云栖社区
yq.aliyun.com

Terark.com
——重新定义数据技术

雷鹏
Terark Inc. CTO

主办单位：
Alibaba Group
战略合作伙伴：
扫码观看大会视频

--Preface--

扫码观看大会视频

Terark是一个数据技术提供商

Terark成立于2015年11月

致力于研发领先世界的高压缩存储和高性能检索技术

Terark目前的性能已经大幅超越了rocksdb等开源存储引擎

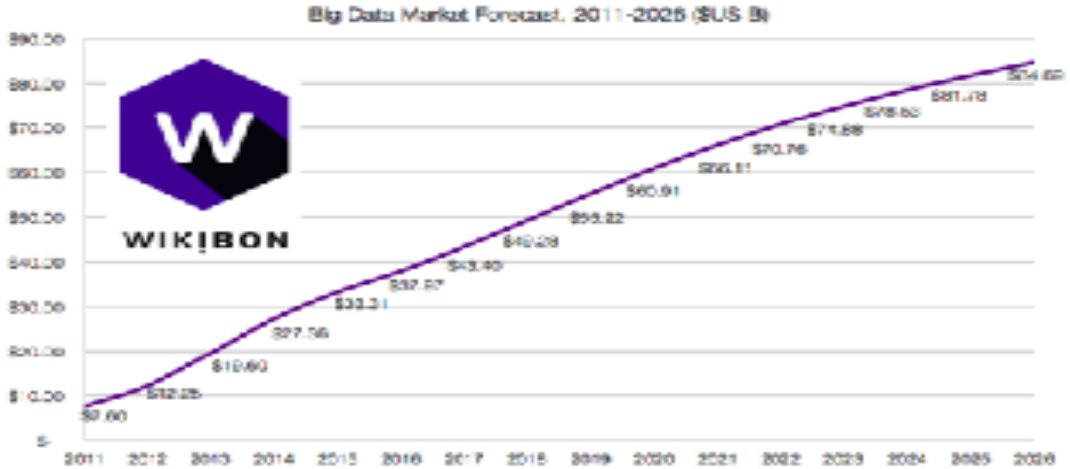
我们的存储引擎能为典型的大数据应用节省30%以上的成本

扫码观看大会视频

市场需求巨大

- 世界上92%的数据是在过去两年产生, 并且还在加速, 每年增长40%~50%
- 云端, 终端, 都在处理越来越大的数据量, 需要新型数据技术极高的性能和存储能力

Wikibon报告2011~2026 Market Forecast



Big Data Market Revenue Trend 2011~2026

扫码观看大会视频

核心技术

可检索压缩 SeComp (Seekable Compression) 技术

这个技术拥有超高的压缩率(一般在5倍以上), 同时可以直接在压缩的数据上进行定点访问(微秒级), 避免了传统数据库使用的分块压缩技术固有的缺点

索引技术

我们实现了多种独有的索引技术。其中最关键的是“数据即索引，索引即数据”，从而节约了空间，同时并保持甚至提高了性能

存储引擎技术

Terark存储引擎可以便捷的融入各种数据系统, 为其提供高效的核心存储技术,大幅提升整个系统的容量和性能

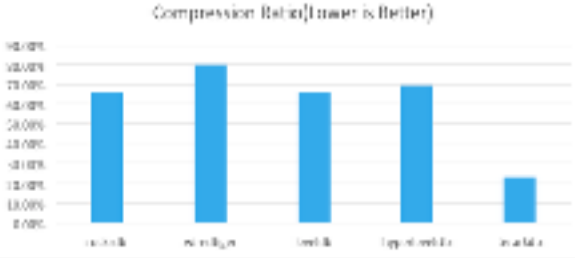
扫码观看大会视频

TerarkDB 是高性能、高压缩的存储引擎

★ 功能丰富：支持丰富数据类型的Schema定义和灵活的索引, 原生支持正则表达式检索

★ 性能强大：TerarkDB的QPS比同类产品提高1～2个数量级, 降低延迟, 提高吞吐量

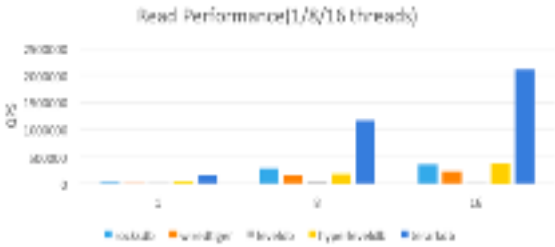
★ 容量大成本低：压缩率5倍以上 极大的降低IO压力, 提高数据容量, 降低基础设施成本



