Postgres-XC
Postgres Open 2011

Michael PAQUIER
2011/09/16

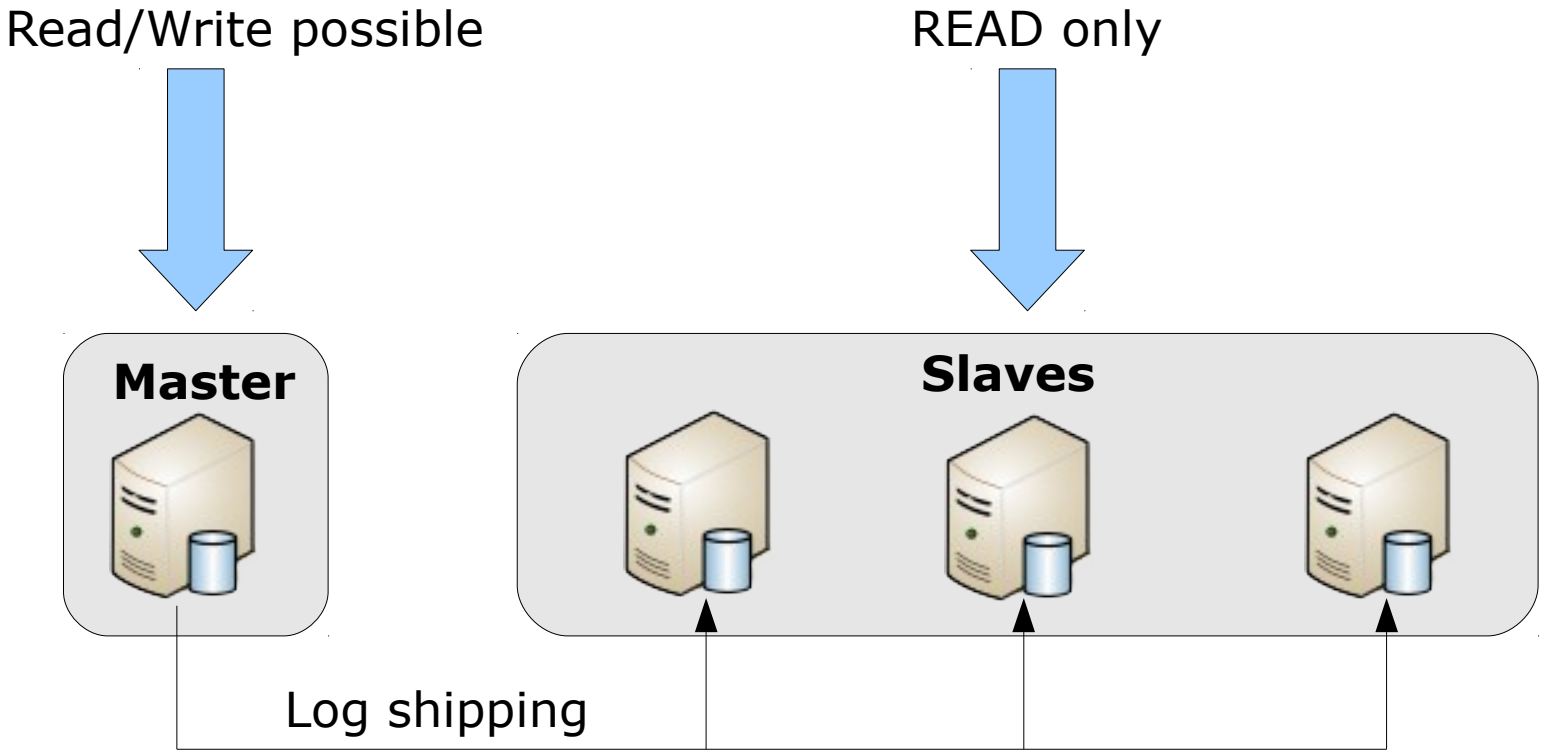
What is Postgres-XC?

- Project page: http://postgres-xc.sourceforge.net
- Write-scalable, multi-master clustering solution for PostgreSQL ?? @-@
- Symetric cluster of PostgreSQL
  - No Slave and no Master
  - Transparent Transaction Management
  - Every node can issue both READ/WRITE
  - Shared-nothing
- PostgreSQL license

Core architecture

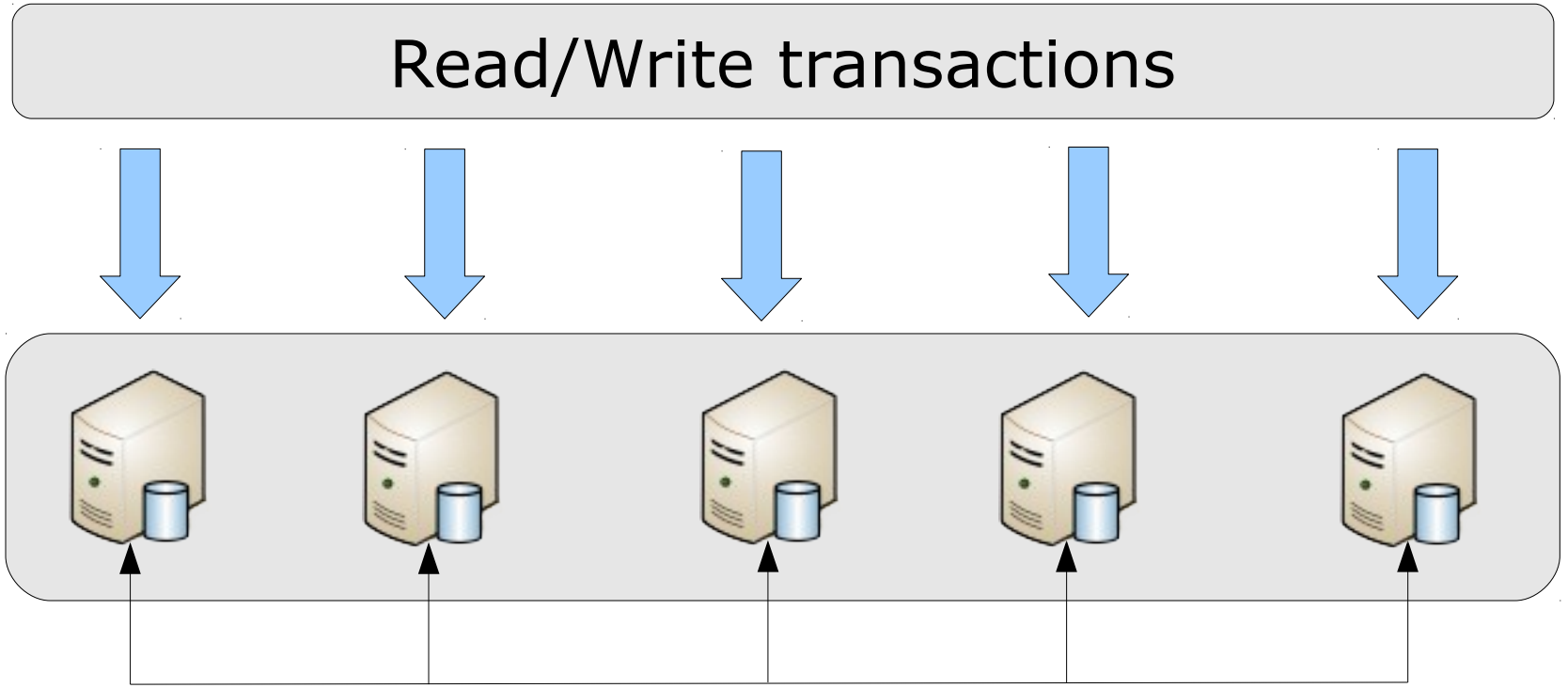
About PostgreSQL 9.1

- Streaming replication and HOT-Standby



- Asynchronous mode => timestamp view not consistent
- Synchronous mode => timestamp view consistent

