Network-Aware Optimization of MPDATA on Homogeneous Multi-core Clusters with Heterogeneous Network

Tania Malik[1](✉), Lukasz Szustak[2](✉), Roman Wyrzykowski[2], and Alexey Lastovetsky[1]

[1] School of Computer Science, University College Dublin, Belfield, Dublin 4, Ireland
tania.malik@ucdconnect.ie, Alexey.Lastovetsky@ucd.ie
[2] Czestochowa University of Technology,
Dabrowskiego 69, 42-201 Czestochowa, Poland
{lszustak,roman}@icis.pcz.pl

Abstract. The communication layer of modern HPC platforms is getting increasingly heterogeneous and hierarchical. As a result, even on platforms with homogeneous processors, the communication cost of many parallel applications will significantly vary depending on the mapping of their processes to the processors of the platform. The optimal mapping, minimizing the communication cost of the application, will strongly depend on the network structure and performance as well as the logical communication flow of the application. In our previous work, we proposed a general approach and two approximate heuristic algorithms aimed at minimization of the communication cost of data parallel applications which have two-dimensional symmetric communication pattern on heterogeneous hierarchical networks, and tested these algorithms in the context of the parallel matrix multiplication application. In this paper, we develop a new algorithm that is built on top of one of these heuristic approaches in the context of a real-life application, MPDATA, which is one of the major parts of the EULAG geophysical model. We carefully study the communication flow of MPDATA and discover that even under the assumption of a perfectly homogeneous communication network, the logical communication links of this application will have different bandwidths, which makes the optimization of its communication cost particularly challenging. We propose a new algorithm that is based on cost functions of one of our general heuristic algorithms and apply it to optimization of the communication cost of MPDATA, which has asymmetric heterogeneous communication pattern. We also present experimental results demonstrating performance gains due to this optimization.

A. Lastovetsky—This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number 14/IA/2474. This research was conducted with the financial support of NCN under grants no. UMO-2015/17/D/ST6/04059. This work is partially supported by EU under the COST Program Action IC1305: Network for Sustainable Ultrascale Computing (NESUS). Experiments were carried out on Grid'5000 developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr).

J. Carretero et al. (Eds.): ICA3PP 2016 Workshops, LNCS 10049, pp. 30–42, 2016.
DOI: 10.1007/978-3-319-49956-7_3

Keywords: Heterogeneous computing · Communication optimization · Topology-aware optimization · Performance-aware optimization · Data partitioning · MPDATA

1 Introduction

Modern high performance computing (HPC) platforms are becoming increasingly complex, heterogeneous and hierarchical. Heterogeneity appears not only in the computing devices but also in networks. Even with homogeneous processors, efficient execution of data-parallel applications is a big challenge due to ever increasing heterogeneity and complexity of the underlying networks. Optimization of data-parallel applications on such platforms is typically achieved by minimizing the cost of data movement between the processors. In this work, we consider the network heterogeneity rather than the processor heterogeneity. Thus, the target platform comprises homogeneous processors connected with a heterogeneous network. Assuming that the workload is balanced among the processors, we propose a mapping approach that optimizes the overall communication performance of a parallel computational fluid dynamics (CFD) application on such a platform.

We target HPC platforms with heterogeneous networks having two levels of hierarchy, such as interconnected compute nodes and clusters. These networks are very common in the computing world. Popular examples include Grid and Cloud infrastructures. Even supercomputers with thousand of nodes are also examples of heterogeneous network where the communication cost is different on different hierarchical levels e.g. intra-node vs inter-node communication. In data-intensive parallel applications, data transfer between different hierarchical levels is a primary cause of the execution delay. Application scalability has been highly hampered from this data transfer communication overhead. Communication cost can significantly vary depending on mapping of the application processes to the processors of the platform, and the optimal solution minimizing the communication cost strongly depends both on the structure and performance characteristics of the network and on the logical communication flow of the application. In our previous work [1], we proposed a general approach and two heuristic algorithms aimed at minimization of the communication cost of data parallel applications which have symmetrical two-dimensional communication pattern on heterogeneous hierarchical networks, and tested these algorithms in the context of the parallel matrix multiplication application. In this work, we propose a new algorithm that is built on top of cost functions and heuristics of one of our previously proposed algorithms. This algorithm reduces overall message hops and increases data throughput for a wider range of applications, and we apply it to a real-life CFD application.

