作业 3

171850532 司富元

1. **简述 HDFS 的基本特征。**

- 模仿 Google GFS 设计实现
- 可存储极大数目的信息，将数据保存到大量节点中；支持存储很大的单个文件
- 提供数据的高可靠性和容错能力。单个或者多个节点故障/不工作不会影响系统和数据可用性。通过一定的数据复制保证数据存储的可靠性和出错恢复能力
- 提供对数据的快速访问，并提供良好的可扩展性。简单加入更多服务器即可快速扩充系统容量，服务更多客户端
- 类似 GFS，HDFS 是 MapReduce 的底层数据存储支撑，并使数据尽可能根据其本地局部性进行访问与计算
- 优化了顺序读，支持大量数据的快速顺序读出，代价是对于随机访问负载较高
- 支持数据一次写入，多次读取；不支持已写入数据的更新操作
- 数据不进行本地缓存（文件太大，且顺序读没有局部性）
- 基于块的文件存储，默认块的大小是 64MB
    - 减少元数据的量
    - 有利于顺序读写（在磁盘上数据顺序存放）
- 多副本数据块形式存储，按照块的方式随机选择存储节点，默认副本数目是 3

2. **HDFS 的可靠性是如何设计的?**

- DataNode 节点检测

    - HDFS 采用心跳包(Heartbeats)机制,位于 HDFS 核心的节点 NameNode 不断周期性向各个管理的 DataNode 发送心跳包，而收到包的 DataNode 则需要回复，进而检测该 DataNode 是否有效
    - 如果某节点失效，则寻找新节点替代，并将失效节点数据重新分布
- 集群负载均衡
    - HDFS 支持数据的均衡负载。若某 DataNode 空闲空间低于特定临界点，则自动将数据搬迁至空闲 DataNode
    - 若对某个文件请求突然增加，也可能会创建文件新副本并将其分布到集群以满足应用要求
- 数据一致性
    - HDFS 采用校验和（CheckSum）机制。创建文件时为其生成一校验和，保存到和文件同一空间中。传输数据时会将数据和校验和一并传输，应用收到数据后进行校验。若两个校验和结果不同，则文件一定出错，数据块无效。判定数据块无效后，则需要从其他 DataNode 读取副本
- 主节点元数据失效（SecondaryNameNode 机制）
    - HDFS 使用 SecondaryNameNode 备份 NodeNode 元数据，以便在 NameNode 失效时从其上恢复 NameNode 元数据
    - SecondaryNameNode 使用文件镜像数据 Fslmage 和编辑日志数据 EditLog(相当于 HDFS 的 Checkpoint)。NameNode 启动时读取 Fslmage 内容，并将其与 EditLog 日志中信息合并生成新 Fslmage，EditLog 负责记录 HDFS 的所有修改，从而做到周期性保存主节点元数据
    - 在 NameNode 失效时，可通过 Fslmage 和 EditLog 数据恢复得到最近状

态的 NameNode，尽量减少损失

3. 简述 Google MapReduce 的主要功能和设计思想。

主要功能

- 任务调度
    - 主要负责为计算作业（job）划分成的计算任务（task）分配和调度计算节点（map/reduce 节点）、监控节点执行状态和 map 节点执行同步控制
    - 也负责一些计算性能优化处理，如对最慢计算任务采用多备份执行、选最快完成者作为结果
- 数据/代码互定位
    - 计算节点尽量处理本地磁盘上分布存储的数据，实现代码向数据的迁移
    - 若无法进行本地化数据处理（locality）时，再寻找其他可用节点并将数据从网络上传送给该节点，实现数据向代码迁移
- 出错处理
    - MapReduce 能检测并隔离出错节点，并调度分配新节点接管计算任务
- 分布式数据存储与文件管理
    - 能把海量数据分布存储在各个节点的本地磁盘上，同时保持整个数据逻辑上完整
    - 提供数据块的多备份存储管理能力，作为数据存储容错机制
- Combiner 和 Partitioner
    - 中间结果数据传入 reduce 节点前进行合并（combine）处理，合并具有相同主键数据避免重复传送

- Map 节点输出的中间结果使用一定策略进行划分（partition）处理，保证相关数据发送到同一个 reduce 节点

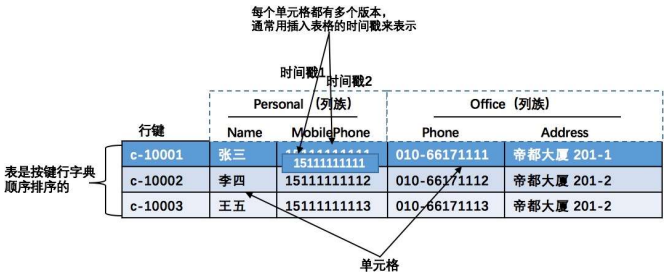
设计思想

- 向“外”横向扩展，而非向“上”纵向扩展（scale out 而非 scale up）
- 把失效看作常态
- 把处理向数据迁移
- 顺序处理数据（避免随机访问数据）
- 为开发者隐藏系统层细节
- 实现平滑无缝的可扩展性

4. 简述 Hadoop MapReduce 架构 v1.0 和 v2.0 的主要区别。

Hadoop MapReduce v2.0 和 v1.0 主要区别在于运行时环境。MR v2.0 运行在资源管理框架 YARN 上，运行环境不再由 JobTracker 和 TaskTracker 等组件构成。通过引入 YARN，v1.0 JobTracker 中的资源管理和作业控制被分开，分别交由 ResourceManager（全局群集资源管理器，负责资源分配）和 ApplicationMaster（负责作业控制）。这种架构上的调整，解决了 v1.0 单点故障（所有 Job 由 JobTracker 调度分配）和可扩展性差（JobTracker 任务繁重）的问题。

5. 对照 Google BigTable 的数据模型，简述 HBase 的基本数据模型。



上图是 HBase 数据模型的典型示例。

不同于传统的关系型数据库，HBase 有以下特点

- 结构松散
- 分布式
- 多维度
- 持久化
- 有序映射

HBase 索引依据是行键、列键和时间戳，其数据模型主要结构包括表（Table）、行（Row）、列（Column）、列族（Column family）、列限定符（qualifier）、单元格（Cell）、时间戳（Timestamp）。其中

- Table：组织存储于 HBase 的数据
- Row：每一行数据被唯一行键（Row key）标识。Row key 没有数据类型，在 HBase 中被看作为一个 byte 数组。所有行按 Row key 字典序排序
- Column, column family, qualifier：列族是访问控制的单位。数据通过列限定符在列中寻址。一个唯一的 Column family:qualifier 确定唯一列和唯一列值。同样，列在 HBase 中也被看作 byte 数组
- Cell：根据 Row key、Column family:qualifier 可以映射到一个对应的单元格。单元格是 HBase 存储数据的具体地址，数据在 Cell 中以 byte 数组形式存储，也没有具体类型
- Timestamp：给定值的版本标识号。每个值都会对应一个 Timestamp，在值一起一并写入 HBase

主要参考文章

Hadoop：HDFS 的健壮性设计，CSDN

https://blog.csdn.net/oraclestudyroad/article/details/52082915

HDFS 的可靠性，CSDN

https://blog.csdn.net/u012762573/article/details/46821645

Hadoop1.0 和 Hadoop2.0 的区别，OSChina

https://my.oschina.net/zgl2008/blog/895629