软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2014,25(10):2189−2205 [doi: 10.13328/j.cnki.jos.004680]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于知识描述和遗传算法的跨域虚拟网络映射*

肖蔼玲[1], 王 颖[1], 孟洛明[1], 邱雪松[1], 李文璟[1], 尹 斌[2]

[1](网络与交换技术国家重点实验室(北京邮电大学),北京 100876)

[2](中国西安卫星测控中心 喀什卫星测控站,新疆 喀什 844000)

通讯作者: 肖蔼玲, E-mail: xiao_ailing@bupt.edu.cn

摘 要: 网络虚拟化环境下的跨域虚拟网络映射是指当物理网络由多个自治域构成时,以最小化虚拟网络映射开销为目标,将虚拟网络请求恰当地划分为多个虚拟子网请求,并分别指派给相应自治域以完成映射.资源匹配和虚拟网络划分是跨域虚拟网络映射中的两个关键阶段.然而,现有的资源匹配算法无法支持精确的数值属性匹配,也无法满足虚拟网络用户对表达多样化映射约束的需求,故实用性不高.此外,虚拟网络划分属于 NP 问题,目前也缺乏高效的求解方法.针对上述两个阶段中存在的问题,分别提出了基于 OWL 及 SWRL 的资源匹配算法和基于遗传算法的虚拟网络划分算法.理论分析证明了该方法的正确性.仿真实验从效率、性能及稳定性方面验证了该方法的有效性.

关键词: 网络虚拟化;跨域虚拟网络映射;资源匹配;知识描述;虚拟网络划分;遗传算法

中图法分类号: TP393

中文引用格式: 肖蔼玲,王颖,孟洛明,邱雪松,李文璟,尹斌.基于知识描述和遗传算法的跨域虚拟网络映射.软件学报,2014,25(10):2189−2205. http://www.jos.org.cn/1000-9825/4680.htm

英文引用格式: Xiao AL, Wang Y, Meng LM, Qiu XS, Li WJ, Yin B. Knowledge description and genetic algorithm based multi-domain virtual network embedding. Ruan Jian Xue Bao/Journal of Software, 2014,25(10):2189−2205 (in Chinese). http://www.jos.org.cn/1000-9825/4680.htm

Knowledge Description and Genetic Algorithm Based Multi-Domain Virtual Network Embedding

XIAO Ai-Ling[1], WANG Ying[1], MENG Luo-Ming[1], QIU Xue-Song[1], LI Wen-Jing[1], YIN Bin[2]

[1](State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications), Beijing 100876, China)

[2](China Xi'an Satellite Control Center, Kashi Satellite Station, Kashi 844000, China)

Corresponding author: XIAO Ai-Ling, E-mail: xiao_ailing@bupt.edu.cn

Abstract: In a network virtualization environment, multi-domain virtual network embedding (MVNE) deals with properly partitioning a virtual network (VN) request into sub-VN requests across multiple domains for embedding with the aim of minimizing the embedding cost. Resource matching and VN partitioning are two key phases in MVNE. However, a resource matching algorithm capable of providing accurate numerical attribute matching and VN users' diversified mapping constraints specification has not been explicitly studied by the existing research. Moreover, an efficient solution is needed for MVNE which is NP-hard. This paper provides an OWL and SWRL based resource matching algorithm and a genetic algorithm based VN partitioning algorithm to tackle the problems mentioned above. The correctness of the presented method is proved by theoretical analysis, and the validity of the method is assessed by simulation experiments in terms of efficiency, performance, and stability.

Key words: network virtualization; multi-domain virtual network embedding; resource matching; knowledge description; virtual network partitioning; genetic algorithm

* 基金项目: 国家高技术研究发展计划(863)(2013AA013502); 中央高校基本科研业务费专项资金(BUPT-2013RC1103); 国家科技支撑计划(2012BAH35F02)

收稿时间: 2014-02-28; 修改时间: 2014-07-07; 定稿时间: 2014-07-31

© 中国科学院软件研究所 http://www.jos.org.cn

网络虚拟化技术被视为解决网络僵化问题的重要途径[1,2],该技术支持多个虚拟网络(virtual network,简称VN)彼此隔离地共享物理网络资源.利用网络虚拟化技术,服务提供商(service provider,简称SP)可以通过向虚拟网络用户(VN user)提供多样化的网络服务来获得收益,基础设施提供商(infrastructure provider,简称InP)可以通过提高自身的资源利用率来减少运营成本和设备开销.

虚拟网络映射是网络虚拟化的关键,其目标是在物理网络中为VN的虚拟节点和虚拟链路分别找到满足其映射约束的物理节点和物理路径.目前,研究人员已经对单一自治域(单域)场景下的VN映射进行了诸多有益的探索[3−9],即,在获悉单个自治域内全部物理网络信息的前提下提供 VN 的映射方案.然而在实际中,由于客观因素的限制(如资源类型、价格等),往往需要多个InP的共同合作来完成VN映射.由于各InP不会公开其内部网络的详细信息,故单域映射方法不适用于VN的跨域部署,于是,学者们开始研究跨域映射问题[10−19].

已有的跨域VN映射方法可以分为两类:分布式和集中式.

- 分布式的跨域VN映射[13,14]通过SP与InP及InP与InP间的资源协商来实现,能够充分尊重SP和InP双方的意愿且有良好的可扩展性.但分布式方法在协商过程中会引起额外的网络传输代价,又由于缺乏对全局信息的掌握,故并不能得到跨域VN映射的最优解;
- 而集中式方法在SP和InP间引入虚拟网络提供商(virtual network provider,简称VNP)[20],简化了SP与InP的供需匹配过程,并以最小化映射开销为目标为VN求解跨域映射方案.

集中式跨域VN映射包括如下3个阶段:

1) 资源匹配阶段,在收到SP发来的VN请求后,VNP将请求中对虚拟资源的映射约束与各InP公开的物理网络信息进行匹配,得到每个虚拟资源的匹配集合;
2) VN划分阶段,VNP以最小化映射开销为目标,为每个虚拟资源从其匹配集合中选择最适合的映射目标,选择结果会将VN划分为多个虚拟子网,并确定映射每个虚拟子网的InP,至此,跨域VN映射已化归为多个单一自治域场景下的VN映射;
3) 虚拟子网映射阶段,VNP将各虚拟子网请求转发给相应的InP,各InP可采用已有的单域映射方法[3−9]求解虚拟子网映射.

然而对于跨域映射的前两个阶段,目前的研究还存在一些不足:

- 基于概念群集技术,已有的资源匹配算法[10−12]按相似性对物理网络可提供的虚拟资源进行分类,生成概念分类树,再利用概念分类树完成资源匹配.虽然这类方法的资源匹配效率较高,但却限制了 VN user 对映射约束的表达形式,使映射约束只能描述为合取式,而不能表达蕴含、非等较为复杂的语义.另一方面,数值属性在概念群集技术中是按取值区间分类的[21],容易导致资源匹配结果不精确,影响后续VN划分阶段的效率和性能.因此,如何对VN user多样化映射约束的表达及精确数值属性的匹配提供支持,是资源匹配阶段需要解决的问题;
- 对于VN划分问题,Houidi等人[15]第一次从理论上证明了该问题是NP难的,并基于边界节点间全连接的物理网络拓扑提出了 VN 划分的确切式和启发式算法.Dietrich 等人[17]则以 InP 公开边界节点的相关信息为前提,给出了一种确切式 VN 划分算法.与 Houidi 等人的方法相比,该方法虽然建立了更加准确的问题模型且有效减少了跨域VN映射的开销,但却增加了求解VN划分的时间复杂度,使该方法很难完成对较大规模 VN 划分的计算.因此,如何高效地对 VN 划分进行近似求解,是一个亟待解决的问题.

本文针对上述集中式跨域 VN 映射的资源匹配和 VN 划分两个阶段中存在的问题,分别提供了解决方案.本文的贡献主要包括:

1) 基于OWL知识描述和SWRL查询工具设计了一种资源匹配算法,该算法可支持VN user多样化的映射约束表达以及精确的数值属性匹配,提高了资源匹配过程的实用性;
2) 提出了一种基于遗传算法的VN划分算法.该算法将VN划分方案以矩阵形式编码,从多个初始解开始进行迭代搜索,逐步逼近全局最优解;

© 中国科学院软件研究所 http://www.jos.org.cn

3) 设计并实现了本文 VN 划分算法的仿真实验,从效率、性能和稳定性这 3 个方面验证了该算法的优势,并总结了算法中参数的设置原则.

本文第 1 节对跨域 VN 映射中的资源匹配和 VN 划分问题进行数学描述.第 2 节设计资源描述知识库和资源匹配算法.第 3 节提出基于遗传算法的 VN 划分算法.第 4 节通过仿真实验验证本文给出的 VN 划分算法的有效性.第 5 节和第 6 节讨论相关工作并总结全文.

1 问题描述

VNP 是跨域 VN 映射的核心角色,其任务是根据 VN 请求中的映射约束和各 InP 公开的物理网络信息,将 VN 划分为多个虚拟子网并指派给相应 InP 完成映射,故资源匹配和 VN 划分是 VNP 的工作重点.本节首先介绍跨域 VN 映射中 VNP 可获取的物理网络信息,在此基础上给出资源匹配和 VN 划分的数学描述.

