Aerospike基础知识

一、客户端层（Client Layer）

客户端层能监控cluster的所有节点，并且能够自动感知所有节点的更新，同时掌握数据在cluster内的分布，特点：

- 高效性：Client的基础架构确保请求能够到相应的节点读写数据，减少响应时间。
- 稳定性：如果节点出错，不需要重启Client端，并且保持服务的正确性
- 连接池：为了减少频繁的open/close TCP操作，Client会在内部维护一个连接池保持长连接

二、分布层（Distibution Layer）

负责管理集群内数据的平衡分布。备份、容错和不用集群之前的数据同步。如果新增节点，只需要向集群中添加新的Aerospike server,不需要停止当前的服务，包含三个模块：

- 集群管理模块：基于Paxos-like Consensus Voting Process算法来管理和维护集群内的节点，并用心跳（Heart，包含active和passive）来监听所有节点的状态，用于监听节点间的连通性。
- 数据迁移模块：当有节点添加和删除是，该模块保证数据的重新分布，按照系统配置的复制因子确保每个数据块跨节点和跨数据中心复制。
- 事务处理模块：确保读写一致性，写操作先写Replica，再写master

事务模块主要负责以下任务：

- Sync/Async Replication(同步/异步复制)：为保证写一致性，在提交数据之前向副本传播更新并将结果返回客户端
- Proxy（代理）：集群重新配置期间客户端可能出现短暂过期，透明代理请求到其他节点。
- Duplicate（副本解析）：当集群从活动分区恢复时，解决不同数据副本之间的冲突。

三、数据存储层（Data Layer）

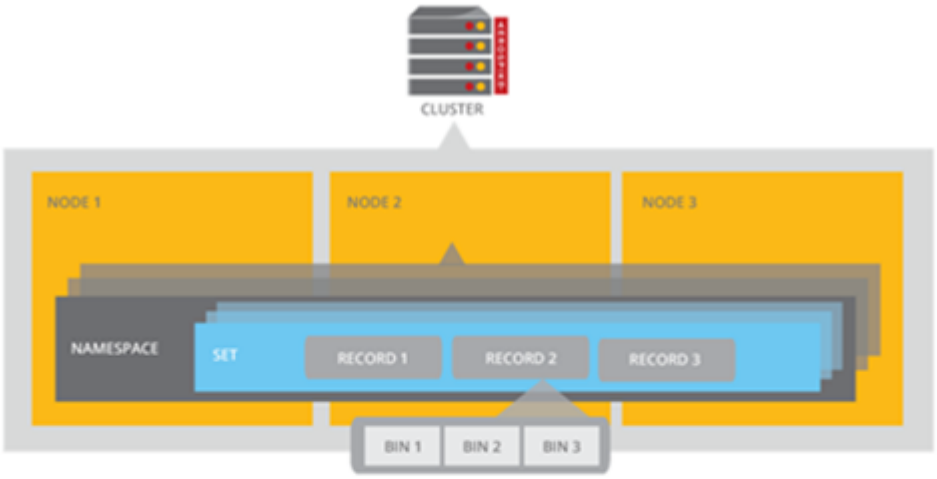
负责数据的存储，Aerospike是schema-less的键-值数据库，数据存储模式：

- 命名空间：数据存储在namespace中；namespace可以分为不同sets和records，每条record包含一个唯一的key和一个或多个bins值。
- 索引：Aerospike Indexs包含Primary Indexs和Second Indexs，索引只存储在内存中。
- 磁盘：与其它基于文件系统数据库的不同之处，在于Aerospike为了达到更好的性能选择了直接访问SSD的raw blocks（row device），并特别优化了Aerospike的最小化读、大块写和并行SSD来真加响应速度和吞吐量。

四、数据模型

Aerospike采用无模式（schema-less）数据模型，这意味着存储在库中的数据不符合严格模式。允许动态添加新类型的bin。尽管如此，仍然需要遵守bin名称与数据的对应关系。应用程序必须利用bin的一致性来保障查询和聚合的正确性。

Aerospike 键值存储（KVS）操作将键与一组命名值相关联。在集群启动时，Aerospike配置策略容器 - namespace（RDBMS 数据库） - 它控制一组数据的保留和可靠性要求。命名空间分为set（RDBMS 表）和record（RDBMS 行）。每条记录都有一个唯一的索引键，以及一个或多个包含记录值的bin（RDBMS 列）。



Aerospike	RDBMS
Namespace	Tablespace or Database
Set	Table
Record	Row
Bin	Column

Bin type
Integer
String
BLOB
List
Map

4.1 命名空间(namespace)

命名空间是顶级数据容器。命名空间实际上可以是数据库或一组数据库的一部分，如标准RDBMS中所述。在命名空间中收集数据的方式与数据的存储和管理方式有关。命名空间包含记录，索引和策略。策略规定了命名空间行为，包括：

