Leveraging Neural Networks to Predict Job IO in HPC Systems

Michael R. Wyatt II
University of Delaware
18 Amstel Ave
Newark, DE 19701
mwyatt@udel.edu

Todd Gamblin (Advisor)
Lawrence Livermore National Laboratory
Livermore, CA 94550
gamblin2@llnl.gov

Adam Moody (Advisor)
Lawrence Livermore National Laboratory
Livermore, CA 94550
gamblin2@llnl.gov

Michela Taufer (Advisor)
University of Delaware
18 Amstel Ave
Newark, DE 19701
taufer@udel.edu

ABSTRACT

The Lustre File System is a shared resource for many HPC systems. It is liable to crash if it cannot handle the IO needs of an entire cluster. Predicting job IO performance allows schedulers to avoid running jobs which may overload the Lustre File System. We use a neural network to predict job execution features related to the Lustre File System on several high performance clusters. A combination of neural network types (i.e., 2D convolution, embedding, and deep) are used to capture information found in the regular and irregular patterns of a user job submission script. We examine the ability of the neural network to predict job features for several months of job submissions. Results indicate that the neural network captures information from job scripts and utilizes other data from the scheduler to provide accurate predictions of several job performance features.

1. MOTIVATION

Application performance is reliant on the cluster, other running jobs, and shared resources. The Lustre File System (LFS) is a shared resource for one or more clusters. The LFS consists of several Object Storage Targets (OST) and a Meta-Data Server (MDS). Information about files stored on the OST is maintained by the MDS. When applications on a cluster need to interact with files in the LFS, they must first consult with the MDS. Figure 1 shows the structure of the LFS. If the MDS cannot support the number of metadata operations needed by a cluster, it is liable to become unresponsive and crash [1]. To avoid LFS crashes and maintain a high throughput of applications on a cluster, knowledge of application runtime and metadata operations is necessary for a smarter job scheduler. If runtime and metadata operations can be accurately predicted, the job scheduler will avoid running IO-intensive operations simultaneously.

SC '16 November 13–18, 2016, Salt Lake City, UT, USA

DOI: 10.1145/1235

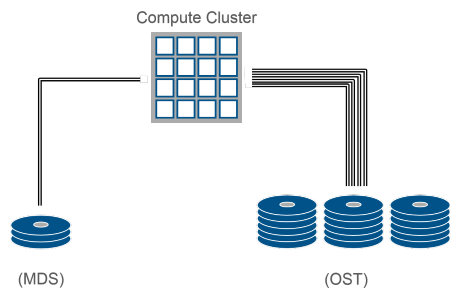

Figure 1: The Lustre File System.

2. CONTRIBUTIONS

Previous efforts to predict features of applications (e.g., queue time) have used machine learning models such as decision trees. These projects relied on relatively few job features which were collected from the job scheduler [2, 3]. We demonstrate the use of the entire user submission scripts from which we learn irregular patterns using neural networks.

3. WORKFLOW

To address the problem of parsing information from a script which contains irregular patterns, we utilize a combination of neural networks. Figure 3 shows the structure of the neural network used to predict several job features.

3.1 Data Sources

We collect several months of data from multiple machines at Lawrence Livermore National Laboratory. This data consists of the original user job submission script, strings from the scheduler (e.g., submission directory), and extracted information from the job submission (e.g., user, requested nodes). Additionally, we collect information for 46 metadata operations for each job (e.g., create and delete operations) and observed job runtimes. Jobs are sorted by submission date-time and tested in batches of 100. Training data for

