**BWA-Mem analysis**

1. Overview

In the previous report, modifications to the data collection script have been made. However, there is a reliability problem in the method used, so a new way is used by utilizing the /proc/pid system file. After that, BWA-mem metrics data was collected and it was found that the number of threads, RSS, and VM size were negatively correlated with elapsed time. Meanwhile, disk write and disk read show no correlation.

1. Method of collecting system metrics

In the previous system metrics collection mechanism, there were several problems. This problem causes the data to be less reliable. The problems are as follows:

1. The program fetches data from the top compute nodes and directly stores snapshot job statistics in a file on the master over time causing inconsistent snapshot intervals (there is delay time due to ssh protocol).
2. If the memory usage of a job is more than 1 GB, top will round the number to 1 decimal place so that the data tends to be uniform.

To solve this problem we need to collect data from the /proc/pid system file. The advantage of this method is that the numerical data are still raw (compared to the top) so that more detailed data can be obtained. In addition, there is information about IO in this file system that is not available on top and is always missing in slurm accounting. Then, the collected data is not directly sent to the master node but is compiled to the compute node first. Only when the system metrics of all jobs are recorded, the data that has been collected in the compute nodes is sent to the master node.

1. Experimental setup

The machine used in collecting system metrics is a Haswell instance of the Chameleon Texas server. This instance uses Intel(R) Xeon(R) CPU E5-2670 v3 @ 2.30GHz with detailed specifications in Table 1.

| CPUs | 48 |
| --- | --- |
| Threads per core | 2 |
| Cores per socket | 12 |
| Sockets | 2 |

Table 1. Haswell instance specifications.

1. Results

So far, the analysis that has been carried out is to see how the relationship between RSS, VM size, disk read, and disk write has on elapsed time. The analysis is carried out by making a scatter plot of the four metrics for the elapsed time for each file. In addition, the analysis was carried out by looking at the relationship between the number of threads and the elapsed time and RSS and VM size. And it was found that the number of threads, RSS and VM size were quite correlated with the elapsed time. Meanwhile, disk write and disk read do not appear to be correlated with elapsed time.

| RSS | the average use of physical memory by a job. |
| --- | --- |
| VM Size | average virtual memory usage by a job. |
| Thread | the number of threads in processing the job |
| Disk write | total size of data written to disk on job execution. |
| Disk read | total size of data read from disk on job execution. |

Table 2. Definition of the analyzed system metrics.

In Figure 1, the analysis is carried out by making a boxplot of the number of threads against the elapsed time for 32 CPU allocations in several files. In the figure, it can be seen that there is a negative correlation trend, where the greater the number of threads, the smaller the elapsed time. However, this correlation does not really look absolute, because in some conditions, such as SRR2059430 and SRR2059427 the elapsed time increases after passing a certain number of threads. So it can be assumed that for each file there is an optimum point for each thread.

