开源时序数据库 CeresDB 介绍

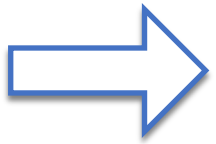
https://github.com/jiacai2050

目录

CeresDB 的设计目标——要解决的问题是什么？

主流的时序数据库

- InfluxDB
- Prometheus
- Opentsdb
- VictoriaMetrics



问题

- 时间线高基数问题
- 社区的分布式方案不够完备

蚂蚁集团的内部实践

- Antmonitor 的底层存储

CeresDB 的设计目标——时序数据模型

时序数据

- 基于时间的一系列数据点的集合，在有时间的坐标系中将这些数据点连成线。
- 从时间维度可以做成多纬度报表，揭示其趋势性、 规律性、异常性。

术语

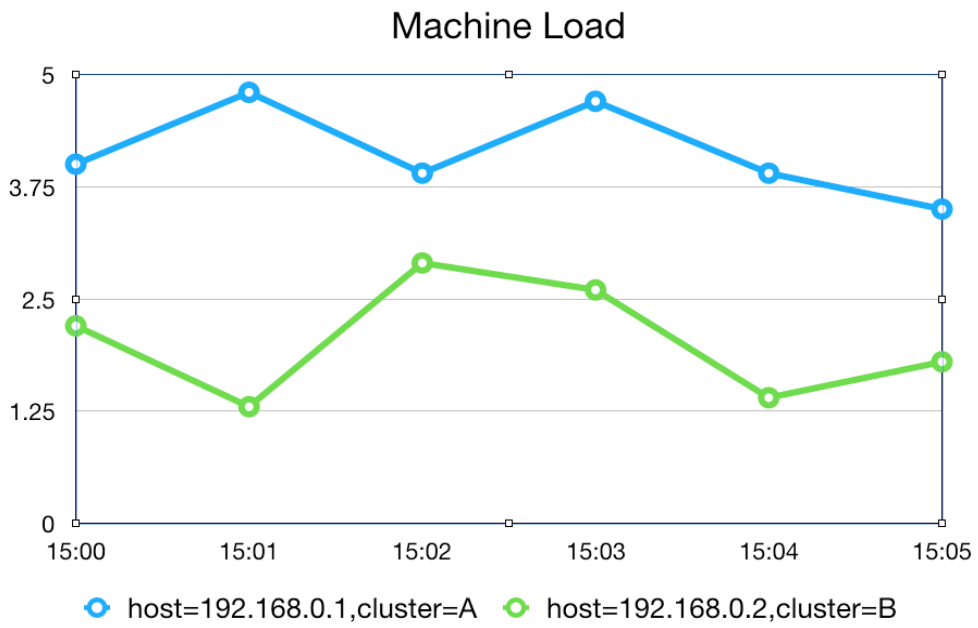
Metric: 指标名/表名

Timestamp: 数据必带时间戳

Tags: 用于标识一根时间线(实体)

Fields: 用于存储数据

Timeseries: 所有 Tags 组合形成的唯一标识



timeseries

```
host=192.168.0.1,cluster=A | timestamp=2020-06-26 15:00 ==> 4.0
host=192.168.0.1,cluster=A | timestamp=2020-06-26 15:01 ==> 4.8
host=192.168.0.1,cluster=A | timestamp=2020-06-26 15:02 ==> 3.9
host=192.168.0.1,cluster=A | timestamp=2020-06-26 15:03 ==> 4.7
host=192.168.0.1,cluster=A | timestamp=2020-06-26 15:04 ==> 3.9
host=192.168.0.1,cluster=A | timestamp=2020-06-26 15:05 ==> 3.5
host=192.168.0.2,cluster=B | timestamp=2020-06-26 15:00 ==> 2.2
host=192.168.0.2,cluster=B | timestamp=2020-06-26 15:01 ==> 1.3
host=192.168.0.2,cluster=B | timestamp=2020-06-26 15:02 ==> 2.9
host=192.168.0.2,cluster=B | timestamp=2020-06-26 15:03 ==> 2.6
host=192.168.0.2,cluster=B | timestamp=2020-06-26 15:04 ==> 1.4
host=192.168.0.2,cluster=B | timestamp=2020-06-26 15:05 ==> 1.8
```

