Data Analysis of Jobs Run On the Titan Supercomputer from 2015 to 2019

Manas P P
Undergraduate at Department of Computer Science and Engineering, Amrita University, Kollam, Kerala, India
manas@am.students.amrita.edu

Abstract. In this paper, major characteristics of jobs run on the Titan supercomputer at the Oak Ridge National Laboratory (ORNL) from the years 2015 to 2019 has been studied in great detail. A rigorous analysis of the Resource Utilization Report (RUR) provided statistics and insights into the jobs that were run on Titan. A relationship between the CPU memory usage, GPU memory usage and the job size (node count) of a job is explored in this paper. User characteristics and errors encountered by the users is also studied. Success of jobs in the different file systems were noted. Jobs were also categorized based on the domain they belonged to and statistics based on domain provided insight into certain domains having much lesser success percentage when compared to the best performing domains.

Keywords: Titan Supercomputer, Resource Utilization, Data Analysis.

1 Introduction

Data collected on the various characteristics of a job run on a supercomputer can provide insights into the functioning of the hardware and the software of the system. Usage patterns of users, information about job failures, statistics of jobs based on the domain can help in the building of better scheduling algorithms as well as machine learning algorithms that improve reliability and user experience on High Performance Computing (HPC) systems. Sajal et al., [1] provides the general information of the challenges that can be solved by using the RUR and ProjectAreas datasets which were used for this study.

This paper provides a thorough study of the various characteristics of jobs run at Oak Ridge National laboratory (ORNL) on the Titan supercomputer using statistics and visualization. Data collected from the years 2015 to 2019 amass to 12,981,186 jobs from different domains of science like engineering, chemistry, physics, etc. I explored statistics of various job characteristics, user behavior, jobs based on domain and errors encountered by the users. Results from this observational study gives a thorough understanding of jobs run on Titan.

Cray XK7 Titan. Titan was a hybrid-architecture Cray® XK7™ system with a theoretical peak performance exceeding 27 PF and a sustained performance of 17.59 PF. It had 18,688 nodes, each with a 16-core AMD Opteron CPU having 32 GB memory and an NVIDIA K20x (Kepler) GPU with 6 GB memory. With 200 cabinets, 512 ser-

vices and I/O ports and 710 TB total system memory it has 8.9 MW peak energy measurement. The nodes are connected using Cray Gemini 3D Torus Interconnect. Titan employs the parallel distributed file system from Lustre and there were three different file systems under it – atlas, atlas1 and atlas2. Titan was decommissioned on August 2, 2019 [3].

Accessing the Job Statistics. At the conclusion of every job, Cray uses a revised NVIDIA API to query every compute node associated with a job, extracting the accumulated GPU usage and memory usage statistics on each individual node. By aggregating that information with data from the job scheduler, statistics is generated that describe the GPU usage on a per-job basis. Information like GPU seconds, command name, start time, end time, etc. is collected from multiple sources like the workload manager and Cray's RUR [4].

2 Related Work

The work in Wang et al.,[2] gives us an overall understanding of the RUR dataset. This paper furthers the study of jobs run on Titan. The codes for this study is made available to public at https://github.com/Manas641/RUR-Dataset-Analysis.

3 Data Overview

The Resource Utilization Report (RUR) dataset is collected from the Titan supercomputer at Oak Ridge National Laboratory from 2015 to 2019 using the Cray developed resource usage data collection and reporting system. A row in the dataset corresponds to the job's total resource utilization statistics. In this paper, the CPU memory usage is estimated using the feature "max_rss" which is the estimate of the maximum resident CPU memory used by an individual compute node through the lifespan of a job run. Every job is associated with a project ID (Area0 - Area60) found in the "command" feature. The ProjectAreas dataset provides the mapping of project ID to its science domain. However, many jobs did not have the file system, the user or the project information.

4 Findings

GPU Mode. There were two GPU modes available for a job, the "exclusive_process" and "default" modes. However, the "default" GPU mode made up just 0.00188% of the total jobs over the five year period. In figure 5, the graph shows an increasing success of jobs over the years for jobs with the "exclusive_process" mode.

