Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 8890721, 13 pages
https://doi.org/10.1155/2020/8890721

Research Article

Communication Optimization Technology Based on Network Dynamic Performance Model

Xiang Cui,[1,2] **Xiaowen Li**,[3] **and Bei Wang**[2]

[1]*College of Computer & Information Engineering, Henan University, Kaifeng 475000, China*
[2]*HCST Key Lab, at School of EECS, Peking University, Beijing 100871, China*
[3]*Henan Finance University, Zhengzhou 450000, China*

Correspondence should be addressed to Xiaowen Li; 1206375360@pku.edu.cn

Received 31 August 2020; Revised 24 September 2020; Accepted 28 September 2020; Published 24 October 2020

Academic Editor: S. A. Edalatpanah

This work analyses different communication modes in applications of supercomputing, proposes a communication dynamic performance model based on topology awareness, and realizes the prototype system of all-to-all communication and stencil communication optimization based on this model. Basic tests on the optimization of all-to-all communication and stencil communication were carried out on the Sunway TaihuLight System, and this achieved obvious optimization results. Several applications, including molecular dynamics simulation and turbulence simulation, have been optimized and tested. The average performance has been improved obviously. It can be expected that, for other large-scale applications, this optimization method can also be used to obtain significant improvement in communication performance.

1. Introduction

Although supercomputers have been making breakthroughs at peak computing rates, their application levels have lagged behind. While researchers strive to improve the application level of high performance computing, researchers in the field of computer science also need to do research work to improve the availability and ease of use of large heterogeneous systems. When the system expands to a certain scale, not only the scalability of performance needs to be solved, but also the scalability of system availability and ease of use.

An important aspect of improving the performance of HPC applications (especially communication-intensive applications) is to improve the performance and stability of the communication part of the application. From the perspective of architecture, for large heterogeneous systems, the health status of each node in the system and the use of the network all change at any time. Therefore, the communication performance must be optimized according to the dynamic performance model of the system. Based on the architectural characteristics of heterogeneous systems, its dynamic performance model needs to consider not only the network communication performance between nodes, but also the data transmission performance between different types of memory within nodes (such as main memory at different locations in the NUMA structure or main memory and MIC memory in the MIC accelerating system). In addition, support for heterogeneous systems of different types of memory transfer mode not only need to include simple transposes of data dimensions between nodes but also should coordinate data distribution dimension and the structure of the system network topology and support more general complex dimension transformation.

The research idea of this paper is that, in addition to considering the physical structure of the network, optimization should be carried out based on the dynamic performance model of the network. As for the supercomputing system, after the system reaches a certain scale, the delay, bandwidth, and blocking of the communication between nodes are greatly affected by the network topology. In order to achieve the reasonable map between data distribution dimension and the system network topology, it is necessary to detect system data communication dynamic topology, through test sets and test system (including nodes between

the storage unit within and between network nodes) communication performance, build the dynamic topology model of heterogeneous system communication, and finally realize the process/thread-nuclear efficient mapping optimization.

The significance of this research for the supercomputing system is that, with the expansion of the system and network scale, the scalability of the set communication performance will become a prominent problem. This problem is exposed on existing systems and will become more prominent on future larger systems. Therefore, it is necessary to optimize the communication implementation according to the network topology structure to alleviate such problems to some extent.

The research idea adopted in this paper is to analyze the communication characteristics of different types of applications and study the implementation of the dynamic topology detection mechanism of data communication. Considering not only the physical structure of the network but also the dynamic performance model of the network, this paper optimizes the implementation of complex set communication by improving the process-computation kernel mapping.

2. Related Work

Many research studies focus on the optimization of set communication for static topology structure of system network. Faraj et al. [1] optimized MPI set communication on the Blue Gene/P system according to the process distribution on the node, which was divided into global distribution, Torus cube distribution, and irregular distribution. Jain and Sabharwal [2] optimized bucket algorithms (including Allgather, Reduce-Scatter, and Allreduce) based on IBM Blue Gene/P 3D Torus network topology. The performance of symmetric Torus network is close to the theoretical constraints, while the performance of asymmetric Torus network is close to the theoretical constraints of the maximum dimension. Sack and Gropp [3] implemented and optimized Allgather and Reduce-Scatter algorithms on BlueGene/P. Almási et al. [4] optimized MPI set communication based on BlueGene/L high-speed Torus/Collective network topology. Adachi et al. [5] optimized MPI set communication for the K system mesh/Torus network topology. Faraj and Yuan [6] took the topology description of the system network as input and used the generator to generate the corresponding efficient algorithm automatically. Similarly, Faraj and Yuan [7] designed an automatic program generator to generate Alltoall algorithm for big data messages with network topology information as input, which achieved better performance than LAM/MPI and MPICH in Ethernet switch clusters. Nicolai et al. [8] proposed the concept of average logical communication distance and its calculation formula and designed an algorithm called neighbor exchange to optimize Allgather performance. Paul and Gropp [9] optimized the aggregation communication algorithm on the torus network connected with multiple ports.

Some researches focus on dynamic optimization of set communication for system network topology. Faraj et al. [10, 11] designed a method called star-MPI (self-tuning adaptive routines for MPI collective operations), which can dynamically select the algorithm for ensemble communication in a network with unpredictable performance. This method tests various possible schemes and uses a certain prediction mechanism to delete the algorithm with low performance to save testing time. Vadhiyar et al. [12] used an automatic optimization technique similar to FFTW for aggregate communication tuning. First, test the optimal buffer size applicable to the algorithm under a certain number of processes, then test the performance of different algorithms against a certain message size, and finally repeat the above steps for different numbers of processes, so as to determine the optimal set communication algorithm under different number of processes. Subramoni et al. [13] analyzed the factors causing network congestion in the large-scale InfiniBand cluster, represented the dynamic topology characteristics of the system by generating path matrix, and optimized Alltoall implementation, which achieved 12% performance improvement for P3DFFT on the 4,096 core network. Mamadou et al. [14] used p-Log$p$ point-to-point model to predict the performance of different algorithms to determine the optimal implementation algorithm of Alltoall based on the dynamic changes of system network load, and achieved good results on Infiniband and Gigabit Ethernet networks. Patarasuk and Yuan [15] optimized big-message All-Reduce under the tree network structure, enabling each process to send and receive the minimum amount of data and avoid the occurrence of blocking, and achieved performance improvement on Myrinet, InfiniBand, and Ethernet clusters. Kandalla et al. [16] modeled the communication performance by detecting the topology information of the large-scale InfiniBand network, analyzed the performance overhead of collection communication, and optimized Gather and Scatter routines. Ma et al. [17], based on the process distance, network hardware topology, and runtime communicator information, generated topology aware Broadcast and All-Gather implementations. Gallardo et al. [18] implemented the MPI Advisor, an easy-to-use software tool for programmers to dynamically monitor application execution and optimize the MPI environment to improve performance. Bhatele et al. [19] speculated the possible causes of network communication blocking by dynamically monitoring the performance of the application.

