Report Template

I. Introduction

High-Performance Computing (HPC) systems generate a massive amount of data. The data generated on these supercomputers are the resource use data and system logs. These data has been shown to be very valuable for the development of HPC failure prediction tools [1]–[10]. In this assignment, I will describe the resource use data obtained on a large HPC system, validate three linear regression models and discuss the findings.

II. Case Study: Ranger HPC System

The Ranger HPC system was operated from 2007 to 2013 by the Texas Advanced Computing Center at The University of Texas at Austin. It has 4,048 nodes that provided batch job processing and data storage services. Job resource usage is monitored by the TACC_Stats resource usage monitor [11]. TACC_Stats monitors the resource usage by jobs and nodes and it performs the monitoring online. The values of all the resource use counters are aggregated at a default time interval of 10 minutes. The values for the resource use counters are set to zero only when a node is reset. A resource use log is given below:

```
20665 Aug 16 15:20:01 i151-312 eth0 rx_bytes
424689 tx_bytes 25178
```

The resource use log contains the following fields: job-id (`20665`), time-stamp (`Aug 16 15:20:01`), node-id (`i151-312`), device group (`eth0`) and the resource use counter and its value (`rx_bytes 424689`, `tx_bytes 25178`). `eth0` represents the network interface card, `rx_bytes 424689` represents the amount of bytes received and `tx_bytes 25178` represents the amount of bytes transmitted. The resource use data contains 410 resource use counters that are divided into 9 groups of devices. The resource use counters and their device groups are given in Table I.

TABLE I
List of resource use counters and their device groups.

Device group	Qty.	Resource use counters
Lustre network	6	tx_msgs, rx_msgs, rx_msgs_dropped, tx_bytes, rx_bytes, rx_bytes_dropped
Lustre /work, /share, /scratch	23 23 23	read_bytes, write_bytes, direct_read, direct _write, dirty_pages_hits, dirty_pages_misses, ioctl, open, close, mmap, seek, fsync, setattr, truncate, flock, getattr, statfs, alloc_node, setxattr, getxattr, listxattr, removexattr, inode_permission
Virtual memory	21	pgpgin, pgpgout, pswpin, pswpout, pgalloc-_normal, pgfree, pgactivate, pgdeactivate, pgfault, pgmajfault, pgrefill_normal, pgsteal _normal, pgscan_normal, pgscan_direct_no-rmal, pginodesteal, slabs_scanned, kswapd _steal, kswapd_inodesteal, pageoutrun, allocstall, pgrotated
Block md0, hdd	11 11	rd_ios, rd_merges, rd_sectors, rd_ticks, wr_ios, wr_merges, wr_sectors, wr_ticks, in_flight, io_ticks, time_in_queue
Cpu 0 to 15	112	user, nice, system, idle, iowait, irq, softirq
Mem 0 to 3	80	MemTotal, MemFree, MemUsed, Active, Inactive HighTotal, HighFree, LowTotal, LowFree, Dirty, Writeback, FilePages, Mapped, AnonPages, PageTables, NFS_Un-stable, Bounce, Slab, HugePages_Total, HugePages_Free
Net ib0, lo, eth0,	23 23 23	collisions, multicast, rx_bytes, rx_compressed, rx_crc_errors, rx_dropped, rx_errors, rx_fifo _errors, rx_frame_errors, rx_length_errors, rx _missed_errors, rx_over_errors, rx_packets, tx_aborted_errors, tx_bytes, tx_carrier_errors, tx_compressed, tx_dropped, tx_errors, tx_fifo _errors, tx_heartbeat_errors, tx_packets, tx_window_errors
Numa 0 to 3	24	numa_hit, numa_miss, numa_foreign, interleave_hit, local_node, other_node
Ps	7	ctxt, processes, load_1, load_5, load_15, nr_running, nr_threads

