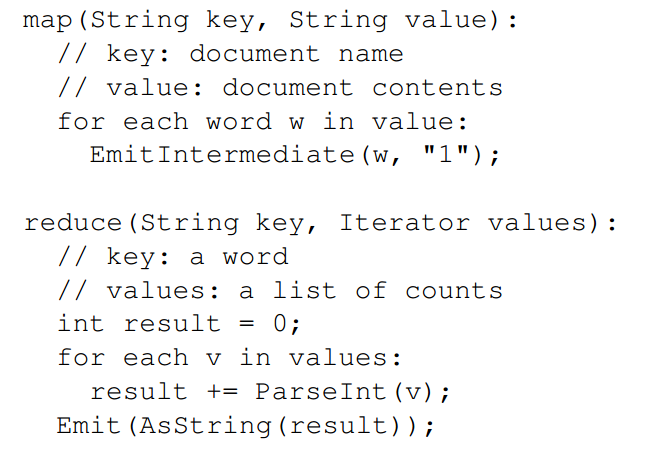
# Google Map Reduce

## 中文翻译

<https://blog.csdn.net/hguisu/article/details/7244981>

## 示例代码



## 类型

map(k1,v1) ---> list(k2,v2)

reduce(k2,list(v2)) ---> list(v2)

## 容错

### Worker故障

当worker故障时，由于已经完成的Map任务的输出存储在这台机器上，Map任务的输出已不可访问了，因此必须重新执行。

而已经完成的Reduce任务的输出存储在全局文件系统上，因此不需要再次执行。

### Master故障

如果master失效，就中止MapReduce运算。客户可以检查到这个状态，并且可以根据需要重新执行MapReduce操作。

## 存储位置（Locality）

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(例如，分配到一个和包含输入数据的机器在一个switch里的worker机器上执行)。

当在一个足够大的cluster集群上运行大型MapReduce操作的时候，大部分的输入数据都能从本地机器读取，因此消耗非常少的网络带宽。

## Backup Task

影响一个MapReduce的总执行时间最通常的因素是“落伍者”：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。

我们有一个通用的机制来减少“落伍者”出现的情况。当一个MapReduce操作接近完成的时候，master调度备用（backup）任务进程来执行剩下的、处于处理中状态（in-progress）的任务。

无论是最初的执行进程、还是备用（backup）任务进程完成了任务，我们都把这个任务标记成为已经完成。

## Combiner函数

我们允许用户指定一个可选的combiner函数，combiner函数首先在本地将这些记录进行一次合并，然后将合并的结果再通过网络发送出去。

Combiner函数在每台执行Map任务的机器上都会被执行一次。

一般情况下，Combiner和Reduce函数是一样的。

Combiner函数和Reduce函数之间唯一的区别是MapReduce库怎样控制函数的输出。Reduce函数的输出被保存在最终的输出文件里，而Combiner函数的输出被写到中间文件里，然后被发送给Reduce任务。

# Google Big Table

## 中文翻译

<https://blog.csdn.net/hguisu/article/details/7244991>