



(12)发明专利

(10)授权公告号 CN 104050042 B

(45)授权公告日 2017.06.13

(21)申请号 201410240314.9

(22)申请日 2014.05.30

(65)同一申请的已公布的文献号

申请公布号 CN 104050042 A

(43)申请公布日 2014.09.17

(73)专利权人 北京先进数通信息技术股份公司

地址 100089 北京市海淀区车道沟1号滨河大厦D座六层

(72)发明人 完献忠 唐遂

(74)专利代理机构 北京润泽恒知识产权代理有

限公司 11319

代理人 苏培华

(51)Int.Cl.

G06F 9/50(2006.01)

G06F 9/445(2006.01)

(56)对比文件

CN 101819540 A,2010.09.01,

CN 101788920 A,2010.07.28,

CN 102096602 A,2011.06.15,

CN 102236580 A,2011.11.09,

CN 102609295 A,2012.07.25,

CN 102739771 A,2012.10.17,

US 2011047084 A1,2011.02.24,

US 2008103861 A1,2008.05.01,

审查员 邵娜娜

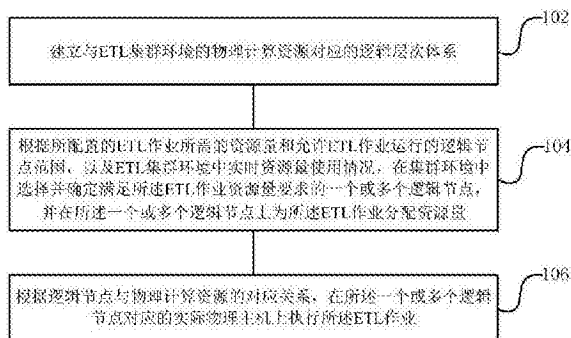
权利要求书3页 说明书17页 附图13页

(54)发明名称

ETL作业的资源分配方法及装置

(57)摘要

本申请提供了一种ETL作业的资源分配方法及装置,该方法包括:建立与ETL集群环境的物理计算资源对应的逻辑层次体系;根据所配置的ETL作业所需的资源量和允许ETL作业运行的逻辑节点范围,以及ETL集群环境中实时资源量使用情况,在集群环境中选择并确定满足所述ETL作业资源量要求的一个或多个逻辑节点,并在所述一个或多个逻辑节点上为所述ETL作业分配资源量;根据逻辑节点与物理计算资源的对应关系,在所述一个或多个逻辑节点对应的实际物理主机上执行所述ETL作业。本申请提高了资源分配的效率。



1. 一种抽取、转换和加载ETL作业的资源分配方法,其特征在于,包括:

建立与ETL集群环境的物理计算资源对应的逻辑层次体系,其中,所述逻辑层次体系包括:直接与硬件计算资源对应的物理节点,以及在物理节点上创建的、对应物理节点部分或全部计算资源的逻辑节点,所述逻辑节点包括:位于特定物理节点的用于执行ETL作业的逻辑节点ETL服务器ETL-SERVER节点,从属于ETL-SERVER节点的位于特定物理节点之上的逻辑节点ETL客户端ETL-CLIENT节点,以及包含一个或多个ETL-SERVER节点和ETL-CLIENT节点的ETL域,其中,所述物理节点、所述ETL-SERVER节点、和所述ETL-CLIENT节点所提供的计算资源采用资源量进行量化;

根据所配置的ETL作业所需的资源量和允许ETL作业运行的逻辑节点范围,以及ETL集群环境中实时资源量使用情况,在集群环境中选择并确定满足所述ETL作业资源量要求的一个或多个逻辑节点,并在所述一个或多个逻辑节点上为所述ETL作业分配资源量;

根据逻辑节点与物理计算资源的对应关系,在所述一个或多个逻辑节点对应的实际物理主机上执行所述ETL作业。

2. 如权利要求1所述的方法,其特征在于,根据所配置的ETL作业所需的资源量和允许ETL作业运行的逻辑节点范围,以及ETL集群环境中实时资源量使用情况,在集群环境中确定满足所述ETL作业资源量要求的一个或多个逻辑节点,包括:

从ETL作业的配置信息中获取可运行所述ETL作业的ETL域清单、以及所述ETL作业运行限定的ETL-SERVER节点清单;

在所述ETL作业运行限定的ETL-SERVER节点及所述ETL-SERVER节点所属的ETL客户端节点中,确定所有满足所述ETL作业运行资源量要求的ETL服务器和ETL客户端节点;

从满足所述ETL作业资源量要求的ETL服务器中选择剩余资源量最多的ETL服务器,作为执行所述ETL作业的逻辑节点;或,当所述ETL作业为数据阶段DataStage并行作业时,从所述剩余资源量最多的ETL服务器ETL服务器所属的且满足ETL作业资源量要求的ETL客户端中选择占比最低的N个ETL客户端,作为DataStage并行作业的执行节点,其中,N即DataStage并行作业的并发度,N为大于或等于1的整数。

3. 如权利要求2所述的方法,其特征在于,确定所有满足所述ETL作业运行资源量要求的ETL服务器和ETL客户端节点,包括:

当所述ETL作业为非DataStage并行作业时,确定剩余资源量大于或等于所述ETL作业所需资源量的ETL服务器为满足所述ETL作业运行资源量要求的ETL服务器;

当所述ETL作业为DataStage并行作业时,判断第一ETL服务器的剩余资源量是否大于或等于DataStage并行作业主节点所需资源量,其中,DataStage并行作业主节点所需资源量是一个预设值;如果是,则判断从属于所述第一ETL服务器所属的ETL客户端中,满足ETL作业运行的资源量要求的ETL客户端是否大于或等于N个,其中,N为DataStage并行作业的并发度,N为大于或等于1的整数,如果是,则确定所述第一ETL服务器以及从属于所述第一ETL服务器的满足资源量要求的ETL客户端为满足所述ETL作业运行资源量要求的ETL服务器和ETL客户端节点,否则,确定所述第一ETL服务器及其所属的所有ETL客户端均不满足ETL作业运行资源量要求;

重复上述步骤,直到对所有ETL作业运行限定ETL服务器清单中的ETL服务器以及其所属的ETL客户端完成判断。

4. 如权利要求3所述的方法,其特征在于,通过以下方法判断从属于所述第一ETL服务器的满足资源量要求的ETL客户端是否满足所述资源量要求:

对于从属于所述第一ETL服务器的一个或多个ETL客户端,当且仅当ETL客户端的剩余资源量大于或等于所述ETL作业需要的资源量的N分之一,判定该ETL客户端满足ETL作业运行的资源量要求。

5. 如权利要求1至4任一项所述的方法,其特征在于,在根据所配置的ETL作业所需的资源量和允许ETL作业运行的逻辑节点范围,以及ETL集群环境中实时资源量使用情况,在集群环境中选择并确定满足所述ETL作业资源量要求的一个或多个逻辑节点,并在所述一个或多个逻辑节点上为所述ETL作业分配资源量之前,所述方法还包括:

根据所述ETL作业的复杂度,确定所述ETL作业所需的资源量。

6. 一种抽取、转换和加载ETL作业的资源分配装置,其特征在于,包括:

配置模块,用于建立与ETL集群环境的物理计算资源对应的逻辑层次体系,并对集群环境所能提供计算资源,使用资源量的概念进行量化,其中,所述逻辑层次体系包括:直接与硬件计算资源对应的物理节点,以及在物理节点上创建的、对应物理节点部分或全部计算资源的逻辑节点,所述逻辑节点包括:位于特定物理节点的用于执行ETL作业的逻辑节点ETL服务器ETL-SERVER节点,从属于ETL-SERVER节点的位于特定物理节点之上的逻辑节点ETL客户端ETL-CLIENT节点,以及包含一个或多个ETL-SERVER节点和ETL-CLIENT节点的ETL域,其中,所述物理节点、所述ETL-SERVER节点、和所述ETL-CLIENT节点均采用资源量来对节点的计算资源进行量化;

计算资源分配模块,用于根据所配置的ETL作业所需的资源量和允许ETL作业运行的逻辑节点范围,以及ETL集群环境中实时资源量使用情况,在集群环境中选择并确定满足所述ETL作业资源量要求的一个或多个逻辑节点,并在所述一个或多个逻辑节点上为所述ETL作业分配资源量;

ETL作业执行模块,用于根据逻辑节点与物理计算资源的对应关系,在所述一个或多个逻辑节点对应的实际物理主机上执行所述ETL作业。

7. 如权利要求6所述的装置,其特征在于,所述计算资源分配模块用于:

从所述ETL作业的配置信息中获取可运行所述ETL作业的ETL域清单、以及所述ETL作业运行限定的ETL-SERVER节点清单;

在所述ETL作业运行限定的ETL-SERVER节点及所述ETL-SERVER节点所属的ETL客户端节点中,确定所有满足所述ETL作业运行资源量要求的ETL服务器和ETL客户端节点;

从满足所述ETL作业资源量要求的ETL服务器中选择剩余资源量最多的ETL服务器,作为执行所述ETL作业的逻辑节点;或,当所述ETL作业为数据阶段DataStage并行作业时,从所述剩余资源量最多的ETL服务器ETL服务器所属的且满足ETL作业资源量要求的ETL客户端中选择占比最低的N个ETL客户端,作为DataStage并行作业的执行节点,其中,N即DataStage并行作业的并发度,N为大于或等于1的整数。

8. 如权利要求7所述的装置,其特征在于,所述计算资源分配模块用于:

当所述ETL作业为非DataStage并行作业时,确定剩余资源量大于或等于所述ETL作业所需资源量的ETL服务器为满足所述ETL作业运行资源量要求的ETL服务器;

当所述ETL作业为DataStage并行作业时,判断第一ETL服务器的剩余资源量是否大于

或等于DataStage并行作业主节点所需资源量,其中,DataStage并行作业主节点所需资源量是一个预设值;如果是,则判断从属于所述第一ETL服务器所属的ETL客户端中,满足ETL作业运行的资源量要求的ETL客户端是否大于或等于N个,其中,N为DataStage并行作业的并发度,N为大于等于1的整数,如果是,则确定所述第一ETL服务器以及从属于所述第一ETL服务器的满足资源量要求的ETL客户端为满足所述ETL作业运行资源量要求的ETL服务器和ETL客户端节点,否则,确定所述第一ETL服务器及其所属的所有ETL客户端均不满足ETL作业运行资源量要求;

重复上述步骤,直到对所有ETL作业运行限定ETL服务器清单中的ETL服务器以及其所属的ETL客户端完成判断。

9.如权利要求8所述的装置,其特征在于,所述计算资源分配模块用于通过以下方式判断从属于所述第一ETL服务器的满足资源量要求的ETL客户端是否满足所述资源量要求:

对于从属于所述第一ETL服务器的一个或多个ETL客户端,当且仅当ETL客户端的剩余资源量大于或等于所述ETL作业需要的资源量的N分之一,判定该ETL客户端满足ETL作业运行的资源量要求。

10.如权利要求6至9任一项所述的装置,其特征在于,所述配置模块用于:

在根据所配置的ETL作业所需的资源量和ETL作业所能运行的逻辑节点,以及ETL集群环境中实时资源量使用情况,确定运行所述ETL作业的所需的一个或多个逻辑节点之前,根据所述ETL作业的复杂度,确定所述ETL作业所需的资源量。

