Revealing the Power of Neural Networks to Capture Accurate Job Resource Usage from Unparsed Job Scripts and Application Inputs

Extended Abstract

Michael R. Wyatt II
University of Delaware
Newark, Delaware
mwyatt@udel.edu

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

Adam Moody (Advisor)
Lawrence Livermore National Laboratory
Livermore, California
moody20@llnl.gov

Michela Taufer (Advisor)
University of Delaware
Newark, Delaware
taufer@udel.edu

ABSTRACT

Next generation HPC schedulers will rely heavily on accurate information about resource usage of submitted jobs. The information provided by users is often inaccurate and previous prediction models, which rely on parsed job script features, fail to accurately predict for all HPC jobs. We propose a new representation of job scripts and inclusion of application input decks for resource usage predictions with a neural network. Our contributions are a method for representing job scripts as image-like data, an automated method for predicting job resource usage from job script images and input deck features, and validation of our methods with real HPC data. We demonstrate that when job scripts for an application are very similar, our method performs better than other methods. We observe an average decrease in error of 2 node-hours compared to state of the art methods.

1 MOTIVATION

To promote high throughput and efficient sharing of high-performance computing (HPC) machines, users submit thier jobs to batch queues which, are managed by job schedulers. Current schedulers consider only user requested runtime and nodes when allocating resources to jobs. The next generation of HPC schedulers will consider higher dimensional resource spaces, which includes IO, power, and network usage. These new schedulers will require high accuracy information about job resource usage in order to optimize job execution on HPC machines [1].

The accurate resource usage information required by next generation schedulers cannot be collected from users because they do not provide all the necessary information and the information provided is inaccurate. Consequently, researchers have turned to machine learning to predict resource usage of jobs, using historical job data [3] [2]. A major focus of this work has been extracting relevant features from job scripts for the machine learning algorithms to use. Predicting resource usage is non-trivial because many jobs have highly similar job scripts which appear identical from the perspective of parsed features. However, similar job scripts may have very different resource usage profiles due to small changes in the application input deck (i.e., the set of application parameters). As a result, resource prediction for jobs with similar job scripts is inaccurate.

SC'17, November 2017, Denver, Colorado USA

ACM ISBN . . . $15.00
https://doi.org/

2 CONTRIBUTIONS

We provide a new method of interpretting and using data from HPC jobs to provide more accurate resource usage predictions. We focus on jobs which have traditionally been difficult to predict correctly, such as jobs running the *GEOS* application. We propose a new representation of job scripts which uses entire job scripts rather than only parsed features. We expand the data used for prediction to include the application input decks. In this work, we make the following contributions:

(1) We develop a technique for interpreting entire job scripts as images
(2) We develop an automated method for predicting job resource usage from job script images and input decks
(3) We validate the accuracy of our method with runtime predictions on real HPC job traces of the *GEOS* application

3 WORKFLOW

We collect execution data for HPC jobs that ran the *GEOS* application. The *GEOS* application was chosen due to the high similarity between job scripts running the application and the accompanying difficulty of predicting resource usage. Our approach transforms job script text data into image-like data to take advantage of the automatic feature detection abilities of Convolutional Neural Networks (CNN). We parse features from application input decks for use with a Neural Network (NN).

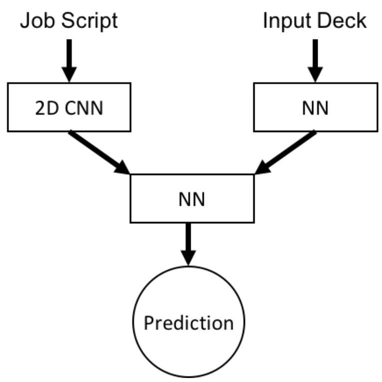

Figure 1: Our neural network structure with two inputs for the job script and input deck and one output for resource prediction.

3.1 Data and Job Script Transformation

Job scripts and input decks for 5,037 submissions of the *GEOS* application were collected. These runs occurred during 2016 on an HPC machine, "Cab," at Lawrence Livermore National Laboratory. This machine allows for submitted jobs to run up to 24 hours. *GEOS* is a simulation with high sensitivity to input deck parameters, which are stored in a separate XML file.

We process entire job scripts into image-like data to exploit the automatic feature detection abilities of CNNs. We apply a character embedding using Google's word2vec algorithm and transform each character in a job script to a vector. We adapt job script images to a standard size of 100 rows and 80 characters per row.