与谷歌、Facebook的产品相比

占空间更小同时检索更快

非时间空间的折衷, 而是技术革新



我们测试过一组800G的数据，Terark压缩后只有47.9G，同时Query延迟达到微秒级。使用Terark，只需要1台普通64G内存的服务器就够了，而用其它数据库可能就需要搭建一个小集群了。

适用场景

云平台　互联网　企业大数据　数据库　手机

扫码观看大会视频

TerarkDB 技术精要

- 功能概要
- 索引压缩
- 数据（Value）压缩
- Succinct
- TerarkDB 架构

扫码观看大会视频

索引压缩

	Hash	B+Tree	Terark Nest Succinct Trie
压缩率	膨胀	还行	很高
搜索速度	极快	较快	很快
精确搜索	支持	支持	支持
范围搜索	不支持	支持	支持
前缀搜索	不支持	支持	支持
正则搜索	不支持	不支持	支持
反向搜索(id到key)	可支持	不支持	支持

扫码观看大会视频

动态索引： TRB: Terark Thread Red Black Tree

只有 Left/Right，用数组下标代替指针；使用两个 bit 表示 threadtag，iterate更快

	传统 RBTree	B+Tree	TRB
数据结构消耗	4 ptr ☹	~ 0.75 keylen ☺	64 bits ☺
搜索速度	较快 ☺	很快 ☺	很快 ☺
数据耦合	紧耦合 ☹	紧耦合 ☹	松耦合 key可与结点分离
反向搜索(id到key)	不支持	不支持	支持

Key 数据可以保存在另外的数组，用平行的数组下标访问

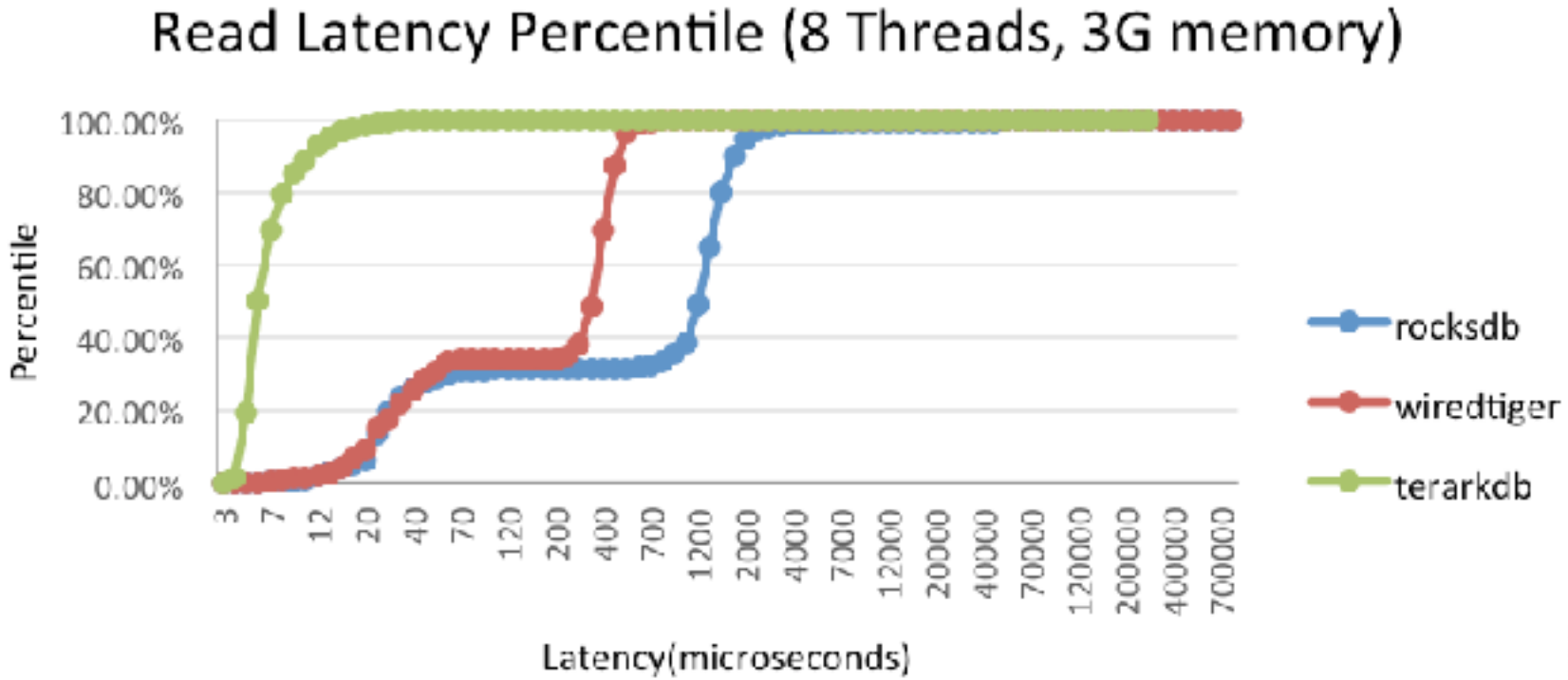
例如，以最小代价，用作链式 HashMap 的冲突链，用作 DFA 的状态转移表