ETL作业的资源分配方法及装置

技术领域

[0001] 本申请涉及平台技术领域,特别是涉及一种ETL作业的资源分配方法及装置。

背景技术

[0002] 以DataStage为基础构建的抽取、转换和加载(Extraction-Transformation-Loading,ETL)集群环境中,通常会包括DataStage服务器作业、DataStage并行作业及可执行程序多种类型的ETL作业,这几种类型的ETL作业的底层架构有很大差异。DataStage服务器作业及可执行程序仅运行于单台服务器上,而DataStage并行作业由DataStage并行引擎运行。

[0003] 并行作业的并行度是可以通过控制执行节点的数目来设定的。具体来说,需要设定一个并行引擎配置文件,并且可以通过并行作业预设的名为\$APT_CONFIG_FILE的环境变量参数,为特定的并行作业指定各自不同的配置文件。通过并行引擎配置文件,可以有效控制作业的并发度及计算资源的使用。

[0004] DataStage虽然提供了并行引擎配置文件,这个“接口”可以对作业的并行度及资源分配进行控制;但是,ETL平台上通常会有数量众多的并行作业运行,每次作业执行前都需要人工介入调整并行引擎配置文件的内容,管理较为繁琐,资源分配的效率较低。

[0005] 另外,并行引擎配置文件只针对并行作业,对于服务器作业或可执行程序ETL作业则没有效果,所以单靠并行引擎配置文件,并不能达到统一管理ETL平台计算资源的效果。

发明内容

[0006] 本申请所要解决的技术问题是提供一种ETL作业的资源分配方法及装置,能够提高对ETL作业的资源分配效率。

[0007] 为了解决上述问题,本申请公开了一种基于资源量的对抽取、转换和加载ETL作业进行并发调度的方法,包括:建立与ETL集群环境的物理计算资源对应的逻辑层次体系,其中,所述逻辑层次体系包括:直接与硬件计算资源对应的物理节点,以及在物理节点上创建的、对应物理节点部分或全部计算资源的逻辑节点,所述逻辑节点包括:位于特定物理节点的用于执行ETL作业的逻辑节点ETL服务器ETL-SERVER节点,从属于ETL-SERVER节点的位于特定物理节点之上的逻辑节点ETL客户端ETL-CLIENT节点,以及包含一个或多个ETL-SERVER节点和ETL-CLIENT节点的ETL域,其中,所述物理节点、所述ETL-SERVER节点、和所述ETL-CLIENT节点所提供的计算资源采用资源量进行量化;根据所配置的ETL作业所需的资源量和允许ETL作业运行的逻辑节点范围,以及ETL集群环境中实时资源量使用情况,在集群环境中选择并确定满足所述ETL作业资源量要求的一个或多个逻辑节点,并在所述一个或多个逻辑节点上为所述ETL作业分配资源量;根据逻辑节点与物理计算资源的对应关系,在所述一个或多个逻辑节点对应的实际物理主机上执行所述ETL作业。

[0008] 为了解决上述问题,本申请公开了一种基于资源量的ETL作业并行调度,包括:配置模块,用于建立与ETL集群环境的物理计算资源对应的逻辑层次体系,并对集群环境所能

提供计算资源,使用资源量的概念进行量化,其中,所述逻辑层次体系包括:直接与硬件计算资源对应的物理节点,以及在物理节点上创建的、对应物理节点部分或全部计算资源的逻辑节点,所述逻辑节点包括:位于特定物理节点的用于执行ETL作业的逻辑节点ETL服务器ETL-SERVER节点,从属于ETL-SERVER节点的位于特定物理节点之上的逻辑节点ETL客户端ETL-CLIENT节点,以及包含一个或多个ETL-SERVER节点和ETL-CLIENT节点的ETL域,其中,所述物理节点、所述ETL-SERVER节点、和所述ETL-CLIENT节点均采用资源量来对节点的计算资源进行量化;计算资源分配模块,用于根据所配置的ETL作业所需的资源量和允许ETL作业运行的逻辑节点范围,以及ETL集群环境中实时资源量使用情况,在集群环境中选择并确定满足所述ETL作业资源量要求的一个或多个逻辑节点,并在所述一个或多个逻辑节点上为所述ETL作业分配资源量;ETL作业执行模块,用于根据逻辑节点与物理计算资源的对应关系,在所述一个或多个逻辑节点对应的实际物理主机上执行所述ETL作业。

[0009] 与现有技术相比,本申请具有以下优点:

[0010] 通过在硬件和ETL作业之间设置逻辑节点,根据ETL作业需要的资源量,确定ETL作业对应的逻辑节点,从而分配相应的硬件给ETL作业,避免了每次作业执行前都需要人工介入调整并行引擎配置文件的内容,提高了资源分配的效率。

附图说明

[0011] 图1是根据本发明实施例的ETL作业的资源分配方法的流程图;

[0012] 图2是根据本发明实施例的MOIA典型系统拓扑结构示意图;

[0013] 图3是根据本发明实施例的MOIA逻辑集群示意图;

[0014] 图4是本发明实施例中,与平台环境资源量及资源分配相关的配置表和状态表的后台数据表的E-R(实体-关系)示意图,示意图中仅保留了和资源分配相关的字段;

[0015] 图5是本发明实施例中,作业预计算用资源量及资源分配相关的配置表和状态表的后台数据表的E-R(实体-关系)示意图,示意图中仅保留了和资源分配相关的字段;

[0016] 图6是根据本发明实施例的资源分配流程图;

[0017] 图6A是图6中604处理步骤的详细说明;

[0018] 图7是根据本发明实施例的作业执行服务的处理流程图;

[0019] 图7A是根据本发明实施例的ETL作业执行程序的处理过程的示意图;

[0020] 图8是根据本发明实施例的新建物理节点ETL_node1的示意图;

[0021] 图9是根据本发明实施例的设定物理节点资源的示意图;

[0022] 图10是根据本发明实施例的新建物理节点ETL_node2和ETL_node3的示意图;

[0023] 图11是根据本发明实施例的新建ETL域的示意图;

[0024] 图12是根据本发明实施例的新建ETL_Server1的示意图;

[0025] 图13是根据本发明实施例建立的逻辑节点结构图;

[0026] 图14是根据本发明实施例的建立备用的ETL域的示意图;

[0027] 图15是根据本发明实施例的测试作业的示意图;

[0028] 图16是根据本发明实施例的设置并行作业的示意图;

[0029] 图17是根据本发明实施例的设置动态调度属性的示意图;

[0030] 图18至图20是根据本发明实施例的新建另一个作业的示意图;

- [0031] 图21是根据本发明实施例的一种ETL作业的资源分配装置的结构框图；
- [0032] 图22是根据本发明实施例的一种优选的ETL作业的资源分配装置的结构框图；
- [0033] 图23是根据本发明实施例的另一种优选的ETL作业的资源分配装置的结构框图。

具体实施方式

[0034] 为使本申请的上述目的、特征和优点能够更加明显易懂，下面结合附图和具体实施方式对本申请作进一步详细的说明。

[0035] 为了便于理解，下面对本发明实施例中出现的名词进行解释。

[0036] DataStage：一种企业级ETL工具。

[0037] ETL(也称为Etl或etl)：Extraction-Transformation-Loading的缩写，即抽取、转换和加载。ETL指将分散的、异构数据源中的数据如关系数据、平面文件等抽取后进行清洗、转换，最后加载到目标数据平台中的整个数据处理过程。ETL是企业数据集成过程中最典型的数据加工处理方式。

[0038] 数据集成：数据集成是把不同来源、格式、特点性质的数据在逻辑上或物理上有机地集中，从而为企业提供全面的数据共享。

[0039] ETL作业(ETLJob)：ETL作业是进行ETL处理过程的最小可执行单元。

[0040] DataStage作业(DataStageJob)：在DataStage中开发的ETL作业，包括服务器作业(ServerJob)及并行作业(ParallelJob)两类。

[0041] Stage：一个ETL作业包括各个独立的数据处理阶段(Stage)，比如从特定的数据源读取数据，对数据进行特定的加工处理，装载到特定的目标数据库中等。在DataStage作业中，特定的处理阶段都用名为Stage(阶段)的“控件”表示，在DataStage作业的设计面板中，不同的Stage都以特定的图标显示，每个Stage都代表在底层实现特定数据加工逻辑的组件。因为用Stage的中文译名“阶段”比较容易混淆，后续就只使用英文名Stage。

[0042] DataStage并行作业(ParallelJob)：依赖于DataStage并行引擎执行的DataStage作业，可在单台计算机或多台计算机组成的集群环境中运行，通过作业中不同处理阶段并行执行的方式提高达到较高的数据处理效率。

[0043] DataStage并行引擎：可部署于单台或多台服务器上的DataStage数据处理引擎，可构建单机多路并行及多机集群环境。

[0044] DataStage服务器作业(ServerJob)：仅允许在单台服务器上执行的DataStage作业。

[0045] 集群(Cluster)：一个计算机集群是指一组连接起来的电脑，它们共同工作对外界来说就像一个电脑。集群一般用于单台电脑无法完成的高性能计算，整体上拥有较高的性价比。

[0046] 对称多处理(Symmetrical Multi-Processing, SMP)：SMP意指“对称多处理”技术，是指在一个计算机上汇集了一组处理器—即多CPU，各CPU之间共享内存子系统以及总线结构。它是一种应用十分广泛的并行技术。

[0047] 大规模并行处理(Massively Parallel Processing, MPP)：大规模并行处理系统，这样的系统是由许多松耦合的处理单元组成的，每个单元内的CPU都有自己私有的资源，如总线，内存，硬盘等。通常可通过由多台服务器组件集群构建MPP并行计算系统。

[0048] MOIA (Moia) :是先进数通信息技术股份公司开发的企业级调度平台,MOIA针对DataStage进行了深度集成。

[0049] 参照图1,示出了本申请一种ETL作业的资源分配方法,包括:

[0050] 步骤102,建立与ETL集群环境的物理计算资源对应的逻辑层次体系,其中,所述逻辑层次体系包括:直接与硬件计算资源对应的物理节点,以及在物理节点上创建的、对应物理节点部分或全部计算资源的逻辑节点,所述逻辑节点包括:位于特定物理节点的用于执行ETL作业的逻辑节点ETL服务器ETL-SERVER节点,从属于ETL-SERVER节点的位于特定物理节点之上的逻辑节点ETL客户端ETL-CLIENT节点,以及包含一个或多个ETL-SERVER节点和ETL-CLIENT节点的ETL域,其中,所述物理节点、所述ETL-SERVER节点、和所述ETL-CLIENT节点所提供的计算资源采用资源量进行量化;

[0051] 本步骤中,对ETL计算资源进行逻辑上的抽象,包括ETL-域(也称为ETL域、ETL-domain、ETLdomain)、ETL-SERVER(也称为Et1SERVER或et1-SERVER)、ETL-CLIENT(也称为Et1CLIENT或et1-CLIENT)和物理节点这一系列概念,使得对ETL计算资源的分配管理及不同类型ETL作业对计算资源的不同要求纳入到一套统一的框架体系中。此时,可以维护一套数据结构,存储包括ETL-域、ETL-SERVER、ETL-CLIENT和物理节点的属性资源信息以及实时的状态信息,以及作业的信息及状态。优选的,这套数据结构包含在MOIA资料数据库的库表中,可选的,这套数据结构也可以存储于内存数据结构中。可开发一套人机交互界面,用以管理上述包括ETL-域、ETL-SERVER、ETL-CLIENT、物理节点及作业的属性信息,并对状态信息进行展现。在后台调度层面,则需要可以基于“资源量”的算法,具体实现资源分配管理的程序。

