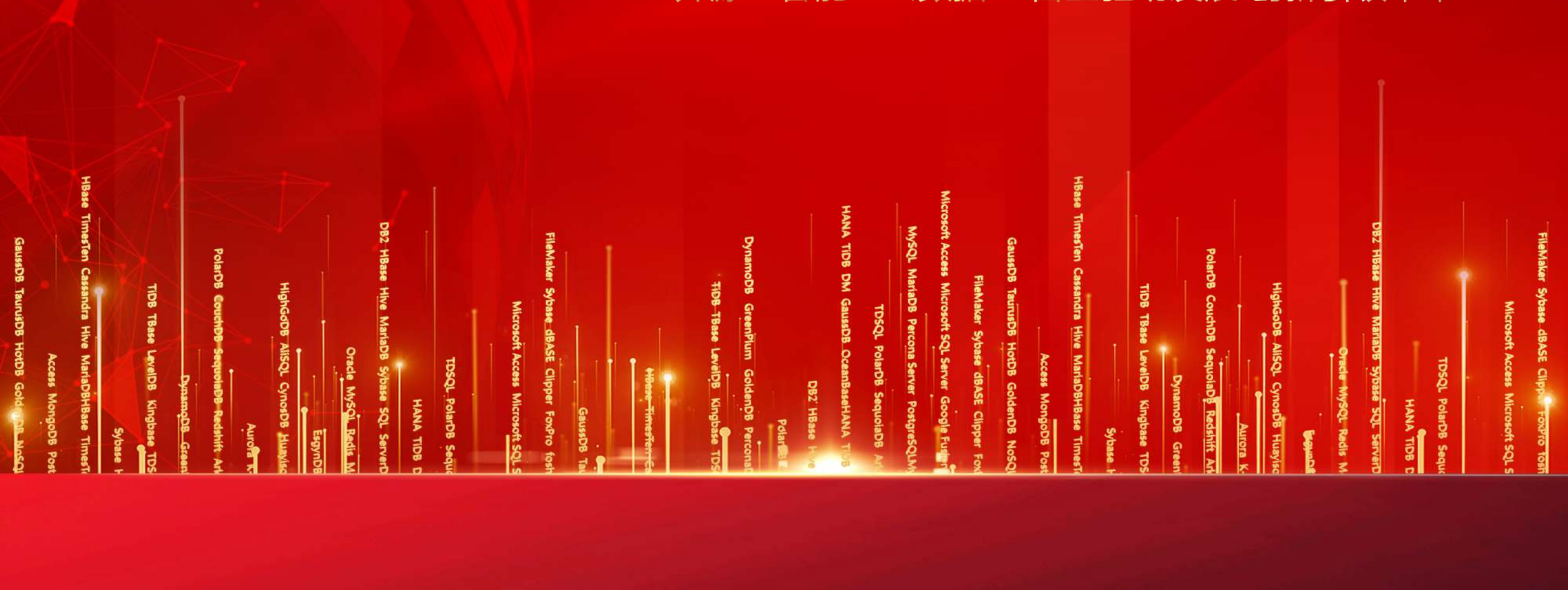
11/15-16 | 北京
新云南皇冠假日酒店

2019

# 数据技术嘉年华

## Data Technology Carnival

开源 · 智能 · 云数据 – 自主驱动发展 创新引领未来



# POLARDB 产品架构与实现

周振兴 阿里云资深数据库产品专家

# POLARDB 领先的云原生数据库

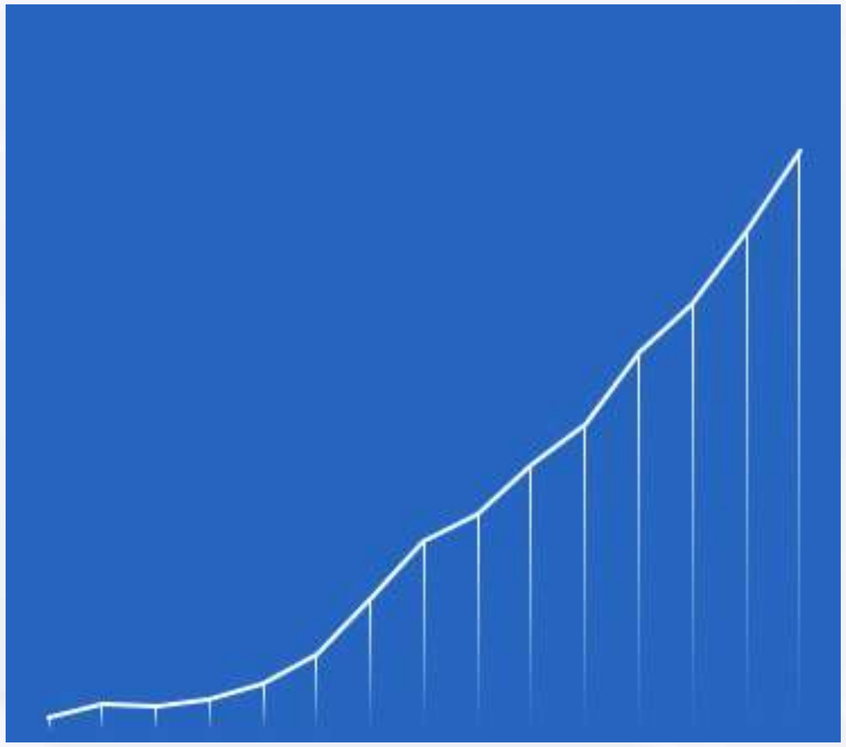
100%兼容MySQL、PostgreSQL，高度兼容Oracle

Very Large Data Bases

VLDB2018

**PolarFS: An Ultra-low Latency and Failure Resilient Distributed File System for Shared Storage Cloud Database**

Wei Cao, Zhenjun Liu, Peng Wang, Sen Chen, Caifeng Zhu,
Song Zheng, Yuhui Wang, Guoqing Ma
{mingsong.cw, zhenjun.lzj, wangpeng.wangp, chensen.cs, caifeng.zhucaifeng,
catttree.zs, yuhui.wyh, guoqing.mgq}@alibaba-inc.com



