



(12)发明专利申请

(10)申请公布号 CN 106201754 A

(43)申请公布日 2016. 12. 07

(21)申请号 201610529219.X

(22)申请日 2016.07.06

(71)申请人 乐视控股(北京)有限公司

地址 100123 北京市朝阳区姚家园路105号
3号楼10层1102

申请人 乐视网信息技术(北京)股份有限公司

(72)发明人 刘光华

(74)专利代理机构 北京风雅颂专利代理有限公司 11403

代理人 李莎 李弘

(51)Int. Cl.

G06F 11/07(2006.01)

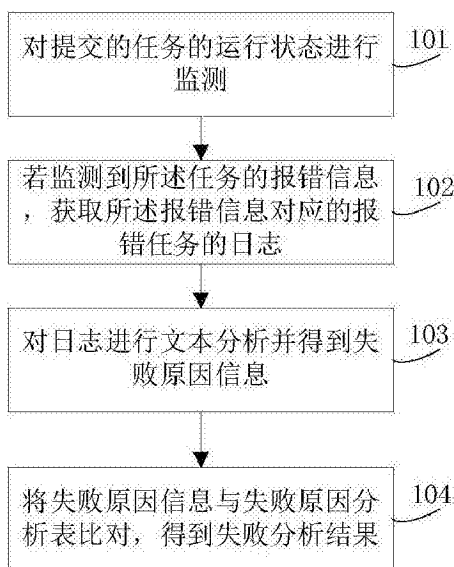
权利要求书3页 说明书16页 附图3页

(54)发明名称

任务信息分析方法及装置

(57)摘要

本发明公开了一种任务信息分析方法及装置,包括:对提交的任务的运行状态进行监测;若监测到所述任务的报错信息,获取所述报错信息对应的报错任务的日志;对所述报错任务的日志进行文本分析并得到所述报错任务的报错原因信息;将所述报错原因信息与预存的报错原因分析表进行比对,得到与所述报错原因信息相对应的报错分析结果;所述报错原因分析表中包括预存的报错原因信息以及与所述报错原因信息相对应的报错分析结果。本发明实施例提出的任务信息分析方法及装置,能够自动分析任务的运行情况,无需手动操作。



1. 一种任务信息分析方法,其特征在于,包括:
 - 对提交的任务的运行状态进行监测;
 - 若监测到所述任务的报错信息,获取所述报错信息对应的报错任务的日志;
 - 对所述报错任务的日志进行文本分析并得到所述报错任务的报错原因信息;
 - 将所述报错原因信息与预存的报错原因分析表进行比对,得到与所述报错原因信息相对应的报错分析结果,所述报错原因分析表中包括预存的报错原因信息以及与所述报错原因信息相对应的报错分析结果。
2. 根据权利要求1所述的方法,其特征在于,应用于MapReduce运算,所述任务包括map任务和reduce任务;
 - 所述对提交的任务的运行状态进行监测的步骤之后还包括:
 - 创建map队列和reduce队列,根据所述任务的运行状态,将报错的map任务和报错的reduce任务分别存储在所述map任务报错列表和reduce任务报错列表中;
 - 在所述map任务报错列表中记录报错的map任务的ID和执行所述map任务的机器的列表;
 - 在所述reduce任务报错列表中记录报错的reduce任务的ID和执行所述reduce任务的机器的列表;
 - 所述获取报错任务的日志的步骤具体包括:
 - 采用随机算法,从所述map任务报错列表和reduce任务报错列表中选取报错任务进行分析。
3. 根据权利要求1或2所述的方法,其特征在于,应用于MapReduce运算,所述任务包括map任务和reduce任务;
 - 所述对提交的任务的运行状态进行监测的步骤之后还包括:
 - 监测所述reduce任务的处理进度;所述处理进度包括需要抓取文件数、已经抓取文件数和文件抓取速度;
 - 在与所述reduce任务有关联的map任务结束后,启动从零开始的计时;
 - 判断计时时长超过第一预设时间阈值时是否存在未结束文件抓取的reduce任务;
 - 若存在未结束文件抓取的reduce任务,判断所述文件抓取速度是否低于预设文件抓取速度;
 - 若所述文件抓取速度低于预设文件抓取速度,判定当前网络状态不佳。
4. 根据权利要求1或2所述的方法,其特征在于,应用于MapReduce运算,所述任务包括map任务和reduce任务;
 - 所述对提交的任务的运行状态进行监测的步骤之后还包括:
 - 对于联接的所述reduce任务,监测所述reduce任务的日志;
 - 判断联接的所述reduce任务中,未完成的reduce任务占联接的全部所述reduce任务的比例是否低于预设未完成比例;
 - 若未完成的reduce任务占联接的全部所述reduce任务的比例低于预设未完成比例,则判断所述未完成的reduce任务的持续时间是否超过第三预设时间阈值;
 - 若所述未完成的reduce任务的持续时间超过第三预设时间阈值,则判断所述reduce任务的同一键的数据量是否超出预设键数据量阈值;

若所述reduce任务的同一键的数据量超出预设键数据量阈值,判定所述reduce任务属于数据倾斜,并发出数据倾斜警告信息。

5. 根据权利要求1或2所述的方法,其特征在于,应用于hive任务;

所述对提交的任务进行监测步骤之后还包括:

监测所述任务的失败信息;

根据所述任务的失败信息,判断所述任务是否连续请求分配AM失败达到预设失败次数;

若所述任务连续请求分配AM失败达到预设失败次数,则判定所述任务请求分配AM失败并输出AM的IP地址。

6. 一种任务信息分析装置,其特征在于,包括:

监测模块,用于对提交的任务的运行状态进行监测;

日志获取模块,用于若监测到所述任务的报错信息,获取所述报错信息对应的报错任务的日志;

日志解析模块,用于对所述报错任务的日志进行文本分析并得到所述报错任务的报错原因信息;

分析比对模块,用于将所述报错原因信息与预存的报错原因分析表进行比对,得到与所述报错原因信息相对应的报错分析结果,所述报错原因分析表中包括预存的报错原因信息以及与所述报错原因信息相对应的报错分析结果。

7. 根据权利要求6所述的装置,其特征在于,应用于MapReduce运算,所述任务包括map任务和reduce任务;

所述任务信息分析装置还包括报错列表创建模块,具体用于:

创建map队列和reduce队列;

根据所述任务的运行状态,将报错的map任务和报错的reduce任务分别存储在所述map任务报错列表和reduce任务报错列表中;

在所述map任务报错列表中记录报错的map任务的ID和执行所述map任务的机器的列表;

在所述reduce任务报错列表中记录报错的reduce任务的ID和执行所述reduce任务的机器的列表;

所述日志获取模块,具体用于:

采用随机算法,从所述map任务报错列表和reduce任务报错列表中选取报错任务进行分析。

8. 根据权利要求6或7所述的装置,其特征在于,应用于MapReduce运算,所述任务包括map任务和reduce任务;

所述监测模块,还用于监测所述reduce任务的处理进度;所述处理进度包括需要抓取文件数、已经抓取文件数和文件抓取速度;

所述任务信息分析装置还包括:

计时模块,用于在与所述reduce任务有关联的map任务结束后,启动从零开始的计时;

未结束任务判断模块,用于判断计时时长超过第一预设时间阈值时是否存在未结束文件抓取的reduce任务;

抓取速度判断模块,若存在未结束文件抓取的reduce任务,用于判断所述文件抓取速度是否低于预设文件抓取速度;

网络状态判断模块,若所述文件抓取速度低于预设文件抓取速度,用于判定当前网络状态不佳。

9.根据权利要求6或7所述的装置,其特征在于,应用于MapReduce运算,所述任务包括map任务和reduce任务;

所述监测模块,还用于对于联接的所述reduce任务,监测所述reduce任务的日志;

所述任务信息分析装置还包括:

未完成比例判断模块,用于判断联接的所述reduce任务中,未完成的reduce任务占联接的全部所述reduce任务的比例是否低于预设未完成比例;

持续时间判断模块,若未完成的reduce任务占联接的全部所述reduce任务的比例低于预设未完成比例,用于判断所述未完成的reduce任务的持续时间是否超过第三预设时间阈值;

数据量判断模块,若所述未完成的reduce任务的持续时间超过第三预设时间阈值,用于判断所述reduce任务的同一键的数据量是否超出预设键数据量阈值;