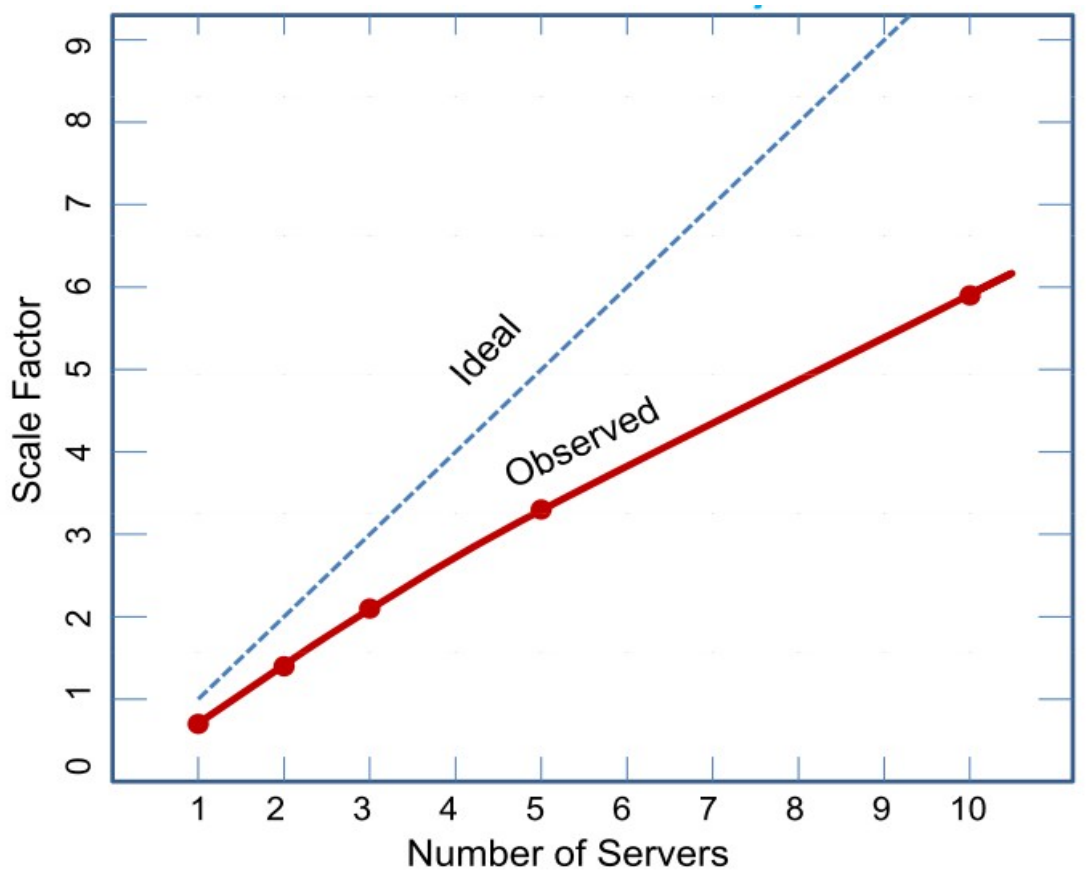
And Postgres-XC itself?



Distributed/replicated tables
Same data consistency/transparency
Same timestamp view

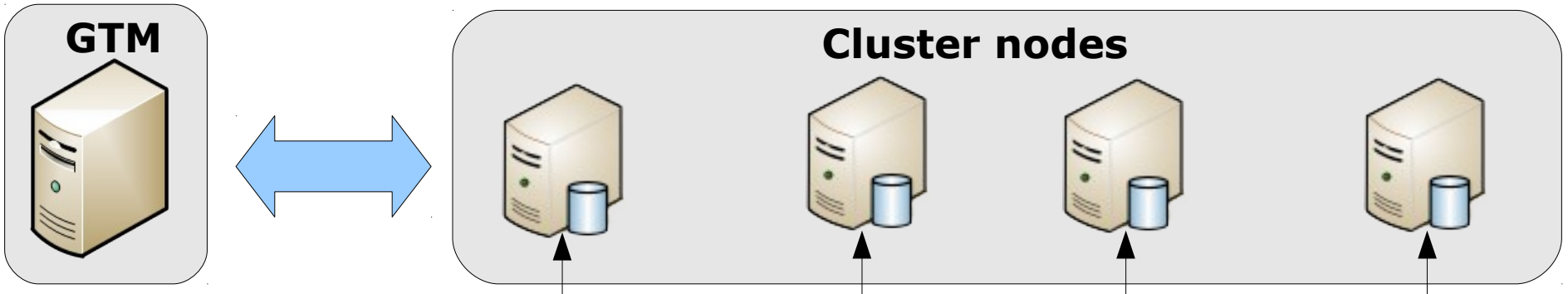
Scalability measurements

- Tests done with DBT-1 (TPC-W)



GTM: designed for transparency

- Cluster nodes are fed with a global snapshot obtained from a unique GTM node.

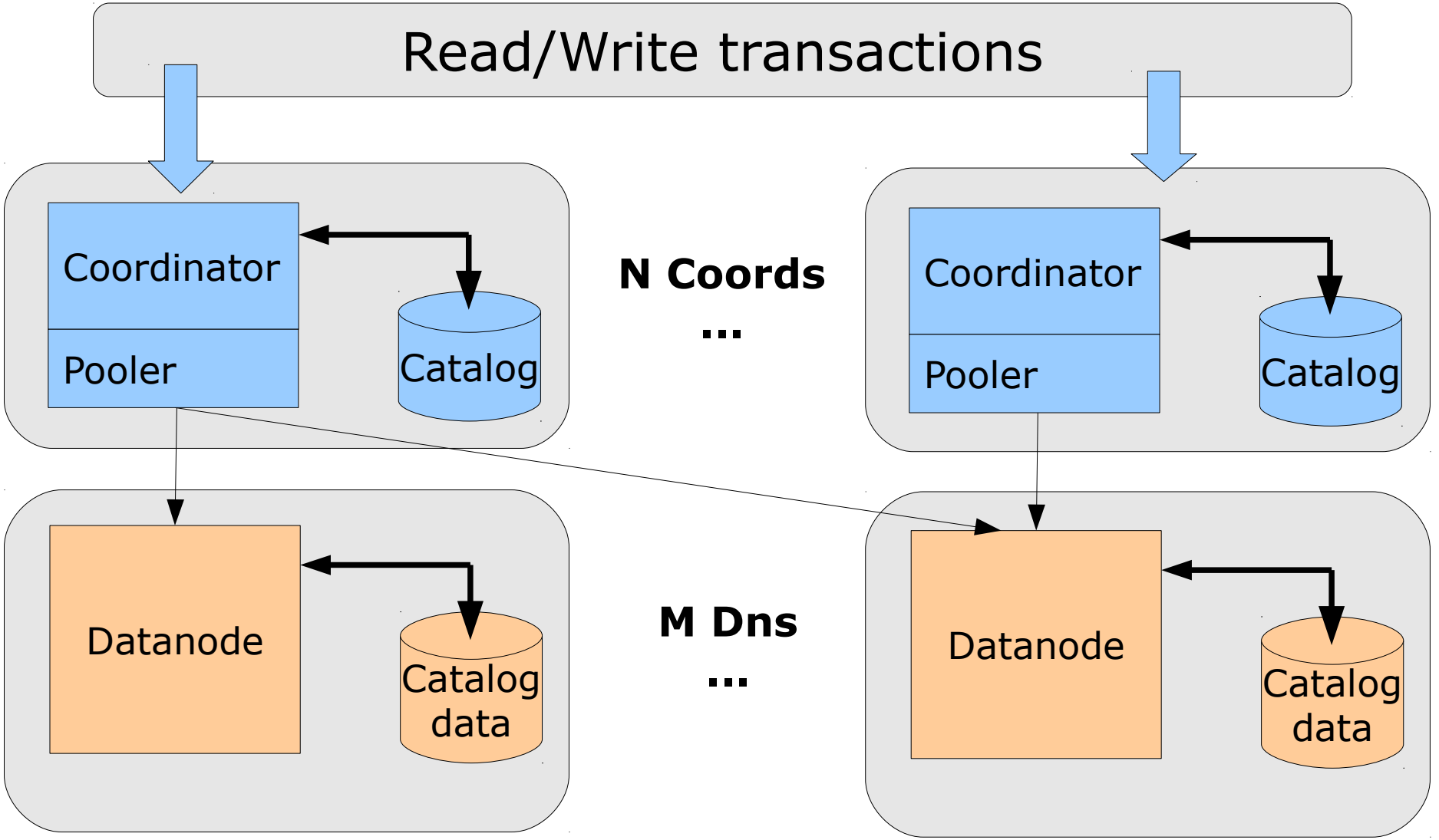


Snapshot
GXID
Timestamp
Sequence values

Node types (1)

