MapReduce

出自开放百科 - 灰狐

您可以在Wikipedia上了解到此条目的英文信息 **MapReduce** Thanks, Wikipedia.

什么是MapReduce? Google的分布运算开发工具!

The Google MapReduce framework is implemented in C++ and Java.

MapReduce是Google开发的C++编程工具，用于大规模数据集（大于1TB）的并行运算。概念"Map（映射）"和"Reduce（化简）"，和他们的主要思想，都是从函数式编程语言里借来的，还有从矢量编程语言里借来的特性。[1]

当前的软件实现是指定一个Map（映射）函数，用来把一组键值对映射成一组新的键值对，指定并发的Reduce（化简）函数，用来保证所有映射的键值对中的每一个共享相同的键组。 映射和化简 简单说来，一个映射函数就是对一些独立元素组成的概念上的列表（例如，一个测试成绩的列表）的每一个元素进行指定的操作（比如前面的例子里，有人发现所有学生的成绩都被高估了一分，他可以定义一个“减一”的映射函数，用来修正这个错误。）。事实上，每个元素都是被独立操作的，而原始列表没有被更改，因为这里创建了一个新的列表来保存新的答案。这就是说，Map操作是可以高度并行的，这对高性能要求的应用以及并行计算领域的需求非常有用。

而化简操作指的是对一个列表的元素进行适当的合并（继续看前面的例子，如果有人想知道班级的平均分该怎么做？他可以定义一个化简函数，通过让列表中的元素跟自己的相邻的元素相加的方式把列表减半，如此递归运算直到列表只剩下一个元素，然后用这个元素除以人数，就得到了平均分。）。虽然他不如映射函数那么并行，但是因为化简总是有一个简单的答案，大规模的运算相对独立，所以化简函数在高度并行环境下也很有用。 分布和可靠性 MapReduce 通过把对数据集的大规模操作分发给网络上的每个节点实现可靠性；每个节点会周期性的把完成的工作和状态的更新报告回来。如果一个节点保持沉默超过一个预设的时间间隔，主节点（类同Google File System中的主服务器）记录下这个节点状态为死亡，并把分配给这个节点的数据发到别的节点。每个操作使用命名文件的原子操作以确保不会发生并行线程间的冲突；当文件被改名的时候，系统可能会把他们复制到任务名以外的另一个名字上去。（避免副作用）。

化简操作工作方式很类似，但是由于化简操作在并行能力较差，主节点会尽量把化简操作调度在一个节点上，或者离需要操作的数据尽可能进的节点上了；这个特性可以满足Google的需求，因为他们有足够的带宽，他们的内部网络没有那么多的机器。 用途 在Google， MapReduce用在非常广泛的应用程序中，包括“分布grep，分布排序，web连接图反转，每台机器的词矢量，web访问日志分析，反向索引构建，文档聚类,机器学习，基于统计的机器翻译...”值得注意的是，MapReduce实现以后，它被用来重新生成Google的整个索引，并取代老的ad hoc程序去更新索引。

MapReduce会生成大量的临时文件，为了提高效率，它利用Google文件系统来管理和访问这些文件。

开发者只需要为数据集编写特定的Map/Reduce操作，有时甚至只需25-50行代码就够了，而MapReduce软件微架构会处理并行任务并且向分布在各处的计算机分发任务，同时处理机器错误和数据中的错误条件并进行优化操作，例如把计算过程推移到靠近数据的一方执行来减少I/O带来的带宽消耗，还提供了系统监控并且通过数以千计的计算机保持服务的可拓展性。

其他实现 Nutch项目开发了一个实验性的MapReduce的实现:Apache Hadoop。

参考

- Dean, Jeffrey & Ghemawat, Sanjay (2004)."MapReduce:大规模集群上的简单数据处理方式 (http://labs.google.com/papers/mapreduce.html) " 2005年4月6日。

^ "我们的灵感来自lisp和其他函数式编程语言中的古老的映射和化简操作." -"MapReduce:大规模集群上的简单数据处理方式"

(来源: http://www.tinydust.net/prog/diary/diary.htm)

Map Reduce - the Free Lunch is not over? (http://www.mengyan.org/blog/archives/2006/11/15/138.html)

目录

李开复: 算法的力量

Google最资深的计算机科学家Jeff Dean认识到，Google所需的绝大部分数据处理都可以归结为一个简单的并行算法：MapReduce。这个算法能够在很多种计算中达到相当高的效率，而且是可扩展的（也就是说，一千台机器就算不能达到一千倍的效果，至少也可以达到几百倍的效果）。MapReduce的另外一大特色是它可以利用大批廉价的机器组成功能强大的server farm。最后，它的容错性能异常出色，就算一个 server farm宕掉一半，整个fram依然能够运行。正是因为这个天才的认识，才有了Map and Reduce算法。借助该算法， Google几乎能无限地增加计算量，与日新月异的互联网应用一同成长。

http

Implementation

In general, these functions could be defined as:

```
List2 map(Functor1, List1);
Object
```

The

```
map
  foreach element in list {
```

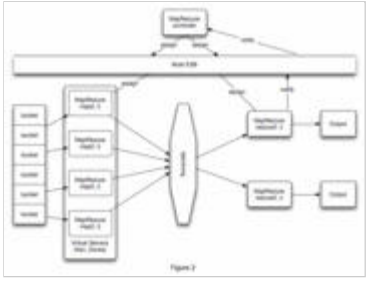
Implementation of

```
  v = function(element)
  intermediateResult.add(v)
 }
}
```

MapReduce

The

```
reduce(function, list, init) {
 result = init
 foreach value in list {
  result = function(result, value)
 }
 outputResult.add(result)
}
```



Java-based MapReduce Implementation

Java的方案中涉及 Xen , Terracotta , Mule 等技术

Why Should You Care About MapReduce? (http://www.theserverside.com/tt/knowledgecenter-tc/knowledgecenter-tc

Java

- Apache Hadoop
- IBM MapReduce Tools for Eclipse
- Ope

C/C++

- Phoenix is a shared-memory implementation of MapReduce implemented in C.
- Qt Concurrent

Ruby

Skynet Ruby MapReduce Framework

Erlang

- Apache CouchDB uses a MapReduce framework for defining views over distributed documents

Python

Writing An Hadoop MapReduce Program In Python (http://www.michael-noll.com/wiki/Writing_An_Hadoop_MapReduce_Program_In_Python)

Python 提供了map() 和 reduce() 函数, 是否能悟出些什么 :)

zip(list1, list2)”是把一对序列变成元素对序列的便捷方式

```
>>> seq = range(8)
```

```
>>> def square(x): return x*x
...
>>> zip(seq, map(square, seq))
[(0, 0), (1, 1), (2, 4), (3, 9), (4, 16), (5, 25), (6, 36), (7, 49)]
```

reduce(func, sequence)" 返回一个单值，它是这样构造的：首先以序列的前两个元素调用函数，再以返回值和第三个参数调用，依次执行下去。

```
>>> def add(x,y): return x+y
...
>>> reduce(add, range(1, 11))
55
```

0 comments [隐藏] 选项 **发表评论**

取自 “http://wiki.huihoo.com/wiki/MapReduce”
2个分类: Google | Cloud Computing

0

- 此页面最后修订于2010年9月16日 (星期四) 09:03。
- 此页面已被浏览过10,144次。
- 本站的全部文本内容在知识共享 署名-相同方式共享 3.0协议之条款下提供。

- 隐私政策
- 关于开放百科 - 灰狐
- 免责声明