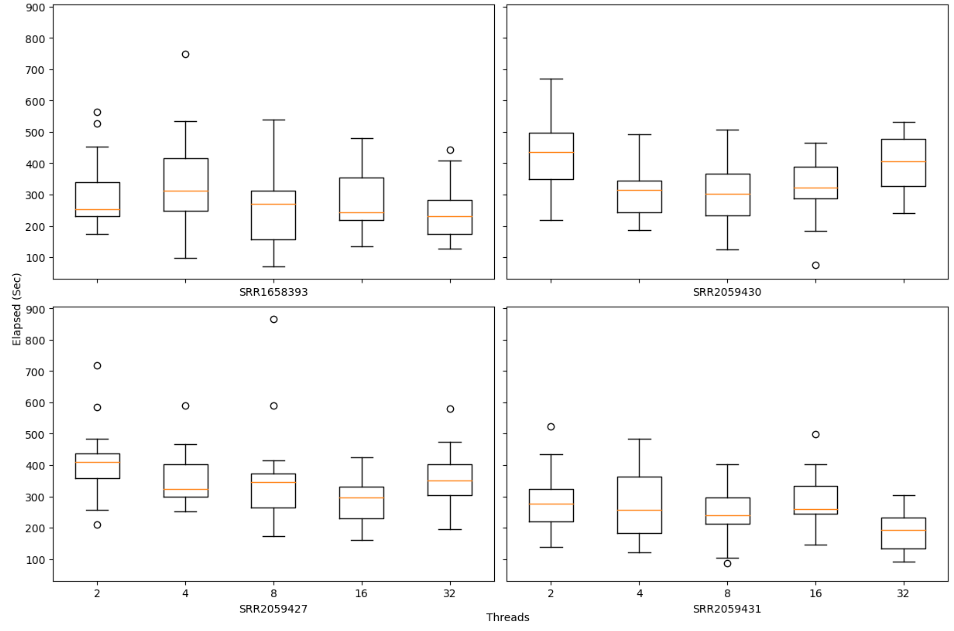


Figure 1. Boxplot of number of threads against elapsed time. Each file has a different optimal number of threads.

In Figure 2, it can be seen that RSS tends to have a negative correlation with elapsed time. That is, the greater the RSS value, the smaller the elapsed time. This is also indicated by the scattering of dots that is decreasing to the right. Similar to RSS, VM size also has a negative correlation with elapsed time. This negative correlation occurs because the use of RSS and VM size increases with the number of threads, as can be seen in Figure 6 and Figure 7. In other words, the positive correlation between RSS and VM size for threads causes a negative correlation with elapsed time. Meanwhile, write disks (Figure 5) and read disks (Figure 6) do not show a correlation with elapsed time. However, it is still not known exactly why this happens.

