Supervised Learning for Parallel Application Performance Prediction

Andrew R. Titus
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139 USA
Email: atitus@mit.edu

Abhinav Bhatele
Center for Applied Scientific Computing
Lawrence Livermore National Laboratory
Livermore, California 94551 USA
Email: bhatele@llnl.gov

***Abstract*—We evaluate supervised machine learning methods as tools for prediction of communication time of large parallel applications. Through these methods, we correlate communication time for different task mappings to the corresponding network hardware counters accessible on the IBM Blue Gene/Q system. We use the results from these machine learning regression algorithms to provide insights into the relative importance of different hardware counters and metrics for predicting application performance. These results are explored in the context of two production applications, MILC and pF3D.**

I. Introduction

We generate a spread of task mappings, using both random ordering of MPI ranks and Rubik, a mapping tool developed at LLNL [1]. We then run pF3D and MILC, two multiphysics codes, on 1024 and 4096 nodes of Vulcan, an IBM Blue Gene/Q installation at LLNL, using these mappings. Using the data from these runs, we evaluate six different supervised machine learning algorithms provided by scikit-learn [2]:

- Bayesian Ridge
- Gradient Boosted Regression Trees (GBRT)
- Decision Tree
- Forests of Randomized Decision Trees (Random Forests)
- Ridge
- Support Vector Machines (SVM)

We compare single metrics derived from hardware network counter data, as well as "hybrid" combinations of these, to generate predicted communication times. Using the results from various algorithms, we can analyze their effectiveness graphically and statistically.

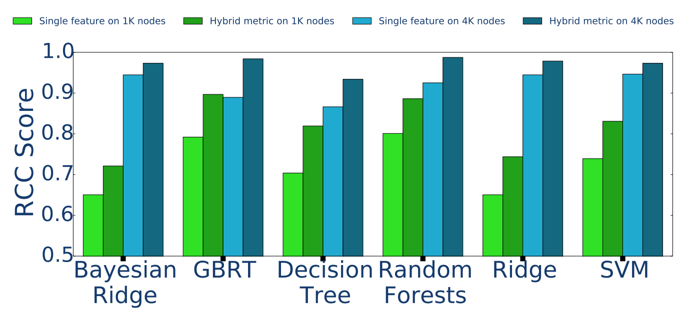
II. Results

Success of the predicted times is determined by the **Rank Correlation Coefficient (RCC)**, a metric that relates the number of correctly predicted pair rankings by performance to the actual performance rankings. If $\{x_1, x_2, \cdots, x_n\}$ are the actual performance rankings and $\{y_1, y_2, \cdots, y_n\}$ are the predicted performance rankings, then RCC, as defined by Bhatele *et al.* [3] is:

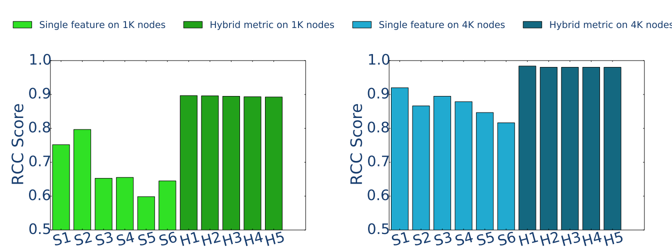
$$concord_{ij} = \begin{cases} 1, & \text{if } x_i >= x_j \text{ \& } y_i >= y_j \\ 1, & \text{if } x_i < x_j \text{ \& } y_i < y_j \\ 0, & \text{otherwise} \end{cases}$$

$$RCC = \Big(\sum_{0<=i<n} \sum_{0<=j<i} concord_{ij} \Big) / (\frac{n(n-1)}{2})$$

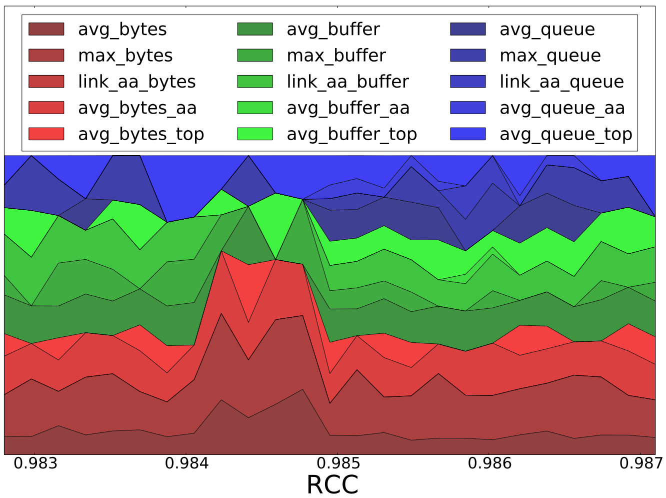
The first set of plots in the poster (MILC example below) visualizes the relative strength of each machine learning algorithm. They display the best RCC scores among all single metrics, as well as the best scores achieved with hybrid metrics, for runs of MILC and pF3D on 1024 and 4096 nodes. Hybrid metrics consistently outperform single metrics for all algorithms and among these, GBRT and Random Forests consistently perform the best. It is also interesting that RCC values tend to be higher for runs on 4096 nodes, as opposed to 1024 nodes.



The next set of plots (MILC example below), shows in more detail that hybrid metrics outperform single metrics. This was done by juxtaposing the best six single metric prediction models with the best five hybrid prediction models for runs of MILC and pF3D on 1024 and 4096 nodes.



The last set of plots (pF3D on 1024 nodes example below) provides easy visualization of the relative importance of single metrics in the generation of the best hybrid metrics. They display the top 25 hybrid metrics developed using GBRT, in order of increasing RCC, in a stacked area plot. The plots show that a combination of single metrics from various sources (bytes on links, buffers and queues) come together to create the well-performing hybrid metrics.



III. Conclusion and Future Work

We see high correlations between network counters data and the communication time for production applications. Hybrid metrics tend to consistently have superior prediction ability as compared to single metrics, particularly with those generated by *Ensemble Methods*. Ensemble methods used in this poster included Gradient Boosted Regression Trees (GBRT) and Forests of Randomized Trees, which combine weaker individual decision trees together in different ways to generate a single strong method. Unfortunately, what actually contributes most to network contention is still uncertain, as the relative importance of single metrics within the best hybrid GBRT models varies widely across applications and node counts.

Therefore, the goals of future work are to expand the search for the explanation of this network effect. This includes looking at more communication features as candidates for single metrics, more machine learning algorithms, and more applications over different node counts. Using stacked area plots, as well as other visualization methods, will allow researchers to further study network contention and move closer to an explanation for it.

Acknowledgment

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This work was funded by the Laboratory Directed Research and Development Program at LLNL under project tracking code 13-ERD-055 (LLNL-ABS-662732).

References

[1] A. Bhatele, T. Gamblin, S. H. Langer, P.-T. Bremer, E. W. Draeger, B. Hamann, K. E. Isaacs, A. G. Landge, J. A. Levine, V. Pascucci, M. Schulz, and C. H. Still, "Mapping applications with collectives over sub-communicators on torus networks," in *Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '12. IEEE Computer Society, Nov. 2012, LLNL-CONF-556491.

[2] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[3] N. Jain, A. Bhatele, M. P. Robson, T. Gamblin, and L. V. Kale, "Predicting application performance using supervised learning on communication features," in *ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '13. IEEE Computer Society, Nov. 2013, LLNL-CONF-635857.