

DOI: 10.7652/xjtuxb201508012

采用资源划分的云环境下 Hadoop 资源许可调度方法

周墨颂¹, 朱正东¹, 董小社¹, 陈衡¹, 王寅峰²

(1. 西安交通大学电子与信息工程学院, 710049, 西安; 2. 深圳信息职业技术学院软件学院, 518172, 广东深圳)

摘要: 针对云计算环境中 Hadoop 平台由于节点计算能力差异、多样混合负载共存等原因而出现的性能不佳的问题,提出一种采用资源划分的资源许可方法。该方法在云计算环境下通过减少资源浪费或负载过重等情况的出现来提高系统性能。该方法采集资源信息并推测任务资源需求,根据可用计算资源和任务需求动态划分、调度资源;使用与资源无耦合的资源许可可启动任务并控制任务数量调节资源利用率以适应云环境。使用该方法对比公平调度器在国家高性能计算中心(西安)进行测试发现:单作业在资源竞争环境中优于公平调度器的静态结果;混合负载在 3 种测试环境中完成时间分别平均减少了 27.5%、37.1%和 50.98%,性能显著提升。实验结果表明,该方法可以适应负载资源需求和可用计算资源的变化,灵活划分计算资源,解决 Hadoop 在云环境中的性能不佳问题。

关键词: 云计算;Hadoop;资源划分;作业调度;混合负载

中图分类号: TP391 **文献标志码:** A **文章编号:** 0253-987X(2015)08-0069-06

A Resource License Scheduling Method for Hadoop in Cloud Computing Using Resource Allocation

ZHOU Mosong¹, ZHU Zhengdong¹, DONG Xiaoshe¹, CHEN Heng¹, WANG Yinfeng²

(1. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China;

2. School of Software, Shenzhen Institute of Information Technology, Shenzhen, Guangdong 518172, China)

Abstract: A resource license scheduling method using resource allocation is proposed to improve the poor performance caused by different computing capacities of nodes and mixed workload in cloud computing. The method improves the performance through reducing the resources wasting or overload. It collects resource information and estimates resource requirements of workloads and allocates computing resources dynamically according to available resources and resource requirements of workloads. Licenses uncoupled with resources are used to launch tasks and to adjust the number of parallel tasks to adapt the cloud environment by controlling the number of licenses. The method is evaluated in the national high performance computing center (Xi'an). Results show that the completion times of single job workloads of the proposed method are better than those of the FAIR scheduler in competitive environments. Moreover, the completion times of mixed workloads of the method in three environments reduce 27.5%, 37.1% and 50.98%

收稿日期: 2015-03-03。 作者简介: 周墨颂(1988—),男,博士生;朱正东(通信作者),男,高级工程师,硕士生导师。 基金项目: 国家自然科学基金资助项目(61173039);国家自然科学基金青年基金资助项目(61202041);国家高技术研究发展计划资助项目(2012AA01A306);深圳市科技计划资助项目(JCYJ20120615101127404)。

网络出版时间: 2015-05-21

网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20150521.0901.002.html>

respectively on average, that is, the performance of the method has a significant improvement. It can be concluded from the results that the proposed method adapts the complex environment and solve the performance problem in cloud computing.

Keywords: cloud computing; Hadoop; resource allocation; job scheduling; mixed workload

Google 公布的大型集群使用数据表明,其集群上运行着来自不同部门的负载,其中包括多种类型负载^[1]。集群中节点计算能力的差异性非常显著,这大大降低了基于 slot 和 core 的传统资源划分和调度方式的有效性,导致性能下降^[1]。Google 使用共享集群满足不同部门计算需求的情况是云计算的一个缩影,云计算的重要意义之一是使用共享的基础设施满足不同租用者的需求,以达到提高数据、计算资源的使用效率,方便集中管理,减少维护成本等目的。因此,Google 集群中的问题在云计算环境中普遍存在。云环境负载同样具有多样性,且由于多种原因节点计算能力也呈现出差异化:首先,集群硬件的扩充和更新换代造成了节点间计算资源异构;另外,各节点上负载的差异导致节点剩余计算能力有差异。云计算中系统性能深受节点计算能力差异化、负载多样化的影响,因而有效地划分和调度计算资源成为了云计算研究中的关键问题。

