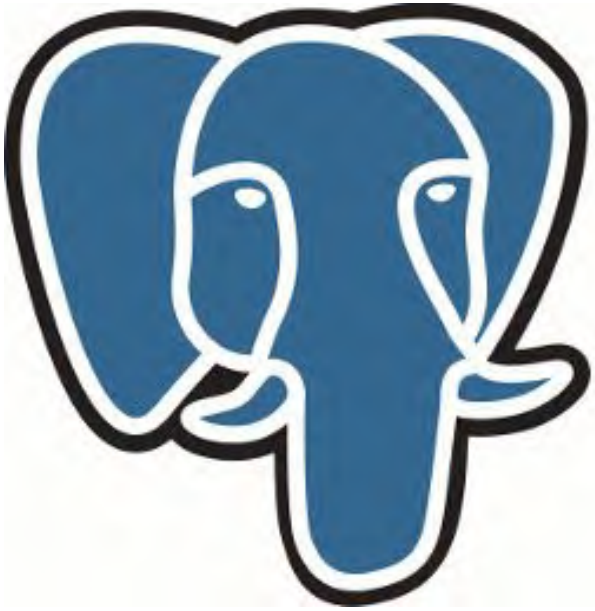
Postgres-X2介绍

李元佳

自我介绍

目录



PostgreSQL

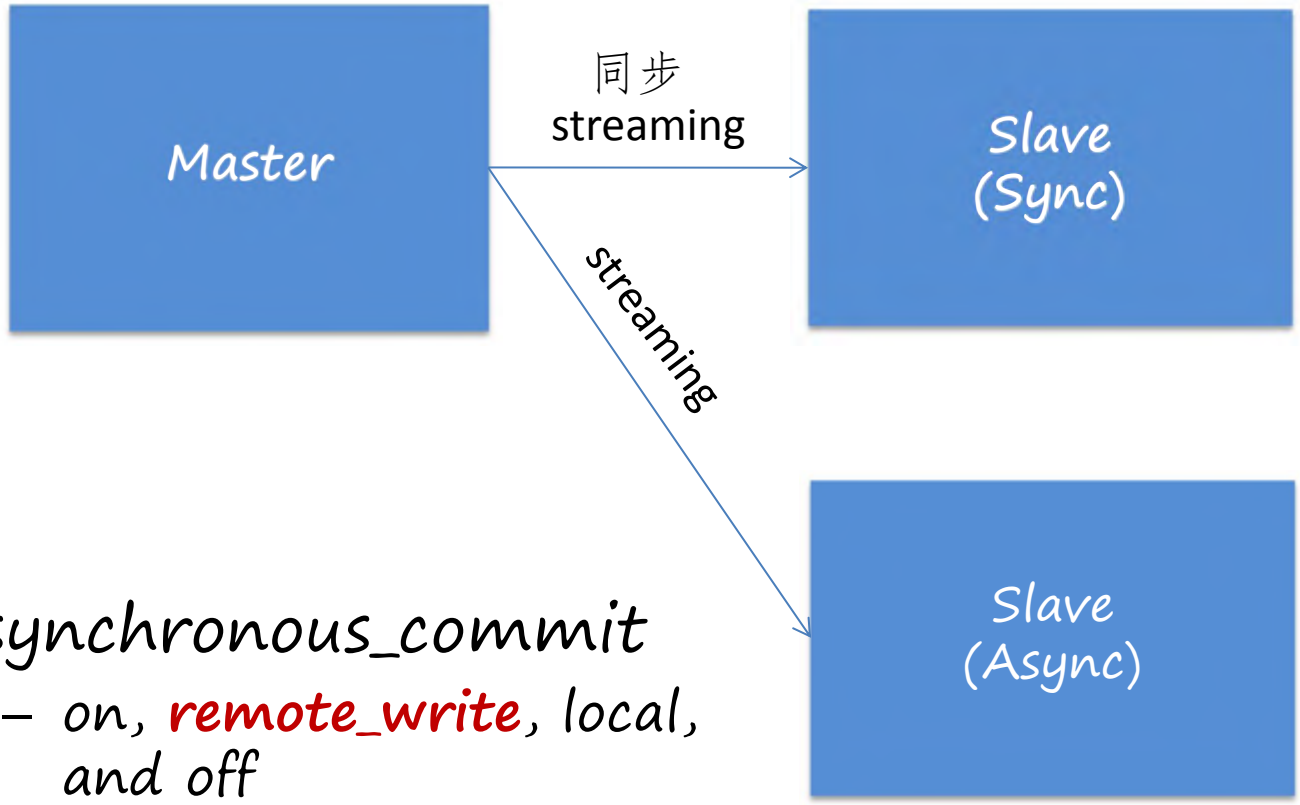
the world's most advanced open source database

Postgres的简介

- 开源的RDBMS
- 功能丰富
  - 完整SQL、事务、存储过程、同步复制
- 企业应用领域比较多
- 国内案例
  - 去哪儿、平安科技、国家电网等


Postgres 2015 中国用户大会（大象会）
2015年11月20日-21日 中国.北京
Postgres 2015中国用户大会
Postgres Conference China 2015
Postgres 2015中国用户大会

