1	T2	T3
	BEGIN	
		BEGIN
		DELETE 3
	INSERT 1	
BEGIN		
INSERT 2		
		UPDATE 2 <wait>
	UPDATE 4	
COMMIT		<wakeup>
	COMMIT	
		COMMIT

Different query execution order on slave?

Solution: Optimistic parallel replication

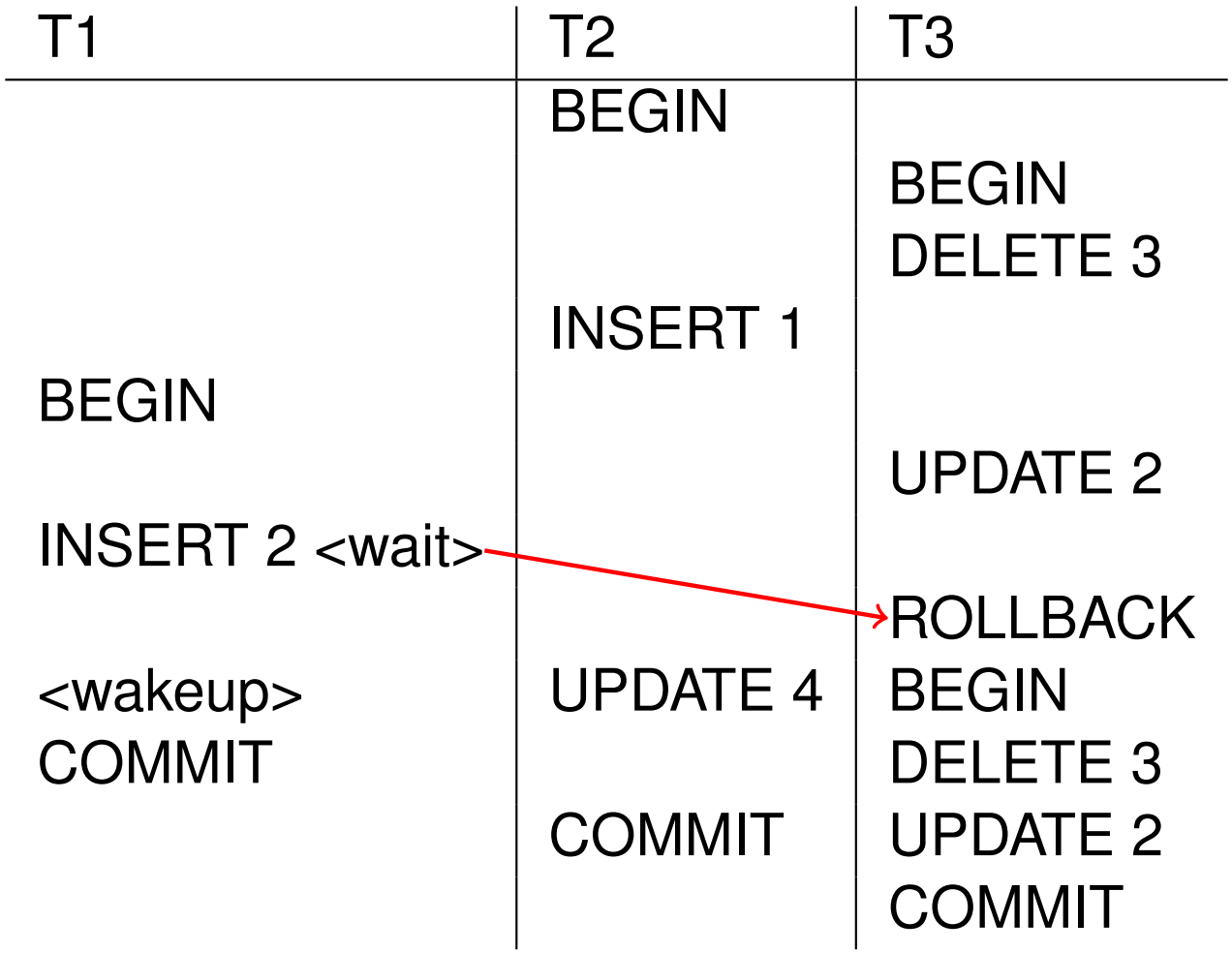
Central idea:

