Accurate and Efficient Estimation of Parameters of Heterogeneous Communication Performance Models

Alexey Lastovetsky and Vladimir Rychkov

International Journal of High Performance Computing Applications 2009; 23; 123
DOI: 10.1177/1094342009103947

ACCURATE AND EFFICIENT ESTIMATION OF PARAMETERS OF HETEROGENEOUS COMMUNICATION PERFORMANCE MODELS

Alexey Lastovetsky
Vladimir Rychkov

SCHOOL OF COMPUTER SCIENCE AND INFORMATICS,
UNIVERSITY COLLEGE DUBLIN, BELFIELD, DUBLIN 4,
IRELAND
(ALEXEY.LASTOVETSKY@UCD.IE,
VLADIMIR.RYCHKOV@UCD.IE)

Abstract

Analytical predictive communication models play an important role in the optimization of communication operations in scientific applications running on computational clusters. The effectiveness of this model-based optimization strongly depends on the accuracy of the estimation of the parameters of these models. The task of accurate estimation of the model is particularly challenging for heterogeneous communication models that use a much larger number of point-to-point parameters than their homogeneous counterparts. One particular challenge occurs when the number of point-to-point parameters describing communication between a pair of processors becomes larger than the number of independent point-to-point communication experiments traditionally used for estimation of the parameters. In this paper, we address this and other related issues and propose an approach that allows us to design a set of communication experiments sufficient for the accurate and efficient estimation of the parameters of a heterogeneous communication performance model. The experiments on heterogeneous clusters demonstrate the accuracy and efficiency of the proposed solution.

Key words: heterogeneous cluster, heterogeneous communication performance model, MPI, communication model estimation

The International Journal of High Performance Computing Applications,
Volume 23, No. 2, Summer 2009, pp. 123–139
DOI: 10.1177/1094342009103947

Figures 2–4 appear in color online: http://hpc.sagepub.com

1 Introduction

A programming system for high performance computing on heterogeneous platforms, such as mpC (Lastovetsky 2002), HeteroMPI (Lastovetsky and Reddy 2006), or GridSolve (Arnold, Casanova, and Dongarra 2002), strongly relies on the performance model of the executing platform, which is used for prediction of the execution time of different configurations of the application in order to find the optimal one. The prediction includes computation and communication costs and depends on the accuracy of estimation of parameters of the model. In this paper, we deal with assessing the parameters of communication performance models of heterogeneous clusters based on a switched network, which are arguably the most common platform for parallel computing.

Traditionally, communication performance models for high performance computing are analytical and built for homogeneous clusters. The basis of these models is a point-to-point communication model characterized by a set of integral parameters, having the same value for each pair of processors. The execution time of other operations (which are, in fact, collective), is expressed as a combination of the point-to-point parameters, and is analytically predicted for different message sizes and numbers of processors involved. The core of this approach is the choice of a point-to-point model that is the most appropriate to the targeted platform, allowing for easy and natural expression of different algorithms of collective operations. For homogeneous clusters, the point-to-point parameters are found statistically from communication experiments between any two processors. Typical experiments include sending and receiving messages of different sizes, with the communication execution time being measured on one side.

A homogeneous communication model can be applied to a cluster of heterogeneous processors by averaging values obtained for every pair of processors. In this case, the heterogeneous cluster will be treated as homogeneous in terms of the performance of communication operations. If some processors or links in the heterogeneous cluster significantly differ in performance, predictions based on the homogeneous communication model may become inaccurate. More accurate performance models would not average the point-to-point communication parameters. In this paper, we show that the use of such heterogeneous communication models in model-based optimization of MPI collective operations on heterogeneous clusters does improve their performance.

While more accurate, the heterogeneous models have a significantly larger number of parameters. This results in a higher cost of their estimation. In particular, when applied to the heterogeneous communication model, the statistical methods of finding the point-to-point parameters, traditionally used in the case of homogeneous communication models, require a significantly larger number of measure-

ments. For our target architecture, which is a heterogeneous cluster based on a switched network, we can address this problem by performing most of the communication experiments in parallel, using the fact that the network switches provide no-contention point-to-point communications, appropriately forwarding packets between sources and destinations.