**增长最快的**

云数据库产品

POLARDB

世界互联网大会 World Internet Conference

世 界 互 联 网

## 领先科技成果



2019中国数据库年度最佳创新产品


POLARDB获得2018年度最佳创新产品奖

开源 · 智能 · 云数据 – 自主驱动发展 创新引领未来

# 关于作者

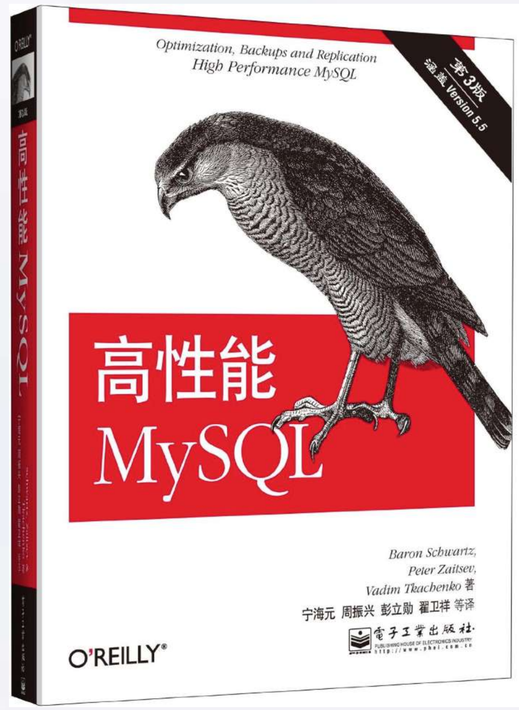
周振兴（花名：苏普） 资深数据库产品专家

2009年加入阿里巴巴

2009~2014年 负责淘宝核心交易系统的去IOE

2014~2019年 负责云上数据库产品规划与设计

联系方式： supu@alibaba-inc.com
orczhou@gmail.com
个人博客： http://www.orczhou.com




2013年最具技术影响力引进图书TOP10

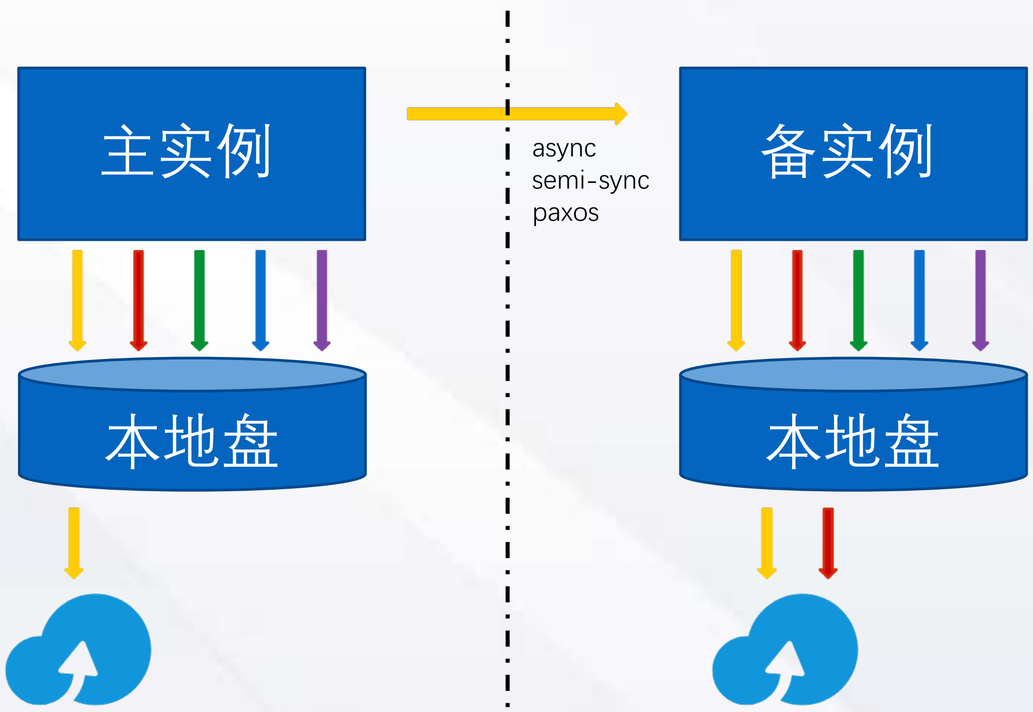
# 目录

# 阿里巴巴数据库发展

| 商业数据库 | 开源数据库 | 云数据库 | 自研云原生时代 |
| --- | --- | --- | --- |
| ~2008 | 2008~2014 | 2011~2017 | 2017~ |
| | 去IOE | RDS 系列产品 | POLARDB |
| Oracle ACE | AliSQL | NoSQL系列产品 | AnalyticDB |
| 亚洲最大的Oracle RAC | 异地多活 | 生态工具 | OceanBase |
| 大量使用Oracle数据库 | 垂直拆分 | OLAP | |
| 近百个 | 近万个 | 数十万 | |

# 传统数据库架构面临的挑战

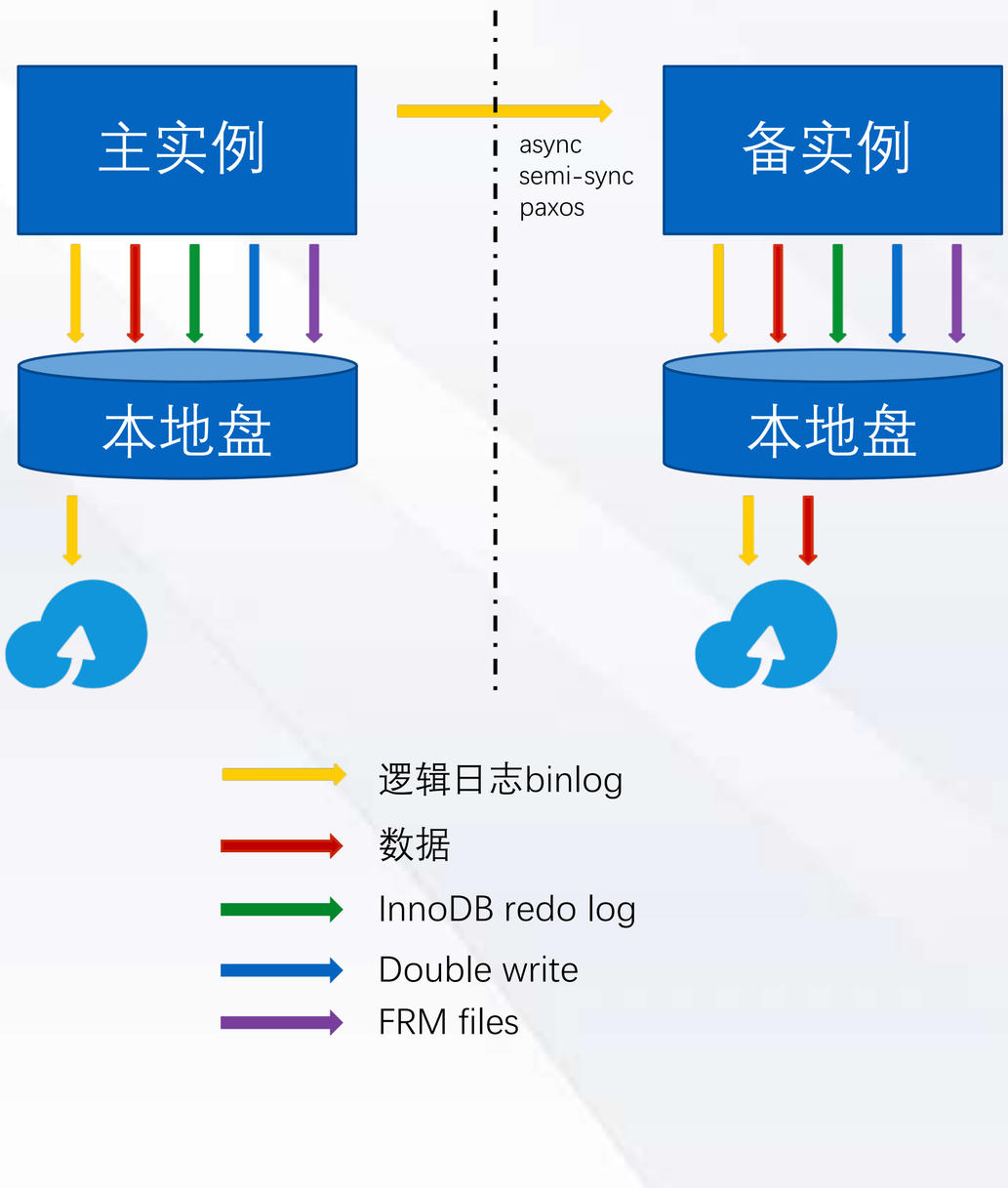
MySQL高可用 典型主备架构




适合大多数的业务场景

# 传统数据库架构面临的挑战

MySQL高可用 典型主备架构




适合大多数的业务场景，但是

- 主备延迟
  - 主备切换时，备库无法及时提供服务
  - 数据写入后，只读实例中却查不到
- 升级变配慢
  - 跨机迁移时，可能要数小时，甚至更久
- 新增只读节点慢
  - 根据备份/日志大小，数小时，甚至更久
- 存储空间限制
  - 数据存储空间有限，不建议超过2TB
  - 数据备份时间长

# 云原生时代的数据库面临什么挑战？

- 能否具备更强的弹性能力，以最低成本应对业务快速增长？
- 能否支持后互联网时代的，海量数据存储的需求？
- 能否继续享受开源红利，同时具备商业数据库的可靠性？
- 能否全托管，让研发资源专注于数据架构与业务？

# POLARDB 产品概述

POLARDB

POLARDB是阿里巴巴自研的新一代云原生数据库，在存储计算分离架构下，利用了软硬件结合的优势，为用户提供具备高性能、海量存储、安全可靠的数据库服务。100%兼容MySQL 5.6/8.0，PostgreSQL 11，高度兼容Oracle。