扫码观看大会视频

数据（Value）压缩

	块压缩: leveldb, rocksdb, wiredtiger…	短数据: Terark Nest Succinct Trie	长数据: Terark Global Compression
压缩率	还行	很高	很高
随机读取速度	很慢	很快	很快
顺序读取速度	很快	较慢	很快
双缓冲问题	有	无	无
压缩速度	快	慢	慢

扫码观看大会视频


Read Latency Percentile (8 Threads, 3G memory)
Percentile
100.00%
80.00%
60.00%
40.00%
20.00%
0.00%
3
7
12
20
40
70
120
200
400
700
1200
2000
4000
7000
12000
20000
40000
70000
120000
200000
400000
700000
Latency(microseconds)
rocksdb
wiredtiger
terarkdb

扫码观看大会视频

索引压缩: Succinct Tree，概念

Succinct Data Structure是一种能够在接近于信息论下限的空间内来表达对象的技术，通常使用位图来表示，用位图上的rank和select来定位。

虽然能够极大的降低内存占用量，但是实现起来较为复杂，目前开源的有SDSL-Lite。注意：Succinct数据结构的性能比相应的传统(基于指针)数据结构更低。

Terark 使用自己实现的Rank-Select，性能远高于开源实现。

二叉树"传统基本版"的表现形式:

```
struct Node { Node *left, *right; };
```

每个结点占用 2ptr，如果我们对传统方法进行优化，结点指针用最小的 bits 数来表达，N个结点就需要 $2 * \lceil log_2(N) \rceil$ 个 bits。

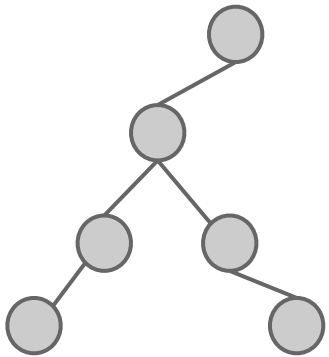
- 对比传统基本版和传统优化版，假设共有2^{16}个结点(包括null结点)，传统优化版需要2 bytes，传统基本版需要4/8 bytes
- 对比传统优化版和Succinct，假设共有10亿(~2^{30})个结点
  - 传统优化版每个指针占用$\lceil log_2(2^{30}) \rceil$ = 30 bits，总内存占用：$\left(\frac{2*30}{8}\right) * 2^{30} \approx 7.5GB$，
  - 使用Succinct，内存占用是$\left(\frac{2.5}{8}\right) * 2^{30} \approx 312.5MB$(每个结点2.5 bits，其中0.5bits是rank-select索引占用的空间)

扫码观看大会视频

索引压缩: Succinct Tree，图示

每个结点用两个bit表示, Pre-Order

DFUDS



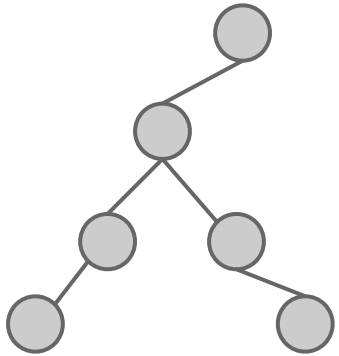
10110100100

Needs ***findopen, findclose, enclose***, which are much slower than rank/ select, rarely used

