1 Modeling of the P2P service migration problem

We suppose there are M videos, and N ISPs. There are one on-premise server and one cloud node in each ISP.

1.1 Progressing stage

Now I can model the cache state as a couple of queue:
$P_m^j(t)$ is the state of storage of video m at the cloud node j. $P_m^j(t) = 0 or 1$. The departure of 1 means moving the video out of the cache while arrival of 1 means moving video into the cache.

Difficulty:
How to decide when to move videos into or out of the cache?

1.2 Optimization of the problem with Lyapunov optimization

This is a combination of optimization for one time deployment and time-average variables. The placement of content is one time deployment while the schedule is for time-average.

Notation definition:

B_s: storage capacity of the on-premise server

B_u: upload bandwidth capacity of the on-premise server

h_j: charging rate for storage on the cloud at the $j-th$ ISP

k_j: charging rate for upload bandwidth on the cloud at the $j-th$ ISP

s_m: storage of $m-th$ video

$y_m^j = \{0,1\}, m = 1, ..., M$: $y_m^j = 1$ if the placement of the $m-th$ video is on the cloud at the $j-th$ ISP; $y_m^j = 0$ otherwise;

D_s^j is the delay from source j to on premise server i

D_c^{ji} is the delay from source j to on cloud node i.

$A_m^j(t)$: at time slot t, number of requests of the $m-th$ video generated from the $j-th$ ISP.

$r_m^j(t)$: at time slot t, number of requests of the $m-th$ video that are admitted into the system. $r_m^j(t) \leq A_m^j(t)$

$S_m^j(t)$: at time slot t, number of requests for video m that are routed from region j to on-premise server i

$C_m^{ji}(t)$: at time slot t, number of requests for video m that are routed from region j to cloud node i

$Q_m^j(t)$: at time slot t, queues of requests from video m from ISP j.

Note: The queue update is: $Q_m^j(t+1) = \max[Q_m^j(t) + r_m^j(t) - S_m^j(t) - \sum_{i=1}^{N} C_m^{ji}(t), 0]$
Different from the previous sub section, $S_m^j(t)$ and $C_m^{ji}(t)$ is not a schedule of fraction of arrival rates for all time slots. Now they are schedule of number of requests (integers) for each time slot.

Note: minimize sum of:

- time average spending cost of upload bandwidth at cloud node
- spending cost of time average upload bandwidth at on premise server
- cost of storage at cloud
- cost of storage at on premise server
- time average weighted delay

maximize $g(\sum_{m=1}^{N}\sum_{j=1}^{N}\overline{r_m^j(t)}) - \alpha_1\overline{\sum_{m=1}^{M}\sum_{j=1}^{N}\sum_{i=1}^{N}(s_m C_m^{ji}(t)k_i)} - \alpha_2\sum_{m=1}^{M}\sum_{j=1}^{N}\sum_{i=1}^{N}\overline{s_m S_m^j(t)} - \alpha_3\sum_{j=1}^{N}\sum_{i=1}^{N}\overline{\sum_{m=1}^{M} s_m(C_m^{ji}(t)D_c^{ji} + S_m^{ji}(t)D_s^j)}$
subject to:
$0 \le C_m^{ji}(t) \le C_m^{ji}(t)y_m^t, \forall j = 1,..,N, \forall i = 1,...,N, \forall m = 1,...,N, \forall t$
$\sum_{m=1}^{M}\sum_{j=1}^{N} s_m S_m^j(t) \le B_u, \forall i = 1,...,N, \forall t$ (on-premise server's upload bandwidth constraint)
Queues $Q_m^j(t)$ is stable, $\forall m, j$, i.e., $\overline{r_m^j(t)} \le \overline{\sum_{i=1}^{N} S_m^j + \sum_{i=1}^{N} C_m^{ji}}$
$Q_m^j(0) = 0, \forall m, j$
$r_m^j(t) < A_m^j(t)$
Note:
known values: B_u, k_j, s_m, $A_m^j(t)$, D_c^{ji}, D_s^j, y_m^j
optimization variables: $S_m^j(t)$, $C_m^{ji}(t)$, $r_m^j(t)$
$\Delta(Q(t)) - Vutility$
$\le B + \sum_{m,j} Q_m^j(t)(r_m^j(t) - S_m^j(t) - \sum_{i=1}^{N} C_m^{ji}(t)) - Vg(\sum_{m,j} r_m^j(t)) + V(\alpha_1\sum_{m,j,i} s_m C_m^{ji}(t)k_i + \sum_{m,j}\alpha_2 s_m S_m^j(t) + \sum_{m,j,i}\alpha_3 s_m C_m^{ji}(t)D_c^{ji} + \sum_{m,j}\alpha_3 s_m S_m^j(t)D_s^j)$
$= B - \sum_{m,j,i} C_m^{ji}(t)(Q_m^j(t) - \alpha_1 V s_m k_i - V\alpha_3 s_m D_c^{ji}) - \sum_{m,j} S_m^j(t)(Q_m^j(t) - V\alpha_2 s_m - V\alpha_3 s_m D_s^j) - [Vg(\sum_{m,j} r_m^j(t)) - \sum_{m,j} r_m^j(t)Q_m^j(t)]$

2 Possible Extension

1. Add time average budget constraint
2. add the constraint of delay
3. (?) Consider the startup and tear-down of virtual machines on cloud nodes.
4. (?) use queue to model the utility difference between the situation when the content is in cache and the situation when the content is out of cache
5. (less attractive) use a flow model to do the deployment of content and another model to do the schedule

3 Note on comparison of related paper

3.1 Optimal placement

constraint: link capacity, disk capacity
optimize: disk
variable: placement, schedule

3.2 Neely p2p

constraint: ratio incentive

3.3 CDN

reduce number of requests so that the analysis of cache eviction is easier.

3.4 Other

Bound delay, queue length, etc.

3.5 Mine

1. disk capacity, budget constraint, delay constraint. 2. variable: placement, schedule. 3. optimize: budget spending. delay.

case 1: cloud A is cheapest, route all requests to cloud A. case 2: server is cheapest, rount all requests to server. if the average arrival rates is smaller than C_u^i, then OK. otherwise, build a virtual queue?

case 3:

if the content in in cache, the queue is decreased faster;

use a queue reflects the utility difference in the past period with and without queue. e.g, with content, the departure is faster, but if the incoming is not large, then the throughput is not large. if the content is not in cache, and the incoming is large, then