The draft for the cost minimization of a federated cloud participant

***Abstract*—We plan to consider the placement and migration of virtual machines under the dynamic request arrival scenario in the cloud federation platform. The objective is to minimize the cost of an autonomy cloud.**

I. System Model

We study the problem that how an individual cloud provider in the cloud federation minimizes its total cost. The cost of an individual cloud provider includes: the operational cost related to the number of running servers; the cost of buying virtual machines from other providers; the cost of data transferring in other providers; the migration cost.

We consider a cloud federation with J cloud providers in total. Each cloud in the federation provides M types of instances(i.e., virtual machines). Each type of VM corresponds to a set of configurations of CPU, storage and memory.We suppose each cloud $j \in [1, J]$ has N_m^j homogeneous servers configured to provide VMs of type m, each of which can provide at most H_m^j VMs of this type. The total number of servers in cloud j is $\sum_{m=1}^{M} N_m^j$.

The price for type m VM of cloud j is p_m^j per unit time. The price for cloud j transferring data out or in is p_{do}^j or p_{di}^j per unit volume.

A. Service demands

Each individual cloud receives job requests independently. There are K types of jobs. Job type $k \in [1, K]$ can be denoted by a tuple $< m_k, v_k, w_k >$. Here, m_k represents job type k requests type m_k VMs. v_k is the number of requested VMs. VMs are labeled from 1 to v_k. w_k is the request service time, i.e., the time duration that requested VMs should run. An individual job l in any job type is associated with a traffic matrix T^l. T^l is a $v_{k_l} \times v_{k_l}$ matrix. The entry of row r and column s, $T_{r,s}^l$ is the traffic from r-th VM to s-th VM.

The system operates in a time slotted manner. Let $t = 0, 1, 2, 3..., T$ be the time slots. Let $\mathcal{A}_k^j(t)$ be the set of type k jobs arriving at time slot t in cloud j, $|\mathcal{A}_k^j(t)| = A_k^j(t)$.

Let $U_k^j(t)$ denote the number of newly served type k jobs in cloud j. Then, the number of type k jobs at cloud j newly served at time slot $t - w, 0 \leq w \leq w_k - 1$ is $U_k^j(t - w), 0 \leq w \leq w_k - 1$. Index the jobs receiving service at time slot $t - w$ from 1 to $U_k^j(t - w)$. Let $\mathcal{L}^j(t)$ be the set of all jobs from cloud j being served in the federation at time slot t, $|\mathcal{L}^j(t)| = \sum_{k=1}^{K} \sum_{w=0}^{w_k-1} U_k^j(t - w)$ A job l in $\mathcal{L}^j(t)$ can be denoted by a 3-tuple (k_l, t_l, h_l), which means the job is the h_l-th type k_l job that received service at time slot t_l.

For each type of job, we use a queue $Q_k^j(t)$ to denote the workload of job type k at cloud j, here the workload refers to the total remaining time slots needed for one type of jobs. The cloud federation guarantees to bound the service responsive time within a delay of d, i.e., the time span from when the job arrives to when it starts to run on scheduled VMs. When a job's service responsive time can not be bounded with d, it is dropped. Let $D_k^j(t) \in [0, D_k^{j(max)}]$ denote the number of type-k jobs dropped by cloud j at time slot t. $D_k^{j(max)}$ is the maximum value of $D_k^j(t)$. Let α_k^j be the penalty to drop one such job.

The dynamic of $Q_k^j(t)$ is as follows:

$$Q_k^j(t+1) = \max\{Q_k^j(t) - \sum_{w=0}^{w_k-1} U_k^j(t-w) - w_k \cdot D_k^j(t), 0\} + w_k \cdot A_k^j(t), 1 \leq j \leq J, 1 \leq k \leq K. \quad (1)$$

Job scheduling satisfies the following SLA constraint:

Each job is either served or dropped (subject to a penalty) before the maximum response delay d. (2)

B. VM provision and placement

Each cloud j needs to place the VMs of all running jobs $\mathcal{L}^j(t)$ in the federation. Let $r_m^{ji}(t)$ denote the number of type m virtual machines that cloud j places in cloud i at time slot t. Let $n_m^j(t)$ be the number of active servers hosting type m VMs in cloud j.

