

一种面向异构大数据计算框架的监控及调度服务

胡雅鹏 丁维龙 王桂玲

(北方工业大学数据工程研究院 北京 100144)

(大规模流数据集成与分析技术北京市重点实验室 北京 100144)

摘 要 各种类型的大数据计算框架存在各自专用的管理方法。传统的监控和调度服务在异构环境下的操作由于无法获取集群整体的运行状态而受到限制,且无法综合多粒度的运行时资源状态来调度不同的计算作业。这不仅浪费了集群的可用资源,而且增加了计算作业的等待时间。针对上述两个问题,提出了一种面向异构大数据计算框架的一体化监控及动态调度管理服务。该服务可以自动适应并监控多种类型的大数据计算框架及计算作业,并对多类型作业提供一体化调度。针对 Hadoop 和 Storm 两种计算框架,实现了原型系统并进行了实验。实验结果表明,所提服务在异构环境下的大数据计算框架中能降低人工操作的复杂度,并且能提高作业的调度效率。

关键词 作业调度,集群监控,管理服务,作业提交

中图法分类号 TP315 文献标识码 A DOI 10.11896/j.issn.1002-137X.2018.06.011

Monitoring and Dispatching Service for Heterogeneous Big Data Computing Frameworks

HU Ya-peng DING Wei-long WANG Gui-ling

(Data Engineering Institute, North China University of Technology, Beijing 100144, China)

(Beijing Key Laboratory on Integration and Analysis of Large-scale Stream Data, Beijing 100144, China)

Abstract Various types of large data computing frameworks have their own management methods. The operation of traditional monitoring and scheduling service in heterogeneous environment is limited by the global status of cluster. It not only wastes resource of cluster, but also suffers long executive latencies of job. To solve these problems above, this paper presented an integrated monitoring and dynamic scheduling management service for heterogeneous big data computing framework. The service can monitor multiple types of computing framework automatically and provide integrated dispatching for diverse computing jobs. The work was implemented on Hadoop and Storm. The experimental results show that the service can reduce the complexity of manual operation in heterogeneous environment and improve job scheduling efficiency.

Keywords Job scheduling, Cluster monitoring, Management service, Job submission

1 引言

物联网的普及,促使着越来越多的设备产生了大量异构的数据。为了分析和处理这些异构数据,研究者们提出了多种类型的分布式计算框架。有的框架专门负责批处理计算,如 Hadoop¹⁾, Spark²⁾等;有的框架则只能处理流式计算,如 Storm³⁾, Flink⁴⁾等。这些计算框架之间彼此独立,并且都有一套单独的调度逻辑,这阻碍了共享环境下对资源的统一管理。由于每个计算框架内的资源使用情况彼此独立,运维人

员需要进入到不同的计算框架内部才能得知系统当前的运行情况;另外,计算框架种类的增加,使得调度不同种类的计算作业更加复杂,这也是当前亟待解决的问题。

近年来,针对上述需求,研究者们从计算框架监控数据的准确度、一体化监控平台的兼容性以及分布式作业的调度方案等方面开展了大量研究^[1-3]。但是,当前的监控平台大多只能监控一种类型的计算框架,如有些研究者提出针对 Hadoop 的监控,这种单一类型的监控平台并不能很好地适应异构的大数据计算环境^[4]。随着业务需求的发展,分布式系统也出

¹⁾ <http://hadoop.apache.org/>

²⁾ <http://spark.apache.org/docs/latest/>

³⁾ <http://storm.apache.org/index.html>

⁴⁾ <http://flink.apache.org/introduction.html>

到稿日期:2017-03-11 返修日期:2017-06-05 本文受国家自然科学基金(61702014),北京市自然科学基金(4172018)资助。

胡雅鹏(1991—),男,硕士生,主要研究方向为分布式系统,E-mail:huyapeng200828@126.com;丁维龙(1983—),男,博士,讲师,CCF 会员,主要研究方向为实时数据处理、分布式系统,E-mail:dingweilong@ncut.edu.cn(通信作者);王桂玲(1978—),女,博士,副研究员,CCF 会员,主要研究方向为服务计算、面向服务的数据集成、大规模流数据处理和集成。

