Dynamic Service Migration in Mobile Edge Computing Based on Markov Decision Process

Shiqiang Wang [1] Rahul Urgaonkar [2] Murtaza Zafer [3] Ting He [4] Kevi Chan [5] Kin K. Leung [6]

[1]IBM T.J. Watson Research Center, Yorktown Heights, NY, US

[2]Amazon Inc., Seattle, WA, US

[3]Nyansa Inc., Palo Alto, CA, US

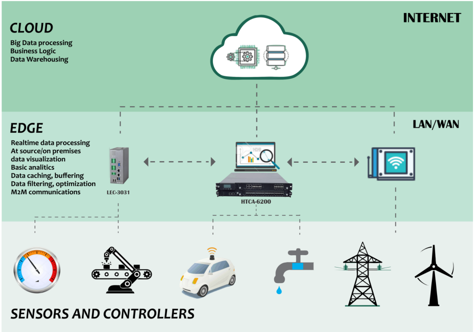
[4]Pennsylvania State University, University Park, PA, US

[5]Army Research Laboratory, Adelphi, MD, US

[6]Department of Electrical and Electronic Engineering, Imperial College London, UK

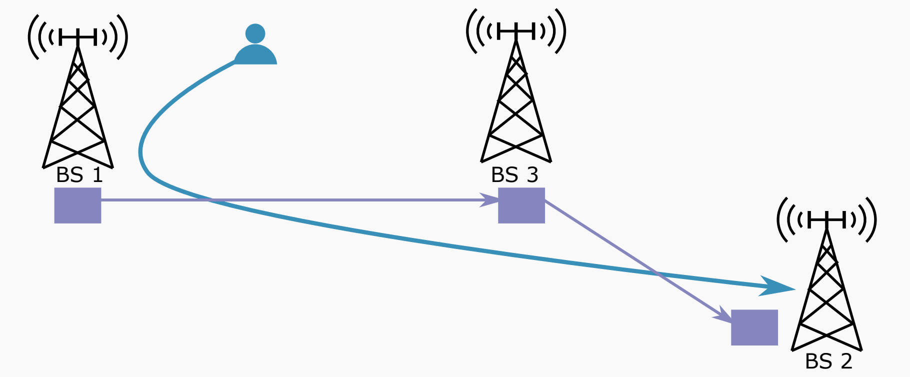
Presented by *Haichuan Wang*, at October 23, 2018

Mobile Edge Computing



- Moving computation closer to users.
- Significantly reduce the service access delay.
- More robust than traditional centralized cloud computing system.

Scenario



As a user moves across different geographical locations, its service may need to be migrated to follow the user so that the benefits of MEC are maintained.

Problems and Challenges

Problems

- Whether the service should be migrated?
- When should the service be migrated?
- Where should the service be migrated?

Challenges

- Uncertainty in user mobility.
- Complex trade-off between the “costs” related to migration and distant data transmission.

Objective and Premises

Objective

Design a control policy to make the service migration decisions, minimizing the long-term cost.

Premises

- Edge server (ES) are co-located with base station (BS).
- Single user accessing a single service.
- Different services are independent of each other.
- One service services a single user.

Locations

- Locations Set: $\mathcal{L}$ (finite, but arbitrarily large).
- Location $l \in \mathcal{L}$ is associated with an ES, represented as a 2-D vectors.
- Distance metric between location $l_1, l_2 \in \mathcal{L}$: $||l_1 - l_2||$.

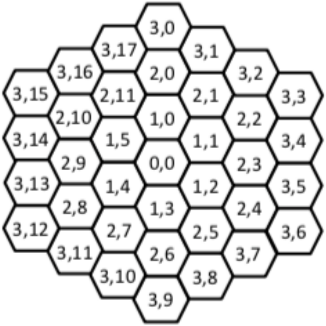
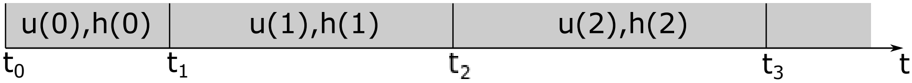


Figure: Locations Example Visualization

User Location and Service Location

A Time Slotted Model

- Duration of slots can be equal or non-equal.
- User location at time slot t: $u(t)$.
- Service location at time slot t: $h(t)$.
- $u(t)$ and $h(t)$ remain fixed during one slot.
- $u(t)$ changes according to a Markovian mobility model.



