

1. HBase-2.x新特性

1.1. HBase的版本定义

2. 主要特性

- 2.1. A new Region assignment manager
- 2.2. Offheaping of Read/Write
- 2.3. In-Memory Compaction
- 2.4. NettyRpcServer
- 2.5. Async RPC Client
- 2.6. RegionServer Group
- 2.7. Support for MOB
- 2.8. Support CF-level Storage Policy
- 2.9. 与1.x版本的兼容性

1. HBase-2.x新特性

2018年4月30日HBase发布了2.0的Release版本。HBase 的 2.0 版本承载了太多的 Features，共 [包含4551个Issues](#)，可以说是迄今最大的一个版本。

在1.0版本正式发布之前，我们看看几个比较大的版本相关信息：

- 0.90.0版本于2011年1月19日发布，共包含1028个Issues。
- 0.92.0版本于2012年1月23日发布，共包含695个Issues。
- 0.94.0版本于2012年5月14日发布，共包含423个Issues。
- 0.96.0版本于2013年10月19日发布，共包含290个Issues。
- 0.98.0版本于2014年2月16日发布，共包含1307个Issues。

而在2.0版本之前，共有如下几个Minor Releases：

- 1.1.0版本于2015年5月14日发布，共包含440个Issues
- 1.2.0版本于2016年2月22日发布，共包含647个Issues
- 1.3.0版本于2017年1月16日发布，共包含895个Issues
- 1.4.0版本于2017年12月18日发布，共包含665个Issues

对比于 1.0 版本，只能说 2.0 版本中承载了太多的改动。一个版本承载了太多的 Features 也许是一件好事，但从质量上说，在短期上可能会带来一点担忧。

1.1. HBase的版本定义

HBase 从 1.0.0 版本开始，在版本定义上正式遵循了 Semantic Versioning 规范：

一个版本号，由三部分组成： MAJOR.MINOR.PATCH，关于这三部分数字的变更，定义如下：

- MAJOR版本号变更：意味着可能带来不兼容的API变更。
- MINOR版本号变更：主要是增加了一些功能特性，但API接口却是向前兼容的。
- PATCH版本号变更：主要是针对Bug修复。

按照该定义，我们举例说明如下：

- 从1.0.0到2.0.0版本，可能带来一些不兼容的API变更。
- 从1.0.0到1.1.0版本，意味着合入了一些新的Feature，但API接口却是兼容的。
- 从1.0.0版本到1.0.1版本，只是在1.0.0版本基础之上，做了一些Bug修复，但没有合入新的Feature。

2. 主要特性

2.1. A new Region assignment manager

[HBASE-14350](#)、[HBASE-14614](#)

AssignmentManager V2("AMv2")基于Procedure V2实现，能够更快速的分配Region，维护的region状态机存储不再依赖于ZooKeeper，能够更好的应对Region长时间RIT问题。

2.2. Offheaping of Read/Write

[HBASE-11425](#)、[HBASE-15179](#)

减少对Heap内存的使用，改用Offheap区的内存，有效减少GC压力

2.3. In-Memory Compaction

[HBASE-17343](#)

重新设计了CompactingMemStore 替代 DefaultMemStore，CompactingMemStore中的数据达到一定大小以后，不是直接Flush成HDFS中的HFile文件，而是先Flush到内存中的一个不可改写的Segment，这样，内存中的多个Segments可以预先合并，当达到一定的大小以后，才Flush成HDFS中的HFile文件，这样做的好处是希望有效降低Compaction所带来的写IO放大问题。

MemStore中的数据达到一定大小以后，不是直接Flush成HDFS中的HFile文件，而是先Flush到内存中的一个不可改写的Segment，这样，内存中的多个Segments可以预先合并，当达到一定的大小以后，才Flush成HDFS中的HFile文件，这样做的好处是希望有效降低Compaction所带来的写IO放大问题。

<https://developer.aliyun.com/article/573702>

2.4. NettyRpcServer

[HBASE-17263](#)

HBase2.0 开始默认使用NettyRpcServer。使用Netty替代HBase原生的RPC server，大大提升了HBaseRPC的吞吐能力，降低了延迟

2.5. Async RPC Client

[HBASE-16833](#)、[HBASE-15921](#)

HBase2.0 Client不在是原来同步等待，而是利用异步RPC机制，大大提高Client端请求并发度，有效提高资源利用率，扩大吞吐。

2.6. RegionServer Group

[HBASE-6721](#)

在 HBase 2.0 中我们可以将 RegionServer 划分到多个逻辑 Group 中，这样可以提供多租户的能力。

2.7. Support for MOB

[HBASE-11339](#)

MOB特性使得HBase支持存储小于10MB 的中等媒体对象数据，这些小对象文件采用独立的HFile文件进行存储，相比原来直接存储大对象插入hbase，其读写效率更高；Mob数据存储还是以hfile格式存储，兼容HBase现有特性，如snapshot、bulkload、replication等。MOB数据文件有独立的compaction和expire clean机制，稳定性更可控。

让HBase更好的支持数KB甚至数MB级别的小对象存储，这些小对象采用独立的HFile文件进行存储，而且不参与普通的Compaction，这样不会因为小对象数据使得Compaction的写IO放大问题进一步恶化。

2.8. Support CF-level Storage Policy

[HBASE-14061](#)

支持通过“hbase.hstore.block.storage.policy”配置来设置HFile的存储策略，并且还支持CF级设置以覆盖配置文件中的设置。目前支持的存储策略包括ALL_SSD/ONE_SSD/HOT/WARM/COLD

2.9. 与1.x版本的兼容性

1.x版本的Client可以访问2.0版本的集群，进行正常的数据读写操作。但从1.x版本不停服务的情况下滚动升级到2.0版本是没法做到的，1.x版本的coprocessor/Endpoint不做改动的情况下也无法在2.0版本中正常使用。