Server Fleet Management at Scale

Tech Arena 2024 - **Phase 2**

Huawei Ireland Research Center

This document contains the instructions to compete in the Huawei Ireland Research Center Tech Arena 2024 challenge. Please take the time to read this document carefully and do not hesitate to contact us if you have any questions.

Problem 1

The goal is to build a model that at each time-step recommends the number of servers of each type to deploy at each data-center in order to maximize the given objective function. **Unlike Phase 1, now there is a single objective, to maximize the profit, and it is possible to implement a pricing strategy**.

Problem 2

Create a presentation (7 mins + 3 mins for Q&A) to illustrate your work on Problem 1:

1. To ensure stakeholders fully understand your solution, present business insights in a clear, concise, and visual manner.
2. To persuade stakeholders to adopt your algorithm, explain why it represents a powerful and reliable approach to the problem at hand.

Here is a limited list of suggestions:

a You could analyze all problem variables e.g. the average revenue by server generation, the average server lifespan, etc.

b You could compare your current solution with your Phase 1 solution.

c You could analyze the resource consumption of your algorithm.

Note: only top-8 performing teams on Problem 1 will be invited to present their results.

1 Overview

The Tech Arena 2024 problem is outlined in Figure 1. There is one decision-maker who is in charge of four data-centers. Each data-center can contain two types of server: CPU, and GPU servers. The decision-maker has one objective: to maximize the profit. At the same time, the decision-maker has to comply with one constraint: each data-center has a fixed-size capacity in terms of the number of servers it can host. In order to achieve the objective the decision-maker can take five actions at each discrete time-step: buy a server, move a server from one data-center to another, hold a server as it is, dismiss a server, and set the price for any segment of demand.

The rest of this document is organized as follows. "Problem 1" formulation is detailed in Section 2. "Problem 1" solution evaluation, solution format, and submission details are provided in Section 3. "Problem 1" code-base and data are described in Section 4. "Problem 2" solution evaluation is described in Section 5. Finally, the overall "Phase 2" evaluation process is outlined in Section 6.

2 Problem 1 - Formulation

1. **Decision-maker**. There is one decision-maker who is in charge of four data-centers.

2. **Data-centers**. Each data-center has four attributes as listed in Table 1. The data-center data can be found in the file "`datacenters.csv`".

	Attribute	Explanation	Variable
1	Data-center ID	This is a unique data-center ID.	k
2	Cost of Energy	This is the electricity price per kilowatt per time-step.	h
3	Latency Sensitivity	This is the time it takes for data to travel from its source to the data-center and back. Latency sensitivity is divided into three categories: low, medium, and high.	i
4	Slots Capacity	A slot is a unit of space designed to hold a server in place. A server can occupy two or more slots.	V

Table 1: Data-center Attributes

3. **Servers**. All data-centers can host a variety of servers. There are two types of servers: CPU, and GPU servers. As technology advances, new servers are available for purchase at certain time-steps. Each server has 13 attributes as listed in Table 2. The server data can be found in the file "`servers.csv`". Servers' selling prices are stored in the file "`selling_prices.csv`" and elasticity data can be found in the file "`price_elasticity_of_demand.csv`".

	Attribute	Explanation	Variable
1	Server ID	This is a unique ID related to each server.	s
2	Server Generation	As technology advances, new servers are available for purchase at certain time-steps. This is the unique ID of a generation of servers. There are four generations of CPU servers, and three generations of GPU servers.	g
3	Server Type	The server type can be: CPU or GPU.	
4	Capacity	The capacity has a different unit of measurement for each server type. CPU servers capacity is measured in number of CPUs; GPU servers capacity is measured in number of GPU cards.	z
5	Release Time	Time-steps at which the server is available for purchase.	
6	Purchase Price	This is the server price.	r
7	Slots Size	This is the number of slots occupied by the server.	v
8	Energy Consumption	This is the server energy consumption in terms of kilowatt per time-step.	$\hat{e}$
9	Cost of Moving	This is the cost required to move a server from one data-center to another.	m
10	Operating Time	This is the number of time-steps since the server has been deployed.	x
11	Life Expectancy	This is the maximum number of time-steps that the server can be used before to be dismissed.	$\hat{x}$
12	Selling Price	This is the base selling price for each unit of measurement of a given server generation. A change to the base price causes a change in the demand for the related pair of latency sensitivity and server generation.	$\hat{p}$
13	Elasticity	This is the price elasticity of demand that gives the percentage change in quantity demanded when, ceteris paribus, there is a percentage change in price.	ϵ

