

Gdevops

Global DevOps Summit

# 全球敏捷运维峰会

TDengine 的创新与最佳应用场景

陶建辉 涛思数据创始人



# 大数据时代

数据采集后被源源不断的发往云端



# 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。



HBASE



elastic



hadoop



ZOOKEEPER



mongoDB

(name: "mongo", type: "DB")

## 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。

## 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。

## 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、IT运维监测数据特征：时序空间数据

- 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





# TDengine 应运而生

# TDengine提供的功能



## 消息队列

自带消息队列，  
无需 Kafka 或其他 MQ



## 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

## TDengine: All in One的时序大数据平台

为时序数据处理提供全栈解决方案，无需再集成Kafka, Redis, Spark, HBase, Zookeeper等软件，大幅降低系统架构的复杂度，而且极大提高数据处理的性能

# TDengine的产品竞争力

## • 十倍以上的性能提升

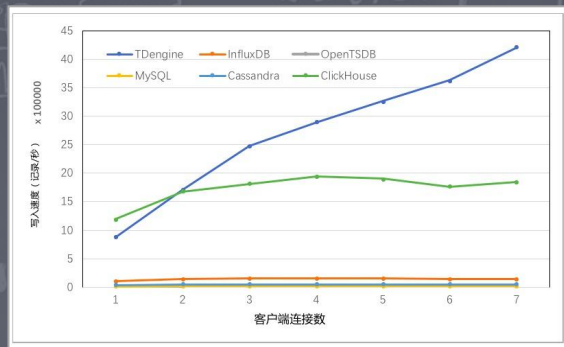
定义了创新的时序数据存储结构，通过采用无锁设计和多核技术，TDengine 让数据插入和读出的速度比现有通用数据库高了10倍以上。

## • 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。

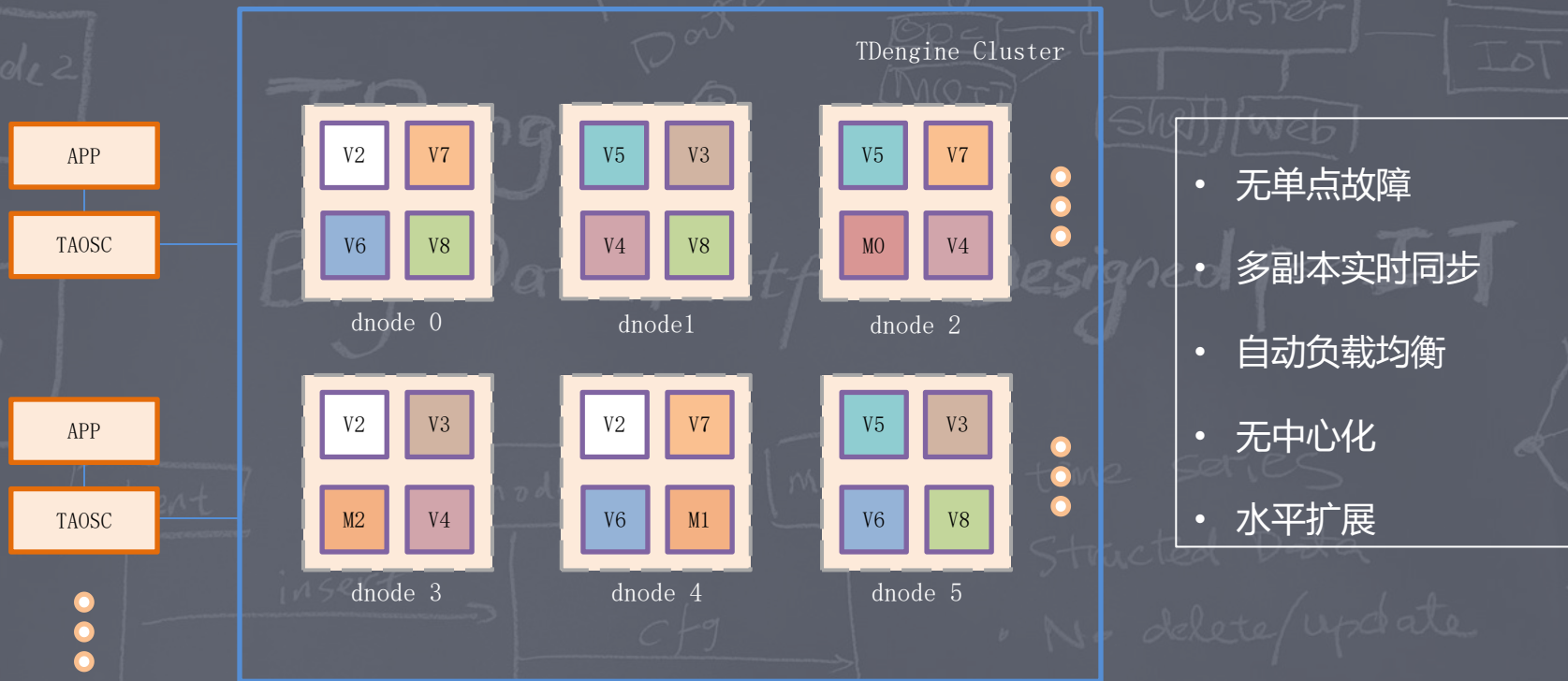
完整对比测试报告，请参阅：[www.tdengine.com](http://www.tdengine.com)



# TDengine 技术创新之处



# TDengine 系统结构





# TDengine 的创新之一

## 数据模型：一个数据采集点一张表



# 物联网典型场景：智能电表

设备 ID	时间戳	采集数据			标签	
Device ID	Time Stamp	Current	Voltage	Phase	Location	Type
d1001	1538548685000	3.13	220	0.31	BJ.chaoyang	1
d1002	1538548685100	8.21	219	0.82	BJ.haidian	2
d1001	1538548686000	3.11	219	0.35	BJ.chaoyang	1
d1003	1538548683000	5.41	110	0.53	BJ.daxing	1
d1002	1538548686100	8.11	223	0.81	BJ.haidian	2
d1002	1538548687130	8.15	215	0.85	BJ.haidian	2
d1001	1538548687000	3.15	223	0.32	BJ.chaoyang	1
d1003	1538548684000	5.51	112	0.54	BJ.daxing	1
d1003	1538548685000	5.60	109	0.53	BJ.daxing	1
d1002	1538548688100	8.19	218	0.87	BJ.haidian	2
d1003	1538548686000	5.62	108	0.56	BJ.daxing	1
d1001	1538548688500	3.19	221	0.31	BJ.chaoyang	1

## 很显著的特点

- 数据是时序的，都带有时间戳
- 数据是结构化的，而且以数字型为主
- 除非固件更新，结构不会改变
- 每个采集点都带有静态属性标签
- 由于网络或其他原因，每个采集点的数据到达服务器的时间是无法控制的，但每个采集点的采集数据到达服务器的相对顺序是基本有保证的

# 一个数据采集点一张表

设备ID: D1001, 标签 loc: BJ.chaoyang type: 1

Time stamp	Current	Voltage	Phase
1538548685000	3.13	220	0.31
1538548686000	3.11	223	0.35
1538548687000	3.15	219	0.32
1538548688500	3.19	221	0.33

设备ID: D1002, 标签 loc: BJ.haidian type:

Time stamp	Current	Voltage	Phase
1538548685100	8.21	219	0.82
1538548686100	8.11	223	0.81
1538548687130	8.15	215	0.85
1538548688100	8.19	218	0.87

设备ID: D1003, 标签 loc: BJ.daxing type: 2

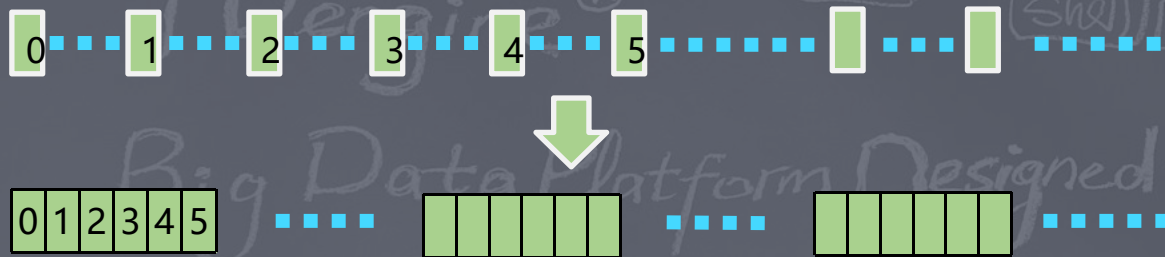
Time stamp	Current	Voltage	Phase
1538548683000	5.41	100	0.53
1538548684000	5.51	109	0.54
1538548685000	5.60	112	0.53
1538548686000	5.62	108	0.56

## 一个采集点一张表带来的直接变化

- 每张表里的记录按时间自动排序
- 新数据记录的写入变成简单的追加操作
- 数值按列的变化范围更小
- 设备ID、标签不会重复存储



# 表的存储：一块一块，块内连续存储



- 每张表的记录按块存储
- 每个数据块包含一定数量的记录条数
- 每个数据块带有预计算
- 每个数据块带有schema
- 一张表往往有多个数据块
- 系统有块索引，根据起止时间，迅速找到数据块

# 块内数据采用列式存储

Time stamp	Current	Voltage	Phase
1538548685000	3.13	220	0.31
1538548686000	3.11	223	0.35
1538548687000	3.15	219	0.32
1538548688500	3.19	221	0.33

- 大幅提高压缩率：同一列数据相近，便于压缩；不同类型数据可采用不同压缩算法
- 大幅提高分析性能：时序数据的分析，往往是针对一个采集量在一个时间范围段进行的。如果行式存储，会读取大量的无效数据

Row-based Store

1538548685000	3.13	220	0.31	38548	1538548686000	3.11	223	0.35	3.11	1538548687000	3.15	219	0.32	9	1538548688500	3.19	221	0.33
---------------	------	-----	------	-------	---------------	------	-----	------	------	---------------	------	-----	------	---	---------------	------	-----	------

# TDengine 通过采用以下几点策略

- 一个数据采集点一张表
- 一张表的数据按块连续存储
- 数据块内采取列式存储

保证了单个数据采集点的插入和查询效率是最好的



## TDengine 的创新之二

# 超级表：多个数据采集点的高效聚合





# 超级表：用以描述某一种类型的数据采集点

- 描述一张超级表，即一个数据采集点类型，需要定义
  - 采集量的数据结构 schema
  - 静态标签的数据结构 schema
- 为一个具体数据采集点创建表
  - 以超级表为模板，表的schema就是超级表的采集量schema
  - 给静态标签指定具体值

# TDengine超级表使用实例

为智能电表这个设备类型建立一个 STable, 采集量有电流、电压和相位, 标签有位置和类型

```
create table smeter (ts timestamp, current float, voltage int, phase float)  
tags (loc binary(20), type int);
```

用 smeter 做模板, 为6个智能电表创建6张表, 地理位置标签为北京朝阳、海淀、上海浦东等

```
create table t1 using smeter tags('BJ.chaoyang', 1);  
create table t2 using smeter tags('BJ.haidian', 2);  
create table t3 using smeter tags('BJ.daxing', 1);  
create table t4 using smeter tags('BJ.chaoyang', 2);  
create table t5 using smeter tags('SH.pudong', 1);  
create table t6 using smeter tags('SH.Hongqiao', 1);
```

# 通过超级表进行聚合查询、多维分析

查询北京朝阳区所有智能电表的电压平均值和电流最大值

```
Select avg(voltage), max(current) from smeters where loc = "BJ.chaoyang"
```

查询北京类型为1的智能电表的电压平均值

```
Select avg(voltage) from smeters where type = 1 and loc like "BJ%"
```

- 超级表可以象普通表一样查询，但可以指定标签的过滤条件
- 标签可以多至128个，每个标签代表一个维度
- 标签可以事后增加、删除、修改。这样数据建模时，可以先不确定标签或分析维度
- 每个标签，可以是一树状结构，比如 北京.朝阳.望京，这样便于缩小搜索范围

# 标签的存储

标签存储策略：

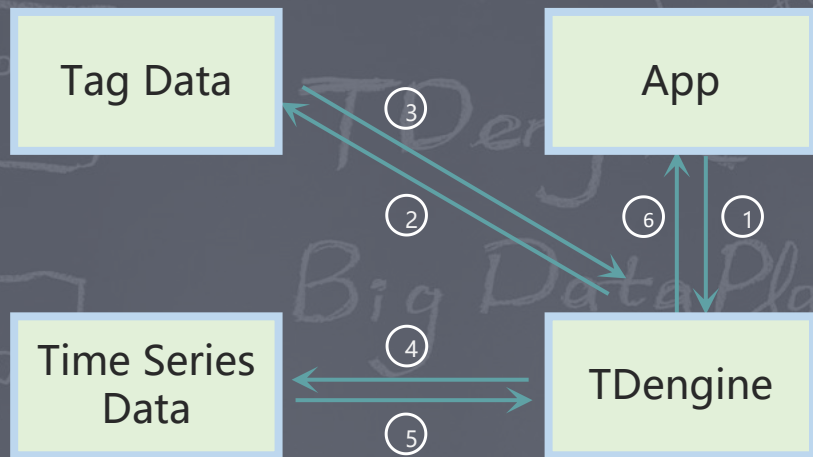
- 标签数据单独存储，与时序数据完全分离
- 采用Key-Value型存储，便于增删改操作
- 每个数据采集点一条标签记录
- 标签记录集中在一起存储，并建有索引

优势：

- 与典型的NoSQL数据库相比，因为不重复存储标签值，大幅节省存储空间
- 做多维聚合分析时，先标签过滤，找到需要聚合的采集点，大幅减少聚合的数据集
- 标签记录总条数等于采集点的数目，但总量不大，可全内存处理，进一步提升查询效率



# TDengine聚合计算流程



## 主要流程:

1. App向TDengine发起查询请求
2. TDengine将标签过滤条件发往标签数据处理模块
3. 标签查询模块将符合过滤条件的采集点列表返回
4. TDengine通知时序数据处理模块对选择的采集点在指定的时间段进行聚合操作
5. TDengine收到聚合后的结果
6. TDengine将结果返回给App

- 本质上, Tag Data是Dimension Table, 而TS Data是Facts Table。
- 换一个角度看, 超级表的设计是两级索引结构, 一级索引是标签, 用于过滤数据采集点。二级索引是时间戳, 用于过滤采集的时序数据

# TDengine 通过超级表的设计

- 大幅节省标签的存储空间
- 大幅提升多个数据采集点的聚合效率
- 让多维分析方便、高效

TDengine 是一强大的多维分析工具



# TDengine其他亮点

- time series
- Structured Data
- No delete/update
- Data Source Unique & Index

# 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。

查询智能电表t1记录的电压每五分钟的平均值

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

查询北京所有智能电表记录的电压每五分钟的平均值

```
select avg(voltage) from smeter where loc like "BJ%" 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。

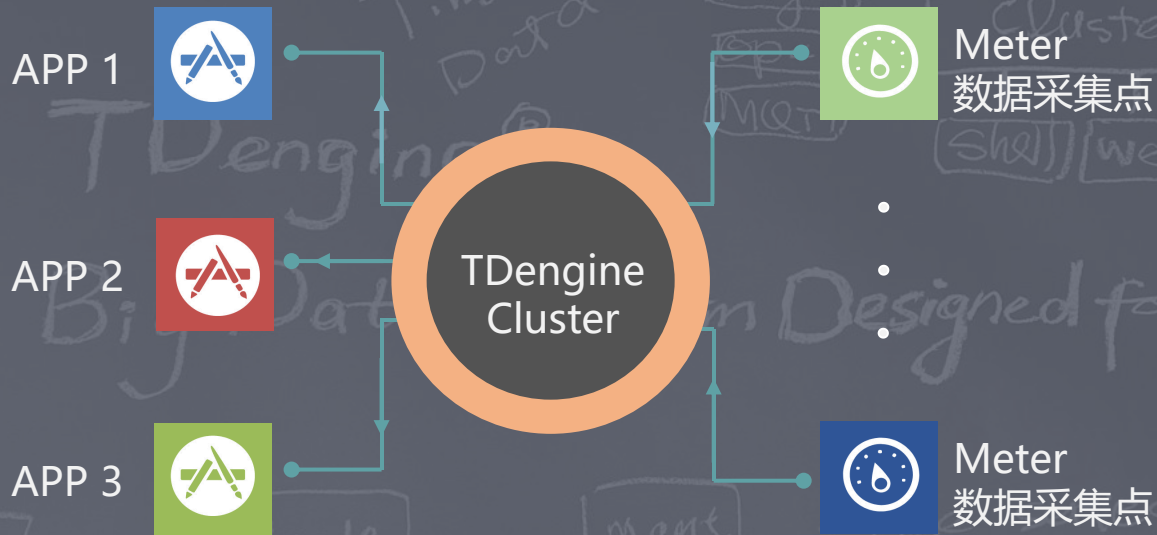
每隔一分钟计算北京刚刚过去的五分钟电压平均值

