KunlunDB 基本管理

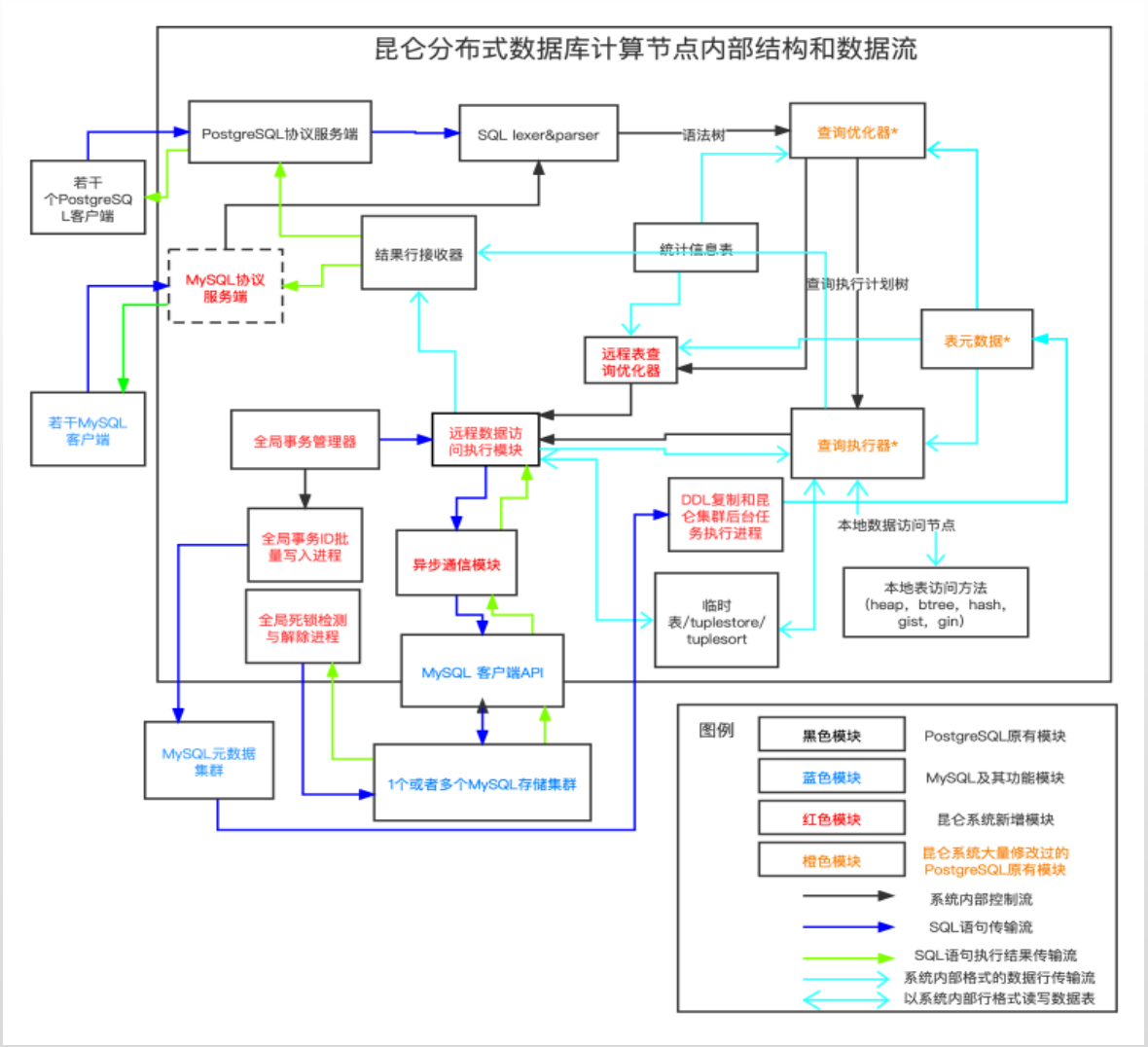
何革新

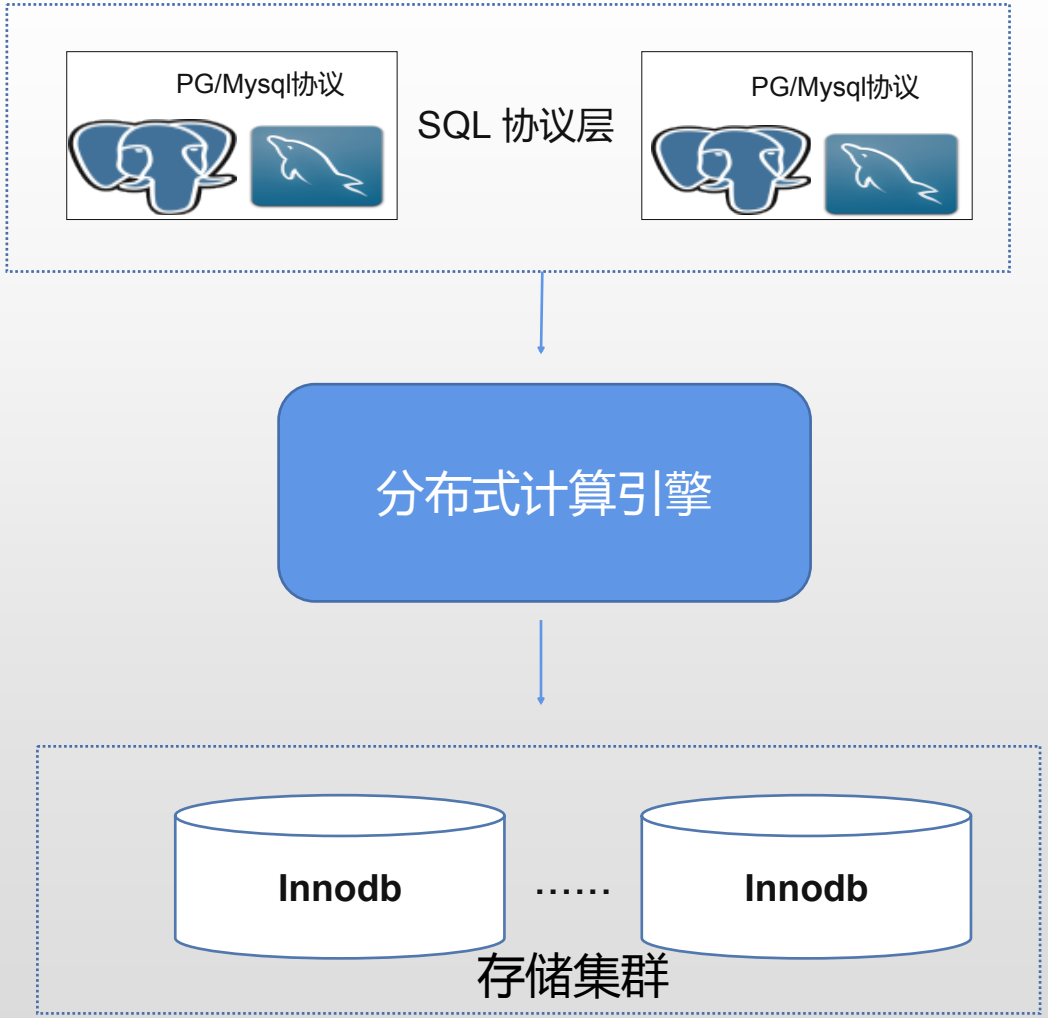
泽拓科技（深圳）有限责任公司

目录

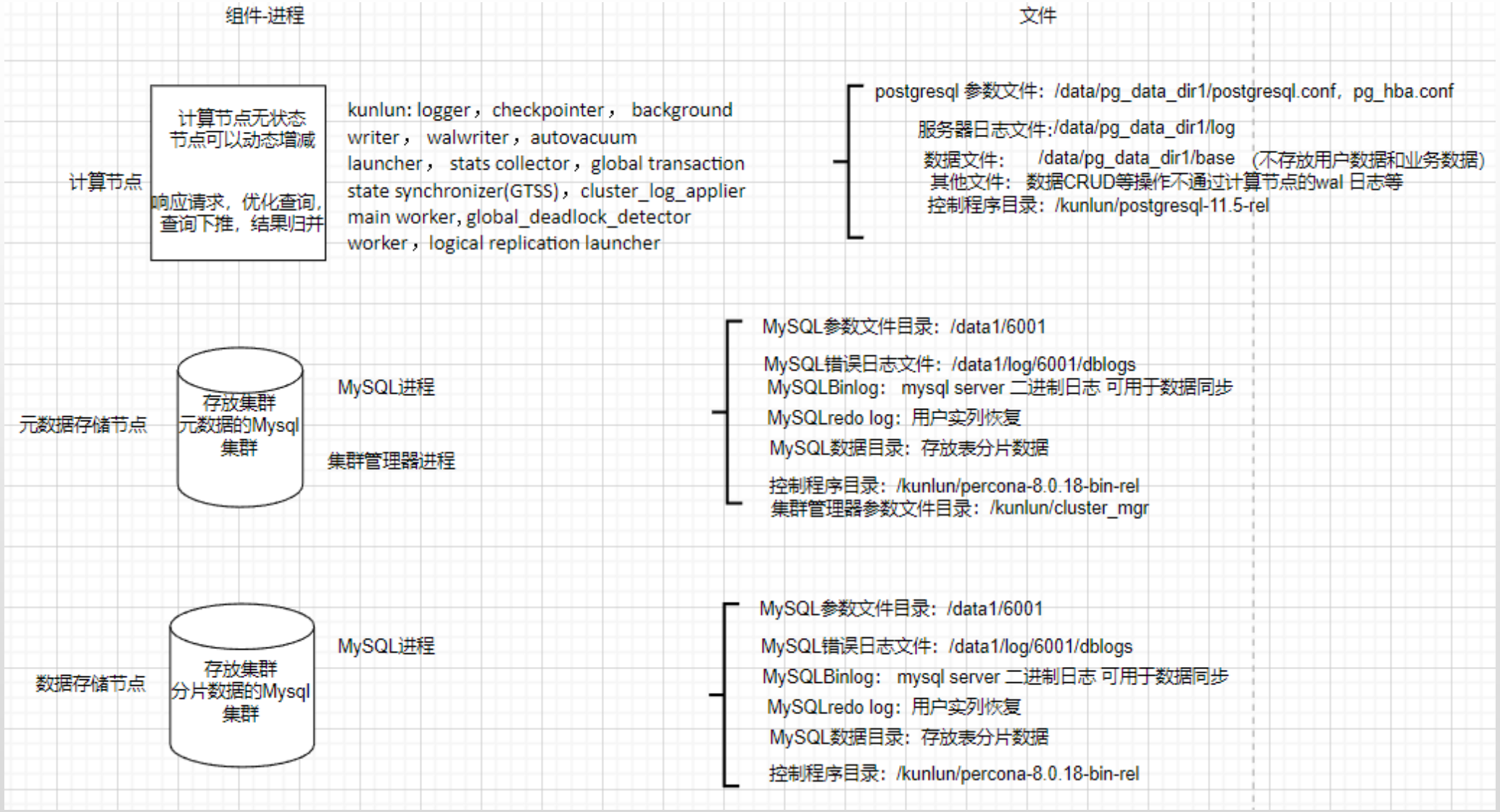
CONTENTS

KunlunDB 计算流程图



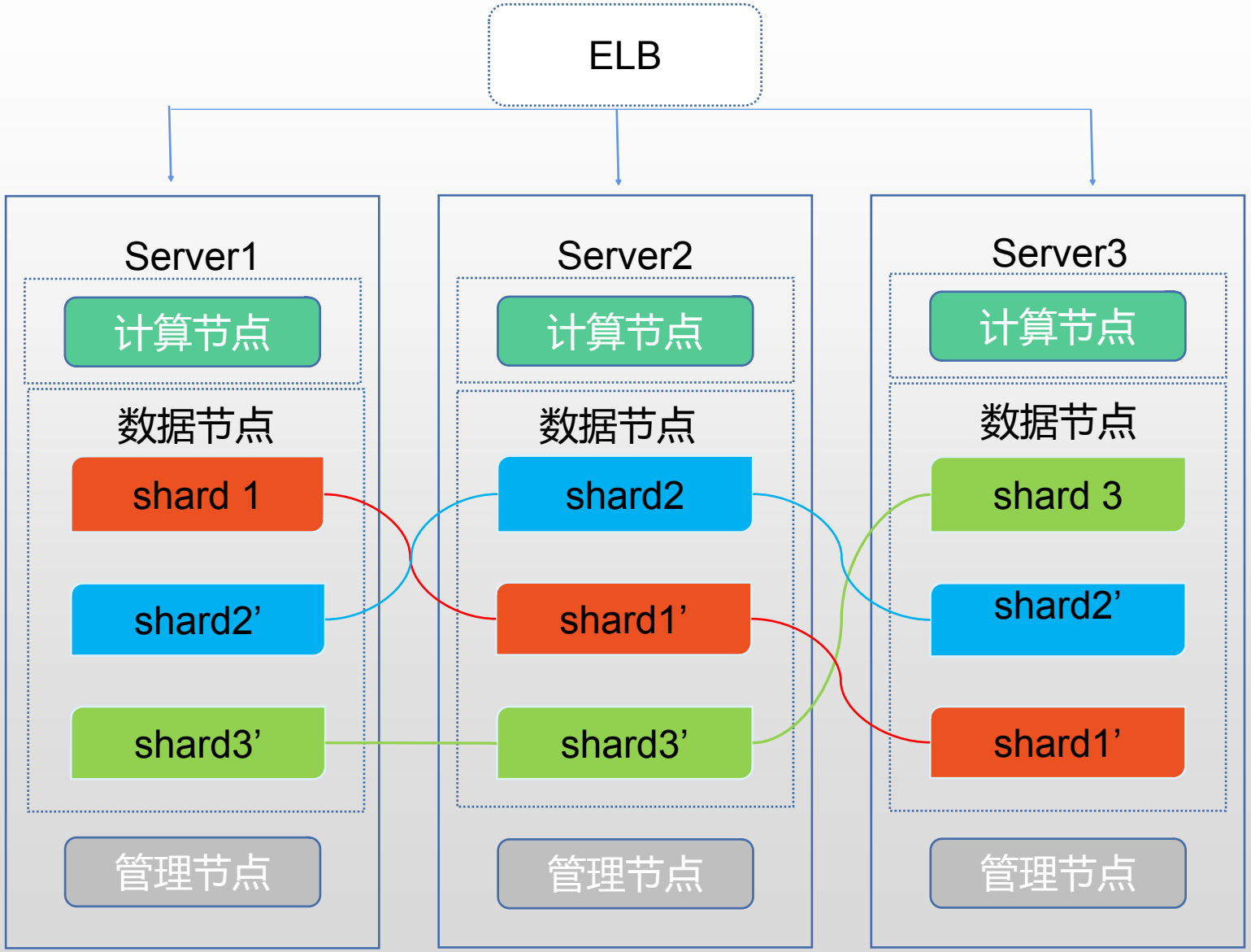


KunlunDB 物理结构



部署架构-对等部署方案

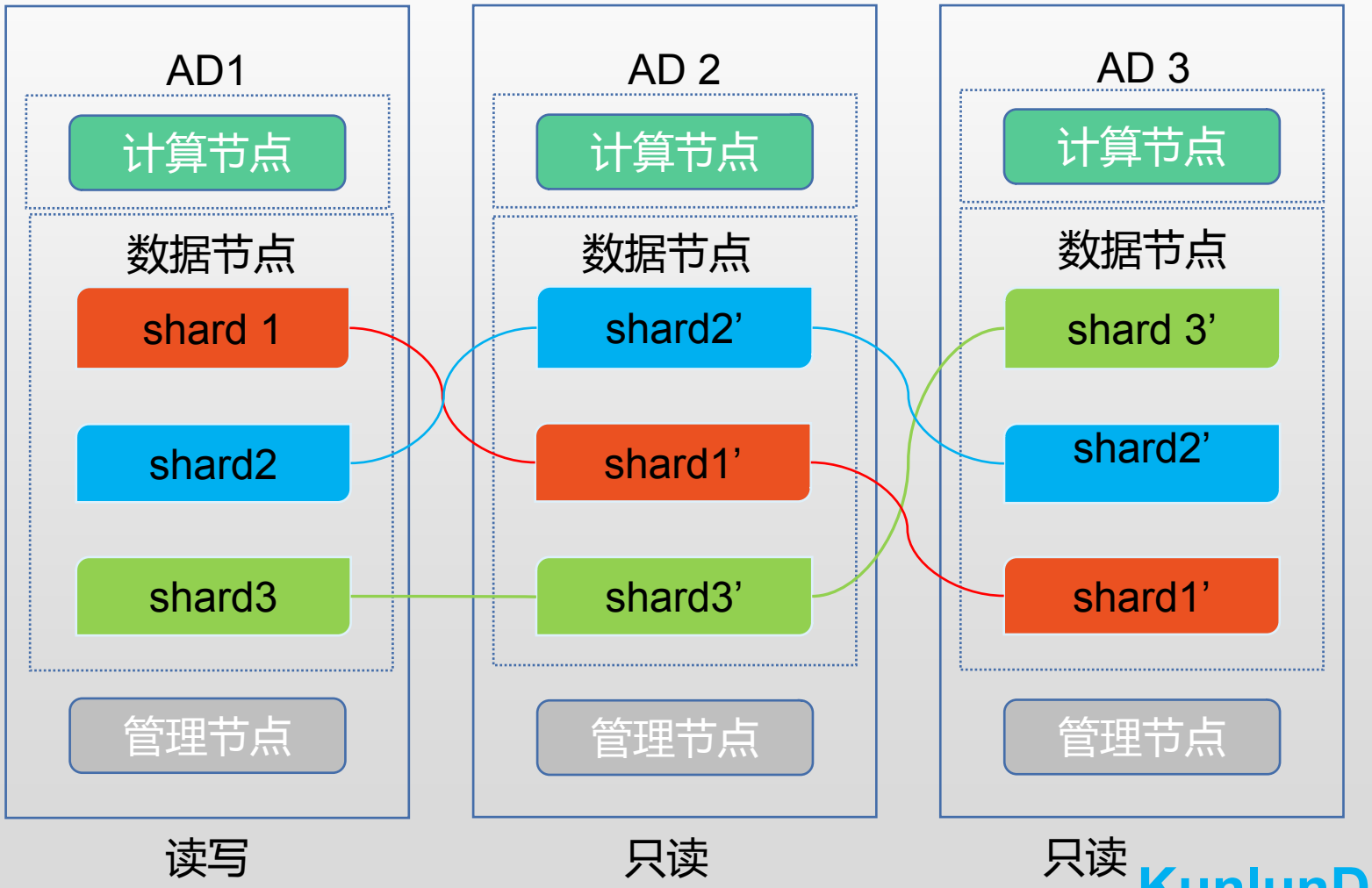
- 服务器：每台服务器是一个shard集群，集群数据库的主从副本散列在各个服务器上，实现负载均衡