CeresDB 的设计目标——经典时序数据模型的缺陷

数据组织

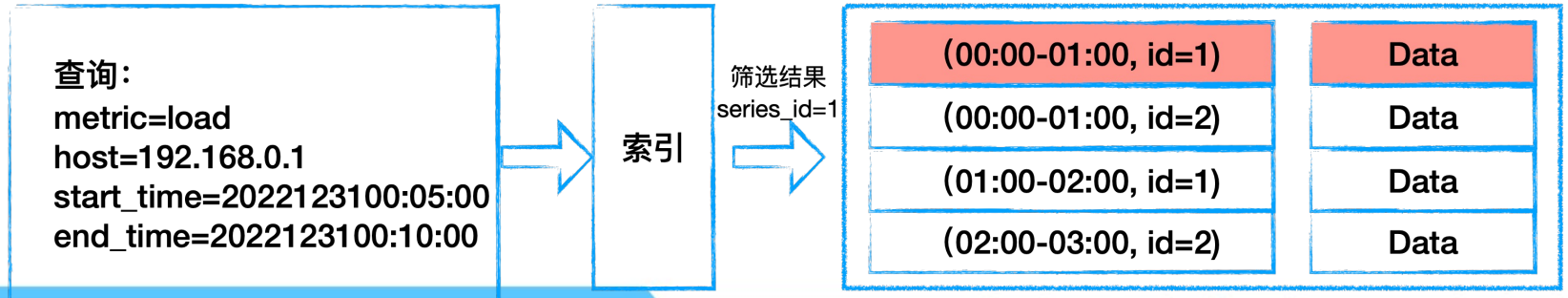
- 根据 Tag 构建出倒排索引
- 数据按照时间段划分
- 时间段内数据按照时间线组织

