Huawei Sweden Hackathon 2023

Final Phase Task Description

Task updated – Transition time

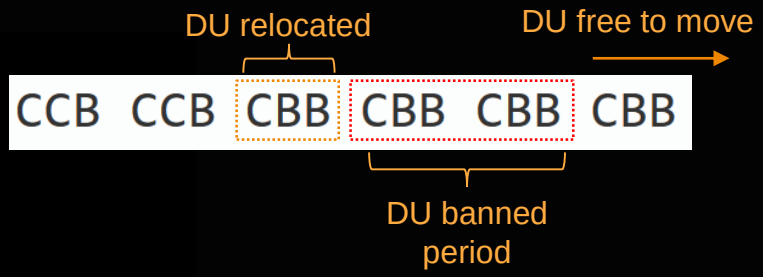
- During the qualification task, relocating an app instance had an action cost. This cost represents partially the difficulties for the real world deployment problem.

- In this phase of the hackathon we will **remove the action cost**, and in turn **introduce a transition time**, which we will call *Z*.

- This time will be an integer number and represents the number of time steps that a relocated instance (CU, DU, PHY) needs to wait until it is allowed to be relocated again.

Task updated – Transition time (II)

- Formally, an instance relocating at time *t* cannot be relocated again in the interval (*t*, *t* +Z] (note *t* +Z is included in the banned period).
- t_0 allocation (initial allocation) is not considered as a relocation, i.e. it is always allowed to relocate at t_0 + 1 regardless of Z value
  - Example: slice S_e is relocated from CCB to CBB at *t* = 3 and Z is defined in the test case as Z = 2. Then S_e DU instance cannot be relocated again during *t* = 4 or *t* = 5. The first time step DU from S_e is allowed to relocate is *t* = 6
  - Note that S_e could still be updated to BBB since the CU is not affected by the DU transition time.



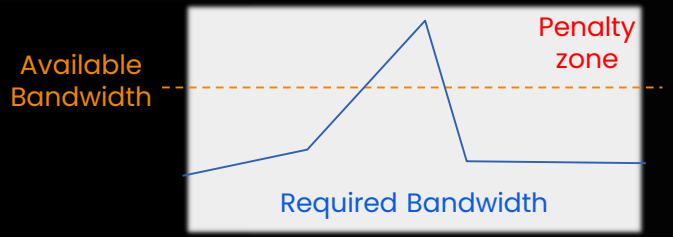
Task updated - IO

- In this phase we will also refine the limitations regarding IO.
- Of course, the output IO from the cloud provider premises is well-known cost in cloud computing.
- However, we also need to take into account that BBU equipment is deployed in a distributed fashion. Wiring each BBU is a relevant expense, among other factors.

- Therefore, in reality, the bandwidth capacity between **the Cloud and the BBU is not unlimited**.

Task updated– IO limitation (II)

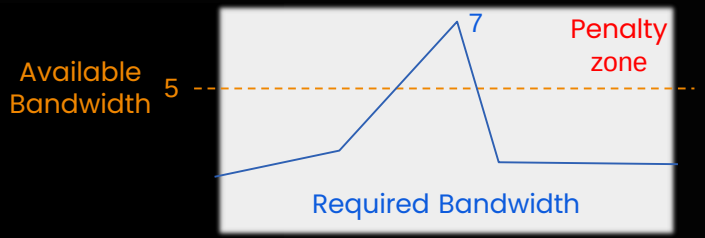
- For each test case we define the available Bandwidth, *W*, between the Cloud and the BBU as an integer.
- Also, a Penalty Cost *P* will be defined as an integer.
- We will consider the IO cost of each link (as defined in the Qualification Phase) to also represent the Bandwidth Units needed for that link.
- At any time step, if the required Bandwidth Units required is bigger than the available Bandwidth *W*, the operator has to pay a penalty of *P* per exceeding unit.



Task updated– IO limitation (III)

- Formally, we define W_r as the amount of bandwidth required given the allocation and traffic (sum of link costs per slice). Then if $W_r - W > 0$ there is a penalty cost of $(W_r – W) * P$
- Example: available bandwidth is W = 5 and we have 2 slices, requiring at time t_x 3 and 4 bandwidth units (i.e. the IO cost for the relevant link given their allocation and traffic is 3 and 4).

  Then the IO cost for t_x is: 3 + 4 + ((3+4) - 5) * P = 7 + 2*P



Cost calculations formalized

$$CloudCost_t = \sum_s CPURequired_s * Traffic_{t,s} * CPUCost + MEMRequired_s * Traffic_{t,s} * MEMCost$$

