

Introduction

We need to find the optimal storage to choose from a pool of storages in order to utilize the overall storage capacity. A certain amount of Capacity, IOPS, Throughput will be requested, and we need to find the most optimal storage to assign these resources.

Approach

Initial solution: Assign the Capacity, IOPS, Throughput to the storage with the least capacity that is enough and make sure it has enough IOPS and Throughput. For example: If we have 2 storage devices, one with 30/40 TB and the other with 20/40 TB and the request requires 10TB of storage, we will choose the one with 30/40 TB.

Reasoning: This approach will always utilize the capacity over all the existing storages.

If we have 2 storages both with the same capacity like 30/40TB and the request requires 10TB then we will look at other resources like IOPS and Throughput to find the one that matches best.

Implementation

We will receive a list of json objects each containing 4 fields (WWN, capacity, throughput, iops) (example included in storage.json). Then we will load this json data and store it in an array.

Filtering: The filtering will happen in 2 stages. The first stage we are going to choose the storages that satisfy our request, regardless of which is more optimal and we will return an array only with the viable storages.

In the second stage we go over every json object and calculate a score for each storage. We will keep the storage that gives us the least score. The reason we choose the least score is that this means the storage fits the requested storage the most.

Priority: In the current implementation Capacity, Iops, and Throughput all have the same priority. But this can be easily changed in calculateScore(storage) method. We can multiply one of the fields by a certain value to give it a higher priority on the score.

Usage

```
python3 compute.py storages.json requestedCapacityInTB requestedIOPS requestedThroughputInGB/s
```

Output: prints a json object as a string.

Example: `python3 compute.py storages.json 20 1000 5`

Output: `{'wwn': 3, 'capacity': 20, 'iops': 6000, 'throughput': 6}`