

Galera Replication

Synchronous Multi-Master Replication
...for any DBMS

Seppo Jaakola - Codership

Contents

1. Technical Overview

- Certification based replication

2. Replication API

- wsrep API

3. Benchmarks

- Sysbench, DBT2, 100% inserts, WAN

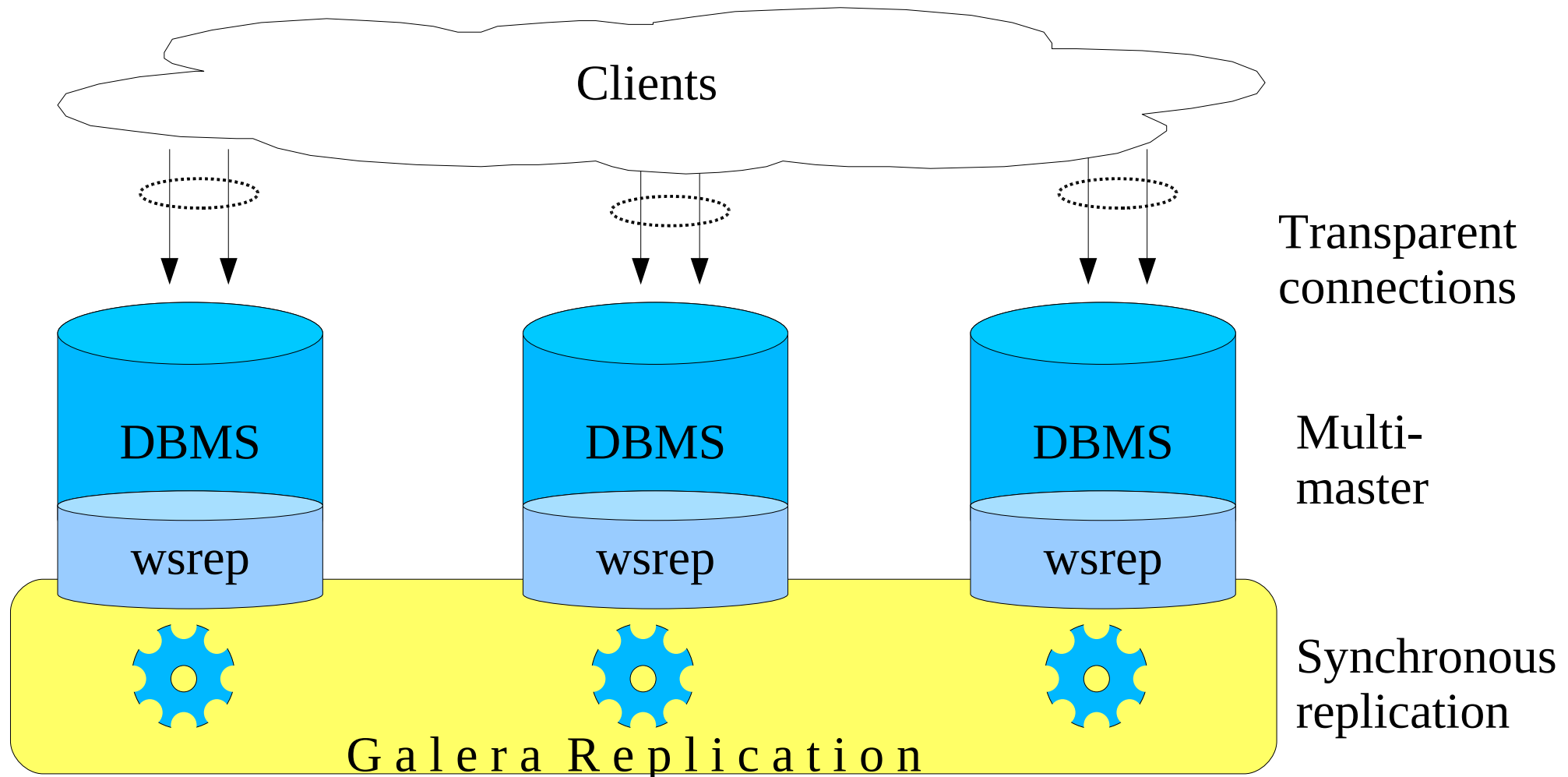
4. Galera Project

- Release status
- Road map
- Codership

Focusing

- Galera is generic solution – fits any DBMS
- Replication API is important
- Galera replication is very efficient

