# Two Algorithms of Irregular Scatter/Gather Operations for Heterogeneous Platforms

Kiril Dichev, Vladimir Rychkov, and Alexey Lastovetsky

UCD School of Computer Science and Informatics,
University College Dublin,
Belfield, Dublin 4, Ireland

**Abstract.** In this work we present two algorithms of irregular scatter/gather operations based on the binomial tree and Träff algorithms. We use the prediction provided by heterogeneous communication performance models when constructing communication trees for these operations. The experiments show that the model-based algorithms outperform the traditional ones on heterogeneous platforms.

**Keywords:** Heterogeneous platform, communication performance model, collective communication, scatterv, gatherv.

## 1 Introduction and Related Work

Algorithms for MPI collective communication operations typically implement them as a combination of point-to-point operations in a tree representing the communication partners and the way messages are exchanged between them. Traditional tree-based implementations target homogeneous platforms, implicitly assuming identical processors and a homogeneous communication layer. When applied to heterogeneous platforms, these implementations may be far from the optimal.

We propose to use heterogeneous communication performance models and their prediction for finding more optimal communication trees for these algorithms. The models take into account the underlying heterogeneous network of computers when constructing communication trees. In this work, we only use $t(i, j, m)$ , the prediction of the execution time of sending a message of size $m$ from process $i$ to process $j$, in the algorithm design.

Optimization of collectives is not a new research topic and a wide range of optimized collective algorithms have been proposed in the past [1,3,5]. Communication performance models [4] have been used for collectives by predicting the runtime of various collective algorithms and switching between them accordingly [6]. In this work, we use model predictions during the dynamic construction of communication trees either by changing the mapping or changing the tree structure altogether. This topic has not been of practical interest to the best of our knowledge, except for [7], where the regular gather collective operation is optimized.

R. Keller et al. (Eds.): EuroMPI 2010, LNCS 6305, pp. 289–293, 2010.

## 2 Model-Based Algorithms of Irregular Scatter/Gather

The shortcomings of the homogeneous algorithms are also valid for scatter and gather algorithms. While the regular variants of these operations only allow for same-sized chunks of data to be scattered or gathered, their irregular counterparts support different data sizes at each process. Irregular scatter/gather operations are used particularly for heterogeneous algorithms which distribute data according to the different computational capacity of the different processes. In this work, we demonstrate our approach on the example of two existing tree construction algorithms for irregular scatter/gather operations. We integrate the model prediction $t(i, j, m)$ into the algorithms to produce more optimal communication trees.

A model-based algorithm for tree construction derived from an algorithm that does not use models can differ from it by changing the process mapping to nodes or by constructing a tree with a different tree structure. From the two algorithms we present, the first one changes the mapping, while the second one may construct a different tree structure as well.

*Model-based binomial tree scatterv/gatherv.* In this algorithm we use point-to-point predictions to map processes to a binomial tree. The binomial tree is constructed in a depth-first manner, starting with the lowest-order subtrees. Each new tree node receives the process number i from the set of free processes that has minimal ($minimum - first$) or maximal ($maximum - first$) predicted communication time $t(parent, i, m_i)$, where $m_i$ is the message size assigned to process $i$. A good choice of mapping depends on the runtime platform. For example, on a heterogeneous cluster with a single switch, $maximum - first$ mapping may be better since the subtrees of a parent node will be balanced in their communication costs to it. In a hierarchy of clusters $minimum - first$ mapping may be better because intra-cluster processes are likely to be mapped to the same communication subtree.

*Model-based Träff algorithm for scatterv/gatherv.* We will significantly modify an algorithm by Träff [2] which targets irregular scatter/gather operations when constructing a communication tree. Träff considers the message size assigned to each process and assumes identical links between all nodes, while we consider both the message size and the characteristics of the links between nodes by using the prediction function $t(i, j, m)$. Even for a fixed node count, the original algorithm can generate different trees depending on the message sizes at the node level. Since our modified algorithm observes the weight of the links instead, both the process mapping as well as the tree structure can differ from the original algorithm.

*Given* :

- set of nodes S with corresponding sendcounts/recvcounts arrays defining the message size to be sent to/received from each process
- defined predictor $t(i, j, m)$

*Result* :

– a communication tree

*Algorithm* :

Starting from a set of processes to build a tree with a given root (Fig. 1a), we sort them decreasingly using $t(i, j, m)$ and partition the sorted set into subsets(Fig. 1b). These subsets are balanced in their total communication cost with the root node - left subtrees have less processes with slower transfer times while the right subtrees have more processes with faster transfer times. Each subset then chooses the process $i$ which would take the least time to transfer all messages of this subset to/from the parent node $j$, removes it from the subset and an edge $(i, j)$ is created (Fig. 1c). We repeat the algorithm for all further subsets until the tree is fully constructed. The outlined phases have to be repeated in each step since we predict the communication time of new process pairs.