**100%**

100%兼容MySQL，PG

**100TB**

存储空间最大支持100TB

**5分钟**

5分钟新增只读节点
15分钟完成节点升降级

MySQL

PostgreSQL

ORACLE

**RPO=0**

ParalleRaft协议同步写入
RPO=0

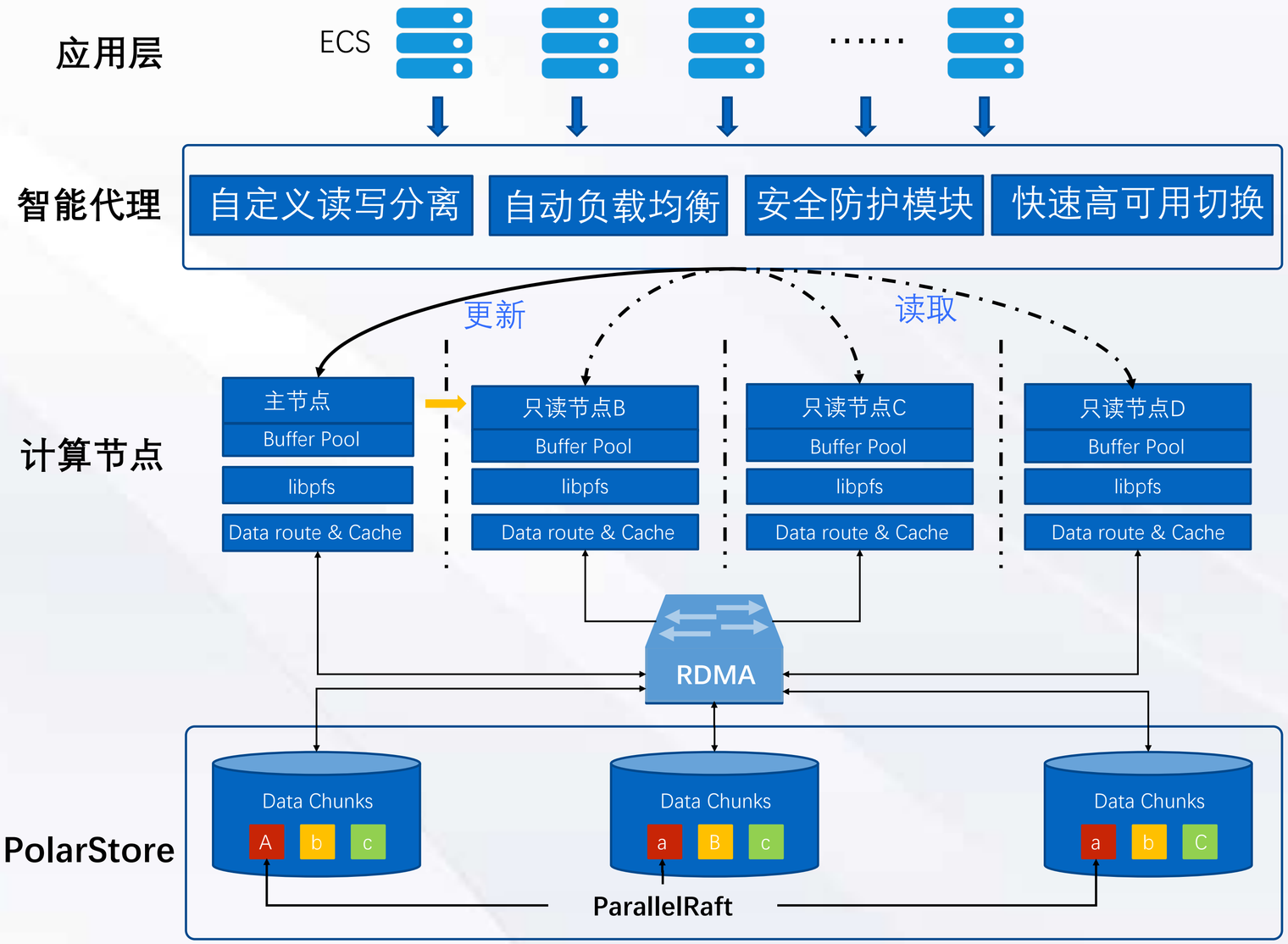
**6倍**

6倍RDS MySQL性能

**毫秒**

主备延迟为毫秒级别

# POLARDB 产品概述




## 架构特点

### 存储计算分离

分钟级新增只读节点

分钟级实现实例升降级

### 智能代理转发

透明的读写分离、负载均衡

自定义Endpoint，隔离不同业务

### 存储分布式

100TB 存储空间

TB级数据，分钟级完成备份

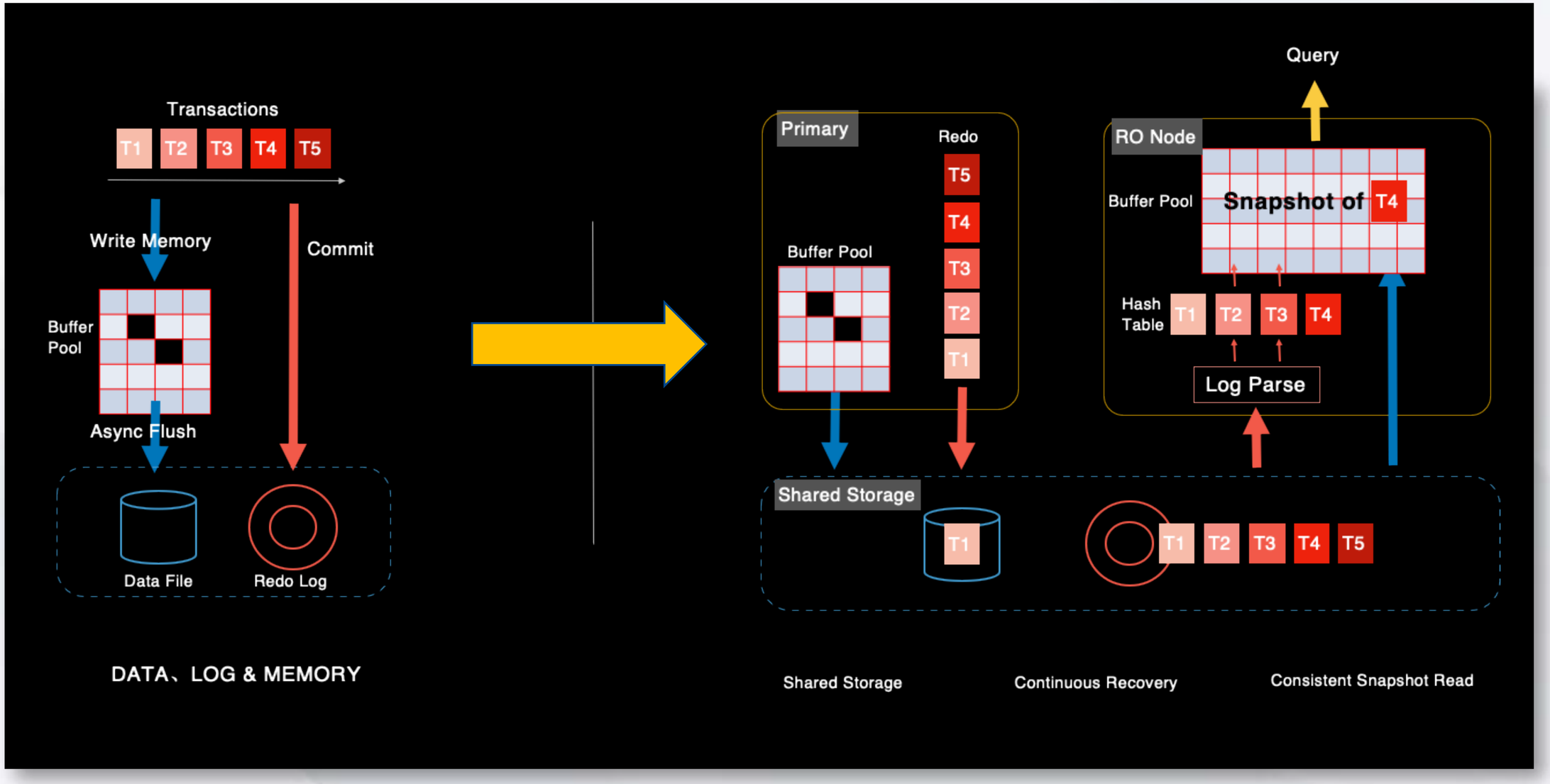
### libpfs+rdma/optane

6倍RDS MySQL性能

### 基于redo复制

只读实例，毫秒级的延迟

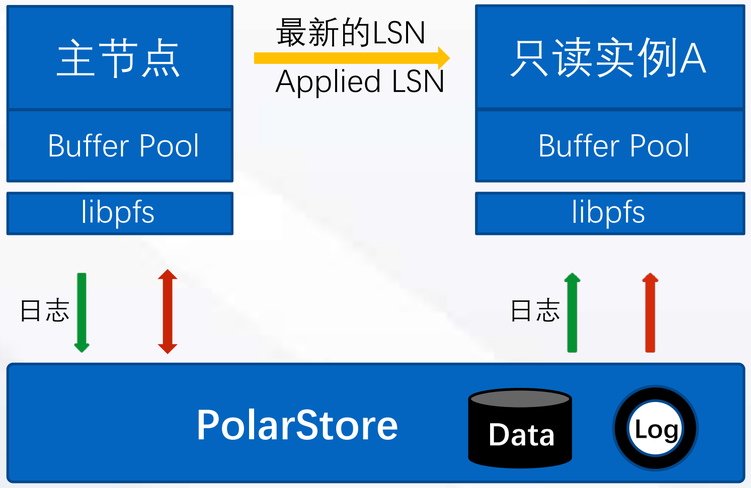
# POLARDB 底层实现：从逻辑复制到物理复制




- 使用InnoDB redo复制，默认关闭Binlog
- 共享一份datafile、redo log、undo log等

# POLARDB 底层实现：RW与RO节点的通信

## 保持只读节点Buffer Pool数据一致




- 只读节点需要不断的从Log文件中读取最新的redo日志
- 只读节点读取redo后：
  如果BP中有这些Page则，立刻更新
  如果BP中**没**有这些Page，则读取page 并应用redo
- 根据Applied LSN，可以删除所有之前的内存中的redo
- 只读节点上多个Log Apply Threads可以加速redo apply