```
select avg(voltage) from smeter where loc like 'BJ%' interval(5m) sliding(1m);
```

每分钟计算一次北京刚过去的5分钟的电压平均值，并写入新的表d1

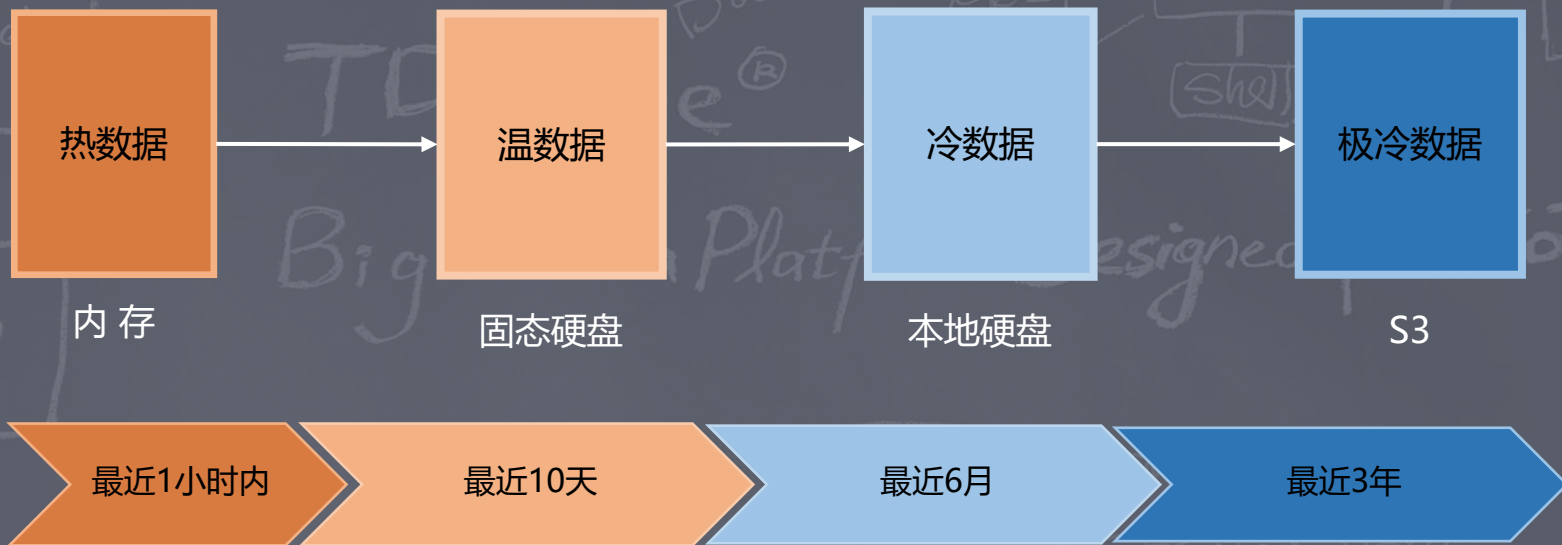
```
create table d1 as  
select avg(voltage) from smeter where loc like 'BJ%' interval(5m) sliding(1m);
```

# 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，应用只要指定表名（可以是超级表）和开始时间即可，还可指定过滤条件

# TDengine 数据的多级存储



按照时间自动迁移数据到不同存储介质，只需简单配置时间范围和对应存储路径。



www.github.com/taosdata/TDengine



# TDengine核心代码全部开源

单机版开源

2019.07.12

16,100+

Star 数

集群版开源

2020.08.03

4,100+

Fork 数

8,000+

PR & Issue 数

项目	开源时间	Star 数据	Fork 数据
InfluxDB	2013	22,200	3,000
OpenTSDB	2011	4,500	1,200
TimeScale	2017	11,800	626



Reset Filters

Grouping

Category

Sort By

Stars (high to low)

Category

Database

CNCF Relation

Any

License

Any

Organization

Any

Headquarters Location

Any

Example filters:

Cards by age