- Coordinator
  - Holds table information (distribution type, key)
  - Entry point for applications, remote node
  - Connection pooling
- Datanode
  - Holds the data, backend node in cluster
  - More or less like a normal Postgres server
- So what?
  - All nodes share the same synchronized catalogs

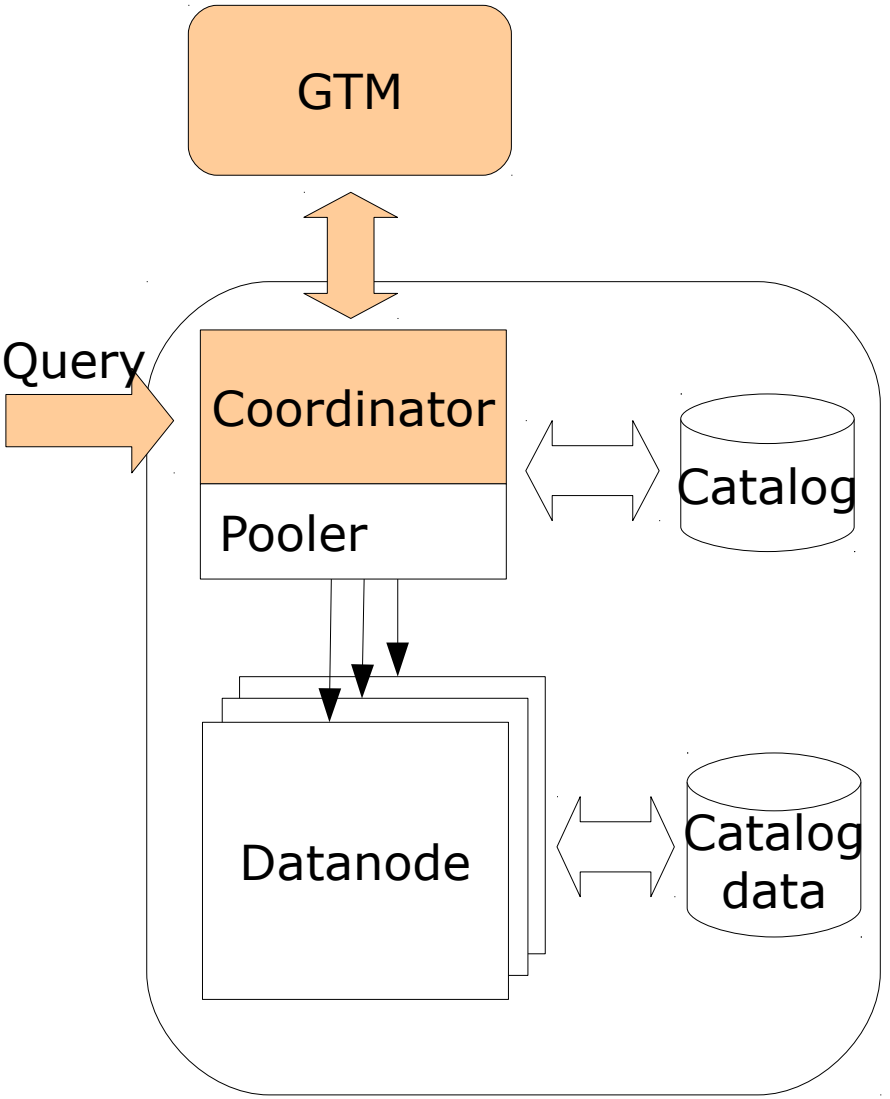
Node types (2)



Key algorithm

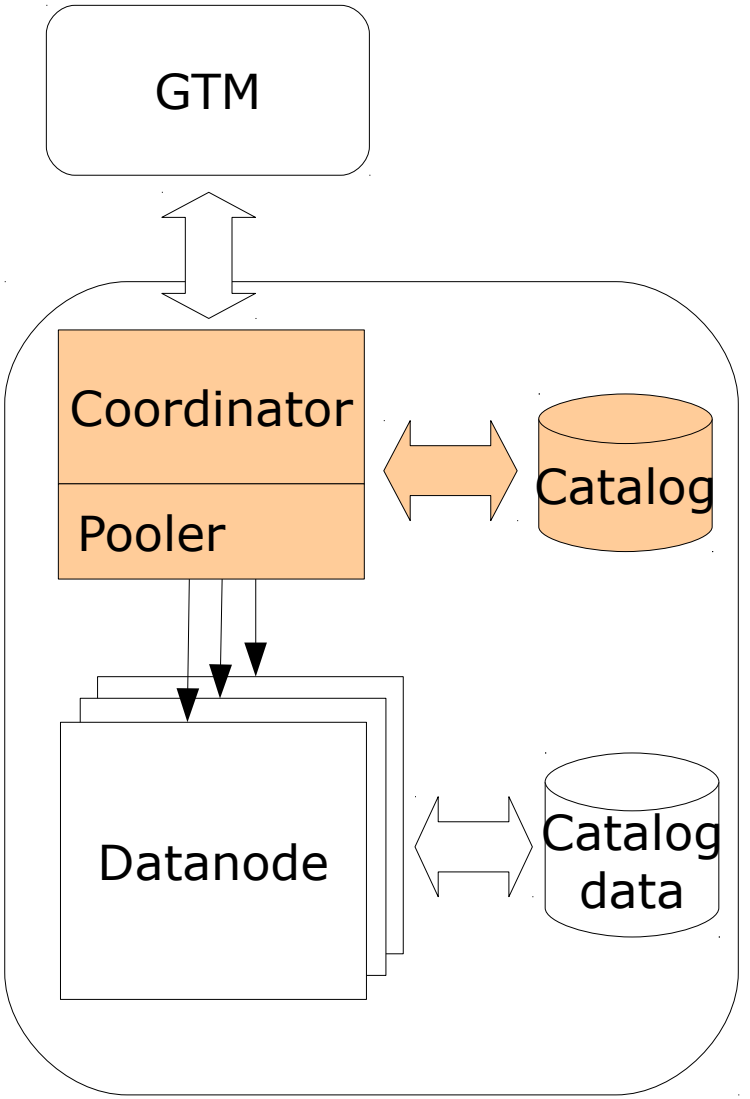
Query algorithm (1)

- Receive query from application
- Get snapshot, GXID and timestamp from GTM



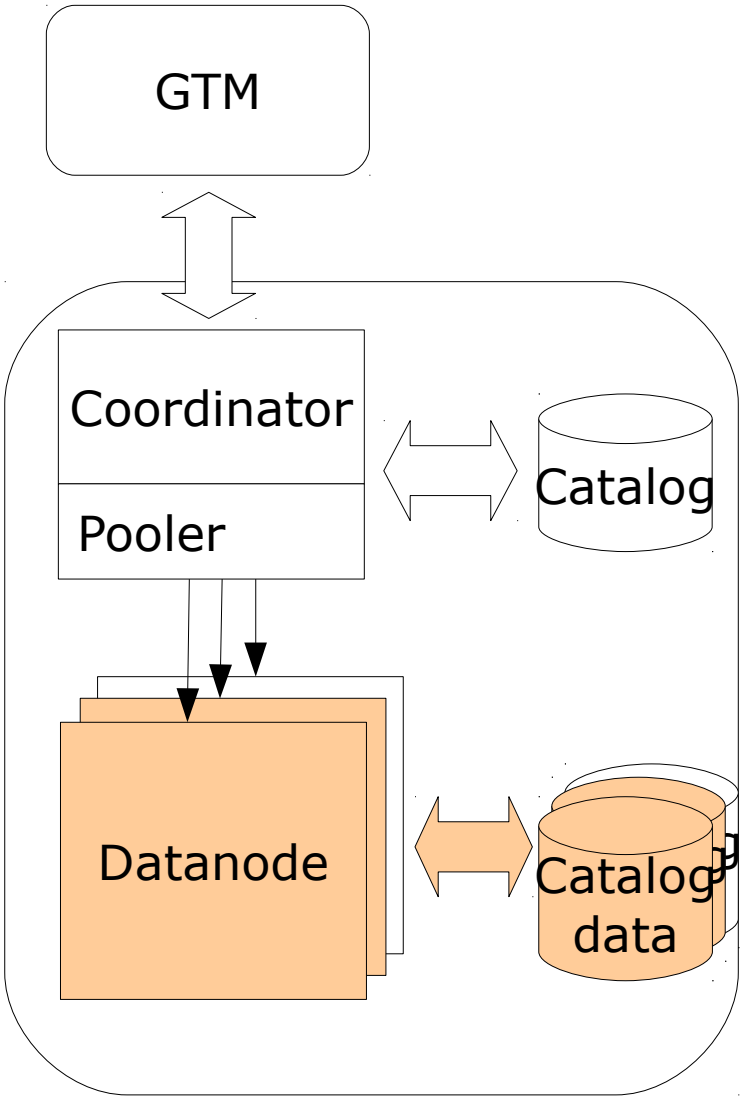
Query algorithm (2)

