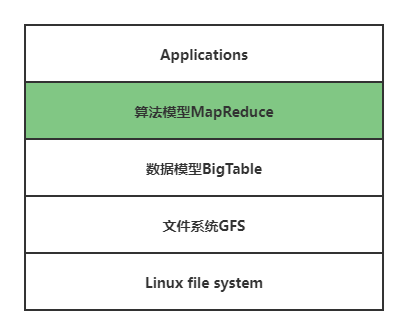
MapReduce读后感

1. **MapReduce介绍**

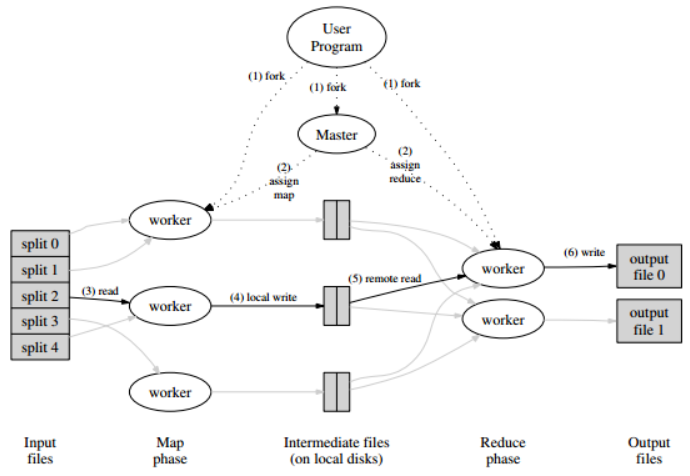
Google搜索引擎的支撑架构图如下，本文将思考并分析其中的MapReduce部分：

****

MapReduce是一个编程模型，用于处理和产生大数据。用户首先定义map函数来处理key-value键值对的对应，然后定义一个reduce函数来合并拥有相同中间key值的中间value。在现实中，许多分布式应用都能够使用这种模型进行表达，如文档抓取，Web请求日志等数据处理运算。

1. **设计原理**
   1. **MapReduce执行架构**

具体架构如下图所示：

****

* 1. **MapReduce的六大过程**

MapReduce有六个主要过程：Input，Split，Map，Shuffle，Reduce，Finalize。MapReduce首先将输入的原始数据(input)进行分片，然后在集群的多台机器上进行拷贝(split)。然后Map过程将会读取相应的输入块内容(map)，并从中解析出键值对然后传送给用户自定义的Map函数，再将函数返回值缓存入内存中(shuffle)。此时，处理进程将读取这些缓存的位置，再进入Reduce过程。通过获取的中间键值对进行排序，然后映射到一个task中。在排序后，处理进程每当遇到一个新的key，它的中间键值对将会被调用Reduce函数(reduce)，然后处理进程将函数的输出存储在文件中。当所有task完成后，执行结果将被存放在多个输出文件中，能够用于以此为输入的分布式应用(finalize)。

* 1. **如何统计出现次数**

MapReduce的库中提供了一个counter用于统计不同事件发生的次数。如用户能够统计处理过的单词的数量或索引文件的数量。这些counter能够由MapReduce自动进行维护。

* 1. **故障处理**

在MapReduce中，使用一个Master多个worker进程的方式对数据进行处理和管理。MapReduce使用与GFS相似的方式检测是否出现错误，即定时“握手”的方式。如果worker在特定时间没有相应，则会被标记为failed。此时，这个worker所有完成的task将被回退，然后分配给其他worker进行重新执行。这是因为每个worker的输入被保存在本地，若worker不再相应则输入也无法获取，因此需要全部重新执行。