[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 9/46 (2006.01)

[12] 发明专利申请公开说明书

[21] 申请号 200610026430.6

[43] 公开日 2006 年 10 月 11 日

[11] 公开号 CN 1845075A

[22] 申请日 2006.5.11

[21] 申请号 200610026430.6

[71] 申请人 上海交通大学

地址 200240 上海市闵行区东川路 800 号

[72] 发明人 翁楚良

[74] 专利代理机构 上海交达专利事务所

代理人 王锡麟 王桂忠

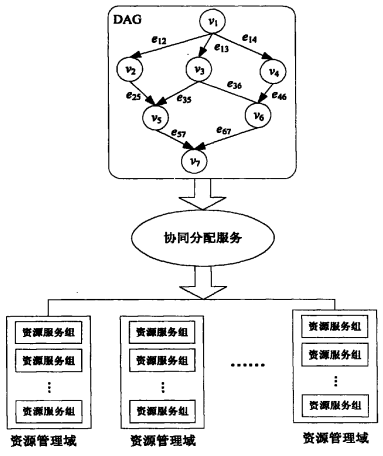
权利要求书 3 页 说明书 7 页 附图 2 页

[54] 发明名称

面向服务的网格高性能计算作业调度方法

[57] 摘要

一种面向服务的网格高性能计算作业调度方法，采用面向服务的方式，网格系统由分布的资源服务组构成，根据高性能计算应用的特征、网格计算中不同计算站点之间协作处理作业的特点，将一个高性能计算应用分解为一组存在数据输入输出关系的计算子任务。 采用有向无循环图表示一个高性能计算应用中各个子任务之间的数据相关性，采用改进型的动态优先级调度算法同时匹配就绪子任务和可用的资源服务组，将选定的子任务调度到选定的资源服务组中，考虑了一个子任务只能在特定的一些资源服务组上执行的情况，实现网格范围内的高性能计算作业的高效调度。



1、一种面向服务的网格高性能计算作业调度方法，其特征在于，采用面向服务的方式，网格系统由分布的资源服务组构成，根据高性能计算应用的特征、网格计算中不同计算站点之间协作处理作业的特点，将一个高性能计算应用分解为一组存在数据输入输出关系的计算子任务，用有向无循环图表示一个高性能计算应用中各个子任务之间的数据相关性，并采用动态优先级调度算法同时匹配就绪子任务和可用的资源服务组，将选定的子任务调度到选定的资源服务组中，实现网格范围内的高性能计算作业的高效调度。

2、根据权利要求 1 所述的面向服务的网格高性能计算作业调度方法，其特征是，包括如下步骤：

（1）确定作业有向无循环图 DAG 中的就绪任务$\{v_i\}$

用有向无循环图表示一个高性能计算应用的多个子任务，一个有向无循环图$G=(V,E)$，其中顶点集合$V=\{v_1,v_1,...,v_m\}$表示高性能计算应用被分解的多个子任务，$e_{ij}=(v_i,v_j)\in E$表示从子任务v_i到子任务v_j的通信，而$|e_{ij}|$则表示通信量，如果有向无循环图中子任务v_i所有前绪子任务均以执行，则该子任务为就绪任务；

（2）获取当前可用资源服务组的实时信息$\{R_j\}$

网格环境定义为由一系列的资源服务组构成，资源服务组集合形式化地表示为$R=\{R_1,R_2,...,R_n\}$，开销函数$C:V\times R\rightarrow \mathbf{R}$，表示一个计算子任务在资源服务组上执行的时间开销，子任务v_i在资源服务组R_j上执行的时间开销定义为$C(v_i,R_j)$，通过查询网格系统中的信息服务，获得相应的资源服务组的实时信息$\{R_j\}$；

（3）计算$MDL(v_i,R_{j|j\in A(v_i)},\sum(t))$

采用动态优先级调度算法，动态优先级表示为$MDL(v_i,R_j,\sum(t))$，其反映了在状态集$\sum(t)$情况下，子任务v_i调度到资源服务组R_j上的匹配程度，状态集$\sum(t)$表

示时刻t时所有资源服务组和通信资源的状态信息，动态优先级MDL定义为：

$$MDL(v_i, R_{j|j\in A(v_i)}, \sum(t)) = SL(v_i) - \max(t_d(v_i, R_j), t_a(R_j)) + \Delta(v_i, R_j)$$