- Incoming statements: analyzer and rewriter
- Planning: analyze nodes to be involved. Build queries for local nodes (push down if necessary)



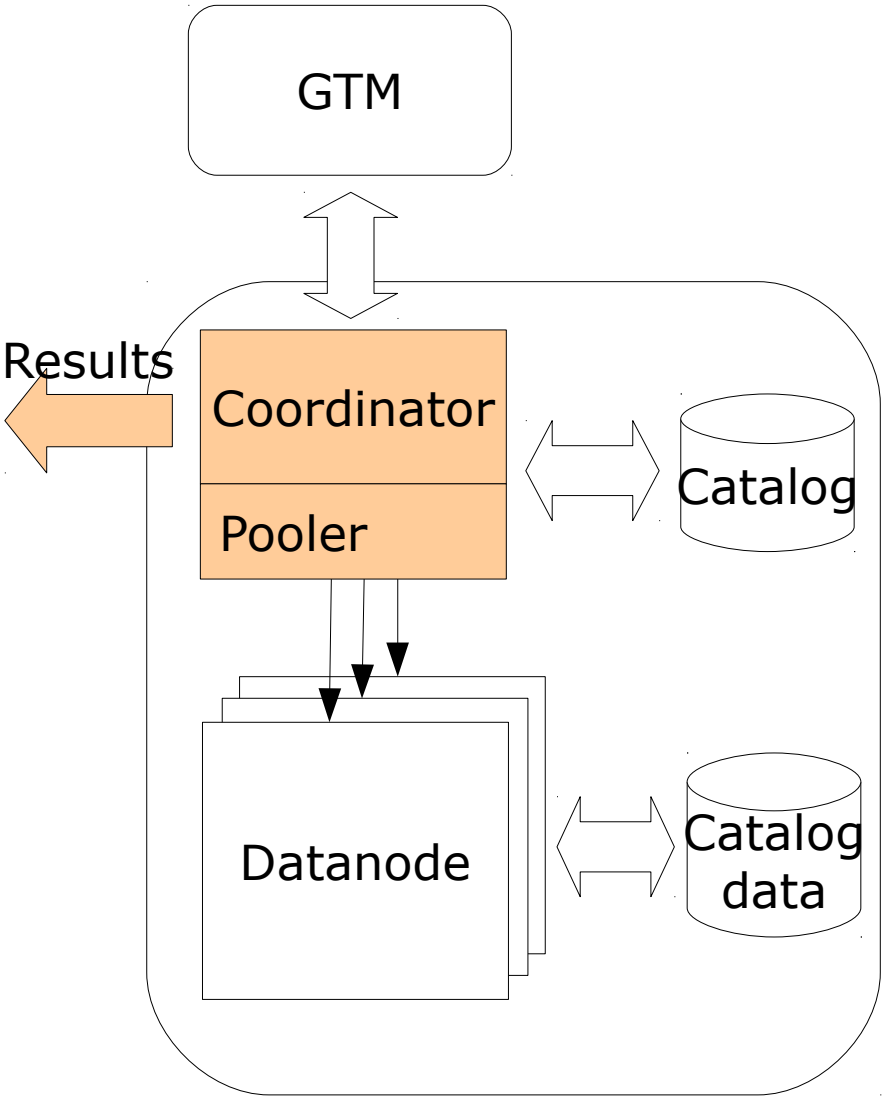
Query algorithm (3)

- Run queries on remote Datanodes and send back results to Coordinator

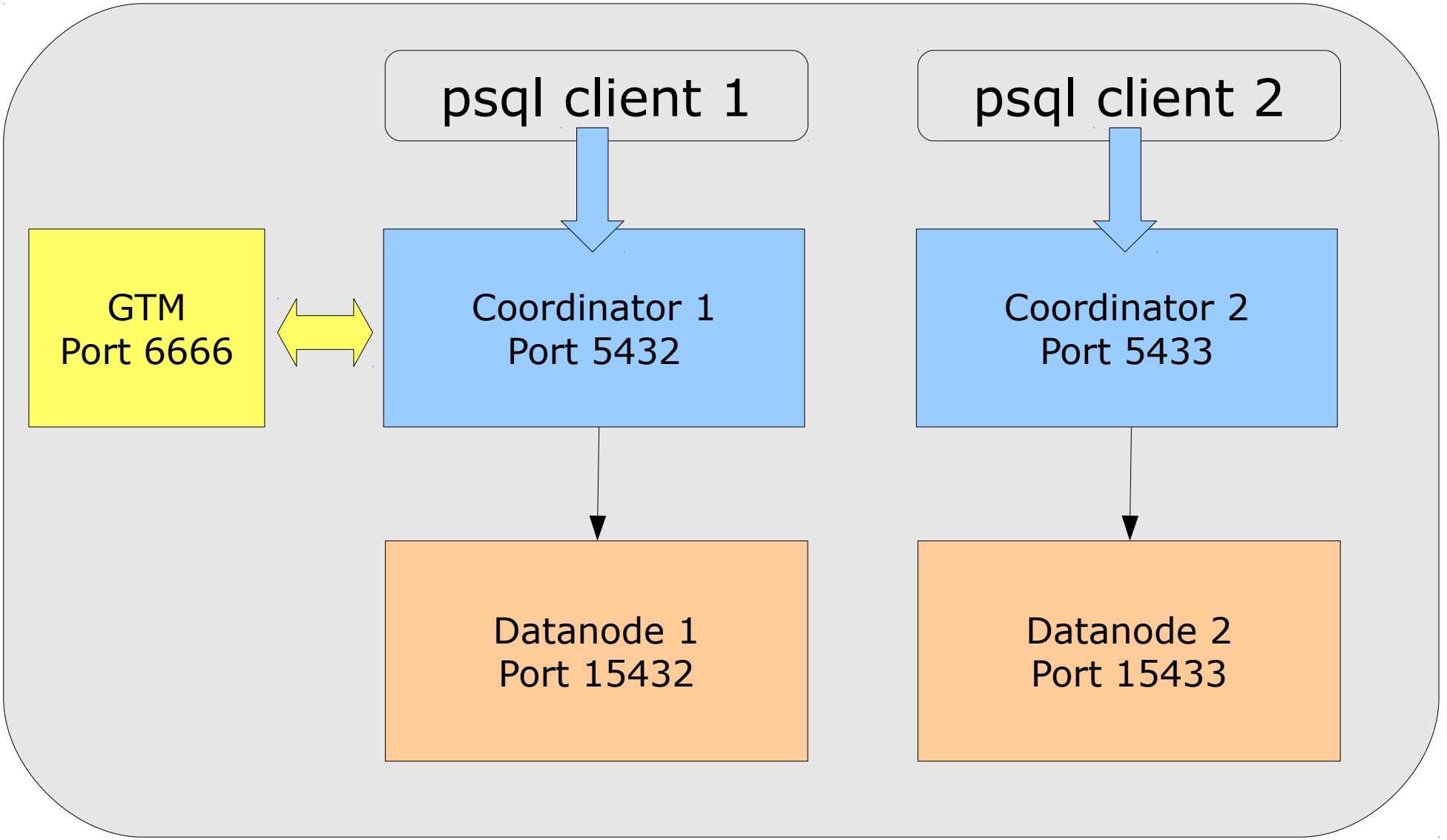


Query algorithm (4)