Galera Cluster



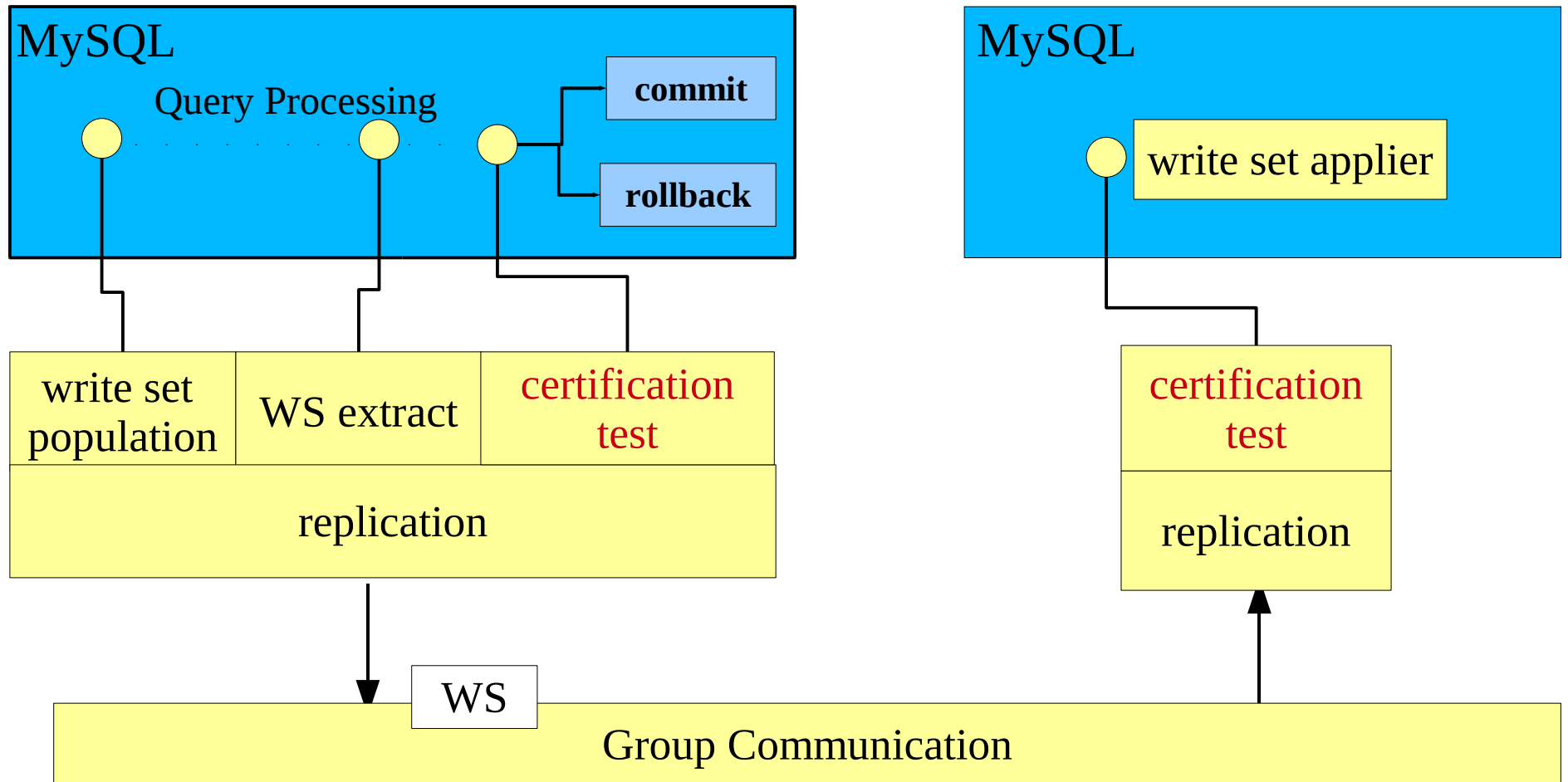
Galera Replication

- Synchronous multi-master replication
 - High Availability
- No middle-ware, connections directly to DBMS
 - Transparency
- Row events, row level locking
 - Write scalability
- Certification based replication method

