

基于虚拟网节点迁移的虚拟网映射优化算法

郭贺彬

(北京京北职业技术学院 北京 101400)

摘要 为提高虚拟网请求接收率和底层网络资源利用率,提出了基于虚拟网节点迁移的虚拟网映射优化算法,算法能够将资源竞争最大化的虚拟节点划分在一个组,实现虚拟网映射结果全局最优。通过时间复杂度分析和仿真实验验证了提出的算法能够显著降低虚拟网络节点迁移算法的运行时间;通过仿真实验,将提出的算法与 No-Migration 算法和 Long-Duration 算法进行对比,验证了提出的算法有更高的请求接收率、更高的平均收益,可以节省底层网络资源开销。

关键词 网络虚拟化,底层网络,虚拟网络,资源分配,节点迁移

中图分类号 TP393 **文献标识码** A

Virtual Network Mapping Optimization Algorithm Based on Virtual Network Node Migration

GUO He-bin

(Northern Beijing Vocational Education Institute, Beijing 101400, China)

Abstract In order to improve the virtual network request receiving rate and the substrate network resource utilization, this paper presented a virtual network node migration algorithm based on partition. The algorithm can put virtual nodes in a group which have resource competition relationship to achieve global optimization results of virtual network mapping. Through the time complexity analysis and simulation experiment, the proposed algorithm can significantly reduce the running time of the virtual network node migration algorithm. Through experiments, comparisons with No-Migration algorithm and Long Duration algorithm were made. Experiment results show that the proposed algorithm has a higher request receiving rate, a higher average income, and saves the substrate network resources.

Keywords Network virtualization, Substrate network, Virtual network, Resource allocation, Node migration

1 引言

网络虚拟化(Network Virtualization, NV)是在底层物理网络(Substrate Network, SN)和用户之间增加的一个抽象层,该抽象层向下对物理网络资源进行分割,向上提供虚拟网络(Virtual Network, VN)^[1-5],网络虚拟化得到了越来越多国家和研究机构的关注。为了保证 VN 正常运行,提高 SN 资源利用率,虚拟网资源分配已成为当前的研究热点^[6-9]。

文献[6]提出底层网络中的一条虚拟链路可以由多个分割的物理路径组成,采用启发式算法进行映射。文献[7]利用混合整数规划方法来解决虚拟网映射问题。文献[8]提出了考虑均衡的自适应虚拟网构建算法。文献[9]提出了解决路径跳数限制约束的虚拟网映射算法。

上述文献主要研究虚拟网构建过程中如何提高底层网络资源利用率,很少考虑对底层网络上的虚拟资源进行优化。虽然文献[6,7]采用对虚拟请求进行迁移的方法来实现动态环境下的资源优化,但文献[6]对所有虚拟请求整体迁移,时间复杂度较高,导致不能及时处理新到来的虚拟请求。文献[7]对虚拟请求进行单个迁移,只能达到局部最优映射结果。

为了解决上述问题,本文提出基于虚拟网节点迁移的虚拟网映射优化算法,算法能够将资源竞争最大化的虚拟节点划分在一个组,实现虚拟网映射结果全局最优化。实验比较表明,本文提出的算法有更高的请求接收率、更高的平均收益、可以节省网络的平均资源开销,很好地解决了动态环境下虚拟网映射优化问题。

2 算法

2.1 基于虚拟网节点迁移的虚拟网映射优化算法

基于虚拟网节点迁移的虚拟网映射优化算法如算法 1 所示,算法使用网络划分算法(如算法 2 所示)将多个虚拟请求组成的虚拟网络进行划分,划分时保证组内节点间的资源竞争最大化,实现映射结果的全局最优化。划分完毕后,分别对 k 个组进行节点和链路迁移,每组的结点个数为总的结点个除以 k 值,之后对组间的链路进行迁移。

算法 1 基于虚拟网节点迁移的虚拟网映射优化算法

1. If 时间窗口到达时
2. 将 Run_Qur 中的 VN 请求合并成一个网络,放入 Merge_VN
3. 使用网络划分算法将 Merge_VN 划分为 k 组;

本文受北京市职业院校教师培养计划项目(bjms201318)资助。

郭贺彬(1974-),男,硕士,副教授,主要研究方向为云计算与大数据, E-mail: ghbsxy@163.com.

