



(12) 发明专利

(10) 授权公告号 CN 109564528 B

(45) 授权公告日 2021. 11. 19

(21) 申请号 201780026323.X

(22) 申请日 2017.10.16

(65) 同一申请的已公布的文献号  
申请公布号 CN 109564528 A

(43) 申请公布日 2019.04.02

(30) 优先权数据  
15/643,429 2017.07.06 US

(85) PCT国际申请进入国家阶段日  
2018.10.26

(86) PCT国际申请的申请数据  
PCT/CN2017/106359 2017.10.16

(87) PCT国际申请的公布数据  
W02019/006907 EN 2019.01.10

(73) 专利权人 华为技术有限公司  
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼

(72) 发明人 胡振华 柯晓棣  
卡勒德·齐亚·艾恩 赵志  
罗志刚

(74) 专利代理机构 广州三环专利商标代理有限公司 44202  
代理人 熊永强 李稷芳

(51) Int.Cl.  
G06F 9/50 (2006.01)  
H04W 72/04 (2006.01)

(56) 对比文件  
CN 102546379 A, 2012.07.04  
CN 103699445 A, 2014.04.02  
CN 102088719 A, 2011.06.08  
CN 102193833 A, 2011.09.21  
CN 106020941 A, 2016.10.12

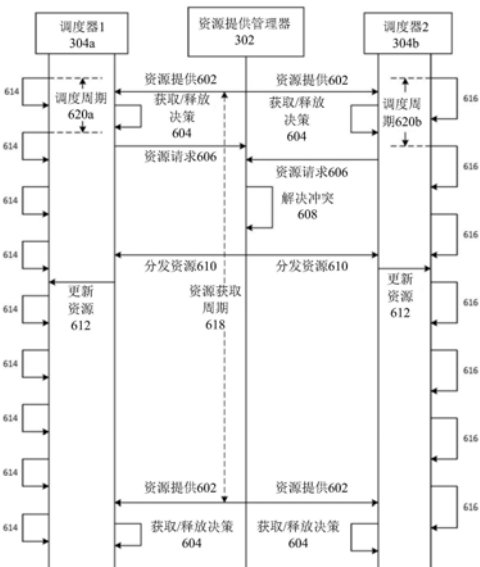
审查员 甘晶萌  
权利要求书3页 说明书15页 附图7页

(54) 发明名称

分布式计算中计算资源分配的系统和方法

(57) 摘要

本文公开了利用资源获取周期和工作负载调度周期分配计算资源以执行分布式计算系统中的工作负载的系统和方法。一个或多个资源提供管理器向一个或多个调度器中的每个传输资源提供。每个调度器传输资源请求和/或资源释放请求以响应每个资源提供管理器。在解决冲突之后，每个资源提供管理器计算分配给每个调度器的资源，并根据资源分配决策将所述资源分配给每个调度器。所述资源获取周期与所述工作负载调度周期分离。



1. 一种在分布式计算系统中两级调度计算资源和工作负载的方法,其特征在于,所述方法包括:

周期性地执行与第一资源提供管理器相关联的第一资源获取周期,所述第一资源提供管理器用于管理由多个工作负载调度器共享和获取的分布式计算系统的计算资源池,所述计算资源池包括所述分布式系统中的所述计算资源的子集;所述第一资源获取周期包括:

所述第一资源提供管理器向每个工作负载调度器传输第一资源提供,所述资源提供指定计算资源池中第一多个计算资源中的每个计算资源的状态;

从多个工作负载调度器中的每个工作负载调度器接收第一资源获取请求,其中,从每个相应工作负载调度器接收的所述第一资源获取请求包括由相应工作负载调度器基于计算资源池中的每个计算资源的状态获取所述计算资源池中的计算资源;

所述第一资源提供管理器在解决所述多个工作负载调度器之间的冲突之后,基于从每个工作负载调度器接收的第一资源获取请求,计算每个工作负载调度器的资源分配;以及

根据对所述相应的工作负载调度器的资源分配,将所述计算资源池中的计算资源分配给所述多个工作负载调度器中的每个相应的工作负载调度器;

执行第一工作负载调度周期,其中,通过调度与每个相应的工作负载调度器相关联的工作负载,以使用分配给相应的工作负载调度器的计算资源来执行。

2. 根据权利要求1所述的方法,其特征在于,还包括在所述第一资源获取周期执行过程中执行第二工作负载调度周期。

3. 根据权利要求1所述的方法,其特征在于,还包括:

执行与第二资源提供管理器相关联的第二资源获取周期,所述第二资源获取周期包括:

所述第二资源提供管理器向所述多个工作负载调度器中的每个传输第二资源提供,所述第二资源提供指定不同于所述第一多个计算资源的第二多个计算资源中的每个的状态;

从所述多个工作负载调度器中的至少一个接收至少一个第二资源请求,所述至少一个第二资源请求基于所述第二多个计算资源中的每个的所述状态;以及

将所述第二多个计算资源中的资源分配给任何所述多个工作负载调度器。

4. 根据权利要求3所述的方法,其特征在于,所述第一资源获取周期独立于所述第二资源获取周期,由所述第一资源提供管理器管理的计算资源池包括与由所述第二资源提供管理器管理的计算资源池不同的分布式计算系统的计算资源。

5. 根据权利要求1至4中任一项所述的方法,其特征在于,所述第一资源获取请求包括获取所述计算资源池中的计算资源量的请求,所述计算资源量指定为一个范围。

6. 根据权利要求1所述的方法,其特征在于,所述第一资源获取请求指定所需的亲和性。

7. 根据权利要求1所述的方法,其特征在于,所述第一资源获取请求指定所需的反亲和性。

8. 根据权利要求1所述的方法,其特征在于,所述计算资源池中的每个计算资源的所述状态包括以下各项中的至少一个:所述第一多个计算资源中当前正在使用的计算资源的标识、所述第一多个计算资源中当前免费使用的计算资源的标识、与所述第一多个计算资源中的特定计算资源相关联的所有权状态以及至少两个所述工作负载调度器之间的计算资

源共享策略。

9. 根据权利要求1所述的方法,其特征在于,所述第一资源提供基于与其中一个所述工作负载调度器相关联的一个或多个订阅。

10. 根据权利要求1所述的方法,其特征在于,还包括:

从其中一个所述工作负载调度器接收补充资源获取请求;以及

基于所述补充资源获取请求将所述第一多个计算资源中的其它资源分配给其中一个所述工作负载调度器。

11. 根据权利要求10所述的方法,其特征在于,分配所述第一多个计算资源中的所述其它资源包括:

基于所述补充资源获取请求从不同于发送所述补充资源获取请求的所述工作负载调度器的工作负载调度器将至少一个先前获取的资源释放给所述第一多个计算资源;以及

将所述释放的至少一个先前获取的资源分配给发送所述补充资源获取请求的所述工作负载调度器。

12. 根据权利要求1所述的方法,其特征在于,还包括:

其中一个所述工作负载调度器执行工作负载调度周期以调度与所述工作负载调度器相关联的第一工作负载,以使用分配给并行于且独立于其它调度器的所述工作负载调度器的计算资源来执行。

13. 根据权利要求1所述的方法,其特征在于,还包括:

资源提供管理器从至少一个工作负载调度器抢占资源并基于至少一个资源共享策略将所述抢占到的资源分配给至少一个其它工作负载调度器。

14. 一种在分布式计算系统中两级调度计算资源和工作负载的系统,其特征在于,包括:

至少一个处理器;

存储器,包含计算机可读指令,所述计算机可读指令在由所述至少一个处理器执行时使得所述至少一个处理器执行包括以下操作的方法:

周期性地执行与第一资源提供管理器相关联的第一资源获取周期,所述第一资源提供管理器用于管理由多个工作负载调度器共享和获取的分布式计算系统的计算资源池,所述计算资源池包括所述分布式系统中的所述计算资源的子集;所述第一资源获取周期包括:

向每个工作负载调度器传输第一资源提供,所述资源提供指定计算资源池中第一多个计算资源中的每个计算资源的状态;

响应于从多个工作负载调度器中的每个工作负载调度器接收第一资源获取请求,其中,从每个相应工作负载调度器接收的所述第一资源获取请求包括由相应工作负载调度器基于计算资源池中的每个计算资源的状态获取所述计算资源池中的计算资源;

在解决所述多个工作负载调度器之间的冲突之后,基于从每个工作负载调度器接收的第一资源获取请求,计算每个工作负载调度器的资源分配;以及

根据对所述相应的工作负载调度器的资源分配,将所述计算资源池中的计算资源分配给所述多个工作负载调度器中的每个相应的工作负载调度器;

执行第一工作负载调度周期,其中,通过调度与每个相应的工作负载调度器相关联的工作负载,以使用分配给相应的工作负载调度器的计算资源来执行。

15. 根据权利要求14所述的系统,其特征在于,所述方法还包括在所述第一资源获取周期执行过程中执行第二工作负载调度周期。

16. 根据权利要求14所述的系统,其特征在于,还用于:

执行与第二资源提供管理器相关联的第二资源获取周期,所述第二资源获取周期包括:

所述第二资源提供管理器向所述多个工作负载调度器中的每个传输第二资源提供,所述第二资源提供指定不同于所述第一多个计算资源的第二多个计算资源中的每个的状态;

从所述多个工作负载调度器中的至少一个接收至少一个第二资源请求,所述至少一个第二资源请求基于所述第二多个计算资源中的每个的所述状态;以及

将所述第二多个计算资源中的资源分配给任何所述多个工作负载调度器。

17. 根据权利要求16所述的系统,其特征在于,所述第一资源获取周期独立于所述第二资源获取周期,由所述第一资源提供管理器管理的计算资源池包括与由所述第二资源提供管理器管理的计算资源池不同的分布式计算系统的计算资源。

18. 根据权利要求14所述的系统,其特征在于,所述第一资源请求包括获取所述计算资源池中的计算资源量的请求,所述计算资源量指定为一个范围。

19. 根据权利要求14所述的系统,其特征在于,所述第一资源获取请求指定所需的亲和性。

20. 根据权利要求14所述的系统,其特征在于,所述第一资源获取请求指定所需的反亲和性。

21. 根据权利要求14所述的系统,其特征在于,所述计算资源池中的每个计算资源的所述状态包括以下各项中的至少一个:所述第一多个计算资源中当前正在使用的计算资源的标识、所述第一多个计算资源中当前免费使用的计算资源的标识、与所述多个计算资源中的特定计算资源相关联的所有权状态以及至少两个所述工作负载调度器之间的计算资源共享策略。

