EnterpriseDB®
The Enterprise PostgreSQL Company

Postgres-XC

Write-scalable, Synchronous Multi-master, Transparent PostgreSQL Cluster with Shared Nothing Approach

Presented by Ashutosh Bapat

November 22, 2011

@ OSI Days 2011

Agenda

- High-lights
- Architecture overview
- Performance
- Release and development processes
- Your contributions

Postgres-XC - in short

- Write scalable
  - Scalability by adding multiple servers
  - Each server capable of handling writes
- Synchronous multi-master
  - Multiple database servers that client can connect to
  - A single, consistent cluster-wide view of the database
  - Writes from any server are immediately visible to transactions on other servers
- Transparent
  - The applications do not have to worry about where data is stored
- Shared nothing cluster
  - Servers do not share any resources
  - Loosely coupled to large extent
  - Ease of deployment and scaling out by addition of commodity hardware

Postgres-XC - high-lights

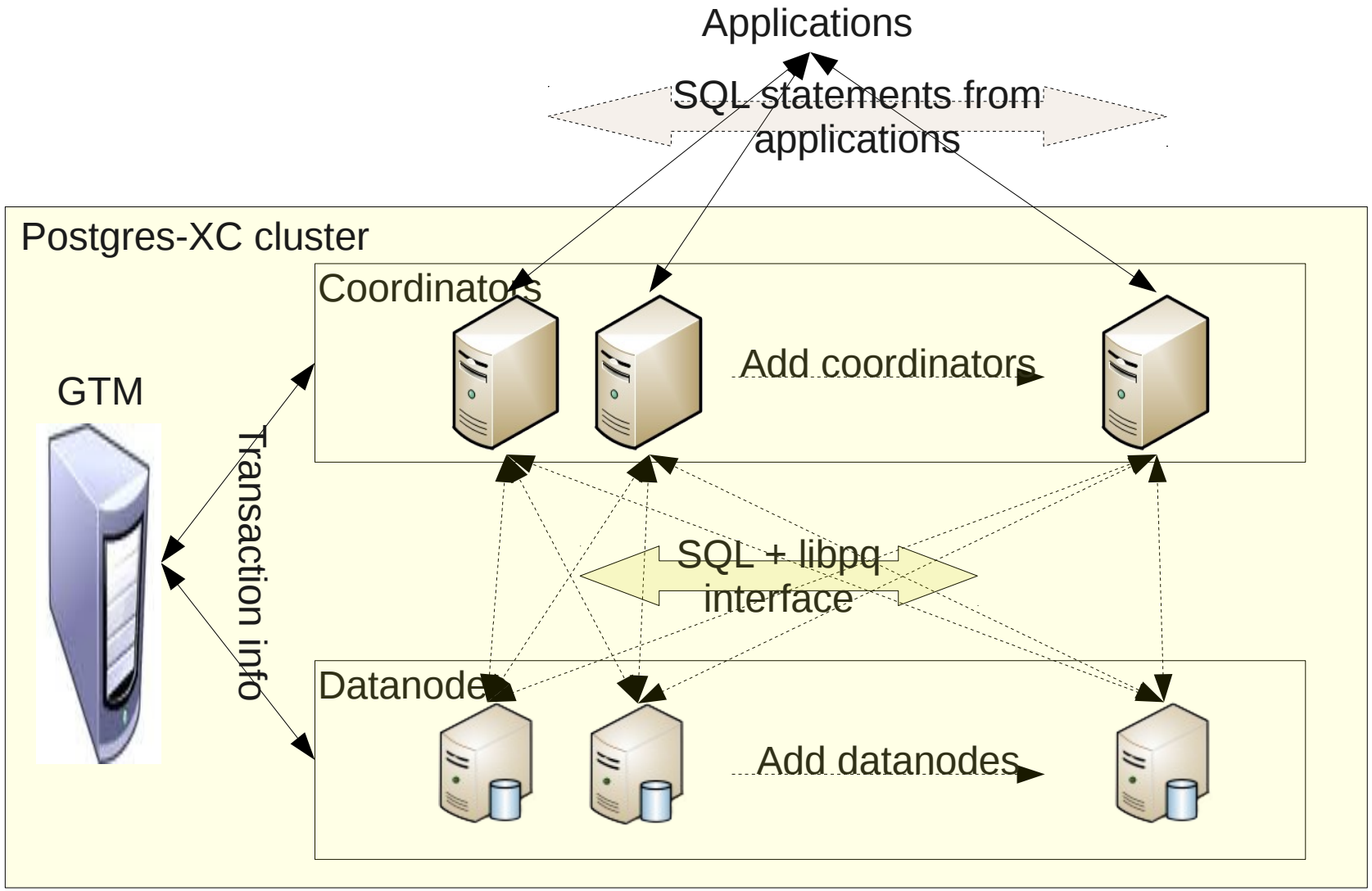
- Based on world's most advance open source database – PostgreSQL
- Same client APIs as PostgreSQL
  - Ease of application migration from existing PostgreSQL deployment
- Licensing – same as PostgreSQL license
  - Free to use, modify and redistribute for commercial purposes