4. For 每个组 i
5. 使用 CNLM 算法进行节点映射;
6. 使用 MCF 算法进行链路映射;
7. If 第 k 组的节点 & 链路映射成功
8. 更新 Merge_VN 中的映射信息;
9. End if
10. $i++$;
11. End for
12. If Rest 的节点映射成功
13. 更新 Merge_VN 中的映射信息;
14. End if
15. If k 个组和 Rest 映射都成功
16. 将原来 Run_Queue 中 VN 请求的映射资源撤销;
17. 给 Merge_VN 分配资源;
18. 更新 Run_Queue 中的 VN 请求的映射信息;
19. 更新性能指标的值;
20. End if
21. End if

虚拟请求队列 VN_Queue 中存储每一个到来的 VN 请求及其映射信息。Run_Queue 中存储当前网络中正在运行的 VN 请求编号和离开时间。Merge_VN 是一个虚拟网络,存储的是将运行队列中的 VN 请求整合后的网络。CNLM^[7]综合考虑了物理节点的 CPU 资源和相关的带宽资源,是一种能使资源被较优化利用的节点映射算法。MCF^[7]解决了多个 VN 链路请求同时被考虑前提下的链路映射问题。Rest 表示划分后组与组之间的链路的集合。

2.2 网络划分算法

使用网络划分算法划分后的虚拟网络可以抽象为: $G_i^v = (N_i^v, L_i^v, Rest)$, G_i^v 表示划分后的第 i 组, $1 \leq i \leq k$ 。 k 为设定好的一个常量,表示将网络划分成 k 组。 N_i^v, L_i^v 分别表示第 i 组的节点和链路的集合。 w_{mn} 中存储 m 与 n 两个虚拟节点之间的边权值大小,权值越小表示竞争性越大,越可能被划分在同一组内。 $P(n^v)$ 表示 n^v 是否被划分过,0 表示未被划分过,1 表示被划分过。网络划分算法过程如算法 2 所示, $H(n)$ 表示资源拥有量, $n \in \Omega(m)$; h_m 表示排序后 m 节点的第 i 个物理节点的编号; s_{mnum} 和 v_{mnum} 分别表示物理网络和虚拟请求的节点的个数, s_{bnum} 和 v_{bnum} 分别表示物理网络和虚拟请求的链路的个数。 $\text{Min} \sum_{n_i^v \in N_i^v} w_{n_i^v, n_j^v}$ 表示组内的所有节点之间的权值之和最小。如果合并之后大的虚拟网络 Merge_VN 中的结点个数为 k 的值大,则不执行划分算法。

网络划分算法首先计算每个虚拟节点 m 可被映射的物理节点的 H 值,并按照 H 值从大到小排序后存储在 h_m 中。之后对比任意两个虚拟节点 m 和 n 的 h_m 和 h_n 中的数据项,若找到相等的数据项 h_m 和 h_n ,则将 w_{mn} 赋值为 i 与 j 的和。找到每两个虚拟节点的边权值 w_{mn} 之后,开始将节点划分到 k 个组内。若节点 m 能使第 i 组边权值的和最小,则将节点 m 加入第 i 组内,循环上面的做法,直到所有节点都被划分到某一个组内。

算法 2 网络划分算法

1. $\forall m \in N^v$, 将 $\Omega(m)$ 中的节点按 $H(n)$ 降序排列存储在 $\forall h_m$ 中;
2. $\forall m, n \in N^v$, While $i < v_{mnum}$
3. While $j < v_{nnum}$
4. If $h_{mi} = h_{nj}$
5. $w_{mn} \leftarrow i + j$
6. End if
7. 对 $\forall n^v \in N^v, P(n^v) \leftarrow 0$;
8. 将 $n^v \in N^v (P(n^v) = 0)$ 加入 N_i^v 中,且 $P(n^v) \leftarrow 1$;
9. 求使得 $\text{Min} \sum_{n_i^v \in N_i^v} w_{n_i^v, n_j^v}$ 的 $n_i^v \notin N_i^v (P(n_i^v) = 0)$, 加入 N_i^v 中,且 $P(n^v) \leftarrow 1$;
10. 若 N_i^v 中结点个数为 v_{mnum}/k , 则跳转到第 11 步; 否则跳转到第 9 步;
11. 若 k 个组都划分完毕, 则跳转到第 12 步; 否则 $i++$, 并跳转到第 8 步;
12. 对 $l^v \in L^v$, If l^v 的起始点属于 N_i^v
13. If l^v 的终结点属于 N_i^v
14. 将 l^v 加入 L_i^v ;
15. Else
16. 将 l^v 加入集合 Rest;
17. Else
18. 将 l^v 加入集合 Rest;
19. 对 G_i^v 重映射;
20. 对 Rest 重映射;

