

应用于云中 MapReduce 计算的动态资源调度算法

郭欢欢, 张水平, 王 蓉, 陈爱网, 张月玲, 张凤琴

(空军工程大学 信息与导航学院, 陕西 西安 710077)

摘 要: 越来越多的 MapReduce 计算由运行在云中虚拟机组成的虚拟集群完成, 为增强虚拟 MapReduce 集群计算中任务的数据本地性, 充分利用云计算的资源可动态配置的优势, 提出了一种基于队列的动态资源调度算法, 并模拟了云计算平台进行测试, 实验结果表明动态资源调度算法提高了虚拟 MapReduce 集群的计算性能。

关键词: 云计算; MapReduce; 虚拟集群; 虚拟机资源重配置

中图分类号: TP301.1

文献标识码: A

文章编号: 1000-7180(2014)03-0044-04

The Dynamic Resource Scheduler for MapReduce in the Cloud

GUO Huan-huan, ZHANG Shui-ping, WANG Rong,

CHEN Ai-wang, ZHANG Yue-ling, ZHANG Feng-qin

(Institute of Information and Navigation, Air Force Engineering University, Xi'an 710077, China)

Abstract: It has been increasingly adopted that MapReduce jobs runs on a set of virtual machines (VMs) in the cloud. To enhance the data locality of the tasks during the job, leverages the advantage of the cloud's dynamic resource reconfiguration, this paper proposed a Queue-based dynamic resource schedule algorithm. Cloud platform had been simulated for the evaluation and the result shows that the scheduler efficiently improved the virtual clusters' performance.

Key words: cloud computing; MapReduce; virtual clusters; VM resource reconfiguration

1 引言

随着电子商务、移动及社交网络的兴起, 互联网每天产生巨大规模的数据, 为了从数据中发掘出有价值的信息并加以利用, 急需有效的手段对这类数据进行分析。

MapReduce^[1]由 Google 公司提出的并行计算模型。结合分布式文件系统, 可以方便地通过增加计算节点和存储节点来扩展系统的计算和存储性能。用户只需要编写的 Map 函数和 Reduce 函数, 就可以自动地在可伸缩的大规模集群上并行处理和分析大规模的数据。

完全由物理机组成的 MapReduce 集群其计算和存储绑定在一起, 节点故障恢复代价大, 并且每个节点的配置是静态的, 无法满足整个作业周期内需

求的动态变化。而使用由虚拟机组成的集群则提高了整个集群计算的灵活性、可靠性^[2], 并且用户可以按需租用云服务提供商提供的云服务来进行集群计算, 降低成本、专注核心业务。但云计算的物理资源虚拟化与 MapReduce 的数据本地性存在冲突, 且计算过程中节点对资源的需求不断变化, 需要动态调整分配给节点的资源。

2 云平台和 MapReduce

2.1 云平台架构

云计算^[3]是一种通过网络统一组织和灵活调用各种信息资源包括计算与存储、应用运行平台、软件等, 实现大规模计算的信息处理方式。按照云计算服务提供的资源层次, 可以分为 IaaS(基础设施即服务)、PaaS(平台即服务)、SaaS(软件即服务)等。

收稿日期: 2013-05-09; **修回日期:** 2013-07-02

基金项目: 陕西省自然科学基金项目(2011JM8035)

提供基础服务的云平台的体系架构(如图1所示)分为三个层次,即云计算硬件资源池、云计算管理平台、虚拟化管理层。云计算硬件资源池由服务器节点、存储设备组成。云计算管理平台负责将资源池中的硬件资源虚拟化,向上层的应用服务提供运算、存储资源。虚拟服务器节点层通过虚拟化管理底层驱动,根据用户需求创建虚拟机,通过网络将云中的虚拟机提供给终端用户使用。