Other studies optimize the aggregation communication for the system network characteristics. Usually, MPI collection communication is designed according to the assumption that one node can only communicate with another node at a certain time. Chan et al. [20] improved several collection communication functions including Broadcast, Reduce, Scatter, Gather, All_gather, Reduce_scatter, and all-Reduce, aiming at the feature that one node can communicate with multiple other nodes at the same time in the IBM Blue Gene/L system. Faraj et al. [21] analyzed that, in the network composed of cut-through and store-and-forward switches, when the message is large enough, the subnet composed of a minimum spanning tree connection can achieve nearly optimal performance for Alltoall broadcast communication. Zhang and Deng [22] proposed that the

average distance between nodes could be reduced more effectively and the broadcast communication performance could be improved by adding shortcut connections with strategies rather than network dimensions on Torus network. Song and Hollingsworth [23] proposed a new broadcast communication algorithm using MPI-2 unilateral communication and pipe-logging mechanism, and the quantitative analysis and experiment of P Log$P$ parallel computing model verified that the algorithm had better performance improvement than the traditional algorithm. Mamidala et al. [24] analyzed performance scalability and performance/memory consumption in achieving set communication and unilateral communication using InfiniBand Reliable Connection (RC) and Unreliable Datagram (UD). In systems using InfiniBand network, MPI communication function was usually used in transmission mode RC. However, in large-scale networks, in order to save memory consumption in establishing full connection in RC, Koop et al. [25] suggested that using Unreliable Datagram (UD) realizes MPI's aggregation communication function. Qian and Ahmad [26] implemented several RDMA multiport communication functions based on the characteristics of its network multi-Rail on the QsNetII cluster. Hasanovn [27] optimized the parallel matrix multiplication algorithm on large-scale network systems by reducing communication overhead. Mistry et al. [28] found that switching components on InifiniBand network would become the bottleneck of Alltoall communication.

Some researchers have developed set communication optimization based on process-node and process-CPU core mappings. Karlsson et al. [29] improved the performance of multidimensional process groups in broadcast communication in different dimensions by applying hierarchical optimization process-CPU core mapping. Balaji et al. [30] analyzed the influence of process-node correspondence in three-dimensional Torus network topology structure of Blue Gene/P system on application performance and provided application communication mode information to optimize the communication performance before application loading. Based on Torus network topology, Mittal et al. [31] designed methods for each subcommunicator's nonblocking routing data when the subcommunicator formed by multiple discontinuous nodes concomitant communication in a loosely synchronized manner and verified the performance in the Blue Gene/P system. Bhatele et al. [32] developed a tool called Rubik to optimize the communication performance of the subcommunicator in the application by adjusting the process-node mapping relationship. Karlsson et al. [33] optimized the multidimensional MPI set communication on the multidimensional Torus network structure and reduced the communication traffic between nodes on Jaguar system by changing the process-CPU kernel mapping relationship to optimize the performance. Zahavi et al. [34] proposed that when an application runs on a fully or partially filled fat tree structure, the MPI process-node mapping relationship should reflect the structural characteristics of the network, and the simulation verified that its nonblocking routing method has higher performance in Alltoall communication.

3. Communication Characteristics of Different Types of Applications

In order to carry out the research of communication performance optimization technology based on topological structure, it is necessary to study the characteristics of communication mode applied in the supercomputing system. Therefore, the communication characteristics of turbulent flow application and crystal silicon solidification process simulation application are studied.

3.1. All-to-All Communication. The communication characteristics of direct numerical turbulence simulation applications are all-to-all communication. The core of direct numerical turbulence simulation is the Fourier transform of a three-dimensional cube, which is also the most difficult part of optimization. This part of the data volume is large. For the 3 d cube with side length of 16,384, the data volume is huge, up to 16 TB. Standard practice requires the entire data to be transposed, resulting in frequent data transfers, one data transfer per iteration time step, and more than 10 such cube FFTs.

The calculation design of this part is as follows. The original data are stored in ordinary three dimensions, and the right-most dimension is the continuous dimension. The whole cube has $N3$ singularly complex numbers. The array dimension representation method is used, and the initial data is marked as an array type $(x/[N])(x/[N])(y/[N])(z/[N])$. We use P processes to participate in the calculation. The data is divided equally into P parts, and each process is allocated N/P squares with $N * N$ sides. That is, the cube slices are assigned to each process on the first dimension. At this point, the data distribution is denoted as $(x_1/[p])(x_2/x_2)(y/[N])(z/[N])$. Then the local FFT of the two-dimensional matrix is completed in each process. Then, an all-to-all communication takes place between all processes to complete a transpose of the 3D data on x dimension, transforming the dimension into a continuous dimension on a single process. To do this, the second dimension also needs to be split into $(x_1/[p])(x_2/[N/P])(y_1/[p])(y_2/y_2)(z/[N])$. First, x and y are swapped, the transposition becomes $(y_1/[p])(y_2/[N/p])(x_1/[p])(x_2/[N/p])(z/[N])$, and then a local data transpose is done; that is, z and y are swapped, and $(y_1/[p])(y_2/[N/p])(z/[N])(x_1/[p])(x_2/[N/p])$ distribution is achieved. Finally, one-dimensional FFT of x is done. This completes the transformation of 3D FFT.

It is found that there are significant performance differences when using different nodes for communication. As shown in Figure 1, the abscissa represents different node groups; each group has 64 nodes, a total of 32 groups for all-to-all communication, and different curves represent 5 performance measurements. It can be seen that the performance of different groups differs significantly, and the performance of each node of the same group has certain stability. This shows that, by changing the process-computational kernel mapping to optimize the implementation of complex set communication, effective performance improvement can be expected.

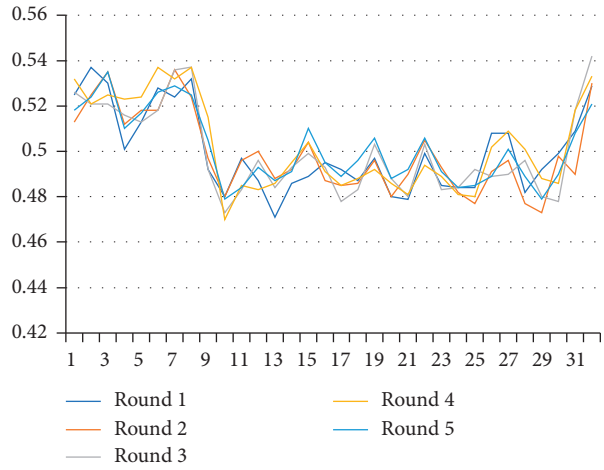

Figure 1: Comparison of the communication performance of different groups of processes.

3.2. Stencil Communication. The communication features of the silicon solidification process simulation application are stencil communication mode. We tested the effect of different communication patterns and process dimensional distribution patterns on performance.

In one-dimensional communication mode, each process sends data of unit message length (2 K) to 26 surrounding neighborhoods at the same time. After communication, each process receives all messages from 26 surrounding neighborhoods. An example of a one-dimensional communication pattern is shown in Figure 2.

