Improving Multilevel Approach for Optimizing Collective Communications in Computational Grids

Boro Jakimovski[1], Marjan Gusev[1]

[1] University Sts. Cyril and Methodius,
Faculty of Natural Sciences and Mathematics,
Institute of Informatics,
Arhimedova 5, 1000 Skopje, Macedonia
{boroj, marjan}@ii.edu.mk

Abstract. Collective operations represent a tool for easy implementation of parallel algorithms in the message-passing parallel programming languages. Efficient implementation of these operations significantly improves the performance of the parallel algorithms, especially in the Grid systems. We introduce an improvement of multilevel algorithm that enables improvement of the performance of collective communication operations. An implementation of the algorithm is used for analyzing its characteristics and for comparing its performance it with the multilevel algorithm.

1 Introduction

Computational Grids [1] represent a technology that will enable ultimate computing power at the fingertips of users. Today Grids are evolving in their usability and diversity. New technologies and standards are used for improving their capabilities. Since this powerful resources need to be utilized very efficiently we need to adopt the programming models used in the parallel and distributed computing. One of the main problems facing parallel and distributed computing when introduced to the Grid environment is scalability.

Currently most widely used parallel programming environment is the MPI standard [2]. MPI represents a programming standard that enables implementation of parallelism using message passing. Operations for Collective communication represent a part of the MPI standard that involves communications between a group of processes. Optimizations of collective communications have been the focus of many years of research. This research has led to development of many different algorithms for implementation of collective communications [3]. These algorithms were optimized mainly for cluster computations where the characteristics of the communications between every two nodes are the same.

Main problem of introducing MPI to the Grid environment is the big latency of the communications. Even bigger problem lies in the different latencies of different pairs of processes involved in the communication. This led to the development of new improved algorithms for implementation of collective communication in the Grid environment. Most algorithms for implementation of collective communications are based on tree like communication pattern. There have been many efforts for optimization of the topology of this communication trees for better performance in the Computational Grids.

2 Topology aware collective communications

There have been different approaches for solving the problem of optimizing communication tree for collective operations in Computational Grids. First efforts started with the development of algorithms that involved Minimal Spanning Tree [4], followed by variations of this approach by changing the weights and conditions in the steps for building the communication tree (SPOC [5], FNF [5], FEF [6], ECEF [6], Look-ahead [6], TTCC [7]).

Currently best performing solution is the solution utilizing the network topology information for building the communication tree. This approach, later called topology aware collective communication, was introduced in [7] and later improved in [8][9]. This algorithm involved grouping of the processors in groups where each group represent either processors from one multiprocessor computer or processors from one cluster. Once the groups are defined the communication tree is defined in two levels. The first level contains one group consisting of the root processes from each group. The second level contains the groups defined previously.

The main disadvantage of the two-level algorithm was the utilization of only two levels of communication, local area communication (low latency) and wide area communication (high latency). This disadvantage was overthrown by implementation of multilevel communication pattern introduced by Karonis et. all in [10]. Their approach, implemented in MPICH-G2 [11], defines up to four levels of network communication. Each level consists of several groups of processors where the communications have common characteristics. This way they achieve more adequate topology aware communication pattern which is very close to the optimal.

