A Process-to-Processor Mapping of HHC Topology in MPI

Min-Hwan Ok'

Dept. of Computer Science, Korea University,
Seongbuk-gu, Seoul, Korea
mhok@krri.re.kr

Abstract. In the case network topology is none of the representative ones including mesh and hypercube, the process topology constructed by MPI is not well suited to that network topology. In this work, a transformation for topology-aware mapping is proposed since MPI does not provide a proper construction of process topology adequate to other new network topology. We show the procedure for a transformation case with a specific topology called HHC.

Keywords: Physical Topology, Virtual Topology, Topology-aware Mapping, Collective Communication.

1 Introduction

Multicomputer is a unified system of multiple computing elements, and the computing elements are mostly computing nodes with hardware independent from each other. Interconnection network unifies the computing nodes in hardware, and delivers messages among the computing nodes. As multiple computing nodes send and receive their messages, there could be contentions on the interconnection network. The message delivery is delayed in the contention and this could prolong the executions related with the message. An evaluative study[1] has presented the effect on message latencies by the contention, along the number of hops. As the messages travel more hops, links are shared by more messages increasing the contention and decreasing the available effective bandwidth.

The contention would be reduced by schematic formations of the task assignment onto processors. According to the proximity of the communicating task pairs, the contention is reduced as the hop count decreases between them. Topology-aware mapping was exploited to reduce the contention on mesh topology[2]. The mapping scheme is adequate to mesh and mesh topology is one of representative topologies for interconnection networks. MPI, *Message Passing Interface,* is the standard for

[1] Please note that the LNCS Editorial assumes that all authors have used the western naming convention, with given names preceding surnames. This determines the structure of the names in the running heads and the author index.

message passing in multicomputer, which has been evolved for two decades. Many works have proposed mapping schemes for the process assignment to processors. However almost of them considered little of collective communications which are also quite sensitive to process mapping as indicated in [3]. Communication Topology Graph (CTG) generated from composition of point-to-point communications and collective ones is overlaid on Network Topology Graph (NTG) describing bandwidth and latency between processors (cores). Processes of CTG are mapped onto processors of NTG with an optimization technique. The approach outperformed previous approaches by decomposing a given collective operation to a collection of point-to-point communications and integrating them with existing mapping schemes, although what network topology is not stated in the work.

Collective communications have the higher probability of contention than point-to-point communications, whereas point-to-point ones take place much frequently than collective ones. The collective communication is asymmetric communication with many or all of the other processes. Asymmetric communication naturally incurs contention and the numbers of hop counts are larger than those of point-to-point communications, resulting in much higher probability of contention since many or all the processes take part in the collective communications. However orthogonal topologies including mesh and hypercube are not appropriate for asymmetric communication and collective communication although they are mostly representative topologies of interconnection networks in multicomputers. In the case network topology is none of the representative ones, the process topology constructed by MPI is not well suited to that network topology. In this work, a transformation for topology-aware mapping is proposed since MPI does not provide a proper construction of process topology adequate to other new network topology. We show the procedure for a transformation case with a specific topology called *HHC*. The next section describes the benefit of the topology both in point-to-point and collective communications. A transformation scheme is proposed later, what is founded on a translation of node IDs into process ranks for process-to-processor mapping in MPI. Each node is considered to have one processor of a single core and only one process would be allotted to each node in this work.

2 A Hierarchical Topology of Cubes with Center Nodes

Orthogonal topologies including mesh and hypercube are appropriate for symmetric communication in form of one-to-many. Based on mesh topology, a hierarchical topology of cubes with center nodes has been proposed[4], and this topology is appropriate for both symmetric communication and asymmetric one. The role of the center node is the central operation with communications, i.e. collective communication in each cube. The topology was named as HHC and a 3-dimensional HHC is depicted in Fig. 1. This topology is a good fit for a variety of bulk-synchronous parallel applications, especially MPI-based parallel ones, in which a master creates a group of slave processes across a set of nodes. It would also fit well for a communication-aware load-balancing[5] that attempts to balance the communication load by allocating the tasks to a group of nodes with a lower utilization of network resources.

