



(12) 发明专利申请

(10) 申请公布号 CN 114096961 A

(43) 申请公布日 2022. 02. 25

(21) 申请号 202180001736.9

闫家奇

(22) 申请日 2021.05.25

(74) 专利代理机构 北京安信方达知识产权代理有限公司 11262

(30) 优先权数据

16/889,033 2020.06.01 US

代理人 李琰 杨明钊

(85) PCT国际申请进入国家阶段日

2021.06.30

(51) Int.Cl.

G06F 16/2453 (2006.01)

(86) PCT国际申请的申请数据

PCT/US2021/034020 2021.05.25

(87) PCT国际申请的公布数据

W02021/247286 EN 2021.12.09

(71) 申请人 斯诺弗雷克公司

地址 美国蒙大拿州

(72) 发明人 蒂埃里·克吕安斯

伊戈尔·戴姆拉 瓦伦·加内什

普拉桑纳·拉贾佩鲁马尔 王立波

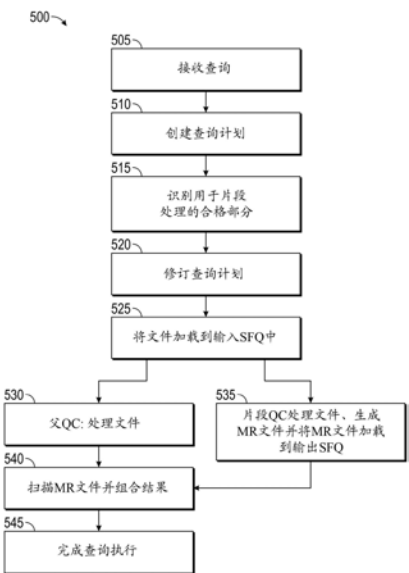
权利要求书3页 说明书19页 附图14页

(54) 发明名称

可扩展查询处理

(57) 摘要

本公开的实施例可以提供动态查询执行模型。该查询执行模型可以通过将查询的并行部分(也称为片段)向外扩展到额外的计算资源,例如从计算资源池租用的计算资源,来提供加速。查询各部分的执行可以由父查询协调器和片段查询协调器来协调,父查询协调器是查询的起源。



1. 一种方法,包括:
 - 由一个或多个处理器接收针对数据集的查询;
 - 创建执行所述查询的查询计划;
 - 基于一组标准,识别所述查询计划中的有资格进行片段处理的部分;
 - 由父查询协调器在所述数据集的第一批次文件上执行所述查询计划的识别的部分,以生成第一批次结果;
 - 由片段查询协调器在所述数据集的第二批次文件上执行所述查询的所述识别的部分,以生成实例化结果文件;
 - 扫描所述实例化结果文件以生成扫描结果;
 - 组合所述第一批次结果和所述扫描结果以生成组合结果;和
 - 对所述组合结果执行所述查询计划的剩余部分,以生成对所述查询的响应。
2. 根据权利要求1所述的方法,还包括:
 - 将所述数据集的文件作为连续扫描集加载到第一共享文件队列中;
 - 将第一组文件分组为所述第一批次,并将所述第一批次提供给所述父查询协调器;和
 - 将第二组文件分组为所述第二批次,并将所述第二批次提供给所述片段查询协调器。
3. 根据权利要求2所述的方法,还包括:
 - 串行地提供附加批次,直到所述连续扫描集中的所有文件都已被提供。
4. 根据权利要求2所述的方法,其中所述片段查询协调器将所述实例化结果文件加载到输出共享文件队列中。
5. 根据权利要求1所述的方法,其中所述片段查询协调器使用一个或多个片段计算资源来执行所述查询的所述识别的部分。
6. 根据权利要求5所述的方法,其中所述片段计算资源的数量在所述查询的执行期间改变。
7. 根据权利要求1所述的方法,其中所述一组标准包括所述识别的部分的执行输出是否小于所述识别的部分的输入。
8. 根据权利要求1所述的方法,其中所述一组标准包括所述识别的部分是否能由计算资源在不与另一计算资源通信的情况下执行。
9. 根据权利要求8所述的方法,其中所述查询计划包括多个操作符和连接所述操作符的链路,每个链路连接所述多个操作符中的第一操作符和第二操作符,并指示所述第一操作符是否能由计算资源在不与另一计算资源通信的情况下执行。
10. 根据权利要求1所述的方法,还包括:
 - 将所述实例化结果文件与至少另一个实例化结果文件聚合在一起;和
 - 其中扫描所述实例化结果文件以生成扫描结果包括扫描聚合的实例化结果文件以生成所述扫描结果。
11. 一种系统,包括:
 - 机器的一个或多个处理器;和
 - 存储器,其存储指令,当由所述一个或多个处理器执行时,所述指令使得所述机器执行操作,所述操作包括:
 - 接收针对数据集的查询;

创建执行所述查询的查询计划；

基于一组标准，识别所述查询计划中的有资格进行片段处理的部分；

由父查询协调器在所述数据集的第一批次文件上执行所述查询计划的识别的部分，以生成第一批次结果；

由片段查询协调器在所述数据集的第二批次文件上执行所述查询的所述识别的部分，以生成实例化结果文件；

扫描所述实例化结果文件以生成扫描结果；

组合所述第一批次结果和所述扫描结果以生成组合结果；和

对所述组合结果执行所述查询计划的剩余部分，以生成对所述查询的响应。

12. 根据权利要求11所述的系统，所述操作还包括：

将所述数据集的文件作为连续扫描集加载到第一共享文件队列中；

将第一组文件分组为所述第一批次，并将所述第一批次提供给所述父查询协调器；和

将第二组文件分组为所述第二批次，并将所述第二批次提供给所述片段查询协调器。

13. 根据权利要求12所述的系统，所述操作还包括：

串行地提供附加批次，直到所述连续扫描集中的所有文件都已被提供。

14. 根据权利要求12所述的系统，其中所述片段查询协调器将所述实例化结果文件加载到输出共享文件队列中。

15. 根据权利要求11所述的系统，其中所述片段查询协调器使用一个或更多个片段计算资源来执行所述查询的所述识别的部分。

16. 根据权利要求15所述的系统，其中所述片段计算资源的数量在所述查询的执行期间改变。

17. 根据权利要求11所述的系统，其中所述一组标准包括所述识别的部分的执行输出是否小于所述识别的部分的输入。

18. 根据权利要求11所述的系统，其中所述一组标准包括所述识别的部分是否能由计算资源在不与另一计算资源通信的情况下执行。

19. 根据权利要求18所述的系统，其中所述查询计划包括多个操作符和连接所述操作符的链路，每个链路连接所述多个操作符中的第一操作符和第二操作符，并指示所述第一操作符是否能由计算资源在不与另一计算资源通信的情况下执行。

20. 根据权利要求11所述的系统，所述操作还包括：

将所述实例化结果文件与至少另一个实例化结果文件聚合在一起；和

其中扫描所述实例化结果文件以生成扫描结果包括扫描聚合的实例化结果文件以生成所述扫描结果。

21. 一种包含指令的机器存储介质，当由机器执行时，所述指令使所述机器执行操作，所述操作包括：

接收针对数据集的查询；

创建执行所述查询的查询计划；

基于一组标准，识别所述查询计划中的有资格进行片段处理的部分；

由父查询协调器在所述数据集的第一批次文件上执行所述查询计划的识别的部分，以生成第一批次结果；

由片段查询协调器在所述数据集的第二批次文件上执行所述查询的所述识别的部分，以生成实例化结果文件；

扫描所述实例化结果文件以生成扫描结果；

组合所述第一批次结果和所述扫描结果以生成组合结果；和

对所述组合结果执行所述查询计划的剩余部分，以生成对所述查询的响应。

22. 根据权利要求21所述的机器存储介质，还包括：

将所述数据集的文件作为连续扫描集加载到第一共享文件队列中；

将第一组文件分组为所述第一批次，并将所述第一批次提供给所述父查询协调器；和

将第二组文件分组为所述第二批次，并将所述第二批次提供给所述片段查询协调器。

23. 根据权利要求22所述的机器存储介质，还包括：

串行地提供附加批次，直到所述连续扫描集中的所有文件都已被提供。

24. 根据权利要求22所述的机器存储介质，其中所述片段查询协调器将所述实例化结果文件加载到输出共享文件队列中。

25. 根据权利要求21所述的机器存储介质，其中所述片段查询协调器使用一个或更多个片段计算资源来执行所述查询的所述识别的部分。

26. 根据权利要求25所述的机器存储介质，其中所述片段计算资源的数量在所述查询的执行期间改变。

27. 根据权利要求21所述的机器存储介质，其中所述一组标准包括所述识别的部分的执行输出是否小于所述识别的部分的输入。

28. 根据权利要求21所述的机器存储介质，其中所述一组标准包括所述识别的部分是否能由计算资源在不与另一计算资源通信的情况下执行。

29. 根据权利要求28所述的机器存储介质，其中所述查询计划包括多个操作符和连接所述操作符的链路，每个链路连接所述多个操作符中的第一操作符和第二操作符，并指示所述第一操作符是否能由计算资源在不与另一计算资源通信的情况下执行。

30. 根据权利要求21所述的机器存储介质，还包括：

将所述实例化结果文件与至少另一个实例化结果文件聚合在一起；和

其中扫描所述实例化结果文件以生成扫描结果包括扫描聚合的实例化结果文件以生成所述扫描结果。

可扩展查询处理

[0001] 优先权申请的交叉引用

[0002] 本申请要求2020年6月1日提交的美国专利申请序列号16/889,033的优先权,其内容通过引用结合于此。

技术领域

[0003] 本公开一般涉及使用并行处理的可扩展查询处理。

[0004] 背景

[0005] 随着世界变得更加数据驱动,数据库系统和其他数据系统正在存储越来越多的数据。对于使用这些数据的企业来说,不同的操作或查询通常是在这些大量的数据上运行的。一些操作,例如那些包括大型表扫描的操作,可能要花费大量的时间来执行大量的数据。执行这种操作的时间可以与用于执行的计算资源的数量成比例,因此可以通过使用更多的计算资源来缩短时间。

[0006] 为此,一些数据系统可以提供一个计算资源池,这些资源可以被分配来执行不同的操作。然而,在这样的系统中,分配的计算资源通常协同工作(例如作为一个进程组)。因此,他们的分配是固定不变的。因为被分配的计算资源一起工作来执行操作,所以它们通常在开始时被分配它们各自的部分,例如,角色和分配被提前固定。因此,对于可用资源数量的动态变化,这些系统并不灵活。

[0007] 此外,这些系统无法跟踪所有计算资源的性能。因此,如果计算资源之一出现错误,整个作业将不得不重新执行,从而浪费时间和资源。

[0008] 附图简述

[0009] 各种附图仅示出了本公开的示例性实施例,不应被视为限制其范围。

[0010] 图1示出了根据一些示例实施例的示例计算环境,其中基于网络的数据仓库系统可以在共享数据库对象上实现流。

[0011] 图2是示出根据一些示例实施例的计算服务管理器的组件的框图。

[0012] 图3是示出根据一些示例实施例的执行平台的组件的框图。

[0013] 图4是示出根据一些示例实施例的前台全局服务的框图。

[0014] 图5示出了根据一些示例实施例的用于执行查询的流程图。

[0015] 图6示出了根据一些示例实施例的查询计划的示例。

[0016] 图7示出了根据一些示例实施例的修订的查询计划的示例。

[0017] 图8是根据一些示例实施例的查询处理系统的框图。

[0018] 图9示出了根据一些示例实施例的输入到输出映射的示例。

[0019] 图10是根据一些示例实施例的片段处理系统的框图。

[0020] 图11示出了根据一些示例实施例的使用检查点(checkpoint)的片段计划的示例。

[0021] 图12是根据一些示例实施例的片段处理环境的框图。

[0022] 图13示出了根据一些示例实施例的用于跟踪批处理的流程图。

[0023] 图14示出了根据本公开一些实施例的以计算机系统形式的机器的示意图表示,用

于使机器执行本文所讨论方法中的任何一个或更多的指令集合可以在该计算机系统内执行。

[0024] 详细描述

[0025] 以下描述包括体现本公开的说明性实施例的系统、方法、技术、指令序列和计算机程序产品。在以下描述中,出于解释的目的,阐述了许多具体细节,以便提供对本发明主题的各种实施例的理解。然而,对于本领域技术人员来说,很明显,本发明主题的实施例可以在没有这些具体细节的情况下实施。通常,众所周知的指令实例、协议、结构和技术不一定详细示出。

[0026] 本公开的实施例可以提供动态查询执行模型。该查询执行模型可以通过将查询的并行部分(也称为片段)向外扩展到额外的计算资源,例如从计算资源池租用的计算资源,来提供加速。查询各部分的执行可以由父查询协调器(查询起源的地方)和片段查询协调器来协调。用于查询的文件可以作为连续扫描集加载到共享文件队列中,当父查询协调器和片段协调器完成处理它们当前分配的批次时,它们可以向该连续扫描集串行地(例如,一次一个)请求文件批次。片段计算资源可以生成实例化结果(materialized results),并将它们加载到另一个共享文件队列中。在连续扫描集中的所有文件被处理之后,实例化结果可以被父查询协调器使用。因此,这种查询执行模型提供了更快的速度和更大的灵活性,尤其是当资源的数量在查询执行过程中可以动态变化时。

[0027] 图1示出了根据本公开的一些实施例的实现部署之间的安全消息传递的示例共享数据处理平台100。为了避免用不必要的细节模糊本发明主题,图中省略了与传达对本发明主题的理解不密切相关的各种功能组件。然而,本领域技术人员将容易认识到,各种附加功能组件可以被包括作为共享数据处理平台100的一部分,以促进本文没有具体描述的附加功能。

[0028] 如图所示,共享数据处理平台100包括基于网络的数据仓库系统102、云计算存储平台104(例如,存储平台、AWS®服务、Microsoft Azure®或Google Cloud Services®)和远程计算设备106。基于网络的数据仓库系统102是基于网络的系统,用于以集成方式存储和访问数据(例如,内部存储数据,访问外部远程数据),以及报告和分析来自一个或多个不同源(例如,云计算存储平台104)的集成数据。云计算存储平台104包括多个计算机器,并向基于网络的数据仓库系统102按需提供计算机系统资源,如数据存储和计算能力。虽然在图1所示的实施例中,描绘了数据仓库,但是其他实施例可以包括其他类型的数据库或其他数据处理系统。

[0029] 远程计算设备106(例如,诸如膝上型计算机的用户设备)包括一个或多个计算机器(例如,诸如膝上型计算机的用户设备),其执行远程软件组件108(例如,浏览器访问的云服务)以向基于网络的数据仓库系统102的用户提供附加功能。远程软件组件108包括一组机器可读指令(例如,代码),当由远程计算设备106执行时,该指令使得远程计算设备106提供某些功能。远程软件组件108可以对输入数据进行操作,并基于处理、分析或以其他方式转换输入数据来生成结果数据。作为示例,远程软件组件108可以是数据提供者或数据消费者,其实现数据库跟踪过程,例如共享表和视图上的流,如下面进一步详细讨论的。

[0030] 基于网络的数据仓库系统102包括访问管理系统110、计算服务管理器112、执行平台114和数据库116。访问管理系统110使得管理用户能够管理对基于网络的数据仓库系统

102所提供的资源和服务的访问。管理用户可以创建和管理用户、角色和组,并使用权限来允许或拒绝对资源和服务的访问。访问管理系统110可以存储共享数据,该共享数据在基于网络的数据仓库系统102的不同用户之间安全地管理对云计算存储平台104的存储资源的共享访问,如下面进一步详细讨论的。