现了越来越多的分类,这使得传统的大数据计算管理服务面临两个问题。

1)传统的大数据计算框架监控方法只针对单一类型的框架,针对多种集群下的监控则需要配置多套监控系统,从而导致资源消耗增加、集群计算能力降低以及人工开销增加。

2)传统的作业管理系统使用的是分布式系统框架默认的调度方式,这种静态调度方法没有结合当前集群的负载来进行动态调整,无法最大化利用当前的系统资源,并且会增加作业的等待时间。

针对以上问题,本文提出了一种大数据计算环境中的适应不同类型框架的监控与调度服务。该服务将监控指标重新进行分类,实现一体化的监控;与此同时,针对不同类型的大数据计算作业,提供了统一的作业提交和调度方法。该服务可提高整个系统运行维护的效率,缩短计算作业的等待时间,提升系统资源的利用率。

本文第2节介绍相关工作;第3节主要介绍一体化监控和作业调度方法;第4节通过实验验证动态调度的效率以及服务的应用与评价;最后总结全文并进行展望。

2 相关工作

近年来,许多专家对云环境的监控和作业调度进行了研究,研究方向包括环境监控、云环境架构设计、集群负载预测以及云平台作业调度等。除此之外,还有很多商用或者开源监控框架对大数据计算框架进行了监控。

在学术界,郭晓慧等人^[5]使用 Zabbix 在分布式服务器中进行监控。Zabbix 能够提供完整的报警机制,相比于其他监控框架,不需要下载和配置多种插件。除此以外,Zabbix 功能强大,适用范围广。但是,其配置客户端较为复杂,从 API 获取数据的难度较大。于国防等人^[6]对集群服务器进行延时响应,以预测集群的负载情况。该方法在网络带宽高负载的情况下能够很好地反映集群的负载情况,但是如果集群内 CPU 或者磁盘负载较高,而网络却处于空闲状态,那么通过网络延时来判断集群负载就会产生误差。

在工业界,Cloudera Manager 是一个商用监控框架^[7],运维人员可以使用该框架来对大数据计算节点进行图形化的控制。该框架可以自动地对节点进行配置,并且能定期生成系统性能评估报告等;但是由于没有对大数据作业的提交和调度进行管理,无法对计算作业提供统一的提交方式。Apache Ambari 是一个开源监控框架,运维人员可以通过使用该开源框架对计算节点进行简单方便的管理^[8];但是其没有提供统一的作业提交接口,同时也没有根据当前集群的运行情况对作业调度进行动态调整,无法最大化地利用集群资源。

针对此类问题,大部分研究仅停留在对某一种计算框架的监控,或者是针对某一种计算框架的作业调度算法进行优化上,使得集群管理服务无法对异构的集群提供一体化的监控方法,也无法对异构的计算作业提供一个更为合理的调度算法。

3 大数据计算的监控与调度服务

当集群中运行着多种类型的大数据计算框架时,通过一

体化的监控可以对这些异构的计算框架统一进行监控。一体化作业调度是指可以通过统一的入口将不同类型的作业提交到合适的集群并对其进行调度。通过将不同种类的集群、系统、服务以及环境进行分类,可以将相同的部分放到一个界面下进行统一管理,而对不同的部分则根据特征进行区分管理。如图1所示,底层运行着多种类型的计算框架,这些计算框架彼此独立。在计算框架之上的是一体化监控服务和作业调度服务。

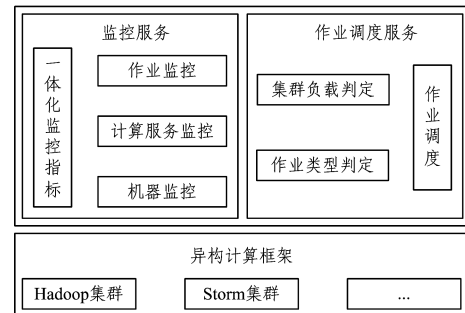


图1 大数据计算监控与调度服务框架

Fig. 1 Big data computing monitoring and scheduling service framework

监控服务主要提供的功能有:1)获得当前集群的运行指标,包括CPU负载、内存使用情况、磁盘I/O值以及网络带宽吞吐率等,为上层逻辑控制层的分析提供数据,并且能对机器进行诸如关机和重启等操作;2)对计算服务框架进行监控,对其运行作业数量、空闲资源进行统计;3)对运行中的作业进行监控,包括重新运行作业、结束作业以及重新对作业进行负载均衡等操作。作业调度服务提供的功能有:1)根据监控服务传来的监控指标对集群负载进行判定;2)在提交作业时对作业的类型进行判定;3)结合集群负载以及作业类型,将作业提交到最合适的节点上,并动态指定运行顺序。

