ECM5605 (5085) S'21: Homework 01

!!! DUE: 2022/02/17 23:59:59 !!!

[Problem 1] (100%) Service Chain Deployment Problem

What is service chain?

Service Chain (SC) is a set of linked network functions, which are provided by specific virtualized network functions (VNFs) or services, with a defined direction for traffic flow and defined ingress and egress points. The services could be firewall, encryption/decryption, spam detector… etc., and must serve in a specific order in a service chain.

Figure 1 shows an example of two service chains and four kinds of services: firewall, antivirus, parental control and video optimizer. For the web traffics, the service chain includes firewall, antivirus and parental control; while for the video traffics, the service chain includes firewall and video optimizer.

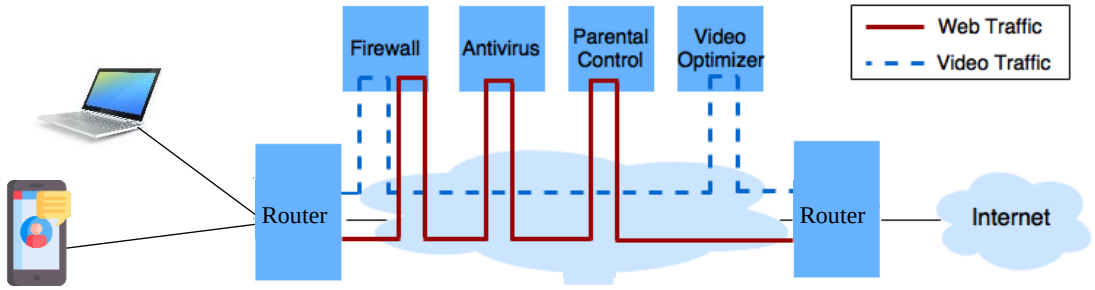

Figure 1. An example of service chains[1]

Service deployment in datacenters for service chains

Let's say that you are a CEO of a technology company who owns couple datacenters, and your task is to deploy the services to satisfy the subscripted service chains with the lowest costs (i.e., the bandwidth you need to purchase to connect between datacenters).

Figure 2 shows an example with three datacenters and a deployment of the two service chains in Figure 1. For the video traffic, because both of the services are deployed in the same datacenter, there is no need to purchase any bandwidth to

[1] Modified from: S. Kulkarni, M. Arumaithurai, K. K. Ramakrishnan and X. Fu, "Neo-NSH: Towards scalable and efficient dynamic service function chaining of elastic network functions," 2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN), 2017, pp. 308-312, doi: 10.1109/ICIN.2017.7899429.

connect between datacenters. However, for the web traffic, you need to purchase 1 Mb/s for each link to connect between (firewall, antivirus) and (antivirus, parental control) deployed in three separate datacenters. Therefore, the total cost of this deployment is 1 Mb/s + 1 Mb/s = 2 Mb/s.

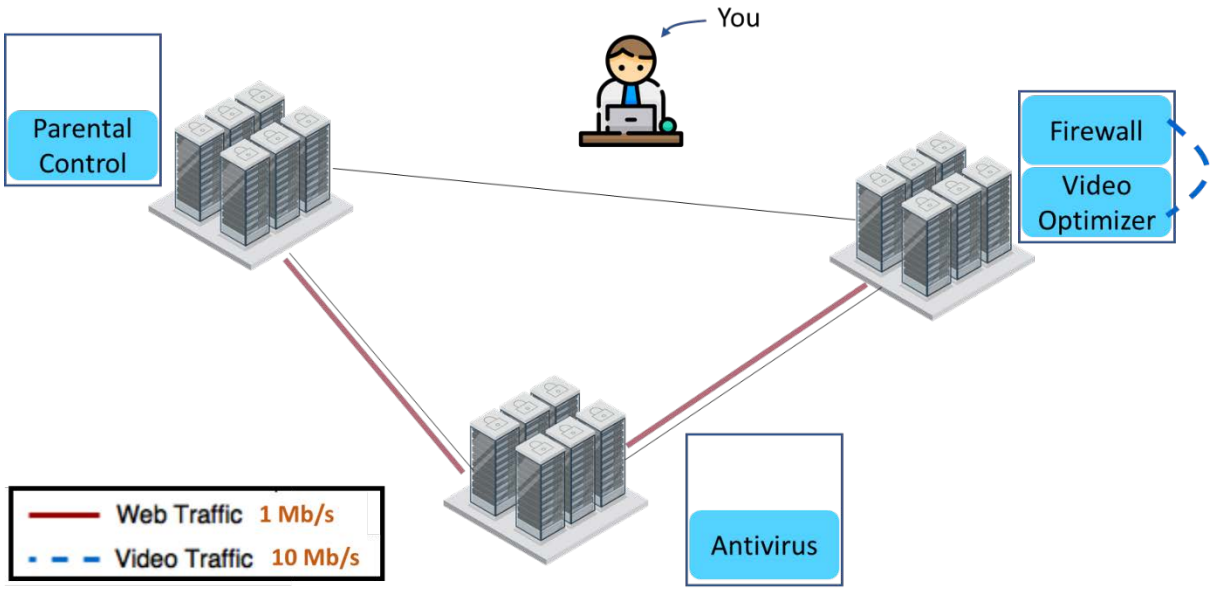

Figure 2. An example of service chain deployment

As the CEO of this company, you have several requirements for the deployment:

1. Each service is deployed in one datacenter (i.e., a service cannot be deployed in two or more datacenters).
2. Each datacenter only accepts a given number of services.
3. Your ultimate goal is to minimize the bandwidth cost between datacenters.

[Requirement]

In this homework, you are going to design an algorithm to deploy service chains and **minimize the bandwidth cost** under the following constraints.

1. Each service is deployed in only one datacenter.
2. For given **k** datacenters, **n** services and **c** (imbalance tolerance rate), the number of services that can be deployed in each datacenter is at least $\left\lfloor \frac{n}{ck} \right\rfloor$ and at most $\left\lceil \frac{cn}{k} \right\rceil$, where $c \geq 1$. (For example, if k=10, n=100 and c=2.0, then each datacenter can have 5 to 20 services.)
3. Other given constraints on the input: $\left\lceil \frac{cn}{k} \right\rceil > 1$, k > 1, n > k, t > 0

Input file <.in>

Format:

<**k** datacenter> <**n** services> <**c** imbalance tolerance rate> <**t** number of SC>
Chain 1: <bandwidth> <service number> <service 1 index> <service 2 index> …
Chain 2: <bandwidth> <service number> <service 1 index> <service 2 index> …
…

Example:

```
$ cat test1.in
3 6 1 3                 // k=3, n=6, c=1.0, t=3
7.2 2 0 3               // First service chain with two services
9.3 4 2 3 4 5           // Second service chain with four services
6.1 7 1 3 4 5 0 1 2     // Third service chain with seven services
```

Please note that bandwidth is a floating number between 1 and 100, and the number of services per service chain is between 1 and 20.

Output file <.out>

Format:

Datacenter 0: <service number> <service index 1> <service index 2> …
Datacenter 1: <service number> <service index 1> <service index 2> …
Datacenter 2: <service number> <service index 1> <service index 2> …
…

Example:

```
$ cat test1.out
2 0 1       // Datacenter 0 has 2 services: 0 and 1
2 2 3       // Datacenter 1 has 2 services: 2 and 3
2 4 5       // Datacenter 2 has 2 services: 4 and 5
```

Notes that you need to be aware of:

1. You should implement your code in servie_chain_deployment() in **your_id_hw1.cpp** and rename it as "**<your student id>_hw1.cpp**" (e.g., 0580706_hw1.cpp)
2. Your .out file must have the same name as the .in file (i.e., your algorithm should generate **test1**.out for the input **test1**.in).
3. For the given **n** services, although some of the services might not be used by any service chain, you still need to allocate those services into one of the

datacenters. The allocation of those unused services will not affect the cost.

4. Your code should be able to compile with
   $g++ -std=c++17 validate_code.cpp *_hw1.cpp -o validate_code

Things you need to provide:

1. Your code in C++ named by "**<your student id>_hw1.cpp**" with the function **servie_chain_deployment** implemented
   (e.g., 0580706_hw1.cpp)
2. A simple report describing how you design your algorithm named by "**<your student id>_hw1_report.pdf**". (e.g., 0580706_hw1_report.pdf)
3. Don't zip your files

Please note that wrong file name and file format will result in a zero for this homework.

How we evaluate your code:

1. Correctness: Check whether your deployment satisfies the constraints
   a. Each service is deployed in only one datacenter
   b. Each datacenter has least $\frac{n}{ck}$ and at most $\frac{cn}{k}$ services
2. Effectiveness (80%): Check whether your algorithm does improve the deployment
   a. Your deployment cost must be lower than the benchmark (i.e., the average of random deployment).

   We will accept your answer only when your result is correct and effective. (Obtain 70 points)
3. Execution time (10%)
   You are **NOT encouraged** but welcome to use parallel programming. Note that we will accumulate CPU clocks in all the threads/tasks you use.
   *Your code will be terminated after running for **5 minutes**, so please be aware of the execution time.
4. Memory Usage (10%)
   We will measure the resident set size, which is your RAM usage including your code size. *RSS = Heap size + MetaSpace + OffHeap size.*
   *Your code will be terminated if you try to use too much RAM, so please be aware of the memory usage.

You score will be based on your ranking on the effectiveness, the execution time and the memory usage among your classmates. (Partial of 30 points)

Good luck and happy holiday ~