In the two-dimensional communication mode, in the first communication, each process sends data (2 K) of unit message length to the surrounding 8 neighborhoods at the same time. After the communication, each process receives all messages from the surrounding 8 neighborhoods. On the second communication, each process will send the message data containing its 8 neighborhoods (2 K $*$ 9) to the upper and lower neighborhoods at the same time. After the communication, each process receives all the messages from the surrounding 26 neighborhoods. An example of two-dimensional communication mode is shown in Figure 3.

In 3D communication mode, for the first communication, each process sends data (2 K) of unit message length to left and right neighborhood at the same time. After communication, each process receives all messages from about 2 neighborhoods. In the second communication, each process will send the message data containing its two neighborhoods (2 K $*$ 3) to one neighborhood before and after at the same time. After the communication, each process receives all the messages from the surrounding eight neighborhoods. In the third communication, each process will send the message data containing its 8 neighborhoods (2 K $*$ 9) to the upper and lower neighborhoods at the same time. After the communication, each process receives all the messages from the surrounding 26 neighborhoods. The example figure of 3D communication mode is shown in Figure 4.

The performance comparison of the three communication modes is shown in Figure 5. It can be seen that the 3D 2-2-2 mode has obvious performance advantages.

The ranking of processes in different dimensions demonstrates the complexity of neighborhood relationships, which is also critical to performance. Under the three different permutations, the above three dimensional communication mode is adopted in the communication mode. We test the performance trend of computing plus communication (unit message length is 2 K), communication only (unit message length is 2 K), and communication only (unit message length is 8 K) under different sizes. It can be seen that the choice of different communication modes has a significant impact on performance, and it can also be expected that the improvement of process-computational kernel mapping optimization can also promote the improvement of communication performance.

4. Communication Dynamic Performance Model

In addition to considering the physical structure of the network, this scheme considers the dynamic performance model based on the network for optimization, which is an innovative work of this study.

The work of this paper is carried out on the Sunway Taihulight supercomputer. The Sunway Taihulight supercomputer consists of 40 computing cabinets and 8 network cabinets. In each computing cabinet, four supernodes composed of 32 computing plug-ins are distributed among them. Each plug-in is composed of four operation nodal plates, and one operation nodal plate contains two high-performance processors "Shenwei 26010." One cabinet has 1024 processors, and the whole machine has 40,960 processors. Each single processor has 260 cores, the motherboard is designed for double nodes, and each CPU has 32GBDDR3-2133 solidified on-board memory. This optimization method may not be directly applicable to other nontree network structures. The corresponding performance model should be established according to the specific network structure. However, the thought in this paper can be used for reference.

The communication dynamic performance model based on topology awareness is designed as follows: $M = (N, E)$, where $N(M)$ represents the set of all nodes in the network; the elements in $E(M)$ are triplets; for any $<a, b, d> \in E(M)$, there is $a, b \in N(M)$, and d is real number, indicating the network communication performance between node a and node b. It can be seen that what the model describes is actually a fully connected directed graph weighted by the network performance between nodes, as shown in Figure 6.

The technical route proposed in this paper is to test the communication performance of each link of the system (including the communication between storage components within the node and the network between nodes) through the example test set, so as to build the dynamic topology model of the communication of the whole heterogeneous system. The specific communication instance test set can include the following:

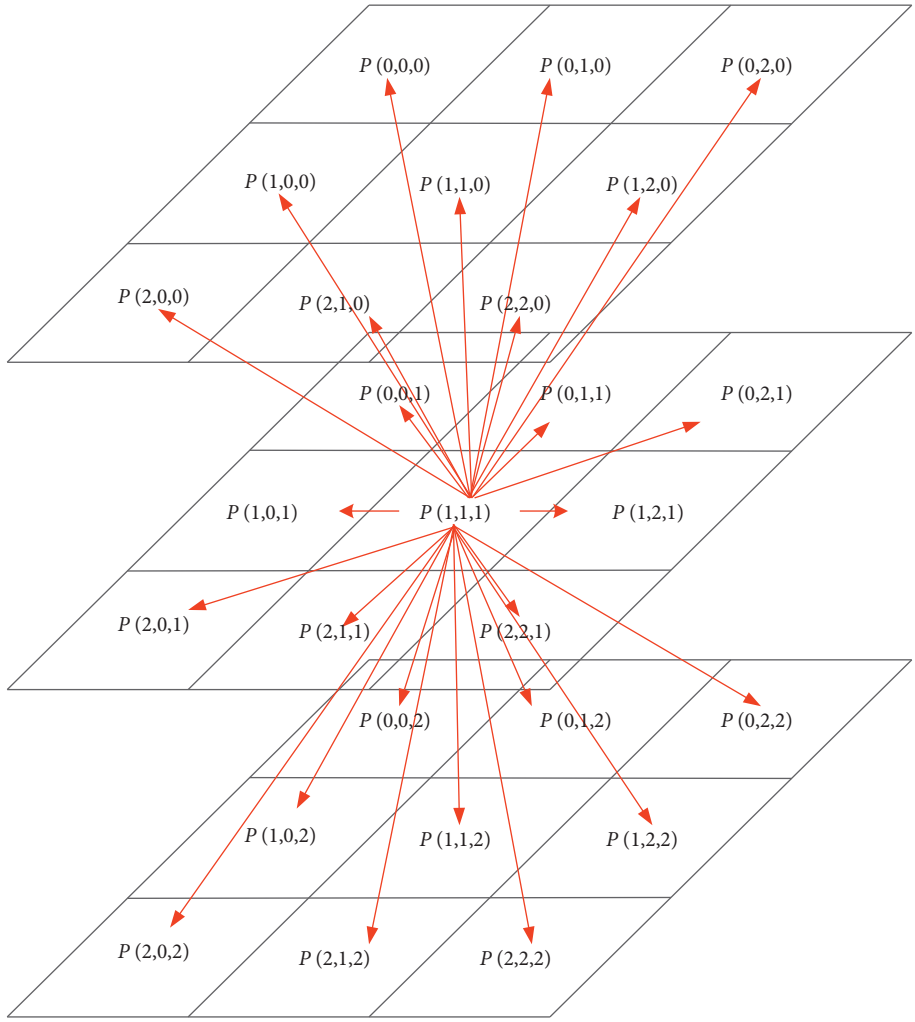

Figure 2: One-dimensional communication mode.

① For the internal nodes, the data transmission performance between storage components under different granularity is tested to fully describe the "distance" between each storage component.

② For nodes, the bandwidth and delay of communication between nodes under different transmission granularity are tested, and the "distance" between nodes is depicted. Stress test the throughput performance and other constraints of network switches at all levels.

③ Test the model of the interplay between the performance of various concurrent transports.

④ This profiling process should be conducted in an efficient and automated manner and can be retested at intervals during application execution to modify the dynamic topology model.

The dynamic communication model is constructed by detecting the dynamic topology of data communication. The dynamic communication model is represented by graph structure: each point in the graph represents network nodes, and the edge between nodes represents network characteristics such as bandwidth between node pairs.

Considering the network dynamic performance model, the process-computation kernel mapping optimization is carried out for applications with different communication characteristics:

① Different types of communication characteristics have different requirements for communication. For example, whereas full-to-full communication requires network relationships between nodes, stencil 2-2-2 communication only requires network relationships between associated neighbor nodes.

