CS-466 CLOUD COMPUTING

VIRTUAL NETWORK EMBEDDING FOR CONTENT DELIVERY NETWORKS

Submitted by:
Nihar KG Rai (181CO235)
Nishtha Kumari (181CO236)
Dhiraj Lokesh(181CO118)

TABLE OF CONTENTS

Abstract

Network virtualization is a powerful way to run multiple architectures or experiments simultaneously on a shared infrastructure. However, making efficient use of the underlying resources requires effective techniques for virtual network embedding—mapping each virtual network to specific nodes and links in the substrate network. Since the general embedding problem is computationally intractable, past research restricted the problem space to allow efficient solutions, or focused on designing heuristic algorithms. In this paper, we have proposed a way of dealing with the VN embedding algorithm which is called VNE-CDN algorithm. The purpose of our algorithm is two aspects. Firstly, we introduce centralized node importance criteria such as degree centrality and closeness centrality to measure the average distance, and then we will choose the virtual node with the largest bandwidth and CPU resources to map. In order to further accelerate content acquisition, we take into account the distance between virtual network and content server.

Introduction

Below we have introduced certain important concepts with respect to VNE to aid the understanding of this report.

Virtualization

Virtualization is a technique of how to separate a service from the underlying physical delivery of that service. It is the process of creating a virtual version of something like computer hardware. It was initially developed during the mainframe era. It involves using specialized software to create a virtual or software-created version of a computing resource rather than the actual version of the same resource. With the help of Virtualization, multiple operating systems and applications can run on same machine and its same hardware at the same time, increasing the utilization and flexibility of hardware.

Network Virtualization

The ability to run multiple virtual networks with each has a separate control and data plan. It co-exists together on top of one physical network. It can be managed by individual parties that potentially confidential to each other.

Network virtualization provides a facility to create and provision virtual networks—logical switches, routers, firewalls, load balancer, Virtual Private Network (VPN), and workload security within days or even in weeks.

Virtual Network Embedding

In this section, we first describe the general VN embedding problem. Then, we present an original solution to this problem without assuming that the substrate can split a
virtual link over multiple underlying paths.

Virtual Network Embedding Problem

Substrate network. We denote the substrate network by an undirected graph Gs = (Ns, Ls, As N , As L), where Ns and Ls refer to the set of nodes and links, respectively. We use superscript to refer to substrate or virtual network, and use subscript to refer to nodes or links, unless otherwise specified. Substrate nodes and links are associated with their attributes, denoted by As N and As L, respectively. In this paper, we consider CPU capacity and location for node attributes, and bandwidth capacity for link attributes. We also denote by Ps the set of all loop-free paths in the substrate network. The right side of Figure 1 shows a substrate network. The numbers near the links represent available bandwidths and the numbers in rectangles are the available CPU resources at the nodes.

Virtual network request

We denote by an undirected graph Gv = (Nv, Lv, Cv N , Cv L) a virtual network request. A VN request typically has link and node constraints that are specified in terms of attributes of the substrate network. We denote by Cv L and Cv N the set of link and node constraints, respectively. Figure 1 depicts two VN requests: the VN request 1 requires the bandwidth 20 over the links (a, b) and (a, c), and the CPU resource 10 at all nodes, a, b, and c; the VN request 2 is: “connect two nodes d, e ∈ Nv with constraints that node d should be in Atlanta (where substrate nodes D and G are located), and node e should be in New Jersey (where substrate nodes E and I are located), with ten units of bandwidth on the virtual link between them.”

VN embedding

A virtual network embedding for a VN request is defined as a mapping M from Gv to a subset of Gs, such that the constraints in Gv are satisfied, i.e., M : Gv → (N , P , RN , RL), where N ⊂ Ns and P ⊂ Ps, and RN and RL are the node and link resources allocated for the VN requests. The VN network embedding can be naturally decomposed into node and link mapping as follows: Node Mapping: MN : (Nv, Cv N) → (N , RN), Link Mapping: ML : (Lv, Cv L) → (P , RL). The right side of Figure 1 shows the VN embedding solutions for the two VN requests. For example, the nodes a, b, and c in VN request 1 are mapped to the substrate nodes A, E, and F, and the virtual links (a, b) and (a, c) are mapped to the substrate paths (A,D,E) and (A,D,F) with the CPU and bandwidth constraints all satisfied. A similar mapping occurs for VN request 2

Content Delivery Network

A content delivery network (CDN) refers to a geographically distributed group of servers which work together to provide fast delivery of Internet content.

A CDN allows for the quick transfer of assets needed for loading Internet content including HTML pages, javascript files, stylesheets, images, and videos. The popularity of CDN services continues to grow, and today the majority of web traffic is served through CDNs, including traffic from major sites like Facebook, Netflix, and Amazon.