1.1 VNP可获取的物理网络信息

VNP 对物理网络信息(substrate network information,简称 SNI)的可见度是其进行资源匹配和 VN 划分的重要依据.实际上,只有得到全部的物理网络信息,VNP 才能计算出最优的 VN 划分方案.考虑到 InP 不可能公开其全部物理网络信息(网络拓扑及资源),与 Dietrich 等人[17]相同,本文参考现有云计算平台(如 Amazon EC2)中 InP 公开计算资源信息的方式,假设在网络虚拟化环境下,InP 会将其物理资源抽象为多种虚拟资源类型.虚拟资源的属性用二元组(属性名,属性值)表示.所有属性取值都相同的虚拟资源被分为一类,例如,将有相同节点功能、操作系统、虚拟化环境及地理区域的虚拟节点资源视为同一类型的虚拟节点.如图 1 所示,InP 向 VNP 公开的是其所在自治域可提供的虚拟资源类型及相应资源类型的价格信息,该信息存放于资源描述知识库(resource description knowledge base)中,用于对资源匹配算法(matching algorithm)及 VN 划分算法(partitioning algorithm)提供支持.此外,VNP 还可以从互联网交换点[22]及 peering 数据库[23]公开的信息中获取边界节点间的拓扑信息及连接开销信息(P_node information),以支持 VN 划分算法.

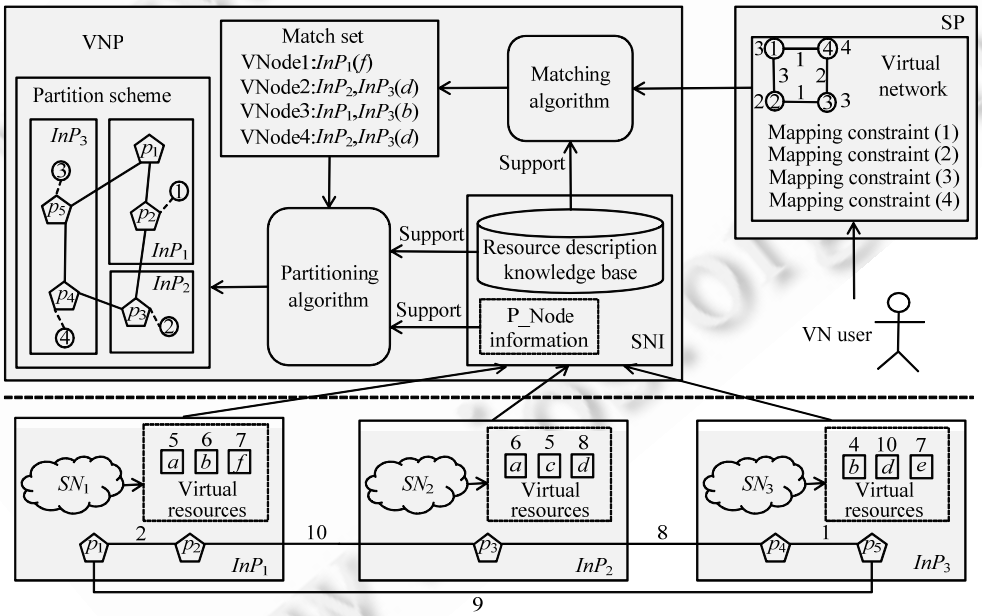


Fig.1 VNP's working mechanism in multi-domain VN embedding

图 1 跨域 VN 映射中 VNP 的工作机制

需要特别说明的是:在资源匹配算法中,对虚拟节点的匹配,即找到满足其映射约束的虚拟节点类型;对虚

© 中国科学院软件研究所 http://www.jos.org.cn

拟链路的匹配,则是要找到一条物理路径,使该路径上的每条物理链路都能满足该虚拟链路的映射约束.考虑到在不能掌握全部物理网络信息的条件下,VNP 无法对虚拟链路进行有效的匹配,因此,本文的资源匹配主要是对虚拟节点匹配;对虚拟链路的匹配是在生成 VN 划分方案后,由各虚拟子网的指定 InP 负责完成.故资源描述知识库中的虚拟资源类型只包括虚拟节点类型,并不涉及虚拟链路类型.

图 1 所示中虚线以下的部分展示了 VNP 可获取的物理网络信息,可以看出,整个物理网络由 3 个自治域及其相互连接构成.各域内的正方形代表该域可提供的虚拟节点类型,以不同字母区分,正方形上的数字代表该虚拟节点类型的单价.各域内的正五边形代表该域的边界节点,正五边形间的连线表示边界节点间的连接关系,连线上的数字代表相应边界节点间建立连接的单位开销.一个 InP 可提供多种虚拟节点类型,一种虚拟节点类型可由多个 InP 提供,如 InP_1 可提供的虚拟节点类型有 a,b,f,其中,a 还可由 InP_2 提供.对于同一虚拟节点类型,各 InP 可有不同报价,如对虚拟节点类型 a,InP_1 和 InP_2 的单价分别是 5 和 6.同域的不同边界节点有不同的域间连接和连接开销,如 InP_1 的 p_1 可与 InP_3 连接且单价为 9,p_2 可与 InP_2 连接且单价为 10.

1.2 资源匹配的问题描述

如图 1 所示,资源匹配是指 VNP 根据 VN 请求中对虚拟节点的映射约束(mapping constraint)及资源描述知识库中的信息,通过一定的资源匹配算法得到每个虚拟节点的可映射范围,即,可提供满足映射约束的虚拟节点类型的 InP 集合,称该集合为匹配集(match set).

将各 InP 能够提供的所有虚拟节点类型记为全集 U,虚拟节点类型的全部属性记为集合 A,其中的每种属性 $A_i\in A$ 可以有不同的取值 A_{ij},其取值集合记为 $V(A_i)=\{A_{i1},A_{i2},\ldots\}$.虚拟节点类型 $T\in U$ 的属性 A_i 的取值记为 $Value(T,A_i)$.根据 A_i 取值的不同对 U 进行划分,可以得到多个子集 U_{ij}:

$$U_{ij}=\{T|Value(T,A_i)=A_{ij},T\in U\}.$$

虚拟节点 v 的映射约束 mc_v 可表达为一个由非、合取、析取、蕴含及等价符号连接的逻辑表达式,其中的每个原子式是虚拟资源类型的一种属性,故由上式可将每个原子式对应到一个 U_{ij}.考虑到逻辑运算中的交、并、补分别对应集合运算中的合取、析取、非,而蕴含和等价则可通过非、合取、析取的转换得到,故可将 mc_v 的逻辑表达式看作对多个 U_{ij} 进行一系列的交、并、补,其运算结果 U_v 中的每个元素都满足 mc_v:

$$U_v=\{T|satisfy(T,mc_v),T\in U\}.$$

将物理网络中全部 InP 组成的集合记为 INP,$InP_i\in INP$ 可提供的全部虚拟节点类型组成的集合记为 $PVT(InP_i)$,则虚拟节点 v 的匹配集可表示为

$$MatchSet(v)=\{InP_i|PVT(InP_i)\cap U_v\neq\varnothing,InP_i\in INP\}.$$

1.3 虚拟网络划分的问题描述

如图 1 所示,VN 划分是指 VNP 以降低跨域 VN 映射的开销为目标,根据资源匹配得到的匹配集、资源描述知识库中虚拟资源类型的价格信息以及边界节点的相关信息,将 VN 请求划分为多个虚拟子网,形成 VN 的划分方案(partition scheme).

跨域 VN 映射的开销($cost$)由 3 部分组成:节点映射开销($NODE_{cost}$)、域间链路映射开销($LINK_{cost}$)和域内链路映射开销.其中,域内链路映射开销依赖于 InP 内部具体的网络信息.由于在实际中 InP 不会完全公开其网络信息,且域内链路映射开销远小于域间链路映射开销[17],故本文在计算跨域 VN 映射开销时忽略了域内链路开销,即:$Cost=NODE_{cost}+LINK_{cost}$.

由第 1.1 节可知:当一个自治域中的虚拟节点与另一个自治域中的虚拟节点间需要建立虚拟链路时,选择不同的边界节点会产生不同的 $LINK_{cost}$.因此,VN 划分方案中不仅要为每个虚拟节点指明承担其映射的 InP,还要指明通过相应 InP 中的哪个边界节点完成域间连接.本文将虚拟节点 v 的匹配集 $MatchSet(v)$扩展为由该集合中的 InP 包含的所有边界节点组成的集合 $MatchSet'(v)$.需要说明的是,边界节点并不承担具体的虚拟节点映射.本文在提到将一个虚拟节点映射到一个边界节点上时,指的是在 VN 划分阶段,将该虚拟节点划分到该边界节点所在的 InP,后续虚拟节点映射由该 InP 完成,且相关虚拟链路映射要使用该边界节点与其他 InP 建立连接.