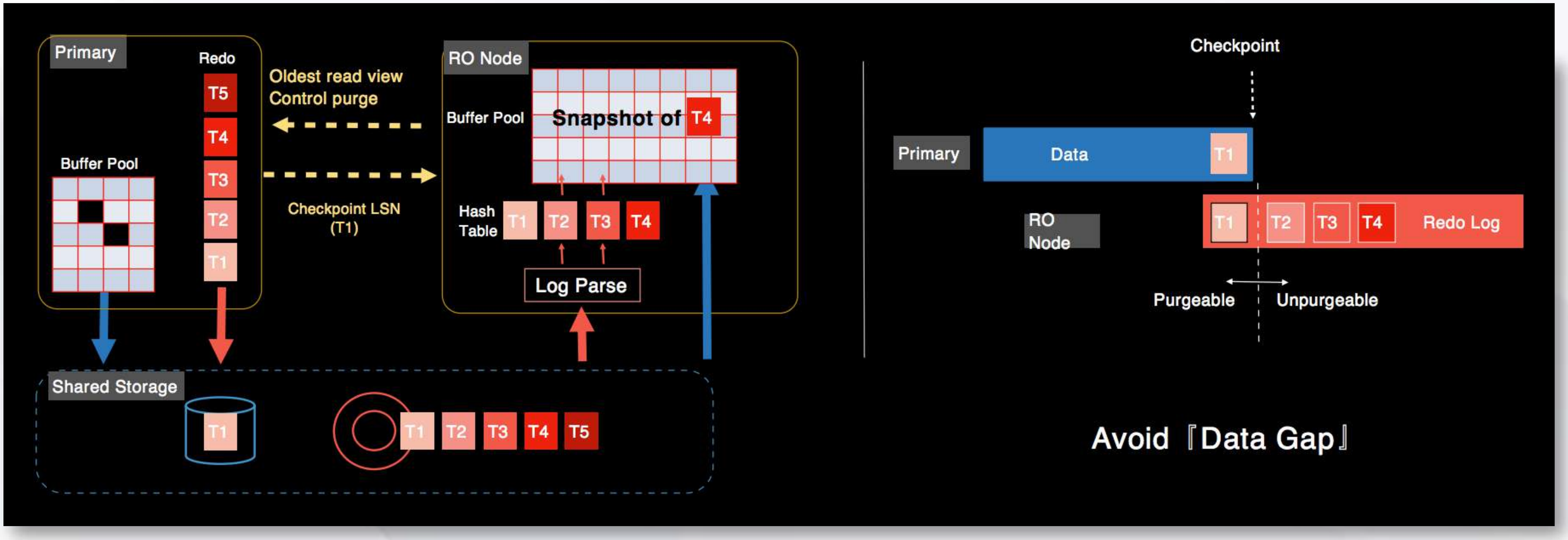
## 备/RO节点Buffer Pool更新

POLARDB：主节点通知 | 读取redo | BP 更新

传统主备架构：主binlog推送 | 存放relay log | 读取relay log | 解析binlog | 执行计划生成 | 存储引擎执行 | BP 更新