3.1 计算框架的一体化监控

一体化监控最核心的问题是监控指标的时效性以及不同计算框架的兼容性。在监控的过程中,不仅要随时获取运行集群的物理机或虚拟机的参数指标,而且要对计算框架的当前状态进行监控。在获取监控数据的同时,要尽量减少监控对整体集群的资源占用率。

对于大数据计算框架而言,主要监控指标有CPU负载、内存使用率、磁盘使用率以及网络吞吐率。

定义1(集群单项负载高位阈值、集群单项负载低位阈值) 导致集群性能严重下降的单项资源监控值被称为集群负载高位阈值,表示为 $utiload_r^{high}$;集群利用率较低时的单项资源监控值被称为集群负载低位阈值,表示为 $utiload_r^{low}$ 。其中, $r \in \{cpu, disk, \dots, memory\}$ 代表集群可用资源,如CPU资源、内存资源、磁盘利用率、网络I/O等。

不同的集群对高/低位阈值区间的设置也有差异。例如,配置较低的集群, $utiload_r^{high}$ 的值达到80%就会导致集群性能严重下降;但是对于配置较高的集群,80%的资源使用率的影响较小。

定义2(集群综合负载高位阈值,集群综合负载低位阈值) 导致集群性能严重下降的综合资源监控值被称为集群

综合负载高位阈值,表示为 $multiload_r^{high}$;导致集群利用率较低的综合资源监控值被称为集群综合负载低位阈值,表示为 $multiload_r^{low}$ 。

当集群单项监测值没有达到单项高/低位阈值时,则需要通过综合集群所有指标来判断集群当前的负载值。不同的集群对综合负载阈值区间的划分也有差异,需要根据集群的配置进行设置。

定义 3(探测周期、探测时间间隔和探测次数) 探测周期为记录集群运行状态的时间跨度,表示为 T ;探测时间间隔为在探测周期内每两次采样的时间间隔,表示为 l ,具体探测间隔如表 1 所列;探测次数 n 为某一个完整探测周期内总共采样的次数, $n = \frac{T}{l}$ 。

表 1 监控指标
Table 1 Monitoring index

监控指标	CU	MU	DU	NT
探测间隔	l_c s	l_m min	l_d min	l_t s

表 1 中, CU 表示 CPU 负载率, MU 表示内存使用率, DU 表示磁盘使用率, NT 表示网络上下行吞吐率。

定义 4(资源 r 高负载集群) 在时间 T 内,若一个集群中有任意一个资源 r 的利用率 E_r 超过 $utiload_r^{high}$ 的次数大于 f_1 ($0 < f_1 \leq n$),则将该集群标记为针对资源 r 导致的高负载集群。

定义 5(资源 r 低负载集群) 在时间 T 内,若一个集群中所有资源 r 的利用率 E_r 低于 $utiload_r^{low}$ 的次数大于 f_2 ($0 < f_2 \leq n$),则将该集群标记为针对 r 导致的低负载集群。

资源利用率的单次探测数值只能代表集群当前的瞬时负载。如果只将瞬时数值作为集群负载的判断依据,会造成集群负载的判断出现误差。只有当某集群资源的利用率超过高/低位阈值一段时间之后,才能判断该虚拟机在这一资源上是高负载或是低负载的。对于一个给定的 n 值, f 值越大,则对集群负载稳态的判断越保守; f 值越小,则对集群负载稳态的判断越敏感。对于检测到的高负载集群,同时记录其高负载资源集合 $Hload_r$ 以及触发次数 m 。例如,某集群在 CPU 和内存这两个资源上触发了高负载,则 $Hload_r = \{cpu, memory\}$ 。