- 服务器之间是对等关系，但每个分区数据的主从副本不存储在同一台服务器里
- 每台服务器部署全部的集群组件：计算节点， 存储节点和管理节点



KunlunDB

部署架构-多AD 部署方案

- AD （可用区）：每个可用区是一个服务器&网络等基础设施的共用区，处于同一个可用区的服务器具有一致的可靠性特性，每个可用区部署多个计算节点及多个存储节点。可用区可以是一台服务器单位，一个机架单位，或一个机房单位，或同一个城市单位。同一AD内的服务器可以纵向扩展
- 不同的可用区是处于对等的单位的部署位置，数据在可用区之间互为冗余实现高可靠性
- 每个可用区的计算节点运行 postgresql 服务器，postgresql 实列数目根据负载动态增减
- 数据节点运行MySQL服务器进程，每个MySQL 实列是一个shard
- 管理节点运行MySQL服务器进程作为元数据管理



KunlunDB 一键安装

1. 准备服务器：操作系统，补丁，依赖软件包清单（参考安装手册）
2. 规划安装：计算节点、存储节点、管理节点 、磁盘空间等
3. 下载安装工具包： git clone https://github.com/zettadb/cloudnative.git
4. 下载集群软件
5. 修改配置文件：

```
kunlun@kunlun-server:~/cloudnative/cluster$ ls
 backup                        generate_scripts.py
 backup.json                  'how archive_mode'
 clean                         init_ubuntu.sh
 clustermgr.cnf.template       install
 cluster_mgr_rel               install.json
```

安装指令：bash install/commands.sh

```
{
    "machines":[
        {
            "ip":"172.18.0.136",
            "basedir":"/kunlun",
            "user":"kunlun"
        }
    ],
    "cluster":{
        "name":"clust1",
        "meta":{
            "nodes":[
                {
                    "is_primary":true,
                    "ip":"172.18.0.136",
                    "port":6001,
                    "xport":60010,
                    "mgr_port":60011,
                    "innodb_buffer_pool_size":"64MB",
                    "data_dir_path":"/kunlun/data1",
                    "log_dir_path":"/kunlun/data1/log",
```