其中，t是当前时间，式中的第一部分表示子任务v_i在任务图的静态优先级，其值为从子任务v_i到任务图终点不同路径上执行时间总和的最大值，$t_d(v_i, R_j)$表示在状态集合$\sum(t)$情况下，子任务v_i所需所有数据的最早可用时间，$t_a(R_j)$表示资源服务组最早空闲时间，

（4）确定使得MDL值最大的资源服务组R_J和就绪任务v_I；

（5）调度就绪任务v_I到资源服务组R_J；

（6）等待下一次调度事件，并重复上述过程。

3、根据权利要求2所述的面向服务的网格高性能计算作业调度方法，其特征是，所述的步骤（1）中，对于一个子任务，并不是在网格环境中的所有资源管理域中均能运行，一些子任务需要相应的科学工程计算库，而这个科学工程计算库仅存在于相应的计算结点上。

4、根据权利要求2所述的面向服务的网格高性能计算作业调度方法，其特征是，所述的步骤（2）中，对于不能执行某子任务的资源服务组，定义其时间开销为无穷大。

5、根据权利要求1或者2或者4所述的面向服务的网格高性能计算作业调度方法，其特征是，所述的资源服务组，是指：若将一个高性能计算应用划分为多个子任务，分别通过相应的计算程序求解，这些子任务属于计算密集型任务，因此一个计算资源上只会运行一个程序拷贝，这一计算程序和其运行的计算资源即构成了资源服务组。

6、根据权利要求2所述的面向服务的网格高性能计算作业调度方法，其特征是，所述的步骤（3）中，$\Delta(v_i, R_j)$是表示资源服务组处理能力的差异，定义为：

$$\Delta(v_i, R_j) = \bar{C}(v_i) - C(v_i, R_j)$$

其中，$\bar{C}(v_i)$表示子任务v_i在所有资源服务组上执行时间的平均值，如果计算得到的平均值为无限大，则取在所有资源服务组上执行时间为有限时间值的

最大值，$A(v_i)$表示可执行子任务v_i的所有可用资源服务组的集合。

7、根据权利要求6所述的面向服务的网格高性能计算作业调度方法，其特征是，在计算$\Delta(v_i,R_j)$时，$\bar{C}(v_i)$定义为：

$$\bar{C}(v_i)=\sum_{j\in A(v_i)} C(v_i,R_j)/\|A(v_i)\|$$

其中，$\|A(v_i)\|$表示集合$A(v_i)$中元素的个数。

面向服务的网格高性能计算作业调度方法

技术领域

本发明涉及的是一种计算机技术领域的方法，具体而言是一种面向服务的网格高性能计算作业调度方法。

背景技术

随着网络技术和计算机技术的发展，以及对高性能计算不断增长的需求，在集群计算的基础之上，出现了早期的元计算和近年来的计算网格。由于网格计算在高性能计算上的成功应用，促成了网格计算在信息服务、数据处理等多方面得到进一步的应用和深化。同时，Web 服务技术在业界的推动下已得到长足的发展。基于网格技术和 Web 服务，由 IBM 和 Globus 联盟提出了 WSRF(WS-Resource Framework)，以整合 Web 服务和网格技术的优势，实现在跨区域和机构上的资源共享，充分利用各类计算资源。无论基于 Web 服务和网格技术的 WSRF，还是其它面向服务的架构(SOA)，都需要针对在动态、开放的计算环境下确立如何构造、部署和使用面向服务应用的有效方法和机制。在传统计算网格的基础之上，高性能计算应用也将会得益于面向服务的架构。与传统的架构类似，在面向服务的架构下，同样需要解决资源的组织管理、任务的分解和调度等问题。

经对现有技术文献的检索发现，以余海燕为主提出了一种面向服务的网格作业管理方法（余海燕，查礼，李伟. 一种面向服务的网格作业管理机制. 计算机研究与发展，2003，40(12):1770-1774)。该方法基于一种面向服务的作业管理机制,它作为用户访问网格资源(服务) 的代理,为用户提供透明的、与资源物理位置无关的并带有会话支持的作业服务接口。 并引入了服务水平协议(SLA)的概念来表示用户需求的不同网格服务级别,作业管理系统则根据可定制的服务水平实现配置(SLAP) 将 SLA 中规定的各项 QoS 特性映射到具体的作业管理行为。该作业管理机制已应用于织女星网格系统软件中,并能够为基于服务网格的应用提供灵活有效的支持。该方法主要针对网格环境下的事务处理作业，没