定义 6(负载值) 负载值是集群当前资源利用率的评判标准。负载值的计算如式(1)所示。其中, AE_r 为集群资源 r 的利用率在时间 T 内的 n 次检测结果的平均值; ρ_r 为单项资源 r 在所有资源中所占的权值,且满足 $\sum \rho_r = 1$ 。

$$loadValue = \begin{cases} \sum_{r \in \{cpu, disk, \dots, memory\}} AE_r \times \rho_r, & AE_r < utiload_r^{high} \\ \max_{r \in \{Hload_1, \dots, Hload_m\}} (AE_r), & AE_r \geq utiload_r^{high}, 0 < h \leq m \end{cases} \quad (1)$$

在式(1)中,若单项平均负载值超过高负载阈值,则集群负载值等于所有超过阈值的单项负载值的最大值。例如,如果时间 T 内集群 CPU 和内存的 AE_r 高于单项高位阈值,那么在这两项指标中选取最大值作为集群负载;若单项平均负载值没有超过高负载阈值,则需要通过对集群中所有的监控

指标进行加权求和,来得到集群当前的综合负载值。

通过对不同资源进行加权求和,可以更加准确地得到集群当前的负载值。在大数据计算框架中,计算作业属于计算资源密集型。在这种情况下,除非单点达到阈值,否则硬盘或者网络吞吐率等单项相比于 CPU 和内存而言权重较低。通过加权求和,能够以计算密集型资源为重点进行集群状态的评估。

3.2 计算作业的一体化管理

作业的一体化管理分为作业的一体化提交、一体化调度,以及一体化监控作业的一体化调度。其核心目标是解决不同种类的计算作业的统一提交以及不同种类的作业在不同集群中的调度问题。作业的提交过程如图 2 所示。

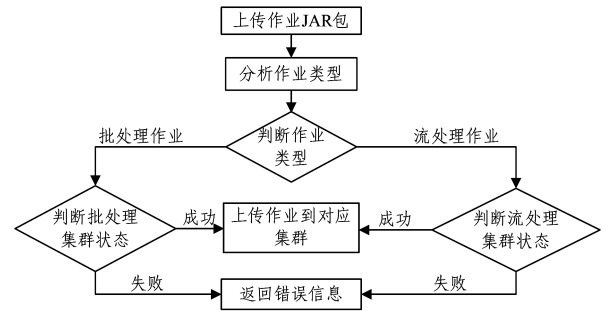


图 2 作业的一体化提交流程

Fig. 2 Integrated submission process of jobs

当用户在客户端提交 JAR 包后,系统对作业类型进行判断,同时结合当前的集群信息来判断集群能否运行该类型的作业。通过对不同类型的计算框架的作业提交接口进行封装,作业一体化提交服务可以将判定成功的作业提交到指定集群中;若没有合适的集群或者集群负载过高,则拒绝本次提交并返回错误信息。

在 Hadoop 调度算法中,公平调度算法的吞吐率高于 FIFO 算法的吞吐率^[9]。在公平调度算法中,按照资源池来组织作业,并把集群资源公平地分到这些资源池中;也可以手动指定某个资源池的相应权重。在公平调度算法中,需要配置的变量有: $map\ slot$ (map 操作的逻辑资源单位)、 $reduce\ slot$ ($reduce$ 操作的逻辑资源单位)、 $pool\ weight$ (资源池权重)、 $max\ running\ jobs$ (最大运行作业数)、 $min\ maps$ (最少 map 数量)、 $min\ reduce$ (最少 $reduce$ 数量)。对于管理员而言,上述参数需要进行手动配置。但是,集群的负载在时刻变化,固定的配置文件不能很好地配合当前集群的状况。

作业的一体化调度针对传统的公平调度算法进行了优化。经过优化后的公平算法,其配置文件会根据集群当前的运行情况对参数进行动态调整,以最大化地利用集群资源。对于已经完成的作业,作业一体化调度系统会将本次作业的名称、运行作业时集群的负载情况以及完成时间等信息写入历史作业数据库。当下次再执行作业时会判断作业是否已在集群运行过,如果运行过,则根据数据库信息对作业的运行时间以及占用资源的情况进行估计。

作业一体化调度服务会根据集群的配置来定时执行调度器优化模块。优化模块根据集群的当前运行状态及当前作业的队列长度来更改公平调度器的配置文件,具体流程如图 3 所示。