The jobs in "default" mode had its best success rate as well as job count (Fig. 1, Fig. 5) in 2017 with 94.6% success compared to the 88.2% for "exclusive_process". The year 2016 witnessed the highest total number of jobs. The following years of 2017, 2018 and 2019 saw a lower number of jobs (Fig. 1).

GPU Usage. The annual ratio of tasks with zero GPU usage to the tasks with non-zero GPU usage was always higher than one. The ratio saw an increasing figure each year with 2019 seeing a 12.15 times increase over the previous year 2018 with a value of 90.55 (Fig. 2). The average GPU usage peaked in 2015 with a value of 54.9 GB per job with the second highest average coming in 2017 with a value of 51.57 GB. 2016 and 2018 observed much lower average values of 31.9 and 25.25 GB respectively. The year 2019 reported the lowest average GPU usage per job with just 0.181 GB (Fig. 2).

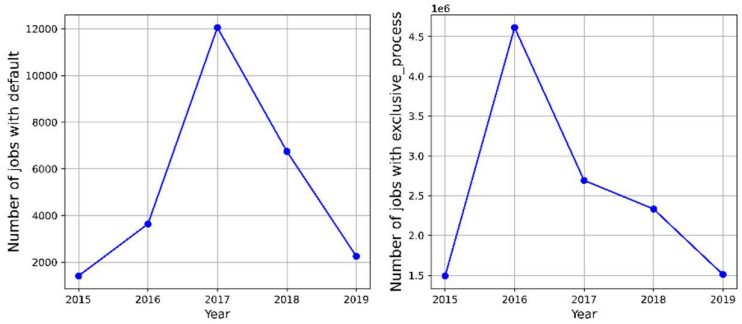

Fig. 1. (left) The number of jobs with GPU mode as "default" each year (right) The number of jobs with GPU mode as "exclusive_process" each year where 1 unit = 1e6, because of a small percentage of default jobs the total number of jobs follows the same plot.

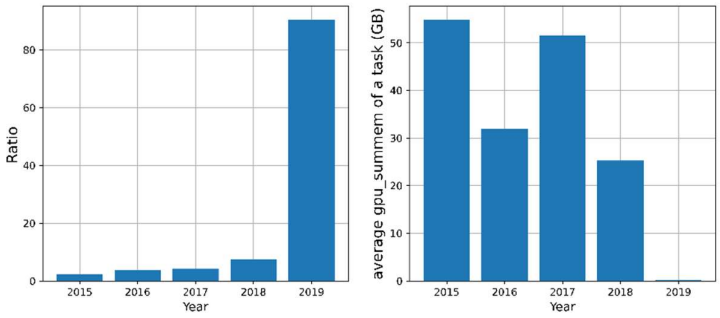

Fig. 2. (left) The annual ratio of jobs with zero GPU usage to jobs with non-zero GPU usage. (right) The annual averages of total GPU memory used by the jobs in GB.

Furthermore, a job without GPU usage tends to be smaller when compared to jobs with GPU usage. For example, the 75 percentile value of the time span of a job without GPU in 2015 was 40 times less than the corresponding value for jobs with GPU. Table 1 compares four important characteristics each year.

The average annual ratio of the amount of time spent on all the GPUs by a job to the total job time was highest in 2015 with a ratio of 18.35 followed by 2018 with 12.6. 2017, 2016 and 2019 had smaller ratios of 8.5, 5.6 and 2.8 respectively (Fig. 4).

