

QCon 全球软件开发大会 【北京站】2016

从InfluxDB看时 序数据的处理

陈超
七牛云 技术总监

QCon

2016.10.20~22

上海·宝华万豪酒店

全球软件开发大会 2016

[上海站]



购票热线: 010-64738142

会务咨询: qcon@cn.infoq.com

赞助咨询: sponsor@cn.infoq.com

议题提交: speakers@cn.infoq.com

在线咨询 (QQ): 1173834688

团 · 购 · 享 · 受 · 更 · 多 · 优 · 惠

7折

优惠 (截至06月21日)
现在报名, 立省2040元/张

Agenda

- 时序数据库
- InfluxDB的基本概念
- InfluxDB存储引擎的演化
- InfluxDB的cluster设计
- 七牛的时序数据架构
- 总结

Agenda



- *时序数据库*
- InfluxDB的基本概念
- InfluxDB存储引擎的演化
- InfluxDB的cluster设计
- 七牛的时序数据架构
- 总结

时序数据库

- A **time series database** (TSDB) is a software system that is optimized for handling **time series** data, arrays of numbers indexed by **time** (a datetime or a datetime range). In some fields these **time series** are called profiles, curves, or traces.

时序数据库的排名

19 systems in ranking, April 2016

Rank			DBMS	Database Model	Score		
Apr 2016	Mar 2016	Apr 2015			Apr 2016	Mar 2016	Apr 2015
1.	1.	1.	InfluxDB	Time Series DBMS	3.96	+0.01	+2.66
2.	2.		RRDtool	Time Series DBMS	2.53	-0.08	
3.	3.		Graphite	Time Series DBMS	1.56	-0.01	
4.	4.		OpenTSDB	Time Series DBMS	1.41	+0.02	
5.	5.	↓ 2.	Kdb+ 	Multi-model 	1.21	-0.04	+0.32
6.	↑ 7.		Druid	Time Series DBMS	0.23	+0.08	
7.	↓ 6.		KairosDB	Time Series DBMS	0.19	+0.03	
8.	↑ 9.		Prometheus	Time Series DBMS	0.15	+0.02	
9.	↓ 8.		Axibase	Time Series DBMS	0.11	-0.03	
10.	10.		Riak TS	Time Series DBMS	0.02	-0.02	
11.	11.		TempoIQ	Time Series DBMS	0.01	-0.00	
12.	12.		Blueflood	Time Series DBMS	0.00	±0.00	
12.	12.		Cityzen Data	Time Series DBMS	0.00	±0.00	
12.	12.		Hawkular Metrics	Time Series DBMS	0.00	±0.00	
12.	12.		Infiniflux	Time Series DBMS	0.00	±0.00	
12.	12.		Newts	Time Series DBMS	0.00	±0.00	
12.	12.		SiteWhere	Time Series DBMS	0.00	±0.00	
12.	12.		TimeSeries.Guru	Time Series DBMS	0.00	±0.00	
12.	12.		Yanza	Time Series DBMS	0.00	±0.00	

时序数据库关键特性

- 以时间为维度的高效查询
- 方便的down sampling
- 自动简单高效的处理过期数据
-

时序数据的场景

- 监控
- IoT

Agenda

- 时序数据库
- **InfluxDB 的基本概念**
- InfluxDB的使用
- InfluxDB存储引擎的演化
- InfluxDB的cluster设计
- 七牛的时序数据架构
- 总结

选择InfluxDB的理由

- 无外部依赖
- 快速使用
- 优雅的REST API
- 强大的基于SQL的查询语言
- 水平扩展(*)
- 纯Go编写
-

InfluxDB的核心概念

- measurement
- tag/tagset (tag被索引)
- series (measurement + tagset)
- field
- timestamp
- continuous queries
- retention policy

retention policy

- 指定数据的过期时间
- 指定数据副本数
- `CREATE RETENTION POLICY two_hours ON food_data DURATION 2h REPLICATION 1`
- `SHOW RETENTION POLICIES ON food_data`

line protocol

- measurement[,tag1,tag2,...] field1[,field2,...] ts
- cpu,host=qn00001 value=0.1 1434055562000000000

写数据

- `curl -i -XPOST 'http://localhost:8086/write?db=mydb' --data-binary 'cpu_load_short,host=server01,region=us-west value=0.6414340555620000000000'`
- 也可以批量写入数据

