

文章编号:1001-9081(2015)02-0299-06

doi:10.11772/j.issn.1001-9081.2015.02.0299

可靠性感知下的虚拟数据中心映射算法

左 成*, 虞红芳

(光纤传感与通信教育部重点实验室(电子科技大学), 成都 611731)

(* 通信作者电子邮箱 zuocheng2009@gmail.com)

摘要:介绍现阶段虚拟数据中心(VDC)映射的研究进展,根据租户对VDC可靠性的需求,提出一种可靠性感知下的VDC映射启发式算法。对于每个VDC,该算法通过限制能放置在同一个服务器上的最大虚拟机数目来保证租户VDC可靠性需求,然后以降低数据中心网络带宽消耗和服务器能耗为主要目标进行VDC映射。其具体做法是:首先将相互之间带宽需求量大的虚拟机合并并部署来降低数据中心网络带宽的消耗;然后把合并后的虚拟机优先部署到已开启的服务器上,从而减少开启的服务器数目,降低数据中心的服务器能耗。利用基于胖树结构的数据中心拓扑对提出的算法进行了仿真,结果表明,与2EM算法相比,该算法能够满足租户VDC的可靠性需求,能在不增加额外能耗的前提下最多减少数据中心网络约30%的带宽消耗。

关键词:可靠性;带宽消耗;虚拟数据中心映射;能耗;胖树

中图分类号: TP393.01 **文献标志码:**A

Reliability-aware virtual data center embedding algorithm

ZUO Cheng*, YU Hongfang

(Key Laboratory of Optical Fiber Sensing and Communications, Education Ministry of China
(University of Electronic Science and Technology of China), Chengdu Sichuan 611731, China)

Abstract: By introducing the current research progress of Virtual Data Center (VDC) embedding, and in accordance with the reliability requirement of VDC, a new heuristic algorithm to address reliability-aware VDC embedding problem was proposed. It restricted the number of Virtual Machines (VMs) which can be embedded onto the same physical server to guarantee the VDC reliability, and then regarded reduction of the bandwidth consumption and energy consumption as main objective to embed the VDC. Firstly, it reduced bandwidth consumption of data center by consolidating the virtual machines, which had high communication services, into the same group and placed them onto the same physical server. Secondly, the consolidated groups were mapped onto the powered physical servers to decrease the number of powered servers, thus reducing the power consumption of servers. The results of experiment conducted on fat tree topology show that, compared with 2EM algorithm, the proposed algorithm can satisfy VDC reliability requirement, and effectively reduce a maximum of 30% bandwidth consumption of data center without increasing extra energy consumption.

Key words: reliability; bandwidth consumption; virtual data center embedding; power consumption; fat tree

0 引言

在云计算时代,随着社会对计算需求的不断扩大,数据中心的规模也在迅速变大。但是,在庞大的数据中心背后,其资源的平均利用率却相对较低,大部分设备空闲,给数据中心增加了巨大的能耗负担^[1]。目前,数据中心中使用虚拟化技术,可以有效提高数据中心资源利用率。在这种新趋势下,每个租户的资源请求可抽象为一组虚拟机(Virtual Machine, VM)构成的虚拟数据中心(Virtual Data Center, VDC)^[2],每个VM对应一定的计算资源(包括CPU、内存以及硬盘等);同时为了传递数据和中间文件,VM之间需要建立具有带宽保障的通信链路,以满足VM之间的通信需求。由于VM放置与VM间通信带宽的路由的紧耦合,使得把VDC映射到数据中心的这个过程变得非常复杂。

虚拟网络(Virtual Network, VN)映射问题^[3]与VDC映射问题很类似,在VN映射上已有许多研究,但VDC映射的研究还较少。VDC映射与VN映射不同的是:在映射VN时,每个物理服务器上只能映射一个VM,而映射VDC时,每个物理服务器上可以同时映射多个VM。因此现有的VN映射算法并不能直接用于解决VDC映射问题^[4]。

Guo等^[4]提出的SecondNet算法解决了VDC映射的带宽保障问题,它可以在一定程度上提高数据中心网络的利用率;但是由于SecondNet算法在一个服务器上只能映射一个VM,所以网络资源利用率仍然较低,VM之间的带宽需求会造成数据中心带宽的巨大消耗。Fuerst等^[5]提出的LOCO算法利用对VM进行自动分组的方式来提高网络资源利用率。与SecondNet算法相比,LOCO算法提高了数据中心网络的利用率和VDC的映射成功率;但是由于LOCO算法把VDC中的

收稿日期:2014-09-01;修回日期:2014-10-21。

基金项目:国家973计划项目(2013CB329103);国家自然科学基金资助项目(61271171)。

作者简介:左成(1990-),男,四川绵阳人,硕士研究生,CCF会员,主要研究方向:数据中心网络虚拟化、软件定义网络;虞红芳(1975-),女,浙江萧山人,教授,博士,主要研究方向:网络虚拟化、软件定义网络。

VM 尽可能合并映射,所以 VDC 的可靠性无法得到保障。Luo 等^[6]提出的 2EM 算法首次以节能为目标来进行 VDC 映射。2EM 算法对 LOCO 算法进行了优化,使得能关闭的空闲服务器和链路尽可能多;但是 2EM 也是把 VM 尽可能合并映射,所以 VDC 的可靠性也无法保障。