② The structure of dynamic communication graph is taken as a complete graph, and the optimal subgraph is sought to make it match the performance requirements of different communication characteristics mentioned above.

③ The node characteristics of the subgraph should conform to the known network physical structure model.

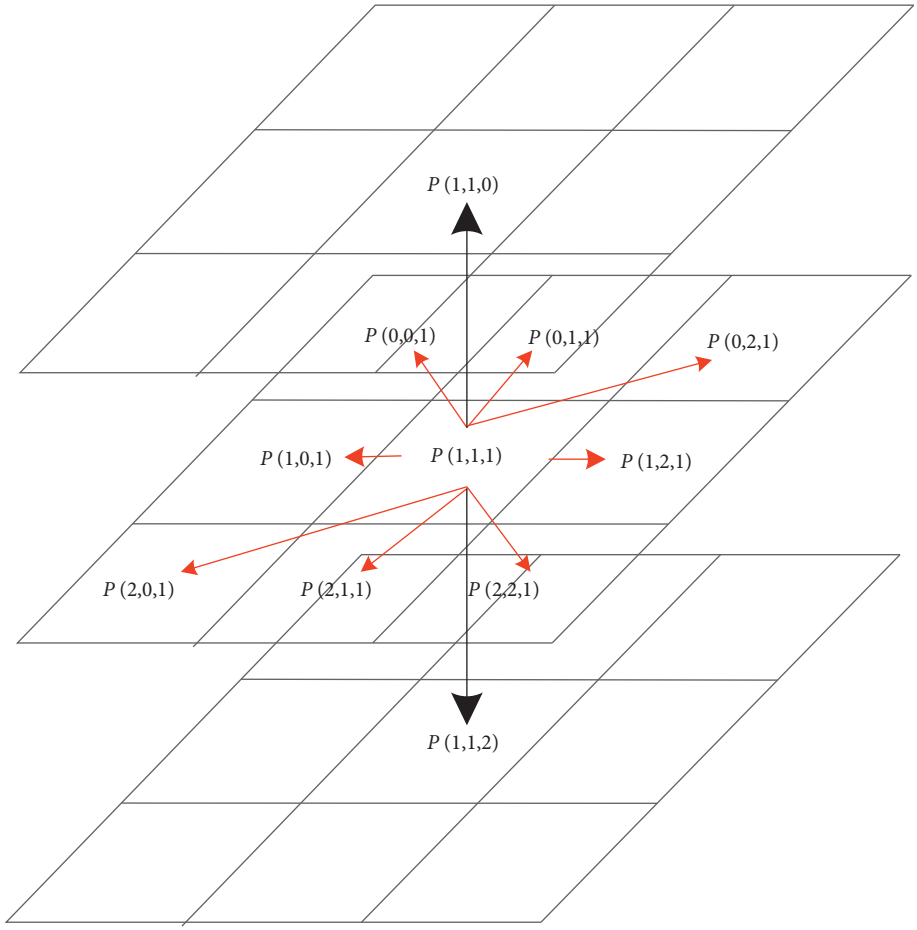

Figure 3: Two-dimensional communication mode.

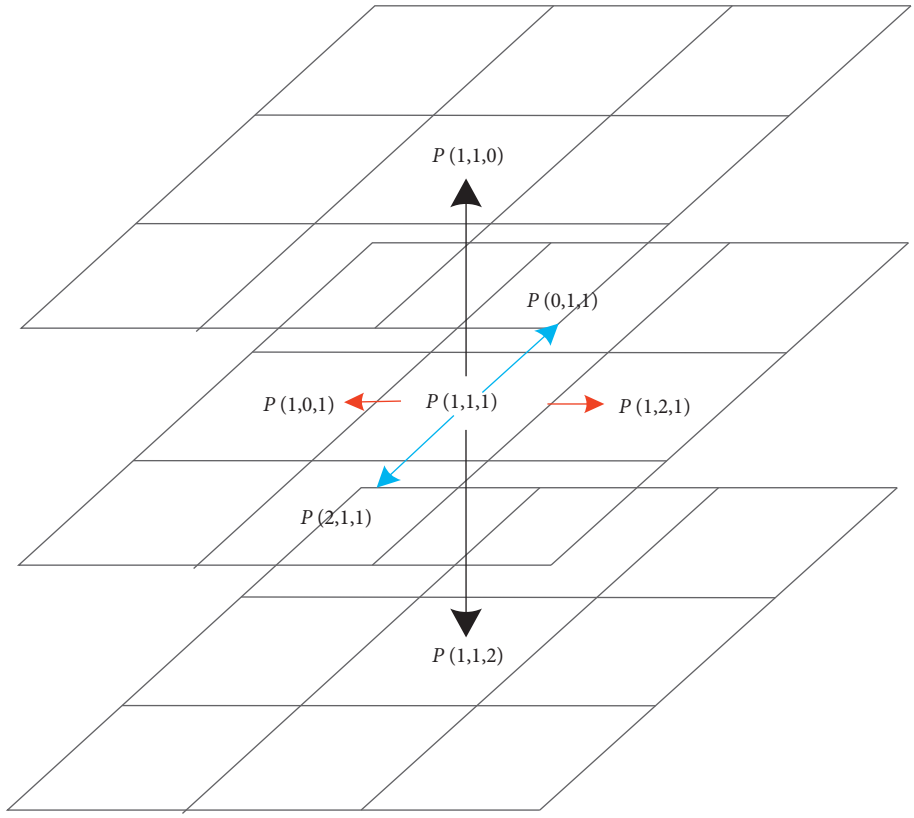

Figure 4: Three-dimensional communication mode.

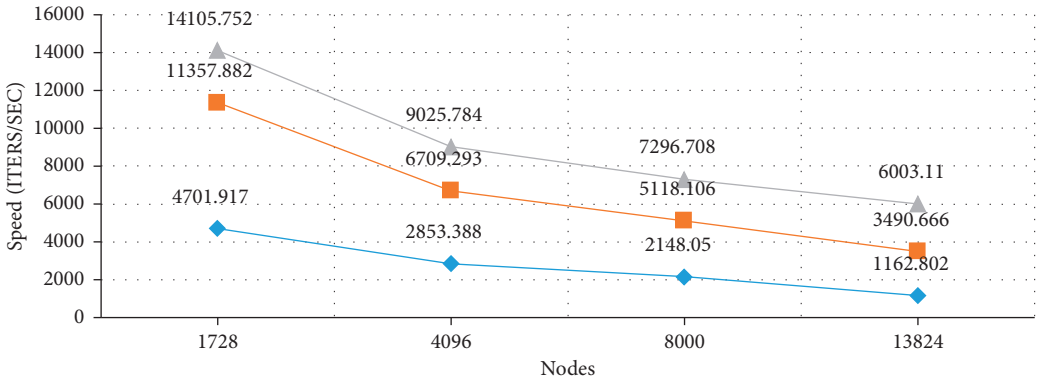

Figure 5: Performance comparison of various communication modes.

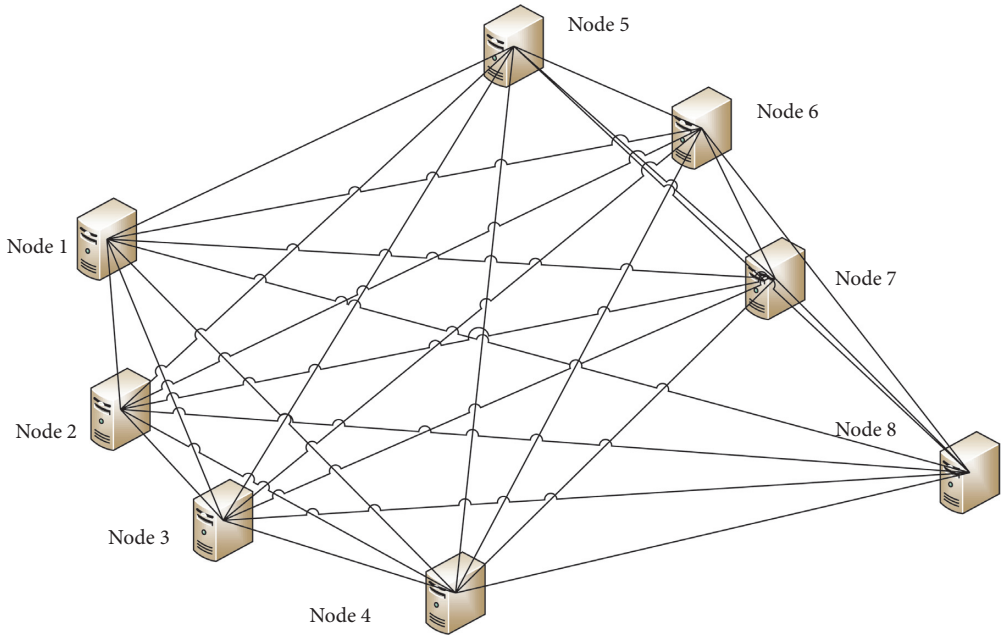

Figure 6: The full connection diagram of the network communication dynamic performance model.

④ Validate process-computational the availability of kernel mapping optimizations with examples: in addition to the all-to-all communication and stencil 2-2-2 communication modes described above, consider using other MPI collection communication modes for validation. For example, for broadcast communication mode, it is necessary to construct a subgraph to form a tree structure corresponding to the implementation of broadcast communication mode and make this tree structure reach the optimal level.

5. Optimize All-to-All Communication Based on the Dynamic Performance Model

This section takes optimal set communication based on dynamic performance model as an example to demonstrate the design idea of the scheme. As shown in Figure 7, under the communication pressure condition, the bandwidth and delay of communication between nodes under different transmission granularity were tested.

According to the bandwidth and latency characteristics, the topology of each node is represented as a full connection diagram, and the distance between nodes represents the network performance between nodes. It can be seen that in this dynamic topology that nodes 1–4 are located in the switch network of the same layer, while nodes 5–8 are located in the switch network of another layer.

If the all-to-all-communication process-computational kernel mapping optimization is carried out at this time, if two nodes are needed, then 2–3 nodes are selected as the best; if four nodes are needed, then 1–4 nodes are selected as the best.

The dynamic topology structure can not only optimize the node selection and process-computation kernel mapping optimization, but also optimize the implementation of set communication. For example, if the broadcast communication of the eight nodes in the figure is realized, it is

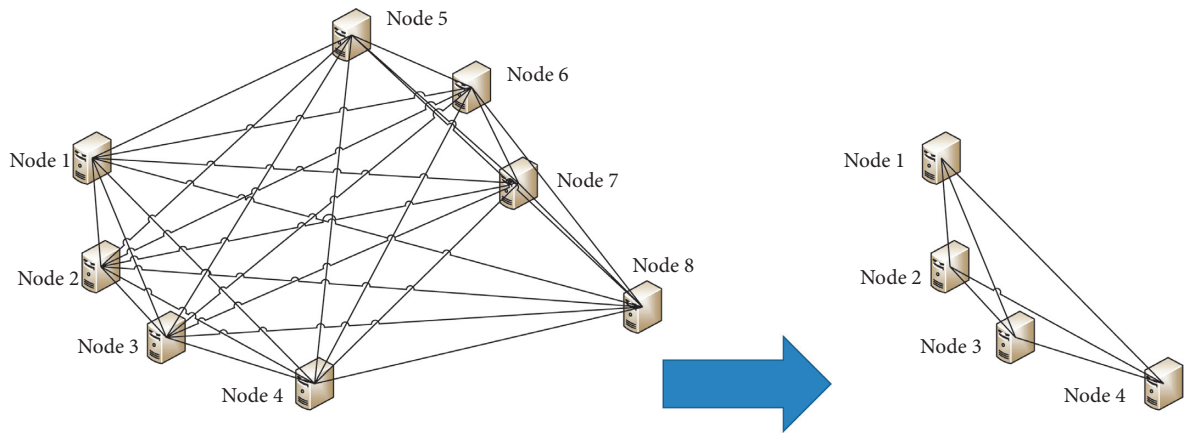

Figure 7: Test network connections between nodes in the network.

advisable for the upper nodes of the forwarding tree structure to select nodes 1–4.

According to the design of dynamic performance model, the prototype system began to implement and test.

Through the example test set, the communication performance of each link of the system is tested, and the dynamic topology model of the whole system communication is built. The test method of the test set is as follows: only considering the network communication performance between the main core, repeated ping-pong communications will be carried out between any node pair at the same time. Several rounds will be conducted in this process to record the communication performance between each node. The dynamic communication model is expressed as a graph structure. According to the graph structure, the optimal fully connected subgraph is sought, and all-to-all communication performance is tested. The algorithm to find the optimal fully connected subgraph is shown in Figure 8.

According to the above implementation methods, based on the network dynamic performance model, the all-to-all communication features of the program are tested by changing the process-computational kernel mapping.

Since the test is carried out in a shared partitioned environment and the workload and network load change at any time, the following factors will be considered for the test: the test operation program is a program that has carried out several rounds of MPI_Alltoall communication. For each batch of tests, several times will be performed to eliminate data with obvious abnormal performance results (there is an order of magnitude difference between the performance results of the two adjacent tests). The test job before optimization is issued with command and uses the default node allocation mode. When the optimized test job is submitted, specify the nodes and mapping mode selected by the optimization. To ensure fair competition, the two types of work will be submitted in different terminals at the same time. If the nodes selected by both parties have duplicates, the two test jobs are submitted in turn.

From the test results shown in Table 1, it can be seen that significant optimization effect of communication performance can be achieved after node optimization selection and process-computational kernel mapping optimization based on dynamic topology structure. The values in the table represent the time in seconds needed to complete a round of communication. For the operation with large communication data volume and node size, the performance improvement before and after optimization is more obvious. It is also expected that the larger the job node size is, the easier it is to benefit from node optimization selection and process-computational kernel mapping optimization.

Note that the test is carried out in a shared partitioned environment. The workload and network load change at any time, so the acceleration effect test may be different each time (but it also meets the requirements of the real scenario). After repeating several tests, the optimization effect can be clearly reflected.

6. Optimize Stencil Communication Based on the Dynamic Performance Model

This section takes the stencil communication optimization based on dynamic performance model as an example to demonstrate the design idea of the scheme.

In all nodes on the network, the communication performance between each node is tested. Combine nodes that do better at communicating into smaller stencil blocks (2 by 2 by 2) and then build larger stencil blocks (4 by 4 by 4) from smaller stencil blocks. This process iterates until the node size required for the application is constructed as shown in Figure 9.

Through the example test set, the communication performance of each link of the system is tested, and the dynamic topology model of the whole system communication is built. This process is similar to the all-to-all communication optimization implementation process and will not be repeated here. The algorithm to construct a communication node block using stencil is shown in Figure 10.

Based on the above implementation method, a program with communication characteristics of stencil is tested by changing process-computational kernel mapping based on the network dynamic performance model.

Since the test is carried out in a shared partition environment and the workload and network load change from time to time, the following factors will be considered

```
The initial set of selected nodes is set empty
The two closest nodes are selected from all candidate nodes
While (number of selected nodes < number of required nodes){
    Select the node newOne from the candidate nodes,
        Ensure that the sum of newOne and all selected nodes is minimum;
    Add the newOne node to the selected node set;
}
```

Figure 8: Algorithm for finding the optimal fully connected subgraph.

Table 1: All-to-all communication optimization test results of the Sunway TaihuLight system.

Number of nodes	512			1024		
Message size	Before optimization	After optimization	Speed-up ratio	Before optimization	After optimization	Speed-up ratio
1 k	0.123503	0.124768	0.989861	0.304584	0.297603	1.023457
2 k	0.165302	0.158945	1.039995	0.452202	0.45197	1.000513
4 k	0.176427	0.174712	1.009816	0.438695	0.421362	1.041136
8 k	0.166236	0.162356	1.023898	0.425682	0.419812	1.013982
16 k	0.179189	0.163515	1.095857	0.438668	0.382685	1.14629
32 k	0.197772	0.181229	1.091282	0.502881	0.414985	1.211805
64 k	0.465669	0.44584	1.044476	0.730972	0.631371	1.157754
128 k	0.492282	0.482712	1.019825	1.457561	1.301542	1.119872
256 k	0.808534	0.777597	1.039785	2.295667	2.181182	1.052488
512 k	1.648359	1.501278	1.097971	4.616991	4.168189	1.107673
1 m	3.168055	2.958922	1.070679	9.865502	8.007522	1.232029
2 m	6.457053	6.02656	1.071433	—	—	—

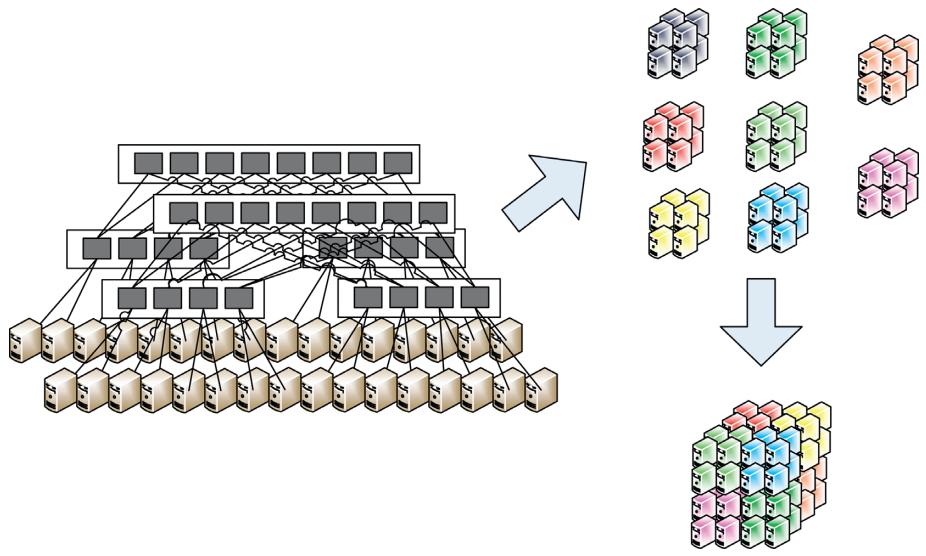

Figure 9: An example of stencil communication dynamic performance model optimization.

for the test. The test operation program is a program that has conducted several rounds of 3D mode stencil communication. For each batch of tests, several times will be performed to eliminate data with obvious abnormal performance results. The test job before optimization is issued with command and uses the default node allocation mode. When the optimized test job is submitted, specify the nodes and mapping mode selected by the optimization. To ensure fair competition, the two types of work will be submitted in different terminals at the same time. If the nodes selected by both parties have duplicates, the two test jobs are submitted in turn. As the message length decreases, the number of communication iterations increases, making the observation time easy to measure.

From the test results shown in Table 2, it can be seen that significant optimization effect of communication performance can be achieved after node optimization selection and process-computational kernel mapping optimization based on dynamic topology. The values in the table represent the time in seconds needed to complete a round of communication.

```
The initial node block set is empty;
Smaller node blocks are constructed from all candidate nodes and added into the
node block set.
While (Node block size < number of nodes required){
    In the node block set, larger node blocks are constructed from smaller node
blocks.
    Add the larger node blocks to the node block set;
    Clearing smaller node blocks in the node block set;
}
```

Figure 10: Algorithm to construct stencil communication node block.

Table 2: Test results of stencil communication optimization of the Sunway TaihuLight system.

Number of nodes	2048			4096		
Message size	Before optimization	After optimization	Speed-up ratio	Before optimization	After optimization	Speed-up ratio
1 k	9.234411	9.001827	1.025837	14.381728	13.382712	1.07465
2 k	8.231239	8.138279	1.011423	14.381979	13.391783	1.073941
4 k	8.123486	8.01416	1.013642	16.737168	15.773643	1.061084
8 k	8.123227	8.108371	1.001832	14.391076	13.847271	1.039272
1 6k	9.549201	9.132407	1.045639	14.837273	13.838275	1.072191
32 k	8.888959	8.566869	1.037597	16.948927	14.989327	1.130733
64 k	9.611331	9.086707	1.057735	14.441525	13.426739	1.075579
128 k	8.888959	8.566869	1.037597	14.611412	13.532244	1.079748
256 k	8.785386	8.33516	1.054015	16.599016	14.64771	1.133216
512 k	8.778187	8.078187	1.086653	14.517939	13.429475	1.08105
1 m	9.338786	8.463902	1.103367	14.325862	13.001551	1.101858
2 m	9.311491	8.435159	1.10389	17.006372	14.313058	1.188172
4 m	9.308542	8.410059	1.106834	16.290833	14.371748	1.133532
8 m	10.218814	8.56815	1.192651	15.568335	14.043371	1.10859

7. Application Optimization Examples

At present, several applications including molecular dynamics simulation and turbulence simulation have been optimized using this technique. The performance of these applications in the Sunway TaihuLight system was tested.