The total number of type m VMs running in cloud j satisfy the following two constraints:

$$\sum_{i=1}^{J} r_m^{ij}(t) \leq n_m^j(t) \cdot H_m^j, \quad n_m^j(t) \leq N_m^j, \quad 1 \leq j \leq J, 1 \leq m \leq M. \quad (3)$$

$$\sum_{k:m_k=m} \sum_{w=0}^{w_k-1} v_k \cdot U_k^j(t-w) \leq \sum_{i=1}^{J} r_m^{ji}(t), \quad 1 \leq j \leq J, 1 \leq m \leq M. \quad (4)$$

Constraint (3) means all type m VMs placed on cloud j should be no larger than the maximum number of VMs it can host. Constraint (4) means the total number of VMs owned by cloud j should be larger than the required VMs from its running jobs.

Hence, cloud j could place at most $r_m^{ji}(t)$ type m VMs in cloud i. For each individual running job in cloud j, $l \in \mathcal{L}^j(t)$, let $I_{l,s}^i(t)$ be the indicator whether the VM $s \in [1, v_{k_l}]$ is placed in cloud i or not at time slot t.

$$I_{l,s}^i(t) = 1 \text{ if instance } s \text{ is placed in cloud } i \text{ at time slot } t.$$
$$I_{l,s}^i(t) = 0 \text{ if not.} \quad (5)$$

The number of type m instances that is placed by cloud j in cloud i is $\sum_{l \in \mathcal{L}^j, m_{k_1}=m} \sum_{s=1}^{v_{k_l}} I_{l,s}^i(t), 1 \le i \le J$.

$$\sum_{l \in \mathcal{L}^j, m_{k_1}=m} \sum_{s=1}^{v_{k_l}} I_{l,s}^i(t) \le r_m^{ji}(t), i, j \in [1, J], m \in [1, M] \quad (6)$$

$$\sum_{i=1}^{J} I_{l,s}^i(t) = 1, l \in \mathcal{L}, s \in [1, v_{k_l}] \quad (7)$$

Constraint (6) means the placement of cloud j's type m VMs in cloud i can not exceed $r_m^{ji}(t)$. Constraint (7) means any VM from running jobs is placed.

C. Data traffic among clouds

Let us consider the data traffic among different cloud providers in the federation due to the placement and migration of VMs. There are two parts of traffic: one is the data traffic among different VMs of the same job; the other is the traffic induced by VM migrations.

First, we consider the traffic due to traffic among VMs of jobs. The traffic transferring out of cloud j due to jobs of cloud i is $\sum_{l \in \mathcal{L}^i} \sum_{s_2 != s_1} I_{l,s_1}^j \cdot (1 - I_{l,s_2}^j) \cdot T_{s_1,s_2}^l$.

The traffic transferring into cloud j due to jobs of cloud i is $\sum_{l \in \mathcal{L}^i} \sum_{s_2 != s_1} (1 - I_{l,s_1}^j) \cdot I_{l,s_2}^j \cdot T_{s_1,s_2}^l$

Next, let us consider the traffic due to migration. We assume the VM downtime during migration is short and can be ignored. We also assume the bandwidth used for migration is large, and the migration time can be ignore compared to the time slot. Let B_m be the size of transferred data for migrating a type m instance. We consider the transferred data due to the VM migration. The migration traffic out of cloud j due to migration of jobs in cloud i can be calculated as $\sum_{l \in \mathcal{L}^i} \sum_{s=1}^{v_{k_l}} [I_{l,s}^j(t-1) - I_{l,s}^j(t)]^+ \cdot U_{m_{k_l}}$

The migration traffic into cloud j due to migration of jobs in cloud i is $\sum_{l \in \mathcal{L}^i} \sum_{s=1}^{v_{k_l}} [I_{l,s}^j(t) - I_{l,s}^j(t-1)]^+ \cdot U_{m_{k_l}}$.

