Layout-Aware Scientific Computing A Case Study using MILC

Jun He
Illinois Institute of Technology
Chicago, Illinois 60616
jhe24@iit.edu

Jim Kowalkowski
Fermi National Accelerator Laboratory
Batavia, Illinois 60510
jbk@fnal.gov

Marc Paterno
Fermi National Accelerator Laboratory
Batavia, Illinois 60510
paterno@fnal.gov

Don Holmgren
Fermi National Accelerator Laboratory
Batavia, Illinois 60510
djholm@fnal.gov

James Simone
Fermi National Accelerator Laboratory
Batavia, Illinois 60510
simone@fnal.gov

Xian-He Sun
Illinois Institute of Technology
Chicago, Illinois 60616
sun@iit.edu

ABSTRACT

Nowadays, high performance computers have more cores and nodes than ever before. Computation is spread out among them, leading to more communication. For this reason, communication can easily become the bottleneck of a system and limit its scalability. The layout of an application on a computer is the key factor to preserve communication locality and reduce its cost. In this paper, we propose a simple model to optimize the layout for scientific applications by minimizing inter-node communication cost. The model takes into account the latency and bandwidth of the network and associates them with the dominant layout variables of the application. We take MILC as an example and analyze its communication patterns. According to our experimental results, the model developed for MILC achieved a satisfactory accuracy for predicting the performance, leading to up to 31% performance improvement.

Categories and Subject Descriptors

C.4 [**PERFORMANCE OF SYSTEMS**]: Modeling techniques; J.2 [**PHYSICAL SCIENCES AND ENGINEERING**]: Physics

General Terms

Performance

Keywords

Performance model, MILC, communication

ScalA'11, November 14, 2011, Seattle, Washington, USA.

1. INTRODUCTION

Scientists in many domains of research have relied on the growth of computing power to improve and extend their research. Hundreds of thousands of cores are built into modern supercomputers, yielding huge amounts of computing power [5] [3]. However, spreading the computing power across multiple cores and nodes leads to more inter-core and inter-node communication. Inter-node communication is much slower than intra-node communication. As the size of a computation scales up, the inter-node communication may gradually dominate the application running time, even when advanced networking technology like Infiniband [1] is used. Slow communication leads to longer overall running time of applications. The costs of running application on supercomputers are very high. A few percent improvement in application speed can enhance the efficiency of the supercomputers and save significant amounts of money. Therefore, it is crucial to minimize inter-node communications in order to improve the overall performance.

MILC (MIMD Lattice Computation) [2] has been developed to simulate four dimensional SU(3) lattice gauge theory on MIMD parallel machines. It uses the conjugate gradient method, which involves both computation and communication. MILC can easily eat up computing cycles and it is often used as benchmark for supercomputers [6] [7]. MILC is designed to overlap communication with computation, while the degree of overlapping depends on many factors. Long communication time or low degree of overlapping make applications communication bounded. To achieve optimal performance on any given problem, application parameters must be chosen to communication cost. [7]. Communication costs can be even larger when the same amount of computing is spread to more cores on many nodes, leading to more inter-node communication. Therefore it is crucial to reduce communication cost in MILC.

In MILC, a 4-d lattice is divided to many 4-d subvolumes of equal size. Each subvolume communicates only with its neighbors, to synchronize their data periodically after each phase of computing. The *layout* of a lattice is the way in which it is divided into subvolumes, and how the subvolumes are placed on computer nodes and cores. The layout has great impact on the communication cost and therefore the overall performance, since it influences the communi-

cation locality. Because MILC is not aware of the topology of computers, it cannot find the best layout by itself. Physicists who use MILC must choose what layout to use, typically after trying several. This is cumbersome and does not guarantee an optimal solution. To solve this problem, we have explored different layout strategies, and find the optimal one by minimizing the cost of inter-node communication. We propose a model to determine the best layout for a given lattice and a given set of computational resources.