- Replicate **queries** freely in parallel
- Replicate **commits** strictly in sequence
- Different query order can lead to conflicts
- **Detect** any conflicts
- **Resolve** conflicts by rollback and retry

Benefits:

- Reuse all existing row locking code etc.
- Strict commit sequence ensures correctness
- No need for separate complex conflict analysis

Solution: Optimistic parallel replication 2



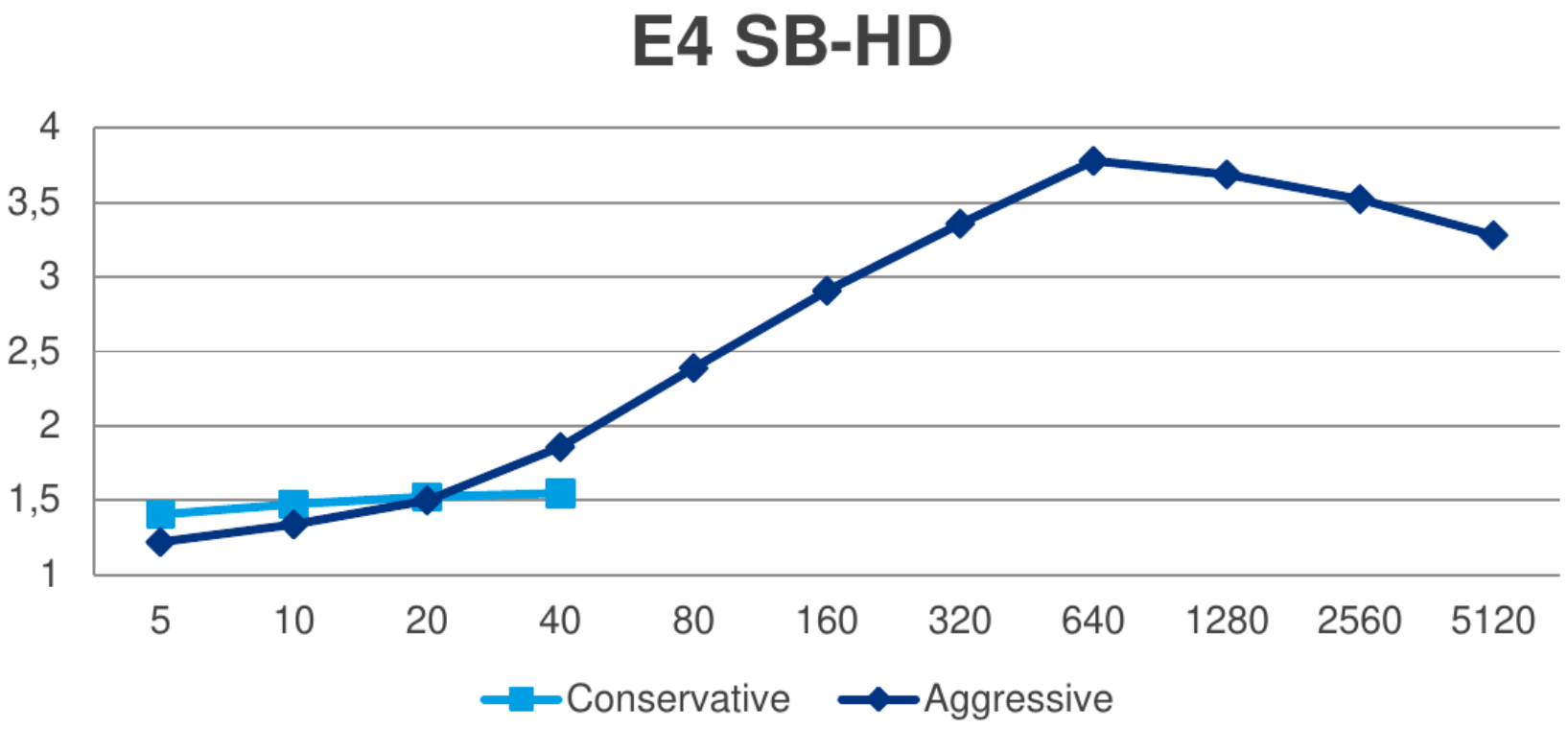
Booking.com

MySQL Parallel Replication: inventory, use-cases and limitations

Jean-François Gagné (System Engineer)
jeanfrancois DOT gagne AT booking.com

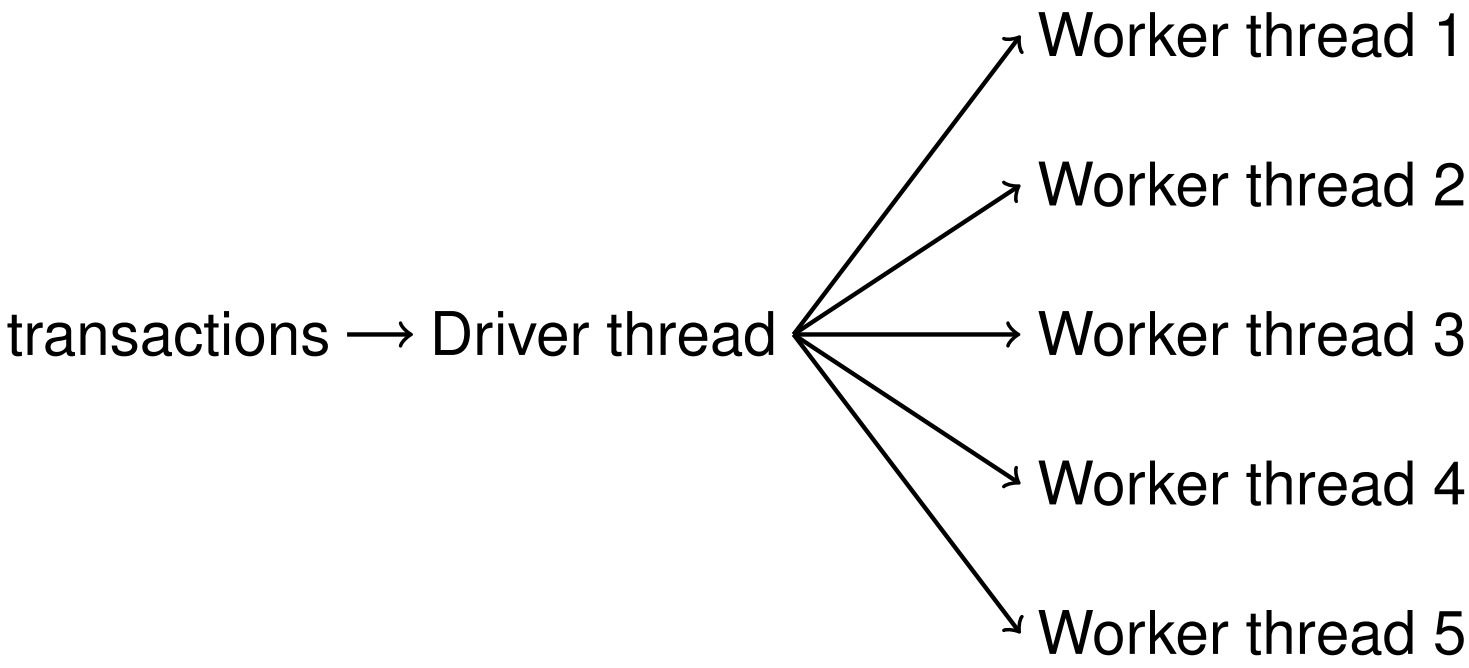
Presented at FOSDEM 2016

Benchmarks 2



Scheduling

Schedule round-robin amongst *N* worker threads:



See `do_event()` and `handle_rpl_parallel_thread()` in `sql/rpl_parallel.cc`

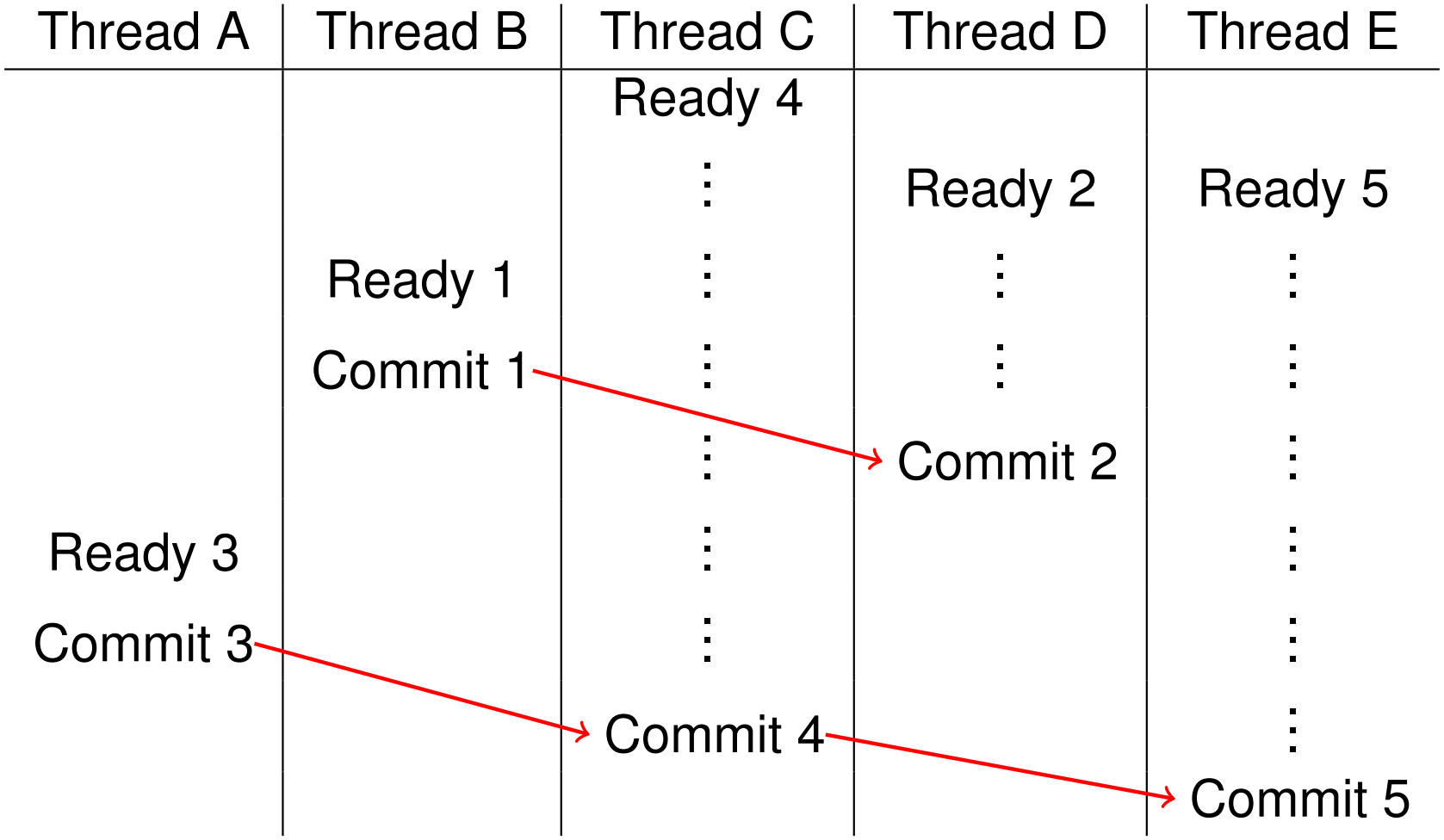
Handling conflicts

- We know the commit order from the master
  - Say $T_1, T_2, T_3, \ldots$
- If T_2 waits for T_1, that is fine
- If T_1 waits for T_2, it is a conflict
  - T_1 will be blocked from committing before T_2
  - Must abort and roll back T_2
- Hook the InnoDB locking code to report lock waits
- Check the commit order in the hook and handle any conflicts.