查询数据

- `curl -G 'http://localhost:8086/query?pretty=true' --data-urlencode "db=mydb" --data-urlencode "q=SELECT value FROM cpu_load_short WHERE region='us-west'"`
- `curl -G 'http://localhost:8086/query?pretty=true' --data-urlencode "db=mydb" --data-urlencode "q=SELECT value FROM cpu_load_short WHERE region='us-west';SELECT count(value) FROM cpu_load_short WHERE region='us-west'"`

continuous queries

- ```
CREATE CONTINUOUS QUERY cq_30m ON food_data
BEGIN
SELECT mean(website) AS mean_website, mean(phone) AS mean_phone INTO
food_data."default".downsampled_orders FROM orders GROUP BY time(30m)
END
```



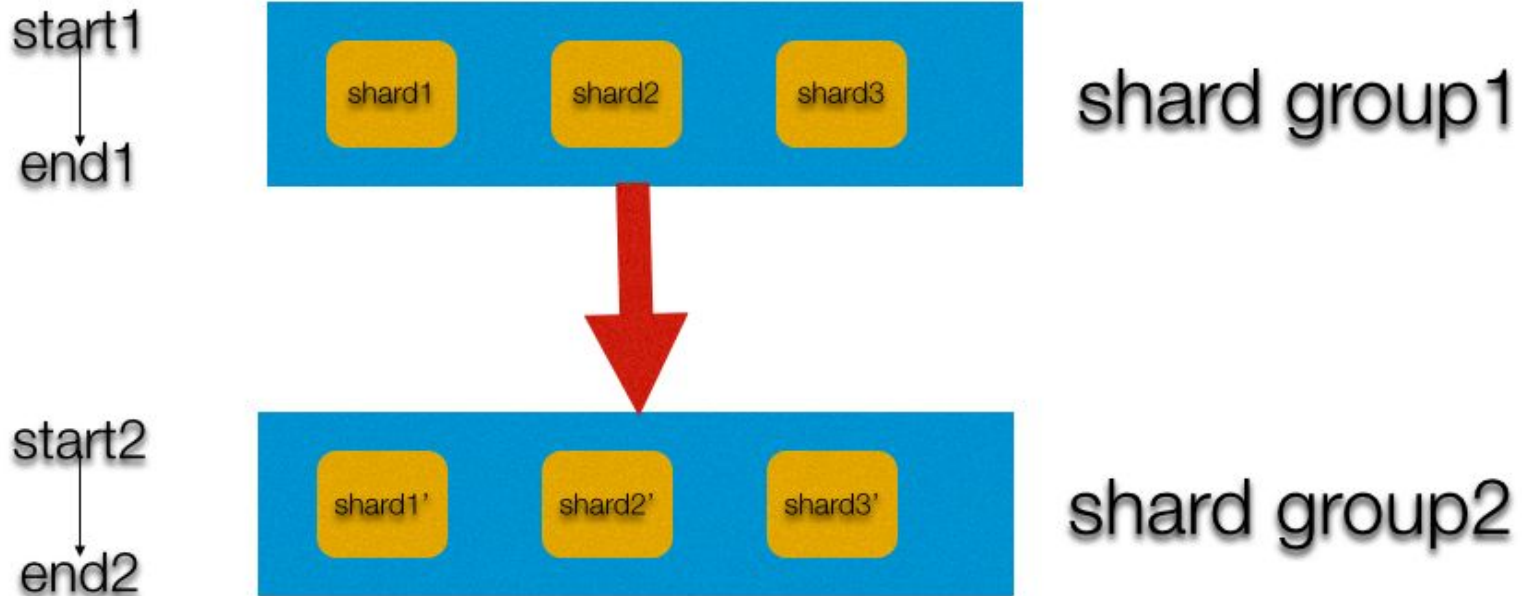
# query language

- `SELECT * FROM h2o_feet WHERE time > now() - 7d`
- `SELECT * FROM h2o_feet WHERE location = 'coyote_creek' AND water_level > 8`
- `SELECT MEAN(water_level) FROM h2o_feet GROUP BY location`
- `SELECT MEAN(water_level) FROM h2o_feet GROUP BY time(10m), location`
- `SELECT water_level INTO h2o_feet_copy FROM h2o_feet WHERE location = 'coyote_creek'`
- `SELECT water_level INTO h2o_feet_copy FROM h2o_feet WHERE location = 'coyote_creek' limit 10`
- .....