[0031] 计算服务管理器112协调和管理基于网络的数据仓库系统102的操作。计算服务管理器112还执行查询优化和编译,以及管理提供计算资源的计算服务集群(例如,虚拟仓库、虚拟机、EC2集群)。计算服务管理器112可以支持任意数量的客户端账户,如提供数据存储和检索请求的终端用户、管理本文描述的系统和方法的系统管理员以及与计算服务管理器112交互的其他组件/设备。

[0032] 计算服务管理器112还耦合到数据库116,数据库116与存储在共享数据处理平台100上的全部数据相关联。数据库116存储数据,该数据涉及与基于网络的数据仓库系统102及其用户相关联的各种功能和方面。

[0033] 在一些实施例中,数据库116包括存储在远程数据存储系统中的数据以及可从一个或多个本地高速缓存获得的数据的概要。另外,数据库116可以包括有关在远程数据存储系统和本地高速缓存中如何组织数据的信息。数据库116允许系统和服务确定是否需要访问数据段,而无需从存储设备加载或访问实际数据。计算服务管理器112进一步耦合到执行平台114,执行平台114提供执行各种数据存储和数据检索任务的多个计算资源(例如,虚拟仓库),如下面更详细地讨论的。

[0034] 执行平台114耦合到作为云计算存储平台104的一部分的多个数据存储设备124-1至124-n。在一些实施例中,数据储存设备124-1到124-n是位于一个或多个地理位置的基于云的储存设备。例如,数据存储设备124-1到124-n可以是公共云基础设施或私有云基础设施的一部分。数据存储设备124-1到124-n可以包括硬盘驱动(HDD)、固态驱动(SSD)、存储集群、亚马孙(Amazon) S3存储系统或任何其它数据存储技术。另外,云计算存储平台104可以包括分布式文件系统(诸如,Hadoop分布式文件系统(HDFS))、对象存储系统等。

[0035] 执行平台114包括多个计算节点(例如,虚拟仓库)。计算节点上的进程集合执行由计算服务管理器112编译的查询计划。进程集合可以包括:执行查询计划的第一进程;使用最近最少使用(least recently used,LRU)策略来监视和删除微分区文件,并实现内存不足(OOM)错误缓解进程的第二进程;从进程日志和状态信息提取健康信息以发送回计算服务管理器112的第三进程;在系统引导(boot)之后建立与计算服务管理器112的通信的第四进程;以及第五进程,该第五进程处理与由计算服务管理器112提供的给定作业的计算集群的所有通信,并将信息传送回计算服务管理器112和执行平台114的其他计算节点。

[0036] 云计算存储平台104还包括访问管理系统118和web代理120。如同访问管理系统110一样,访问管理系统118允许用户创建和管理用户、角色和组,并使用许可来允许或拒绝对云服务和资源的访问。基于网络的数据仓库系统102的访问管理系统110和云计算存储平台104的访问管理系统118可以通信和共享信息,以便能够访问和管理由基于网络的数据仓库系统102和云计算存储平台104的用户共享的资源和服务。web代理120处理涉及接受和处理并发API呼叫的任务,包括流量管理、授权和访问控制、监控和API版本管理。web代理120提供HTTP代理服务,用于创建、发布、维护、保护和监控API(例如,REST API)。

[0037] 在一些实施例中,共享数据处理平台100的元件之间的通信链路通过一个或多个

个数据通信网络来实现。这些数据通信网络可以利用任何通信协议和任何类型的通信介质。在一些实施例中,数据通信网络是彼此耦合的两个或更多个数据通信网络(或子网络)的组合。在可替代的实施例中,这些通信链路被使用任何类型的通信介质和任何通信协议来实现。

[0038] 如图1所示,数据存储设备124-1至124-N从与执行平台114相关联的计算资源解耦。也就是说,可以在执行平台114中创建和终止新的虚拟仓库,并且可以以独立的方式在云计算存储平台104上创建和终止附加的数据存储设备。该体系结构支持基于网络的数据仓库系统102基于变化的数据存储/检索需求以及访问共享数据处理平台100的用户和系统的变化需求的动态变化。对动态变化的支持允许基于网络的数据仓库系统102响应于对基于网络的数据仓库系统102内的系统和组件的不断变化的需求而快速变化规模(scale)。计算资源与数据存储设备124-1到124-n的分离支持大量数据的存储,而无需相应的大量计算资源。类似地,资源的这种分离支持在特定时间使用的计算资源的显著增加,而无需相应增加可用数据存储资源。此外,资源的解耦使不同的账户能够处理创建额外的计算资源,以处理由其他用户共享的数据,而不会影响其他用户的系统。例如,数据提供者可以具有三个计算资源,并且与数据消费者共享数据,并且数据消费者可以生成新的计算资源来对共享数据执行查询,其中新的计算资源由数据消费者管理,并且不影响数据提供者的计算资源或者与数据提供者的计算资源交互。

[0039] 计算服务管理器112、数据库116、执行平台114、云计算存储平台104和远程计算设备106在图1中显示为单独的组件。然而,计算服务管理器112、数据库116、执行平台114、云计算存储平台104和远程计算环境中的每一个可以被实现为由API连接并访问信息(例如,令牌、登录数据)的分布式系统(例如,分布在多个地理位置的多个系统/平台上)。此外,计算服务管理器112、数据库116、执行平台114和云计算存储平台104中的每一个都可以接收到的请求的变化和共享数据处理平台100的需求的变化而扩大或缩小(彼此独立)。因此,在所描述的实施例中,基于网络的数据仓库系统102是动态的,并且支持频繁变化以满足当前的数据处理需求。

[0040] 在典型操作期间,基于网络的数据仓库系统102处理由计算服务管理器112确定的多个作业(例如,查询)。这些作业由计算服务管理器112调度和管理,以确定何时以及如何执行该作业。例如,计算服务管理器112可以将作业划分为多个离散任务,并且可以确定需要什么数据来执行多个离散任务中的每个离散任务。计算服务管理器112可以将多个离散任务中的每个离散任务分配给执行平台114的一个或更多个节点以处理任务。计算服务管理器112可以确定需要哪些数据来处理任务,并且进一步确定执行平台114内的哪些节点最适合于处理该任务。一些节点可能已经缓存了处理任务所需的数据(由于节点最近已经从云计算存储平台104下载了用于先前作业的数据),因此是处理任务的良好候选。存储在数据库116中的元数据帮助计算服务管理器112确定执行平台114中的哪些节点已经缓存了处理该任务所需的至少一部分数据。执行平台114中的一个或更多个节点使用由节点高速缓存的数据以及必要时从云计算存储平台104检索到的数据来处理任务。期望从执行平台114内的高速缓存中检索尽可能多的数据,因为检索速度通常比从云计算存储平台104中检索数据快得多。

[0041] 如图1所示,共享数据处理平台100将执行平台114与云计算存储平台104分开。在

这种布置中,执行平台114中的处理资源和高速缓存资源独立于云计算存储平台104中的数据存储设备124-1到124-n进行操作。因此,计算资源和高速缓存资源不限于具体的数据存储设备124-1到124-n。相反,所有计算资源和所有高速缓存资源可以从存储平台104中的任何数据存储资源检索数据,并将数据存储到其中。

[0042] 图2是示出根据本公开一些实施例的计算服务管理器112的组件的框图。如图2所示,请求处理服务202管理接收到的数据存储请求和数据检索请求(例如,将对数据库数据执行的作业)。例如,请求处理服务202可以确定处理接收到的查询(例如,数据存储请求或数据检索请求)所需的数据。数据可以存储在执行平台114内的高速缓存中,或者存储在云计算存储平台104中的数据存储设备中。管理控制台服务204支持管理员和其他系统管理员对各种系统和进程的访问。另外,管理控制台服务204可以接收执行作业并监视系统上的工作负荷的请求。根据一些示例实施例,流共享引擎225管理对诸如数据共享(例如,共享表)或共享视图的数据库对象的改变跟踪,如下面进一步详细讨论的。

[0043] 计算服务管理器112还包括作业编译器206、作业优化器208和作业执行器210。作业编译器206将作业解析为多个离散任务,并为多个离散任务中的每个离散任务生成执行代码。作业优化器208基于需要处理的数据来确定执行多个离散任务的最佳方法。作业优化器208还处理各种数据修剪操作和其他数据优化技术,以提高执行作业的速度和效率。作业执行器210执行从队列接收的或由计算服务管理器112确定的作业的执行代码。

[0044] 作业调度器和协调器212将接收到的作业发送到适当的服务或系统,以进行编译、优化并分派到执行平台114。例如,可以对作业进行优先级排序并按该优先级顺序进行处理。在实施例中,作业调度器和协调器212确定由计算服务管理器112调度的内部作业与其他“外部”作业(诸如可以由数据库中的其他系统调度但可以利用执行平台114中的相同处理资源的用户查询)的优先级。在一些实施例中,作业调度器和协调器212识别或分配执行平台114中的特定节点以处理特定任务。虚拟仓库管理器214管理在执行平台114中实现的多个虚拟仓库的操作。如下所述,每个虚拟仓库包括多个执行节点,每个执行节点包括高速缓存和处理器(例如,虚拟机、操作系统级容器执行环境)。

[0045] 另外,计算服务管理器112包括配置和元数据管理器216,其管理与存储在远程数据存储设备和本地高速缓存(即,执行平台114中的高速缓存)中的数据有关的信息。配置和元数据管理器216使用元数据来确定需要访问哪些数据微分区来检索用于处理特定任务或作业的数据。监视器和工作负荷分析器218监督由计算服务管理器112执行的进程,并管理跨执行平台114中的虚拟仓库和执行节点的任务(例如,工作负荷)的分配。监视器和工作负荷分析器218还根据需要,基于整个基于网络的数据仓库系统102中不断变化的工作负载来重新分配任务,并且还可以基于也可以由执行平台114处理的用户(例如,“外部”)查询工作负载来重新分配任务。配置和元数据管理器216以及监视器和工作负荷分析器218耦合到数据存储设备220。图2中的数据存储设备220表示基于网络的数据仓库系统102中的任何数据存储设备。例如,数据存储设备220可以表示执行平台114中的高速缓存、云计算存储平台104中的存储设备或任何其他存储设备。

[0046] 图3是示出根据本公开一些实施例的执行平台114的组件的框图。如图3所示,执行平台114包括多个虚拟仓库,它们是计算实例的弹性集群,例如虚拟机。在所示的示例中,虚拟仓库包括虚拟仓库1、虚拟仓库2和虚拟仓库n。每个虚拟仓库(例如,EC2集群)包括多个执

行节点(例如,虚拟机),每个执行节点包括数据高速缓存和处理器。虚拟仓库可以通过使用多个执行节点并行执行多个任务。如本文中所讨论的,执行平台114可以基于系统和用户的当前处理需求实时地添加新的虚拟仓库并丢弃现有的虚拟仓库。这种灵活性允许执行平台114在需要时快速部署大量计算资源,而不必在不再需要它们时继续为那些计算资源付费。所有虚拟仓库都可以访问任何数据存储设备(例如,云计算存储平台104中的任何存储设备)中的数据。

[0047] 尽管图3中所示的每个虚拟仓库包括三个执行节点,但是特定的虚拟仓库可以包括任意数量的执行节点。此外,虚拟仓库中执行节点的数量是动态的,使得在存在额外需求时创建新的执行节点,并在不再需要现有执行节点时将其删除(例如,在查询或作业完成时)。

[0048] 每个虚拟仓库能够访问图1中所示的数据存储设备124-1到124-n中的任何一个。因此,虚拟仓库没有必要被分配给特定的数据存储设备124-1到124-n,而是可以从云计算存储平台104内的数据存储设备124-1到124-n中的任何一个访问数据。类似地,图3所示的每个执行节点可以从数据存储设备124-1至124-n中的任何一个访问数据。例如,第一用户(例如,提供商账户用户)的存储设备124-1可以与另一用户(例如,消费者账户用户)的虚拟仓库中的工作器节点共享,使得另一用户可以创建数据库(例如,只读数据库)并直接使用存储设备124-1中的数据,而不需要复制数据(例如,将其复制到由消费者账户用户管理的新磁盘)。在一些实施例中,可以将特定的虚拟仓库或特定的执行节点临时分配给特定的数据存储设备,但是该虚拟仓库或执行节点可以稍后从任何其他数据存储设备访问数据。

[0049] 在图3的示例中,虚拟仓库1包括三个执行节点302-1、302-2和302-n。执行节点302-1包括高速缓存304-1和处理器306-1。执行节点302-2包括高速缓存304-2和处理器306-2。执行节点302-n包括高速缓存304-n和处理器306-n。每个执行节点302-1、302-2和302-n与处理一个或多个数据存储和/或数据检索任务相关联。例如,虚拟仓库可以处理与内部服务(诸如,聚类服务、实例化视图刷新服务、文件压缩服务、存储程序服务或文件升级服务)相关联的数据存储和数据检索任务。在其他实现中,特定虚拟仓库可以处理与特定数据存储系统或特定类别的数据相关联的数据存储和数据检索任务。

[0050] 类似于上面讨论的虚拟仓库1,虚拟仓库2包括三个执行节点312-1、312-2和312-n。执行节点312-1包括高速缓存314-1和处理器316-1。执行节点312-2包括高速缓存314-2和处理器316-2。执行节点312-n包括高速缓存314-n和处理器316-n。另外,虚拟仓库3包括三个执行节点322-1、322-2和322-n。执行节点322-1包括高速缓存324-1和处理器326-1。执行节点322-2包括高速缓存324-2和处理器326-2。执行节点322-n包括高速缓存324-n和处理器326-n。

[0051] 在一些实施例中,相对于执行节点正在高速缓存的数据,图3所示的执行节点是无状态的。例如,这些执行节点不存储或以其他方式维护有关执行节点的状态信息或由特定执行节点高速缓存的数据。因此,在执行节点发生故障的情况下,可以用另一个节点透明地替换发生故障的节点。由于没有与发生故障的执行节点相关联的状态信息,因此新的(替换)执行节点可以轻松替换发生故障的节点,而无需考虑重新创建特定状态。

[0052] 尽管图3所示的执行节点各自包括一个数据高速缓存和一个处理器,但是替代实施例可以包括包含任意数量的处理器和任意数量的高速缓存的执行节点。另外,高速缓存

的大小在不同执行节点之间可能有所不同。图3所示的高速缓存在本地执行节点(例如,本地磁盘)中存储从云计算存储平台104中的一个或多个数据存储设备检索的数据(例如,给定节点最近访问的S3对象)。在一些示例实施例中,当查询仅下载该查询所需的列时,高速缓存存储文件头和文件的各个列。

[0053] 为了提高高速缓存命中并避免重叠存储在节点高速缓存中的冗余数据,作业优化器208使用一致性散列方案将输入文件集分配给节点,以散列所访问的数据(例如,数据库116或数据库122中的数据)的表文件名。根据一些示例实施例,访问同一表文件的后续或并发查询因此将在同一节点上执行。

[0054] 如所讨论的,节点和虚拟仓库可以响应于环境条件(例如,灾难场景)、硬件/软件问题(例如,故障)或管理变化(例如,从大集群改变到小集群以降低成本)而动态改变。在一些示例实施例中,当节点集改变时,没有数据被立即重新混洗。相反,实施最近最少使用的替换策略来最终替换多个作业中丢失的缓存内容。因此,高速缓存减小或消除了从远程存储系统中不断检索数据的平台中出现的瓶颈问题。本文描述的系统和方法不是从远程存储设备重复访问数据,而是从执行节点中的高速缓存访问数据,这明显更快并且避免了上面讨论的瓶颈问题。在一些实施例中,使用提供对高速缓存的数据的快速访问的高速存储器设备来实现高速缓存。每个高速缓存可以存储来自云计算存储平台104中的任何存储设备的数据。