流复制

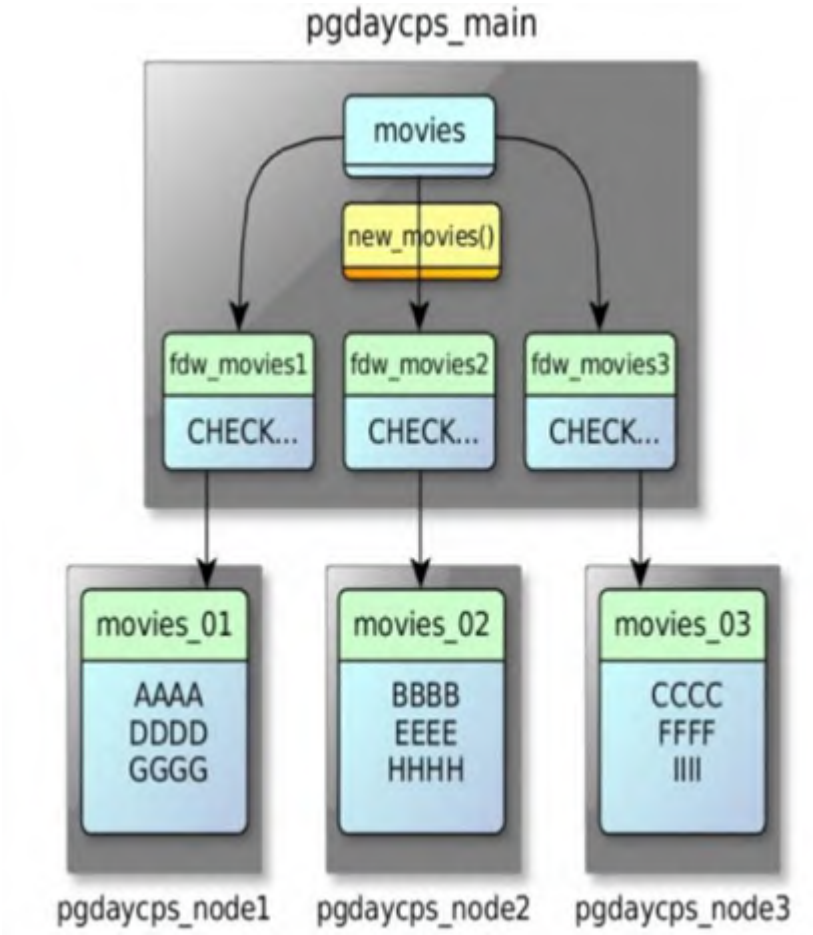


- synchronous_commit
  - on, **remote_write**, local, and off

数据零丢失+高性能

高可用问题已经解决

Sharding: postgres_fdw



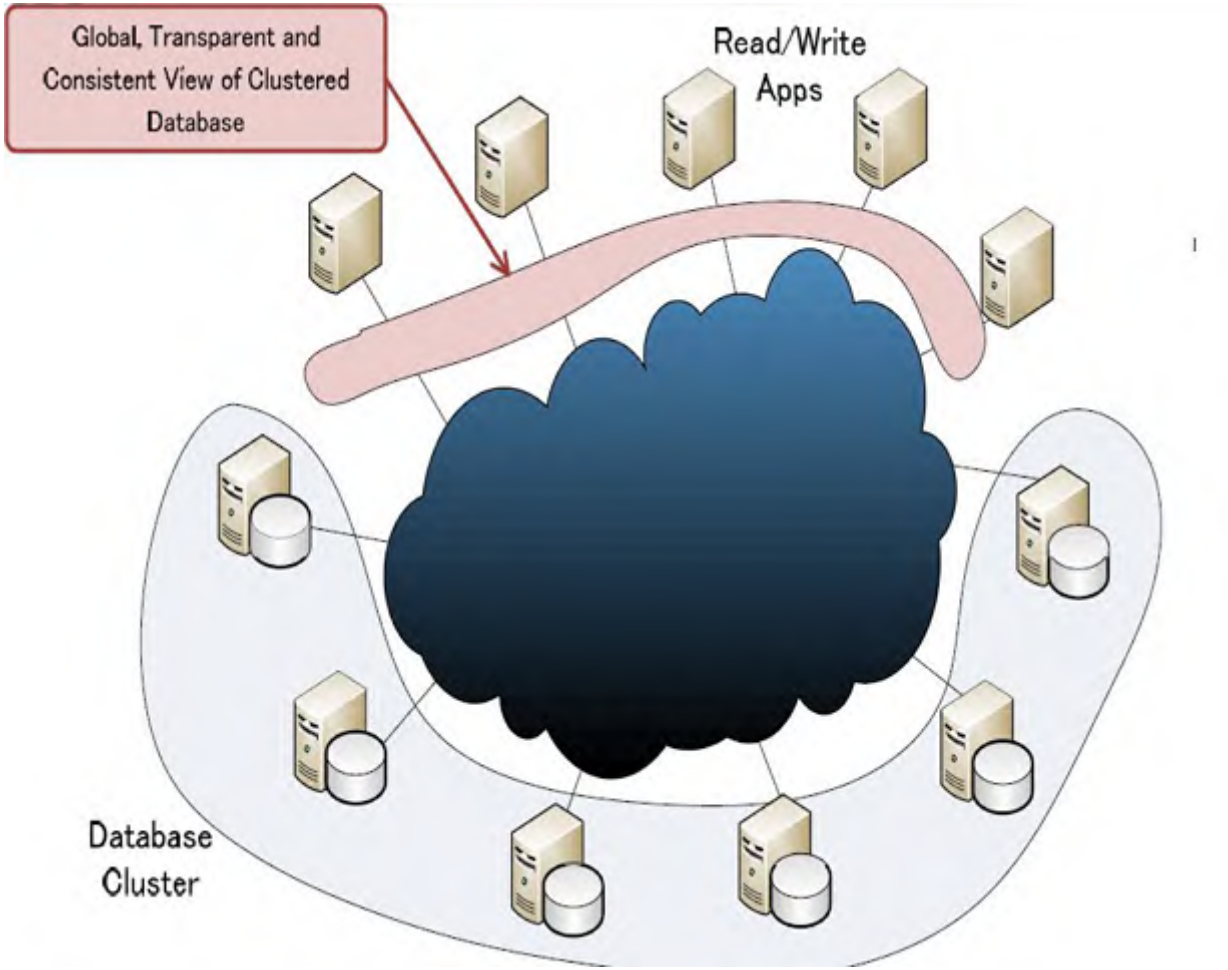
高扩展呢？

分布式数据库

- 面向大数据量、高并发的OLTP场景
- 多主多读、横向扩展
- 全功能关系型数据库(ACID、SQL几乎无限制)

全功能 + 高扩展

Postgres-XC/XL



Postgres-XC/XL简介

- 开源
  - Postgres-XC采用Postgres协议(类似BSD协议)
  - Postgres-XL以前是Mozzila协议，目前已经改为PostgreSQL协议
- 面向OLTP及OLAP场景
- 采用Share-Nothing架构、弹性扩展
- 基于Postgres改造、功能几乎完全继承

社区发展历史

- 2004~2008 NTT Data构建了模型Rita-DB
- 2009年 NTT Data与EnterpriseDB合作进行社区化开发
- 2012, Postgres-XC 1.0正式发布
- 2012, StormDB在XC基础上增加MPP功能.
- 2013, XC 1.1发布； TransLattice 收购 StormDB
- 2014, XC 1.2发布； StormDB 开源为 Postgres-XL.
- 2015, 两个社区合并为Postgres-X2