22. 根据权利要求14所述的系统,其特征在于,所述第一资源提供基于与其中一个所述工作负载调度器相关联的一个或多个订阅。

23. 根据权利要求22所述的系统,其特征在于,还用于:

从其中一个所述工作负载调度器接收补充资源获取请求;以及

基于所述补充资源获取请求将所述第一多个计算资源中的其它资源分配给其中一个所述工作负载调度器。

24. 根据权利要求23所述的系统,其特征在于,分配所述第一多个计算资源中的所述其它资源包括:

基于所述补充资源获取请求从不同于发送所述补充资源获取请求的所述工作负载调度器的工作负载调度器将至少一个先前获取的资源释放给所述第一多个计算资源;以及

将所述释放的至少一个先前获取的资源分配给发送所述补充资源获取请求的所述工作负载调度器。

25. 一种计算机可读存储介质,其特征在于,所述计算机可读存储介质上存储有计算机可执行指令,所述计算机可执行指令在由一个或多个处理器执行时使得所述一个或多个处理器执行根据权利要求1至13中任一项所述的方法。

## 分布式计算中计算资源分配的系统和方法

[0001] 相关申请案交叉申请

[0002] 本申请要求2017年7月6日递交的发明名称为“分布式计算中计算资源分配的系统和方法(SYSTEMS AND METHODS FOR ALLOCATING COMPUTING RESOURCES IN DISTRIBUTED COMPUTING)”的第15/643,429号美国专利申请案的在先申请优先权,该在先申请的全部内容以引用的方式并入本文本中。

### 技术领域

[0003] 本发明涉及分布式计算系统,尤其涉及管理分布式计算系统中的调度器的计算资源分配的系统和方法。

### 背景技术

[0004] 在云计算系统等现代分布式计算系统中,通过计算设备从多个源(例如租户、项目、组和工作站)聚合许多不同的计算任务(下文称为工作负载)。同样地,可以聚合或集合多个计算设备或资源服务器中的计算资源以在工作负载之间分发可用计算资源。

[0005] 资源管理系统可以通过使用调度器来管理工作负载。调度器从计算设备接收工作负载并将工作负载分配给计算设备或资源服务器上的计算资源以供执行。调度工作负载可能会比较困难,因为可用计算资源有限。在一些情况下,工作负载比运行这些工作负载的可用资源要多。一些资源管理系统包括多个调度器,它们专用于不同类型的工作负载,负责不同类型的资源或代表不同的租户。在分布式计算系统中存在不止一个调度器的情况下,当两个调度器尝试同时对工作负载使用相同的计算资源时可能会出现冲突。调度器的需求之间的冲突可以得到解决,但是会导致调度效率低,因为失败的调度器将必须花费额外的时间来尝试寻找具有可以执行工作负载的可用资源的其它计算设备或资源服务器。调度效率低会对工作负载吞吐量、资源利用率和工作负载运行时间等性能指标产生负面影响。

[0006] 随着分布式计算系统的扩展以及工作负载量的增加,工作负载调度本身会变成一个复杂且耗时的过程。有效地调度工作负载会比较困难,尤其是当工作负载总量接近或超过可从计算设备或资源服务器中获得的计算资源时。但是,随着工作负载量的增加,现有工作负载调度方法容易降低效率和性能,并且解决调度器之间的冲突会成为系统性能的瓶颈。

### 发明内容

[0007] 一种分配计算资源以在分布式计算系统中执行工作负载的方法,包括:执行与第一资源提供管理器相关联的第一资源获取周期,所述第一资源获取周期包括:所述第一资源提供管理器向一个或多个调度器中的每个传输第一资源提供,所述资源提供指定第一多个计算资源中的每个的状态;从至少一个所述调度器接收相应的资源请求,所述相应的资源请求基于所述第一多个计算资源中的每个的所述状态;所述第一资源提供管理器基于所述相应的资源请求来计算分配给所述一个或多个调度器的资源;以及根据所述相应的资源

分配将所述第一多个计算资源中的资源分配给所述一个或多个调度器。

[0008] 在任意前述实施例中,所述方法可以包括执行工作负载调度周期,包括:调度与每个所述调度器相关联的第一工作负载,以使用分配给其中一个所述调度器的计算资源来执行。

[0009] 在任意前述实施例中,所述方法可以包括在所述第一资源获取周期过程中开始第二工作负载调度周期。

[0010] 在任意前述实施例中,所述方法可以包括执行与第二资源提供管理器相关联的第二资源获取周期,所述第二资源获取周期包括:所述第二资源提供管理器向所述一个或多个调度器中的每个传输第二资源提供,所述第二资源提供指定不同于所述第一多个计算资源的第二多个计算资源中的每个的状态;从所述一个或多个调度器中的至少一个接收资源请求,所述资源请求基于所述第二多个计算资源中的每个的所述状态;以及将所述第二多个计算资源中的资源分配给任何所述一个或多个调度器。

[0011] 在任意前述实施例中,所述第一资源获取周期和所述第二资源获取周期可以同步。

[0012] 在任意前述实施例中,所述至少一个所述相应的资源请求包括获取所述第一多个计算资源中的资源量的请求,所述资源量指定为一个范围。

[0013] 在任意前述实施例中,所述相应的资源请求可以指定所需的亲和性。

[0014] 在任意前述实施例中,所述相应的资源请求可以指定所需的反亲和性。

[0015] 在任意前述实施例中,所述第一多个资源中的每个的所述状态可以包括以下各项中的至少一个:所述第一多个计算资源中当前正在使用的资源的标识、所述第一多个计算资源中当前免费使用的资源的标识、与所述第一多个计算资源中的特定资源相关联的所有权状态以及至少两个所述调度器之间的资源共享策略。

[0016] 在任意前述实施例中,所述第一资源提供可以基于与其中一个所述调度器相关联的一个或多个订阅。

[0017] 在任意前述实施例中,所述方法可以包括:从其中一个所述调度器接收补充资源请求;以及基于所述补充资源请求将所述第一多个计算资源中的其它资源分配给其中一个所述调度器。

[0018] 在任意前述实施例中,分配所述第一多个计算资源中的所述其它资源可以包括:基于所述补充资源请求从不同于发送所述补充资源请求的所述调度器的调度器将至少一个先前获取的资源释放给所述第一多个计算资源;以及将所述释放的至少一个先前获取的资源分配给发送所述补充资源请求的所述调度器。

[0019] 在任意前述实施例中,所述方法可以包括:其中一个所述调度器执行工作负载调度周期以调度与所述调度器相关联的第一工作负载,以使用分配给并行于且独立于其它调度器的所述调度器的计算资源来执行。

[0020] 在任意前述实施例中,所述方法可以包括:资源提供管理器从至少一个调度器抢占资源并基于至少一个资源共享策略将所述抢占到的资源分配给至少一个其它调度器。

[0021] 一种示例系统,包括:至少一个处理器;存储器,包含计算机可读指令,所述计算机可读指令在由所述至少一个处理器执行时使所述至少一个处理器执行包括以下操作的方法:执行与第一资源提供管理器相关联的第一资源获取周期,所述第一资源获取周期包括:

向一个或多个调度器中的每个传输第一资源提供,所述资源提供指定第一多个计算资源中的每个的状态;响应于从至少一个所述调度器接收相应的资源请求,基于所述相应的资源请求来计算分配给所述一个或多个调度器的资源,所述相应的资源请求基于所述第一多个计算资源中的每个的所述状态;以及根据所述相应的资源分配将所述第一多个计算资源中的资源分配给所述一个或多个调度器。

[0022] 在任意前述实施例中,所述方法可以包括:执行工作负载调度周期,包括:调度与其中一个所述调度器相关联的第一工作负载,以使用分配给其中一个所述调度器的计算资源来执行。

[0023] 在任意前述实施例中,所述方法可以包括在所述第一资源获取周期过程中开始第二工作负载调度周期。

[0024] 在任意前述实施例中,所述方法可以包括:执行与第二资源提供管理器相关联的第二资源获取周期,所述第二资源获取周期包括:所述第二资源提供管理器向所述一个或多个调度器中的每个传输第二资源提供,所述第二资源提供指定不同于所述第一多个计算资源的第二多个计算资源中的每个的状态;从所述一个或多个调度器中的至少一个接收资源请求,所述资源请求基于所述第二多个计算资源中的每个的所述状态;以及将所述第二多个计算资源中的资源分配给任何所述一个或多个调度器。

[0025] 在任意前述实施例中,所述第一资源获取周期和所述第二资源获取周期可以同步。

[0026] 在任意前述实施例中,至少一个所述相应的资源请求可以包括获取所述第一多个计算资源中的资源量的请求,其中,所述资源量指定为一个范围。

[0027] 在任意前述实施例中,所述相应的资源请求可以指定所需的亲和性。

[0028] 在前述实施例中的任一者中,所述相应的资源请求可以指定所需的反亲和性。

[0029] 在任意前述实施例中,所述第一多个资源中的每个的所述状态可以包括以下各项中的至少一个:所述第一多个计算资源中当前正在使用的资源的标识、所述第一多个计算资源中当前免费使用的资源的标识、与所述多个计算资源中的特定资源相关联的所有权状态以及至少两个所述调度器之间的资源共享策略。

[0030] 在任意前述实施例中,所述第一资源提供可以基于与其中一个所述调度器相关联的一个或多个订阅。

[0031] 在任意前述实施例中,所述方法还可以包括:从其中一个所述调度器接收补充资源请求;以及基于所述补充资源请求将所述第一多个计算资源中的其它资源分配给其中一个所述调度器。

[0032] 在任意前述实施例中,分配所述第一多个计算资源中的所述其它资源可以包括:基于所述补充资源请求从不同于发送所述补充资源请求的所述调度器的调度器将至少一个先前获取的资源释放给所述第一多个计算资源;以及将所述释放的至少一个先前获取的资源分配给发送所述补充资源请求的所述调度器。

[0033] 一种示例计算机可读存储介质上存储有计算机可执行指令,所述计算机可执行指令在由一个或多个处理器执行时使得所述一个或多个处理器执行分配计算资源的方法以执行分布式计算系统中的工作负载,所述方法包括:执行与第一资源提供管理器相关联的第一资源获取周期,所述第一资源获取周期包括:向一个或多个调度器中的每个传输资源