© 中国科学院软件研究所 http://www.jos.org.cn

将边界节点组成的物理网络抽象表示为一个有权无向图,记为 $G^S=(N^S,E^S)$,其中,N^S 为边界节点集合,E^S 为边界节点间的物理链路组成的集合.对连接边界节点 u,v 的物理链路 $l^S(u,v)\in E^S$,其单位开销用 $cost(l^S)$表示.同样地,将虚拟网络请求表示为一个有权无向图,记为 $G^V=(N^V,E^V)$,其中,N^V 为虚拟节点集合,E^V 为虚拟链路集合.虚拟节点 $n^v\in N^V$ 的节点能力约束用 $c(n^v)$表示.对连接虚拟节点 m,n 的虚拟链路 $l^v(m,n)\in E^V$,其带宽约束用 $bw(l^v)$表示.VN 划分可理解为以最小化跨域 VN 映射开销为目标,找到以下两个映射:

$$f_n: n^v \to MatchSet'(n^v), n^v \in N^V,$$

$$f_l: l^v(i,j) \to Path(i',j'), l^v(i,j) \in E^V, i' = f_n(i), j' = f_n(j),$$

其中,$Path(i',j')$示 G^S 中边界节点 i'和 j'之间的无环路径组成的集合.f_n 表示虚拟节点映射,即把虚拟节点映射到其匹配集中的某个边界节点上;f_l 表示虚拟链路映射,即:当 VN 中虚拟链路 $l^v(i,j)$的两个端点 i,j 分别映射到边界节点 i',j'上时,为完成 VN 划分,$l^v(i,j)$必须映射到 $Path(i',j')$中的路径上.

2 资源描述知识库及资源匹配算法

作为 VN 划分的必要准备,资源匹配的结果必须精确,从而为 VN 划分提供可靠依据,以提高 VN 划分的效率和性能.本节首先用 OWL 语言[24]为物理网络提供的虚拟资源创建资源描述知识库,再利用 SWRL[25]查询语言规范 VN user 的映射约束,最后利用 SWRL 查询工具实现资源匹配算法.

2.1 资源描述知识库

由前一节的分析可知:在网络虚拟化环境下,对虚拟资源的描述可由虚拟资源的类型和属性来表达.这是一种典型的“概念-属性”知识描述体系,故可用 OWL 语言为其创建相应的资源描述知识库.本节先给出资源描述知识库的概念模型,包括概念关系(如图 2 所示)和属性层次结构(如图 3 所示),再基于概念模型创建资源描述知识库的具体内容.

如图 2 所示,概念模型中的概念关系可描述为:一个 InP 可提供多种虚拟节点类型(virtual node type),一种虚拟节点类型也可由多个 InP 提供.一种虚拟节点类型可有多个可枚举属性(eAttribute)和多个数值属性(nAttribute).可枚举属性对应可枚举的概念,可被泛化为多个子概念,用于表示具体的可枚举属性;数值属性对应于一个具体数值,采用 Double 表示.图 2 给出了 3 种常见的可枚举概念:节点功能类型(function type)、操作系统类型(OS type)和虚拟化环境类型(VE type).另外,概念模型中还有一个特殊概念称为候选集(candidate set),这是一个临时保存资源匹配结果的变量,一个候选集可包含多个 InP.

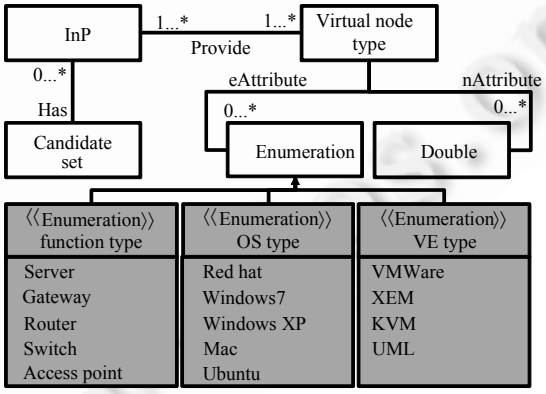


Fig.2 Concept relationship of resource description knowledge base

图 2 资源描述知识库的概念关系

概念模型中的属性层次结构如图 3 所示,其中,可枚举属性(eAttribute)被泛化为 3 种:节点功能类型(functionType)、操作系统类型(osType)和虚拟化环境类型(veType).数值属性(nAttribute)被泛化为 6 种:节点最

© 中国科学院软件研究所 http://www.jos.org.cn

大能力(maxCap,可用主频、内存大小等指标衡量)、节点可靠性(reliability,可用平均故障恢复时间、平均无故障时间等指标衡量)和 4 个用于表达节点地理位置的属性,包括最大经度(maxLon)、最小经度(minLon)、最大纬度(maxLat)和最小纬度(minLat).

图 3 中还给出了每个属性的定义域(domain)和值域(range),具体含义参见文献[24].

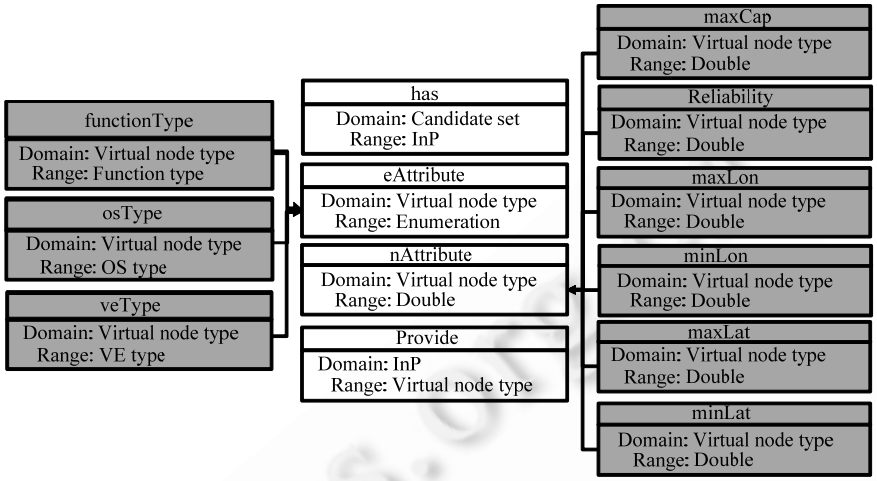


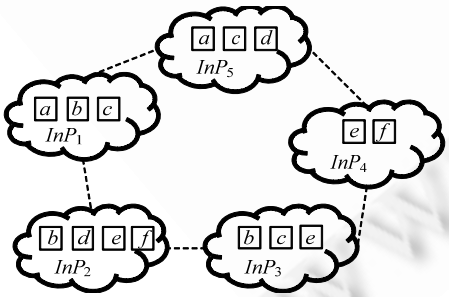
Fig.3 Attribute hierarchical structure of resource description knowledge base

图 3 资源描述知识库的属性层次结构

图 2 和图 3 的灰色部分是可配置的,即,VNP 可根据需要对概念或属性进行添加或删除操作.其中,Enumeration 泛化出的子概念应与 eAttribute 泛化出的子属性一一对应,即,子属性的值域应指向对应的子概念.由于 Double 无需泛化,故由 nAttribute 泛化出的子属性的值域均指向 Double.应该明确的是:概念模型是资源匹配的基础,VN user 要根据概念模型来表达映射约束,InP 要根据概念模型来发布物理网络信息.

进一步地,基于上述概念模型来创建资源描述知识库的具体内容.在跨域 VN 映射中,知识库的创建过程即 InP 将可提供的虚拟节点类型以实例的形式发布到知识库中,并创建各实例间属性关系的过程.对创建好的知识库可进行两类修改:VNP 对概念模型的修改和 InP 对知识的修改.对概念模型的修改必须通知所有 InP 和 VN user,修改代价较大,应尽量避免;对知识的修改仅涉及提出修改的 InP 和 VNP,对其他 InP 和所有 VN user 都是透明的,修改代价较小,InP 可根据自身情况较为自由地修改已发布的虚拟节点类型信息.

图 4 给出了一个资源描述知识库的例子,该知识库包含 1 个可枚举属性和 4 个表示地理位置信息的数值属性.图 4 的左侧部分展示了各 InP 间的连接关系以及各 InP 可提供的虚拟节点类型,右侧部分以表格的形式给出了各虚拟节点类型的属性值.



Attribute / Virtual node type	osType	maxLon	minLon	maxLat	minLat
a	Linux	120	100	60	15
b	Windows	110	75	40	20
c	Linux	130	90	75	50
d	Mac	105	85	65	35
e	Windows	125	105	80	40
f	Linux	115	90	65	25

Fig.4 An example of resource description knowledge base

图 4 一个资源描述知识库的例子

© 中国科学院软件研究所 http://www.jos.org.cn

2.2 映射约束和资源匹配算法

2.2.1 映射约束

若要利用上述资源描述知识库进行资源匹配,必须对 VN user 提出的映射约束加以规范.本节采用 SWRL 查询语言规范 VN user 对映射约束的表达形式.表 1 给出了本节使用的 SWRL 内置函数.