A heterogeneous communication performance model, LMO, has recently been proposed to allow for easy and intuitive expression of the execution time of collective operations (Lastovetsky, Mkwawa, and O'Flynn 2006; Lastovetsky and O'Flynn 2007). Unfortunately, this goal can only be achieved by introducing more point-to-point parameters than it is possible to estimate with help of the standard point-to-point communication experiments. The estimation of such communication models is a new and non-trivial problem. This paper proposes a solution to this problem. The idea is to introduce additional collective communication experiments involving more than two processors. To make use of the results of these additional independent experiments, we propose to extend the heterogeneous point-to-point communication model by a model of these collective operations and use this extended model to obtain additional independent equations for the estimated parameters.

This paper is organized as follows. Section 2 outlines related work on the estimation of the homogeneous communication performance models for homogeneous platforms and their use for optimization of communication. In Section 3, we discuss and compare different approaches to communication performance models for heterogeneous clusters. We conclude that while the advanced heterogeneous models are the best option for model-based optimization of communications, the task of estimation of these models is particularly challenging and cannot be solved by traditional methods. In Section 4, we propose our approach to accurate and efficient estimation of the parameters of these models. In Section 5, we apply this approach to estimation of a particular heterogeneous communication model. Section 6 presents the experimental results that demonstrate the accuracy and efficiency of the proposed solution. In Section 7, we show the results of the model-based optimization of collective communication operations for parallel matrix multiplication on a heterogeneous cluster.

2 Communication Models for Homogeneous Platforms: Estimation and Use

In this section, we discuss how the point-to-point parameters of traditional communication performance models are estimated for homogeneous platforms. In this case, the parameters will be the same for any pair of processors. Therefore, to estimate them, communication experiments between any two processors will be sufficient.

The Hockney model (Hockney 1994) estimates the execution time of point-to-point communication as $\alpha + \beta M$, where α is the latency, β is the bandwidth, and M is the message size. There are two ways to obtain a statistically reliable estimation of the Hockney parameters:

1. To perform two series of roundtrips: with empty messages (to get the latency parameter from the average execution time), and with non-empty ones (to get the bandwidth).
2. To perform a series of roundtrips with messages of different sizes and use results in a linear regression which fits the execution time into a linear combination of the Hockney parameters and a message size.

The LogP model (Culler et al. 1993) predicts the time of network communication for small fixed-sized messages in terms of the latency, L, the overhead, o, the gap per message, g, and the number of processors, P. The latency, L, is an upper bound on the time to transmit a message from its source to destination. The overhead, o, is the time period during which the processor is engaged in sending or receiving a message. The gap, g, is the minimum time between consecutive transmissions or receptions; it is the reciprocal value of the end-to-end bandwidth between two processors, so that the network bandwidth can be expressed as L/g. According to LogP, the time of point-to-point communication can be estimated by $L + 2o$. In Culler et al. (1996), the estimation of the LogP parameters is presented, with the sending, o_s, and receiving, o_r, overheads being distinguished. The set of communication experiments used for estimation of the LogP parameters is as follows:

- To estimate the sending overhead parameter, o_s, a small number of messages are sent consecutively in one direction. The averaged sending time measured on the sender side will approximate o_s.
- The receiving overhead, o_r, is found directly from the time of receiving a message in the roundtrip. In this experiment, after completion of the send operation, the sending processor waits for some time, sufficient for the reply to reach the receiving processor, and only then posts a receive operation. The execution time of the receive operation is assumed to approximate o_r.
- The latency is found from the execution time of the roundtrip with small messages $L = RTT/2 - o_s - o_r$.
- To estimate the gap parameter, g, a large number of messages are sent consecutively in one direction. The gap is estimated as $g = T_n/n$, where n is a number of messages

and T_n is the total execution time of this communication experiment measured on the sender processor. The number of messages is chosen to be large to ensure that the point-to-point communication time is dominated by the factor of bandwidth rather than latency. This experiment, also known as a saturation, reflects the nature of the gap parameter but takes a long time.

