**Don’t cry over spilled records: Memory elasticity of data-parallel applications and its application** **to cluste scheduling**

1. **实验目的**

由于条件有限，本实验只验证非理想内存条件下的性能损失情况。即给定不同的理想内存百分比，观察并记录任务的运行时间。

1. **实验原理**

弹性内存允许任务以比理想需要的内存少得多的内存运行，同时只付出适度的性能损失。

弹性内存中，资源与时间权衡成为任务排队时间与任务运行时间权衡。

shuffle内存不足可能导致任务内存分配大幅度减少，而任务运行时间只能适度增加。两个因素导致大量的内存减少。首先，shuffle内存通常是任务内存分配的很大一部分（Hadoop中默认为70％）。第二，外部归并排序算法可以运行非常少的内存，因为它们可以通过使用辅助存储进行补偿。几个因素也解释了为什么shuffle内存不足的时候任务运行时间只是适度增加。首先，数据并行任务将shuffle操作与CPU密集处理相结合，从而与外部存储和内存的归并排序之间性能差距的相关性更小。第二，磁盘访问是有效的，因为磁盘是被顺序访问。第三，尽管shuffle内存显着减少（k路合并在k中是取对数的），外部归并排序算法的性能保持稳定。

由于等待时间缩短，任务可能会在更少的内存调度时完成。 我们显示调度程序可以将此任务级权衡转换为改进的作业完成时间和集群范围的内存利用率。由于任务的完成时间是等待时间加上运行时间的总和，由于弹性，等待时间的显着减少可能会抵消掉运行时间的增加，并且总体上导致任务完成时间的更快。

**三、实验环境**

1. **论文实验环境**

实验在两个不同的硬件平台（Dual Intel Xeon E5-2630v3 + 40Gb NIC，Dual AMD Opteron 6212 + 10GB NIC），两个操作系统（RHEL 7，Ubuntu 14.04），三种不同的磁盘配置（HDD，SDD，RAID 0中的2\*HDD），三个IO调度程序（CFS，截止日期，noop）和三个JVM（HotSpot 1.7,1.8，OpenJDK 1.7）。

1. **实验环境搭建**

本论文中对实验环境配置要求较高，由于条件限制，我们小组的电脑配置不高。

硬件环境如下：

CPU：Core i5-3230M

内存：DDR3 6GB

硬盘：西部数据500G 黑盘7200转

软件环境配置如下：

Ubuntu16.04下搭建Hadoop环境，通过使用理想内存、理想内存10%、50%，90%的内存来运行PageRank程序，记录在Hadoop弹性负载情况下Reducer的执行情况。

搭建hadoop的步骤如下：

1. 创建Hadoop用户，并赋予相应的权限
2. 切换到Hadoop用户，并修改/etc/hostname 主机名为 master或slave
3. 修改 /etc/hosts
4. 安装JDK,配置JDK的环境变量，手动设置系统默认JDK
5. 配置SSH免密码登录
6. 下载hadoop-2.71.tar.gz并解压安装
7. 配置HADOOP环境变量

* 分别修改conf/Hadoop-env.sh，conf/core-site.xml，conf/hdfs-site.xml，etc/hadoop/yarn-site.xml四个文件。
* 配置conf/masters文件，设置内容为master
* 配置conf/slaves文件，设置内容为slave1，slave2

8.初始格式化文件系统hadoop namenode -format

1. **实验过程**

运行1GB的PageRank数据，设定理想内存为1.2GB

1. 给理想内存的10%运行PageRank，并记录程序运行时间
2. 给理想内存的50%运行PageRank，并记录程序运行时间
3. 给理想内存的90%运行PageRank，并记录程序运行时间
4. 在理想内存条件（100%）下运行PageRank，并记录程序 运行时间。
5. **实验结果**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 内存大小  时间 | 10% | 50% | 90% | 100% |
| 运行时间 | 27.8 | 28.5 | 29.3 | 23.6s |
| 相对值 | 1.18 | 1.21 | 1.24 | 1 |

结果分析：

分别用10%，50%，90%的理想内存条件下运行PageRank程序，实验数据表明，在非理想内存条件下有适当的性能损失，10%与90%有相似的性能损失，但差别不是很大。

三个因素限制了惩罚。 首先，映射器通过地图函数调用完成了shuffling CPU密集型工作。 第二，磁盘访问是有效的，因为顺序访问磁盘。 第三，外部融合排序算法的性能保持稳定，尽管洗牌记忆显着减少。 记忆弹性形状的形状映射器的弹性分布类似于阶梯函数。 原因是尺寸不足的映射器执行一个额外的合并阶段，对于许多大小不足的分配，它需要相似的时间。

1. **结论**

由于硬件限制，无法重现论文中实验的全部结果，但是实验结果可以表明弹性内存的优势，在非理想内存条件下，以少得多的内存运行任务，只付出适度的性能损失，并且性能损失差别不大。