Open source landscape

Member cards

Cards by stars

Cards from China

Certified K8s/KCSP/KTP

Cards by MCap/Funding

Download as CSV



**redis**

Redis  
Redis Labs

★ 42,380  
Funding: \$146.6M



**RethinkDB**

RethinkDB  
Linux Foundation

★ 23,503



**TiDB**

TiDB  
PingCAP

★ 23,302  
Funding: \$71.6M



**Cockroach**  
LABS

CockroachDB  
Cockroach Labs

★ 18,089  
Funding: \$108.5M



**mongoDB**

MongoDB  
MongoDB

★ 17,767  
MCap: \$9.21B



**SEATA**

Seata  
Ant Financial

★ 15,301  
MCap: \$563.59B



**Dgraph**

Dgraph  
Dgraph Labs

★ 13,014  
Funding: \$14.45M



**TAOS**  
DATA

TDengine  
Taos Data

★ 11,116  
Funding: \$3M



**ShardingSphere**

ShardingSphere  
Apache Software Foundation

★ 10,818



**APACHE**  
**hadoop**

Apache Hadoop  
Apache Software Foundation

★ 10,360



**Vitess**

Vitess  
Cloud Native Computing Foundation (CNCF)

★ 9,889



**FoundationDB**

FoundationDB  
Apple

★ 9,855  
MCap: \$1.24T



**druid**

Druid  
Apache Software Foundation

★ 9,549



**ArangoDB**

ArangoDB  
ArangoDB

★ 9,484  
Funding: \$19.22M



**neo4j**

Neo4j  
Neo4j

★ 7,657  
Funding: \$160.1M



**KV**

TIKV  
Cloud Native Computing Foundation (CNCF)

★ 7,201



**noms**

NomsDB  
Salesforce

★ 7,182  
MCap: \$137.81B



**PostgreSQL**

PostgreSQL  
PostgreSQL

★ 6,617



**SCYLLA**

Scylla  
ScyllaDB

★ 5,970  
Funding: \$60M



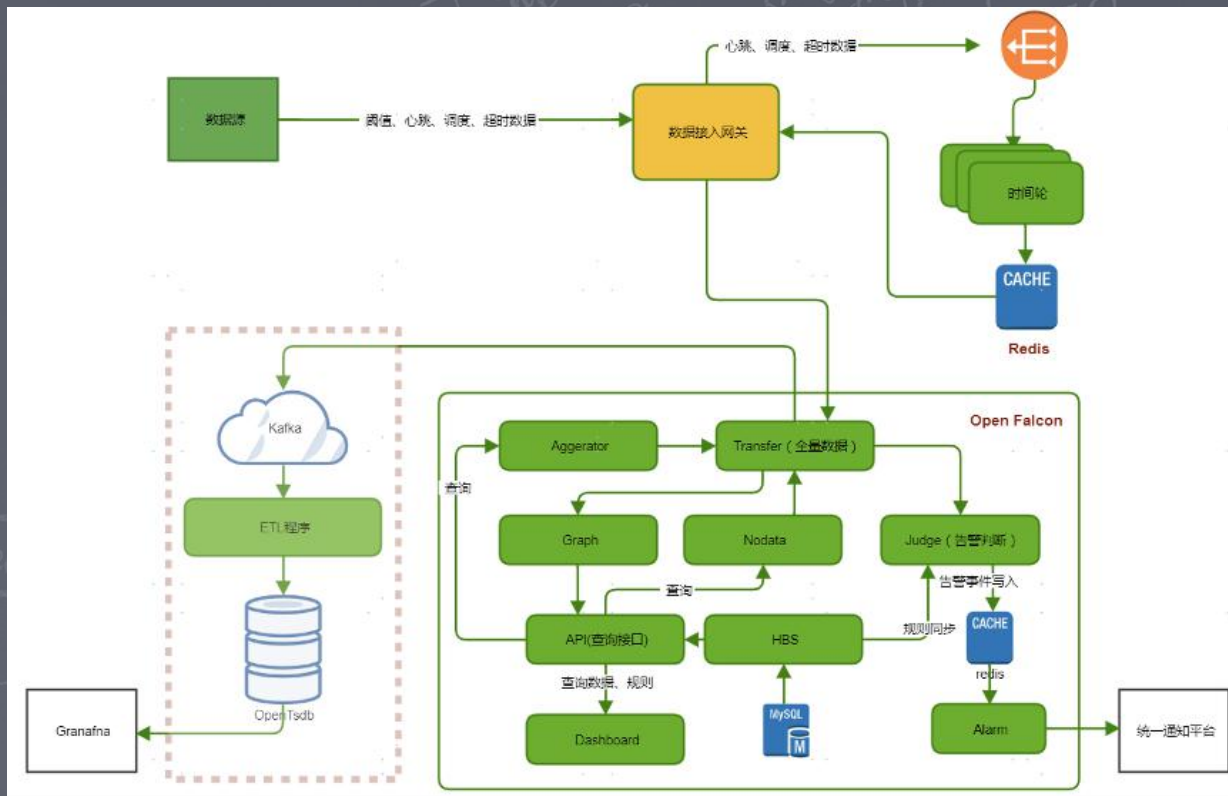
**cassandra**

Cassandra  
Apache Software Foundation

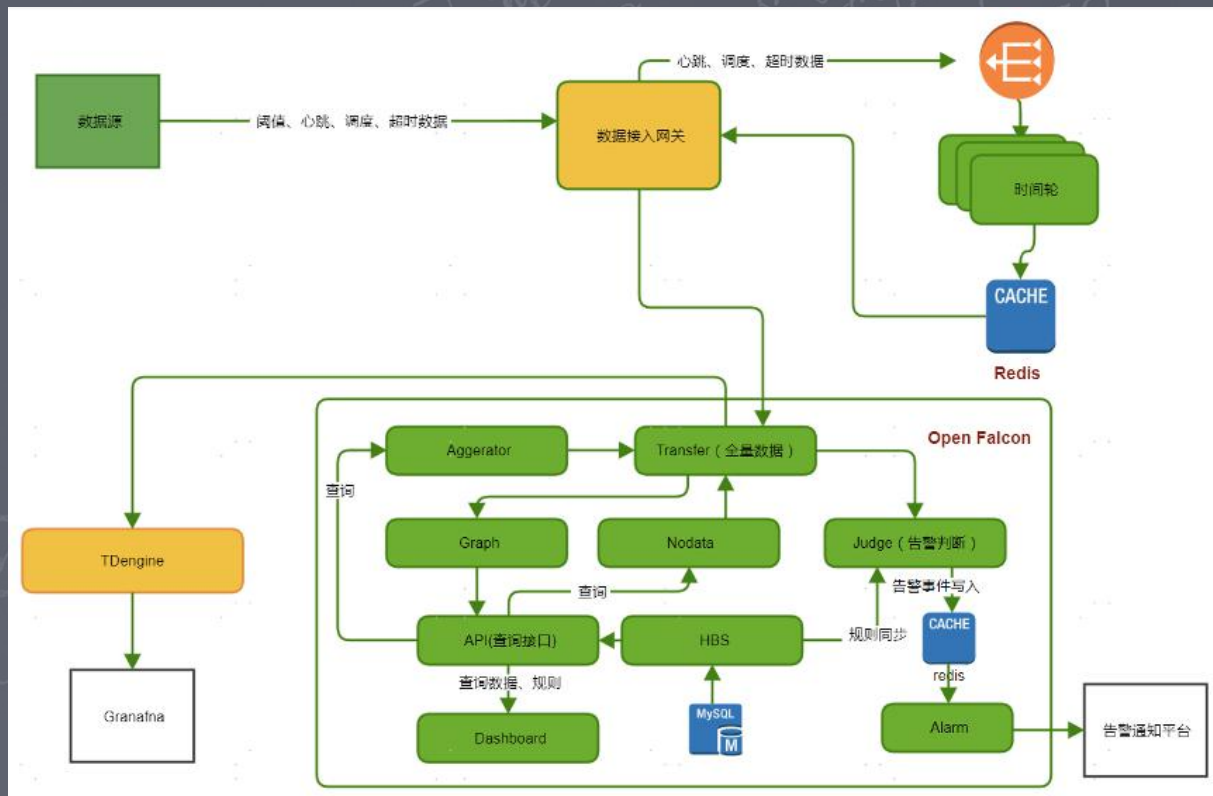
★ 5,867

# TDengine 最佳应用场景

# 顺丰科技大数据监控：改造前



# 顺丰科技大数据监控：改造后

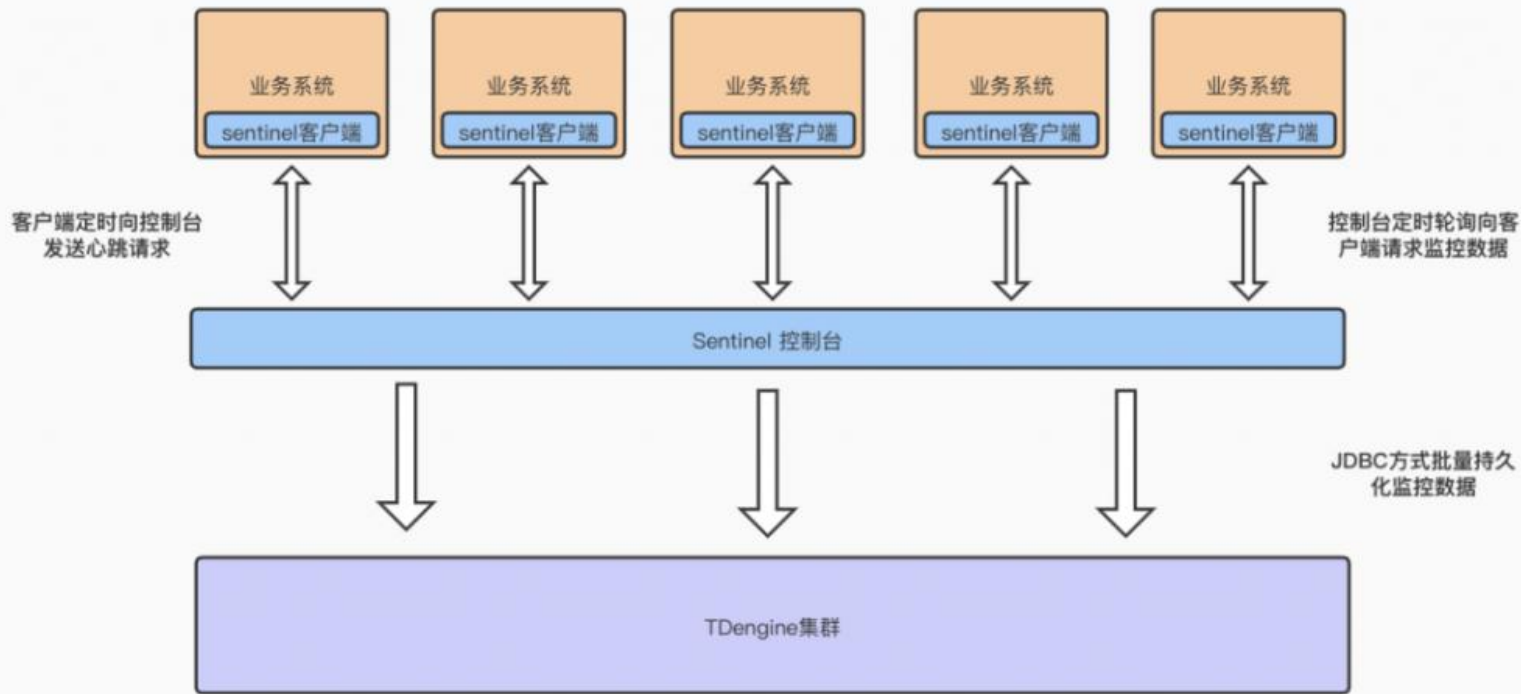




# 采用TDengine之后的表现

- 写入性能：集群写入速度最高达到90w条/s，通常为20w条/s
- 查询性能（OpenTSDB需要十几秒）
  - 在使用预计算函数情况下，查询p99都在0.7秒以内
  - 在做大跨度（6个月）非预计算查询情况下，首次查询耗时在十秒左右，后续类似查询耗时会有大幅下降（2-3s）