[0055] 此外,高速缓存资源和计算资源可以在不同的执行节点之间变化。例如,一个执行节点可包含大量的计算资源和最少的高速缓存资源,从而使该执行节点可用于需要大量计算资源的任务。另一个执行节点可包含大量的高速缓存资源和最少的计算资源,从而使该执行节点可用于需要高速缓存大量数据的任务。又一个执行节点可包含提供更快输入-输出操作的高速缓存资源,这对于需要快速扫描大量数据的任务很有用。在一些实施例中,执行平台114实现偏斜处理,以在与特定执行相关联的高速缓存资源和计算资源之间分配工作,其中分配可以进一步基于由执行节点执行的预期任务。例如,如果执行节点执行的任务变得更加处理器密集,则可以为执行节点分配更多的处理资源。类似地,如果执行节点执行的任务需要更大的高速缓存容量,则可以为执行节点分配更多的高速缓存资源。此外,由于各种问题(例如,虚拟化问题、网络开销),一些节点的执行速度可能比其他节点慢得多。在一些示例实施例中,使用文件窃取方案在扫描级别解决不平衡。特别是,每当节点进程完成扫描其输入文件集时,它都会向其他节点请求额外的文件。如果其他节点中的一个接收到这样的请求,则该节点分析其自己的集合(例如,当接收到该请求时,在输入文件集合中还剩多少文件),然后在当前作业(例如,查询)的持续时间内转移一个或多个剩余文件的所有权。请求节点(例如,文件窃取节点)然后接收数据(例如,头部数据)并从云计算存储平台104(例如,从数据存储设备124-1)下载文件,并且不从传输节点下载文件。这样,滞后节点可以通过文件窃取来传输文件,而不会加重滞后节点的负载。

[0056] 尽管虚拟仓库1、2和n与同一执行平台114相关联,但是可以使用在多个地理位置处的多个计算系统来实现虚拟仓库。例如,虚拟仓库1可以由第一地理位置处的计算系统来实现,而虚拟仓库2和n由第二地理位置处的另一计算系统来实现。在一些实施例中,这些不同的计算系统是由一个或多个不同的实体维护的基于云的计算机系统。

[0057] 另外,每个虚拟仓库在图3中被示为具有多个执行节点。可以使用位于多个地理位

置处的多个计算系统来实现与每个虚拟仓库相关联的多个执行节点。例如,虚拟仓库1的实例在一个地理位置处的一个计算平台上实现执行节点302-1和302-2,并且在另一个地理位置处的不同计算平台上实现执行节点302-n。选择特定的计算系统以实施执行节点可以取决于各种因素,诸如特定的执行节点所需的资源水平(例如,处理资源要求和高速缓存要求)、特定计算系统处可用的资源、地理位置内或地理位置之间的网络的通信能力以及哪些计算系统已在虚拟仓库中实现其他执行节点。

[0058] 执行平台114也是容错的。例如,如果一个虚拟仓库发生故障,那么该虚拟仓库将很快被位于不同地理位置处的不同虚拟仓库替换。

[0059] 特定执行平台114可以包括任意数量的虚拟仓库。另外,特定执行平台中虚拟仓库的数量是动态的,使得在需要附加的处理和/或高速缓存资源时创建新的虚拟仓库。类似地,当不再需要与虚拟仓库关联的资源时,可以删除现有的虚拟仓库。

[0060] 在一些实施例中,虚拟仓库可以在云计算存储平台104中的相同数据上操作,但是每个虚拟仓库具有其自己的具有独立处理和高速缓存资源的执行节点。此配置允许对不同虚拟仓库上的请求进行独立处理,并且请求之间不会相互干扰。这种独立的处理与动态添加和移除虚拟仓库的能力相结合,支持为新用户添加新的处理能力,而不会影响现有用户观察到的性能。

[0061] 图4示出了根据一些示例实施例的示例前台全局服务(GS) 400。前台GS 400可以接收查询请求并开发查询计划来执行查询请求。前台GS 400可以代理对执行查询计划的计算节点或资源的请求,如这里进一步详细解释的。前台GS 400可以包括耦合到本地后台服务(BG) 404的查询协调器(QC) 402.1-402.3。在一个实施例中,前台GS 400可以被定义用于特定类型的服务,例如复制、摄取、计算、大表扫描等。QC 402.1-402.3可以从不同的源接收查询请求,这些源可能具有不同的账户ID。对于某些操作,例如那些涉及多个计算资源一起工作来执行操作的不同部分的操作(例如,大型表扫描),可以在数据仓库级粒度定义源。QC 402.1-402.3可以向本地BG 404传送关于查询请求及其源的信息。

[0062] 如下面进一步详细解释的,本地BG 404可以将计算资源(有时也称为执行平台(XP))分配给QC 402.1-402.3。计算资源可以是从计算资源池中分配给前台GS 400的计算节点。在一个实施例中,计算资源可以是机器、服务器和/或处理器。在一个实施例中,计算资源可以是机器的处理核心。一旦接收到它们的计算资源分配,QC 402.1-402.3可以直接与分配的计算资源通信,以执行各自的查询计划。

[0063] 即使在执行查询期间,分配的计算资源的数量也可能动态变化。例如,可以使用2020年5月14日提交的题为“Flexible Computing”的美国专利申请第16/874,388号中描述的技术来动态分配分配的计算资源的数量,该申请通过引用整体结合于此,包括但不限于在下文中具体出现的那些部分,通过引用结合有以下例外:在上述引用的申请的任何部分与本申请不一致的情况下,本申请取代上述引用的申请。

[0064] 图5示出了根据一些示例实施例的由前台GS执行查询的方法500的流程图。在一个实施例中,前台GS内的QC可以执行方法500的部分。在操作505,QC或前台GS可以接收要在数据集(例如表)上运行的查询请求。接收查询的QC可以被称为父QC。在操作510,父QC可以创建查询计划来执行接收到的查询。查询计划可以包括多个操作符和连接操作符的链路;这些链路可以定义如何将一个操作符的结果传递给下一个操作符。

[0065] 图6示出了根据一些示例实施例的示例查询计划600。仅为了说明的目的,查询计划600被示出用于过滤器聚合查询类型,并且查询计划600可以是如下所述的其他类型的查询。查询计划600可以包括多个操作符来执行查询:表扫描操作符605、过滤器操作符610、本地聚合操作符615、最终聚合操作符620和结果操作符625。操作符可以通过链路连接,在示例中显示为“L”、“X”和“S”类型。如图所示,表扫描操作符605的结果可以使用本地链路(“L”链路)传送到过滤器操作符610。也就是说,计算资源可以执行表扫描操作符605、过滤器操作符610和本地聚合操作符615,而不与另一计算资源通信。相反,本地聚合操作符615的结果可以使用与最终聚合操作符620的交换(“X”链路)来传送。并且最终聚合操作符610的结果可以使用单个链路(“S”链路)传送到结果操作符625。交换和单个链路都可能需要计算资源之间的通信;因此,在不与其他资源通信的情况下,这些连接的操作符可能无法由独立的计算资源来执行。

[0066] 返回到图5的方法500,在创建查询计划之后,可以在操作515识别查询计划中的用于片段处理的合格部分。片段是指查询计划中可以使用额外计算资源并行(和隔离)执行的部分,即查询计划的可并行部分。可以使用一组标准来标识查询计划中可能有资格进行片段处理的一个或更多个部分。在一个实施例中,可以包括一个操作符或一组操作符、仅具有本地链路的计划块可能有资格进行片段处理。此外,该操作符或操作符组可以从表扫描操作开始,以适合片段处理。因为只有本地链路连接所标识的操作,所以计算资源可以执行那些所标识的操作,而不与其他资源通信,从而使其有资格进行片段处理。例如,那些操作可以包括可以由计算资源在数据库、数据集、微分区等中的数据上本地执行的操作。这种操作符的例子可以包括但不限于表扫描、过滤器、子聚合、投影、布隆过滤器等。

[0067] 考虑图6的查询计划600的例子。在那里,前三个操作符(表扫描操作符605、过滤器操作符610和本地聚合操作符615)可能有资格进行片段处理,因为它们通过本地链路连接,即它们可以由计算资源独立执行,而不与其他计算资源通信,并且以表扫描操作符开始。

[0068] 片段处理资格(fragment processing eligibility)的另一个标准可以是提高处理速度。也就是说,该标准可以基于与正常执行(即,没有片段处理)相比,使用具有额外计算资源的片段处理是否会产生更快的执行。在一个实施例中,该确定可以不依赖于额外计算资源的最小数量,因为该数量可以是动态的,并且在如上所述的查询执行期间改变。由于计算资源数量的动态性质,查询处理的速度可能不是静态的或定义的,而是取决于可用计算资源的数量。因此,根据执行时可用计算资源的数量,使用片段处理执行相同查询的时间可以在不同的时间变化。

[0069] 片段处理资格的其他标准也可以用于确定片段处理的资格,例如表扫描的大小、输出与输入的比率、租用计算资源的可用性等等。更大的表扫描可能更适合片段处理。输出与输入的比率是指片段处理的输出相对于片段处理的输入的大小。如下所述,计算资源的输出不直接流水线化到父QC中,而是可以以实例化结果(materialized result)的形式提供,父扫描(例如,父作业执行平台机器)该实例化结果。因此,输出可以指实例化结果的大小,输入可以指用于生成实例化结果的输入文件的大小。输出与输入的比率越小,查询计划中的该部分用于片段处理可能就越合适。

[0070] 在操作520,可以修订查询计划,以便为原始查询计划的已识别操作添加片段处理。查询可能包含多个片段。为了清楚和简洁,接下来描述具有单个片段的查询计划的示例

情况,但是应该理解,这些教导可以扩展到查询计划中的多个片段。

[0071] 图7示出了根据一些示例实施例的经修订的查询计划700,其具有针对查询计划600的片段处理。如上所述,操作符605-610可能已经被识别用于片段处理。因此,对于某些文件或文件批次,操作符605-610可以由父QC执行,如上所述。这些操作也可以由片段QC使用额外的计算资源(例如,租用的计算资源)同时执行。父QC和片段QC可以作为不同的QC提供,但是可以扫描相同的表(例如,操作符605)。他们可以扫描表的不同方面或部分,如下所述。片段QC可以接收和执行修订的查询计划的片段步骤(例如,操作符605-615)。

[0072] 这些操作符通过片段处理的输出可以作为实例化结果提供,并且可以被存储,如操作705所示。例如,实例化结果可以存储在诸如云存储部的存储位置,并且实例化结果文件列表可以存储在存储器(例如,输出共享文件队列)中。这样,片段计算资源可以不需要与父QC或彼此通信。片段计算资源可以不直接链接到父QC,因此它们的操作可以独立于其他组件。

[0073] 经修订的查询计划700还可以包括:扫描操作符710,用于通过父QC扫描存储的实例化结果;本地聚合操作符715,用于聚合扫描的实例化结果;以及用于将由父QC执行的操作符605-615的结果和本地聚合715的结果(例如,聚合的扫描实例化结果)结合起来的联合所有(union all)操作720。在一个实施例中,联合所有操作720可以首先消耗来自左分支(操作符605-615)的结果,然后,它可以消耗右分支(操作符705-715)的结果。通过与父查询相关的左分支开始,然后切换到处理来自右分支的实例化结果,经修订的查询计划确保连续扫描集中的所有文件都得到处理,并且使用片段处理不会有性能损失。

[0074] 此外,首先在操作符715本地聚合实例化结果可以缩减联合所有操作符720所需的数据量。因此,这种本地聚合可以提供另一个缩减数据的机会,因为片段处理可以具有小的缩减因子。修订的查询计划700还可以包括查询计划600的剩余部分,例如,如上所述的最终聚合操作符620和结果操作符625。

[0075] 再次返回图5中的方法500,然后可以执行经修订的查询计划。在操作525,来自数据集的文件可以作为连续扫描集被加载到输入共享文件队列(SFQ)中,以便于父和片段处理。输入SFQ可以被提供在可由父QC和片段QC访问的存储器中。如下面进一步详细描述(参考图8),在操作530和535,输入SFQ可以将来自连续扫描集的文件提供给父和片段QC,用于以串行方式处理,例如一次一个或更多个批次。使用批次,即一组文件,可以减少来自片段计算资源的REST调用量。父QC可以使用其分配的计算资源(父计算资源)来处理其分配的文件,片段QC可以使用租用的计算资源(片段计算资源)来处理其分配的文件。

[0076] 来自输入SFQ的文件分配可以使用连续扫描集而不是固定分配来批量执行。也就是说,父和片段QC(以及它们的计算资源)都可以耦合到输入SFQ,并且当它们可用时(例如,当它们已经完成处理它们当前分配的批次时),父和片段QC中的每一个的计算资源可以向输入SFQ请求一批次的文件(例如,一组文件)。父QC可以与其分配的计算资源协调以执行分配的批次,并且片段QC可以与其分配的计算资源协调以执行其分配的批次。输入SFQ可以将下一批次分配给相应QC的请求计算资源。例如,QC的计算资源可以在处理完当前批次后并且当它有能力处理下一个批次时,请求下一个批次。例如,SFQ可以将一批次的文件分配给父QC和片段QC中的每一个的相应计算资源,然后根据请求将后续批次分配给每个QC的计算资源,直到连续扫描集中的所有文件都被分配。

[0077] 这种批处理技术提供了灵活性,因为在查询执行期间,片段计算资源的数量可能会增加或减少。与固定分配技术相反,这种批处理技术不依赖于在整个查询执行期间所有计算资源的可用性。例如,在查询执行期间片段计算资源变得不可用的情况下,父QC(及其计算资源)可以继续处理连续扫描集中的文件。也就是说,如果片段QC(及其计算资源)不能再处理任何批次,父QC(及其计算资源)可以继续其批处理(一次一个),直到连续扫描集中的所有文件都被处理。因此,执行的结果将保持不变,但是当片段计算资源在查询执行期间丢失时(与片段计算资源可用时的时间相比),执行查询的时间可能会增加。

[0078] 此外,在操作535,片段处理的结果可以作为实例化结果被提供,其被存储在存储区域(例如,云存储部)中,并且实例化结果的文件列表(例如,URL)可以被存储在输出SFQ中,在输出SFQ中它们可被父QC访问。在操作540,在父和/或片段QC的计算资源已经处理了整个连续扫描集(例如,输入SFQ中的文件)之后,父QC计算资源可以读取或扫描实例化结果,处理扫描的实例化结果,并且将它们与由父QC处理的文件的批次结果组合(例如,图7中修改的查询计划的操作符705-720)。在操作545,父QC可以完成查询执行的协调并提供结果。例如,父QC可以协调修订的查询计划的任何剩余部分(例如,图7的操作符620和625)在组合结果上的执行。

[0079] 图8示出了根据一些示例实施例的查询处理系统800的示例。查询处理系统800可以包括父QC 805、片段QC 810、输入SFQ 815和输出SFQ 820。在一个实施例中,输入和输出SFQ 815、820可以在可由父QC 805和片段QC 810访问的本地存储器中提供。

[0080] 父QC 805可以如本文所述的创建使用片段处理的查询计划。父QC 805可以与其分配的计算资源相协调,以根据查询计划执行查询。片段QC 810可以与其分配的计算资源相协调,以执行查询计划的片段部分。

[0081] 用于查询的扫描集可以被分成文件,并且这些文件可以作为连续的扫描集存储在输入SFQ 815中。父QC 805和片段QC 810可以向输入SFQ 815请求文件进行处理。作为响应,输入SFQ 815可以将一组文件组合成一批次,为该批次分配唯一的批次ID,然后将该批次分配或转移给请求QC。

[0082] 在图8所示的例子中,父QC 805可能已经从输入共享队列815接收到批次101;批次101包括文件 T_{10} 和 T_{11} 。当父QC 805完成处理批次101并且(其计算资源)为该批次生成结果时,它可以向输入SFQ 815请求更多文件。此时,输入SFQ 815可以将一批次中的另一组文件与其自己唯一的批次ID组合,并将该批次分配或转移到父QC 805。

