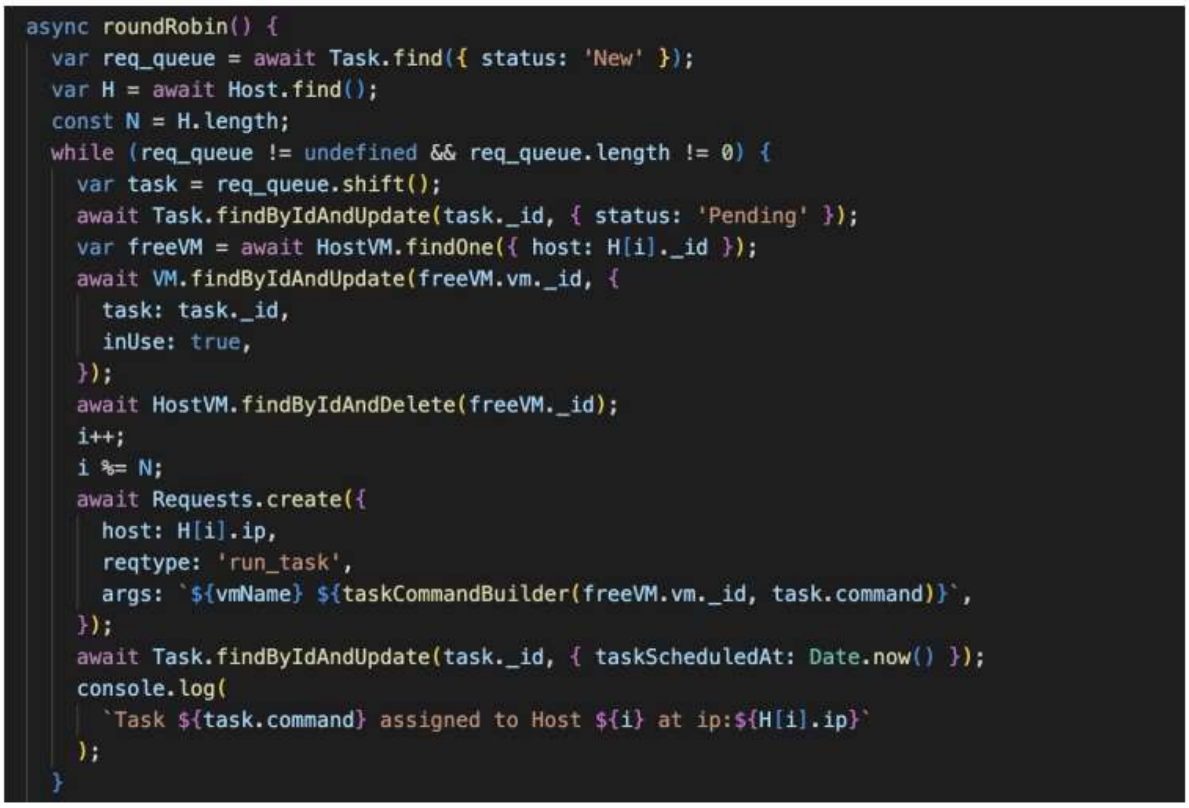
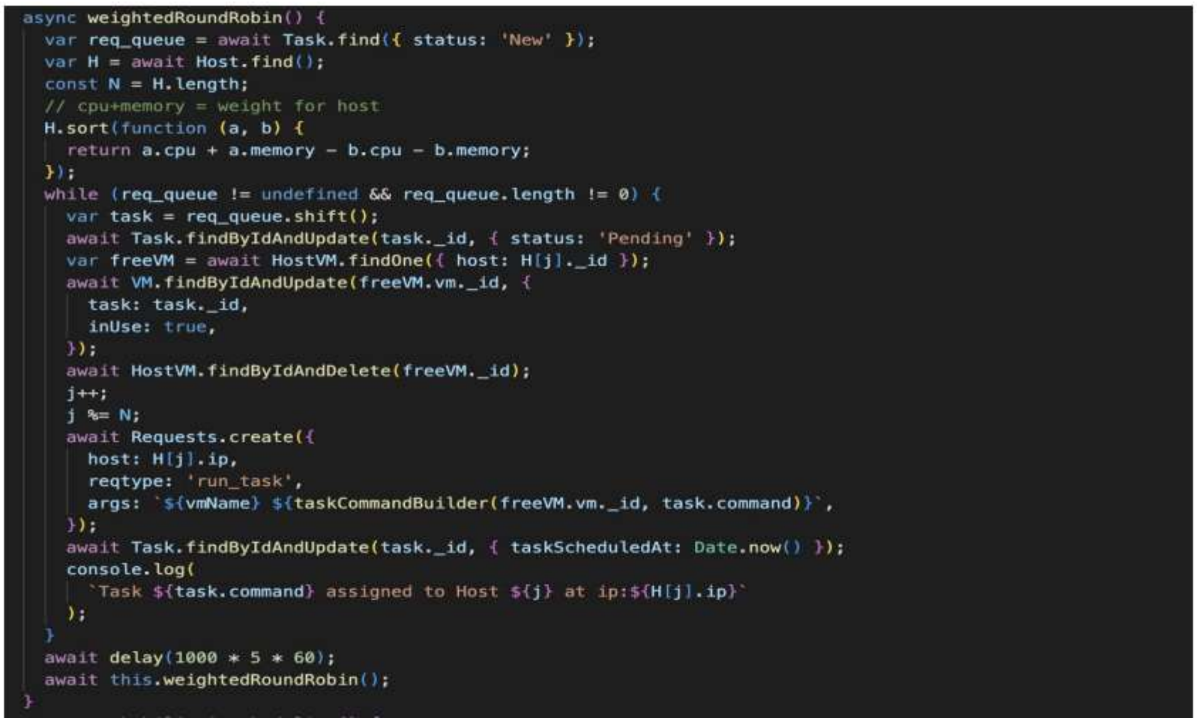
Load Balancing Policies – Round Robin