The CFD application we consider in this work is the multidimensional positive definite advection transport algorithm (MPDATA), which is one of the major parts of the dynamic core of the EULAG geophysical model [2,3]. This geophysical model can be used for simulating thermo-fluid flows across a wide range of scales and physical scenarios, including the numerical weather prediction. The MPDATA belongs to the group of non-oscillatory forward-in-time algorithms, and performs a sequence of stencil computations. The original version of MPDATA has been implemented in FORTRAN 77 and parallelized using MPI library. In our previous work [4] we proposed to rewrite the MPDATA code and replace conventional HPC systems with modern homogeneous and heterogeneous multi- and many-core based platforms. In particular, we have successfully developed a new version of MPDATA that allowed us to much better exploit the available computational features of novel processors and Intel Xeon Phi coprocessors.

However, the communication cost of MPDATA on modern HPC clusters has not been properly optimized. The current approach to mapping of the partitions of the MPDATA computational domain onto computing resources take into account neither the actual properties of the MPDATA communication flow nor the heterogeneity, hierarchy and performance of the communication network.

In this work, we first study and analyse the communication pattern of the MPDATA application. The analysis reveals that MPDATA is very sensitive to the choice of logical topology of processes as the cost per byte of horizontal communications is higher than that of vertical communications even for homogeneous communication networks. This property of MPDATA further complicates the task of partitioning of the MPDATA computational domain and mapping of the sub-domains to the processors in a way that minimizes the cost of communications between different levels of the network hierarchy. In general, finding the optimal arrangement of processors in a 2-D grid is an NP-complete combinatorial optimization problem [5] but it can be approximately solved by using heuristics [6]. For MPDATA, we propose a new heuristic algorithm based on one of our general heuristic approach presented in [1] and apply it to optimization of the communication cost of MPDATA. This algorithm is non-intrusive to the source code of the application and, compared to [1], is not application specific. Our previous algorithms deal with two-dimensional symmetric communication patterns that is why we tested these algorithms in the context of the parallel matrix multiplication application. With this new algorithm, any data-parallel application with two-dimensional homogeneous computational domain and asymmetric heterogeneous communication pattern can benefit. We demonstrate the accuracy and efficiency of the proposed solution using experiments on two-level hierarchical networks, namely, interconnected nodes (intra- and inter-node communication levels) and interconnected clusters (intra- and inter-cluster communication levels).

The rest of the paper is organized as follows. In Sect. 2, we introduce MPDATA and overview existing approaches to topology-aware optimization of communications for MPI applications. In Sect. 3, we analyze the communication pattern of MPDATA and describe its implementation in a cluster environment. In Sect. 4, we present the proposed approach to finding the optimal configuration of MPDATA. In Sect. 5, we give experimental results demonstrating performance gains due to this optimization.

2 Related Work

In this section, we describe MPDATA and its modifications over time. We also overview related work on topology-aware optimization of communications.

2.1 MPDATA

The MPDATA application is used to solve the advection equation on a moving grid according to the subsequent time steps [7]. This real-life application offers several advanced options that allow for modeling a wide range of complex geophysical flows. Depending on the type of modeled phenomena, this application can demand a high computing performance of HPC clusters. Therefore, the configurable code of MPDATA was developed and delivered over the years [2,7,8]. This code was implemented in FORTRAN 77 and parallelized using MPI library, however, without taking into account of the features of todays computing architectures.

The MPI parallelization of the MPDATA computations on x86-based clusters as a part of the EULAG model was thoroughly studied in [8], using tens of thousands of cores, or even more than 100 K cores in the case of IBM Blue Gene/Q. The parallelization strategy of this implementation is based on 3D domain decomposition, and executes computations according to the distributed memory model where each core is assigned to a single MPI_rank. This approach ignores the advantages of shared memory systems available in modern multicore platforms. Moreover, it also does not take into account the network-aware partitioning of communications across computing resources.

