(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 102882973 A
(43)申请公布日 2013.01.16

(21)申请号 201210384185.1

(22)申请日 2012.10.11

(71)申请人 北京邮电大学
地址 100876 北京市海淀区西土城路 10 号

(72)发明人 赵耀 宋颖莹 彭书凯 邹志勇
杨放春 邹华 孙其博 林荣恒
李静林 刘志晗

(74)专利代理机构 北京德琦知识产权代理有限公司 11018
代理人 夏宪富

(51)Int.Cl.
H04L 29/08(2006.01)

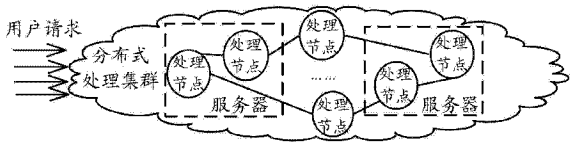
权利要求书 5 页 说明书 10 页 附图 4 页

(54)发明名称

基于 P2P 技术的分布式负载均衡系统和方法

(57)摘要

一种基于 P2P 技术的分布式负载均衡系统和方法，系统由均衡负载功能的分布式处理集群所组成，该集群内只设有分设于不同服务器的、由处理层和通信层构成的多个同构或异构处理节点，每个处理节点都能独立承担分布式仲裁的负载均衡功能，且其权限与功能都相同，差异只是事件的处理速率不同；当客户端向分布式处理集群实时发送海量数据处理请求时，某个处理节点完成请求处理后，将产生的中间结果封装为事件，发往其它处理节点继续处理；各处理节点在分发事件过程中，始终采用分布式负载均衡方法保证事件分发的负载均衡，直到产生最终结果并返还给客户端。本发明有效避免系统出现单点失效的负载均衡，能够在分布式集群系统中实现节点动态变化的负载均衡。



CN 102882973 A

1. 一种基于点对点 P2P（Peer-to-Peer）技术的分布式负载均衡系统，其特征在于：该系统由具有均衡负载功能的分布式处理集群所组成，该处理集群内没有集中控制的中心节点，只设有分设于不同服务器、并分别用作该系统分布式控制核心的多个同构或异构的处理节点，每个处理节点都能独立提供分布式仲裁的负载均衡功能，且各自的处理事件的权限与均衡负载的功能都相同，异构处理节点的差异只是其事件的处理速率不同；当客户端向分布式处理集群实时发送海量数据处理请求时，该系统的某个处理节点完成请求处理后，将产生的中间结果封装为事件，发往其它处理节点继续处理；各处理节点在分发事件过程中，始终采用分布式负载均衡方法保证事件分发的负载均衡，直到产生最终结果并返还给客户端；该处理节点设有处理层和通信层；其中：

处理层，由包含多个处理单元的处理单元容器所组成，负责进行事件处理：接收来自通信层的事件，再根据事件类型和每个事件所设定属性的属性值，将该事件交由特定处理单元进行处理，同时负责将该事件处理完成后产生的新事件或最终结果，交给通信层进行分发与发送；

通信层，用于监听、分发事件及实现分布式负载均衡：负责从网络接收事件并转交给处理层；然后接收处理层完成处理后产生的新事件或最终结果，再使用分布式负载均衡方法将该新事件或最终结果分发给其他处理节点或客户端，保证该分布式处理集群中的各处理节点负载均衡；设有：事件监听、事件分发、事件发送、负载管理和分布式集群管理共五个模块。

2. 根据权利要求 1 所述的系统，其特征在于：所述处理层的各个组成部件功能如下：

处理单元容器，用于管理和控制其内部的各个处理单元，采用设定顺序激活相应的处理单元来处理不同事件；

处理单元，用于处理特定类型和属性的事件，负责藉由通信层的事件分发模块和事件发送模块，将产生的中间处理结果作为新事件，发往其他处理单元继续处理；或者将处理的最终结果返回给客户端；每个处理单元都是系统预先设置的，或者由第三方程序员开发并部署到处理节点上。

3. 根据权利要求 1 所述的系统，其特征在于：所述通信层的各个组成部件功能如下：

事件监听模块，负责接收网络事件，并交由处理层的处理单元进行处理；

事件分发模块，负责接收处理层的新事件或最终结果，并按照负载管理模块提供的负载均衡信息来分发新事件或最终结果，再经由事件发送模块将其发送到其他处理节点或客户端；

事件发送模块，用于将来自事件分发模块的新事件或最终结果分别发送到设定的处理节点或客户端；

负载管理模块，负责创建与维护事件分发模块所需的负载信息，即虚拟节点映射表与动态负载信息表：在系统启动时，借助分布式集群管理模块中的信息创建虚拟节点映射表；在系统启动后，利用已经建立的虚拟节点映射表信息创建动态负载信息表，且在分布式处理集群中有新增或退出处理节点时，对虚拟节点映射表和动态负载信息表进行实时更新；

分布式集群管理模块，负责管理和维护分布式处理集群及其中的全部处理节点与其对应的虚拟节点，该分布式处理集群中的各处理节点也经由该分布式集群管理模块实时获取该集群中自身与其他处理节点的链接地址、当前存活状态、负载状态与相关信息。

4. 根据权利要求 3 所述的系统，其特征在于：所述虚拟节点是为了实现分布式负载均衡方法而设置的一个与处理节点相互映射的逻辑符号，每个虚拟节点只能映射到一个处理节点，但每个处理节点能够映射到多个虚拟节点；虚拟节点个数是在系统启动时设置的，其数量远大于处理节点个数；且在系统运行后，虚拟节点个数保持不变；当系统新增或退出处理节点时，处理节点及其虚拟节点之间的映射关系也根据调整算法作相应改变；

所述虚拟节点映射表设于每个处理节点的负载管理模块，用于表示每个虚拟节点编号及其对应的处理节点编号，以便在系统分发事件时，查找该虚拟节点映射表，根据虚拟节点编号获取其对应的处理节点；该虚拟节点映射表是根据每个处理节点的事件处理速率高低降序排列形成的；因每个处理节点处理事件的速率不同，故每个处理节点所分配的对应虚拟节点个数也不相等；且在系统运行后，为节省更新时间，采取增删处理节点的动态负载信息表和虚拟节点映射表的更新方法来实时更新虚拟节点映射表；

所述动态负载信息表设于每个处理节点负载管理模块，用于表示每个处理节点编号、按照其事件处理速率在理论上分配的虚拟节点个数与其实际对应的虚拟节点个数；该动态负载信息表是按照每个处理节点的负载数量与其理论上分配的负载数量之差的绝对值降序排列形成的，且其信息与虚拟节点映射表的信息逐一对应；故在虚拟节点映射表更新时，动态负载信息表也要相应更新。

5. 一种基于 P2P 技术的分布式负载均衡系统的工作方法，其特征在于：每个处理节点的通信层中的事件分发模块和负载管理模块协同完成基于虚拟节点与处理节点之间映射的事件分发方法，实现基于 P2P 技术的分布式负载均衡；所述方法包括下述两个操作步骤：

步骤 1，每个处理节点的通信层中的负载管理模块创建或更新虚拟节点映射表和动态负载信息表：

步骤 2，每个处理节点的通信层中的事件分发模块与负载管理模块交互，根据虚拟节点映射表执行事件分发决策操作，将每个事件分发到相应的处理节点，并实现负载均衡：

事件分发模块计算每个事件的关键属性的哈希值，再将该哈希值除以虚拟节点总个数所得到的余数，作为将该事件准备分发到的虚拟节点编号；

事件分发模块与负载管理模块交互，从虚拟节点映射表查找该虚拟节点编号及其所对应的处理节点编号，以获取准备将该事件分发到的处理节点编号。

6. 根据权利要求 5 所述的方法，其特征在于：所述方法的应用场景有三种：

分发事件：客户端向分布式负载均衡系统中的分布式处理集群发送海量数据处理请求，分布式处理集群中的各个处理节点分别独立、均等地完成各自的负载决策，且在处理过程中，保证各个处理节点的负载均衡；并实时向客户端返回处理结果；

新增处理节点：客户端向分布式处理集群发送海量数据处理请求过程中，为减轻该处理集群中各处理节点的负载压力，在该处理集群中实时增加新的处理节点；并在尽可能短时间内，由新增处理节点分担原处理集群中各处理节点的部分负载，使得新增处理节点和原处理集群中各处理节点重新实现负载均衡；