A properly configured CDN may also help protect websites against some common malicious attacks, such as Distributed Denial of Service (DDOS) attacks.

Outline of this report

In this report, we shall be discussing network model and our objectives of VNE based on CDN, node important criteria including the centralize node and substrate node, details of VNE-CDN algorithm and experimental results.

Network Model

A. Content Delivery Network

We model the content delivery network (CDN) as a weighted undirected graph and it can be denoted as content delivery network G = (N,, E,,C, C(), B() , P) where N represents the set of substrate nodes, E represents the set of substrate links, C represents our content server. For any substrate node, N has a symbol C() represents the CPU capacity and storage capacity. For each substrate link e(i, j), there is a symbol B() represents the link's resources where i and j represent the source node and the destination node of link. Respectively, the path set of all substrate nodes are represented by P, and P (,) , represents the set of all path from the source's node to the destination node. The lower of Fig.1 shows the CDN infrastructure. The number in the rectangular boxes represent the total capacity and the remaining capacity of substrate nodes, respectively. The numbers alongside the links represent the total bandwidth resource and the remaining bandwidth resource.

B. The Model of Virtual Network Request

We model VN requests as weighted undirected graphs and denote a VN request as G = (N, E, C(), B()) where N represents the set of virtual nodes, E represents the set of the virtual links. For any node $n \in N$, there is a symbol C() represents the CPU the resources constraint set of virtual nodes. For each link e (i, j) $\in$ E, there is a symbol B() represents the resources constraint set of virtual links. The upper of fig.1 shows the virtual network, where in the rectangular box represents the CPU capacity of virtual network 1, we can see the virtual node symbolized by (a) is 20 units. (b) is 20 units, (c) is 15 units. The bandwidth requirement link symbolized by (a, b) is 5 units, (b, c) is 10 units and (a, c) is 5 units. Respectively, virtual network 2 virtual node symbolized by (d) is 20 units. (e) is 20 units, (f) is 15. The bandwidth requirement link symbolized by (d, e) is 10 units, (e, f) is 5 units and (d, f) is 5 units.

C.The Problem of Virtual Network Embedding

In the definition of the problem, each resource, either virtual or substrate, has one or more parameters. Node and link parameters are attributes that refer to nodes and links, respectively. These parameters are extremely important in order to obtain a valid embedding. An example of a virtual node parameter is the CPU capacity demand, i.e., the computational capacity that this node should have to run some workload. An example of a substrate node parameter is the CPU capacity, i.e., the processing capabilities the node can provide to one or more virtual nodes that are mapped on it. The same stands for virtual links and substrate links: virtual links may demand a certain bandwidth, while substrate links have some maximum bandwidth that can be used by the virtual links mapped on it. Therefore, Virtual network embedding can be divided into node and link mapping as follow.

The Objective of VNE Based on CDN

Below are some evaluation criteria for the performance of the virtual network embedding based on content delivery network.

1. The Speed of Contents to the user

S = distance/time

Where S represents the speed (meters/second), D represents the distance has been travelled (meters), and T represents Time (seconds).

2. Revenue and Cost Ratio

The revenue and cost ratio is one of the most important evaluation criteria for measuring the performance of a virtual network embedding algorithm. The main goal is to maximize the revenue of InPs and increase the resources utilization of substrate networks by accommodating as many VN requests as possible.

3. Request of Acceptance Ratio

The acceptance ratio of virtual network requests is also one of the most important criteria for evaluating the performance of a virtual network embedding algorithm. It represents how many requests are successfully embedded in the entire number of requests. It is defined by the following formula:

Acceptance Ratio = count of successful request / count of all request

Node Importance Criteria

A. Centralized Node Importance Criteria

1. **Degree Centrality**
   Degree centrality reflects the ability of a node to communicate directly with other nodes. The greater the value of degree centrality, the more important the node is.

2. **Closeness Centrality**
   Closeness centrality can be treated as a measurement of a node's importance through the average spread time of information in the network. The greater the value of closeness centrality, the more important the node.

B. Substrate Node Importance Criteria

It is the main optimization method of the formula in this paper. It improves the definition of node residual resources in the step of calculating substrate node residual resources (SRR) in the node mapping stage.