Table 2: Server Attributes

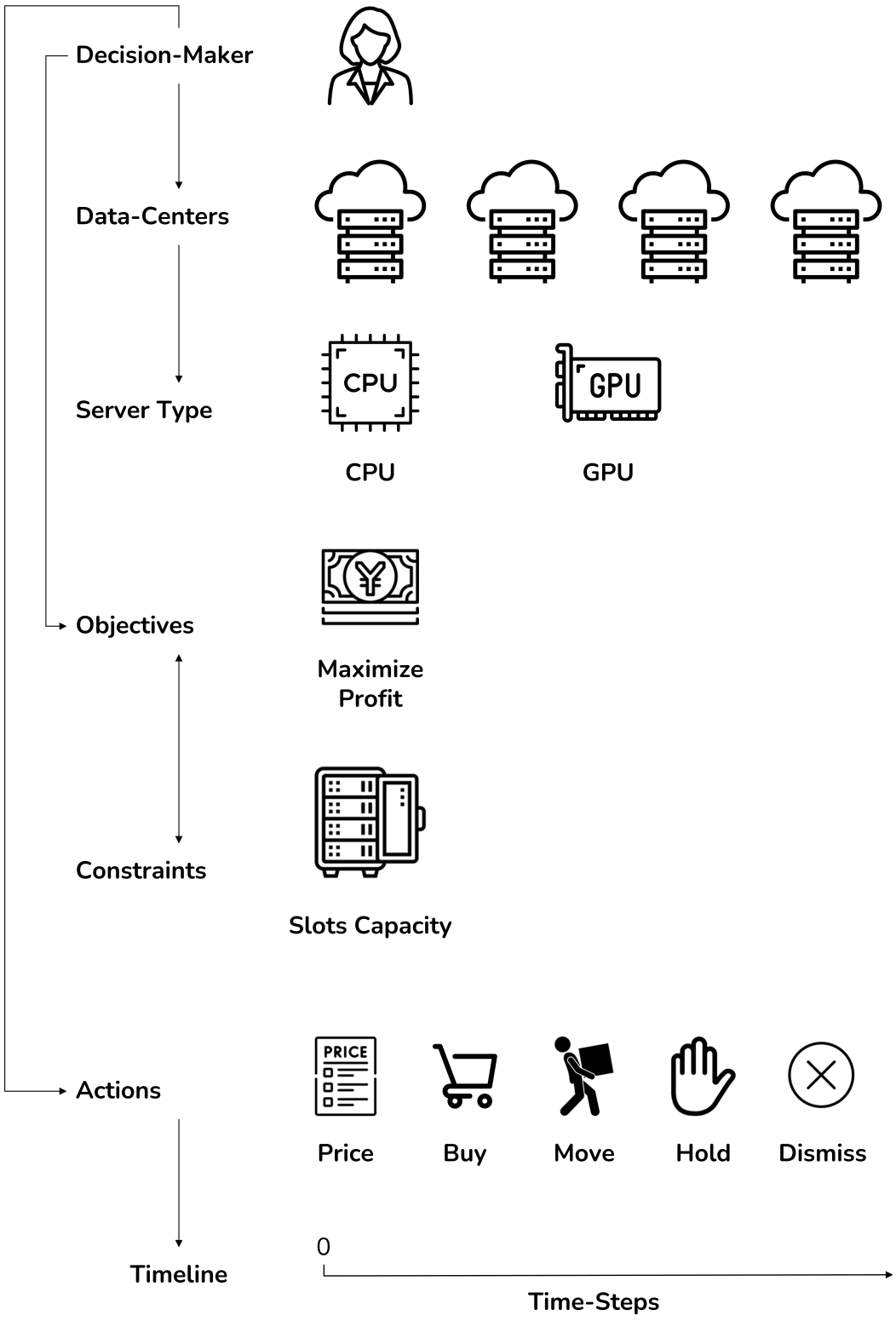

Figure 1: Problem Overview

4. **Objective**. At each time-step, the decision-maker wants to maximize the profit as defined in Eq. 1.

   The profit P. This is defined in Eq. 1 as the difference between the revenue R and the cost C.

$$P = R - C \tag{1}$$

   The revenue R is defined in Eq. 2.1 as the sum of the revenue generated by the capacity $Z^f_{i,g}$ deployed to satisfy the demand $D_{i,g}$ for a certain latency sensitivity i and server generation g. The revenue equals the met demand $min(Z^f_{i,g}, D_{i,g})$ times the price $p_{i,g}$. Here, f represents the failure rate that is sampled from a truncated Weibull distribution with $f \in [0.05, 0.1]$. Specifically, the capacity $Z^f_{i,g}$ is equal to the sum of the capacities of all servers of generation g deployed across all data-centers with latency sensitivity i adjusted by the failure rate f as follows: $Z^f_{i,g} = (1 - f) \times Z_{i,g}$. Selling prices can be set by the decision maker with implications for the demand that are described in Eq. 5.2. If no price is set by the decision maker, then the base price is used (see "`selling_prices.csv`").

$$R = \sum_{i \in I} \sum_{g \in G} \min(Z^f_{i,g}, D_{i,g}) \times p_{i,g} \tag{2.1}$$

   The cost C is defined in Eq. 2.2 as the sum of the cost all servers S_k deployed across all data-centers K. The cost of a server is equal to the sum of the server purchase price r_s, the cost of the server energy consumption e_s, and the server maintenance cost $\alpha(\cdot)$. If the server is moved from one data-center to another it is necessary to account for the moving cost m. The server energy consumption, as defined in Eq. 2.2.1, is equal to the product of the server energy consumption $\hat{e}_s$ times the cost of energy h_k that is the cost of energy at the data-center k where the server s is deployed. Finally, the maintenance cost is calculated according to a function $\alpha(\cdot)$ defined in Eq. 2.2.2. This function takes as input: the server operating time x_s, the server life expectancy $\hat{x}_s$, and average maintenance fee b_s.

$$C = \sum_{k \in K} \sum_{s \in S_k} \begin{cases} r_s + e_s + \alpha(x_s) & \text{if } x_s = 1 \\ e_s + \alpha(x_s) + m & \text{if action} = \text{move} \\ e_s + \alpha(x_s) & \text{otherwise} \end{cases} \tag{2.2}$$

$$e_s = \hat{e}_s \times h_k \tag{2.2.1}$$

$$\alpha(x_s) = b_s \times \left[1 + \frac{1.5 x_s}{\hat{x}_s} \times \log_2\left(\frac{1.5 x_s}{\hat{x}_s}\right)\right] \tag{2.2.2}$$

5. **Constraint: the number of slots**. As defined in Eq. 7, the number of slots occupied at each data-center k must be less than or equal to its slots capacity V_k. In this equation, $v_{s,k}$ represents the slots size of server s deployed at the data-center k while S_k represents the set of servers deployed at date-center k.

$$V_k \geq \sum_{s \in S_k} v_{s,k} \qquad \forall \quad k \tag{3}$$

6. **Actions**. At each time-step, the decision-maker can take four types of action to maximize the objective function. These actions are detailed in Table 3. Again, at each time-step the decision-maker can take as many actions as needed. As an example, at time-step 1 the decision-maker may choose to buy 50 CPU servers for data-center 1 and 10 GPU servers for data-center 2.

	Action	Explanation
1	Buy	With this action it is possible to buy a new server and deploy it at a given data-center. This action requires a cost as mentioned in Eq. 2.2.
2	Move	With this action it is possible to remove a server from a given data-center and deploy it into another. This action requires a cost as mentioned in Eq. 2.2.
3	Hold	This action is equivalent to "do nothing". With this action a server will continue to be used as it is.
4	Dismiss	With this action it is possible to remove a server from a given data-center. This action is applied automatically when a server achieves its life expectancy.
5	Price	With this action it is possible to set the price for any latency sensitivity and server generation pair.

Table 3: Actions

7. **Demand**. As defined in Eq. 4, at each time-step t, there is a certain demand for each pair of latency sensitivity i and server generation g. Such demand is computed by the "get_actual_demand" function provided in the "evaluation.py" file. Finally, it should be noted that the demand is estimated according to the base price. Any change to the base price causes a change in the demand as described in Eq. 5.2.

$$D_{i,g,t} = D_{i,g,t-1} + \mathcal{N} \tag{4}$$

8. **Price**. For each pair of latency sensitivity i and server generation g it is possible to set a non-negative price p. The change in price, as defined in Eq. 5.1, causes a change in demand, as defined in Eq. 5.2. In these equations, $\hat{p}$ is the base price, p is the price set by the decision maker, and ϵ is the price elasticity of demand. The price elasticity gives the percentage change in quantity demanded given a percentage change in price. For instance, given $\epsilon = -2$ a price increase of 10% causes a 20% decrease in quantity demanded. In practice, when a new price is set the demand is updated through the "update_demand_according_to_prices" function provided in the "evaluation.py" file.

$$\Delta p_{i,g} = \frac{p_{i,g} - \hat{p}_{i,g}}{\hat{p}_{i,g}} \tag{5.1}$$

$$\Delta D_{i,g} = \Delta p_{i,g} \times \epsilon_{i,g} \tag{5.2}$$

9. **Timeline**. The timeline consists of 168 discrete time-steps. At time-step 0 data-centers are empty.

3 Problem 1 - Solution

3.1 Solution Evaluation

Solutions are evaluated according to the cumulative score achieved through Eq. 6 over all the time-steps T. Solutions that violate the constraint are discarded.

As discussed below (Section 3.3), it is required to submit multiple solutions each of which is related to a different realization of the demand. In order to calculate the "Problem 1" score, we first calculate the team score as the average score over the solutions. Then, we calculate the relative score of each team as the ratio between the team score and the team score of the best performing team. Note that the "Problem 1 Score" is a number between 0 and 1.

$$P = \sum_{t=1}^{T} P_t \tag{6}$$

3.2 Solution Format

A solution must be submitted as a `json` file with the same format as outlined in Example 1. All the variables that a solution must contain and the values they can assume are listed in Table 4. A solution example is provided in the file "`solution_example.json`". Finally, the following conditions must be met:

- The "server_id" variable must be unique for all servers. In other words, it is not possible to buy two (or more) servers with the same "server_id".
- At each time-step, it is possible to submit only one {"buy", "move", "hold", "dismiss"} action for each "server_id".
- At each time-step it is possible to submit only one "price" action for each pair of latency sensitivity i and server generation g. If no price is set, then the default one is used. If a price is set at any time-step, then that price is used for all subsequent time-steps unless a new "price" action is submitted.
- It is not required to submit the "hold" action in your solution, in other words, if you buy a server this will be deployed until you submit a "dismiss" action or the server achieves its life expectancy.

3.3 Solution Submission

It is required to evaluate your approach against multiple realizations of the demand. To this end, please consider the following:

- Multiple realizations of the demand can be generated by setting different random seeds as shown in the "mysolution.py" file. When you evaluate a solution based on a certain random seed, that seed must be set as argument of the "evaluation_function" function provided in the "evaluation.py" file.
- **10 random seeds** can be retrieved through the "known_seeds" function provided in the "seeds.py" file.
- It is necessary to create a solution for each random seed using the naming convention "seed.json". All `json` files should be compressed within a `zip` folder that can finally be submitted.

4 Problem 1 - Code-base and Data

All the files required to build and evaluate a solution to the problem at hand can be found in the compressed folder "`tech_arena_24_phase_2.zip`". A brief description of the folder content is provided in Table 5.

	Fleet				Price		
	Variable	**Data Type**	**Values**		**Variable**	**Data Type**	**Values**
1	"time_step"	int	$[1, 168]$	1	"time_step"	int	$[1, 168]$
2	"datacenter_id"	string	See "`datacenters.csv`".	2	"latency_sensitivity"	string	{"high", "medium", "low"}
3	"server_generation"	string	See "`servers.csv`".	3	"server_generation"	string	See "`servers.csv`".
4	"server_id"	int or string	Your choice.	4	"price"	float	$[0, \infty)$
5	"action"	string	{"buy", "move", "hold", "dismiss"}				