移出处理节点：客户端向该分布式处理集群发送海量数据处理请求过程中，因故障或管理需要，有节点要迁移离开该处理集群时，由该处理集群中的其它处理节点承担该移出处理节点的负载；并在尽可能短时间内，使得移出一个或多个处理节点后的该处理集群中的各个处理节点重新达到负载均衡。

7. 根据权利要求 5 所述的方法，其特征在于：所述步骤 1 包括下列操作内容：

（11）系统初始建立时，负载管理模块根据每个处理节点的事件处理速率高低降序排列方式创建初始化虚拟节点映射表，此时对于事件处理速率不同的处理节点，分别设置相应不同的虚拟节点个数；

（12）系统运行后，当新增或移出处理节点时，为使系统尽快重新达到负载均衡，负载管理模块实时更新虚拟节点映射表和动态负载信息表。

8. 根据权利要求 7 所述的方法，其特征在于：所述步骤（11）包括下列操作内容：

（11A）为保证负载均衡，负载管理模块先计算系统当前每个处理节点理论上获取的负载个数，即其理论负载个数 loadAfter(i) 为：$loadAfter(i)=VIRTUAL\_NODES\_NUM\times\frac{P\_CAPACITY(i)}{\sum_{i=0}^{i=N-1}P\_CAPACITY(i)}$；式中，VIRTUAL_NODES_NUM 是虚拟节点总个数，其数值远大于系统处理节点总个数 N；自然数 i 是处理节点编号，其最大值为 N；P_CAPACITY(i) 是编号为 i 的处理节点的事件处理速率，该事件处理速率取决于包括 CPU 和内存的权衡因子，或者由专业测试工具事先测定的；$\sum_{i=0}^{i=N-1}P\_CAPACITY(i)$ 是系统当前所有处理节点的事件处理总速率；

（11B）负载管理模块顺序给每个处理节点随机分配 loadAfter(i) 个负载，直到所有虚拟节点都有一个与其对应的处理节点；且每次分配都要分别在动态负载信息表和虚拟节点映射表执行相应更新操作。

（11C）负载管理模块对动态负载信息表中各个处理节点按照其当前的实际负载数与理论负载数之差的绝对值进行降序排列。

9. 根据权利要求 6 所述的方法，其特征在于：所述方法应用于分发事件场景时，包括下列操作内容：

（A1）处理节点的事件监听模块监测到网络中新出现需要处理的事件，就根据该事件类型将其发送到处理单元容器，交给相应的处理单元进行处理；

（A2）该处理单元完成事件处理后，将新产生的事件准备发往其他处理节点或由其自身继续处理；

（A3）事件分发模块先计算该新事件属性的哈希值，再利用哈希方法、即将该哈希值除以虚拟节点总个数所得到的余数，作为对应的虚拟节点编号；

（A4）事件分发模块与负载管理模块交互，根据该虚拟节点编号查询虚拟节点映射表，得到该新事件准备分发到的处理节点编号；然后，将该新事件与要分发的对应处理节点编号信息传给事件发送模块；

（A5）事件发送模块将该新事件发往其他处理节点或自身节点继续处理，直到产生事件最终结果，返回给客户端。

10. 根据权利要求 6 所述的方法，其特征在于：所述方法应用于新增处理节点场景时，包括下列操作内容：

（B1）处理节点的分布式集群管理模块检测到新增处理节点加入，就将该信息通知分布式集群中的所有处理节点的负载管理模块，并提供该新增处理节点的包括编号、IP 地址、事件处理速率和端口号的配置信息；

（B2）其他各个处理节点负载管理模块接收上述信息，再根据该新增处理节点信息对动

态负载信息表和虚拟节点映射表进行更新；

（B3）每个处理节点负载管理模块完成虚拟节点映射表和动态负载信息表的更新后，新增处理节点就分担该分布式处理集群中的所有原处理节点的部分负载，使得新增处理节点和该处理集群中的所有原处理节点很快重新达到负载均衡。

11. 根据权利要求 10 所述的方法，其特征在于：所述步骤（B2）中，动态负载信息表和虚拟节点映射表的更新包括下列操作内容：

（B2A）为保证负载均衡，负载管理模块先计算系统当前包括新增处理节点在内的所有处理节点理论上将获取的负载个数，即其理论负载个数 loadAfter(i) 为：$loadAfter(i) = VIRTUAL\_NODES\_NUM * \frac{P\_CAPACITY(i)}{\sum_{i=0}^{i=N} P\_CAPACITY(i)}$，式中，自然数 i 是处理节点编号，其最大值为 N；，VIRTUAL_NODES_NUM 是虚拟节点总个数，其数值远大于系统处理节点总个数 N；P_CAPACITY(i) 是编号为 i 的处理节点的事件处理速率，$\sum_{i=0}^{i=N} P\_CAPACITY(i)$ 是系统当前所有处理节点的总事件处理速率；

（B2B）负载管理模块将动态负载信息表按照处理节点的实际负载数与理论负载数之差的绝对值进行降序排列；

（B2C）负载管理模块按照调整后的动态负载信息表的排列顺序，从绝对值最大的处理节点 k 开始，将（|loadAfter(k)-loadActual(k)|）个负载转移给新增处理节点，式中，loadActual(k) 为处理节点的实际负载个数；直到新增处理节点 i 的理论负载个数与实际负载个数相等；需要注意的是：对于需要转移负载给新增处理节点的最后一个处理节点 1，其转移的负载个数应不多于（|loadAfter(1)-loadActual(1)|），以保证该新增处理节点所分配的实际负载个数不大于其理论分配的负载个数，且负载数多的处理节点移出更多的负载；

（B2D）根据上述步骤调整结果，对虚拟节点映射表和动态负载信息表做相应更新：遍历处理节点的动态负载信息表，对于该表中每个处理节点转移（|loadAfter(k)-loadActual(k)|）个负载给新增处理节点，直到新增处理节点 i 理论负载个数 loadAfter(i) 与实际负载个数相等 loadActual(i)；且每次转移都要分别在动态负载信息表和虚拟节点映射表执行相应更新操作；再在动态负载信息表上新建一行，用于表示该新增处理节点的负载信息；最后，对动态负载信息表中各个处理节点重新按照其实际负载数与理论负载数之差的绝对值进行降序排列。

12. 根据权利要求 6 所述的方法，其特征在于：所述方法应用于移出处理节点场景时，包括下列操作内容：

（C1）处理节点的分布式集群管理模块采用各个处理节点之间互发心跳信息或其他方法，检测到该集群中有移出处理节点的状况：处理节点正常关闭或因故障而宕机，使得该集群中的处理节点减少的现象；

（C2）发现移出处理节点的处理节点中的分布式集群管理模块，通知该分布式集群中的所有处理节点的负载管理模块有处理节点移出，并提供该移出处理节点包括其编号、IP 地址、事件处理速率和端口号的配置信息；

（C3）其他各个处理节点的负载管理模块接收上述信息后，根据该移出的处理节点信息更新动态负载信息表和虚拟节点映射表；

（C4）每个处理节点负载管理模块更新完成虚拟节点映射表和动态负载信息表后，该集群中的其他各个处理节点就承担了该退出处理节点的负载，并在尽可能短的时间内，移出处理节点后的该分布式处理集群重新实现负载均衡。

13. 根据权利要求 12 所述的方法，其特征在于：所述步骤（C3）中，动态负载信息表和虚拟节点映射表的更新包括下列操作内容：

（C31）假设原处理节点总个数为 N，虚拟节点总个数为 VIRTUAL_NODES_NUM，且 VIRTUAL_NODES_NUM 远大于 N，此时移出一个处理节点，即当前处理节点总个数为（N-1）；同时，计算当前系统各处理节点的理论负载为：$loadAfter(i) = VIRTUAL\_NODES\_NUM \times \frac{P\_CAPACITY(i)}{\sum_{i=0}^{i=N-2} P\_CAPACITY(i)}$；并对动态负载信息表中各个处理节点重新按照其实际负载数与理论负载数之差的绝对值降序排列；

（C32）为保证移出处理节点的负载转移后，仍然实现均衡负载，采取下述更新策略：按照调整后的动态负载信息表的排列顺序，从绝对值最大的处理节点 k 开始，逐个进行下述操作：从该移出处理节点处获取（|loadAfter(k)-loadActual(k)|）个负载，直到该移出处理节点的所有负载都已被重新分配给其他各个处理节点；需要注意的是，操作到动态负载信息表中最后一个处理节点时，如果该移出处理节点的负载还没有全部被分配出去，则将该移出处理节点剩余的所有负载都转移给最后一个处理节点，保证负载数少的处理节点能从该移出处理节点处获取更多负载；

（C33）根据上述操作结果，相应修改虚拟节点映射表和动态负载信息表：遍历处理节点的动态负载信息表，该表中的每个处理节点从该移出处理节点处获取（|loadAfter(k)-loadActual(k)|）个负载，其中，自然数 k 为遍历时的当前处理节点编号；直到该移出处理节点的所有负载都已被重新分配给其他各个处理节点；且每次转移负载都要分别在动态负载信息表和虚拟节点映射表执行相应修改操作；

（C34）在动态负载信息表上删除已经移出的处理节点信息；最后，对动态负载信息表中各个处理节点重新按照其实际负载数与理论负载数之差的绝对值进行降序排列。

基于 P2P 技术的分布式负载均衡系统和方法

技术领域

[0001] 本发明涉及一种基于点对点 P2P（Peer-to-Peer）技术的分布式负载均衡系统和方法，属于计算机网络的技术领域。

背景技术

[0002] 随着网络业务量、访问量和数据流量的快速增长，服务器集群技术得到广泛的应用和研究，而负载均衡技术作为服务器集群技术中的重要组成部分，也得到了广泛、深入的研究和应用。

[0003] 目前，负载均衡技术在商业化的 web 服务的业务负载管理、网格计算中的计算任务负载管理、并行计算中的计算任务负载管理、云计算的计算资源负载管理等领域问题都有大量的研究和实践应用。

[0004] 按照仲裁方式，负载均衡技术主要分为集中式仲裁的负载均衡和分布式仲裁的负载均衡。由于集中式仲裁需要设置一个集中的负载均衡器，因此容易出现单点失效；并且随着分布式集群处理节点的增多，其处理性能也会受到一定的限制，容易成为性能瓶颈。分布式仲裁的负载均衡已经逐渐成为新的研究方向。P2P 技术由于具有全分布、无中心节点等优势，进行负载决策时不会出现性能瓶颈和单故障点问题，而且，集群的伸缩性比较好，因此成为分布式负载均衡研究和应用的新方向。在 P2P 技术中，分布式哈希表 DHT(Distributed HashTable)是一种广泛应用的方法。其主要思想是通过哈希函数建立各个请求和分布式集群处理节点之间的映射关系，并在分布式集群处理节点处于动态增加和移出时，能够最小化地改变每个请求和分布式集群处理节点之间映射关系。

[0005] 基于 P2P 技术的分布式负载均衡方法的现有技术存在两个问题亟待解决：节点动态变化的管理和节点请求的负载均衡。

[0006] 现有技术中，基于 DHT 实现的负载均衡经典算法方案有两种：Chord 算法方案和 Kademlia 算法方案。这两种方案都很好地解决了分布式集群环境下的节点动态变化管理和数据快速存取的处理过程，但是，它们都没有很好地解决节点请求负载均衡的问题，系统中的节点可能出现负载过重或过轻的现象。其中，

[0007] Chord 算法在一致性哈希算法的基础上，提高了查找指定请求所在节点的效率，但是，在节点数量不多时，很难保证每个节点在哈希环上的分布是否具有随机性，这个问题又会造成哈希环上相邻节点之间的间隔不够均匀，从而带来某些节点负载过重或者过轻的后果。

[0008] Kademlia 算法在一致性哈希算法的基础上，根据请求的关键字和节点 ID 的相似度来选择为其服务的节点。在查询选择时，通过异或运算来优化查询效率，即查找与请求关键字的异或运算结果最小的节点，用作为其服务的节点，这个方法具有一定的随机性。但是，该方案没有考虑节点的负载情况，而且，系统中每个节点的负载分配跟请求的关键字有较大的关联，造成系统中每个节点的负载出现不均衡的概率比较大，因此，基于 Kademlia 算法虽然和 Chord 算法同样解决了分布式集群环境下的处理节点动态变化和数据快速存

取过程。但是，它们的共同缺点是：均没有负载均衡的策略，系统中的节点都可能出现负载过重或者过轻的现象。

[0009] 查找现有的论文和专利申请等资料，发现有 2 篇专利申请涉及到基于 P2P 技术的分布式负载均衡机制，分别介绍和对比分析如下。

[0010] 《一种集群服务的负载均衡方法和装置》（公开号：CN 102137128A）介绍一种基于反馈机制的动态负载均衡技术，其方法是：首先获得集群中负载节点的负载能力，再根据负载能力获得负载节点的负载因子，再根据负载因子生成负载分配序列；当接收到待分配的服务请求时，根据服务请求生成随机数，再把生成的随机数根据所有节点负载因子总和取模余，得到一个参考值，再以负载节点的负载因子为参考值的节点做为接受请求的节点。但是，其没有提供解决节点动态变化的管理问题，在分布式集群环境下应用具有一定的局限性。

[0011] 《一种 DHT 网络负载均衡装置及虚节点划分的方法》（公开号：CN101834897A）介绍了一种 DHT 网络负载均衡装置结构组成，以及其虚节点的划分方法：节点加入网络时，通过性能模型定义自身节点级别，若本身为弱节点，找到临近的弱节点，并与邻近的弱节点合并；接着，该节点退出网络并与找到的临近弱节点建立连接，共同组成一个强虚节点，同时退出节点，通过相邻弱节点与系统中的其他节点进行交互，其他节点将退出节点看成为强虚节点的一部分；节点合并后的调整，使得节点查询装置只包含强虚节点 ID，由强虚节点给各弱节点分配负载，实现负载均衡。该方法有效解决了节点动态变化的管理问题，但是，因为强虚节点的数量少于弱节点的数量，不可避免造成弱节点扎堆和系统整体均衡性不佳的情况。

[0012] 综合上述两个专利申请的方技术案，尽管其在一定程度上解决节点动态变化和系统负载均衡的问题。其系统负载均衡性是通过三层来保证的：第一层是请求到参考值的均衡分布，第二层是参考值到强虚节点的均衡映射，第三层是强虚节点将请求均衡分配给其负责的各弱节点。但是，这样处理仍然不能很好地解决分布式集群环境中的负载均衡问题，其原因在于：首先因其存在强虚节点层，当一个物理节点变动时，需要更改弱节点到强虚节点的映射关系和强虚节点到参考值的映射关系，一个弱节点的变动可能还会引起其邻居强虚节点的变动；因此当系统规模较大、节点动态变化较频繁时，强虚节点的变更给系统带来的开销会严重影响系统的整体性能和稳定性。其次，由于其强虚节点作为多个弱节点、即实际处理节点的集合，在第二层参考值到强虚节点的均衡映射只是保证了请求到弱节点集合的均衡性，并没有实现分布式集群系统全局意义上的负载均衡。因此，如何对现有的分布式负载均衡技术继续进行改进和提高，就成为业内科技人员关注的新课题。

发明内容

[0013] 有鉴于此，本发明的目的是提供一种基于 P2P 技术的分布式负载均衡系统和方法，本发明系统用于分布式网络环境下，能够实现具有分布式仲裁功能、有效避免系统出现单点失效情况的负载均衡，该系统和方法能够支持分布式集群系统中节点增加和移出等情况下实现动态变化的负载均衡，并且可以有效保证环境变化后各节点的负载均衡。

[0014] 为了达到上述目的，本发明提供了一种基于 P2P 技术的分布式负载均衡系统，其特征在于：该系统由具有均衡负载功能的分布式处理集群所组成，该处理集群内没有集中