The data volume transferring out of cloud j due to jobs of cloud i is:

$$G_{out}^{ji}(t) = \sum_{l \in \mathcal{L}^i} \sum_{s_2 != s_1} I_{l,s_1}^j \cdot (1 - I_{l,s_2}^j) \cdot T_{s_1,s_2}^l + \sum_{l \in \mathcal{L}^i} \sum_{s=1}^{v_{k_l}} [I_{l,s}^j(t-1) - I_{l,s}^j(t)]^+ \cdot U_{m_{k_l}}$$

The data volume transferring into cloud j due to jobs of cloud i is:

$$G_{in}^{ji}(t) = \sum_{l \in \mathcal{L}^i} \sum_{s_2 != s_1} (1 - I_{l,s_1}^j) \cdot I_{l,s_2}^j \cdot T_{s_1,s_2}^l + \sum_{l \in \mathcal{L}^i} \sum_{s=1}^{v_{k_l}} [I_{l,s}^j(t) - I_{l,s}^j(t-1)]^+ \cdot U_{m_{k_l}}$$

The data transfer should satisfy the bandwidth constraints:

$$\sum_{i=1}^{J} G_{out}^{ji}(t) \le B_{out}^j$$
$$\sum_{i=1}^{J} G_{in}^{ji}(t) \le B_{in}^j \quad (8)$$

B_{out}^j is the upload bandwidth limit of cloud j. B_{in}^j is the download bandwidth limit of cloud j.

D. Cost minimization problem definition

We first formulate the cost minimization problem for one cloud provider in the federation.

Let us first consider the cost of cloud j in the federation. The total cost is the operational cost for running servers and transferring data plus the cost for buying VMs or transferring data from other cloud providers in the federation minus the income for selling VMs to other cloud providers or transferring other cloud providers' data.

The time-averaged operational cost for running servers is related to the number of active servers:

$$C_1^j = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{\beta^j(t) \cdot \sum_{m=1}^{M} n_m^j(t)\}, j \in [1, J]$$

The time-averaged network cost related to transferring data out and in is :

$$C_2^j = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{\beta_{do}^j(t) \cdot \sum_{i=1}^{J} G_{out}^{ji}(t) + \beta_{di}^j(t) \cdot \sum_{i=1}^{J} G_{in}^{ji}(t)\}, \quad j \in [1, J]$$

The cost due to penalty of dropping jobs is

$$C_3^j = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{\sum_{k=1}^{K} \alpha_k^j \cdot D_k^j(t)\}$$

The time-averaged cost of buying instances from other cloud providers minus the time-averaged revenue by selling instances to other providers is:

$$C_4^j = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{\sum_{i!=j} \sum_{m=1}^{M} [p_m^i \cdot r_m^{ji}(t) - p_m^j \cdot r_m^{ij}(t)]\} \quad j \in [1, J]$$

The time-averaged cost of transferring data into or out of other cloud providers minus the time-averaged revenue by transferring data of other providers is:

$$C_5^j = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}\{\sum_{i!=j} [p_{do}^i \cdot G_{out}^{ij}(t) + p_{di}^i \cdot G_{di}^{ij} - p_{do}^j \cdot G_{out}^{ji}(t) - p_{di}^j \cdot G_{in}^{ji}(t)]\}$$
$$j \in [1, J]$$

Hence, the total cost at time slot t of cloud j is:

$$C^j = C_1^j + C_2^j + C_3^j + C_4^j + C_5^j$$

The cost minimization problem at cloud j can be formulated as follows:

$$\min C^j$$
$$\text{constraints } 1-8.$$

E. Social cost minimization

The social cost of the federation is the overall cost of the cloud federation:

$$\sum_{j=1}^{J} [C_1^j + C_2^j + C_3^j + C_4^j + C_5^j]$$

Since the income and expenditure due to buying/selling instances and data transferring among clouds cancel each other, the formula above equals to $\sum_{j=1}^{J}[C_1^j + C_2^j + C_3^j]$. The social cost minimization problem is:

$$\min \sum_{j=1}^{J} [C_1^j + C_2^j + C_3^j]$$
$$\text{constraints } 1-8.$$

The variables include $U_k^j(t)$, i.e., the number of new type k jobs served at time t, $I_{l,s}^j(t)$, i.e., the instance s assignment indicator, $n_m^j(t)$, i.e., the number of active servers providing type m VMs, $r_m^{ij}(t)$, i.e., the number of type m VMs cloud i buys from cloud j.