数据倾斜判断模块,若所述reduce任务的同一键的数据量超出预设键数据量阈值,用于判定所述reduce任务属于数据倾斜;

警告模块,用于发出数据倾斜警告信息。

10.根据权利要求6或7所述的装置,其特征在于,应用于hive任务;所述监测模块,还用于监测所述任务的失败信息;

所述任务信息分析装置还包括:

AM请求判断模块,用于根据所述任务的失败信息,判断所述任务是否连续请求分配AM失败达到预设失败次数;

请求失败判断模块,若所述任务连续请求分配AM失败达到预设失败次数,用于判定所述任务请求分配AM失败并输出AM的IP地址。

任务信息分析方法及装置

技术领域

[0001] 本发明涉及数据处理技术领域,特别是指一种任务信息分析方法及装置。

背景技术

[0002] Hadoop是一种分布式系统基础架构,用户可以在不了解分布式底层细节的情况下开发分布式程序,从而充分利用集群的威力进行高速运算和存储。Hadoop实现了一个分布式文件系统(Hadoop Distributed File System),简称HDFS。HDFS有高容错性的特点,并且设计用来部署在低廉的硬件上;而且它提供高吞吐量来访问应用程序的数据,适合那些有着超大数据集的应用程序。HDFS放宽了POSIX(Portable Operating System Interface,可移植操作系统接口)的要求,可以以流的形式访问文件系统中的数据。Hadoop的框架最核心的设计就是:HDFS和MapReduce(可译为“映射与化简”)。MapReduce是一种编程模型,用于大规模数据集的并行运算。HDFS为海量的数据提供了存储,MapReduce则为海量的数据提供了计算。

[0003] hive是基于Hadoop的一个数据仓库工具,可以将结构化的数据文件映射为一张数据库表,并提供sql(Structured Query Language,结构化查询语言)查询功能,可以将sql语句转换为MapReduce任务进行运行。在hadoop2.x中,资源调度管理器是yarn。Apache Hadoop YARN(Yet Another Resource Negotiator,另一种资源协调者)是一种新的Hadoop资源管理器,它是一个通用资源管理系统,可为上层应用提供统一的资源管理和调度,它的引入为集群在利用率、资源统一管理和数据共享等方面带来了巨大好处。在hive任务中,某些情况下任务会失败,而任务失败会有许多原因,对于初学者或者单纯的数据分析人员来说,很多时候hive给出的报错提示太少了,而不能帮助这些人员来分析他们的任务失败的原因。

[0004] 发明人在实现本发明的过程中发现,现有的方案,对于hive失败的任务,需要用户自己去查看logserver(日志记录服务器)上的hive运行的日志。但是logserver上保存的日志是有时间和数量限制的。并且,想要查看日志还需要配置集群所有的机器在本地的hosts映射,当集群节点发生改变时,还需要更新相关的配置,对于数据分析人员造成很多的不便。而且,在很多情况下,处于对于安全因素的考虑,Hadoop的相关机器并不能够被外网地址访问,这样就看不到具体的分布式运算节点上的信息。此外,hive直接在控制台上打印的日志,往往信息不足,无法直接用来定位错误的原因。

发明内容

[0005] 有鉴于此,本发明的目的在于提出一种任务信息分析方法及装置,能够自动分析任务的运行情况,无需手动操作。

[0006] 基于上述目的本发明实施例提供的任务信息分析方法,包括:

[0007] 对提交的任务的运行状态进行监测;

[0008] 若监测到所述任务的报错信息,获取所述报错信息对应的报错任务的日志;

- [0009] 对所述报错任务的日志进行文本分析并得到所述报错任务的报错原因信息；
- [0010] 将所述报错原因信息与预存的报错原因分析表进行比对，得到与所述报错原因信息相对应的报错分析结果；所述报错原因分析表中包括预存的报错原因信息以及与所述报错原因信息相对应的报错分析结果。
- [0011] 在一些实施方式中，所述的方法，应用于MapReduce运算，所述任务包括map任务和reduce任务；
- [0012] 所述对提交的任务的运行状态进行监测的步骤之后还包括：
- [0013] 创建map队列和reduce队列，
- [0014] 根据所述任务的运行状态，将报错的map任务和报错的reduce任务分别存储在所述map任务报错列表和reduce任务报错列表中；
- [0015] 在所述map任务报错列表中记录报错的map任务的ID和执行所述map任务的机器的列表；
- [0016] 在所述reduce任务报错列表中记录报错的reduce任务的ID和执行所述reduce任务的机器的列表；
- [0017] 所述获取报错任务的日志的步骤具体包括：
- [0018] 采用随机算法，从所述map任务报错列表和reduce任务报错列表中选取报错任务进行分析。
- [0019] 在一些实施方式中，所述的方法，应用于MapReduce运算，所述任务包括map任务和reduce任务；
- [0020] 所述对提交的任务的运行状态进行监测的步骤之后还包括：
- [0021] 监测所述reduce任务的处理进度；所述处理进度包括需要抓取文件数、已经抓取文件数和文件抓取速度；
- [0022] 在与所述reduce任务有关联的map任务结束后，启动从零开始的计时；
- [0023] 判断计时时长超过第一预设时间阈值时是否存在未结束文件抓取的reduce任务；
- [0024] 若存在未结束文件抓取的reduce任务，判断所述文件抓取速度是否低于预设文件抓取速度；
- [0025] 若所述文件抓取速度低于预设文件抓取速度，判定当前网络状态不佳。
- [0026] 在一些实施方式中，所述的方法，应用于MapReduce运算，所述任务包括map任务和reduce任务；
- [0027] 所述对提交的任务的运行状态进行监测的步骤之后还包括：
- [0028] 对于联接的所述reduce任务，监测所述reduce任务的日志；
- [0029] 判断联接的所述reduce任务中，未完成的reduce任务占联接的全部所述reduce任务的比例是否低于预设未完成比例；
- [0030] 若未完成的reduce任务占联接的全部所述reduce任务的比例低于预设未完成比例，则判断所述未完成的reduce任务的持续时间是否超过第三预设时间阈值；
- [0031] 若所述未完成的reduce任务的持续时间超过第三预设时间阈值，则判断所述reduce任务的同一键的数据量是否超出预设键数据量阈值；
- [0032] 若所述reduce任务的同一键的数据量超出预设键数据量阈值，判定所述reduce任务属于数据倾斜，并发出数据倾斜警告信息。

- [0033] 在一些实施方式中,所述的方法,应用于hive任务;
- [0034] 所述对提交的任务的运行状态进行监测的步骤之后还包括:
- [0035] 监测所述任务的失败信息;
- [0036] 根据所述任务的失败信息,判断所述任务是否连续请求分配AM失败达到预设失败次数;
- [0037] 若所述任务连续请求分配AM失败达到预设失败次数,则判定所述任务请求分配AM失败并输出AM的IP地址。
- [0038] 本发明实施例的另一方面,还提供了一种任务信息分析装置,包括:
- [0039] 监测模块,用于对提交的任务的运行状态进行监测;
- [0040] 日志获取模块,用于若监测到所述任务的报错信息,获取所述报错信息对应的报错任务的日志;
- [0041] 日志解析模块,用于对所述报错任务的日志进行文本分析并得到所述报错任务的报错原因信息;
- [0042] 分析比对模块,用于将所述报错原因信息与预存的报错原因分析表进行比对,得到与所述报错原因信息相对应的报错分析结果;所述报错原因分析表中包括预存的报错原因信息以及与所述报错原因信息相对应的报错分析结果。
- [0043] 在一些实施方式中,所述的装置,应用于MapReduce运算,所述任务包括map任务和reduce任务;
- [0044] 所述任务信息分析装置还包括报错列表创建模块,具体用于:
- [0045] 创建map队列和reduce队列,
- [0046] 根据所述任务的运行状态,将报错的map任务和报错的reduce任务分别存储在所述map任务报错列表和reduce任务报错列表中;
- [0047] 在所述map任务报错列表中记录报错的map任务的ID和执行所述map任务的机器的列表;
- [0048] 在所述reduce任务报错列表中记录报错的reduce任务的ID和执行所述reduce任务的机器的列表;
- [0049] 所述日志获取模块,具体用于:
- [0050] 采用随机算法,从所述map任务报错列表和reduce任务报错列表中选取报错任务进行分析。
- [0051] 在一些实施方式中,所述的装置,应用于MapReduce运算,所述任务包括map任务和reduce任务;
- [0052] 所述监测模块,还用于监测所述reduce任务的处理进度;所述处理进度包括需要抓取文件数、已经抓取文件数和文件抓取速度;
- [0053] 所述任务信息分析装置还包括:
- [0054] 计时模块,用于在与所述reduce任务有关联的map任务结束后,启动从零开始的计时;
- [0055] 未结束任务判断模块,用于判断计时时长超过第一预设时间阈值时是否存在未结束文件抓取的reduce任务;
- [0056] 抓取速度判断模块,若存在未结束文件抓取的reduce任务,用于判断所述文件抓