Galera Replication

- Galera is pluggable software library
- **Generic** replication system to make a cluster from any transactional DBMS
- First implementation MySQL/InnoDB cluster

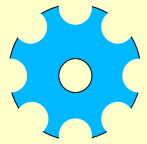
Certification Based Replication



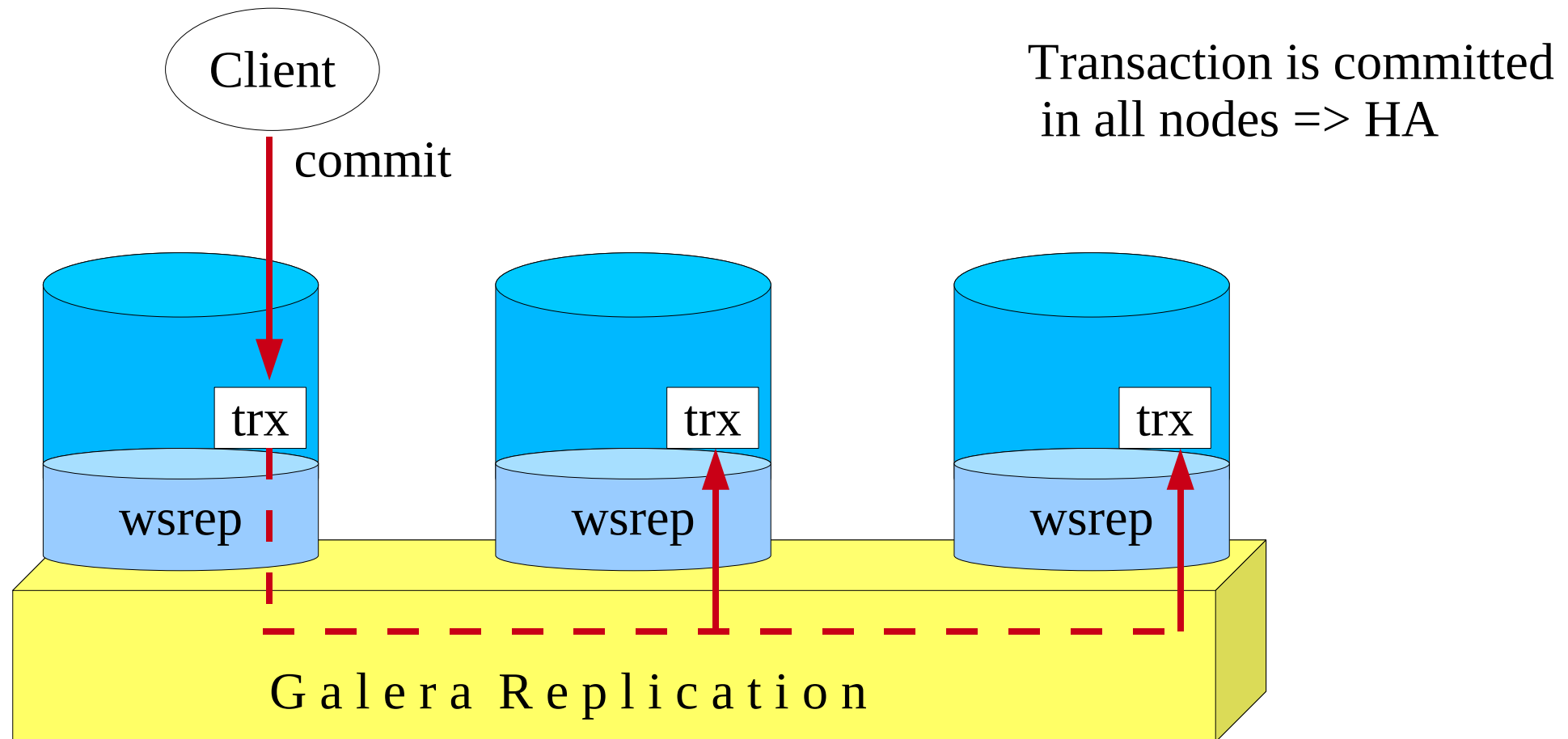
Write Set

Seqno
last_committed_seqno
Keys
Applying info

- Sequence number of the trx
- Sequence number of the last committed trx, which affected the processing state
- All row changes are identified with keys
- SQL statements or RBR events