Our approach extracts features from the input deck by parsing tag and attribute values from the XML tree structure using depth-first search. We encode non-numerical features as vectors with feature hashing. Additionally, we fix the number of tags and attributes we consider to 2048 due to input constraints of the NN.

3.2 Predictions with Neural Network

We learn from the job scripts and input decks with a NN containing 2 input layers, seen in Figure 1. The first input is a CNN for analyzing job scripts. Our process of transforming job scripts to images allows us to take advantage of the CNN's automativ detection of features, without any text parsing. The second input is a NN for analyzing input decks. Each input has hidden layers connecting to an output layer. The outputs are concatenated and used as input to a final NN with several hidden layers. The output layer contains 1,440 nodes, where each corresponds to a minute between 0 and 24 hours.

4 RESULTS

We validate our methods by predicting runtime of *GEOS* jobs. The high similarity between *GEOS* job scripts make them ideal for our evaluation. They represents a set of jobs which prove difficult to accurately predict resource usage using only traditional parsed features. We train and test our method with 10-fold crossvalidation. For comparison, we also train and test a random forest regressor using features similar to Smith et al. [3]. We calculate the minimum distance between a job script and all other job scripts in our dataset.

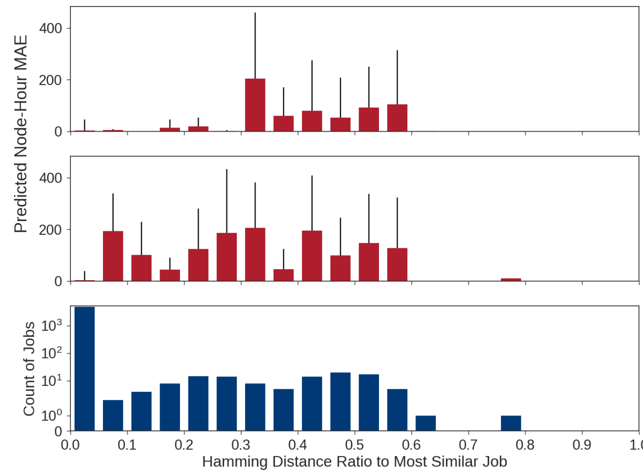

Figure 2: Runtime prediction error for GEOS jobs using our neural network method (TOP), runtime prediction error for GEOS jobs using a random forest regressor and traditional parsed features (MIDDLE) and distribution of jobs by minimum distance to the closest neighbor (BOTTOM).

We use the hamming distance ratio, shown in Equation 1, to measure distances between job script texts.

$$hammingRatio(i, j) = \frac{hamming(i, j)}{max(length(i), length(j))} \quad (1)$$

Figure 2 shows the predicted node-hour mean absolute error of our method and the random forest regressor as well as the distribution of jobs by hamming distance ratio to the most similar neighbor. Our method predicts runtimes for jobs having similar job scripts with much less error than the random forest regressor and parsed features. This demonstrates that our methods are able to accurately predict runtime of jobs which cannot be accurately predicted with the traditional parsed features. We reduce the average rate of error from 6.6 node-hours with the random forest to 4.6 node-hours with our method. This is a total reduction in runtime error of 420 node-days across all GEOS jobs in the dataset. In future work, we will expand the number of applications and the types of resource predictions we use to validate our work. Additionally, we will explore optimization of our NN to obtain higher accuracy predictions.

ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. DoE by LLNL under Contract DE-AC52-07NA27344. IM: LLNL-ABS-736466. This material is based upon work supported by the NSF under the grant no. CCF 1318445.

REFERENCES

[1] Stephen Herbein et al. 2016. Scalable I/O-Aware Job Scheduling for Burst Buffer Enabled HPC Clusters. In *Proceedings of the 25th ACM International Symposium on High-Performance Parallel and Distributed Computing*. ACM, 69–80.
[2] Ryan McKenna et al. 2016. Machine Learning Predictions of Runtime and IO Traffic on High-End Clusters. In *Cluster Computing (CLUSTER), 2016 IEEE International Conference on*. IEEE, 255–258.
[3] Warren Smith et al. 1999. Using run-time predictions to estimate queue wait times and improve scheduler performance. In *JSSPP*. Springer, 202–219.