取速度是否低于预设文件抓取速度；

[0057] 网络状态判断模块,若所述文件抓取速度低于预设文件抓取速度,用于判定当前网络状态不佳。

[0058] 在一些实施方式中,所述的装置,应用于MapReduce运算,所述任务包括map任务和reduce任务；

[0059] 所述监测模块,还用于对于联接的所述reduce任务,监测所述reduce任务的日志；

[0060] 所述任务信息分析装置还包括：

[0061] 未完成比例判断模块,用于判断联接的所述reduce任务中,未完成的reduce任务占联接的全部所述reduce任务的比例是否低于预设未完成比例；

[0062] 持续时间判断模块,若未完成的reduce任务占联接的全部所述reduce任务的比例低于预设未完成比例,用于判断所述未完成的reduce任务的持续时间是否超过第三预设时间阈值；

[0063] 数据量判断模块,若所述未完成的reduce任务的持续时间超过第三预设时间阈值,用于判断所述reduce任务的同一键的数据量是否超出预设键数据量阈值；

[0064] 数据倾斜判断模块,若所述reduce任务的同一键的数据量超出预设键数据量阈值,用于判定所述reduce任务属于数据倾斜；

[0065] 警告模块,用于发出数据倾斜警告信息。

[0066] 在一些实施方式中,所述的装置,应用于hive任务；所述监测模块,还用于监测所述任务的失败信息；

[0067] 所述任务信息分析装置还包括：

[0068] AM请求判断模块,用于根据所述任务的失败信息,判断所述任务是否连续请求分配AM失败达到预设失败次数；

[0069] 请求失败判断模块,若所述任务连续请求分配AM失败达到预设失败次数,用于判定所述任务请求分配AM失败并输出AM的IP地址。

[0070] 从上述实施例可以看出,本发明实施例提供的所述任务信息分析方法及装置,通过设置任务守护线程对任务进行监控并得到报错任务,对报错任务的日志进行文本分析并提取得到报错原因信息,将所述报错原因信息与预先设置的报错原因分析表进行比对,最终得到并输出报错分析结果,使得在出现任务报错时,无需技术人员进行人工排查,就能自动得出报错任务的错误原因(即报错分析结果),从而大大提高了技术人员的工作效率。

附图说明

[0071] 图1为本发明提供的任务信息分析方法的第一个实施例的流程示意图；

[0072] 图2为本发明提供的任务信息分析方法的第二个实施例中步骤103的具体流程示意图；

[0073] 图3为本发明提供的任务信息分析方法的第三个实施例的流程示意图；

[0074] 图4为本发明提供的任务信息分析装置的第一个实施例的模块结构示意图；

[0075] 图5为本发明提供的任务信息分析装置的第二个实施例的模块结构示意图。

具体实施方式

[0076] 为使本发明的目的、技术方案和优点更加清楚明白,以下结合具体实施例,并参照附图,对本发明进一步详细说明。

[0077] 需要说明的是,本发明实施例中所有使用“第一”和“第二”的表述均是为了区分两个相同名称非相同的实体或者非相同的参量,可见“第一”“第二”仅为了表述的方便,不应理解为对本发明实施例的限定,后续实施例对此不再一一说明。

[0078] 在本文的一开始,先大致介绍hive任务的运算大体流程,其主要包括以下几个步骤:

[0079] 首先,向Resourcemanager(资源管理器,简称RM,是yarn资源控制框架的中心模块,负责集群中所有的资源的统一管理和分配)提交任务请求。

[0080] 然后,RM分配一个ApplicationMaster给这个任务;这里,ApplicationMaster简称AM,可译为“应用主控器”,是一个具体的框架库,它的任务是与RM协商获取应用所需资源,以及,与Nodemanager(简称ND,节点管理器)合作,以完成执行和监控所述任务。对于MapReduce计算模型而言有它自己的AM实现,对于其他的想要运行在yarn上的计算模型而言,必须得实现针对该计算模型的AM用以向RM申请资源运行任务。比如运行在yarn上的spark(可译为“火花”)框架也有对应的AM实现,这里,spark是UC Berkeley AMP lab(加利福尼亚大学伯克利分校安普实验室)所开源的类Hadoop MapReduce的通用并行框架,其拥有Hadoop MapReduce所具有的优点。归根结底,yarn是一个资源管理的框架,并不是一个计算框架,要想在yarn上运行应用程序,还得有特定的计算框架来实现,例如AM。

[0081] 最后,AM负责给这个任务在空闲的ND上分配相关的计算资源,根据执行计算启动与该任务相应的map任务或reduce任务。

[0082] 其中,计算任务的日志被保存到任务各自的本地。

[0083] 在hive任务的运算过程中,影响hive任务运行进度以及可能导致失败或报错的原因主要有以下几类:

[0084] (1)AM所在的机器节点发生故障,导致AM分配失败,需要重新分配AM,每次重新分配就会导致任务需要全部重新计算;如果AM连续两次分配失败,则任务失败;

[0085] (2)当前用户所在的资源队列资源不足,导致任务运行缓慢;

[0086] (3)当发生数据倾斜问题时,会导致其中某几个reduce任务运行特别缓慢;

[0087] (4)由于数据异常,或者自定义的udf函数(User-Defined Function,用户自定义函数)、udaf函数(User-Defined Aggregation Funcation,用户自定义聚合函数)等自身的问题,会导致map任务或者reduce任务报错。

[0088] 基于上述错误原因,本发明实施例的第一个方面,提出了一种能够自动分析任务的运行情况且无需手动操作的任务信息分析方法的第一个实施例。如图1所示,为本发明提供的任务信息分析方法的第一个实施例的流程示意图。

[0089] 所述任务信息分析方法,包括以下步骤:

[0090] 步骤101:对提交的任务的运行状态进行监测;可选的,可通过启动任务守护线程的方式来进行监测,所述守护线程则用于对提交的任务进行跟踪并获取所述任务的运行状态;

[0091] 举例来说,在Hadoop中,每一个hive任务都会被解析成MapReduce任务,每一个MapReduce任务都会有一个唯一的applicationid(应用ID);在hive启动过程中,则会启动

所述任务守护线程,当hive每提交一个MapReduce任务时,所述任务守护线程则会对所述MapReduce任务进行跟踪,从而在yarn中可以监测到所述MapReduce任务的运行状态;

[0092] 步骤102:若监测到所述任务的报错信息;获取所述报错信息对应的报错任务的日志;

[0093] 这里,对于hive任务,可以通过yarn相关的API(Application Programming Interface,应用程序编程接口),监测到任务的报错信息,以及获取到报错任务的日志;

[0094] 步骤103:对所述报错任务的日志进行文本分析并得到所述报错任务的报错原因信息;这里,通过对报错任务的日志进行文本分析,能够得到所述报错任务的报错原因信息,所述报错原因信息中包含了所述报错任务为什么会报错的关键信息;

[0095] 所述日志中遵循一定的格式,经过文本分析之后,可以提取出具体报错的数据的样本信息以及详细的报错信息,能够用于帮助分析人员更快的定位错误原因;

[0096] 步骤104:将所述报错原因信息与预存的报错原因分析表进行比对,得到与所述报错原因信息相对应的报错分析结果;所述报错原因分析表中包括预存的报错原因信息以及与所述报错原因信息相对应的报错分析结果,所述报错分析结构最终可以输出到Hadoop的控制台;