从前面分析可知,现有的 VDC 映射算法还未涉及可靠性问题。本文提出一个可靠性感知的 VDC (Reliability-Aware Virtual Data Center, RAVDC) 映射算法来解决可靠性感知下的 VDC 映射问题。RAVDC 算法会首先保证 VDC 的可靠性需求,然后以减少带宽消耗和降低能耗为主要目标进行 VDC 映射。

1 问题描述

1.1 可靠性感知下的 VDC 映射问题

VDC 是一系列虚拟资源的组合,主要包括:虚拟机、虚拟交换机、虚拟路由器以及虚拟链路^[2]。在云数据中心中的主要挑战是 VDC 映射问题,其主要目标是找到虚拟资源到物理资源的映射来提高数据中心的资源利用率。除此之外,VDC 映射还需要考虑可靠性、带宽保障和能耗等其他目标,以便 VDC 算法能应用于实际生产环境。

VDC 映射问题中的可靠性是指在有单个服务器失效时,VDC 中仍然有效的虚拟机数目占 VDC 中总虚拟机数目的百分比。可靠性给 VDC 映射增加了新的挑战,提高可靠性要求 VDC 映射时 VM 尽量分散部署,而减少带宽消耗则要求 VDC 中的 VM 尽量集中部署,所以可靠性和带宽消耗是相互矛盾的目标。降低能耗要求开启的服务器数目少,这就要求尽可能集中放置 VM。所以减少带宽消耗和降低能耗的实现方式都是尽可能集中放置 VM,从而使得如果能减少带宽消耗,就能相应降低能耗。降低能耗不是本文的主要目标,本文的研究重点是解决可靠性和带宽消耗之间的矛盾。

使用图 1 来说明 VDC 映射问题中的可靠性与带宽消耗的矛盾。其中:圆圈代表 VM,六角菱形代表数据中心服务器,正方形代表数据中心交换机,服务器与交换机或交换机与交换机之间连线上的数字表示该物理链路上的带宽消耗。

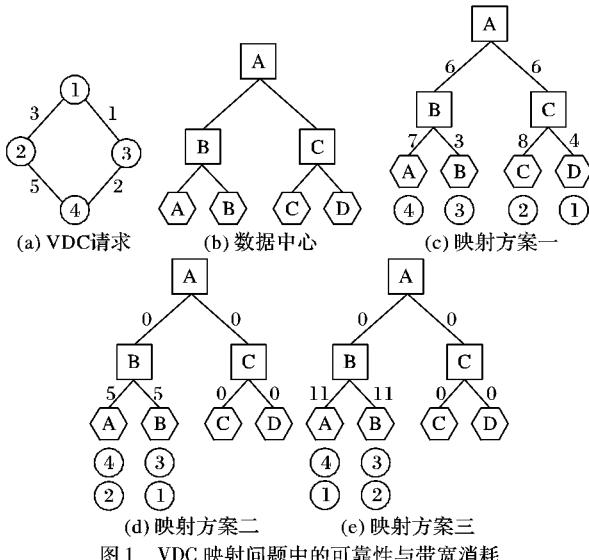


图 1 VDC 映射问题中的可靠性与带宽消耗

图 1(a)表示一个有 4 个 VM 节点的 VDC 请求,VM 之间连线上的数字表示虚拟链路的带宽需求;图 1(b)表示一个数据中心;图 1(c)~(e)表示三种不同的映射方案,服务器下方

的圆圈表示映射到该服务器上的 VM。图 1(c)中,每个服务器上只映射一个 VM,此时 VDC 的可靠性为 75%,而带宽消耗为 34 个单位;图 1(d)中,每个物理服务器映射了 2 个 VM,此时 VDC 的可靠性为 50%,而带宽消耗为 10 个单位;图 1(e)中,每个服务器也映射 2 个 VM,虽然此时 VDC 的可靠性同样为 50%,但带宽消耗变成 22 个单位。从这三个映射方案中可以得到两点启示:1) VM 集中放置会引起可靠性降低,但带宽消耗减少,因此为了满足 VDC 的可靠性需求,需要控制 VM 整合放置的数目;2) 相同的 VM 整合放置数量会导致相同的可靠性,但 VM 的不同组合方式会产生不同的带宽消耗,因此选择哪些 VM 整合在一起是本文需要解决的关键问题。

本文用 $C(V, A)$ 表示数据中心,其中: V 表示数据中心的节点集合, A 表示数据中心网络的链路集合;节点分为交换机和服务器,交换机集合用 W 表示,服务器集合用 S 表示。用 $G^k(V^k, A^k)$ 表示第 k 个 VDC 请求,其中: V^k 表示 VDC 的 VM 集合, A^k 表示 VDC 中 VM 之间的虚拟链路集合。对于每个 VDC 请求,VM 的需求为一定数量的虚拟 CPU(virtual CPU, vCPU),虚拟链路的需求为一定数量的带宽。

1.2 可靠性模型

数据中心中的失效域^[7]有多种类型,主要包括服务器、柜顶(Top of Rack, TOR)交换机、汇聚交换机和供电设备等,本文只考虑服务器失效,并使用最坏情况有效率作为可靠性指标。最坏情况有效率是指在最坏情况下发生单个失效时,VDC 中仍然有效的 VM 数目占 VM 总数目的百分比。