Table 1 Built-In functions of SWRL

表 1 SWRL 的内置函数

函数	含义	取值类型
Concept(?*a*)	*a* 是否为概念“Concept”的实例	布尔值
Attribute(?*a*,?*b*)	*a* 和 *b* 之间是否存在属性“Attribute”	布尔值
Attribute(?*c*)	*c* 的“Attribute”值	数值
greaterThan(d_1,d_2)	d_1 是否大于 d_2	布尔值
lessThan(d_1,d_2)	d_1 是否小于 d_2	布尔值

这里结合图 4 中的资源描述知识库,给出一个用 SWRL 查询语言规范映射约束的例子.VN user 的映射约束为“某虚拟节点 *N* 的操作系统类型是 Mac 或 Linux,且 *N* 的最大经度小于 125,最小纬度大于 20”.为增强该映射约束的可读性,将其分为 3 部分:

- stm_1:*osType*(?*b*,*Mac*)|*osType*(?*b*,*Linux*)
- stm_2:*lessThan*(*maxLon*(?*b*),125)&*greaterThan*(*minLat*(?*b*),20)
- stm_3:*InP*(?*a*)&*VirtualNodeType*(?*b*)&*provide*(?*a*,?*b*)&(stm_1&stm_2)=>*Candidate Set*(?*a*)

其中,stm_1 描述了 *N* 对操作系统类型的约束,stm_2 描述了 *N* 对地理位置的约束.将 stm_1 和 stm_2 带入 stm_3 可以得到 *N* 的完整映射约束.stm_3 可直观地理解为:若 *a* 是一个 InP,*b* 是 *N* 请求的一种虚拟节点类型,且 *a* 可以提供 *b*,同时 *b* 能够满足 stm_1 和 stm_2 的约束,则 *a* 属于 *N* 的候选集 *Candidate Set*(*N*).调用 SWRL 查询工具可得到最终的匹配结果:*Candidate Set*(*N*)={InP_2,InP_4,InP_5}.

2.2.2 资源匹配算法流程

如图 5 所示,资源匹配算法的输入是 VN 请求中指定的一组映射约束,每个映射约束对应 VN 请求中的一个虚拟节点,输出是各虚拟节点的匹配集.算法共包括 4 步:第 1 步,从输入的映射约束中读取虚拟节点 *i* 的映射约束(*i*=1,2,…,*m*,*m* 为 VN 请求中虚拟节点数目);第 2 步,调用 SWRL 查询工具,在资源描述知识库的支持下对虚拟节点 *i* 的映射约束进行匹配,将满足条件的 InP 放入候选集(candidate set)中;第 3 步,将候选集中的元素放入虚拟节点 *i* 的匹配集中;第 4 步,置空候选集.以上 4 步循环执行,直到将所有映射约束处理完毕.候选集用于临时保存资源匹配结果,初始时应置为空集.

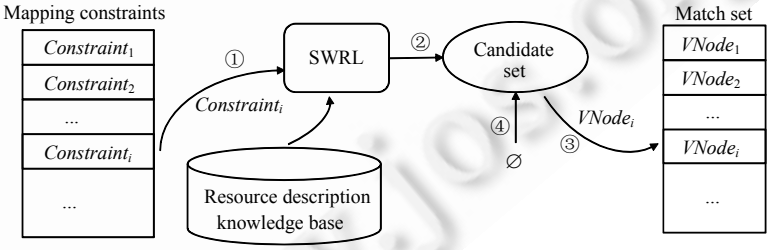


Fig.5 Algorithm flow for resource matching

图 5 资源匹配算法的流程

2.3 资源匹配算法评估

本节基于 OWL 语言为物理网络提供的虚拟资源创建了资源描述知识库,并利用 SWRL 工具设计了资源匹配算法.其中,资源描述知识库的概念模型规定了映射约束中可以使用的术语,SWRL 查询语言明确了映射约束

© 中国科学院软件研究所 http://www.jos.org.cn

的逻辑形式.相比于已有算法,本节的资源匹配算法支持 VN user 在概念模型和 SWRL 查询语言规范的范围内表达多样化的映射约束,且无需对映射约束进行逻辑范式转化即可完成匹配.此外,本节算法还能支持精确的数值属性匹配,避免了采用概念群集技术带来的数值属性模糊匹配问题.

理论上,本节算法的时间复杂度为 $O(n)$(n 为各 InP 可提供的虚拟节点类型总数),高于 Houidi 等人[10]算法的时间复杂度 $O(\lg n)$.但一方面,SWRL 查询工具优化了算法的查询部分,故实践中其复杂度接近 $O(\lg n)$,若能利用缓存中常用映射约束的匹配结果,则还可进一步提高算法效率;另一方面,后续 VN 划分问题属于 NP 问题,其时间消耗远大于资源匹配阶段,故从跨域 VN 映射的整体过程来看,资源匹配阶段并非影响全局效率的瓶颈.综上,本节资源匹配算法的效率是可以接受的.

3 基于遗传算法的虚拟网络划分算法

本节首先给出基于遗传算法的 VN 划分算法(GA-partition)的变量定义和目标函数的计算方法,然后介绍算法流程和 3 种遗传算子,最后证明算法的两个重要性质.

3.1 变量定义及目标函数

将 VN 请求的每一个可行划分方案表示为一个矩阵 $PM_{m\times n}$,其中,m 为 VN 请求中虚拟节点的数目,n 为物理网络中边界节点的数目.元素 $PM[i][j]$的取值代表虚拟节点 i 与边界节点 j 之间的映射关系.

- $PM[i][j]=1$,表示 $j\in MatchSet'(i)$且 $f_n(i)=j$;
- $PM[i][j]=0$,表示 $j\in MatchSet'(i)$但 $f_n(i)\neq j$;
- $PM[i][j]=-1$,表示 $j\notin MatchSet'(i)$.

因此,对于特定的 PM 矩阵,有唯一的划分方案与之对应.

同时,一个合法的 PM 矩阵必须满足以下条件:

1) 若 $j\in MatchSet'(i)$,则 $PM[i][j]=1$ 或 $PM[i][j]=0$;
2) 若 $j\notin MatchSet'(i)$,则 $PM[i][j]=-1$;
3) PM 的每一行有且仅有一个元素可被赋值为 1(每个虚拟节点都将映射到唯一的边界节点上).

对于特定的虚拟网络和物理网络,在完成资源匹配后,便确定了 PM 矩阵中值为−1 的元素,其他元素则暂时赋 0.对于一个划分方案,若 $f_n(i)=j$,则将 $PM[i][j]$的值由 0 改为 1,故划分方案也对应唯一的 PM 矩阵.综上,合法的 PM 矩阵与可行的 VN 划分方案一一对应.又由于 PM 矩阵的构建过程即遗传算法的编码过程,故该过程满足完备性、健全性和非冗余性[26].以图 1 中的 VN 划分方案为例,该方案可矩阵表示为表 2 的形式.

Table 2 VN partition scheme expressed as a matrix

表 2 VN 划分方案的矩阵表示

	P_1	P_2	P_3	P_4	P_5
$VNode_1$	0	1	−1	−1	−1
$VNode_2$	−1	−1	1	0	0
$VNode_3$	0	0	−1	0	1
$VNode_4$	−1	−1	0	1	0

VN 划分的求解目标是找到使映射开销最小的 VN 划分方案,其目标函数 f 可表示为

$$\text{Min}:f(PM_{m\times n},TM_{m\times m},CAP_{1\times m},VNC_{m\times n},PMC_{n\times n}).$$

f 中 5 个参数的具体含义如下:

1) $PM_{m\times n}$:虚拟节点与边界节点的映射关系矩阵;
2) $TM_{m\times n}$:VN 请求的流量矩阵,$TM[i][j]$表示虚拟节点 i 与虚拟节点 j 之间的带宽约束,参见文献[17,27];
3) $CAP_{1\times m}$:虚拟节点的能力约束矩阵,存放 VN 请求中所有虚拟节点的节点能力约束;
4) $VNC_{m\times n}$:虚拟节点映射的最小单位开销矩阵.令 InP_j 表示边界节点 j 所属的 InP,若 $InP_j\in MatchSet(i)$,则 InP_j 可提供至少一个满足 i 映射约束的虚拟节点类型,$VNC[i][j]$取这些虚拟节点类型的单位开销

© 中国科学院软件研究所 http://www.jos.org.cn

的最小值.若 $InP_j \notin MatchSet(i)$,则 $VNC[i][j]$取无穷大;

5) $PMC_{n\times n}$:边界节点间连接的最小单位开销矩阵.$PMC[i][j]$取边界节点 i,j 间所有路径的单位开销的最小值,可由弗洛伊德算法计算得到.同时,弗洛伊德算法还能得到任意两个边界节点间单位开销最小的路径,可用于提供虚拟链路的域间映射方案.