提供,所述资源提供指定第一多个计算资源中的每个的状态;响应于从至少一所述调度器接收相应的资源请求,基于所述相应的资源请求来计算分配给所述一个或多个调度器的资源,所述相应的资源请求基于所述第一多个计算资源中的每个的所述状态;以及根据所述相应的资源分配将所述第一多个计算资源中的资源分配给所述一个或多个调度器。

## 附图说明

[0034] 附图中描述了以下示例实施例:

[0035] 图1为示例分布式计算系统的方框图;

[0036] 图2A为示例资源服务器的方框图;

[0037] 图2B为示例计算设备的方框图;

[0038] 图3为示例资源管理系统的方框图;

[0039] 图4为调度工作负载的示例方法的流程图;

[0040] 图5为从调度器角度为调度器分配资源以及调度工作负载以通过计算资源来执行的示例方法的流程图;

[0041] 图6A为示例资源管理系统的序列图,示出了并行操作的资源获取周期和工作负载调度周期;以及

[0042] 图6B为操作中的示例资源管理系统的序列图。

[0043] 为了进行说明,这些附图描述了示例实施例的各方面。

## 具体实施方式

[0044] 图1为示出示例分布式计算系统100的图。在分布式计算系统100中,一个或多个计算设备102可以直接或间接连接到一个或多个资源服务器103,以访问或利用由资源服务器103提供的一个或多个资源150。

[0045] 分布式计算系统100包括硬件和软件组件。例如,如图所示,分布式计算系统100包括计算设备102、资源服务器103和资源管理系统109的组合,它们通过网络107连接。如图所示,资源服务器103具有一个或多个资源150,可以分配该一个或多个资源150以对来自一个或多个计算设备102的工作负载进行计算。资源服务器103提供存储器(例如随机存取存储器(Random Access Memory,RAM))、处理器或处理器内核等处理单元、图形处理单元(graphics processing unit,GPU)、存储设备、通信接口等,它们在此分别或统一称为资源150。

[0046] 资源管理系统109可用于协调对资源服务器103中的资源150的分配,以执行由计算设备102生成的工作负载。在一些实施例中,除了资源服务器103中的资源之外,资源150还包括计算设备102中的资源。在一些实施例中,资源服务器103生成工作负载以通过计算资源150来执行。如图所示,资源管理系统109实现为一个单独的硬件设备。资源管理系统109还可以在一个或多个资源服务器103上以软件、硬件或其组合的形式实现。

[0047] 计算设备102可以包括个人计算机、膝上型电脑、服务器、工作站、超级计算机、智能电话、平板电脑、可穿戴计算设备等。如图所示,计算设备102和资源服务器103可以通过一个或多个局域网、广域网、无线网络、因特网等网络107互连。

[0048] 分布式计算系统100可以包括位于一个或多个资源服务器103处的一个或多个处

理器101。一些资源服务器103可以具有多个处理器101。

[0049] 在一些实施例中,分布式计算系统100是异构的。也就是说,分布式计算系统100的硬件和软件组件可以彼此不同。例如,一些计算设备102可以具有不同的硬件和软件配置。同样地,一些资源服务器103也可以具有不同的硬件和软件配置。在其它实施例中,分布式计算系统100是同构的。也就是说,计算设备102可以具有类似的硬件和软件配置。同样地,资源服务器103也具有类似的硬件和软件配置。

[0050] 在一些实施例中,分布式计算系统100可以是物理上或逻辑上的单个设备,例如单个计算设备102,也可以是具有一个或多个资源150的单个资源服务器103。在一些实施例中,分布式计算系统100可以包括以各种方式连接的多个计算设备102。

[0051] 一些资源150可以与单个计算设备102或一组设备在物理或逻辑上相关联,而其它资源150是可以是在计算设备102之间共享并由分布式计算系统100中的多个设备使用的共享资源。也就是说,一些资源150只能分配给来自计算设备102的子集的工作负载,而其它资源150可以分配给来自任何计算设备102的工作负载。在一些实施例中,分布式计算系统100根据共享策略进行操作。共享策略是规定如何使用特定资源的规则。例如,资源管理系统109可以实施共享策略,该共享策略规定来自特定计算设备102的工作负载应该使用特定资源服务器103中的资源150来执行。共享策略可以针对资源服务器103中的特定类型的资源150设置,并且还可以更广泛地应用于资源服务器103中的全部资源或应用于系统范围内。计算设备102还可以表示用户、用户组或租户,或者项目。共享策略可以规定如何在用户、用户组或租户,或者项目之间共享资源。

[0052] 分布式计算系统100中的资源150与或可以与一个或多个属性相关联。这些属性可以包括资源类型、资源状态、资源位置、资源标识符/名称、资源值、资源容量、资源能力或任何其它资源信息等,这些信息可以作为选择或识别适合于由一个或多个工作负载使用的资源的标准。

[0053] 分布式计算系统100可以在概念上视为具有多种硬件、软件和其它构件资源的单个实体,能够用于运行来自分布式计算系统100自身组件的工作负载以及分布式计算系统100外部的计算设备102的工作负载。

[0054] 可以使用各种策略来调度计算资源上的工作负载。根据第一种策略,单个集中调度器,这里也称为“单片调度器”,处理所有工作负载到机器调度功能。然而,随着计算机系统的扩大,使用单个调度器的系统的性能可能会降低。而且,可能难以在单个调度器系统中实现定制的调度或共享策略,因为单个调度器系统的速度和效率会迅速变慢变低。

[0055] 根据另一种策略,可以分两个层次完成调度。也就是说,可以使用单个资源管理器,将集群中的所有可用资源从资源服务器103提供给多个独立调度器,一次一个调度器。当将资源提供给调度器时,调度器能够获取资源并将工作负载分配给那些所获取的资源。

[0056] 根据另一种策略,资源管理器一次只向一个调度器提供群集中的可用资源的子集。因此,不止一个调度器无法尝试同时使用相同的计算机来执行不同的工作负载。然而,虽然这样的系统有效地锁定了该特定机器以供一个调度器使用,但是会导致效率低。此外,虽然这样的系统有效地以先到先得的原则运行,但是会使实现工作负载策略的优先级或优先顺序变得困难。最后,这样的系统会有效地耗尽第二优先级或低优先级的调度器,因为第一优先级或最高优先级的调度器可能会用尽所有资源,并且剩余的资源不足以用于一

个或多个其它调度器。一种选择是允许较高优先级的调度器抢占来自所获取的计算机器的工作负载,但是这种选择导致效率更低,因为抢占的工作负载将不得不重新启动。

[0057] 根据另一种策略,也称为“乐观”策略,资源管理器向每个调度器提供资源列表,用以识别分布式计算系统内的所有资源。因此,可以同时向不止一个调度器提供资源。在不止一个调度器尝试将工作负载分配给同一计算机器的情况下,资源管理器解决冲突并决定哪个调度器成功地将其工作负载分配给计算机器。冲突可以根据调度器之间的权限优先顺序、优先级和工作重要性或资源管理器中实现的其它策略来解决。然而,冲突的结果是其中一个调度器被拒绝,并且需要进行另一次尝试以将被拒绝的工作负载分配给另一个合适的计算机器。因此,在这种情况下,一个或多个调度器会在执行最终被拒绝的调度工作上浪费时间和资源。这是一种二进制方法,在该方法中,调度器的请求或被接受或被拒绝,没有灵活性。此外,该策略要求确切数量的计算机器应该由调度器请求,这些计算机器参考来自特定资源服务器或计算设备的工作负载的资源请求。也就是说,调度器在工作负载到机器的基础上发出请求。

[0058] 在使用乐观调度器的系统中,每个乐观调度器不知道其它乐观调度器的工作负载要求和调度策略。在没有完整信息的情况下,调度器存在一种倾向:就是提高所请求的计算资源量。然而,在二进制方法中,当可能已经接受充足且更适度的资源请求时,会导致拒绝过高的请求。在请求计算机器和资源方面允许一定程度的灵活性将是有益的。

[0059] 随着调度策略和调度性能指标的实现越来越复杂,资源管理器在解决冲突时做出的决策也变得越来越复杂且耗时。冲突解决的过程会成为大规模系统中的瓶颈,尤其是当执行工作负载由于多次调度请求拒绝而被延迟时。

[0060] 本文描述的各实施例至少可以减轻或消除部分与上述策略相关联的上述缺陷。

[0061] 图2A是示例资源服务器103的方框图。如图所示,资源服务器103包括一个或多个处理器101、存储器104、存储设备106、I/O设备108和网络接口110及其组合。资源服务器103中的处理器101、存储器104、存储设备106、I/O设备108和网络接口110中的一个或多个作为资源150,用于执行来自分布式计算系统100中的计算设备102的工作负载。

[0062] 处理器101是任何合适类型的处理器,例如执行ARM或x86指令集的处理器。在一些实施例中,处理器101是图形处理单元(graphics processing unit,GPU)。存储器104是由处理器101访问的任何合适类型的随机存取存储器。存储设备106可以是存储器中的一个或多个模块、硬盘驱动器或其它永久计算机存储设备。

[0063] I/O设备108包括用户界面设备,例如屏幕,包括能够将渲染图像显示为输出并且以触摸的形式接收输入的电容式屏幕或其它触敏屏幕。在一些实施例中,I/O设备108另外或可替代地还包括扬声器、麦克风、加速度计和全球定位系统(global positioning system,GPS)接收器等传感器、小键盘等中的一个或多个。在一些实施例中,I/O设备108包括将计算设备102连接到其它计算设备的端口。在一个示例中,I/O设备128包括连接到外围设备或主机计算设备的通用串行总线(universal serial bus,USB)控制器。

[0064] 网络接口110能够将计算设备102连接到一个或多个通信网络。在一些实施例中,网络接口110包括有线接口(例如有线以太网)和WiFi或蜂窝(例如GPRS、GSM、EDGE、CDMA、LTE等)等无线电设备中的一个或多个。

[0065] 资源服务器103在软件程序的控制下操作。计算机可读指令存储在存储设备106

中,并且由处理器101在存储器104中执行。

[0066] 图2B是示例计算设备102的方框图。计算设备102可以包括一个或多个处理器121、存储器124、存储设备126、一个或多个输入/输出(input/output,I/O)设备128、网络接口130及其组合。

[0067] 处理器121是任何合适类型的处理器,例如执行ARM或x86指令集的处理器。在一些实施例中,处理器121是图形处理单元(graphics processing unit,GPU)。存储器124是由处理器121访问的任何适当类型的随机存取存储器。存储设备126可以是存储器中的一个或多个模块、硬盘驱动器或其它永久计算机存储设备。