Table 1. A Comparison of the 75 percentile values of four important job characteristics of jobs with and without GPU usage. (ttime is the total time spent on cpu)

75 percentile value	Time span(sec)	Max_rss(GB)	ttime (sec)	Node count
With GPU 2015	5298	1.3	30.9	32
Without GPU 2015	130	0.3	7.2	15
With GPU 2016	3682	2.187	104.24	94
Without GPU 2016	38	0.471	1.428	56
With GPU 2017	3542	0.58	141.30	54
Without GPU 2017	12	0.073	0.062	15
With GPU 2018	3342	0.45	113.05	240
Without GPU 2018	574	0.986	18.12	8
With GPU 2019	5264	5.99	115.33	12
Without GPU 2019	1301	0.192	7.91	2

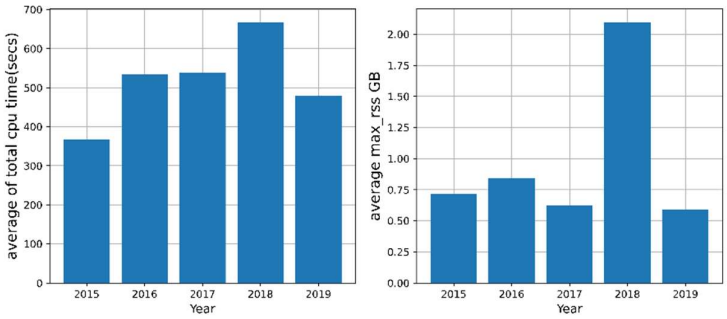

Fig. 3. (left) The annual averages of total time spent on CPU by a job in seconds. (right) The annual averages of max_rss of a job in GB.

CPU Usage. The average max_rss of the jobs was at the highest point in 2018 with a value of 2.0978 GB, which was at least double of the values in the other years (Fig. 3). The average total time spent by a job on the CPU saw a gradual increase over the years and peaked in 2018 at an average of 667 seconds for a task, which was followed by a decrease to 479 seconds in 2019 (Fig. 3). Although the mean was a few hundred seconds each year, the standard deviations were much higher. For example, in 2015, 75 percentile of the jobs had a total CPU time of 10.775 seconds with a standard deviation

of 18,429 seconds. The median was 0.07 seconds and the mode 0.000269 seconds, indicating a positively skewed distribution with a median skewness of 0.0757.

The average ratio of total CPU time to the total job time was a small value each year, with 2015 having the highest ratio of 0.18 while 2019 had just 0.088 (Fig. 4).

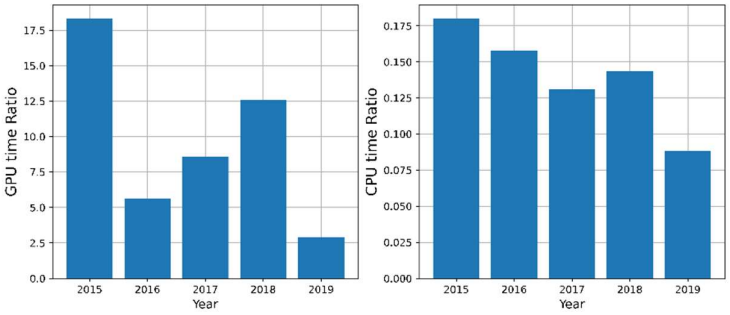

Fig. 4. (left) The annual average ratio of time spent on GPUs to the total job time. (right) The annual average ratio of time spent on CPUs to the total job time.

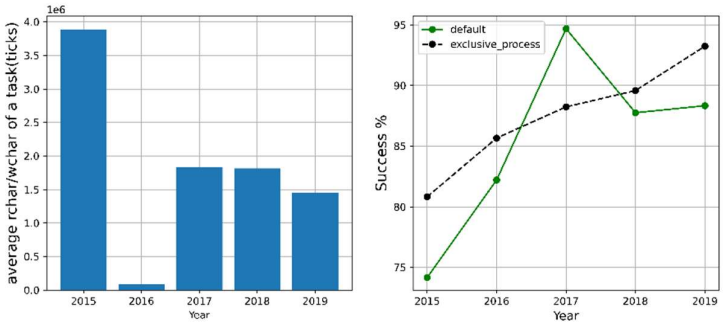

Fig. 5. (left) The annual average ratio of characters read to characters written where 1 unit = 1e6. (right) The success percentage of jobs using the two GPU modes each year.