$$\text{BBU}Cost_t = BBUSetsRequired_t * BBUSetCost$$

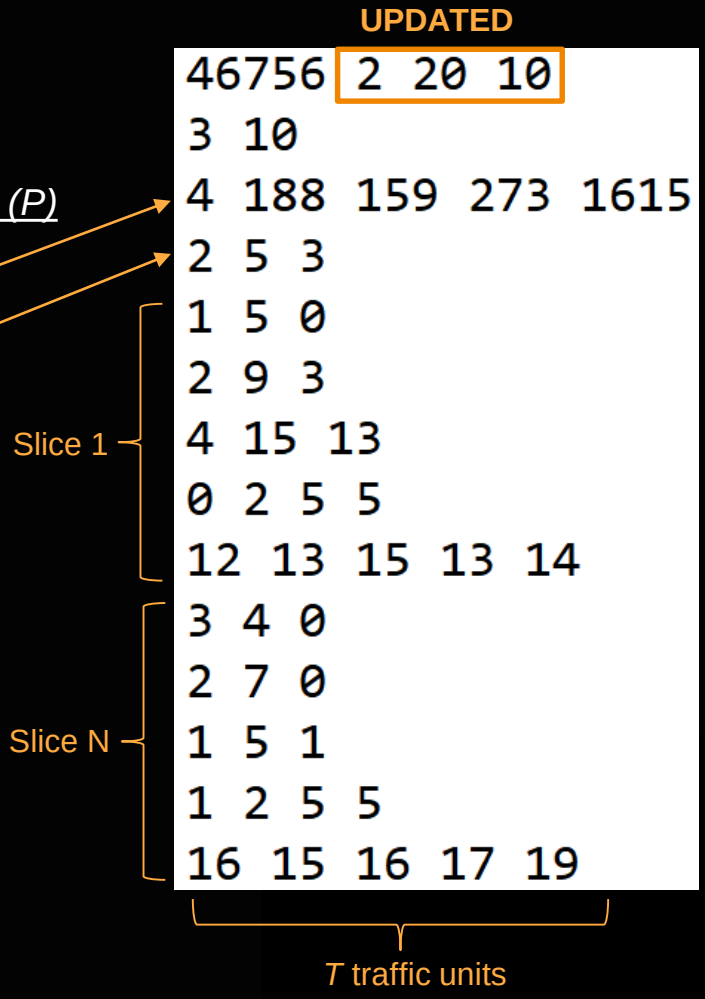
$$\boldsymbol{BandwithRequired_t = \sum_s IOCost_{LinkUsed_{t,s}} * Traffic_{t,s}}$$

$$\boldsymbol{PenaltyCost_t = \max(0, BandwithRequired_t - Bandwidth) * Penalty}$$

$$\boldsymbol{IOCost_t = \sum_s IOCost_{LinkUsed_{t,s}} * Traffic_{t,s} + PenaltyCost_t}$$

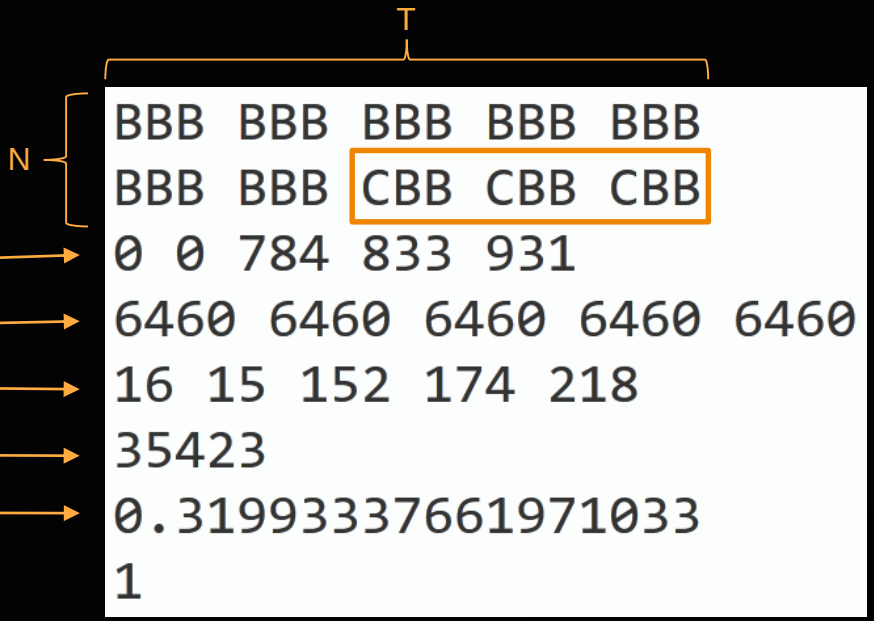
Input for the test cases

- *baseline_cost, transition_time, bandwidth (W), penalty_cost (P)*
- *Cloud Costs :CPU_cost, MEM_cost*
- *BBU Profile : B, CPU, MEM, ACC, cost_per_set*
- *# of slices, time horizon, CPU/ACC: N, T, X*
- *The following will be repeated for each slice s (N repeats)*
  - CU resource unit: cpu, mem, acc
  - DU resource unit: cpu, mem, acc
  - PHY resource unit: cpu, mem, acc
  - IO cost: c1, c2, c3, c4
  - T traffic units: u1, u2, … uT



Output (same as Qualy phase)

- *Solution matrix*
- *Cloud costs*
- *BBU costs*
- *I/O costs*
- *OPEX*
- *Score*
- *Execution time(ms)*



Competition rules

- **This phase consists of 40 test cases.**
  - You will receive 20 in the initial Final Kit.
  - The remaining 20 will be released in an updated Final Kit at 10AM on Saturday

- **Leaderboard will freeze at 10AM**
  - You will see your updated score under Results tab but Ranking tab will not be updated

Ranking rules

- **Your Submission Score for the final Ranking will calculated as follows:**
  - Your code will be executed 5 times against the 40 test cases set
  - Testcases execution not meeting the time restriction will not be considered
  - Your final score will be the average of the 3 best results
- **6 teams will be shortlisted for final presentation to the Jury**

- **Final Score of top 6 teams will be:**
  - 0.6 * Submission Score (normallized to 100) + 0.4 * Jury Score (normallized to 100)

GOOD LUCK!

Organized BY: