Analysis of Energy and Power Consumption by Remote Data Transfers in Distributed Programming Models

Siddhartha Jana, Barbara Chapman
University of Houston, Texas, USA
Email: sidjana,chapman@cs.uh.edu

***Abstract*—Recent studies of the challenges facing the Extremescale era express a need for understanding the impact on energy profile of applications due to inter-process communication on large-scale systems. Programming models like MPI provide the user with explicit interfaces to initiate data-transfers among distributed processes.**

This poster presents empirical evidence that: controllable factors like the size of the data-payload to be transferred and the number of explicit calls used to service such transfers have a direct impact on the power signatures of communication kernels. Moreover, the choice of the transport layer (along with the associated interconnect) and the design of the inter-process communication protocol are responsible for these signatures. Results discussed in this poster motivate the incorporation of energy-based metrics for fine tuning communication libraries that target Extremescale machines.

I. Introduction

One of the primary challenges on the pathway to Exascale Computing is the 20MW power consumption envelope established by the U.S. Department of Energy's Exascale Initiative Steering Committee [1]. The direct outcome of this has been a rising concern about the energy and power consumption of large-scale applications that rely on various communication libraries for efficient data movement in distributed systems. As part of this work, we establish the notion that the factors responsible for the performance of such libraries, also govern the power-profiles of such applications.

Through this work, we have learned that managing small-sized data transfers on RDMA-capable networks are more energy efficient than handling large bulk transfers. The poster presents empirical evidence highlighting the contribution of design factors within the software stack to the power consumption by the underlying system. Our takeaway from this study is that the protocols used to implement such interfaces, play a significant role in impacting its power-cost. In addition, since the design of communication libraries are tuned to specific interconnect solutions, the choice of the transport layer adopted for servicing data transfers plays an equally significant role.

II. Factors Explored

Fig. 1a lists many such factors throughout the hardware and software stack.

Two-sided data-transfer in distributed-memory models like MPI, sockets, etc. require the active participation of both the sender and the receiver of the data. The impact on the achievable latency and bandwidth of such transfers depend on the design of the transport layer (and the associated interconnect) and the data transfer protocol. As part of this work, we learned that the impact of these factors on the energy metrics is very important.

A. Choice of transport layer and the associated interconnect

If the target platform relies on an OS-based TCP protocol for servicing data transfers, CPU cores undergo multiple switches between user and supervisor operating modes. In addition, relying on Ethernet-based fabric has the potential of degrading the achievable efficiency both in performance and energy consumption (as discussed later). To avoid this, a communication library may exploit kernel-bypass mechanisms and RDMA-based capabilities of the OFED stack on top of modern interconnects like InfiniBand, etc.

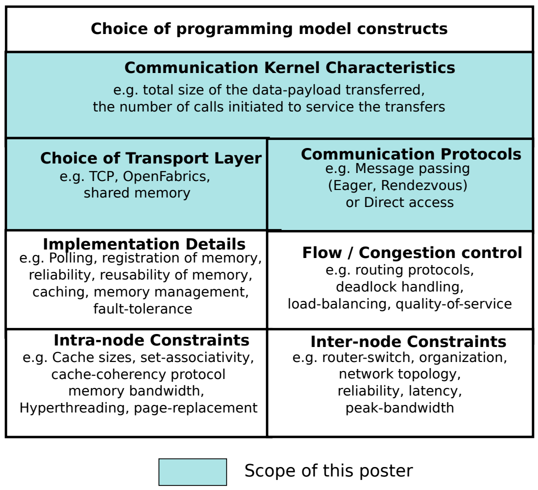
B. Design of data-transfer protocols

Data transfers within message-passing libraries are based on two well-established protocols - the eager and rendezvous protocols. We study the impact on the energy consumption based on the type of protocol a participating process employs.