In contrast to the Hockney model, LogP is not designed for communications with arbitrary messages, but its extension LogGP model (Alexandrov et al. 1997), takes into account the message size by introducing the gap per byte parameter, G. The point-to-point communication time is estimated by $L + 2o + (M - 1)G$. The gap per byte, G, can be assessed in the same way as the gap parameter of the LogP model, saturating the link with large messages of size $\overline{M}$: $G = \frac{T_n(\overline{M})/n}{\overline{M}}$.

In the PLogP (parameterized LogP) model (Kielmann, Bal, and Verstoep 2000), all parameters except for latency are piecewise linear functions of the message size, and the meaning of parameters slightly differs from LogP. The meaning of latency, L, is not intuitive; rather it is a constant that combines all fixed contribution factors such as copying to/from the network interfaces and the transfer over the network. The send, $o_s(M)$, and receive, $o_r(M)$, overheads are the times that the source and destination processors are busy for the duration of communication. They can be overlapped for sufficiently large messages. The gap, $g(M)$, is the minimum time between consecutive transmissions or receptions of messages of a given size M. The gap is assumed to cover the overheads: $g(M) \geq o_s(M)$ and $g(M) \geq o_r(M)$. According to the PLogP model, the point-to-point execution time is equal to $L + g(M)$ for the message of M bytes. This model is adaptive in nature: the number and location of breaks of piecewise linear functions are determined during its construction, the total number of parameters may become too large. The estimation of the PLogP parameters includes the following experiments:

- The overheads $o_s(M)$ and $o_r(M)$ are measured directly from the time of sending/receiving a message of M bytes in the roundtrips $i \xrightarrow[0]{M} \xleftarrow{} j$, consisting of a single sending of the message of M bytes from processor i to processor j and a single zero reply, and $i \xrightarrow[M]{0} \xleftarrow{} j$, consisting of an empty send and non-empty reply. For each message size, these tests are initially run a small number of times. They are repeated as many times as the statistical parameters, which determine the confidence interval likely to include the parameters, require. These experiments are performed for multiple message sizes, which are selected adaptively. For example, if the $o_s(M_k)$ is not consistent with the linearly extrapolated value based on $o_s(M_{k-2})$ and $o_s(M_{k-1})$, then another measurement is performed for the message size $M_k^* = (M_k + M_{k-1})/2$, and the $o_s(M_k^*)$ is estimated.
- To assess the gap parameter, the saturation for each message size has to be performed. The authors of PLogP suggested a less time-consuming indirect method of finding the gap parameter, which requires only one saturation experiment to estimate $g(0)$ and single roundtrips with requests $M > 0$ and empty replies to estimate $g(M)$. As the execution time of such a roundtrip is $RTT(M) = L + g(M) + L + g(0)$, $g(M)$ will be expressed as follows $g(M) = RTT(M) - RTT(0) + g(0)$.
- The latency is found as $L = RTT(0)/2 - g(0)$.

In our experiments on various clusters, we observed that the estimates of the gap obtained by the indirect method could be several times less accurate than the value obtained by the direct method. Moreover, the gap values found with the optimized technique are often less than send/receive overheads for small and medium messages, which contradicts the assumption that $g(M) \geq o_s(M)$ and $g(M) \geq o_r(M)$.

One application of communication performance models is the optimization of MPI collective operations. Traditionally, the research on optimization of collective communications focuses on the analysis of such collective communication operations that allow for many different algorithms to be implemented via point-to-point communications on different tree topologies. The goal is to find the optimal algorithm for each particular network configuration with respect to the prediction provided by the communication performance model.

Thakur, Rabenseifner, and Gropp (2005) used the Hockney model to estimate the communication performance of different algorithms of collective operations. For a particular collective operation they suggested switching between algorithms to choose the fastest one for each given message size and number of processors. Kielmann et al. (1999) used the PLogP model to find an optimal algorithm for collective operations on clusters connected by a wide area network. The design of their algorithms of collective operations is based on intra- and inter-cluster graphs of processors; they switch between different shapes of graphs for different message sizes to get the best prediction of execution time. Pjesivac-Grbovic et al. (2007) applied the Hockney, LogP/LogGP, and PLogP models to different algorithms and topologies for barrier, broadcast, reduce, and all-to-all operations. They showed that the estimations provided by the traditional models might differ from the observed communication execution times, resulting in non-optimal switch between algorithms, and

presented the decision functions based on empirical data obtained from excessive measurements of the execution time of different algorithms.