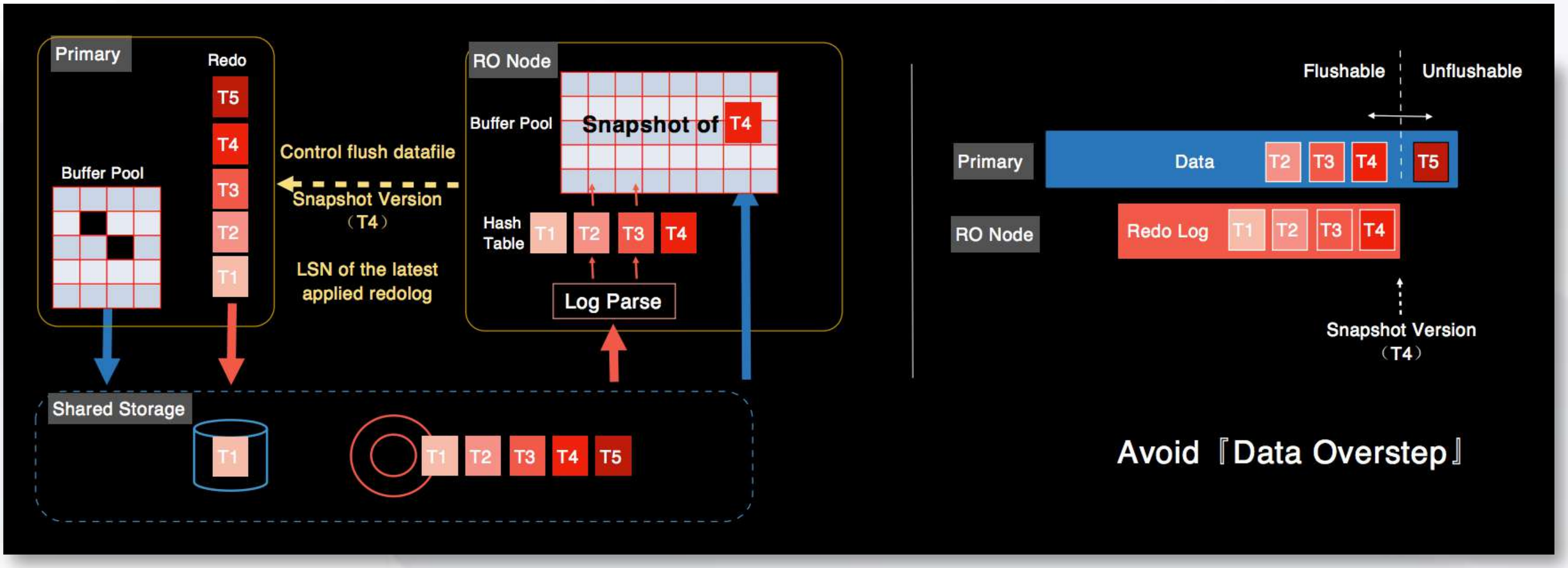
# POLARDB 底层实现：RW与RO节点的通信

主库上undo log的purge需要考虑备库上read view的情况



# POLARDB 底层实现：RW与RO节点的通信

主库的checkpoint需要考虑备库的redo apply进度

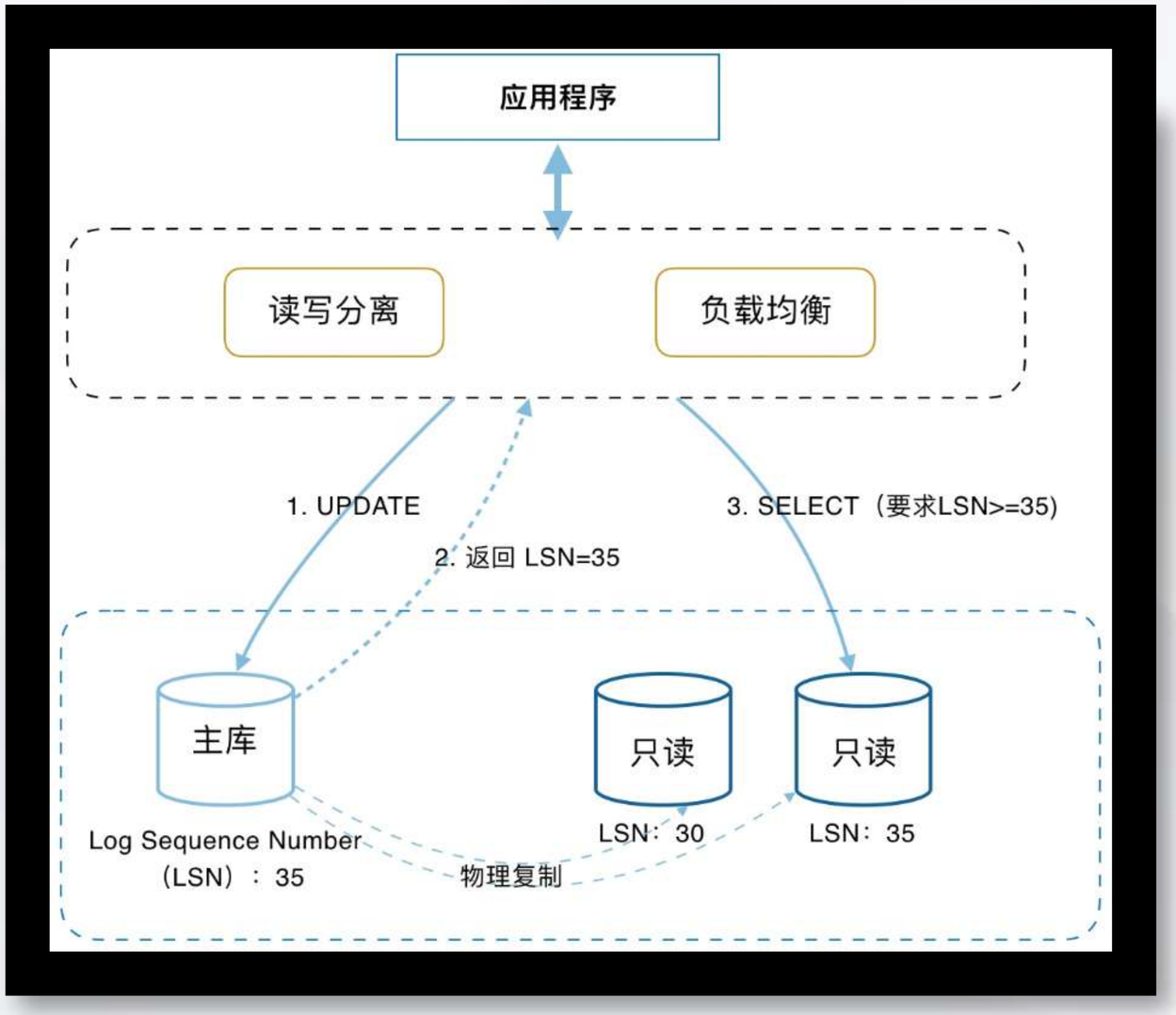


# POLARDB 底层实现：Proxy与RO/RW的通信

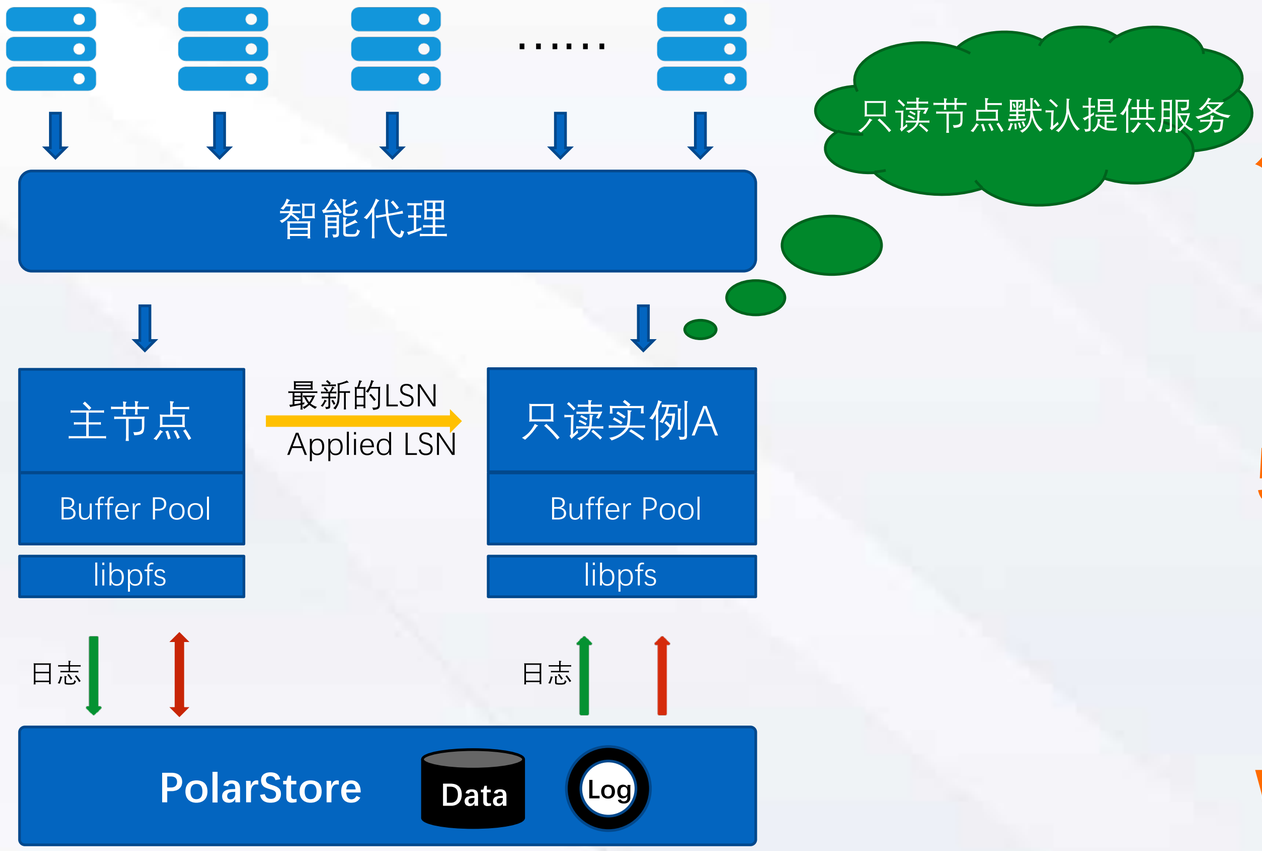
只读节点的会话一致性

解决“**读不到最新数据**”的问题

```
connection.query
{
  UPDATE user SET name= 'Jimmy' WHERE id=1;
  COMMIT;
  SELECT name FROM user WHERE id=1;
  // name is Jimmy
}
```



# POLARDB 产品架构：只读实例




## 低成本

- 备节点默认提供只读服务（集群地址）注

## 低延迟，备库数据更一致

- 通过InnoDB redo复制，并行apply，保障高效与一致
- DDL仅在主库执行一次，不会导致备库延迟

## 5分钟新增只读节点，流量自动分发

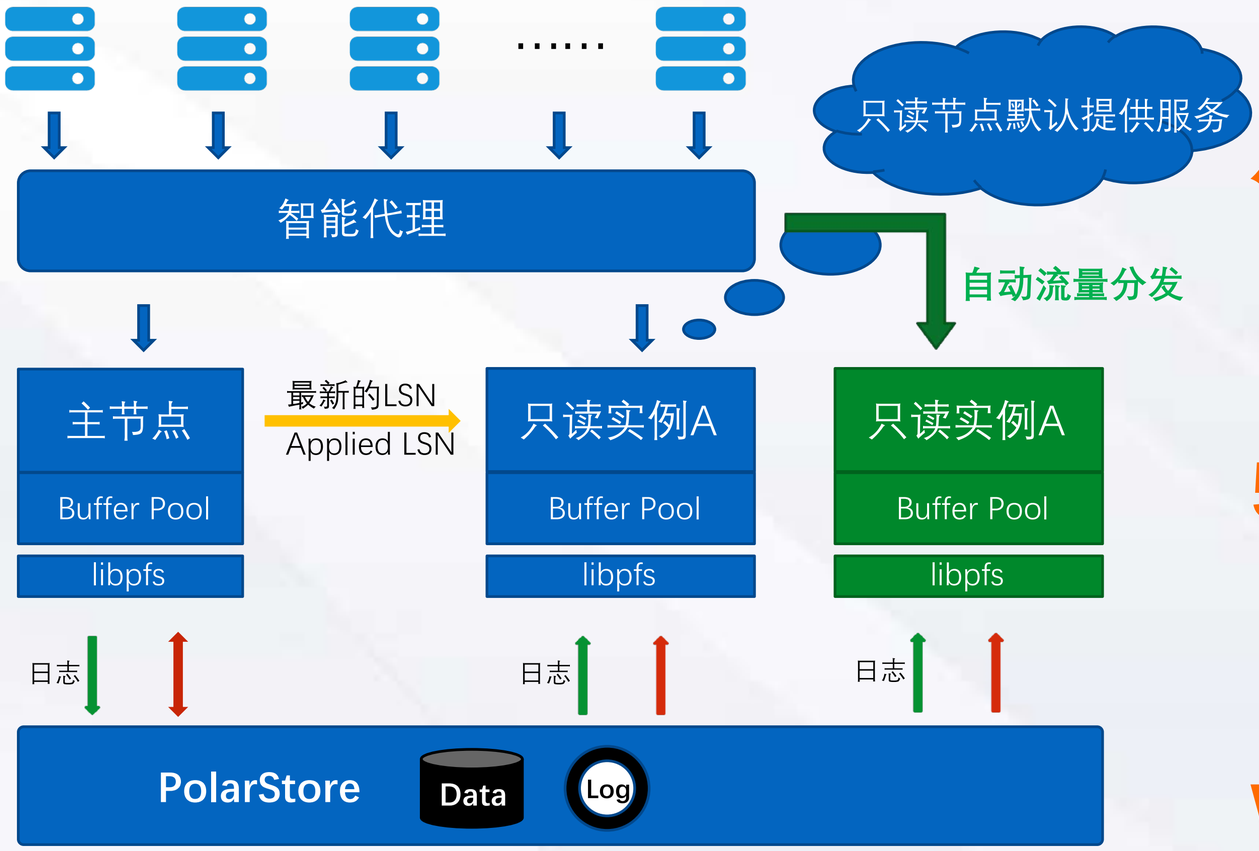
- 共享存储架构下，快速扩容只读节点，应对业务高峰
- 智能代理根据负载，自动分发只读业务流量