See `lock_wait()` in `storage/innobase/lock/lock0lock.cc` and `thd_rpl_deadlock_check()` in `sql/sql_class.cc`

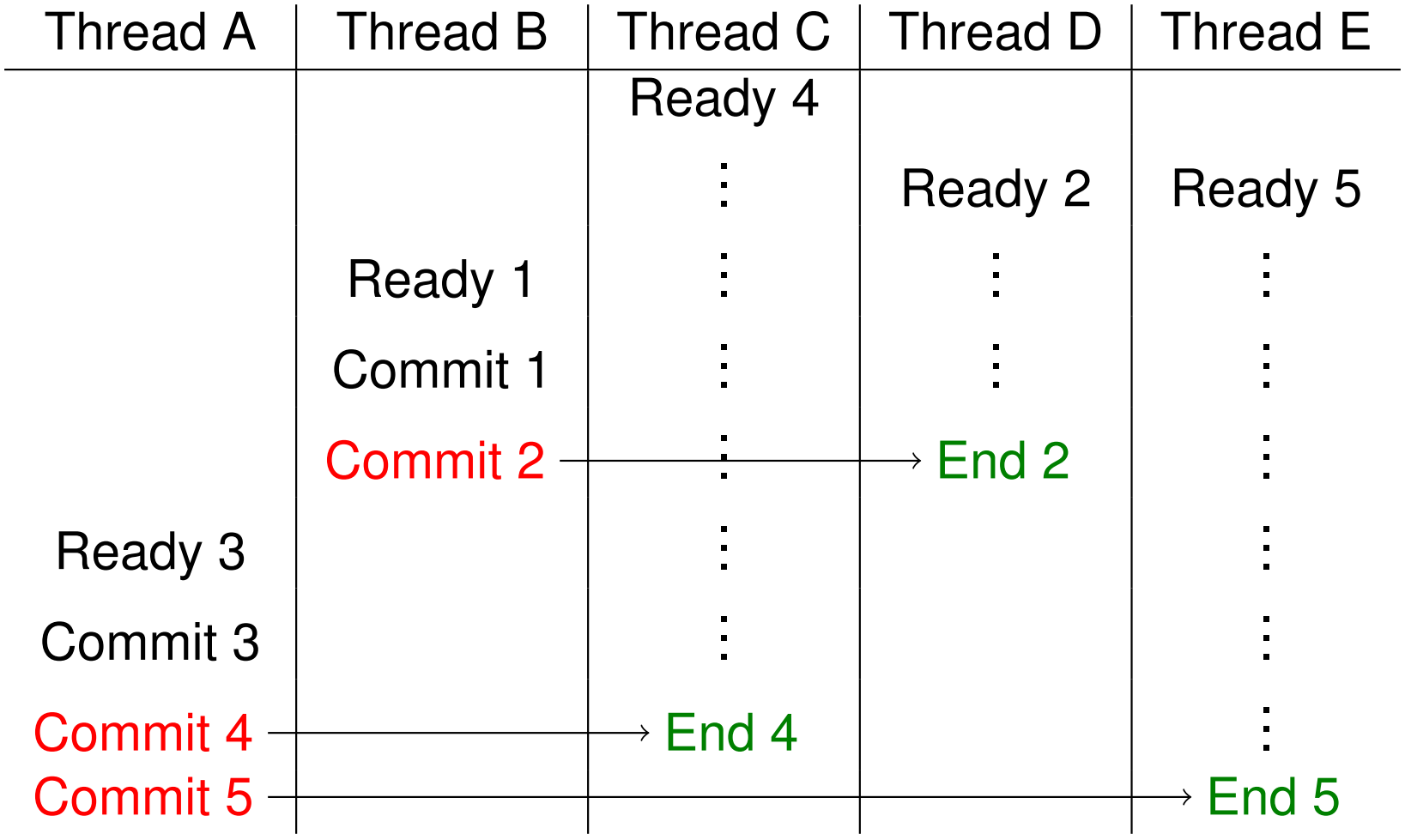
Ordering commits 1

How to coordinate commits between threads?