Molecular dynamics simulation is a computer simulation method, usually using computer software, according to the basic principles of Newtonian mechanics, molecular movement as the main object of simulation, and all the motion of the particles in the research system with the evolution of the time. Molecular dynamics simulation can not only get the molecular motion but also observe the microscopic details of atomic motion. The application mode of communication is stencil mode. For molecular dynamics simulation application, the single-step communication performance before and after optimization is compared as shown in Table 3. The values in the table represent the time in seconds needed to complete a round of communication.

A common numerical method for turbulence simulation is to directly solve the NS equation with periodic boundary conditions, namely, the Fast Fourier Transform method, more accurately known as the potential pseudo-spectral method, which has the advantage of being able to deal with periodic boundary conditions and has high order accuracy. A typical turbulence program requires more than 12 arrays to represent different physical components. The communication mode of this application is all-to-all communication mode. For turbulence simulation application, the single-step communication performance before and after optimization is compared as shown in Table 4. The values in the table represent the time in seconds needed to complete a round of communication.

It can be seen from the above data that this technology can effectively optimize the communication performance of each application. Especially for molecular dynamics simulation applications, the communication performance was improved about twice under the size of the Sunway TaihuLight system half machine and full machine, as shown in Figure 11. The time in the figure represents the time in seconds needed for one round of communication.

This technology also improves the scalability of application communication performance. As shown in Figure 12, the horizontal axis is the number of nodes used in the application, and the vertical axis is the single-step communication time. The time in the figure represents the time in seconds needed for one round of communication. It can be seen that the single-step communication time after optimization has better scalability than before optimization.

Table 3: Performance results for molecular dynamics simulation applications of the Sunway TaihuLight system.

Number of nodes	Before optimization	After optimization	Speed-up ratio
512	0.68	0.668	1.017964071
1728	0.865	0.785	1.101910828
4096	1.246	1.135	1.097797357
8000	1.386	1.186	1.168634064
13824	1.574	1.399	1.12508935
21952	3.292	1.532	2.148825065
32768	3.78	2.321	1.628608358

Table 4: Performance results for turbulence simulation applications of the Sunway TaihuLight system.

Number of nodes	Before optimization	After optimization	Speed-up ratio
1024	1.31	1.27	1.031496
2048	2.65	2.43	1.090535
4096	7.4	6.46	1.144802
8192	26.9	24.7	1.089069
16384	97.5	91.1	1.070252

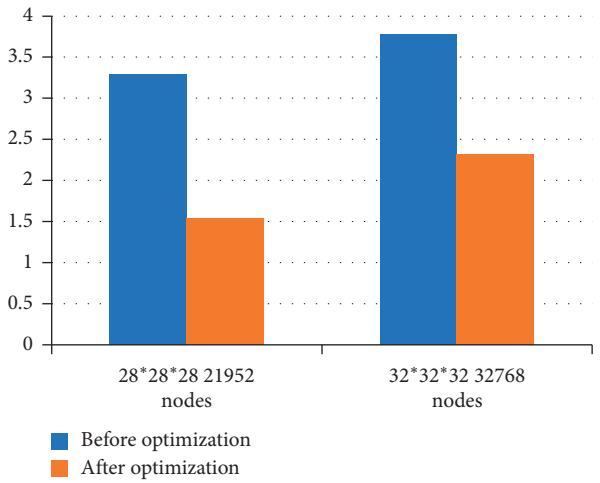

Figure 11: Comparison of communication optimization performance in large-scale applications.

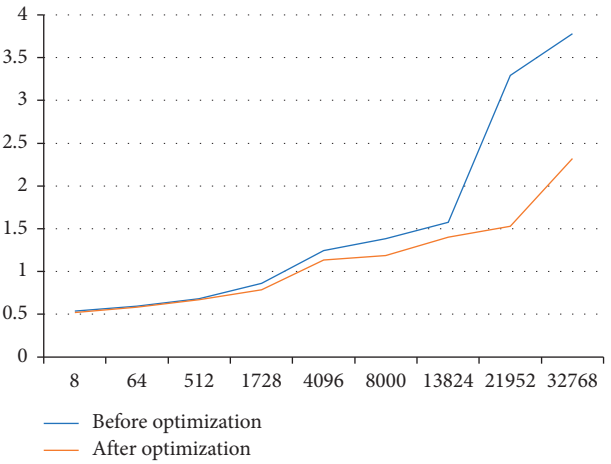

Figure 12: Scalability of communication performance for large-scale applications.

8. Conclusions

In this paper, the communication performance optimization technology based on topological structure is presented. The communication characteristics of different types of applications are analyzed, and the implementation of dynamic topology detection mechanism of data communication is studied. According to the dual factors of network physical structure and network dynamic performance model, complex set communication is optimized by improving process-computation kernel mapping. Several applications, including molecular dynamics simulation and turbulence simulation, have been optimized and tested. The average performance has been improved obviously. It can be expected that, for other large-scale applications, this optimization method can also be used to obtain significant improvement in communication performance.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was supported by the National Key R&D Program of China under Grant no. 2017YFB0202001 and the National Natural Science Foundation of China under Grant nos. 61672208 and 61432018.

References

[1] A. Faraj, S. Kumar, B. Smith, A. Mamidala, and J. Gunnels, "MPI collective communications on the blue gene/*P* supercomputer: algorithms and optimizations," in *Proceedings of*

the 17th IEEE Symposium on High Performance Interconnects, HOTI 2009, pp. 63–72, IEEE, New York, USA, August 2009.

[2] N. Jain and Y. Sabharwal, "Optimal bucket algorithms for large MPI collectives on torus interconnects," in *Proceedings of the 24th ACM International Conference on Supercomputing*, pp. 27–36, ACM, Tsukuba, Japan, June 2010.

[3] P. Sack and W. Gropp, "Faster topology-aware collective algorithms through non-minimal communication," in *Proceedings of the ACM SIGPLAN Notices*, vol. 47, no. 8, pp. 45–54, ACM, New Orleans, LA, USA, September 2012.

[4] G. Almási, P. Heidelberger, C. J. Archer et al., "Optimization of MPI collective communication on BlueGene/L systems," in *Proceedings of the 19th Annual International Conference on Supercomputing*, pp. 253–262, ACM, Cambridge, MA, USA, June 2005.

[5] T. Adachi, N. Shida, K. Miura et al., "The design of ultra scalable MPI collective communication on the K computer," *Computer Science-Research and Development*, vol. 28, no. 2-3, pp. 147–155, 2013.

[6] A. Faraj and X. Yuan, "Automatic generation and tuning of MPI collective communication routines," in *Proceedings of the 19th Annual International Conference on Supercomputing*, pp. 393–402, ACM, Cambridge, MA USA, June 2005.

[7] A. Faraj and X. Yuan, "Message scheduling for all-to-all personalized communication on ethernet switched clusters," in *Proceedingsof the 19th IEEE International Parallel and Distributed Processing Symposium*, p. 85a, April 2005.