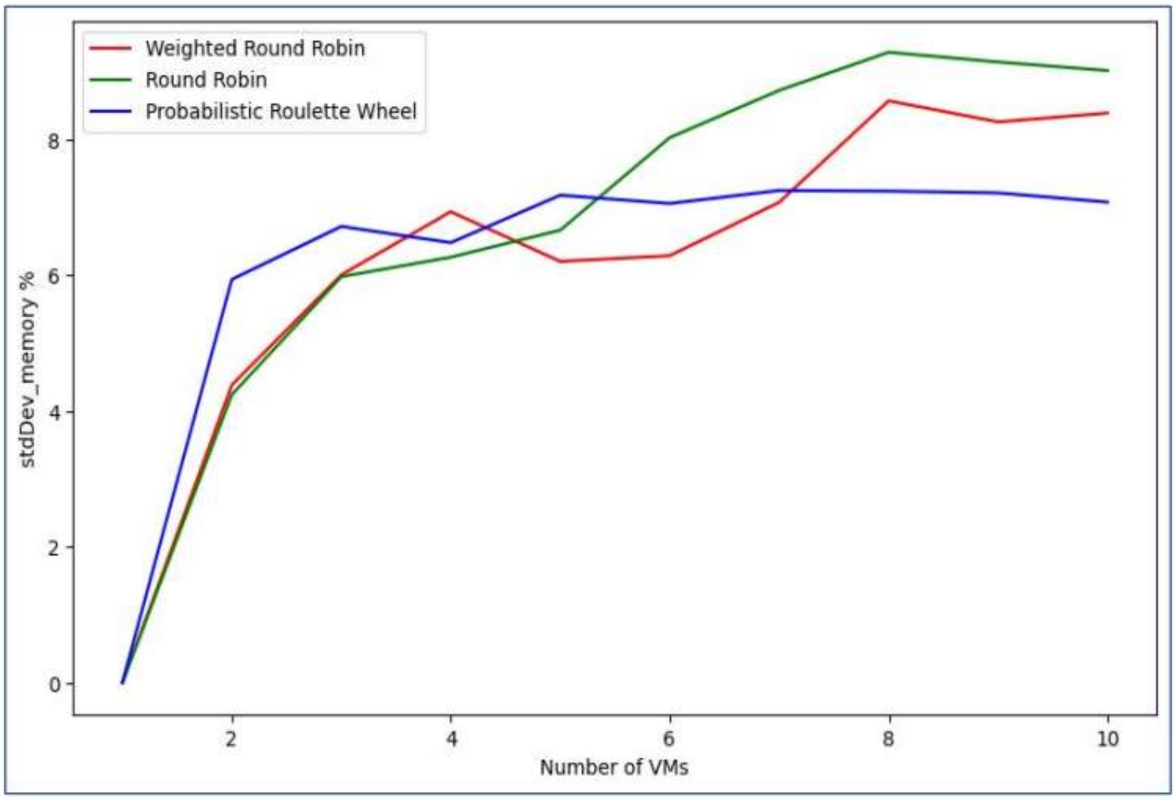
```
async roundRobin() {
  var req_queue = await Task.find({ status: 'New' });
  var H = await Host.find();
  const N = H.length;
  while (req_queue != undefined && req_queue.length != 0) {
    var task = req_queue.shift();
    await Task.findByIdAndUpdate(task._id, { status: 'Pending' });
    var freeVM = await HostVM.findOne({ host: H[i]._id });
    await VM.findByIdAndUpdate(freeVM.vm._id, {
      task: task._id,
      inUse: true,
    });
    await HostVM.findByIdAndDelete(freeVM._id);
    i++;
    i %= N;
    await Requests.create({
      host: H[i].ip,
      reqtype: 'run_task',
      args: `${vmName} ${taskCommandBuilder(freeVM.vm._id, task.command)}`,
    });
    await Task.findByIdAndUpdate(task._id, { taskScheduledAt: Date.now() });
    console.log(
      `Task ${task.command} assigned to Host ${i} at ip:${H[i].ip}`
    );
  }
```

Load Balancing Policies – Weighted Round Robin

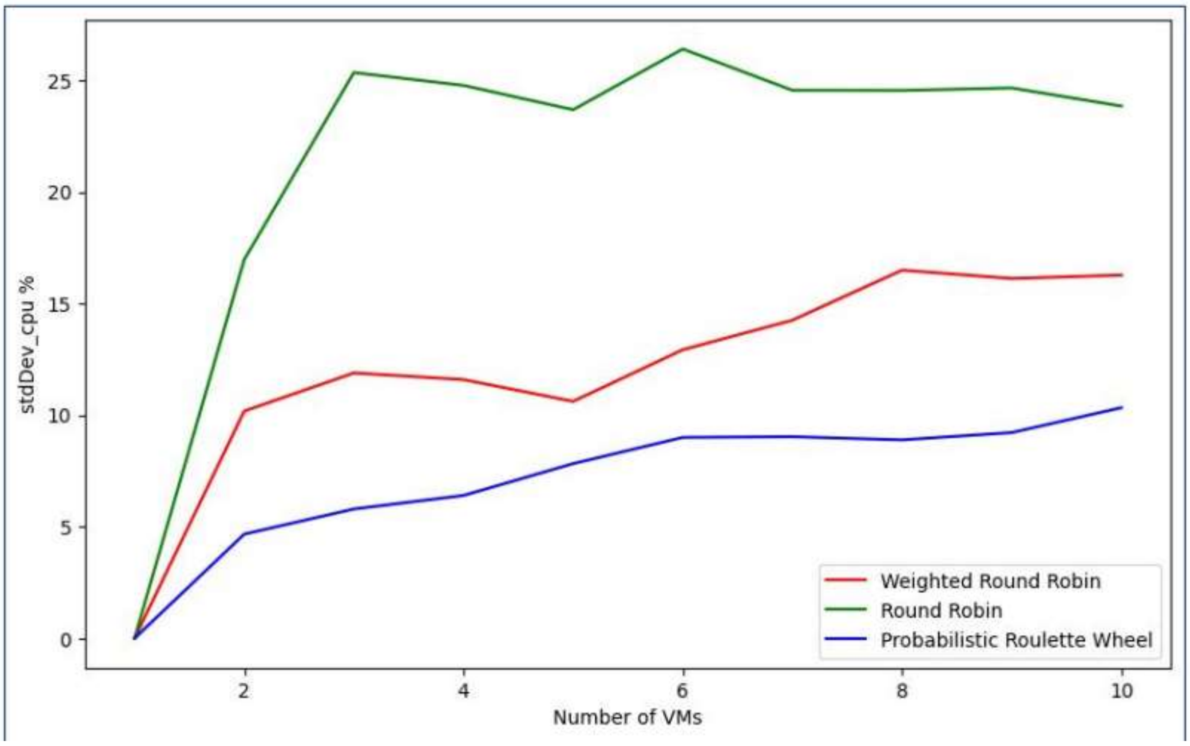


```
async weightedRoundRobin() {
  var req_queue = await Task.find({ status: 'New' });
  var H = await Host.find();
  const N = H.length;
  // cpu+memory = weight for host
  H.sort(function (a, b) {
    return a.cpu + a.memory - b.cpu - b.memory;
  });
  while (req_queue != undefined && req_queue.length != 0) {
    var task = req_queue.shift();
    await Task.findByIdAndUpdate(task._id, { status: 'Pending' });
    var freeVM = await HostVM.findOne({ host: H[j]._id });
    await VM.findByIdAndUpdate(freeVM.vm._id, {
      task: task._id,
      inUse: true,
    });
    await HostVM.findByIdAndDelete(freeVM._id);
    j++;
    j %= N;
    await Requests.create({
      host: H[j].ip,
      reqtype: 'run_task',
      args: `${vmName} ${taskCommandBuilder(freeVM.vm._id, task.command)}`,
    });
    await Task.findByIdAndUpdate(task._id, { taskScheduledAt: Date.now() });
    console.log(
      `Task ${task.command} assigned to Host ${j} at ip:${H[j].ip}`
    );
  }
  await delay(1000 * 5 * 60);
  await this.weightedRoundRobin();
}
```