有考虑高性能计算应用的特点，因此不适用于网格高性能计算应用。

发明内容

本发明针对现有技术的不足，结合高性能计算应用和网格计算特点，提出一种面向服务的网格高性能计算作业调度方法，使其实现高性能计算作业在网格环境下的有效调度，提高资源的利用率，降低作业的执行时间。

本发明是通过以下技术方案实现的，本发明采用面向服务的方式，网格系统由分布的资源服务组构成，根据高性能计算应用的特征、网格计算中不同计算站点之间协作处理作业的特点，将一个高性能计算应用分解为一组存在数据输入输出关系的计算子任务。采用有向无循环图表示一个高性能计算应用中各个子任务之间的数据相关性，采用改进型的动态优先级调度算法同时匹配就绪子任务和可用的资源服务组，将选定的子任务调度到选定的资源服务组中，考虑了一个子任务只能在特定的一些资源服务组上执行的情况，实现网格范围内的高性能计算作业的高效调度。

本发明方法包括如下步骤：

（1）确定作业有向无循环图（DAG）中的就绪任务$\{v_i\}$

通常一个高性能计算应用任务可以分解成多个子任务，针对广域网的特点，计算网格通常适用于子任务之间通信量较小的高性能计算应用。在这些子任务之间存在一定的数据相关性，其中输入输出相关情况较多，因此，一个高性能计算应用的多个子任务可以用有向无循环图(DAG)加以表示。

一个有向无循环图$G=(V,E)$，其中顶点集合$V=\{v_1,v_1,...,v_m\}$表示高性能计算应用被分解的多个子任务，$e_{ij}=(v_i,v_j)\in E$表示从子任务v_i到子任务v_j的通信，而$|e_{ij}|$则表示通信量，通信可以分为两大类：一类是大数据的文件数据通信，另一类是小数据的参数数据通信。需要注意的是，对于一个子任务，并不是在网格环境中的所有资源管理域中均可以运行，如一些子任务需要特定的科学工程计算库，而这个科学工程计算库仅存在于一些特定的计算结点上。

如果有向无循环图中子任务v_i所有前绪子任务均以执行，则该子任务为就绪任务。

（2）获取当前可用资源服务组的实时信息$\{R_j\}$

若将一个高性能计算应用划分为多个子任务，分别通过相应的计算程序求解，通常这些子任务是属于计算密集型任务，因此一个特定计算资源上通常只会运行一个程序拷贝。这一计算程序和特定运行的计算资源即构成了资源服务组。

网格环境可以定义为由一系列的资源服务组构成，其中每个资源服务组由相应的计算软件和特定的硬件计算资源组成，通过服务封装机制可以很便利地被外界访问。资源服务组集合形式化地表示为$R=\{R_1,R_2,...,R_n\}$。开销函数$C:V\times R\to \mathbf{R}$，表示一个计算子任务在资源服务组上执行的时间开销，子任务v_i在资源服务组R_j上执行的时间开销定义为$C(v_i,R_j)$。对于不能执行某子任务的资源服务组，定义其时间开销为无穷大。

通过查询网格系统中的信息服务，可获得相应的资源服务组的实时信息$\{R_j\}$。

基于（1）和步聚（2）的设定，网格高性能计算应用调度问题可以表示为图的映射问题，即将一个有向无循环图映射到资源服务组集合上，以最小化总的执行时间为目标。

（3）计算$MDL(v_i,R_{j|j\in A(v_i)},\sum(t))$

作业调度方法采用一种改进的动态优先级调度算法(DLS)，DLS 算法是通过任务优先级进行调度任务，DLS 算法的特点在于一个子任务的优先级与任务图中已调度的子任务相关。

改进的动态优先级(Dynamic Level)表示为$MDL(v_i,R_j,\sum(t))$，其反映了在状态集$\sum(t)$情况下，子任务v_i调度到资源服务组R_j上的匹配程度，状态集$\sum(t)$表示时刻t时所有资源服务组和通信资源等的状态信息。形式地，动态优先级MDL可以定义为：

$$MDL(v_i,R_{j|j\in A(v_i)},\sum(t))=SL(v_i)-\max(t_d(v_i,R_j),t_a(R_j))+\Delta(v_i,R_j)$$

