物联网时序大数据的高效处理

涛思数据联合创始人　李广

物联网的技术产业链

传感器
数据采集

通讯模组
边缘计算

云数据引擎
（存储·查询·计算）

分析·应用
系统

TalkingData

传统的实时数据库

为解决流程控制领域的实时/时序数据处理问题，从上世纪八十年代起，出现一批实时数据库，以美国的OSIsoft PI为代表，具有较高的数据处理能力，很好的解决了传统工业生产问题。

传统的实时数据库面临的挑战

在测点数暴涨、数据采集频次不断提高的大数据时代，传统实时数据库暴露出下列问题：

没有水平扩展能力，数据量增加，只能依靠硬件scale up

技术架构陈旧，使用磁盘阵列，还运行在windows环境下

数据分析能力偏弱，不支持现在流行的各种大数据分析接口

不支持云端部署，更无法支持PaaS

TalkingData

通用大数据方案的挑战：低效、复杂、高成本

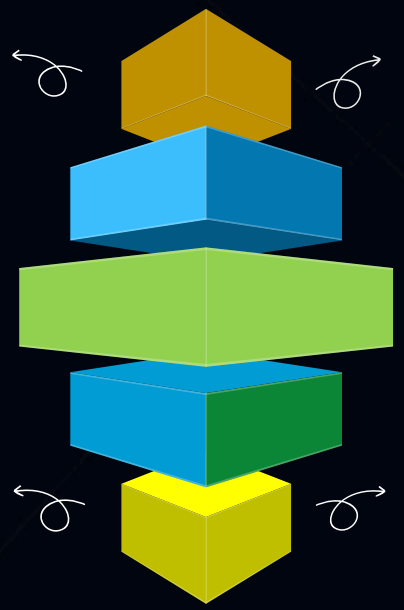
通常将开源的Kafka, Redis, Hbase, MongoDB, Cassandra, ES, Hadoop, Spark，Zookeeper等大数据软件拼装起来，利用集群来处理海量数据。

开发效率低

因牵涉到多种系统，每种系统有自己的开发语言和工具，开发精力花在了系统联调上，而且数据的一致性难以保证

运维复杂

每个系统都有自己的运维后台，带来更高的运维代价，出问题后难以跟踪解决，系统的不稳定性大幅上升



运行效率差

非结构化数据技术来处理结构化数据，整体性能不够，系统资源消耗大。因为多套系统，数据需要在各系统之间传输，造成额外的运行代价

应用推向市场慢

集成复杂，得不到专业服务，项目实施周期长，导致人力攀升，利润缩水

物联网、工业4.0数据特征：时序空间数据

采集的数据量巨大，但有典型特征:

1. 所有采集的数据都是时序的
2. 数据都是结构化的
3. 一个采集点的数据源是唯一的
4. 数据很少有更新或删除操作
5. 数据一般是按到期日期来删除的
6. 数据以写操作为主，读操作为辅
7. 数据流量平稳，可以较为准确的计算
8. 数据都有统计、聚合等实时计算操作
9. 数据一定是指定时间段和指定区域查找的
10. 数据量巨大，一天的数据量就超过100亿条