The MPDATA code has been recently re-written and optimized for execution on modern CPU and Intel co-processors based high-performance computing platforms. The new C++ implementation proposed in [4] allows for more efficient distribution of computational tasks on the available resources. It makes use of the (3+1)D decomposition strategy for the stencils computation, that transfers the data traffic from the main memory to cache hierarchy by proper reusing of the cache memory. Additionally, to improve the computational efficiency the algorithm groups the cores (threads) into independent work teams in order to reduce inter-cache communication overheads due to the communications between neighbouring threads/cores, and synchronizations.

2.2 Communication Optimization for Parallel Applications

Communication optimization is a very broad field that comprises a number of different approaches. The goal of all such optimization approaches is to reduce the overall runtime of the communication operations. Communication optimization on heterogeneous HPC platform is comprehensively covered in [9], where all the existing approaches were classified as performance or topology-aware. The increasing complexity of HPC platforms has made topology awareness a critical component of HPC application optimization. A number of topology-aware approaches have been proposed in [10,11]. The main idea behind the topology-aware optimizations is to reduce communication traffic and contention by taking into account the network topology so that most of the communication occurs between nearby processors. Whereas, in performance-aware optimizations, network properties are reconstructed with performance measurements by using communication benchmarks. This approach is used in the absence of topology information.

Topology information has been used in developing a number of topology-aware implementations of MPI collectives for optimal scheduling of messages on heterogeneous HPC platforms [12–15]. In [12], the optimization of the MPICH broadcast algorithm was proposed for efficient execution of broadcast on interconnected clusters. Interconnected clusters are presented as two-level communication graphs, inter- and intra-cluster ones. Clusters communicate via selected nodes, coordinators, which form the inter-cluster communicator. Within a cluster all nodes communicate with the cluster coordinator, forming the intra-cluster communicator. This topology-aware implementation of broadcast algorithm aims to minimize the amount of data sent over the slow wide-area links and results in significant improvement.

Further performance improvement is realized in [13] and [16], where collective operations were optimized by adopting multilevel hierarchical heterogeneous networks and Grid. Pipelining and offloading techniques were used to overlap the inter- and intra-node communications in multi-core clusters. [11] has shown that topology-aware collectives can be used to reduce the communication cost on homogeneous supercomputers which have complex network topologies, like BlueGene and Cray.

Many existing MPI applications can be executed efficiently on hierarchical heterogeneous HPC platforms by using topology-aware collectives and does not require to modify application source code. However, it is applicable to collective operations only and does not affect the applications that are based on point-to-point exchanges. In this case, the communication cost can be reduced by placing frequently communicating tasks on physically nearby processors. This closeness is application-specific and depends on the logical communication flow of the application.

In [10,17], the problem of topology-aware optimization of point-to-point communications is solved by introducing a graph, which represents the logical communication flow of the application and is mapped onto the network topology. [17] applied this approach to the mesh and graph virtual MPI topologies and SMP clusters. In [10], it was applied to the mesh topology on BlueGene/L.

3 MPDATA on Clusters

One of the common methods for exploiting the multicore clusters is to employ the hybrid programming model, that allows for efficient usage of the distributed and shared memory hierarchies of these systems. This implies to combine different programming paradigms, such as MPI and OpenMP. Such a mixture is successfully utilized for the MPDATA computation, where a single MPI_rank is

