

网格环境中带度约束的多播资源查找算法^{*}

于显平 蒲 汛 余建桥

(西南大学计算机与信息科学学院 重庆 400715)

摘 要 网格计算的前提是资源查找。本文分析研究了几种适应某些网格资源模型的现有资源查找算法及其时间和空间复杂度。针对有多播特征的网格环境中的资源查找,基于多播功能,同时赋予资源节点不同权值,构造带度的多播网格资源模型,提出带度约束的多播资源查找算法。与现有算法相比,此算法能更有效实现多播网格环境中资源的快速查找。

关键词 网格,资源查找,多播,带度约束

A Degree-constrained Multicasting Algorithm of Resource Lookup in Grid

YU Xian-Ping PU Xun YU Jian-Qiao

(College of Computer and Information Science, Southwest University, Chongqing 400715)

Abstract The premise of Computing-Grid is searching of resource. This article studied the existing resource locating algorithms and analyzed their cost of time and space. Aiming at searching of resource under the grid environment having characteristic-multicasting, this paper constructed a multicasting grid resource model using the degree of the resource node to represent the resource locating ability of each node, presented a Degree-Constrained Multicasting Algorithm which provide a more effective resource locating method than existing others in that model.

Keywords Grid, Resource locating, Multicasting, Degree-constrained

1 引言

随着高性能应用需求的迅猛发展,单台高性能计算机已经不能胜任一些超大规模应用问题的解决。这就需要将地理上分布、系统异构的多种计算资源通过高速网络连接起来,共同解决大型应用问题,即广域高性能的元计算技术(Meta-computing),也称为网格(Grid)技术。

在单计算机系统和机群系统中,计算资源和信息资源的分布比较集中,计算在使用资源之前可以快速、可靠地进行资源定位,资源的查找操作对计算性能的影响很小。但在网格计算中,由于资源的广域分布以及现有 Internet 存在的带宽和延迟的限制,在广域范围内资源定位在很大程度上影响网格计算的性能。针对各种网格资源存在的网络环境的有效的资源查找算法的研究有了很大的发展,有的算法还应用到实际的环境中,比如穷举查找、集中查找、路由转发,以及这些方法的结合。在实际应用中常见网格资源存在的环境有多播特征,网格中采用多播功能能有效利用带宽与降低时延,但前边这些算法却没考虑到其多播特点,同时这些算法没有根据资源节点提供能力有侧重地选择资源节点进行查询请求。论文研究的多播资源查找算法在前面这些算法技术基础上,利用多播功能的优点,根据资源提供能力大小给不同资源节点赋予不同权值,同时以节点约束度的方式限制多播分支数;相邻节点权值的动态交换使算法能动态跟踪资源节点资源提供能力权值的改变,以便能很好适应网格多播资源拓扑模型。

2 网格资源基础模型分析

网格系统本质上只是一个基础平台,它允许获取与位置无关的资源和服务,这些资源和服务是由分布在不同物理位置上的机器和网络提供。因此可以把网格分为两个部分:即

信息或服务节点,以及连接这些节点的网络。

网格中每个节点对请求的处理能力有可能不完全相同,通常网格中的资源可以归结为三种分布形式:1)不平衡,少数节点提供绝大多数的资源,而大多数节点都只提供少量资源;2)平衡,所有节点上的资源数量相当;3)两者的折衷。根据以上三种形式,我们为每个节点添加一个衡量资源提供能力的权值,根据权值大小,网格中的资源发现就存在从网格某资源节点开始,找到网格目的资源节点的快速查询问题。

2.1 基础模型的定义

定义 1 网格的资源模型可以定义为一个加权无向图,即 $G=(V,E,D)$,其中 $|V|=n$,即节点的个数, $|E|=m$ 表示两节点间的通信链路的全体,在 E 上定义一个代价函数,表示通信时链路的代价, $|D|=q$ 节点的资源提供能力。