- Materialize results if necessary and send back to client



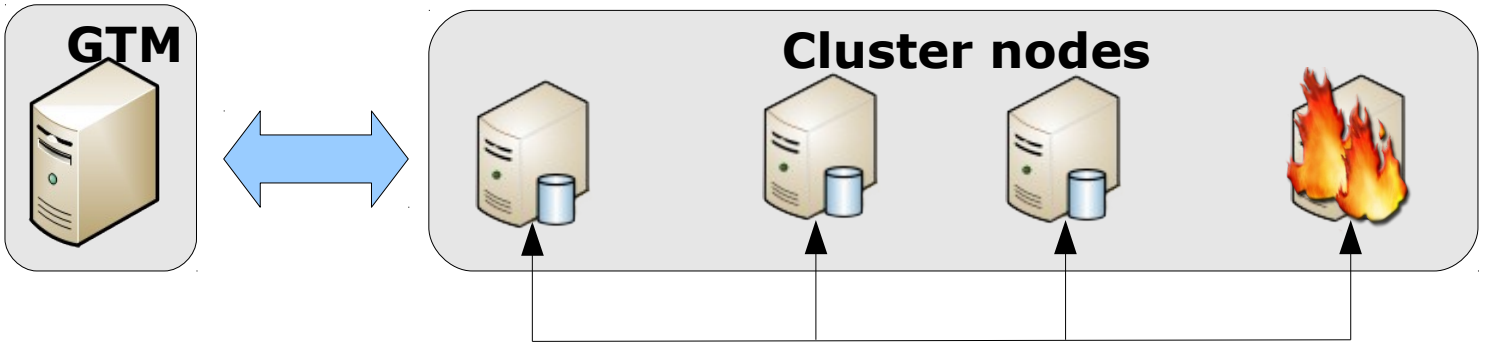
Demonstration



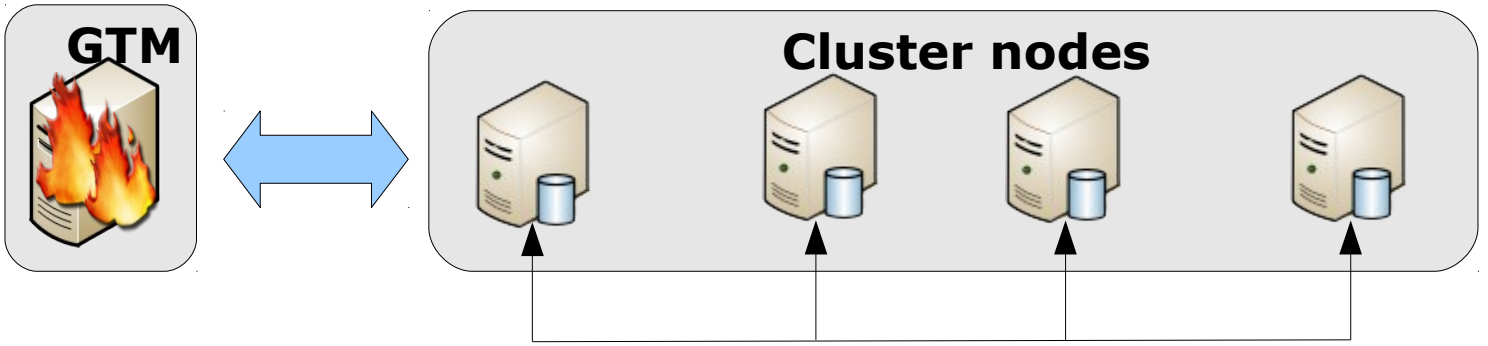
About high-availability

Cluster SPOF problem

- Datanode is a SPOF if it has a portion of distributed table.