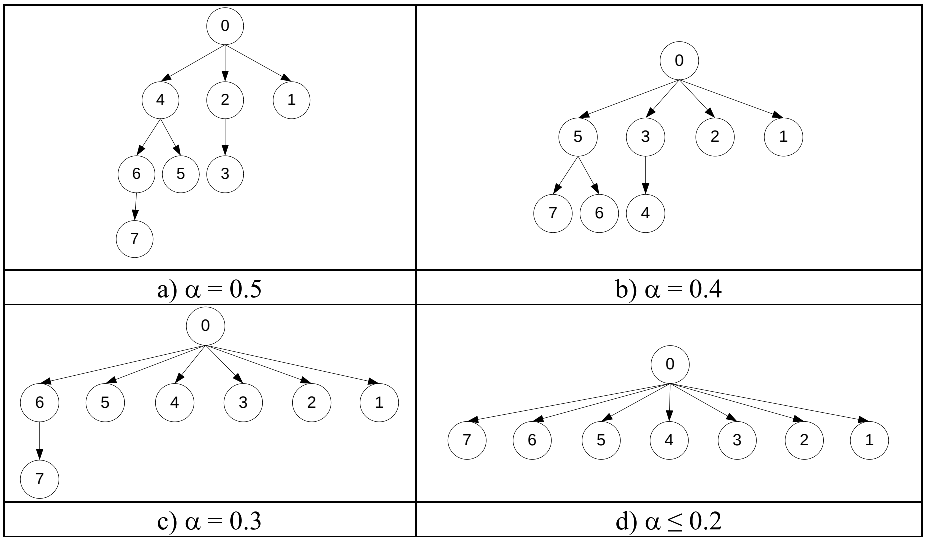
3 Multilevel communication α tree

The multilevel communication tree improves the communication time of collective communications by introducing better topology awareness. The only disadvantage of multilevel communication tree is the choice of communication algorithms. Authors settle for simple solution where they choose one algorithm for high latency level (the first level – wide area level) and another algorithm for low latency levels (all other levels – local area and intra machine). The algorithm chosen for high latency communications is flat tree, which has been shown to behave optimally in such conditions. For low latency communications, the authors choose binomial tree, which also has been shown by LogP model [12] to be optimal in such conditions.

3.1 α communication tree

One of the most advanced algorithms for implementation of collective communication operations is the α-tree algorithm [13]. The algorithm is derived from the theoretically optimal algorithm of λ-tree. The α-tree algorithm overcomes the problems for implementation of the λ-tree but with reduced optimality.

The α-tree algorithm represents a communication tree dependent of the value of the parameter α. The value of the parameter is in the range between 0 and 0.5. The main characteristic of the α-tree is that for α=0, the α-tree looks like flat tree, and for α=0.5 the α-tree looks like binomial tree. This characteristic shows that the α-tree algorithm can adjust itself according to the network characteristics, i.e. if the latency of the communications is low then the value of the parameter will shift towards 0.5, and if the latency of the communications is high then the value of the parameter will move towards 0. This behavior is visually represented in Fig. 1.



a) α = 0.5

b) α = 0.4

c) α = 0.3

d) α ≤ 0.2

Fig. 1. Tree topology changes from binomial tree a) to flat tree d) as the value of the parameter α changes from 0.5 to 0

3.2 Multilevel communication α tree

The disadvantage of the Multilevel communication tree algorithm lies in the choice of the algorithms for communication inside the groups of processes on each level. Our approach tries to improve this disadvantage by defining new communication scheme which will be more efficient then the multilevel tree.

Multilevel communication α tree represents an improvement of the multilevel algorithm. The improvement lies in the ability to properly choose the communication algorithm for each of the four levels of the multilevel communication tree algorithm. The best way to implement the communication algorithm for each group of the communication tree is to properly adjust the communication algorithm according to the latency characteristics of the network. We choose to use the α-tree algorithm for each level of the multilevel tree but with different values of the parameter α for each group. This should enable more flexibility for the algorithm, enabling it to achieve better performance. As it can be seen from the characteristics of the multilevel α tree algorithm, the multilevel tree algorithm represents a special case of the new algorithm when we choose value of α=0 for the first level and value of α=0.5 for all of the other levels. Therefore we can conclude that the new algorithm should enable better performance then the multilevel algorithm.

4 Experimental results

4.1 Simulation environment

The evaluation of the proposed solution for implementation of collective communication was conduced in our relatively small Grid infrastructure. For evaluation purposes we have changed the topology of our Grid by making it more suitable to achieve real results from the simulation. Our simulation resources were four laboratories each with 20 PCs installed with Red Hat Linux and Globus 2.4. The laboratories are separated in two buildings connected between with a link, which we have reduced to 10 Mbit link. Each laboratory we further separated in several clusters of 4-5 PCs. The overall Grid infrastructure used for evaluation of the new algorithm is depicted in Fig.2.

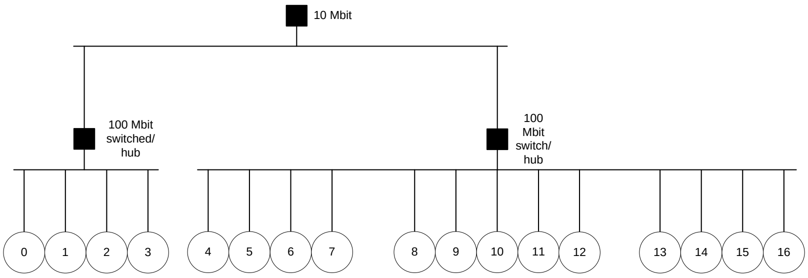


Fig. 2. Topology of the Grid infrastructure used for the experiments for evaluating the new algorithm

From the figure it can be seen that the communication infrastructure on the second and third level is 100 Mbit switch/hub. To simulate different network scenarios we made different experiments in the grid infrastructure either by using switch technology, or by using hub technology.