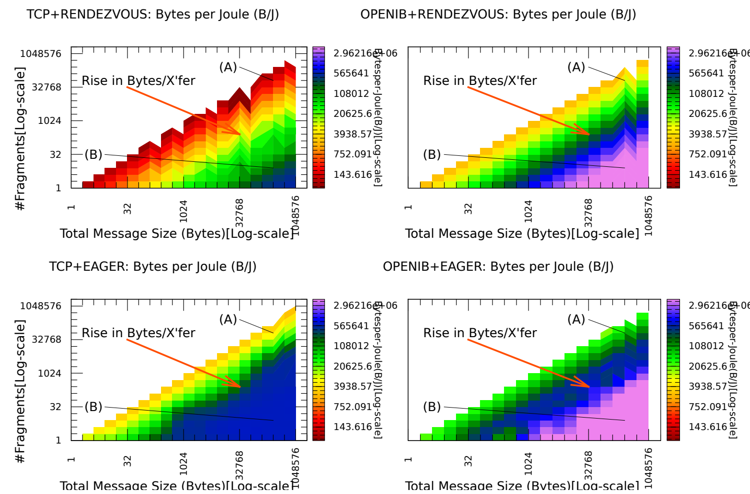
III. Empirical results

To study the net impact of the choice of the communication protocols and the transport layer, we evaluated the power efficiency using a metric suitable for evaluating communication-intensive kernels - Bytes/Joule i.e. the number of bytes that can be transferred across the network for each joule of energy consumed by the sender and receiver.

Figure 1b summarizes the results of our empirical study based on two different transport layers - TCP and OpenIB paired with two different data transfer protocol - Eager and Rendezvous. For each of the four plots, the X-axis represents the size of the data-payload that is transferred to a remote node while executing a communication-intensive kernel. The Y-axis represents the number of explicit MPI calls that were used to make this transfer. Each colored point for a tuple $\{X,Y\}$ represents the value of the metric - Bytes/Joule and



(a) Factors impacting the energy and power consumption across the hardware and software stack



(b) A summary of the total bytes transferred per Joule of energy consumed by the sender and the receiver while participating in remote data transfers.

maps to the colored scale on the right of every plot. The primary observations are:

- The net bandwidth achievable using an interconnect directly impacts the maximum value of energy-efficiency. Thus the peak bytes transmitted per joule is an order of a magnitude higher when using the OpenIB over InfiniBand as compared to TCP over Ethernet. Moreover, irrespective of the type of transport adopted, energy-efficient communication can be achieved using an eager-based protocol.
- *Impact of number of bytes packed per transfer:*
  - In the figure, the arrow points towards the direction of the increase in the number of bytes transferred per call. For TCP+Rendezvous configuration (Fig. 1b(I)), we see that the peak energy efficiency during a data transfer (about 0.565MB/Joule) may be attained only when the total message size per transfer is higher than 128KB. With the TCP+Eager protocol however, this peak is attained for message sizes beyond 1KB in size.
  - The highest power efficiency among all the configurations is achievable while using an eager-based protocol over the OpenIB stack - A maximum of 3MB of data is transferred for every joule of energy consumed.

IV. Conclusion

Data movement across large-scale systems has the potential of impacting not only the performance of distributed programming models, but also the power-signatures. This poster establishes the notion that the choice of the transport layer and the design of communication protocols play a significant role in terms of the energy and power consumption. The empirical results discussed highlight the behavior of this impact on the CPU cores and the memory. Empirical results indicate that the power consumption by CPU cores and the memory bandwidth is not only impacted by the latency of the remote transfers, but also the memory bandwidth between the CPU cores and the memory.

While using traditional TCP over Ethernet, energy savings can be obtained by choosing an eager-based protocol over a rendezvous-based one. While using an eager protocol, an efficiency of up to 600bytes/joule may be obtained. Despite these savings, it must be noted that mapping an eager protocol over a non-RDMA based fabric leads to high power consumption by the memory. While using an RDMA-capable network like InfiniBand, the use of eager-based protocol lends itself naturally to the semantics of the transport layer (OpenFabrics OFED, in our case).

Irrespective of the type of transport and protocol, higher efficiency (bytes transferred per joule) can be achieved by aggregating user buffers into contiguous larger fragments before servicing the transfer. In addition, the net bandwidth achievable during a transfer impacts this efficiency. We hope that results of energy-efficiency as well as a detailed study of the impact on the various sub-components of the system would motivate the design of "power-aware" middleware for use with HPC applications.

As the next step, we plan to extend this study to evaluate the impact on large-scale systems. It is equally essential to study the contribution of communication kernels to the energy profiles of large scale real world HPC applications.

References

[1] J. Shalf, S. Dosanjh, and J. Morrison, "Exascale Computing Technology Challenges," pp. 1–25, 2011.