[0083] 同样,在图8所示的例子中,片段QC 810可能已经从输入SFQ 815接收到批次102;批次102包括文件 T_{13} 和 T_{14} 。当片段QC 810完成处理批次102时,它可以向输入SFQ 815请求更多文件。此时,输入SFQ 815可以将一批次中的另一组文件与其自己唯一的批次ID组合,并将该批次分配或转移到片段QC 810。父QC和片段QC串行地请求和处理批次的过程可以继续,直到连续扫描集中的所有文件都被处理。

[0084] 然而,在分配给片段QC 810的计算资源(租用的计算资源)已经被撤销的情况下,片段QC可以允许该计算资源完成其当前批次的处理,然后可以释放被撤销的计算资源。因此,片段QC 810可能不会向输入SFQ 815请求更多文件用于该计算资源,并且如果没有更多可用的片段计算资源,父QC 805可能是请求和处理连续扫描集中的文件的唯一实体。如果在查询仍在被处理的同时,额外的计算资源被重新分配给片段QC 810,那么它可以向输入

SFQ 815请求文件,并且可以继续其与父QC 805的并行操作,如上所述。因此,当可用计算资源的数量在查询执行期间可能变化时,在输入SFQ中使用连续扫描集的这种方案在文件处理中提供了鲁棒性和灵活性。

[0085] 与处理其分配的批次并生成批次结果的父QC 805不同,片段QC 810可以处理其分配的批次并生成实例化结果(存储在存储区域中),并将实例化结果(MR)文件列表发送到输出共享队列820。每个MR文件可能对应单个批次,一个MR文件可能不包含不同批次的结果。这允许跟踪批次处理,并且还允许在出现错误的情况下重复处理特定的批次,如下面进一步详细描述的。但是,一个输入批次可能会导致输出几个MR文件。例如,如果一个输入批次很大,并且输出不适合一个MR文件,则输出可能被分成几个MR文件。

[0086] 父QC 805可以访问在输出SFQ 820中识别的实例化结果。父QC 805通过其计算资源可以处理实例化结果(例如,表扫描),并且可以将它们与来自输入SFQ 815的文件的处理结果(例如,批次结果)组合,以完成查询的执行。在一个实施例中,在连续扫描集中的所有文件被处理之后,父QC 805的计算资源可以开始扫描在输出SFQ 820中识别的实例化结果。

[0087] 片段处理的另一个好处是它可以提供更多的容错能力和更鲁棒的故障恢复技术。通过将实例化结果与其输入批次进行映射,系统可以跟踪哪些输入文件生成了哪些结果。因此,如果在处理中存在错误,例如片段计算资源失败,系统可能只需要重新调度提供给失败资源的批次。

[0088] 图9示出了根据一些示例实施例的输入到输出映射的示例。这里示出了输入批次910、912、914。如本文所述,每个输入批次910、912、914可以包括连续扫描集的一个或更多个文件。输入批次910、912、914可以由一个或更多个片段计算资源使用片段执行来处理。片段执行的输出可以作为MR文件(也称为实例化批次(MB))920、922、924来提供。MR文件可能只包括特定输入批次的处理输出。例如,MB 920可以包括输入批次910中文件的处理结果,MB 922可以包括输入批次912中文件的处理结果,等等。如果一个输入批次很大,并且输出不适合一个MR文件,则输出可能被分成几个MR文件,称为实例化批次(MB)。然而,MB中的每个MR文件可能只包含一个输入批次的结果。

[0089] 接下来将描述用于确保每个实例化结果文件仅包含来自一个输入批次的数据的技术。这些跟踪技术可以提供有价值的元数据信息,例如哪些输入批次已被成功处理并生成了实例化结果,以及将相应的实例化结果与特定的输入批次相关联。该信息可以允许快速有效地恢复计算资源故障,例如通过重新调度相应的输入批次,而不是再次处理所有文件。

[0090] 为了确保正确的输入批次到实例化结果的关系,行集操作符(RSO)链接可以被设计为本地同步的。也就是说,来自一个输入批次的所有数据都可以包含在一个RSO线程中,并且不会干扰其他线程。如本文所述,这可以通过使用本地聚合器操作符来执行。此外,可以设置屏障,将不同输入批次的数据分开。这些屏障可以通过使用检查点和暂存区(staging area)来实现,如下文进一步详细描述的。

[0091] 图10示出了根据一些示例实施例的片段处理系统1000的示例。片段处理系统1000可以包括具有QC暂存区1010的QC 1005、输入SFQ 1015、具有片段暂存区1025的片段计算资源1020和输出SFQ 1030。QC 1005可以协调和调度片段计算资源1020上的作业。当片段计算资源1020可用于执行片段处理时,QC 1005可以向输入SFQ 1015请求文件。

[0092] 如上所述,所请求的文件可以被分组为一个批次,并且该批次可以被分配一个唯一的批次ID。该批次可以放置在QC暂存区1010中。在那里,可以存储元数据,例如批次ID以及该批次中的文件列表。QC 1005可以将具有唯一批次ID的批次转移到片段计算资源1020。片段计算资源1020可以将该批次放入其片段暂存区1025中。片段计算资源1020可以包括每个线程的暂存区。每个片段计算资源1020可以包括多个执行线程(例如,八个执行线程)。在一个实施例中,线程的数量可以对应于片段计算资源的处理核心的数量。每个线程可以有自己的流水线,并在计算资源中并行工作。

[0093] 批次ID和批次中的文件列表可以存储在片段暂存区1025中。片段计算资源1020可以处理该批次(例如,执行片段计划中的一组操作符)。

[0094] 如下文进一步详细解释的,操作符可以使用检查点来保持不同批次数据的分离。在执行最后一个操作之后,片段暂存区1025可以接收检查点信号。片段暂存区1025可以最终化(finalize)处理的批次。此外,处理的输出可以被上传到存储区域(例如,云存储部),并且文件列表(例如,URLS)可以以实例化结果文件列表的形式被上传到输出SFQ 1030。在一个实施例中,文件可以滚动上传。片段暂存区1025还可以将实例化文件与相应的输入批次相关联,并将该信息存储为元数据。在所有实例化结果文件都已上传并且批次已最终化后,可以认为该批次已完成。

[0095] 文件批次登记请求可以从片段计算资源1020发送到QC 1005。在一个实施例中,文件批次登记请求可以以JavaScript对象符号(JSON)格式提供。

[0096] 因此,如果没有接收到特定批次的文件批次登记请求,则该批次ID的信息保留在QC暂存区1010中。因此,QC 1005可以重新调度该批次的处理。在一个实施例中,为了避免可能的结果重复,QC 1005可以在重新调度之前为该批次分配新的、不同的唯一批次ID。在另一个实施例中,QC 1005可以指示由父计算资源而不是片段计算资源来执行该批次的处理。

[0097] 如这里所讨论的,片段的执行是通过操作符来执行的。分离不同输入批次的数据的屏障可能因操作符而异。例如,一些操作符,如过滤器和投影操作符,可能是无状态的,因此可能不会实现屏障。其他操作符,如表扫描、聚合和插入操作符,不是无状态的,可能会实现屏障来将不同的输入批次的数据分离。收到检查点信号后,可以刷新这些操作符的状态。屏障的形式可能因操作符而异。

[0098] 图11示出了根据一些示例实施例的使用检查点的示例片段计划1100。片段计划1100可以包括由片段计算资源执行片段计划的多个操作符:表扫描操作符1105、过滤器操作符1110、聚合操作符1115和插入操作符1120。关于表扫描操作符1105,在向QC请求更多文件(或批次)之前,使用检查点的异步扫描器可以暂停,直到所有扫描器都用完文件。异步扫描器扫描完一批次中的所有文件后,会生成一个检查点信号,并将其传播给本地分布的其余下游操作符。每个执行线程可以产生一个或多个异步扫描器。出于创建屏障的目的,异步扫描器可能只会干扰从同一个执行线程产生的同类扫描器。

[0099] 过滤器操作符1110可以是无状态的,因此不需要检查点变更,并且可以忽略检查点信号。另一方面,聚合操作符1115可以不是无状态的,并且可以维护本地散列表。使用检查点,可以在批次之间刷新本地散列表。也就是说,聚合操作符1115可以接收检查点信号,并作为响应,通过输出链路刷新其数据,清除散列表。插入操作符1120可以缓冲数据,直到它接收到足够的数据来填充输出文件(例如,新的实例化结果文件)。如下面进一步详细解

释的,当缓冲器在每个批次结束时接收到检查点信号时,它可以被刷新。

[0100] 图12示出了根据一些示例实施例的具有暂存区的示例片段处理环境。该环境可以包括RSOInsert操作符1205、片段暂存区1210、异步上传器1215和文件批次登记队列1220。如上所述,片段计算资源可以包括每个RSO线程的暂存区和由所有RSO线程共享的文件批次登记队列。

[0101] RSOInsert操作符1205可以缓冲当前批次的元数据(如图所示),其被放置在片段暂存区1210中。也就是说,RSOInsert操作符1205可以向当前批次的MR文件添加数据。一旦接收到检查点信号,片段暂存区1210可以最终化当前批次。异步上传器1215然后将MR文件上传到存储区域,并将MR文件列表上传到输出SFQ。MR文件全部上传后,可以认为当前批次已经完成。接下来,可以将具有已完成批次的批次ID的元数据放置在文件批次登记队列中,并且可以清除或擦除片段暂存区1210中的当前批次的数据。

[0102] 文件批次登记队列1220可以接收MR文件的元数据及其对应批次ID。文件批次登记队列然后可以向QC发送文件批次登记请求,该请求带有与MR文件及其对应的批次ID相关的信息。QC可以登记文件,然后从其暂存区删除批次ID。在一个实施例中,MB中的所有MR文件可以在同一请求中登记。

[0103] 在一个实施例中,文件批次登记队列1220可以在其发送文件批次登记请求之前具有阈值数。例如,考虑文件批次登记队列1220在能够发送文件批次登记请求之前具有十个文件的阈值的例子。文件批次登记队列1220可以接收五个MR文件,全部对应于批次ID 1。但是,由于5小于10,此时它可能不会发送文件批次登记请求,因为它还未达到10个文件的阈值。随后,它可能会收到另外五个MR文件,所有这些文件都对应于批次ID 2。现在,文件批次登记队列1220已经达到其十个文件的阈值。因此,文件批次登记队列1220可以发送具有十个文件的文件批次登记请求,其中五个被识别用于批次ID 1,而另外五个被识别用于批次ID 2。

[0104] 在一个实施例中,片段暂存区1210可以由RSOInsert操作符1205和异步上传器1215共同拥有。每个所有者可以使用计数器来确定当前批次的状态。例如,RSOInsert操作符1205最初可以拥有新的批次,并且每当组成新的MR文件时,计数器可以递增。此时,异步上传器1215可以获得共同所有权,并且每当MR文件已经完成上传时,计数器可以递减。在接收到检查点信号之后,RSOInsert操作符1205可以最终化批次并释放其所有权。然后,一旦计数器达到零(或负1),当前批次可被视为完成,因为所有文件都已由异步上传器1215上传。异步上传器1215可以释放它们的所有权,并且片段暂存区1210中的数据可以在如上所述被放置在文件登记队列中之后被删除。

[0105] 图13示出了根据一些示例实施例的用于跟踪输入批次以输出实例化结果的方法1300的流程图。如图所示,方法1300的部分可以由QC和片段计算资源(例如,执行平台(XP))执行。

[0106] 在操作1305,片段计算资源可以向QC请求文件。例如,片段计算资源可以通过REST API等请求文件。在操作1310,作为响应,QC可以从输入SFQ中检索一批次的文件,为该批次分配唯一的批次ID,并将包括批次ID的该批次的元数据和该批次中的文件列表存储在QC暂存区中。在一个实施例中,如果父计算资源请求文件,QC可以不使用QC暂存区;在该实施例中,QC暂存区可用于片段计算资源请求的文件。在操作1315,QC可以向请求的片段计算资源

发送批次分配。

[0107] 在操作1320,片段计算资源可以在其暂存区中加载该批次的元数据。在操作1325,片段计算资源可以对批次中的文件执行片段计划的第一操作符。例如,片段查询计划中的第一操作符可以是表扫描操作,因此片段计算资源可以执行表扫描操作。在操作1330,在完成片段计划中的第一操作符时,可以生成检查点信号,并且可以将其传播到片段计划中的下游操作符。

[0108] 在操作1335,可以执行片段计划的剩余部分,每个操作符接收检查点信号并在需要时采取适当的行动(例如,刷新状态)。在操作1340,当片段计划的最后一个操作符被执行并且在最后一个操作符处接收到检查点信号时,该批次可以被最终化。在操作1345,可以将所有MR文件上传到存储区域,并且可以将MR文件的文件列表上传到输出SFQ(例如,通过文件登记)。在一个实施例中,MR文件可以滚动上传。也就是说,MR文件可以在它们完成时上传,并且一些MR文件可以在批次最终化之前上传。

[0109] 在操作1350,MR文件的元数据可以被发送到文件批次登记队列,并且文件批次登记请求可以被发送到QC。例如,文件批次登记请求可以通过REST API等发送。文件批次登记请求可以包括与MR文件及其对应的批次ID相关的信息。例如,每个单文件登记请求可能包含其对应的批次ID。在一个实施例中,批次ID的列表可以被添加到请求中,以确保产生空输出的批次也被考虑在内。在操作1355,QC可以在文件批次登记请求中登记文件。此外,QC可以擦除存储在其暂存区中的关于该批次的信息。

[0110] 这里描述的跟踪技术有助于故障和错误恢复。QC可以以多种方式检查计算资源的状态。例如,如果计算资源正在运行,则该计算资源可以向QC提供其运行状态。在计算资源崩溃的情况下,QC可以检测到崩溃。此外,计算资源可以向QC发送周期性心跳状态检查,并且如果QC在指定时间内没有接收到心跳状态检查,则QC可以确定计算资源中发生了错误。因此,如果QC检测到计算资源中的错误或崩溃,QC可以使用这里描述的跟踪技术(例如,使用批次ID、暂存区和/或检查点)来确定要重新调度哪个特定批次或文件,而不必重复整个作业的执行。

[0111] 此外,尽管在片段处理上下文中讨论了本文描述的跟踪技术的一些实施例,但是本文描述的跟踪技术不限于片段处理,并且可以用于其他应用。例如,这里描述的跟踪技术可以用于大型插入操作。大的插入操作可以被分成批次,并且这里描述的跟踪技术可以提供输入批次和输出文件的映射。因此,如果一个特定批次的处理失败,系统可能不必重复整个大的插入操作,而是只重复失败的批次。因此,这里描述的跟踪技术可以在出现某些错误时识别哪些输出批次文件是可用的,以及哪些输出批次文件可能需要重做。这里描述的跟踪技术也可以用于任何合适的应用,例如大数据操作语言(DML)操作或多步骤查询计划等。

[0112] 根据示例实施例,图14以计算机系统的形式示出了机器1400的示意图表示,用于使机器1400执行本文所讨论方法中的任何一个或更多的指令集合可以在该计算机系统内执行。具体地,图14以计算机系统的示例形式示出了机器1400的示意图表示,用于使机器1400执行本文所讨论方法中的任何一个或更多的指令1416(例如,软件、程序、应用、小程序、小应用(app)或其他可执行代码)在该计算机系统内执行。例如,指令1416可以使机器1400执行这里描述的方法中的任何一个或更多的任何一个或更多个操作。作为另一个例子,指令1416可以使机器900实现这里描述的数据流的部分。以这种方式,指令1416将通用

的、未编程的机器转换成特定的机器1400 (例如, 远程计算设备106、访问管理系统110、计算服务管理器112、执行平台114、访问管理系统118、web代理120、远程计算设备106), 该特定的机器1400被特别配置成以这里描述的方式执行所描述和示出的功能中的任何一个。

