# Google Bigtable论文读后感

## 数据模型

Bigtable 在存储数据时会按照 Cell 的 Row Key 对 Table 进行字典排序，并提供行级事务的支持（类似于 MongoDB，不支持跨行事务）。作为分布式的存储引擎，Bigtable 会把一个 Table 按 Row 切分成若干个相邻的 Tablet，并将 Tablet 分配到不同的 Tablet Server 上存储。如此一来，客户端查询较为接近的 Row Key 时 Cell 落在同一个 Tablet 上的概念也会更大，查询的效率也会更高。

除外，Bigtable 会按照由若干个 Column 组成的 Column Family（列族）对 Table 的访问权限控制。Column Key 由 family:qualifier 的形式组成，用户在使用前必须首先声明 Table 中有哪些 Column Family，声明后即可在该 Column Family 中创建任意 Column。由于同一个 Column Family 中存储的数据通常属于同一类型，Bigtable 还会对属于同一 Column Family 的数据进行合并压缩。由于 Bigtable 允许用户以 Column Family 为单位为其他用户设定数据访问权限，数据统计作业有时也会从一个 Column Family 中读出数据后，将统计结果写入到另一个 Column Family 中。

Table 中的不同 Cell 可以保存同一份数据的多个版本，以时间戳进行区分。时间戳本质上为 64 位整数，可由 Bigtable 自动设定为数据写入的当前时间（微秒），也可由应用自行设定，但应用需要自行确保 Cell 间不会出现冲突。对于拥有相同 Row Key 和 Column Key 的 Cell，Bigtable 会按照时间戳降序进行排序，如此一来最新的数据便会被首先读取。在此基础上，用户还可以设定让 Bigtable 只保存最近若干个版本的数据或是时间戳在指定时间范围内的数据。

## 系统原理

一个完整的 Bigtable 集群由两类节点组成：Master 和 Tablet Server。

Master 负责检测集群中的 Tablet Server 组成以及它们的加入和退出事件，会将 Tablet 分配至 Tablet Server，并负责均衡 Tablet Server 间的存储负载以及从 GFS 上回收无用的文件。除外，Master 还负责管理如 Table、Column Family 的创建和删除等 Schema 修改操作。

每个 Tablet Server 会负责管理若干个由 Master 指定的 Tablet，负责处理针对这些 Tablet 的读写请求，并负责在 Tablet 变得过大时对其进行切分。

Bigtable 集群会管理若干个 Table，每个 Table 由若干个 Tablet 组成，每个 Tablet 都会关联一个指定的 Row Key 范围，那么这个 Tablet 就包含了该 Table 在该范围内的所有数据。初始时，Table 会只有一个 Tablet，随着 Tablet 增大被 Tablet Server 自动切分，Table 就会包含越来越多的 Tablet。

## 集群成员变化与tablet分配

Bigtable Master 利用了 Chubby 来探测 Tablet Server 加入和离开集群的事件。每个 Tablet Server 在 Chubby 上都会有一个对应的唯一文件，Tablet Server 在启动时便会拿到该文件在 Chubby 上的互斥锁，Master 则通过监听这些文件的父目录来检测 Tablet Server 的加入。如果 Tablet Server 失去了互斥锁，那么 Master 就会认为 Tablet Server 已退出集群。尽管如此，只要该文件仍然存在，Tablet Server 就会不断地尝试再次获取它的互斥锁；如果该文件已被删除（见下文），那么 Tablet Server 就会自行关闭。

在了解了集群中有哪些 Tablet Server 后，Master 便需要将 Tablet 分配给 Tablet Server。同一时间，一个 Tablet 只能被分配给一个 Tablet Server。Master 会通过向 Tablet Server 发送 Tablet 载入请求来分配 Tablet。除非该载入请求在 Master 失效前仍未被 Tablet Server 接收到，那么就可以认为此次 Tablet 分配操作已成功：Tablet Server 只会接受来自当前 Master 的节点的请求。当 Tablet Server 决定不再负责某个 Tablet 时，它也会发送请求通知 Master。

Master 在检测到 Tablet Server 失效（互斥锁丢失）后，便会将其负责的 Tablet 重新分配。为此，Master 会尝试在 Chubby 上获取该 Tablet Server 对应的文件的互斥锁，并在成功获取后删除该文件，确保 Tablet Server 能够正确下线。之后，Master 便可顺利将 Tablet 分配至其他 Tablet Server。

如果 Master 与 Chubby 之间的通信连接断开，那么 Master 便会认为自己已经失效并自动关闭。Master 失效后，新 Master 恢复的过程如下：

1. 在 Chubby 上获取 Master 独有的锁，确保不会有另一个 Master 同时启动
2. 利用 Chubby 获取仍有效的 Tablet Server
3. 从各个 Tablet Server 处获取其所负责的 Tablet 列表，并向其表明自己作为新 Master 的身份，确保 Tablet Server 的后续通信能发往这个新 Master
4. Master 确保 Root Tablet 及 METADATA 表的 Tablet 已完成分配
5. Master 扫描 METADATA 表获取集群中的所有 Tablet，并对未分配的 Tablet 重新进行分配

## Locality Group

* Bigtable 允许客户端为 Column Family 指定一个 Locality Group，并以 Locality Group 为基础指定其实际的文件存储格式以及压缩方式。
* 首先，在进行上面我们提到的 Compaction 操作时，Bigtable 会为 Tablet 中的每个 Locality Group 生成独立的 SSTable 文件。由此，用户便可将那些很少同时访问的 Column Famliy 放入到不同的 Locality Group 中，以提高查询效率。除外 Bigtable 也提供了其他基于 Locality Group 的调优参数设置，如设置某个 Locality Group 为 in-memory 等。
* 在压缩方面，Bigtable 允许用户指定某个 Locality Group 是否要对数据进行压缩以及使用何种格式进行压缩。值得注意的是，Bigtable 对 SSTable 的压缩是基于 SSTable 文件的 Block 进行的，而不是对整个文件直接进行压缩。尽管这会让压缩的效率下降，但这也使得用户在读取数据时 Bigtable 只需要对 SSTable 的某些 Block 进行解压。