目录

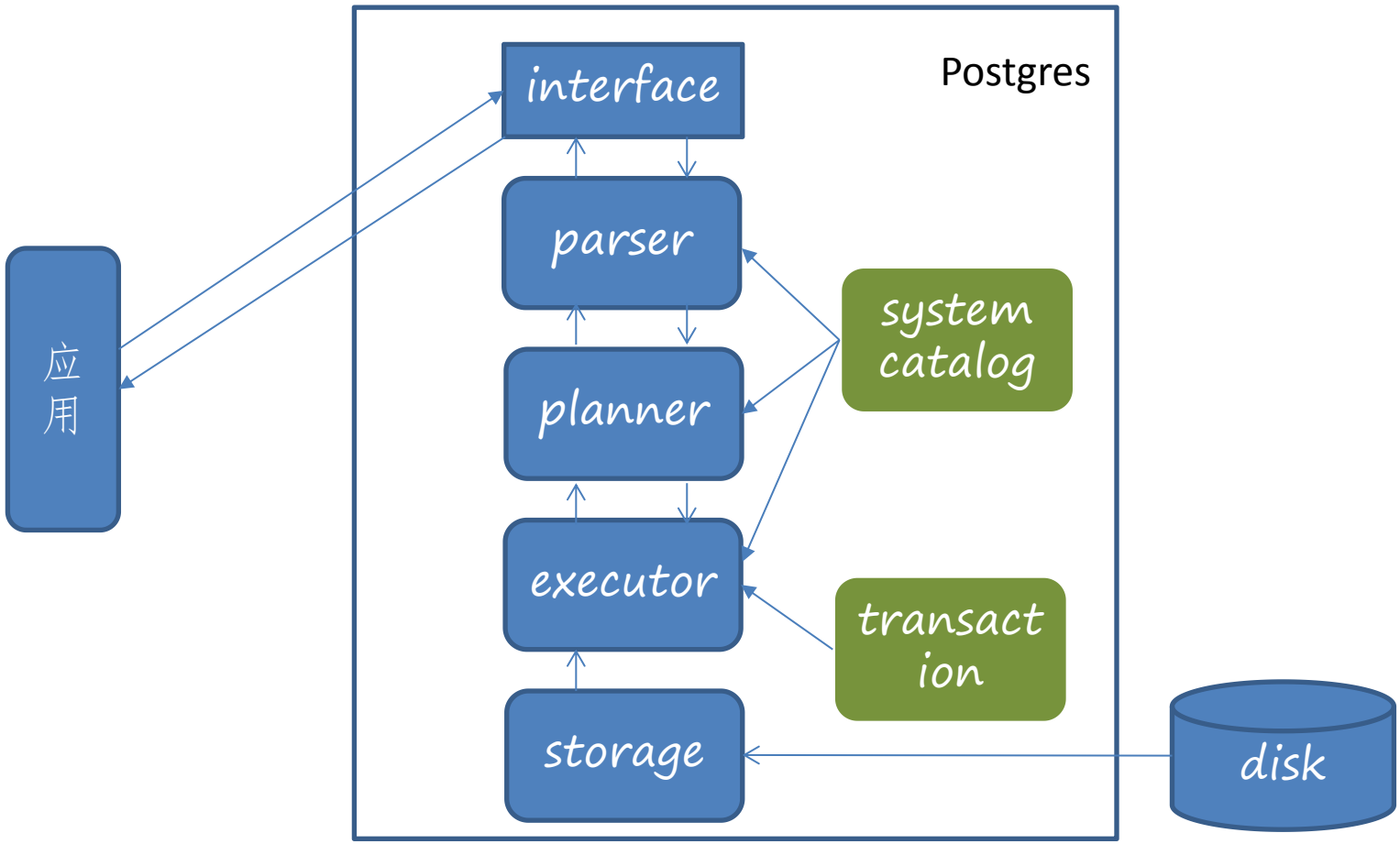
设计理念

- Share nothing 架构
- 功能解耦、分层扩展
- 数据分散在多个节点

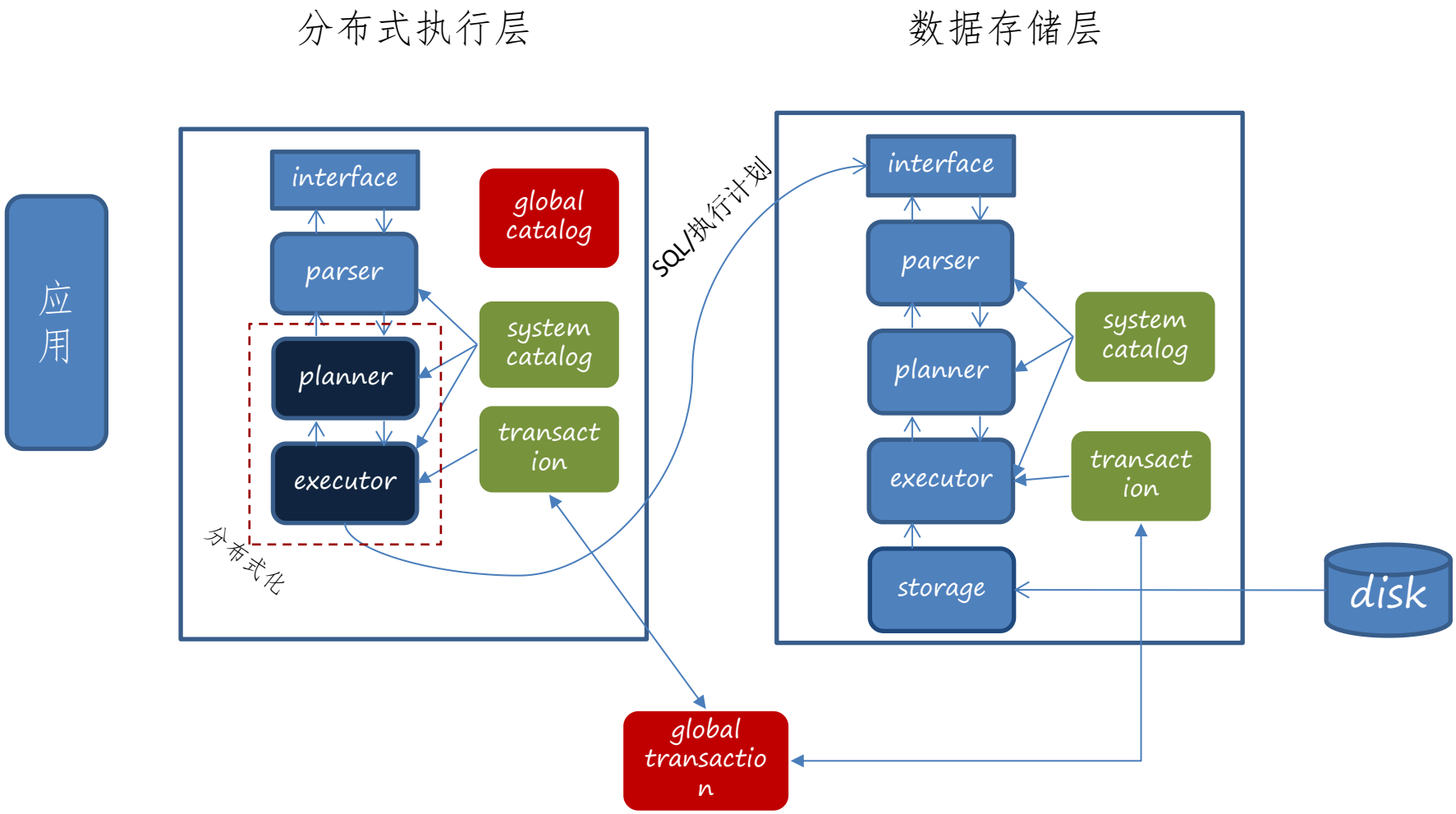
- 继承Postgres功能
- 继承Postgres生态
- SQL能力不受限制
- 支持存储过程

- MVCC
- 全局事务支持强一致

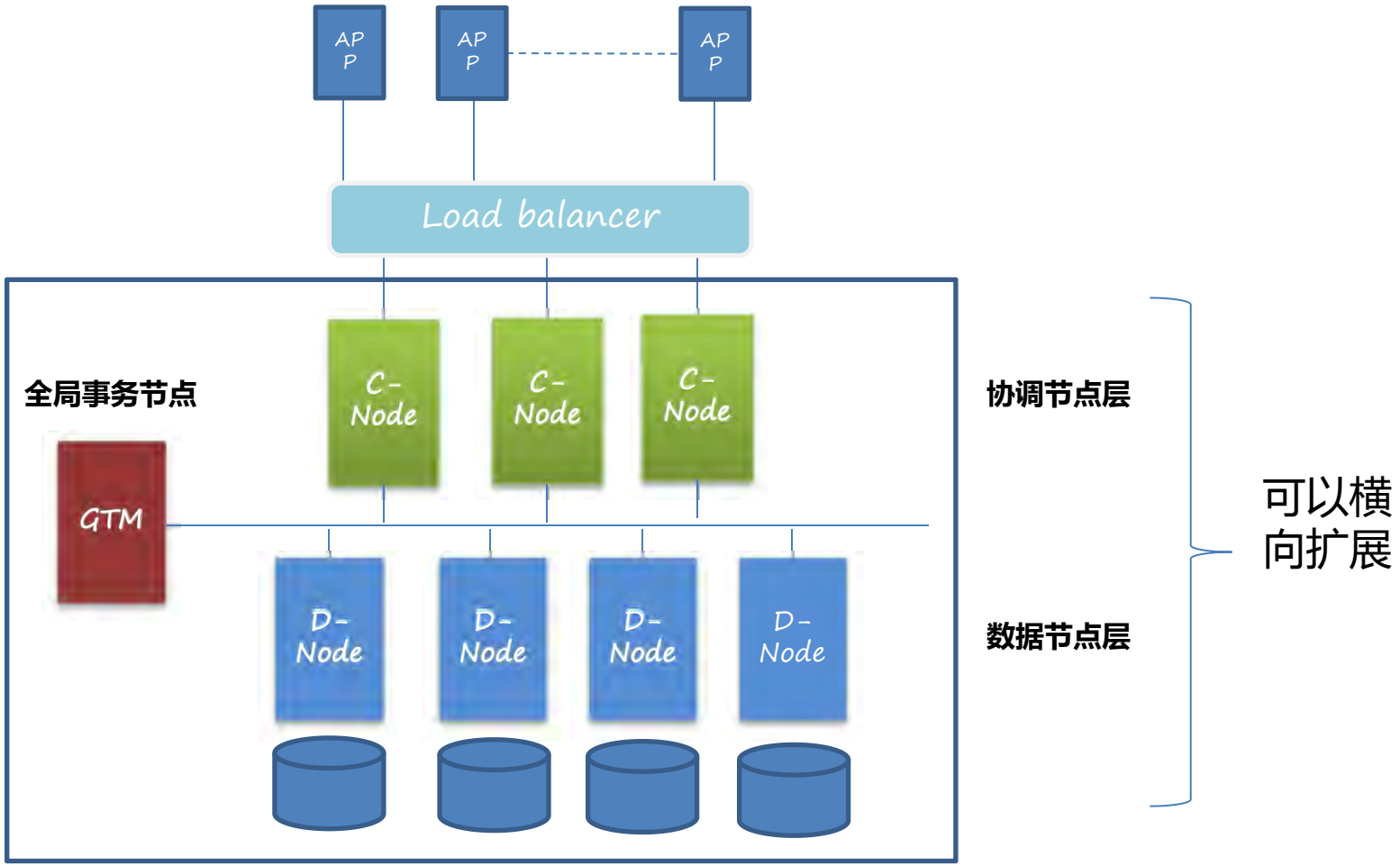
Postgres的架构



架构的解耦及分布式化


分布式执行层
数据存储层
应用
interface
parser
planner
executor
global catalog
system catalog
transact ion
分布式化
SQL/执行计划
interface
parser
planner
executor
storage
system catalog
transact ion
disk
global transactio n