其中，t是当前时间。式中的第一部分表示子任务v_i在任务图的静态优先级，

其值为从子任务 v_i 到任务图终点不同路径上执行时间总和的最大值，对于一个高性能计算应用的任务图，这是一个静态信息。$t_a(v_i,R_j)$ 表示在状态集合 $\sum(t)$ 情况下，子任务 v_i 所需所有数据的最早可用时间。$t_a(R_j)$ 表示资源服务组最早空闲时间。$\Delta(v_i,R_j)$ 则是表示资源服务组处理能力的差异，定义为：

$$\Delta(v_i,R_j)=\bar{C}(v_i)-C(v_i,R_j)$$

其中，$\bar{C}(v_i)$ 表示子任务 v_i 在所有资源服务组上执行时间的平均值。如果计算得到的平均值为无限大，则取在所有资源服务组上执行时间为有限时间值的最大值。$A(v_i)$ 表示可以执行子任务 v_i 的所有可用资源服务组的集合。在计算 $\Delta(v_i,R_j)$ 时，$\bar{C}(v_i)$ 定义为：

$$\bar{C}(v_i)=\sum_{j\in A(v_i)} C(v_i,R_j)/\|A(v_i)\|$$

其中，$\|A(v_i)\|$ 表示集合 $A(v_i)$ 中元素的个数。

（4）确定使得 *MDL* 值最大的资源服务组 R_j 和就绪任务 v_j；

（5）调度就绪任务 v_j 到资源服务组 R_j；

（6）等待下一次调度事件，并重复上述过程。

其中，资源是指各类计算资源，可以是物理上的 CPU 资源、存储资源，也可以是逻辑上的数据库、工程计算库等；Web 服务是一种接口，它描述了在网络上可通过相关协议进行访问的操作集合；而资源服务组是指在特定资源上的操作集合；任务是指代高性能计算应用，而子任务指代高性能计算应用分解后的计算模块。

在本发明提出的作业调度方法中，以最大化动态优先级 *MDL* 为目标，匹配就绪子任务和可用的资源服务组，并将该子任务调度到选定的资源服务组中，然后更新系统的状态信息，重新计算 *MDL*，确定下一个子任务的调度。其优点在于子任务和资源服务组是在同时选择的，优于单独选取子任务或者资源服务组。

本发明提出的作业调度方法，是针对高性能计算应用，采用面向服务的方式，

通过网格基础设施，为在广域范围内实现协同求解高性能计算问题提供了一种高效、灵活的方法，同时考虑了完成一个计算作业需要专用计算资源的情况，不同于已有技术中一个计算作业可以在所有计算资源上执行的情况。因此，提出的作业调度方法能够很好地应用于网格环境中高性能计算的实际情况，可以获得更好的性能。

附图说明

图 1 为面向服务的层次化资源管理体系结构图

图 2 为高性能计算应用分解的多个子任务构成的有向无循环图示意图

图 3 为本发明任务调度流程图

具体实施方式

为实现高效组织分布、异构、自治的计算资源，网格系统采用以下组织方式：网格门户、全局资源管理层和局部资源管理层，如图 1 所示。

1）网格门户

在组织结构的上层是网格门户，也是终端用户与网格高性能计算环境交互的组件。网格门户为终端用户提供工具以便于在网格环境下协同调度大规模高性能应用，通过简洁的图形界面或 Web 界面，终端用户可以很便利地获取服务信息，例如计算处理能力、科学与工程库、系统负载等，同时还可便利地调用计算服务、配置参数、监控中间结果和下载最终计算结果等。从理论上，终端用户无须关注网格协议和网格基础设施；另一方面，高性能应用领域的开发人员亦无须关注网格环境下应用软件的配置。终端用户通过网格平台可以很便利地利用多个高性能计算应用软件完成大规模的科学计算任务。

2）全局资源管理层

在全局资源管理层，需要实现服务请求的解析、计算任务的分解、计算子任务的协同调度以及信息服务等。

代理服务(broker service)是网格中间件的一个重要组成部份，由它解析由网格门户传递来的用户请求，最终根据不同应用问题，将一个大规模的高性能计算任务划分为多个子任务。例如，飞机整机模拟任务可以分解为多个子任务，即多个功能模块，各个功能模块分别计算飞机的机身、两个机翼、水平尾翼和垂直尾翼。任务解析与分解是和相应的计算任务类型密切相关，不同的高

性能计算应用需要不同的分解策略。

通常被分解的多个子任务之间存在一定的数据相关性，设定这些子任务间存在输入输出关联性。通过协同分配服务可以将这些子任务分别由地理上分布的、跨管理域的多个应用软件执行。信息服务则提供当前整个系统的状态信息。