All these approaches were applied to homogeneous platforms. They considered a fixed set of commonly used algorithms for each collective with a predetermined form of communication trees. The heterogeneous communication performance models can provide another approach to the model-based optimization: the building of optimal communication trees by using the prediction of the execution time for each link.

Bhat, Prasanna, and Raghavendra (2003) and Hatta and Shibusawa (2000) used a heterogeneous Hockney model to build the optimal communication trees for broadcast and gather. They applied different heuristics based on the Hockney prediction on either the whole or some of its parameters, for example, "broadcast: fastest node first" (selection of the node, the average latency with which is minimal), "broadcast, gather: fastest edge first/last" (selection of the communication for which the estimation of the execution time is the smallest), "gather: widest bandwidth last" (selection of the communication with the maximum bandwidth) and so on. The authors of these works used the heterogeneous Hockney extension just for relative estimation of the point-to-point communications but did not build models of collective operations. To the best of the authors' knowledge, there are no publications on the modeling of MPI collective operations on heterogeneous clusters.

3 Communication Performance Models for Heterogeneous Clusters

Heterogeneous computational clusters with MPI as the principle programming system have become a popular platform for parallel computing. Unfortunately, many MPI-based applications that were originally designed for homogeneous platforms do not demonstrate the same performance on heterogeneous ones and require optimization. The optimization of parallel applications is typically based on the performance models of heterogeneous clusters, which are used for prediction of the execution time of different configurations of the application, including its computation and communication costs. The optimization of communications, in particular MPI collective operations, is an important aspect of the optimization of parallel applications. The model-based optimization can significantly improve the performance of collective operations on both homogeneous and heterogeneous platforms (Kielmann et al. 1999; Hatta and Shibusawa 2000; Bhat et al. 2003; Thakur et al. 2005; Pjesivac-Grbovic et al. 2007).

There are two main approaches to modeling the performance of communication operations on heterogeneous clusters. The first is to apply traditional *homogeneous* communication performance models to heterogeneous clusters. In this case, the parameters of the models are estimated for each pair of processors and the average values for all pairs are then used in modeling. The second approach is to use dedicated *heterogeneous* models, where different pairs of heterogeneous processors are characterized by different parameters. In this section, we compare these two approaches and show that while not that simple in use, heterogeneous communication models are more accurate and outperform their homogeneous counterparts in the model-based optimization of communication operations on heterogeneous clusters.

The number of parameters is the principle difference between the homogeneous and heterogeneous models. A homogeneous model uses a small number of the average parameters that are applied to any pair of processors, whereas in heterogeneous models, different pairs of processors are characterized by different parameters, making the total number of parameters needed to describe an n-processor heterogeneous cluster of the order of $O(n^2)$.

The small number of parameters is an obvious advantage of the homogeneous models over the heterogeneous ones. It allows the expression of the execution time of any communication operation by a simple compact formula, which is independent of the processors involved in the operation. For example, with the homogeneous Hockney model the execution time of the binomial (minimum spanning tree) algorithm of scatter/gather can be approximated by the following formula: $(\log_2 n)\alpha + (n-1)\beta M$, where M is the size of the receive (scatter) and send (gather) buffers. In each sub-tree, the largest messages are sent first (scatter) or received last (gather). The formula includes parallel (constant contributions in sub-trees of the same height) and sequential (accumulated variable contributions) parts. The binomial communication tree for 16 participating processors is shown in Figure 1. The nodes of the tree represent the processors. The arcs represent the logical communication links between the processors. Given 16 data blocks are to be scattered/gathered, each arc is marked by the number of blocks communicated over the corresponding link during the execution of the algorithm. According to the formula, the variable contributions are accumulated for all links, while the constant contributions are taken once for all sub-trees of the same height: C_k, $k = 1, \ldots, 4$.