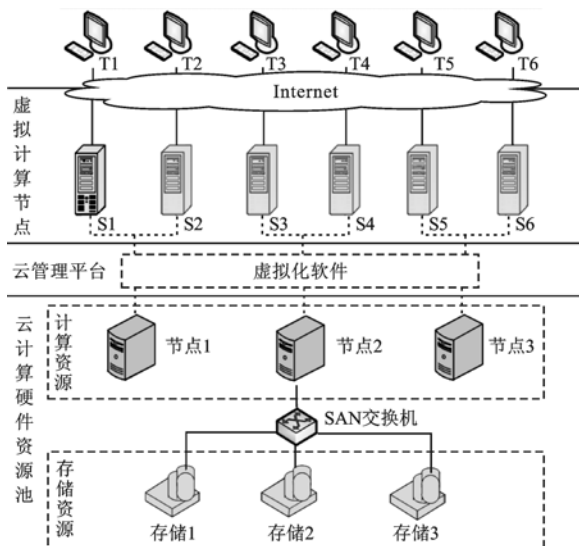


图1 云平台架构

2.2 MapReduce 框架

MapReduce 并行编程模型把计算过程分解为两个主要阶段,即 Map 阶段和 Reduce 阶段。Map 函数处理 Key/Value 对,产生一系列的中间 Key/Value 对,Reduce 函数用来合并所有具有相同 Key 值的中间键值对,计算最终结果。

Hadoop^[4]是对 MapReduce 框架的开源实现项目,核心由 MapReduce 计算模块和 HDFS^[5](分布式文件系统)组成。在一个 Hadoop 集群中,同时运行 NameNode 进程和 JobTracker 进程的节点被称为 Master 节点,而同时运行 DataNode 进程和 TaskTracker 进程的节点被称为 Slave 节点。Hadoop 核心架构如图2所示,可分为 MapReduce 计算模块和 HDFS 存储模块两部分。

(1) MapReduce 计算模块如图2所示上半部分。集群中只有 Master 节点拥有 JobTracker 进程,JobTracker 进程负责控制 MapReduce 任务的分配和 TaskTracker 的使用。其余 Slave 节点各有一个 TaskTracker 进程,TaskTracker 进程负责运行 Map 任务和 Reduce 任务。

(2) HDFS 存储模块如图2所示下半部分。

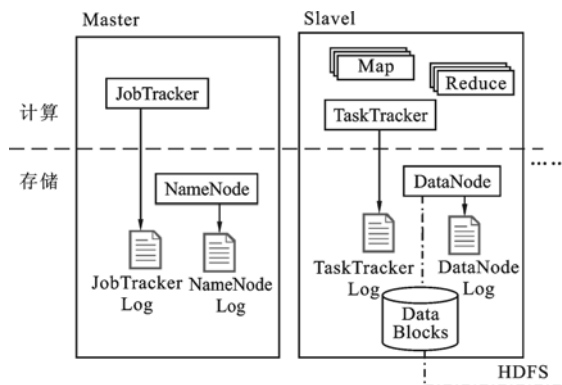


图2 Hadoop 架构

Master 节点是 HDFS 的控制节点,拥有 NameNode 进程,存放数据块索引的元数据。其余每个 Slave 节点各有一个 DataNode 进程,对实际的数据块进行读写。数据采用大块存储(64M 及以上),方便对数据进行压缩和传输。每份数据在 HDFS 中都至少会有三个备份,以支持高度容错。

3 云平台和 MapReduce 结合的问题

云计算利用分布式计算和虚拟资源管理技术,通过网络将分散的资源集中起来形成共享的资源池,并以动态按需和可度量的方式向用户提供服务。云计算的资源虚拟化与 MapReduce 计算中数据本地性的要求存在冲突,使用云平台提供的虚拟机作为计算和存储节点需要解决如下问题:

(1)增强数据本地性。MapReduce 集群计算中要输入的数据和任务执行在同一个物理节点上,而云计算对物理资源进行了统一的管理,因此由虚拟机组成的集群中各节点的计算和存储并不一定在同一个物理节点上,其调度算法需要考虑增强作业执行的数据本地性。

(2)动态调整虚拟 MapReduce 集群中节点的资源。完全由物理机组成的 MapReduce 集群中各个节点的资源配置是静态的,并且在执行作业时只能在初始化时设定各项参数,运行时无法再调整。云平台中的虚拟机节点可以动态调整资源配置,这赋予了虚拟 MapReduce 集群更大的灵活性。因此需要改进 MapReduce 的调度算法以利用云平台可动态调整虚拟机资源的优势。

4 适合云中 MapReduce 动态资源调度算法

数据的本地化要求计算节点要有灵活的 CPU 资源。为了达到负载均衡,任务必须在节点间平均分配,然而这样一个纯平衡的任务调度会忽视数据的

本地性,引起数据远程读写的数据传输代价^[6].为了维护数据的本地性,有输入数据的虚拟机需要临时地增加其计算能力来处理这样的任务.