4.2 Measurement technique

Since the measurement of the performance is the crucial part of the evaluation process it is very important to choose the correct measuring technique. Measuring collective communication requires measurement of consequent execution of many operation calls since one operation call is very short and cannot be measured correctly. Main measurement problem is the pipelining effect that is generated by consecutive calls of the collective operation [14].

The pipelining effect can easily be solved by introducing barriers between consecutive calls of the operations. This approach cannot be implemented straight forward because of the problems that arise with the barrier implementation. When using already implemented barrier from the MPICH library (topology unaware), the main problem lies in the very slow solution for the barrier. This slow solution cannot be used since if the barrier execution time varies only by few percents then the results from the collective operation will disappear completely. In such case the use of some specially tailored barrier is needed. This barrier will not be a real barrier but a barrier that in this circumstances will ensure that the execution will not be pipelined.

Our solution for choice for semi-barrier is the use of opposite ring topology for communication. This means that the between each collective communication operation the processes synchronize between each other by communicating in a ring where each process (with rank r) receives a message from the process with rank $r + 1$ and once it receives the message it resends it to the process with tank $r - 1$. This process starts with the root process and ends with it. This solution represents simple, efficient barrier that reduces pipelining effect in the operation and the barrier.

4.3 Experimental results

Performance measurement of the new algorithm is evaluated by using the broadcast operation. In order to fully evaluate the new algorithm we have measured the performance using many different scenarios. To achieve this in our experiments we have changed the following parameters:

- message size
- parameter α for the second level
- parameter α for the third level
- characteristics of the network topology

The parameter α for the first level is fixed to the value of 0 because this level contains only two processes and any value of α will lead to the same communication topology.

The results of the experiments are depicted in the figures of this chapter. The figures represent three dimensional charts where on the two axes (x and y) are represented the α parameters for the second and the third level of the communication tree. The third axes (z – depth) represents the measured time of the operation for the particular values of α for the second and third level.

On Fig. 3 we can see the results for the measurements for different network characteristic. The figure shows how the performance changes once the network latency increments. The first part of the figure shown the performance of the hub infrastructure. This infrastructure characterizes with bigger latency because of the non-parallel communications. This makes the binomial tree very inefficient because of its low latency and parallel nature. This shows that the optimum shifts towards the lower values of the parameters α. On the other hand the for the switch infrastructure the optimum moves toward the higher values for α, but still doesn't achieve value of 0.5 for both levels which is the case of the multilevel algorithm.