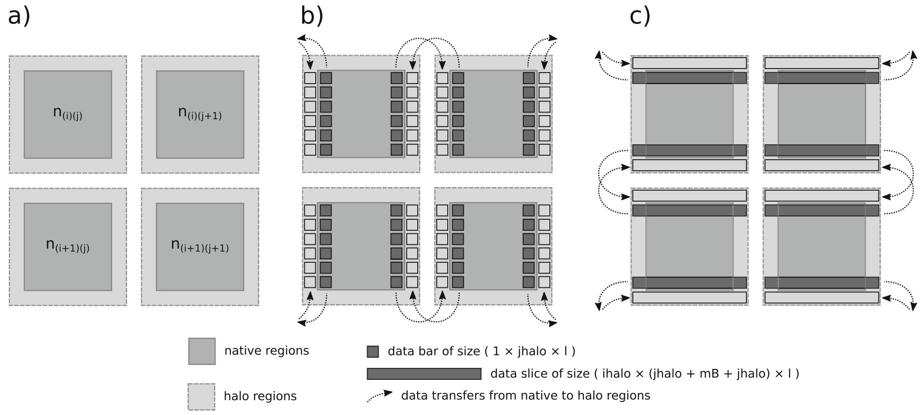
a)
b)
c)
n(i)(j)
n(i)(j+1)
n(i+1)(j)
n(i+1)(j+1)
native regions
halo regions
data bar of size (1 × jhalo × l)
data slice of size (ihalo × (jhalo + mB + jhalo) × l)
data transfers from native to halo regions

Fig. 1. Data flow between nodes for the MPDATA application: (a) 2D domain decomposition between computing nodes: $n_{ij}, n_{ij+1}, ...$, (b) the communication pattern for the horizontal direction, (c) the communication pattern for the vertical direction

assigned to every multicore node while OpenMP threads are employed to utilize the multicore computational resources.

The 3D $n \times m \times l$ MPDATA domain is firstly partitioned in two dimensions n and m into equal sub-domains that are further one-to-one mapped to adequate nodes of the homogeneous clusters. Every sub-domain of size $nB \times mB \times l$ is decomposed according to the (3+1)D decomposition proposed in our previous works [4]. This strategy contributes to ease the main-memory and communications bounds, that characterize MPDATA, and to better exploit modern computational resources such as cores and vector units.

Since the (3+1)D strategy allows for independent calculation of every subdomain for a single time step, the inter-node communications and synchronization points have to take place only between subsequent time steps in order to exchange the required partial outcomes. The exchanged data corresponds to the halo regions determined by data dependencies of MPDATA computations. These regions take place on the border of the MPDATA domain partitioning. As a result, the data traffic is generated only between nodes that are mapped onto adjacent sub-domains in both directions: vertical and horizontal. Figure 1 illustrates the data flow between nodes of MPDATA application.

After every time step each node has to send/receive in horizontal direction the adequate halo regions to/from adjacent nodes placed on the left and right sides (Fig. 1b). Since the necessary halo regions for this direction are periodically placed in the main memory, each node exchanges nB data bar of size $1 \times jhalo \times l$ to the left node, and to the right one. Then, the same node is responsible for sending/receiving in vertical direction the adequate halo regions to/from adjacent nodes placed on the top and bottom sides (Fig. 1c). Transferred data in this communication path is placed in the contiguous memory areas, thus this node moves the data slices of size $ihalo \times (jhalo + mB + jhalo) \times l$ to/from the top and bottom nodes.

4 Communication-Optimal Mapping Arrangement for MPDATA

In this section, we first propose an extension of the network-bandwidth-based cost function [1] to accurately measure the communication cost of the MPDATA application. Then we formulate the heuristic solution that efficiently constructs a near-optimal arrangement for MPDATA based on the extended cost function by using information about network topology and the application communication flow. This heuristic solution reduces the search space of sub-domain arrangements and finds the one that minimizes the communication cost of the MPDATA.

4.1 Cost Function Based on Asymmetric Bandwidth

In our previous work [1], we defined the cost function based on network bandwidth. The main idea was to estimate the communication cost accurately by using information about the network topology and the application communication flow. That cost function proved to work well with applications having symmetric communication patterns. However, MPDATA has asymmetric communication behavior, namely, even in the case of a homogeneous communication layer the effective bandwidth of horizontal communications is higher than that of the vertical ones. One of the reasons behind this phenomenon is that data communicated vertically is stored in a contiguous region of memory while the data communicated horizontally is not. As a result, this cost function fails to accurately characterize the communication cost of MPDATA. Therefore, we propose to extend this bandwidth-based cost function to account for applications with asymmetric communication patterns. The proposed extension characterizes the communication time, using the asymmetric bandwidths properties. We call it a cost function based on asymmetric bandwidth in the rest of the paper. The function takes into account two bandwidth values, one for horizontal communication and the other is for vertical one. The problem of finding the communication-optimal arrangement can be formulated as minimization of the sum of the horizontal and vertical communication costs.