为便于表示,在介绍目标函数的计算方法之前,先定义运算符“⊗”:

$$a\otimes b=\begin{cases}0, & a\times b\leqslant 0\\ a\times b, & a\times b>0\end{cases}.$$

节点映射开销可表示为 $NODE_{cost}=\sum_{i=1}^{m}CAP[i]\times\sum_{k=1}^{n}PM[i][k]\otimes VNC[i][k]$,其中,$\sum_{k=1}^{m}PM[i][k]\otimes VNC[i][k]$,即虚拟节点 i 映射到相应边界节点的单位开销.域间链路映射开销可表示为 $LINK_{cost}=\sum_{i=1}^{m}\sum_{j=1}^{m}TM[i][j]\times PMC[u][v]$,其中,$u=\sum_{k=1}^{n}PM[i][k]\otimes k, v=\sum_{k=1}^{n}PM[j][k]\otimes k$.若 $f_n(i)=i$,则 $k=i'$时,$PM[i][k]=1$,$PM[i][k]\otimes k=i'$;否则,$PM[i][k]\otimes k=0$.故 $u=i'$,表示虚拟节点 i 映射到的边界节点.同理,v 表示虚拟节点 j 映射到的边界节点.计算可得,图 1 中 VN 划分方案的 $NODE_{cost}$ 和 $LINK_{cost}$ 分别为 89 和 53,故 $Cost$ 为 142.

3.2 算法流程及3种算子

3.2.1 GA-partition 算法流程

首先,依据遗传算法相关文献[26,28],解释遗传算法中几个重要概念在本文中的含义:

- 个体:一个可行的 VN 划分方案(一个 PM 矩阵)称为一个个体;
- 基因:一个虚拟节点称为一个基因;
- 基因性状:一个虚拟节点与全部边界节点的映射关系(PM 矩阵中的一行)称为该基因的性状;
- 种群:由一组个体形成的集合称为种群.

GA-partition 算法流程如下:

1. 随机生成一组个体,由这组个体形成一个种群,记为 PM_SET;
2. 对 PM_SET 中的个体两两进行交叉运算,并将得到的新个体加入 PM_SET,对其扩充;
3. 对扩充后 PM_SET 中的个体进行变异运算;
4. 对 PM_SET 中的个体进行选择运算,淘汰一部分个体;
5. 评估 PM_SET,看其是否满足算法终止条件:若满足,则算法终止,输出 PM_SET 中令目标函数最优的解,并结合弗洛伊德算法得到的单位开销最小的路径形成划分方案;若不满足,则跳至步骤 2.

GA-partition 算法的终止条件如下:

算法的终止由迭代次数 k 来控制.在一次迭代完成后,对当前迭代次数 i 进行判断:若 i 小于 k,则算法继续;若 i 大于或等于 k,则取出前 k 次迭代中每次迭代的最优解,形成最优解集合 $s=\{s_{i-k+1},s_{i-k+2},\ldots,s_i\}$.若 k 次迭代中第 1 次迭代的最优解与 s 中的最优解相等,即 $s_{i-k+1}=\min(s)$,则表明算法经 $k-1$ 次迭代后未找到比 s_{i-k+1} 更优的解,算法终止;若 $s_{i-k+1}>\min(s)$,则表明算法在 $k-1$ 次迭代内找到了比 s_{i-k+1} 更优的解,算法继续执行.

3.2.2 GA-partition 算法的 3 种算子

1. 交叉算子

交叉运算的目的是由两个个体 PM_1(父亲)和 PM_2(母亲)得到一个新个体 PM_3(孩子),让 PM_3 既具备 PM_1 的基因性状,也具备 PM_2 的基因性状.交叉运算的操作可以描述为:将 PM_1 和 PM_2 的基因进行随机分割,并分别选择部分基因性状遗传给 PM_3,具体的公式表达如下:

对 m 个基因,随机生成一个全排列,记为 $c_j(j=1,2,\ldots,m)$:

1) 若 m 是奇数,则变异操作为

© 中国科学院软件研究所 http://www.jos.org.cn

$$\begin{cases} 1 \leqslant j \leqslant \dfrac{m-1}{2}, & PM_3[c_j] = PM_1[c_j] \\ \dfrac{m-1}{2} < j \leqslant m-1, & PM_3[c_j] = PM_2[c_j] \\ j = m, & \text{随机生成}PM_3[c_j]\text{的基因性状} \end{cases};$$

2) 若 m 是偶数,则变异操作为

$$\begin{cases} 1 \leqslant j \leqslant \dfrac{m-2}{2}, & PM_3[c_j] = PM_1[c_j] \\ \dfrac{m-2}{2} < j \leqslant m-2, & PM_3[c_j] = PM_2[c_j] \\ j > m-2, & \text{随机生成}PM_3[c_j]\text{的基因性状} \end{cases}.$$

2. 选择算子

选择运算的目的是基于种群中个体适应度及淘汰概率的定义,淘汰种群中适应度较低的个体.本文采用基于排序的方法来定义适应度函数,即 PM_SET 中所有个体的目标函数按降序排列,将每个个体的适应度 s 定义为该个体在序列中的位置.淘汰概率函数 $P(s)$是适应度的函数.参考遗传算法的相关文献[26,28],本文设计淘汰概率的原则主要有:

1) 保证适应度最高的个体不被淘汰;
2) 适应度高的个体被淘汰的概率小;
3) 种群数目保持稳定(保证算法在执行过程中不会因为种群数目增加而降低搜索效率,也不会因为种群数目减少而降低优化程度).

基于以上原则,将 $P(s)$定义为

$$P(s) = \frac{k-s}{3\times(k-1)}, 1 \leqslant s \leqslant k, s \in N,$$

其中,k 为种群数目(k>1),N 为自然数.$P(s)$可以满足:

1) $P(k)$=0,保证适应度最高的个体不被淘汰;
2) $P(s)$为单调递减函数,使适应度高的个体被淘汰的概率小;
3) 设 x 为经选择运算后种群中被淘汰个体的数目,则 x 的期望可以表示为

$$E(x) = \int_1^k 1\times P(s)\mathrm{d}s = k/3.$$

即,选择运算可淘汰 1/3 的个体.考虑到每两个个体的交叉运算可生成一个新个体,故经交叉运算后种群数目会变为原来的 3/2.故,本文对 $P(s)$的定义可使种群数目保持稳定.

3. 变异算子

变异运算的目的是扩大 GA-partition 算法的搜索范围,以避免算法过早陷入局部最优.变异运算由变异概率 p 控制,其操作可描述为随机选择个体的一个基因,并随机生成一个新的基因性状替代原基因性状.对种群中适应度最高的个体不进行变异运算.

3.3 算法性质证明

性质 1. GA-partition 算法是可以终止的.

证明:采用反证法证明.设算法不可终止,即,算法会一直迭代下去.考虑经过第 i 次迭代后算法没有终止,则由第 3.2.1 节中算法的终止条件可知,s_{i-k+1}>min($s_{i-k+1},s_{i-k+2},\ldots,s_i$),设 min($s_{i-k+1},s_{i-k+2},\ldots,s_i$)=$s_j$,$i-k+2\leqslant j\leqslant i$.又由算法不可终止可知,算法在第 $j+k-1$ 次迭代后也不会终止,因此,同样有 $s(j)$>min($s_j,s_{j+1},\ldots,s_{j+k-1}$),此时,又可以找到一个比 $s(j)$更优的解.以此类推,由于迭代一直进行,可以找到无数个可行解,而可行解的数量是有限的,故矛盾. □

性质 2. GA-partition 算法的搜索空间包含整个解空间.

证明:性质 2 即证在不考虑变异的情况下,任意给定初始种群,解空间中的任意个体都能以一定概率被搜索

© 中国科学院软件研究所 http://www.jos.org.cn

到.以下证明中的“可以”均是概率意义上的.

假设虚拟节点数目为 i,且 i 为奇数(偶数同理).对任意的可行解 s,s 有 i 个基因及相应的基因性状.从任意初始种群开始,交叉运算可生成 s 的第 1 个基因性状,下次交叉运算又可以在保留 s 第 1 个基因性状的基础上生成 s 的第 2 个基因性状.以此类推,可以搜索到可行解 s_1,s_1 具有 s 的第 1 至第$(i-1)/2$ 个性状.同理,可以搜索到可行解 s_2,s_2 具有 s 的第$(i+1)/2$ 至第 $i-1$ 个基因性状.此时,可以将 s_1 和 s_2 进行交叉运算,保留 s 中的前 $i-1$ 个基因性状,并生成 s 中第 i 个基因性状.得证. □

4 仿真实验

本节仿真实验的目的包括如下 3 个方面:

1) 将本文提出的基于遗传算法的 VN 划分算法(GA-partition)与 Dietrich 等人[17]的基于边界节点信息的确切式 VN 划分算法(LID-partition)以及 Houidi 等人[15]的不公开边界节点信息的确切式 VN 划分算法(NID-partition)和其启发式算法(NID-partition-h)在效率(算法执行时间)和性能(解的最优性)上进行比较;
2) 在不同的物理网络拓扑下分析 GA-partition 的稳定性;
3) 分析 GA-partition 中 3 个重要参数(即迭代次数、种群数目、变异概率)对算法效率和性能的影响,并讨论参数的设置原则.