Postgres-XC - history

- Started through a collaboration between EnterpriseDB and NTT Open Source Software Center in 2009
- Mandate to build a PostgreSQL based clustering solution which can serve as an alternative to Oracle RAC
- Very well represented by NTT in terms of engineers, hardware and funding
- EnterpriseDB provides key technical resources and PostgreSQL expertise to the project
- Licensing terms changed from GPL to BSD (or PostgreSQL) this year

Architecture Overview

Postgres-XC architecture



Global Transaction Manager (GTM)

- Gather and manage information about transaction activities in the cluster
- Issue global transaction identifiers to transactions and MVCC snapshots for a consistent view on all nodes.
- Help guarantee ACID properties
- Provide support for other global data such as sequences and time-stamps
- Store no data
  - Except some control information
- Separate binary from coordinator and datanodes

Postgres-XC node - Coordinator

- Point of contact for the application/client
- Parse and partially plan the statements
- Determine the data to be fetched from the datanodes and also location of the data
- Fetch the required data by issuing queries to the datanodes
- Combine and process the data to evaluate the results of the query (if needed)
- Pass the results to the applications
- Manage two-phase commit
- Store catalog data
- Need space for materializing results from datanodes
- Binary same as the datanode, based on the latest PostgreSQL release

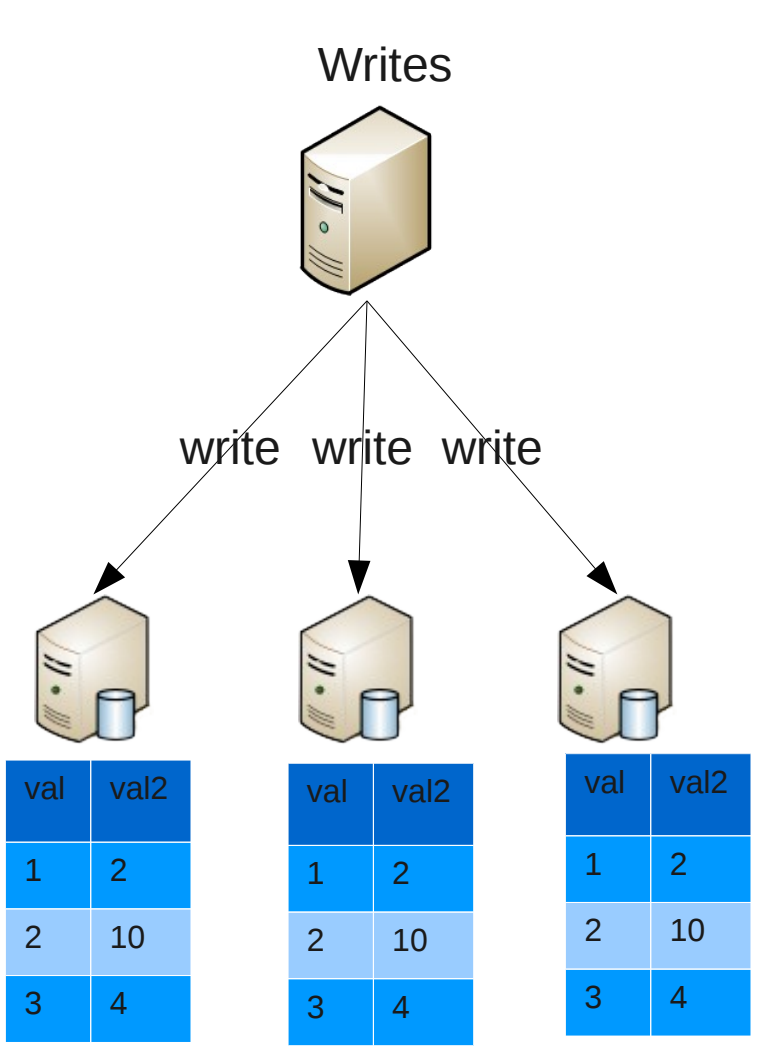
Postgres-XC node type - Datanode

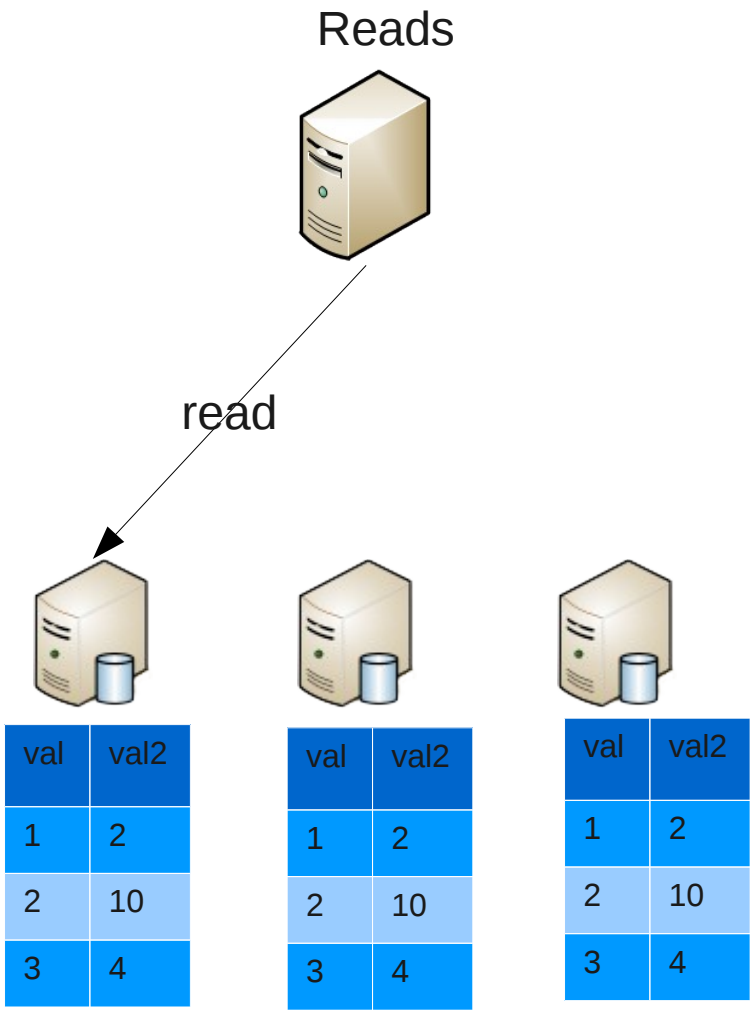
- Stores tables and catalogs
- Executes the queries from the coordinator and return results to the coordiantor
- Data nodes can be made fault tolerant by Hot-Standby and Synchronous Replication technologies available with standard PostgreSQL
- Binary same as coordinator, based on lastest PostgreSQL release