MapReduce^[2]是一种高效的并行处理模型,用于在商用集群上提供 PB 级数据处理能力。Hadoop 作为 MapReduce 的开源实现,是目前大数据处理中研究、应用最广泛的平台^[3]。由 Facebook 公司开发的公平调度算法^[4]实现了多个用户对集群硬件资源的公平共享,该算法被广泛应用于各数据处理平台并取得了良好的效果。Hadoop 以槽划分节点上的计算资源,调度器通常假设计算节点能力相同。在云计算环境中,Hadoop 依然忽略节点差异化的计算能力和多样负载并存带来的可用资源持续变化,仍然使用固定数目的 slot 划分资源,导致计算节点频繁出现资源浪费或负载过重,进而严重影响负载完成时间。

近年来,文献[1,5-9]分别指出了节点计算能力差异、多样混合负载共存等因素带来的性能问题。文献[7-12]从调度算法、资源划分、负载迁移中的一或两方面着手优化,但 Hadoop 在云环境中的问题由多方面因素共同引起,仅从少数几个方面着手解决的效果并不理想。本文结合多种优化思路,提炼出涉及资源管理、划分、调度和负载需求特点区分等多方面的资源许可方法,感知计算节点资源情况并推测负载资源需求,使用资源许可控制并行负载数

量以适应资源变化,解决其中节点计算能力差异和多样负载并存导致的负载完成时间增加、性能下降的问题。

1 相关工作

Yang 等提出了一种研究负载特性、Hadoop 配置、系统性能之间关系的统计分析方法,分析指出负载多样的资源需求对性能产生严重影响^[5]。Huang 等将负载按特点分组,指出负载的资源需求存在很大差异,为此设计了更接近现实的基准测试套件 HiBench^[6]。针对多种负载资源需求差异引起的问题,研究者们提出了不同的解决方案。Tian 提出根据历史信息预测负载资源需求以识别 CPU 密集型和 IO 密集型任务,并设计了一个三重队列调度器^[7]。Lu 等提出面向负载特性调度器,通过采样分析和静态分析预测负载的资源特性,并以此为依据进行调度^[8]。

Zaharia 指出,节点间计算能力差异化不满足 Hadoop 调度器所做的所有节点能力相同的假设,在性能差节点上执行的任务可能拖慢整个作业。LATE 方法通过在其他节点上重复启动落后任务,缓解节点间计算能力差异化对作业完成时间的影响。但是,任务重复执行消耗资源,集群负载过重时可能造成严重的性能下降^[9]。SkewTune 将落后任务划分为小任务,并将未完成的提交为新任务迁移到其他节点以缓解节点的性能差异^[10]。Rasooli 等指出了负载差异、集群差异和用户差异带来的影响,并提出了在多个调度算法间切换的调度器^[11]。Tang 等通过在任务间借用槽来解决类型限制引起的资源浪费问题^[12]。

以往研究中解决问题的思路可分为以下几种:优化调度算法、区分负载资源需求、改进资源划分策略、负载冗余执行、负载迁移。研究者一般从单一方面入手进行优化,但由于 Hadoop 在云环境中出现的性能问题由多种因素引起,所以优化效果不佳。优化调度算法^[11]受限于资源划分方式,改进资源划分策略^[12]仍基于固定数目的 slot,因此不适合在可用资源持续变化的环境中使用;区分负载资源需求^[7-8]没有考虑节点差异化的计算能力,不适用于云

计算环境;负载冗余执行^[9]和负载迁移^[10]虽然可以在性能出现问题后进行补救,但是并不能彻底解决问题。

与之前研究不同,本文从多个方面入手解决问题,加入资源机制感知节点资源情况并根据资源情况控制任务申请;在运行时推测负载的资源需求并以此为依据对计算资源进行细粒度划分调度;提出资源许可,通过控制并行负载数量屏蔽节点能力差异化并调节资源利用率。使用本文提出的资源许可方法可以在节点计算能力差异化、负载多样化的云环境中显著提高性能。

2 资源许可方法

2.1 资源许可架构

针对 Hadoop 在云计算环境中的问题,本文从感知资源变化、资源划分、区分负载特性、调度等方面入手提出资源许可方法,其结构如图 1 所示。

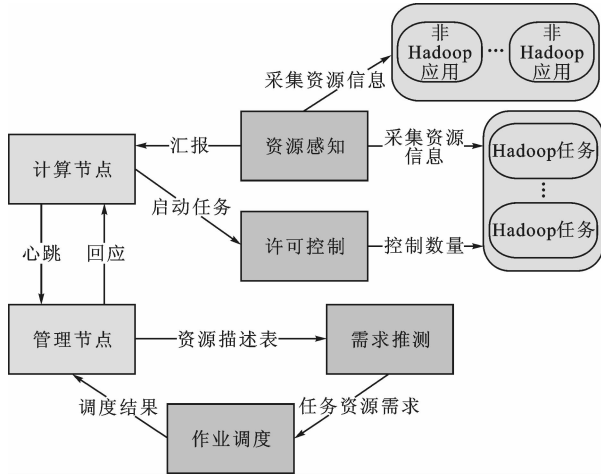


图 1 资源许可方法整体架构和流程

资源许可方法分为 4 个模块,分别为资源感知、许可控制、需求推测、作业调度,其中资源感知和许可控制模块运行在计算节点上,而需求推测和作业调度模块运行在管理节点上。资源许可方法运行流程为:首先,资源感知模块采集资源信息并有计算节点组织成资源描述表加入心跳中发送给管理节点;然后,需求推测模块使用资源信息推测、校正任务的资源需求并提供给作业调度器使用;接着,作业调度器根据资源和需求情况作出调度决策,并由管理节点将被调度任务及其资源需求信息回应给计算节点;最后,许可控制模块根据当时可用资源和任务资源需求动态控制许可数量,在充分利用计算资源的同时避免负载过重。

2.2 资源感知与许可控制

资源感知模块通过周期性分析 Linux 系统 Proc 文件夹下数据来采集节点上负载的资源使用信息以及可用资源信息。

$$R_{idle} > \begin{cases} (\min(T_c, C_a)/m) \log_m n, & \text{CPU 资源} \\ (\min(T_m M_t, M_a)) \log_m n, & \text{内存资源} \end{cases} \quad (1)$$

资源感知模块负责控制新任务的申请,不等式(1)为申请条件,式中 R_{idle} 表示当前可用资源量(具体包括 CPU 和内存两种资源)。不等式右侧是资源阈值和资源预留因子 $\log_m n$ 的乘积,其中资源阈值取任务资源平均需求(C_a 或 M_a)与固定阈值(T_c 或 T_m)中最小值,因为 CPU 平均使用量及阈值针对单个核心计算所以需要除以逻辑核数 m ,而由于内存平均使用量不是百分比所以需要乘以内存总量 M_t , n 为节点上资源使用者数目。资源预留因子被设计用来动态预留资源,这样当资源使用者多于逻辑核数时,因子升高以预留更多资源;反之,因子降低使申请条件更容易满足。当节点资源信息使式(1)中两个不等式同时成立,即可用资源满足申请条件时,进行任务申请。

许可控制模块是整个方法的关键,它控制并行任务数量来适应环境中可用计算资源的变化。资源许可是任务被允许启动的依据,分为静态许可和动态许可两种,任务使用任意一种许可均能获得弹性的计算资源。静态许可数目由配置文件指定,它可以使 Hadoop 在资源竞争中保有一定资源。动态许可由控制模块动态生成,生成动态许可使用的资源会在可用资源中减去,动态许可的数量随可用资源变化来避免资源浪费和负载过重。

资源许可控制模块优先使用静态许可启动任务,静态许可用尽后开始生成动态许可启动任务。生成动态许可时,控制模块将可用资源与资源趋势相乘后与任务需求比较,若不满足需求则不分配。资源趋势 R_t 是为了缓冲资源变化而设计的,其定义及推导参见式(2)。

$$R_t = R_t / R_c = (R - \sum_{\gamma \in \text{node}} \gamma_t) / (R - \sum_{\gamma \in \text{node}} \gamma_c) = 1 + (\sum_{\gamma \in \text{node}} (\gamma_c - \gamma_t)) / (R - \sum_{\gamma \in \text{node}} \gamma_c) = 1 - \gamma' / R_c \quad (2)$$