定义 1 VDC 的可靠性。第 i 个 VDC 的可靠性的计算公式如下:

$$R(i) = (s_i - \max_{j \in S} s_{i,j}) / s_i \quad (1)$$

其中: $R(i)$ 表示第 i 个 VDC 的可靠性; s_i 表示第 i 个 VDC 中 VM 的总数目; $s_{i,j}$ 表示第 i 个 VDC 在服务器 j 上部署的 VM 数目,如果服务器 j 上没有映射 VDC i 的 VM,则 $s_{i,j}$ 为 0。

在可靠性感知下的 VDC 映射问题中,租户给出 VDC 的可靠性需求,即 $R(i)$ 。根据 $R(i)$ 可计算出虚拟机整合门限 K , K 的物理意义表示 VDC 在单个服务器上最多能部署的 VM 个数。虚拟机整合门限 K 的计算公式如下:

$$K = \lfloor (1 - R(i)) \cdot s_i \rfloor \quad (2)$$

利用 K ,可以保证 VDC 映射后实际的可靠性不会低于租户的需求。

1.3 带宽消耗模型

当有带宽需求的 VM 映射到同一物理服务器上时,这些 VM 间的通信可以由物理服务器的 hypervisor 完成,不需要占用额外的数据中心网络带宽资源。只有在不同物理服务器上的 VM 间需要通信时,才会占用网络带宽资源。本文将使用一个简单的模型来衡量 VDC 映射方案的带宽消耗。

定义 2 VDC 的带宽消耗。第 i 个 VDC 的带宽消耗的计算公式如下:

$$BC(i) = \sum_{u, v \in V_i} (Hops_{u,v} \cdot BD_{u,v}) \quad (3)$$

其中: $BC(i)$ 表示第 i 个 VDC 的带宽消耗量; $Hops_{u,v}$ 表示 VM u 和 VM v 之间虚拟链路所对应的底层路径的跳数,如果 u 和 v 在同一个服务器上,则 $Hops_{u,v}$ 为 0; $BD_{u,v}$ 表示 VM u 和 VM v 之间虚拟链路的带宽需求量。

显然,从带宽消耗角度看,VM 应该尽可能多地整合放置到同一物理服务器。但这样会导致这个物理服务器失效时,有

大量的 VM 失效,从而无法保障租户的可靠性需求。因此,如何协调可靠性和带宽消耗间的矛盾是本文要解决的关键问题。

2 RAVDC 算法

由于单纯的 VDC 映射已经是 NP-Hard 问题,因此本文针对可靠性感知下的 VDC 映射问题提出了启发式算法 RAVDC。

RAVDC 算法采用迭代的方式,首先从 VDC 所有未映射的 VM 中找出带宽资源节约最多的 K 个 VM(K 是虚拟机整合门限,这 K 个 VM 与 VDC 中其他 VM 之间的总带宽需求最小)作为一组,然后把这一组作为一个整体放置到某个物理服务器上,直到 VDC 被成功映射时停止迭代。

RAVDC 算法需要使用两个重要的数据结构: M 和 U, M 表示 VDC 中已映射的 VM 集合, U 表示 VDC 中待映射的 VM 集合。在完成一个 VM u ($u \in U$) 映射到服务器的工作之后,需要映射 VM u 和已映射 VM v ($v \in M$) 之间的所有虚拟链路。映射虚拟链路时,是找 u 和 v 所在服务器之间的一条满足虚拟链路带宽需求的路径。当把所有的虚拟链路映射完成之后, RAVDC 算法会把 u 加入到 M 之中。初始时, $M = \emptyset, U = A^k$ 。

RAVDC 算法的输入参数有三个,分别是 $G(V, A)$ 、 $G^k(V^k, A^k)$ 和 $R(i)$ 。

RAVDC 算法的主要步骤分为三步:

步骤 1 利用租户给定的可靠性需求 $R(i)$ 计算出虚拟机整合门限 K (如式(2)所示)。 K 表示 VDC 在单个服务器上最多能部署的 VM 个数,通过 K 可以确保单个物理服务器失效只影响租户 VDC 的少量 VM,从而保障租户 VDC 的可靠性需求。

步骤 2 基于门限 K ,从 U 中找一个大小为 K 的最好分组 P (P 中含有 VM 个数为 K)。通过式(5)来计算一个分组的好坏程度。分组越好表示分组需要消耗的带宽越少,所以找最好分组可以保证分组之间的通信量最少,从而减少物理数据中心网络的带宽消耗。

步骤 3 把得到的最好分组 P 整合映射到一个开启且满足需求的服务器上,同时映射分组 P 中 VM 与 M 中 VM 之间的所有虚拟链路。若映射失败,则开启一个新服务器来重新映射。这可以减少开启的服务器数目,从而降低数据中心服务器能耗。

RAVDC 算法循环执行步骤 2、3,直到 VDC 所有的 VM 和 VM 之间的虚拟链路全部被映射成功。通过这种机制, RAVDC 算法可以在保证可靠性的前提下,减少带宽消耗、降低服务器能耗。