[0113] 在替代实施例中, 机器1400作为独立的设备操作或者可以耦合 (例如, 联网) 到其它机器。在联网部署中, 机器1400可以在服务器-客户端网络环境中作为服务器机器或客户端机器来操作, 或者作为对等 (或分布式) 网络环境中的对等机器来操作。机器1400可以包括但不限于服务器计算机、客户端计算机、个人计算机 (PC)、平板计算机、膝上型计算机、上网本 (netbook)、智能电话、移动设备、网络路由器、网络交换机、网络桥接器或能够顺序地或以其它方式执行指令1416的任何机器, 指令1416指定由机器1400采取的动作 (action)。此外, 虽然仅示出了单个机器1400, 但术语“机器”还应被理解为包括单独或联合地执行指令1416以实施本文所讨论方法中的任何一个或更多个的机器1400的集合。

[0114] 机器1400包括处理器1410、存储器1430和输入/输出 (I/O) 组件1450, 它们被配置成例如经由总线1402彼此通信。在示例实施例中, 处理器1410 (例如, 中央处理单元 (CPU)、精简指令集计算 (RISC) 处理器、复杂指令集计算 (CISC) 处理器、图形处理单元 (GPU)、数字信号处理器 (DSP)、专用集成电路 (ASIC)、射频集成电路 (RFIC)、另一处理器或其任何合适的组合) 可以包括例如可以执行指令1416的处理器1412和处理器1414。术语“处理器”旨在包括多核处理器1410, 多核处理器1410可以包括可以同时执行指令1416的两个或更多个独立处理器 (有时称为“核”)。尽管图14示出了多个处理器1410, 但是机器1400可以包括具有单核的单个处理器、具有多核的单个处理器 (例如, 多核处理器)、具有单核的多个处理器、具有多核的多个处理器或者它们的任意组合。

[0115] 存储器1430可以包括主存储器1432、静态存储器1434和存储单元1436, 它们都可以由处理器1410例如经由总线1402访问。主存储器1432、静态存储器1434和存储单元1436存储指令1416, 指令1416体现本文描述的方法或功能中任何一种或更多种。在由机器1400执行期间, 指令1416也可以全部或部分地驻留在主存储器1432内、静态存储器1434内、存储单元1436内、至少一个处理器1410内 (例如, 在处理器的高速缓冲存储器内) 或其任何合适的组合内。

[0116] I/O组件1450包括用于接收输入、提供输出、产生输出、传输信息、交换信息、捕获测量值等的组件。包含在特定机器1400中的具体的I/O组件1450将取决于机器的类型。例如, 诸如移动电话的便携式机器将可能包括触摸输入设备或其他这样的输入机制, 而无头服务器机器可能不包括这样的触摸输入设备。将认识到的是, I/O组件1450可以包括图14中未示出的许多其他组件。I/O组件1450根据功能进行分组仅仅是为了简化下面的讨论, 并且该分组决不是限制性的。在各种示例实施例中, I/O组件1450可以包括输出组件1452和输入组件1454。输出组件1452可以包括视觉组件 (例如, 诸如等离子体显示面板 (PDP)、发光二极管 (LED) 显示器、液晶显示器 (LCD)、投影仪或阴极射线管 (CRT) 的显示器)、声学组件 (例如, 扬声器)、其他信号发生器等。输入组件1454可以包括字母数字输入组件 (例如, 键盘、被配置成接收字母数字输入的触摸屏、光电键盘或其他字母数字输入组件)、基于点的输入组件 (例如, 鼠标、触控板、轨迹球、操纵杆、运动传感器或另一指向仪器)、触觉输入组件 (例如, 物理按钮、提供触摸或触摸手势的位置和/或力的触摸屏或其他触觉输入组件)、音频输入组件 (例如, 麦克风) 等。

[0117] 通信可以使用多种技术来实现。I/O组件1450可以包括通信组件964,通信组件964可操作来分别经由耦合1482和耦合1472将机器1400耦合到网络1480或设备1470。例如,通信组件1464可以包括网络接口组件或与网络1480接口的另一合适的设备。在进一步的示例中,通信组件1464可以包括有线通信组件、无线通信组件、蜂窝通信组件和经由其他模态提供通信的其他通信组件。设备1470可以是另一台机器或多种外围设备中的任一种(例如,经由通用串行总线(USB)耦合的外围设备)。例如,如上所述,机器900可以对应于远程计算设备106、访问管理系统110、计算服务管理器112、执行平台114、访问管理系统118、web代理120中的任何一个,并且设备1470可以包括这些系统和设备中的任何其他系统和设备。

[0118] 各种存储器(例如,1430、1432、1434和/或处理器1410和/或存储单元1436的存储器)可以存储指令1416和数据结构(例如,软件)的一个或更多个集合,这些指令和数据结构体现本文所描述方法或功能中的任何一种或更多种或被本文所描述方法或功能中的任何一种或更多种所利用。当由处理器1410执行时,这些指令1416促使各种操作来实现所公开的实施例。

[0119] 如本文所使用的,术语“机器存储介质”、“设备存储介质”和“计算机存储介质”意思相同,并且可以在本公开中互换使用。这些术语指的是存储可执行指令和/或数据的单个或多个存储设备和/或介质(例如,集中式或分布式数据库和/或相关联的高速缓存和服务器等)。相应地,这些术语应当被认为包括但不限于包括处理器内部或外部的存储器的固态存储器以及光和磁介质。机器存储介质、计算机存储介质和/或设备存储介质的具体示例包括非易失性存储器,非易失性存储器包括例如半导体存储器设备,诸如可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)、现场可编程门阵列(FPGA)和闪存设备;磁盘,如内部硬盘和可移除磁盘;磁光盘;以及CD-ROM和DVD-ROM光盘。术语“机器存储介质”、“计算机存储介质”和“设备存储介质”具体排除了载波、调制数据信号和其他这样的介质(其中的至少一些包含在下面讨论的术语“信号介质”中)。

[0120] 在各种示例实施例中,网络980的一个或更多个部分可以是自组织网络(ad hoc network)、内联网、外联网、虚拟专用网络(VPN)、局域网(LAN)、无线LAN(WLAN)、广域网(WAN)、无线WAN(WWAN)、城域网(MAN)、互联网、互联网的一部分、公共交换电话网络(PSTN)的一部分、普通老式电话服务(POTS)网络、蜂窝电话网络、无线网络、**Wi-Fi®**网络、另一种类型的网络、或两种或更多种这样的网络的组合。例如,网络980或网络1480的一部分可以包括无线或蜂窝网络,并且耦合1482可以是码分多址(CDMA)连接、全球移动通信系统(GSM)连接或另一种类型的蜂窝或无线耦合。在该示例中,耦合982可以实现多种类型的数据传输技术中的任何一种,如单载波无线电传输技术(1xRTT)、演进数据优化(EVDO)技术、通用分组无线电服务(GPRS)技术、增强型数据速率GSM演进(EDGE)技术、包括3G、第四代无线(4G)网络、通用移动通信系统(UMTS)、高速分组接入(HSPA)、微波接入全球互操作性(WiMAX)、长期演进(LTE)标准的第三代合作伙伴计划(3GPP)、由各种标准设置组织定义的其他技术、其他长程协议、或其他数据传输技术。

[0121] 指令1416可以使用传输介质经由网络接口设备(例如,包括在通信组件1464中的网络接口组件)并利用多种众所周知的传输协议(例如,超文本传输协议(HTTP))中的任何一种在网络1480上传输或接收。类似地,指令1416可以使用传输介质经由耦合1472(例如,对等耦合)传输或接收到设备1470。术语“传输介质”和“信号介质”意思相同,在本公开中可

以互换使用。术语“传输介质”和“信号介质”应被理解为包括能够存储、编码或携带由机器1400执行的指令1416的任何无形介质,并且包括数字或模拟通信信号或便于这种软件的通信的其他无形介质。因此,术语“传输介质”和“信号介质”应被理解为包括任何形式的调制数据信号、载波等。术语“调制数据信号”意指这样的信号:其具有以对信号中的信息进行编码的这样的方式设置或更改的其特性中的一个或更多个特性。

[0122] 术语“机器可读介质”、“计算机可读介质”和“设备可读介质”意思相同,并且在本公开中可以互换使用。这些术语被定义为包括机器存储介质和传输介质。因此,这些术语包括存储设备/介质和载波/调制数据信号。

[0123] 本文所述的示例方法的各种操作可以至少部分地由(例如,通过软件)临时配置或永久地配置为执行相关操作的一个或更多个处理器来执行。类似地,本文描述的方法可以至少部分地是处理器实现的。例如,这里描述的方法的至少一些操作可以由一个或更多个处理器执行。某些操作的执行可以分布在一个或更多个处理器中,不仅驻留在单个机器内,而且部署在多个机器上。在一些示例实施例中,一个或更多个处理器可以位于单个位置(例如,在家庭环境、办公室环境或服务器场内)中,而在其他实施例中,处理器可以分布在多个位置上。

[0124] 尽管参考具体的示例实施例描述了本公开的实施例,但显然地,在不背离本发明主题的更广泛范围的情况下,可以对这些实施例做出各种修改和变化。相应地,说明书和附图被认为是说明性的而不是限制性的。形成其一部分的附图通过说明方式并且不以限制方式显示了其中可以实施主题的具体实施例。所示出的实施例以充分细节描述,以便使本领域那些技术人员能够实施本文披露的教导。可以使用其他实施例和从其中衍生的实施例,使得可以作出结构性或逻辑性替换和变化而不偏离本公开的范围。因此不以限制性意义理解这一详细说明,并且各种实施例的范围仅由所附权利要求连同授权的此类权利要求的完整范围的等同物限定。

[0125] 发明主题的此类实施例可以在本文中单独地和/或共同地由术语“发明”提到,该术语仅出于便利目的并且不意在自愿将本申请的范围限于任何单一发明或发明构思,如果实际上披露的多于一种的话。因此,虽然已经在此说明和描述了特定实施例,但是应当理解,可以用经计算以实现相同目的的任何布置替换用于所示的具体实施例。本公开旨在覆盖各种实施例的任何和所有修改或变型。在阅读以上描述后,本领域的技术人员将明白以上实施例的组合以及本文未具体描述的其他实施例。

[0126] 在本文档中,术语“一个(a)”或“一个(an)”,如专利文档中常见的那样,用于包括一个或多于一个,与“至少一个”或“一个或更多个”的任何其他实例或用法无关。。在本文档中,术语“或”用于指非排他性的或,因此“A或B”包括“A但不包括B”、“B但不包括A”和“A和B”,除非另有说明。在所附权利要求中,术语“包括(including)”和“其中(in which)”被用作相应术语“包含(comprising)”和“其中(wherein)”的纯英语等效词。此外,在所附权利要求中,术语“包括”和“包含”是开放式的;也就是说,包括除了权利要求中这样的术语之后列出的那些元素之外的元素的系统、设备、物品或过程仍然被认为落入该权利要求的范围内。

[0127] 以下编号的示例是实施例:

[0128] 示例1.一种方法,包括:由一个或更多个处理器接收针对数据集的查询;创建执行查询的查询计划;基于一组标准,识别有资格进行片段处理的查询计划的一部分;由父查询

协调器在数据集的第一批次文件上执行查询计划的识别的部分,以生成第一批次结果;由片段查询协调器在数据集的第二批次文件上执行查询的识别的部分,以生成实例化结果文件;扫描实例化结果文件以生成扫描结果;组合第一批次结果和扫描结果以生成组合结果;以及对组合结果执行查询计划的剩余部分,以生成对查询的响应。

[0129] 示例2.根据示例1所述的方法,还包括:将数据集的文件作为连续扫描集加载到第一共享文件队列中;将第一组文件分组为第一批次,并将第一批次提供给父查询协调器;以及将第二组文件分组为第二批次,并将第二批次提供给片段查询协调器。

[0130] 示例3.根据示例1-2中任一实施例的方法,还包括:串行地提供附加批次,直到已经提供了连续扫描集中的所有文件。

[0131] 示例4.根据示例1-3中任一示例的方法,其中片段查询协调器将实例化结果文件加载到输出共享文件队列中。

[0132] 示例5.根据示例1-4中任一示例的方法,其中片段查询协调器使用一个或多个片段计算资源来执行查询的所识别的部分。

[0133] 示例6.根据示例1-5中任一示例的方法,其中片段计算资源的数量在查询执行期间改变。

[0134] 示例7.根据示例1-6中任一示例所述的方法,其中该组标准包括所识别的部分的执行输出是否小于所识别的部分的输入。

[0135] 示例8.根据示例1-7中任一示例所述的方法,其中该组标准包括所识别的部分是否可由计算资源执行而无需与另一计算资源通信。

[0136] 示例9.根据示例1-8中任一示例所述的方法,其中查询计划包括多个操作符和连接操作符的链路,每个链路连接多个操作符中的第一操作符和第二操作符,并指示第一操作符是否可由计算资源执行而无需与另一计算资源通信。

[0137] 示例10.根据示例1-9中任一示例的方法,还包括:将实例化结果文件与至少另一个实例化结果文件聚合;并且其中扫描实例化结果文件以生成扫描结果包括扫描聚合的实例化结果文件以生成扫描结果。

[0138] 示例11.一种系统,包括:机器的一个或多个处理器;和存储指令的存储器,当指令被一个或多个处理器执行时,使得机器执行实现示例方法1至10中任一个的操作。