Synchronous Replication



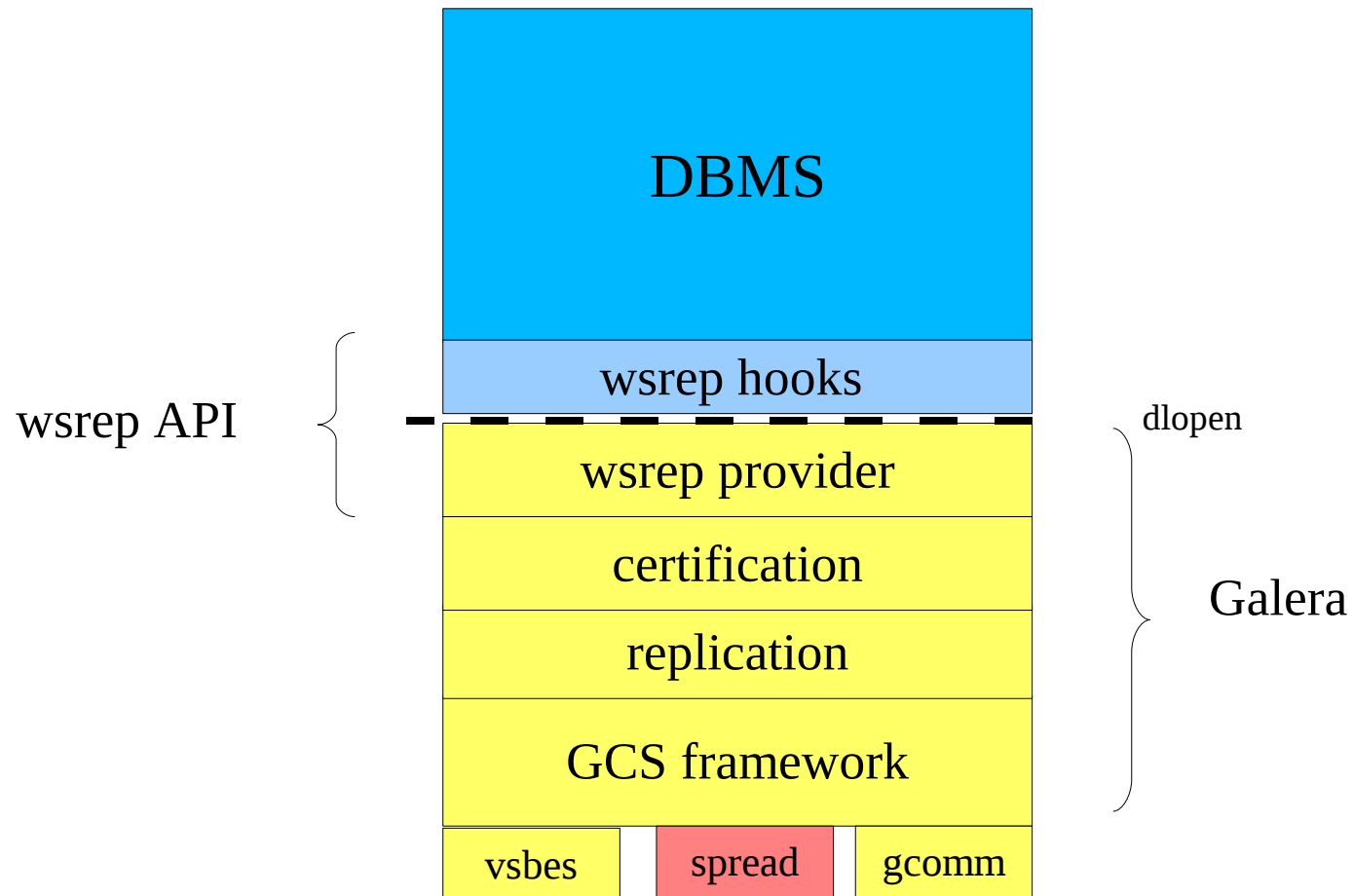
Synch vs. Async

- Asynchronous replication is easy to implement
- ...but makes application more complicated
 - reads and writes must go to different nodes
 - write node is a moving target
 - read nodes contain old data, always
- All nodes must synchronize eventually, one way or another
- Synchronous cluster looks like one database to the application

Replication API

- Galera integrates closely in DBMS transaction processing
- There must be an interface between DBMS and replication system

Replication API





wsrep API

- Codership's replication API
- A generic interface for DBMS and replication system
- Defines:
 - **Write Set** for describing changes
 - Write set replication API for transactions
 - DDL replication using TO isolation
- <https://launchpad.net/wsrep>

Other Replication APIs

- MySQL's API cooking up:
 - http://forge.mysql.com/wiki/MySQL_Replication:_Walk-through_of_the_new_5.1_and_6.0_features