Table 4: Solution Variables

	File	Explanation
1	"`solution_example.json`"	This file contains a solution that can be evaluated using the script provided in the file "`example.py`".
2	"`evaluation_example.py`"	This file can be run to evaluate a solution. By default, this file evaluates the solution provided in the file "`solution_example.json`".
3	"`mysolution.py`"	This file contains a simple example of a pipeline that can be used to solve the problem.
4	"`evaluation.py`"	This file contains all the functions needed to evaluate a solution. Of these, "evaluation_function" is the main function needed to evaluate a solution.
5	"`utils.py`"	This file contains a few functions needed to load and save some challenge-related data.
6	"`seeds.py`"	This file contains only one function that returns the random seeds.
7	"`datacenters.csv`"	This file contains the data-centers data described in Table 1.
8	"`servers.csv`"	This file contains the servers data described in Table 2 with the exception of "selling prices" that are provided in the file "`base_prices.csv`".
9	"`selling_prices.csv`"	This file contains the servers base selling prices (see Table 2).
10	"`price_elasticity_of_demand.csv`"	This file contains the price elasticity of demand for each pair of server type and latency sensitivity.
11	"`demand.csv`"	This file contains the baseline demand data that, along with Eq. **??**, is needed to compute the actual demand for each pair of latency sensitivity i and server generation g at a given time-step.
12	"`requirements.txt`"	This file lists the Python libraries needed to run the evaluation function.

Table 5: Challenge Files & Data

5 Problem 2 - Evaluation Criteria

The 8 highest-performing teams on "Problem 1" will be invited to present their results. A panel of mentors will evaluate the presentations assigning a score from 1 to 10 to each of the following 9 evaluation criteria. The normalized score across all criteria constitutes the "Problem 2 Score". Note that the "Problem 2 Score" is a number between 0 and 1.

1. To ensure stakeholders fully understand your solution, present business insights in a clear, concise, and visual manner.

   C1 **Data analysis**: How insightful are the conclusions drawn from the analysis? Have notable business insights been uncovered?

   C2 **Visualization**: Did the presentation successfully communicate complex analyses? Are the visual representations clear and intuitive?

   C3 **Storytelling**: Is the presentation well-structured and engaging?

2. To persuade stakeholders to adopt your algorithm, explain why it represents a powerful and reliable approach to the problem at hand.

   C4 **Solution design**: How well the proposed solution is described? Is the relation between the business context and the proposed solution well-defined?

   C5 **Robustness**: What is the variance between the solutions arising from different random seeds? What accounts for this variance?

   C6 **Extensibility**: Is the solution adaptable to changes in the problem space?

   C7 **Scalability**: How scalable is the proposed solution? Has its resource consumption been thoroughly analyzed?

   C8 **Technical execution**: How sophisticated are the techniques employed in developing the solution?

   C9 **Innovation and creativity**: Is there any use of novel techniques or perspectives on the optimization problem?

6 Overall Phase 2 Evaluation

The final “Phase 2 Score” is defined in Eq. 7 as the weighted sum of “Problem 1 Score” and “Problem 2 Score”.

$$\text{Phase 2 Score} = 0.6 \times \text{Problem 1 Score} + 0.4 \times \text{Problem 2 Score} \tag{7}$$

```
{
"fleet": [
            {"time_step": 1,
             "datacenter_id": "DC1",
             "server_generation": "CPU.S1",
             "server_id": "abc1",
             "action": "buy"},
            ...
            {"time_step": 1,
             "datacenter_id": "DC4",
             "server_generation": "GPU.S1",
             "server_id": "abc2",
             "action": "buy"},
            ...
            {"time_step": 70,
             "datacenter_id": "DC1",
             "server_generation": "CPU.S2",
             "server_id": "abc3",
             "action": "buy"},
            ...
         ]
"pricing_strategy": [
                        {"time_step": 1,
                         "latency_sensitivity": "medium",
                         "server_generation": "CPU.S1",
                         "price": 1},
                        {"time_step": 1,
                         "latency_sensitivity": "high",
                         "server_generation": "CPU.S1",
                         "price": 1.2},
                        ...
                    ]
}
```

Example 1: Solution Format

Disclaimer

- This document is only intended to enable the “Tech Arena 2024” event hosted by Huawei Ireland Research Center. Under no circumstances should the information hereby presented be interpreted as representative of any real entity or organization.
- This document is for the “Tech Arena 2024” participants only and should not to be distributed to external parties.