索引膨胀

- 多个高基数的 Tag 会导致非常夸张的 Tag 组合数
- 写入性能差：索引复杂度高
- 查询性能差：索引有效性低

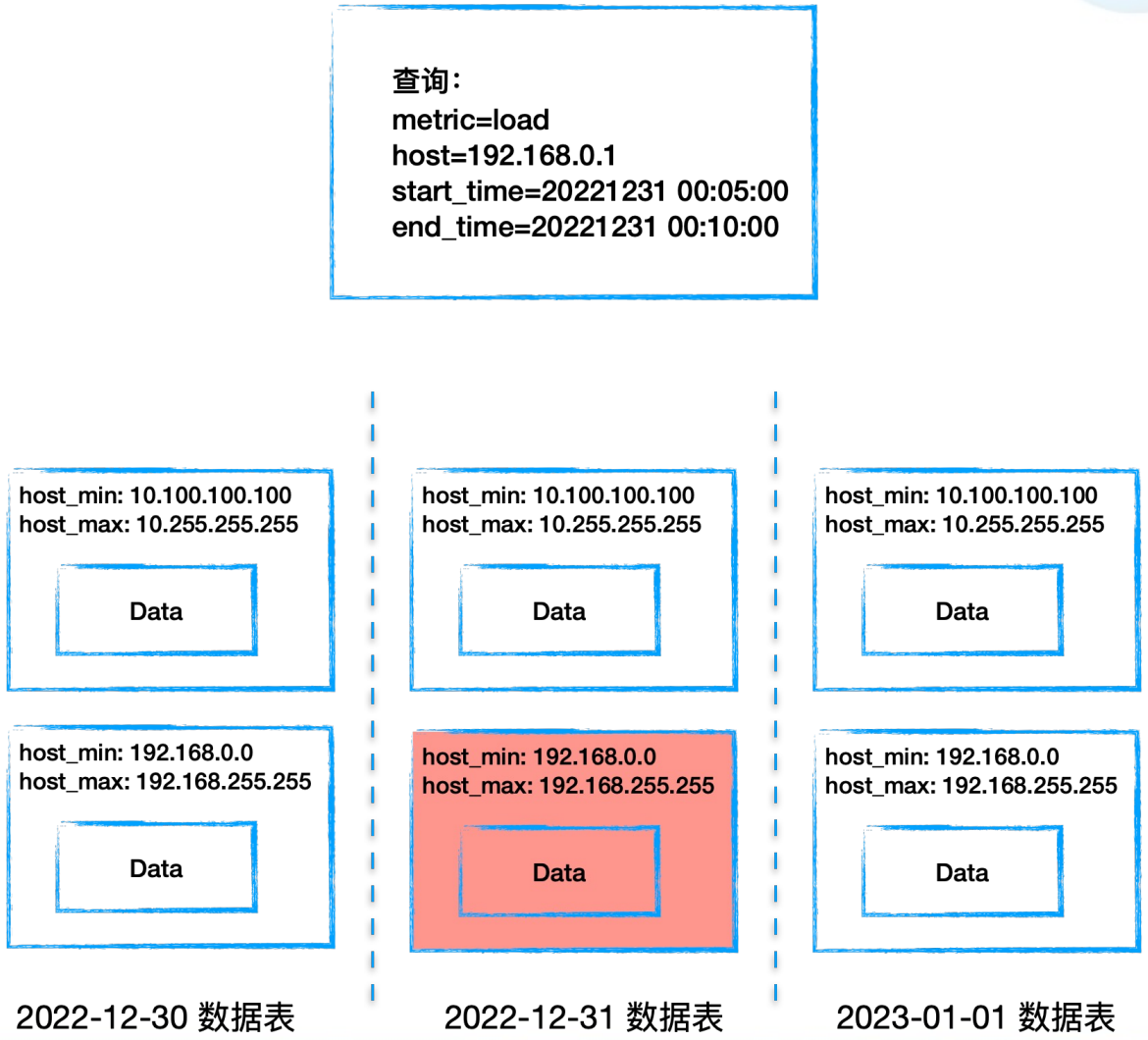
可能出现问题的场景

- 短生命周期的 pod 指标任务监控
- 业务监控：时间上不持续，较稀疏



CeresDB 的设计目标——解决方案

- 列式存储 + 混合存储
- 分区扫描 + 剪枝 + 倒排索引(可选)