控制的中心节点，只设有分设于不同服务器、并分别用作该系统分布式控制核心的多个同构或异构的处理节点，每个处理节点都能独立提供分布式仲裁的负载均衡功能，且各自的处理事件的权限与均衡负载的功能都相同，异构处理节点的差异只是其事件的处理速率不同；当客户端向分布式处理集群实时发送海量数据处理请求时，该系统的某个处理节点完成请求处理后，将产生的中间结果封装为事件，发往其它处理节点继续处理；各处理节点在分发事件过程中，始终采用分布式负载均衡方法保证事件分发的负载均衡，直到产生最终结果并返还给客户端；该处理节点设有处理层和通信层；其中：

[0015] 处理层，由包含多个处理单元的处理单元容器所组成，负责进行事件处理：接收来自通信层的事件，再根据事件类型和每个事件所设定属性的属性值，将该事件交由特定处理单元进行处理，同时负责将该事件处理完成后产生的新事件或最终结果，交给通信层进行分发与发送；

[0016] 通信层，用于监听、分发事件及实现分布式负载均衡：负责从网络接收事件并转交给处理层；然后接收处理层完成处理后产生的新事件或最终结果，再使用分布式负载均衡方法将该新事件或最终结果分发给其他处理节点或客户端，保证该分布式处理集群中的各处理节点负载均衡；设有：事件监听、事件分发、事件发送、负载管理和分布式集群管理共五个模块。

[0017] 为了达到上述目的，本发明还提供了一种基于 P2P 技术的分布式负载均衡系统的工作方法，其特征在于：每个处理节点的通信层中的事件分发模块和负载管理模块协同完成基于虚拟节点与处理节点之间映射的事件分发方法，实现基于 P2P 技术的分布式负载均衡；所述方法包括下述两个操作步骤：

[0018] 步骤 1，每个处理节点的通信层中的负载管理模块创建或更新虚拟节点映射表和动态负载信息表：

[0019] 步骤 2，每个处理节点的通信层中的事件分发模块与负载管理模块交互，根据虚拟节点映射表执行事件分发决策操作，将每个事件分发到相应的处理节点，并实现负载均衡：

[0020] 事件分发模块计算每个事件的关键属性的哈希值，再将该哈希值除以虚拟节点总个数所得到的余数，作为将该事件准备分发到的虚拟节点编号；

[0021] 事件分发模块与负载管理模块交互，从虚拟节点映射表查找该虚拟节点编号及其所对应的处理节点编号，以获取准备将该事件分发到的处理节点编号。

[0022] 本发明系统和方法的主要创新技术是：采用在系统的通信层建立虚拟节点映射表和动态负载信息表来实现分布式负载均衡。本发明中的虚拟节点概念与 DHT 算法中一致性哈希的哈希环值有相似之处，但是，本发明的事件分发方法是对一致性哈希中的均衡策略的改进，更具优势。究其原因在于：DHT 算法中的一致性哈希是将资源映射到哈希环上的数值，然后，某段哈希环上的数值就与某个处理节点相对应；由于处理节点在哈希环上是随机分布的，在查询处理节点时，需要顺时针或逆时针遍历所有处理节点的集合，这样就会产生处理节点扎堆和查询效率不高的缺陷。

[0023] 本发明是先用哈希方法获取虚拟节点编号，然后再映射到处理节点。本发明保证虚拟节点编号可以均匀地分布在哈希环上。此外，通过第一阶段的哈希方法分散性和第二阶段的虚拟节点映射到处理节点的均衡性，以保证系统全局意义上的均衡性，并且，在查询

处理节点时，只需在虚拟节点映射表中一次查询便能获知处理节点，查询效率高。因而，本发明分布式负载均衡方法的实质是一种改进的 DHT 实现方法。多次的仿真实施例的试验证明，本发明具有很好的推广应用前景。

附图说明

[0024] 图 1 是本发明基于 P2P 技术的分布式负载均衡系统总体结构组成示意图。

[0025] 图 2 是本发明分布式负载均衡系统中的处理节点组成结构示意图。

[0026] 图 3 是本发明基于 P2P 技术的分布式负载均衡系统的工作方法流程图。

[0027] 图 4 是本发明分布式负载均衡系统中的事件分发方案示意图。

[0028] 图 5 是本发明分布式负载均衡系统中分布式负载均衡事件分发时序图。

[0029] 图 6 是本发明分布式负载均衡系统中新增处理节点表的更新操作时序图。

[0030] 图 7 是本发明分布式负载均衡系统中处理节点动态移出表的更新操作时序图。

具体实施方式

[0031] 为使本发明的目的、技术方案和优点更加清楚，下面结合附图和实施例对本发明作进一步的详细描述。

[0032] 本发明是一种基于对等 P2P（Peer-to-Peer）技术的分布式负载均衡系统，同时，提供了一种基于该系统的负载均衡方法。首先介绍系统结构组成：

[0033] 参见图 1，本发明基于 P2P 技术的分布式负载均衡系统由具有均衡负载功能的分布式处理集群所组成，该处理集群内没有中心节点（即集中控制节点），只有分设于不同服务器、并分别用作该系统分布式控制核心的多个同构或异构的处理节点（实际环境部署时，同一台服务器也可以设置多个处理节点），每个处理节点都能独立提供分布式仲裁的负载均衡功能，且各自处理事件的权限与均衡负载的功能都相同，异构处理节点的差异只是其事件的处理速率不同。当客户端向分布式处理集群实时发送海量数据处理请求时，该系统的某个处理节点完成请求处理后，将产生的中间结果封装为事件，并发往其它处理节点继续处理。而且，各处理节点在分发事件过程中，始终采用分布式负载均衡方法保证事件分发的负载均衡，直到产生最终结果并返还给客户端。

[0034] 参见图 2，介绍处理节点内部功能模块，其设有处理层和通信层；其中：

[0035] 处理层由包含多个处理单元的处理单元容器所组成，负责进行事件处理：接收来自通信层的事件，再根据事件类型和每个事件所设定属性的属性值，将该事件交由特定处理单元进行处理，同时负责将该事件处理完成后产生的新事件或最终结果，交给通信层进行分发与发送。处理层的两种组成部件功能如下：

[0036] 处理单元容器，用于管理和控制其内部的各个处理单元，采用设定顺序激活相应的处理单元来处理不同事件。

[0037] 处理单元，用于处理特定类型和属性的事件，负责藉由通信层的事件分发模块和事件发送模块，将产生的中间处理结果作为新事件，发往其他处理单元继续处理；或者将处理的最终结果返回给客户端；每个处理单元都是系统预先设置的，或者由第三方程序员开发并部署到处理节点上。

[0038] 通信层用于监听、分发事件及实现分布式负载均衡（参见第 4 节）：负责从网络接收

事件并转交给处理层，然后接收处理层完成处理后产生的新事件或最终结果，再使用分布式负载均衡方法将该新事件或最终结果分发给其他处理节点或客户端，保证该分布式处理集群中的各处理节点负载均衡。设有下述五个模块：事件监听、事件分发、事件发送、负载管理和分布式集群管理模块：

[0039] 事件监听模块，负责接收网络事件，并交由处理层的处理单元进行处理。

[0040] 事件分发模块，负责接收处理层的新事件或最终结果，并按照负载管理模块提供的负载均衡信息来分发新事件或最终结果，再经由事件发送模块将其发送到其他处理节点或客户端。

[0041] 事件发送模块，用于将来自事件分发模块的新事件或最终结果分别发送到设定的处理节点或客户端，分发过程中保证系统负载均衡。

[0042] 负载管理模块，其为处理节点实现分布式负载均衡方法的关键模块，负责创建与维护事件分发模块所需的负载信息，并为此定义了下述几个重要概念：

[0043] （A）虚拟节点是为了实现分布式负载均衡方法而设置的一个与处理节点相互映射的逻辑符号，每个虚拟节点只能映射到一个处理节点，但是，每个处理节点能够映射到多个虚拟节点。虚拟节点个数是在系统启动时设置的，其数量远大于处理节点个数；且在系统运行后，虚拟节点个数保持不变；当系统新增或退出处理节点时，处理节点及其虚拟节点之间的映射关系也要根据调整算法作相应改变。

[0044] （B）虚拟节点映射表是本发明的重要数据结构，设于每个处理节点的负载管理模块，用于表示每个虚拟节点编号及其对应的处理节点编号，以便在系统分发事件时，查找该虚拟节点映射表，根据虚拟节点编号获取其对应的处理节点。虚拟节点映射表是根据每个处理节点的事件处理速率高低降序排列形成的；因每个处理节点处理事件的速率不同，故每个处理节点所分配的对应虚拟节点个数也不相等；且在系统运行后，为节省更新时间，采取增删处理节点的动态负载信息表和虚拟节点映射表的更新方法来实时更新虚拟节点映射表。

[0045] 下表 1 就是一个虚拟节点映射表的示例：其中第一项是虚拟节点编号，第二项是该虚拟节点对应的处理节点编号。在需要进行事件分发时，即本发明事件分发第二阶段就是通过根据虚拟节点编号查找虚拟节点映射表的方式，获取该虚拟节点对应的处理节点。

[0046]

[0047]

虚拟节点	处理节点
虚拟节点 0	处理节点 0
虚拟节点 1	处理节点 1
虚拟节点 2.....	
虚拟节点 3	处理节点 K
虚拟节点 4	处理节点 0
虚拟节点 5	处理节点 1
虚拟节点 6.....	