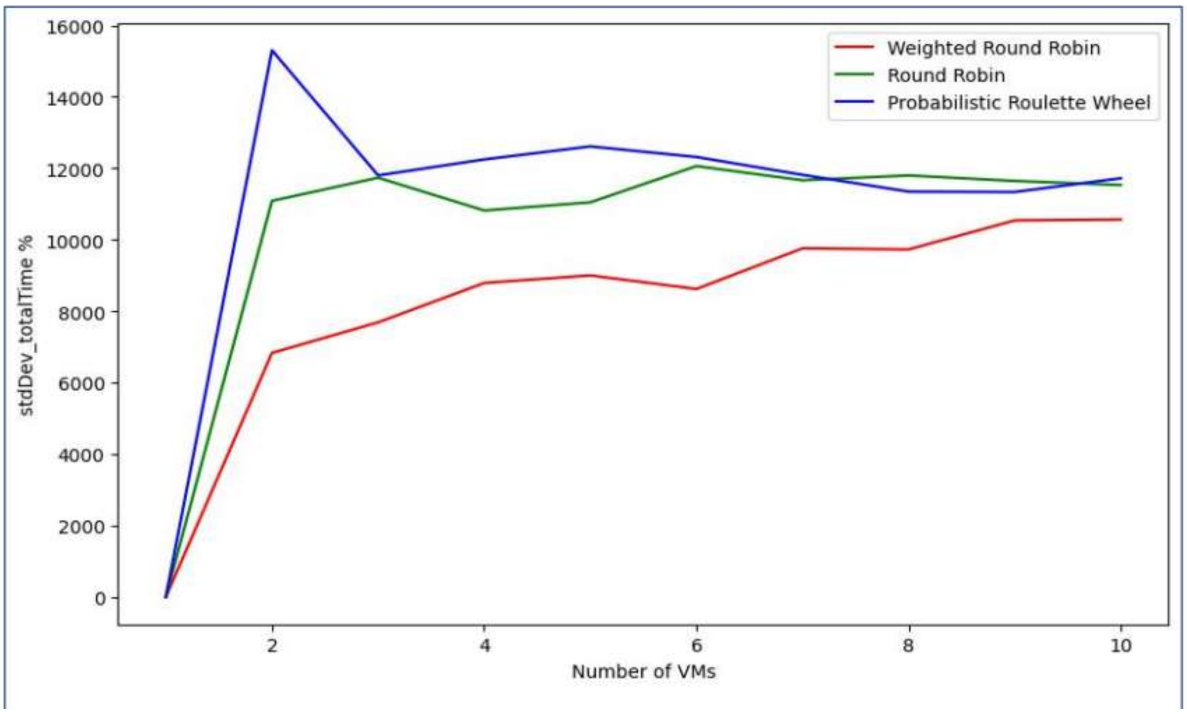
RESULTS

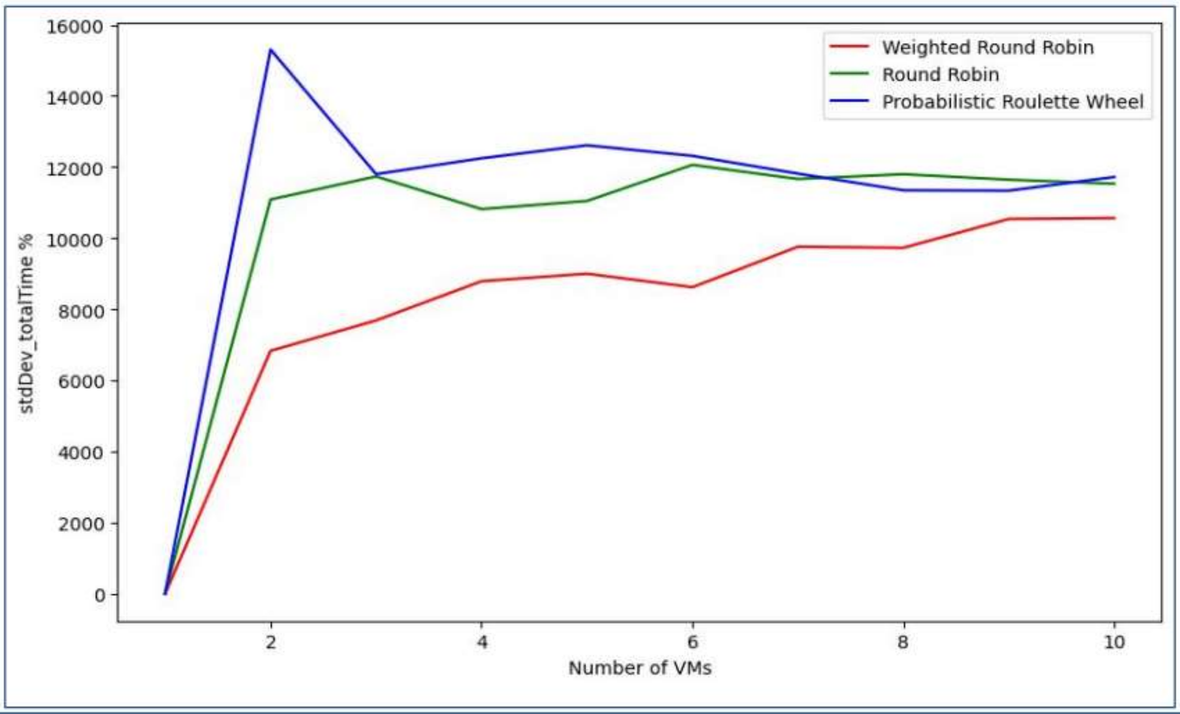


Weighted Round Robin
Round Robin
Probabilistic Roulette Wheel
stdDev_memory %
Number of VMs



Weighted Round Robin
Round Robin
Probabilistic Roulette Wheel
stdDev_cpu %
Number of VMs


16000
14000
12000
10000
8000
6000
4000
2000
0
stdDev_totalTime %
2
4
6
8
10
Number of VMs
Weighted Round Robin
Round Robin
Probabilistic Roulette Wheel


16000
14000
12000
10000
8000
6000
4000
2000
0
stdDev_totalTime %
2
4
6
8
10
Number of VMs
Weighted Round Robin
Round Robin
Probabilistic Roulette Wheel