

Solution Report

Vicente Opazo

December 5, 2024

General Description

The primary insight was to distinguish between small and large cases based on the virtual machine (VM) types. Initially, inefficient machines with unfavorable resource-to-cost ratios were filtered out.

The solution employs a **greedy approach**. When a new container arrives, the algorithm searches for the most suitable VM that can accommodate it. The selection criterion prioritizes the VM with the **oldest allocated container** because about data exploring this VM is more likely to be open more time (because that oldest allocated container).

In the large case, VMs with significantly higher costs than the cheapest suitable option are ignored due to the non-uniform ratio of machines cpu and memory. If no suitable VM is found, a new one is created:

- **Large Case:** The cheapest VM type that meets resource requirements is selected.
- **Small Case:** A heuristic is applied, balancing cost and capacity based on the percentage of total time elapsed. I just try a polynomial depending on the percentage of the time elapsed

If a VM no longer hosts any containers, it is immediately shut down to minimize costs.

Algorithms and Data Structures Used

- **Data Structures:**
 - `multiset<int>`: Efficiently retrieves the **oldest container**, minimizing frag-

mentation and improving VM reuse.

- `map<int, VM>`: Tracks active VMs and their resource states.
- `map<int, vector<int>>`: Records container allocations, tracking associated VMs and resource usage.

- **Heuristic Decision-Making:**

- In the **small case**, the heuristic prioritizes VMs with moderate cost and sufficient capacity for future requests, adjusting dynamically based on the remaining time.
- In the **large case**, a cost-based approach ensures the most economical option without compromising resource requirements.

Potential Improvements

Currently, the approach prioritizes container age for VM selection. Potential enhancements include:

- Considering the **total number of containers** allocated per VM.
- Evaluating **resource utilization ratios**.
- Balancing multiple factors, such as the two oldest containers or available free resources.

During the contest, I experimented with a **neural network model** trained on synthetic data to predict optimal VM selections. While initial results were suboptimal, refining this model or exploring alternative machine learning techniques could improve future performance.