Data management

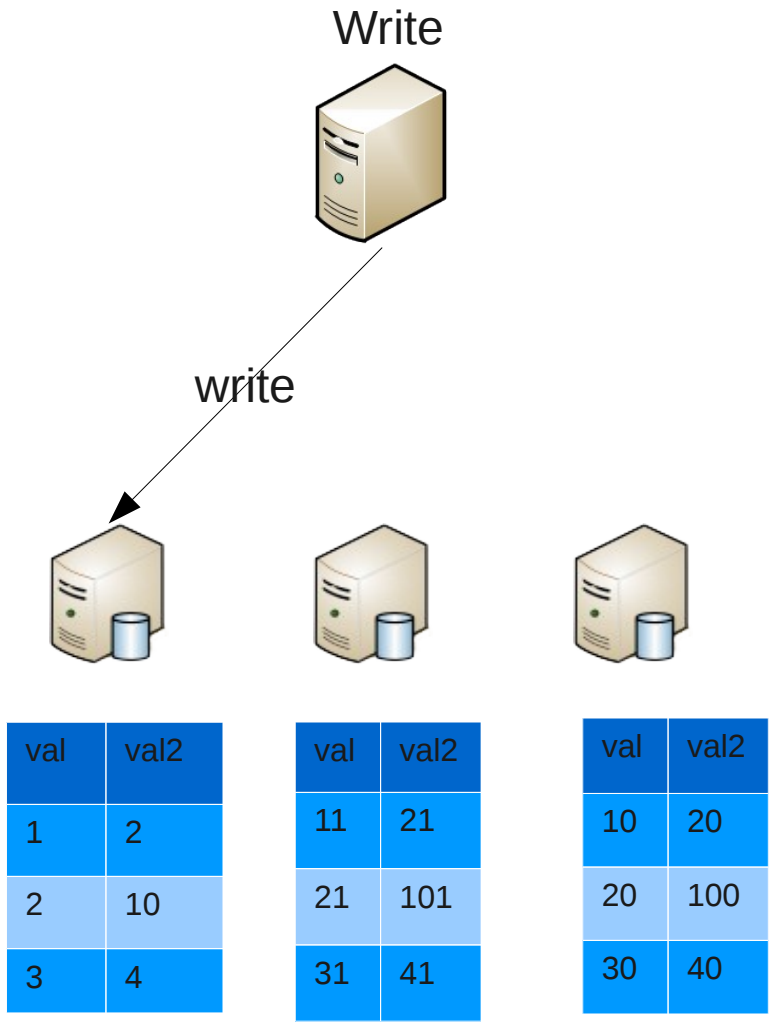
- A table is stored on datanode/s as
- Replicated table
  - Each row in the table is replicated to the datanodes
  - Statement based replication
- Distributed table
  - Each row of the table is stored on one datanode, decided by one of following strategies
    - Hash
    - Round Robin
    - Modulo
    - Range and user defined function – TBD