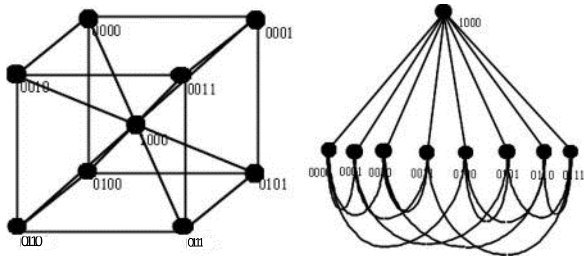

Fig. 1. 3-dimensional HCC *(left)* and an equivalent graph *(right)*

2.1 Characteristics of HHC

As the name, HHC is a hierarchical topology and advantageous in group communications, one-to-many and many-to-many. By placing processes with their communication localities, a process group is placed around one center node of a cube. Process groups are placed from outer cubes to inner cubes. This is analogous to deciding whether a team or a group of teams with respect to the whole work size when deciding on the size of a workgroup of people. Thus the center node takes rather a role of central operation than a partial processing of the whole work. Groups of remained nodes could serve for other jobs that do not involve frequent group communications.

Compared to hypercube, the diameter is shorter(ri *n1)* but the average distance and the total distance are longer in HHC[4]. A message destined for out of the node's cube traverses via center nodes and makes the average and the total distances longer and thus the processes are necessarily placed with their communication localities. Process topologies should also show better communication performance by placing with communication locality. In contrast, HHC has the fixed number of node degrees. The maximum of a center node degree is 12 (8+4), and node degrees of every source and destination nodes are always 4 if they are not one of center nodes. The system with interconnection network of HHC topology is easily expanded by this fixed node degrees.

2.2 Routing in HHC

The node ID comprises a concatenation of 4-bit nibble addresses. HHC topology could be decomposed to one tree and 3-cubes recursively expanding to lower level of the tree as shown in Fig.2. A nibble of '1000' is the specific address of a center node, and no other *nibble* has a higher value than '1000'. Along the levels in the tree, the number of concatenated nibbles increases as the depth of the tree increases.

Intra-cube routing is for communication within a 3-cube. The LSB in the lowest nibble of destination node ID indicates the first direction and the MSB in the lowest nibble of the destination indicates the last direction in the route. Inter-cube routing is for communication among 3-cubes, which is distinguished when the upper nibbles of the destination node ID differs from those of the sender (or current) node ID. The first direction is to the center node of the current level, and then the higher level becomes current level and the upper nibble becomes this nibble. Intra-cube routing is applied against the new this nibble. Once this nibble corresponds in current level, the lower level becomes current level and the lower nibble becomes this nibble. Intra-cube routing is applied against the new this nibble, and the direction is determined. Inter-cube routing involves intra-cube routing recursively.

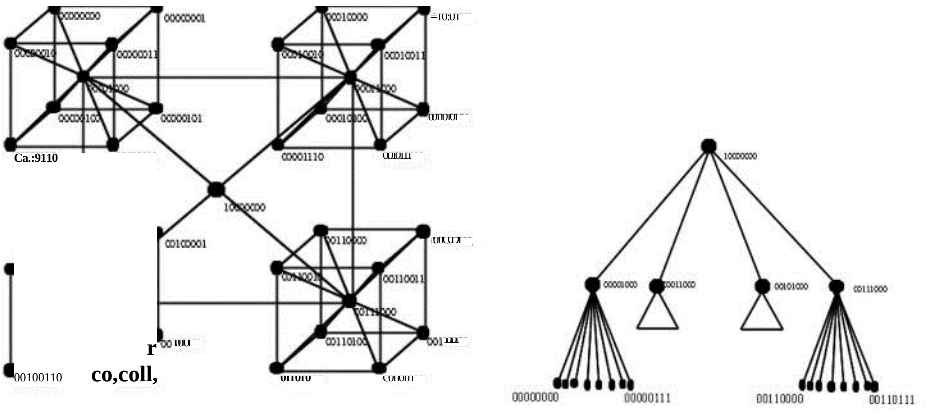

Fig. 2. 5-dimensional HHC *(left)* and an equivalent graph *(right)* with links to center nodes only

3 Process-to-Processor Mapping for Message Routing

MPI 2.2 supports two kinds of topology construction, *Cartesian* (or grid) and *Graph* (or tree). The process ranks are assigned in accordance with the virtual topology created on the physical topology. For the case of a Cartesian, those ranks are assigned in the form of node IDs in a 3-D mesh. However this assignment of process ranks is far different from the form of the node IDs in HHC, and this is the same situation for the case of a Graph. The process ranks assigned automatically by MPI should be translated into proper node IDs in HHC for message routing. Further a transformation of virtual topologies created would necessarily precede for translations of node IDs into process ranks with communicators into on the physical topology.

3.1 Topology Construction with MPI Functions

HHC is a composite topology of a tree with 3-cubes. 3-dimensional HHC of 9 nodes is constructed by the following procedure.