We extend the Hockney model for heterogeneous clusters and introduce different parameters α_{ij} and β_{ij} for different pairs of processors. We assume symmetric communications, $\alpha_{ij} = \alpha_{ji}$, $\beta_{ij} = \beta_{ji}$, which is natural for the switched network. For the execution time of the binomial scatter/gather, we suggest the following formula, recursively applying to the sub-trees:

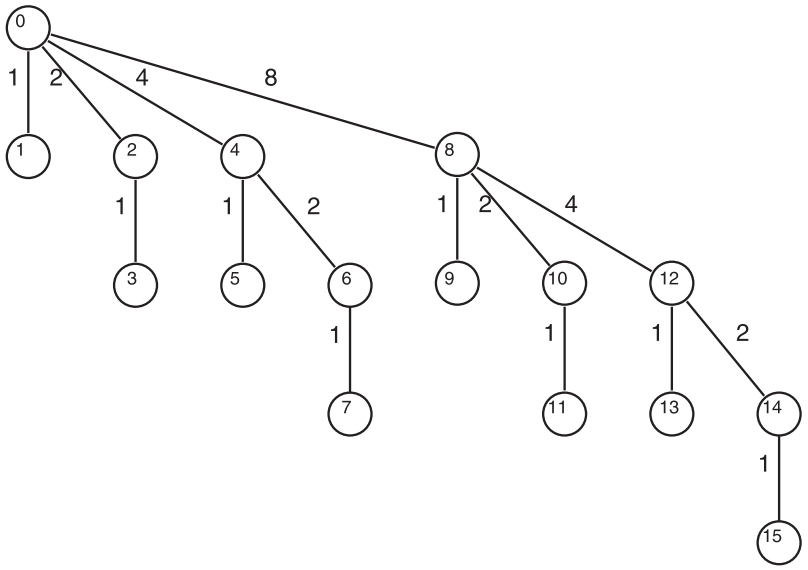

Fig. 1 The binomial communication tree for scatter (gather) involving 16 processors. The nodes represent the processors. Each arc represents a logical communication link and is marked by the number of data blocks communicated over this link.

$$T(k) = \alpha_{rs} + \beta_{rs} 2^{k-1} M + \max_{c \in C_{k-1}} T_c(k-1) \tag{1}$$

where k is an order of the sub-tree (starts with $\log_2 n$ – the whole tree), r is a root processor of the sub-tree (0, for the whole tree), and s is a root of a sub-sub-tree with the highest order (8, for the whole tree in Figure 1). $T_c(k-1)$ is the execution time of the sub-tree c of order $k-1$ from the set C_{k-1}. For the tree in Figure 1, C_3 consists of two sub-trees, with roots 0 and 8.

The execution times of sending/receiving of the largest block in each sub-tree are summed (sequential part). As communications in the sub-trees of the same height are performed in parallel, the formula includes maximums and recursion. For eight participating processors with the root 0 the formula will look as follows:

$$\alpha_{04} + 4\beta_{04} M + \max \left\{ \begin{array}{l} \alpha_{02} + 2\beta_{02} M + \max \left\{ \begin{array}{l} (\alpha_{01} + \beta_{01} M) \\ (\alpha_{23} + \beta_{23} M) \end{array} \right\} \\ \alpha_{46} + 2\beta_{46} M + \max \left\{ \begin{array}{l} (\alpha_{45} + \beta_{45} M) \\ (\alpha_{67} + \beta_{67} M) \end{array} \right\} \end{array} \right\} \tag{2}$$

One can see that the formula for the homogeneous Hockney model is a special case of this formula. If all the point-to-point parameters are the same, then in the case of eight processors it will be rewritten as:

$$\begin{aligned} &\alpha + 4\beta M + \alpha + 2\beta M + \alpha + \beta M \\ &\approx \log_2 8\alpha + (8-1)\beta M \end{aligned} \tag{3}$$

While simpler in use, the homogeneous models are less accurate. When some processors or links in the heterogeneous cluster significantly differ in performance, predictions based on the homogeneous models may become quite inaccurate.