[0139] 示例12.一种包含指令的机器可读存储设备,当指令被机器执行时,使得机器执行实现示例方法1至10中任一个的操作。

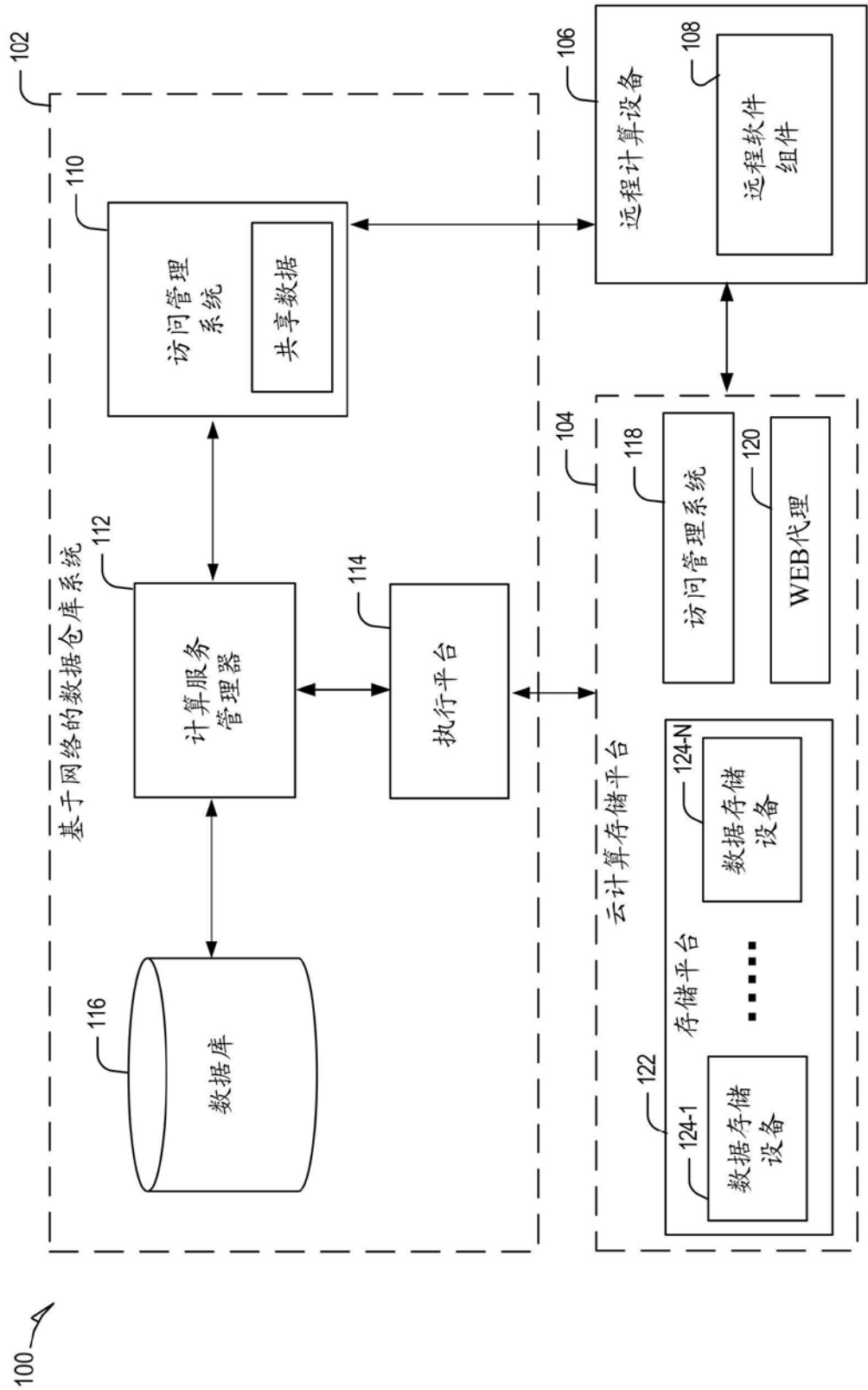


图1

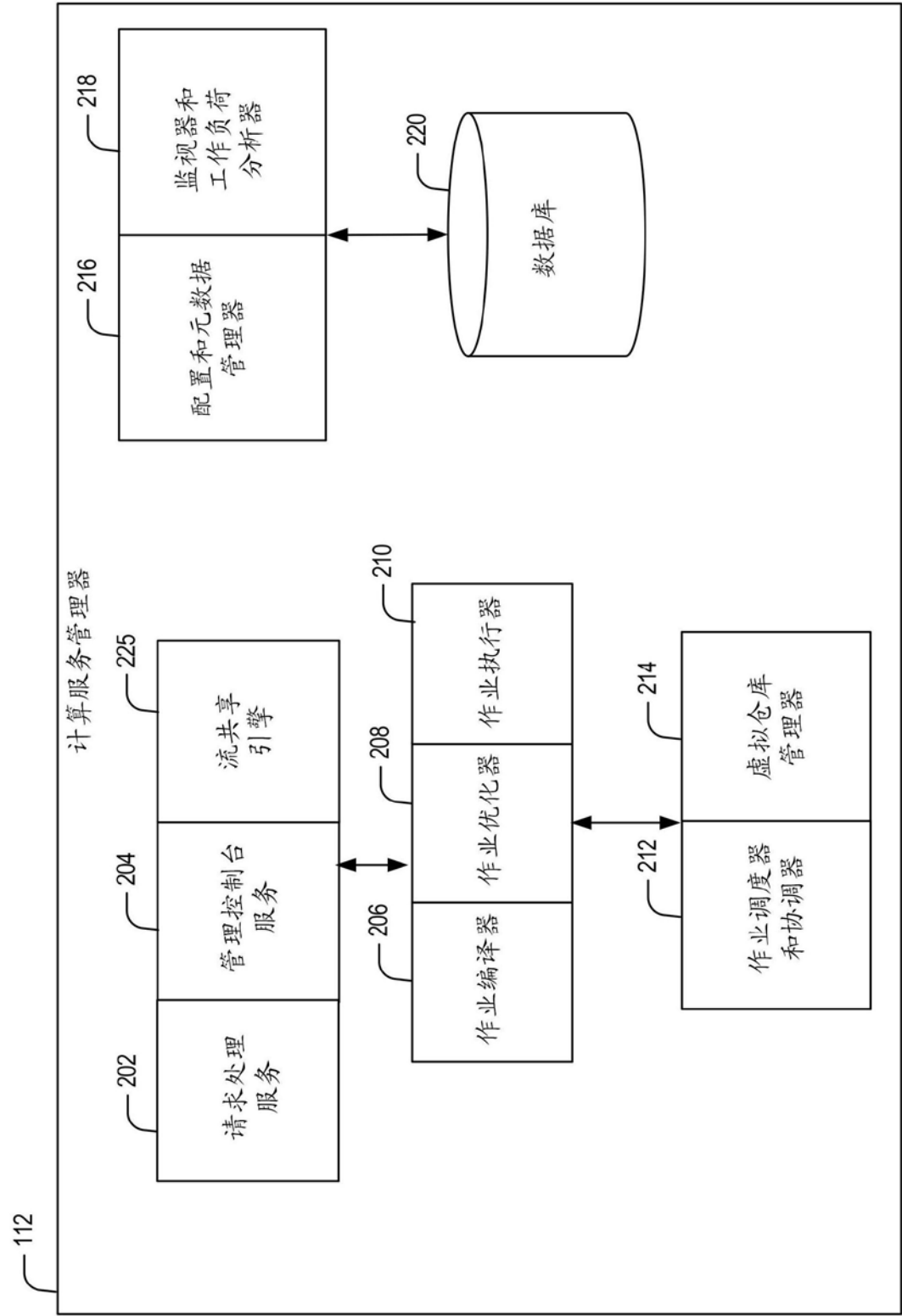


图2

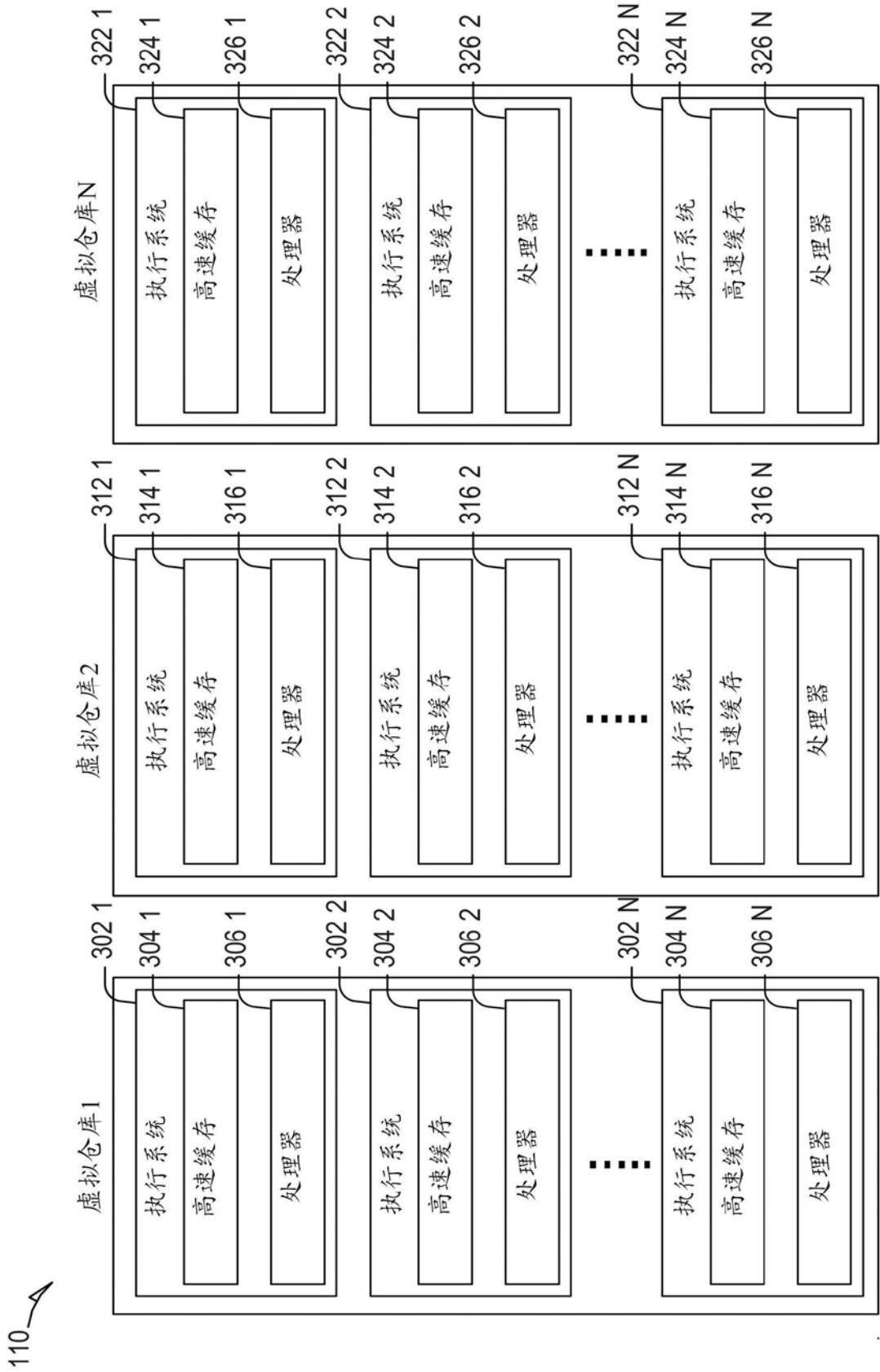


图3

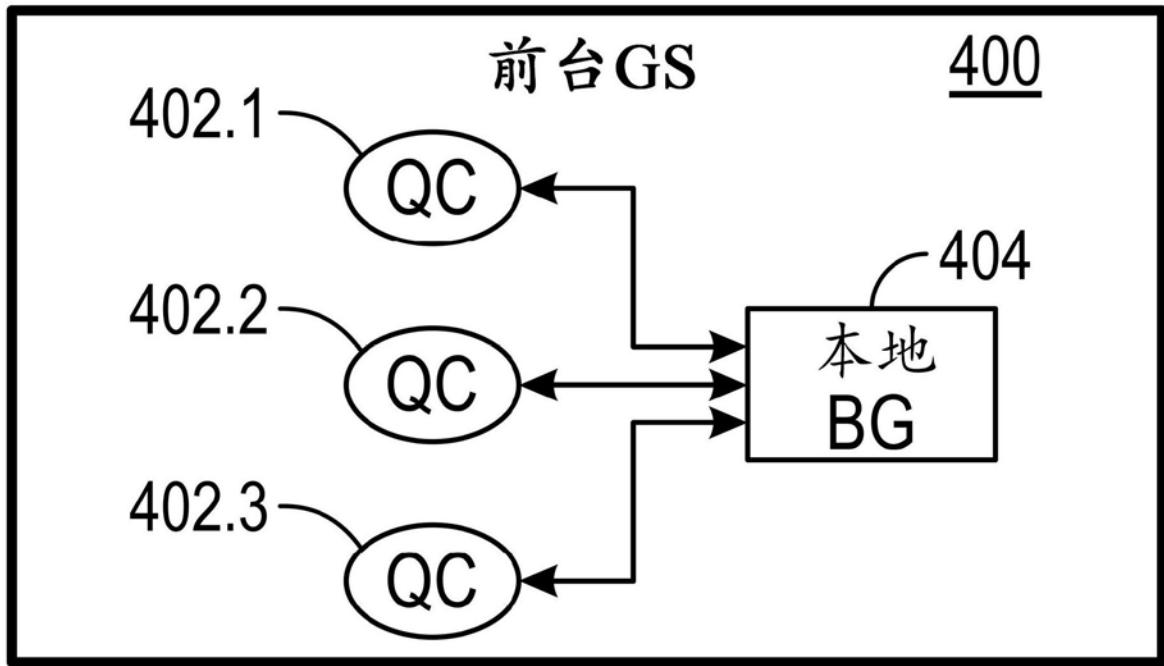


