PREDICTING APPLICATION PERFORMANCE USING SUPERVISED LEARNING ON COMMUNICATION FEATURES

Nikhil Jain*, Abhinav Bhave†, Michael P. Robson*, Todd Gamblin†, Laxmikant V. Kale*

*University of Illinois at Urbana-Champaign
†Lawrence Livermore National Laboratory

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 IM Release:
SUPERCOMPUTERS

48 GB/s, 1-2 µs

40 GB/s, 1-3 µs

150 GB/s, 0.8 µs

420 GB/s, 1-2 µs

Higher Bandwidth
Lower Latency
Fewer hops
WHY STUDY NETWORK PERFORMANCE?

- Peak bandwidth and latency are never obtained in presence of congestion
- High raw bandwidth does not guarantee proportionate observed performance
  - Topology, job interference, I/O
- Find the next generation topology
- Savings are proportionate to core-count
Mapping via logical operations in Rubik

What about others mappings?

How far are we from the best performance?

Which is the best performing mapping?

PERFORMANCE PREDICTION METHODS

- Theoretically: NP hard
- Simulations: too slow
  - Few days to simulate one use case*
- Real runs: very expensive
  - Application/allocation specific information

<table>
<thead>
<tr>
<th></th>
<th>2012</th>
<th>2013</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intrepid</td>
<td>4.16M</td>
<td>0.73M</td>
</tr>
<tr>
<td>Mira</td>
<td>0.17M</td>
<td>7.67M</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>4.33M</strong></td>
<td><strong>8.40M</strong></td>
</tr>
</tbody>
</table>


13 million core hours!
HEURISTICS
PRIOR FEATURES

2D-Halo: predicting performance using a linear regression model for prior features
SUPERVISED LEARNING: OVERVIEW

- Collect/generate data and summarize
- Build models: train performance prediction based on independent features
- Predict and correlate

```
2D Halo Predicted  2D Halo Observed
```

![Graph showing execution times](image)
MESSAGE LIFE CYCLE
ON BLUE GENE/Q
A PMPI based BG/Q-Counter collection module

- Packets sent on links in specific directions: A, B, C, D, E
  - deterministic, dynamic
- Packets received on a link
- Packets in buffers
INPUT FROM SIMULATION

- Simulate the injection mechanism
- Selection of memory injection FIFO
- Mapping of memory FIFO to network injection FIFO
- Simulate routing to obtain hops/dilation
## INPUT DATA

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Source</th>
<th>Derived from</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bytes on links</td>
<td>Counters</td>
<td>Sent chunks</td>
</tr>
<tr>
<td>Buffer length</td>
<td>Counters</td>
<td>#Packets in buffers</td>
</tr>
<tr>
<td>Delay per link</td>
<td>Counters</td>
<td>#Packets in buffers / #received packets</td>
</tr>
<tr>
<td>Dilation</td>
<td>Analytical</td>
<td>Shortest path routing</td>
</tr>
<tr>
<td>FIFO length</td>
<td>Analytical</td>
<td>Based on PAMI</td>
</tr>
</tbody>
</table>
BUILDING MODEL

- Derive features from the raw data on entities, e.g. average bytes on links
- Create a database of derived features and performance; we have used 100 mappings
  - 33% mappings generated randomly
  - 33% using Rubik
  - Rest are based on better performing mappings
- Select two-thirds entries as training set:
  - Derived features are independent variables
  - Performance is a dependent variable
BUILDING MODEL

- The training set is used to create a model for prediction
- Remaining entries from the database are used as the test set - derived features as input
- Prediction is compared with observed values
- Experimented with a large number of algorithms - linear, bayesian, SVM, near-neighbors, etc.

http://scikit-learn.org
LEARNING ALGORITHM

Decision trees

Randomized forest of trees

Rank Correlation Coefficient (RCC): fraction of the number of pairs of task mappings whose ranks are in the same partial order in predicted and observed performance list

\[
\text{RCC} = \left( \sum_{0 \leq i < n} \sum_{0 \leq j < i} \text{concord}_{ij} \right) / (\frac{n(n-1)}{2})
\]

