

# CSE Recitation-4 20161024

Name: ChenJiayang

ID: 5140379036

**Question1: What exactly would happen if one block of one hard drive got erased ...**

- GFS 把每个文件按 64MB 一个 Block 分隔，每个 Block 保存在多台机器上，环境中就存放了多份拷贝。MapReduce 的 master 在调度 Map 任务时会考虑 输入文件的位置信息，尽量将一个 Map 任务调度在包含相关输入数据拷贝的机器上执行；如果上述努力失败了，master 将尝试在保存有输入数据拷贝的机器附近的机器上执行 Map 任务。

**Question2: How do "stragglers" impact performance?**

- 在运算过程中，如果有一台机器花了很长的时间才完成最后几个 Map 或 Reduce 任务，导致 MapReduce 操作总的执行时间超过预期。出现“落伍者”的原因：
  - 1.如果一个机器的硬盘出了问题，在读取的时候要经常的进行读取纠错操作，导致读取数据的速度从 30M/s 降低到 1M/s。如果 cluster 的调度系统在这台机器上又调度了其他的任务，由于 CPU、内存、本地硬盘和网络带宽等竞争因素的存在，导致执行 MapReduce 代码的执行效率更加缓慢。
  - 2.最近遇到的一个问题是由于机器的初始化代码有 bug，导致关闭了的处理器的缓存：在这些机器上执行任务的性能和 正常情况相差上百倍。

**Question3: There is an article named "MapReduce: A Major Step Backwards" ...**

- 观点：
  1. 在大规模的数据密集应用的编程领域，它是一个巨大的倒退
  2. 它是一个非最优的实现，使用了蛮力而非索引
  3. 它一点也不新颖——代表了一种25年前已经开发得非常完善的技术
  4. 它缺乏当前DBMS基本都拥有的大多数特性
  5. 它和DBMS用户已经依赖的所有工具都不兼容
- 原因：
  - 1.1 MapReduce没有遵循一个特定的Schemas机制，没有任何控制数据集的预防垃圾数据机制
  - 1.2 MapReduce没有系统目录用来储存records的结构，导致程序员必须检查程序的代码来获得数据的结构
  - 1.3 MapReduce在low-level语言基础上做low-level的记录操作
  - 2.1 MapReduce没有索引，因此处理时只有蛮力一种选择
  - 2.2 MapReduce忽略了skew问题，导致当有同样key的记录分布变化很广时，导致一些reduce实例比其他实例要运行更长时间
  - 2.3 reduce实例同时访问同一个map节点来获取输入文件是不可避免的-导致大量的硬盘查找，有效的硬盘运转速度至少降低20%
  - 3.1 MapReduce使用的技术已有20年。实质上, 所有现代数据库系统已经提供这样的功能很久了，大约起源于1995年左右的Illustra引擎。
  - 4.1 MapReduce缺乏许多现代DBMS都已经缺省提供的特性，而仅仅提供了现代DBMS一小部分的功能
  - 5.1 MapReduce和DBMS工具不兼容，无法使用它们，也没有自己的工具可用，导致在完成一个终端应用时MapReduce会一直很难用

#### **Question4: Please raise at least one question of your own for the discussion.**

- 为什么MapReduce不适合做实时分析系统?
- Hadoop有种机制叫推测执行，即：如果某一个map or reduce跑的太忙，会起一个副本，谁先完成算谁的，它具体是为了解决怎样的问题而设计的?