- Drizzle's API, already there:
 - <http://www.jpipes.com/index.php?/archives/290-Towards-a-New-Modular-Replication-Architecture.html>
- MariaDB specifying new API
 - <https://lists.launchpad.net/maria-developers/msg01998.html>

Optimistic Concurrency Control

- Transactions proceed independently in each cluster node assuming they can eventually commit
 - There can be cluster wide conflicts
- Victim trx must abort in master node, and avoid committing in other nodes
- ER_DEADLOCK returned for cluster aborts

Optimistic Concurrency Control

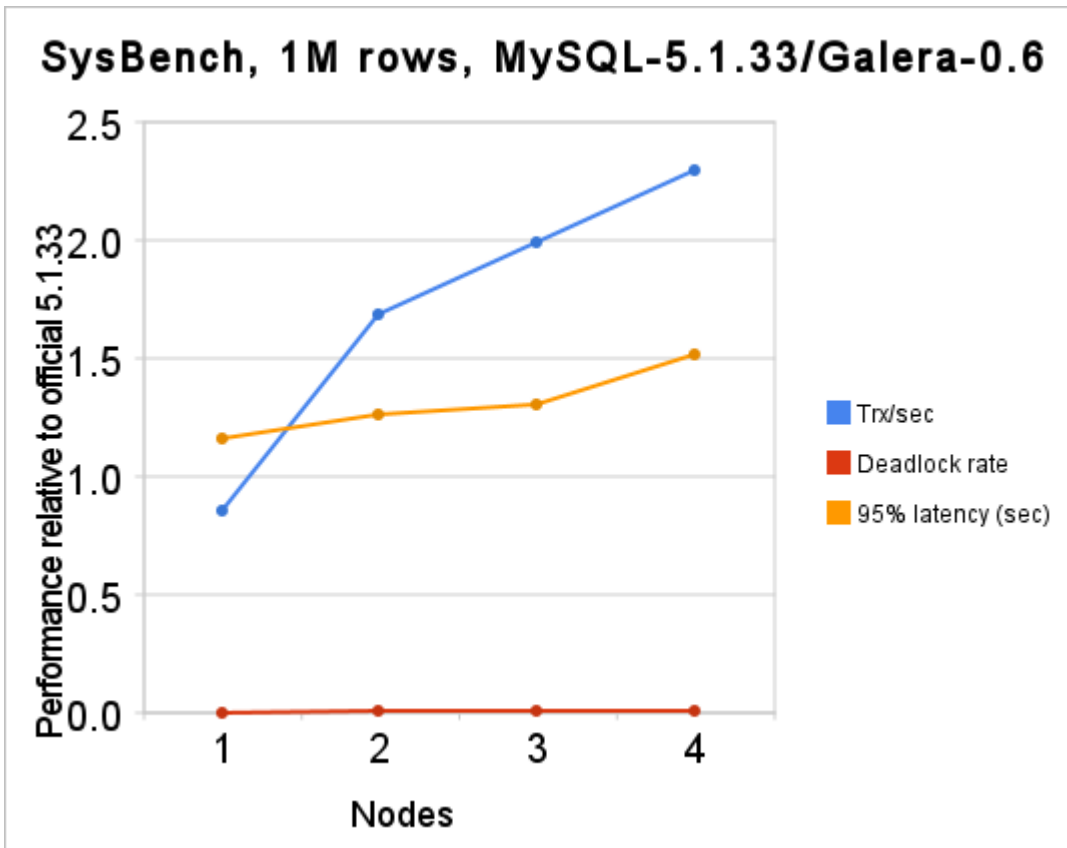
- **Hot spots** are bad
 - .e.g. DBT2 shows pretty high conflict rate
- Long lasting transactions are vulnerable
- Rollbacks eat performance, but rollback happens only in one node, all other nodes just avoid applying
- With problematic SQL load, cluster can be adjusted to have a smaller number of write capable nodes