[8] C. Nicolai, B. Jacob, P. Gualtieri et al., "Inertial particles in homogeneous shear turbulence: experiments and direct numerical simulation," *Flow Turbulence & Combustion*, vol. 92, no. 1-2, pp. 65–82, 2014.

[9] S. Paul and W. Gropp, "Collective algorithms for multiported torus networks," *ACM Transactions on Parallel Computing*, vol. 1, no. 2, 2015.

[10] A. Faraj, X. Yuan, and D. Lowenthal, "STAR-MPI: self tuned adaptive routines for MPI collective operations," in *Proceedings of the 20th Annual International Conference on Supercomputing*, pp. 199–208, ACM, Cairns, Australia, June 2006.

[11] T. Nanri and M. Kurokawa, "Effect of dynamic algorithm selection of all to all communication on environments with unstable network speed," in *Proceedings of the International Conference on High Performance Computing and Simulation (HPCS)*, pp. 693–698, IEEE, Istanbul, Turkey, July 2011.

[12] S. S. Vadhiyar, G. E. Fagg, and D. Jack, "Automatically tuned collective communications," in *Proceedings of the 2000 ACM/IEEE Conference on Supercomputing*, p. 3, November 2000.

[13] H. Subramoni, K. Krishna, J. Jose et al., "Designing topology-aware communication schedules for Alltoall operations in large InfiniBand clusters," in *Proceedings of the International 43rd Conference on,Parallel Processing (ICPP)*, pp. 231–240, IEEE, Minneapolis, USA, September 2014.

[14] H. N. Mamadou, T. Nanri, and K. Murakami, "A robust dynamic optimization for MPI all to all operation," in *Proceedings of the IEEE International Symposium on Parallel & Distributed Processing*, pp. 1–15, IEEE, Rome, Italy, Europe, May 2009.

[15] P. Patarasuk and X. Yuan, "Bandwidth efficient all-reduce operation on tree topologies," in *Proceedings of the IEEE International Parallel and Distributed Processing Symposium*, pp. 1–8, IEEE, Silicon Valley, California, USA, March 2007.

[16] K. Kandalla, H. Subramoni, A. Vishnu, and D. K. Panda, "Designing topology-aware collective communication algorithms for large scale infiniband clusters: case studies with scatter and gather," in *Proceedings of the IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW)*, pp. 1–8, IEEE, Atlanta, Georgia, April 2010.

[17] T. Ma, T. Herault, B. George, and J. J. Dongarra, "Process distance-aware adaptive MPI collective communications," in *Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER), 2011*, pp. 196–204, IEEE, Austin, Texas, US, September 2011.

[18] E. Gallardo, V. Jerome, L. Fialho, P. Teller, and J. Browne, "A minimal overhead tool for MPI library performance tuning," in *Proceedings of the 22nd European MPI Users' Group Meeting*, pp. 21–23, Bordeaux, France, September 2015.

[19] A. Bhatele, A. R. Titus, J. J. Thiagarajan et al., "Identifying the Culprits behind Network Congestion," in *Proceedings of the IEEE International On Parallel and Distributed Processing Symposium (IPDPS)*, pp. 113–122, Hyderabad, India, May 2015.

[20] E. Chan, G. Robert van de, W. Gropp, and R. Thakur, "Collective communication on architectures that support simultaneous communication over multiple links," in *Proceedings of the Eleventh ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pp. 2–11, ACM, New York, USA, March 2006.

[21] A. Faraj, P. Patarasuk, and X. Yuan, "Bandwidth efficient all-to-all broadcast on switched clusters," *International Journal of Parallel Programming*, vol. 36, no. 4, pp. 426–453, 2008.

[22] P. Zhang and Y. Deng, "Design and analysis of pipelined broadcast algorithms for the all-port interlaced bypass torus networks." *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2245–2253, 2012.

[23] S. Song and J. K. Hollingsworth, "Designing and auto-tuning parallel 3-D FFT for computation-communication overlap," in *Proceedings of the 19th ACM Sigplan Symposium on Principles & Practice of Parallel Programming*, February 2014.

[24] A. R. Mamidala, S. Narravula, A. Vishnu, G. Santhanaraman, and D. K. Panda, "On using connection-oriented vs. connection-less transport for performance and scalability of collective and one-sided operations," in *Proceedings of the 12th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pp. 46–54, ACM, San Jose, California, USA, March 2007.

[25] M. J. Koop, S. Sur, Qi Gao, and D. K. Panda, "High performance MPI design using unreliable datagram for ultra-scale InfiniBand clusters," in *Proceedings of the 21st Annual International Conference on Supercomputing*, pp. 180–189, ACM, Reno, Nevada, 2007.

[26] Y. Qian and A. Ahmad, "Efficient RDMA-based multi-port collectives on multi-rail QsNet II clusters," in *Proceedings of the 20th International Conference on Parallel and Distributed Processing Symposium*, p. 273, April 2006.

[27] K. Hasanov, J.-N. Quintin, and A. Lastovetsky, "Hierarchical approach to optimization of parallel matrix multiplication on large-scale platforms," *The Journal of Supercomputing*, vol. 71, no. 11, pp. 3991–4014, 2015.

[28] N. Mistry, J. Ramsey, B. Wiley et al., "Throughput studies on an InfiniBand interconnect via all-to-all communications," in *Proceedings of the Symposium on High Performance Computing (HPC'15)*, pp. 93–99, Society for Computer Simulation International, Alexandria, VA, USA, April 2015.

[29] C. Karlsson, T. Davies, C. Ding, H. Liu, and Z. Chen, "Optimizing process-to-core mappings for two dimensional broadcast/reduce on multicore architectures," in *Proceedings*

of the International Conference onParallel Processing (ICPP), 2011, pp. 404–413, IEEE, Taipei, Taiwan, September 2011.

[30] P. Balaji, R. Gupta, A. Vishnu, and P. Beckman, "Mapping communication layouts to network hardware characteristics on massive-scale blue gene systems," *Computer Science-Research and Development*, vol. 26, no. 3-4, pp. 247–256, 2011.

[31] A. Mittal, N. Jain, T. George, Y. Sabharwal, and S. Kumar, "Collective algorithms for sub-communicators," in *Proceedings of the 26th ACM International Conference on Supercomputing*, pp. 225–234, ACM, Venice, Italy, Europe, June 2012.

[32] A. Bhatele, G. Todd, S. H. Langer et al., "Mapping applications with collectives over sub-communicators on torus networks," in *Proceedings of the International Conference for, High Performance Computing, Networking, Storage and Analysis (SC), 2012*, pp. 1–11, IEEE, Salt Lake, Utah, USA, 2012.

[33] C. Karlsson, T. Davies, and Z. Chen, "Optimizing process-to-core mappings for application level multi-dimensional MPI communications," in *Proceedings of the IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 486–494, IEEE, Beijing, China, September 2012.

[34] E. Zahavi, "Fat-trees routing and node ordering providing contention free traffic for MPI global collectives," in *Proceedings of the 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW)*, pp. 761–770, IEEE, Ottawa, Canada, May 2011.