Our contributions are as follows:

- We evaluated many parameters of MILC layout, and identified the two key factors which determine communication cost: *inter-node subvolume paths* and *sites on the inter-node surfaces*.
- We developed a model to estimate the communication cost of MILC, which enabled us to find the optimal layout. Our experimental results show that the performance improvement can be up to 31%.

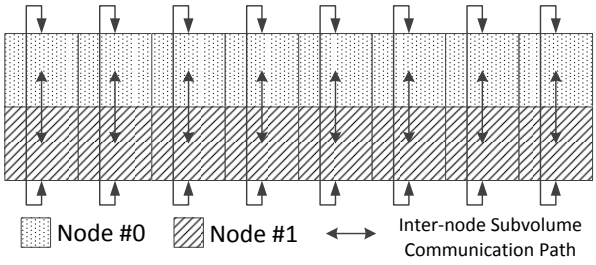
2. ANALYSIS OF COMMUNICATION AND LAYOUT IN MILC

In MILC there are several basic elements, including the *lattice*, *subvolume* and *site*. A lattice is comprised of many sites, which form a hyper-rectangular region in four space-time dimensions. An $A \times B \times C \times D$ lattice has A sites in the $\hat{x}$ direction, B sites in the $\hat{y}$ direction, C sites in the $\hat{z}$ direction and D sites in the $\hat{t}$ (time) direction The lattice is divided to one or more subvolumes, each of identical shape. MILC performs calculations on a site-by-site basis, and requires communication between neighboring sites. All sites for a given subvolume are restricted within that subvolume and a single core can only hold one subvolume. The major communication in MILC is between subvolumes. Each subvolume communicates only with its neighbors.

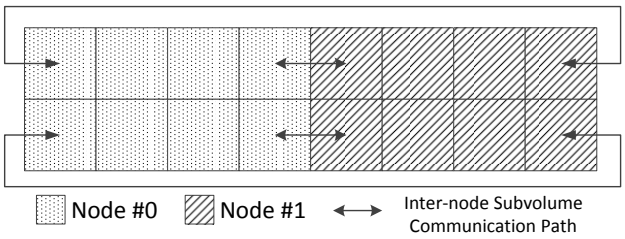
Inter-node communication is much more costly than intra-node communication. Due to the communication characteristics of MILC, in order to achieve good performance, it is very important to cut the lattice in appropriate subvolumes and to assign those subvolumes to appropriate nodes. If the bandwidth of the inter-node communication is B, the size of the message is S and the latency is L, then the inter-node communication time t, can be calculated according to Formula 1:

$$t = L + S/B \tag{1}$$

To demonstrate how layout affects communication times, we present the following example. Suppose we have 8×2 subvolumes and 2 nodes. Each node has 8 cores. As illustrated in Figure 1, Layout 1 (Figure 1(a)) has more inter-node communication paths than Layout 2 (Figure 1(b)), due to different layouts of the subvolumes. The greater number of inter-node paths leads to more messages and thus to larger latency. The number of sites on the inter-node surfaces has impact on the total size of messages, leading to larger aggregated message size.



(a) Layout 1



(b) Layout 2

Figure 1: A simplified illustration of the number of inter-node subvolume communication paths and sites under different layouts. Placing the subvolumes on nodes differently can lead to different number of paths and size of messages. More inter-node communication paths do not necessarily lead to more sites on inter-node surface.

3. LAYOUT REMAPPING

3.1 NodeVolume

MILC is not aware of the topology of the machine on which it runs. In order to enable MILC to map the subvolumes to cores on different nodes, we introduced the concept of the *NodeVolume* and *NodeCut*.