Benchmarking

Benchmarking

- Tested with several benchmarks
 - Sysbench, dbt2, DOTS, osdb, jmeter, sqlgen...
- Benchmarks testing with 'physical hardware' and with Amazon EC2 small and large instances
- Currently tests range up to 5 cluster nodes
- In general, shows good scalability even with write intensive work loads

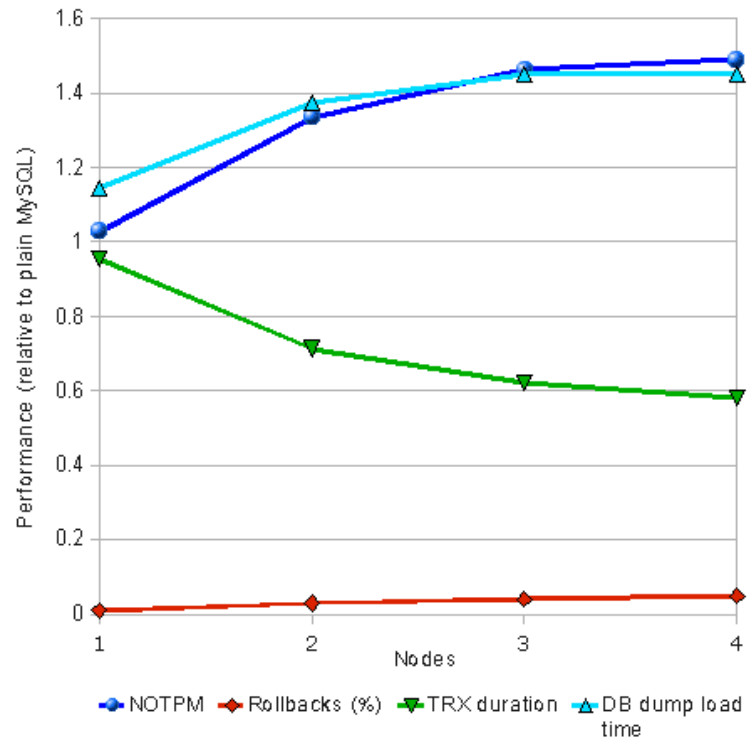
Sysbench Benchmarks



- Sysbench oltp mode test
- EC2 Large instances

nodes	users	trx/s	deadlks	95%lat
1	18	385	0	0.092
2	36	761	2.54	0.100
3	45	900	3.42	0.103
4	60	1034	4.54	0.120
official 5.1.33 binary:				
1	18	451	0	0.079