We summarize important notations in Table I for ease of reference.

TABLE I
IMPORTANT NOTATIONS

J	total number of cloud providers in the federation.
M	types of instances.
N_m^j	number of servers running type m instances in cloud j.
H_m^j	number of type m instances a server in cloud j can host.
p_m^j	price of buying an instance m in cloud j.
p_{do}^j	price of transferring one volume data out of cloud j.
p_{di}^j	price of transferring one volume data into cloud j.
K	the number of all job types.
v_k	the number of VMs required by job type k.
w_k	the service time of job type k.
m_k	type of VMs required by job type k.
$\mathcal{A}_k^j(t)$	set of type k jobs arriving at t in cloud j.
$A_k^j(t)$	number of type k jobs arriving at t in cloud j.
$U_k^j(t)$	number of newly served type k jobs at t in cloud j.
Q_k^j	queue backlog for workload of type k jobs in cloud j.
$D_k^j(t)$	number of dropped type k jobs at t in cloud j.
d	maximum responsive delay.
$\mathcal{L}^j$	set of all jobs from cloud j being served.
T^l	the traffic matrix of job l.
r_m^{ij}	number of type m instances cloud i buys from cloud j.
$I_{l,s}^i$	indicator whether instance s of job l is placed in cloud i or not.
U_m	the data size of migrating a type m instance.
B_{out}^j	upload bandwidth limit of cloud j.
B_{in}^j	download bandwidth limit of cloud j.
β^j	operational cost of running one server.
β_{do}^j	cost of transferring one volume of data out of cloud j.
β_{di}^j	cost of transferring one volume of data into cloud j.

II. ALGORITHM DESIGN

Let $\mathbf{Q}^j = (Q_1^j, Q_2^j, ..., Q_K^j)$ be the queue backlog vector at cloud j. To make the job scheduling satisfy delay constraint, we apply ϵ-persistent queue technique [], to define a virtual queue associated with each job queue.

$$Z_k^j(t+1) = \max\{Z_k^j(t) + 1_{Q_k^j(t)>0} \cdot [\epsilon_k - \sum_{w=0}^{w_k-1} U_k^j(t-w)] - w_k \cdot D_k^j(t) - 1_{Q_k^j(t)=0} \cdot U_k^{max}, 0\} \quad (9)$$

Let $\Theta^j(t) = (\mathbf{Q}^j, \mathbf{Z}^j)$ be the vector of actual queues and virtual queues in cloud j. $\Theta(t) = (\Theta^1(t), \Theta^2(t), ..., \Theta^J(t))$. The Lyapunov function of $\Theta(t)$ is:

$$L(\Theta(t)) = \frac{1}{2} \sum_{j=1}^{J} \sum_{k=1}^{K} [Q_k^j(t)^2 + Z_k^j(t)^2] \quad (10)$$

The one slot drift is

$$\Delta(\Theta(t)) \le B + \sum_{j=1}^{J} \sum_{k=1}^{K} Q_k^j(t) \cdot [w_k \mathbb{E}[A_k^j(t)] - \sum_{w=0}^{w_k-1} U_k^j(t-w) - w_k \cdot D_k^j(t)] + \sum_{j=1}^{J} \sum_{k=1}^{K} Z_k^j(t) \cdot [1_{Q_k^j(t)>0}(\epsilon_k - \sum_{w=0}^{w_k-1} U_k^j(t-w)) - w_k D_k^j(t) - 1_{Q_k^j(t)=0} U_k^{max}]$$

B is a constant.