每个结点用两个bit表示, Level-Order

LOUDS

101110010000



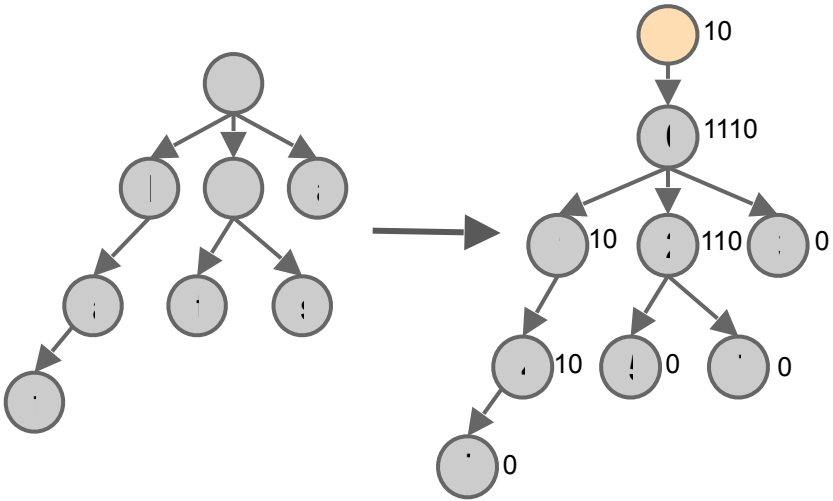
Simple and fast, small:

Parent(c) = rank0(select1(c))

Child(p, i) = select0(p) – p + i

扫码观看大会视频

存储了 *hat,is,it,a* 的Trie



对于第i个节点 = Node[i]，其孩子节点：

child[0] = Node[Select$_0$(i+1) - i]
child[1] = Node[Select$_0$(i+1) - i + 1]
child[2] = Node[Select$_0$(i+1) - i + 2]
…

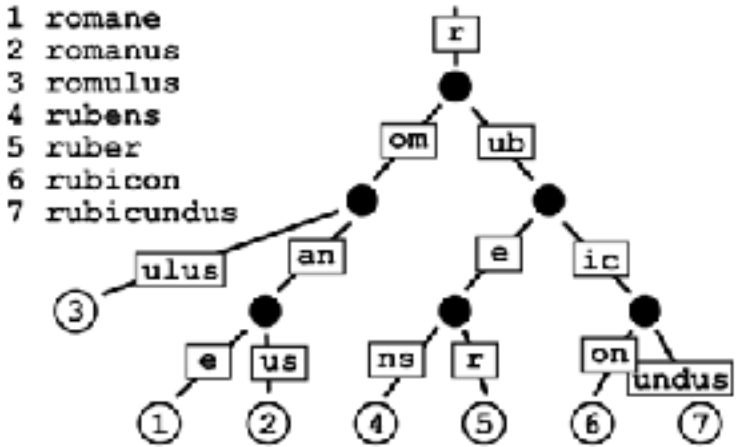
Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits	1	0	1	1	1	0	1	0	1	1	0	0	1	0	0	0	0
Node	**super**		**0**				**1**		**2**			**3**	**4**		**5**	**6**	**7**

扫码观看大会视频

Patricia Trie + 嵌套

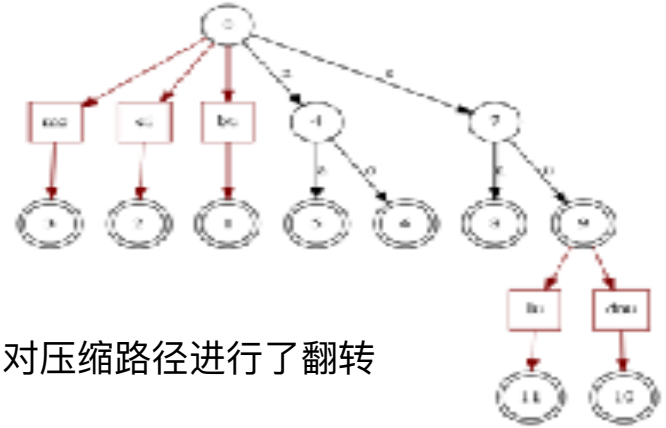
Patricia Trie: 路径压缩的Trie

路径压缩: 把一串仅包含一个孩子的结点压缩成一个



嵌套: 把压缩的路径存入另一个棵Trie

需要: Trie 数据结构支持“反向搜索”，即从结点搜索出能到达该结点的字符串(除了 Succinct，Double Array Trie也支持反向搜索，但无压缩效果)