图4

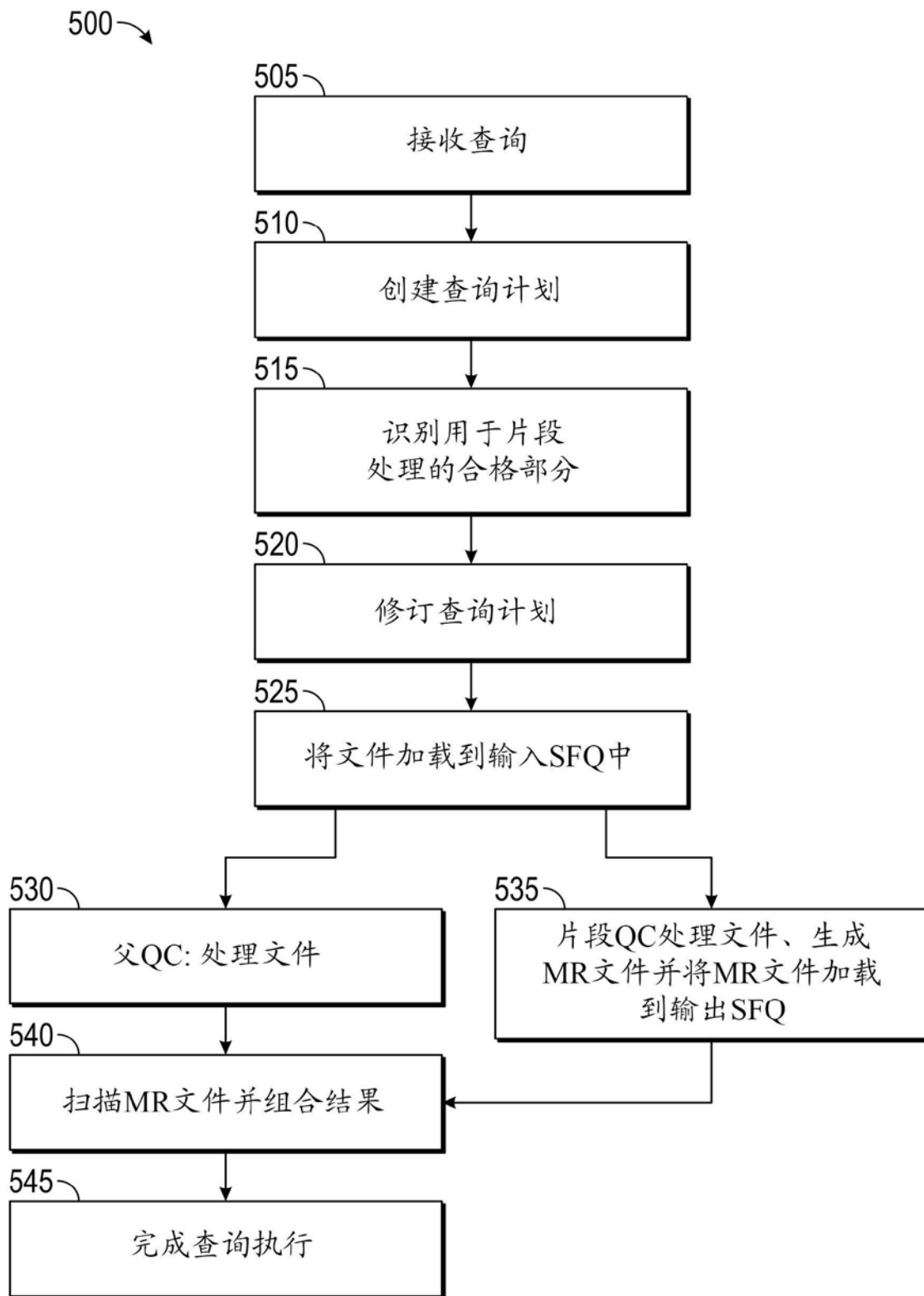


图5

600

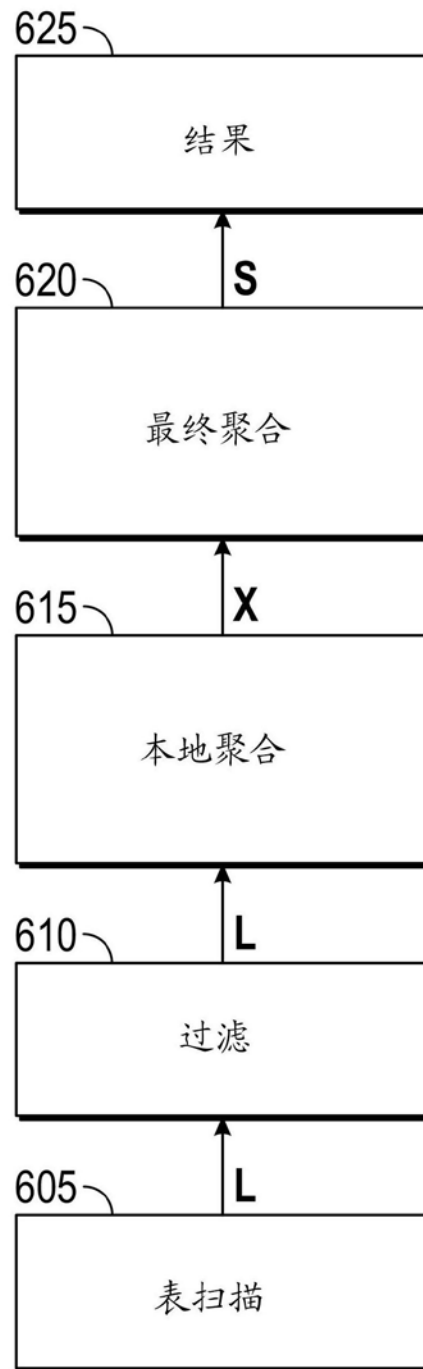


图6

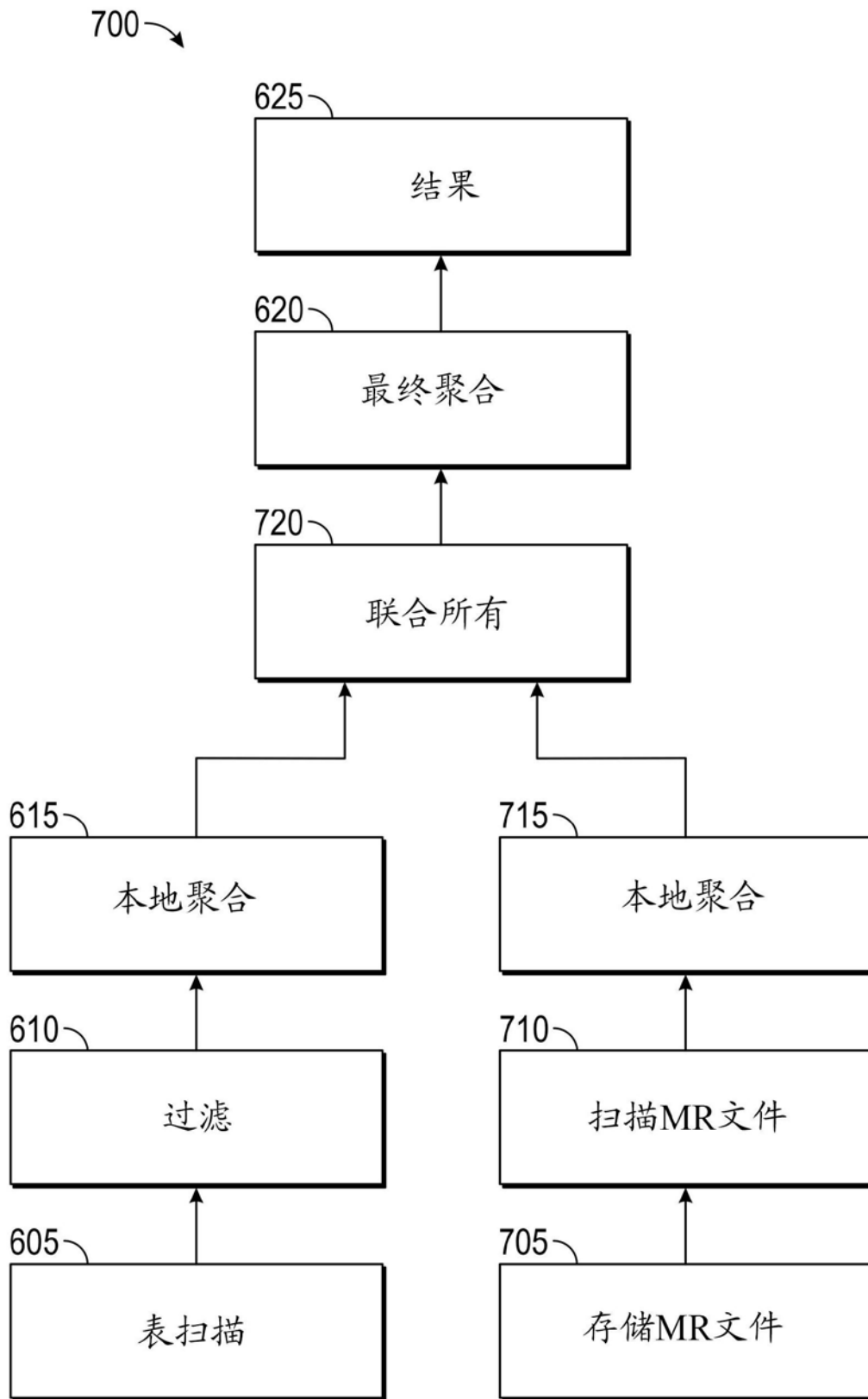


图7

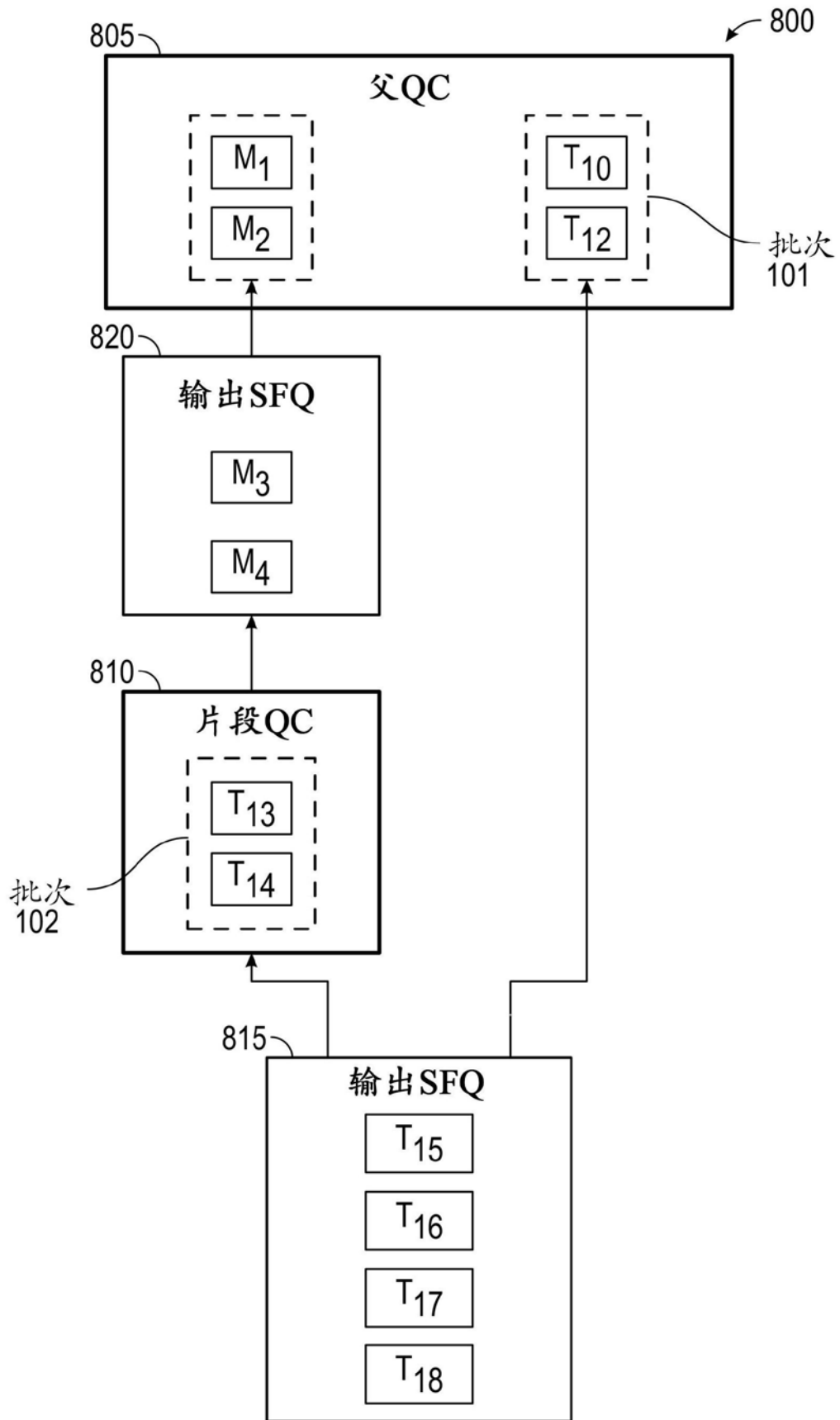


图8

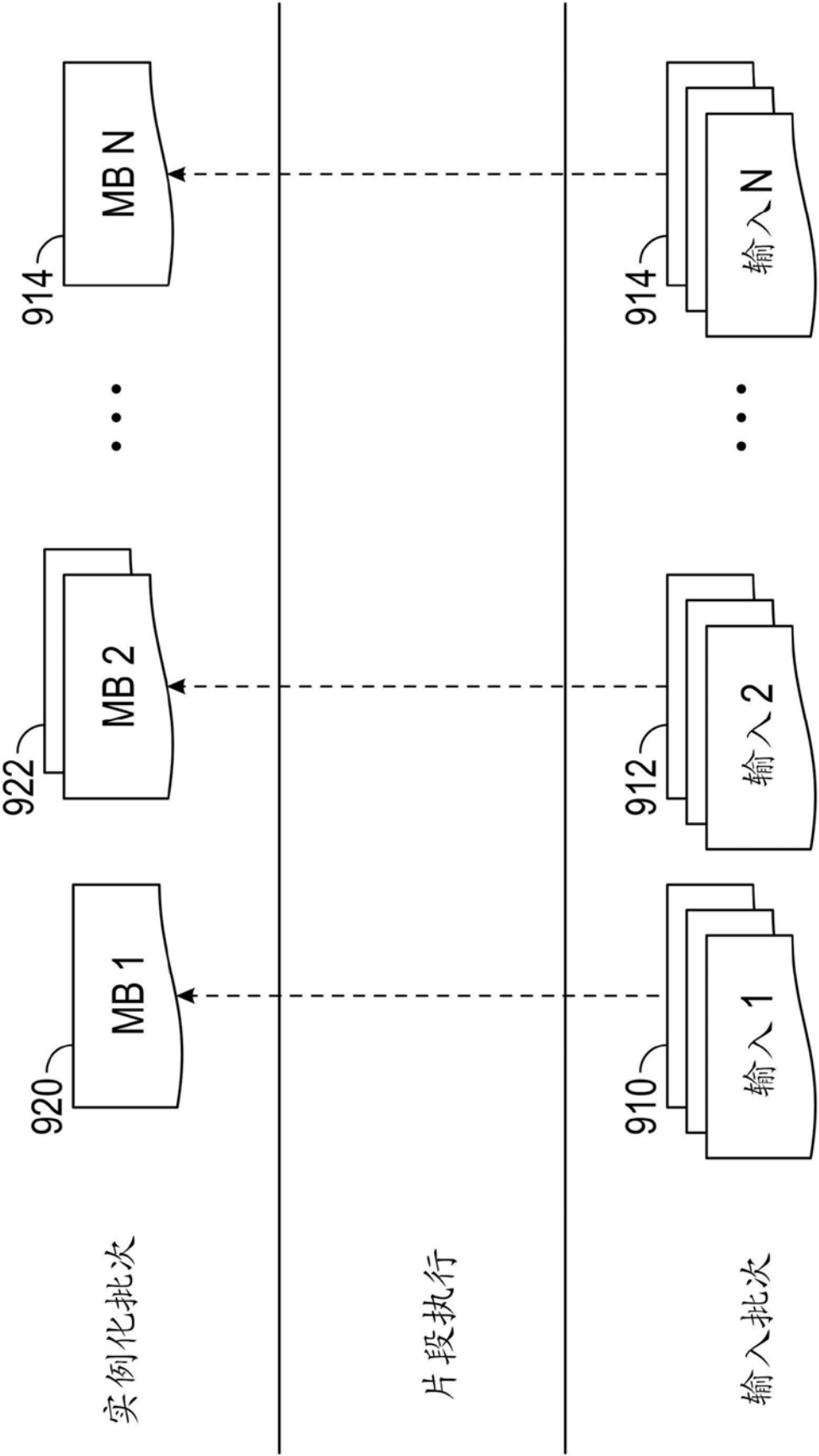