仿真实验使用配置为 4GB 内存、64 位 Win8 操作系统、Intel i5 处理器的计算机进行上述评估,实验代码使用 Java 语言编写,虚拟机版本为 jdk1.6.实验中物理网络及虚拟网络拓扑都采用 GT-ITM[29]工具随机生成,并利用 MATLAB 工具对实验结果进行分析.

4.1 算法效率和性能

考虑到若使用边界节点连接比较稀疏的物理网络拓扑,则在应用 NID-partition 时必须将不存在连接关系的边界节点间的链路开销设为无穷大,使一些虚拟网络用NID-partition无法找到划分方案,不便于NID-partition与其他算法进行性能比较.因此,为保证 NID-partition 对任意 VN 请求都存在划分的可行解,本节实验中使用全连接的物理网络拓扑.但应该指出的是,全连接的物理网络在实际中并不具备代表性,后续的第 4.2 节将讨论 GA-partition 在不同物理网络拓扑下的效率和性能.

随机生成分为 10 个自治域的物理网络,每个自治域中有 2 个边界节点,边界节点间为全连接,共 190 条链路.链路单位开销服从[1,10]的均匀分布.随机生成 2 000 个 VN,VN 中虚拟节点数目服从[1,8]的均匀分布,虚拟节点间以 50%的概率连接.本节比较 4 种 VN 划分算法在不同 VN 划分问题规模下的效率和性能.VN 划分的问题规模(以下简称问题规模)即虚拟节点数目与虚拟节点候选集中元素数目的乘积.本节将每个虚拟节点匹配集的元素个数固定为 2(即,每个虚拟节点有 4 个可映射的边界节点).将 GA-partition 算法中的相关参数依次设定为:迭代次数为 30,种群数目为 200,变异概率为 0.1.

- 首先,比较 4 种 VN 划分算法的运行效率.

本文用平均划分时间来衡量算法的运行效率,即,算法完成一个 VN 划分需要的平均计算时间.图 6 的左图展示了随 VN 请求中虚拟节点数目的增加,4 种 VN 划分算法在平均划分时间上的差异,图 6 的右图为左图中 GA-partition 和 NID-partition-h 的放大图,可以看出:

1) LID-partition 和 NID-partition 的平均划分时间随着问题规模的增大呈指数增长,当虚拟节点较多时,可用性较差;而 GA-partition 和 NID-partition-h 的平均划分时间则随虚拟节点数目的增多呈近似线性增长,在问题规模较大时,也能在可接受的时间内完成划分.文献[15]的实验结果也展示了 NID-partition 和 NID-partition-h 的平均划分时间随着问题规模的增加分别呈指数和近似线性增长,与本文的实验结果一致;
2) 在问题规模较大时,LID-partition 的平均划分时间比 NID-partition 要长.其原因在于:与 NID-partition 相比,LID-partition 的二元整数规划中还涉及对虚拟链路的划分,其二元变量的数目比 NID-partition

© 中国科学院软件研究所 http://www.jos.org.cn

要多;

3) 当虚拟节点数目小于 4 时,GA-partition 和 NID-partition-h 的平均划分时间都小于 50ms,数量级较小,实验数据受算法外部环境因素(如 CPU 进程调度、磁盘读写速率等)影响较大,两种方法互有优劣;但当虚拟节点数目较大时,GA-partition 基本上优于 NID-partition-h.

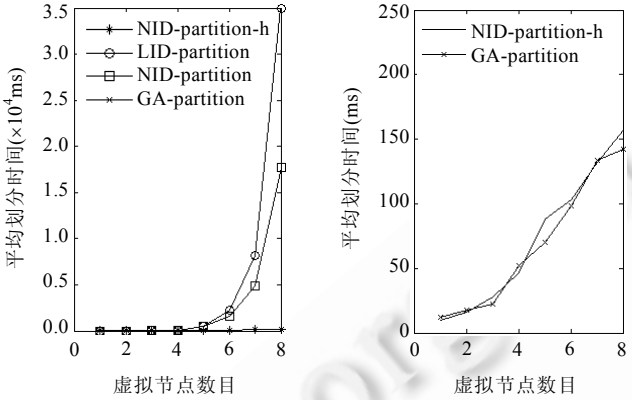


Fig.6 Comparison of four VN partitioning algorithms in efficiency

图 6 4 种 VN 划分算法的效率比较

- 其次,比较 4 种 VN 划分算法的性能.

本文用额外映射开销来衡量算法性能.额外映射开销越小,表明算法的性能越好.LID-partition 的额外映射开销为 0(最优解),故将其作为性能比较的基准.

设 $Cost(M)$表示算法 M 的映射开销,则其额外映射开销 $extraCost(M)$按如下方式计算:

$$extraCost(M)=[Cost(M)-Cost(LID-partition)]/Cost(LID-partition).$$

由图 7 的左图可以看出:相比于 NID-partition 和 NID-partition-h,GA-partition 大幅提高了 VN 划分的性能,即使在问题规模较大时,也能将平均额外映射开销控制在 5%以内,另外两种算法则超过了 10%.此外,随着问题规模的增大,GA-partition 的平均额外映射开销逐渐增加,而 NID-partition 和 NID-partition-h 的平均额外映射开销没有显著变化.其原因在于,后两种算法的最优化瓶颈是未公开边界节点信息.

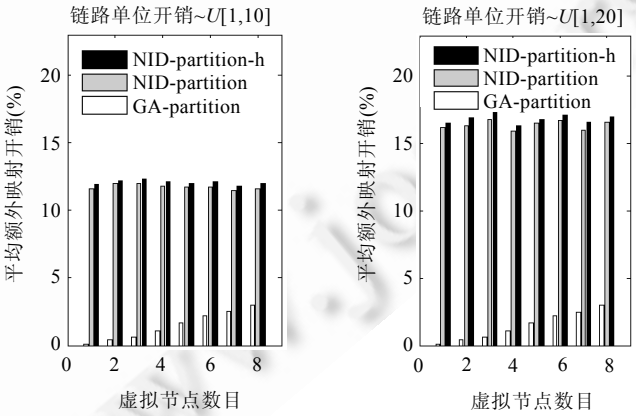


Fig.7 Comparison of four VN partitioning algorithms in performance

图 7 4 种 VN 划分算法的性能比较

进一步地,考察边界节点信息对性能的影响.使物理链路的单位开销服从[1,20]的均匀分布,以增加链路单

© 中国科学院软件研究所 http://www.jos.org.cn

位开销的差异.此时,两个边界节点间的最小开销路径就可能不是直连链路,而是经其他边界节点中继的物理路径.公开边界节点信息的算法(LID-partition 和 GA-partition)会使用边界节点间的最小开销路径来优化划分方案,而不公开边界节点信息的算法(NID-partition 和 NID-partition-h)仍使用直连链路进行划分.如图 7 中右图所示,边界节点信息对 GA-partition 的影响不大,而 NID-partition 和 NID-partition-h 的性能则有明显下降.

4.2 GA-partition算法的稳定性

本节在不同物理网络拓扑上对 GA-partition 的稳定性进行验证.令物理网络中自治域的数目及自治域中边界节点的数目保持不变,调整边界节点间的连接概率生成 3 个物理网络拓扑,链路数分别是 44,102,190.链路单位开销服从[1,10]的均匀分布.虚拟网络的生成与第 4.1 节相同.图 8 展示了 GA-partition 在上述 3 个拓扑上的效率和性能.可以看出:GA-partition 的效率和性能并没有受物理网络拓扑变化的影响,稳定性较好.

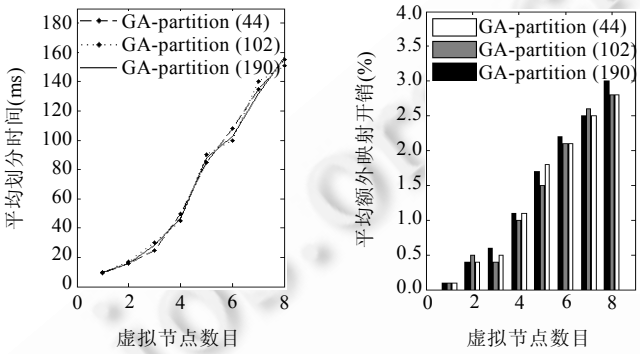


Fig.8 Efficiency and performance of GA-partition in different substrate topology

图 8 不同物理网络拓扑下 GA-partition 的效率和性能

4.3 GA-partition算法的参数设置

本节讨论迭代次数、种群数目和变异概率对 GA-partition 算法的性能、效率和稳定性的影响,并总结其设置原则.随机生成 10 个自治域,每个自治域中有 2 个边界节点,共 44 条链路,链路的单位开销服从[1,10]的均匀分布.虚拟网络的生成与第 4.1 节相同.

首先,将变异概率固定为 0.1,讨论种群数目(范围:50~500)和迭代次数(范围:10~50)对 GA-partition 算法效率的影响.