Control Decisions and Costs

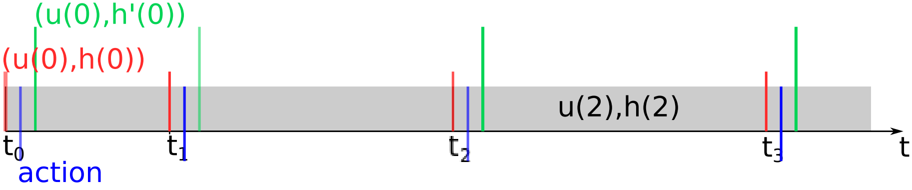
Decisions at the beginning of time slot t

- Migrate the service from location $h(t)$ to location $h'(t)$, cost:

  $$c_m(x), x = ||h(t) - h'(t)||, \text{migration cost, non-decreasing with } x$$

- Do not migrate the service, set $h'(t) = h(t)$, cost:

  $$c_d(y), y = ||u(t) - h'(t)||, \text{transmission cost, non-decreasing with } y$$



System State and Control Policy

System State

- Initial state at beginning of time slot t: $s(t) = (u(t), h(t))$.
- State after the action of the control policy: $s'(t) = (u(t), h'(t))$.

Control Policy

π, defines the action a_π took at beginning of each time slot:

$$a_\pi(s(t)) = s'(t) = (u(t), h'(t))$$

Cost incurred by policy π at time slot t

$$C_{a_\pi}(s(t)) = c_m(||h(t) - h'(t)||) + c_d(||u(t) - h'(t)||)$$

Long-Term Expected Cost

Long-term expected discounted sum cost from any initial state s_0 incurred by policy π:

$$V_\pi(s_0) = \lim_{t\to\infty} \mathbb{E}\left\{\sum_{\tau=0}^{t} \gamma^\tau C_{a_\pi}(s(\tau)) \middle| s(0) = s_0\right\}$$

γ is a discount factor, $0 < \gamma < 1$.

Objective with Formulation

Design the control policy π to minimize the long-term expected expected discounted sum total cost starting from any initial state:

$$V^*(s_0) = \min_{\pi} V_\pi(s_0), \forall s_0$$

$$\left(V_\pi(s_0) = \lim_{t \to \infty} \mathbb{E} \left\{ \sum_{\tau=0}^{t} \gamma^\tau C_{a_\pi}(s(\tau)) \middle| s(0) = s_0 \right\} \right)$$

MDP with infinite horizon discounted sum cost

The optimal solution is given by a stationary policy and can be obtained as the unique solution to the Bellman's equation:

$$\begin{aligned} V^*(s_0) &= \min_{\pi} V_\pi(s_0) \\ &= \min_{a} \left\{ C_a(s_0) + \gamma \sum_{s_1 \in \mathcal{L}\times\mathcal{L}} P_{a(s_0),s_1} V^*(s_1) \right\} \end{aligned}$$

$P_{a(s_0),s_1}$ denotes the probability of transmission from state:

$$s'(0) = s'_0 = a(s_0)$$

Large state space, challenging to derive optimal control policy

Simplify the Search Space

Observation

The cost functions only depend on the distance:

$$C_{a_\pi}(s(t)) = c_m(||h(t) - h'(t)||) + c_d(||u(t) - h'(t)||)$$

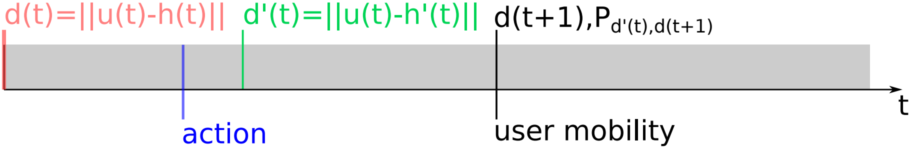
System State Re-defination

Only use the distance between user's and service's locations:

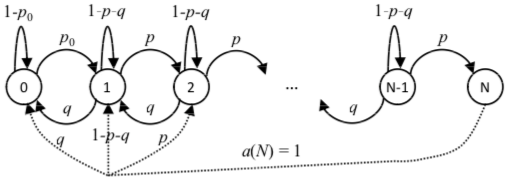
$$d(t) = ||u(t) - h(t)||$$

which is reasonable it as an approximation of the state space for many practical scenarios of interest.