TalkingData

TDengine

TalkingData

TDengine提供的功能

消息队列

自带消息队列

缓存

所有设备最新记录实时返回

数据库

实时数据库，历史数据库，操作合一透明

流式计算

对一个或多个数据流实时聚合计算

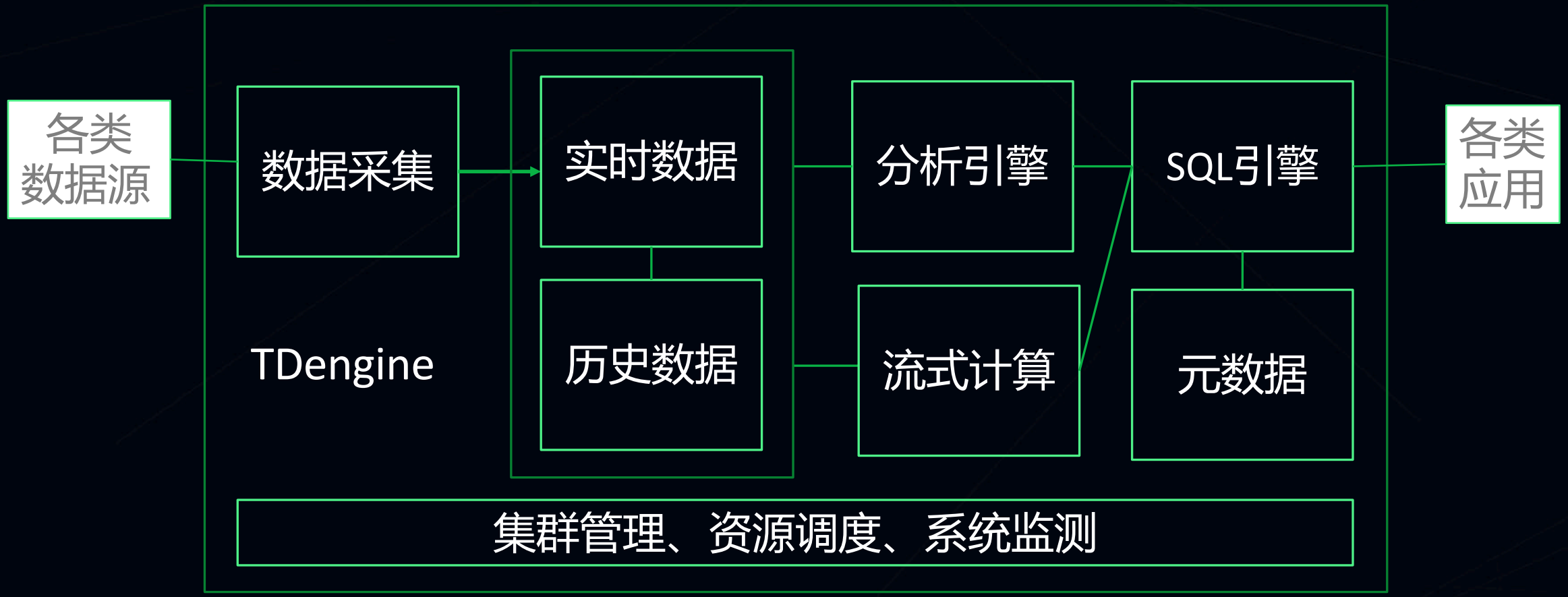
数据订阅

最新的数据可实时推送到应用

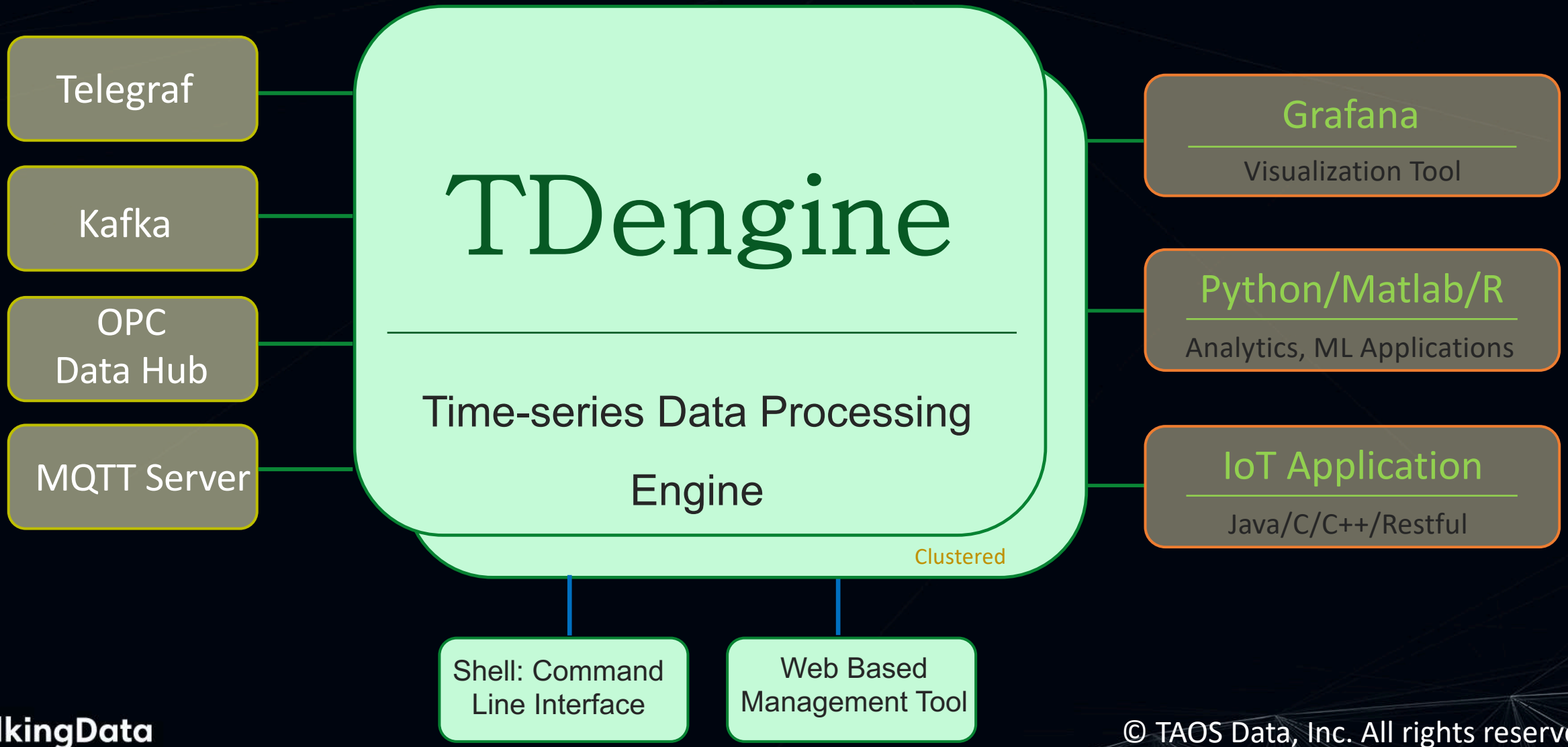
物联网数据的全栈解决方案，物联网大数据平台

TalkingData

TDengine 内部架构图



TDengine 生态图

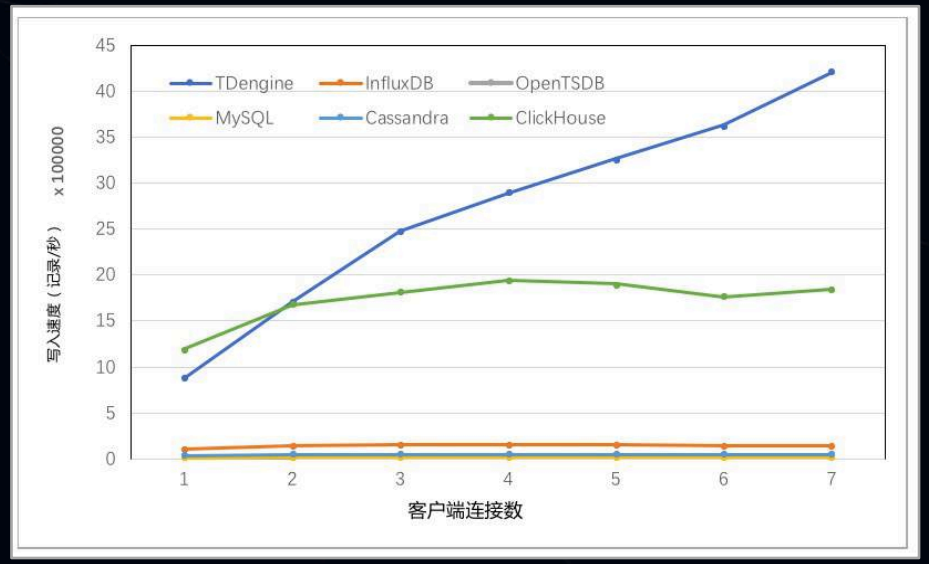


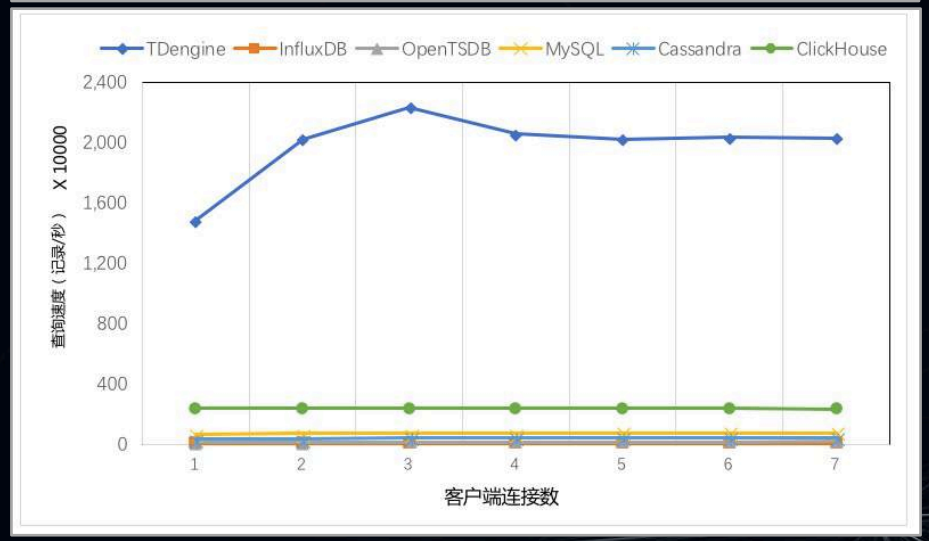
TalkingData

TDengine的产品竞争力

10倍以上的性能提升

- 定义了创新的时序数据存储结构，通过采用无锁设计和多核技术，TDengine 让数据插入和读出的速度比现有通用数据库高了10倍以上。
- 单核一秒就可处理2万以上插入请求，插入数百万数据点，可从硬盘读出一千万以上数据点。
- 数据都有预聚合处理，多表聚合查询保证只扫描一次数据文件，查询速度数量级的提升。8核服务器，100亿条记录的平均值计算不到2秒。





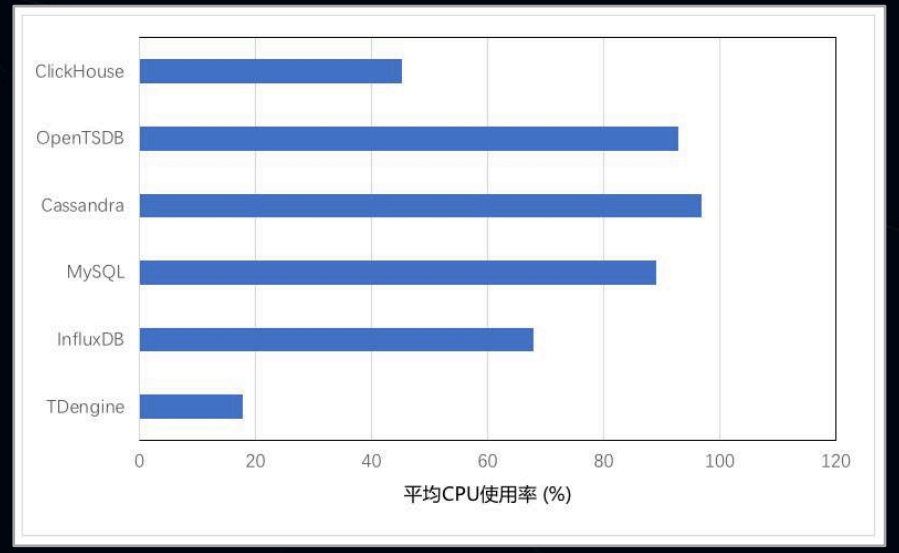
完整对比测试报告，请参阅：www.tdengine.com

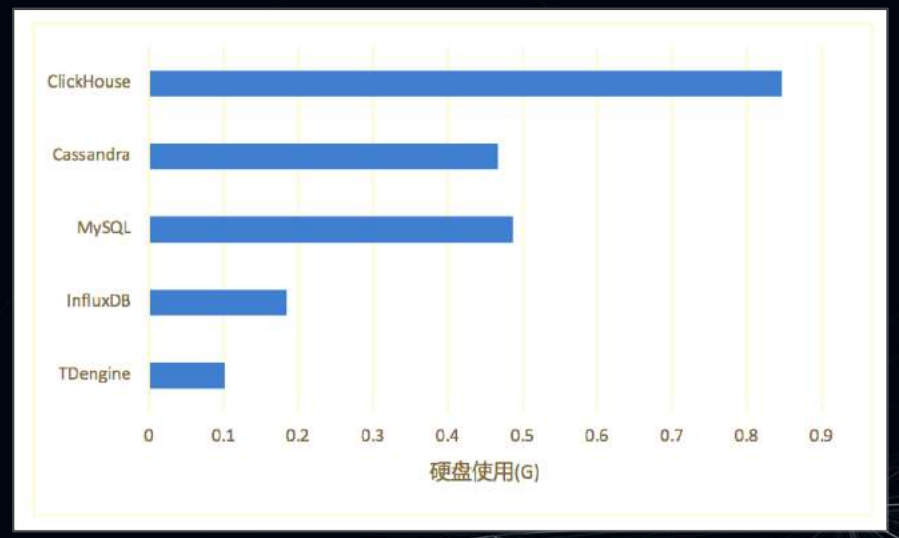
TalkingData

TDengine的产品竞争力

总拥有成本大幅下降

- 由于超强性能，计算资源不到通用大数据方案的1/5；通过列式存储和先进的压缩算法，存储空间不到通用数据库的1/10。
- 不用再集成Kafka, Redis, Spark, HBase等系列软件，系统架构大幅简化，产品研发成本大幅下降。
- 零管理，不用分库分表、不分历史库、实时库，数据实时备份，运维成本大幅下降。