## Warmed up，切换后快速应对业务

- 只读节点提供服务，Buffer Pool天然被warmed up

注：“主地址”总是指向主节点

# POLARDB 产品架构：只读实例（续）




## 低成本

- 备节点默认提供只读服务（集群地址）注

## 低延迟，备库数据更一致

- 通过InnoDB redo复制，并行apply，保障高效与一致
- DDL仅在主库执行一次，不会导致备库延迟

## 5分钟新增只读节点，流量自动分发

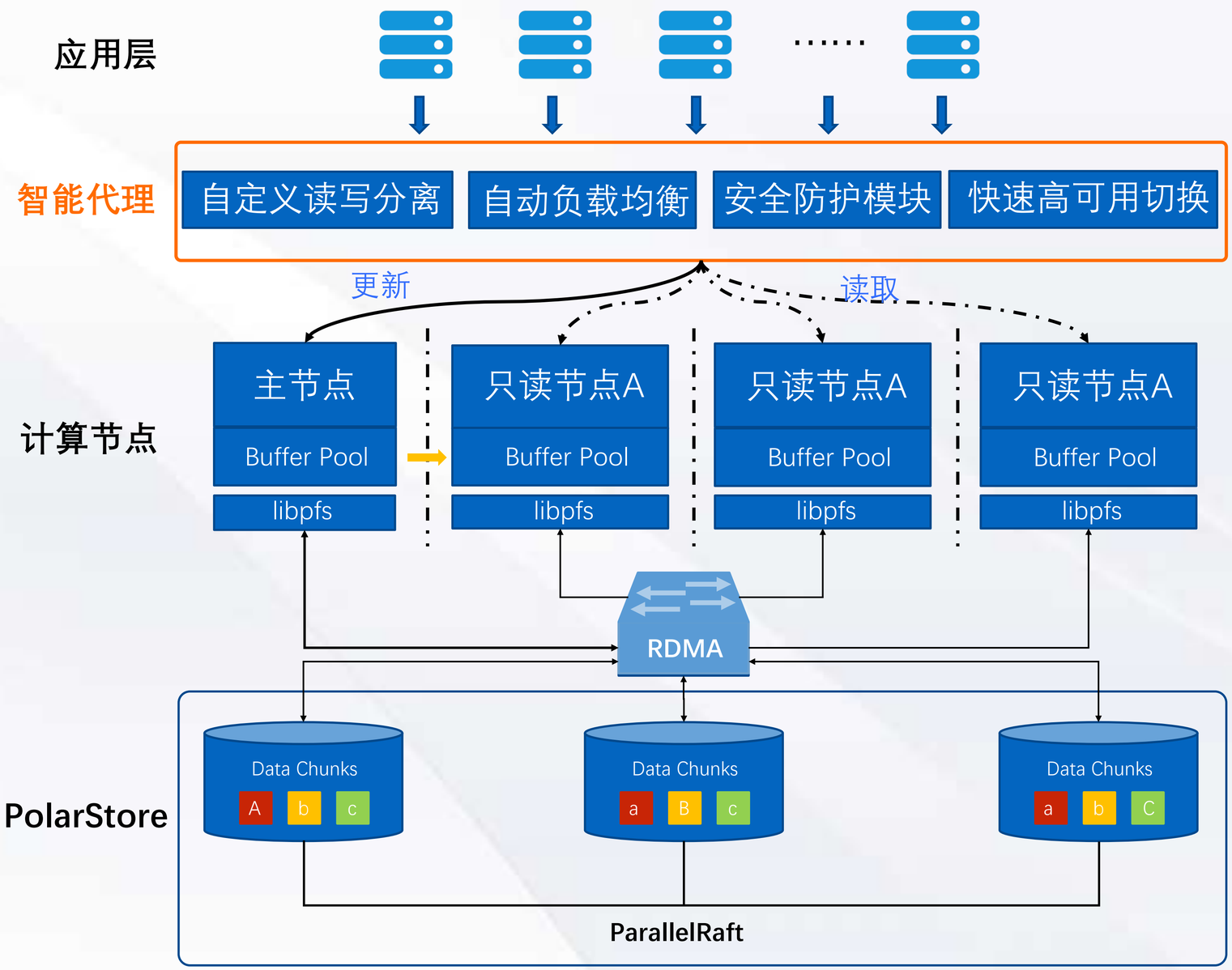
- 共享存储架构下，快速扩容只读节点，应对业务高峰
- 智能代理根据负载，自动分发只读业务流量

## Warmed up，切换后快速应对业务

- 只读节点提供服务，Buffer Pool天然被warmed up

注：“主地址”总是指向主节点

# POLARDB 产品架构：只读实例（续）




- 默认自动负载均衡

  集群地址具备负载均衡（根据active session）、高可用等能力。

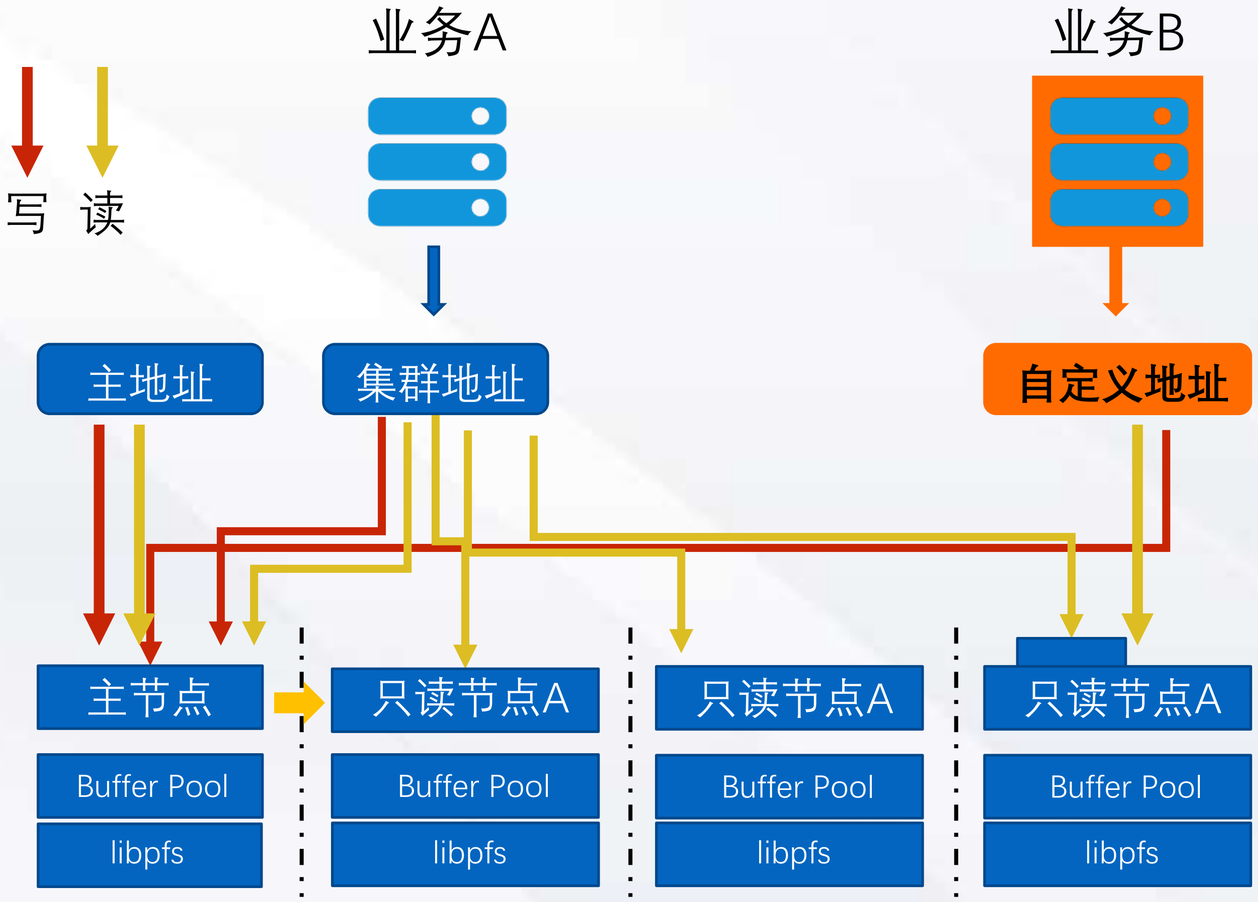
- 可以自定义Endpoint

  让某些只读节点只做特定用途，不影响主路径业务。例如某些分析型的查询

- 只读节点具备强一致的能力

```
connection.query
{
  UPDATE user SET name=  ‘Jimmy’  WHERE id=1;
  COMMIT;
  SELECT name FROM user WHERE id=1;
  // name is Jimmy
}
```