- 成本方面
  - 服务器物理机从21台降至3台
  - 所需存储空间约为OpenTSDB+HBase的1/10

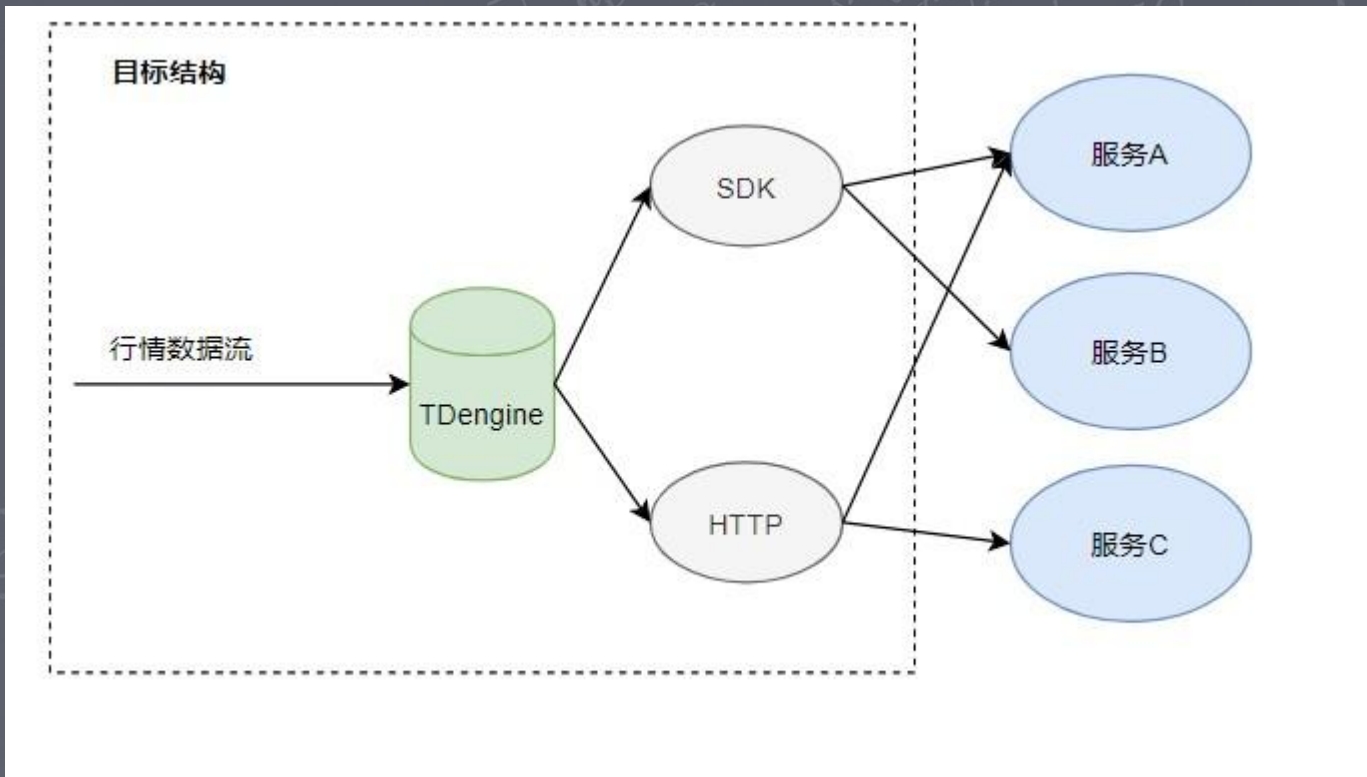
# 得物流量监控



# 采用TDengine之后的表现

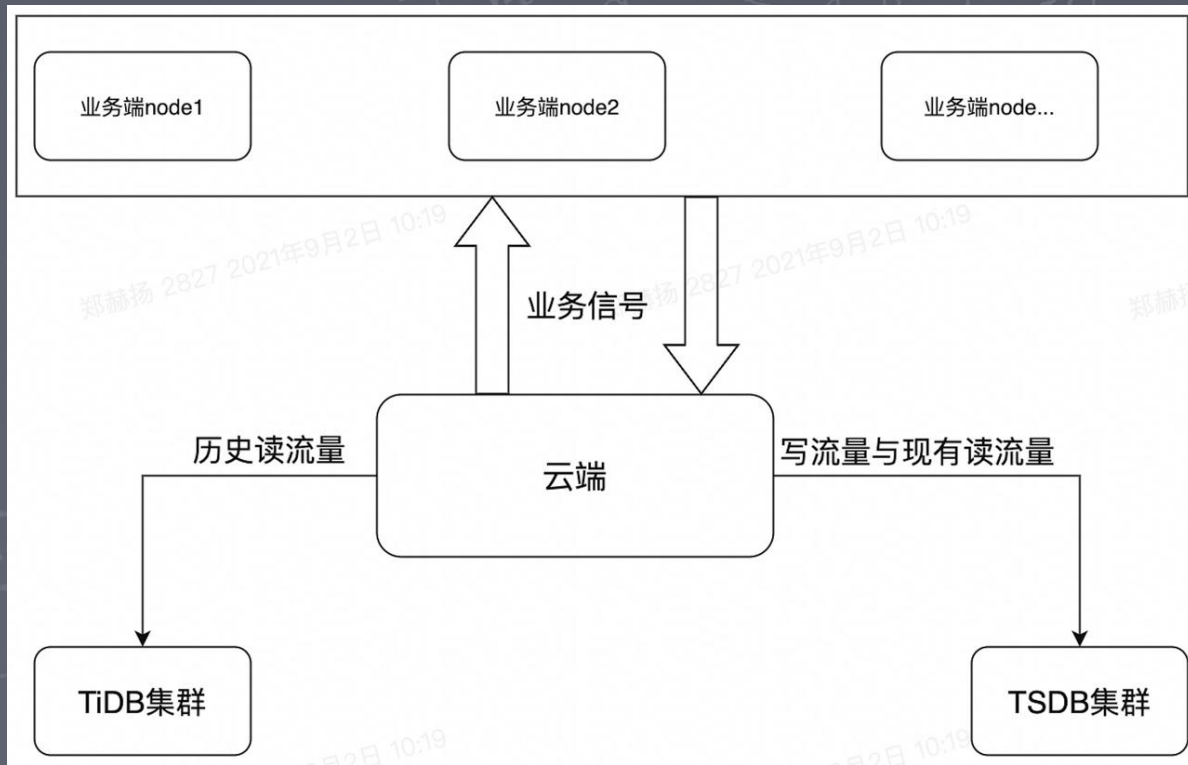
- 大批量写入，耗时基本在10ms
- 可以通过调整SQL语句进一步优化
- 存储容量：压缩率达10%
- 查询性能优异

# 同花顺组合管理业务

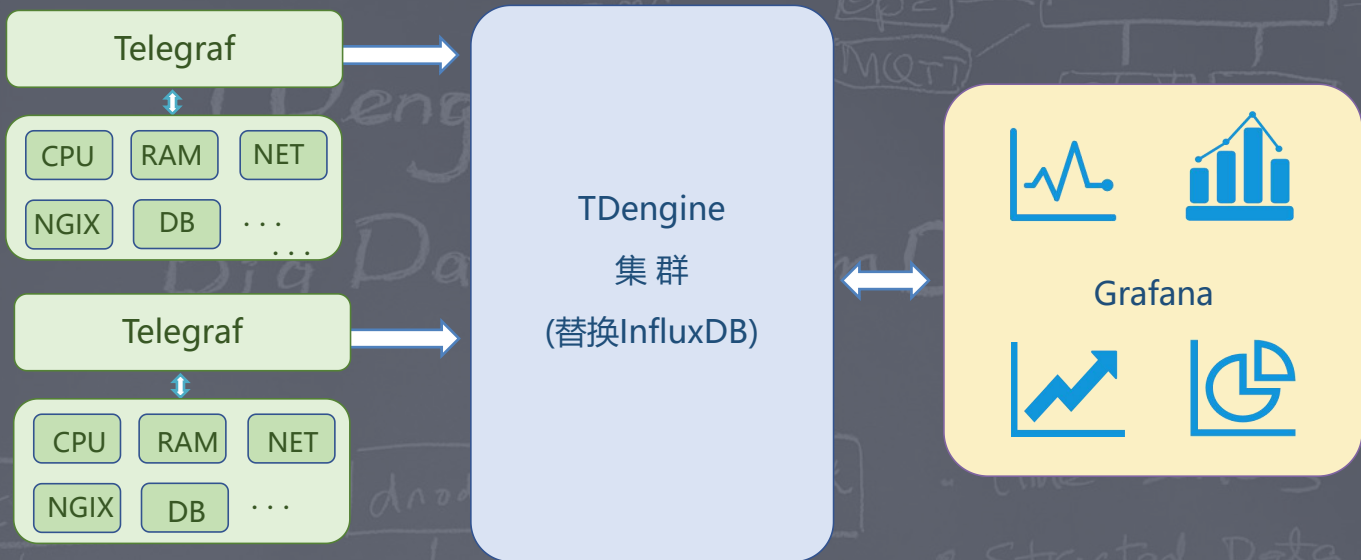




# 理想汽车车联网场景



# InfluxDB用户



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

# 与InfluxDB相比的优势

- 集群: TDengine集群开源, 而InfluxDB集群闭源
- 写入速度: TDengine是InfluxDB的2-8倍
- 查询速度: TDengine是InfluxDB的1-20倍, 特别是历史数据查询, 差异明显
- 存储效率: TDengine存储空间是InfluxDB的50%
- 缓存: TDengine具备缓存, 不需要安装redis

# 从InfluxDB迁移的成本

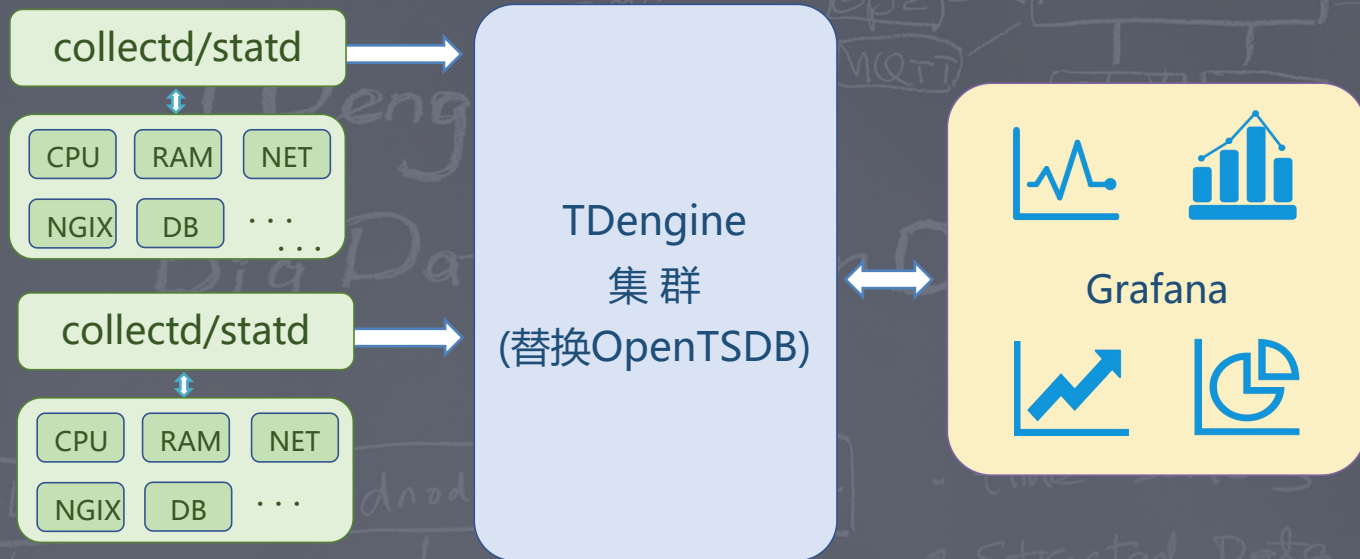
## 简单三步

1. 将TDengine 与涛思数据提供开源的 BLM 同时部署
2. 将Telegraf指向InfluxDB的URL改为BLM即可
3. 在Grafana里配置涛思数据提供的对应dashboard，或自己重新配置

不用写任何代码，轻松搞定



# OpenTSDB用户



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