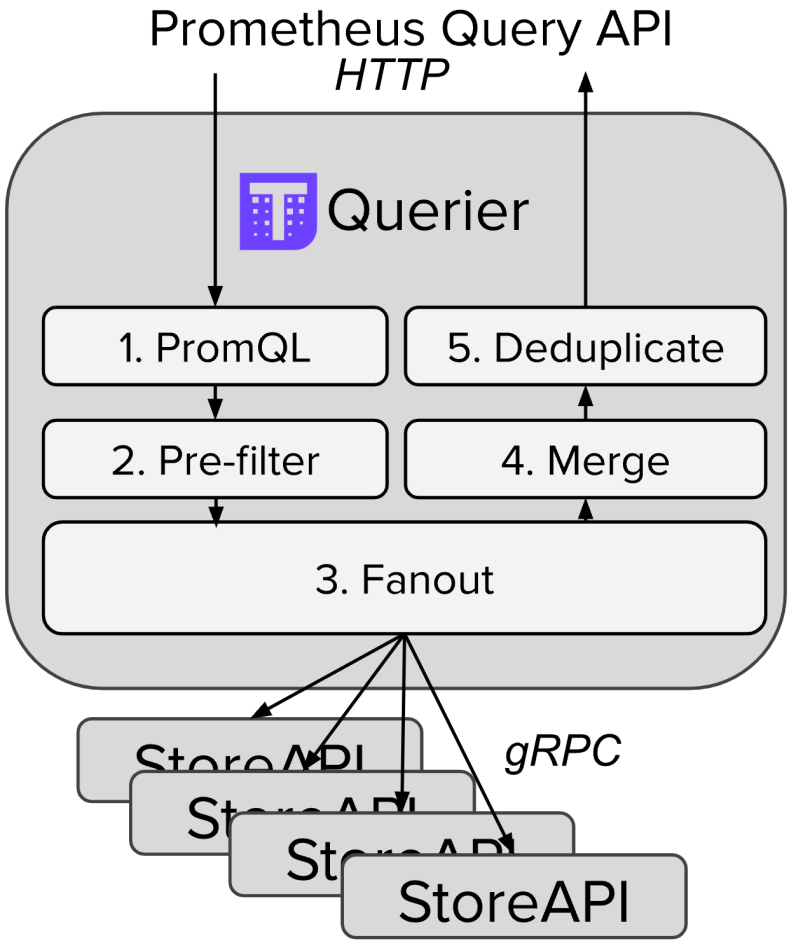
CeresDB 的设计目标——不完善的分布式方案

主流时序数据库的分布式方案

- InfluxDB：商业版
- Prometheus：Thanos

问题

- 商业版未开源
- 第三方构建，存在性能损耗



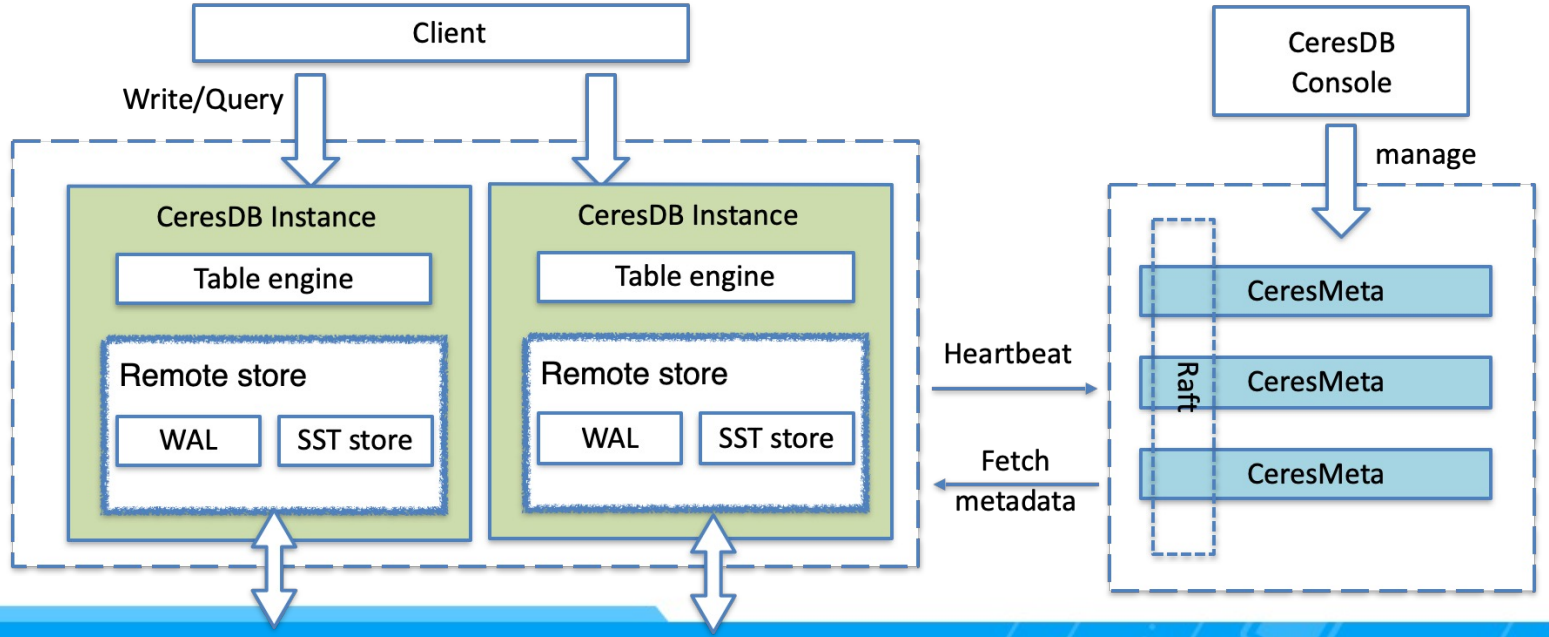
CeresDB 的设计目标——分布式方案

CeresDB 的分布式方案

- 云原生的分布式时序数据库
- 计算存储分离的设计架构

分布式特性

- 服务高可用、高可靠
- 数据具备容灾
- 支持水平扩容



CeresDB 的设计目标——总结

解决时间线高基数问题

- 能高效处理好 APM 型时序数据
- 同时能高效处理好高基数时间线场景

提供原生分布式方案

- 大规模部署
- 提供高可用、高可靠的服务
- 支持水平扩容
- 支持高效的分布式查询

1.0 特性

性能

- 向量化查询引擎（基于 Arrow + Datafusion）
- 高效剪枝（ XOR 过滤器）

分布式

- 计算存储分离，支持（OSS、OBKV、Kafka）
- 支持 HASH 分区表

周边生态

- 支持 SQL
- Java、Python、Go、Rust SDK
- 支持 Prometheus 协议
- 支持 Prometheus + Grafana 搭建自监控