2.3 时间复杂度分析

2.3.1 整体迁移算法时间复杂度

整体迁移算法时间复杂度为节点迁移和链路迁移的时间开销之和,节点迁移时间开销为: $T(\frac{(n+m)^2}{n(n-1)/2+1})$, 链路迁移时间开销为: $T(\frac{(n+m)^2}{n(n-1)/2})$, 所以整体迁移总的时间开销为: $(\frac{(n+m)^2}{n(n-1)/2+1})^a + (\frac{(n+m)^2}{n(n-1)/2})^a$, n 表示虚拟网络节点个数, m 表示物理网络节点个数。

2.3.2 划分算法时间复杂度

节点和链路映射使用线性规划中的单纯形法。单纯形法时间复杂度平均情况下为多项式,最坏情况下为指数。本文计算时设其复杂度为: $O(N^a)$, N 为线性规划中变量的个数, a 为大于等于 1 的常数。用 $T(n)$ 来表示划分算法的时间复杂度, n 为虚拟网络的结点个数为, 则有:

$$T(n) = \sum_{i=1}^{n/k} [i(n-i) + i(n - \frac{n}{k} - i) + \dots + i(\frac{2n}{k} - i)] + O(n^4)$$

$$= \sum_{i=1}^{n/k} [\frac{i(k-1)(n + \frac{2n}{k})}{2} - (k-1)i^2] + O(n^4)$$

$$= O(n^3) + O(n^4)$$

上述计算表明,划分算法的时间复杂度为 $O(n^4)$, n 表示虚拟网络节点个数。

2.3.3 先划分后迁移算法的时间复杂度

先划分后迁移算法的时间复杂度为划分算法时间复杂度、 k 组节点迁移时间开销、 $(k+1)$ 组链路迁移的时间开销之

和。其中, 每个小组节点迁移的时间开销: $T(\frac{n/k+m}{n(n/k-1)/2k+1})$, 每个小组的链路迁移的时间开销为: $T(\frac{n/(k+1)+m}{n(n/(k+1)-1)/2(k+1)})$, 所以划分后再迁移的总的的时间开销可以表示为: $O(n^4) + k(\frac{n/k+m}{n(n/k-1)/2k+1})^a + (k+1)(\frac{n/(k+1)+m}{n(n/(k+1)-1)/2(k+1)})^a$ 。

2.3.4 算法时间复杂度对比

设 $f(k) = \text{整体迁移算法时间} / \text{先划分后迁移算法时间}$, 当 $O(n^a)$ 中的 $a \geq 2$ 时, 对 k 求导可得出当 $k > 1$ 时, $f(k)$ 是单调上升的, 即有 $f(2) \leq f(k) \leq f(n)$; 又有 $f(2) \in [12.3, 2^{4a}]$, $f(n) \in [n^{2a-1}, (n+1)^{4a-1}]$, 则得到 $f(k) \in [12.3, (n+1)^{4a-1}]$; 当 $O(n^a)$ 中的 $a=1$ 时, 对 k 求导可得出当 $k > 1$ 时, $f(k)$ 是单调上升的, 即有 $f(2) \leq f(k) \leq f(n)$, 又有 $f(2) \in (0.48, 3.8]$,

$$f(n) \in [\frac{2}{4 + \frac{1}{n^3} + \frac{1}{(n+1)^3}},$$

$$\frac{16n^5 + 52n^4 + 52n^3 + 8n^2 - 8n + 12n + 4}{64n^5 + 160n^4 + 72n^3 - 32n^2 - 32n + 8}]$$

从上述 $f(k)$ 的取值可以分析出, 整体迁移的时间复杂度, 只有在最好情况下即 $a=1$ 时, 才有可能比先划分后迁移算法好, 而且此时 $f(2)$ 的取值小于 4, $f(n)$ 的取值也是远小于 n 的, 两个算法时间上的差别很小。在绝大部分情况下, 整体迁移算法所用的时间最少是先划分后迁移算法的 12.3 倍, 最多能够达到 $(n+1)^{4a-1}$ 倍, 所以, 划分后迁移算法在时间上的优势很明显。

3 实验数据分析

本文使用 GT-ITM 网络拓扑生成工具来生成物理网络和虚拟网络请求^[10]。本文所生成的物理网络拥有 100 个节点, 物理节点是在 100×100 的坐标范围内随机确定的, 每两个物理节点之间按照 0.5 的概率进行随机的连接来生成链路, 物理节点的 CPU 资源数和物理链路的带宽资源数都在 50~100 的范围内随机确定。