早期的 Hadoop 采用 FIFO(先进先出)调度算法,作业必须等待直到轮到自己运行,为了使每个用户公平地共享集群能力,Hadoop 提供了使用公平调度算法的调度器.此后为改进性能提出了虚拟集群同步调度算法.

为了更好地支持数据本地性,本文提出了基于队列的动态资源调度算法(Q-DRS).当一个虚拟机有空闲的任务槽时,将其空闲的 CPU 核注册到释放队列中.如果一个虚拟机有附加的本地任务时,将该任务添加到申请分配资源的队列.只有当释放资源和申请资源的队列都不为空的情况下,才进行资源重配置.基于 Q-DRS 算法的 MapReduce 作业调度模型如图 3 所示.其中 Q_a 表示资源申请队列, Q_r 表示资源释放队列.

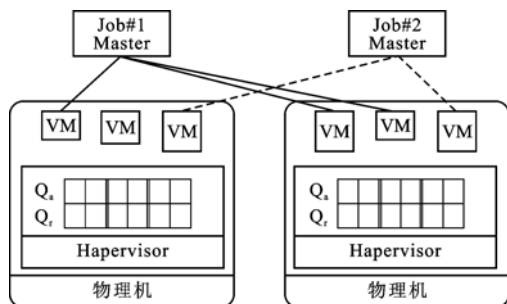


图 3 基于 Q-DRS 算法的 MapReduce 作业调度

Q-DRS 算法的描述如表 1 所示.在作业分配的任务中, map 任务对数据本地性要求高,而 reduce 任务并不要求,因此在对任务进行分配时应充分考虑任务的类型.如果是 reduce 任务,则可以直接在有空闲任务槽的节点上分配任务,而不用考虑其数据的本地性问题;对于 map 任务,如果空闲节点满足数据本地性,则在该节点上启动任务,如果不满足则首先检查空闲待释放资源队列,尽可能将可用的 CPU 资源分配给任务.如果不可用,则检查申请资源队列,找出申请资源记录最少的节点,将任务分配给该节点.同时,在物理机的虚拟机管理中对申请和释放资源的队列进行处理,完成对资源的调度.

Q-DRS 算法不同于同步资源调度算法,其区别是同步调度算法只有在虚拟机所在的物理机有空闲的 CPU 核资源时才分配给该虚拟机节点,而 Q-DRS 算法则不考虑虚拟机所在的物理机有无空闲 CPU 资源,将分配和释放资源的请求加入队列,等待该物理机有空闲 CPU 资源后再进行实际分配,

从而最大程度地达到数据本地性的目的.同时 Q-DRS 算法又对任务的类型进行了分类, reduce 任务直接在空闲节点上启动,减少了任务分配的等待时间. Q-DRS 避免了同步调度算法中必须要有可用资源的限制,减少了对物理机预留 CPU 资源的需求,更好地支持了数据本地性,更适用于云平台上虚拟集群的计算环境.

表 1 Q-DRS 算法

1:	从节点 n 收到“心跳”;
2:	if n 有空闲的任务槽 then
3:	将 jobs 按照所含任务数的多少从小到大排列
4:	for j in jobs do
5:	if j 有未启动的任务 t and (其任务类型为 reduce or 其输入数据在 n 上) then
6:	在节点 n 上启动任务 t
7:	else if j 有未启动的任务 t then
8:	在 n 所在物理机的 Q_r 中找出含有任务 t 执行所需数据的节点的集合 s_d
9:	将 s_d 按照释放记录的数量降序排列
10:	if s_d not $\emptyset$ then
11:	挑选集合 s_d 中第一个节点 m
12:	else
13:	找出含有任务 t 所需数据的节点集合 s_a
14:	将集合 s_a 按照申请分配队列中记录的数目升序排列
15:	选择集合 s_a 中第一个节点 m
16:	end if
17:	增加申请 m 记录到 m 所在物理机 Q_a
18:	增加释放 n 记录到 n 所在物理机 Q_r
19:	在 m 上启动任务 t
20:	end if
21:	end for
22:	end if

5 评估测试

本实验采用 8 台 Dell 服务器,每台服务器含有 8 个 CPU(Intel®Xeon™ CPU 2.40 GHz),8 GB 主存,模拟了基于 Xen 4.0 的云计算平台. Xen 的 credit 调度算法支持虚拟机中虚拟 CPU 的热插拔.