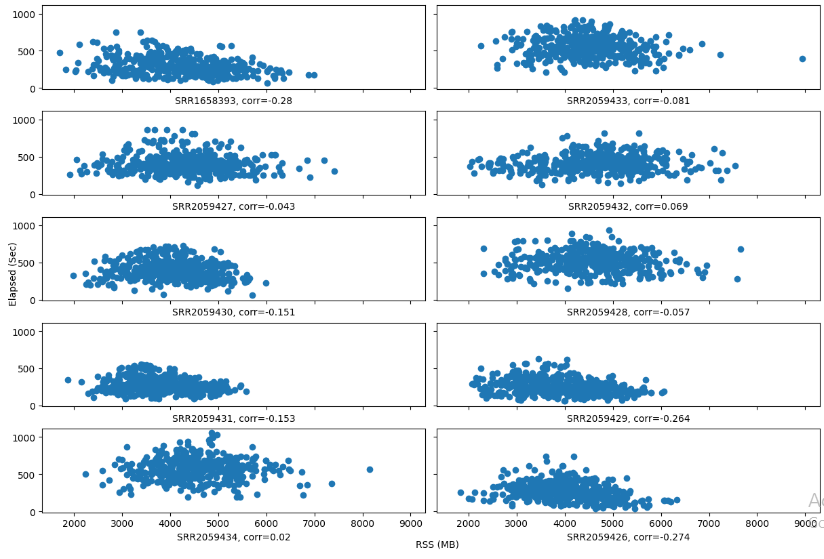


Figure 2. The correlation between RSS and elapsed time tends to be negative

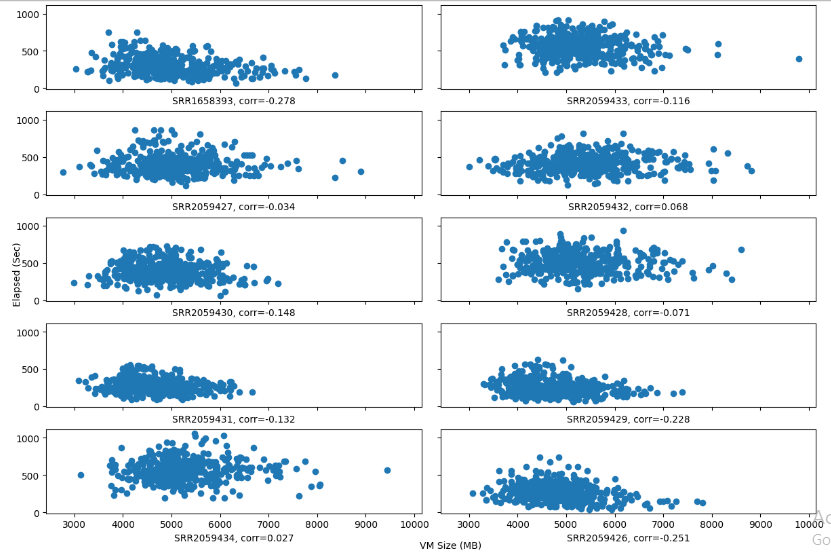


Figure 3. The correlation between VMSize and elapsed time tends to be negative

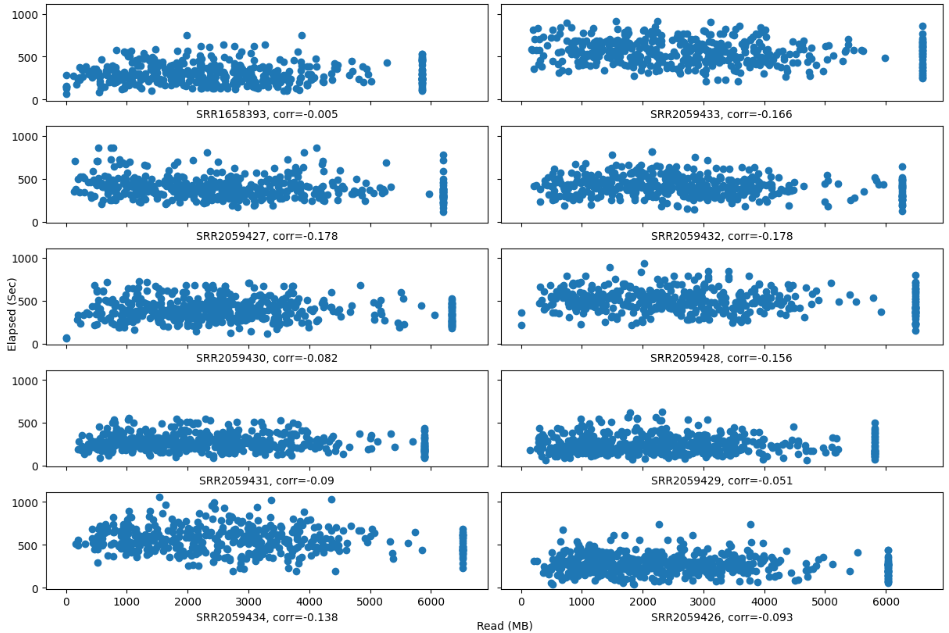


Figure 4. The correlation between disk write and elapsed time tends to be close to 0.