# Agenda

- 时序数据库
- InfluxDB的基本概念
- **InfluxDB 存储引擎的演化**
- InfluxDB的cluster设计
- 七牛的时序数据架构
- 总结

# Shard & ShardGroup



# 演化

- LSM Tree(LevelDB) ==> LMDB(BoltDB) ==> TSM Tree(tsm)

# B+ Tree

- 在keyspace做append是非常高效的
- 在每一个time series上都要做append操作
- 最终演变成随机插入，而不是append only的操作

# LSM Tree

- Cassandra, Hbase , LevelDB etc ...
- Log(WAL) → Memtable → SStable

# LevelDB

- 较高的写吞吐      ✓
- 内置压缩            ✓
- 不支持hot backup    ×    (RocksDB & HyperLevelDB)
- 过期数据支持不好    ×    (LSM的delete比较昂贵)
- too many open files   ×    (shard的设计及LevelDB自身的设计导致)

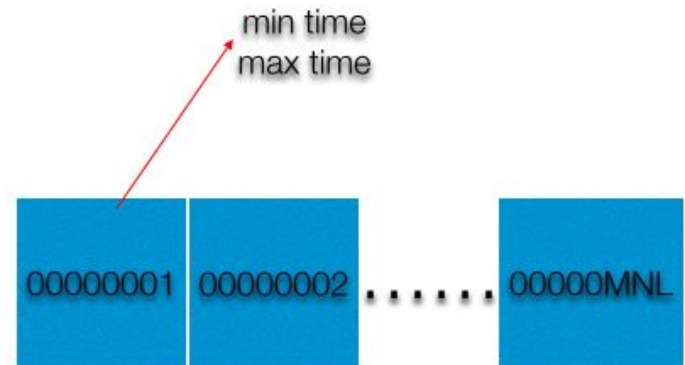
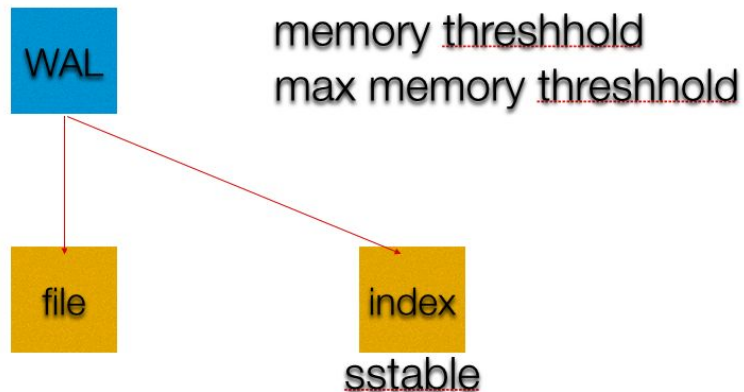
# BoltDB

- 几乎就是一个用Go写的LMDB
- 一个数据库一个文件
- 读写吞吐都比较好
- IOPS spike严重 (引入WAL缓解)



# 目前的方案--TSM

- 本质上还属于LSM
- meta files(启动时加载)
- data files(时间上连续且无交叉)

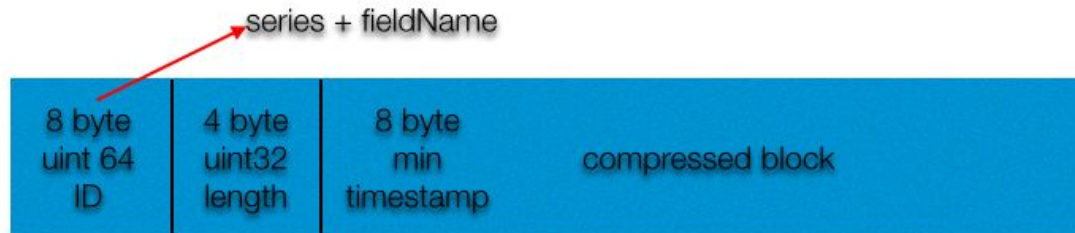


# data files

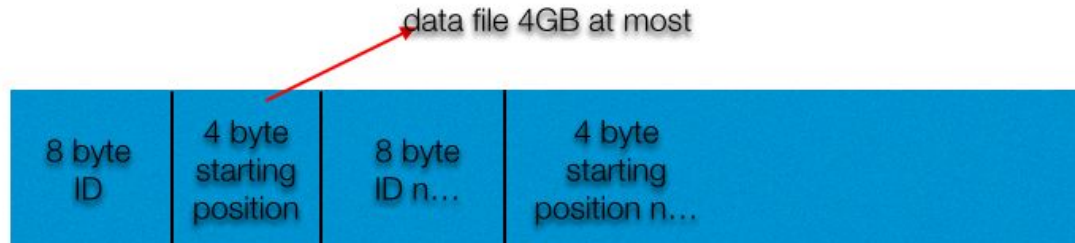
data files:



data block:



index block:



# Agenda

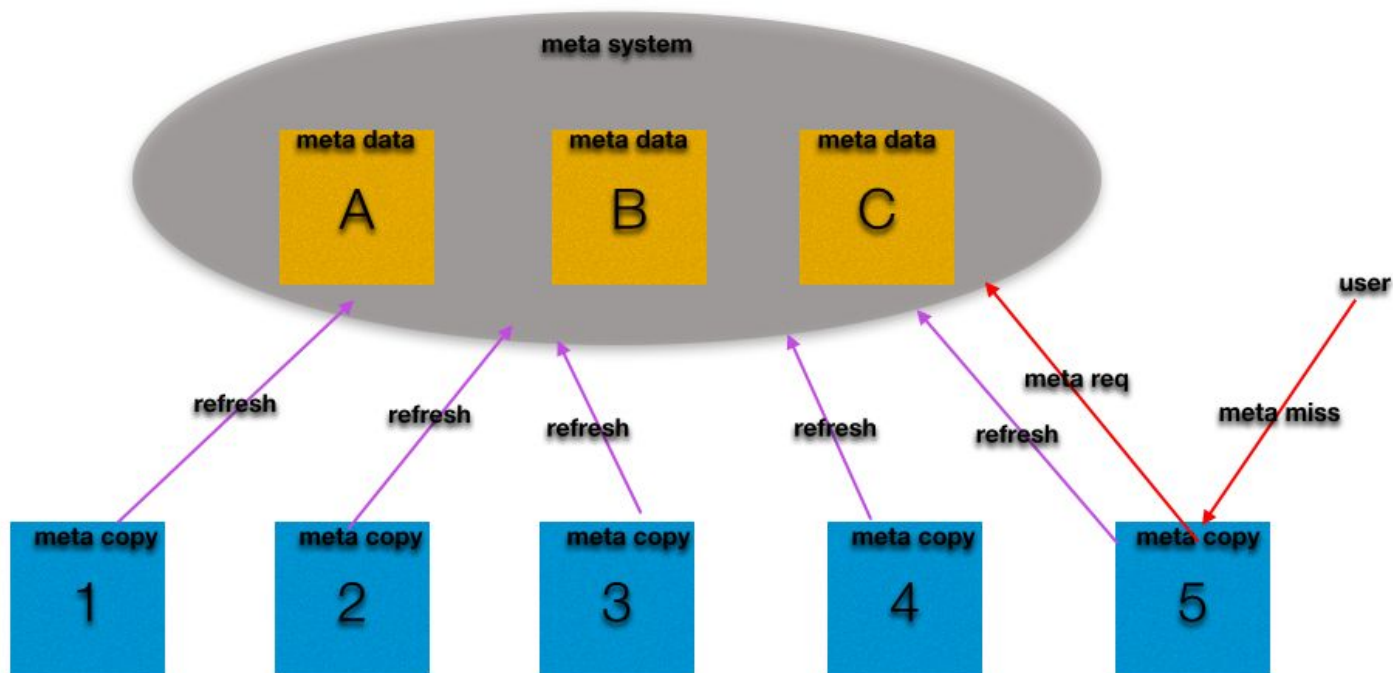
- 时序数据库
- InfluxDB的基本概念
- InfluxDB存储引擎的演化
- ***InfluxDB的cluster设计***
- 七牛的时序数据架构
- 总结