Read to Write Ratio. The average ratio of characters read to characters written witnessed a very high value each year, with 2016 having a relatively smaller value when compared to the other years (Fig. 5).In addition, tasks that were successful had a much higher ratio than the tasks that were not successful (Table 2).

Users. Titan saw an increase in the number of users each year, with 2018 having the highest number of users at 850. Now let's consider the statistics each year based on the four quartiles (0-25, 25-50, 50-75 and 75-100 percentiles) of the job time. We observe in Fig. 6 that in the year 2015 the percentage of users (unique users) involved in jobs in the 50-75 percentile was higher than the percentage in the other three divisions. 2016 and 2017 saw the percentage at its highest in the 75-100 percentile range. However, in

the years 2018 and 2019 we see that percentage kept decreasing with the higher job time percentiles and 2018 recorded the percentages at its lowest values for the 50-75 and 75-100 percentiles.

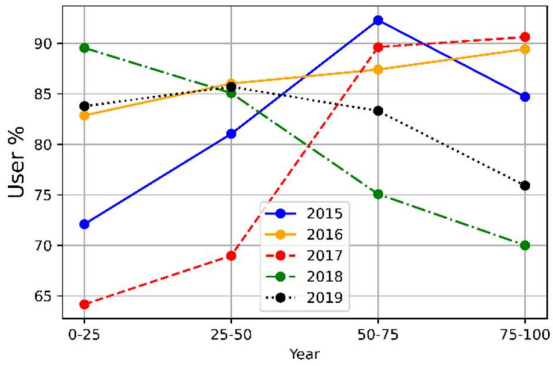

Fig. 6. The percentage of users that ran jobs in each quartile of job times. We see after 2017, the percentage of users is decreasing for the higher percentiles. This shows there are fewer users in the years 2018 and 2019 who haven't run jobs in the lower quartiles.

The average ratio of time spent in user mode to time spent in kernel mode of the jobs after the year 2016 was higher for successful jobs when compared to unsuccessful jobs. The years 2015 and 2016 had observed the value being smaller for successful jobs (Table 3).

Table 2. The annual average ratio of characters read to characters written based on job success.

Year	Unsuccessful tasks	Successful tasks	Overall tasks
2015	448,372	4,715,308	3,889,537
2016	71,583	82,783	81,057
2017	84,592	2,074,072	1,835,697
2018	1,13,411	2,012,885	1,813,590
2019	1,80,568	1,538,249	1,445,683

Table 3. The annual average ratio of job time spent in user mode to time spent in kernel mode.

Year	Ratio (Unsuccessful tasks)	Ratio (Successful tasks)
2015	58.289	37.242
2016	66.365	44.730
2017	36.761	61.277
2018	43.138	124.681
2019	36.509	394.969

The distribution of the number of users with a particular success rate in a year can be modelled by using a curve based on equation 1. For each year, a curve could be fit that best describes the estimated number of users with a particular success ratio in that year. We use the histogram data, i.e., the bin locations (ratio of successful jobs to total jobs) and data entries (number of users) and fit a curve using least squares fit using the function in equation 1 (Fig. 7).

$$F(x) = (a * e^{(b*x)}) + c \quad (1)$$

The values of a, b and c over the years are shown in Table 4.

Errors. The different types of exit conditions (excluding 0) encountered peaked in 2016 with 104 different exit conditions and the following years saw a decline in the types of exit conditions encountered. The most frequently occurring exit conditions each year were 0, 137, 1, 127, 143, 139, 2, 130, 255 and 134. The percentage of successful jobs increased over the years; however, 2016 which had the largest job count reported the highest number of errors. Almost two-thirds of the errors in 2016 were the error codes 127 and 137 with the total errors at 662,095. A child process returns error 127 when a command is not found. Error 137 is the SIGKILL termination signal (9) that usually indicates that the application ran out of memory.

Table 4. The values of a, b and c of equation 1 in each year.