其伪码如算法 1 所示。

算法 1 RAVDC 算法。

输入 $G(V, A)$ 、 $G^k(V^k, A^k)$ 和 $R(i)$ 。

- 1) 令 S_Y 表示已开启的服务器集合, $S_Y \leftarrow \emptyset$
- 2) 令 S_N 表示未开启的服务器集合, $S_N \leftarrow S$
- 3) 令 $Flag$ 表示是否已经开启了一个新服务器
- 4) $M \leftarrow \emptyset, U \leftarrow V^k, US \leftarrow S_Y, FS \leftarrow \emptyset, Flag \leftarrow \text{false}$
- 5) 根据式(2)计算 K , 令 $K^* \leftarrow K$
- 6) repeat
- 7) 令 P 表示从 U 中找到的最好分组, $P \leftarrow \emptyset, OB_P \leftarrow \infty$
- 8) 令 Z 表示分组大小小于 K^* 的小分组集合, $Z \leftarrow \emptyset$
- 9) for all $u \in U$ do
- 10) 使用贪婪图增长算法计算从 u 出发的一个大小为 K^* 的分组 P^* (P^* 的大小可能小于 K^*)
- 11) if P^* 的大小 $< K^*$ then
- 12) $Z \leftarrow Z \cup \{P^*\}$, continue
- 13) if $OB_{P^*} < OB_P$ then
- 14) $P \leftarrow P^*, OB_P \leftarrow OB_{P^*}$
- 15) if $P = \emptyset$ then
- 16) 从 Z 中组合出一个尽可能接近 K^* 的备选分组 P'
- 17) $P \leftarrow P', OB_P \leftarrow OB_{P'}$
- 18) 令 CS 表示 P 的所有候选服务器集合, $CS \leftarrow \emptyset$
- 19) for 每个未使用的服务器 $i \in US$ do
- 20) if $pl_i \geq OB_P$ and 服务器 i 满足 P 中所有 VM 的资源需求 and $i \notin FS$ then
- 21) $CS \leftarrow CS \cup i$
- 22) 对 CS 中的服务器按照 pl 升序排列
- 23) for 每个候选服务器 $i \in CS$ do
- 24) 尝试映射 P 到 i 上
- 25) 映射虚拟链路 (u, v) , $\forall u \in P, \forall v \in M$, 当 $(u, v) \in A^k$
- 26) if 映射 P 成功 then
- 27) $M \leftarrow M \cup P, U \leftarrow U \setminus P, US \leftarrow US \setminus i, Flag \leftarrow \text{false}$
- 28) break
- 29) if 映射 P 失败 then
- 30) if $K^* = 1$ then
- 31) if 所有服务器已经开启 then
- 32) VDC k 映射失败
- 33) if $Flag = \text{true}$ then
- 34) $PS \leftarrow FS \cup$ 所有 VM i ($i \in M$) 所在的服务器
- 35) $M \leftarrow \emptyset, Flag \leftarrow \text{false}, continue$
- 36) 从 S_N 中任意开启一个服务器 $i, Flag \leftarrow \text{true}$
- 37) $S_N \leftarrow S_N \setminus i, S_Y \leftarrow S_Y \cup i, K^* \leftarrow K, US \leftarrow US \cup i$
- 38) else
- 39) $K^* \leftarrow K^* - 1$
- 40) until 已映射的 VM 集合 $M = V^k$

2.1 找分组

RAVDC 算法的第二步是找分组,由于分组内部的各个 VM 之间的虚拟链路不产生带宽消耗,所以应当把相互之间带宽需求大的 VM 合并在同一个分组中。找分组的主要方法是从 U 中的每个 VM 出发,使用贪婪图增长(Greedy Graph Growing, GGGP)算法^[8]计算出一个分组 P ($P \subseteq U$),然后从所有这些分组中找 OB (OB 的计算公式如式(5)) 最小的分组作为最好分组。

GGGP 算法的基本流程是从一个 VM s 开始,先把 s 加入 P (初始时, $P = \emptyset$)中,然后计算 s 的所有在 Q ($Q = U - P$) 中的邻接 VM v 的增益 g_v (增益的计算如式(4))。 g_v 表示 VM v 从分组 Q 移动到分组 P 之后,带宽需求的减少量。接着将拥有最大增益的 VM 加入到分组 P 中。当一个 VM 被加入到分组 P 中时,同时计算它的所有在 Q 中的邻接 VM 的增益,然后从所有计算了增益的 VM 中找到具有最大增益的 VM 加入分组 P ,如此循环,直到分组 P 的大小为 K 时停止。

定义 3 VM 的增益。用数据结构 B 来表示 VM 所属分组,对于任意两个 VM u 和 v ,若 $B[u] = B[v]$,则表示 VM u 和 v 属于同一个分组;若 $B[u] \neq B[v]$,则表示 VM u 和 v 属于不同分组。在找分组过程中,VM u ($u \in U$) 不是在 P 中,就是在 Q 中。VM u 的增益 g_u 的物理意义表示当 VM u 从其所在分组移动到另一个分组时带宽需求的减少量。 g_u 的计算公式如下:

$$g_u = \sum_{(u,v) \in A^k, B[u] \neq B[v]} BW(u,v) - \sum_{(u,w) \in A^k, B[u] = B[w]} BW(u,w) \quad (4)$$

其中: $BW(u, x)$ 表示 VM u 与其邻接的 VM x 之间的虚拟链路的带宽需求; (u, x) 表示 VM u 与 VM x 之间的一条虚拟链路。

当从 U 中每个 VM 出发找到一个分组后, 接下来需要从中找一个最好分组。用符号 OB_P 来表示分组 P 的好坏程度, OB_P 的物理意义表示 P 中所有 VM 连接到 M 和 Q 中 VM 的所有虚拟链路的带宽需求之和, OB_P 的计算公式如下:

$$OB_P = \sum_{u \in P, v \in M, (u,v) \in A^k} BW(u,v) + \sum_{u \in P, w \in Q, (u,w) \in A^k} BW(u,w) \quad (5)$$