Absolute Correlation

\[
R^2(y, \hat{y}) = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}
\]

Higher is better!
RESULTS: SETUP

- Three communication kernels
  - Five-point 2D stencil
  - 14-point 3D stencil
  - All-to-all over sub-communicators
- Four message sizes to span MPI and routing protocols
PRIOR FEATURES

- Entities
  - Bytes on a link
  - Dilation
- Derivation Methods
  - Maximum
  - Average
  - Sum

Figure 1: Performance variation with prior metrics. A large variation in performance is observed for the same value of the metric in different plots. For example, in the maximum bytes plot (right), for the same x-value, there are mappings with performance varying from 0 to 60. When used for prediction, in the rest of the paper, we use average metrics (from hop-bytes) and all three are practically equivalent. Sources of contention motivate the need for more precise metrics. Derived graphs and metrics are described in Section 4.
RESULTS
PRIOR FEATURES

max bytes is good, but incorrect in 10% cases
NEW FEATURES

- Entities
  - Buffer length (on intermediate nodes)
  - FIFO length (packets in injection FIFO)
  - Delay per link (packets in buffer / packets received)

- Derivation methods
  - Average Outliers (AO)
  - Top Outliers (TO)
RESULTS
NEW FEATURES

Rank correlation coefficient

Absolute performance correlation

R²

16K 4M 512 16K 4M 8 512 16K 4M
2D Halo 3D Halo Sub A2A

avg buffer
sum dilation AO
avg bytes AO
avg bytes TO

0.6 0.7 0.8 0.9 1.0
HYBRID FEATURES

- Combine multiple metrics to complement each other

- Some combinations
  - H1: avg bytes + max bytes + max FIFO
  - H3: avg bytes + max bytes + avg buffer + max FIFO
  - H4: avg bytes + max bytes + avg buffer TO
  - H5: avg bytes TO + avg buffer TO + avg delay AO + sum hops AO + max FIFO
RESULTS

HYBRID FEATURES

hybrid metrics provide high accuracy
SUMMARY ON 64K CORES

Rank correlation coefficient

Absolute performance correlation

max bytes
avg bytes TO
H3
H5

Nikhil Jain @ SC ‘13
RESULTS: TREND

Figure 7: Prediction success based on hybrid features from Table 3. We obtain RCC and $R^2$ values exceeding 0.99 for 3D Halo and Sub A2A. Prediction success improves significantly for 2D Halo also.

$0.93$ to $0.975$ and $0.955$ for the $16$ KB and $4$ MB message sizes respectively. For the more communication intensive benchmarks, we obtained $R^2$ values as high as $0.99$ in general. Hence, the use of hybrid features not only predicts the correct pairwise ordering of mapping pairs but also does so with high accuracy in predicting their absolute performance.

Figure 8 presents the scatter-plot of predicted performance for the three benchmarks for the $4$ MB message size. On the x-axis are the task mappings sorted by observed performance, while the y-axis is the predicted performance. The feature set $H3$: avg bytes, max bytes, avg buffer, max FIFO was used for these predictions. It is evident from the figure that an almost perfect ordering is achieved for all three benchmarks.

Figure 9 shows the prediction success for the three benchmarks on $65,536$ cores of BG/Q. From all the previously presented features (prior, new and hybrid), we selected the ones with the highest RCC scores for $16,384$ cores, and present only those in this figure. We obtain significant improvements in the prediction scores using hybrid features for prediction in comparison to single features such as max bytes TO $^2$. For Sub A2A, RCC improved by $14\%$ from $0.86$ to $0.98$, with a RCC value of $1.00$ for both $512$ bytes and $4$ MB message sizes. For 2D Halo and 3D Halo, an improvement of up to $8\%$ was observed in the prediction success. Similar trends were observed for $R^2$ values.