Year	a	b	c
2015	4.40	2.79	11.02
2016	9.33	2.49	-4.33
2017	6.54	2.87	-0.46
2018	0.03	8.50	20.32
2019	0.02	8.09	9.66

Relationship between CPU Memory Usage, GPU Memory Usage and Job Size. The pearson correlation coefficient between logarithm of node count and logarithm of total gpu memory used by a job with non-zero GPU usage took the values of 0.8677, 0.8725, 0.8895 and 0.8896 in the years of 2015 to 2018 respectively. However correlation of the two variables in the year 2019 was 0.4675. The year 2019 had a large number of jobs with zero GPU usage and the graph for jobs with GPU usage was highly dispersed.

A linear regression model for jobs with non-zero GPU usage can give a good prediction of the logarithm of node count of a job given its logarithm of max_rss and logarithm of the total gpu memory used. The regression model was built on data from 2015 and gave the relation in equation 2 (Fig. 9).

$$\log node\ count = (-0.4846 * \log \max rss) + (0.8596 * \log gpu\ summem) \quad (2)$$

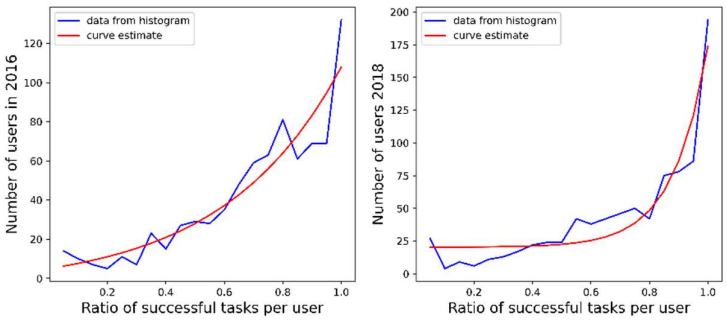

Fig. 7. (left) An exponential curve estimates the number of users who had a particular ratio of successful tasks to total tasks in 2016. Similar gradually increasing curves were seen for 2015 and 2017 as well. (right) The curve fit for users in 2018. 2019 saw a similar curve with a steep slope after 0.8 showing greater number of users with high success rates.

The model gave an adjusted-R^2 score of 0.831, 0.844 and 0.855 for the years 2016, 2017 and 2018. The year 2019 gave a poor result of -2.447. Upon plotting the points in 2019 it was observed that the points were much fewer and deviated from the plane that was observed every other year. 2019 saw the launch of Summit supercomputer at ORNL and the jobs in that year were fewer and not consistent with the other years in terms of the job feature trends. For this reason this paper treats the year 2019 as an exception for this model, which performed well on the other years.

File Sytems in Lustre. Among the jobs which had a science department assigned to it, the file_system had three categories atlas, atlas1 and atlas2. In 2017 there was another category encountered, the atlas1_thin (Table 5). Atlas2 has the highest success percentage over the years with at least 91% each year and an impressive 98.67 % in 2019. Atlas1_thin had only 254 jobs in 2017 and recorded a success percentage of 85.433% (Fig. 8).

Table 5. The job count (in thousands) in each of the file system under Lustre. NA referes to jobs with no file system assigned to it in the command.

File system	2015	2016	2017	2018	2019
NA	895.9	3359.8	1411.6	1737.7	1229.1
atlas	404.9	430.7	236.7	317.6	149.8
atlas1	25.9	30.6	14.8	42.2	9.8
atlas2	165.2	795.7	1040.9	241.7	121.5
atlas1_thin	0	0	0.254	0	0

Science. Titan had projects from nine fields of science and each field further had areas in them, which made up 29 in total. Sciences enjoyed variable levels of success

as shown in Table 8. No job in Chemistry was encountered in the year 2016. Considering the years 2015 to 2018, Chemistry had the least successful jobs over the four-year period followed by Physics and Nuclear.

The jobs in all the sciences over the 5 year period saw jobs with a higher time spent in user mode than in kernel mode having a higher average job time of 3669 seconds when compared to 458.8 seconds when the ratio was less than 1, showing a greater reliance on user mode for larger jobs (Table 6).