- 如何存储数据：在DRAM或磁盘上。
- 记录存在多少个副本。
- 记录到期时间等

```
namespace test {
        replication-factor 2
        memory-size 4G
        default-ttl 30d # 30 days, use 0 to never expire/evict.

        storage-engine memory
}

namespace bar {
        replication-factor 2
        memory-size 4G
        default-ttl 30d # 30 days, use 0 to never expire/evict.

        storage-engine memory

        # To use file storage backing, comment out the line above and use the
        # following lines instead.
#       storage-engine device {
#               file /opt/aerospike/data/bar.dat
#               filesize 16G
#               data-in-memory true # Store data in memory in addition to file.
#       }
}
```

4.2 集合（sets）

在命名空间（namespace）中，记录可以属于一个称作集合（set）的逻辑容器。集合（set）为应用程序提供了收集记录组的能力。集合（set）继承了包含它的namespace的策略，也可为set指定其他的策略。例如，可以仅针对特定集合的数据指定二级索引，或者可以对特定集合执行扫描操作。

4.3 记录（records）

Aerospike数据库是一个行存储，专注于单个记录（RDBMS 行）。记录是数据库中的基本存储单元。记录可以属于命名空间或命名空间中的集合。记录使用密钥作为其唯一标识符。

记录包括以下内容：

Component	Description
Key	唯一标识，记录可以使用其密钥的散列来寻址，称为摘要。
Metadata	记录版本信息和配置到期时间，称为生存时间（TTL）
Bins	相当于传统数据中的字段

- key
  使用键在应用程序中读取或写入记录。当密钥被发送到数据库时，它及其设置信息被散列为160位摘要，用于解决所有操作的记录。因此，您在应用程序中使用密钥，而摘要用于寻址数据库中的记录。
- Metadata
  每条记录包括以下元数据

生成跟踪记录修改周期。该数字将在读取时返回给应用程序，可以使用它来确保自上次读取后未写入的数据已被修改

- **生存时间（TTL）**指定记录到期。Aerospike根据其TTL自动过期记录。每次写入记录时TTL都会递增。对于服务器版本3.10.1及更高版本，客户端可以将策略设置为在更新记录时不更新TTL。请参阅相应的客户端API文档以获取详细信
- **last-update-time（LUT）**指定更新的时间戳记录。这是数据库内部的元数据，不会返回给客户端。

4.4 bins

在记录中，数据存储在一个或多个bin中，包含名称和值。Bins不指定数据类型，数据类型由bin中包含的值定义，这种动态数据类型为数据模型提供了灵活性。例如，记录可以包含由字符串值bob组成的bin id。bin的值总是可以更改为不同的字符串值，但也可以更改为不同数据类型的值，例如整数。Bin中支持的数据类型：

- Integer
- String
- Bytes
- Double（3.6.0及更高的版本）
- List
- Map
- GeoJSON（3.7.0及更高的版本）
- native-language serialized (blobs)
- 命名空间或集合中的记录可以由非常不同的箱组成。记录没有架构，因此每条记录可以有不同的bins。您可以在记录的生命周期中的任何位置添加和删除容器。
- 目前，命名空间中并发唯一bin名称的数量限制为32K。这是由于优化的字符串表实现。

4.5 已知限制

AS数据库中的限制摘要

Item	限制	详细限制
Set	Set的数量	每个命名空间限制为1023个Set
Set	名称	集合名称不能包含‘：’或‘；’
Bin	bins的数量	每个命名空间最多有32,767个bin。这是硬编码的，不能更改。
Bin	名称长度	任何单个字节的15个字符。不允许使用双字节字符。对于4.2之前的服务器版本，限制为14个字符。
Namespace	数量	企业最多32个，社区最多2个（从4.0版开始）。集名不能包含’：‘或’;’ 字符。
Namespace	存储	命名空间就像传统SQL数据库的表空间。请注意，您可以将多个SSD与命名空间关联，但用作原始设备的任何SSD只能与单个命名空间关联。每个命名空间最多限制为128个设备。
Namespace	名称	命名空间名称不能包含’：‘或’;’ 字符。
Record	数量	实际数量受RAM和存储限制。索引条目的每条记录占用64个字节。索引条目仅存储在RAM中。密钥本身实际上并不存储在索引中，而是密钥的哈希值（使用RIPE-MD 160算法）。这个带有开销的哈希值恰好是64个字节。在给定节点上每个命名空间的最大记录数限制为Community Edition上的4,294,967,296（由于4个字节用于存储引用，因此为2 ^ 32），企业版上为34,359,738,368（2 ^ 35）。这表示2TiB的RAM。
Record	总数量大小	每个记录值大小限制为写入块大小（版本4.2及更高版本允许的最大大小为8388608（或8M）。对于4.2之前的版本，允许的最大大小为1048576（或1M）。较大write-block-size可能会对性能产生负面影响。），对于Flash设备，通常在配置中设置为128KB，但从版本4.2开始可以增加到8MiB（对于先前版本，最大值为1MiB，这也是如果未在配置文件中指定，则为default。