6. COMBINING ALL TRAINING SETS

In the previous section, we presented high correlation for predicting performance of the three benchmarks. For the prediction of individual benchmarks, the training and testing sets were generated from the $84$ different mappings of the same benchmark for a particular message size on a fixed core count. In this section, we relax these requirements, and explore the space where the training and testing sets are a mix of different benchmarks, message sizes and core counts.

6.1 Combining samples from different kernels

We first explore the use of training and testing sets that are a combination of all three benchmarks and both $16$ KB and $4$ MB message sizes.
RESULTS

ABSOLUTE PERFORMANCE

 Execution Time (s)

 Mappings sorted by actual execution times

FFT Predicted  3D Halo Predicted  2D Halo Predicted
FFT Observed  3D Halo Observed  2D Halo Observed
COMBINING BENCHMARKS

Rank correlation coefficient

Pairwise ordering misprediction

Number of mispredictions
PREDICTING FOR 64K CORES USING 16K CORES

Rank correlation coefficient

Absolute Performance Correlation

<table>
<thead>
<tr>
<th>RCC</th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
<th>H4</th>
<th>H5</th>
<th>H6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.95</td>
<td>0.85</td>
<td>0.90</td>
<td>0.75</td>
<td>0.65</td>
<td>0.90</td>
<td>0.85</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>R²</th>
<th>avg bytes</th>
<th>max bytes</th>
<th>avg dilation A0</th>
<th>avg buffer</th>
<th>H1</th>
<th>H2</th>
<th>H3</th>
<th>H4</th>
<th>H5</th>
<th>H6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.85</td>
<td>0.80</td>
<td>0.75</td>
<td>0.60</td>
<td>0.50</td>
<td>0.80</td>
<td>0.75</td>
<td>0.60</td>
<td>0.50</td>
<td>0.80</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Nikhil Jain @ SC ’13
RESULTS: PF3D

Rank correlation coefficient

RCC

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg bytes</td>
<td>0.6</td>
</tr>
<tr>
<td>max bytes</td>
<td>0.7</td>
</tr>
<tr>
<td>sum dilation AO</td>
<td>0.8</td>
</tr>
<tr>
<td>avg buffer TO</td>
<td>0.9</td>
</tr>
<tr>
<td>H1</td>
<td>1.0</td>
</tr>
<tr>
<td>H3</td>
<td>0.9</td>
</tr>
<tr>
<td>H4</td>
<td>0.8</td>
</tr>
<tr>
<td>H5</td>
<td>0.7</td>
</tr>
<tr>
<td>H6</td>
<td>0.6</td>
</tr>
</tbody>
</table>

Absolute performance correlation

$R^2$

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>avg bytes</td>
<td>0.6</td>
</tr>
<tr>
<td>max bytes</td>
<td>0.7</td>
</tr>
<tr>
<td>sum dilation AO</td>
<td>0.8</td>
</tr>
<tr>
<td>avg buffer TO</td>
<td>0.9</td>
</tr>
<tr>
<td>H1</td>
<td>1.0</td>
</tr>
<tr>
<td>H3</td>
<td>0.9</td>
</tr>
<tr>
<td>H4</td>
<td>0.8</td>
</tr>
<tr>
<td>H5</td>
<td>0.7</td>
</tr>
<tr>
<td>H6</td>
<td>0.6</td>
</tr>
</tbody>
</table>
RESULTS: PF3D

Blue Gene/Q (16,384 cores)

Mappings sorted by actual execution times

pF3D Observed
pF3D Predicted
Communication is not just about peak latency / bandwidth

Simultaneous analysis of various aspects of network is important

Complex models are required for accurate prediction

There are patterns waiting to be identified!
FUTURE WORK

- More applications!
- More metrics
- Weighted analysis
- Offline prediction of entities

Questions?