完整对比测试报告，请参阅：www.tdengine.com

TalkingData

TDengine的产品竞争力

零学习成本

- 安装包仅仅1.5M，不依赖任何其他软件。从下载、安装到成功运行几秒搞定
- 使用标准的SQL语法，并支持C/C++, JAVA, GO, Python, RESTful接口，应用API与MySQL高度相似，让学习成本几乎为零
- 无论是十年前还是一秒钟前的数据，指定时间范围即可查询。数据可在时间轴或多个设备上进行聚合。临时查询可通过Shell/Python/R/Matlab随时进行
- 与第三方工具Telegraf, Grafana, Matlab, R等无缝链接

```
create database demo;
use demo;
create table t1(ts timestamp, degree float);
insert into t1 values(now, 28.5);
insert into t1 values(now, 29.0);
select * from t1;
select avg(degree), count(*) from t1;
```

TalkingData

TDengine 各项指标为何这么出众

充分利用物联网数据特点

- 对于一个数据采集点而言，只有一个写
- 结构化数据
- 时序的。。。

不基于任何开源产品，C/C++开发了

- 存储引擎
- 集群调度、管理
- 计算模块
- SQL解析。。。

量身定制
只为物联网数据而设计

不适合
电商、社区、ERP、CRM

TalkingData

TDengine 应用领域

上网记录、通话记录
交易记录、卡口数据

智能电表、水表、气表
电网、管道、智慧城市

火车/汽车/出租/飞机
自行车的实时监测

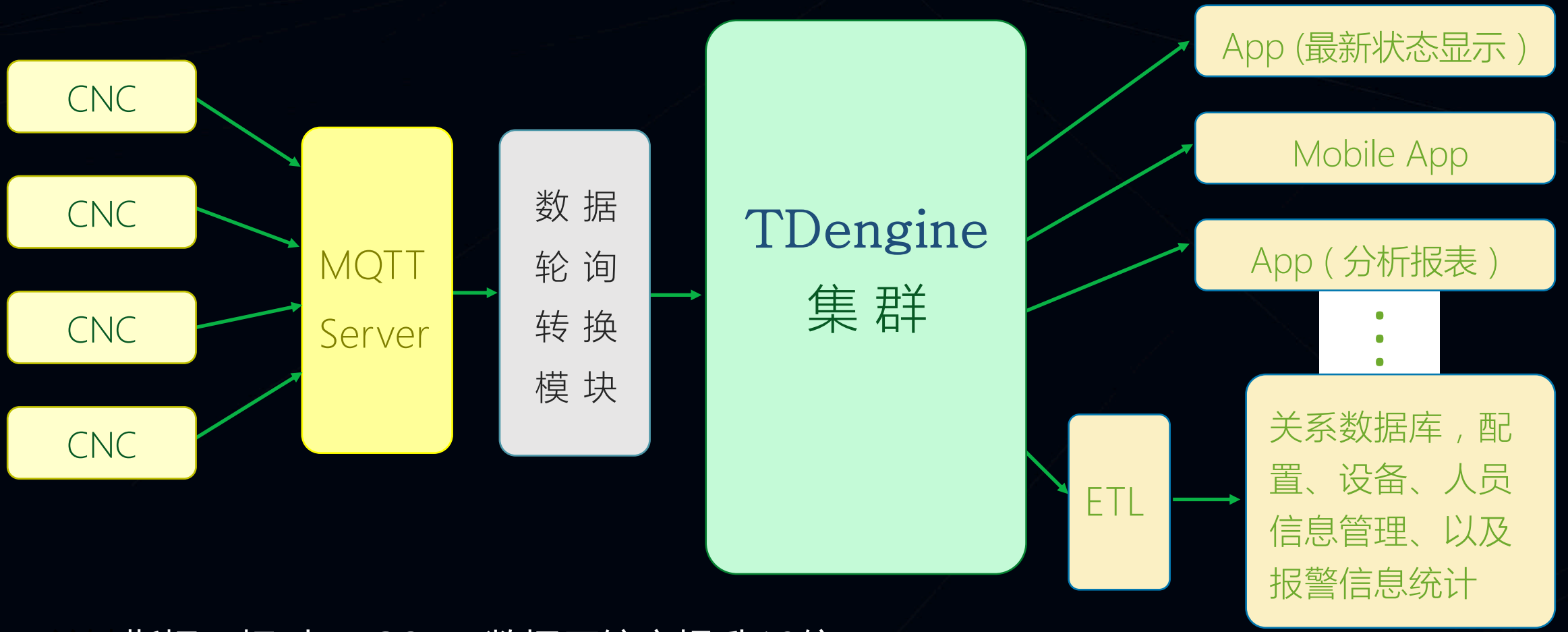
服务器/应用监测
用户访问日志、广告点击日志

电梯、锅炉、机床
机械设备实时监测

天气、空气、水文
地质环境监测

TalkingData

CNC计算机数控应用场景



关键指标：相对MySQL，数据压缩率提升18倍

查询速度提升20倍以上

TalkingData

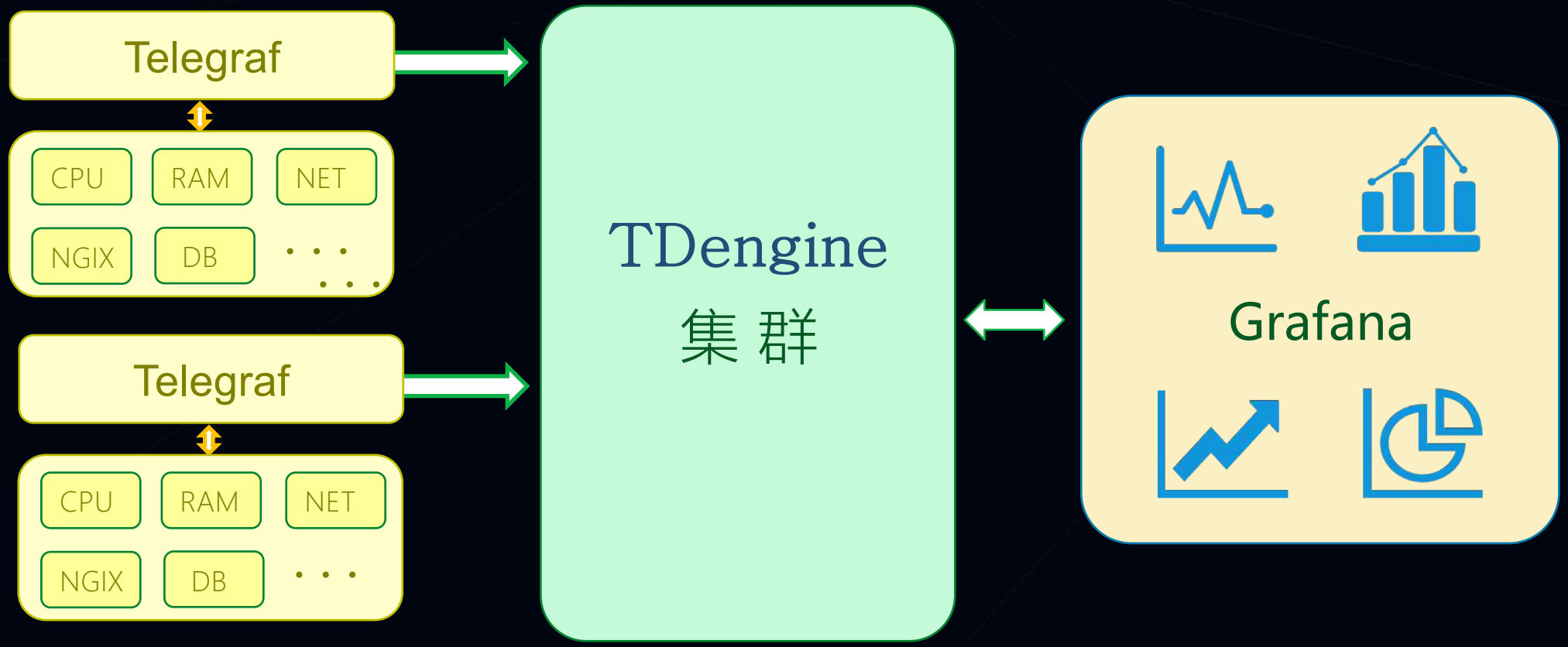
车联网轨迹数据应用场景

关键指标：CDH LZO 数据压缩率：44.3%

TDengine 数据压缩率：3.34%

TalkingData

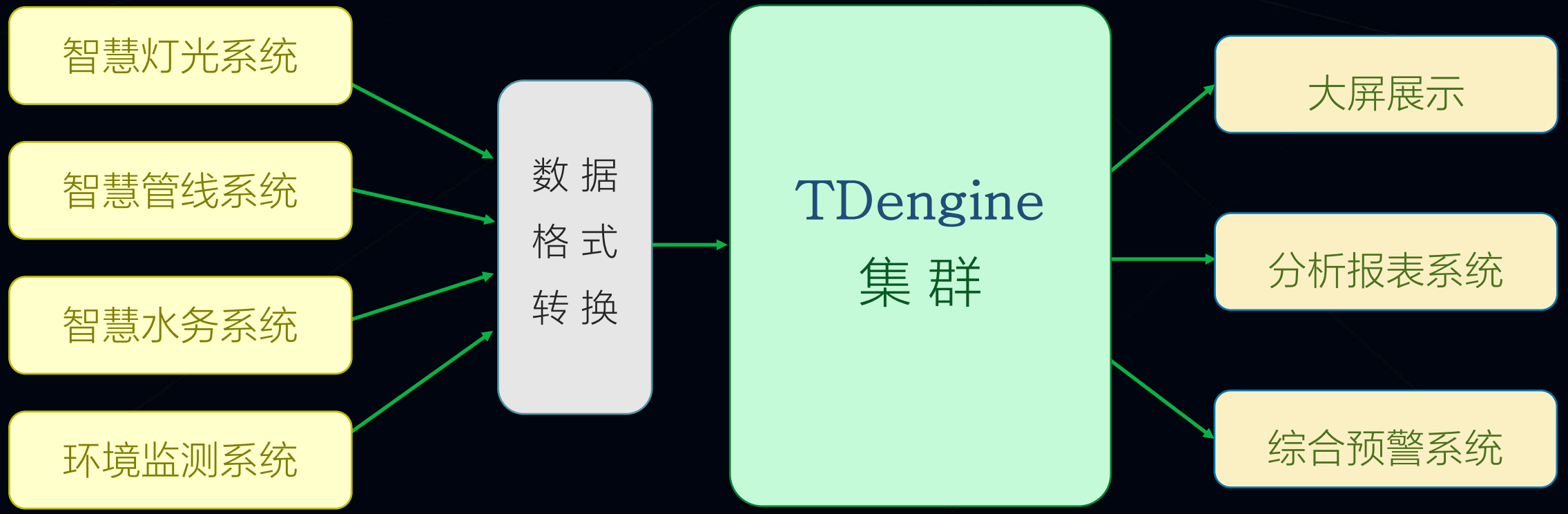
电力IDC运维监测场景



Telegraf+TDengine+Grafana 组合：无需代码、搭建一个高效的IT运维监测平台

TalkingData

智慧城市CIM系统

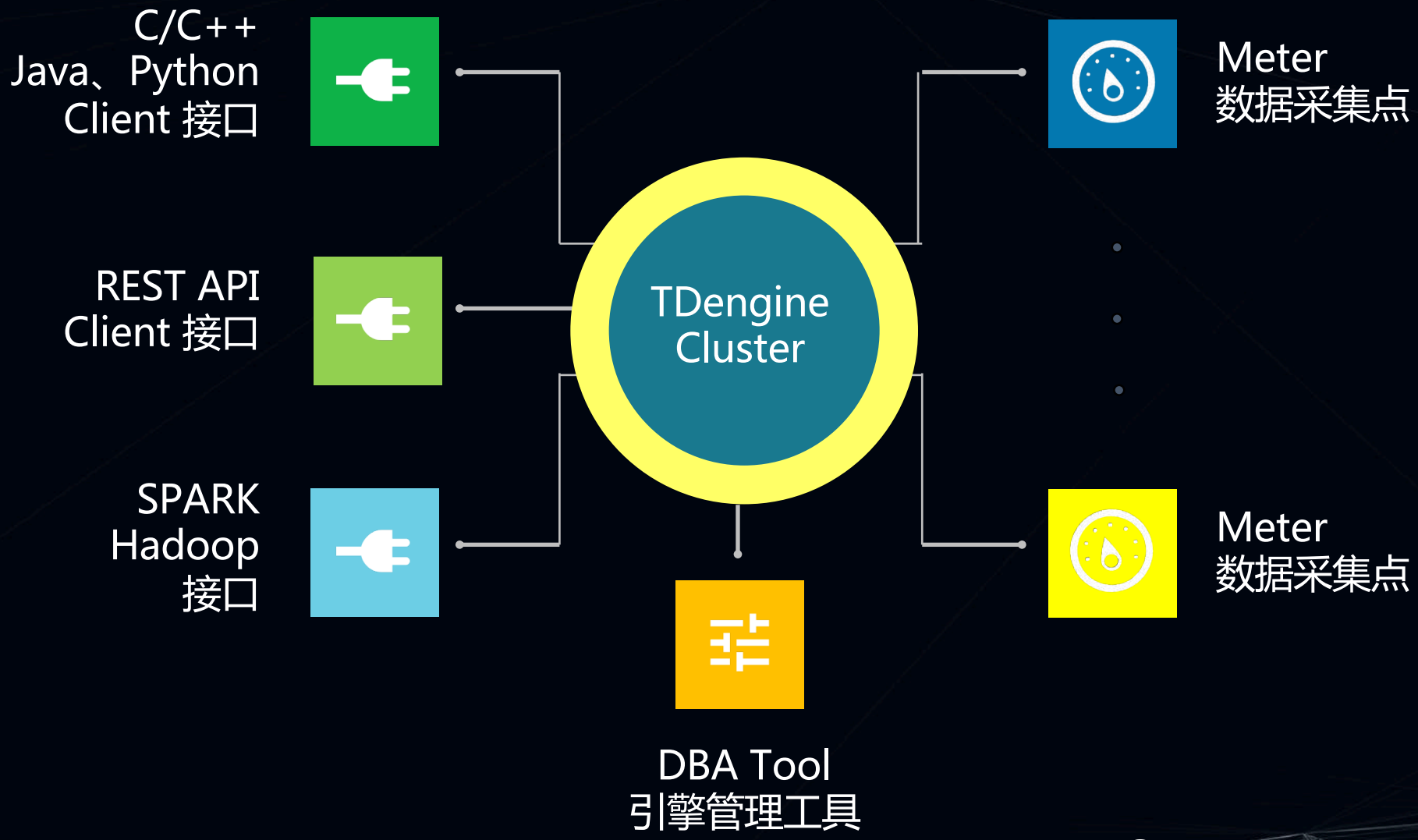


TalkingData

TDengine 技术介绍

TalkingData

TDengine 对外接口



TDengine 对外接口

TDengine本身运行在一组服务器上，提供高效可靠的时序数据采集、插入、查询和计算等基础功能

TDengine可与SPARK、Hadoop等大数据分析架构无缝对接，便于各种分析工具的使用

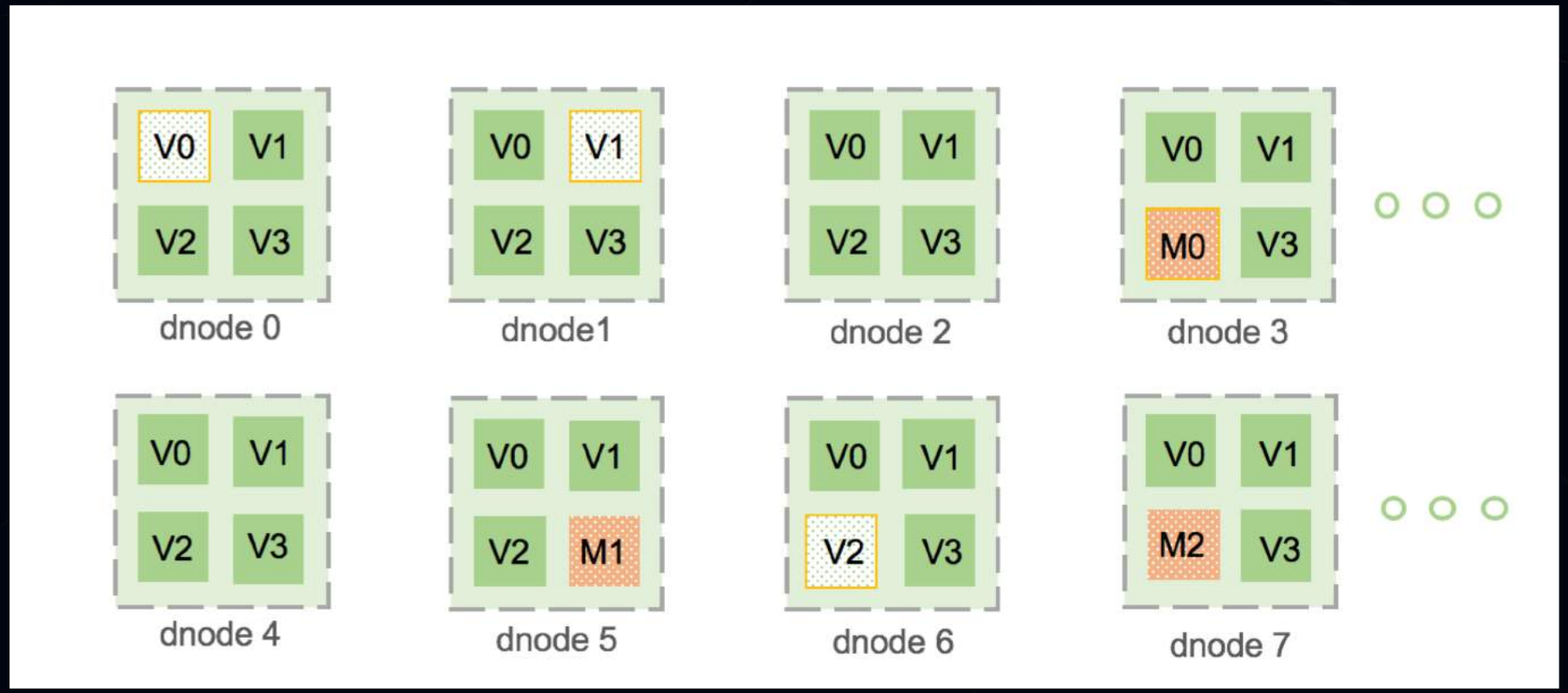
TDengine提供客户端各种语言的开发接口，包括C/C++，Java，Python等，API与MySQL一样，但是一个子集

TDengine提供一DBA工具，用于管理TDengine

TDengine提供一组REST API，这样便于跨平台的开发

为最大程度提高效率，采集点可以直接通过安全加密的链接将采集的数据写入引擎。采集点可以使用HTTPS或使用TAOS提供的SDK来传送数据

TDengine 系统结构



TalkingData

TDengine 系统结构

物理节点

一台实际服务器或虚拟机，根据具体的CPU、内存和存储资源，一个物理节点可以配置多个虚拟节点。

虚拟管理节点

负责所有物理节点运行状态的采集、负载均衡，和所有Meta Data的管理。系统会自动在整个集群里的三个物理节点上创建三个虚拟管理节点，以形成一个虚拟管理节点组，保证系统的高可靠

虚拟数据节点

存储具体的时序数据，所有针对时序数据的插入和查询操作，都在虚拟数据节点上进行。位于不同物理节点上的虚拟节点可以组成一个虚拟数据节点组，虚拟节点组里的虚拟节点的数据是实时同步的，保证系统的高可靠

TDengine 完全无中心化的设计

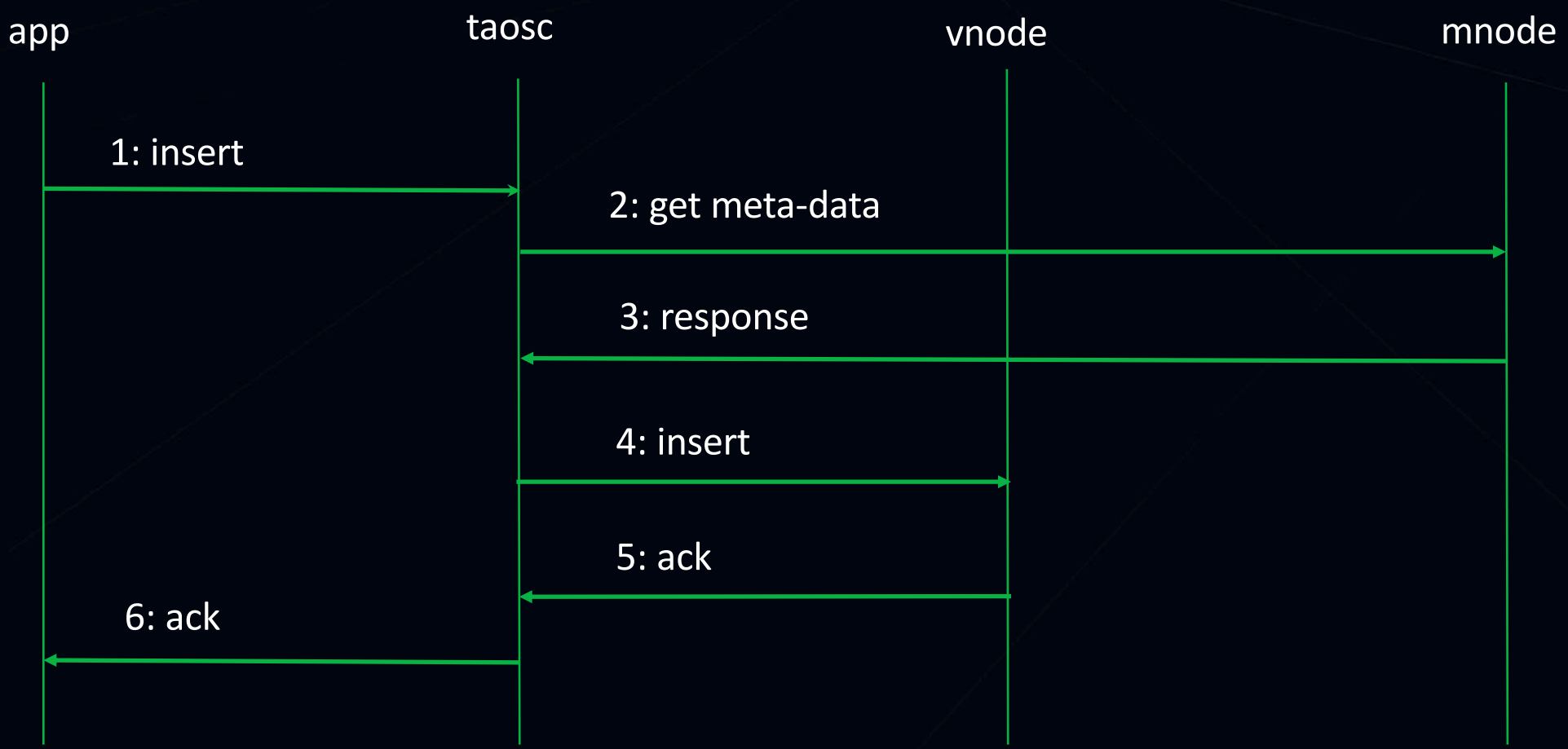
应用访问系统，可以连接集群中的任何一个节点进行

一个物理节点宕机或网络故障，不会影响系统的正常运行

节点的增加、删除或过热，系统会自动进行计算和存储的负载均衡

TalkingData

TDengine 一典型流程



TDengine 存储结构

将每一个采集点的数据作为数据库中的一张独立的表来存储，无论在内存还是硬盘上，一张表的数据点在介质上是连续存放的，这样大幅减少随机读取操作，数量级的提升读取和查询效率

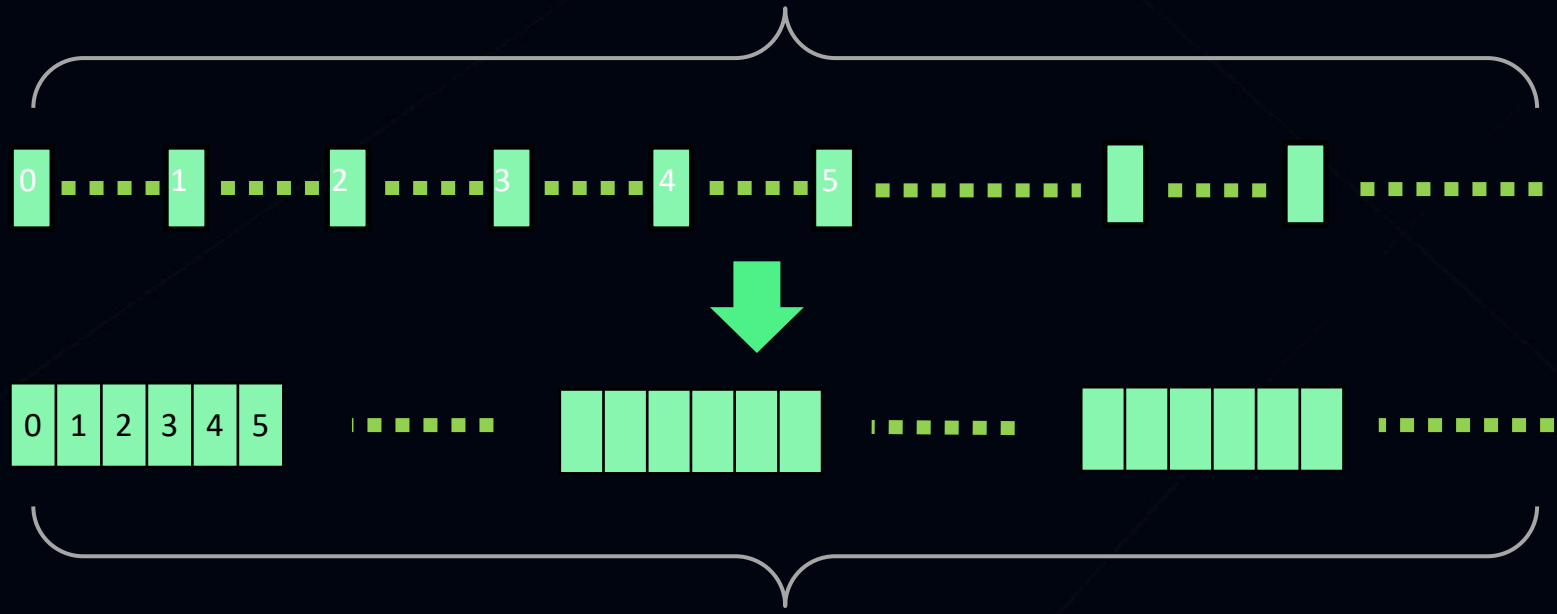
数据写入硬盘是以添加日志的方式进行的,但每个数据文件仅仅保存固定一段时间的数据，大幅提高数据落盘、同步、恢复、删除等操作的效率