The drift plus penalty is:

$$\begin{aligned}
&\Delta(\Theta(t)) + V \cdot \sum_{j=1}^{J}[\beta^j(t) \cdot \sum_{m=1}^{M} n_m^j(t) + \beta_{do}^j(t) \cdot \sum_{i=1}^{J} G_{out}^{ji}(t) \\
&+ \beta_{di}^j(t) \cdot \sum_{i=1}^{J} G_{in}^{ji}(t) + \sum_{k=1}^{K} \alpha_k^j \cdot D_k^j(t)] \\
&\leq B + \sum_{j=1}^{J}\sum_{k=1}^{K} Q_k^j(t) \cdot [w_k \mathbb{E}[A_k^j(t)] - \sum_{w=0}^{w_k-1} U_k^j(t-w) - w_k \cdot D_k^j(t)] \\
&+ \sum_{j=1}^{J}\sum_{k=1}^{K} Z_k^j(t) \cdot [(\epsilon_k - \sum_{w=0}^{w_k-1} U_k^j(t-w)) - w_k D_k^j(t)] \\
&+ V \cdot \sum_{j=1}^{J}[\beta^j(t) \cdot \sum_{m=1}^{M} n_m^j(t) + \beta_{do}^j(t) \cdot \sum_{i=1}^{J} G_{out}^{ji}(t) \\
&+ \beta_{di}^j(t) \cdot \sum_{i=1}^{J} G_{in}^{ji}(t) + \sum_{k=1}^{K} \alpha_k^j \cdot D_k^j(t)]
\end{aligned}$$

We minimize the right-hand-side of the above drift-plus-penalty expression.

$$\begin{aligned}
&\min \sum_{j=1}^{J}\sum_{k=1}^{K}[Q_k^j(t) + Z_k^j(t)] \cdot \\
&[- \sum_{w=0}^{w_k-1} U_k^j(t-w) - w_k \cdot D_k^j(t)] \\
&+ V \cdot \sum_{j=1}^{J}[\beta^j(t) \cdot \sum_{m=1}^{M} n_m^j(t) + \beta_{do}^j(t) \cdot \sum_{i=1}^{J} G_{out}^{ji}(t) \\
&+ \beta_{di}^j(t) \cdot \sum_{i=1}^{J} G_{in}^{ji}(t) + \sum_{k=1}^{K} \alpha_k^j \cdot D_k^j(t)]
\end{aligned}$$

Constraints$3-8$.

A. Solving one slot optimization

To minimize the one slot optimization, we decompose the variables among different cloud providers.

We derive the dual problem of the one slot optimization by relaxing constraints (3), (4), (6).

Associating dual variables with those constraints, the Lagrangian is:

$$\begin{aligned}
&L[\mathbf{U}(t), \mathbf{r}(t), \mathbf{I}(t), \mathbf{n}(t), \lambda(t), \mu(t), \nu(t)] \\
&= \sum_{j=1}^{J}\sum_{k=1}^{K}[Q_k^j(t) + Z_k^j(t)] \cdot \\
&[- \sum_{w=0}^{w_k-1} U_k^j(t-w) - w_k \cdot D_k^j(t)] \\
&+ V \cdot \sum_{j=1}^{J}[\beta^j(t) \cdot \sum_{m=1}^{M} n_m^j(t) + \beta_{do}^j(t) \cdot \sum_{i=1}^{J} G_{out}^{ji}(t) \\
&+ \beta_{di}^j(t) \cdot \sum_{i=1}^{J} G_{in}^{ji}(t) + \sum_{k=1}^{K} \alpha_k^j \cdot D_k^j(t)] \\
&+ \sum_{j=1}^{J}\sum_{m=1}^{M} \lambda_m^j \cdot [\sum_{i=1}^{J} r_m^{ij}(t) - n_m^j(t) H_m^j] \\
&+ \sum_{j=1}^{J}\sum_{m=1}^{M} \mu_m^j \cdot [\sum_{k:m_k=m} \sum_{w=0}^{w_k-1} v_k \cdot U_k^j(t-w) - \sum_{i=1}^{J} r_m^{ji}(t)] \\
&+ \sum_{i=1}^{J}\sum_{j=1}^{J}\sum_{m=1}^{M} \nu_m^{ji} \cdot [\sum_{l \in \mathcal{L}^j, m_{k_1}=m} \sum_{s=1}^{v_{k_l}} I_{l,s}^i(t) - r_m^{ji}]
\end{aligned}$$