2-D random walk model to 1-D



To simplify the solution, we restrict the transition probability $P_{d'(t),d(t+1)}$ according to the parameters p_0, p and q



The New Cost Function

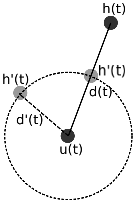
Rule

For an action $d'(t) = a(d(t))$, service location $h'(t)$ is chosen such that:

$$x = ||h(t) - h't(t)|| = |d(t) - d'(t)|$$
$$y = ||u(t) - h'(t)|| = d'(t)$$

New one-timeslot cost

$$C_a(d(t)) = c_m(||h(t) - h'(t)||) + c_d(||u(t) - h'(t)||)$$
$$= c_m(|d(t) - d'(t)|) + c_d(d'(t))$$



Universal Constant-Plus-Exponential Cost Functions

$$c_m(x) = \begin{cases} 0, & if \ \ x = 0 \\ \beta_c + \beta_l \mu^x, & if \ \ x > 0 \end{cases}$$

$$c_d(y) = \begin{cases} 0, & if \ \ x = 0 \\ \delta_c + \delta_l \theta^y, & if \ \ y > 0 \end{cases}$$

Closed-Form Solution to Discounted Sum Cost

$$V_\pi(d(0)) = C_{a_\pi}(d(0)) + \gamma \sum_{d(1)=a_\pi(d(0))-1}^{a_\pi(d(0))-1} P_{a_\pi(d(0)),d(1)} V_\pi(d(1))$$

$$V(d) = A_k m_1^d + B_k m_2^d + D + \begin{cases} H \cdot \theta^d, & if \ 1 - \frac{phi_1}{\theta} - \phi_2 \theta \neq 0 \\ Hd \cdot \theta^d, & if \ 1 - \frac{phi_1}{\theta} - \phi_2 \theta = 0 \end{cases}$$

Numerically, we can find $V(d)$ for all $d \in [0, N]$ in $O(N)$ time.

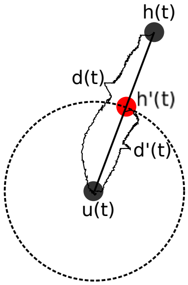
Algorithm for Finding the Optimal Policy

Description

1. Initialize the policy π.
2. Compute $V_\pi(d)$.
3. Update the policy π.
4. Chose $h'(t)$ based on the rule aforementioned.

Complexity

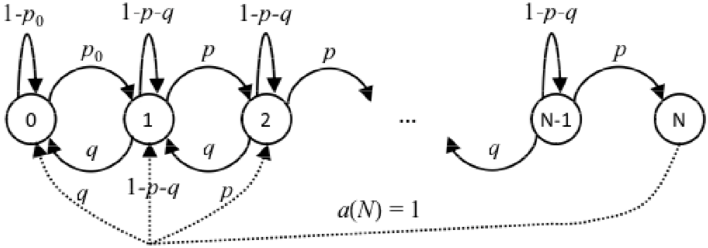
proposed: $O(N^2)$
standard MDP based alg: $O(N^3)$



Summary

- Cost formulation
- Markov Decision Process
- Bellman Equation
- Search Space Simplification

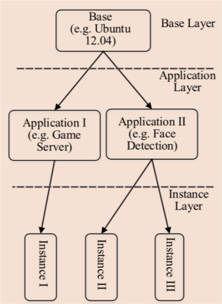
User Mobility Model: Fixed to Dynamic



Dynamic User Mobility Model

Setup a machine learning model to provide dynamic user mobility model.

Service Placement and Service Migration



This paper focuses on the "Instance" layer and assumes the "Application" layer is always ready on each edge server or can be fetched in negligible time, which is not practical in many situations.