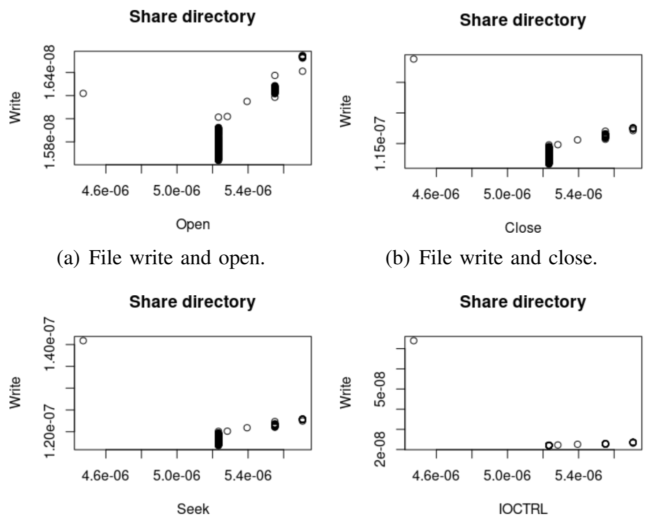
The Lustre filesystem is an object-based high-performance networked filesystem that is designed for high-throughput I/O tasks and it is commonly used to provide high-speed data I/O on many HPC systems. However, Lustre filesystem I/O problems such as resource contention [12] and hung Lustre clients [13] were widely reported. I have defined the following research question to provide an insight into resource usage on the Lustre filesystem:

- Can we predict file system writes given the file open, close, seek and I/O control resource use counters on the Lustre filesystem's share partition?

III. Evaluation Results

To begin my analysis of the Lustre filesystem I/O resource use counters, I generated scatter plots to visualize the relationship between (a) file write and open resource use counters, (b) file write and close resource use counters, (c) file write and seek resource use counters and (d) file write and ioctrl resource use counters. The scatter plots are shown in Fig. 1. From Fig. 1(a), we observed that the relationship between the file write and open resource use counters is relatively strong. From Fig. 1(b), Fig. 1(c) and Fig. 1(d), we observed that the relationships between the file write and close, file write and seek and file write and ioctrl resource use counters are weak.

Next, I used a simple linear regression model to obtain the best fit line for the file write and open, file write and close, file write and seek and file write and ioctrl resource use counters. My objective is to determine whether the file open, close, seek or ioctrl resource use counter can be used as a predictor. Fig. 2



(a) File write and open. (b) File write and close.

(c) File write and seek. (d) File write and io control.

Fig. 1. Scatter plots for file write, open, close, seek and ioctrl resource use counters.

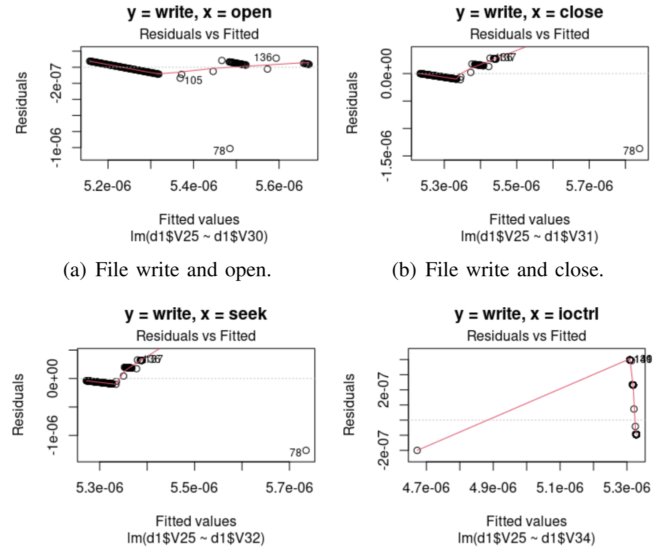
shows the residuals vs fitted values. For the file write and open resource use counters, an adjusted R^2 value of 0.68 or 68% of the variation in file open can be explained by the model. For the file write and close resource use counters, an adjusted R^2 value of 0.17 or 17% of the variation in file close can be explained by the model. For the file write and seek resource use counters, an adjusted R^2 value of 0.07 or 7% of the variation in file seek can be explained by the model. For the file write and ioctrl resource use counters, an adjusted R^2 value of 0.09 or 9% of the variation in file ioctrl can be explained by the model.

> While the prediction scores from the regression models for the file close, file seek and file ioctrl resource use counters are low, the prediction score from the regression model for the file write and open resource use counters is quite high, indicating that file open can be used as a predictor for file writes.

A. Multiple Linear Regression

The first phase of my analysis is characterized by fitting a simple linear regression model to the Lustre filesystem I/O resource use counters. I specifically observed that the file open resource use counter can predict file write fairly reliably. However, my objective is to compare regression models and choose the best one. To achieve this, I used multiple linear regression to obtain a better model.