所生成的虚拟请求的到达时间按照泊松分布进行, 每两个相邻的虚拟请求的到达时间平均间隔有 1.5 个时间单位, 每个虚拟请求的平均生存周期为 20 个时间单位。每个虚拟请求的节点个数在 2~10 之间随机生成, 虚拟节点的坐标在 100×100 的范围内随机确定, 每两个虚拟节点之间按 0.5 的概率进行随机链接来生成链路, 虚拟请求节点 CPU 资源数在 1~20 范围内随机确定, 虚拟链路的带宽资源数在 1~50 的范围内随机确定。在算法的实现中, 使用线性规划库 glpk 来解决节点和链路的映射问题。

3.1 时间开销分析

用网络生成工具生成若干虚拟请求, 算法每运行一次增加一个虚拟请求。运行结果如图 1 和图 2 所示。两图中, 横轴表示当前正在运行的所有虚拟请求的节点个数的总和, 纵

轴表示节点迁移算法的运行时间。

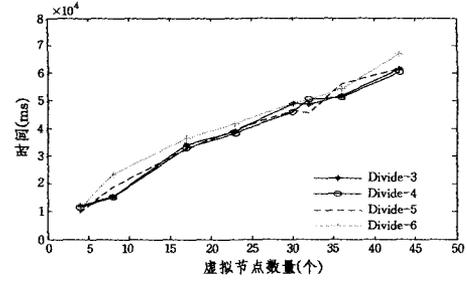


图 1 划分组数对时间开销的影响

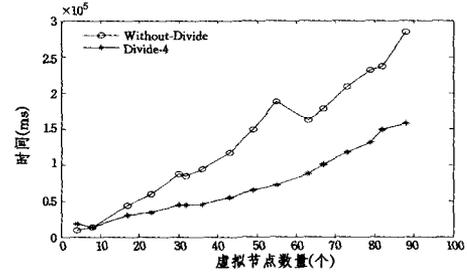


图 2 时间开销比较

如图 1 所示, divide-3, divide-4, divide-5, divide-6 分别表示划分组数 k 为 3, 4, 5, 6 时先划分后迁移的算法。从运行的结果图 1 中可以看出划分组数 k 的值在 3~6 之间, 总的虚拟请求节点数在 0~50 之间变化时, 先划分后迁移的算法的时间复杂度没有明显变化。

图 2 显示了未划分算法 Without-Divide 和 Divide-4 的时间开销对比。从图中可以看出, 在划分组数 k 值为 4, 总的虚拟请求节点个数在 0~100 之间变化时, 先划分后迁移的算法与整体迁移不划分算法相比有明显优势, 需要迁移的节点总数越多, 越能够突显出划分后再迁移的算法在时间上的优势。

3.2 性能指标分析

图 1 所示为时间开销的走势, 在实际运行过程中, 整体迁移算法的时间开销是惊人的。因此选择虚拟网络的一部分虚拟请求进行节点迁移, 即对运行时间较长的虚拟请求进行迁移。本节将对网络先划分后迁移的算法与不对网络节点进行迁移的算法、对运行时间较长的虚拟请求进行迁移的算法进行对比。本文使用 Long-Duration^[6] 表示对已经运行较长时间(超过 15 个时间单位)的虚拟请求进行迁移的算法; No-Migration^[7] 表示不对网络节点进行迁移的算法; Divide 表示对虚拟网络先划分后迁移的算法。程序运行 150 个时间单位, 运行效果如图 3~图 7 所示。