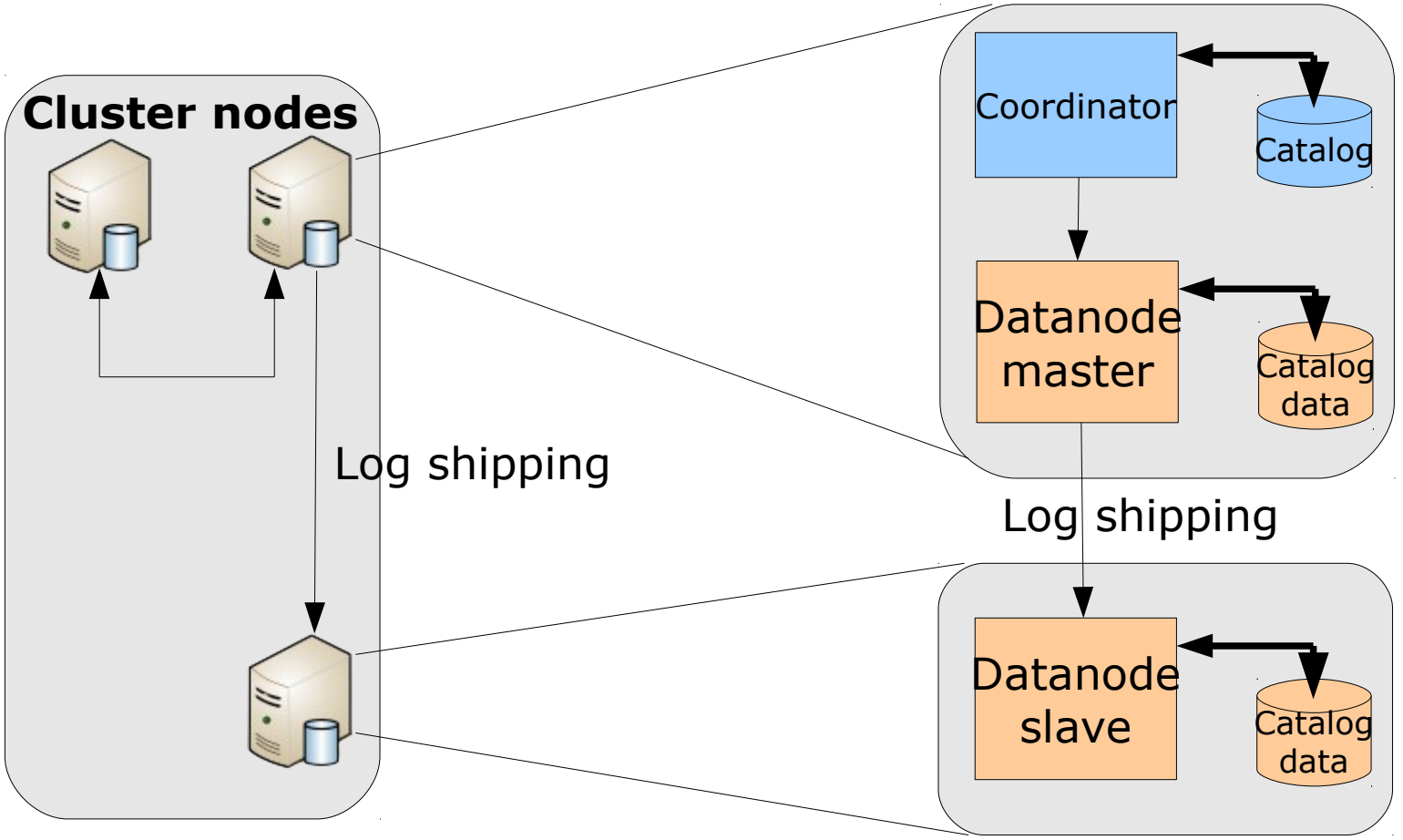


- GTM case



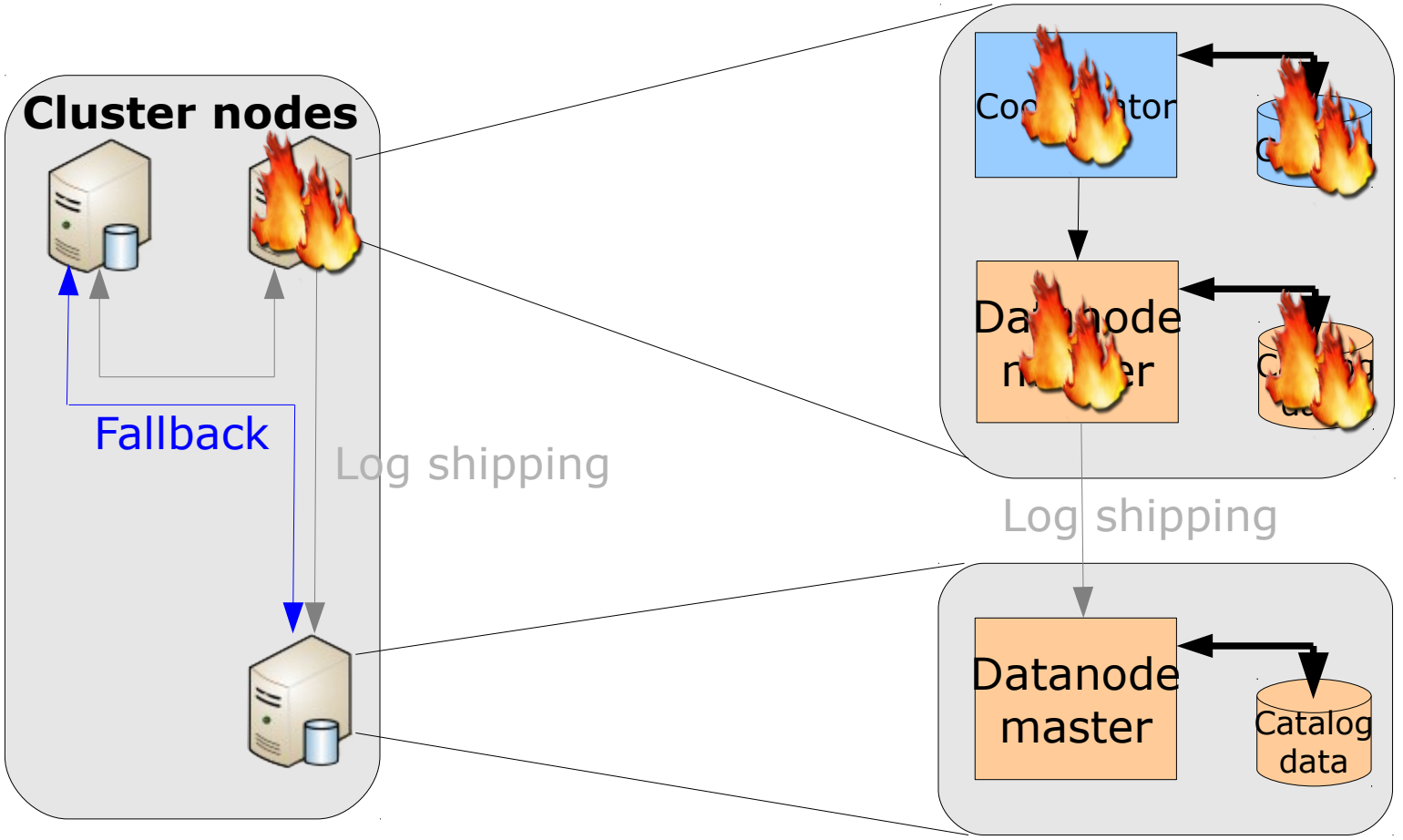
Datanode SPOF resolution (1)

- PostgreSQL 9.1 synchronous strrep



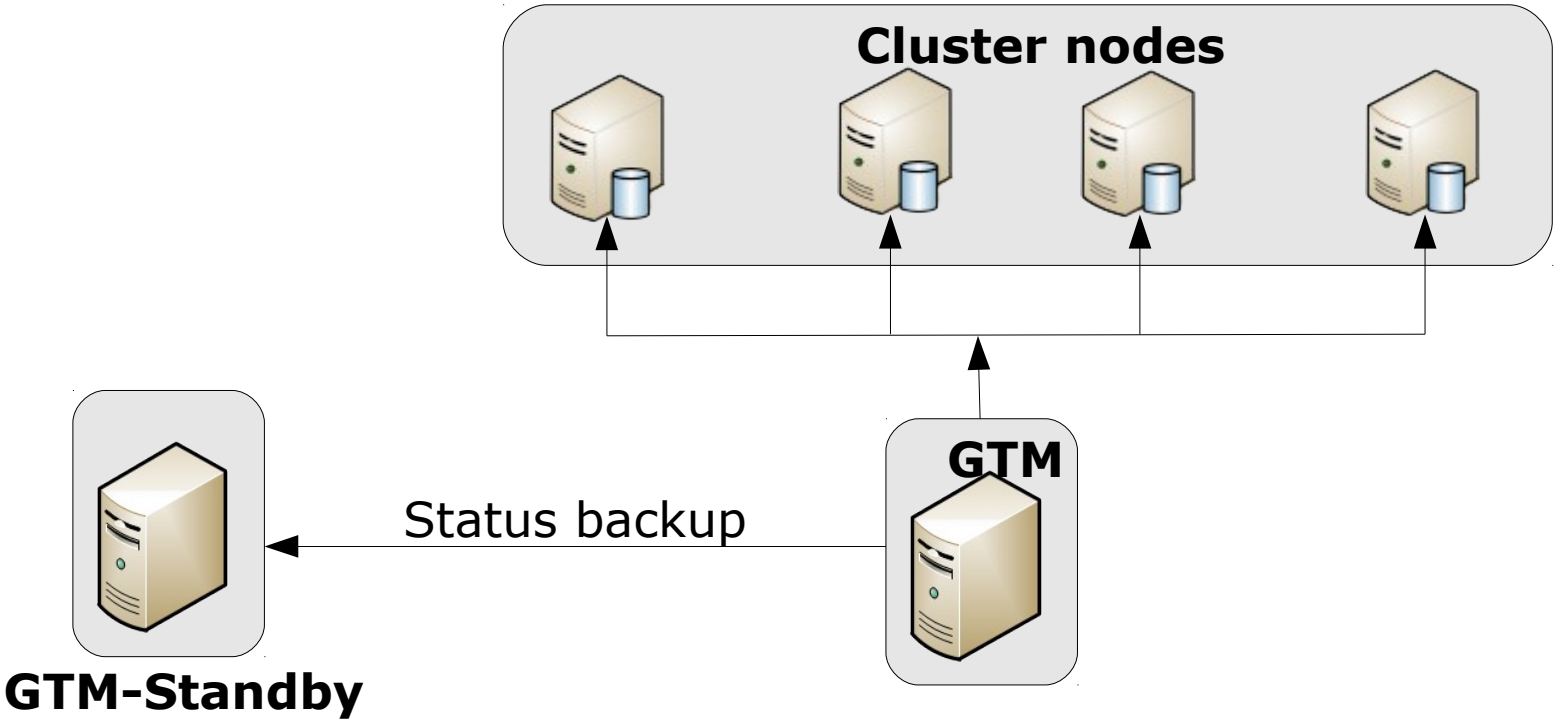
Datanode SPOF resolution (2)