定义 2 图 G 中的节点分为两类,即资源节点和信息节点,分别表示为 V_r, V_i 。资源节点是指资源所在的节点,信息节点用于存储资源和资源节点的对应关系。在访问这两种节点时,都需要查找资源的位置信息。通过高效的查找算法(如哈希表),访问节点的时间可以为常数。

定义 3 网格的资源集合表示为 R ,资源的个数表示为 $|R|$,我们假设 $|R| \geq n$,且图中每个节点都是资源节点,即任意节点 $v_i \in V_r$ 。对于每个资源 r ,资源与资源节点的关系表示为 $at(v_i, r)$,其中 $v_i \in V_r$ 。如果 $at(v_i, r)=1$ 表示资源在该节点上,而 $at(v_i, r)=0$ 表示资源不在该节点上。资源节点 v_i 的位置表示为 1_{v_i} 。

定义 4 查找资源 r 的过程可以看成是在图 G 中从资源请求起点 v_0 到达目标资源所在节点 v_i 所经过的路径,即, $P=v_0, v_1, v_2, \dots, v_i$,其中 $1 \leq i \leq n$,且 $at(v_i, r)=1$,此路径的权值表示为 W_p ,假设其存在的最长路径的权值表示为 W_{max} 。

2.2 查找算法

^{*}重庆市科委项目(20038035)资助。于显平 硕士,讲师,主要研究方向是人工智能、软件工程、网络等。蒲 汛 硕士,主要研究方向是计算机网络。余建桥 博士,教授,主要研究方向是人工智能、数据库系统等。

根据上述基础模型,资源查找主要有以下几种方式:穷举查找方法、集中查找方法和路由查找法。在早期的 Globus 系统中使用的是集中查找方法,而在 Globus 的最近实现中采用了穷举查找方法和集中查找方法相结合的方式。

(1) 穷举查找

对于穷举查找方法来说,每次查找的起点 $v_0 \in V_r$ 。 v_0 中存储了所有资源节点的位置信息,而不包括资源与资源节点之间的对应关系。即在查找资源 r 之前,算法已知所有资源节点所在的位置。在节点 v_0 存储的信息中,对于任何 v_i 都已知。即在 v_0 中存有所有 l_{v_i} 的信息。因此这种穷举查找算法在最坏情况下的时间复杂度为 $O(|V| \cdot W_{\max}) = O(n \cdot W_{\max})$;而空间复杂度在最坏情况下为 $O(|V| \cdot |V|) = O(n^2)$ 。

(2) 集中查找

对于集中查找方法来说,整个图 G 中只有一个信息节点 v_s 。 v_s 中存放着所有资源节点的位置信息和资源与资源节点的对应关系。即在存储的信息中,对于任何 $v_i \in V_r$ 都已知 l_{v_i} ,同时都已知 $at(v_i, r) = 1$ 。因此查找资源 r 的过程将分为两部分:首先从起点 v_0 查找找到 v_s 的路径用 W_{ps} 表示;然后通过对 v_s 上相关信息的查询,找到目的节点 v_i 的路径,再从 v_0 开始查找找到 v_i 的路径用 W_{pi} 表示。可以看出整个查找所需的时间为 $W_{ps} + W_{pi}$ 。则集中查找的算法的时间复杂度为 $O(|W_{ps}| + |W_{pi}|)$,最坏情况下是 $O(|W_{ps}| + |W_{\max}|)$,其空间复杂度是 $O(|V_r| + |R|)$,最坏情况下是 $O|R|$ 。

相对而言,这种集中式和穷举查找相结合的方式比较成熟,在 Globus 正是通过 GIIS(Grid Index Information Service)和 GRIS(Grid Resource Information Service)进行集中式的资源查找。

(3) 路由转发 RT(Routing-Transfer) 资源查找算法

在 RT 算法中,图 G 中的每个节点都是信息节点,即对于任意节点 v_i ,都有 $v_i \in V_r$ 。每个信息节点中存放着所有资源的路由信息,而不是该资源的直接位置,即在查找某个资源时,路由信息需要把搜索的方向指向相邻的某个节点。我们用路由函数来表示这种关系:

$$route(v_i, r) = v_n, v_n \in neighbor(v_i).$$

路由转发查找算法的具体如下:

1) 起点 v_0 出发查找资源 r ,选择与 v_0 相连的信息节点 v_{i0} ,根据路由算法求出将要访问的下一个节点 v_{i1} ;

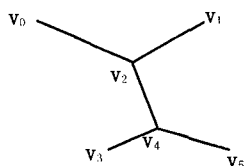


图1 网络的资源分布和节点图

2) 如果 v_{i1} 是资源所在节点,那么查找结束;如果不是资源所在节点,那么根据路由算法算出下一个节点的位置,将此资源请求转发给与当前节点相邻的下一个节点,直到找到所需的资源。如图1所示: V_0 节点到 V_5 节点的查找过程如图2。

图2为每个节点内存储的相关节点的路径。以此为基础查找请求在各节点中不断搜索相关信息最终到达目的地。

此算法的时间复杂度为其查找路径 P_1 所花费的时间 W_{p1} ,最坏情况下时间复杂度为 $O(W_{\max})$ 。其空间复杂度是每个信息节点需要存储所有资源与此信息节点相邻的映射关系,其空间复杂度为 $O|R|$ 。

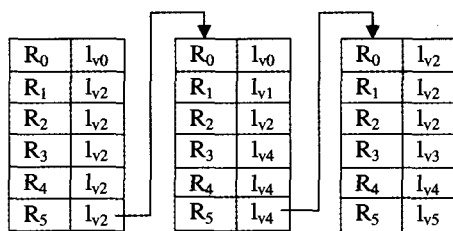


图2 算法流程图

3 带度约束的多播资源查找算法

3.1 网格的多播资源模型

前边的算法处理只涉及到最简单情况,在实际应用中常常会有多播特征的网格环境,如图3所示网格资源模型。因此需要研究适应网格多播资源模型的高效资源查找算法。下面研究针对该模型的带度约束的多播资源查找算法。

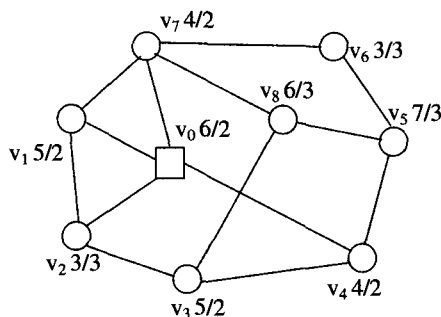


图3 网格的资源分布和节点图

图3中, M/N 数字中 M 表示节点相互注册的资源提供能力的权值, N 表示度约束值。

3.2 带度约束的资源查找多播树构造算法

在图3的资源模型中,根据各个网格节点提供资源的能力赋给每个节点资源提供能力大小的权值,即图3中的 M 参数。权值与动态路由协议一样,节点的权值改变后会动态通知它的邻节点。我们用 deg_i 表示节点 i 在多播模型中的邻节点的个数。带度约束的多播资源查找算法目的是在复杂的网格多播环境中高效的查找到源节点需要的资源所在的节点位置。算法执行查询的要求:

(1) 要求查找过程中查找资源询问请求集中在重要资源节点,不能过分扩散。我们研究的算法执行路径形成了以源节点 v_0 为根,以节点权值大小为依据,连接目的节点集合 $V = \{v_1, v_2, \dots, v_m\}$ 的最大可能结果的资源查找多播树 T (multicast tree) (这里,最大可能结果是因为权值大的节点提供需求资源的可能性最大)。

(2) 为防止每个节点盲目向外多播,给每个节点 i 设置各自的度约束 k_i (图3中的 N 参数),节点多播的度约束条件是:节点 i 可以向与其邻接的 $k_i - 1$ 个节点传送信息。由于度约束的限制,算法根据节点提供资源的能力来选择当前节点的哪些邻节点能连接到多播树上去。因为节点还要接收信息,所以每个节点的度约束应大于或等于2。这个度的限定可以依据需要改变,当其为2时表示只能传1个最佳邻节点,到度约束为无穷大时表示当前节点能将查询信息传给所有邻节点。

(3) 为了防止在查询资源的过程中,查询请求形成查询回路,算法在多播资源查找数据报中采用队列方式携带前边所有父节点与子节点相关信息,当前节点在发出多播查找信息时能根据这些信息判断是否会构成回路。

查找算法如下:

设 $Tc(v_i)$ 表示图中所有节点的权值, Q_i 集合表示某个节点的邻节点的集合, 并且将邻节点按处理能力分成等级同时按等级高低排列, S_p 集合表示接收到查询请求的当前节点的父节点与祖先节点产生的多播树的节点集合,

第一步: 对源节点 v_0 , 设数组 $S_0 = v_0$, 设置查询等待时间, 执行步骤二;

第二步: 求出数组 Q_0 中 Tc 值较大且等级不大于 v_0 节点度约束值 $k_0 - 1$ 的所有节点集合 U_0 , 计算 $S_0 = S_0 \cup U_0$, v_0 向 U_0 中所有节点发送连接请求, 并等待集合 U_0 内所有节点的应答, 如果所有子节点都已返回终止信息, 资源查找结束, 返回失败信息; 而 U_0 中接收到请求的节点 v_i 将顺序执行下列步骤;

第三步: 节点 v_i 在收到任何请求后, 首先判断当前请求是否处理过, 通过对请求队列中的请求源节点、资源、及时间比对后确定是否继续处理当前请求。如果请求处理过, 立刻向父节点发出终止信息, 否则转向步骤四;

第四步: 如果查询资源在当前节点内登记, 则向父节点返回成功信息同时向根节点 v_0 直接返回资源所在节点位置信息, 转到步骤六; 否则转到步骤五;

第五步: 对所有与 v_i 相邻的节点集合 Q_i 中的元素, 求出集合 Q_i 中 Tc 值较大且等级不大于 v_i 节点度约束值 $k_i - 1$ 的所有节点集合 U_i , 对所有 $U_i = U_i - (U_i \cap S_p)$, 判断 U_i 是否为空, 如为空则向父节点发出终止信息, 否则计算 $S_i = S_p \cup U_i$, 然后 v_i 向 U_i 中所有节点发送查询请求, 并等待 U_i 所有节点的应答, 如果所有子节点都已返回终止信息, 则当前节点将终止信息返回给父节点。

第六步: 如果有节点返回成功信息或者所有被查询节点返回终止信息, 或者查询等待时间到期, 则算法结束; 否则转第三步。

3.3 算法仿真

算法仿真于图 3 所示网格模型, 产生的多播树如图 4 所示:

虚线表示从 v_0 源节点开始在网格中资源查询多播数树的路径。a 表示没有资源时算法形成的多播树, 图中的叉表示避免回路的两种情况, 一是向父节点或者祖先节点请求, 二是向被其他树分支已经查询过的节点请求。b 假设度约束改变、资源在节点 v_5 上的执行情况。从 v_0 出发根据 v_0 的度约束, 将请求发给 v_1 、 v_7 、 v_4 节点, v_1 发到 v_2 点, v_7 发到 v_8 、 v_6 , v_4 查询到 v_5 , 由于资源在 v_5 上, 由 v_5 直接返回给 v_0 表示查询成功, 同时 v_5 也向父节点 v_4 反馈查询成功信息。

结论 下面对几种算法就时间、空间复杂度进行综合分析:

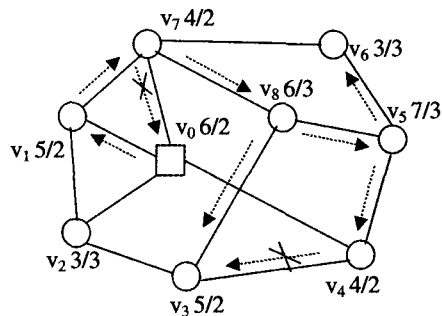
(1) 时间复杂度有: $O(n^2)$ 算法 $1 \geq O(|W_{ps}| + |W_{max}|)$ 算法 $2 \geq O(W_{max})$ 算法 3, 因此从时间复杂度来看有: 算法 $1 \geq$ 算法 $2 \geq$ 算法 3。