虚拟节点 7	处理节点 K ....
虚拟节点 N	处理节点 K

[0048] 本发明实施例是根据每个处理节点的事件处理速率的高低，采用降序排列方式来初始化虚拟节点映射表。实际应用时，也可用其它能够保证按照处理节点的事件处理速率高低降序排列的方法来初始化虚拟节点映射表，而且，对于每个处理效率不同的处理节点，可以给其分配数量不同的虚拟节点。

[0049] 需要注意的是，虚拟节点映射表的初始化方法仅适用于初始构建的时候。系统运行后，为节省虚拟节点映射表的更新时间，不再采用该初始化的方法来更新虚拟节点映射表。在动态新增或移出处理节点后，本发明是采用动态新增或移出处理节点的动态负载信息表和虚拟节点映射表的更新方法来实时更新虚拟节点映射表。

[0050] (C)动态负载信息表是是本发明的另一重要数据结构，设于每个处理节点负载管理模块，用于表示每个处理节点编号、按照其事件处理速率在理论上分配的虚拟节点个数与其实际对应的虚拟节点个数。本发明系统中各处理节点对应的虚拟节点个数称为节点负载数。在初始建立虚拟节点映射表时，为方便维护每个处理节点的负载信息，根据虚拟节点映射表信息建立了一张动态负载信息表。该动态负载信息表是按照每个处理节点的负载数量与其理论上分配的负载数量之差的绝对值降序排列形成的，且其信息与虚拟节点映射表的信息逐一对应的。故在虚拟节点映射表更新时，动态负载信息表也要相应更新。

[0051] 下表 2 是一个动态负载信息表的示例：其中第一项是处理节点编号，第二项是该处理节点对应的虚拟节点个数，第三项是一个列表，包含该处理节点对应的各个虚拟节点编号。动态负载信息表中每一项是按照处理节点负载数从大到小降序排列的。同时，为了保证分布式处理集群中增加处理节点时，新增的处理节点和虚拟节点的映射集均匀分开，在创建动态负载信息表时，对每个处理节点对应的虚拟节点编号执行乱序操作。

[0052]

处理节点	节点负载数	虚拟节点列表
处理节点 0	256	40,4,0,.....
处理节点 1	256	5,41,1,..... ....
处理节点 K	256	19,7,3,.....

[0053] 虚拟节点映射表与动态负载信息表之间的关系：在系统启动时，借助分布式集群管理模块中的信息创建虚拟节点映射表；在系统启动后，利用已经建立的虚拟节点映射表

信息创建动态负载信息表，且在分布式处理集群中有新增或退出处理节点时，对虚拟节点映射表和动态负载信息表进行实时更新。

[0054] 分布式集群管理模块，负责管理和维护分布式处理集群及其中的全部处理节点与其对应的虚拟节点，该分布式处理集群中的各处理节点也经由该分布式集群管理模块实时获取该集群中自身与其他处理节点的链接地址、当前存活状态、负载状态与相关信息。

[0055] 本发明系统的应用场景主要有下述三种：

[0056] (1)分发事件场景：客户端向分布式负载均衡系统中的分布式处理集群发送海量数据处理请求，分布式处理集群中的每个处理节点分别独立、均等地完成各自的负载决策，且在处理过程中，保证每个处理节点的负载均衡；并实时向客户端返回处理结果；

[0057] (2)新增处理节点场景：客户端向分布式处理集群发送海量数据处理请求过程中，为减轻该处理集群中各处理节点的负载压力，向该处理集群中实时增加新的处理节点；并在尽可能短时间内，由新增处理节点分担原处理集群中各处理节点的部分负载，使得新增处理节点和原处理集群中各处理节点重新实现负载均衡。

[0058] (3)移出处理节点场景：客户端向该分布式处理集群发送海量数据处理请求过程中，因故障或管理需要等原因，有节点要迁移离开该处理集群时，由该集群中其它处理节点承担该移出处理节点的负载；并在尽可能短时间内，使得移出一个或多个处理节点后的处理集群中的每个处理节点重新达到负载均衡。

[0059] 参见图3，介绍本发明基于P2P技术的分布式负载均衡系统的工作方法：每个处理节点的通信层中的事件分发模块和负载管理模块协同完成基于虚拟节点与处理节点之间映射的事件分发方法，实现基于P2P技术的分布式负载均衡。包括下述两个操作步骤：

[0060] 步骤1，每个处理节点的通信层中的负载管理模块创建或更新虚拟节点映射表和动态负载信息表。该步骤1包括下列操作内容：

[0061] (11)系统初始建立时，负载管理模块根据每个处理节点的事件处理速率高低采用降序排列方式创建初始化虚拟节点映射表，此时对于事件处理速率不同的处理节点，分别设置相应不同的虚拟节点个数。该步骤包括下列操作内容：

[0062] (11A)为保证负载均衡，负载管理模块先计算系统当前每个处理节点理论上获取的负载个数，即其理论负载个数loadAfter(i)为：

[0063] $$loadAfter(i)=VIRTUAL\_NODES\_NUM\times\frac{P\_CAPACITY(i)}{\sum_{i=0}^{i=N-1}P\_CAPACITY(i)}$$；式中，VIRTUAL_NODES_NUM是虚拟节点总个数，其数值远大于系统处理节点总个数N；自然数i是处理节点编号，其最大值为N；P_CAPACITY(i)是编号为i的处理节点的事件处理速率，该事件处理速率取决于包括CPU和内存的权衡因子，或者由专业测试工具事先测定的；$\sum_{i=0}^{i=N-1}P_CAPACITY(i)$是系统当前所有处理节点的事件处理总速率；

[0064] (11B)负载管理模块顺序给每个处理节点随机分配loadAfter(i)个负载，直到所有虚拟节点都有一个与其对应的处理节点；且每次分配都要分别在动态负载信息表和虚拟节点映射表执行相应更新操作。

[0065] (11C)负载管理模块对动态负载信息表中各个处理节点按照其当前的实际负载数与理论负载数之差的绝对值进行降序排列。

[0066] (12)系统运行后，当新增或移出处理节点时，为使系统尽快重新达到负载均衡，负

载管理模块实时更新虚拟节点映射表和动态负载信息表。

[0067] 步骤 2（参见图 4），每个处理节点的通信层中的事件分发模块与负载管理模块交互，根据虚拟节点映射表执行事件分发决策操作，将每个事件分发到相应的处理节点，并实现负载均衡：

[0068] 事件分发模块计算每个事件的关键属性的哈希值，再将该哈希值除以虚拟节点总个数所得到的余数，作为将该事件准备分发到的虚拟节点编号；

[0069] 事件分发模块与负载管理模块交互，从虚拟节点映射表查找该虚拟节点编号及其所对应的处理节点编号，以获取准备将该事件分发到的处理节点编号。

[0070] 图 4 所示的系统事件分发机制示意图是本发明方法的关键，为了更详细地介绍其操作流程，下面根据本发明三种应用场景分别介绍三种不同场景下的分布式负载均衡事件分发流程。

[0071] 参见图 5，说明分发事件场景的分布式负载均衡事件分发操作流程，其对应的是本系统进行海量数据处理的场景。分发事件场景包括下列操作内容：

[0072] （A1）处理节点的事件监听模块监测到网络中新出现需要处理的事件，就根据该事件类型将其发送到处理单元容器，交给相应的处理单元进行处理。需要说明的是，分布式处理集群中每个处理节点都在时刻监听是否有需处理事件。