对压缩路径进行了翻转

扫码观看大会视频

对数据(Value)的压缩

- 全局压缩
- 全局字典+局部字典，贪心最长匹配
- 可选熵编码
- 对短数据友好（~50字节）
- 数据集越大，压缩效果越好
- 定点访问（根据 record id）
- 可认为是 lz77 系列变种（加上熵编码，整体类似 gzip）

扫码观看大会视频

TerarkDB 架构

TerarkDB 最上层逻辑是“表”(Table)，每个表可以通过 Schema 定义其字段及相应的索引、特性等。

TerarkDB 可以嵌入其他上层数据库，如MongoDB、MySQL、SSDB等，只需要将其引擎替换掉即可。

TerarkDB目前在Github开源：https://github.com/Terark/terark-db

扫码观看大会视频

2018 [飞天·进化] APSARA EVOLUTION

表结构(Table)

云栖社区 yq.aliyun.com

一个“表”由多个“段”(Segment)组成，表中的所有记录的id，在逻辑上是连续且不变的。

Segment 中每一行有逻辑删除和物理删除的标记，物理删除的记录**一定也是**逻辑删除的，逻辑删除的**可能没有**物理删除。逻辑ID（record id）和物理ID(physical id) 之间通过 Rank Select 算法进行双向映射。

seg		del_mark	col[0]	col[1]	...	col[N-1]
s[0]	record[0]	0,0				
	record[1]	0,0				
	record[2]	1,0	← 逻辑删除，还没有物理删除			
s[1]	record[3]	1,1	← 逻辑删除，并且也物理删除			
	record[4]					

逻辑Id	0	1	2	3	4	5	6	7
物理删除标记	1	0	0	1	1	0	1	0
物理id = Rank0(逻辑id)	0(NA)	0	1	2(NA)	2(NA)	2	3(NA)	3

物理id	0	1	2	3
逻辑id = Select0(物理id)	1	2	5	7

扫码观看大会视频

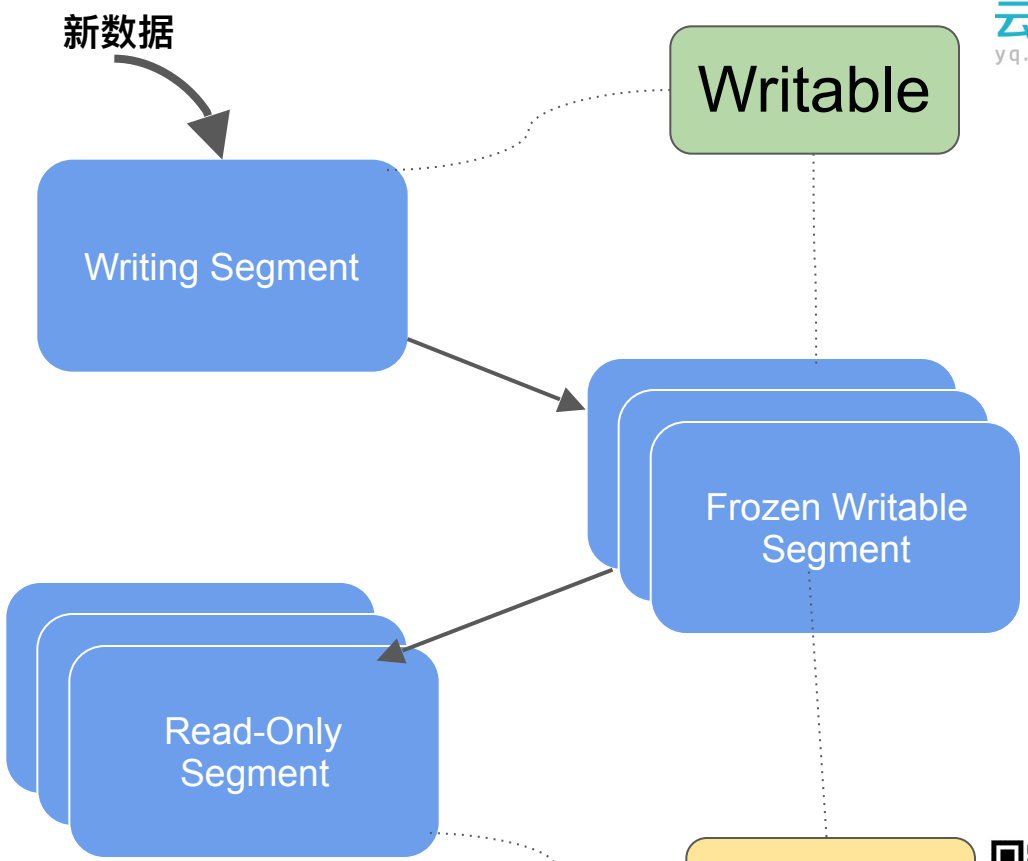
分段(Segment)