式中: R 、 R_c 、 R_t 分别代表总共、目前可用、将来可用的计算资源; γ 表示任务; γ_c 、 γ_t 、 γ' 分别代表任务当前、将来、变化的资源需求,这些信息从需求推测信息以及资源感知模块获得。

静态许可数目推荐设置为较小数值,这样少数

任务可以不等待许可生成而快速启动,同时为使用动态资源许可调节资源利用率留下空间。

2.3 需求推测与作业调度

现有的需求推测方法有历史信息分析、程序分析、沙盒执行等,这些方法存在延迟高、代价大、结果不准确等问题。需求推测模块利用 Hadoop 任务执行特点,在运行时使用迭代校正方法推测任务资源需求。Hadoop 采用分布式文件系统将数据均匀冗余地存放在多个节点上,并规定每个任务只处理一部分数据,因而同作业同类型的任务处理的数据大小相近且逻辑相同。相同大小、分布均匀的数据经过相同逻辑处理时,所需的计算资源近似相等。因此需求推测模块根据运行中任务的资源使用信息推测同作业同类型任务的资源需求,并使用后续任务资源使用情况对推测结果进行校正。

$$\alpha = \begin{cases} e^{-5/60}\alpha + (1 - e^{-5/60})\beta, & \alpha \neq 0 \\ \beta, & \alpha = 0 \end{cases} \quad (3)$$

式(3)为推测、修正任务资源需求 α 时采用的公式,该公式借鉴了 Linux 系统平均负载计算公式。首次收到任务资源信息 β 则直接作为推测信息,之后收到资源信息则进行迭代修正。式(3)避免了单次数据抖动对推测结果造成重大影响,且避免了遍历计算算术平均值的计算量。需求推测模块计算任务每个阶段内的平均资源需求,即阶段内每个进度资源需求的算术平均值,作为该作业同类型任务该阶段的资源需求。

作业调度器复用了公平调度器^[4]的一些机制,保持了对公平性和数据本地性的支持。它针对不同类型的资源许可将调度分为两个阶段:第一个阶段针对静态资源许可,调度中只考虑可用的静态许可数目,使 Hadoop 在资源竞争中获得一定资源;第二阶段的资源适应调度首先减去上一阶段使用的资源量,调度器根据剩余可用资源和任务需求选择合适的任务,并考虑 reduce 任务分配情况对选出的任务进行过滤以避免 IO 瓶颈,适应资源变化。

3 实验验证

3.1 测试环境与测试负载

本文使用国家高性能计算中心(西安)的部分集群对资源许可方法进行验证。集群由 3 类共 37 个节点组成:①A 类高性能节点 4 个,每节点配有 2 个至强 E5430 2.66 GHz CPU,共 8 核心,内存为 16 GB,2 块 146 GB Ultra320 SCSI 硬盘;②B 类中性能节点 25 个,每节点配有 2 个至强 E5310 2.33 GHz CPU,

共 8 核心,内存为 8 GB,1 块 146 GB Ultra320 SCSI 硬盘;③C 类低性能节点 8 个,每节点配有 2 个至强 E5310 1.6 GHz CPU,共 8 核心,内存仅为 4 GB,1 块 146 GB Ultra320 SCSI 硬盘。

上述所有节点配有 Jdk1.6.0_45 版本 Java 环境和 1.0.1 版本 Hadoop 平台。Hadoop 中管理节点由 1 个 B 类节点充当,其余 36 个节点均作为计算节点。