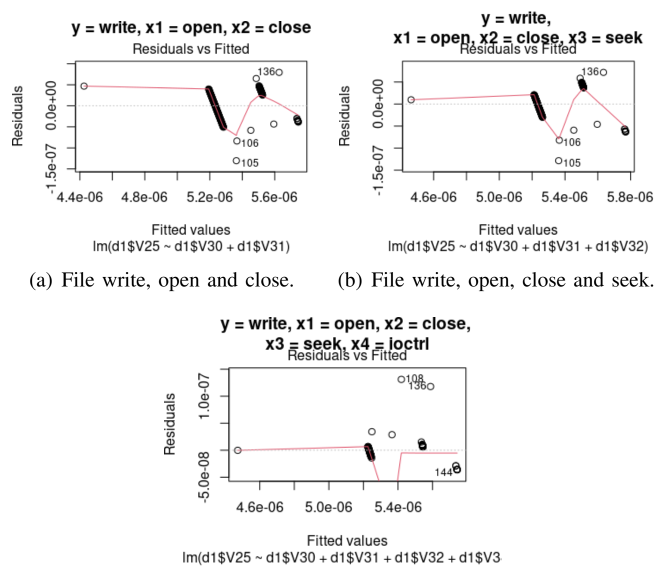
Fig. 3 shows the residuals vs fitted values for the multiple linear regression models. For the first model in Fig. 3(a), an adjusted R^2 value of 0.96 or 96% was obtained. For the second model in Fig. 3(b), an adjusted R^2 value of 0.97 or 97% was obtained. For the third model in Fig. 3(c), an adjusted R^2 value of 0.98 or 98% was obtained. The change in the variation that can be explained between those models is 1%.



(a) File write and open. (b) File write and close.

(c) File write and seek. (d) File write and io control.

Fig. 2. Residuals vs fitted values of file write and open, file write and close, file write and seek and file write and ioctrl operations.



(a) File write, open and close. (b) File write, open, close and seek.

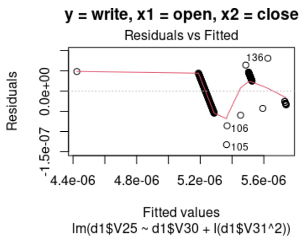
(c) File write, open, close, seek and ioctrl

Fig. 3. Residuals vs fitted values for file write, open, close, seek and ioctrl resource use counters.

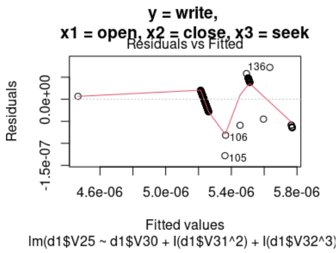
> All three multiple linear regression models performed well on the data. The multiple linear regression model that was trained on four features produced the highest prediction score, indicating that it is the best model for predicting file writes.

B. Polynomial Regression

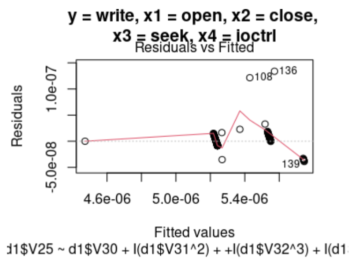
The second phase of my analysis is characterized by fitting a multiple linear regression model to the Lustre filesystem I/O resource use counters. I specifically observed that all the three multiple linear regression models performed well on the data. However, my objective is to compare different regression models and choose the best one. To achieve this, I used polynomial regression to obtain the best model.



(a) File write, open and close.



(b) File write, open, close and seek.



(c) File write, open, close, seek and ioctrl

Fig. 4. Residuals vs fitted values for file write, open, close, seek and ioctrl resource use counters.

Fig. 4 shows the residuals vs fitted values for the polynomial regression models. For the first model in Fig. 4(a), an adjusted R^2 value of 0.95 or 95% was obtained. For the second model in Fig. 4(b), an adjusted R^2 value of 0.96 or 96% was obtained. For the third model in Fig. 4(c), an adjusted R^2 value of 0.98 or 98% was obtained. Between the three models, the change in the variation that can be explained by those models range between 1% to 2%.

All three polynomial regression models performed well on the data. The polynomial regression model that was trained on four features produced the highest prediction score, indicating that it is the best model for predicting file writes.

C. Discussion

From these results, I showed that a multiple linear regression model trained on the file open and close resource use counters is suitable as a model to predict file writes. My analysis over the resource use data on a large HPC system helps to become cognizant of the extent to which the filesystem I/O writes may be predicted by other resource use counters. The fact that many resource use counters are not the primary indicators of file writes is not obvious, for example, an increase in the number of resource use counters used to train both the multiple linear regression model and polynomial regression model does not necessarily produce a better prediction model. The model prediction scores are summarized in Table II.