KunlunDB 部署配置文件

- 从元数据库获取整个集群的配置信息

登录元数据库：mysql> use kunlun_metadata_db ;

1. 计算节点的信息 ： 参考表comp_nodes;
2. 元数据节点信息： 参考表meta_db_nodes
3. 存储节点信息 ： 参考表：shard_nodes

```
mysql> use kunlun_metadata_db;
Database changed
mysql> show tables;
+-------------------------------+
| Tables_in_kunlun_metadata_db |
+-------------------------------+
| commit_log                    |
| commit_log_clust1             |
| comp_nodes                    |
| comp_nodes_id_seq             |
| db_clusters                   |
| ddl_ops_log_clust1            |
| ddl_ops_log_template_table    |
| meta_db_nodes                 |
| shard_nodes                   |
| shards                        |
+-------------------------------+
```

KunlunDB 计算节点管理概要（一）

- 计算节点运行一套Postgresql 数据库服务进程 ，响应SQL 会话及请求的处理
- 计算节点是客户应用程序访问数据库的入口，因此需要做基本的账号及权限管理,数据库对象的创建，数据库分片策略的执行等
- 计算节点的不承担数据持久化功能，所有数据的管理功能将被下推到存储节点
- 计算节点无状态，可以按需增减计算节点，所有计算节点可读写，计算节点的配置信息记录在元数据库中

KunlunDB 计算节点管理概要（二）

计算节点集群专用系统表：

pg_cluster_meta :

comp_node_id|*cluster_id*|*cluster_master_id*|*ha_mode*|*cluster_name*|*comp_node_name*

pg_cluster_meta_nodes :

server_id|*cluster_id*|*is_master*|*port*|*user_name*|*hostaddr* |*passwd*

pg_shard :name
|id|master_node_id|num_nodes|space_volumn|num_tablets|db_cluster_id|when_created

pg_shard_node:id|port|shard_id|svr_node_id|ro_weight|user_name|hostaddr |passwd
|when_created

pg_computing_node_stat:

pg_ddl_log_progress:

pg_class :

KunlunDB 计算节点管理概要（三）

计算节点管理任务(演示）：

- Postgresql服务进程的启动、停止及异常日志检测

默认情况下， 集群启动后所有相关进程会自动启动， 如需要手工启动， 可以通过控制目录的程序或脚本
启动&停止 postgresql 主进程， 服务器异常日志位于data/log 目录中

- Postgresql访问权限的配置

✓ 通过配置pg_hba.conf文件及pg 数据库中的权限管理，创建数据库，用户，角色，及 schema

- 服务器参数配置和管理

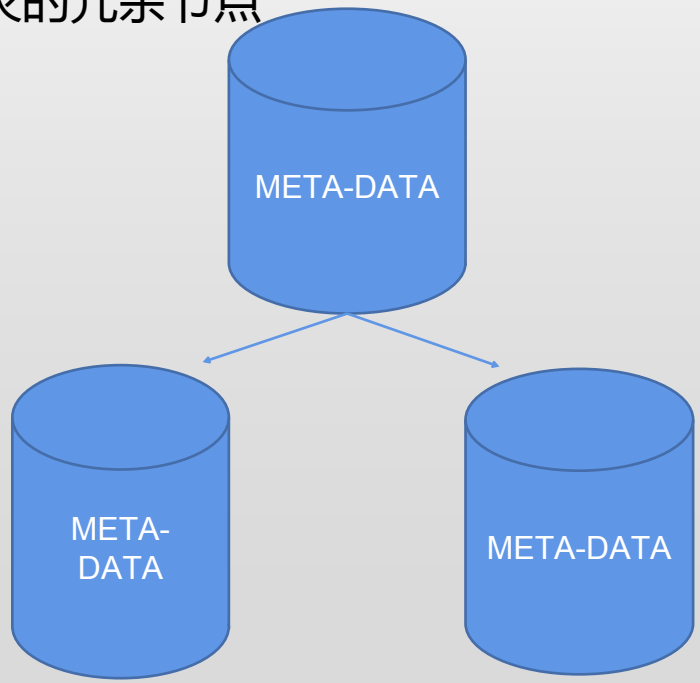
✓ 内存参数：shared buffers，work_mem
✓ 并发度参数：max_connections = 1000
✓ timeout 参数：
✓ 全局死锁检测参数：
✓ 查询下推参数：

enable_remote_join_pushdown = true
enable_remote_agg_pushdown = true
enable_remote_orderby_pushdown = true
enable_remote_distinct_pushdown = true
enable_remote_limit_pushdown = true