MILC maps subvolumes to cores by assigning each subvolume a rank by the subvolume's coordinates. In order to map subvolumes to cores in the manner that we want, we introduced a new remapping layer by which we can assign a particular group of subvolumes to a node. In order to describe the group of subvolumes that are assigned to a given node, the concept of NodeVolume is introduced. We describe a 4-d NodeVolume by the notation $A \times B \times C \times D$, which means in this node there are A consecutive subvolumes in the $\hat{x}$ direction, B consecutive subvolumes in the $\hat{y}$ direction, C consecutive subvolumes in the $\hat{z}$ direction, and D consecutive subvolumes in the $\hat{t}$ direction. Figure 2 shows how a lattice is divided into NodeVolumes. Given a lattice size, and the choice of size of NodeVolumes and subvolumes, we can calculate the number of inter-node subvolume communication paths and sizes of inter-node messages.

We define the *lattice size* L by the four-tuple (l_x, l_y, l_z, l_t), where l_i is the length of the lattice, measured in sites, in the $\hat{i}$ direction. We define a *subvolume* as the hyper-rectangular volume of sites which will be handled by a single core, and the *subvolume size* $S = (s_x, s_y, s_z, s_t)$ similarly to L. We next define a *nodevolume* as the hyper-rectangular volume

of sites which will be handled by the cores of a single node, and the *nodevolume size* $N = (n_x, n_y, n_z, n_t)$, where n_i is the length of the nodevolume, in subvolumes (not lattice sites), in the $\hat{\imath}$ direction. We define the four-tuple *nodecut* $C = (c_x, c_y, c_z, c_t)$, where c_i is the number of nodevolumes in the $\hat{\imath}$ direction into which the lattice is divided. Finally, we define the four-tuple Q by

$$Q_i = N_i C_i.$$

For example, $C = (1, 2, 1, 2)$ means there are two nodevolumes in each of the $\hat{y}$ and $\hat{t}$ directions, and one in each of the the $\hat{x}$ and $\hat{z}$ directions. Q_i indicates how many subvolumes are in $\hat{\imath}$ direction.

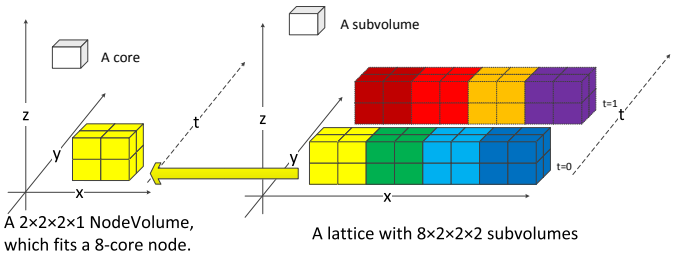


Figure 2: Subvolumes of the same color form a nodevolume, and are placed on the same node. (To see the color, please refer to the electronic version of this paper.)

3.2 Inter-node Subvolume Communication Paths

The Inter-node Subvolume Communication Path (ISP) in $\hat{\imath}$ direction is the number of pairs of subvolumes exchanging data in that direction. This parameter affects the number of packages sent in $\hat{\imath}$ direction. The number of packages has great impact on the performance of network, since each package has individual network startup time and cleanup time, which causes delay.

The total ISP of a lattice is determined by Q and C. If $Q = (q_x, q_y, q_z, q_t)$, $C = (c_x, c_y, c_z, c_t)$, the overall ISP is

$$\begin{aligned} ISP = {} & k_x \times c_x \times q_y \times q_z \times q_t \\ & + k_y \times q_x \times c_y \times q_z \times q_t \\ & + k_z \times q_x \times q_y \times c_z \times q_t \\ & + k_t \times q_x \times q_y \times q_z \times c_t \\ \forall .i \in & \{x, y, z, t\}, if\ c_i > 1, then\ k_i = 1,\ else\ k_i = 0 \end{aligned}$$

$c_i > 1$ indicates the lattice is divided and put on different nodes in that direction, which leads to inter-node communication. For example, if $Q = (4, 4, 2, 1)$ and $C = (1, 1, 2, 2)$, ISP in $\hat{z}$ direction is $4 \times 4 \times 1 = 32$ between any two neighbor nodes. The ISP in $\hat{t}$ direction is $4 \times 4 \times 2 = 32$ between any two neighbor nodes. In total, ISP is $16 \times 2 + 32 \times 2 = 96$.