[0097] 可选的,所述报错原因分析表,是在程序内部维护的一个已知的哈希表,其中,键(key)代表经常报错的报错原因信息,值(value)代表已经定义好的与所述报错原因信息相对应的报错分析结果;然后,根据抓取到的报错原因信息跟这个哈希表进行比对,从而得出诊断的信息,即报错分析结果。

[0098] 从上述实施例可以看出,本发明实施例提供的所述任务信息分析方法,通过设置任务守护线程对任务进行监控并得到报错任务,对报错任务的日志进行文本分析并提取得到报错原因信息,将所述报错原因信息与预先设置的报错原因分析表进行比对,最终得到并输出报错分析结果,使得在出现任务报错时,无需技术人员进行人工排查,就能自动得出报错任务的错误原因(即报错分析结果),从而大大提高了技术人员的工作效率。

[0099] 本发明实施例的第二个方面,提出了一种能够自动分析任务的运行情况且无需手动操作的任务信息分析方法的第二个实施例。

[0100] 结合参照附图1和附图2,所述任务信息分析方法,应用于Hadoop,可包括以下步骤:

[0101] 步骤101:对提交的任务的运行状态进行监测;可选的,可通过启动任务守护线程的方式来进行监测,所述守护线程则用于对提交的的任务进行跟踪并获取所述任务的运行状态;

[0102] 步骤102:若监测到所述任务的报错信息;获取所述报错信息对应的报错任务的日志;

[0103] 步骤103:对所述报错任务的日志进行文本分析并得到所述报错任务的报错原因信息;这里,通过对报错任务的日志进行文本分析,能够得到所述报错任务的报错原因信息,所述报错原因信息中包含了所述报错任务为什么会报错的关键信息;

[0104] 步骤104:将所述报错原因信息与预存的报错原因分析表进行比对,得到与所述报错原因信息相对应的报错分析结果;所述报错原因分析表中包括预存的报错原因信息以及与所述报错原因信息相对应的报错分析结果,所述报错分析结构最终可以输出到Hadoop的

控制台；

[0105] 进一步的,在一些可选实施方式中,所述任务信息分析方法,所述对所述报错任务的日志进行文本分析并得到所述报错任务的报错原因信息的步骤103还可以具体包括以下步骤:

[0106] 步骤1031:在启用yarn的日志聚集的情况下,选取出报错任务;

[0107] 步骤1032:根据MapReduce任务的applicationid和container(容器)获取详细的所述报错任务的日志信息;容器中封装了机器资源,如内存,CPU,磁盘,网络等,每个任务都会被分配一个容器,该任务只能在该容器中执行,并使用该容器封装的资源;

[0108] 步骤1033:通过Hadoop HDFS API(Application Programming Interface,应用程序编程接口)来读取所述报错任务的日志信息;

[0109] 步骤1034:通过grep、awk和/或正则等文本分析算法,对所述报错任务的日志信息中包含的标准化错误进行抓取和分析;

[0110] 步骤1035:得到所述报错任务的报错原因信息;

[0111] 其中,grep(Globally search a Regular Expression and Print,全局搜索正则表达式和打印)是一种强大的文本搜索工具,它能使用正则表达式搜索文本,并把匹配的行打印出来;AWK(沃克)是一个优良的文本处理工具,Linux及Unix环境中现有的功能最强大的数据处理引擎之一;正则,在各种编程语言中,正则代表“正则表达式”,从最早的Perl (Practical Extraction and Report Language,实用报表提取语言)到后来的Java、.NET,都提供了这种文本分析的高级工具;正则表达式是对字符串操作的一种逻辑公式,就是用事先定义好的一些特定字符、及这些特定字符的组合,组成一个“规则字符串”,这个“规则字符串”用来表达对字符串的一种过滤逻辑。

[0112] 通过上述实施例,使得在Hadoop中,能够通过上述方法完成对所述报错任务的日志进行文本分析以及得到所述报错任务的报错原因信息。

[0113] 本发明实施例的第三个方面,提出了一种能够自动分析任务的运行情况且无需手动操作的任务信息分析方法的第三个实施例。如图3所示,为本发明提供的任务信息分析方法的第三个实施例的流程示意图。

[0114] 所述任务信息分析方法,包括以下步骤:

[0115] 步骤201:启动任务守护线程;

[0116] 举例来说,在Hadoop中,每一个hive任务都会被解析成MapReduce任务,每一个MapReduce任务都会有一个唯一的applicationid(应用ID);在hive启动过程中,则会启动所述任务守护线程,当hive每提交一个MapReduce任务时,所述任务守护线程则会对所述MapReduce任务进行跟踪,从而在yarn中可以监测到所述MapReduce任务的运行状态;

[0117] 可选的,所述任务信息分析方法,应用于MapReduce运算,所述任务包括map任务和reduce任务;

[0118] 一个hive任务,会被解析成若干个MapReduce job(工作)来执行,利用yarn相关的API,可以监听到正在运行的这个MapReduce job对应的任务的具体情况;比如,这个job启动了多少个map任务,多少个reduce任务,以及各个任务的状态(包括启动、正在运行、报错、失败);

[0119] MapReduce是一种编程模型,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)”,是它们的主要思想,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;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;

[0120] 在一些实施方式中,所述任务信息分析方法,可包括以下步骤:

[0121] 步骤202:创建map队列和reduce队列,分别用于存储map任务报错列表和reduce任务报错列表;

[0122] 步骤203:根据所述任务的运行状态,将报错的map任务和报错的reduce任务分别存储在所述map任务报错列表和reduce任务报错列表中;

[0123] 举例来说,在Hadoop中,每一个hive任务都会被解析成MapReduce任务,每一个MapReduce任务都会有一个唯一的applicationid(应用ID);在hive启动过程中,则会启动所述任务守护线程,当hive每提交一个MapReduce任务时,所述任务守护线程则会对所述MapReduce任务进行跟踪,从而在yarn中可以监测到所述MapReduce任务的运行状态;根据这个监测到的任务运行状态,能够知道报错的任务是map任务还是reduce任务,从而将报错的map任务和报错的reduce任务区别开来,因此,通过这种方法,可以将报错的map任务和报错的reduce任务进行分别存储;

[0124] 步骤204:在所述map任务报错列表中记录报错的map任务的ID和执行所述map任务的机器的列表;

[0125] 步骤205:在所述reduce任务报错列表中记录报错的reduce任务的ID和执行所述reduce任务的机器的列表;

[0126] 由于报错任务的数量较多,因此,还可具体包括以下步骤:

[0127] 步骤206:采用随机算法,从所述map任务报错列表和reduce任务报错列表中选取报错任务进行分析;

[0128] 步骤207:对所述报错任务的日志进行文本分析并得到所述报错任务的报错原因信息;

[0129] 步骤208:将所述报错原因信息与预存的报错原因分析表进行比对,得到与所述报错原因信息相对应的报错分析结果;

[0130] 这里,通过创建map队列用于存储map任务报错列表、报错的map任务的ID和执行所述map任务的机器的列表,创建reduce队列用于存储reduce任务报错列表、报错的reduce任务的ID和执行所述reduce任务的机器的列表,并采用随机算法选取报错任务进行分析,从而能够较为规范的管理报错任务,并能根据需要从中选取报错任务进行分析,使得报错任务处理结果更加准确,技术人员的工作量大大减少,工作效率大大提高;

[0131] 可选的,在另一些实施方式中,所述任务信息分析方法,还可包括以下步骤:

[0132] 步骤209:通过yarn监测所述reduce任务的处理进度;所述处理进度包括需要抓取文件数、已经抓取文件数和文件抓取速度;

[0133] 可选的,在所述reduce任务开始的时候,首先会从所有的map任务的输出的数据中拉取与所述map任务对应的reduce任务的数据;通过监测reduce任务的抓取数据的子任务,能够知道当前reduce任务的处理进度,即,共需要抓取多少个文件,已经抓取到了多少个文件,以及抓取文件的速度;其中,每隔第二预设时间阈值(例如1s),监测一下文件抓取速度;

[0134] 步骤210:在与所述reduce任务有关联的map任务结束后,启动从零开始的计时;

[0135] 步骤211:判断计时时长超过第一预设时间阈值时是否存在未结束文件抓取的reduce任务;这里,在与所述reduce任务有关联的map任务结束后,对于超过第一预设时间阈值(例如5分钟)还没有结束文件抓取的reduce任务,则说明这些reduce任务远远落后于其他reduce任务;

