Cluster computing/Pooling of Resources

Problem Title : SpaceX

Problem Statement:

Come up with a solution to manage a cluster of computers (They can be physical computers or virtualized like VMs or Containers) and utilize their resources efficiently for long running and resource intensive tasks.

Input to the system will be dynamic and can change. Sample inputs are mentioned in problem description.

Problem Background:

Aman is a student in college and he often face the need to run programs which are long running tasks, which involves high usage of *resources*.

Note : *Resources* consists of CPU, Memory and Disk Space.

He has many computers available, but those are not utilized efficiently. Some of the computers are running a single task, but an efficient solution would have clubbed them to a single machine.

Eg:
Say there are 4 machines A, B, C and D having RAM 10GB, 8GB, 8GB and 4GB respectively.
Let's say 4 users submitted their tasks a, b, c and d with requirements (in terms of memory here for simplicity) as 5GB, 4GB, 2GB and 3GB.

Requirement 1: Run minimum machines

An efficient solution will require *only two machines* to be running to cater these tasks.
Machines A and D
OR B and C
OR A and B
OR A and C

These all are efficient solutions. As total number of machines running are 2.

Other machines should be in shutdown state until new tasks are submitted.

Say we chose to go with A and B. And with below machine allocation

a -> A (Task a is allocated machine A)
b -> A
c -> B
d -> B

Resources left on machines:

A -> 1GB
B -> 3GB

Requirement 2: Utilize an underutilized machine first (Bin Pack)

Two new tasks “e” and “f’ with RAM requirement of 1GB and 4GB are submitted.
Then our current running machines cannot run both of the tasks.

We will allocate task “e” to machine A, so that it is fully utilized.
But to run task “f” , we will power up a new machine and include it into our cluster of computers. Say we power up machine C.

Requirement 3: Shutdown machines which are done running all the tasks

Now say all the tasks submitted to machine A are complete. And no new tasks
were submitted. Then after some wait machine A should shutdown itself.

Challenge: How will one figure out that task is complete ? Options:

1. Monitor PID of tasks
2. Trigger callback via task (All tasks will follow this approach)
3. Create an agent which will be present on each of the machine on pool which will monitor tasks status.
4. Use your creativity

Extra Points:

Requirement 4: Trigger a callback to predefined endpoint when a task complete

When a submitted task is completed it should trigger a callback to a predefined endpoint.

Requirement 5: Use docker containers to run tasks instead of simple binaries

Instead of storing binaries in central repositories docker images can be stored. Cluster will pull the image and spin up a container by capping the resources of container. This will also eliminate the constraint of installing external libraries on cluster machine as those will be present in docker image.

Input Description:

(This is one approach. But teams can decide their own Apis on similar lines)

1. Create Cluster api: This api will bring up required number of virtual machines.

```
{
        "cluster_name": "string",
        "initial_capacity": {
                "cpu": "int",
                "ram": "int",
                "disk": "int"
        },
    "buffer_capacity": {
                "cpu": "int",
                "ram": "int",
                "disk": "int"
    }
}
```

2. Submit task api:

```
{
  "cpu": 10, // 10 percent CPU should be idle
  "ram": 1, // unit is GB
  "disk": 2, // unit is GB,
  "binary_path": "/home/ubuntu/tasks/taska.bin" // Some path from where SCP can be done
  "callback": "https://taska.callback" // Some endpoint which will be hit once task is complete
        }
```

End Goal and Expected Outcome:

1. Implementation with mandatory requirements and a working system.
2. Fault tolerance. If few machines goes down how system recovers.
3. Approach taken to meet the requirements.
4. Flexibility for further scalability.
5. Innovation and creativity in solution.