Assuming that the data is equally partitioned among the processors, so that the size of each sub-domain is same, we define the asymmetric cost function for horizontal communication as follows:

$$cost_H = \sum_{i=1}^{r} \left(h \times \sum_{j=1}^{c} \frac{1}{b_H(Q_{ij}, Q_{i,(j+1)\%c})} \right), \tag{1}$$

where i iterates over the rows and j iterates over the partitioned sub-domains in each row. h is the height of a row (in bytes) that is same for each row because data is equally partitioned. Function $b_H(X, Y)$ returns the horizontal bandwidth (in bytes per second) between processors X and Y, and Q_{ij} designates the processor holding the j-th sub-domain in row i. Thus, this cost function estimates the communication time in seconds. The inner sum represents sending a part of the pivot column in a row. The outer sum represents the upper bound on the communication time required to send the whole pivot column to all rows. We use the upper bound because the bandwidth of some links may be divided between multiple communications corresponding to different rows.

We define the asymmetric cost function for vertical communication in a similar way:

$$cost_V = \sum_{j=1}^{c} \left(w \times \sum_{i=1}^{r} \frac{1}{b_V(Q_{ij}, Q_{i,(j+1)\%r})} \right), \tag{2}$$

Here j iterates over the columns, and i iterates over the partitioned sub-domains in each column. w is the width of a column (in bytes) that is same for each column because data is equally partitioned. Function $b_V(X, Y)$ returns the vertical bandwidth (in bytes per second) between processors X and Y.

The communication cost associated with arrangement A is represented by two values $(cost_H(A), cost_V(A))$. The problem of finding the communication-optimal arrangement can be formulated as minimization of their sum:

$$cost_H(A) + cost_V(A) \to \min . \tag{3}$$

4.2 Heuristic Based on Asymmetric Bandwidth Cost Function

The heuristic algorithm using the asymmetric bandwidth cost function for estimating the volume of communications is built on top of the bandwidth-based heuristic presented in [1]. It assumes that the target platform consists of p interconnected homogeneous processors. The processors are naturally partitioned into a number of groups based on their communication proximity, which reflects the two-level hierarchy of the communication layer. If processors x_0, x_1, y_0 and y_1 belong to the same group then $b_H(x_0, y_0) = b_H(x_1, y_1)$ and $b_V(x_0, y_0) = b_V(x_1, y_1)$.

The algorithm starts with any initial arrangement $P_1, P_2, \ldots, P_p$ of the processors such that processors from the same group will follow one other in this linear arrangement. Note, the orders naturally determined by application configuration files typically satisfy this assumption. Alternatively, a simple clustering algorithm guided by functions $b_H(x, y)$ and $b_V(x, y)$ can be applied to re-order the original arrangement if it does not satisfy this assumption.

The algorithm then repeatedly executes the following two steps. The first step finds the optimal two-dimensional arrangement of the processors, $m \times n$, which preserves their linear order as follows. For each factor pair $r \times c = p$, the processors are arranged column-wise and row-wise into r rows and c columns forming arrangement A. The cost of these arrangements are estimated as $cost(P_1, \ldots, P_p, r, c) = cost_H(A) + cost_V(A)$, and the optimal pair $m \times n$ is found as the one that minimizes this cost, $cost(P_1, \ldots, P_p, m, n) = \min_{r,c} cost(P_1, \ldots, P_p, r, c)$.