Ordering commits 2

Commit a group of transactions in one thread

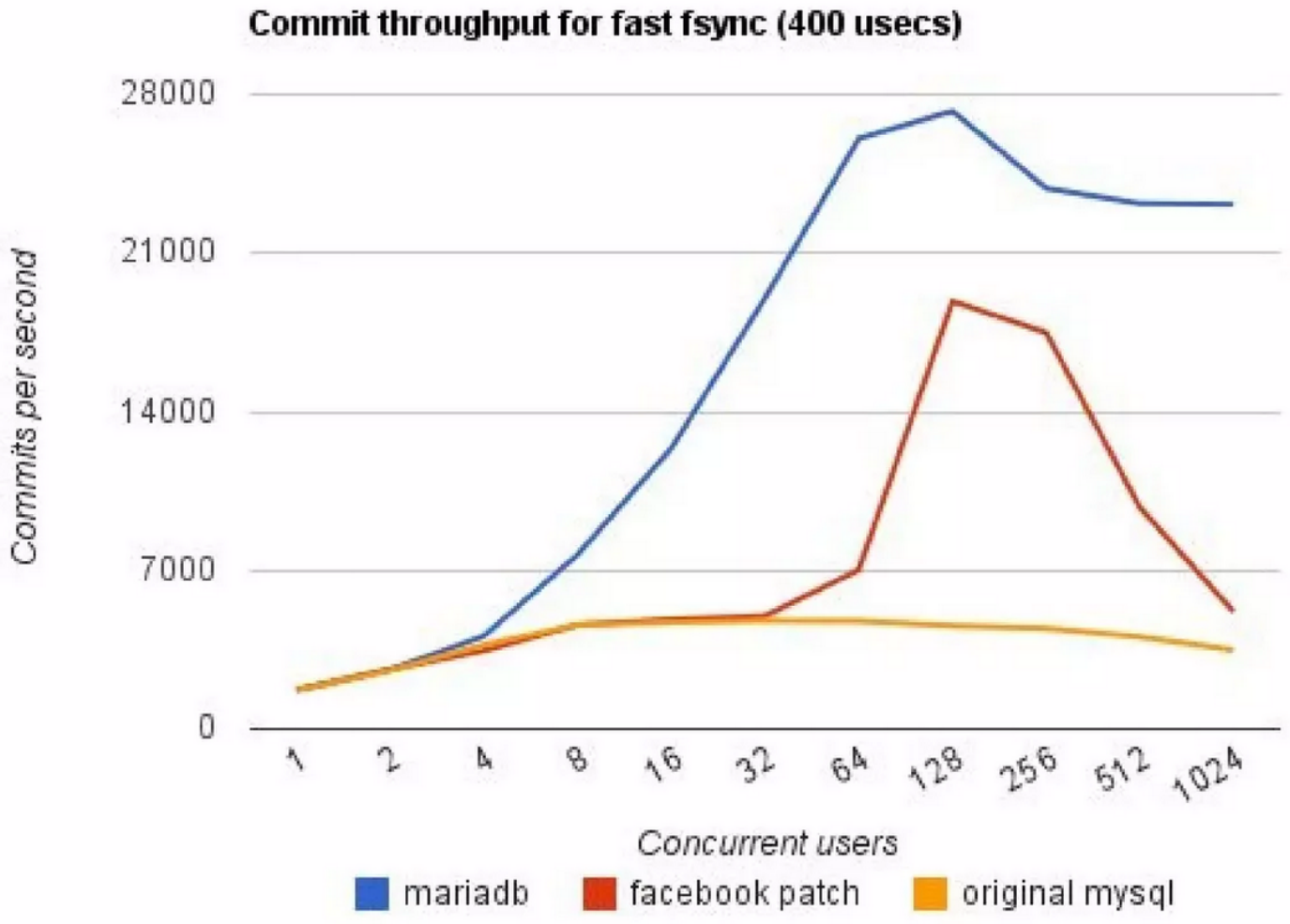


Ordering commits 3

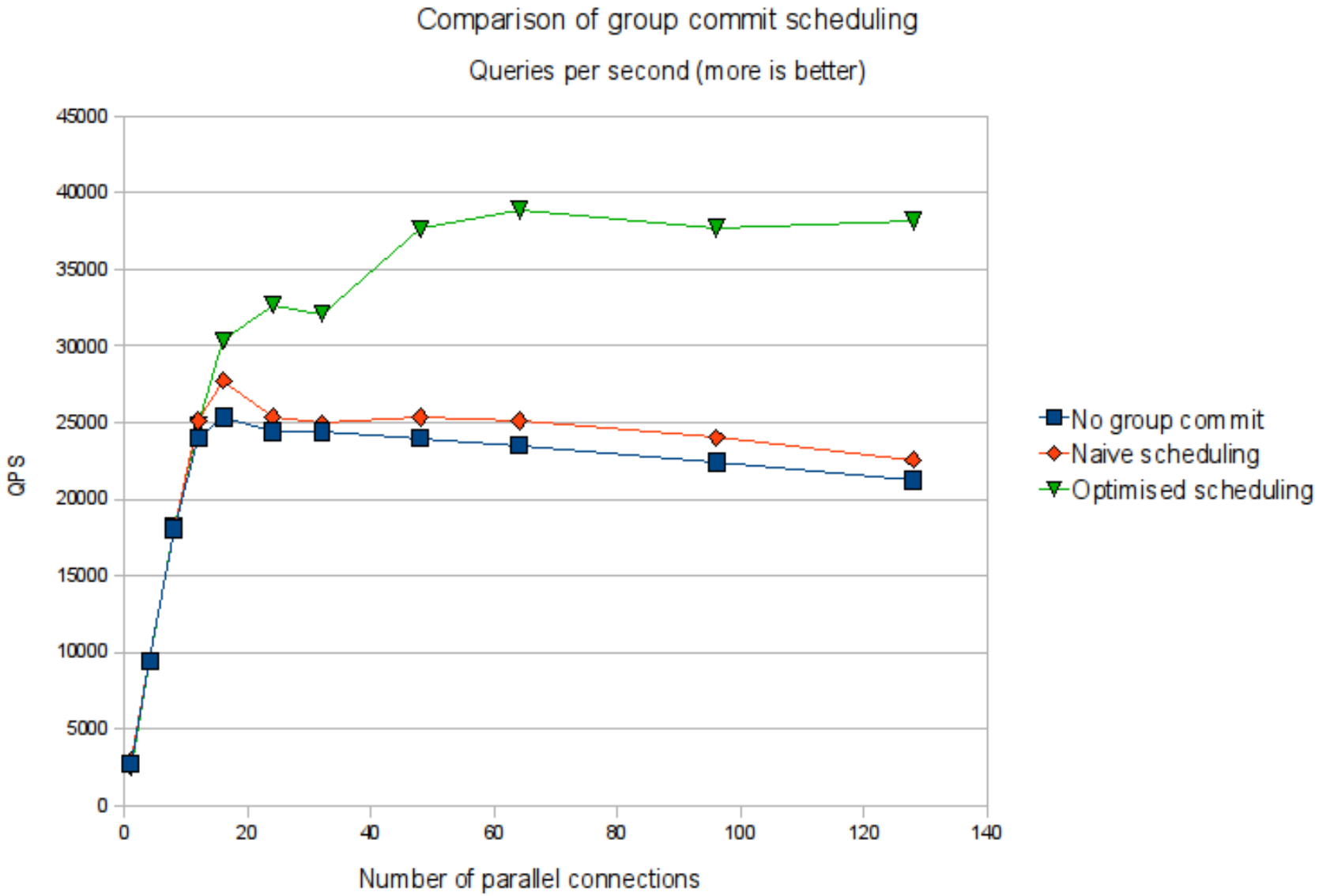
- The sequencing of commits needs to happen in sequence
- Do the serialized execution in a single thread
- Avoid the overhead of context switches in the critical path
- Completion of transactions can happen out-of-order, no waiting

See `wait_for_prior_commit()` in `sql/sql_class.h`, `wait_for_prior_commit2()` in `sql/sql_class.cc`, and `queue_for_group_commit` in `sql/log.cc`

Optimizing thread scheduling



Optimizing thread scheduling



Conclusion

- Parallel processing essential for replication on busy databases
- **Optimistic** parallel replication a great way to get high parallelism while ensuring correctness
- Careful design needed to reduce bottlenecks around thread scheduling and coordination
- A nice practical use-case of concurrency

Links

Further reading:

- Blog: `https://knielsen-hq.org/w/`
- Parallel replication: `https://mariadb.com/kb/en/parallel-replication/`
- MariaDB Foundation: `https://mariadb.org/`

Kristian Nielsen:

- Mail: `knielsen@knielsen-hq.org`
- Consulting: `consulting@kristiannielsen.dk`

Questions?