For the lattice with the same number of total subvolumes, the numbers of communication paths between subvolumes are the same. However, numbers of inter-node paths can be different when subvolumes are assigned to nodes in different ways. This is why layout is important.

3.3 Sites on the Surfaces of NodeVolume

Number of Sites on the Surfaces of NodeVolumes (SSN) has impact on the size of data transferred over network. When the difference of the number of sites on inter-node surface is big enough, it can have combined effects to the communication cost with ISP.

The data of the outer three layers of sites in a subvolume is packaged together and sent to its adjacent subvolume. Due to the symmetrical communication pattern of MILC, the amount of overall data exchanged is proportional to SSN.

$$\begin{aligned} nvs &= N \times S \\ &= (nvs_x, nvs_y, nvs_z, nvs_t) \\ SSN &= nvs_y \times nvs_z \times nvs_t \\ &+ nvs_x \times nvs_z \times nvs_t \\ &+ nvs_x \times nvs_y \times nvs_t \\ &+ nvs_x \times nvs_y \times nvs_z \end{aligned} \tag{2}$$

nv is the size of NodeVolume in terms of subvolumes. $subsize$ is the size of a subvolume in terms of sites.

3.4 Communication Cost Model

Since we have found the number and size of messages is correlated with ISP and SSN separately, we can get the following model.

$$cost = \alpha \times ISP + (1 - \alpha) \times SSN,\ 0 \le \alpha \le 1$$

$cost$ is the inter-node communication cost. α is the weight to be determined. For example, if SSN is the same for different layouts of the lattice, α should be large. If the difference in terms of SSN is large, α should be smaller.

4. EVALUATION

4.1 Environment

The experiments in this paper were conducted on D/S cluster at Fermilab. D/S is a 245-node cluster with quad-socket eight-core Opteron 6128 (2.0 GHz) processors and a quad-data-rate Infiniband fabric. A group of 8 nodes (256 cores) was reserved for the experiments. The nodes within the same group are connected to the same Infiniband switch. So the inter-node connections are identical between any two nodes of the same group.

We use the application ks_imp_dyn in MILC for evaluation, which is used for simulating full QCD in staggered fermion scheme. It is compiled and ran by mvapich-1.2rc1 via NUMA binding wrapper numa_32_mv2. One exception was that we used Openmpi with TAU [4] to instrument MILC.

4.2 Metrics

In our experiment, we use TPI to evaluate the performance. TPI stands for Time Per Iteration of each conjugate gradient step.

In the graphs presented in this paper, we plot the median TPI of conjugate gradient steps for each layout. By doing so, the actual and predicted TPI can be easily compared. In addition, the accuracy of the model can be easily verified.

We ran MILC with each layout for over 5 times with over hundreds of iterations, in order to get enough samples for statistics.

4.3 Linear Model

In the evaluation, we first use the experimental results to build a linear model, and then use the model to predict the performance based on the layout.

4.4 Combined Impacts of ISP and SSN

The network performance is mainly determined by the latency and bandwidth, as presented in Formula 1. In MILC, they are associated with ISP and SSN. In this part, we build a linear model based on both of ISP and SSN. Figure 3 and Figure 4 show the actual and model-based TPI on 128 cores of 4 nodes and 256 cores of 8 nodes. As shown in the figure, this linear model is accurate. In each figure, there are several groups of performance results. The predicted results fit the acutal results well. In Figure 3, when $Q = (2, 4, 2, 8)$, the optimum TPI based on our prediction is around 31% less than the worst case, showing a great potential of improvement.