[0068] I/O设备128包括用户界面设备,例如屏幕,包括能够将渲染图像显示为输出并且以触摸的形式接收输入的电容式屏幕或其它触敏屏幕。在一些实施例中,I/O设备128另外或可替代地还包括扬声器、麦克风、加速度计和全球定位系统(global positioning system,GPS)接收器等传感器、小键盘等中的一个或多个。在一些实施例中,I/O设备128包括将计算设备102连接到其它计算设备的端口。在一个示例中,I/O设备128包括连接到外围设备或主机计算设备的通用串行总线(universal serial bus,USB)控制器。

[0069] 网络接口130能够将计算设备102连接到一个或多个通信网络。在一些实施例中,网络接口130包括有线接口(例如有线以太网)和WiFi或蜂窝(例如GPRS、GSM、EDGE、CDMA、LTE等)等无线电设备中的一个或多个。

[0070] 计算设备102在软件程序的控制下操作。计算机可读指令存储在存储设备126中,并且由处理器121在存储器124中执行。

[0071] 图3为示例资源管理系统109的方框图。资源管理系统109包括一个或多个资源提供管理器302以及一个或多个调度器304。资源提供管理器302和调度器304通信地耦合到资源150。在一些实施例中,资源提供管理器302和调度器304彼此独立运行,可能位于不同的地理位置。资源150可以包括一个或多个资源服务器103中的资源。在一些实施例中,资源150包括资源服务器103和计算设备102中的资源。

[0072] 资源管理系统109可以由分布式计算系统100中的一个或多个计算设备102或资源服务器103中的一个或多个处理器101来实现。在一些实施例中,资源管理系统109是可以在分布式计算环境顶层运行的基础设施中间件。分布式环境可以包括不同种类的硬件和软件。

[0073] 资源管理系统109处理资源管理、工作负载管理和调度。工作负载可以指分布式计算系统100中待运行的任何进程、作业、业务或任何其它计算任务。例如,工作负载可以包括批量作业(例如高性能计算(high performance computing,HPC)批量作业)、串行和/或并行批量任务、实时分析、虚拟机、容器等。工作负载的特征可能有相当大的差异。例如,工作负载可以是CPU密集型的工作负载、内存密集型的工作负载、批量作业(需要快速周转的短任务)、服务作业(长时间运行的任务)或实时作业。

[0074] 资源提供管理器302向调度器304提供和分配资源150。在一些实施例中,单个资源提供管理器302为每个调度器304提供和分配资源150。在其它实施例中,资源管理系统109包括不止一个资源提供管理器302,并且资源提供管理器302中的每个负责将资源150的特定子集分配给一个或多个调度器304。作为示例,如果存在多个资源服务器103,则可以有一个单独的资源提供管理器302用于分配每个相应的资源服务器103或服务器组中的资源

150。

[0075] 资源提供管理器302根据资源获取周期进行操作。资源获取周期是调度器304获取的资源150保持不变的时段。在一些实施例中,出于策略原因可以抢占获取的资源,然后释放。资源获取周期的持续时间可以为一分钟或五分钟等。在资源获取周期过去之后,资源提供管理器302可以更改对一个或多个调度器304的资源150的分配。

[0076] 在存在不止一个资源提供管理器302的一些实施例中,各资源提供管理器302同步并共享公共资源获取周期。可替代地,不同的资源提供管理器302可以根据不同的资源获取周期进行操作。虽然在本文描述的示例实施例中,资源获取周期是根据时间长度来描述的,但是资源获取周期可替代地可以对应于一段计算周期。

[0077] 调度器304可用于临时从资源提供管理器302获取资源150,然后将工作负载分配给所获取的资源150,以执行工作负载。

[0078] 调度器304根据工作负载调度周期进行操作。工作负载调度周期是由调度器304执行的一系列操作,然后重复。在一些实施例中,特定调度器的工作负载调度周期独立于任何其它调度器304的工作负载调度周期,并且独立于任何其它资源提供管理器302的资源获取周期。工作负载调度周期与工作负载调度时间段相关联。工作负载调度时间段是一个时间段,在该时间段中,已调度的由特定资源150执行的工作负载保持不变。在一些实施例中,工作负载调度周期可由使得工作负载调度周期过早结束的事件来驱动。在一些实施例中,工作负载调度时间段是固定的并以设定的时间间隔、时钟周期的数量等来定义。在其它实施例中,工作负载调度时间段是可变的并通过要执行工作负载调度周期所花的时间来定义。例如,调度时间段的长度可以是五秒。在一些实施例中,资源管理系统109中的每个调度器304具有一个工作负载调度周期,但是工作负载调度周期的调度时间段不同。在一些实施例中,一个或多个调度器304具有同步的工作负载调度周期,这些同步的工作负载调度周期的调度时间段相同。

[0079] 在调度过程中,调度器304确定是否有任何其它工作负载需要执行或者在完成先前分配的工作负载之后是否有任何获取的资源可用。然后,调度器304可以将其它工作负载分配给部分或全部可用的所获取的资源150。

[0080] 调度器304可以根据调度策略将工作负载分配给所获取的资源。可以实施各种调度策略以实现不同的性能目标。例如,调度器304的目标可以是:最大化吞吐量(每单位时间完成的工作量),最小化延迟(从创建工作负载到完成工作负载所需的时间),或者保持一定程度的公平性(例如将一定量的资源分配给不同的调度任务或计算设备)。

[0081] 实际上,这些性能目标通常是冲突的。例如,最大化吞吐量需要调度工作负载以在任何给定时间内尽可能地使用所获取的资源。然而,最小化紧急工作负载的延迟可能包括使某些所获取的资源不被调度器使用,以便在接收到紧急工作负载时这些资源可用。因此,尝试实现既最大化吞吐量又最小化延迟的调度策略是很困难的。

[0082] 在一些实施例中,资源获取时间段比工作负载调度时间段长。因此,在一些实施例中,每个调度器304获取的且可用的资源对于至少两个连续的工作负载调度周期来说是不变的。

[0083] 调度器304可以以硬件、软件或硬件和软件的组合(例如由处理器101执行的程序)的方式来实现。调度器304可用于从一个或多个计算设备102接收一个或多个工作负载,并

调度工作负载以供从资源提供管理器302获取的资源150来执行。

[0084] 图4为调度工作负载的简化示例方法400的流程图。在图4中,除方框420(从调度器304的角度)之外,每个方框都从资源提供管理器302的角度呈现。应该理解的是,方法400中的各方框在下面结合图5和图6进一步详细解释。

[0085] 在402处,向一个或多个调度器304发送关于资源的当前全状态资源信息。在一些实施例中,基于调度器304的订阅发送全状态资源信息。全状态资源信息由资源提供管理器302提供。当前全状态资源信息包括当前正在使用的资源的列表、当前未使用的资源以及任何资源共享或所有权策略中的一个或多个,或者它们的组合和/或子集。在一些实施例中,调度器304可以订阅某些类型的待提供的资源或未使用的资源,这可以减少与其它调度器304发生冲突的可能性。在一些实施例中,最初没有为调度器304分配任何资源。

[0086] 在一些实施例中,资源管理系统具有多个资源提供管理器302,每个资源提供管理器302管理不同的资源150组。可选地,资源获取周期可以在每个资源提供管理器302之间同步。可替代地,资源提供管理器302可能具有不同(例如相对较短或较长)的资源获取周期。

[0087] 在404处,每个调度器304向资源提供管理器302发送资源获取和释放请求。应当理解,在存在不止一个资源提供管理器302的实施例中,调度器304可以向不止一个资源提供管理器302发送资源获取和释放请求。

[0088] 在一些实施例中,调度器304基于对全状态资源信息和未决工作负载的审查以及任何有效的共享或性能策略或其任何组合来确定是释放还是获取资源。每个资源获取请求包括调度器304所请求的每个资源的类型和数量。资源释放请求向资源提供管理器302指示特定调度器304不再需要当前分配给该调度器304的资源以及该资源可以被释放并且可能提供给其它调度器304。

[0089] 在一些实施例中,调度器304的获取和释放请求包含在到资源提供管理器302的单个传输中。调度器304可以请求以下各项中的任何一个:a)维护分配给特定调度器304的当前资源组,b)获取其它资源给特定调度器304,c)释放特定调度器304先前获取的一个或多个资源,及d)其任何组合。

[0090] 在一些实施例中,调度器304请求的是一系列资源而不是特定量。例如,调度器304可以请求一种资源的最小量和最大量。相对于对一定量的资源的请求,范围的使用在分配资源时提供了更大的灵活性。

[0091] 例如,调度器对500MB内存的请求会被只有400MB可用于分配的内存的现有系统拒绝。因此,那些资源将不会分配给调度器304,并且需要这些资源的工作负载将保留在该调度器304中并且在下一次资源获取时被纳入到资源请求中。然而,同一调度器304对最小400MB内存和最大600MB内存的请求导致为该调度器分配400MB。这样可以使一些工作负载更快地调度到所获取的资源上,因此相对于现有系统可使一些工作负载更快地完成。

[0092] 在一些实施例中,分配给调度器的资源可以从分布式计算系统100内的任何位置上的任何计算资源获得。例如,可以通过从一个计算资源获取200MB以及从单独的计算资源获取200MB来满足调度器304对400MB内存的请求。可选地,资源请求可以指定待获取资源满足与物理邻近度、位置等有关的要求。

[0093] 例如,在一些实施例中,调度器304传输多个资源请求和亲和性或反亲和性参数。亲和性或反亲和性参数可以作为资源请求的一部分(例如以字段的形式)包含其中,或者可

以由调度器304分别发送并且添加亲和性或反亲和性限制(例如,标识由调度器304做出的特定请求的索引号)所适用的请求的标识符。亲和性是指提供给该调度器的资源满足彼此之间的某些地点关系的要求。例如,调度器304可以请求200MB内存并且还请求20个CPU。在一些实施例中,调度器304指定200MB内存和20个CPU位于同一机架的资源上。在接收到来自调度器304的资源请求时,资源提供管理器302考虑需要将分配给该调度器304的资源驻留在同一机架上。