本文测试环境分为 3 种,分别为独享集群环境、模拟竞争环境、真实云环境。测试负载分为单作业负载和混合负载,均由单词统计、字符串匹配、排序、Terasort、Quasi-Monte Carlo 方法计算圆周率、倒排索引、全球平均温度计算等作业组成。本文对比方法为 Facebook 公司提出的公平调度算法^[4],混合负载测试中各作业处理数据大小服从文献[4]中描述的 Facebook 公司作业数据大小分布,且提交时间服从泊松分布。每种测试在相同环境下重复至少 3 次,测得使用资源许可方法和在 10 种静态配置下的公平调度器的负载完成时间。具体静态配置为从 1mlr 到 10ml0r,1mlr 即配置 1 个 map 类型 slot 和 1 个 reduce 类型 slot,依此类推。

本文测试图以公平调度器所处的静态配置作为横坐标,以相对公平调度器在某种静态配置下取得的作业完成时间的变化作为纵坐标,将使用资源许可方法后性能的变化呈现出来。测试图中的散点代表一次测试结果,而折线由散点平均值绘得。图中纵坐标值大于 0 的点意味着使用资源许可方法相比使用公平调度器的 Hadoop 在横坐标所示的静态配置下取得的负载完成时间减少、性能提升,而纵坐标值小于 0 的点意味着负载完成时间增加、性能下降。

3.2 独享环境测试

独享环境测试中,整个集群中只执行 Hadoop 任务,环境中无其他应用的资源竞争影响。资源许可方法在独享集群环境中相对公平调度器在各种静态配置下取得的性能变化如图 2、3 所示。

由图 2 可见,不同作业的性能变化曲线趋势相同,均为先下降后升高,这表示在达到最优完成时间之前,负载完成时间随节点配置 slot 数增加而减少,之后随节点配置 slot 数增加而增加。这说明充分利用资源启动任务可以提升性能,而过多启动任务带来的资源竞争使性能下降。另外,各负载最优完成时间,即折线最低点处,在不同静态配置下取得,这证明了负载资源需求存在差异。图 2 中各折线有少数点纵坐标为负,即资源许可方法少数情况

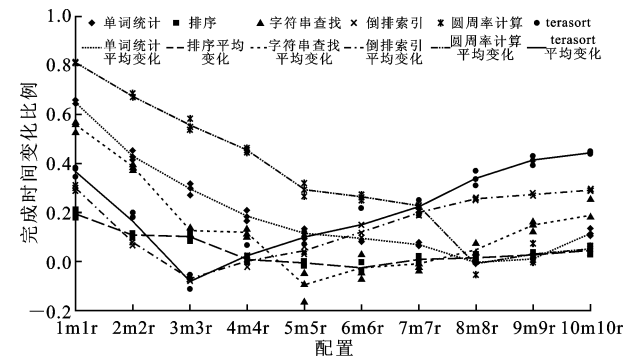


图 2 资源许可方法在独享集群中针对单作业取得的性能变化

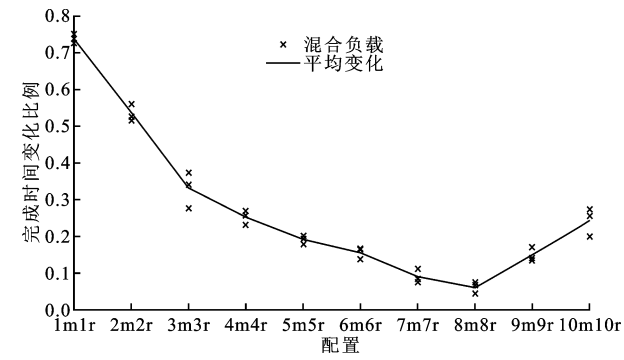


图 3 资源许可方法在独享集群中针对混合负载取得的性能变化

相对静态配置下的最优结果而言性能略有下降,程度从不足 1%到 10%不等。出现这种情况的原因是资源许可方法在采集资源信息、推测任务需求、调度及许可管理等方面付出的代价。资源许可方法相对最优静态结果略有下降是可以接受的,原因在于:①各作业最优完成时间在多个不同静态配置下取得,而 Hadoop 运行时配置只有一种,无法使所有作业取得最优,而资源许可方法相比公平调度器在所有单一静态配置均有性能提升,相比 1m1r 性能提升 48.1%最多,相比 5m5r 性能提升 7.4%最少,平均提升 18.1%;②作业在何种静态配置下取得最优完成时间受多种因素影响,即使集群中只运行一种作业,仍然很难对其手动配置;③资源许可方法在无资源竞争环境针对所有单作业取得只比静态最优略差的结果,整个过程自动完成。