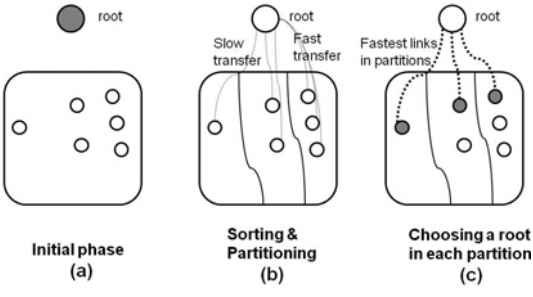


**Fig. 1.** Main phases in the modified Träff algorithm [2]

## 3 Experimental Results

We present results for the two irregular scatter/gather algorithms described in section 2. Our experiments used benchmarks implemented in the CPM/MPIBlib framework [4]. We used two different platforms the HCL cluster with a single Gigabit Ethernet switch and a larger and more heterogeneous multi-site cluster of clusters known as Grid5000. On both platforms, we used the Hockney model for our predictions. Since the prediction we use is not model-specific, other communication models can be used as well. An important consideration is that scatterv/gatherv operations can use any distribution of message sizes. We used a setup which assigns each node a message size based on its CPU speed (delivered by a trivial benchmark).

We first tested this message size distribution on the HCL cluster and Open MPI (1.2.8) on 14 nodes, running one process per node. Since this platform did not provide a high level of network heterogeneity, the improvement demonstrated by our algorithms was not significant. We then experimented with a message size distribution with a larger ratio between maximal and minimal message size. The

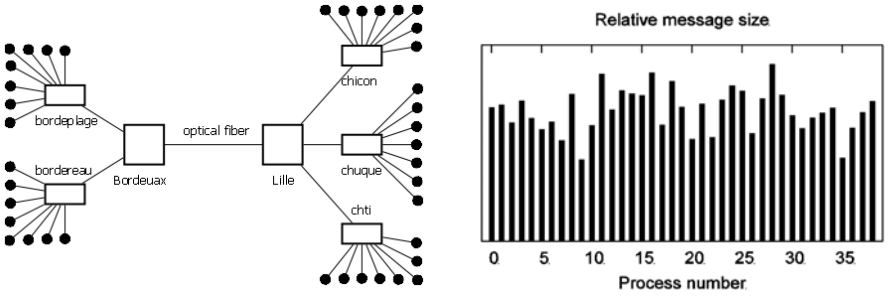


**Fig. 2.** Experimental setup and message size distribution on Grid5000

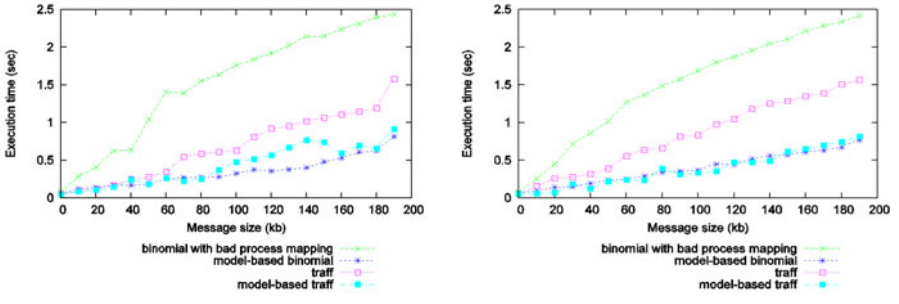


**Fig. 3.** Benchmarks on Scatterv (a) and Gatherv (b) operations on Grid5000

modified algorithm of Trff had similar timings to the original algorithm. The results were best for the model-based binomial tree algorithm with maximum-first mapping - compared to the original algorithm using an arbitrary binomial tree, it was faster by 25-35% or more for larger messages. In the case of a high message size variation, the prediction still had a positive impact for this cluster.

The experiments on Grid5000 used 39 nodes from 5 clusters located on 2 sites, running one process per node. Fig. 2 displays the experimental setup and the CPU-based message size distribution. MPICH2 (version 1.2.1) was used with TCP/IP as communication layer. The results (Fig. 3) demonstrate that on heterogeneous networks both model-based algorithms clearly outperform their non-model-based counterparts we observed time reductions for scatterv and gatherv of up to 75% for the binomial algorithm (minimum-first mapping) and up to 60% for Träffs algorithm. This confirms that our approach is particularly useful for platforms with high network heterogeneity.

**Acknowledgments.** This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grant Number 08/IN.1/I2054.

## References

1. Thakur, R., Rabenseifner, R., Gropp, W.: Optimization of Collective Communication Operations in MPICH. Int. J. of High Perf. Comp. App. 19, 49–66 (2005)
2. Träff, J.L.: Hierarchical Gather/Scatter Algorithms with Graceful Degradation. In: IPDPS 2004, vol. 1, pp. 80–89. IEEE, Los Alamitos (2004)
3. Worringen, J.: Pipelining and Overlapping for MPI Collective Operations. In: LCN 2003, pp. 548–557. IEEE, Los Alamitos (2003)
4. Lastovetsky, A., Rychkov, V., OFlynn, M.: Accurate heterogeneous communication models and a software tool for their efficient estimation. Int. J. of High Perf. Comp. App. 24, 34–48 (2010)
5. Chan, E.W., Heimlich, M.F., Purkayastha, A., van de Geijn, R.A.: On optimizing collective communication. In: Cluster 2004, pp. 145–155. IEEE, Los Alamitos (2004)
6. Pjesivac-Grbovic, J., Angskun, T., Bosilca, G., Fagg, G., Gabriel, E., Dongarra, J.: Performance analysis of MPI collective operations. Cluster Comput. 10(2), 127–143 (2007)
7. Hatta, J., Shibusawa, S.: Scheduling algorithms for efficient gather operations in distributed heterogeneous systems. In: WPP 2000, pp. 173–180. IEEE, Los Alamitos (2000)