Fast Prediction of Network Performance: k-packet Simulation

Nikhil Jain and Laxmikant V. Kale

University of Illinois at Urbana-Champaign, Urbana, Illinois 61801 USA
E-mail: {nikhil, kale}@illinois.edu

I. Introduction

Prediction of application communication performance on supercomputer networks, without doing actual runs, is useful for a variety of what-if analyses: how does the performance change with different task mappings, does the performance improve when using a different algorithm to implement collectives, designing and studying the performance of future networks, and for analyzing different routing schemes, etc. A significant amount of research is conducted to study these aspects because the scalability of many applications is adversely affected by communication overheads when running on large systems.

Flit-level and packet-level simulations have been shown to be useful for prediction, but they are inefficient and such simulations are very slow even for moderate-sized networks. It takes hours to days of execution time to simulate a few seconds of an application run on large systems using these approaches. We propose a new simulation methodology that relies on decreasing the granularity of simulation to k-packets and use of heuristics for predicting the stable state of the network. Preliminary results show that the proposed approach accurately models network behavior for simple communication operations. It is also orders of magnitude faster than the previous methods for simple uniform communication operations – up to two minutes per prediction for 16,384 cores of Blue Gene/Q using the given benchmarks.

II. Proposed Approach

The proposed approach is motivated by following insights:

- Reasonably accurate predictions can be made by analyzing the network at the granularity of message-level events (k-packets where k is variable) instead of flit/packet-level events.
- To predict network behavior over a period of time, complex simultaneous flows of packets from different messages that share a resource can be approximated by message flows that are disjoint in time.
- For many use cases, prediction of trends is sufficient; knowledge of exact execution time is not required.

The flowchart presented in the poster shows our implementation of a simulator based on the above insights. A typical simulation begins with aggregation of input data such as the communication pattern to be simulated, bandwidth of the links, a task mapping, etc. In each iteration, the simulator checks for a list of messages to be sent. Based on the source-destination pair and the routing algorithm (currently restricted to deterministic), possible paths for all messages are computed. Then, using a greedy heuristic, a list of messages to be sent in this iteration is computed. This computation takes into account all the resource constraints, such as limited injection bandwidth, link bandwidths etc., that may prevent a message from being sent.

The next step is to compute the granularity of the current iteration. Granularity is the amount of time for which the selected messages will occupy the resources in this iteration, which in turn is proportional to the number of packets of a message that will be sent out. Since, our iterations begin and end at message-level events, we assign the granularity as the minimum of the individual time taken by the selected outgoing message(s) if all of its packets are sent. This step is followed by the bookkeeping related to updating the packets left for each message, and deletion of messages whose send has been completed. From here, the simulator jumps to the beginning of the next iteration. The simulation is stopped when all the communication is complete.

III. Results

We have used two communication benchmarks to test the accuracy of the proposed approach:

- Near-neighbor: 14-point 3D stencil
- All-to-all: simultaneous all-to-alls over sub-communicators of size 64 each

To validate our simulator, we compare actual observed performance on 16,384 cores of Blue Gene/Q with predicted performance for 84 different task mappings of the two benchmarks. The proposed approach is highly accurate for predicting the trend in network behavior. For All-to-all, the predicted values are very close to the observed values as shown in the poster. For Near-neighbor, the predicted values follow the same trend as the observed values, but are overly optimistic for bad performing mappings.

In terms of the execution time required for simulation, the proposed approach is orders of magnitude faster than the known methods. The speed-up is $153\times$ for All-to-all, and $223\times$ for Near-neighbor in comparison to BigSim executed only for evaluating network performance [1]. The huge difference in performance can be attributed to following reasons: 1) Owing to uniformity in the given benchmarks (all messages are of same size), the granularity (k) selected by the proposed approach is very large (full message). 2) In its current form,

the proposed approach is targeted to find the time spent in the given communication operation as a whole. Unlike BigSim, one will not be able to obtain the network state at intermediate points. This shortcoming will be addressed in a future publication.

In terms of absolute time, it takes only around 15 seconds to predict the performance for Near-neighbor on 16,384 cores. The scalability plot in the poster (for Near-Neighbor) shows that as the application is weak scaled, the simulation time increases linearly. Given the positive preliminary results, we plan to augment the proposed simulators with important functionalities that will lead to a full-fledged simulator. Enabling the simulator to be driven by a trace and being able to examine the state of the network at intermediate points are examples of functionalities that we plan to add to the simulator.

IV. Use Cases

We envision the following use cases (among many others) for the proposed approach:

- A good mapping can be found by exploring a large set of generated task mappings, possibly in parallel using the allocated system at the beginning of a job execution; use of this mapping for the real execution should be beneficial and offset the time spent for initial exploration.
- What-if studies: As shown in the poster, the proposed approach enables analysis of various scenarios, such as different link bandwidths, new routing schemes etc, in a fast manner. For example, we found that if the link bandwidth on Blue Gene/Q was 20 GB/s (instead of 2 GB/s), the proposed approach predicts that mapping may not affect the observed network performance for Near-neighbor.

References

[1] G. Zheng, G. Kakulapati, and L. V. Kalé, "Bigsim: A parallel simulator for performance prediction of extremely large parallel machines," in *18th International Parallel and Distributed Processing Symposium (IPDPS)*, Santa Fe, New Mexico, April 2004, p. 78.