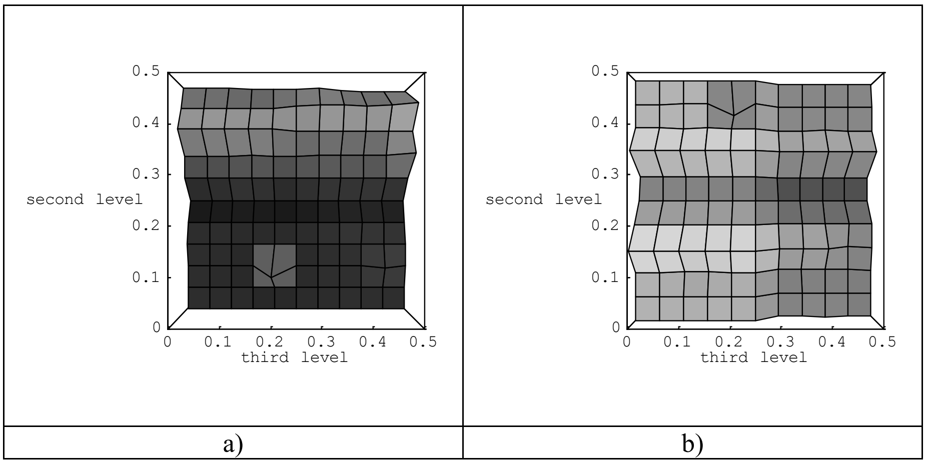
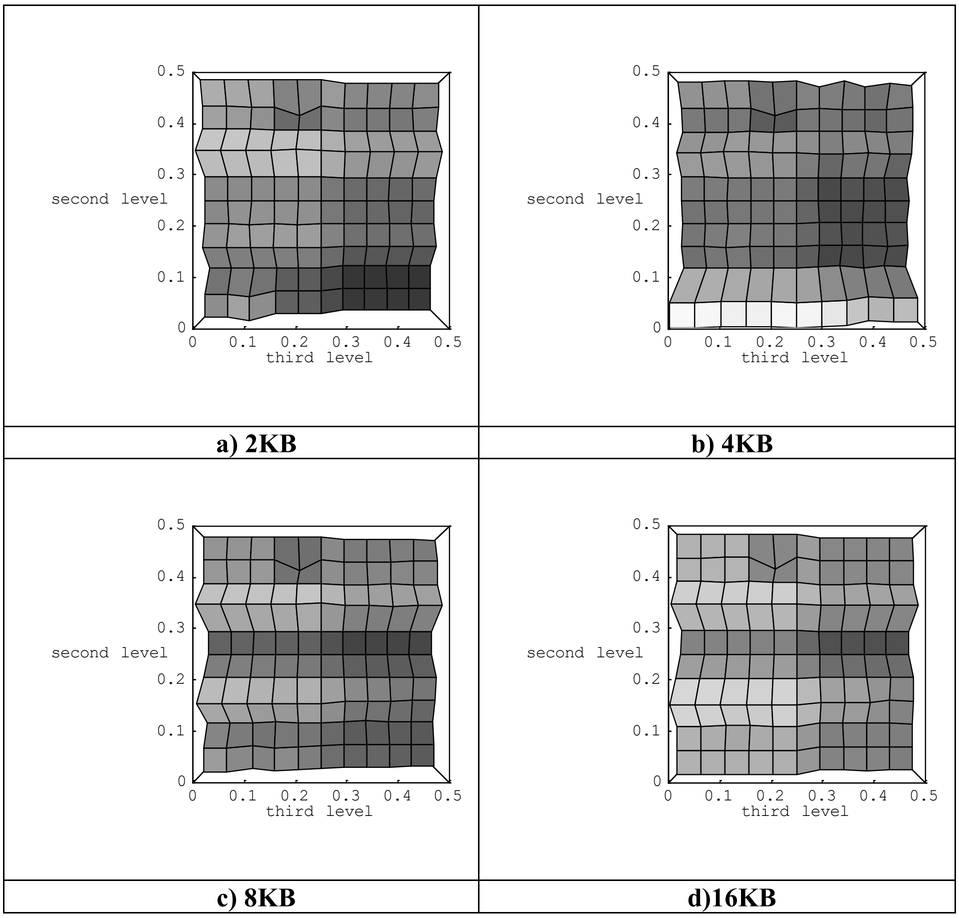


Fig. 3. Results from the simulations using different network topology: a) hub infrastructure, b) switch infrastructure

The second analyzed aspect during the simulations was the effect of the message size over the performance of the operations. Results gathered from the simulations are shown on Fig. 4. As it can be seen from the figures the optimum for the values of the parameter α change as the message size grows. The conducted experiments were for message sizes from 1 byte to 16 KB with the step "power of 2". The results shown on the picture are from 2 KB and above since the results less then 2 KB are identical to the 2 KB results. Reason for this is the usage of TCP/IP protocol that sends messages of approximately 1KB in size even if we send messages with 1 byte of data. The results show the desired performance improvement. It can be concluded that for smaller packet sizes the optimum tends to move towards the lower values of parameters α, and for bigger packet size the optimum moves towards the higher values of the parameters. This is expected since the usage of TCP/IP as transport protocol. When TCP/IP segments large packets into small packets the network communication is flooded with packets and in such case the parallelism in communication is increased.



a) 2KB

b) 4KB

c) 8KB

d)16KB

Fig. 4. Results from the simulations using different message size

4.4 Performance improvement

As it could be expected from the analytical characteristics, experimental results show that the new algorithm gives the opportunity for improvement of the performance of collective communications. The overall improvements gathered from the experimental results are shown on Table 1 and Table 2. The results show that in switch topology the improvements are around 15% compared to the multilevel algorithm. In hub infrastructure the improvements are smaller for the bigger data size, since this introduces many packets to the network which makes even flat tree behave as parallel algorithm, but for smaller data size the improvements are significant and achieve 40% improvement.