[0052] 可选的,在步骤102,可根据ETL作业的复杂度,确定所述ETL作业所需的资源量。该实施例允许根据作业的复杂度,对单个作业的资源量进行区分设置。对于ETL作业,可通过作业所含Stage种类及数量估算作业资源量。

[0053] 步骤104,根据所配置的ETL作业所需的资源量和允许ETL作业运行的逻辑节点范围,以及ETL集群环境中实时资源量使用情况,在集群环境中选择并确定满足所述ETL作业资源量要求的一个或多个逻辑节点,并在所述一个或多个逻辑节点上为所述ETL作业分配资源量。

[0054] 在具体实现时,可以通过以下方式实现步骤104:从ETL作业的配置信息中获取可运行所述ETL作业的ETL域清单、以及所述ETL作业运行限定的ETL-SERVER节点清单;在所述ETL作业运行限定的ETL-SERVER节点及所述ETL-SERVER节点所属的ETL客户端节点中,确定所有满足所述ETL作业运行资源量要求的ETL服务器和ETL客户端节点;从满足所述ETL作业资源量要求的ETL服务器中选择剩余资源量最多的ETL服务器,作为执行所述ETL作业的逻辑节点;或,当所述ETL作业为数据阶段DataStage并行作业时,从所述剩余资源量最多的ETL服务器ETL服务器所属的且满足ETL作业资源量要求的ETL客户端中选择占比最低的N个ETL客户端,作为DataStage并行作业的执行节点,其中,N即DataStage并行作业的并发度,N为大于或等于1的整数

[0055] 在本发明实施例的一个优选实例汇总,当ETL作业为DataStage并行作业时,在设定的可运行所述ETL作业的ETL域、ETL服务器及ETL客户端的范围内,确定资源量满足所述DataStage并行作业运行的资源量需求的ETL服务器及ETL客户端的列表;然后从该列表中

选择当前剩余资源量最多的ETL服务器作为执行该DataStage并行作业的主节点,然后再从从属于该ETL服务器,且剩余资源量大于DataStage并行作业所需资源量的N分之一的ETL客户端中选择剩余资源量最多的N个ETL客户端作为DataStage并行作业的执行节点;其中主节点用于发起并控制DataStage并行作业的执行过程,而执行节点用于实际的数据加工计算。

[0056] 在本发明实施例的另一个优选实例中、对于除DataStage并行作业外的ETL作业,本发明支持单逻辑节点执行方式,即直接在ETL服务器上执行ETL作业,不区分主节点和执行节点;对于此类ETL作业,需在设定可运行所述ETL作业的ETL域、ETL服务器范围内,确定资源量满足所述ETL作业资源量需求的ETL服务器列表;然后在该列表中选择当前剩余资源量最多的ETL服务器作为执行该ETL作业的逻辑节点。

[0057] 其中,可以通过以下方式确定资源量满足所述DataStage并行作业运行的资源量需求的ETL服务器及ETL客户端的列表:

[0058] 从ETL作业的配置信息中获取可运行ETL作业运行的ETL域清单、以及ETL作业运行限定的ETL-SERVER节点清单,只有位于ETL作业运行限定的ETL域中,且在ETL作业运行限定ETL-SERVER节点清单中列出的ETL服务器才可执行ETL作业。在所述可执行ETL作业的ETL服务器,及这些ETL服务器节点所属的ETL客户端节点中,确定所有满足ETL作业运行资源量要求的所有ETL服务器、ETL客户端节点;如不存在满足ETL作业资源量要求的ETL服务器及ETL客户端节点,表明当前环境资源不足,所述ETL作业当前无法执行。如存在满足ETL作业资源量要求的ETL服务器和ETL客户端节点,从满足ETL作业资源量要求的ETL服务器中选择剩余资源量最多的ETL服务器,作为执行ETL作业的逻辑节点;对于DataStage并行作业,需从前述剩余资源量最多的ETL服务器ETL服务器所属的且满足ETL作业资源量要求的ETL客户端中选择占比最低的N个ETL客户端,作为DataStage并行作业的执行节点,其中,N即DataStage并行作业的并发度,N为大于等于1的整数。

[0059] 其中,可以通过以下方式确定所有满足ETL作业运行资源量要求的ETL服务器和ETL客户端节点:

[0060] 当所述ETL作业为非DataStage并行作业时,确定剩余资源量大于或等于所述ETL作业所需资源量的ETL服务器为满足所述ETL作业运行资源量要求的ETL服务器;当所述ETL作业为DataStage并行作业时,判断第一ETL服务器的剩余资源量是否大于或等于DataStage并行作业主节点所需资源量,其中,DataStage并行作业主节点所需资源量是一个预设值;如果是,则判断从属于所述第一ETL服务器所属的ETL客户端中,满足ETL作业运行的资源量要求的ETL客户端是否大于或等于N个,其中,N为DataStage并行作业的并发度,N为大于等于1的整数,如果是,则确定所述第一ETL服务器以及从属于所述第一ETL服务器的满足资源量要求的ETL客户端为满足所述ETL作业运行资源量要求的ETL服务器和ETL客户端节点,否则,确定所述第一ETL服务器及其所属的所有ETL客户端均不满足ETL作业运行资源量要求;重复上述步骤,直到对所有ETL作业运行限定ETL服务器清单中的ETL服务器及其所属的ETL客户端完成判断。

[0061] 在本发明实施例的一个优选实例中,可以通过以下方式判断从属于所述第一ETL服务器的满足资源量要求的ETL客户端是否满足所述资源量要求:对于从属于所述第一ETL服务器的一个或多个ETL客户端,当且仅当ETL客户端的剩余资源量大于或等于所述ETL作

业需要的资源量的N分之一,判定该ETL客户端满足ETL作业运行的资源量要求。

[0062] 步骤106,根据逻辑节点与物理计算资源的对应关系,在所述一个或多个逻辑节点对应的实际物理主机上执行所述ETL作业。

[0063] 在步骤106,根据之前步骤所分配用于执行ETL作业对应的逻辑节点——即ETL服务器和/或ETL客户端——在相应在逻辑节点对应的物理节点,及物理存在的服务器上执行ETL作业。

[0064] 在106步骤中的ETL作业执行过程,包括以下几种方法:

[0065] 方式一、对于非DataStage并行作业,包括DataStageServerJob(服务器作业)及可执行程序ETL作业两种类型。对于DataStageServerJob(服务器作业),直接在104步骤分配用于执行该作业ETL服务器对应的物理节点,调用DataStageAPI执行该作业;对于可执行程序ETL作业,则在操作系统上直接执行对应可执行程序;

[0066] 方式二、对于DataStage并行作业,执行过程分为两个步骤:第一步,根据104步骤确定的执行该DataStage并行作业的ETL服务器及ETL客户端节点,自动生成对应的包含相应物理节点的DataStageParallelJob(并行作业)节点配置文件;第二步,调用DataStageAPI执行该作业,并指定前一步生成的节点配置文件为DataStageParallelJob(并行作业)运行时使用的节点配置文件。

[0067] 本方法可应用于ETL调度平台,在ETL作业运行前按步骤102所述方式完成多个ETL作业的配置,配置完成后,ETL调度平台在调度执行ETL作业时,就可按照步骤104和步骤106完成ETL作业的资源分配及执行过程。

[0068] 对于DataStage并行作业,本实施例通过在硬件和DataStage并行作业之间增加抽象的逻辑节点层次,根据DataStage作业需要的资源量,确定DataStage作业对应的逻辑节点,根据逻辑节点动态生成节点配置文件,从而分配相应的硬件给DataStage作业,避免了每次作业执行前都需要人工介入调整并行引擎配置文件的内容,统一并简化了ETL计算资源管理的需要,提高了资源分配的效率,实现了对资源的统一管理。

[0069] 另外,现有技术中的并行引擎配置文件只能用于DataStage并行作业的并行度及资源分配,对其它类型ETL作业无效,本发明实施例通过本实施例通过对ETL节点,ETL域,ETL-Server,ETL-Client的逻辑抽象,能够满足DataStage并行作业及其它类型ETL作业不同资源分配要求的ETL平台资源管理体系,从而实现了对ETL计算资源的统一管理。

[0070] 基于资源量的对ETL作业并行调度的方法,通常应用于以下典型场景:在以DataStage为主要ETL处理工具的ETL平台,对ETL作业的并发调度进行控制。

[0071] 因为ETL计算资源实际存在于性能及平台各异的物理上存在的服务器上,为解决底层软硬件环境和上层各类型ETL作业对计算环境不同要求二者之间的差异,实现对ETL计算资源的统一管理,本方法在二者中间增加一个逻辑层次,对ETL计算资源进行逻辑上的抽象,以此为基础实现对计算管理的统一管理。

[0072] 现有的调度工具基于实时的系统资源占用情况进行并发控制,但因为缺少对ETL作业所需计算资源的预估机制,所以很容易出现短时间系统负载过大的问题。本方法应用于一种基于资源量的ETL集群环境的并发调度控制框架体系;通过本实施例,能够合理分配ETL作业的资源,实现了集群环境下多机的负载均衡控制,解决了现有技术中的问题。

[0073] 本方法根据基于资源量的ETL集群环境的并发调度控制框架体系中,统一对不同

类型的ETL作业——包括DataStageParallelJob(并行作业)和DataStageServerJob(服务器作业)及可执行程序ETL作业进行统一的资源分配过程,并且在执行ETL作业的阶段,可以根据在DataStage并行作业的资源分配情况,自动创建DataStageParallelJob(并行作业)节点配置文件。通过本方法,构建一个能够对包含有DataStageParallelJob(并行作业)和DataStageServerJob(服务器作业)及可执行程序ETL作业的ETL集群环境,进行统一并发调度控制的完整体系。

[0074] 而现有技术中,DataStage并行作业和其它类型ETL作业的资源分配机制有很大差异。单个DataStage并行作业的并发控制及运行节点由节点配置文件完成,而且必须统一手工配置,且只能用于DataStage并行作业,对其它类型ETL作业无效。本发明实施例通过统一的ETL作业分配资源,统一并简化了ETL计算资源管理的需求,能够满足DataStage并行作业及其它类型ETL作业不同资源分配要求的ETL平台资源管理体系。

[0075] 本方法所述的对ETL计算资源进行逻辑抽象,具体包括如下概念抽象:

[0076] 物理节点:对实际存在的ETL处理能力的抽象,其具体计算能力使用逻辑上的“资源量”概念进行量化,物理节点需配置为实际物理存在的服务器。

[0077] ETL-SERVER:执行ETL作业的逻辑节点,ETL-SERVER存在于具体物理节点之上

[0078] ETL-CLIENT:在集群环境中,进行实际运算处理的逻辑节点,此节点对应DataStage并行作业的执行节点。

[0079] ETL域:对ETL-SERVER和ETL-CLIENT在逻辑上的划分,一个ETL域可以对应特定的业务范围或者应用场景。

[0080] 对“物理节点”、ETL-SERVER和ETL-CLIENT所能提供的计算能力,以及特定ETL作业对ETL计算能力的需求,都使用“资源量”进行度量。然后使用基于“资源量”的分配算法,就可以实现对ETL作业的并发调度进行控制,从而实现对整个ETL平台计算资源的统一管控。