The second step applies the bandwidth-based algorithm from [1] slightly modified by the use of the asymmetric cost function to this 2D arrangement. This step may changes the linear order of the processors within the arrangement in order to reduce its communication cost while preserving the shape of the arrangement, $m \times n$. The reordering is guided by the 2D partitioning of the computational domain induced by the 2D processor arrangement and uses the fact that within each column of the domain, sub-domains held by processors from the same group will also make a group of adjacent sub-domains. In brief, we first try permutations of the groups in the first column and pick the one that minimizes the vertical communication cost for this column. Then, for each following column $k = 2, \ldots, n$, we try permutations of the groups in this column and pick the one that minimizes the sum of vertical and horizontal costs for first k columns. This guarantees that while improving communications horizontally, we will not deteriorate the vertical routes. Permutation of groups rather than individual processors in a column will significantly reduce the solution space that otherwise would be $p!$. Finally, we try all permutations of whole columns and pick the one that minimizes the sum of horizontal and vertical communication costs for the whole domain.

This step can change our original linear arrangement of the processors. If this is the case, we will feed the new arrangement to the first step of next iteration of our heuristic algorithm that will find the optimal $m \times n$ arrangement for this new order. Then, this 2D arrangement will be re-arranged by the second step of this iteration. This procedure continues until we find a fixed point of the transformation performed by one iteration of the algorithm.

The presented iterative algorithm does not require to run the application or any benchmarks to compare the communication cost of the application for different arrangements. Instead, it uses information about the network topology and the application communication flow. This heuristic is efficient for applications having 2D communication pattern on heterogeneous networks. Not only it reduces unnecessary exchanges between the sub-networks but also employs the fastest routes between them.

5 Experimental Results

In this section, we demonstrate that the communication performance of MPDATA can be significantly improved due to optimization proposed by the asymmetric bandwidth heuristic not only for heterogeneous but also for a perfectly homogeneous communication network.

We perform experiments on the Grid'5000 infrastructure, which is a large scale distributed platform. It consists of a number of clusters distributed between 10 sites in France and connected via the Renater network. Each site hosts several clusters of identical nodes. For our experiments, we choose two clusters, Grisou and Grimoire, from the Nancy site and the other two, Paravance and Parasilo, from the Rennes site. All clusters have identical Intel Xeon E5-2630 v3 processors with 8 cores per node. To demonstrate performance gains, we first perform two types of experiments on interconnected clusters. These interconnected clusters form a two-level hierarchy, with very heterogeneous inter-cluster links. Then, we conduct experiments on a single fully homogeneous cluster, with homogeneous processors and a homogeneous communication network. We have a priori information about the network topology and asymmetric bandwidths of MPDATA. We have tried ten different initial mappings as an input and our experiment shows that all of these mappings converges to the optimal solutions have same communication cost after applying asymmetric bandwidth heuristic. It has been noted that there is more than one optimal solutions exist. However, the communication cost and execution time of all optimal solutions are same. To make sure the experimental results are reliable, the application is repeatedly executed until the sample mean lies in the 95% confidence interval and a precision of 0.025 (2.5%) has been achieved and results follows the normal distribution. We also make sure the nodes are fully reserved and dedicated to our experiments.

5.1 Inter-cluster Experiments

In these experiments, we use four clusters with 12 nodes in total: Grimoire(3), Parasilo(4), Grisou(2), Paravance(3). We spawn one MPI process per node. Because logical communication links of MPDATA has different bandwidths, we have two bandwidth values for each link. Horizontal and vertical bandwidths are shown in Table 1. MPDATA is configured with problem size 512 × 512 × 64.

Table 1. Horizontal/Vertical bandwidths of communicating links(GB/sec)

	Grimoire	Parasilo	Grisou	Paravance
Grimoire	0.03963/0.48068	0.00007/0.00056	0.03889/0.49341	0.00007/0.00056
Parasilo	0.00007/0.00056	0.03876/0.48858	0.00007/0.00056	0.03732/0.45943
Grisou	0.03889/0.49341	0.00007/0.00056	0.03834/0.48916	0.00007/0.00056
Paravance	0.00007/0.00056	0.03732/0.45943	0.00007/0.00056	0.03920/0.46808