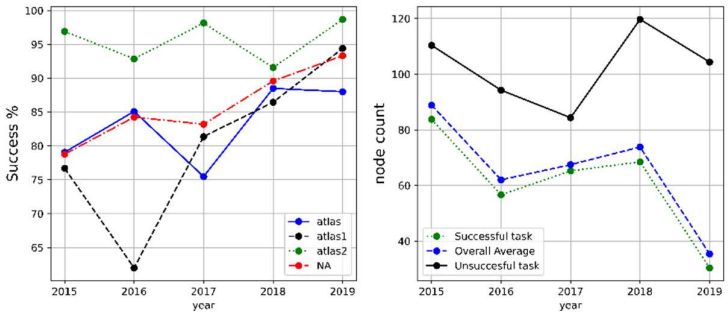

Fig. 8. (left) The annual success percentage of jobs run in each file system. (right) The annual average node count of tasks based on success or failure.

The average node count of unsuccessful tasks was higher than the average node count of overall tasks and the average for successful tasks each year (Fig. 8).

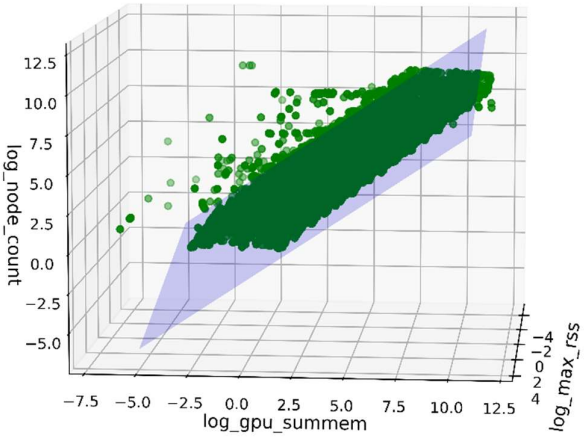

Fig. 9. The plot for the model predicting log of node count of a job using log of max_rss of the job and log of total gpu memory used by the job.

Areas. The RUR datasets also provided an ID for each job based on the area of science it belonged. Table 7 shows the three least successful Areas each year. No area repeated in the table for more than two consecutive years.

Table 6. The annual average job times of jobs with science domain based on the ratio of time spent in user mode to time spent in kernel mode.

Year	Utime/stime < 1	Utime/stime > 1
2015	1838.657	4276.664
2016	76.380	4148.029
2017	211.662	2674.864
2018	90.751	3024.673
2019	76.557	4223.082

Table 7. Least successful areas of science each year and the success percentage separated by '-'

2015	Systems Biology- 0%	Nano- Science- 34.7%	Turbulence- 35.7%
2016	Nano Electronics- 22.5%	Nano Science- 25%	Nuclear Fuel Cycle- 35.5%
2017	Nuclear Physics- 30.1%	Vendor- 39.6%	Biophysics- 41.2%
2018	Turbulence- 9.4%	Vendor- 21.6%	Condensed Matter Physics- 32.8%
2019	General (CS)- 8.5%	Medical Science- 50.7%	High Energy Physics- 51.6%

5 Discussion

The year 2016 witnessed the highest number of jobs. This was due to a manufacturing defect in the GPUs that led to smaller but larger number of jobs [2]. After this issue was identified, there was a replacement of nearly half the GPUs of Titan and also the placement of a GPU-aware algorithm by July 2017 [5]. Following this there was a decreased job count and also an increased success rate. A finding from this study that coincides with this is the increased time spent on user mode than in kernel mode after 2016. In Table 3 it can be seen that after 2016 for jobs that were successful, the ratio of time spent in user mode to the time spent in kernel mode was nearly double in 2017, nearly triple in 2018 and was 10.8 times in 2019 when compared to the jobs for unsuccessful tasks.

A large number of jobs did not have the information of the domain it belonged to. The case of a higher dependence on user mode is even more significant for jobs which had a science domain associated with it. Table 6 clearly shows that users preferred to run bigger jobs mainly in the user mode.