为减少文件个数，一个虚拟节点内的所有表在同一时间段的数据都是存储在同一个数据文件里，而不是一张表一个数据文件

连续存放

使用KV或关系型DB，在多个数据采集点存在的情况下

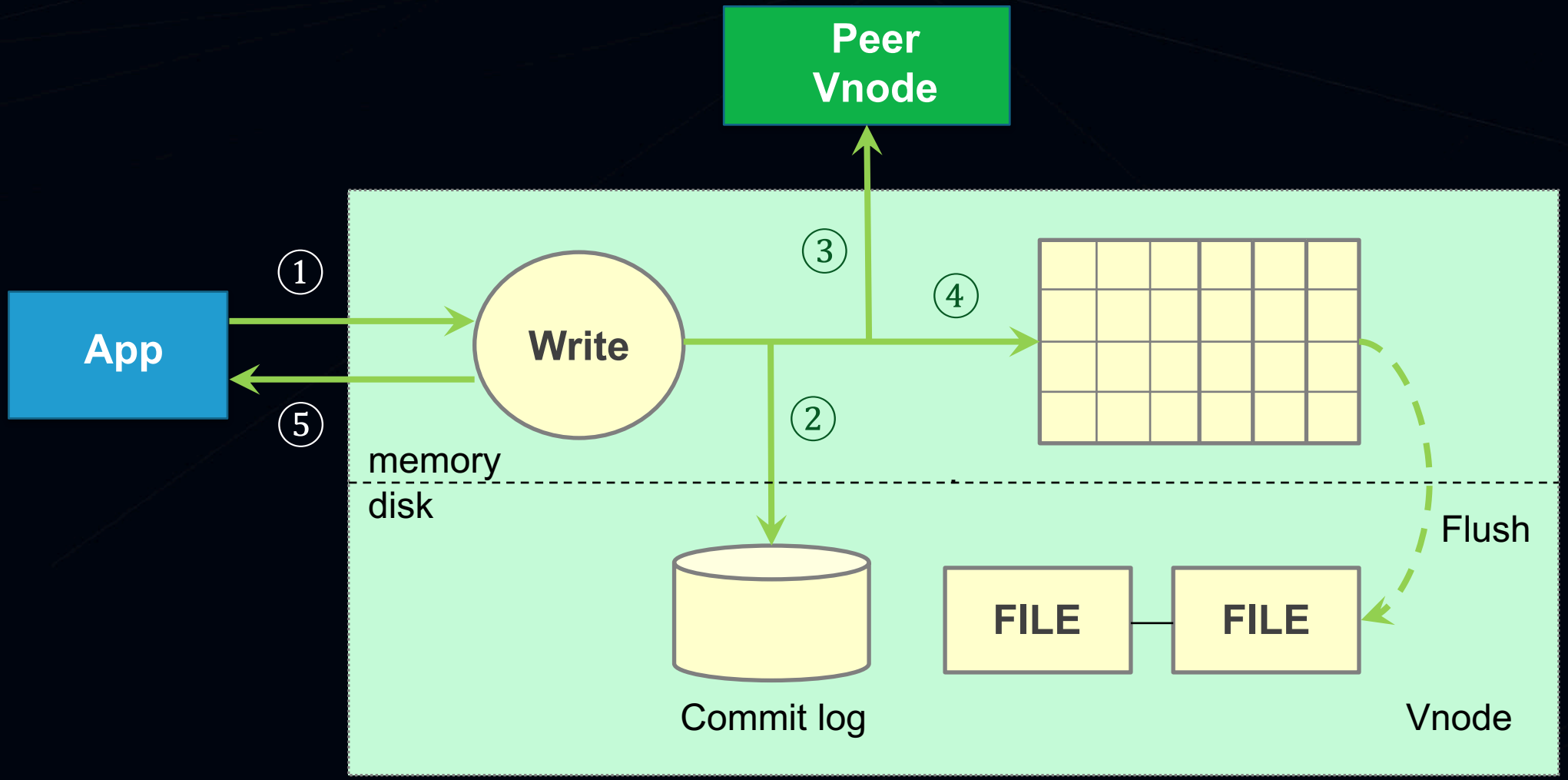
难以保证一个采集点的数据在内存或硬盘上的连续性



TDengine，一个采集点的数据在一个块里是连续存放的，块的大小可配置

采取Block Range Index, 可快速定位要查找的数据所处的块

TDengine 数据写入流程



TDengine 数据的多级存储

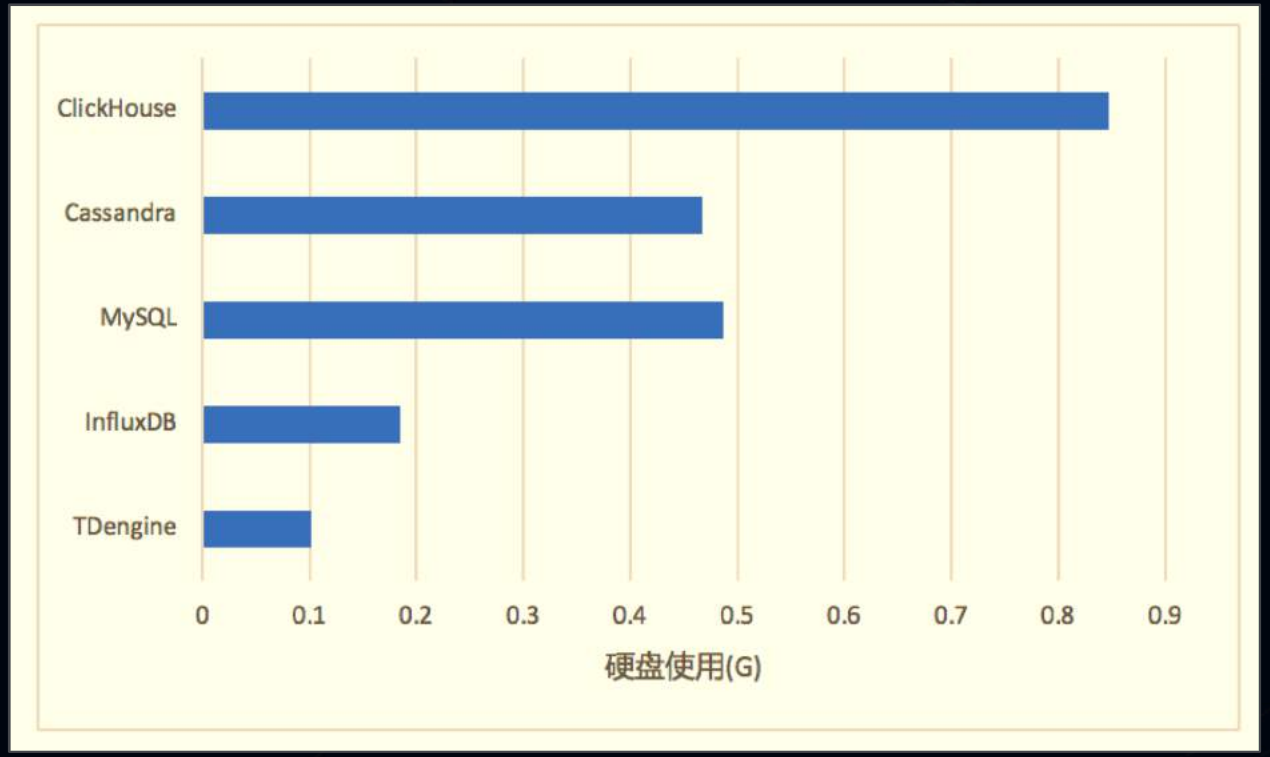
最新采集的数据在内存里，根据业务场景，可配置大小，以保证计算全部在内存里进行。使用TDengine, 无需再集成Redis或其他缓存软件。

当内存满或者设置的时间到，内存的数据会写到持久化存储设备上。TDengine既能处理实时数据，又能处理历史数据，但这一切对应用是透明的。

持久化存储还可以按照时间分级，比如一天之内的数据在SSD盘上，一个月内的数据在普通硬盘上，超过一个月的数据在最廉价的存储介质上，最大程度降低存储成本。

TDengine 数据压缩

- 采取列式存储，便于压缩
- 不同数据类型采用不同的压缩算法，包括 delta-delta 编码、simple 8B 方法、zig-zag编码、LZ4等算法
- 二阶段压缩：压缩的基础上，用通用压缩算法进行再压缩



TDengine 单个采集点数据插入、查询示例

与传统关系型数据库一样，采用SQL语法，应用需先创建库，然后建表，就可以插入、查询

```
create database demo;
use demo;
create table t1 (ts timestamp, degree float);
insert into t1 values(now, 28.5);
select * from t1;
```

上述SQL 为一个温度传感器创建一张表，并插入一条记录、然后查询

TDengine 超级表：多个采集点的数据聚合

实际场景中，经常需要将多个采集点数据进行聚合处理，比如所有温度传感器采集的温度的平均值。因为一个传感器就是一张表，这样需要将多张表聚合。为减少应用的复杂性，TDengine引入STable概念。

STable(超级表）是表的集合，包含多张表，而且每张表的schema是一样的。同一类型的采集设备可以是一个STable，除定义Schema外, 还可定义多个标签。标签定义表的静态属性，如设备型号、颜色等。具体创建表时，指定使用哪个STable（采集点的类型），并指定标签值。

应用可以象查询表一样查询STable, 但可以通过标签过滤条件查询部分或全部数据采集点的记录，并且可以做各种聚合、计算等，方便支持复杂查询，应对业务需求。

每个表（采集点）都有对应一行的标签数据，保存在Meta节点，而且存放在内存并建有索引。标签数据可以任意增加、删除、修改。标签数据与采集数据完全分离，大大节省存储空间，并提高访问效率。而且对于已经采集的历史数据，事后可以打上新的标签。

TDengine 超级表实例

为温度传感器建立一个STable, 有两个标签：位置和类型

```
create table thermometer (ts timestamp, degree float) tags(loc binary(20), type int);
```

用STable创建5张表, 对应5个温度传感器，地理位置标签为北京、天津、上海等

```
create table t1 using thermometer tags('beijing', 1);
create table t2 using thermometer tags('beijing', 2);
create table t3 using thermometer tags('tianjin', 1);
create table t4 using thermometer tags('tianjin', 2);
create table t5 using thermometer tags('shanghai',1);
```

查询北京和天津所有温度传感器记录的最高值和最小值

```
select max(degree), min(degree) from thermometer where loc='beijing' or loc='tianjin';
```

TDengine 时间轴上的数据聚合

实际场景中，经常需要将一段时间的数据进行聚合，比如downsampling, 采样频率为一秒一次，但最终只记录一分钟的平均值。TDengine引入关键词interval, 以进行时间轴上的聚合操作。时间轴的聚合既可以针对单独一张表，也可以针对符合标签过滤条件的一组表进行。

查询温度传感器t1记录的温度每五分钟的平均值

```
select avg(degree) from t1 interval(5m);
```

查询北京所有温度传感器记录的温度每五分钟的平均值

```
select avg(degree) from thermometer where loc='beijing' interval(5m);
```

TDengine 实时Stream计算

目前支持Avg, Max, Min, Percentile, Sum, Count, Dev, First, Last, Diff, Scale, WAvg, Spread等操作。计算是针对时间段，同时可针对一张表或符合过滤条件的一组表进行聚合。

实时计算的衍生数据可以实时写入新的表，方便后续的查询操作。衍生数据还可以与其他原始数据或其他衍生数据进行各种聚合计算，生成新的数据。

每隔一分钟计算北京刚刚过去的五分钟的温度平均值

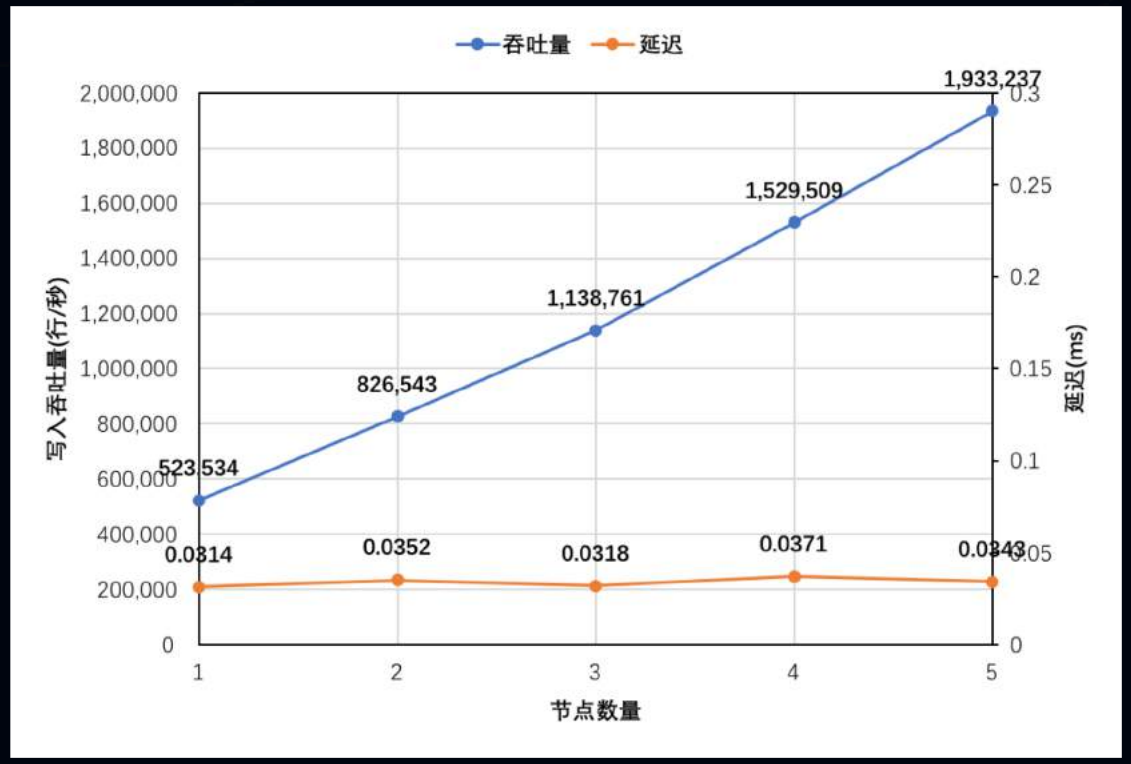
```
select avg(degree) from thermometer where loc='beijing' interval(5m) sliding(1m);
```

每分钟计算一次北京刚过去的5分钟的温度平均值，并写入新的表d1

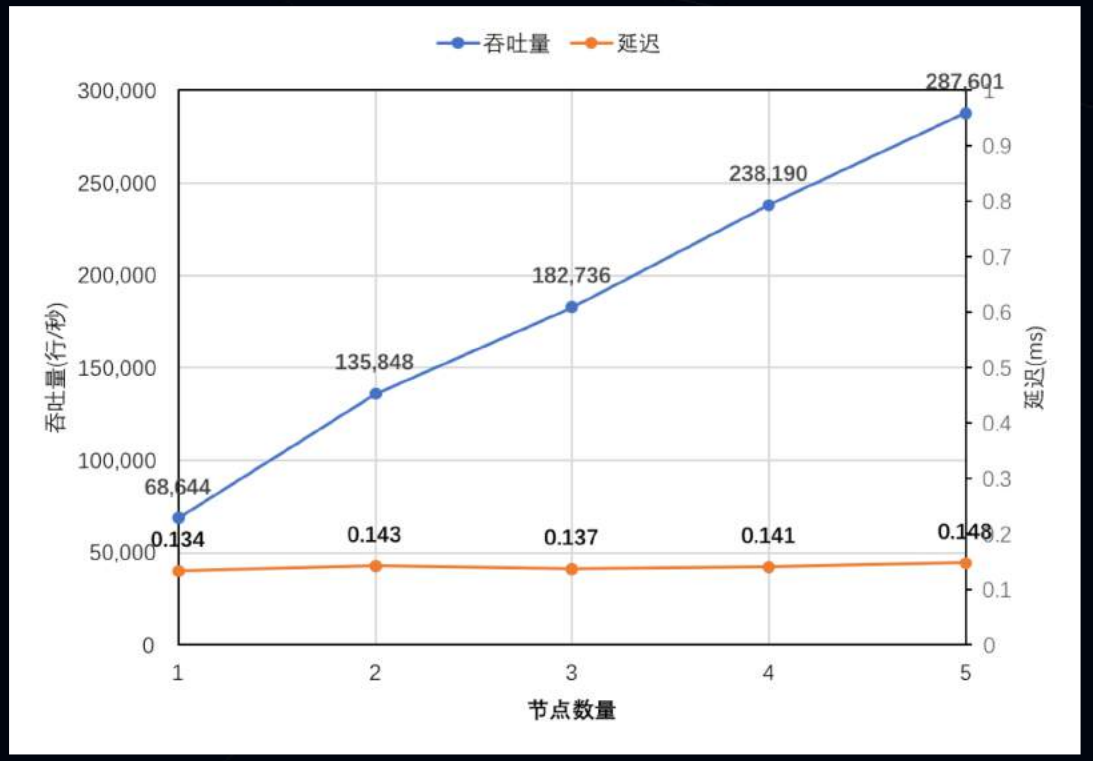
```
create table d1 as
select avg(degree) from thermometer where loc='beijing' interval(5m) sliding(1m);
```

TalkingData

TDengine 水平扩展测试结果：完全线性

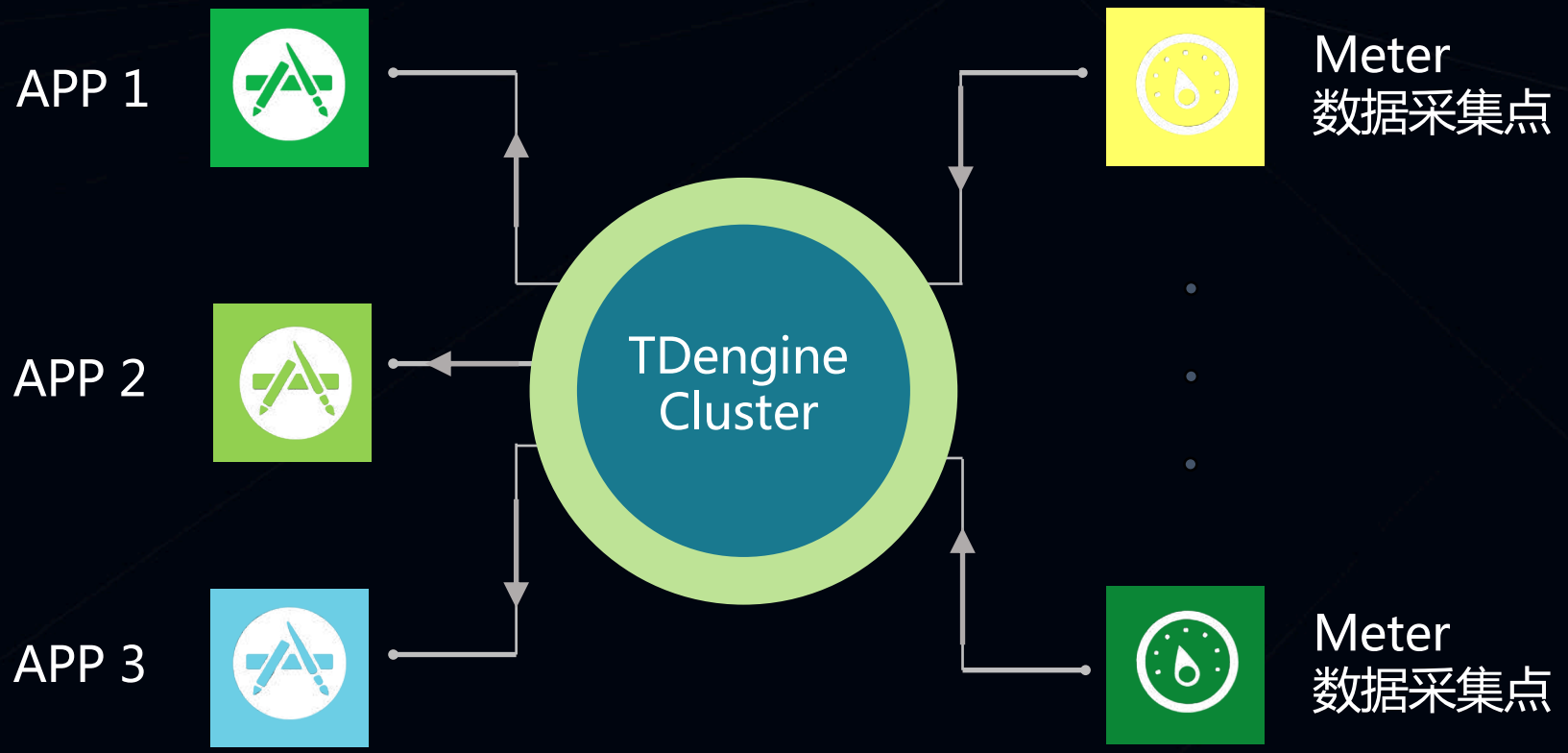


数据插入吞吐量、延时与节点个数关系



数据查询吞吐量、延时与节点个数关系

数据订阅



- 类似流行的Kafka，应用可以订阅数据流，只要数据有更新，应用将得到及时通知
- 订阅时，应用只要指定数据库的表名和开始时间即可

TalkingData

支持异构环境

不同类型、不同性能的服务器可以组建集群，系统根据物理机的配置，自动创建不同数量的虚拟节点

通过Spark/Hadoop Connector,可以与Oracle,MySQL, Cassandra, MongoDB并存，大数据分析处理软件不用做任何修改

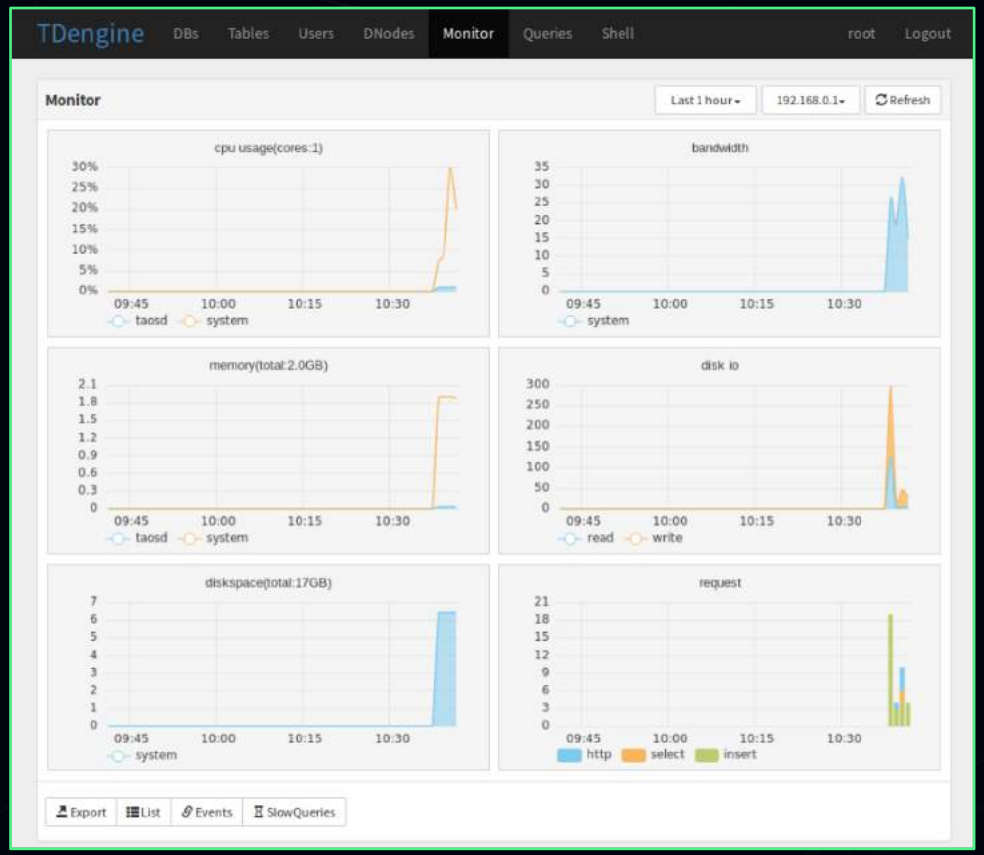
TalkingData

TDengine 零管理：日常运维工作为零

- 不存在分库、分表
- 不存在实时库、历史库之分
- 不存在档案数据之说，只需要配置好多级存储的存储路径
- 扩容，加入新的节点，一条指令搞定
- 系统根据资源情况，自动负载均衡，无需任何人工干预
- 将副本数设置好，数据将自动实时备份

TDengine 运维管理工具

- 数据导入、导出：
  - 支持按SQL脚本文件导入，支持按数据文件导入
  - 在shell里按查询结果直接导出到CSV文件
  - 专用工具taosdump: 导出所有数据库、一个数据库或者数据库中的一张表,所有数据或一时间段的数据，甚至仅仅表的定义
- WEB管理工具
  - 在shell里的操作都可在web页面上进行，更加友好
  - 各种系统监测指标（插入、查询次数，服务器资源等）用图表可视化
  - 各种重要操作日志都可查询
- 与运维监测平台无缝对接
  - 各种监测量和日志都存放在DB里，可用标准SQL获取，便于对接第三方运维监测平台



TDengine 异地容灾、备份

- 对于集群里每台服务器，记录了每个节点的IDC、以及机柜的ID
- 组成一个虚拟节点组时，根据策略，可选不同机柜，不同IDC的节点组成节点组
- 节点之间的数据是实时备份的，但用异步的方式进行，以降低网络延时的影响
- 一个节点组里Master宕机，节点组将实时检测到，立即重新选Master
- 如果一个IDC整个瘫痪，一秒以内，Master就会切换到另外的IDC
- 无需购置其他第三方数据备份工具

TDengine IDC 迁移，可以无服务中断进行

- 对于副本数，用户可以动态的调整，增加、减少都可以
- 增加副本时，数据会立即同步到新的副本，而且服务不受中断
- 为防止同步消耗资源过多，TDengine控制了一个节点并发的同步数量。
- 迁移IDC时，只需要做两件事
  - 将新IDC机房的服务器加入集群，负载均衡将自动启动，等待其完成
  - 将老的IDC机房，一台一台的关机。每关一台，等新的副本在新IDC机房创建完毕，再关下一台。

IDC机房迁移过程中，TDengine能保证服务的正常运行

李广
guangli@taosdata.com
Tel:15801211043

TalkingData