OB_P 越小, 表示分组 P 内的 VM 出分组的带宽需求越小, 这些 VM 合并部署能尽量减少服务器之间的带宽消耗, 所以分组 P 越好; 同理, OB_P 越大, 表示该分组越差。RAVDC 算法通过找 OB 最小的分组, 可以把相互之间带宽需求量大的 VM 合并映射, 从而减少数据中心的带宽消耗。

有两种情况会导致 GGGP 算法无法找到一个大小为 K 的分组 P 。一是 $|U| < K$, 此时直接把 U 作为最好分组。二是 $|U| \geq K$, 此时 U 由多个不相互交叠的小块组成, 这些小块被 M 隔离, 它们的大小均小于 K 。这种情况出现的原因是已映射的 VM 集合 M 可能会把 VDC 分割成许多零散的小块, 当从这些小块中的某个 VM 出发找一个分组时, 由于找分组不会把 M 中的 VM 加入到分组中, 所以此时找到的分组就是这些小块(也称这些分组为小分组)。此时需要从这些小分组中组合出一个大小尽可能接近 K 的备选分组 P' , 并把 P' 作为最好分组。这样组合可以提高数据中心资源的利用率。

2.2 映射分组

当找到一个最好分组之后, RAVDC 算法需要映射该分组。RAVDC 算法在映射分组时, 在每个服务器上最多只映射一个分组, 无论这个分组的大小如何。如果一个服务器上已经部署了一个分组, 则称该服务器在当前 VDC 的映射过程中已被使用, 并且在之后的映射过程中不再考虑该服务器。

RAVDC 算法使用最优匹配法来映射最好分组 P , 在映射 P 之前会先使用 OB_P 进行服务器前向检查。令 pl_i 表示服务器 i 到其连接的数据中心 TOR 交换机之间的物理链路的剩余带宽容量。服务器前向检查的详细过程如下: 首先得到当前 VDC 映射过程中开启且未使用的服务器集合 US , 然后对服务器 i ($i \in US$) 进行检查, 检查 pl_i 是否能满足 OB_P 的需求, 如果能满足需求且 i 不在禁止使用的服务器集合 FS 中, 则把 i 加入到候选服务器集合 CS 中。然后 RAVDC 对得到的 CS 按照 pl_i ($i \in CS$) 升序排列, 并从 CS 中选择具有最小 pl_i 的服务器 i 来映射 P 。如果映射失败, 则从 CS 中选择具有次小 pl_i 的服务器来映射 P 。通过使用服务器前向检查机制, 可以提前发现失败, 减少算法的运行时间。

在计算算法的运行时间时, 除了考虑算法正常执行时所花费的时间外, 还需要考虑重新映射所花费的时间。为了提高 VDC 的映射成功率, VDC 映射算法通常需要在失败时考虑重新映射。发生失败的次数越多, 算法所花费的时间越长。在 2EM 算法^[6] 中, 失败发生的主要原因在于 2EM 算法只使用待映射 VM 和与它邻接且已映射的 VM 之间虚拟链路的带宽需求来进行前向检查, 检查服务器的 pl 是否满足这些链路

的带宽需求之和。这就使得虽然 VM i 映射成功, 但在下一次映射 VM i 的邻接 VM j 时, 如果把 VM j 映射到与 VM i 所在服务器不同的服务器上, 就会很容易发生虚拟链路映射失败。失败的原因是 VM i 所在服务器的 pl 无法满足 VM i 与 VM j 之间虚拟链路的带宽需求。而 RAVDC 算法在一个服务器上只映射一个分组, 并通过找 pl 满足分组 OB 的服务器来映射分组, OB 是分组与所有剩下的 VM(包括已映射和未映射 VM)之间的虚拟链路带宽需求之和, 这就使得造成 2EM 算法大量失败的原因不会造成 RAVDC 算法失败, 从而极大减少失败发生的次数, 减少 RAVDC 算法的运行时间。

如果分组 P 映射成功, 则使用 K -shortest 算法^[6] 映射 P 中的 VM 与 M 中的 VM 之间的所有虚拟链路。 K -shortest 算法会计算所有服务器对之间的多条最短路径, RAVDC 从这些最短路径中选择剩余带宽最小且能满足虚拟链路带宽需求的一条路径来映射虚拟链路, 通过这种选择方式可以充分利用底层物理网络的带宽资源。

如果直到 CS 为空, 也不能成功映射, 则执行简单的回退机制——递减 K , 重新找一个更小的分组进行映射。通过递减 K , 可以尽最大可能地利用现有资源。如果当 K 已经递减到 1 时, 所找的分组也映射失败, 则开启一个新的服务器, 然后重新找一个大小为初始 K 值的分组并重新映射。如果开启一个新服务器仍然不能成功映射一个分组, 则认为已映射分组的映射是不合理的, 即不应该把这些分组映射到当前它们所在的服务器上。因为此时失败的原因是虽然已映射分组所在服务器 i 的 pl_i 能满足分组的 OB 需求, 但却由于服务器 i 与其他服务器之间路径上的中间链路(非服务器与 TOR 交换机之间的链路)非常繁忙, 即剩余带宽容量太少, 而不能成功找到满足该分组与后续分组之间虚拟链路带宽需求的路径。所以如果继续把已映射分组留在这些服务器上, 将不能成功映射 VDC, 最终只有拒绝该 VDC 请求。所以为了提高 VDC 映射的成功率, RAVDC 此时会把已映射分组所在服务器加入禁止使用的服务器集合 FS 中, 然后把 M 清空, 并重新在所有服务器 i ($i \notin FS$) 上再次映射 VDC。如果开启新服务器失败, 即数据中心中的所有服务器已开启, 则认为该 VDC 映射失败。RAVDC 算法通过尽量把 VM 映射到已开启的服务器上, 可以减少开启的服务器数目, 从而降低数据中心的服务器能耗。