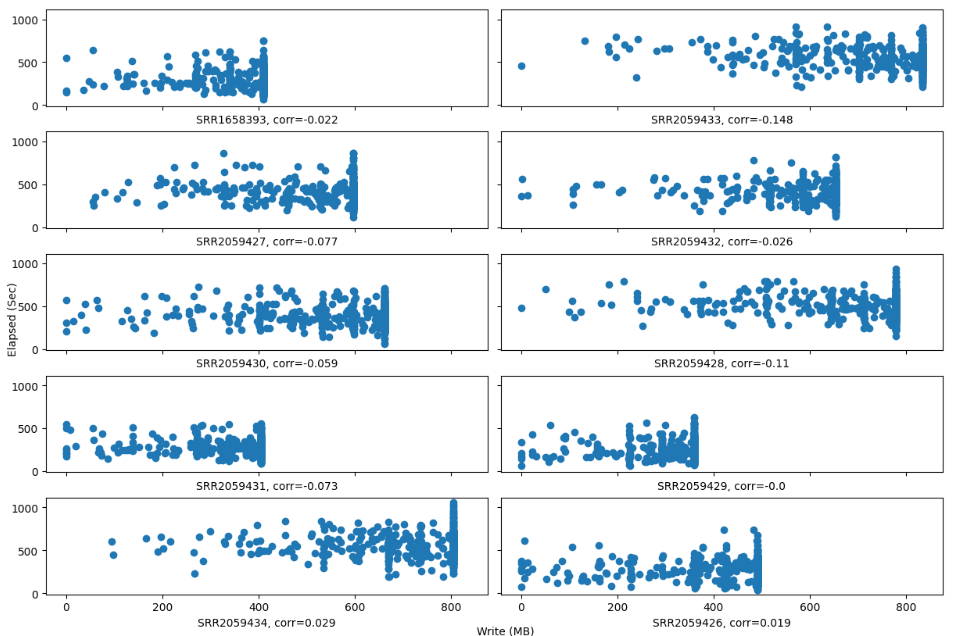


Figure 5. The correlation between disk write and elapsed time tends to be close to 0.

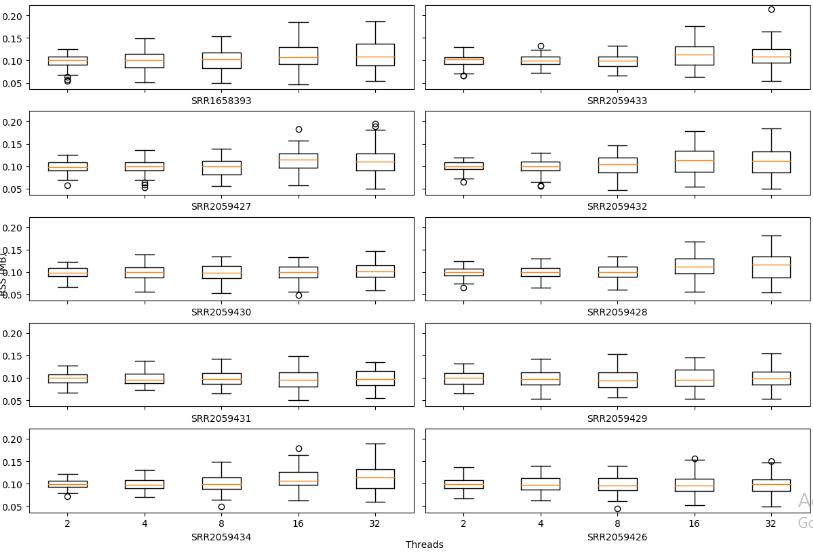


Figure 6. Positive correlation between number of threads and RSS.

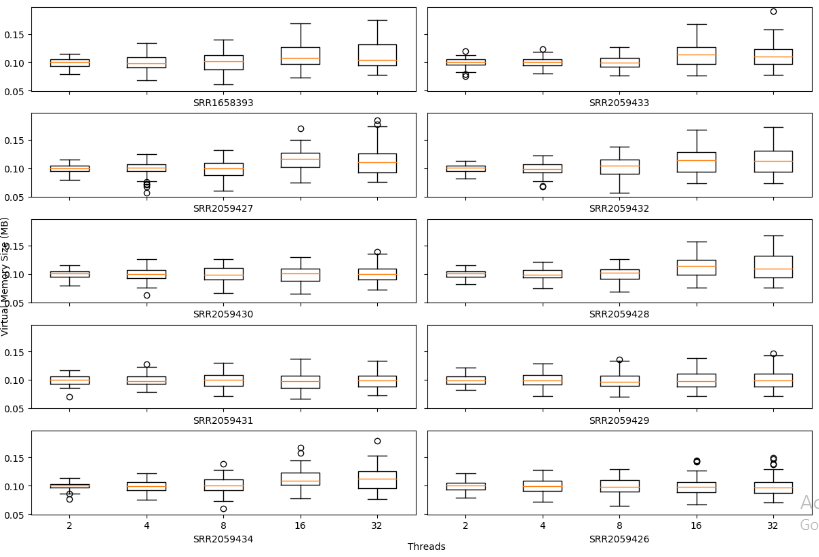


Figure 7. Positive correlation between number of threads and VM size.