[0081] 允许对特定ETL作业所需的“资源量”进行针对设置。另外,ETL作业对ETL计算能力的需求与其复杂度相关,对于DataStage作业,根据作业中的Stage种类及数量对其所需“资源量”进行估算。

[0082] 需要说明的是,本发明实施例中与DataStage相关的内容,仅是利用了DataStage提供的公开接口实现,并未对DataStage自有的功能模块进行修改或者替换。

[0083] 本发明的技术方案可以集成在MOIA调度平台中予以实现,具体实现分散在MOIA的各个功能模块中。MOIA典型系统拓扑结构如图2所示,整体功能划分及其中和集群调度并发控制相关的模块如下所述:

[0084] MOIA资料库:存储调度配置及状态信息,目前的资料库会部署在关系数据库上。本发明实施例中所涉及的ETL域、ETL-SERVER(ETL服务器)、ETL-CLIENT(ETL客户端)、物理节点的相关信息和作业的资源量配置信息等,都可以存储在资料库中。

[0085] Web服务器(网页服务器):用户管理界面,用于管理资料库中的调度配置信息,并且展现当前调度状态。

[0086] ETL域、ETL-SERVER、ETL-CLIENT、物理节点的相关信息以及作业的资源量配置信息也通过此模块进行展现。

[0087] Web客户端(网页客户端):浏览器,用户通过浏览器可以访问Web服务器端进行管理维护。

[0088] MOIA主节点:MOIA的主调度引擎,负责进行调度任务的扫描,资源分配,及集群管理功能。

[0089] 与本发明实施例相关的模块包括:

[0090] 集群管理服务:负责监控集群中各物理节点、ETL-SERVER以及ETL-CLIENT的状态,并将状态更新到资料库相应状态表中。

[0091] 资源分配服务:负责从作业队列中检索出满足调度条件的ETL作业,然后匹配各物理节点、ETL-SERVER、及ETL-CLIENT的当前资源使用情况,为作业分配资源。

[0092] MOIA子节点:实际发起ETL作业执行过程的ETL物理节点、

[0093] 作业执行服务:负责从作业队列中取已完成资源分配的作业,然后执行作业。对于DataStage并行作业,需要根据资源分配情况,先生成相应并行引擎配置文件,并通过\$ATP_CONFIG_FILE参数传递给被调用的并行作业。

[0094] 下面对本实施例的具体实现过程进行说明。

[0095] 首先,建立与硬件计算资源(即ETL集群环境)对应的逻辑层次体系,该体系包括:直接与硬件计算资源对应的物理节点,位于特定物理节点的用于分配执行ETL作业的的逻辑节点ETL-SERVER节点,与ETL-SERVER节点所属的位于特定物理节点的逻辑节点ETL-CLIENT节点,由ETL-SERVER节点和ETL-CLIENT节点所构成的ETL域,以及运行于这些逻辑节点之上的ETL作业。物理节点、ETL-SERVER节点和ETL-CLIENT节点均采用资源量来对节点的计算能力进行量化,与之对应地,ETL作业对计算资源的需求也采用资源量进行量化。也就是说,对ETL计算资源进行逻辑上的抽象,将不同层级的计算资源抽象为一系列逻辑概念,并通过“资源量”这个抽象概念对这些计算资源予以量化。资源量是一个逻辑概念,用以代表作业预计占用的资源权重及平台环境中各逻辑层级所能提供的资源权重。本发明中对表示资源量的权重值的具体设定算法不做限定,实际设定权重值时可以基于经验值,但作业预计占用资源量的权重值和平台环境资源量的权重值需使用相同体系进行评估。下面对各个逻辑概念进行说明。

[0096] 物理节点:物理节点代表可以提供实际ETL处理能力的计算单元,物理节点既可以是对应于实际物理上存在的服务器——即拥有独立的CPU和内存资源有明确分配的主机名和IP地址的物理机器,也可以是对物理机器一部分计算能力的抽象。对于后者,常见于SMP架构的多CPU或者多核高性能服务器,因为单台服务器本身就提供足够强大的计算能力,将其划分为多个逻辑上独立的节点,更有利于通过多路并行提高处理效率。但不论上述哪种情况,这个逻辑概念均有物理上实际存在的计算能力相对应,所以称之为物理节点。

[0097] ETL域:逻辑域,对ETL节点范围的划分,其包含若干ETL_SERVER。从调度的角度来讲,即ETL作业可以执行的逻辑范围。

[0098] ETL_SERVER:即,ETL服务器,ETL作业执行的逻辑节点,对于DataStage作业来说,该逻辑节点又是并行管理节点。每个ETL_SERVER映射一个物理节点。ETL作业指定了运行的逻辑节点,即指定了最终运行该作业的物理节点。

[0099] ETL_CLIENT:即,ETL客户端,对于DataStage作业来说,为ETL作业的并行执行节点,每个ETL_CLIENT映射一个物理节点。对于非并行作业来说,ETL_CLIENT是无意义的。

[0100] 通过这些逻辑概念层次,在ETL作业运行层面,屏蔽不同软硬件环境,不同的硬件部署拓扑结构的差异,满足不同类型,不同复杂度ETL作业对计算资源的不同要求,从而实

现对ETL计算资源的统一管理。

[0101] 然后,设计并实现一套调度服务程序(例如,MOIA调度平台),这一套调度服务程序可根据ETL集群环境及ETL作业的资源量配置及集群环境中特定逻辑节点的实时的资源量的占用情况,指定ETL集群环境中特定的ETL域中的特定ETL-SERVER、ETL-CLIENT运行特定ETL作业,并将实际执行逻辑映射到集群环境中这些逻辑节点实际对应的物理主机予以执行。上述ETL作业的类型可包括:支持分布式方式并行执行的DataStageParallel Job(并行作业)和仅支持单机执行的DataStageServerJob(服务器作业)及可执行程序作业。

[0102] 图3是一个典型的包含上述逻辑概念的MOIA逻辑集群。其中,MOIAServer即为前文所述的MOIA主节点。而ETL-SERVER和ETL-CLIENT均是逻辑节点,而对应实际进行ETL处理的物理节点仅有ETL_node1、ETL_node2和ETL_node3。ETL-SERVER之上的ETL域也可以根据应用场景任意划分,比如划分两个ETL域,其中之一包含ETL-SERVER1和ETL-SERVER2,另一个包含ETL-SERVER3。通过增加这个逻辑层次,可以组合出满足各种场景的集群架构。

[0103] 对上述逻辑概念,在资料库中需分别建立配置表及状态表,配置表中记录属性特征及“资源量”,状态表中记录当前的“资源量”使用情况,如图4和图5所示。

[0104] 配置表图4是作业运行的平台环境相关的包括(ETL域,物理节点等)的配置表及状态表的E-R(实体-关系)示意图;图5则是作业自身的配置表及状态表的E-R(实体-关系)示意图。图4和图5中的配置表及状态表均只保留了和本发明相关的和资源量及资源分配相关的字段。因为两部分之间主要是通过程序逻辑产生的关联关系,并非是直接的关系数据模型层面的关联关系,所以分为两部分。具体来说,就是基于本实施例实现的调度程序会根据作业配置表中的相关资源设置,包括作业并发度、所属ETL域、所属ETL-SERVER这些配置信息,结合图4列出的环境相关的配置表中的配置以及状态表中的当前环境实时运行状态,来从环境中选取合适的资源(即选取ETL域、ETL-SERVER和ETL-CLIENT)运行作业。这两个图均是示意图,只简化保留了和本发明逻辑密切相关的字段。

[0105] 对ETL作业运行所需“资源量”的设定。物理节点的资源数,在进行MOIA安装时候默认给出基础平衡值,用户可根据机器情况进行调整;对可提供资源数,可以通过以下方式设置:注意因为同一台服务器上理论上可以指定多个ETL物理节点以提高并发度,以充分利用单台服务器的资源,在这种情况下,在单个物理节点上,服务器配置以4C8G(4CPU,8G内存)为基数值,以(现Cpu数C-4C)/Cpu基数值4C=Cpu浮动百分比,以(现内存数G-8G)/内存基数值8G=内存浮动百分比,可提供资源数上限=资源基数值2000+1000*(Cpu浮动百分比+内存浮动百分比)。ETL-SERVER及ETL-CLIENT在物理节点资源数的基础上,人工进行分配。

[0106] 对于DataStage作业,通过作业中所包含的Stage的种类及数量确定所需“资源量”的基准值,在此基础上,用户可根据实际情况对作业所需“资源量”进行调整。针对DataStageParallelJob(并行作业),其本身支持集群环境中的分布式方式执行,但其并发控制是通过节点配置文件(通过\$APT_CONFIG_FILE环境变量指定)设定。当MOIA调度平台对ETL集群环境中的DataStageParallelJob(并行作业)进行调度执行时,可根据物理节点、ETL域、ETL-SERVER节点、ETL-CLIENT节点、及作业本身的设置,动态生成DataStageParallelJob(并行作业)的节点配置文件。通过上述方式,可以将DataStageParallelJob(并行作业)的并发调度管理与本实施例所述的框架体系统一。

[0107] 在实际作业运行时,调度程序按图6所示的处理流程进行资源分配并实际调用ETL

作业,如图6所示,该流程包括:

[0108] 步骤601,取作业队列表中满足调度条件的作业,按作业的优先级进行排序;

[0109] 步骤602,循环逐个遍历上述作业,若遍历完成,则结束扫描过程,否则执行步骤603;

[0110] 步骤603,获取作业配置表中的配置信息;

[0111] 步骤604,确定剩余资源量满足作业运行资源量要求的ETL服务器以及ETL客户端,列出所有满足要求的ETL服务器以及ETL客户端的列表,本步骤使用独立的处理逻辑单元实现,其详细处理逻辑见图6A。

[0112] 步骤605,判断604步骤所得列表是否为空,如是,则未找到满足作业运行资源量要求的ETL服务器以及ETL客户端,则跳转到610步骤,标记当前ETL作业状态为“资源不足”,如否,则继续执行步骤606。

[0113] 步骤606,在满足ETL作业运行资源量要求的ETL服务器中选择剩余资源量最多的ETL服务器作为执行ETL作业的逻辑节点,并继续执行步骤607。

[0114] 步骤607,判别是否为DataStage并行作业,如果是,则执行步骤608,否则,直接执行步骤609。

[0115] 步骤608,在满足ETL作业运行要求的ETL客户端中选择剩余资源量最多的N个客户端;

[0116] 步骤609,为指定作业分配ETL服务器和ETL客户端;

[0117] 步骤610,将作业状态标记为“资源不足”,返回步骤602。

[0118] 对步骤604的详细说明见图6A,该流程包括:

[0119] 步骤611,取ETL作业“ETL域列表”及“ETL-Server”列表配置,即限定ETL作业执行的ETL域及ETL-SERVER的范围。

[0120] 步骤612,遍历步骤611列出的范围内的所有ETL-Server。

[0121] 步骤613,取当前ETL域中资源量满足ETL作业运行要求ETL服务器。资源量满足ETL作业运行要求包括两种情况:一种是非DataStage并行作业,满足作业运行要求及ETL服务器的剩余资源量大于ETL作业所需资源量;另一种则是对DataStage并行作业,满足作业运行要求及ETL服务器指ETL服务器的剩余资源量大于“调用DataStage并行作业所需资源量”,“调用DataStage并行作业所需资源量”是一个平台预设的固定值,与ETL作业无关,表示控制DataStage并行作业的资源消耗。

[0122] 步骤614,遍历上述ETL作业运行要求ETL服务器。

[0123] 步骤615,判断ETL作业是否为DataStage并行作业,如是,则继续到616步骤,如否,则直接跳转到618步骤。

[0124] 步骤616,在所述ETL服务器所属的ETL客户端中,查找剩余资源量大于DataStage并行作业所需资源量的N分之一的ETL客户端,N为DataStage并行作业的并发度。统计找到的ETL客户端的数量,并继续617步骤。

[0125] 步骤617,如步骤616中找到的ETL客户端大于等于N,则表明该ETL服务器及其所属的ETL客户端满足ETL作业执行的资源要求,继续到618步骤;如否,则认为ETL服务器不满足DataStage并行作业的资源量需求,跳转到614步骤,继续检查下一个ETL服务器。

[0126] 步骤618,确认ETL服务器以及ETL客户端满足执行ETL作业的资源量要求并记录,

即将满足ETL作业运行的资源量要求的ETL服务器以及ETL客户端添加到列表中。步骤618完成后,跳转到614继续检查下一个ETL服务器。

[0127] ETL作业的资源分配完成后,调度平台就可以根据资源分配情况,在所分配的逻辑节点对应的物理节点上执行ETL作业。作业执行服务的处理流程如图7所示,该流程包括:

[0128] 步骤701,获取作业队列中已完成资源分配的作业。

[0129] 步骤702,循环逐个遍历上述作业,如果遍历完成,则结束扫描过程,否则,执行步骤703。

[0130] 步骤703,取分配用于执行该ETL作业的ETL服务器。。

[0131] 步骤704,更新平台的资源使用情况,标记作业运行状态为执行中。。

[0132] 步骤705,在ETL-Server上调用ETL作业执行程序执行。此调用过程为异步过程,ETL作业执行程序执在独立进程中执行,调用后记录返回并跳转到702步骤继续遍历过程。

[0133] 对ETL作业执行程序的处理过程说明见图7A。其处理逻辑包括如下步骤:

[0134] 步骤711,去ETL作业信息表中的作业配置信息。

[0135] 步骤712,判断ETL作业的是否为DataStage并行作业。如是,则继续713步骤;如否,则直接跳转至714。

[0136] 步骤713,取分配给DataStage并行作业,相应生成DataStage节点配置文件,并将该文件指定为DataStage并行作业执行时使用的节点配置文件。

[0137] 步骤714,直接在已分配的ETL-SERVER上执行作业,并监控作业执行过程,等待作业执行完毕。。

[0138] 步骤715,作业执行完毕后,更新作业状态,并回收ETL作业占用的资源。

[0139] 本发明实施例还提供了一种DataStage任务的资源分配方法,该方法可以应用于图3所示的拓扑结构中。

[0140] 首先,进行资源设置。

[0141] 定义物理节点ETL节点1 (ETL_node1、etl_node1)、ETL节点2 (ETL_node2、etl_node2) 和ETL节点3 (ETL_node3、etl_node3)。

[0142] 新建物理节点ETL_node1,参数设置如图8所示,

[0143] 另外,还需设定物理节点资源 (此处资源特指DataStage并行作业所需的磁盘缓冲区等存储资源),设置的方式如图9所示,列表中列出节点资源名和节点资源路径等信息,选取节点资源名前的选项,即可选择相应的资源。

[0144] 类似的,再新建ETL_node2和ETL_node3两个物理节点,资源数均定义为2000,如图10所示。

[0145] 新建ETL域 (Et1_Domain),参数设置如图11所示;新建ETLServer (ETL-SERVER), ETL_Server1,如图12所示。最终得到图13所示的结构。

[0146] 按相同办法创建备用的ETL域ETL_Domain_bak,如图14所示,最终建立了与图3所示拓扑图一致的逻辑集群。

[0147] 然后,对ETL作业的属性进行设置。

[0148] 1) ETL作业的属性设置

[0149] 以两个测试作业为例,一个作业是名为PJOB_1的DataStage并行作业,另一个则是名为ETL_SCRIPT_1的脚本作业,如图15所示。设置其为DataStage并行作业,如图16所示,

[0150] 在其动态调度属性中,可以设置如下参数,其中和本实施例相关的是ETL域和ETL-Server限定及作业复杂度(即所需资源量)、及并行度上限,数据量权值则用于确认作业的优先级,如图17所示。

[0151] 新建另外一个作业,如图18-20所示。然后,对作业的执行过程的调度进行并发控制。

[0152] 下面以两个测试作业PJOB_1和ETL_SCRIPT_1为例简述本方案中集群并发调度的运作机制。

[0153] 首先说明资源分配环节的运作机制。

[0154] 假设两个作业同时满足调度条件,则资源分配服务会从作业队列表中取出这两个作业进行资源分配。

[0155] 因为ETL_SCRIPT_1的优先级高于PJOB_1,所以ETL_SCRIPT_1会优先被分配资源。

[0156] ETL_SCRIPT_1的资源分配:

[0157] ETL_SCRIPT_1的需求资源量为20,因其为脚本,不能并行执行,并行度为1。

[0158] 因为其指定在名为ETL域(ETL-Domain)的ETL域中运行,所以发起作业执行的ETL-SERVER只能在ETL-Domain中的两个ETL-Server中选择,即ETL-SERVER1和ETL-SERVER2。

[0159] 资源分配服务会查询ETL-SERVER状态表,确认两个ETL-SERVER的资源占比哪一个更低,会优先选择更低的ETL-SERVER分配给ETL-SERVER作业。

[0160] 如果两个ETL-SERVER的剩余资源量均小于20,则资源分配失败,ETL-SERVER更新ETL_SCRIPT_1的状态为“资源不足”,然后结束本轮对ETL_SCRIPT_1的资源分配尝试,下一轮扫描时再重新对ETL_SCRIPT_1进行资源分配。

[0161] 因为ETL_SCRIPT_1并非DataStage并行作业,所以在给其分配了ETL-SERVER后,对其的资源分配的过程就结束了。

[0162] PJOB_1的资源分配:

[0163] ETL_SCRIPT_1的资源分配过程完毕后,继续队列中满足调度条件的其它作业进行资源分配,比如此例中的ETL作业PJOB_1。

[0164] 截止ETL-SERVER的分配过程,PJOB_1与ETL_SCRIPT_1基本相同,但因为PJOB_1是DataStage并行作业,所以选定ETL-SERVER后,还需分配合适的ETL-CLIENT作为并行作业的执行节点。

[0165] 因为PJOB_1指定为两路并行,总资源需求量为200,所以其对单个ETL-CLIENT的资源需求量为100,资源分配服务程序会检索ETL-CLIENT状态表,从之前选定的ETL-SERVER下属的ETL-CLIENT中选择当前资源占比低的两个ETL-CLIENT节点分配给PJOB_1。

[0166] 如不能找到满足资源要求的ETL-CLIENT,则资源分配服务程序会尝试重新指定其他的资源占用率次高,但仍能满足资源要求的ETL-SERVER,然后重新匹配ETL-CLIENT。

[0167] 如最终不能找到满足资源要求的ETL-SERVER和ETL-CLIENT,则认为资源分配失败,结束本轮分配过程。

[0168] 然后,说明作业执行环节的机制。

[0169] 假设对测试作业PJOB_1和ETL_SCRIPT_1的资源分配过程都成功,则负责作业执行的作业执行服务会从作业队列中检索出这两个作业,并在分配的ETL-SERVER和ETL-CLIENT上执行作业。

[0170] 对于ETL_SCRIPT_1,作业执行服务会直接在指定的ETL-SERVER对应的物理节点所在的实际服务器上,执行ETL_SCRIPT_1对应的脚本程序。

[0171] 而对于并行作业PJOB_1,过程稍复杂一些,作业执行服务需要在实际执行作业前,根据其分配的ETL-SERVER和ETL-CLIENT生成节点配置文件。比如,资源分配环节指定使用ETL-SERVER2及ETL-CLIENT4,ETL-client5三个节点执行作业。则服务程序会生成如下所示的节点配置文件。

[0172] 注意ETL-SERVER,和ETL-CLIENT和底层物理节点的对应关系。

[0173]

```
{  
  
  node "ETL-SERVER2" {  
  
    fastname "ETL_node2"  
  
    pool "conductor"  
  
  }  
  
  node "ETL-CLIENT4" {  
  
    fastname "ETL_node1"  
  
    pool ""  
  
    resource disk "/home/dsadm/Ascential/DataStage/Dataset" {}  
  
    resource scratchdisk "/home/dsadm/Ascential/DataStage/Scratch" {}  
  
  }  
}
```

[0174]

```
node "ETL-CLIENT5" {  
  
    fastname "ETL_node2"  
  
    pool ""  
  
    resource disk "/home/dsadm/Ascential/DataStage/Dataset" {}  
  
    resource scratchdisk "/home/dsadm/Ascential/DataStage/Scratch" {}  
  
}  
  
}
```

[0175] 生成上述配置文件后,作业执行服务会将配置文件存储于临时文件中,然后将临时文件的路径作为并行作业PJOB_1的\$APT_CONFIG_FILE参数的参数值,然后通过DataStageAPI在ETL-SERVER2(即物理节点ETL_node2)上执行作业。

[0176] 最终,DataStage并行作业PJOB_1的执行节点(Conductornode)是物理节点ETL_node2,而处理节点(Processingnode)则是ETL_node1和ETL_node2。

[0177] 通过上述本发明实施例,解决了以下三个技术问题:

[0178] 1、以DataStage为基础构建的ETL集群环境中DataStage作业和可执行程序ETL作业的并发调度控制,及集群环境下多机的负载均衡控制问题。

[0179] DataStage并行引擎支持单台SMP服务器的多路并行计算、及多台服务器构建的MPP集群并行环境、以及二者组合而成的SMP集群并行计算环境。但是,DataStage虽然提供了对并行作业的并发度进行控制的接口,但自身并未集成并行控制机制,所以需要集成第三方工具对DataStage并行作业的并发度,以及在集群环境下的多台服务器的负载均衡进行控制。

[0180] 2、对包括DataStage在内的整个ETL平台的计算资源进行统一管理的问题。

[0181] 对于以DataStage为主要数据加工处理工具的ETL平台,平台上运行的ETL作业通常包括DataStage服务器作业、DataStage并行作业及可执行程序多种类型,各类型作业有各自不同的资源分配要求。另外,ETL平台还需满足不同业务领域及不同应用场景(比如测试环境,生产环境的区分),各自不同的资源分配特点及要求。由此,需要有一种抽象机制计算资源进行抽象,从而基于此实现对包括DataStage在内的ETL平台的ETL平台计算资源的统一管理。

[0182] 3、对单个DataStage作业的占用资源的估算问题。

[0183] 因为不同DataStage作业的复杂度有很大差异,其运行时可能占用的系统资源会有很大不同,所以为更好的进行作业的并发控制和集群负载均衡,有必要在对作业的复杂度进行预估的前提下,对作业占用的资源量进行差异化设置。