[0136] 步骤212:若计时时长超过第一预设时间阈值后,还存在未结束文件抓取的reduce任务,说明这些reduce任务远远落后于其他reduce任务,因此,进一步判断所述文件抓取速度是否低于预设文件抓取速度,用于分析这些reduce任务缓慢的原因;

[0137] 步骤213:若所述文件抓取速度低于预设文件抓取速度,判定当前网络状态不佳,并可发出网络状态不佳的提示信息,说明任务进度正在受到网络状态不佳的影响;

[0138] 步骤214:若所述文件抓取速度不低于预设文件抓取速度,判定当前网络状态正常;

[0139] 其中,所述第一预设时间阈值、第二预设时间阈值、预设文件抓取速度均可以根据实际需要进行设置,在此不进行数值限制;

[0140] 通过上述实施例,能够自动判断是否出现了因为网络状态不佳而引起的任务进度缓慢的问题,若出现了这种问题,还可以在判定当前网络状态不佳后发出网络状态不佳的提示信息,用于提醒相关人员及时处理;

[0141] 可选的,在另一些实施方式中,所述任务信息分析方法,还可包括以下步骤:

[0142] 通过hive提交的任務中,MapReduce任务的日志都会按照一定的格式进行输出,之前的步骤中,可以监测所有reduce任务的运行状态;

[0143] 步骤215:对于联接(join)的所述reduce任务,监测所述reduce任务的日志;

[0144] 步骤216:判断联接的所述reduce任务中,未完成的reduce任务占联接的全部所述reduce任务的比例是否低于预设未完成比例;

[0145] 步骤217:若未完成的reduce任务占联接的全部所述reduce任务的比例低于预设未完成比例,当reduce任务绝大部分任务已经结束,只剩余几个reduce任务还在运行时,说明可能出现了数据倾斜的情况;因此,进一步判断所述未完成的reduce任务的持续时间是否超过第三预设时间阈值(例如,5分钟);若这些未完成的reduce任务超过第三预设时间阈值还没有结束,则说明有极大可能性会出现数据倾斜的情况;

[0146] 步骤218:若所述未完成的reduce任务的持续时间超过第三预设时间阈值,则判断所述reduce任务的同一键(key)的数据量是否超出预设键数据量阈值;这里,可以通过监测reduce任务输出的日志来获取所述reduce任务的同一键(key)的数据量,通常,在reduce任务输出的日志中会有类似以下格式输出:

[0147] org.apache.hadoop.hive.q1.exec.CommonJoinOperator:table 0has 8000rows for join key[c80e779e25d4,2,20];

[0148] 步骤219:若所述reduce任务的同一键的数据量超出预设键数据量阈值(例如3万),判定所述reduce任务属于数据倾斜,并发出数据倾斜警告信息;一般情况下,当单个键(key)的记录数超过了3w,基本上就可以判定是发生了数据倾斜,这个时候可以把这些导致数据倾斜的key输出,使相关人员能够清楚的明白发生了数据倾斜,以及是由哪些key导致的数据倾斜;

[0149] 通过上述实施例,能够自动判断是否出现了数据倾斜的问题,若出现了这种问题,还可以在出现了数据倾斜后发出数据倾斜警告信息,用于提醒相关人员及时处理;

[0150] 有的时候,若AM所在的机器节点发生故障,将导致分配AM失败,此时通常需要重新分配AM,若重新分配AM成功,则任务可继续进行,而通常情况下,如果AM连续两次分配失败,则表示任务失败,此时,根据任务的失败信息中出现的请求分配AM失败的失败信息,即可进行相应的失败原因的判断;因此,在一些可选实施方式中,所述任务信息分析方法,应用于hive任务,还可包括以下步骤:

[0151] 步骤220:监测所述任务的失败信息;

[0152] 步骤221:根据所述任务的失败信息,判断所述任务是否连续请求分配AM失败达到预设失败次数;所述预设失败次数可根据需要进行设定,为了保证系统的正常运行,通常可设定为2次,当然,根据实际需要,也可以设置为3次或更多次;

[0153] 步骤222:若所述任务连续请求分配AM失败达到预设失败次数,则判定所述任务请求分配AM(ApplicationMaster)失败并输出AM的IP地址,供集群维护人员查询;

[0154] 通过上述实施例,对任务的失败信息进行监测,从而判断是否因请求分配AM失败而失败,若是,则输出AM的IP地址,供集群维护人员查询,从而方便了集群维护人员对失败原因的判断和对系统的维护。

[0155] 需要说明的是,上述步骤202~208、步骤209~214、步骤215~219、步骤220~222各自所实现的实施例互相之间可以是并列关系,也可以具有先后关系,具体搭配关系可以根据实际需要进行调整,显然,这些调整后得到的实施方式,也应当属于本发明的保护范围;

[0156] 从上述实施例可以看出,本发明实施例提供的所述任务信息分析方法,通过设置任务守护线程对任务进行监控,并能自动得到并输出分析结果,使得在任务运行过程中出现问题时,无需技术人员进行人工排查,就能自动得出问题原因,从而大大提高了技术人员的工作效率。

[0157] 此外,在一些可选实施方式中,所述任务信息分析方法,应用于hive任务;

[0158] 所述对提交的任务的运行状态进行监测的步骤101之后还可进一步包括以下步骤:

[0159] 监测本地运行的hive任务的报错,例如数据路径权限错误的权限问题、mapjoin(映射联接)的小表序列化、sql语法格式错误等等;与在分布式环境下进行的MapReduce任务监测不同,某些hive任务会在运行hive任务的客户端进行,比如说mapjoin的第一步操作;

[0160] 获取报错hive任务的日志;

[0161] 通过grep、awk和/或正则等文本分析算法,对所述报错hive任务的日志进行文本分析,并得到所述报错hive任务的hive任务报错原因信息;

[0162] 将所述hive任务报错原因信息与预存的hive任务报错原因分析表进行比对,得到与所述hive任务报错原因信息相对应的hive任务报错分析结果并输出到Hadoop的控制台;这里,所述hive任务报错原因分析表,也可采用哈希表,将一些基本的hive任务报错类型进行总结,键是经常报错的hive任务报错原因信息,值是已经定义好的诊断信息,即hive任务报错分析结果。

[0163] 通过上述实施例,能够自动得出并输出本地运行的hive任务报错分析结果,从而大大提高了技术人员的工作效率。

[0164] 基于上述任务运行过程中的各种错误原因,本发明实施例的第四个方面,提出了一种能够自动分析任务的运行情况且无需手动操作的任务信息分析装置的第一个实施例。如图4所示,为本发明提供的任务信息分析装置的第一个实施例的模块结构示意图。

[0165] 所述任务信息分析装置,包括:

[0166] 监测模块301,用于对提交的任务的运行状态进行监测;可选的,可通过启动任务守护线程的方式来进行监测,所述守护线程则用于对提交的任务进行跟踪并获取所述任务的运行状态;

[0167] 举例来说,在Hadoop中,每一个hive任务都会被解析成MapReduce任务,每一个MapReduce任务都会有一个唯一的applicationid(应用ID);在hive启动过程中,则会启动所述任务守护线程,当hive每提交一个MapReduce任务时,所述任务守护线程则会对所述MapReduce任务进行跟踪,从而在yarn中可以监测到所述MapReduce任务的运行状态;

[0168] 日志获取模块302,用于若监测到所述任务的报错信息,获取所述报错信息对应的报错任务的日志;

[0169] 这里,对于hive任务,可以通过yarn相关的API(Application Programming Interface,应用程序编程接口),监测到任务的报错信息,以及获取到报错任务的日志;

[0170] 日志解析模块303,用于对所述报错任务的日志进行文本分析并得到所述报错任务的报错原因信息;这里,通过对报错任务的日志进行文本分析,能够得到所述报错任务的报错原因信息,所述报错原因信息中包含了所述报错任务为什么会报错的关键信息;

[0171] 所述日志中遵循一定的格式,经过文本分析之后,可以提取出具体报错的数据的样本信息以及详细的报错信息,能够用于帮助分析人员更快的定位错误原因;

[0172] 分析比对模块304,用于将所述报错原因信息与预存的报错原因分析表进行比对,得到与所述报错原因信息相对应的报错分析结果;所述报错原因分析表中包括预存的报错原因信息以及与所述报错原因信息相对应的报错分析结果,所述报错分析结构最终可以输出到Hadoop的控制台;