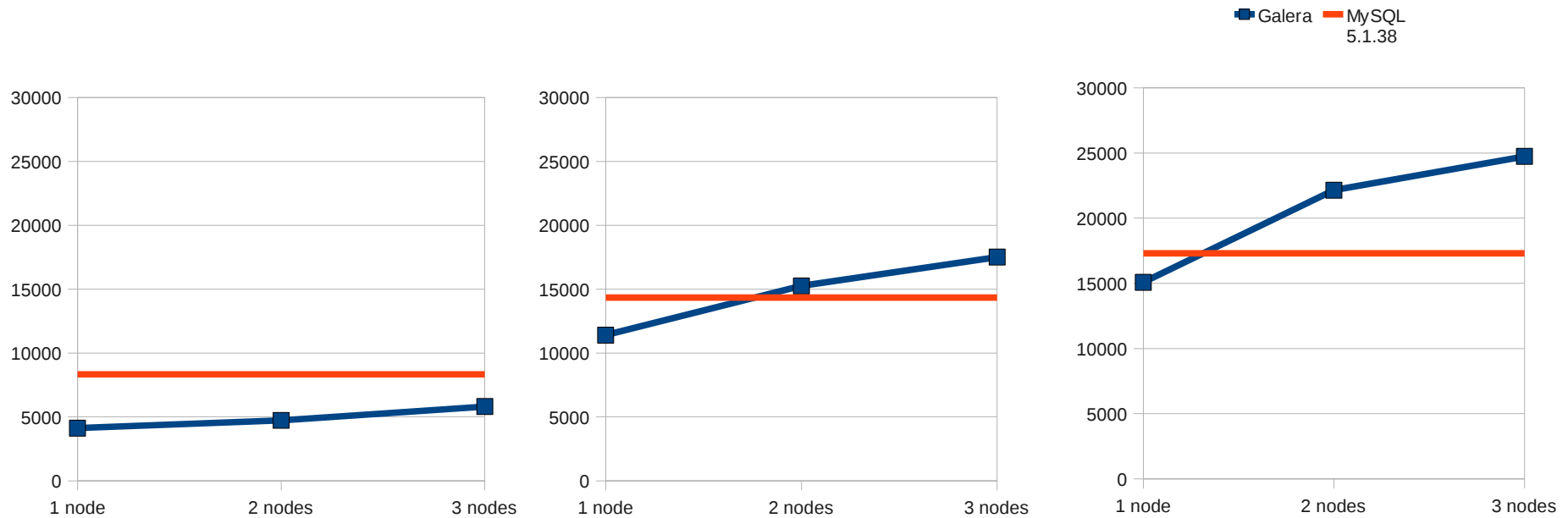
Dbt2 Benchmark



- EC2 large instances
- Dbt2 benchmark
- 60 warehouses

	Conns	NOTPM	Rollbacks(%)	TRX duration(sec)	Dump load(min)
Plain 5.1.30:	20	~7220	1	2.27	26
1 node	: 12	~7420	1	2.17	30
2 nodes	: 24	~9630	3	1.63	36
3 nodes	: 36	~10555	4	1.41	38
4 nodes	: 48	~10753	5	1.32	38