TerarkDB的段根据状态和类型分为：

- 只读段(Readonly Segment)
- 可写段(Writable Segment)
- 正在写段(Writing Segment)
- 冻结段(Frozen Segment)

正在写的段只能有一个，其他的有多个。

- Readonly/Writable 是按类型来区分
- Writing/Frozen 是按状态来区分
- Writing segment 的类型是



云栖社区 yq.aliyun.com
扫码观看大会视频

RocksDB on Terark: TerarkX

使用 Terark 索引压缩与数据压缩算法，实现 RocksDB SSTable

➢利用 Terark 的独特优势，通过 RocksDB，来造福整个生态

➢压缩率远高于 RocksDB 内置的压缩 (~3 倍)

➢随机读性能远高于 RocksDB 内置的压缩 (~10 倍)

➢Terark 数据压缩算法需要扫描数据源两遍，RocksDB 不支持 😕

➢Terark 为了实现高压缩与高速读，牺牲了压缩速度 😕 改进 RocksDB 底层架构

➢仅在最底层 level 使用 Terark　　减小写放大

➢使用 universal compaction

扫码观看大会视频

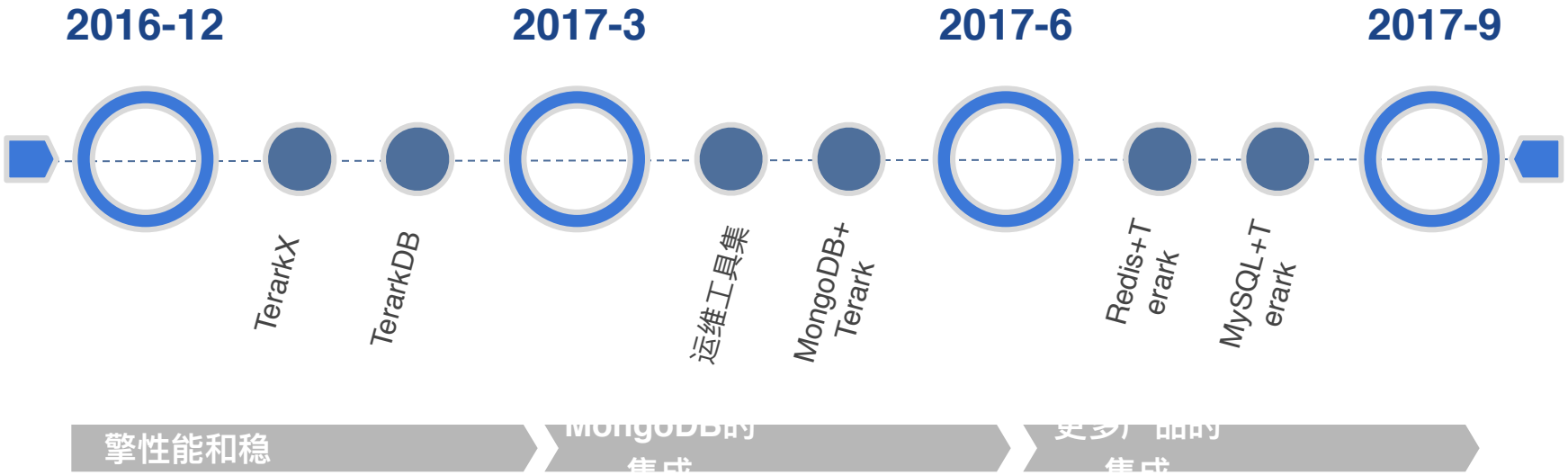
Terark Inc.

我们的技术非常适用于，对性能要求极高，对压缩率有特殊需求，
以及对海量数据高性能处理的场景

欢迎相关厂商的合作

扫码观看大会视频

Terark数据产品发布计划



扫码观看大会视频

云栖社区
yq.aliyun.com

2016 The Computing Conference

THANKS

Terark

扫码观看大会视频