KunlunDB 元数据节点管理概要（一）

- 元数据节点运行一套Mysql 数据库服务进程 ，在数据库内保存整个集群的配置信息，包括节点IP 地址、port，用户名等

- 为提高可用性， 元数据节点采取一主多从的配置，只有一个节点可读写，其他节点作为容灾的冗余节点




```
mysql> use kunlun_metadata_db;
Database changed
mysql> show tables;
+-----------------------------+
| Tables_in_kunlun_metadata_db |
+-----------------------------+
| commit_log                  |
| commit_log_clust1           |
| comp_nodes                  |
| comp_nodes_id_seq           |
| db_clusters                 |
| ddl_ops_log_clust1          |
| ddl_ops_log_template_table  |
| meta_db_nodes               |
| shard_nodes                 |
| shards                      |
+-----------------------------+
```

KunlunDB 元数据节点管理概要（二）

元数据节点管理任务（演示）：

■ Mysql 服务进程的启动、停止
默认情况下， 集群启动后所有相关进程会自动启动， 如需要手工启动， 可以通过控制目录的程序或脚本启动&
停止 Mysql 服务进程
■ Mysql 访问权限的配置
用户及权限在安装的配置文件中设定

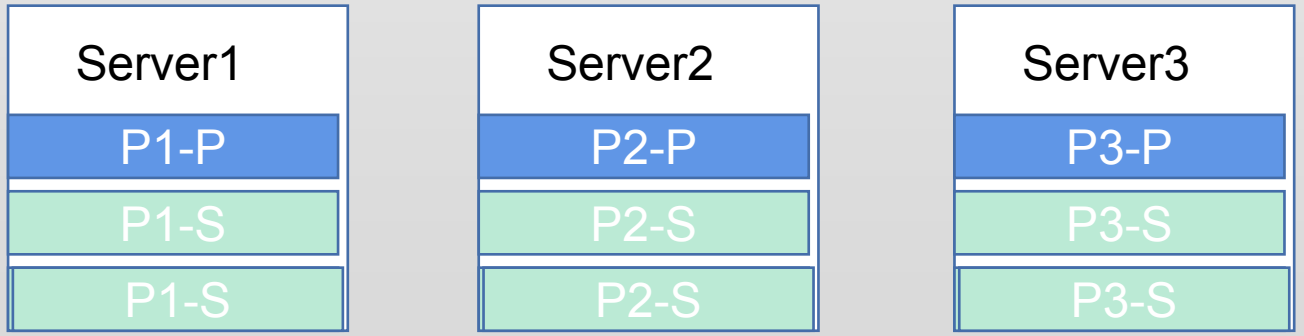
■ Mysql服务器参数的配置：按默认配置即可
✓ 内存参数：innodb_buffer_pool_size ,sort_buffer_size
✓ 并行度参数：
✓ 数据库复制参数：

■ 备份&恢复

通过集群统一备份&恢复工具管理

KunlunDB 存储节点管理概要（一）

- 存储数据节点运行Mysql 数据库服务进程 ，存放系统的业务数据库
- 每个存储节点属于一个SHARD, 各个SHARD分别存放分片的部分行数据
- 每个 SHARD 可以配置多个从副本，复制程序保证SHARD 的主从副本的一致性
- 数据的查询更新等操作主要发生在存储节点，因此可以通过调节数据库的内存参数改善数据库的性能 .



KunlunDB 存储节点管理概要（二）

数据节点管理任务（演示）：

■ Mysql 服务进程的启动、停止
默认情况下， 集群启动后所有相关进程会自动启动， 如需要手工启动， 可以通过控制目录的程序或脚本启动&停止 Mysql 服务进程

■ Mysql 访问权限的配置

■ Mysql服务器参数的配置
✓ 内存参数：innodb_buffer_pool_size ,sort_buffer_size
✓ 并行度参数：
✓ 数据库复制参数：

■ 备份&恢复
✓ 通过集群统一备份&恢复工具管理

Q&A

谢谢观看~