# POLARDB 产品架构：只读实例（续）



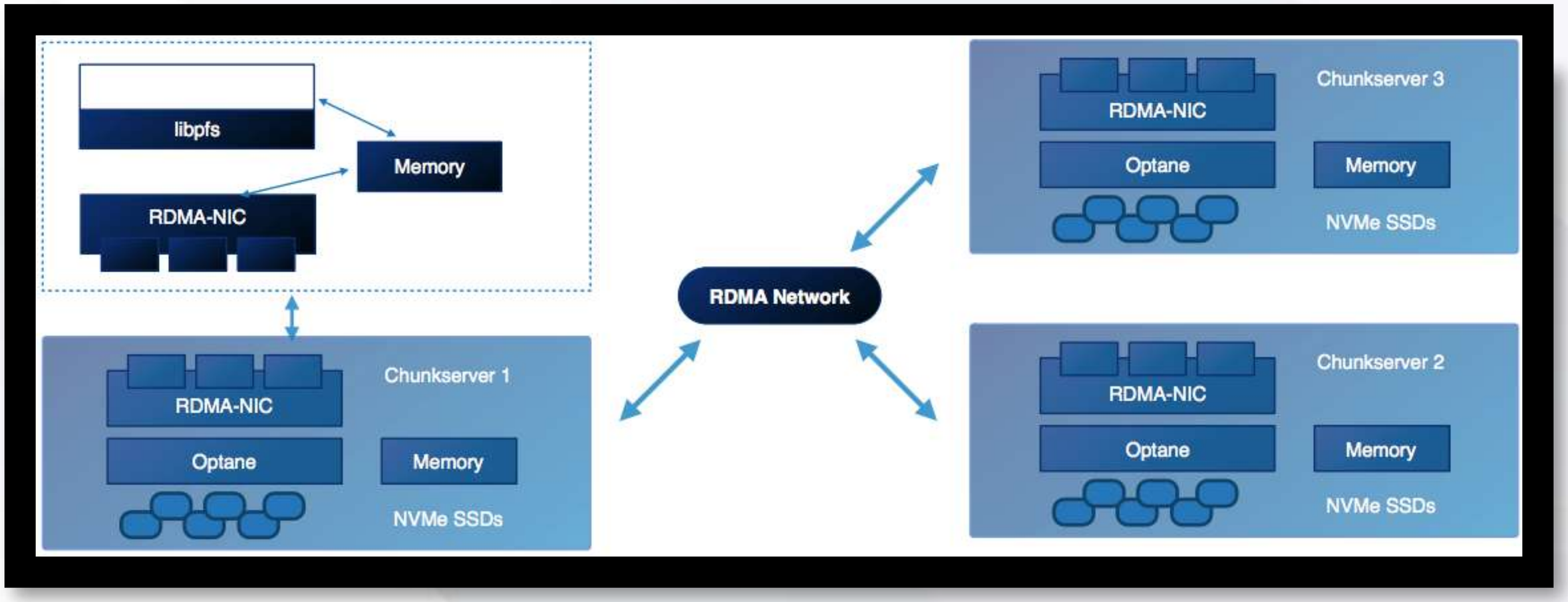

集群地址可实现**自动负载均衡**

自定义地址可**灵活配置只读节点用途**

# POLARDB 存储计算分离的挑战

数据库是延迟敏感的系统

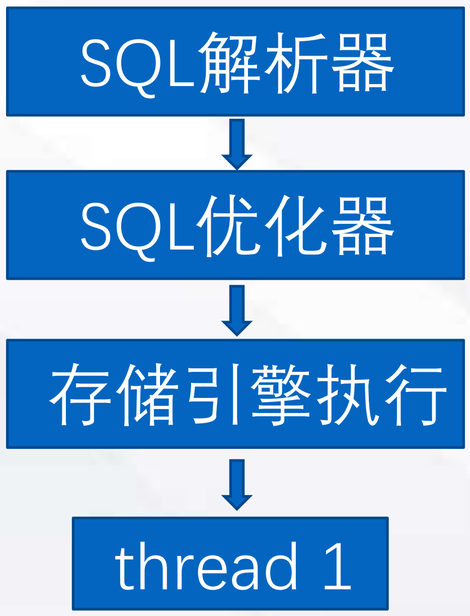
RDMA/libpfs　Intel Optane/NVMe SSD　25G以太网



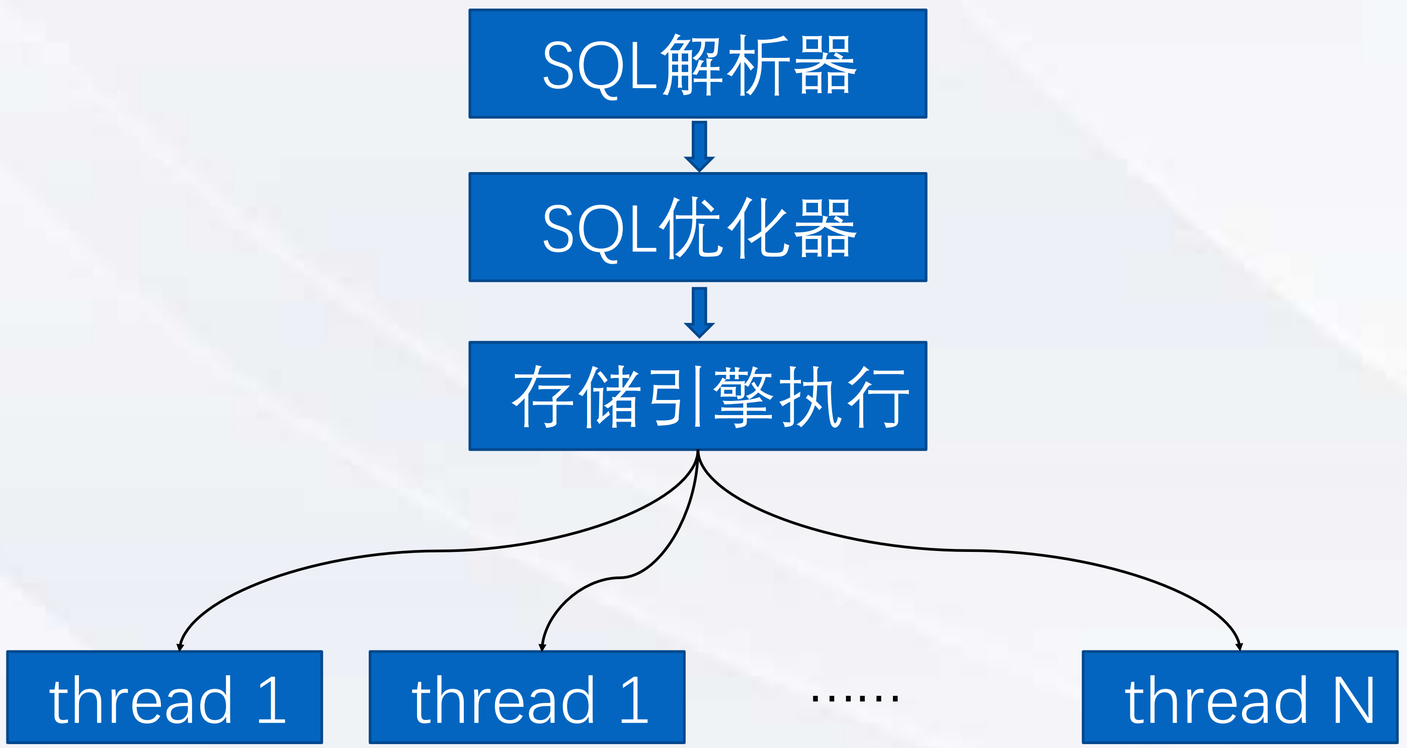
# POLARDB 并行查询

```
SELECT count(*) FROM production.product;
```

MySQL官方版本



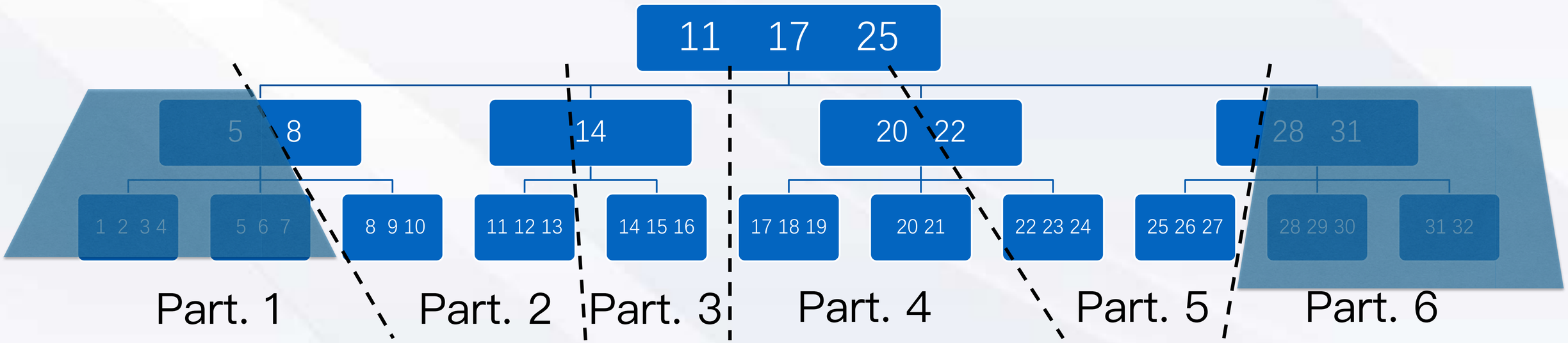

POLARDB 并行查询



# POLARDB 并行查询

```
SELECT count(*) FROM production.product;
```

内核中对对InnoDB的B-Tree进行分区，每个worker处理一部分数据，然后汇总



# POLARDB 并行查询

```
mysql> select /*+ SET_VAR(max_parallel_degree=0) */
           count(1)
       from
           t1
       where
           ASCII(SUBSTRING(md5(sha1(nick)) FROM 16 FOR 1)) > 85;
```

| 并发 | Query执行时间（秒） |
|---|---|
| 0 | 43.94 |
| 2 | 25.10 |
| 4 | 13.98 |
| 8 | 9.07 |
| 16 | 8.34 |
| 32 | 5.17 |
| 48 | 5.16 |