- Fallback slave node



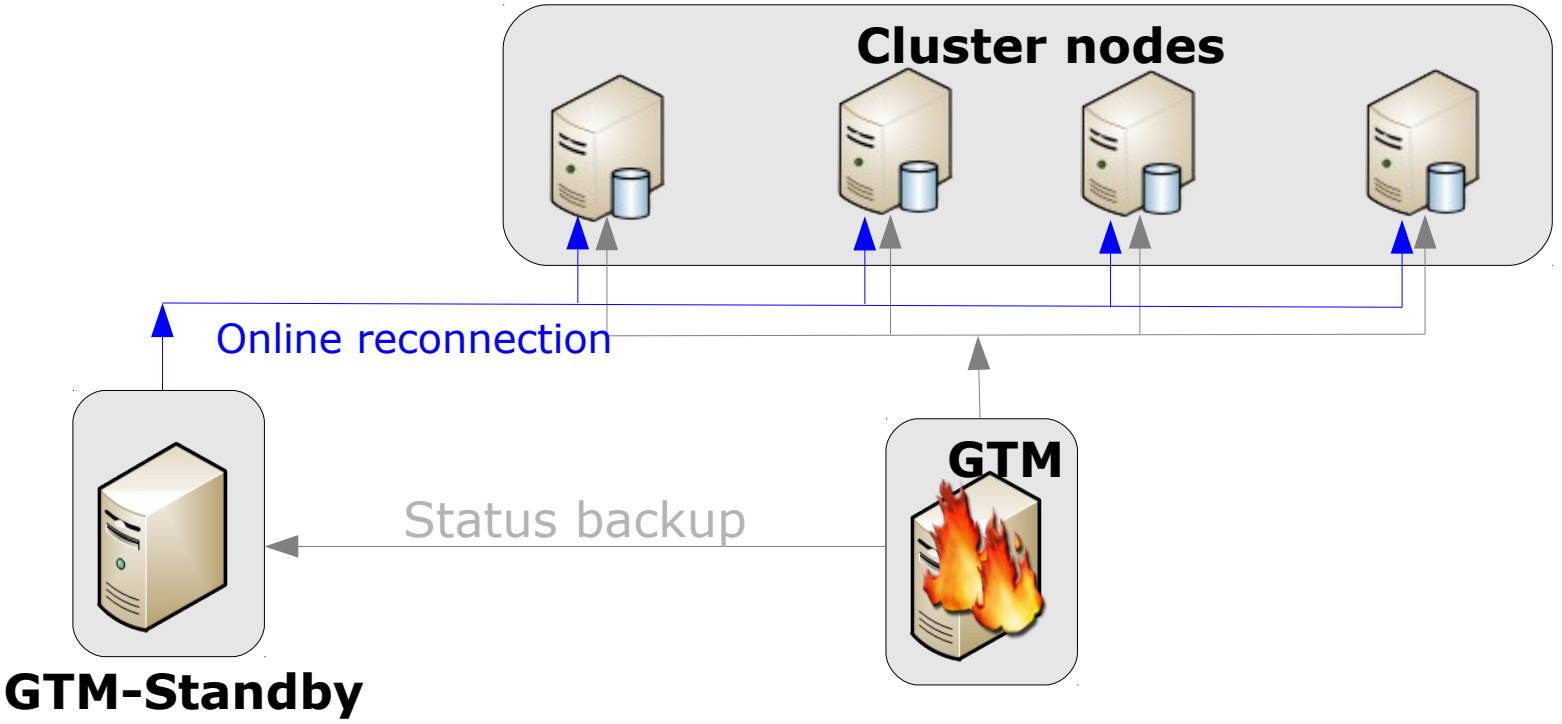
GTM SPOF resolution (1)

- Use of a standby node for GTM



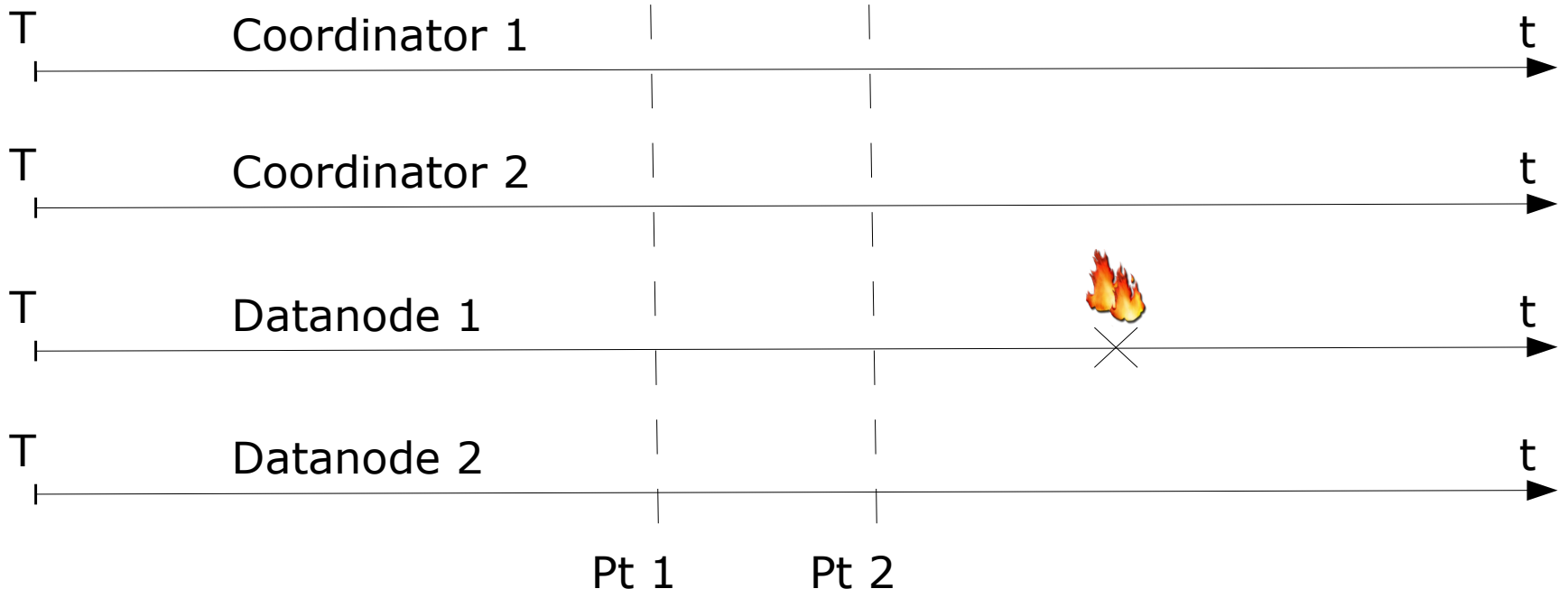
GTM SPOF resolution (2)

- Fallback to standby and reconnect nodes



PITR – requirements (1)

- PITR, Point in-time recovery
  - Rollback the database to a given past state
  - Need consistent points to restore to that



PITR – requirements (2)

- Transaction status has to be consistent in the cluster
- Each transaction must be either:
  - Committed/Prepared/Aborted/Running on all the involved nodes
  - We must avoid cases where transaction is prepared and committed partially, or prepared and rollbacked partially
- Write record in WALs of all the coordinators and datanodes at a moment when all the transaction statuses are consistents.
- External Application can provide such timing as with BARRIER
  - *CREATE BARRIER barrier_id*
- BARRIER:
  - Waits that partially committed or aborted transactions commit (2PC)
  - Block all transaction commit when running a barrier
- When running PITR, specify recovery_target_barrier in recovery.conf

What now and next?

Current functionalities

- Up to 0.9.5
  - Major DDL/DML (TABLE, ROLE, VIEW...)
  - PREPARE/EXECUTE (restrictions on parameters)
  - Session parameters
  - Support for additional distribution types
  - Cursors (no backward, no CURRENT OF)
  - SELECT queries: support extension
    - HAVING, GROUP BY, ORDER BY, LIMIT, OFFSET...

About release 0.9.6

- Release on September 2011
- Temporary objects
- Merge with PostgreSQL 9.1
- PREPARE/EXECUTE (all except CREATE AS)
- HAVING
- Aggregate generally operational
- Regressions updated and maintained consistent

Heading to 1.0 (1)

- SQL support
  - Subqueries (WITH)
  - Generic treatment of function SPI for data distribution (Ex: SERIAL)
  - CREATE AS/SELECT INTO
  - Trigger, rules
  - CURRENT OF, SAVEPOINT
  - TABLESPACE extension (case of multiple Datanodes on same server...)

Heading to 1.0 (2)

- Related to tuple relocation
  - Move tuples from a node to another node
    - Ex: update of a distribution column
  - CREATE/ALTER TABLE to choose list of nodes where a table is distributed (all nodes by default)
- Connection balancing between master and slave Datanodes for read transactions.
- Management of node information with SQL interface => configuration simplified

Challenges after 1.0?

- Global constraints
  - Unique/Reference integrity among partition
  - Exclusion constraint among partition
- Global deadlock detection (wait-for-graph mechanism)
- Online server removal/addition
- SQL/MED mechanisms, FDW integration

What can be done for PostgreSQL?

