

# Cassandra与HBase：双胞胎还是外表相似的陌生人？

知 [zhuanlan.zhihu.com/p/341426602](https://zhuanlan.zhihu.com/p/341426602)

大数据行业码农

最近在学极客时间的王磊老师的《Spark核心原理与实战》，看得很过瘾，正好看到Hbase与Cassandra的接入，

发现两者挺相似的，于是就发现了这篇文章，那我就用谷歌翻译翻译，方便阅读

原文：

[Cassandra vs. HBase: twins or just strangers with similar looks? !\[\]\(e3f8612927870f2e0f9f5989e6dd3064\_img.jpg\)](http://www.scnsoft.com/blog/cassandra-vs-hbase)  
[www.scnsoft.com/blog/cassandra-vs-hbase](http://www.scnsoft.com/blog/cassandra-vs-hbase)

Apache Cassandra和Apache HBase很像两个陌生人，您在街上遇到并认为是双胞胎。您并不是很了解它们，但是它们相似的身高，衣服和发型使您看不到它们之间的差异。但是，仔细观察后，您会发现这两个对象仅在远处看起来相同。

Cassandra和HBase具有许多相似之处，例如NoSQL宽列存储和BigTable的派生。例如，HBase没有查询语言，这意味着您将不得不使用基于JRuby的HBase Shell，并且需要使用诸如Apache Hive，Apache Drill之类的额外技术。虽然Cassandra可以拥有自己的CQL（Cassandra查询语言），但估计只有Cassandra专家认为这最有帮助。

## 1, 数据模型

### HBase

在这里，我们有一个表，该表由按行键(row keys)和列族(Column Family)组织的单元格(cell)组成。有时，列族（CF）具有许多列限定符(column qualifiers)，以帮助更好地组织CF中的数据。

单元格(Cell)包含一个值和一个时间戳。列是公共列限定符和公共CF下的单元格集合。

在一个表中，数据按字典顺序按1列行键进行分区，其中局部相关的数据紧密存储在一起，以最大化性能。行键的设计至关重要，必须在开发人员编写的算法中进行全面考虑，以确保高效的数据查找。

### Cassandra

在这里，我们有一个列族，它由按行键组织的列组成。列包含名称/键，值和时间戳。除了常规列之外，Cassandra还具有包含两个或多个子列的超级列(super column)。这些单元被分组为超级列族（尽管很少使用）。

在集群中，数据由多列主键进行分区，该主键获取一个哈希值，然后将其发送到令牌在数值上大于哈希值的节点。除此之外，数据还会写入额外数量的节点，这取决于Cassandra从业者设置的复制因子。其他节点的选择可能取决于它们在群集中的物理位置。

## HBase与Cassandra (数据模型比较)

术语几乎相同，但含义不同。从列开始：Cassandra的列(column)更像是HBase中的单元格(cell)。Cassandra中的列族(column family)更像是HBase表(table)。HBase中的列限定符(column qualifier)使人联想起Cassandra中的超级列，但后者至少包含2个子列，而前者只有一个。

此外，Cassandra允许主键包含多个列，而HBase与Cassandra不同，HBase仅具有1列的行键，从而将行键设计的负担加在开发人员身上。另外，Cassandra的主键由一个分区键和一个群集列组成，其中分区键也可以包含多个列。

尽管存在这些“冲突”，但这两个数据模型的含义几乎相同。他们没有联接(join)，这就是为什么将局部相关数据组合在一起的原因。它们在一个不占用存储空间的特定单元格/列中都没有价值。它们既需要在架构设计时指定列族，也不能在以后进行更改，同时随时允许列或列限定符具有灵活性。但是，最重要的是，它们都适合存储大数据。

## 2, 架构

---

Cassandra具有无主控架构，而HBase具有基于主控(master based)架构。这与Cassandra和HDFS之间的架构差异相同。

这意味着HBase具有单点故障，而Cassandra没有。HBase客户端无需与主服务器直接联系即可直接与从服务器通信，这使集群在主服务器停机后有一定的工作时间。但是，这几乎无法与始终可用的Cassandra集群竞争。因此，如果您无法承受任何停机时间，则可以选择Cassandra。

但是，为了确保可用性，Cassandra复制和复制数据，这会导致数据一致性问题。如果您的解决方案在很大程度上依赖于数据一致性，那么与强烈一致性的HBase不同，这使Cassandra成为错误的选择。因为后者仅将数据写入一个位置，并且始终知道在哪里可以找到数据（数据复制是在HDFS中“外部”完成的）。

此外，Cassandra的体系结构支持数据管理和存储，而HBase的体系结构仅用于数据管理。从本质上讲，HBase严重依赖于其他技术，例如用于存储的HDFS，用于服务器状态管理和元数据的Apache Zookeeper。同样，它需要额外的技术来运行查询。

## 3, 性能

---

### 写

Cassandra和HBase的服务器上写入路径非常相似。只是略有不同：数据结构的名称以及与Cassandra不同的是，HBase不会同时写入日志和缓存（这会使写入速度变慢）。