Record	bins的数量	记录中的bin数没有内置限制，但命名空间中的bin总数有限制。目前这是32,767。Record Sets 记录只能属于一个集合。请注意，set和key都被放入哈希中，因此为了获得具有set的键的值，您必须知道该set。
Cluster Size	节点数	对于Enterprise Edition，群集中的最大节点数为128.对于运行heartbeat协议v2的3.14之前的版本，限制由配置paxos-max-cluster-size（默认值为31）定义。有关增加已在运行的群集的默认值的过程，请参阅此页面：增加最大群集大小。对于Community Edition，对于4.0之前的版本，最大节点数限制为31，对于4.0及更高版本，最大节点数限制为8。
Devices	限制	每个节点的每个命名空间的最大设备数（或文件数）在版本4.2时为128，对于版本低至3.12.1为64，对于之前的版本为32。

其它限制

Aerospike server item	limits
bin名称	<= 14个字符
set名称	<= 63 个字符(集合名称中不允许使用'：'或';')
namespace名称	<= 31个字符（名称空间名称中不允许使用'：'或';')
命名空间中所有键的总分类数	< 32 K
命名空间中的总二级索引	<= 256 indices
每个键的bin	< 32 K
索引名称	<= 255个字符（索引名称中不允许使用'：'或';')
hist-track-back	86400秒（切片10秒）
复制因子	=节点数是集群
命名空间的每个文件或磁盘的最大可配置大小	= 2 TiB
接口数量	<= 500（对于3.15之前的服务器版本，限制为50）

4.6 索引

AS中有两种索引：主键索引和二级索引

主键索引

在数据库中，最快且最可预测预测的索引是主键索引。在Aerospike中，主键索引是分布式哈希表技术与每个服务器中的分布式树结构的混合。命名空间（数据库）中的整个键空间使用健壮的散列函数划分为分区。共有4096个分区在群集节点上均匀分布。

在最低级别，Aerospike使用红黑内存结构。对于每个分区，可以有可配置数量的这种红黑结构，称为sprigs。在机器上配置正确数量的Spr会降低内存开销，同时优化并行访问。

主键索引在20字节哈希上 - 指定主键的摘要。虽然这会扩展某些记录的密钥大小（例如，只有8字节的整数键），但它是有益的，因为无论输入密钥大小或分布如何，代码操作都是可预测的。

当单个服务器发生故障时，第二个服务器上的索引立即可用。如果故障服务器保持关闭状态，则数据将开始重新平衡，并且复制的索引会在新节点上建立。

索引元数据

目前每个索引条目需要64字节。除了20字节的摘要之外，以下元数据也存储在索引中。

- **写生成：**跟踪密钥的所有更新; 用于解决冲突的更新。
- **到期时间：**跟踪密钥到期的时间。驱逐子系统使用此元数据。
- **最后更新时间：**跟踪对密钥的最后写入（Citrusleaf纪元）。用于冷重启期间的冲突解决，迁移期间的冲突解决（基于配置），谓词过滤，增量备份扫描和截断命令。
  存储地址：数据的存储位置（内存和持久性）。
  索引持久性
  为了保持吞吐量，主索引不会提交到仅存储到RAM。这允许高性能，高度并行写入。数据存储层可以配置为不使用存储。当Aerospike服务器启动时，它会遍历存储上的数据并为所有分区创建主索引。

快速重启功能

为了在最短的停机时间内实现快速集群升级，Aerospike支持快速重启。快速重启功能从Linux共享内存段分配索引内存。对于计划的关闭和重新启动Aerospike（例如，用于升级），在重新启动时，服务器只需重新连接到共享内存段，并激活主索引而无需对存储进行数据扫描。

单仓优化

在命名空间上启用Aerospike 单个bin功能可以提供比在每个记录上支持Aerospike bin结构时更低的内存使用率。当内存中的单bin命名空间中的所有值都是整数或双精度并且命名空间在索引中声明**数据时，**可以实现进一步的优化。然后重新使用主索引中保存的空间来存储整数或双精度值。这意味着命名空间所需的存储量只是其主索引所需的空间。

二级索引

辅助索引位于非主键上，允许您对一对多关系建模。在Aerospike中，二级索引是逐个bin指定的（如RDBMS 列）。这允许有效更新并最小化存储索引所需的资源量。

数据描述（DDL）确定要索引的bin和数据类型。使用Aerospike工具或API动态创建和删除索引。此DDL（类似于RDBMS模式）不用于数据验证 - 即使bin在DDL中作为索引，bin也是可选的。对于索引bin，更新记录以包含bin会更新索引。

索引条目是类型检查的，即如果您有一个存储用户年龄的bin，并且年龄值由一个应用程序存储为字符串而另一个应用程序存储整数，则整数索引会排除包含存储在索引bin中的字符串值的记录，而字符串索引排除存储在索引bin中的整数值的记录。

二级索引：

存储在RAM中以便快速查找。
构建在集群中的每个节点上，并与主索引位于同一位置。每个辅助索引条目仅包含对节点本地记录的引用。
包含指向集群中主记录和复制记录的指针。