3 仿真与分析

3.1 仿真环境设置

仿真使用 128 个服务器、80 个交换机的胖树结构^[9] 作为数据中心拓扑。

为了评估数据中心的总能耗(数据中心总能耗由交换机能耗^[10] 和服务器能耗^[11] 两部分组成), 需要为数据中心交换机和服务器设置能耗参数。使用 EP (Energy Proportionality) 为 0.5 的服务器^[12] 作为数据中心服务器。由于 CPU 能耗在服务器能耗中所占比例最大, 所以在仿真时只考虑 CPU 能耗, 服务器的参数如表 1。选用 Model D 交换机^[10] 作为数据中心边缘和汇聚交换机, 用 Model E 交换机^[10] 作为数据中心核心交换机, 交换机的参数如表 2。

表 1 服务器参数

服务器类型	固定能耗/W	vCPU 能耗/W	vCPU 数量
$EP = 0.5$ [12]	80	2.5	32

表2 交换机参数

交换机类型	端口带宽/ (Mb·s ⁻¹)	带宽能耗/ (mW·Mb ⁻¹ ·s)	固定能耗/W
Model D ^[10]	1 024	0.53	76.4
Model E ^[10]	1 024	2.1	555

VDC的拓扑从星型拓扑、mesh拓扑、二叉树拓扑和随机连通图中随机选择。VDC大小分布在[9,14]区间内。每个VM的计算资源大小(vCPU的数量)分布在[2,4]区间内;每条虚拟链路的带宽资源大小分布在[2,6]区间内,单位是10 Mb/s。

用RAVDC算法与现有的以节能为目标的2EM算法进行比较,由于原有的2EM算法不感知租户的可靠性需求,因此需要对2EM算法进行改进,让其能感知可靠性需求。为了方便区分,称改进后的2EM算法为可靠性感知的2EM(Reliability-aware 2EM, RA-2EM)算法。改进方法如下:在用2EM算法进行VDC映射时,每次映射一个VM到物理服务器之前首先判断该服务器上属于该VDC的VM数量是否已达到可靠性需求对应的门限K,如果没达到,则可以映射,否则放弃映射到该服务器。

随机产生动态到达的VDC拓扑和资源请求,并比较2EM算法、RAVDC算法和RA-2EM算法随着租户VDC可靠性需求从0.05到0.80以间隔为0.05增长时的性能变化,仿真结果如图2,由于当VDC中VM个数为9时,最大可靠性只能到达0.89(根据式(1)计算得到),所以只测到0.80。待比较的性能包括VDC实际平均可靠性(实际测得的所有VDC的总可靠性除以VDC数目)、VDC平均带宽消耗(所有VDC的总带宽消耗除以VDC数目)、VDC平均能耗(所有VDC的总能耗除以VDC数目)和VDC的平均映射时间(所有VDC映射的总时间除以VDC数目)。由于2EM算法不感知可靠性,所以在图2中,随着租户VDC可靠性需求的变化,2EM算法的性能是一条直线。最后从图2中选择租户VDC可靠性需求 $R(i)=0.3$ 进行详细分析,此时RA-2EM算法和RAVDC算法带宽消耗差值最大,结果如图3,横坐标是已映射的VDC数目,纵坐标是与RA-2EM算法相比RAVDC算法所消耗的数据中心总带宽的减少量占RA-2EM算法消耗量的百分比。

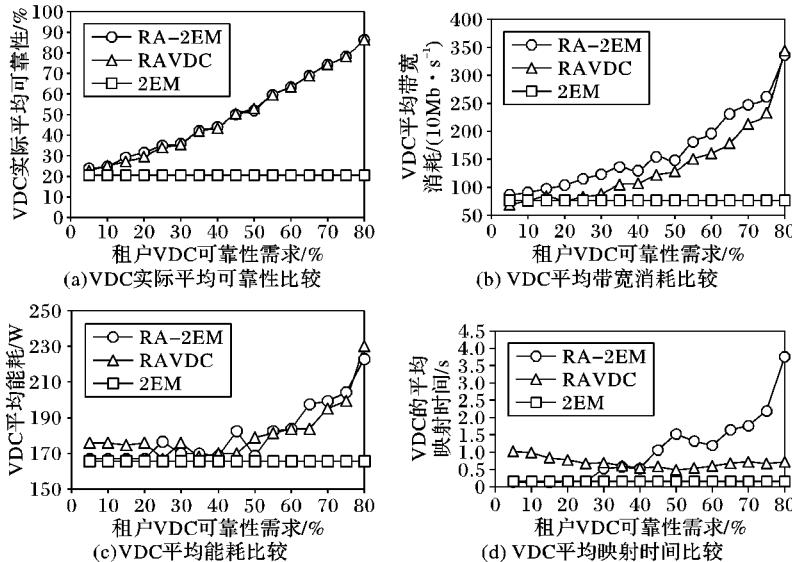
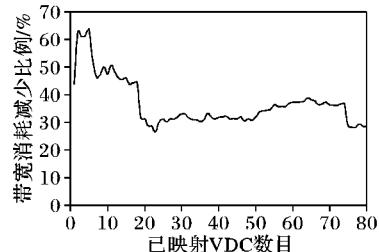


