**Cgroup can Kill Job**

**Last week’s task:**

1. **Activate cgroup**
2. **Collecting BWA-mem systems metrics**

**Next week’s task:**

1. **Test the cgroup for a scenario where the job will be slow due to memory restriction by the cgroup**
2. **Continue collecting BWA-mem systems metrics**
3. **Overview**

In the previous report we discussed a problem where memory allocation does not limit the maximum memory usage of a job, instead it limits the minimum memory required. The solution to this problem is cgroup. Last week, cgroup was activated on Slurm and tested on BWA-mem. The results show that cgroup can kill jobs due to lack of memory due to limited allocations. However, we are not satisfied enough because we cannot see that cgroup can limit memory causing jobs to slow down.

1. **How to configure cgroup**

Cgroup is a feature of the Linux kernel that limits resource usage within a node. Therefore, to limit resource usage using cgroup, it is necessary to configure each compute node. Configuration is done by applying changes to the slurm.conf and cgroup.conf files. In slurm.conf, cgroup activation is done by adding the value "task/cgroup" to the TaskPlugin variable. Then, limiting resources, especially memory, is done by setting the ConstrainRAMSpace value to the value "yes".

1. **Results**

After the cgroup has been successfully activated, we will try to collect system metrics from the BWA-mem tool with 16 threads. The read file used is SRR074979.fastq which is 3.9 GB in size.

Based on the data that has been obtained (see Figure 1), it is known that 100% of jobs with a memory allocation of less than 8GB fail due to OOM-Kill (out of memory kill). This happens because the average BWA-mem memory usage in this case is 9GB. As shown in Figure 2, the smallest memory usage by a job that does not fail is 9200 MB, so more than 9GB of memory is needed for the job to be completed.

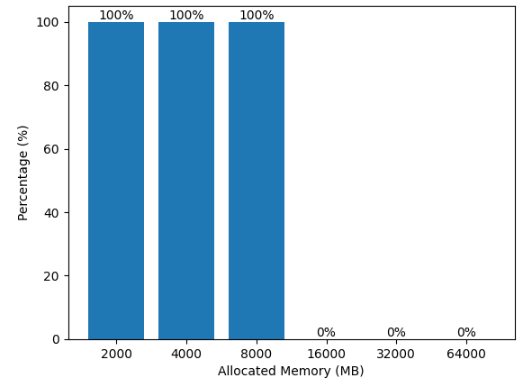


Figure 1. OOM-kill rate. All jobs whose allocation is less than 8GB will be killed.

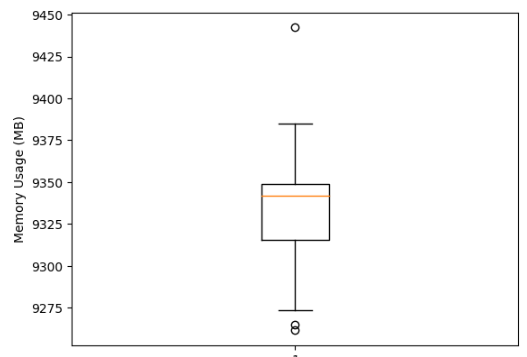


Figure 2. Memory usage boxplot of completed jobs. The smallest memory usage was 9200 MB.

1. **Discussion**

Keep in mind that our goal is to use cgroups to limit memory usage. We hope that with this limitation there are 2 possible scenarios:

1. Job was killed due to lack of memory
2. Job becomes slow due to lack of memory

Based on these 2 scenarios, we want to categorize genomics tools. So it must be ensured that cgroup can cause these 2 scenarios. However, the fact that the job is killed if the memory usage exceeds the allocation is not enough to conclude that cgroup is running as we expect. We can only point out that cgroup works as expected for the first scenario. We don't want every tool that should happen in scenario 2 instead to experience scenario 1 which causes inaccuracies in the research results. Therefore, we need to have prior knowledge of a tool that can suffer from scenario 2 (e.g. by observing memory usage over time) and try to run it to test cgroup. However, so far I still don't really understand big data processing techniques by RAM so it's quite difficult to be able to identify scenarios that are more likely to occur in a tool.