共创建了 24 个虚拟机,每个虚拟机分配有 2 个虚拟 CPU、2 GB 内存,在每个虚拟机上安装了 Ubuntu 12.10 操作系统及 Hadoop 1.04 软件,组成了虚拟 MapReduce 集群.集群中的每个节点设置 map 任务槽和 reduce 任务槽各 2 个.

在由虚拟机组成的 MapReduce 集群上执行了

三种类型的作业:Grep、WordCount、Sort,分别测试了在不同大小的输入数据下的作业完成时间,测试的结果如图4、图5和图6所示。

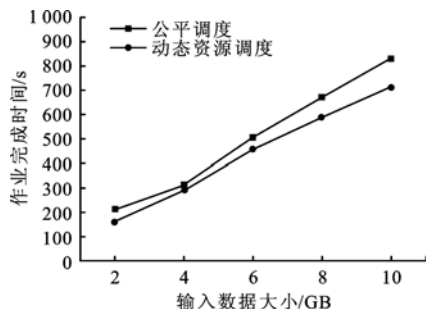


图4 执行 Grep 作业结果

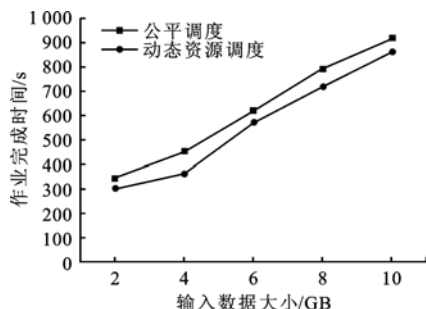


图5 执行 WordCount 作业结果

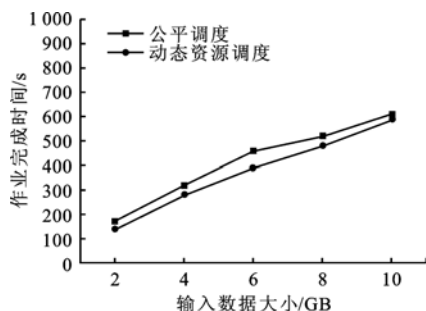


图6 执行 Sort 作业结果

通过作业完成结果可以看出使用基于队列的动态资源调度算法可以提高作业执行的数据本地性,平均加速虚拟 MapReduce 集群计算性能 14%,充分利用了云平台可动态调整虚拟机资源的优势。

6 结束语

针对云平台和 MapReduce 结合中存在的问题,通过将云平台的动态调整虚拟机配置与虚拟 Ma-

pReduce 集群的调度相结合,充分利用了云平台对虚拟机节点的自动化管理能力,增强了作业执行的数据本地性,提高了 MapReduce 集群计算的性能。今后将对云中 MapReduce 集群计算过程中的故障原因进行详细分类并完善其对应的处理措施,进一步优化和提高云平台上的 MapReduce 集群计算的性能。

参考文献:

- [1] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 5(1):107-113.
- [2] 李明, 胥光辉, 耿瑶. MapReduce 编程模型在网络 I/O 密集型程序中的应用研究[J]. 计算机应用研究, 2011, 28(9):3372-3374
- [3] 云计算白皮书[R]. 北京: 工业和信息化部电信研究院, 2012.
- [4] White T. Hadoop: the definitive guide[M]. USA: O'Reilly Media, 2012.
- [5] Borthakur D. HDFS architecture guide[EB/OL]. [2013-04-10]. <http://Hadoop.apache.org/common/docs/current/hdfsdesign.html>.
- [6] Min Li, Dinesh S, Ali R Butt, et al. CAM: a topology aware minimum cost flow based resource manager for map reduce applications in the cloud [C]//HPDC 2012. The Netherlands, ACM, 2012.

作者简介:

郭欢欢 男, (1985—), 硕士研究生. 研究方向为云计算、大数据处理。

张水平 女, (1956—), 教授, 硕士生导师. 研究方向为分布式数据库、信息安全、云计算。

王 蓉 女, (1978—), 讲师. 研究方向为分布式数据库、信息安全。

陈爱网 男, (1975—), 副教授. 研究方向为分布式数据库, 物联网。

张月玲 女, (1963—), 副教授. 研究方向为智能信息处理、物联网。

张凤琴 女, (1964—), 副教授, 硕士生导师. 研究方向为分布式数据库、信息安全、数据挖掘。