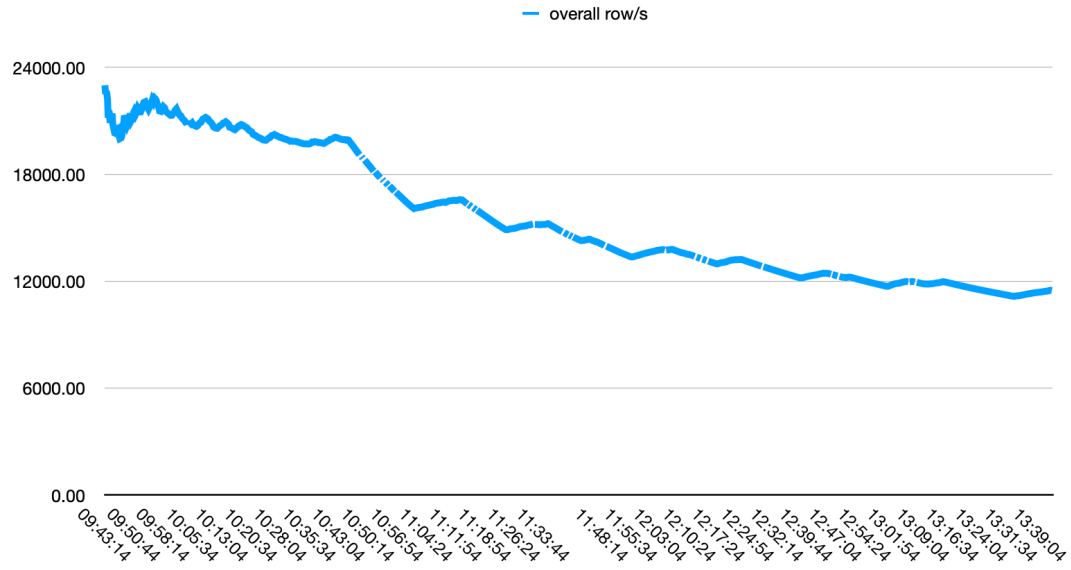
性能测试

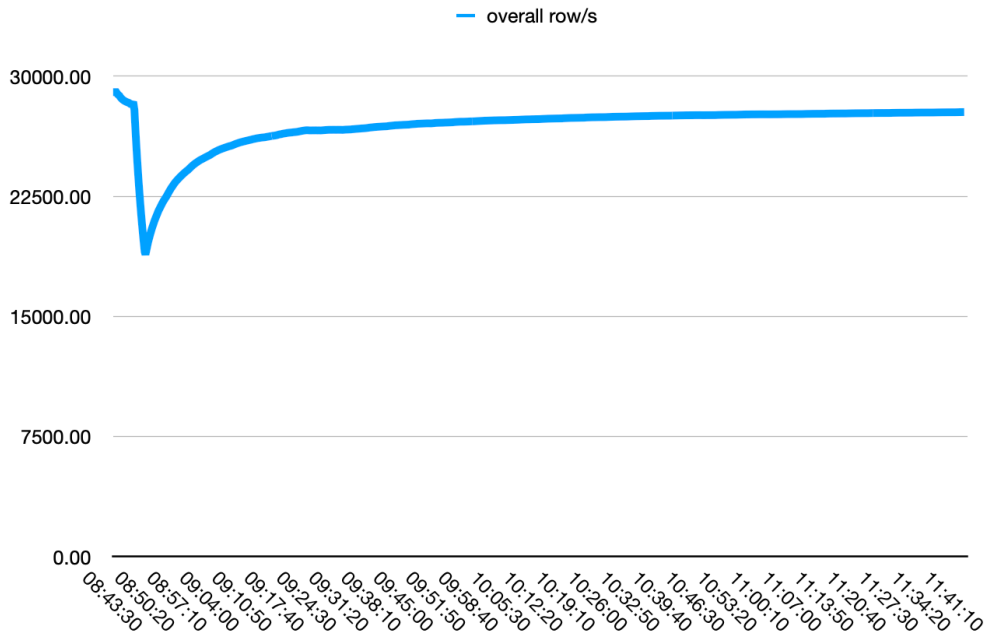
- **数据模型**
  - 10 个 Tag
  - 10 个 field
- **时间线（Tags 组合数）**
  - 100w 量级
- **软件**
  - TSBS
  - InfluxDB 1.8.5
  - CeresDB 1.0

```
CREATE TABLE `cpu` (
  `tsid` uint64 NOT NULL,
  `timestamp` timestamp NOT NULL,
  `hostname` string TAG,
  `region` string TAG,
  `datacenter` string TAG,
  `rack` string TAG,
  `os` string TAG,
  ......
  `usage_user` double,
  `usage_system` double,
   ....
PRIMARY KEY (tsid, timestamp),
  timestamp KEY (timestamp))
```

写入性能

根据文档，当时间线超过 1000w 时，推荐部署商业版





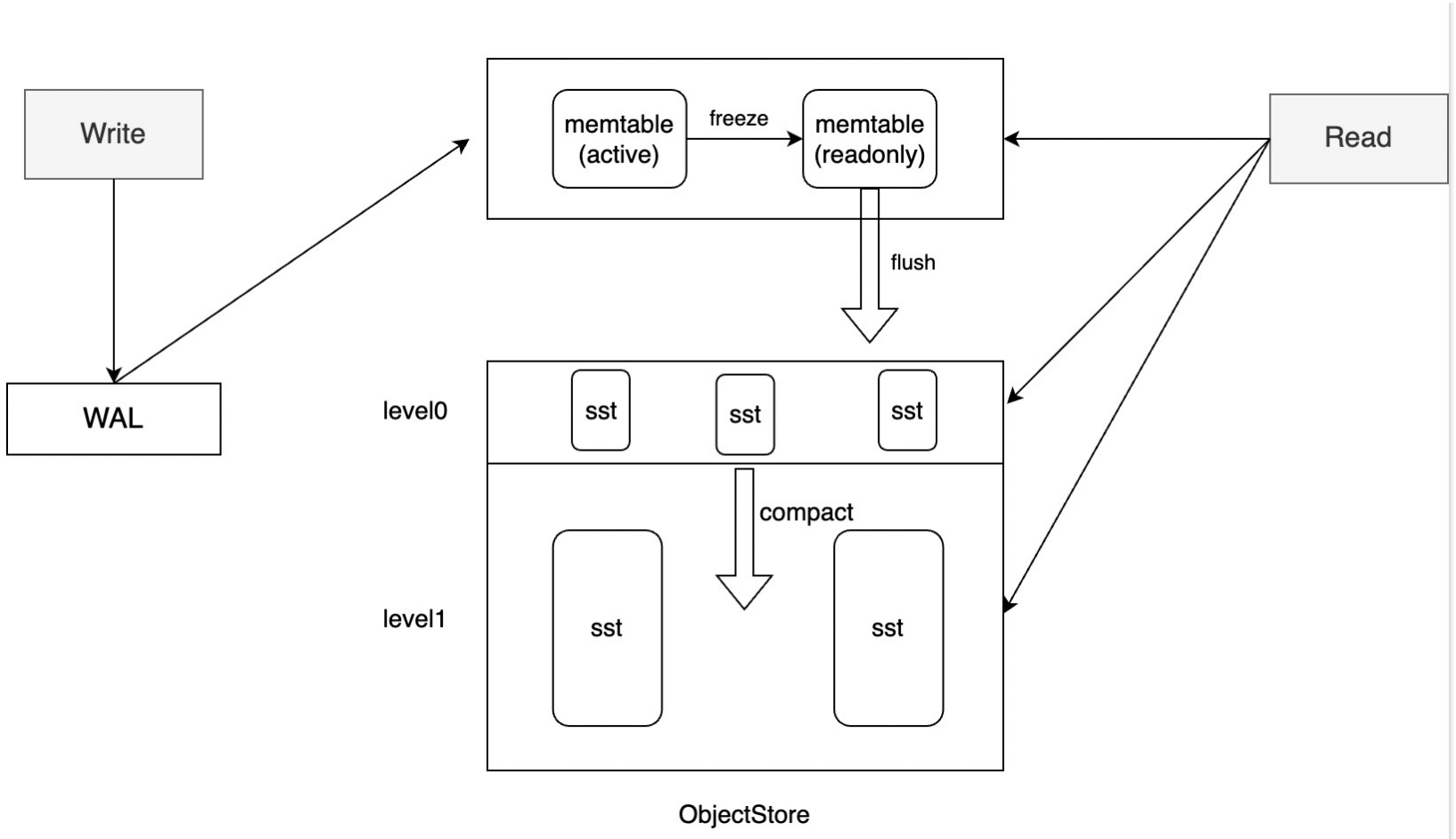
InfluxDB　　10个点/行　　CeresDB

查询性能

- 高筛选度条件（命中的数据较少，hostname=[8个]）
  - CeresDB：500ms、85ms
  - InfluxDB：15ms
  - 慢 5 倍
- 低筛选度条件（os=Ubuntu15.10）
  - CeresDB：15s
  - InfluxDB：6m43s
  - 快 26 倍

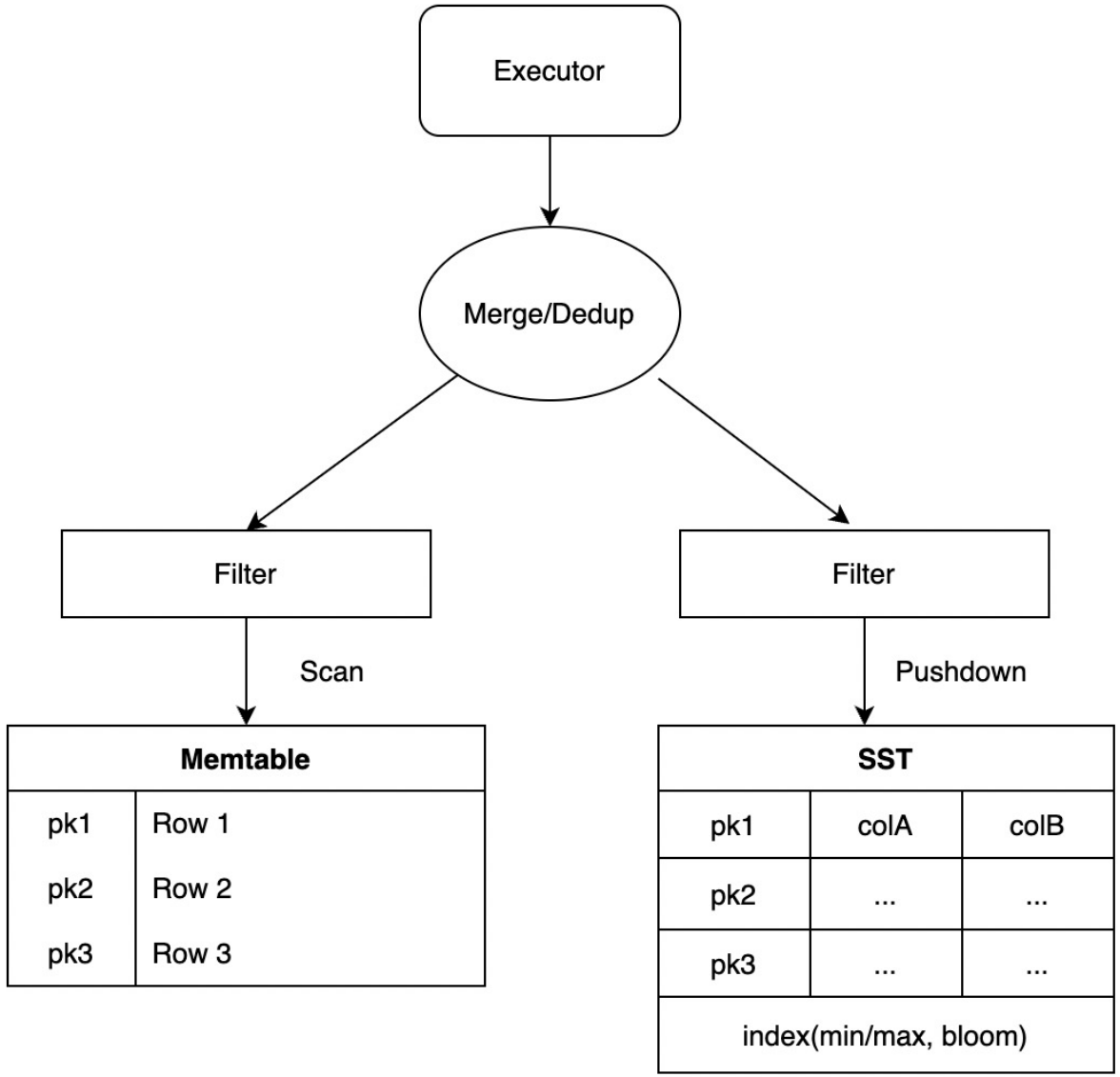
核心设计

- 类 LSM 存储系统
  - WAL
  - Memtable
  - SST



核心设计

- **Memtable: KV 模型**
  - 写入高效
- **SST: 列存模型**
  - 高压缩比
  - 更小的 IO 操作
  - 单字段的高效聚合



Roadmap

性能

- 探索新的存储格式
- 增强索引

分布式

- 自动负载均衡
- 增加可用性、可靠性

周边

- 生态兼容，包括 PromQL、InfluxQL、OpenTSDB
- 运维工具支持，包括 k8s 支持、自监控
- 开发者工具，包括数据导入导出

CeresDB 开源社区

- 代码：https://github.com/ceresdb/ceresdb
- 文档：https://docs.ceresdb.io/
- Slack
- 钉钉： 44602802

CeresDB

微信扫描二维码，关注我的公众号

THANKS / 感谢！