100% Insert Benchmark

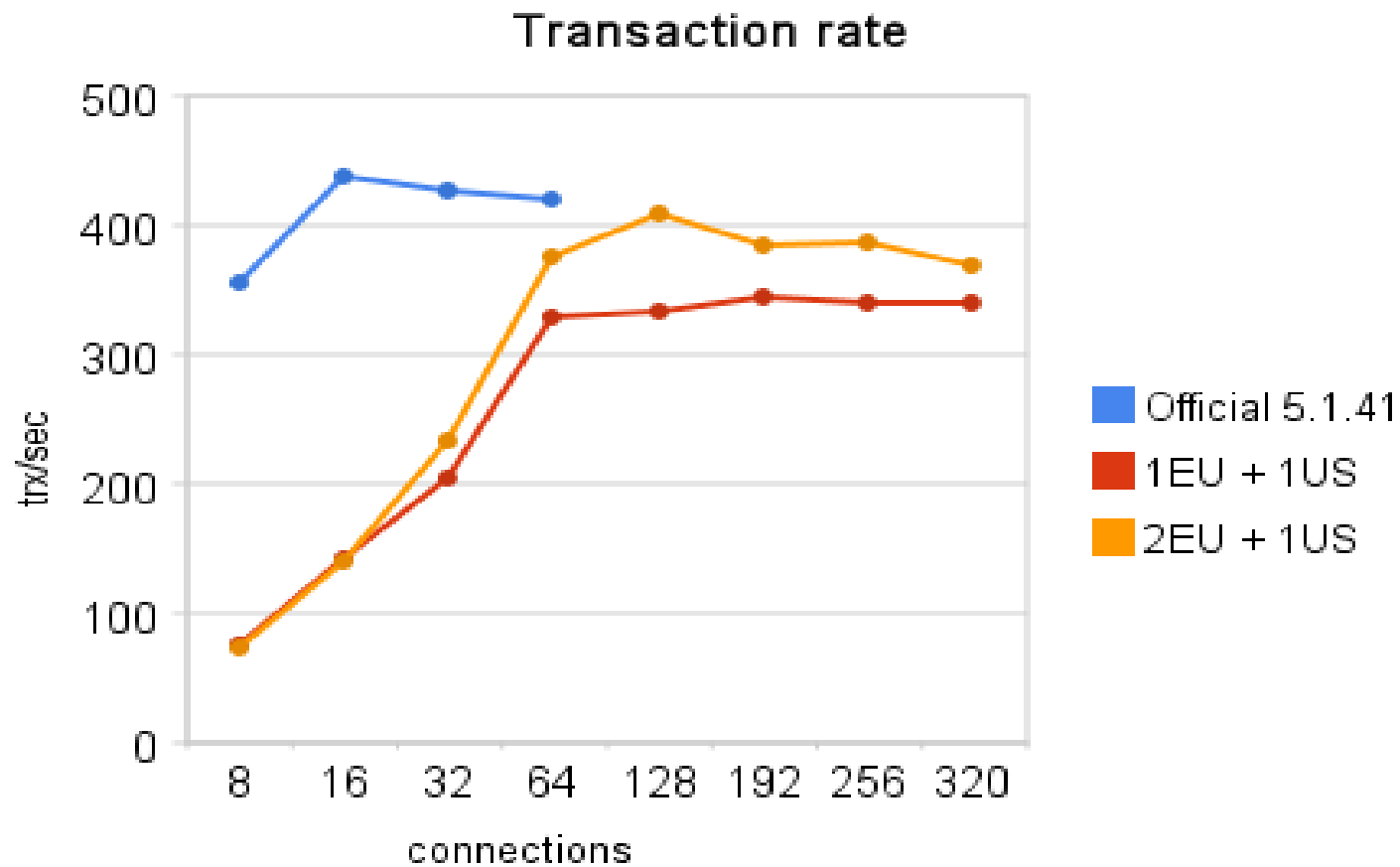


1 insert / trx

10 inserts / trx

100 inserts / trx

Synchronous WAN Replication



- Sysbench oltp
- Amazon EC2

Galera Project

- Development started in 2008
- Public releases since Jan 2009
- Release 0.7 is fully open source and is feature complete for production use

Release 0.7

- Current release 0.7.2
 - Stable release
 - Production readiness
- Simple management & installation utilities
- Nodes can join/drop active cluster
- State transfer by mysqldump
- “Reasonably” good performance

Road Map

- Maintenance releases for 0.7 series
- Release 0.8 – **Optimization Milestone**
 - Incremental State Transfer
 - Xtrabackup, LVM
 - Multicast GCS
 - Q2/10
- Release 0.9 - **Security Milestone**
 - TLS tunneling
- Release 1.0 - **Management Milestone**
 - Management & Monitoring tools

Summary

- Certification based replication turns out effective
 - High Availability
 - Transparency
 - Good scalability even with high write rates
- wsrep API is “not too hard” to implement
- Any (transactional) DBMS can leverage this replication possibility

Codership – The Saga

- Founders Seppo Jaakola, Alexey Yurchenko, Teemu Ollakka
- Fin-Rus community working from Finland
- Experts in distributed systems & DBMS development, information security
- Set Sail Oct 2007
- Products:
 - Galera
 - GLB (Debian ITP)
 - sqlgen

Get in Touch!

codership

- R&D consulting services
- Support subscriptions
- Downloads available: <http://www.codership.com>
- info@codership.com
- Mailing list: codership-team@googlegroups.com