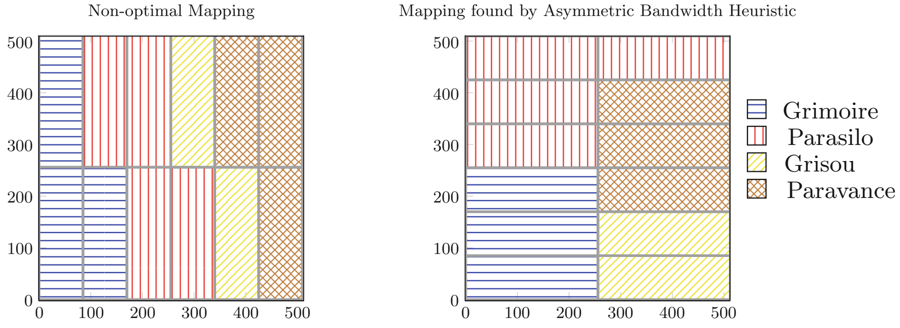

Fig. 2. One of the non-optimal mappings and the mapping returned by the asymmetric bandwidth heuristic for the heterogeneous platform.

Table 2. Inter-cluster experimental results

Nodes	Cost		Ratio	Exec. time (sec)		Ratio
	Non-optimal	Heuristic		Non-optimal	Heuristic	
12	22424946	2143978	10.46	994.02	154.20	6.44

Figure 2 shows one of the considered default initial mappings and the optimal mapping found by the asymmetric bandwidth heuristic. Table 2 shows the communication cost of these mappings, calculated using the cost function, and the measured total execution time of MPDATA. To find the optimal mapping, the asymmetric bandwidth heuristic took 1.130000e-03 s. The mapping found by the asymmetric bandwidth heuristic is more then 6 times faster then the non-optimal case mapping.

5.2 Intra-cluster Experiments

We also perform experiments on a homogeneous multi-core cluster to check the effect of asymmetric bandwidth of MPDATA on the communication performance with a perfectly homogeneous network. We use 12 nodes from the Grisou cluster. MPDATA is configured with problem size 512 × 512 × 64.

Figure 3 shows one of the non-optimal mappings and the mapping returned by the asymmetric bandwidth heuristic. Table 3 shows the calculated communication cost of both mappings and the measured total execution time of MPDATA. The mapping found by the asymmetric bandwidth heuristic is 3 times faster then the non-optimal mapping. Asymmetric bandwidth heuristic took 3.730000e-04 s to find this optimal mapping.

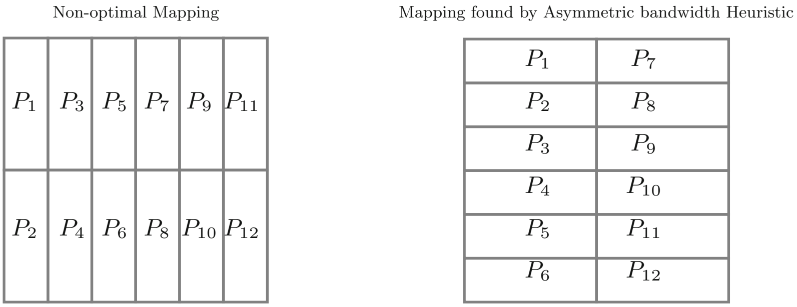
Non-optimal Mapping
P_1 P_3 P_5 P_7 P_9 P_{11}
P_2 P_4 P_6 P_8 P_{10} P_{12}
Mapping found by Asymmetric bandwidth Heuristic
P_1 P_7
P_2 P_8
P_3 P_9
P_4 P_{10}
P_5 P_{11}
P_6 P_{12}

Fig. 3. One of the non-optimal mappings and the mapping returned by the asymmetric bandwidth heuristic for the fully homogeneous platform.

Table 3. Intra-cluster experimental results

Nodes	Cost		Ratio	Exec. time (sec)		Ratio
	Non-optimal	Heuristic		Non-optimal	Heuristic	
12	65658	18535	3.5	3.86	1.32	3.0

6 Conclusions

In this paper, we have applied an approach aimed to minimize the communication cost of a parallel CFD application using information about the network topology/performance and application communication flow. We have also demonstrated that the proposed solution provides significant performance gains.