[0173] 可选的,所述报错原因分析表,是在程序内部维护的一个已知的哈希表,其中,键(key)代表经常报错的报错原因信息,值(value)代表已经定义好的与所述报错原因信息相对应的报错分析结果;然后,根据抓取到的报错原因信息跟这个哈希表进行比对,从而得出诊断的信息,即报错分析结果。

[0174] 从上述实施例可以看出,本发明实施例提供的所述任务信息分析装置,通过设置任务守护线程对任务进行监控并得到报错任务,对报错任务的日志进行文本分析并提取得到报错原因信息,将所述报错原因信息与预先设置的报错原因分析表进行比对,最终得到并输出报错分析结果,使得在出现任务报错时,无需技术人员进行人工排查,就能自动得出报错任务的错误原因(即报错分析结果),从而大大提高了技术人员的工作效率。

[0175] 基于上述任务运行过程中的各种错误原因,本发明实施例的第五个方面,提出了一种能够自动分析任务的运行情况且无需手动操作的任务信息分析装置的第二个实施例。如图5所示,为本发明提供的任务信息分析装置的第二个实施例的模块结构示意图。

[0176] 所述任务信息分析装置,包括:

[0177] 监测模块301,用于对提交的任务的运行状态进行监测;可选的,可通过启动任务守护线程的方式来进行监测,所述守护线程则用于对提交的任务进行跟踪并获取所述任务的运行状态;

[0178] 举例来说,在Hadoop中,每一个hive任务都会被解析成MapReduce任务,每一个MapReduce任务都会有一个唯一的applicationid(应用ID);在hive启动过程中,则会启动所述任务守护线程,当hive每提交一个MapReduce任务时,所述任务守护线程则会对所述MapReduce任务进行跟踪,从而在yarn中可以监测到所述MapReduce任务的运行状态;

[0179] 日志获取模块302,用于若监测到所述任务的报错信息,获取所述报错信息对应的报错任务的日志;

[0180] 这里,对于hive任务,可以通过yarn相关的API(Application Programming Interface,应用程序编程接口),监测到任务的报错信息,以及获取到报错任务的日志;

[0181] 日志解析模块303,用于对所述报错任务的日志进行文本分析并得到所述报错任务的报错原因信息;这里,通过对报错任务的日志进行文本分析,能够得到所述报错任务的报错原因信息,所述报错原因信息中包含了所述报错任务为什么会报错的关键信息;

[0182] 所述日志中遵循一定的格式,经过文本分析之后,可以提取出具体报错的数据的样本信息以及详细的报错信息,能够用于帮助分析人员更快的定位错误原因;

[0183] 分析比对模块304,用于将所述报错原因信息与预存的报错原因分析表进行比对,得到与所述报错原因信息相对应的报错分析结果;所述报错原因分析表中包括预存的报错原因信息以及与所述报错原因信息相对应的报错分析结果,所述报错分析结构最终可以输出到Hadoop的控制台;

[0184] 进一步的,在一些可选实施方式中,所述任务信息分析方法,应用于Hadoop,所述日志解析模块303,还可以具体用于:

[0185] 在启用yarn的日志聚集的情况下,选取出报错任务;

[0186] 根据MapReduce任务的applicationid和container(容器)获取详细的所述报错任务的日志信息;容器中封装了机器资源,如内存,CPU,磁盘,网络等,每个任务都会被分配一个容器,该任务只能在该容器中执行,并使用该容器封装的资源;

[0187] 通过Hadoop HDFS API来读取所述报错任务的日志信息;

[0188] 通过grep、awk和/或正则等文本分析算法,对所述报错任务的日志信息中包含的标准化错误进行抓取和分析;

[0189] 得到所述报错任务的报错原因信息;

[0190] 其中,grep(Globally search a Regular Expression and Print,全局搜索正则表达式和打印)是一种强大的文本搜索工具,它能使用正则表达式搜索文本,并把匹配的行打印出来;AWK(沃克)是一个优良的文本处理工具,Linux及Unix环境中现有的功能最强大的数据处理引擎之一;正则,在各种编程语言中,正则代表“正则表达式”,从最早的Perl (Practical Extraction and Report Language,实用报表提取语言)到后来的Java、.NET,都提供了这种文本分析的高级工具;正则表达式是对字符串操作的一种逻辑公式,就是用事先定义好的一些特定字符、及这些特定字符的组合,组成一个“规则字符串”,这个“规则字符串”用来表达对字符串的一种过滤逻辑。

[0191] 通过上述实施例,使得在Hadoop中,能够通过上述方法完成对所述报错任务的日

志进行文本分析以及得到所述报错任务的报错原因信息。

[0192] 可选的,在一些实施方式中,所述任务信息分析装置,应用于MapReduce运算,所述任务包括map任务和reduce任务;

[0193] 一个hive任务,会被解析成若干个MapReduce job(工作)来执行,利用yarn相关的API,可以监听到正在运行的这个MapReduce job对应的任务的具体的情况;比如,这个job启动了多少个map任务,多少个reduce任务,以及各个任务的状态(包括启动、正在运行、报错、失败);

[0194] MapReduce是一种编程模型,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)”,是它们的主要思想,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;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;

[0195] 所述任务信息分析装置还包括报错列表创建模块305,具体用于:

[0196] 创建map队列和reduce队列,分别用于存储map任务报错列表和reduce任务报错列表;

[0197] 根据所述任务的运行状态,将报错的map任务和报错的reduce任务分别存储在所述map任务报错列表和reduce任务报错列表中;

[0198] 举例来说,在Hadoop中,每一个hive任务都会被解析成MapReduce任务,每一个MapReduce任务都会有一个唯一的applicationid(应用ID);在hive启动过程中,则会启动所述任务守护线程,当hive每提交一个MapReduce任务时,所述任务守护线程则会对所述MapReduce任务进行跟踪,从而在yarn中可以监测到所述MapReduce任务的运行状态;根据这个监测到的任务运行状态,能够知道报错的任务是map任务还是reduce任务,从而将报错的map任务和报错的reduce任务区别开来,因此,通过这种方法,可以将报错的map任务和报错的reduce任务进行分别存储;

[0199] 在所述map任务报错列表中记录报错的map任务的ID和执行所述map任务的机器的列表;

[0200] 在所述reduce任务报错列表中记录报错的reduce任务的ID和执行所述reduce任务的机器的列表;

[0201] 由于报错任务的数量较多,因此,所述日志获取模块302,具体用于:

[0202] 采用随机算法,从所述map任务报错列表和reduce任务报错列表中选取报错任务进行分析。

[0203] 通过创建map队列用于存储map任务报错列表、报错的map任务的ID和执行所述map任务的机器的列表,创建reduce队列用于存储reduce任务报错列表、报错的reduce任务的ID和执行所述reduce任务的机器的列表,并采用随机算法选取报错任务进行分析,从而能够较为规范的管理报错任务,并能根据需要从中选取报错任务进行分析,使得报错任务处理结果更加准确,技术人员的工作量大大减少,工作效率大大提高。

[0204] 可选的,在另一些实施方式中,所述任务信息分析装置,应用于MapReduce运算,所述任务包括map任务和reduce任务;

[0205] 所述监测模块301,还用于通过yarn监测所述reduce任务的处理进度;所述处理进

度包括需要抓取文件数、已经抓取文件数和文件抓取速度；

[0206] 可选的,在所述reduce任务开始的时候,首先会从所有的map任务的输出的数据中拉取与所述map任务对应的reduce任务的数据;通过监测reduce任务的抓取数据的子任务,能够知道当前reduce任务的处理进度,即,共需要抓取多少个文件,已经抓取到了多少个文件,以及抓取文件的速度;其中,每隔第二预设时间阈值(例如1s),监测一下文件抓取速度;

[0207] 所述任务信息分析装置还包括:

[0208] 计时模块306,用于在与所述reduce任务有关联的map任务结束后,启动从零开始的计时;

[0209] 未结束任务判断模块307,用于判断计时时长超过第一预设时间阈值时是否存在未结束文件抓取的reduce任务;这里,在与所述reduce任务有关联的map任务结束后,对于超过第一预设时间阈值(例如5分钟)还没有结束文件抓取的reduce任务,则说明这些reduce任务远远落后于其他reduce任务;