[0184] 本发明实施例还提供了一种ETL作业的资源分配装置,该装置用于实现上述方法,因此,上述方法实施例中的特征可以应用到本实施例中。如图21所示,该装置包括:

[0185] 配置模块2102,用于建立与ETL集群环境的物理计算资源对应的逻辑层次体系,并对集群环境所能提供计算资源,使用资源量的概念进行量化,其中,所述逻辑层次体系包括:直接与硬件计算资源对应的物理节点,以及在物理节点上创建的、对应物理节点部分或全部计算资源的逻辑节点,所述逻辑节点包括:位于特定物理节点的用于执行ETL作业的逻辑节点ETL服务器ETL-SERVER节点,从属于ETL-SERVER节点的位于特定物理节点之上的逻辑节点ETL客户端ETL-CLIENT节点,以及包含一个或多个ETL-SERVER节点和ETL-CLIENT节点的ETL域,其中,所述物理节点、所述ETL-SERVER节点、和所述ETL-CLIENT节点均采用资源量来对节点的计算资源进行量化;

[0186] 计算资源分配模块2104,用于根据所配置的ETL作业所需的资源量和允许ETL作业运行的逻辑节点范围,以及ETL集群环境中实时资源量使用情况,在集群环境中选择并确定满足所述ETL作业资源量要求的一个或多个逻辑节点,并在所述一个或多个逻辑节点上为所述ETL作业分配资源量;

[0187] ETL作业执行模块2106,用于根据逻辑节点与物理计算资源的对应关系,在所述一个或多个逻辑节点对应的实际物理主机上执行所述ETL作业。

[0188] 在本发明实施例的一个优选实例中,所述计算资源分配模块2104用于:从所述ETL作业的配置信息中获取可运行所述ETL作业的ETL域清单、以及所述ETL作业运行限定的ETL-SERVER节点清单;在所述ETL作业运行限定的ETL-SERVER节点及所述ETL-SERVER节点所属的ETL客户端节点中,确定所有满足所述ETL作业运行资源量要求的ETL服务器和ETL客户端节点;从满足所述ETL作业资源量要求的ETL服务器中选择剩余资源量最多的ETL服务器,作为执行所述ETL作业的逻辑节点;或,当所述ETL作业为数据阶段DataStage并行作业时,从所述剩余资源量最多的ETL服务器ETL服务器所属的且满足ETL作业资源量要求的ETL客户端中选择占比最低的N个ETL客户端,作为DataStage并行作业的执行节点,其中,N即DataStage并行作业的并发度,N为大于或等于1的整数。

[0189] 在本发明实施例的一个优选实例中,所述计算资源分配模块2104用于:当所述ETL作业为非DataStage并行作业时,确定剩余资源量大于或等于所述ETL作业所需资源量的ETL服务器为满足所述ETL作业运行资源量要求的ETL服务器;当所述ETL作业为DataStage并行作业时,判断第一ETL服务器的剩余资源量是否大于或等于DataStage并行作业主节点所需资源量,其中,DataStage并行作业主节点所需资源量是一个预设值;如果是,则判断从属于所述第一ETL服务器所属的ETL客户端中,满足ETL作业运行的资源量要求的ETL客户端是否大于或等于N个,其中,N为DataStage并行作业的并发度,N为大于或等于1的整数,如果是,则确定所述第一ETL服务器以及从属于所述第一ETL服务器的满足资源量要求的ETL客户端为满足所述ETL作业运行资源量要求的ETL服务器和ETL客户端节点,否则,确定所述第一ETL服务器及其所属的所有ETL客户端均不满足ETL作业运行资源量要求;重复上述步骤,直到对所有ETL作业运行限定ETL服务器清单中的ETL服务器以及其所属的ETL客户端完成

判断。

[0190] 在本发明实施例的一个优选实例中,所述计算资源分配模块2104用于通过以下方式判断从属于所述第一ETL服务器的满足资源量要求的ETL客户端是否满足所述资源量要求:对于从属于所述第一ETL服务器的一个或多个ETL客户端,当且仅当ETL客户端的剩余资源量大于或等于所述ETL作业需要的资源量的N分之一,判定该ETL客户端满足ETL作业运行的资源量要求。

[0191] 在本发明实施例的一个优选实例中,所述配置模块2102用于:在根据所配置的ETL作业所需的资源量和ETL作业所能运行的逻辑节点,以及ETL集群环境中实时资源量使用情况,确定运行所述ETL作业的所需的一个或多个逻辑节点之前,根据所述ETL作业的复杂度,确定所述ETL作业所需的资源量。如图22所示为适用于非DataStage并行作业的ETL作业的资源分配装置的优选的模块示意图。

[0192] 如图23所示为适用于DataStage并行作业的ETL作业的资源分配装置的优选的模块示意图。

[0193] 计算资源分配模块2104包括:

[0194] 第一确定单元2202,用于在设定可运行指定ETL作业的ETL域及ETL服务器的范围内,确定满足ETL作业运行的资源量需求的ETL服务器的列表;

[0195] 对于DataStage并行作业,在第一确认单元中,需进一步确定ETL服务器所属的ETL客户端中是否存在N个及N个以上的满足DataStage并行作业运行的资源量需求的ETL客户端——N为DataStage并行作业的并发度,如不能,则认为该ETL服务器不满足作业运行的资源量需求,如找到满足条件的至少N个ETL客户端,则认为该ETL服务器满足作业运行的资源量需求,确定所有满足上述条件的ETL服务器和ETL客户端的列表;

[0196] 第一选择单元2204,用于从第一确定单元确定的ETL服务器列表中选择剩余资源量最多的ETL服务器,将其分配给指定ETL作业。

[0197] 对于DataStage并行作业,第一选择单元还需进一步在选定的ETL服务器所属的ETL客户端中选择剩余资源量最多的N个ETL客户端将其分配给指定DataStage并行作业。

[0198] ETL作业执行模块2106包括:第一执行单元2208,用于在计算资源分配模块2104确定的ETL服务器对应的物理节点上执行ETL作业,包括两种情况:对于可执行程序ETL作业,直接在操作系统执行出,对于DataStage并行作业及DataStage服务器作业,通过DataStageAPI执行作业。对于DataStage并行作业,执行时需要指定DataStage节点配置文件。

[0199] 对于DataStage并行作业,ETL作业执行模块2106还包括:第一建立单元2206。用于根据分配模块分配给DataStage并行作业的ETL服务器及其所属的ETL客户端,生成相应的DataStage节点配置文件。

[0200] 本说明书中的各个实施例均采用递进的方式描述,每个实施例重点说明的都是与其他实施例的不同之处,各个实施例之间相同相似的部分互相参见即可。对于系统实施例而言,由于其与方法实施例基本相似,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[0201] 以上对本申请所提供的一种ETL作业的资源分配方法和装置,进行了详细介绍,本文中应用了具体个例对本申请的原理及实施方式进行了阐述,以上实施例的说明只是用于

帮助理解本申请的方法及其核心思想;同时,对于本领域的一般技术人员,依据本申请的思想,在具体实施方式及应用范围上均会有改变之处,综上所述,本说明书内容不应理解为对本申请的限制。