3）局部资源管理层

为了管理本地资源，需要两类 Web 服务：一类是作业工厂服务，另一类是作业管理服务。作业工厂服务为高性能计算子任务请求创建服务实例，并确立相应的计算资源。同时向外界提供本地资源管理域的资源信息，包括硬件资源信息和计算应用软件服务。作业管理服务则是管理已被创建的服务实例，并结合确立的计算资源，形成对应特定请求的资源服务组，并发布已被提交给后台调度系统的作业的状态信息。

这两类特定的服务由专用的计算资源实现，而为高性能计算子任务请求创建的服务则是动态依附于当前可用资源，即是由局部资源管理层根据请求类型和当前可用资源信息进行动态指定。

具体调度过程如下：

网格用户通过网格门户提交一个高性能计算作业，然后由全局资源管理层中的代理服务根据计算应用的特征进行分解，形成相应任务的有向无循环任务图。如图 2 所示，这个计算作业由 7 个子任务 v_i 组成，$i=1,2,...,7$，各个子任务之间的边表示它们之间的通信，也反映了它们执行所需的先后次序。

作业的调度框架如图 3 所示，全局资源管理层中的协同分配服务向信息服务查询当前网格中资源服务组的实时信息，即 $R=\{R_1,R_2,...,R_n\}$ 的信息。

协同分配服务确定有向无循环任务图中就绪子任务，即在有向无循环图中前驱结点已经执行的结点所对应的子任务。针对每个子任务，分别计算 $MDL(v_i,R_{j|j\in A(v_i)},\sum(t))$，并选定调度就绪子任务 v_i 到资源服务组 R_j，即在这些子任务和资源服务组中，资源服务组 R_j 和就绪子任务 v_i 使得 MDL 值最大。协同分配服务根据调度事件重复上述调度过程，直至所有子任务均被调度到相应的资源服务组上。

资源服务组 R_j 所对应的局部资源管理层中作业管理服务获得执行相应子任务 v_i 的请求之后，通过作业工厂服务创建相应的服务实例，确定相应的硬件资源和软件，启动相应的计算服务。在任务执行期间，作业管理服务同时还响应来自用户的查询请求，提供子任务 v_i 的执行中间状态。等待相应的子任务 v_i 执行结束之后，作业管理服务将计算结果返回给协同分配服务。

由协同分配服务整合这个高性能计算应用的多个子任务（子任务 v_i，$i=1,2,...,7$）的计算结果，并返回至网格门户，最后通过网格门户将计算结果返回给用户。

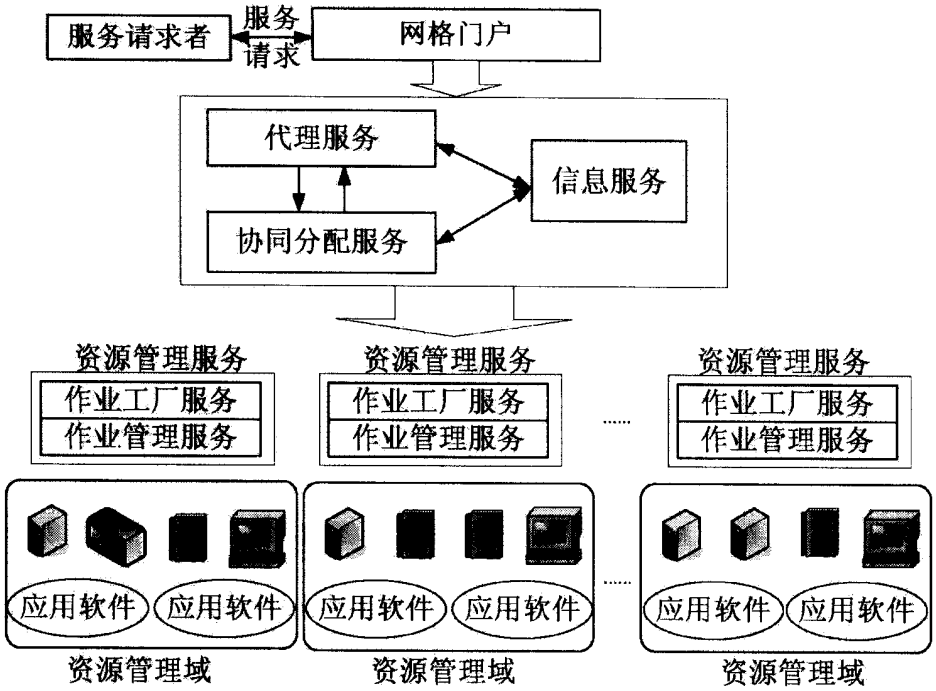

服务请求者
服务请求
网格门户
代理服务
信息服务
协同分配服务
资源管理服务
作业工厂服务
作业管理服务
资源管理服务
作业工厂服务
作业管理服务
资源管理服务
作业工厂服务
作业管理服务
应用软件
应用软件
应用软件
应用软件
应用软件
应用软件
资源管理域
资源管理域
资源管理域