[0210] 抓取速度判断模块308,若存在未结束文件抓取的reduce任务(说明这些reduce任务远远落后于其他reduce任务),用于判断所述文件抓取速度是否低于预设文件抓取速度;

[0211] 网络状态判断模块309,若所述文件抓取速度低于预设文件抓取速度,用于判定当前网络状态不佳,并可发出网络状态不佳的提示信息,说明任务进度正在受到网络状态不佳的影响;

[0212] 其中所述第一预设时间阈值、第二预设时间阈值、预设文件抓取速度均可以根据实际需要进行设置,在此不进行数值限制。

[0213] 通过上述实施例,能够自动判断是否出现了因为网络状态不佳而引起的任务进度缓慢的问题,若出现了这种问题,还可以在判定当前网络状态不佳后发出网络状态不佳的提示信息,用于提醒相关人员及时处理。

[0214] 可选的,在另一些实施方式中,所述任务信息分析装置,应用于MapReduce运算,所述任务包括map任务和reduce任务;

[0215] 所述监测模块301,还用于对于联接的所述reduce任务,监测所述reduce任务的日志;

[0216] 通过hive提交的任务中,MapReduce任务的日志都会按照一定的格式进行输出,之前的步骤中,可以监测所有reduce任务的运行状态;

[0217] 所述任务信息分析装置还包括:

[0218] 未完成比例判断模块310,用于判断联接(join)的所述reduce任务中,未完成的reduce任务占联接的全部所述reduce任务的比例是否低于预设未完成比例;

[0219] 持续时间判断模块311,若未完成的reduce任务占联接的全部所述reduce任务的比例低于预设未完成比例,当reduce任务绝大部分任务已经结束,只剩余几个reduce任务还在运行时,说明可能出现了数据倾斜的情况;因此,用于判断所述未完成的reduce任务的持续时间是否超过第三预设时间阈值(例如,5分钟);若这些未完成的reduce任务超过第三预设时间阈值还没有结束,则说明有极大可能性会出现数据倾斜的情况;

[0220] 数据量判断模块312,若所述未完成的reduce任务的持续时间超过第三预设时间阈值,用于判断所述reduce任务的同一键(key)的数据量是否超出预设键数据量阈值;这里,可以通过监测reduce任务输出的日志来获取所述reduce任务的同一键(key)的数据量,

通常,在reduce任务输出的日志中会有类似以下格式输出:

[0221] org.apache.hadoop.hive ql.exec.CommonJoinOperator:table 0has 8000rows for join key[c80e779e25d4,2,20];

[0222] 数据倾斜判断模块313,若所述reduce任务的同一键的数据量超出预设键数据量阈值,用于判定所述reduce任务属于数据倾斜;

[0223] 警告模块314,用于发出数据倾斜警告信息;一般情况下,当单个键(key)的记录数超过了3w,基本上就可以判定是发生了数据倾斜,这个时候可以把这些导致数据倾斜的key输出,使相关人员能够清楚的明白发生了数据倾斜,以及是由哪些key导致的数据倾斜。

[0224] 通过上述实施例,能够自动判断是否出现了数据倾斜的问题,若出现了这种问题,还可以在出现了数据倾斜后发出数据倾斜警告信息,用于提醒相关人员及时处理。

[0225] 有的时候,若AM所在的机器节点发生故障,将导致分配AM失败,此时通常需要重新分配AM,若重新分配AM成功,则任务可继续进行,而通常情况下,如果AM连续分配失败达到预设失败次数,则表示任务失败,此时,根据任务的失败信息中出现的请求分配AM失败的失败信息,即可进行相应的失败原因的判断;因此,在一些可选实施方式中,所述任务信息分析装置,应用于hive任务;所述监测模块301,还用于监测所述任务的失败信息;

[0226] 所述任务信息分析装置还包括:

[0227] AM请求判断模块315,用于根据所述任务的失败信息,判断所述任务是否连续请求分配AM(ApplicationMaster)失败达到预设失败次数;所述预设失败次数可根据需要进行设定,为了保证系统的正常运行,通常可设定为2次,当然,根据实际需要,也可以设置为3次或更多次;

[0228] 请求失败判断模块316,若所述任务连续请求分配AM失败达到预设失败次数,用于判定所述任务请求分配AM失败并输出AM的IP地址,供集群维护人员查询。

[0229] 通过上述实施例,对任务的失败信息进行监测,从而判断是否因请求分配AM失败而失败,若是,则输出AM的IP地址,供集群维护人员查询,从而方便了集群维护人员对失败原因的判断和对系统的维护。

[0230] 此外,在一些可选实施方式中,所述任务信息分析方法,应用于hive任务;

[0231] 所述监测模块301,还用于监测本地运行的hive任务的报错,例如数据路径权限错误的权限问题、mapjoin(映射联接)的小表序列化、sql语法格式错误等等;与在分布式环境下进行的MapReduce任务监测不同,某些hive任务会在运行hive任务的客户端进行,比如说mapjoin的第一步操作;

[0232] 所述日志获取模块302,还用于获取报错hive任务的日志;

[0233] 所述日志解析模块303,还用于通过grep、awk和/或正则等文本分析算法,对所述报错hive任务的日志进行文本分析,并得到所述报错hive任务的hive任务报错原因信息;

[0234] 所述分析比对模块304,用于将所述hive任务报错原因信息与预存的hive任务报错原因分析表进行比对,得到与所述hive任务报错原因信息相对应的hive任务报错分析结果并输出到Hadoop的控制台;这里,所述hive任务报错原因分析表,也可采用哈希表,将一些基本的hive任务报错类型进行总结,键是经常报错的hive任务报错原因信息,值是已经定义好的诊断信息,即hive任务报错分析结果。

[0235] 通过上述实施例,能够自动得出并输出本地运行的hive任务报错分析结果,从而

大大提高了技术人员的工作效率。

[0236] 所属领域的普通技术人员应当理解:以上任何实施例的讨论仅为示例性的,并非旨在暗示本公开的范围(包括权利要求)被限于这些例子;在本发明的思路下,以上实施例或者不同实施例中的技术特征之间也可以进行组合,步骤可以以任意顺序实现,并存在如上所述的本发明的不同方面的许多其它变化,为了简明它们没有在细节中提供。

[0237] 另外,为简化说明和讨论,并且为了不会使本发明难以理解,在所提供的附图中可以示出或不示出与集成电路(IC)芯片和其它部件的公知的电源/接地连接。此外,可以以框图的形式示出装置,以便避免使本发明难以理解,并且这也考虑了以下事实,即关于这些框图装置的实施方式的细节是高度取决于将要实施本发明的平台的(即,这些细节应当完全处于本领域技术人员的理解范围内)。在阐述了具体细节(例如,电路)以描述本发明的示例性实施例的情况下,对本领域技术人员来说显而易见的是,可以在没有这些具体细节的情况下或者这些具体细节有变化的情况下实施本发明。因此,这些描述应被认为是说明性的而不是限制性的。

[0238] 尽管已经结合了本发明的具体实施例对本发明进行了描述,但是根据前面的描述,这些实施例的很多替换、修改和变型对本领域普通技术人员来说将是显而易见的。例如,其它存储器架构(例如,动态RAM(DRAM))可以使用所讨论的实施例。

[0239] 本发明的实施例旨在涵盖落入所附权利要求的宽泛范围之内的所有这样的替换、修改和变型。因此,凡在本发明的精神和原则之内,所做的任何省略、修改、等同替换、改进等,均应包含在本发明的保护范围之内。