[0094] 来自调度器304的资源请求还可以或者可替代地包括反亲和性参数。反亲和性是指分配给调度器的资源在物理上彼此分离的要求。在调度器304需要对掉电或其它物理灾难具有一定程度的弹性的情况下,反亲和性可能很有用。例如,通过使资源位于不同的物理位置,掉电影响调度器304所获取的所有资源的可能性就小一些,并且允许至少一些工作负载继续执行。例如,调度器304可以请求300MB内存和3个CPU的两种分配。在一些实施例中,调度器304指定必须在明确位于不同机架上的两个节点上找到分配好的两种分配。然后,资源提供管理器302考虑必须将分配给该调度器304的资源驻留在两个不同的机架上。虽然上述示例涉及机架,但是亲和性和反亲和性也可以指代数据中心、国家、地理区域等的位置。例如,亲和性或反亲和性参数中的任一个可以设置为多个可能值中的一个,每个可能值对应于邻近或分离的程度。亲和性参数为1可以指示必须从同一机架分配资源;亲和性参数2可以指示必须从同一建筑物或局域网等分配资源。反亲和性参数值1可以指示资源不能位于同一机架上;反亲和性参数值2可以指示资源不能位于同一建筑物或局域网中。

[0095] 在406处,资源提供管理器302从调度器304接收每个资源获取和释放请求。资源提供管理器302解决可能在各调度器304的请求之间产生的任何冲突。在一些实施例中,资源提供管理器302抢占某些调度器304中的资源并根据资源共享策略将抢占的资源分配给其它调度器304。例如,如果调度器304请求回收由该调度器304拥有且当前正在由另一调度器使用的一些资源,则资源提供管理器302可以对调度器304发起抢占动作。资源提供管理器302最终确定每个调度器304的资源分配。当确定每个调度器304的分配时,资源提供管理器302可考虑任何有效的共享策略或性能策略。

[0096] 在408处,资源提供管理器302向调度器304中的每个传输资源分配信息。资源分配信息包括将由特定调度器获取的每个资源的类型和数量。资源分配信息还可以包括关于每个待获取资源所在的资源服务器103或计算设备102的信息。在一些实施例中,由调度器304先前获取的一个或多个资源释放给资源提供管理器302,以便可能分配给其它调度器304。

[0097] 在410处,调度器304更新内部记录以反映最新的资源分配。在一些实施例中,调度器304在调度器304内执行的获取线程中更新所获取的资源的内部池。在更新所获取的资源的内部池之后,每个调度器304可以调度工作负载以在所获取的资源上执行工作负载。如420处所示,所获取的资源上的工作负载的调度和执行与资源获取周期并行发生,并且在资源提供管理器302的资源获取周期的剩余时间内继续,或者在工作负载的调度和执行的持续时间内继续,除非被资源提供管理器302抢占,如下面结合图6所述。在一些实施例中,调度器304包括处理工作负载的调度和执行的调度线程。调度线程与调度器304中的获取线程并行执行。调度器304根据工作负载调度周期进行操作。在工作负载调度周期完成之后,调度器304可以将更多的工作负载分配给所获取的资源,如420所示。

[0098] 在412处,确定是否开始下一个资源获取周期。如果否,则方法400等待并返回到

412。同时,在420处,调度器304调度更多工作负载以在所获取的资源上执行工作负载。如果确定下一个资源获取周期将开始,则方法返回到402,资源提供管理器302将向一个或多个调度器304中的每一个发送全状态资源信息。关于是否开始下一个资源获取周期可以基于从执行前一个资源获取周期起所经历的时间来确定。

[0099] 图5为从调度器304中的获取线程的角度示出为调度器分配资源以及调度工作负载以供计算资源执行的示例方法500的流程图。在502处,在资源获取周期开始时,调度器304从资源提供管理器302接收资源提供。资源提供包含由资源提供管理器302管理的所有资源150的全状态信息。

[0100] 应该注意的是,在包含不止一个资源提供管理器302的实施例,调度器304从每个资源提供管理器302接收资源提供。每个资源提供管理器302不一定具有相同的资源获取周期。此外,每个资源提供管理器302不一定与其它资源提供管理器302同步。因此,调度器304可以在不同的时间内从不同的资源提供管理器302接收不同的资源提供。

[0101] 在一些实施例中,两个或更多个资源提供管理器302同步并共享相同的资源获取周期。在那些实施例中,调度器304将在类似时间内接收来自同步资源提供管理器302的资源提供,从而向调度器提供可从同步资源提供管理器302获得的资源150的更全面的信息。

[0102] 在504处,调度器304中的获取线程评估在资源提供中接收到的全状态资源信息,并确定调度器304应当获取的每个资源的类型和数量。在一些实施例中,调度器304通过评估当前待决的工作负载进行确定。调度器304还可以考虑工作负载对资源调度策略进行适当“空转”(即测试运行)进行确定。调度器304还可以估计下一个资源获取周期中所需的资源。

[0103] 在506处,在调度器304中的获取线程确定待获取和/或释放的资源类型和量之后,调度器304向资源提供管理器302发送资源获取请求。资源获取请求可以包括待获取资源的范围(例如一种资源的最小量和最大量)以及对资源的亲和性或反亲和性限制(例如要求资源位于相同的地理位置,或位于不同的地理位置)中的一个或多个。这为资源提供管理器302提供了更大的灵活性,能够满足潜在的多种不同调度器304的请求。

[0104] 资源获取请求可以包括资源释放请求。当调度器304确定当前获取的资源多于满足调度器的调度和性能策略所需的资源时,调度器304可以请求释放先前获取的资源。在存在一个资源提供管理器302的实施例中,调度器304向资源提供管理器302发送资源释放请求。在包括不止一个资源提供管理器302的实施例中,调度器304可以向资源提供管理器302中的一个或多个发送资源释放请求。

[0105] 在一些实施例中,调度器304可以在任何时间点发送资源释放请求,包括在开始下一个资源获取周期之前。这样可以更早地释放未使用的资源,从而更有效地利用资源。

[0106] 在一些实施例中,资源提供管理器302用于抢占调度器304已经获取的一个或多个资源。也就是说,不是调度器304自愿向资源提供管理器302发送资源释放请求,而是资源提供管理器302可用于强制释放调度器304已经获取的资源。在一些实施例中,资源提供管理器302可以在任何时间点发起资源抢占,不论资源获取周期是否在进行中。

[0107] 在508处,调度器304从资源提供管理器302接收资源获取和释放决策。在一些实施例中,资源提供管理器302在向调度器304发送将资源获取和释放决策之前解决冲突。在存在多个资源提供管理器302的实施例中,调度器304从每个资源通过管理器302接收资源获

取和释放结果,但是每个资源提供管理器302不一定同时传递结果。

[0108] 在510处,在接收到获取和释放决策时,调度器304基于获取和释放决策来更新调度器304中的所获取的资源的内部池。在一些实施例中,获取和释放决策可能不会使调度器304获取或释放任何资源。从该时刻直到下一次从资源提供管理器302接收资源获取和释放决策的时间,由该调度器304获取的资源可保持静态。然而,如上所述,如果需要,资源提供管理器302可以在任何时间点发起对资源的抢占。

[0109] 在520处,调度器304中的调度线程在调度器304的工作负载调度周期的持续时间内调度工作负载以在所获取的资源150上执行工作负载。在一些实施例中,调度器304中的调度线程与调度器304中的获取线程并行执行。调度器304可以根据各种调度算法和策略将工作负载调度到资源。

[0110] 例如,调度器可以实施如下策略:在任何给定时间内使用尽可能多的资源。通过所获取的资源调度和执行工作负载,一直持续到资源获取时间段过去。在514处,如果下一个资源获取周期尚未开始,则方法500等待并循环回到514。同时,在520处,调度器304中的调度线程继续调度其它工作负载。在514处,当下一个资源获取周期已经开始时,方法500返回到502,从资源提供管理器302接收资源提供。

[0111] 图6A为操作中的示例资源管理系统109的序列图,示出了并行操作的资源获取周期和工作负载调度周期。调度器304a和304b均包括调度线程和资源获取线程。资源获取周期开始于资源提供管理器302向调度器304a和304b分别发送资源提供602。资源提供包括由资源提供管理器302管理的所有资源的全状态信息。全状态信息可以包括,例如当前未使用的资源的列表、当前正在使用的资源的列表、一个或多个调度器304的有效资源所有权和共享计划中的一个或多个。在一些实施例中,可以仅订阅调度器304来接收与调度器304拥有的资源、特定类型的资源或未使用的资源有关的资源信息,从而减少与其它调度程序发生冲突的可能性。在一些实施例中,在602处开始第一资源获取周期之前,不会为任何调度器304分配资源150。

[0112] 在接收全状态信息时,在604处,每个调度器304分析全状态资源信息。每个调度器304确定待获取资源的类型和数量。在一些实施例中,调度器304基于当前分配给调度器304的工作负载来执行上述确定。该估计还可以基于任何有效的调度策略算法。例如,调度器304可空转其工作负载对资源调度策略算法,以估计所需的资源。

[0113] 应该认识到,在本文描述的实施例中,由调度器150获取的资源可以与不止一个资源服务器103相关联。例如,资源150可以由分布式网络中的多个资源服务器103和/或计算设备102提供。尽管每个资源位于特定的资源服务器103中,但调度器304可以请求资源而不必指定从哪个特定资源服务器103获取资源。在一些实施例中,调度器304可以指定资源应来自哪些资源服务器103或哪些类型的资源服务器103。

[0114] 一旦调度器304已确定待获取资源的数量和类型,每个调度器304就在606处创建并发送资源获取请求。对于每个调度器304,资源获取请求包含待由特定调度器304获取的资源的数量和类型。在一些实施例中,一种或多种类型的资源的数量在资源获取请求中被指定为一个范围(例如一种资源的最小量和最大量)。当资源最终被分配给每个调度器304时,在资源获取请求中使用范围可以为资源提供管理器302提供更大的灵活性。在一些实施例中,资源获取请求包含亲和性或反亲和性限制。在资源获取请求中或与资源获取请求一

起使用亲和性或反亲和性参数可使调度器304满足分配给调度器304的工作负载的需求。资源获取请求还可包括释放先前获取的资源的指示,如上所述。

[0115] 在608处,资源提供管理器302基于接收到的资源获取请求来确定每个调度器304所获取的资源150的类型和数量。在一些实施例中,资源提供管理器302至少部分地基于资源所有权和共享计划来确定分配给每个调度器304的资源量。确定向哪些调度器提供哪些资源的过程在这里被称为资源获取冲突解决,并且在很大程度上受资源提供管理器302的策略的影响。

[0116] 在一些实施例中,可能没有足够的资源150可用来处理所有工作负载。在这种情况下,资源提供管理器302基于调度器之间的资源共享策略(例如公平性)和优先级来为调度器304分配资源。