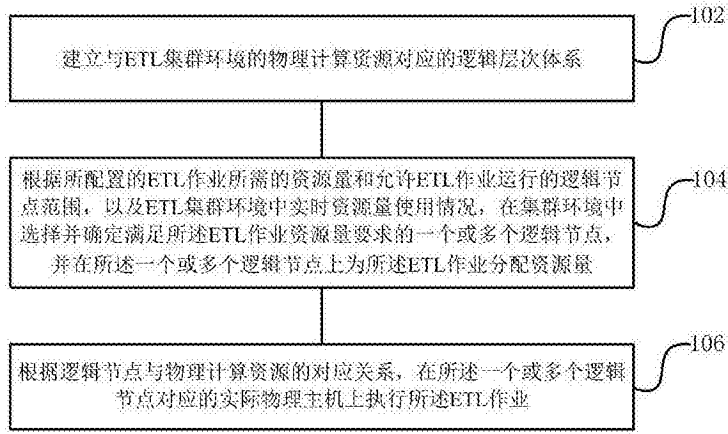


图1

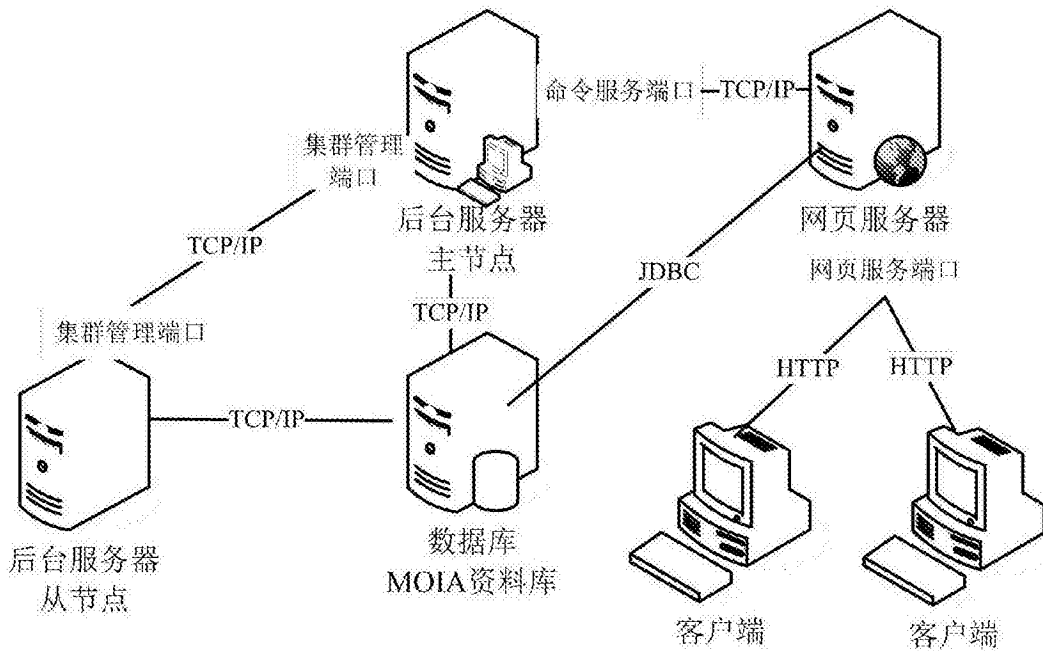


图2

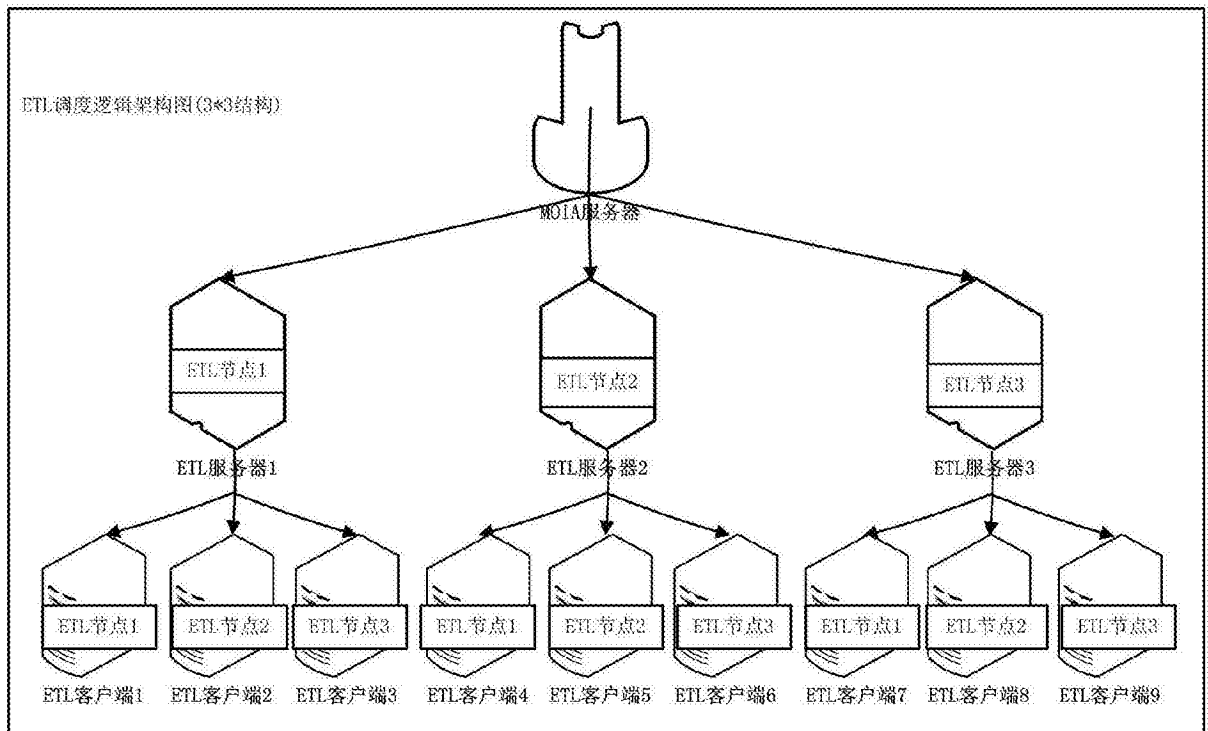


图3

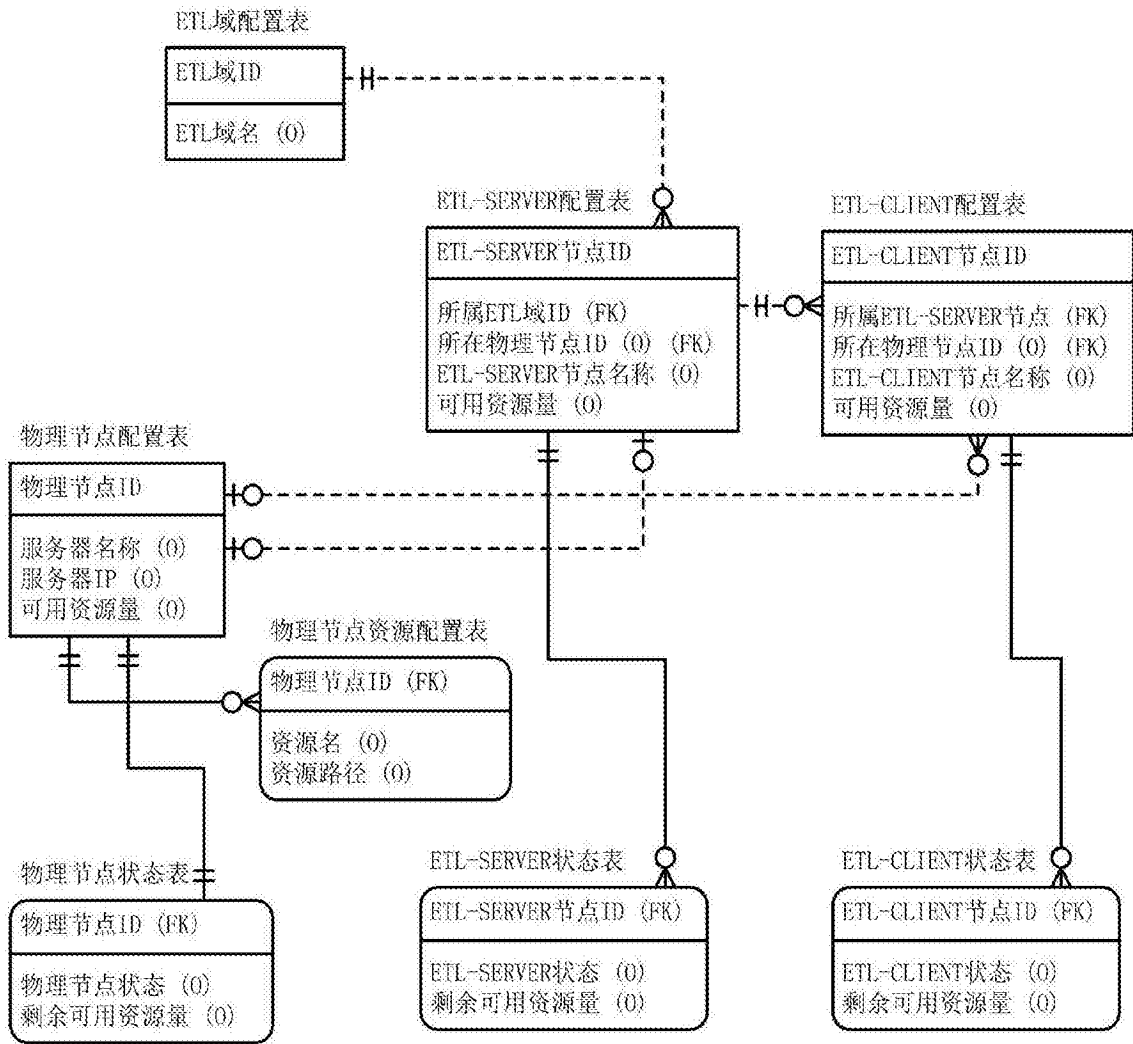


图4

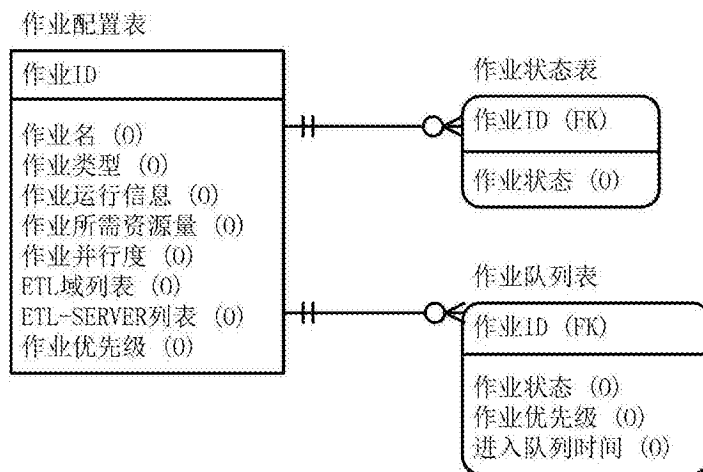


图5

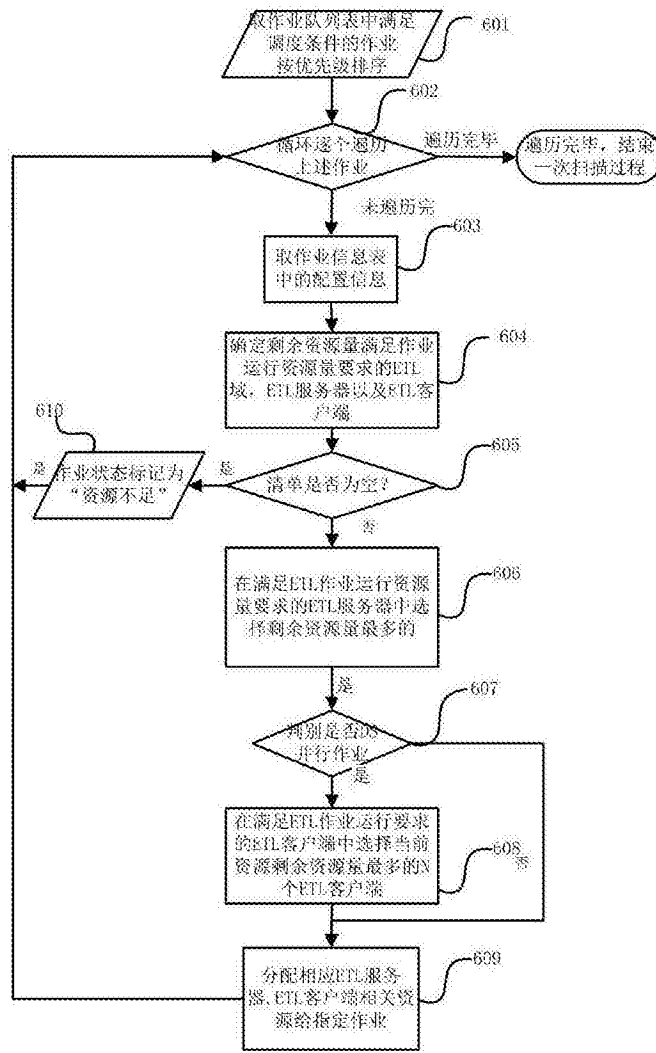


图6

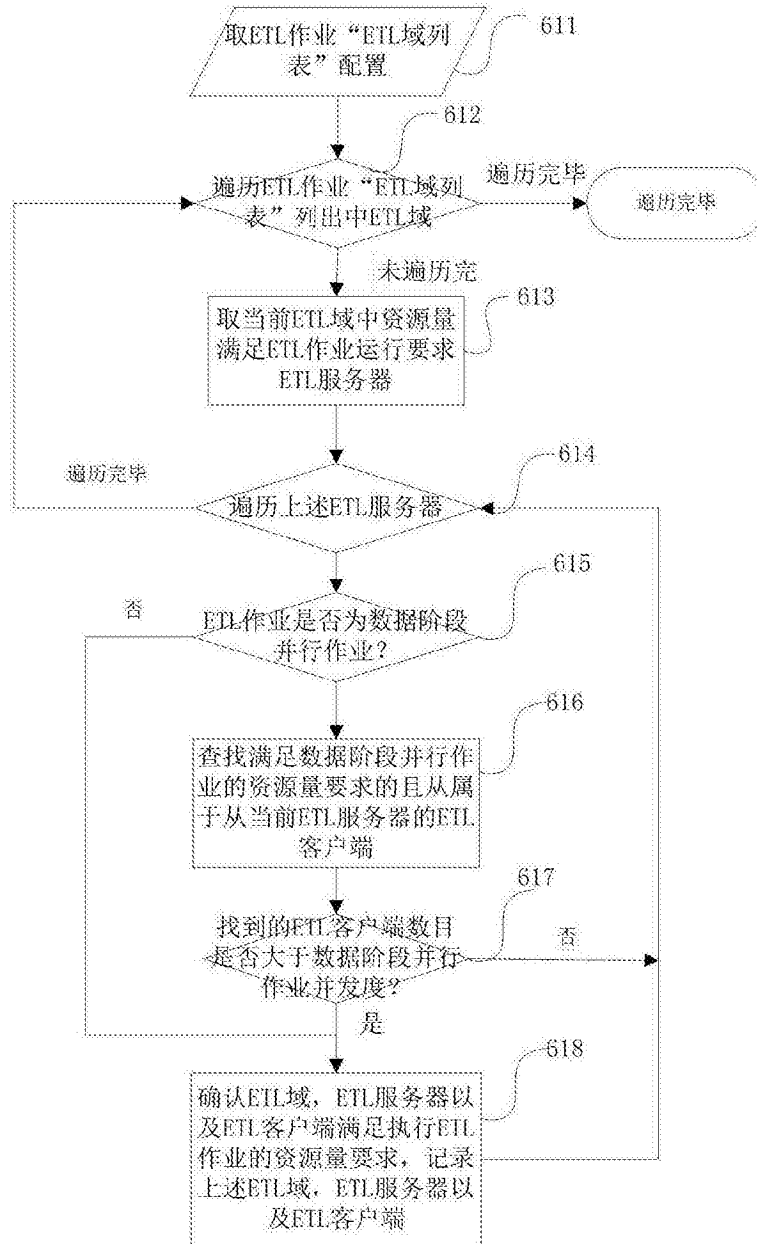


图6A

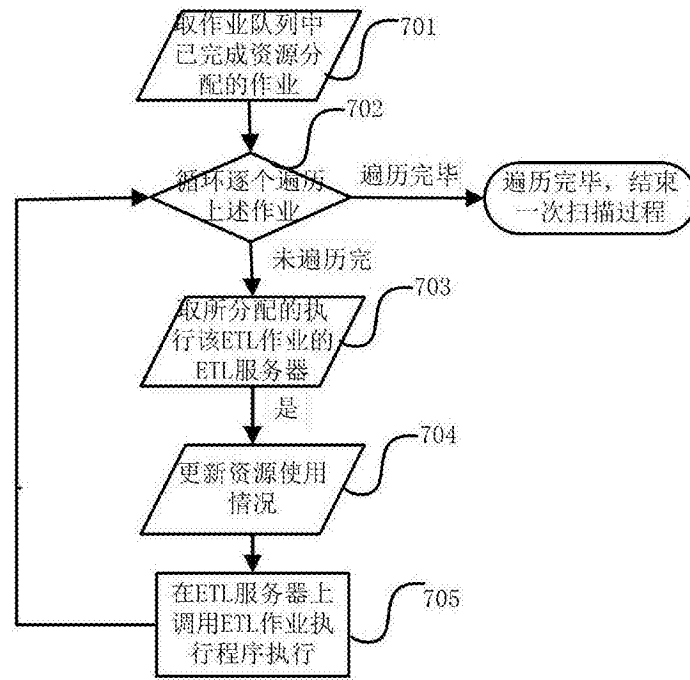


图7

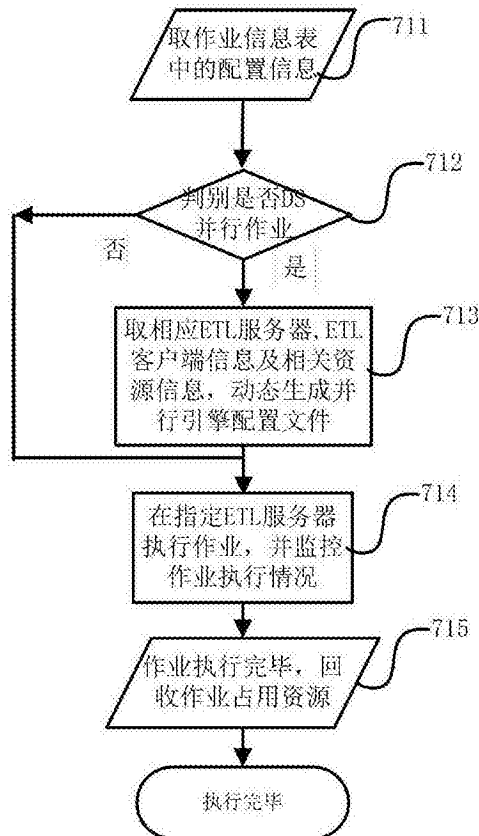


图7A

新建物理节点

※请输入物理节点基本信息，其中打*号为必填项：

*节点ID：1012

*节点名称：etl_node1

*所属节点：TEST_CLUSTER

*主节点/从节点：从节点

*能否做ETL-Server：是

*能否做ETL-Client：是

节点描述：

*节点地址：192.168.1.250

*主机名：etlsrv1

*MoiLsnr端口：57500

*群集服务端口：57501

*命令端口：57502

*事件管理端口：57503

*文件接收端口：57504

*网络带宽：800

*可用CPU数(个)：1

*可用内存(M)：3018

*IO最大吞吐量(G/分钟)：1

*可提供资源数：2000

*最大DSjob数：100

*最大非DSjob数：1

图8

选择	节点资源名	节点资源路径	资源pool列表	节点资源描述