图9

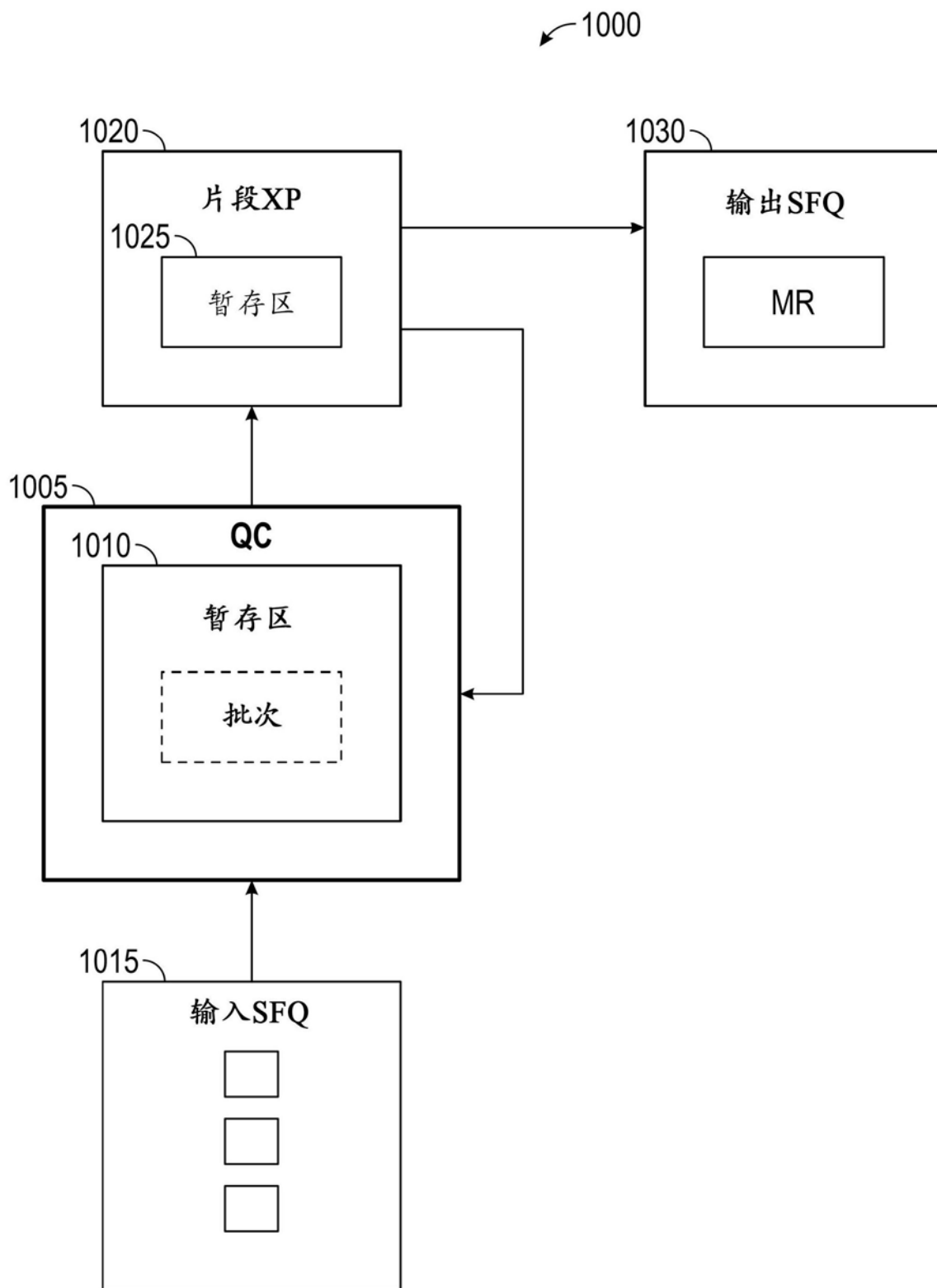


图10

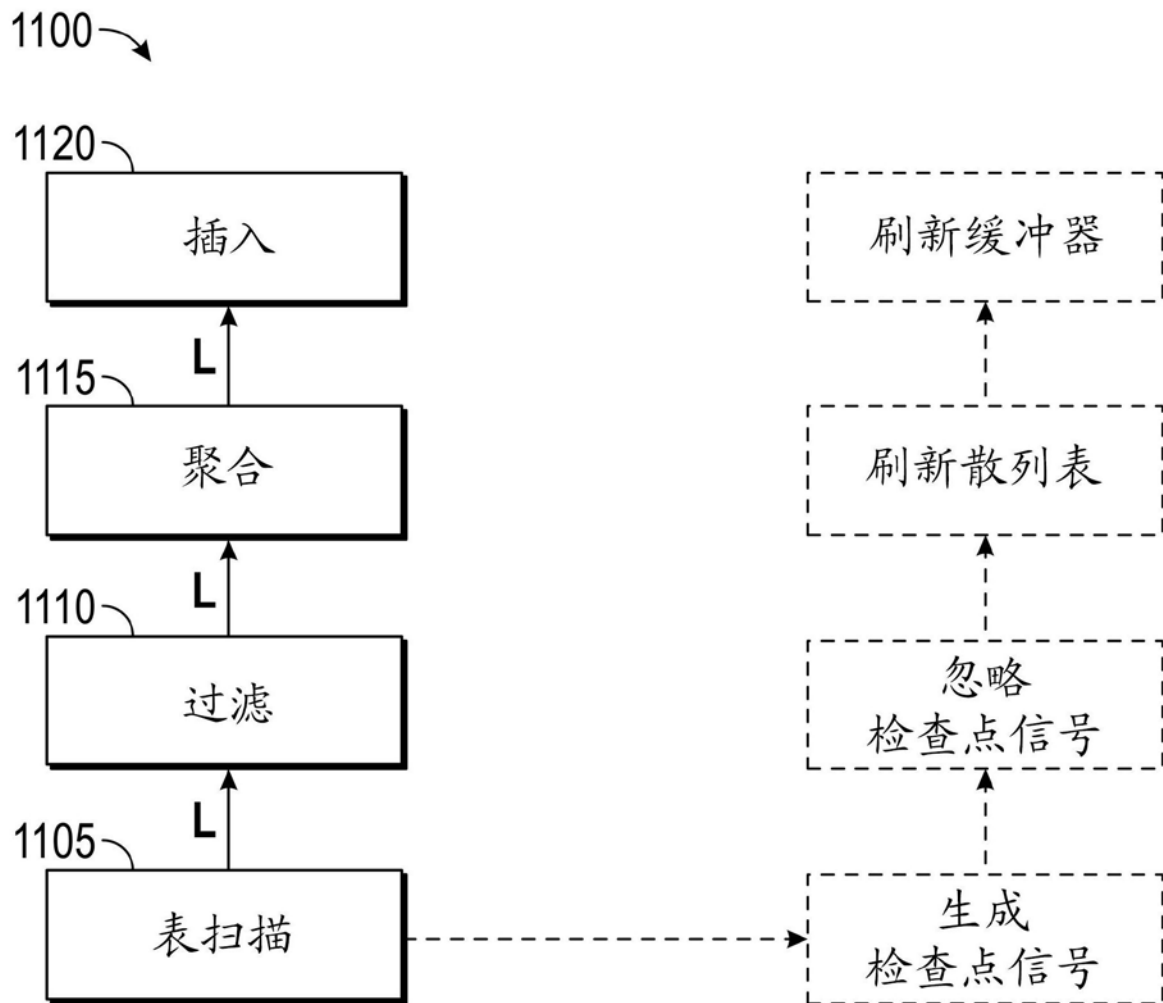


图11

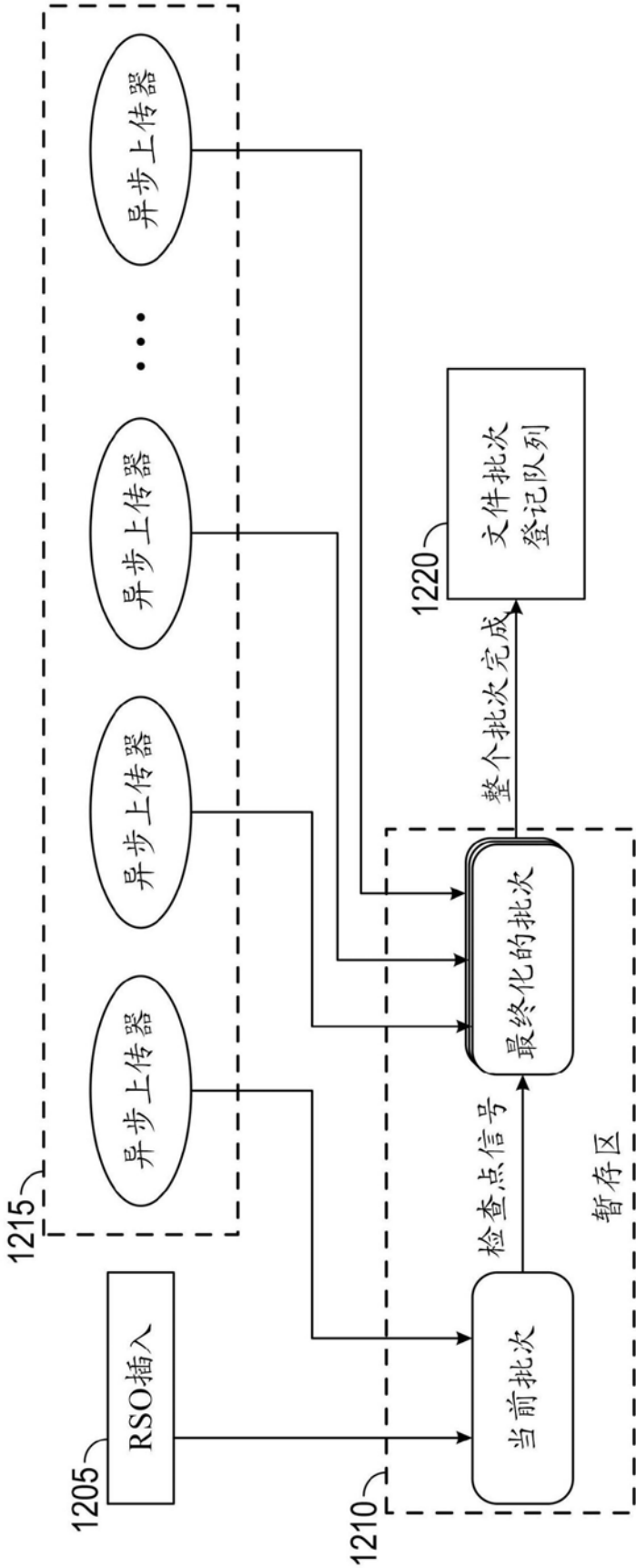


图12

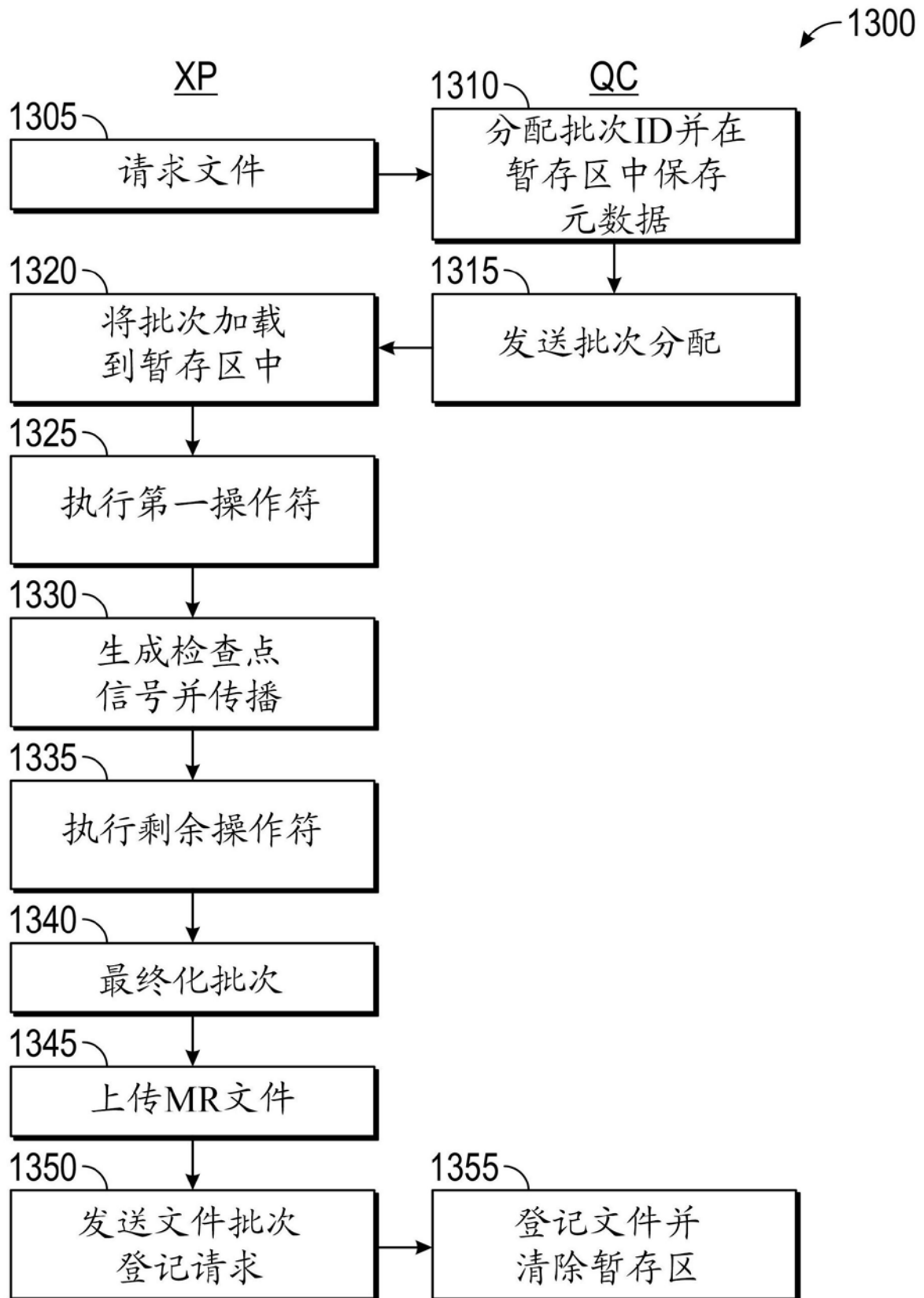


图13

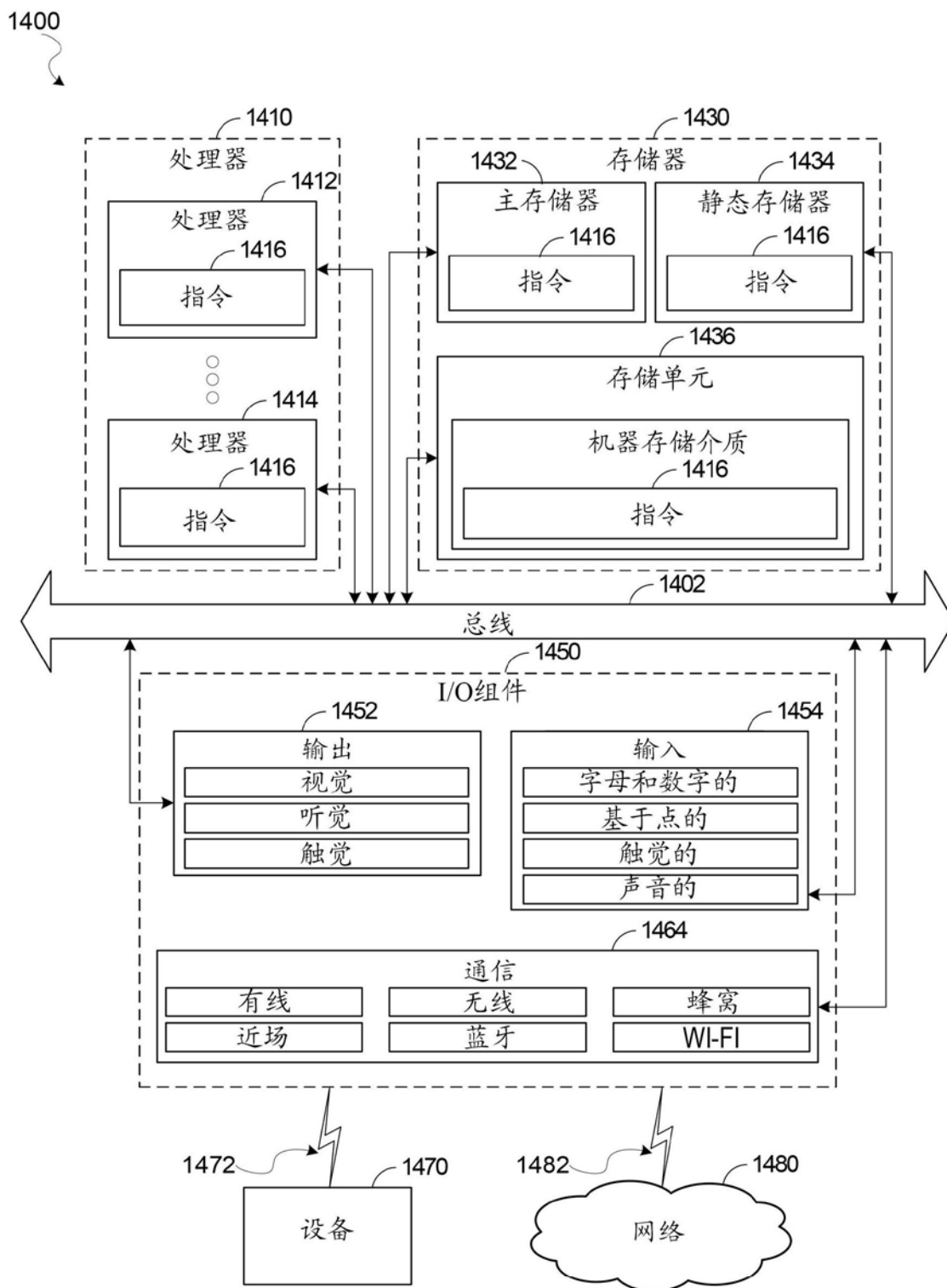


图14