图 3 中所有点纵坐标均大于 0,即混合负载测试中资源许可方法相对于公平调度在所有静态配置均有性能提高,平均提高 27.5%。混合负载会由于作业类型、数据大小等因素在不同静态配置下取得最优结果,因此资源许可方法自动取得优于静态最优的结果是很出色的,这也证明资源许可方法适应

环境中节点差异化的计算能力。

3.3 模拟环境测试

为了模拟真实环境中的资源竞争,测试环境中加入了 3 种资源需求不同的竞争程序,其资源需求见表 1。每次测试时竞争程序均以相同分布与 Hadoop 作业同时开始执行,保证测试环境相同。图 4、5 所示为资源许可方法在模拟环境中分别针对单作业和混合负载的测试结果。

表 1 竞争程序资源需求

程序名称	CPU 需求	内存需求	IO 需求
CPU 密集型程序	较大	较小	无
IO 密集型程序	较小	较大	较大
均衡型程序	较小	较小	较小

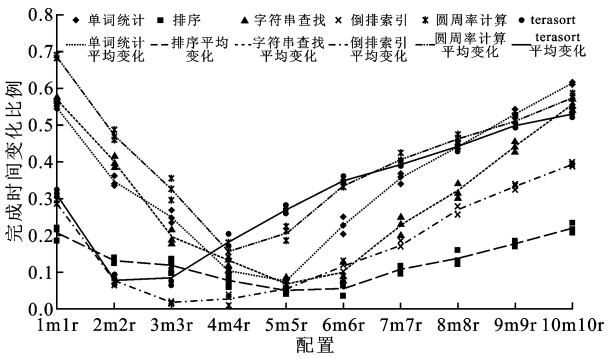


图 4 资源许可方法在模拟集群中针对单作业取得的性能变化

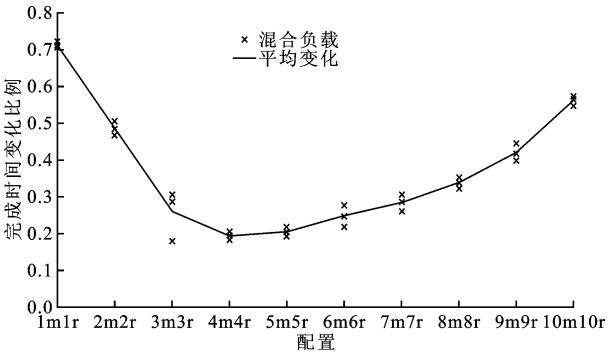


图 5 资源许可方法在模拟环境中针对混合负载取得的性能变化

对比图 2 和图 4 中各曲线,不难发现相同作业在竞争环境中取得最优性能的静态配置发生变化。这证实了多样负载并存带来的资源竞争对 Hadoop 性能的影响是存在的。

在模拟环境中,使用资源许可方法对单作业取得明显的性能提升,所有结果均优于公平调度在静态配置下取得的结果,提升程度从 11.3%到 48.2%

不等,平均提升达到 28.1%。

对比图 5 和图 3,可以发现资源许可方法在竞争环境中发挥更大作用,取得了更大的性能提升:与公平调度器比较,相对最优静态配置提高 19.6%,而相对所有静态配置平均取得 37.1%的提升,这证明资源许可方法适应了多样负载并存的资源竞争环境,并取得了良好的效果。

3.4 真实环境测试

图 6 为资源许可方法在实际使用中的云计算环境下的测试结果。此测试使用国家高性能计算中心(西安)对外服务的部分集群,Hadoop 混合负载与能源与动力、材料科学等学科的真实科学计算及仿真应用共享集群资源,环境中可用资源时刻变化,且各种负载的资源需求呈多样化。