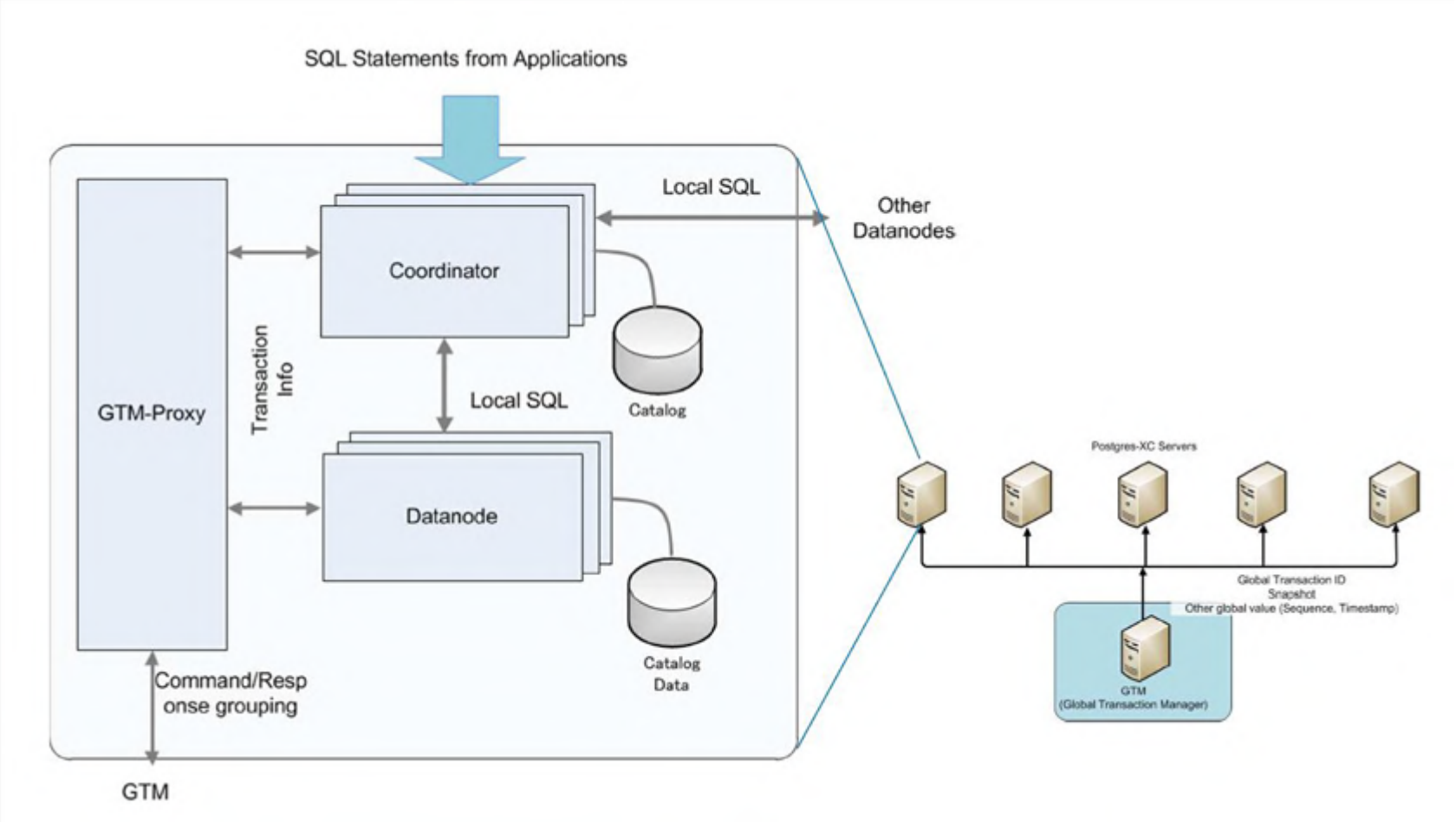
Postgres-XC的整体架构



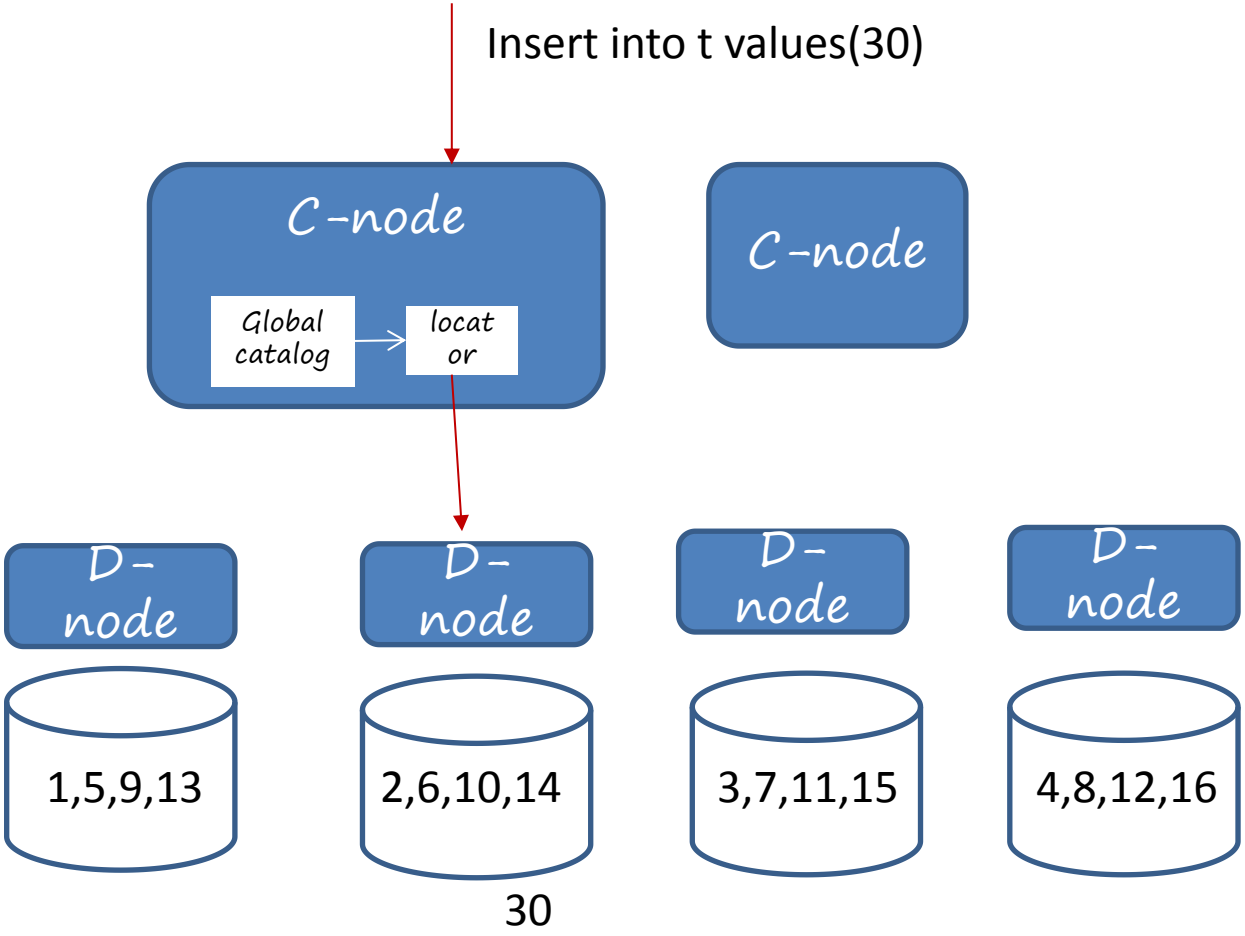
Postgres-XC主要的模块

- **Coordinator node：协调节点**
  - 负责接收用户请求、生成并执行分布式查询、把SQL语句发给相应的数据节点
- **Data node：数据节点**
  - 实际数据存储节点
- **GTM：全局事务节点**
  - 生成全局唯一的事务ID
  - 全局的事务的状态
  - 序列等全局信息

关键模块之间的关系


SQL Statements from Applications
Local SQL
Other Datanodes
Coordinator
Catalog
GTM-Proxy
Transaction Info
Local SQL
Datanode
Catalog Data
Command/Response grouping
GTM
Postgres-XC Servers
Global Transaction ID
Snapshot
Other global value (Sequence, Timestamp)
GTM
(Global Transaction Manager)

数据如何分布


Insert into t values(30)
C-node
Global catalog
locat or
C-node
D-node
D-node
D-node
D-node
1,5,9,13
2,6,10,14
3,7,11,15
4,8,12,16
30

数据分布方式

- replicated table 复制表
  - 表在多个节点复制
- distributed table 分布式表
  - Hash
  - Round robin
  - Range(未实现)
  - User define (未实现)

分布式查询优化

- Parallel query
- Where pushdown
- Join pushdown
- Expression pushdown
- Order by
- Two phase Aggregate
- Fast query shipping

单一节点处理

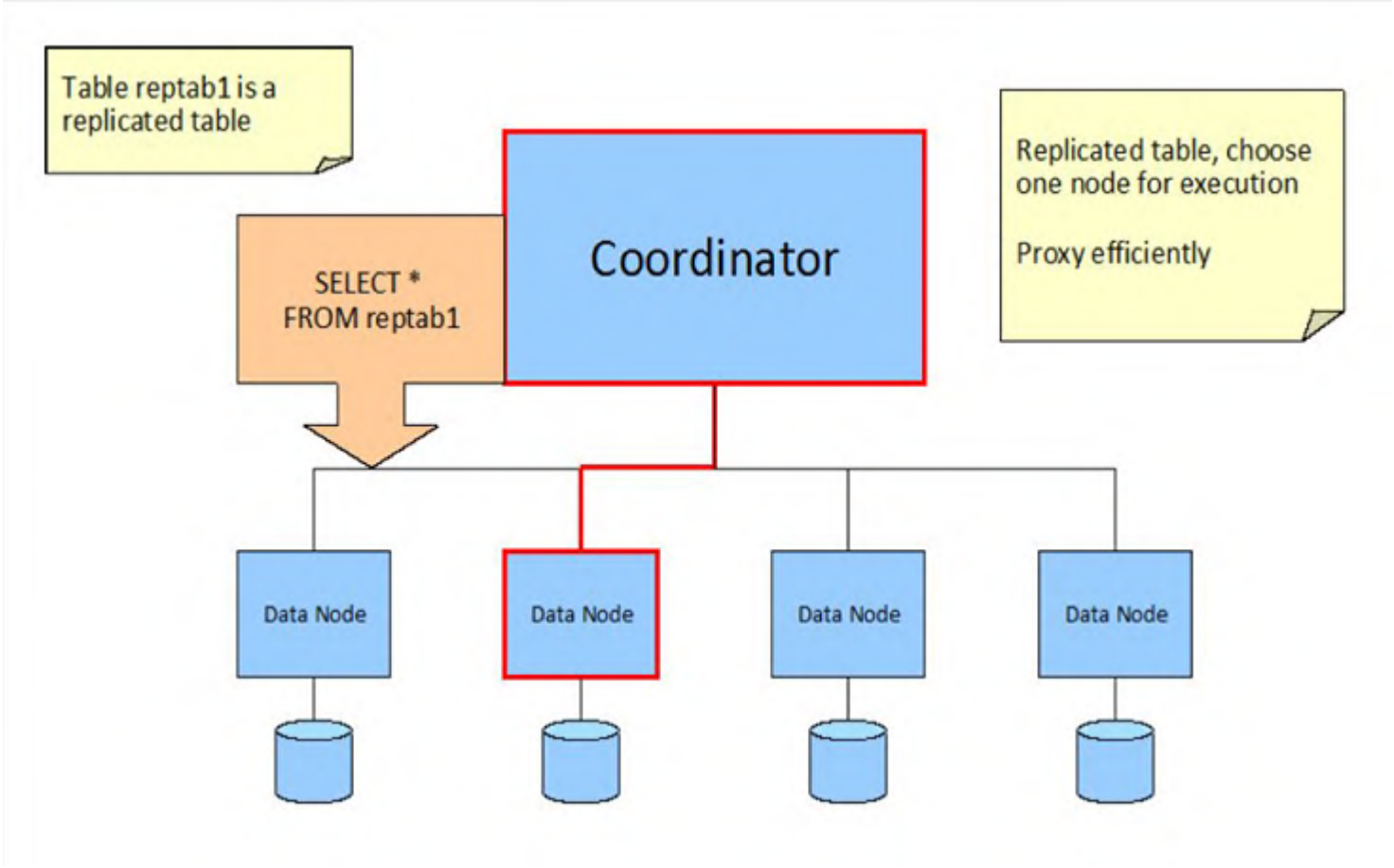


Table reptab1 is a replicated table
SELECT * FROM reptab1
Coordinator
Replicated table, choose one node for execution
Proxy efficiently
Data Node
Data Node
Data Node
Data Node