- Snapshot cloning
  - Several sessions holding the same snapshot
  - parallel pg_dump
- Parallel query execution
- SQL/MED improvements
  - Column projection
  - Join pushdown, ORDER BY, GROUP BY, aggregates
  - Foreign expression pushdown (function stable/volatile/immutable, etc.)
- Materialization of external node
- Cross-node join
- Cross-node aggregation

Project ressources and contacts

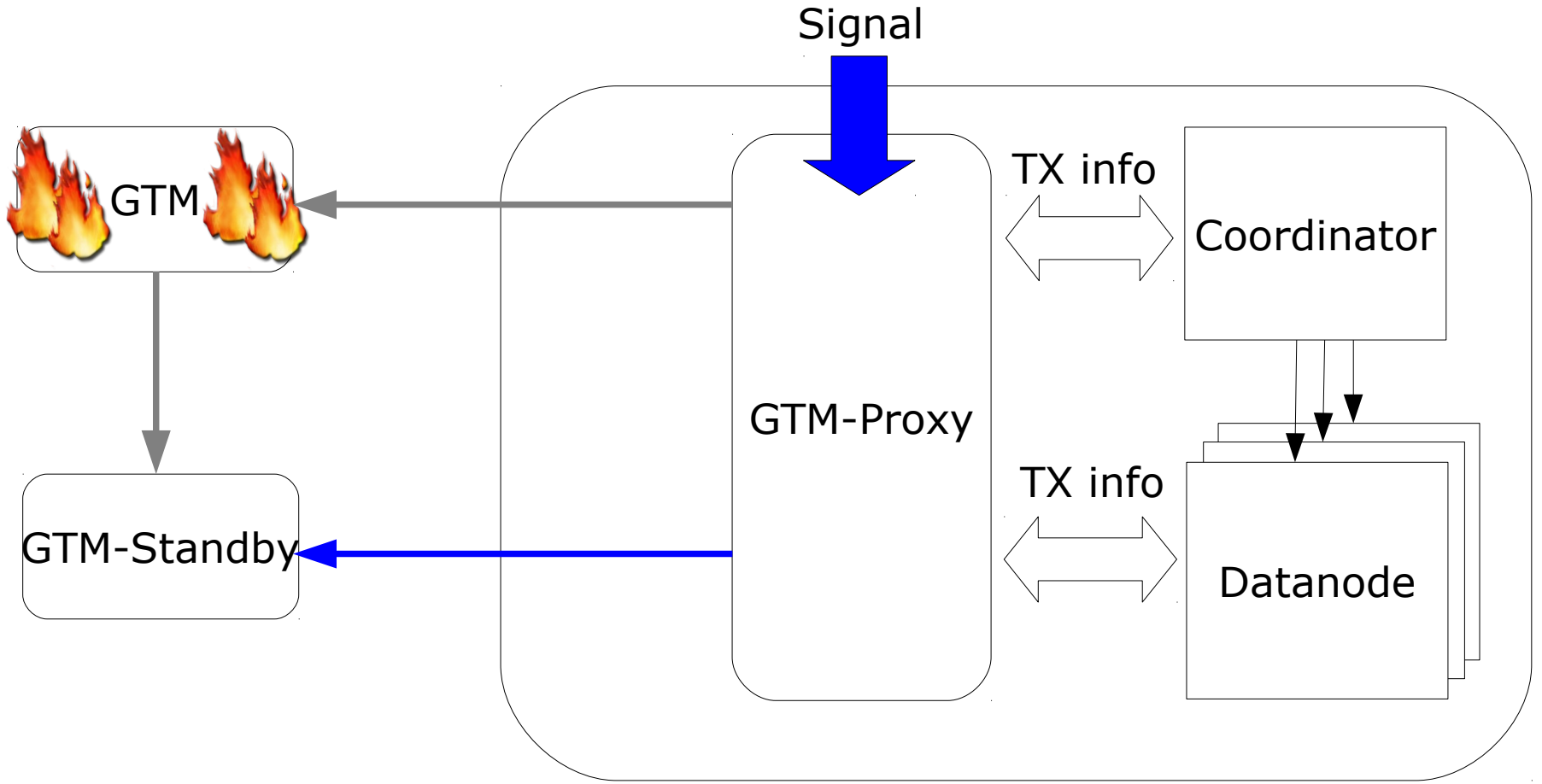
- Project home
  - http://postgres-xc.sourceforge.net
- Developer mailing list
  - postgres-xc-developers@lists.sourceforge.net
  - postgres-xc-general@lists.sourceforge.net
- Contacts
  - michael.paquier@gmail.com
  - koichi.szk@gmail.com
- Twitter: @michaelpq
- Blog: http://michael.otacoo.com

Sponsored and supported by:

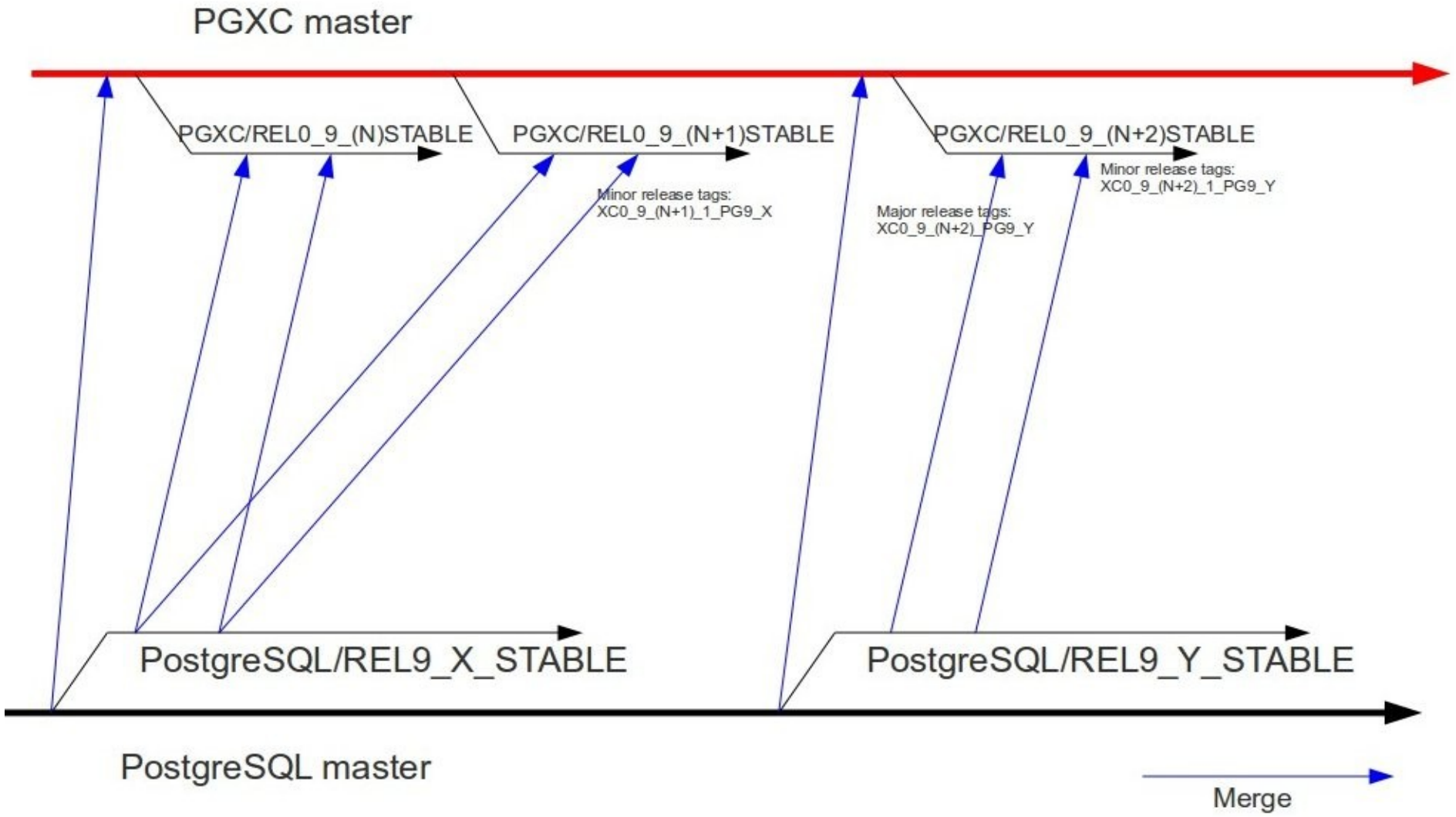
Thanks for your attention.
Questions?

GTM Proxy reconnection

- Signal GTM Proxy and reconnect nodes



Release policy



License

- This document is Copyright © 2011 by Michael Paquier. You may distribute it and/or modify it under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by-nc-sa/3.0/), version 3.0 or later.