由图 9 可以看出,虽然增加迭代次数和种群数目消耗了更多的计算时间,但却可以提高 GA-partition 的性能.具体来说:图 9 的左图表明,GA-partition 的效率与迭代次数和种群数目呈近似线性关系;而图 9 的右图表明,GA-partition 的性能与迭代次数和种群数目虽然没有明显的函数关系,但可以看出呈现如下的一些趋势:

(1) 当种群数目超过 200 时,可以保证平均额外映射开销小于 2%;而当种群数目达到 300 后,再继续增加种群数目,平均额外映射开销没有明显变化;

(2) 当种群数目为 50 时,随着迭代次数的增加,平均额外映射开销显著减少;

(3) 随种群数目的增多,迭代次数对平均额外映射开销的影响逐渐减小.

然后,将迭代次数和种群数目分别固定为 10 和 200(由图 9 可知,此时算法在效率和性能上都有较好的表现),讨论变异概率(范围:0~0.5)对 GA-partition 效率和性能的影响.

由图 10 的左图可以看出,变异概率对 GA-partition 效率的影响是近似线性的.而变异概率对 GA-partition 性能的影响则相对复杂,这是因为变异概率如果太小,则容易陷入局部最优;如果太大,则容易因种群搜索的跳跃性太大而损失最优解.由图 10 的右图可知,变异概率在[0.1,0.2]内算法的性能最好.

© 中国科学院软件研究所 http://www.jos.org.cn

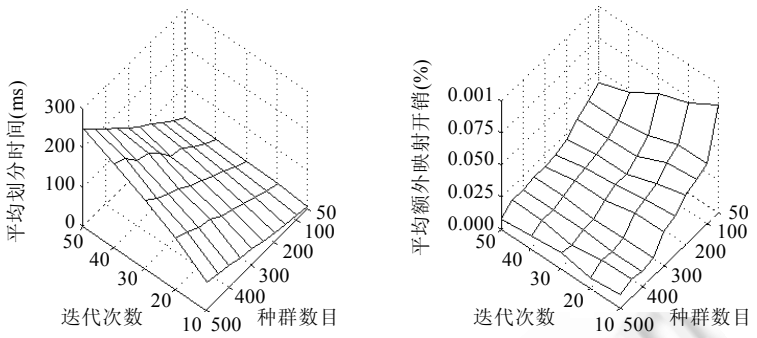


Fig.9 Effects of populations and iterations on the efficiency and performance of GA-partition

图 9 种群数目和迭代次数对 GA-partition 效率和性能的影响

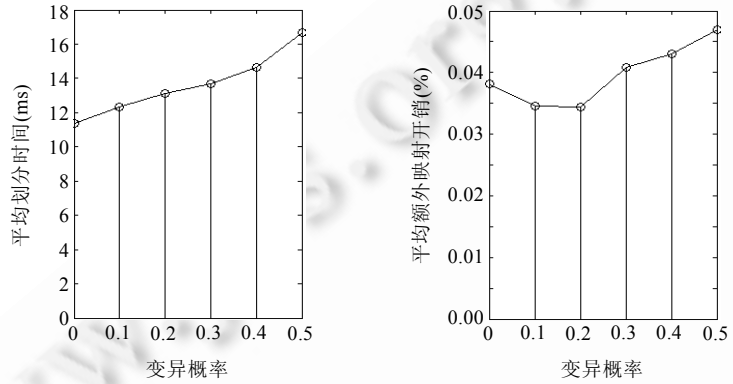


Fig.10 Effects of mutation probability on the efficiency and performance of GA-partition

图 10 变异概率对 GA-partition 效率和性能的影响

进一步地,考察变异概率为 0.1 和 0.2 时 GA-partition 的性能.图 11 中展示了 2 000 个 VN 在两种变异概率下的额外映射开销.

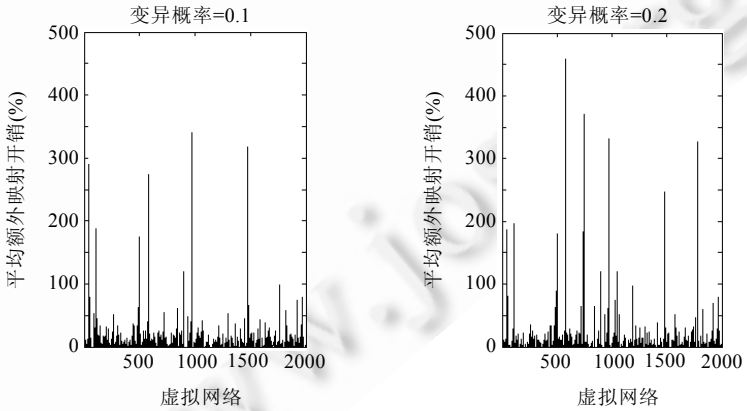


Fig.11 Comparison of GA-partition's performance in two different mutation probabilities

图 11 两种变异概率下 GA-partition 的性能比较

可以看出:在两种变异概率下,本文算法对绝大部分 VN 都可以得到较优的划分方案(额外映射开销在 10%

© 中国科学院软件研究所 http://www.jos.org.cn

以内).但由于遗传算法随机搜索的本质,实验结果中也存在一些极端情况(个别 VN 划分方案的额外映射开销大于 100%).当变异概率为 0.1 时,极端情况较少出现;当变异概率为 0.2 时略有增加.计算可得:当变异概率为 0.1 时,2 000 个 VN 额外映射开销的方差约为 0.031;而当变异概率为 0.2 时的方差约为 0.043.此外,虽然两种变异概率下的平均额外映射开销基本一致(如图 10 所示),但由图 11 可以看出:当变异概率为 0.1 时,算法的稳定性更好.故在平均额外映射开销相同的情况下,应选择较小的变异概率以提高稳定性.

5 相关工作

在获悉单个自治域内全部物理网络信息的前提下,Chowdhury 等人[5]提出了一种同阶段的单域 VN 映射算法,以提高 VN 接受率和均衡网络负载为目标,利用混合整数规划,同时求出映射虚拟节点和虚拟链路的优化解.Fajjari 等人[6]基于蚁群算法提出了一种单域 VN 映射方法,以降低映射开销和提高 VN 接受率为目标,提高了 InP 的长期运营收益.Cheng 等人[7]基于粒子群算法设计的单域 VN 映射方法,显著提高了 InP 的长期运营收益.Huang 等人[8]利用多目标负载均衡的粒子群算法,优化了单域 VN 映射中节点和链路的负载平衡.Guo 等人[9]建立了多目标优化模型,并采用粒子群方法统一分配节点和链路资源,提高了单域 VN 映射的成功率和物理资源的利用率.

然而,物理网络往往被天然地分为多个自治域,这些自治域可提供的虚拟资源种类和价格都不尽相同.为满足 VN 请求中对虚拟资源的映射约束并尽可能地降低映射开销,必须通过跨域映射来完成 VN 部署.

已有的跨域 VN 映射方法可分为两类:分布式和集中式.典型的分布式跨域映射方法包括:Chowdhury 等人[13]提出的基于策略的分布式域间映射框架,并提供地域分级寻址机制和位置感知分发协议为该框架提供支持;Houidi 等人[30]提出的分布式虚拟网络映射算法,并基于多 Agent 技术来确保节点间的通信和同步;Zaheer 等人[14]设计的 V-Mart 机制,该机制不仅为 InP 提供了公平的资源发布平台,也为 SP 提供了客户驱动的 VN 划分方法.与分布式方法不同,集中式跨域映射方法强调在 VNP 的统一协调下完成跨域映射中的资源匹配和 VN 划分.

对于资源匹配,Houidi 等人[10]提供了基于 XML 语言的虚拟资源描述模式,并利用概念群集技术将虚拟资源按相似性进行层级分类,再基于得到的概念分类树给出资源匹配算法.以该方法为基础,Lv 等人[11]提出了基于 VN 的服务特点对虚拟资源分类的方法,避免了盲目搜索;Medhioub 等人[12]提出了基于重要性及使用频率对虚拟资源分类的方法,提高了资源匹配的效率,并克服了概念分类树难以扩展的缺点.

对于 VN 划分,Houidi 等人[15]基于全连接的物理拓扑,给出了 VN 划分的确切式和启发式算法;Dietrich 等人[17]以 InP 公开边界节点的相关信息为前提,利用二元整数规划为 VN 划分问题提供了一种更切合实际的确切式算法;Zhang 等人[19]提出了一种分层线性规划模型,并基于对模型的分解设计了 VN 划分算法.

本文主要研究集中式跨域 VN 映射中的资源匹配和 VN 划分问题.与上述方法不同,本文基于 OWL 知识描述和 SWRL 查询工具来完成资源匹配,并使用遗传算法求解 VN 划分.本文方法可与已有单域 VN 映射方法[3−9]结合使用来完成跨域 VN 映射,即:在 VN 划分阶段首先使用本文方法得到划分方案,将 VN 划分为多个虚拟子网,使跨域 VN 映射转化为多个单域 VN 映射;然后,在虚拟子网映射阶段使用文献[3−9]的方法求解单域 VN 映射.