单一节点处理

```
EXPLAIN VERBOSE DELETE FROM test WHERE a = 100;
                                QUERY PLAN
---------------------------------------------------------------------------
 Data Node Scan on "__REMOTE_FQS_QUERY__"  (cost=0.00..0.00 rows=0 width=0)
   Output: test.a, test.ctid, xc_node_id
   Node/s: node_dn2
   Remote query: DELETE FROM public.test WHERE (a = 100)
(4 rows)
```

多节点处理

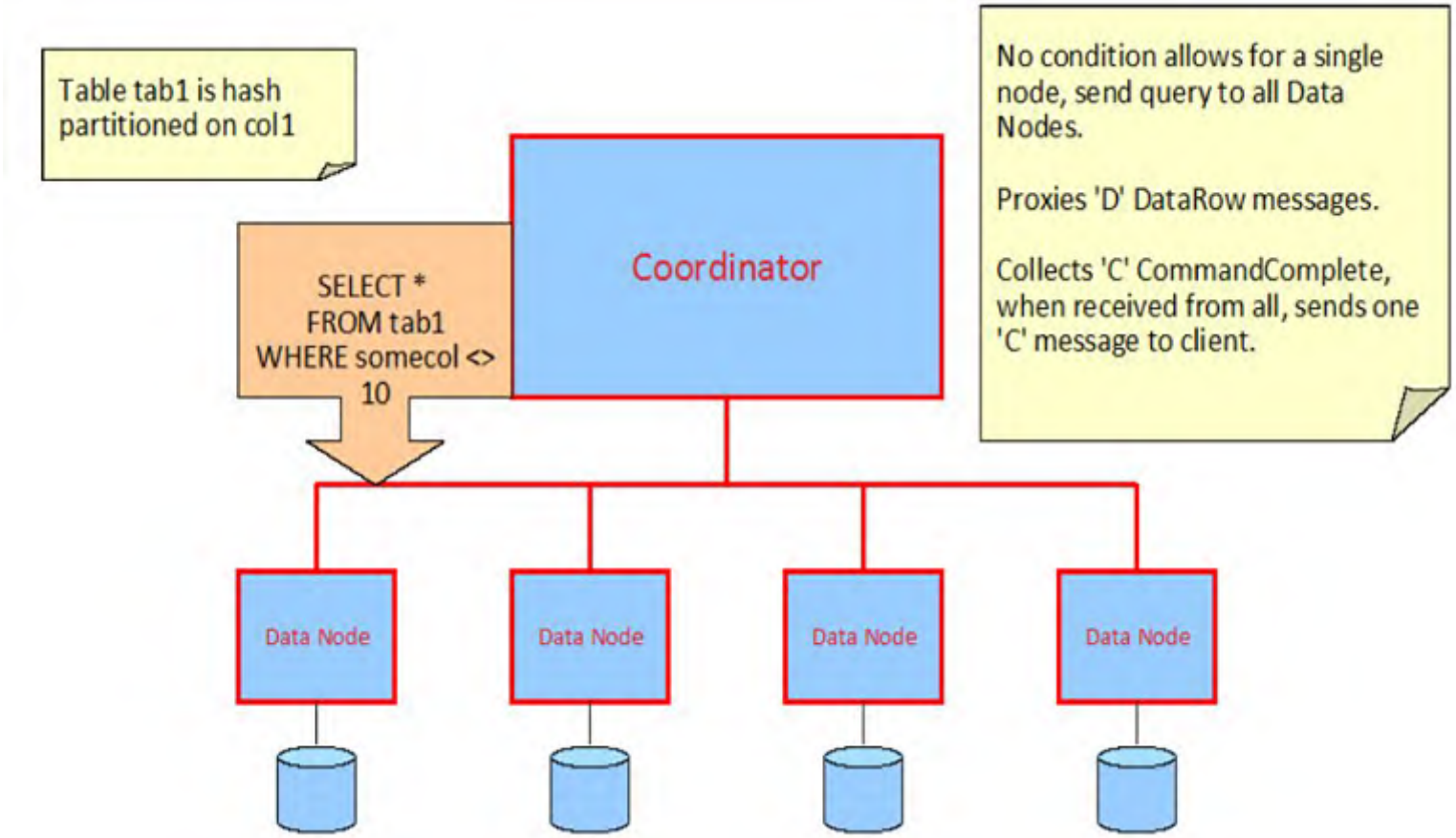


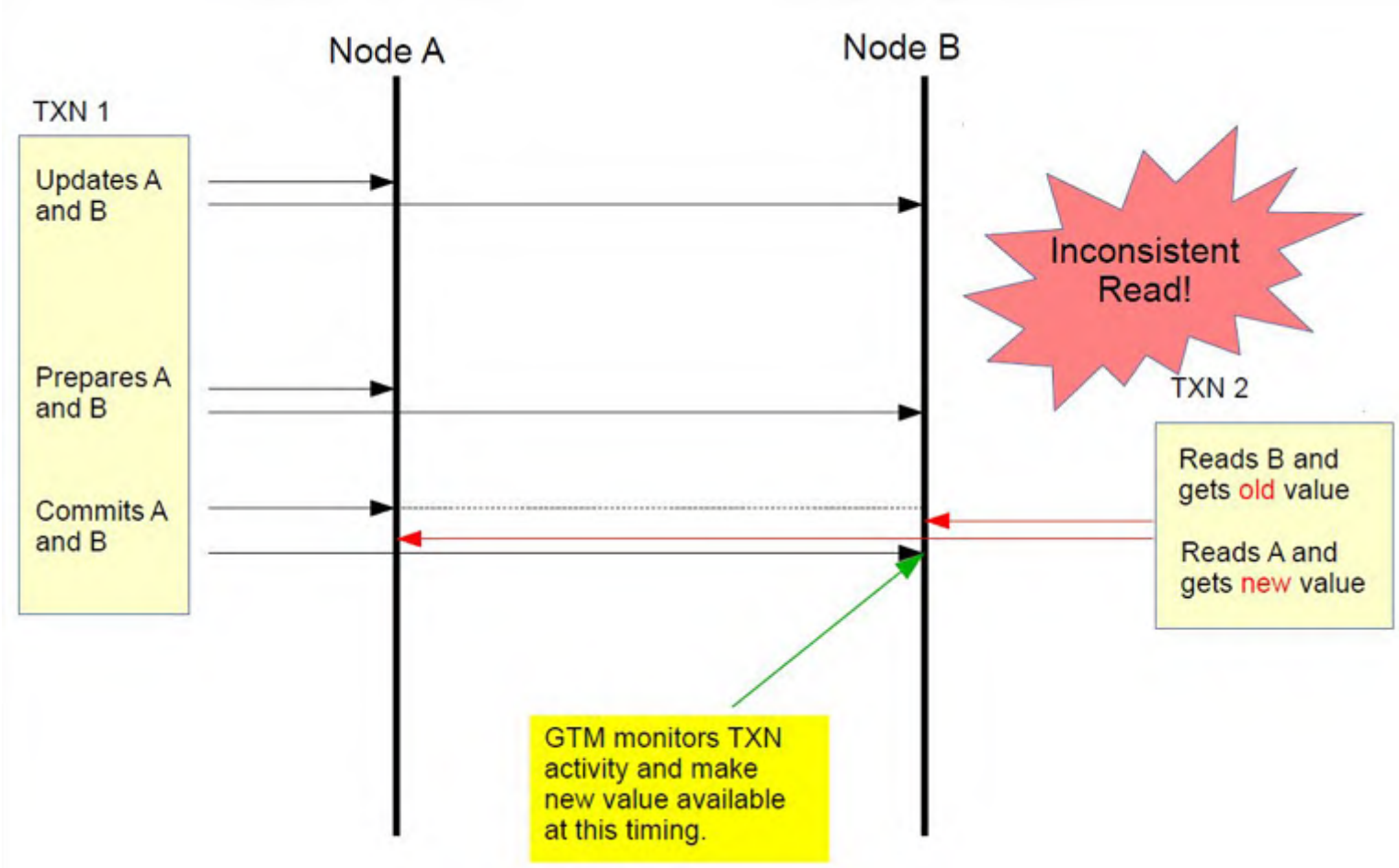
Table tab1 is hash partitioned on col1
SELECT *
FROM tab1
WHERE somecol <> 10
Coordinator
No condition allows for a single node, send query to all Data Nodes.
Proxies 'D' DataRow messages.
Collects 'C' CommandComplete, when received from all, sends one 'C' message to client.
Data Node
Data Node
Data Node
Data Node