[0117] 一旦资源提供管理器302在608处解决资源获取冲突,资源提供管理器302就在610处向每个调度器304发送资源分配。也就是说,每个调度器304从资源提供管理器302接收指令,该指令指定该资源已经分配给该调度器304。在612处,每个调度器304更新分配给该调度器304的资源150。如图所示,分配给该调度器304的资源150将在执行612之后的资源获取周期618的剩余时间内和随后的资源获取周期的一部分中依然分配给该调度器304,直到在612处所获取的资源是在随后的资源获取周期更新。

[0118] 在资源获取周期618结束后(例如,在一段时间过去之后),当资源提供管理器302根据资源提供管理器302管理的所有资源的全状态信息向每个调度器304发送随后的资源提供602时,该过程重复。随后的资源获取周期在604处继续,在604处,调度器确定待获取资源的类型和数量。在一些实施例中,存在多个资源提供管理器302,管理不同的资源150组。在仅包括一个资源提供管理器302的实施例中,资源提供管理器302管理所有可用资源150。

[0119] 应该注意的是,当向每个调度器304发送随后的资源提供602(因此开始随后的资源获取周期)时,前一个资源获取周期中每个调度器304a和304b所获取的资源150保持不变。因此,尽管调度器304a、304b和资源提供管理器302交换各种信息并按照下一个资源获取周期内的所需资源和冲突解决进行确定,但先前资源获取周期内的资源150依然分配给每个调度器304a、304b。

[0120] 相应地,调度器304a和304b上的调度线程可以继续将工作负载614和616分别调度到其所获取的资源150,以与资源获取线程并行执行。因此,资源提供管理器302对资源获取冲突的解决不会妨碍调度器304a、304b中的任何一个调度和执行工作负载。这在大规模计算环境中可能特别有益,在这种环境中,资源获取冲突可能需要复杂的冲突解决算法,否则会成为系统的瓶颈。

[0121] 应该注意的是,资源提供管理器302的资源获取周期618的持续时间可以比调度器304a、304b的工作负载调度周期620a、620b长。在一些实施例中,资源获取周期618明显比工作负载调度周期620a、620b长。因此,多个工作负载调度周期620a、620b可由调度器304a、304b在一个资源获取周期618内执行。这样可以使每个调度器304a、304b获取的资源在多个工作负载调度周期620a、620B中保持静态。当分配给调度器304的资源池在两个或更多个工作负载调度周期内保持不变时,调度器304还具有更大的灵活性来维护任何调度策略。例如,当分配给调度器的资源池在一段时间内不变时,调度器304可以更好地维持公平性(这可以是一种重要的调度策略)。

[0122] 例如,当调度器304a意识到所获取的资源在4个工作负载调度周期内保持不变时,相比于不存在资源获取周期的情况,调度器304a可以以更灵活的方式调度工作负载,因此调度器可用的资源池可以在每个工作负载调度周期之间改变。

[0123] 在一个具体示例中,如果调度器的调度策略要求应该对第一用户的任务使用50%的处理能力,并且应该对第二用户的任务使用50%的处理能力,则调度器304a可以使用多种可能方式在多个工作负载调度周期中调度和调整第一与第二用户之间的公平性。例如,调度器304a可以将50%的资源分配给第一用户并将50%的资源分配给第二用户。调度器304a还可以分配100%的处理能力以在两个周期内完成第一用户的任务,然后分配100%的处理能力以在随后的两个周期内完成第二用户的任务。在该示例中,如果第二用户的任务需要比两个周期更长的时间来完成,则调度器304a随后可以更改资源的分配以补偿和保持公平性。该方案将在调度器级别维护调度策略。在先前系统中,调度器在多个工作负载调度周期620a中都没有稳定的资源分配,这种方案并不可取,因为所分配的资源随后可能发生改变。因此,这里描述的系统和方法的一些实施例可使计算系统更高效地运行。

[0124] 还应该注意的是,根据一些实施例,资源提供管理器302同时从调度器304a、304b接收资源获取和/或释放请求。因此,资源提供管理器302从调度器304获得所有资源获取请求的完整信息。具有关于每个调度器304的资源请求的完整信息可以使资源提供管理器302解决资源获取冲突,同时维护分布式计算系统100中的多个调度器304之间的资源共享策略。此外,资源获取冲突解决过程整体上由资源提供管理器302执行,并且调度器304囤积资源的可能性较小。也就是说,不是资源提供管理器302一次考虑一个调度器304的需求(这可能导致资源用尽并且不可用于其它调度器304的后续资源请求),而是资源提供管理器302能够在解决冲突时考虑一组调度器304的需求,同时在资源提供管理器级别维持资源共享策略。

[0125] 还应该注意的是,在一些实施例中,资源获取周期和工作负载调度周期是分离并且并行执行。因此,调度器304继续在所获取的资源上调度和执行工作负载,同时资源提供管理器302并行工作以解决来自不同调度器304的资源获取请求之间的冲突。当计算系统和工作负载增大时,冲突的解决可能是一个复杂且耗时的过程。资源获取周期与工作负载调度周期的分离可防止或至少降低资源提供管理器302成为计算系统中的瓶颈的可能性,因为在资源提供管理器302解决资源获取冲突的同时调度器304继续调度和执行工作负载。

[0126] 图6B为操作中的示例资源管理系统109的序列图,示出了并行操作的资源获取周期和工作负载调度周期。在一些实施例中,调度器304独立于资源获取周期中的阶段发送资源请求。也就是说,调度器304不需要为了发送资源获取或释放请求而从资源提供管理器302接收资源提供。

[0127] 例如,如图6B所示,调度器304b可以接收意外激增的待执行的工作负载,并且可以在652处在下一个资源获取周期开始之前向资源提供管理器302发送资源请求636。在接收到资源请求636时,资源提供管理器302在638处评估资源的可用性。在640处,资源提供管理器将额外的资源分发给调度器304b。在642处,调度器304b更新其内部记录以反映额外的所获取的资源。在一些实施例中,资源提供管理器302在638处确定可用于分配给调度器304b的资源不足,抢占来自不同调度器304a的一个或多个资源并且将抢占的资源分配给调度器304b。

[0128] 在一些实施例中,资源提供管理器302在接收到资源请求636时不立即采取动作。相反,资源提供管理器302在652处等待直到下一个资源获取周期开始,并且考虑先前从调度器304b接收的资源请求636来确定每个调度器的资源分配。在一些实施例中,如果调度器304b在652处在开始下一个资源获取周期之前已经向资源提供管理器302发送资源请求636,则调度器304b可以不发送资源请求606以响应资源提供602。例如,如果调度器304b的资源需求在发送资源请求636之后没有改变,则调度器304b可以选择不发送其它资源请求以响应资源提供652。

[0129] 本发明的范围并不局限于说明书中所述的过程、机器、制造、物质组分、构件、方法和步骤的特定实施例。所属领域的一般技术人员可从本发明的公开中轻易地了解,可根据本发明使用现有的或即将开发出的,具有与本文所描述的相应实施例实质相同的功能,或能够取得与所述实施例实质相同的结果的过程、机器、制造、物质组分、构件、方法或步骤。相应地,所附权利要求范围旨在包括这些流程、机器、制造、物质组分、构件、方法或步骤。