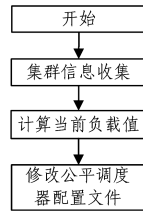


图3 公平调度的优化流程

Fig. 3 Optimization process of fair scheduling

根据第 3.1 节的集群负载判断规则,当集群处于高负载状态时,作业调度控制模块降低最大运行作业数的值,同时降低高权值缓冲池的权值,以减少池中作业总数,尽快降低集群负载;当集群处于低负载时,作业调度控制模块提高有作业的缓冲池的权值,同时适当增加最大作业数的值,以提高 $min\ maps$ 和 $min\ reduce$ 的值,尽量让更多的作业进入集群,提高吞吐率。

如图 4 所示,用户的作业经过一体化提交服务进入到集群以后,提交服务会查询该作业在集群内是否运行过。若作业运行过,则根据作业的历史运行时间对所需资源进行预估,并对作业所需资源进行高/低资源占用区分。当集群负载较高时,则将低资源占用作业排到队首;当集群负载较低时,则优先运行高资源占用作业。若作业以前没有在集群内运行过,则分析作业代码的时间复杂度同时对操作的数据量进行分析,从而对作业的运行时间以及占用的资源进行估计,之后根据集群当前的状态进行调整。对于跨种类作业的一体化管理,其核心目标是解决当前不同种类作业的信息收集、汇总、合理展示以及对以后新出现的计算作业进行兼容等问题。

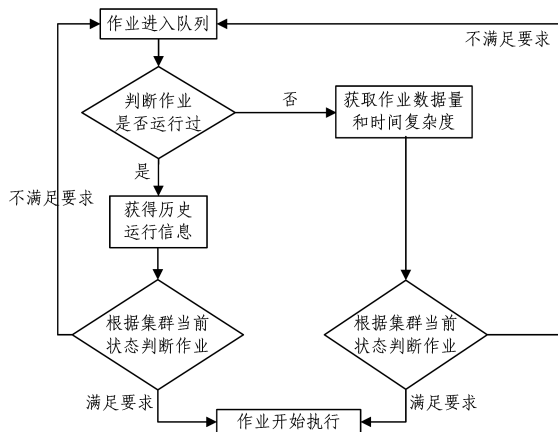


图4 作业调度流程图

Fig. 4 Flowchart of job scheduling

为了实现对作业的一体化管理,对于大多数计算框架,如 Hadoop, Spark 等,作业管理服务通过使用计算框架提供的作业管理接口实现对作业的管理;对于没有提供作业管理接口的计算框架,作业管理服务则通过直接对主节点发送指令来对集群中的作业进行控制。除此以外,一体化管理服务还能够抓取其他监控框架中的数据并对其进行整理汇总,从而实现更加灵活的作业管理。

4 应用与实验评价

本文实验的目的为对动态调整的公平调度算法与传统静

态调度算法的效率进行对比。目前,原型一体化监控及作业调度系统已经被应用于物联网大数据云服务平台上。

实验环境为:CentOS 6.4 版本的 Linux 操作系统, JDK 1.7, TomCat 1.6 版本。集群的环境为:两个 Hadoop 集群,每个集群分别拥有 1 个主节点以及 4 个从节点;一个 Storm 集群,其拥有 1 个主节点和 3 个从节点;13 台 Linux 虚拟机;7 台 Windows 虚拟机;3 台 Linux 物理机。所有节点和服务器的配置均为 4 核心 CPU, 4GB 内存, 10GB 有效存储容量。

4.1 实验结果分析

为了测试所提动态调整公平算法的效率,在 Hadoop 集群中用不同的调度方式运行相同的一批作业,以测试在单位时间内不同调度方式能调度的作业数量,同时测试根据历史运行作业进行动态调度的效果。实验重复多次,以保证实验数据的可靠性。

1) 变量设置

由于集群配置不同,3.1 节中的变量设置也有相应变化。如对于配置较高的集群,可以适当提高 $utiload_r^{high}$ 的值并适当降低 $utiload_r^{low}$ 的值。根据实验环境所使用的集群配置以及使用经验,将 3.1 节定义的变量设置为: $utiload_r^{high}=0.9$, $utiload_r^{low}=0.3$, $multiload_r^{high}=0.85$, $multiload_r^{low}=0.4$, $T=20\ min$, $l_c=30\ s$, $l_m=2\ min$, $l_d=2\ min$, $l_r=30\ s$, $f_1=10$, $f_2=10$, $\rho_{cpu}=0.5$, $\rho_{mem}=0.3$, $\rho_{disk}=0.2$ 。