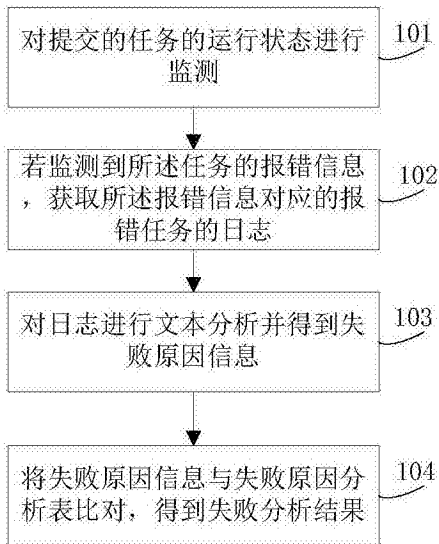


图1

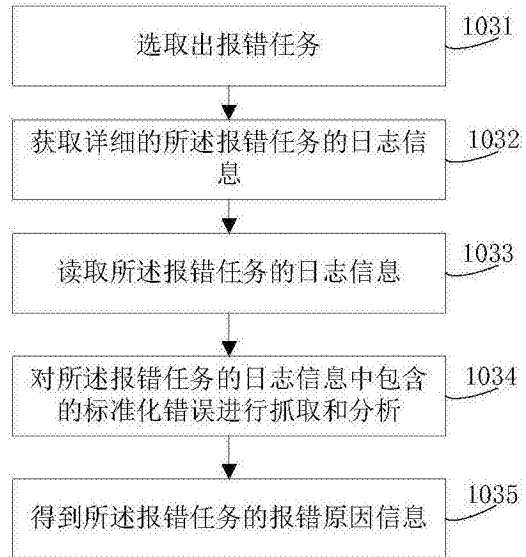


图2

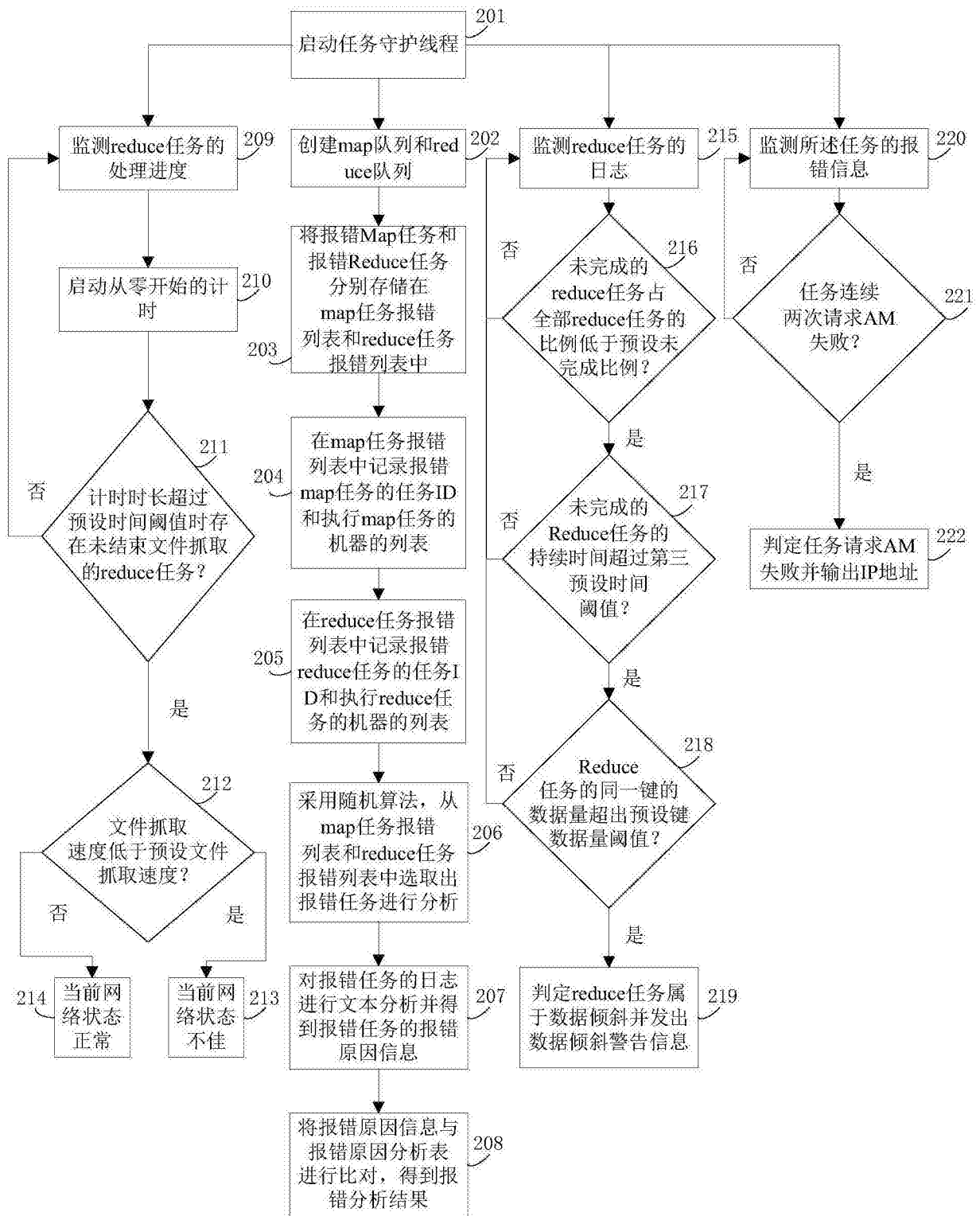


图3

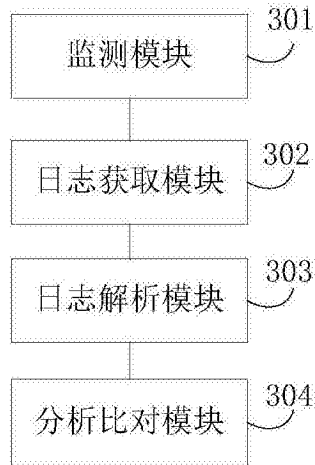


图4

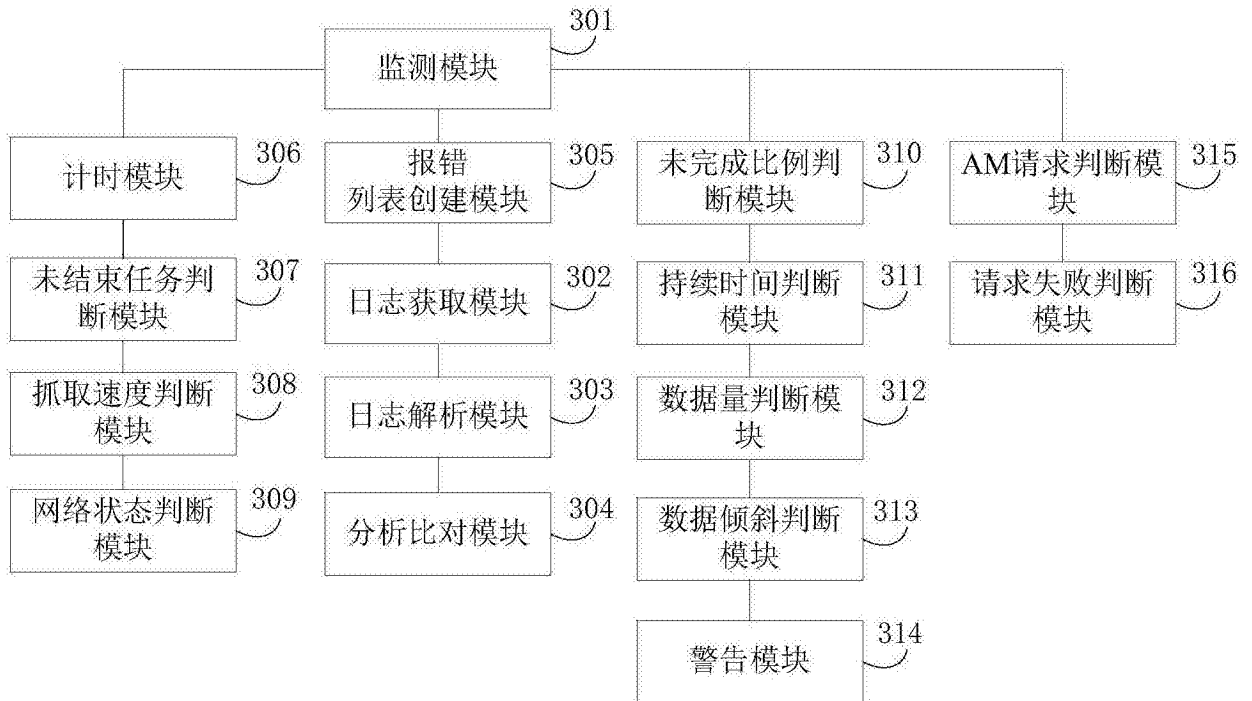


图5