图2 2EM、RAVDC和RA-2EM算法随着租户VDC可靠性需求变化的平均性能变化

图3 $R(i) = 0.3$ 时RA-2EM和RAVDC算法之间的VDC平均带宽消耗比较

3.2 性能比较

3.2.1 VDC实际平均可靠性比较

在可靠性方面,从图2(a)中可以看出,随着租户VDC可靠性需求不断增长,2EM算法所映射VDC的实际平均可靠性很差且稳定,约为20%,原因是2EM算法尽可能集中放置VM且不感知可靠性,因此2EM算法不适用于可靠性感知的VDC映射场景;而RA-2EM和RAVDC会随着租户VDC可靠性需求的增长不断减小VM的集中程度,即不断减小K,从而保证租户VDC的可靠性需求。RA-2EM算法和RAVDC算法的VDC实际平均可靠性性能非常接近,原因是RA-2EM算法和RAVDC算法均是把 $R(i)$ 按照同样的方式转化成K,然后以K为限制进行VM映射。对于每个VDC而言,RA-2EM算法和RAVDC算法均是采用在不超过限制放置的VM个数的前提下选用最小数目的服务器来映射VDC,所以两者在可靠性上的性能几乎没有差异。除此之外,从图2(a)中还可以看出RA-2EM算法和RAVDC算法所映射VDC的实际平均可靠性均大于租户VDC的可靠性需求,这对租户而言是非常有利的。出现这个结果的原因是在把可靠性需求转化成K时,采用向下取整的方式,这就使得实际的可靠性一定大于等于租户需求的可靠性。

3.2.2 VDC平均带宽消耗比较

在带宽消耗方面,从图2(b)中可以看出,随着租户VDC可靠性需求不断增长,2EM算法所映射VDC的带宽消耗很少,约为166 Mb/s,原因是2EM算法尽可能集中放置VM;而RA-2EM和RAVDC会随着租户VDC可靠性需求的增长不断减小K,即把VM更加分散地放置在各个物理服务器上,从而使VM之间的带宽需求占用更多的物理网络带宽资源,所以

VDC的平均带宽消耗不断增加。与RA-2EM算法相比,RAVDC算法所映射VDC的平均带宽消耗会大量减少。结合图3,可以看到RAVDC算法最大可以减少平均约30%的带宽消耗。原因是RAVDC算法在每次选择分组时,尽可能将相互之间带宽需求量大的VM进行合并映射,通过这种聚合方式可以减少VDC对底层物理数据中心的带宽消耗;而RA-2EM算法并没有把带宽消耗作为优化目标。所以在带宽消耗上,与RA-2EM算法相比,RAVDC算法有着很大优势。

3.2.3 VDC平均能耗比较

在能耗方面,从图2(c)中可以看出,随着租户VDC可靠性需求不断增长,2EM算法所映射VDC的能耗较少,约为166 W,原因是2EM算法尽可能集中放置VM;而RA-2EM和RAVDC会随着租户VDC可靠性需求的增长不

断增大 VM 的分散程度,从而增加开启的服务器数目,所以 VDC 的平均能耗不断增加。RA-2EM 算法和 RAVDC 算法的 VDC 平均能耗相差不大,趋势相似,原因是 RA-2EM 算法和 RAVDC 算法都主要通过减少开启的服务器数目来降低能耗。虽然 RA-2EM 算法尝试优化带宽能耗,但是 RA-2EM 算法和 RAVDC 算法在带宽能耗上的差异不大,原因是在实际的数据中心中带宽能耗极小,其在交换机总能耗中所占比例不会超过 5%^[10],所以带宽能耗不会影响两个算法的能耗性能。2EM 算法是专门为减少能耗而提出的 VDC 映射算法,虽然 RA-2EM 算法的能耗性能不如 2EM,但是它并没有改变 2EM 算法节能的核心思想,只增加了 VM 放置数目的限制来满足租户 VDC 的可靠性需求,所以在可靠性感知的问题中,RA-2EM 算法仍然在节能方面有着很高的效率。本文提出的 RAVDC 算法在能耗方面能和 RA-2EM 算法基本接近,表明 RAVDC 算法在能耗方面也有很高的效率。

3.2.4 VDC 平均映射时间比较

从图 2(d)中可以看出,随着租户 VDC 可靠性需求不断增长,2EM 算法的 VDC 平均映射时间很短,约为 158 ms;而 RA-2EM 算法的时间变化起伏很大,在租户 VDC 可靠性需求很小时,VDC 的平均映射时间与 2EM 算法相似,随着租户 VDC 可靠性需求不断增长,VDC 的映射时间也不断增大。出现这种现象的原因是在可靠性需求很小时,大量的 VM 集中映射,大部分虚拟链路的带宽需求通过 hypervisor 完成,这就使得物理链路都很空闲,从而使得虚拟链路的映射几乎不会发生失败,所以 2EM 和 $R(i)$ 较小时的 RA-2EM 算法的 VDC 平均映射时间相同且很小;随着租户可靠性需求不断增加,VM 的部署更加分散,这就使得需要映射的虚拟链路数目增多,物理网络带宽占用高,所以更容易出现两个服务器之间没有任何一条路径能满足虚拟链路带宽需求的现象,即发生失败的次数急剧增加,所以 RA-2EM 算法随着租户 VDC 可靠性需求不断增加,VDC 的平均映射时间急剧增大。而 RAVDC 通过找 pl 满足分组 OB 的服务器来映射分组,就可以大量减少失败发生的次数,从而减少 VDC 的映射时间。所以从图 2(d)中可以看出,RAVDC 算法的 VDC 平均映射时间非常稳定,而且在可靠性需求变高时,也不会增加算法的运行时间。RAVDC 算法在租户 VDC 可靠性需求很少时,由于 K 值较大,GGGP 的运行时间较长,因此此时 RAVDC 算法的 VDC 平均映射时间相对较多。