2) 实验结果

实验结果如图 5 所示。实验结果显示,使用动态参数调度作业的方式在单位时间内调度作业的数量大于静态参数调度方法的数量。尤其当集群负载较高时,所提算法动态地调整公平调度器的参数,以减少进入到队列中抢占资源的作业数量,使得高负载作业能够更快地运行结束。同时,随着实验次数的增多,根据历史运行作业来动态调整调度策略的方式也进一步提高了调度效率。当负载较高的作业在集群中运行时,调度器根据历史作业的运行时间来调整作业队列,将历史运行消耗时间较少的作业放到队伍的首部,以在集群负载允许的情况下尽可能地调度更多的作业。实验结果也验证了这一点。

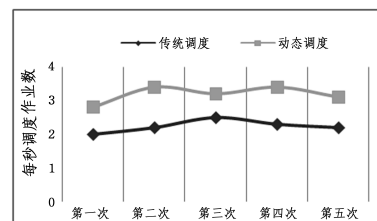


图5 作业调度效率的对比

Fig. 5 Comparison of job scheduling efficiency

3) 实验总结

相较于静态配置的调度器,根据集群当前的负载情况动态调整的公平调度算法在单位时间内可以调度更多的作业。随着调度作业次数的增多,根据历史作业的运行情况动态调整作业执行顺序的调度器可以进一步提高调度效率。

4.2 应用与评价

1) 集群的一体化控制图如图 6 所示。

主机名称	主节点	类型	服务管理	机器撤销	监控详情
1 node0	10.61.8.40	zookeeper, nimbus, storm UI	管理	撤销机器	监控详情
2 node1	10.61.8.41	supervisor	管理	撤销机器	监控详情
3 node2	10.61.8.42	supervisor	管理	撤销机器	监控详情
4 node3	10.61.8.43	supervisor	管理	撤销机器	监控详情
5 node4	10.61.8.44	supervisor	管理	撤销机器	监控详情
6 node5	10.61.8.45	supervisor	管理	撤销机器	监控详情
7 master	10.61.8.51	name node, job tracker	管理	撤销机器	监控详情
8 slaves1	10.61.8.50	data node, task tracker	管理	撤销机器	监控详情
9 slaves2	10.61.8.52	data node, task tracker	管理	撤销机器	监控详情

图 6 集群监控

Fig. 6 Cluster monitoring

在图 6 的列表中,只以机器名称和 IP 地址等机器信息作为标注,并不区分具体是物理机还是虚拟机;同时,在列表中可以对机器进行管理控制,如关机、重启、开机等;另外,通过监控详情还可以查看具体的监控指标,从而能更加具体地得到机器的运行情况、历史和当前负荷,使运维人员能方便地了解机器详情。

2)如图 7 所示,除了对机器进行管理控制,系统还能对集群节点进行控制,从而控制整个集群的运行情况。运维人员发现某节点的负载较高或者异常时,可以直接在管理页面进行处理,而不必调用命令行窗口。

编号	服务	状态	启动服务	停止服务	重启服务
1 120	name node	start	启动	停止	重启
2 121	job tracker	start	启动	停止	重启

图 7 集群控制

Fig. 7 Cluster controlling

3)如图 8 所示,服务可以通过区分不同的系统来显示当前分布式系统运行作业的情况,每一个运行的作业都能被显示出来;而且系统根据集群中运行的分布式框架的类型进行分类,将每种作业进行分开管理。系统在显示条目中展示作业的主要指标,包括运行的节点数、主节点信息、作业列表等,以方便运维人员了解作业运行的情况。在本页面不仅能监视 Hadoop,Storm 等主流分布式计算框架,还能监控 Kafka 的吞吐率等指标。

编号	系统	版本	主节点	Nimbus启动时间(s)	节点数
2	Storm流计算	0.9.4	10.61.8.40	3d 8h 34m 2s	5

编号	系统	主节点	运行中的Map任务	作业提交数	运行中
0	Hadoop 1.0.4	10.61.8.51	0	40	0

图 8 计算框架监控

Fig. 8 Computing framework monitoring

4)作业的一体化提交如图 9 所示。通过上传程序 JAR 包、分析 JAR 包的内容来判断作业类型,并将适合运行作业的集群列在下拉框中。提交程序将作业自动提交到指定集群,并开始调度。

机器服务

提交作业

作业名称:

作业类型:

集群地址:

上传程序包:

启动类名:

请选择作业类型

请选择IP

浏览...