Table 8. Annual success percentages of job based on Science domain

Science	2015	2016	2017	2018	2019
Biology	62.37	50.97	50.85	82.16	69.62
Chemistry	58	-	42.74	51.31	89.36
CS(computer science)	91.74	93.93	97.73	89.67	97.39
Earth Science	90	93.38	87.12	95.32	98.24
Engineering	62.94	68.39	79.74	88.85	95.51
Fusion	92.53	86.60	80.24	95.97	96.15
Materials	59.99	84.34	95.05	92.69	88.96
Nuclear	59.15	79.56	73.88	68.93	100
Physics	93.46	70.66	35.35	60	67.88

An increased number of users ran test/development models in the later years, especially 2018 and 2019. This can be observed in figure 6 where we notice that the ratio of users in the 75 – 100 percentile of job time was higher than the lower quartiles up to 2017. This indicates the presence of users who only ran jobs in the 75 – 100 percentile. Since 2018, the number of users in the higher percentiles decreased which indicates fewer users who deployed large models without proper testing. Table 9 supports this point, as we can see a higher success rate in the 75 – 100 percentile in 2018 and 2019 when compared to previous years and also the success rates in the lower quartiles.

Table 9. Annual Success percentages of jobs in each quartile of job time each year

Year	0-25 percentile	25-50 percentile	50-75 percentile	75-100 percentile
2015	93.67	75.68	79.77	72.06
2016	72.39	91.31	92.24	88.49
2017	98.32	78.15	83.03	92.06
2018	88.30	81.77	92.63	95.57
2019	89.02	92.94	93.70	97.51

Because of a heavy dependence of jobs on the GPU mode “exclusive_process”, the overall results depended on these jobs. The plot for the overall success percentage of jobs overlaps (to the naked eye) the plot for the jobs with “exclusive_process” in Fig. 5. Considering all the years from 2015 to 2019 the error 137 was the most encountered error. This error which usually means that the application ran out of memory also had

the highest average node count which was consistently above 200 nodes each year. No other error had a similar high average node count each year.

6 Conclusion

A thorough study of the jobs run on the Titan supercomputer was covered in this paper. The data used for the study was collected over a span of five years from 2015 to 2019 by ORNL. I also show the presence of a relation between the logarithms of CPU memory usage, GPU memory usage and the node count. A linear regression model was built on data from 2015 that gave good results for the data of remaining years, except 2019 which was an exception. The statistics from this study can be used for documenting the jobs on Titan and the results can be leveraged for building machine learning models for job scheduling and improving user experience. We also see that jobs in the Computer Science domain was very successful, while jobs in domains like Chemistry, Physics, Nuclear and Biology were not at par with Computer Science. The lack of high success rates in these domains can be investigated and jobs in these domains can be given specific assistance based on the results.

Acknowledgement. Analyzing Resource Utilization and User Behavior on Titan Supercomputer is one of challenges of the 5th Annual Smoky Mountains Computational Sciences Data Challenge hosted to tackle scientific data challenges using datasets from ORNL. Support for DOI 10.13139/OLCF/1772811 dataset is provided by the U.S. Department of Energy, project GEN150 under Contract DE-AC05-00OR22725. Project GEN150 used resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

1. Sajal, D., et al.: Analyzing Resource Utilization and User Behaviour on Titan Supercomputer, DOI: 10.13139/OLCF/1772811

2. Wang, F., et al.: Learning from Five-year Resource-Utilization Data of Titan System, DOI: 10.1109/CLUSTER.2019.8891001

3. ORNL https://www.olcf.ornl.gov/olcf-resources/compute-systems/titan/

4. Measuring GPU Usage on Cray XK7 using NVIDIA's NVML and Cray's RUR https://cug.org/proceedings/cug2014_proceedings/includes/files/pap196-file2.pdf

5. Zimmer, C., et al.: GPU age-aware scheduling to improve the reliability of leadership jobs on Titan, Published in SC18, DOI: 10.1109/SC.2018.00010