6 结束语

跨域虚拟网络映射是网络虚拟化领域的一个研究重点.本文对集中式跨域虚拟网络映射的两个重要阶段(资源匹配和虚拟网络划分)中存在的问题分别进行了研究,提出:(1) 一种基于 OWL 知识描述和 SWRL 查询工具的资源匹配算法,与已有资源匹配工作相比,本文算法能够支持 VN user 表达多样化的映射约束以及精确的数值属性匹配;(2) 一种基于遗传算法的 VN 划分算法(GA-partition),并通过理论分析证明了 GA-partition 算法是可以终止的,且算法的搜索空间包含整个解空间.仿真实验结果表明:GA-partition 算法大幅提高了 VN 划分的求解效率,且算法的输出结果与最优解的误差可控制在 5%以内.

未来工作主要从以下 3 个方面进行:(1) 设计高效的资源信息管理机制,将物理网络的即时性信息(如物理

© 中国科学院软件研究所 http://www.jos.org.cn

资源的占用情况、可用能力信息等)引入到资源匹配中,提高资源匹配结果的准确性和有效性;(2) 采用自适应的遗传算法动态调整变异概率和选择概率,进一步提高 VN 划分的性能;(3) 综合考虑其他优化目标(如,功耗或负载平衡等),对本文方法加以完善.

References:

[1] Anderson T, Peterson L, Shenker S, Turner J. Overcoming the Internet impasse through virtualization. Computer, 2005,38(4): 34−41. [doi: 10.1109/MC.2005.136]

[2] Feamster N, Gao L, Rexford J. How to lease the Internet in your spare time. ACM SIGCOMM Computer Communication Review, 2007,37(1):61−64. [doi: 10.1145/1198255.1198265]

[3] Yu M, Yi Y, Rexford J, Chiang M. Rethinking virtual network embedding: Substrate support for path splitting and migration. ACM SIGCOMM Computer Communication Review, 2008,38(2):17−29. [doi: 10.1145/1355734.1355737]

[4] Lischka J, Karl H. A virtual network mapping algorithm based on subgraph isomorphism detection. In: Proc. of the 1st ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures. 2009. 81−88. [doi: 10.1145/1592648.1592662]

[5] Chowdhury N, Rahman M, Boutaba R. Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. IEEE/ACM Trans. on Networking, 2012,20(1):206−219. [doi: 10.1109/TNET.2011.2159308]

[6] Fajjari I, Aitsaadi N, Pujolle G, Zimmermann H. VNE-AC: Virtual network embedding algorithm based on ant colony metaheuristic. In: Proc. of the 2011 IEEE Int'l Conf. on Communications. 2011. 1−6. [doi: 10.1109/icc.2011.5963442]

[7] Cheng X, Zhang ZB, Su S, Yang FC. Virtual network embedding based on particle swarm optimization. Acta Electronica Sinica, 2011,39(10):2240−2244 (in Chinese with English abstract).

[8] Huang BB, Lin RH, Peng K, Zou H, Yang FC. Load-Balancing based on particle swarm optimization in virtual network mapping. Journal of Electronics & Information Technology, 2013,35(7):1753−1759 (in Chinese with English abstract).

[9] Guo ZE, Xue HW, Dai YQ. A multi-objective particle swarm optimization based virtual network embedding algorithm. Journal of National University of Defense Technology, 2013,35(5):163−167 (in Chinese with English abstract).

[10] Houidi I, Louati W, Zeghlache D, Baucke S. Virtual resource description and clustering for virtual network discovery. In: Proc. of the IEEE Int'l Conf. on Communications Workshop on Communications. 2009. 1−6. [doi: 10.1109/ICCW.2009.5207979]

[11] Lv B, Wang Z, Huang T, Chen J, Liu Y. Virtual resource organization and virtual network embedding across multiple domains. In: Proc. of the 2010 Int'l Conf. on Multimedia Information Networking and Security. 2010. 725−728. [doi: 10.1109/MINES.2010.154]

[12] Medhioub H, Houidi I, Louati W Zeghlache D. Design, implementation and evaluation of virtual resource description and clustering framework. In: Proc. of the 2011 IEEE Int'l Conf. on Advanced Information Networking and Applications. 2011. 83−89. [doi: 10.1109/AINA.2011.46]

[13] Chowdhury M, Samuel F, Boutaba R. PolyViNE: Policy-Based virtual network embedding across multiple domains. In: Proc. of the 2nd ACM SIGCOMM Workshop on Virtualized Infrastructure Systems and Architectures. 2010. 49−56. [doi: 10.1145/1851399.1851408]

[14] Zaheer F, Xiao J, Boutaba R. Multi-Provider service negotiation and contracting in network virtualization. In: Proc. of the 2010 IEEE/IFIP Network Operations and Management Symp. 2010. 471−478. [doi: 10.1109/NOMS.2010.5488487]

[15] Houidi I, Louati W, Bean-Ameur W, Zeghlache D. Virtual network provisioning across multiple substrate networks. Computer Networks, 2011,55(4):1011−1023. [doi: 10.1016/j.comnet.2010.12.011]

[16] Werle C, Bless R, Papadimitriou P, Houidi I, Louati W, Zeghlache D, Mathy L. Building virtual networks across multiple domains. ACM SIGCOMM Computer Communication Review, 2011,41(4):412−413. [doi: 10.1145/2018436.2018495]

[17] Dietrich D, Rizk A, Papadimitriou P. Multi-Domain virtual network embedding with limited information disclosure. In: Proc. of the IFIP Networking Conf. 2013. 1−9. http://www.ikt.uni-hannover.de/lavinet.html?&L=1

[18] Dietrich D, Rizk A, Papadimitriou P. AutoEmbed: Automated multi-provider virtual network embedding. ACM SIGCOMM Computer Communication Review, 2013,43(4):465−466. [doi: 10.1145/2486001.2491690]

[19] Zhang M, Wu C, Wang B, Jiang M. Research on mapping method of logical carrying network across multiple domains. Journal on Communications, 2012,33(8):200−207 (in Chinese with English abstract).

© 中国科学院软件研究所 http://www.jos.org.cn

[20] Schaffrath G, Werle C, Papadimitriou P, Feldmann A, Bless R, Greenhalgh A, Wundsam A, Kind M, Maennel O, Mathy L. Network virtualization architecture: Proposal and initial prototype. In: Proc. of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures. 2009. 63−72. [doi: 10.1145/1592648.1592659]

[21] Michalski R, Stepp R. Automated construction of classifications: Conceptual clustering versus numerical taxonomy. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1983,5(4):396−410. [doi: 10.1109/TPAMI.1983.4767409]

[22] DE-CIX. https://www.de-cix.net/

[23] PeeringDB. http://www.peeringdb.com/

[24] OWL. http://www.w3.org/TR/owl-features/

[25] SWRL. http://www.w3.org/Submission/SWRL/

[26] Han RF. Principle of Genetic Algorithm and Applications. Beijing: The Publishing House of Ordnance Industry, 2009 (in Chinese).

[27] Wang C, Wolf T. Virtual network mapping with traffic matrices. In: Proc. of the 7th ACM/IEEE Symp. on Architectures for Networking and Communications Systems. 2011. 225−226. [doi: 10.1109/ANCS.2011.44]

[28] Holland JH. Adaptation in Natural and Artificial Systems. Cambridge: MIT Press, 1992.

[29] GT-ITM. http://www.cc.gatech.edu/projects/gtitm/

[30] Houidi I, Louati W, Zeghlache D. A distributed virtual network mapping algorithm. In: Proc. of the IEEE Int'l Conf. on Communications. 2008. 5634−5640. [doi: 10.1109/ICC.2008.1056]

附中文参考文献:

[7] 程祥,张忠宝,苏森,杨放春.基于粒子群优化的虚拟网络映射算法.电子学报,2011,39(10):2240−2244.

[8] 黄彬彬,林荣恒,彭凯,邹华,杨放春.基于粒子群优化的负载均衡的虚拟网络映射.电子与信息学报,2013,35(7):1753−1759.

[9] 郭智恩,薛海伟,戴一奇.一种基于多目标微粒群优化的虚拟网络映射方法.国防科技大学学报,2013,35(5):163−167.

[19] 张旻,吴春明,王滨,姜明.跨域逻辑承载网映射方法研究.通信学报,2012,33(8):200−207.

[26] 韩瑞锋.遗传算法原理与应用实例.北京:兵器工业出版社,2009.

肖蔼玲(1987－),女,山东烟台人,在读博士生,主要研究领域为云计算,未来网络管理,数据仓库与数据挖掘.
E-mail: xiao_ailing@bupt.edu.cn

邱雪松(1973－),男,博士,教授,博士生导师,主要研究领域为网络与业务管理.
E-mail: xsqiu@bupt.edu.cn

王颖(1976－),女,博士,副教授,主要研究领域为云计算,未来网络管理,IT 服务管理.
E-mail: wangy@bupt.edu.cn

李文璟(1973－),女,博士,副教授,主要研究领域为未来网络管理,无线网络管理,自主管理.
E-mail: wjli@bupt.edu.cn

孟洛明(1955－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为网络管理,通信软件.
E-mail: lmmeng@bupt.edu.cn

尹斌(1984－),男,博士,工程师,主要研究领域为需求工程,软件工程,知识工程,云计算.
E-mail: ybziwen@163.com

© 中国科学院软件研究所 http://www.jos.org.cn