Packet size	Optimal time	Multilevel time	α second	α third	Improvement
16KB	42.5472	48.5963	0.3	0.3	12.45%
8KB	24.4484	26.3964	0.25	0.35	7.38%
4KB	9.82545	12.498	0.1	0.35	21.38%
2KB	5.76205	7.16053	0.1	0.4	19.53%
1KB	3.64349	4.04574	0.1	0.35	9.94%
512B	3.17763	3.68794	0.1	0.45	13.84%

Table 1. Performance improvement for switch topology

Packet size	Optimal time	Multilevel time	α second	α third	Improvement
16KB	116.988	125.192	0.25	0	6.55%
8KB	61.2282	64.0655	0.25	0	4.43%
4KB	33.6797	34.9603	0.5	0.25	3.66%
2KB	18.0641	18.6563	0.25	0.25	3.17%
1KB	10.1231	18.4963	0	0.1	45.27%
512B	5.59515	9.34781	0	0.15	40.14%

Table 2. Performance improvement for hub topology

5 Conclusion

As Computational Grids are more widely used, the need for new techniques for parallel and distributed computing are needed. Topology aware communications are essential aspect of improvement of parallel programs. The currently adopted multilevel tree topology aware solution achieves great performance improvements. Still the simple solution in algorithm performance limits the ability to fully utilize the topology and network characteristics.

Our approach represents an improvement to the multilevel approach for optimizing collective communications in Computational Grids. The usage of the α tree algorithm for adopting the different network characteristics enables significant improvement in the collective operations performance. The new algorithm doesn't increase the implementation issues since the α-tree algorithm is very easy to implement and is easily fitted in the MPICH-G2 implementation of topology aware collective communications that utilize communication trees.

Simulation of the new algorithm shows the possibility for improving the performance of collective communications in Computational Grids. Because the measurement of the performance of the algorithm is a very important issue, we gave special attention on problems concerning the measurement techniques. Our choice of measurement technique is consecutive call of the collective operation followed by a semi-barrier implemented using opposite ring communication topology.

Future work is research in developing an efficient and easy way of utilizing this new approach for use in Computational Grids. This will require a mechanism for gathering and using network characteristics for automatic parameter selection.

References

1. Foster, I., Kesselman, C. ed.: The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, 1999.
2. Message Passing Interface Forum: MPI: A message-passing interface standard. International Journal of Supercomputer Applications, 8(3/4) (1994) 165-414
3. Vadhiyar, S. S., Fagg, G. E., Dongarra, J.: Automatically Tuned Collective Communications. Proceedings of the IEEE/ACM SC2000 Conference, Dallas, Texas (2000)
4. Lowekamp, B. B., Beguelin. A.: ECO: Efficient Collective Operations for communication on heterogeneous networks. Proc. of 10th Intl. Parallel Processing Symposium, (1996) 399–405
5. Banikazemi, M., Moorthy, V., Panda, D.: Efficient Collective Communication on Heterogeneous Networks of Workstations. International Conference on Parallel Processing. Minneapolis, MN (1998) 460–467
6. Bhat, P.B., Raghavendra, C.S., Prasanna, V.K.: Efficient Collective Communication in Distributed Heterogeneous Systems. Proceedings of the International Conference on Distributed Computing Systems (1999)
7. Cha, K., Han, D., Yu, C., Byeon, O.: Two-Tree Collective Communication in Distributed Heterogeneous Systems. IASTED International Conference on Networks, Parallel and Distributed Processing, and Applications (2002)
8. Kielmann, T., Hofman, R. F. H., Bal, H. E., Plaat, A., Bhoedjang, R. A. F.: MAGPIE: MPI's Collective Communication Operations for Clustered Wide Area Systems. Proc. Symposium on Principles and Practice of Parallel Programming (PPoPP), Atlanta, GA, (1999) 131–140
9. Kielmann, T., Bal, H. E., Gorlatch, S.: Bandwidth-efficient Collective Communication for Clustered Wide Area Systems. IPDPS 2000, Cancun, Mexico (2000)

10 Karonis, N., de Supinski, B., Foster, I., Gropp, W., Lusk, E., Bresnahan, J.: Exploiting hierarchy in parallel computer networks to optimize collective operation performance. Proc. of the 14th International Parallel and Distributed Processing Symposium, (2000) 377–384

11 MPICH-G2 web page. http://www.globus.org/mpi.

12 Culler, D.E., Karp, R., Patterson, D.A., Sahay, A., Schauser, K.E., Santos, E., Subramonian, R., von Eicken, T: LogP: Towards a realistic model of parallel computation. Proceedings of the 4th SIGPLAN Symposium on Principles and Practices of Parallel Programming, (1993) 1–12

13 Bernaschi, M., Iannello, G.: Collective Communication Operations: Experimental Results vs. Theory. Concurrency: Practice and Experience, (1998), 10(5):359–386

14 Lacour, S.: MPICH-G2 collective operations: performance evaluation, optimizations. Rapport de stage MIM2, Magistère d'informatique et modélisation (MIM), ENS Lyon, MCS Division, Argonne National Laboratory, USA, (2001)