Postgres-XC - Replicated tables





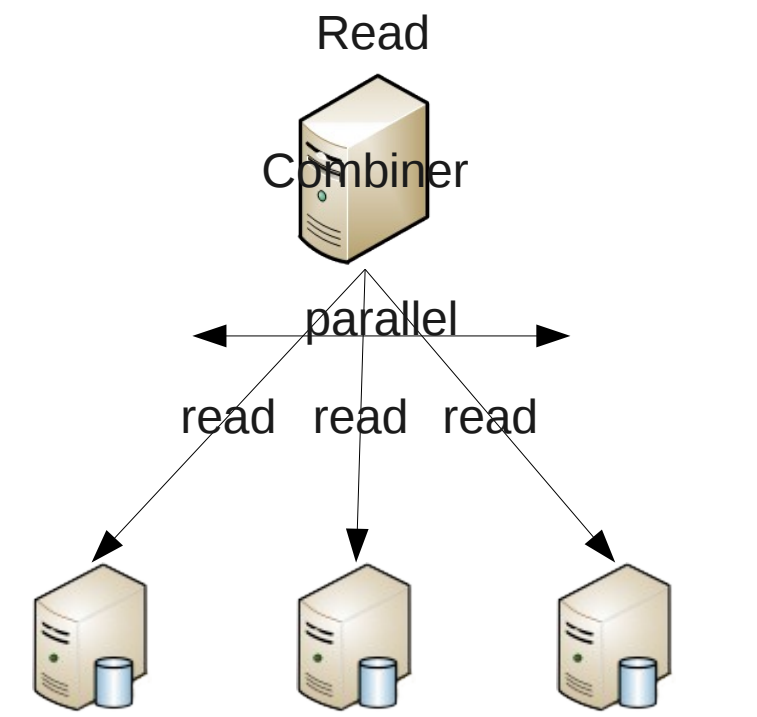
Postgres-XC - Distributed tables



val	val2
1	2
2	10
3	4

val	val2
11	21
21	101
31	41

val	val2
10	20
20	100
30	40

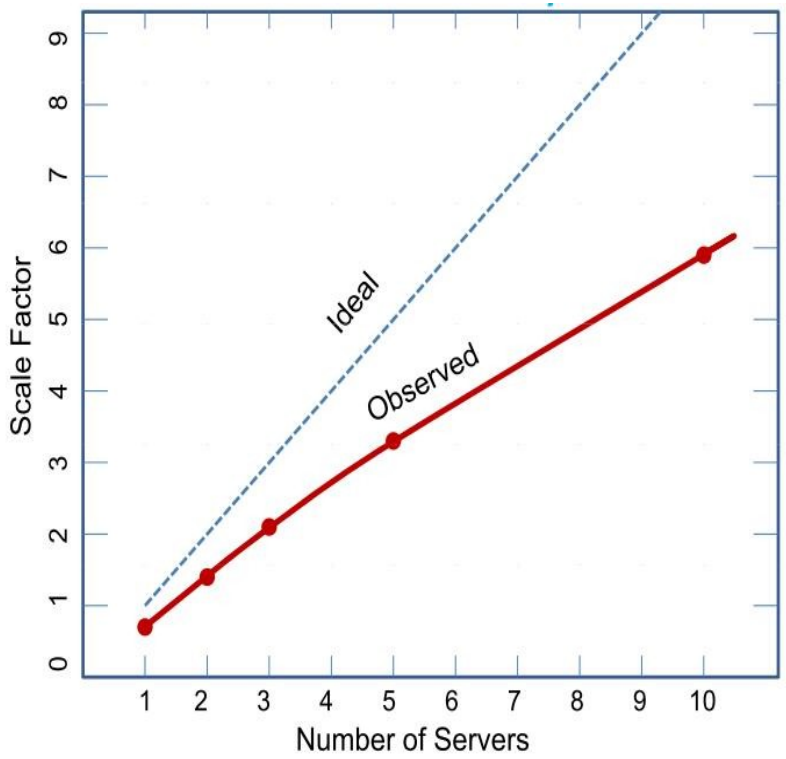


val	val2
1	2
2	10
3	4

val	val2
11	21
21	101
31	41

val	val2
10	20
20	100
30	40

Evaluation



- Conducted by NTT Intellilink
- DBT-1 (TPC-W) benchmark with some minor modification to the schema
- 1 server = 1 coordinator + 1 datanode on same machine
- Coordinator is CPU bound
- Datanode is I/O bound

Release management

Postgres-XC - Development and Release process

- Primary contribution from NTT/EnterpriseDB with at least 4 full time engineers and 2 part time engineers working on the project
- Open source model where many issues are discussed on the public mailing lists
- Increased interests from other community developers
- One release almost every 3-4 months to get new PGXC features to the users as soon as possible
- Keep pace with the PostgreSQL development to benefit from the new features in PostgreSQL
- GIT repository ensures a clean and quick merge process

Upto V 0.9.6

- Based on PostgreSQL 9.1
- SQL support
  - Major DDL/DML (TABLE, ROLE, VIEW...)
  - PREPAREd statements
  - SELECT queries: support extension aggregates, HAVING, GROUP BY, ORDER BY, LIMIT, OFFSET...
    - No CTAS, INSERT SELECT, SELECT INTO (Being worked on)
  - Cursors
    - no backward, no CURRENT OF
- Session parameters
- Temporary objects

Expectations from V 1.0

- SQL support