通过 RAVDC 算法与 2EM 算法和 RA-2EM 算法进行比较,可以得出以下结论:1)2EM 算法不适用于可靠性感知的 VDC 映射问题;2)RA-2EM 算法虽然能保证租户 VDC 的可靠性需求,但是在租户 VDC 可靠性需求较高时,VDC 平均映射时间急剧增加,所以不适用于租户 VDC 可靠性需求较高的场景;3)RAVDC 算法的 VDC 平均映射时间稳定,在任何可靠性需求下,VDC 平均映射时间不超过 1 s;4)RAVDC 算法在不损失 VDC 可靠性和能耗性能的前提下,最多能减少约 30% 的数据中心网络带宽消耗。

4 结语

本文研究了可靠性感知下的 VDC 映射问题,并提出 RAVDC 算法来解决该问题。RAVDC 算法通过三个阶段依次完成目标:首先满足可靠性需求,然后降低带宽消耗,最后降低能耗。仿真结果表明,与 2EM 算法和 RA-2EM 算法相比,该算

法在满足租户 VDC 可靠性需求的前提下,能减少数据中心网络最多约 30% 的带宽消耗而不增加额外的能耗,并且能在平均不超过 1 s 的时间内完成任意可靠性需求下的 VDC 映射。

未来在该领域的研究方向主要包括:1)跨多数据中心的可靠性感知下的 VDC 映射。本文提出的 RAVDC 算法仅仅用于解决单数据中心内可靠性感知下的 VDC 映射问题,但是目前数据中心之间往往可以进行资源共享,所以跨多数据中心的可靠性感知下的 VDC 映射算法还有待研究。2)可靠性未知条件下的 VDC 映射。本文提出的 RAVDC 算法用于解决在可靠性已知条件下如何减少带宽消耗的问题,在未来的研究中可靠性可能不会由租户自己给出,而是由数据中心管理者去权衡,所以可靠性未知条件下的 VDC 映射算法还有待研究。

参考文献:

- [1] YUAN H, KUO C-C J, AHMAD I. Energy efficiency in data centers and cloud-based multimedia services: an overview and future directions [C]// Proceedings of the 2010 International Conference on Green Computing. Piscataway: IEEE, 2010: 375 – 382.
- [2] BARI M F, BOUTABA R, ESTEVES R, et al. Data center network virtualization: a survey [J]. IEEE Communications Surveys & Tutorials, 2013, 15(2): 909 – 928.
- [3] FISCHER A, BOTERO J F, TILL BECK M, et al. Virtual network embedding: a survey [J]. IEEE Communications Surveys & Tutorials, 2013, 15(4): 1888 – 1906.
- [4] GUO C, LU C, WANG H J, et al. SecondNet: a data center network virtualization architecture with bandwidth guarantees [C]// Co-NEXT'10: Proceedings of the 6th International Conference on Emerging Networking Experiments and Technologies. New York: ACM, 2010: Article No. 15.
- [5] FUERST C, SCHMID S, FELDMANN A. On the benefit of collocation in virtual network embeddings [C]// CloudNet 2012: Proceedings of the IEEE 2012 International Conference on Cloud Networking. Piscataway: IEEE, 2012: 161 – 163.
- [6] LUO L, KAI Y, YU H, et al. Energy-efficient virtual data center mapping [C]// ACP 2013: Proceedings of the 2013 Asia Communications and Photonics Conference. Washington, DC: Optical Society of America, 2013: ATh3I.5.
- [7] BODÍK P, MENACHE I, CHOWDHURY M, et al. Surviving failures in bandwidth-constrained datacenters [C]// Proceedings of the ACM SIGCOMM 2012 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication. New York: ACM, 2012: 431 – 442.
- [8] KARYPIS G, KUMAR V. A fast and high quality multilevel scheme for partitioning irregular graphs [J]. SIAM Journal on Scientific Computing, 1998, 20(1): 359 – 392.
- [9] AL-FARES M, LOUKISSAS A, VAHDAT A. A scalable, commodity data center network architecture [C]// Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication. New York: ACM, 2008: 63 – 74.
- [10] MAHADEVAN P, SHARMA P, BANERJEE S, et al. A power benchmarking framework for network devices [C]// NETWORKING '09: Proceedings of the 8th International IFIP-TC 6 Networking Conference. Berlin: Springer, 2009: 795 – 808.
- [11] YE K, WU Z, JIANG X, et al. Power management of virtualized cloud computing platform [J]. Chinese Journal of Computers, 2012, 35(6): 1262 – 1285. (叶可江, 吴朝晖, 姜晓红, 等. 虚拟化云计算平台的能耗管理[J]. 计算机学报, 2012, 35(6): 1262 – 1285.)
- [12] RYCKBOSCH F, POLFLIET S, EECKHOUT L. Trends in server energy proportionality [J]. Computer, 2011, 44(9): 69 – 72.