Accuracy. The use of the homogeneous models for minimization of the communication cost of the application will often result in a solution that is far away from the optimal. Being more accurate, the heterogeneous models have the potential to significantly outperform the homogeneous ones in the model-based optimization of communication operations on heterogeneous clusters. To demonstrate this advantage of the heterogeneous models, we conducted the following experiments on a 16-node heterogeneous cluster (the specification of each of the seven node types of the cluster is given in Table 1).

First, we found the parameters of the homogeneous Hockney model of this cluster: $\alpha = 177$ μs, $\beta = 0.0219$ μs/B. The execution time of the binominal scatter predicted by this model will be the same independent of which processor will be the root of the operation and how the remaining processors will be mapped to the nodes of the communication tree. Therefore, given the root is fixed, according to the homogeneous Hockney model, any

Table 1
Specification of the 16-node heterogeneous cluster.

Node type	Model	OS	Processor	Front side bus	L2 Cache	Number of nodes of the type
1	Dell Poweredge SC1425	FC4	3.6 Xeon	800 MHz	2 MB	2
2	Dell Poweredge 750	FC4	3.4 Xeon	800 MHz	1 MB	6
3	IBM E-server 326	Debian	1.8 AMD Opteron	1 GHz	1 MB	2
4	IBM X-Series 306	Debian	3.2 P4	800 MHz	1 MB	1
5	HP Proliant DL 320 G3	FC4	3.4 P4	800 MHz	1 MB	1
6	HP Proliant DL 320 G3	FC4	2.9 Celeron	533 MHz	256 KB	1
7	HP Proliant DL 140 G2	Debian	3.4 Xeon	800 MHz	1 MB	3

Table 2
Parameters for links between the seven types of nodes specified in Table 1. The first value, α, is measured in μs, the second, β, in μs/B.

Node type	1	2	3	4	5	6	7
1	248 0.0224	244 0.0232	121 0.0211	493 0.0360	126 0.0220	156 0.0218	121 0.0207
2		243 0.0211	117 0.0196	493 0.0331	118 0.0198	118 0.0209	118 0.0196
3			812 0.0190	498 0.0306	888 0.0192	961 0.0196	665 0.0188
4				X	496 0.0305	496 0.0302	497 0.0300
5					X	125 0.0201	993 0.0194
6						X	107 0.0199
7							728 0.0190

mapping of the remaining processors will be equally optimal.

Then, we found the parameters of the heterogeneous Hockney model of this cluster. Table 2 presents these parameters for the links between different types of the nodes (the first value, α, is measured in μs, the second, β, in μs/B). Because of the symmetry of communications, we present only a half of the matrix. The "X" indicates that there is only one node of the corresponding type. We fixed a root processor (Dell Poweredge SC1425, 3.6 Xeon, 800 MHz, 2 MB) and used the heterogeneous Hockney model to predict the execution time of the binomial scatter for different mappings of the remaining processors to the nodes of the tree. Our calculations showed that for some mappings the difference in the execution time exceeded 30%. The same results were obtained for the binomial gather. That is, for this cluster, the use of the heterogeneous Hockney model instead of the homogeneous one has the potential to improve the performance of the collective communication operations.

To validate these calculations, we conducted experiments for two different mappings of the processors to the nodes of the binomial tree with the fixed root processor. Both methods are based on a difference in the size of messages to be passed between the nodes in the binomial tree. The idea of the first mapping is to send larger blocks via faster links. Another, pessimistic, mapping uses the slowest links for large blocks. The speed of each link is estimated