[0130] 可以理解,上文描述和示出的具体实施例仅为示例性的。可以对这些示例实施例进行变形、选择其它配置、选择其它组件以及做出修改。本发明由权利要求书定义。

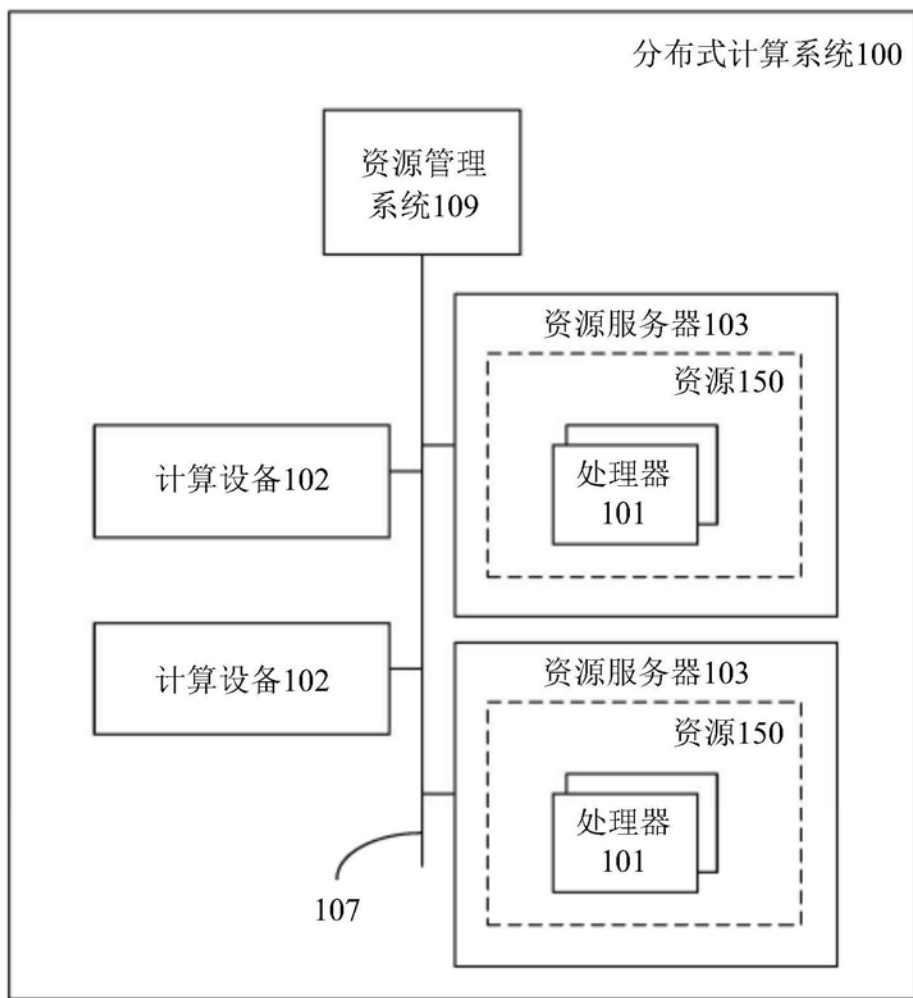


图1

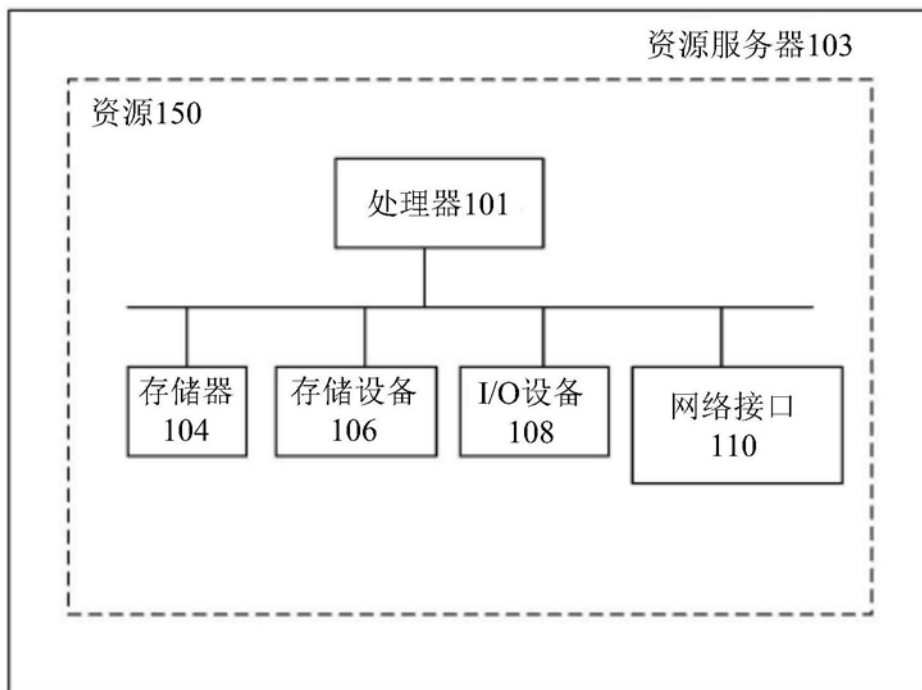


图2A

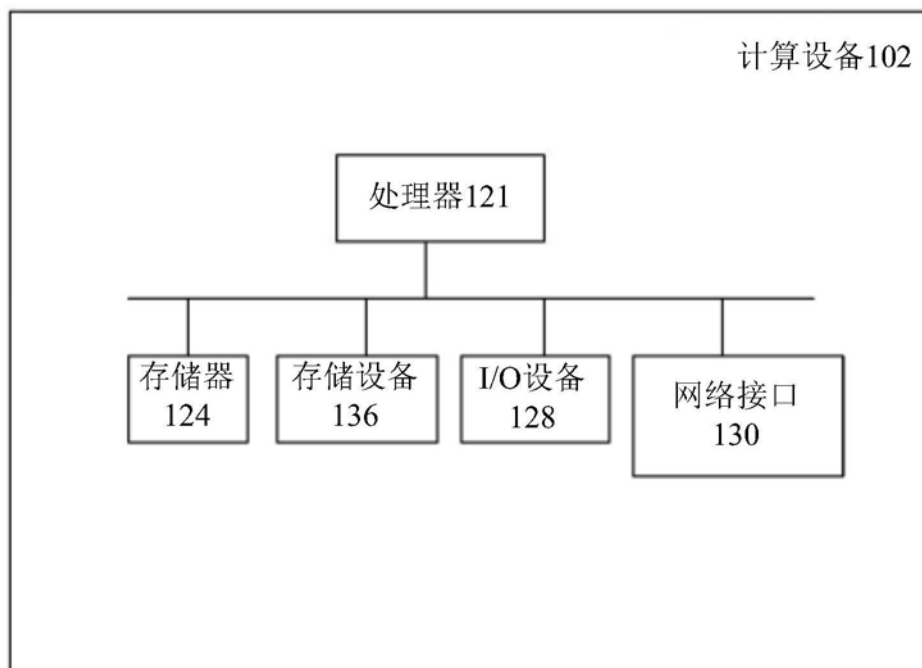


图2B