1. Creating a process group for a tree. A communicator of 9 processes is created using MPI_Comm_Create, at the center node.
2. Creating a tree with the root as the center node. A communicator for a tree is created Using MPI Graph_ Create with its root process of the center node.
3. Creating a process group for a 3-cube. A communicator of 8 processes is created using MPI Comm Create, excluding the process of the center node from the former group.
4. Creating a 3-cube of leaves of the tree. A communicator for a 3-cube is created using MPI Cart Create, excluding the process of the center node.

For 3-dimensional HHC, the procedure would be sufficient however, as the dimension of HCC increases process ranks are far different from the node IDs in HHC.

3.2 Topology Construction in a Higher Level and Translating Node IDs into Process Ranks with Communicators

In the previous subsection, 3-dimensional HHC, *3-HHC* in short, was simply constructed. For the construction of a higher-dimensional HHC, however, the center node of one 3-HHC becomes one leaf node in the higher level. 4-dimensional HCC requires two 3-HHCs and 5-dimensional HHC requires four 3-HHCs, and so forth. The procedure of the previous subsection is conducted by two phases of a tree creation and 3-cubes creations. Thus to construct a HHC of the higher dimension, at the topmost center node, a tree is to be created with all the nodes of the HHC. Since a predetermined rank is not assigned to a specific node in MPI, the assigned ranks are investigated using MPI_Group_rank, from each node. A tree is created based on those ranks. Then, from the topmost center node, 3-cubes are created by post-order traversal on the tree. During these creations, the communicator of each 3-cube is stored in a proprietary data structure (array). For example, in Fig. 2, the virtual topology of 5-demensional HCC has four 3-HHCs and one 3-HHC with the topmost center and (4+1) communicators of 3-cubes. An array of 8x9 {rank, communicator} pairs stores the translation data for four 3-cubes and that of 5x1 {ranks, communicator} pairs stores one for the 3-cube with topmost center node in the proprietary data structure. LSBs of nibbles in the array index distinguish its level of the node in the tree and whether the node is one center node. In the case the node has a couple of communicators, the opposite node determines which communicator.

3.3 Message Passing on the Combined Virtual Topology

In the intra-cube routing, a {rank, communicator} pairs are sufficient for the sender and receiver. Multiple {rank, communicator} pairs suffice collective communications in the intra-cube routing. In the inter-cube routing, one point-to-point communication could bear two point-to-point communications, one on a 3-cube and the other on the tree, or vice versa. When the communicator should be changed on the inter-cube route, an additional point-to-point communication should be called with the other communicator. The collective communication on the intra-cube routes is confined to that on the tree; currently collective communications that require the other communicator is not available.

4 Conclusion

Some recent works on message passing have focused on the contention on the interconnection network demonstrating the effect of process-to-processor mapping. The topology-aware mapping outperforms among the mapping schemes. However the process topology constructed by MPI is not well suited to that network topology in the case the topology is new one other than the representative topologies. A procedure to construct the virtual topology corresponds to the physical topology is presented for the case of HHC topology. Further a composite representation of array index: {rank, communicator} is used for addressing in the combined topology using Graph and Cartesian ones supported by MPI. The communicators would be more useful if they could have numeric names and could be placed in parts of a node ID, for the sake of utilizing the intrinsic routing of a specific network topology.

References

1. Bhatele, A., Laxmikant, V.: An Evaluative Study on the Effect of Contention on Message Latencies in Large Supercomputers. In: IEEE International Symposium on Parallel&Distributed Processing, pp. 1--8. IEEE CS Press, Washington, DC. (2009)
2. Bhatele, A., Laxmikant, V.: Benefits of topology aware mapping for interconnects. Para. Proc. Let. 18(4), 549-566. World Scientific Publishing (2008)
3. Zhang, J., Zhai, J., Chen. W., Zheng, W.: Process Mapping for MPI Collective Communications. In: H. Sips, D. Epema, and H.-X. Lin (Eds.): Euro-Par 2009. LNCS, vol. 5704, pp. 81-92. Springer, Heidelberg (2009)
4. Kim, H-K., Ok, M-H., Yang, J-A., Kim, Y-H., Kim, D-H.: A Design of Experimental System using One Hierarchical Interconnection Network - Hierarchical Body-Centered Cube Testbed. In: High-Performance Computing on the Information Superhighway, HPC Asia '97, pp. 720--723. IEEE CS Press (1997)
5. Qin, X., Jiang, H., Manzanares, A., Ruan, X., Yin, S.: Communication-Aware Load Balancing for Parallel Applications on Clusters. Trans. Comp. 59(1), 42-52. IEEE CS Press (2010)