图 1

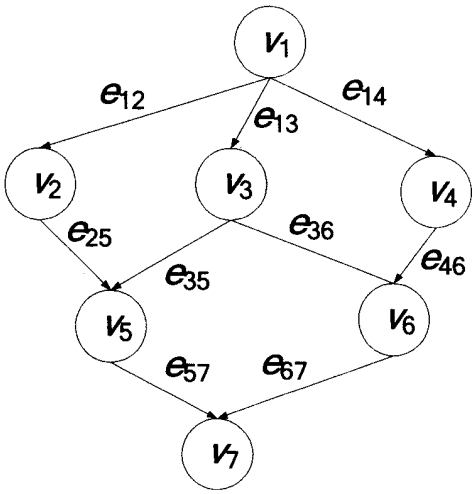

V1
e12
e13
e14
V2
V3
V4
e25
e36
e46
e35
V5
V6
e57
e67
V7

图 2

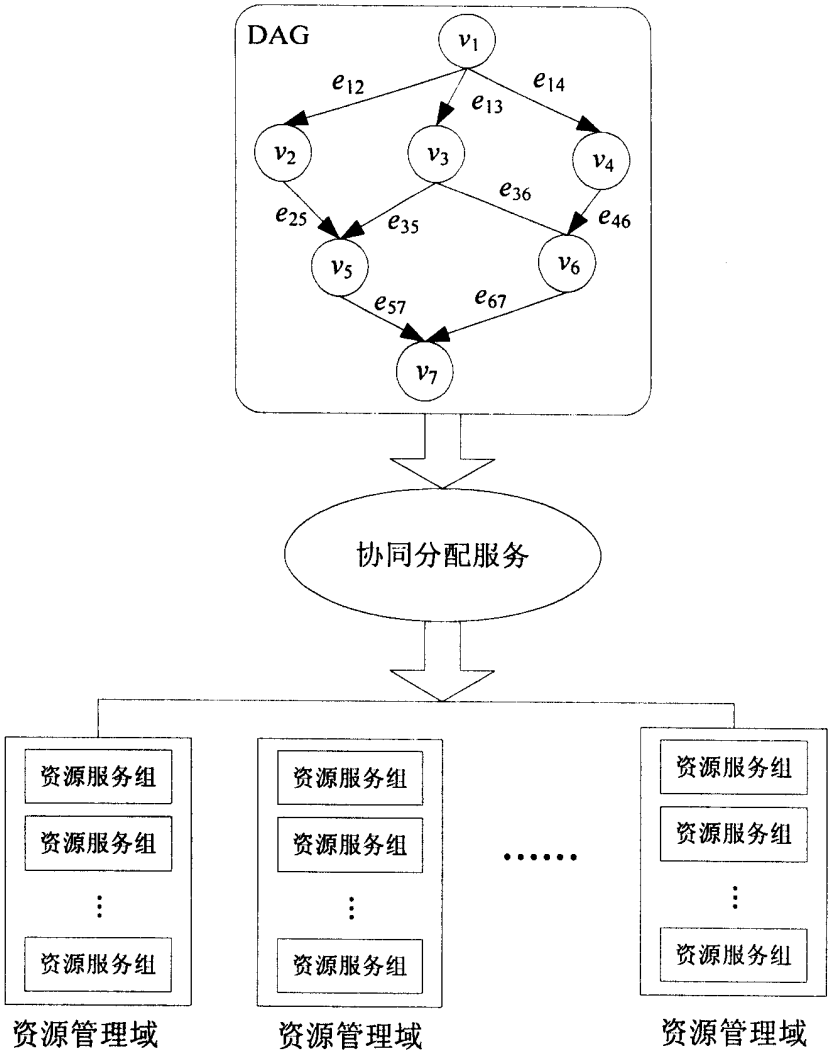

DAG
v_1
e_{12}
e_{13}
e_{14}
v_2
v_3
v_4
e_{36}
e_{25}
e_{35}
e_{46}
v_5
v_6
e_{57}
e_{67}
v_7
协同分配服务
资源服务组
资源服务组
资源服务组
资源服务组
资源服务组
资源服务组
资源服务组
资源服务组
资源服务组
资源管理域
资源管理域
资源管理域

图 3