The VNE Algorithm Based on CDN

A. Node Mapping Stage

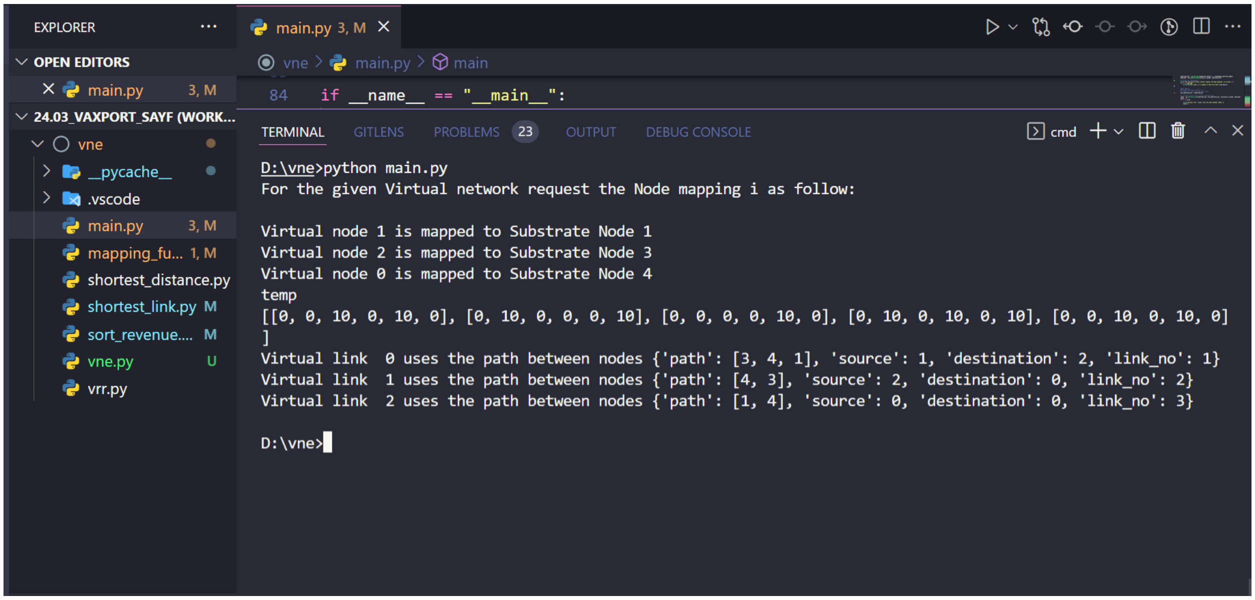
- measure the centralized node by comparing 2 types of centralize node such as: degree centrality, closeness centrality and then we select the biggest virtual node by considering the biggest bandwidth and CPU resource by using VRR formula
- select the substrate node which has the biggest bandwidth and CPU resource by using the SRR formula.
- the closer substrate node is mapped onto the content server.

B. Link Mapping Stage

- After node-mapping has been completed, we apply the k-shortest method. The target is to find the shortest path which has enough bandwidth resources.
- Firstly, we have to remove all the substrate links directly connected to the content server. At the same time, finding the shortest paths which is satisfied the virtual link constraint. if the path meets the constraints, we recover all the substrate links which are directly connected to the content server.
- Next, we search the k-shortest path in order to be mapped substrate node with the biggest resources to get closer to content server.
- Finally, we have to update the resources substrate link and content server for other VNRs.

Simulation

We simulate the process of node-mapping and edge-mapping from the virtual network to the substrate (physical) network using a Python project that handles all the processes required for the same. Unfortunately, we were not able to test our algorithm on a VNE simulator like ALib, and hence we will be presenting our results on test inputs to our Python algorithm.


EXPLORER
OPEN EDITORS
main.py 3, M
24.03_VAXPORT_SAYF (WORK...
vne
__pycache__
.vscode
main.py 3, M
mapping_fu... 1, M
shortest_distance.py
shortest_link.py M
sort_revenue.... M
vne.py U
vrr.py
main.py 3, M
vne main.py main
84 if __name__ == "__main__":
TERMINAL GITLENS PROBLEMS 23 OUTPUT DEBUG CONSOLE
cmd
D:\vne>python main.py
For the given Virtual network request the Node mapping i as follow:
Virtual node 1 is mapped to Substrate Node 1
Virtual node 2 is mapped to Substrate Node 3
Virtual node 0 is mapped to Substrate Node 4
temp
[[0, 0, 10, 0, 10, 0], [0, 10, 0, 0, 0, 10], [0, 0, 0, 0, 10, 0], [0, 10, 0, 10, 0, 10], [0, 0, 10, 0, 10, 0]
]
Virtual link 0 uses the path between nodes {'path': [3, 4, 1], 'source': 1, 'destination': 2, 'link_no': 1}
Virtual link 1 uses the path between nodes {'path': [4, 3], 'source': 2, 'destination': 0, 'link_no': 2}
Virtual link 2 uses the path between nodes {'path': [1, 4], 'source': 0, 'destination': 0, 'link_no': 3}
D:\vne>