<input type="checkbox"/>	disk	/home/dsadm/Ascential/DataStage/Datasets	""	
<input type="checkbox"/>	scratchdisk	/home/dsadm/Ascential/DataStage/Scratch	""	

图9

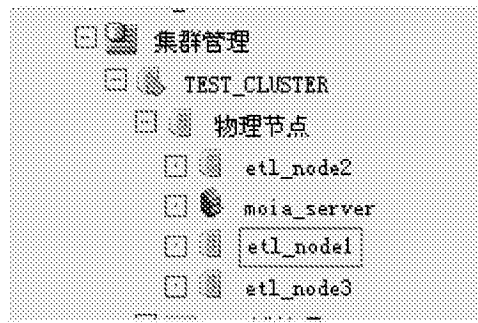


图10



图11



图12

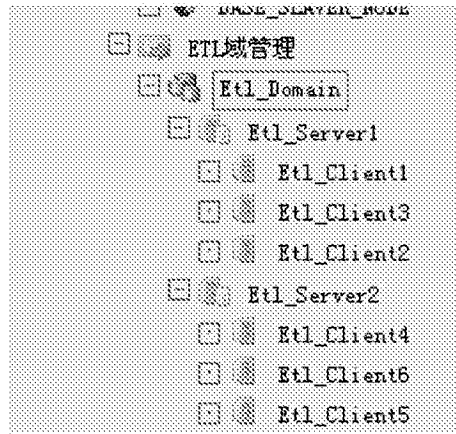


图13

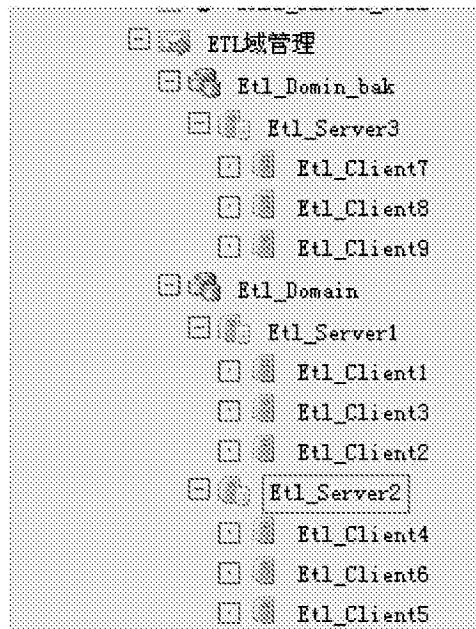


图14

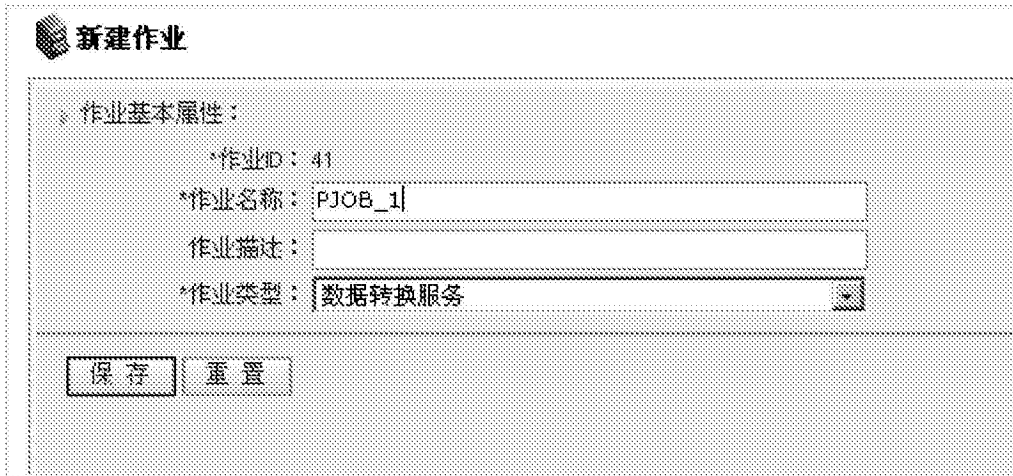


图15



图16

> 作业调度方式： 使用执行节点模式 自定义

*ETL域：

*ETL-Server：

*资源模式：

*作业复杂度：

*数据量权值：

*并行度上限：

作业资源类型：

图17

新建作业

> 作业基本属性：

*作业ID：41

*作业名称：

作业描述：

*作业类型：

图18

基本属性 | 作业参数 | 权限设置 | 返回值预设

> 作业功能配置：

处理类型：

*可执行程序名：

*作业返回值预设： 系统默认值 自定义预设返回值 返回值预设类

图19

作业调度方式： 使用执行节点模板 自定义

*ETL域：

*ETL-Server：

*资源模式：

*作业复杂度：

*数据量权值：

*并行度上限：

作业资源类型：

图20



图21

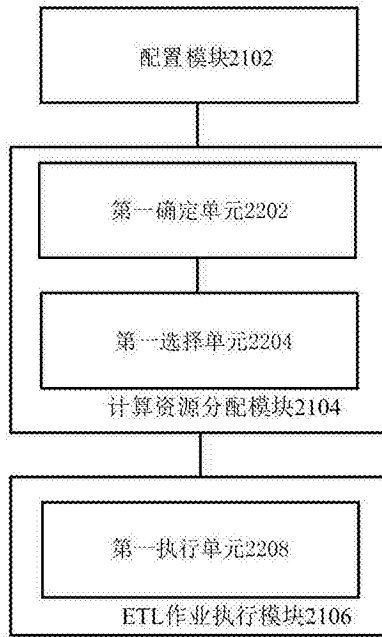


图22

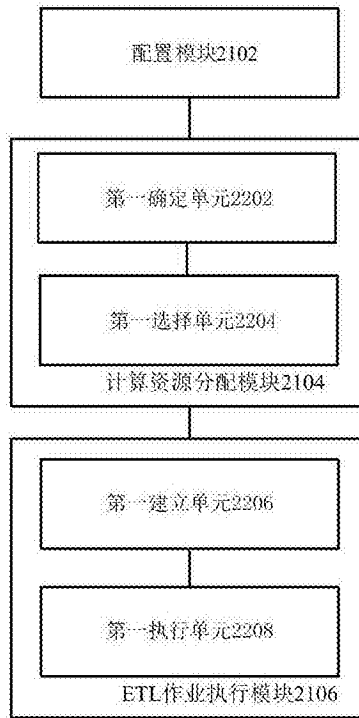


图23