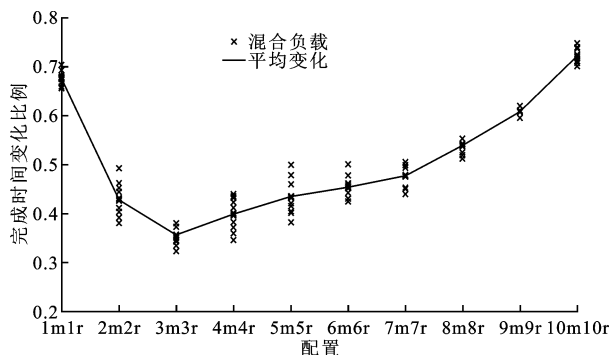


图 6 资源许可方法在真实环境中针对混合负载取得的性能变化

资源许可方法在真实环境中取得的性能提升较之前两种环境更为明显:相比公平调度器,平均取得 50.98%的性能提升,相较于 8 核机器常见的 4m4r 的静态配置有 39.92%的性能提升,即使相对于在静态配置中最优的 3m3r 依然有 35.57%的性能提升。本文认为,Hadoop 在真实环境下性能大幅提升的主要原因是原有基于 slot 的划分、调度资源的方式在可用资源时刻变化的真实环境中时常引起资源浪费或负载过重的情况,导致性能较差。静态配置 slot 数目少时,如果集群可用资源增多,Hadoop 限于静态 slot 数目不能启动更多任务,造成计算资源浪费,无法达到更好的性能。随着配置 slot 数目增加,Hadoop 可以启动更多任务并行执行,性能逐渐提升,但环境中多样负载并存造成可用资源持续变化,而 Hadoop 在运行时不能控制并行的任务数,节点上资源浪费或负载过重的情况仍然时有发生。继续增加 slot 数目配置,节点上并行任务过多,节点上资源浪费逐渐消失而系统负载越来越重,激烈的资源竞争导致性能开始下降,最终下降至比低

slot 数目配置更差。

与上述情况不同的是,资源许可方法感知计算资源变化,根据节点上可用计算资源以及任务资源需求动态划分资源,调度、启动任务,避免资源浪费和负载过重,适应环境中节点计算能力差异化和多样混合负载并存。另外,由于资源许可方法利用空闲资源提高 Hadoop 性能而不会主动与其他负载争抢资源,所以使用该方法时其他负载的完成时间基本保持不变。

4 结 论

针对云计算环境中普遍存在的节点计算能力差异化、多样混合负载共存等特点影响 Hadoop 性能的问题,本文提出了涉及多方面协同优化的资源许可方法,该方法在运行时以更细粒度动态划分计算资源、决定作业调度以屏蔽节点计算能力差异,通过控制发放给任务的资源许可数量适应环境中可用资源变化,避免资源浪费和负载过重等情况,提高 Hadoop 性能。通过独享、模拟、真实环境下的大量实验证明了方法的有效性,该方法在各种环境下均能取得性能提升,尤其在真实云环境中,性能提升显著,效果出色。

参考文献:

- [1] REISS C, TUMANOV A, GANGER G R, et al. Heterogeneity and dynamicity of clouds at scale: google trace analysis [C]// Proceedings of the Third ACM Symposium on Cloud Computing. New York, USA: ACM, 2012: 7.
- [2] DEAN J, GHEMAWAT S. MapReduce: simplified data processing on large clusters [J]. Communications of the ACM, 2008, 51(1): 107-113.
- [3] 李建江, 崔健, 王聘, 等. MapReduce 并行编程模型研究综述 [J]. 电子学报, 2011, 39(11): 2635-2642. LI Jianjiang, CUI Jian, WANG Dan, et al. Survey of MapReduce parallel programming model [J]. Acta Electronica Sinica, 2011, 39(11): 2635-2642.
- [4] ZAHARIA M, BORTHAKUR D, SARMA J S, et al. Job scheduling for multi-user MapReduce clusters, UCB/EECS-2009-55 [R]. Berkeley, CA, USA: University of California, EECS Department, 2009.
- [5] YANG Hailong, LUAN Zhongzhi, LI Wenjun, et al. MapReduce workload modeling with statistical approach [J]. Journal of Grid Computing, 2012, 10(2):

(下转第 108 页)