[0073] （A2）该处理单元完成事件处理后，将新产生的事件准备发往其他处理节点或由其自身继续处理。

[0074] （A3）事件分发模块先计算该新事件属性的哈希值，再利用哈希方法、即将该哈希值除以虚拟节点总个数所得到的余数（启动时配置，系统启动后为常量），作为对应的虚拟节点编号。

[0075] （A4）事件分发模块与负载管理模块交互，根据该虚拟节点编号查询虚拟节点映射表（参见表 1），得到该新事件准备分发到的处理节点编号；然后，将该新事件与要分发的对应处理节点编号信息传给事件发送模块。

[0076] （A5）事件发送模块将该新事件发往其他处理节点或自身节点继续处理，直到产生事件最终结果，返回给客户端。

[0077] 需要注意的是，上述两个步骤（A3）和（A4）是本发明负载均衡方法的核心流程，可以看到各个处理节点分别执行负载均衡策略，无中心负载均衡处理节点，是一种典型的基于 p2p 技术的分布式负载均衡方法。

[0078] 参见图 6，介绍新增处理节点场景时、即系统进行海量数据处理时，新增一个处理节点后，其他处理节点的部分负载分摊给新处理节点，使得分布式处理集群重新达到负载均衡的操作流程。该场景包括下列操作内容：

[0079] （B1）处理节点的分布式集群管理模块检测到新增处理节点加入，就将该信息通知分布式集群中的所有处理节点的负载管理模块，并提供该新增处理节点的配置信息（包括该新增处理节点编号、IP 地址、事件处理速率和端口号）。

[0080] （B2）其他各个处理节点负载管理模块接收上述信息，再根据该新增处理节点信息对动态负载信息表和虚拟节点映射表进行更新。该更新方法借鉴了一致性哈希在节点加入系统时尽量减少系统拓扑维护的思路：设置一个通用场景，用于说明增加一个处理节点时的动态负载信息表和虚拟节点映射表更新方法。假设当前系统设有 N 个处理节点，总共有

VIRTUAL_NODES_NUM 个虚拟节点(VIRTUAL_NODES_NUM 远大于 N)。现在要增加一个处理节点,也就是处理节点从 N 个增加到 N+1 个。具体操作内容如下 :

[0081] (B2A) 为保证负载均衡,负载管理模块先计算系统当前包括新增处理节点在内的所有处理节点理论上将获取的负载个数,即其理论负载个数 loadAfter(i) 为 :$loadAfter(i) = VIRTUAL\_NODES\_NUM * \frac{P\_CAPACITY(i)}{\sum_{i=0}^{i=N} P\_CAPACITY(i)}$,式中,自然数 i 是处理节点编号,其最大值为 N ;,VIRTUAL_NODES_NUM 是虚拟节点总个数 , 其数值远大于系统处理节点总个数 N ;P_CAPACITY(i) 是编号为 i 的处理节点的事件处理速率,$\sum_{i=0}^{i=N} P_CAPACITY(i)$是系统当前所有处理节点的总事件处理速率 ;

[0082] (B2B) 负载管理模块将动态负载信息表按照处理节点的实际负载数与理论负载数之差的绝对值进行降序排列 ;

[0083] (B2C) 负载管理模块按照调整后的动态负载信息表的排列顺序,从绝对值最大的处理节点 k 开始,将(|loadAfter(k)-loadActual(k)|)个负载转移给新增处理节点,式中,loadActual(k) 为处理节点的实际负载个数 ;直到新增处理节点 i 的理论负载个数与实际负载个数相等 ;需要注意的是 :对于需要转移负载给新增处理节点的最后一个处理节点 l,其转移的负载个数应不多于(|loadAfter(l)-loadActual(l)|),以保证该新增处理节点所分配的实际负载个数不大于其理论分配的负载个数,且负载数多的处理节点移出更多的负载 :

[0084] (B2D) 根据上述步骤调整结果,对虚拟节点映射表和动态负载信息表做相应更新 :遍历处理节点的动态负载信息表,对于该表中每个处理节点转移(|loadAfter(k)-loadActual(k)|)个负载给新增处理节点,直到新增处理节点 i 理论负载个数 loadAfter(i) 与实际负载个数相等 loadActual(i) ;且每次转移都要分别在动态负载信息表和虚拟节点映射表执行相应更新操作 ;再在动态负载信息表上新建一行,用于表示该新增处理节点的负载信息 ;最后,对动态负载信息表中各个处理节点重新按照其实际负载数与理论负载数之差的绝对值进行降序排列。

[0085] (B3) 每个处理节点负载管理模块完成虚拟节点映射表和动态负载信息表的更新后,新增处理节点就分担该分布式处理集群中的所有原处理节点的部分负载,使得新增处理节点和该处理集群中的所有原处理节点很快重新达到负载均衡。

[0086] 参见图 7,介绍对于处理节点移出场景的分布式负载均衡表更新流程。该场景是 :本发明系统进行海量数据处理时,某个处理节点因故障或管理原因被移出系统后,分布式处理集群其他处理节点取代移出处理节点工作,重新达到负载均衡的过程。此时的操作步骤如下 :

[0087] (C1) 处理节点的分布式集群管理模块采用各个处理节点之间互发心跳信息或其他方法,检测到该集群中有移出处理节点的状况 :处理节点正常关闭或因故障而宕机,使得该集群中的处理节点减少的现象。

[0088] (C2) 发现移出处理节点的处理节点中的分布式集群管理模块,通知该分布式集群中的所有处理节点的负载管理模块有处理节点移出,并提供该移出处理节点包括其编号、IP 地址、事件处理速率和端口号的配置信息。

[0089] (C3) 其他各个处理节点的负载管理模块接收上述信息后,根据该移出的处理节点

信息更新动态负载信息表和虚拟节点映射表。这个更新过程是重新达到负载均衡的核心，本发明对于处理节点移出的动态负载信息表和虚拟节点映射表的更新方法同样也借鉴了一致性哈希在节点加入系统时尽量减少系统拓扑维护的思想。处理节点移出的动态负载信息表和虚拟节点映射表的详细更新过程包括下列操作内容：

[0090] （C3A）假设原处理节点总个数为N，虚拟节点总个数为VIRTUAL_NODES_NUM，且VIRTUAL_NODES_NUM远大于N，此时移出一个处理节点，即当前处理节点总个数为(N-1)；同时，计算当前系统各处理节点的理论负载为：$loadAfter(i) = VIRTUAL\_NODES\_NUM \times \frac{P\_CAPACITY(i)}{\sum_{i=0}^{i=N-2} P\_CAPACITY(i)}$；并对动态负载信息表中各个处理节点重新按照其实际负载数与理论负载数之差的绝对值降序排列；

[0091] （C3B）为保证移出处理节点的负载转移后，仍然实现均衡负载，采取下述更新策略：按照调整后的动态负载信息表的排列顺序，从绝对值最大的处理节点k开始，逐个进行下述操作：从该移出处理节点处获取(|loadAfter(k)-loadActual(k)|)个负载，直到该移出处理节点的所有负载都已被重新分配给其他各个处理节点；需要注意的是，操作到动态负载信息表中最后一个处理节点时，如果该移出处理节点的负载还没有全部被分配出去，则将该移出处理节点剩余的所有负载都转移给最后一个处理节点，保证负载数少的处理节点能从该移出处理节点处获取更多负载；

[0092] （C3C）根据上述操作结果，相应修改虚拟节点映射表和动态负载信息表：遍历处理节点的动态负载信息表，该表中的每个处理节点从该移出处理节点处获取(|loadAfter(k)-loadActual(k)|)个负载，其中，自然数k为遍历时的当前处理节点编号；直到该移出处理节点的所有负载都已被重新分配给其他各个处理节点；且每次转移负载都要分别在动态负载信息表和虚拟节点映射表执行相应修改操作；

[0093] （C34）在动态负载信息表上删除已经移出的处理节点信息；最后，对动态负载信息表中各个处理节点重新按照其实际负载数与理论负载数之差的绝对值进行降序排列。

[0094] （C4）每个处理节点负载管理模块更新完成虚拟节点映射表和动态负载信息表后，该集群中的其他各个处理节点就承担了该退出处理节点的负载，并在尽可能短的时间内，移出处理节点后的该分布式处理集群重新实现负载均衡。