  - Subqueries (WITH)
  - CREATE AS/SELECT INTO
  - Trigger, rules
  - CURRENT OF
  - TABLESPACE extension (case of multiple Datanodes on same server...)
  - Concurrent index creation
- Changing distribution strategy
  - Distribution column, nodes or type

After V 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
- Connection balancing between master and slave Datanodes for read transactions
- SAVEPOINT

Project resources and contacts

- Project home
  - http://postgres-xc.sourceforge.net
- Developer mailing list
  - postgres-xc-developers@lists.sourceforge.net
  - postgres-xc-general@lists.sourceforge.net
- Contact me
  - ashutosh.bapat@enterprisedb.com
  - EnterpriseDB, the PostgreSQL company

Resources needed for

- Writing Code
  - Backend - New distributed/parallel query handling/optimization, HA capabilities, Utilities, Bug fixes
  - Installers, building binaries, distribution packages
- Test
  - Performance evaluation with various benchmarks
  - Finding bugs
  - New feature proposals
- Deployment
  - Running practical applications against Postgres-XC

Thank you

Installers

- pgxc_configure provides installation capability and is provided as separate resource.
- Need more improvement and to do this

Why DBT-1?

- NTT used this as one of the standard benchmark to show the performance of open source software.
- Evaluation team is now testing TPC-C.
- TPC-W (DBT-1) consumes CPU much more than TPC-2 and considering that XC consumes much CPU at coordinator, TPC-W might show better scalability than TPC-C.