测试实例规格：

节点规格：32核256GB（独享）
数据库引擎：MySQL 8.0
地域：华东 1（杭州）

使用32线程并发后，性能提升**8.5**倍

```
mysql> show create table t1\G
*************************** 1. row ***********************
Table: t1
Create Table: CREATE TABLE `t1` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nick` varchar(32) DEFAULT NULL,
  `age` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)) ENGINE=InnoDB AUTO_INCREMENT=46353005
DEFAULT CHARSET=utf8
1 row in set (0.00 sec)

mysql> select count(1) from t1;
+----------+
| count(1) |
+----------+
| 44424108 |
+----------+
```

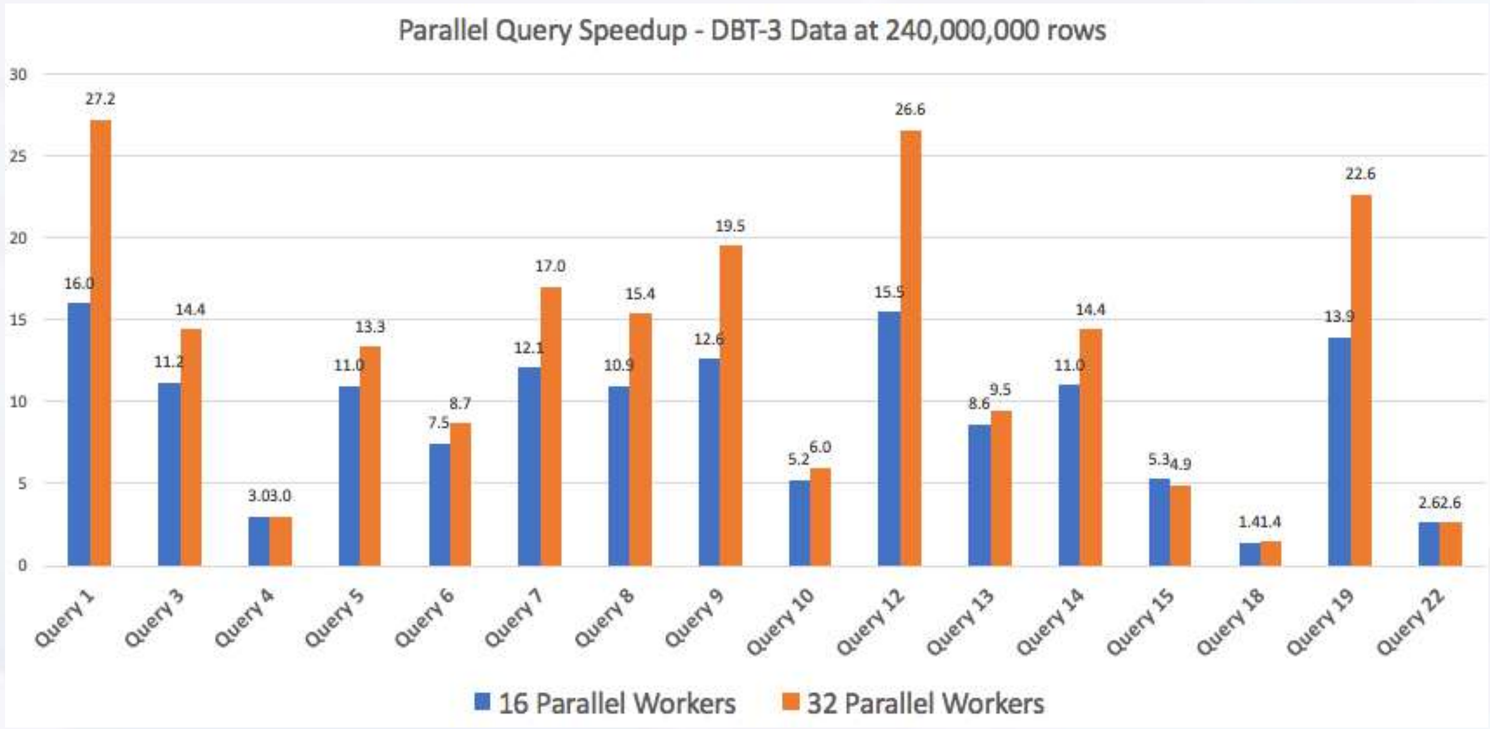
# POLARDB 并行查询

TCP-H压力下，22条Query中

**72%**（16条）可以被加速，

最高性能提升**27倍**。

关于测试：
使用工具DBT-3（参考），tpc-h压力进行测试，测试数量为2.4亿记录。



# POLARDB 并行查询

```
mysql> select /*+ SET_VAR(max_parallel_degree=4) */ count(1)
       from t1 where ASCII(SUBSTRING(md5(sha1(nick)) FROM 16 FOR 1)) > 85;
```

执行计划查看：

```
mysql> explain select /*+ SET_VAR(max_parallel_degree=4) */ count(1)
       from t1 where ASCII(SUBSTRING(md5(sha1(nick)) FROM 16 FOR 1)) > 85;

+----+-------------+----------+------------+------+------+----------+-----------------------------------------------+
| id | select_type | table    | partitions | type | key  | rows     | Extra                                         |
+----+-------------+----------+------------+------+------+----------+-----------------------------------------------+
|  1 | SIMPLE      | <gather2>| NULL       | ALL  | NULL | 40240616 | Merge sort                                    |
|  2 | SIMPLE      | t1       | NULL       | ALL  | NULL | 10060154 | Parallel scan (4 workers); Using where; Using filesort |
+----+-------------+----------+------------+------+------+----------+-----------------------------------------------+
```

# POLARDB 并行查询

主要参数与状态值：

```
mysql> show global variables like '%parallel%';
+-----------------------------------------+---------------+
| Variable_name                           | Value         |
+-----------------------------------------+---------------+
| force_parallel_mode                     | OFF           |
| innodb_parallel_scan_estimate_max_pages | 128           |
| max_parallel_degree                     | 32            |
| max_parallel_workers                    | 16            |
| parallel_partition_factor               | 100           |
+-----------------------------------------+---------------+
8 rows in set (0.00 sec)

mysql> show global status like '%parallel%';
+-------------------------------+-------+
| Variable_name                 | Value |
+-------------------------------+-------+
| Parallel_workers_created      | 176   |
| Total_running_parallel_workers | 0     |
+-------------------------------+-------+
2 rows in set (0.00 sec)
```

→ 忽略优化器执行计划选择，尽量强制走并行执行

→ ……

→ 单个查询的最大并行度

→ 实例中，允许的最大worker线程数，默认C*4

→ ……

→ 总共创建了多少个worker现成

→ 当前系统中并行运行的worker有多少

# POLARDB 典型场景


**TB级数据**

存储容量超过2TB，但无法做分库分表

不希望迁移历史数据

**快速弹性**

底层本的应对，突发流量高峰，类似于周末运营、双十一等活动上线，压力突增，紧急进行扩容

**读写分离**

线上压力激增，快速增加只读节点，保障核心业务稳定

备库天然可读，降低整体业务成本

**数据零丢失**

业务核心数据，保障强一致且不丢失

支付、账务等与钱相关的业务

**复杂查询**

在线业务中的ad-hoc复杂查询

加速部分因为业务数据上涨导致的慢查询

# POLARDB 极致弹性

10 TB数据量 **5分钟**完成新增只读节点；**15分钟**完成实例升降级

**临时升级配置**，最低成本应对业务高峰


**88核 710GB**，超大规格应对业务高速增长

# 如何使用 POLARDB

## 不停机的业务迁移


支持全、增量迁移　　可视化操作

支持业务平滑迁移　　可反向迁移

### 数据传输

- 从RDS for MySQL迁移至POLARDB for MySQL：文档
- 从本地MySQL迁移至POLARDB for MySQL：文档
- 从AWS Aurora迁移至POLARDB for MySQL：文档
- 从ECS上的自建MySQL迁移至POLARDB for MySQL：文档

### 一键升级RDS for MySQL到POLARDB for MySQL：文档

- 物理迁移，更快速

# POLARDB 完整的产品生态

# POALRDB一体机发布

1152 vCPU

9TB 内存

120TB 闪存

1000+ 兼容项

ADAM

数据库和应用迁移 ADAM

应用评估、迁移与实施

数据传输 DTS

一键无缝迁移上云

POLARDB

阿里云市场

丰富的第三方合作生态

AnalyticDB

支持百TB数据库实时分析

关注“**阿里数据库技术**”公众号

后台回复“**嘉年华**”获取PPT资料