Fix the dual variable $(\lambda(t), \mu(t), \nu(t))$, we can define J subproblems. The subproblem j is:

$$\begin{aligned}
&\textbf{inf} - \sum_{k=1}^{K}[Q_k^j(t) + Z_k^j(t)] \cdot [\sum_{w=0}^{w_k-1} U_k^j(t-w) + w_k \cdot D_k^j(t)] \\
&+ V \cdot [\beta^j \cdot \sum_{m=1}^{M} n_m^j(t) + \sum_{i=1}^{J} \beta_{do}^i \cdot G_{out}^{ij} \\
&+ \sum_{i=1}^{J} \beta_{di}^i \cdot G_{in}^{ij} + \sum_{k=1}^{K} \alpha_k^j \cdot D_k^j] \\
&+ \sum_{i=1}^{J}\sum_{m=1}^{M} (\lambda_m^i - \mu_m^j - \nu_m^{ji}) \cdot r_m^{ji} - \sum_{m=1}^{M} \lambda_m^j n_m^j(t) H_m^j \\
&+ \sum_{m=1}^{M} \mu_m^j \sum_{k:m_k=m} \sum_{w=0}^{w_k-1} v_k \cdot U_k^j(t-w) \\
&+ \sum_{i=1}^{J} \nu^{ji} \cdot \sum_{l \in \mathcal{L}^j} \sum_{s=1}^{v_{k_l}} I_{l,s}^i(t)
\end{aligned}$$

Subproblem j can be decomposed into the following 5 small optimization problems:

1) Job dropping:

$$\begin{aligned}
&\min \sum_{k=1}^{K}\{V \cdot \alpha_k^j - [Q_k^j(t) + Z_k^j(t)] w_k\} \cdot D_k^j(t) \\
&0 \leq D_k^j(t) \leq D_k^{max}, 1 \leq k \leq K
\end{aligned}$$

2) Server and VM provision:

$$\min \sum_{m=1}^{M} n_m^j(t) \cdot (V\beta^j - \lambda_m^j H_m^j)$$
$$n_m^j(t) \leq N_m^j, 1 \leq m \leq M$$

3) Job admission rate:

$$\min \sum_{k=1}^{K} \sum_{w=0}^{w_k-1} U_k^j(t-w) \cdot \{\mu_{m_k}^j v_k - [Q_k^j(t) + Z_k^j(t)]\}$$
$$0 \leq U_k^j(t) \leq U_k^{max}, 1 \leq k \leq K$$

4) VM buying/selling:

$$\min \sum_{i=1}^{J} \sum_{m=1}^{M} (\lambda_m^i - \mu_m^j - \nu_m^{ji}) r_m^{ji}(t)$$
$$0 \leq r_m^{ji} \leq N_m^i H_m^i$$

5) VM placement (quadratic assignment problem)

$$\min V \cdot [\sum_{i=1}^{J} \beta_{do}^i \cdot G_{out}^{ij} + \sum_{i=1}^{J} \beta_{di}^i \cdot G_{in}^{ij}] + \sum_{i=1}^{J} \nu^{ji} \cdot \sum_{l \in \mathcal{L}^j} \sum_{s=1}^{v_{k_l}} I_{l,s}^i(t)$$
$$I_{l,s}^i \in \{0,1\}$$
$$\sum_{i=1}^{J} I_{l,s}^i = 1$$

We could apply subgradient algorithm to solve the dual decomposition. By this, we could solve the one shot optimization.

(need more description for algorithm design)

III. Algorithm Performance