# 与OpenTSDB相比的优势

- 部署：部署极其简单，不用HBase, Zookeeper等臃肿庞大的系统
- 写入速度：TDengine是OpenTSDB的10倍以上
- 查询速度：TDengine是OpenTSDB的10倍以上
- 存储效率：TDengine存储空间是OpenTSDB的10%
- 缓存：TDengine具备缓存，不需要安装redis
- 流式计算：TDengine支持连续查询，无需第三方软件

# 从OpenTSDB迁移的成本

## 简单三步

1. 将TDengine 与涛思数据提供开源的 BLM 同时部署
2. 将collectd/statd等Agent指向OpenDB的URL改为BLM即可
3. 在Grafana里配置涛思数据提供的对应dashboard，或自己重新配置

不用写任何代码，轻松搞定

# 灯塔计划

对象：市值或估值在10亿美金以上的公司，有时序数据处理的场景和需求。

计划的价值：

1. 产品赋能：利用TDengine提升时序数据处理能力，大幅降低TOC
2. 原厂支持：涛思数据免费提供2年企业级技术支持，免除后顾之忧
3. 人才培养：涛思数据输送讲师，帮助企业专家深入掌握TDengine
4. 品牌推广：与参加企业联合进行品牌推广活动，提升影响力



# About TAOS Data

北京涛思数据科技有限公司(TAOS Data) 专注时序空间数据的采集、存储、查询、计算和分析。不依赖任何开源或第三方软件，开发了拥有自主知识产权、自主可控的高性能分布式时序空间数据引擎TDengine，公司于2017年6月获得明势资本和蛮子基金的天使投资，后续又获得GGV、红杉中国、经纬中国、永辉瑞金等多家专业机构近7000万美元投资。



北京望京保利国际广场

公司创始人陶建辉在美国留学工作十多年后，回国创业，曾成功创办了“和信”与“快乐妈咪”两家高科技企业。

公司研发团队全部毕业于名牌大学，都拥有硕士或博士学位，在分布式计算、数据存储和数据库上有多年的研发经验。



# 全球敏捷运维峰会

THANK YOU !