MPP方面的优化

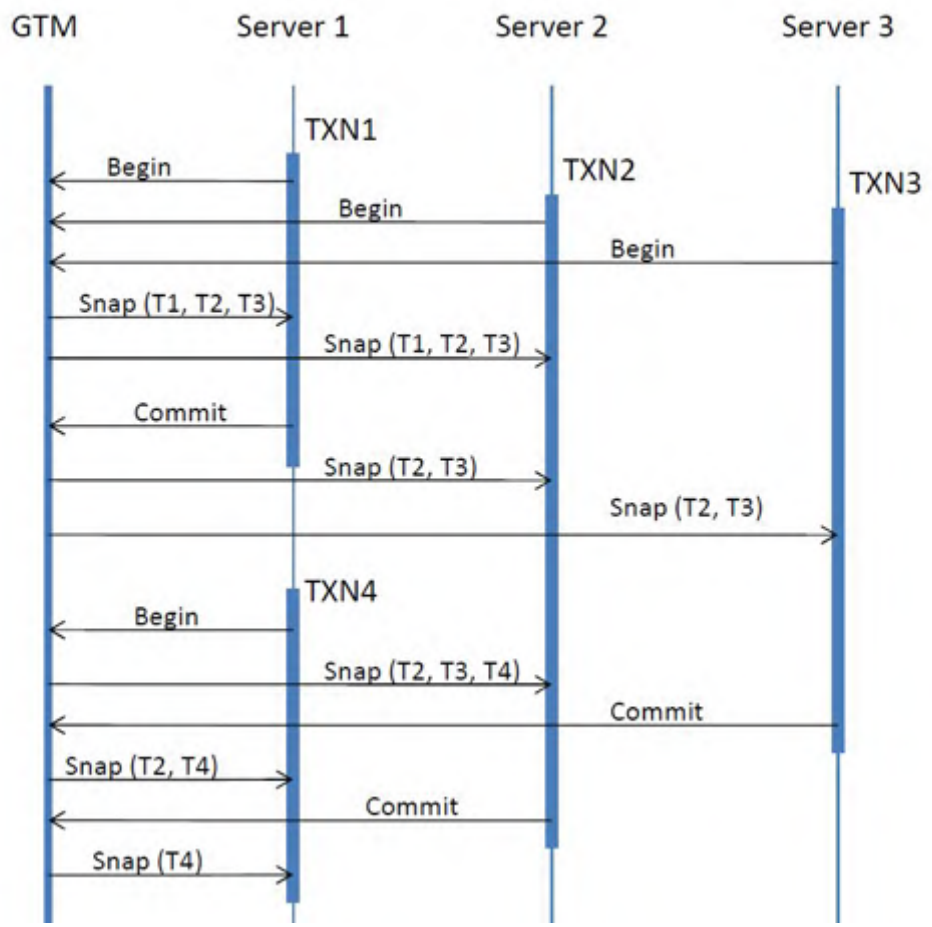
- 数据节点间的数据传输
  - 原来的Postgres-XC的D-Node间不能传数据
  - 数据需要汇聚到C节点进行处理
  - Postgres-XL允许D-Node间进行数据传输
- 执行计划.VS. SQL语句
  - C-node → D-node XC发送SQL语句，XL发送执行计划