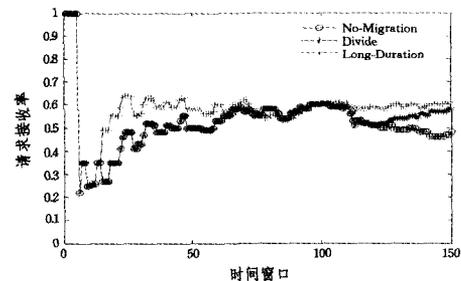


图 3 请求接收率

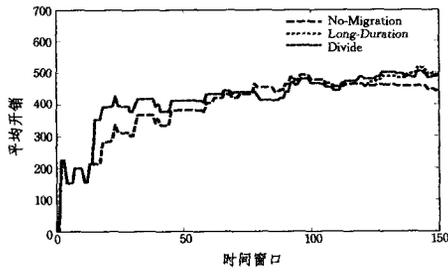


图4 平均资源开销

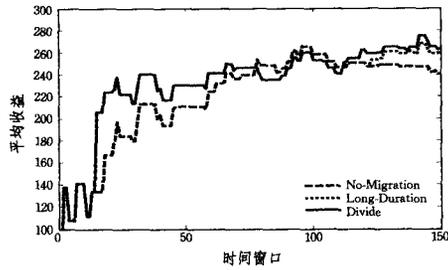


图5 平均收益

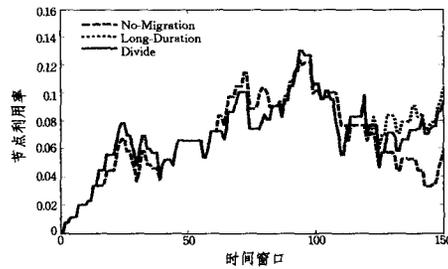


图6 节点利用率

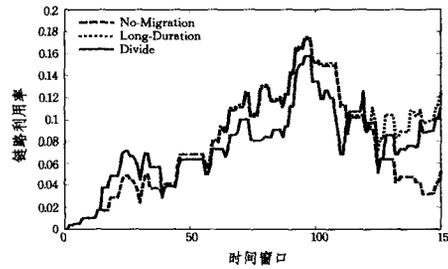


图7 链路利用率

从性能指标的对比图中可以总结出以下几点:

Divide算法有更高的请求接收率和更高的平均收益。图3和图5分别表示请求接收率和平均收益的对比图。Divide比No-Migration和Long-Duration有更高的接收率。因为接收率提高,更多的虚拟网被用一类分配资源,平均收益提高。

Divide算法可以节省底层网络资源开销。图4为平均资源开销的对比图。虽然Divide算法的资源开销比No-Migration和Long-Duration算法高,但是图3中Divide算法的请求接收率比另外两个算法高的幅度更大,而图4中Divide算法的平均资源开销比另两个算法高的幅度更小,这个现象说明

了Divide算法可以将网络映射到较优的资源,可以节省网络的资源开销。

Divide算法节约了底层网络资源。图6和图7为节点利用率和链路利用率的对比图。Divide的节点利用率与其他两个算法相比稍微高一点,这是因为其请求接收率也较高,而Divide的链路利用率比其他算法要低,这从另一方面说明了Divide算法节省资源开销的主要原因是它减少了链路利用率。

结束语 网络虚拟化是未来互联网体系结构的主要组成部分,资源分配作为一个研究关键点,有效的资源分配能提高物理资源利用率,降低服务成本。本文对虚拟网节点迁移算法进行了研究,提出了基于虚拟网节点迁移的虚拟网映射优化算法。通过与相关算法进行实验比较,验证了本文提出的算法有更高的请求接收率和平均收益,可以节省底层网络资源开销,节约底层网络资源。

参考文献

- [1] Papadimitriou P, Houidi I, Louati W. Towards large-scale network virtualization. Springer[J]. *Wired/Wireless Internet Communication, Lecture Notes in Computer Science*, 2012, 727(7): 13-25
- [2] Dong Y, Yang X, Li J, et al. High performance network virtualization with SR-IOV[J]. *Elsevier Communication Architectures for Scalable Systems*, 2012, 72(11): 1471-1480
- [3] Dmitry D, Digital E, Eric K, et al. Scalable Network Virtualization in Software-Defined Networks[J]. *IEEE Internet Computing*, 2013, 17(2): 20-27
- [4] Cheng X, Su S, Zhang Z, et al. Virtual network embedding through topology-aware node ranking [J]. *ACM SIGCOMM Computer Communication Review*, 2011, 41(2): 39-47
- [5] Cheng X, Su S, Zhang Z, et al. Virtual network embedding through topology awareness and optimization [J]. *Computer Networks*, 2012, 56(6): 1797-1813
- [6] Yu M, Yi Y, Rexford J, et al. Rethinking virtual network embedding: Substrate support for path splitting and migration [J]. *ACM SIGCOMM CCR*, 2008, 38(2): 17-29
- [7] Chowdhury N, Rahman M, Boutaba R. ViNEYard: virtual network embedding algorithms with coordinated node and link mapping[J]. *IEEE/ACM Transactions on Networking*, 2012, 20(1): 206-219
- [8] 齐宁,王保进,汪斌强,等. 均衡虚拟网构建算法研究[J]. *电子与信息学报*, 2011, 33(6): 1301-1306
- [9] 姜明,王保进,吴春明,等. 网络虚拟化与虚拟网映射算法研究[J]. *电子学报*, 2011, 39(6): 1315-1320
- [10] Zegura E W, Calvert K L, Bhattacharjee S. How to model an internetwork[C]//*Proc. IEEE INFOCOM*, 1996