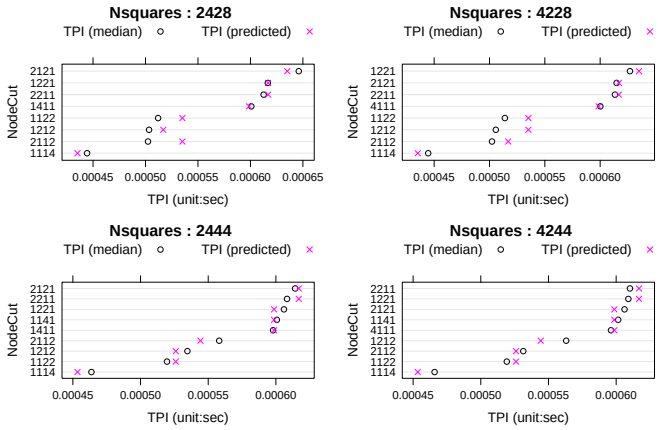


Figure 3: $L = (12, 12, 12, 24)$, 4 nodes, 128 cores, $Nsquares = Q$. Model based on ISP and SSN. We can further categorize the performance by ISP. We can see the performance matches the prediction quite well.

5. CONCLUSION

In this paper, we explore the scalability of scientific applications by a case study of MILC. In order to scale up the system, this paper focuses on the reducing the inter-node communication cost by optimizing layout. We added layout support to MILC and propose a model to estimate the communication costs. This model can yield optimal layout for scientific applications, hence boost their performance This model can help scientists to find the optimal layout for their scientific applications and get results faster. We obtain up to 31% of performance improvement.

In this future, we plan to apply the model to even larger scale experiments to further confirm it. Since the number of processors/cores per node is growing fast, minimizing the intra-node communication cost is also necessary to improve the scalability of scientific applications. That is another one of our future topics. We are also planning to explore other scientific applications with dynamic communication patterns and improve their scalability.

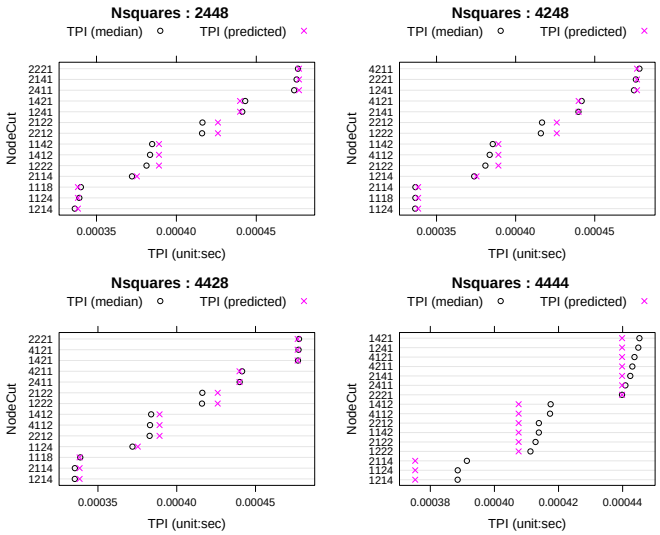


Figure 4: $L = (12, 12, 12, 24)$, 8 nodes, 256 cores, $Nsquares = Q$. In this case, we can also see that the performance can first be categorized by SITEStotal.1nAd, and then be further categorized by ISP. It matches the performance pattern.

6. ACKNOWLEDGMENTS

The authors are thankful to Ce Yu of Tianjin University, Yanlong Yin, Siyuan Ma and Hui Jin of Illinois Institute of Technology for their constructive and thoughtful suggestions toward this study.

7. REFERENCES

[1] Infiniband.
[2] The MIMD Lattice Computation (MILC) Collaboration.
[3] Top 500 supercomputer site.
[4] Tuning and analysis utilities.
[5] S. Borkar. Thousand core chips: a technology perspective. In *Proceedings of the 44th Annual Design Automation Conference*, pages 746–749. ACM, 2007.
[6] J. Carter, Y. He, J. Shalf, H. Shan, E. Strohmaier, and H. Wasserman. The performance effect of multi-core on scientific applications. Technical report, Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA (US), 2007.
[7] T. Hoefler and M. Snir. Performance engineering: a must for petascale and beyond. In *Proceedings of the third international workshop on Large-scale system and application performance*, pages 1–2. ACM, 2011.