Postgres-XL

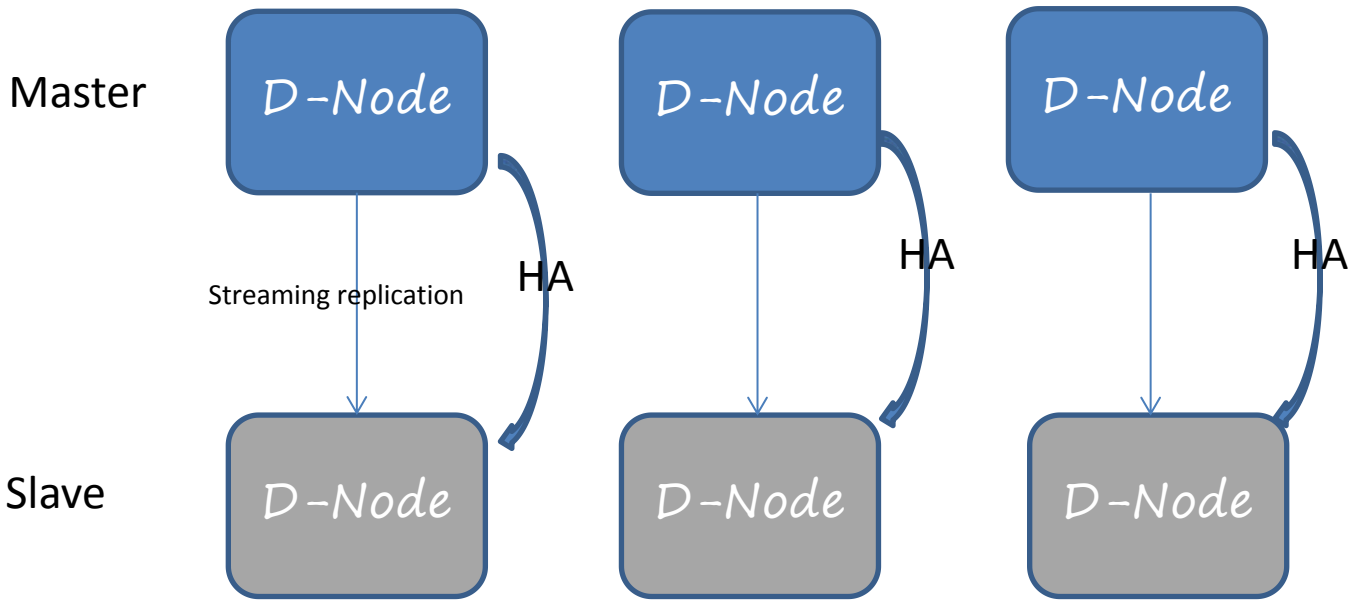
事务处理：2PC就可以了吗？



GTM全局事务状态的处理

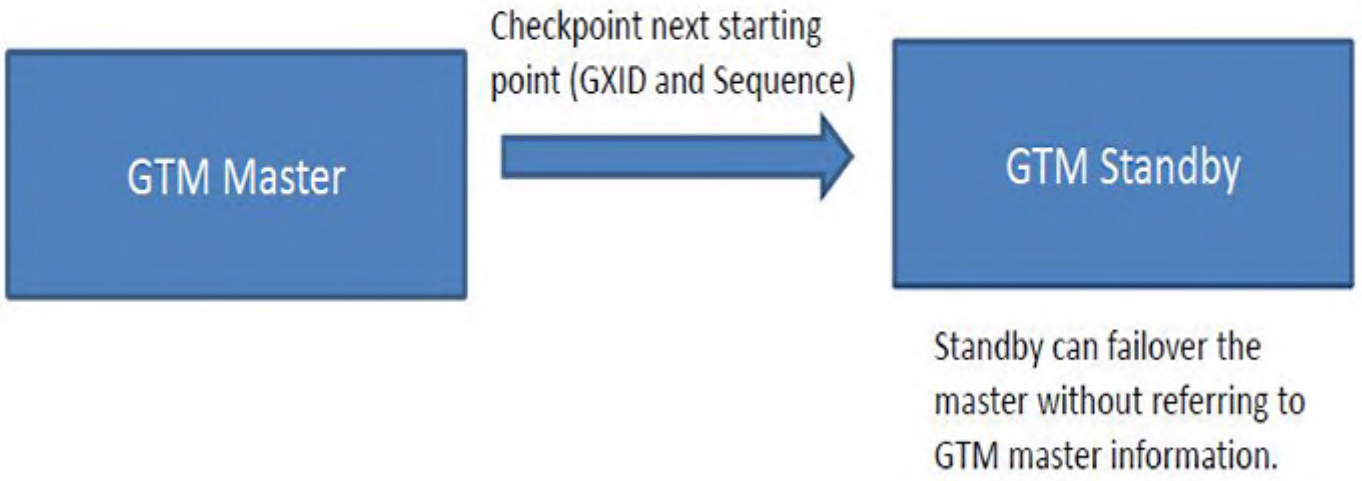


高可用的设计-D-Node



高可用的设计-GTM

- Simple to implement GTM standby



为什么可以处理混合负载

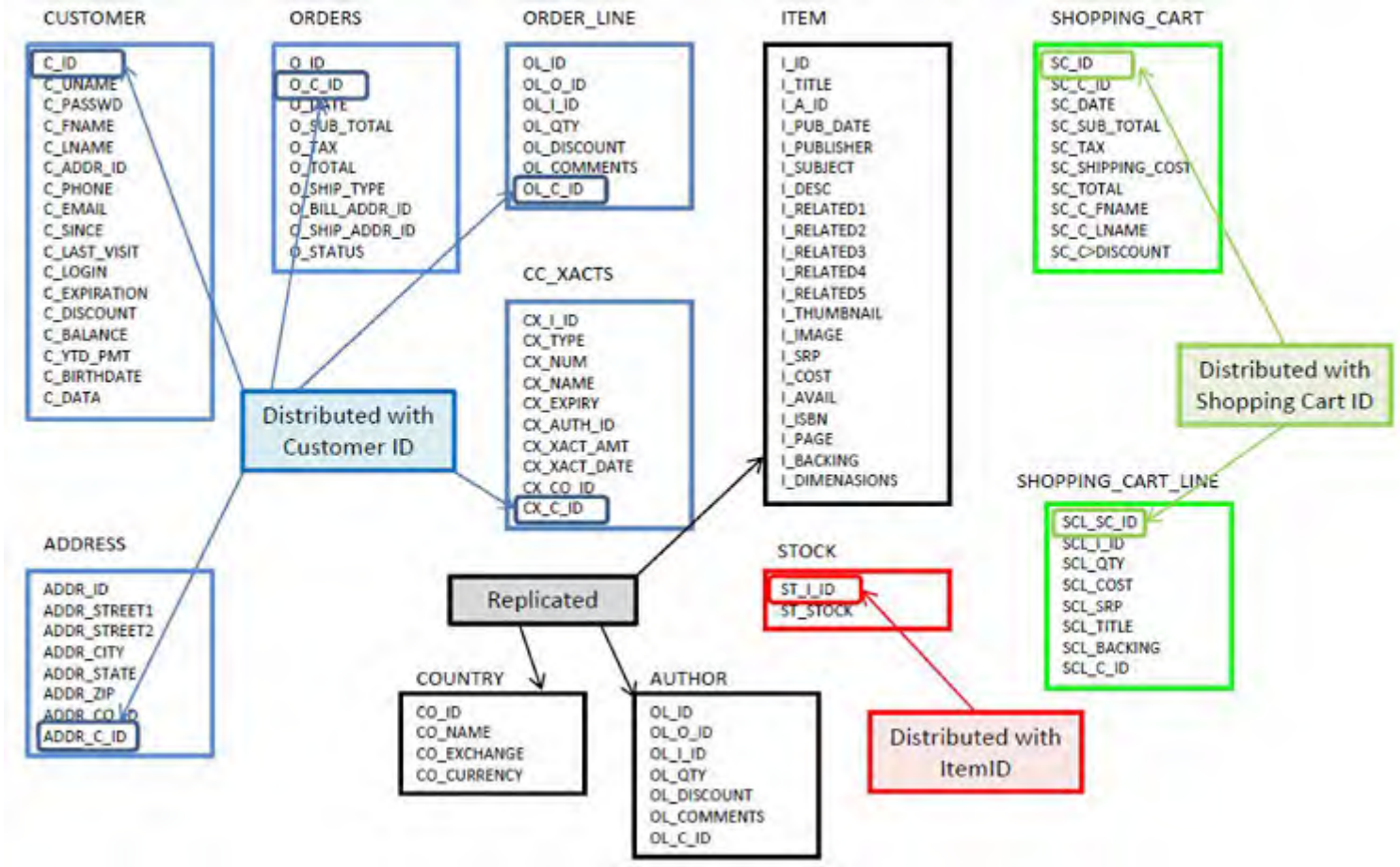
- OLTP
  - 强事务一致性
  - 多主节点可以应付高并发
  - 全功能
- OLAP
  - 多节点并行处理、Share nothing
  - MPP

目录

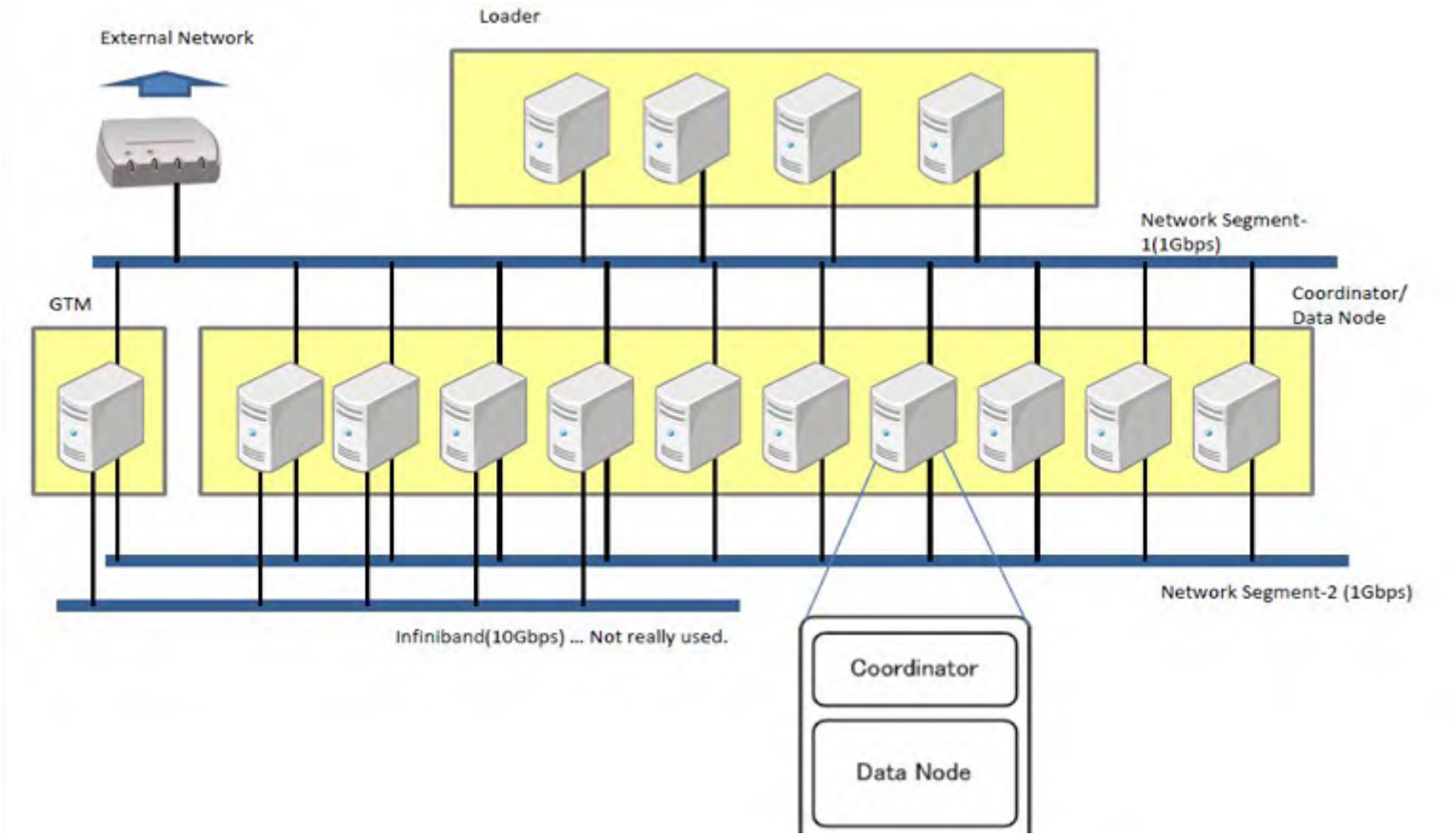
基准测试模型

- DBT-1是TPC-W的开源版
  - 模拟网店的在线交易
  - 主要行为为用户浏览网站、网上购物：primarily shopping (WIPS), browsing (WIPSb) 以及 web-based ordering (WIPSo).
- DBT-3是TPC-H的开源版
  - 数据仓库

DBT-1的表结构


CUSTOMER
C_ID
C_UNAME
C_PASSWD
C_FNAME
C_LNAME
C_ADDR_ID
C_PHONE
C_EMAIL
C_SINCE
C_LAST_VISIT
C_LOGIN
C_EXPIRATION
C_DISCOUNT
C_BALANCE
C_YTD_PMT
C_BIRTHDATE
C_DATA
ORDERS
O_ID
O_C_ID
O_DATE
O_SUB_TOTAL
O_TAX
O_TOTAL
O_SHIP_TYPE
O_BILL_ADDR_ID
O_SHIP_ADDR_ID
O_STATUS
ORDER_LINE
OL_ID
OL_O_ID
OL_I_ID
OL_QTY
OL_DISCOUNT
OL_COMMENTS
OL_C_ID
ITEM
I_ID
I_TITLE
I_A_ID
I_PUB_DATE
I_PUBLISHER
I_SUBJECT
I_DESC
I_RELATED1
I_RELATED2
I_RELATED3
I_RELATED4
I_RELATED5
I_THUMBNAIL
I_IMAGE
I_SRP
I_COST
I_AVAIL
I_ISBN
I_PAGE
I_BACKING
I_DIMENASIONS
SHOPPING_CART
SC_ID
SC_C_ID
SC_DATE
SC_SUB_TOTAL
SC_TAX
SC_SHIPPING_COST
SC_TOTAL
SC_C_FNAME
SC_C_LNAME
SC_C>DISCOUNT
CC_XACTS
CX_I_ID
CX_TYPE
CX_NUM
CX_NAME
CX_EXPIRY
CX_AUTH_ID
CX_XACT_AMT
CX_XACT_DATE
CX_CO_ID
CX_C_ID
Distributed with Customer ID
Distributed with Shopping Cart ID
SHOPPING_CART_LINE
SCL_SC_ID
SCL_I_ID
SCL_QTY
SCL_COST
SCL_SRP
SCL_TITLE
SCL_BACKING
SCL_C_ID
ADDRESS
ADDR_ID
ADDR_STREET1
ADDR_STREET2
ADDR_CITY
ADDR_STATE
ADDR_ZIP
ADDR_CO_ID
ADDR_C_ID
Replicated
STOCK
ST_I_ID
ST_STOCK
Distributed with ItemID
COUNTRY
CO_ID
CO_NAME
CO_EXCHANGE
CO_CURRENCY
AUTHOR
OL_ID
OL_O_ID
OL_I_ID
OL_QTY
OL_DISCOUNT
OL_COMMENTS
OL_C_ID