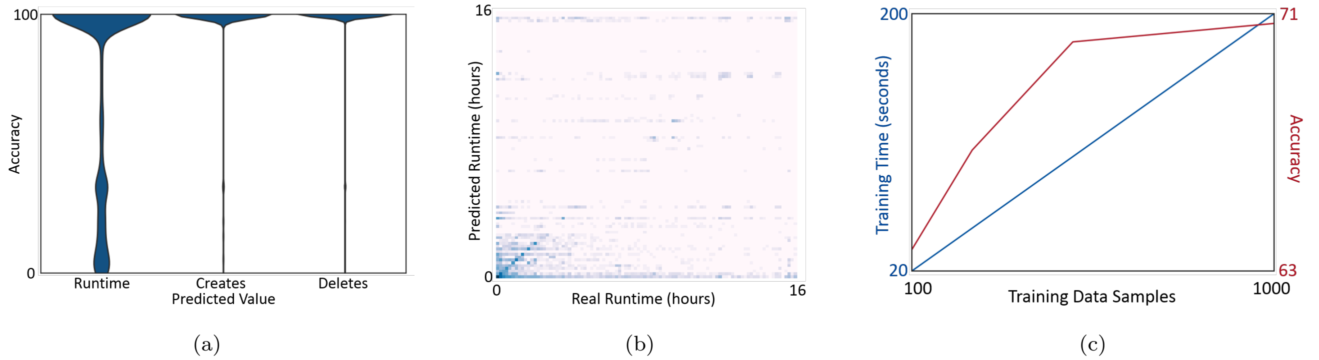

Figure 2: Neural network performance for prediction of job features. Fig. 2a shows the distribution of accuracies for the prediction of three job features. Fig. 2b shows the log normalized heatmap of runtime predictions versus actual runtimes. Fig. 2c shows the tradeoff between training time and accuracy in relation to the number of training samples.

each batch is obtained by selecting the most recent 500 jobs which have a completion date-time before the earliest test data submission date-time.

3.2 Neural Network

Obtaining information from the irregular patterns in job scripts is performed using a 2D convolution network. Long strings obtained from the job scheduler are processed with a neural network containing an embedding layer. The embedding layer maps words to a vector space in which distance determines word similarity. Other features from the submission script consist of short strings and numerical values. Strings are mapped to integer values and the data is placed in an input layer. The output from the two neural networks and additional input layer are concatenated and used as input to a deep neural network. The output of the deep neural network is probabilities for a job belonging in a certain class. The complete structure of the neural network can be seen in Figure 3.

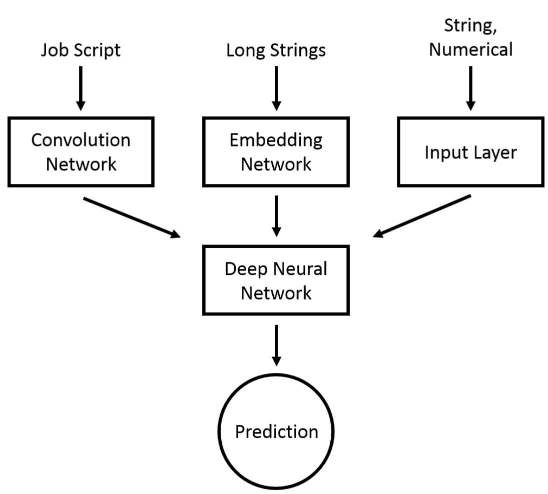

Figure 3: Neural network inputs and structure.

3.3 Evaluation

We evaluate if a prediction is correct based on a relative accuracy score, as seen in Eq. 1. This measure increases the penalty for wrong predictions of small valued features. For example, a runtime prediction with an absolute error of 10 minutes will be penalized more for a job with a runtime of 30 minutes than for a job with a runtime of 6 hours. We assess the distribution of prediction accuracies for job runtime, metadata create operations, and metadata delete operations.

$$Accuracy = 1 - \frac{|true - pred|}{max(true, pred)} \quad (1)$$

4. RESULTS

We predict job runtime, metadata create operations, and metadata delete operations for jobs using different size training data. Figure 2 shows the results of our evaluation. The distribution of prediction accuracies for the three job features is shown in Figure 2a. We achieved an average of 75%, 96%, and 96% accuracy for runtime, create operations, and delete operations, respectively. Additionally, we examine runtime predictions against actual runtimes. Several patterns of correct runtime classification and misclassification can be seen in Figure 2b. The tradeoff between predictive accuracy and training time is shown in Figure 2c.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. IM: LLNL-ABS-698189. NSF support (grants #1318445/ #1318417) is acknowledged.

5. REFERENCES

[1] S. Alam, H. El-Harake, K. Howard, N. Stringfellow, and F. Verzelloni. Parallel I/O and the Metadata Wall. In *Proceedings of the 2004 ACM/IEEE Conference on Supercomputing (SC)*, 2004.

[2] D. Nurmi, J. Brevik, and R. Wolski. QBETS: Queue Bounds Estimation from Time Series. In *Workshop on Job Scheduling Strategies for Parallel Processing*, 2007.

[3] W. Smith and P. Wong. Resource Selection Using Execution and Queue Wait Time Predictions. In *NASA Ames Research Center TR NAS-02-003*, 2002.