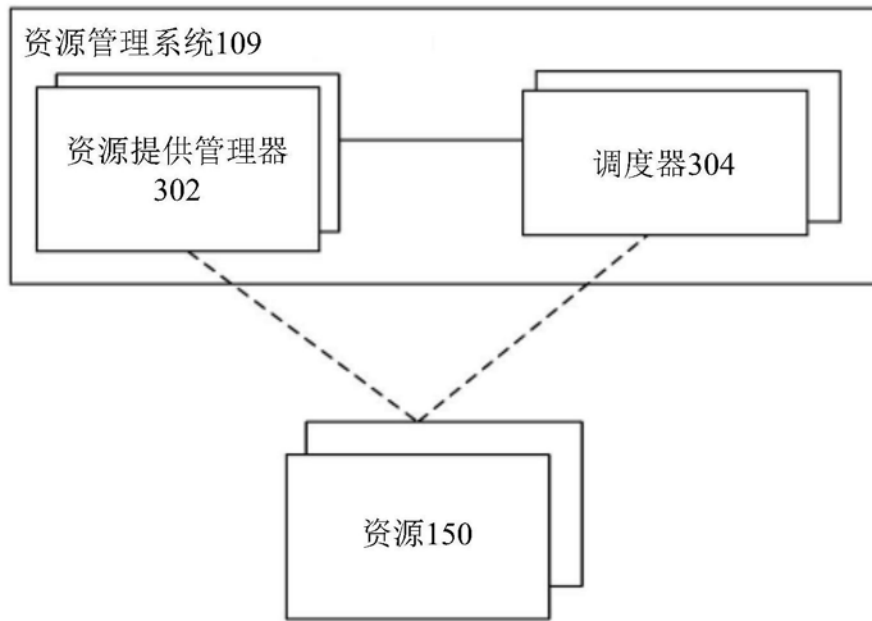


图3

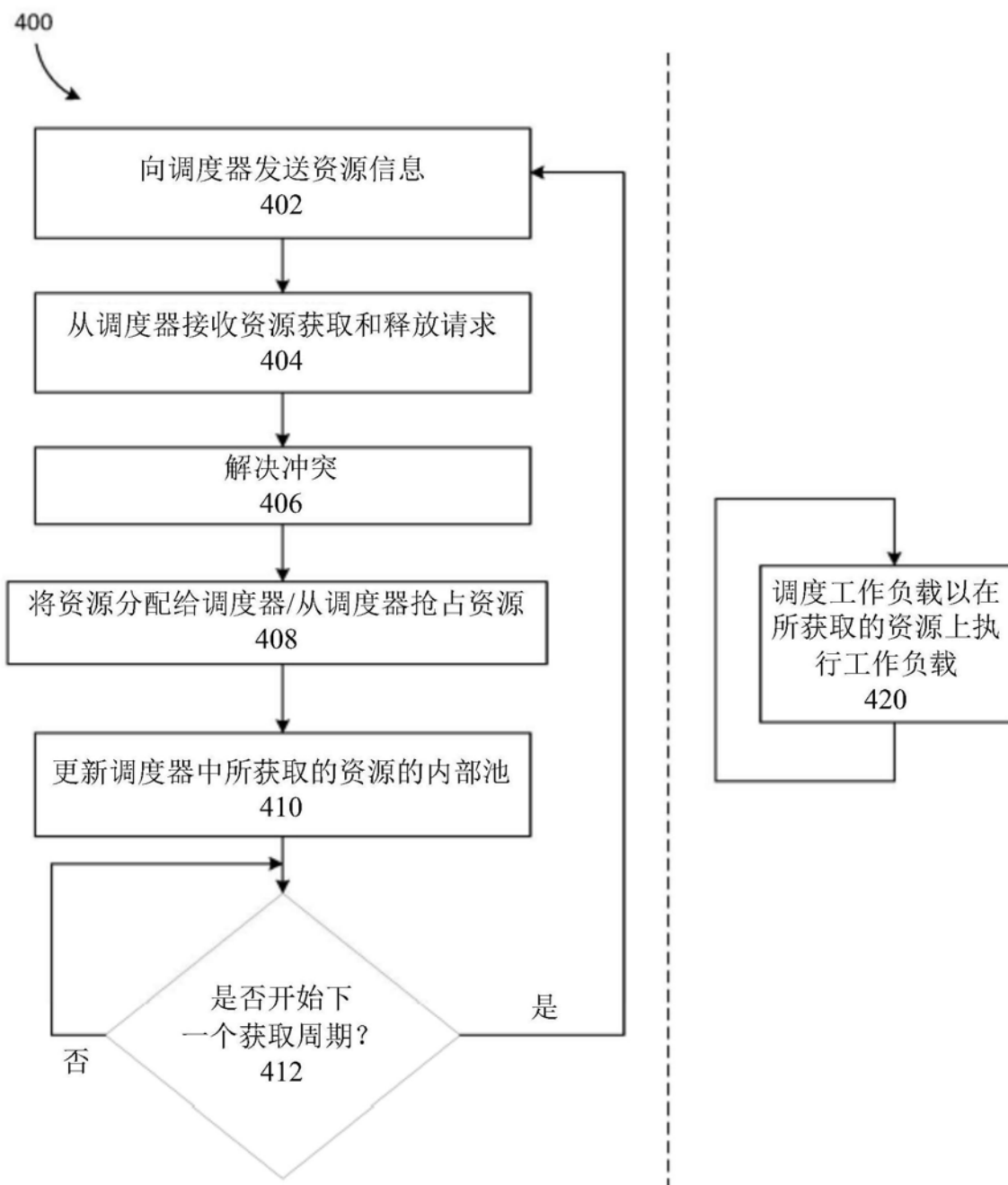


图4

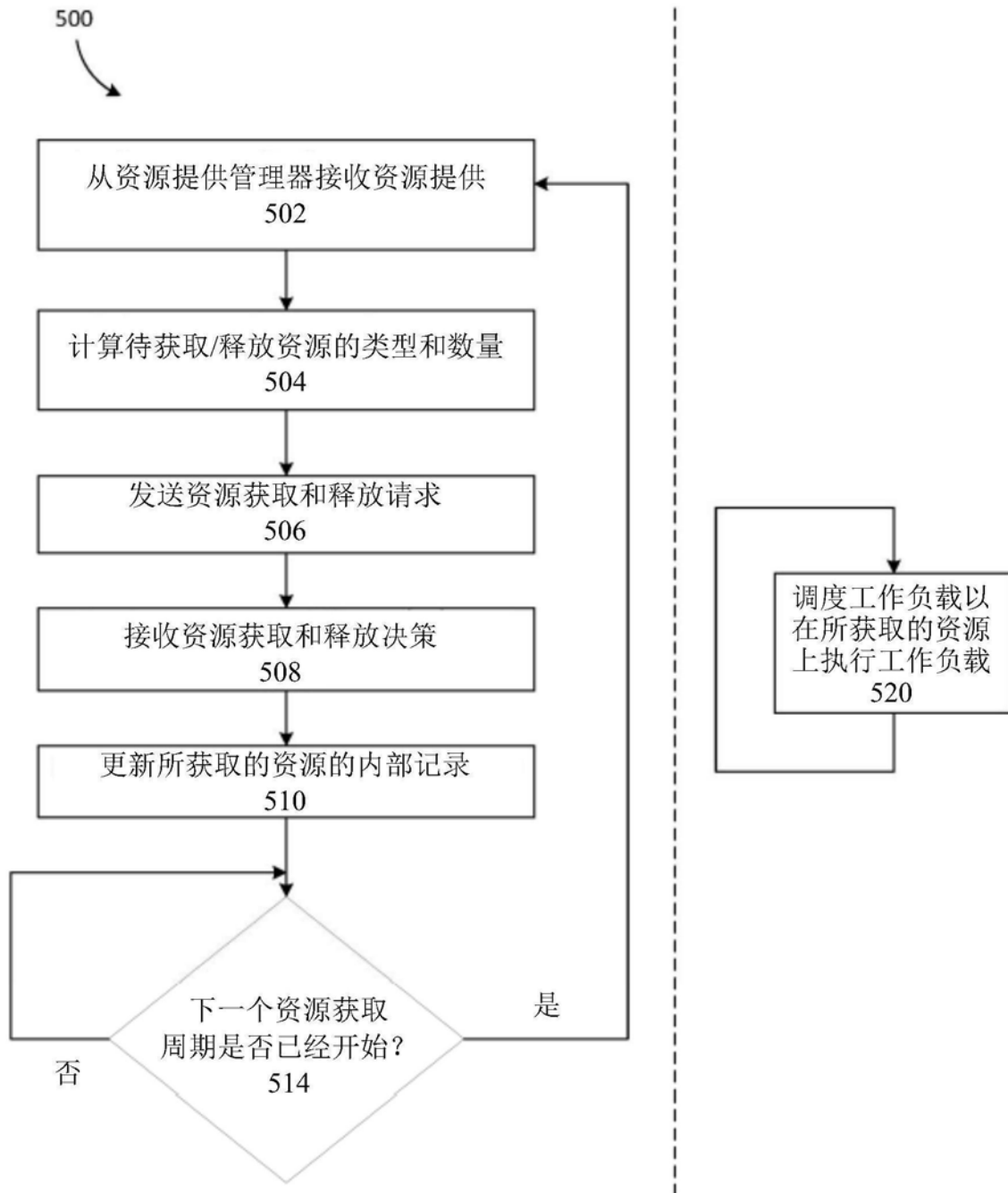


图5

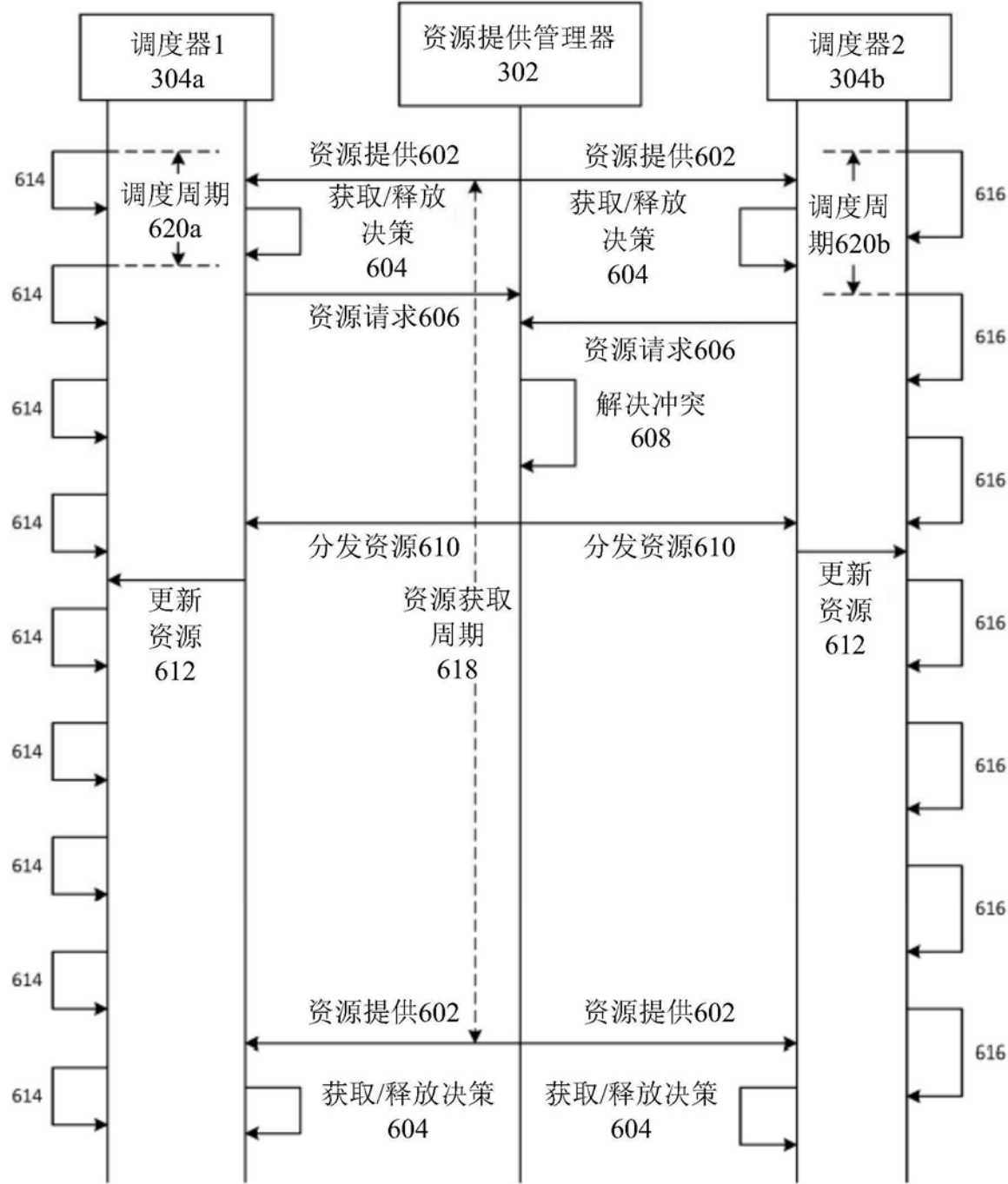


图6A

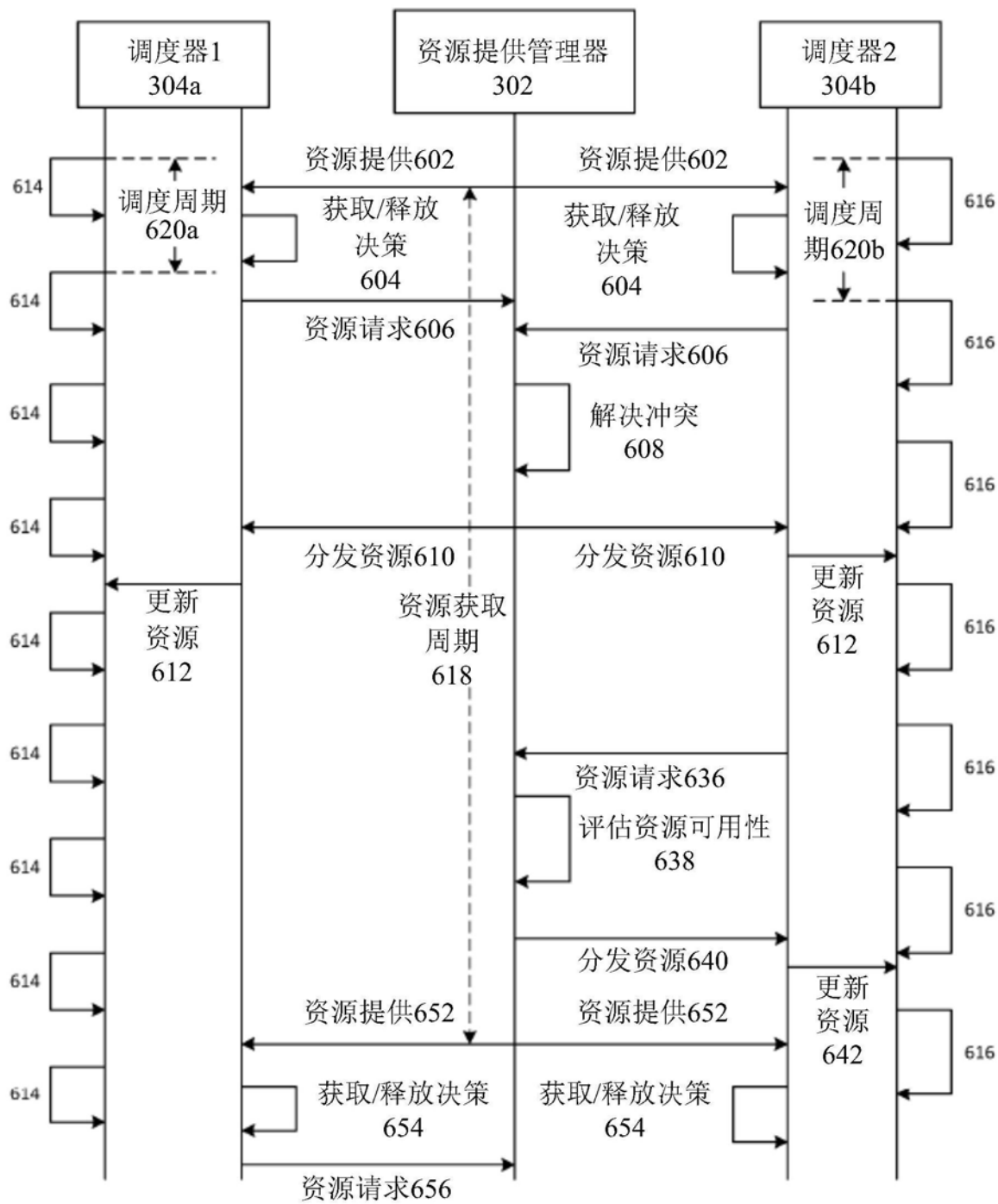


图6B