(2) 空间复杂度有:

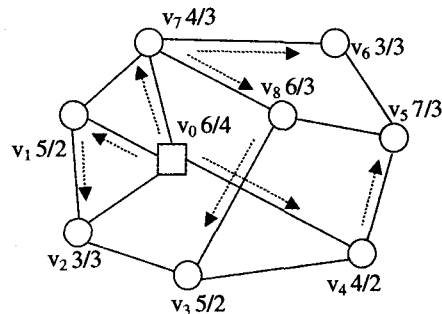
$O(n \cdot W_{max})$ 算法 $1 = O(n \cdot |R|)$ 算法 $3 \geq O(|R|)$ 算法 2 , $n > 1$ 且 $n = |R|$, $O(n \cdot |R|)$ 算法 $3 \geq O(n \cdot W_{max})$ 算法 $1 \geq O(|R|)$ 算法 2 , $n > 1$ 且 $|R| > n$

可以看出通过路由算法查找花费的时间最短, 而空间耗费取决于资源个数和节点的个数。当资源数和节点数相等时, 算法 3 的空间耗费和算法 1 相等, 算法 2 的空间消耗最小。因为算法 2 的资源和节点间对应关系只在一个节点上, 而算法 1 和算法 3 的相关信息都在各资源节点上有冗余, 但是随着节点处理能力的提高, 人们为追求更快的访问速度, 牺

牲一点存储空间也是愿意的, 所以基于路由的资源查找算法将是未来发展的方向。



(a) 无资源且防止回路的多播树



(b) 资源在 v_5 上且度约束改变后的多播树

图 4 算法仿真图

而本文提出的带度约束的多播资源查找算法, 是路由算法的进一步改进。和其他算法比较有三点优势, 一是本算法实现类似于最短路算法的标号法实现, 为此算法复杂度为: $O(|E| + |V| \log |V|)$, $|E|$ 为边的数目, $|V|$ 为节点的数目。对于利用局部信息算法的性能, 文[8]给出了一般结论, 即局部算法得到的解在最坏情况下至少为最优解的 $2k/3$ 倍, 这里 k 为目的节点数。二是本算法利用了网格环境中节点的组播能力, 通过设置适当的度约束值能够有效的提高资源的查询效率, 进而提高网络的运算能力。三是由于本算法是一个局部算法, 它不需要源节点知道整个网格的节点信息, 每个节点只须与其相邻的节点进行信息传递, 实现起来很简单。综上所述本算法更有实际价值。

参考文献

- Broash E, Shavitt Y. Approximation and Heuristic Algorithms for Minimum Delay Application-Layer Multicast Trees. In: Proc. of IEEE INFOCOM 2004, March 2004
- Guo L, Matta I. Search Space Reduction in QoS Routing Computer Networks. The International Journal of Computer and Telecommunications Networking, 2003, 41(1): 73~88
- LI V O K, Zhang Zaichen. Internet Multicast Routing and Transport Control Protocols. Proceedings of the IEEE, 2002, 90(3)
- Foster I, Kesselman C, Nick J, Tuecke S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, January 2002
- Foster I, Kesselman C, Tuecke S. The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International J. Supercomputer Applications, 2001, 15(3)
- Czajkowski K, Fitzgerald S, Foster I, Kesselman C. Grid Information Services for Distributed Resource Sharing. In: Proceedings of the Tenth IEEE International Symposium on High Performance Distributed Computing (HPDC-10), August 2001
- Foster I, Kesselman C, Lee C, Lindell R, Nahrstedt K, Roy A. A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation. In: Intl Workshop on Quality of Service, 1999
- Jaffe J M. Distributed multi-destination routing: the constraints of local information. SIAM J. Comp, 1985, 14(4): 875~888