[0095] 本发明已经进行了大量的仿真实施试验，试验的结果成功的，实现了发明目的。

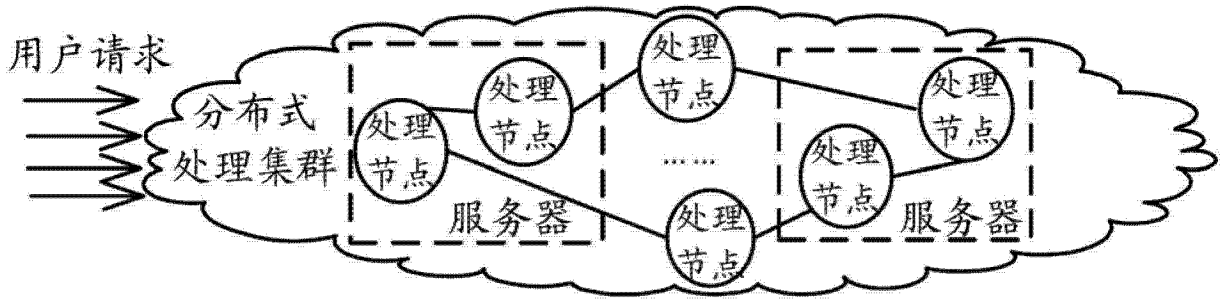

用户请求
分布式
处理集群
处理
节点
处理
节点
处理
节点
……
处理
节点
处理
节点
处理
节点
服务器
服务器

图 1

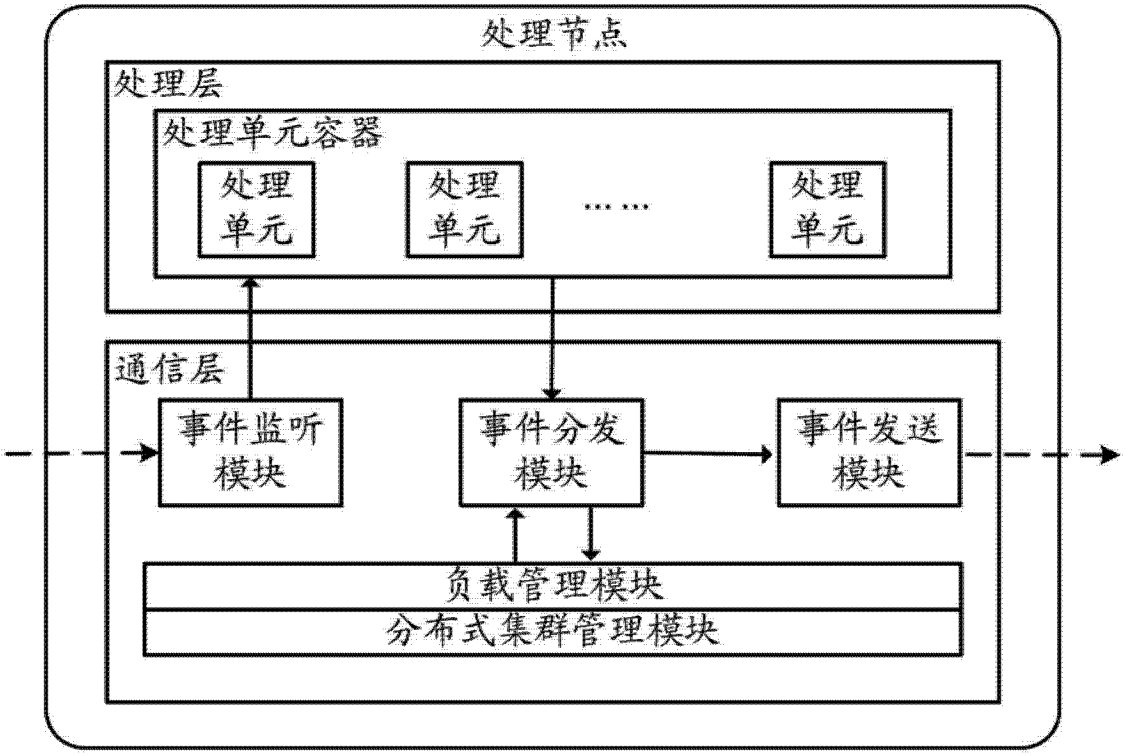

处理节点
处理层
处理单元容器
处理
单元
处理
单元
……
处理
单元
通信层
事件监听
模块
事件分发
模块
事件发送
模块
负载管理模块
分布式集群管理模块

图 2

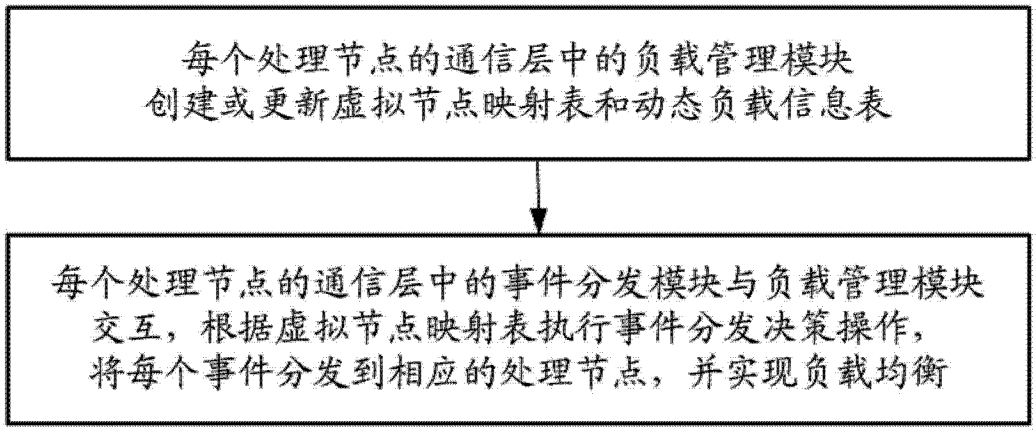

每个处理节点的通信层中的负载管理模块
创建或更新虚拟节点映射表和动态负载信息表
每个处理节点的通信层中的事件分发模块与负载管理模块
交互，根据虚拟节点映射表执行事件分发决策操作，
将每个事件分发到相应的处理节点，并实现负载均衡