TABLE II
MODEL PREDICTION SCORE

No. of features	Simple linear regression	Multiple linear regression	Polynomial regression
1	68%	–	–
2	–	96%	95%
3	–	97%	96%
4	–	98%	98%

I observed that the file close, seek and ioctrl resource use counters are weakly correlated to file writes on the Lustre filesystem and these resource use counters did not increase the model's prediction score significantly. While systems administrators are less concerned with models which do not produce a high prediction score, it is better to equip the resource usage monitor and failure predictors to be aware of early signs of resource contention to reduce service downtime. Machine learning-based failure predictors from resource use data can further help in identifying impending filesystem failures [14]. These findings are suitable for diverse filesystems as well, since parallel filesystems, for example, IBM GPFS can also benefit from resource use data analysis.

REFERENCES

[1] S. Fu and C.-Z. Xu, "Exploring event correlation for failure prediction in coalitions of clusters," in *Proceedings of ACM/IEEE Supercomputing (SC)*, no. 41, 2007.

[2] E. W. Fulp, G. A. Fink, and J. N. Haack, "Predicting computer system failures using support vector machines," in *Proceedings of 1st USENIX Workshop on the Analysis of System Logs*, 2008.

[3] Y. Liang, Y. Zhang, H. Xiong, and R. Sahoo, "Failure prediction in ibm bluegene/l event logs," in *Proceedings of IEEE International Conference on Data Mining (ICDM)*, 2007, pp. 583–588.

[4] F. Salfner, P. Troeger, and S. Tschirpke, "Cross-core event monitoring for processor failure prediction," in *Proceedings of IEEE International Conference on High Performance Computing Simulation*, 2009, pp. 67–73.

[5] Z. Zheng, Z. Lan, R. Gupta, S. Coghlan, and P. Beckman, "A practical failure prediction with location and lead time for blue gene/p," in *1st Workshop on Fault-Tolerance for HPC at Extreme Scale (in conjunction with IEEE/IFIP DSN 2010)*, 2010.

[6] A. Gainaru, F. Cappello, M. Snir, and B. Kramer, "Fault prediction under the microscope: A closer look into HPC systems," in *Proceedings of IEEE/ACM Supercomputing (SC)*, 2012, pp. 1–12.

[7] A. Pelaez, A. Quiroz, J. C. Browne, E. Chuah, and M. Parashar, "Online failure prediction for HPC resources using decentralized clustering," in *Proceedings of IEEE HiPC*, 2014, pp. 1–9.

[8] X. Fu, R. Ren, S. A. McKee, J. Zhan, and N. Sun, "Digging deeper into cluster system logs for failure prediction and root cause diagnosis," in *Proceedings of IEEE Cluster*, 2014, pp. 103–112.

[9] A. Das, F. Mueller, P. Hargrove, E. Roman, and S. Baden, "Doomsday: Predicting which node will fail when on supercomputers," in *IEEE/ACM Supercomputing (SC)*, 2018.

[10] A. Das, F. Mueller, C. Siegel, and A. Vishnu, "Desh: Deep learning for system health prediction of lead times to failure in HPC," in *ACM HPDC*, 2018.

[11] R. T. Evans, J. C. Browne, and W. L. Barth, "Understanding application and system performance through system-wide monitoring," in *Proceedings of IEEE IPDPS Workshops*, 2016, pp. 1702–1710.

[12] J. C. Browne, R. L. DeLeon, C. D. Lu, M. D. Jones, S. M. Gallo, A. Ghadersohi, A. K. Patra, W. L. Barth, J. Hammond, T. R. Furlani, and R. T. McLay, "Enabling comprehensive data-driven system management for large computational facilities," in *Proceedings of ACM Supercomputing (SC)*, Nov 2013, pp. 1–11.

[13] J. L. Hammond, T. Minyard, and J. Browne, “End-to-end framework for fault management for open source clusters: Ranger,” in *Proceedings of ACM TeraGrid Conference*, no. 9, 2010.

[14] S. Lu, B. Luo, T. Patel, Y. Yao, D. Tiwari, and W. Shi, “Making disk failure prediction SMARTer!” in *Proceedings of USENIX Conference on File and Storage Technologies (FAST)*, 2020, pp. 151–167.