# CAP

- P是必需的 (CP or AP)
- metadata是CP
- write是AP

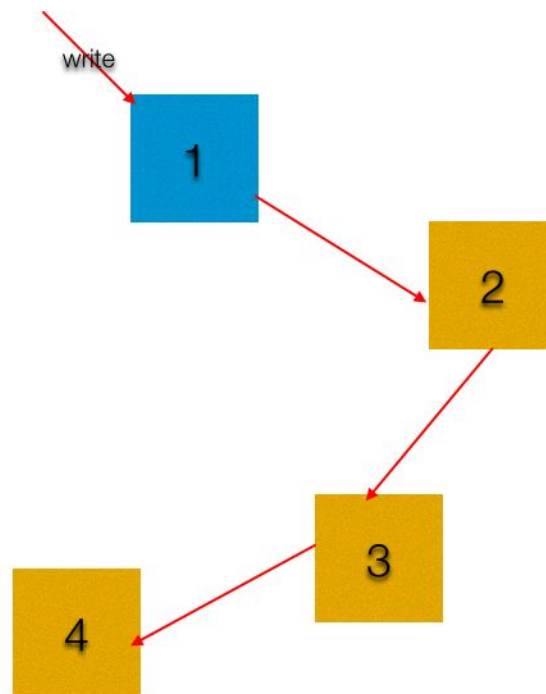
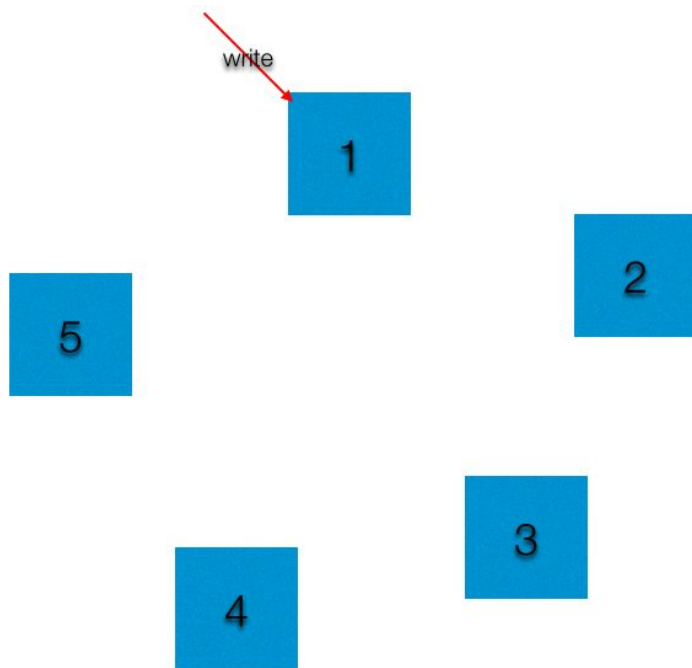
# metadata system -- CP

- 基于raft
- 集群中meta node(id, hostname)
- database, retention policies, series ,continuous queries, etc ...
- 用户与权限
- shard group & shard(起始时间等)



# 写 -- AP

- 一致性要求(Any, One, Quorum, All)



# 容错

- hinted handoff (ttl + full queue)
- anti-entropy repair (merkle tree)
- conflict resolution (greater value wins)

# 不过.....

- InfluxDB从0.12版本开始不再将其cluster源码开源



# Agenda

- 时序数据库
- InfluxDB的基本概念
- InfluxDB存储引擎的演化
- InfluxDB的cluster设计
- *七牛的时序数据架构*
- 总结

# 七牛在InfluxDB上做了哪些工作

- 做了相当多的测试
- 提了不少PR，大多是cluster方面的。
- 定制了内部的版本

# 七牛如何使用InfluxDB

- 单node单实例 (最早期放些不敏感数据，数据量也较小)
- 将InfluxDB放入内部的容器平台中
- InfluxDB cluster (差点让人崩溃)
- 统一的时序数据平台(TSDB)

# 七牛TSDB平台

- 抛弃InfluxDB官方的cluster的方案(所以他们闭源这部分代码我们表示无压力)
- 只将InfluxDB作为存储
- 抽象出repo, 一个repo可以有很多series
- 自研了调度器, 数据计算框架及高可用模块等
- agent负责上报各种信息

# 七牛TSDB平台

- 可拔插的架构
- 超强的scale能力
- 目前还在快速演化中

# Agenda

- 时序数据库
- InfluxDB的基本概念
- InfluxDB存储引擎的演化
- InfluxDB的cluster设计
- 七牛的时序数据架构
- *总结*

# InfluxDB的使用建议

- 用0.10及以上版本
- 单实例，目前的版本已经比较好
- 最好别用集群模式
- 对数据有高可用要求时可以用官方的Relay方案
- 有问题，尽量先看下社区的issue

- we are hiring!!!





# THANKS!