在较高的体系结构级别上，HBase具有更多的缺点：

1，在访问所需服务器之前，客户端必须“询问” Zookeeper，该服务器的hbase：meta表包含有关集群中所有表的位置的信息。然后，客户端要求元表保存服务器存储需要写入的实际表。并且只有在此之后，客户端才会将数据写入所需的位置。如果此类写入（和读取）非常频繁，则此信息当然会被缓存。但是，如果将表区域移至另一台服务器，则客户端需要再次执行完整回合。尽管Cassandra的数据分发和分区基于一致的哈希值比这聪明得多，而且速度更快。

2，HBase中的写入路径结束（缓存的数据被刷新到磁盘）后，HDFS也需要时间来物理存储数据。此外，对Cassandra写入性能的实际测量（在32节点群集中，每秒约326,500次操作，而HBase的297,000次操作）也证明，Cassandra的写入性能优于HBase。

## 读

如果需要大量快速而一致的读取（对数据和扫描的随机访问），则可以选择HBase。它仅在一台服务器上进行写入，因此无需比较不同节点的数据版本。HBase服务器在查找数据之前也没有太多数据结构需要检查。您可能会认为HBase的读取效率很低，因为数据实际上存储在HDFS中，因此HBase每次都需要将其取出。但是HBase具有一个块缓存，其中包含所有经常访问的HDFS数据，以及带有所有其他数据的近似“地址”的布隆过滤器，从而加快了数据检索的速度。本质上，HBase和HDFS的索引系统是多层的，它比Cassandra的索引效率更高（请参阅有关Cassandra性能的文章，以了解有关读取的更多信息）。

如果您读过Cassandra也很擅长read，那么您可能会对HBase更好的结论感到困惑。尤其是当您看到这种基准测试经验时，Cassandra每秒处理129,000次读取，而HBase仅为8,000次（在32节点群集中）。事实是，这些读取是有针对性的（基于已知的主键），而且它们也很不一致。因此，如果我们谈论扫描和一致性，那么Cassandra的庞大数量就会消失。

## 4，安全

像所有NoSQL数据库一样，HBase和Cassandra也存在安全性问题（主要问题是保护数据会破坏性能，从而使系统沉重且不灵活）。

但是可以肯定地说，这两个数据库都有一些确保数据安全的功能：两者中的身份验证和授权以及Cassandra中的节点间+客户端到节点加密。反过来，HBase提供了急需的方法来与其所依赖的其他技术进行安全通信。

更详细点：

Cassandra和HBase不仅提供数据库范围的访问控制，而且还提供一定级别的粒度。Cassandra支持行级访问，而HBase可以深入到单元级。Cassandra定义用户角色，并为这些角色设置条件，这些条件随后确定用户是否可以看到特定数据。HBase具有相反的“移动”功能。它的管理员将可见性标签分配给数据集，然后“告诉”用户和用户组他们可以看到标签。

## 5，应用领域

---

从Cassandra和HBase的数据模型组织方式来看，它们都非常适合时序数据：物联网系统中的传感器读数，网站访问和客户行为，证券交易所数据等。它们都很好地存储和读取了这些值。除此之外，可伸缩性是它们两者都具有的特性：Cassandra –线性，HBase –线性和模块化。

但是，当涉及扫描大量数据以查找少量结果时，由于没有数据重复，因此HBase更好。例如，此原因适用于HBase处理文本分析（基于网页，社交网络帖子，词典等）的功能。此外，由于使用Java的协处理器，HBase可以很好地与数据管理平台和基本数据分析（计数，求和等）配合使用。

Cassandra是一种高效的面向写入的数据库，非常适合大量数据的提取。借助它，您将构建可靠且可用的数据存储。此外，通过Cassandra，您可以在不同国家/地区创建数据中心并使它们同步运行。此外，如果将Cassandra与Spark结合使用，还可以获得良好的扫描性能。

但是在实际项目中应用Cassandra和HBase之间的主要区别在于。Cassandra非常适合“始终在线”的网络或移动应用程序以及具有复杂和/或实时分析的项目。但是，如果不急于获得分析结果（例如，进行数据湖实验或创建机器学习模型），则HBase可能是一个不错的选择。特别是如果您已经投资了Hadoop基础架构和技能。

### Cassandra vs. HBase –回顾

---

Cassandra是一种用于数据存储和管理的“自给自足”技术，而HBase则不是。后者旨在用作HDFS随机数据输入/输出的工具，这就是为什么将其所有数据存储在此处的原因。此外，HBase使用Zookeeper作为服务器状态管理器，并使用“大师”来知道所有元数据在哪里（以避免在包含元数据的主服务器出现故障时立即出现群集故障）。因此，HBase复杂的相互依存系统更加难以配置，保护和维护。

Cassandra擅长写入，而HBase擅长密集读取。Cassandra的弱点是数据一致性，而HBase的痛苦是数据可用性，尽管两者都试图减轻这些问题的不利后果。而且，两者都不会频繁删除和更新数据。

因此，Cassandra和HBase绝对不是双胞胎，而是两个有着相似发型的陌生人。要在两者之间进行选择，您应该彻底分析您的任务。然后，尝试找到一种方法来增强数据库的弱点而又不影响其性能。