未选择文件.

分析

请选择启动类名

提交

图 9 异构作业的统一提交接口

Fig. 9 Integrated submission interface for heterogeneous jobs

5)作业的具体运行情况如图 10 所示。通过点击作业的一体化控制中的详情,可以看到正在运行的作业的具体作业信息,包括作业运行的节点情况、运行状态等信息。

系统	作业名	Id	状态	启动时间
storm-2	FakePlate	FakePlate-7-1497761572	ACTIVE	3d 8h 18
storm-2	BlackList	BlackList-6-1497761552	ACTIVE	3d 8h 18
storm-2	correlation_analysis...	correlation_analysis4plant-9-149...	ACTIVE	3d 5h 4n
storm-2	OverSpeed	OverSpeed-3-1497761473	ACTIVE	3d 8h 19
storm-2	CopyPlate	CopyPlate-5-1497761530	ACTIVE	3d 8h 18
storm-2	YellowRank	YellowRank-8-1497761707	ACTIVE	3d 8h 15
storm-2	PlaceMonitor	PlaceMonitor-4-1497761502	ACTIVE	3d 8h 19

系统	Jobid	优先级
Hadoop1.0.4-1	job_201706021004_0001	NORMAL
Hadoop1.0.4-1	job_201706021004_0006	NORMAL
Hadoop1.0.4-1	job_201706021004_0008	NORMAL
Hadoop1.0.4-1	job_201706021004_0010	NORMAL
Hadoop1.0.4-1	job_201706021004_0011	NORMAL
Hadoop1.0.4-1	job_201706021004_0012	NORMAL
Hadoop1.0.4-1	job_201706021004_0014	NORMAL
Hadoop1.0.4-1	job_201706021004_0015	NORMAL
Hadoop1.0.4-1	job_201706021004_0016	NORMAL
Hadoop1.0.4-1	job_201706021004_0017	NORMAL
Hadoop1.0.4-1	job_201706021004_0018	NORMAL
Hadoop1.0.4-1	job_201706021004_0020	NORMAL
Hadoop1.0.4-1	job_201706021004_0021	NORMAL

图 10 计算作业的监控和调度

Fig. 10 Monitoring and scheduling of computing operations

根据实际应用的效果反馈可以得出,本文所提的一体化大数据计算框架服务可以有效地节省人工成本。通过一体化的监控不仅能够降低操作难度,还可以让整个系统根据监控值动态调整调度策略,从而提高作业调度效率。

结束语 本文提出了一种面向大数据计算框架及作业控制的一体化解决方法,并针对 Hadoop,Storm 等异构计算框架进行了实验。从实验结果来看,本文提出的大数据计算框架一体化管理服务确实能解决异构环境下集群监控复杂的问题,同时能够降低运维人员对异构计算作业的一体化管理和动态调度的工作难度,提高整个系统的调度效率。下一步的研究将聚焦在作业调度算法上,目前的算法由于引入了集群当前的运行信息这一个维度,因此比以往固定环境的算法的复杂度更高;同时,细化作业调度的评价指标,以进一步提高集群的运行效率。

参考文献

- [1] DONG B, SHEN Q, XIAO D B. Research on Monitoring Method of Cloud Computing Cluster Server System [J]. Computer Engineering & Science, 2012, 34(10): 69-71. (in Chinese)
董波,沈青,肖德宝.云计算集群服务器系统监控方法的研究[J].计算机工程与科学,2012,34(10):69-71.
- [2] DARLING C L, GERNAEY M E, KALDESTAD H S, et al. Dynamic monitor and controller of availability of a load-balancing cluster, US 7,296,268 B2[P]. 2007-11-13.
- [3] YUAN K. Design and Implementation of Monitoring System Based on Cloud[D]. Wuhan: Huazhong University of Science and Technology, 2012. (in Chinese)
袁凯.云计算环境下的监控系统设计与实现[D].武汉:华中科技大学,2012.
- [4] LIU J, LIU F, ANSARI N. Monitoring and analyzing big traffic data of a large-scale cellular network with Hadoop[J]. IEEE Network, 2014, 28(4): 32-39.