References

1. Malik, T., Rychkov, V., Lastovetsky, A.: Network-aware optimization of communications for parallel matrix multiplication on hierarchical hpc platforms. Concurrency Comput. Pract. Experience **28**, 02–821 (2016). cpe.3609
2. Wyrzykowski, R., Szustak, L., Rojek, K.: Parallelization of 2D MPDATA EULAG algorithm on hybrid architectures with GPU accelerators. parallel Comput. **40**, 425–447 (2014)
3. Wyrzykowski, R., Szustak, L., Rojek, K., Tomas, A.: Towards efficient decomposition and parallelization of MPDATA on hybrid CPU-GPU cluster. In: Lirkov, I., Margenov, S., Waśniewski, J. (eds.) LSSC 2013. LNCS, vol. 8353, pp. 457–464. Springer, Heidelberg (2014). doi:10.1007/978-3-662-43880-0_52
4. Szustak, L., Rojek, K., Wyrzykowski, R., Gepner, P.: Toward efficient distribution of mpdata stencil computation on intel mic architecture. In: Proceedings of the 1st International Workshop on High-Performance Stencil Computations, pp. 51–56 (2014)
5. Beaumont, O., Boudet, V., Legrand, A., Rastello, F., Robert, Y.: Heterogeneous matrix-matrix multiplication or partitioning a square into rectangles: Np-completeness and approximation algorithms. In: Proceedings of the Ninth Euromicro Workshop on Parallel and Distributed Processing, pp. 298–305 (2001)
6. Lastovetsky, A., Dongarra, J.: High Performance Heterogeneous Computing. Wiley (2009)
7. Smolarkiewicz, P.: Multidimensional positive definite advection transport algorithm: an overview. Int. J. Numer. Meth. Fluids **50**, 1123–1144 (2006)
8. Piotrowski, Z., Wyszogrodzki, A., Smolarkiewicz, P.: Towards petascale simulation of atmospheric circulations with soundproof equations. Acta Geophys. **59**, 1294–1311 (2011)
9. Dichev, K., Lastovetsky, A.: Optimization of collective communication for heterogeneous hpc platforms. Wiley-Interscience (2013)
10. Agarwal, T., Sharma, A., Laxmikant, A., Kale, L.: Topology-aware task mapping for reducing communication contention on large parallel machines. In: IPDPS 2006, p. 10 (2006)
11. Solomonik, E., Bhatele, A., Demmel, J.: Improving communication performance in dense linear algebra via topology aware collectives. In: SC 2011, pp. 77: 1–77: 11. ACM, New York (2011)

12. Kielmann, T., Hofman, R.F., Bal, H.E., Plaat, A., Bhoedjang, R.A.: MagPIe: MPI's collective communication operations for clustered wide area systems. In: ACM Sigplan Notices, vol. 34, pp. 131–140. ACM (1999)
13. Karonis, N., De Supinski, B., Foster, I., Gropp, W., Lusk, E., Bresnahan, J.: Exploiting hierarchy in parallel computer networks to optimize collective operation performance. IPDPS **2000**, 377–384 (2000)
14. Ma, T., Bosilca, G., Bouteiller, A., Dongarra, J.: HierKNEM: an adaptive framework for kernel-assisted and topology-aware collective communications on many-core clusters. In: IPDPS 2012, pp. 970–982 (2012)
15. Kandalla, K., Subramoni, H., Vishnu, A., Panda, D.K.: Designing topology-aware collective communication algorithms for large scale infiniband clusters: case studies with scatter and gather. In: 2010 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), pp. 1–8(2010)
16. Coti, C., Herault, T., Cappello, F.: MPI applications on grids: a topology aware approach. In: Sips, H., Epema, D., Lin, H.-X. (eds.) Euro-Par 2009. LNCS, vol. 5704, pp. 466–477. Springer, Heidelberg (2009). doi:10.1007/978-3-642-03869-3_45
17. Traff, J.: Implementing the MPI process topology mechanism. In: Supercomputing 2002, pp. 1–23 (2002)