图 3

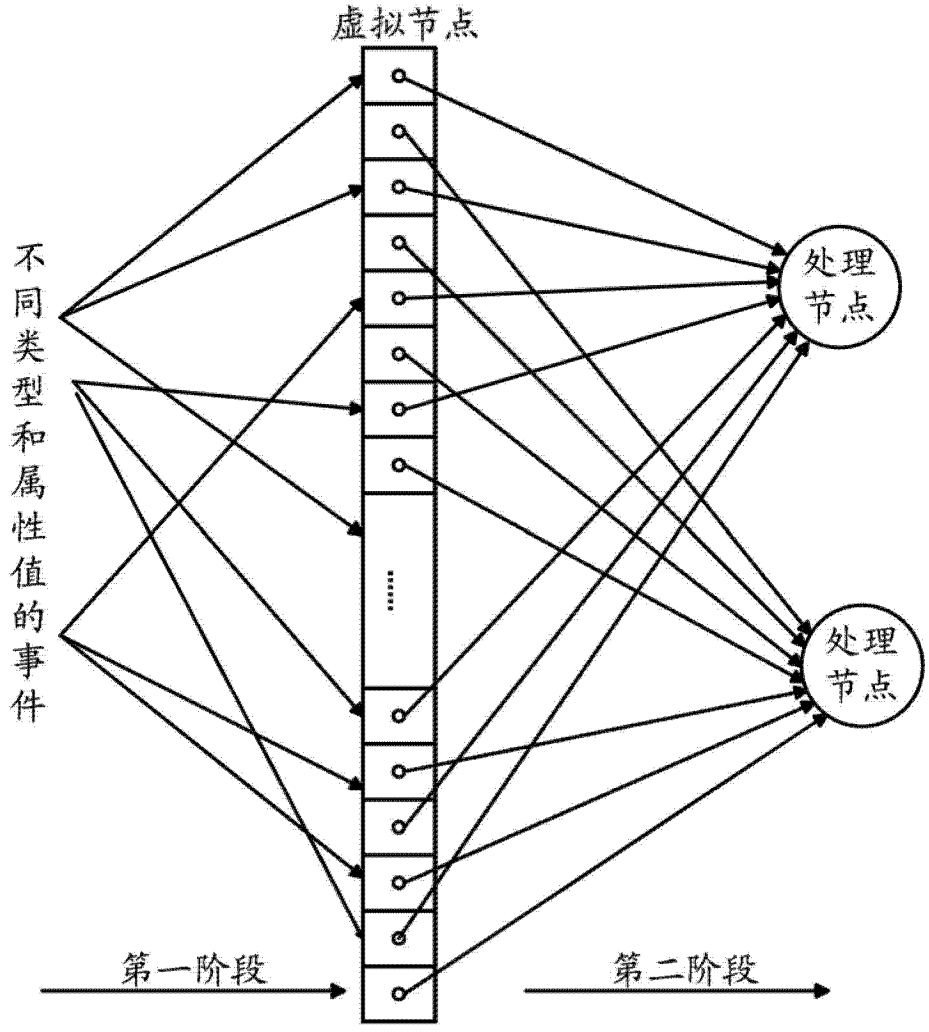

虚拟节点
不同类型和属性值的事件
处理节点
处理节点
第一阶段
第二阶段

图 4

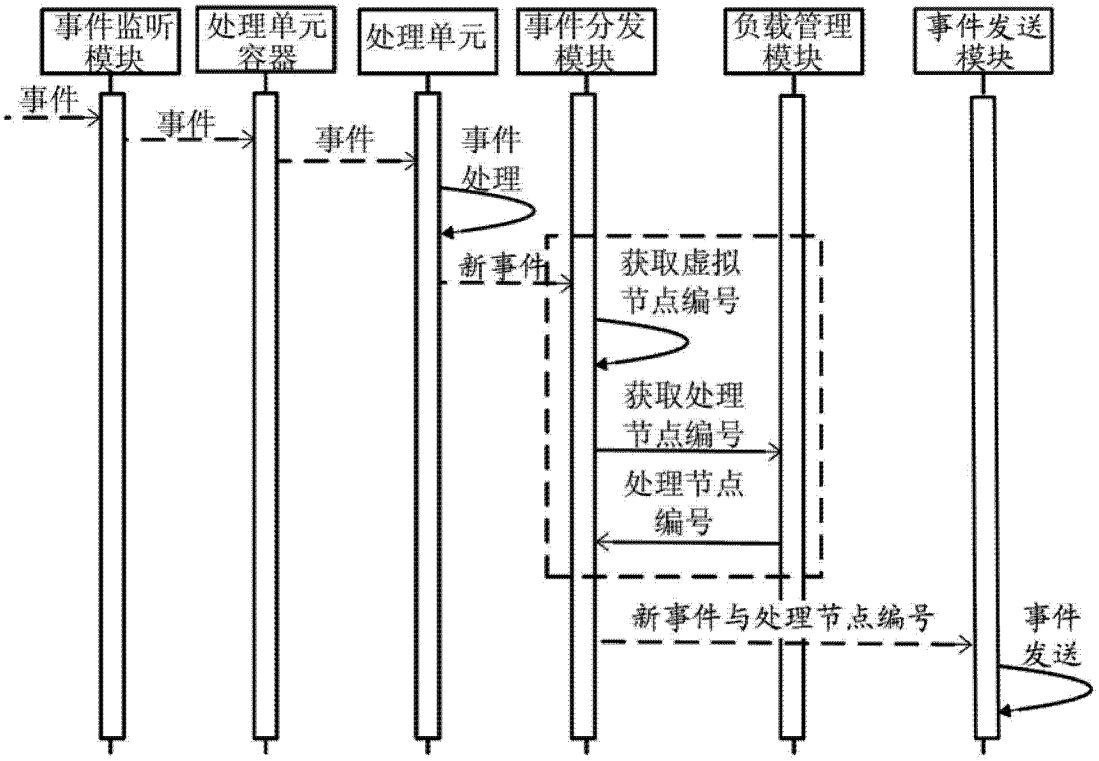

事件监听模块
处理单元容器
处理单元
事件分发模块
负载管理模块
事件发送模块
事件
事件
事件
事件处理
新事件
获取虚拟节点编号
获取处理节点编号
处理节点编号
新事件与处理节点编号
事件发送

图 5

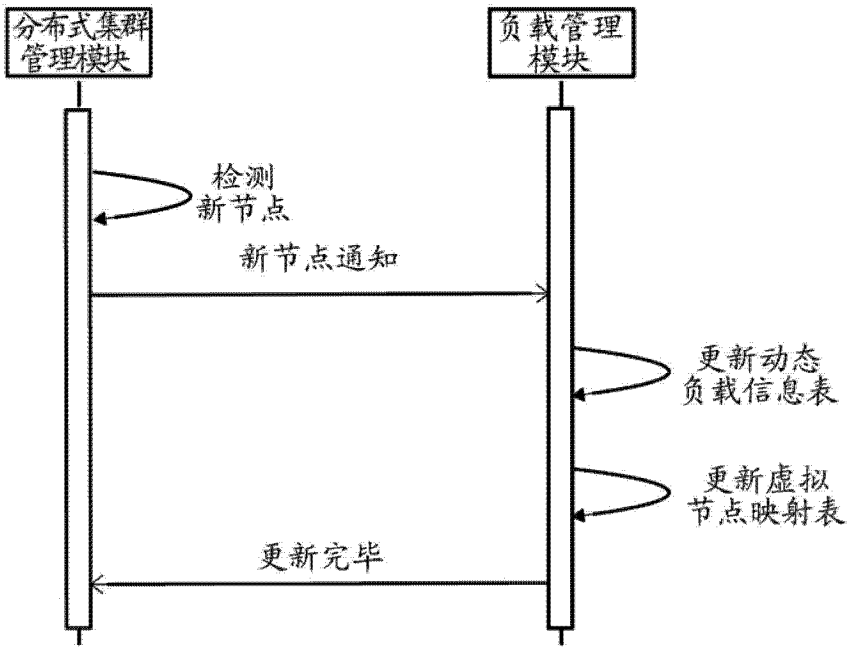

分布式集群管理模块
负载管理模块
检测新节点
新节点通知
更新动态负载信息表
更新虚拟节点映射表
更新完毕

图 6

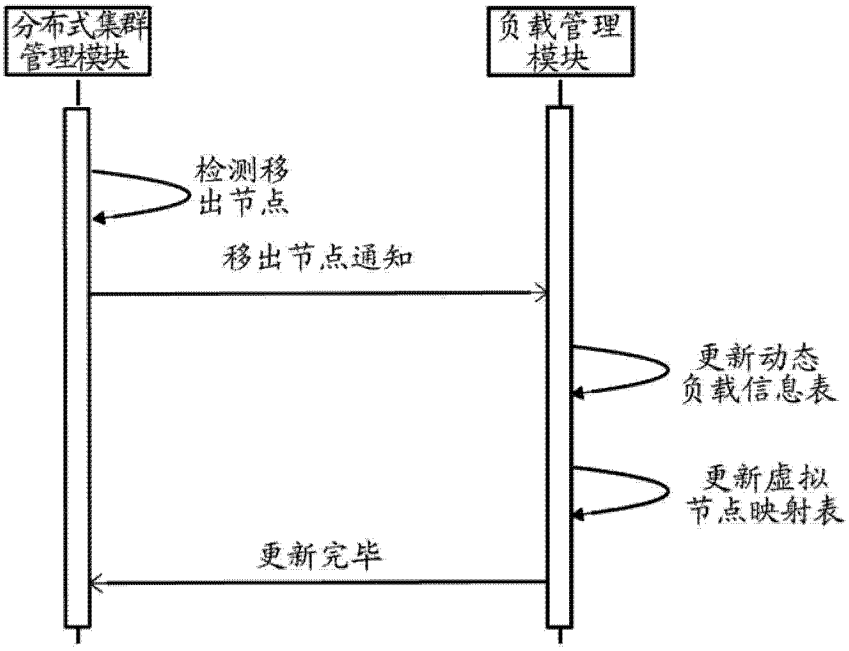


图 7