硬件拓扑结构

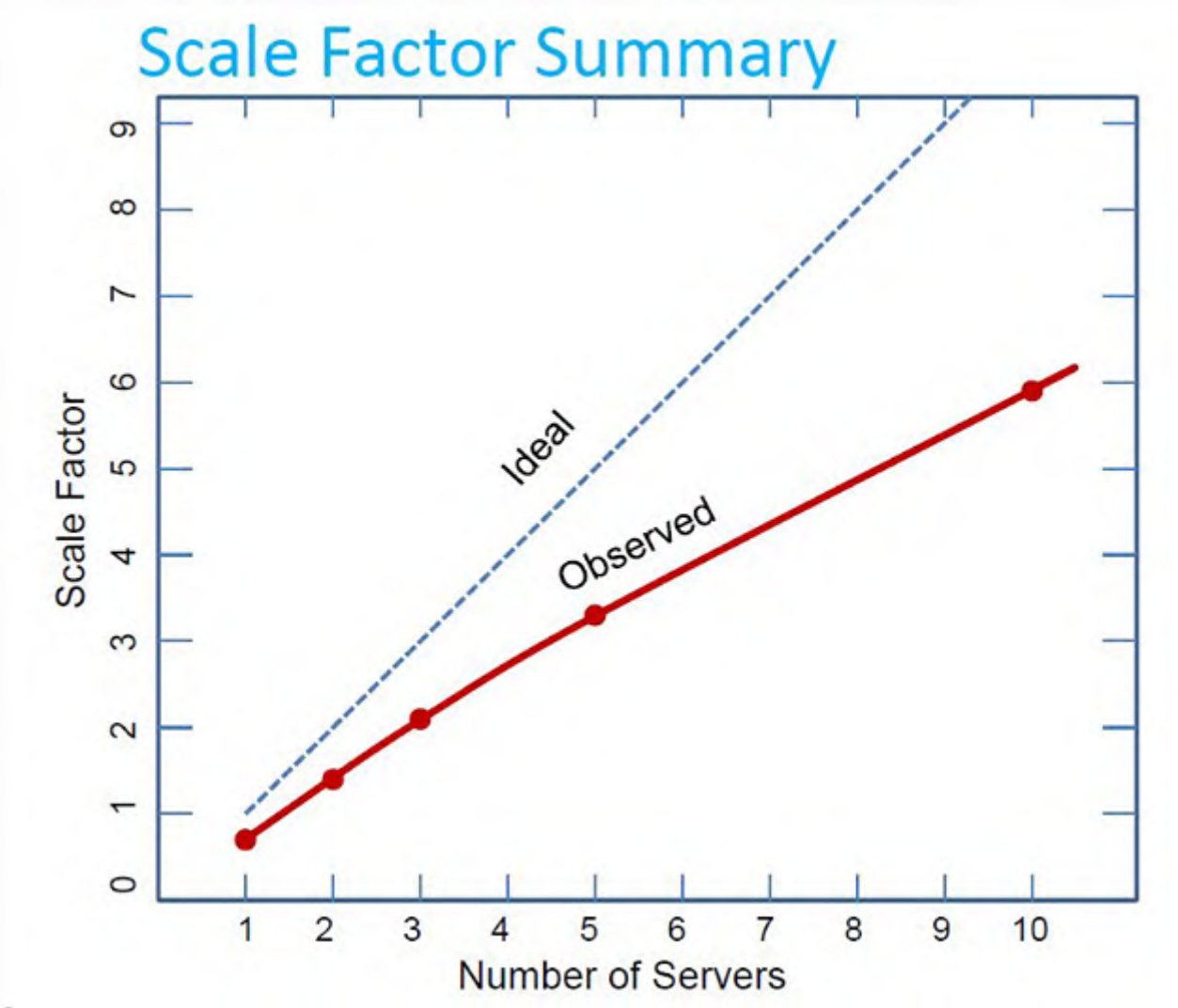

External Network
Loader
Network Segment-1(1Gbps)
GTM
Coordinator/ Data Node
Network Segment-2 (1Gbps)
Infiniband(10Gbps) ... Not really used.
Coordinator
Data Node

OLTP的测试性能

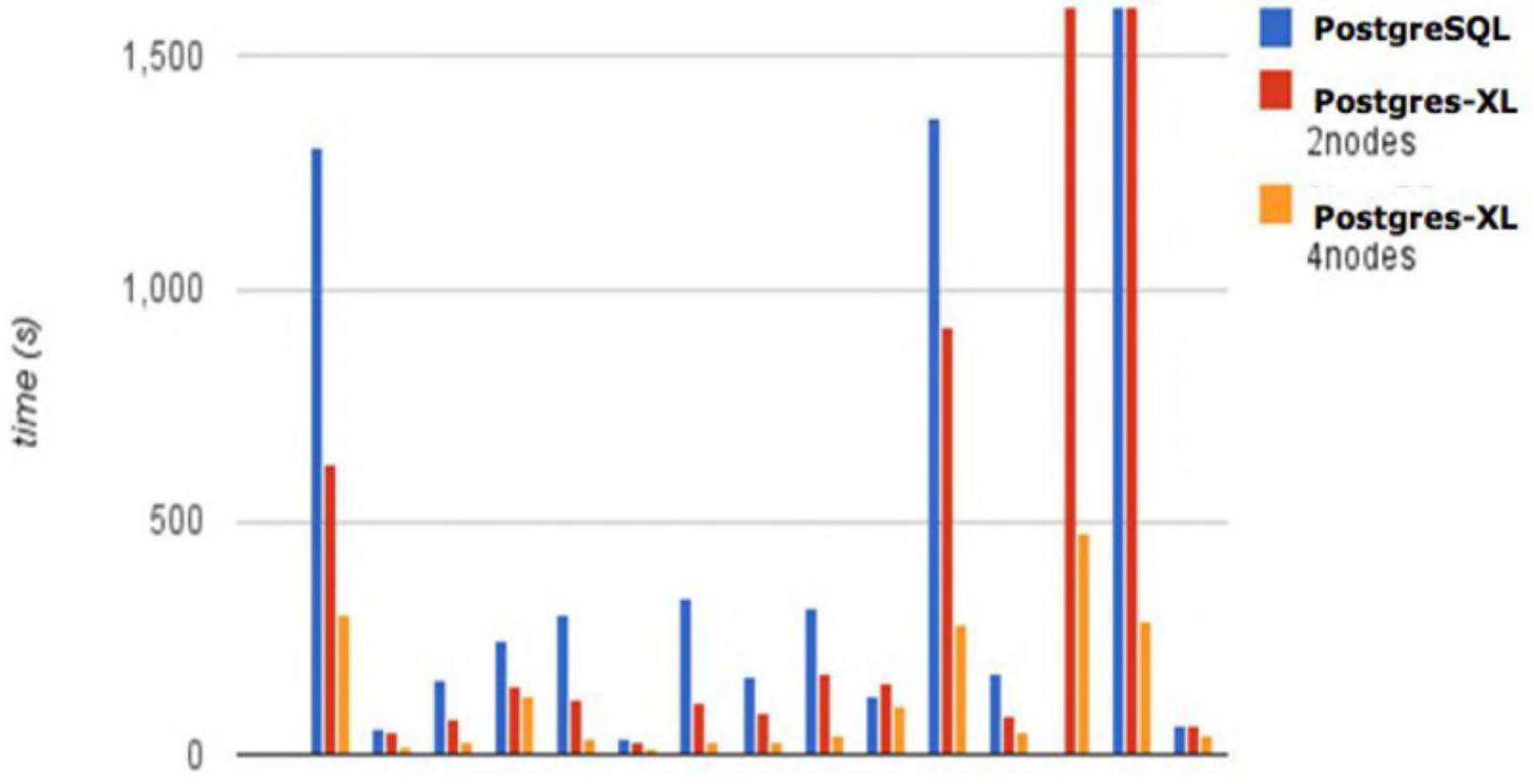
Full Load Throughput

Database	Num. of Servers	Throughput (TPS)	Scale Factor
PostgreSQL	1	2,617	1.0
Postgres-XC	1	1,869	0.71
Postgres-XC	2	3,646	1.39
Postgres-XC	3	5,379	2.06
Postgres-XC	5	8,473	3.24
Postgres-XC	10	15,380	5.88

性能基本可以线性扩展


Scale Factor Summary
Scale Factor
0
1
2
3
4
5
6
7
8
9
Ideal
Observed
1
2
3
4
5
6
7
8
9
10
Number of Servers

TPC-H的性能


1,500
1,000
500
0
time (s)
PostgreSQL
Postgres-XL
2nodes
Postgres-XL
4nodes

应用场景

- 大规模OLTP应用，尤其是企业领域
- 云环境下的弹性伸缩
- OLTP及OLAP混合负载
- 详单查询
- ODS
- …

目录

相关的资料

- https://github.com/postgres-x2/
- http://www.postgres-xl.org/

谢谢

galylee@gmail.com