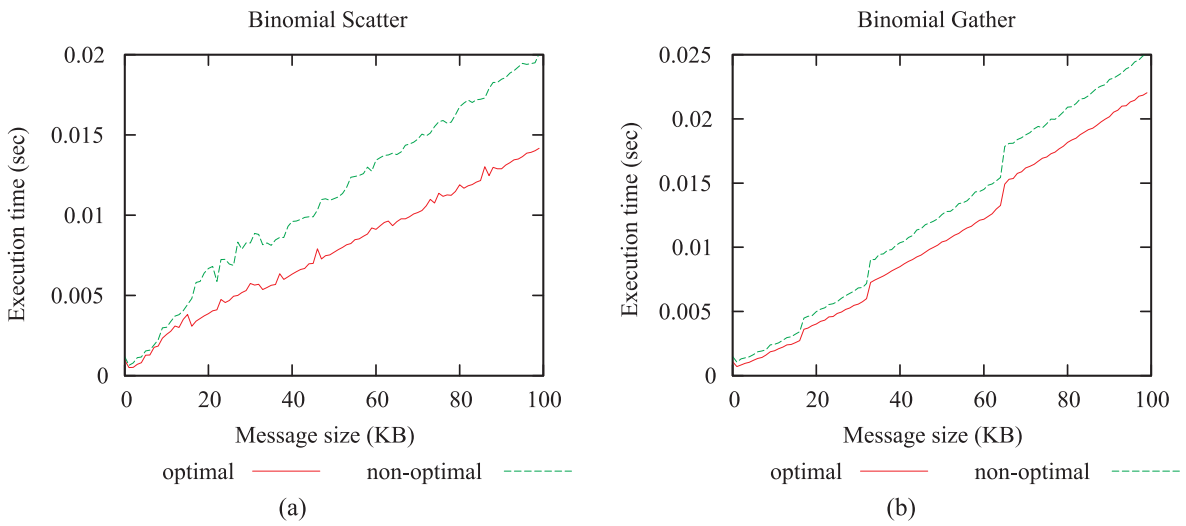

Fig. 2 The execution time of the binomial scatter (a) and gather (b) for two different mappings of the processors of the 16-node heterogeneous cluster to the nodes of the binomial communication tree.

by the heterogeneous Hockney model. Starting with the sub-trees of the highest order, C_k ($k = \log_2 n - 1, \ldots, 1$), through which larger messages are passed, we recursively choose their roots by comparing the Hockney predictions $\alpha_{ri} + \beta_{ri} 2^k \frac{m}{n}$ ($i = 1, \ldots, n-1,\ i \neq r$) for all point-to-point communications $r \xrightarrow{2^k \frac{m}{n}} i$ (scatter) or $r \xleftarrow{2^k M} i$ (gather), where the source, r, is a root of the parent sub-tree (the first r is a root of the whole tree, which is given as an argument of the scatter or gather operation). Figure 2 shows the results of the experiments with these mappings of processors. The algorithm that we used in the experiments does not return the best mapping of the processors to the nodes of the binomial tree. We only presented this algorithm for illustration purposes.

Estimation cost. As we have shown, the heterogeneous models are more accurate than the homogeneous ones but more complicated in use. In particular, algorithms of the optimization of collective communication operations based on the heterogeneous models will be of much higher complexity than their homogeneous counterparts. Do the heterogeneous models also come at a much higher cost of the estimation of their parameters? The answer is "no." Indeed, the cost of the estimation of the model is determined by the number of communication experiments that need to be conducted to collect enough data for accurate approximation of its parameters. The number of parameters in a heterogeneous communication model of an n-processor heterogeneous cluster will be proportional to the number of links, C_n^2, that is, of the order of $O(n^2)$. Given the model is linear, the number of independent communication experiments required to obtain enough data for accurate estimation of the parameters will be proportional to the number of the parameters, that is, of the order of $O(n^2)$. Parameters of a homogeneous model of the heterogeneous cluster are obtained by averaging the corresponding parameters for every pair of processors. Therefore, the number of communication experiments required to estimate the parameters of the homogeneous model will be proportional to the number of pairs, C_n^2, that is, of the order of $O(n^2)$. Thus, the number of communication experiments required for the accurate estimation of both homogeneous and heterogeneous models will be of the same order, $O(n^2)$.

The cost of the accurate estimation of a communication model of the heterogeneous cluster may be quite significant as it typically involves multiple repetitions of the same communication experiments and statistical processing of their results in order to obtain a reliable approximation of the parameters. At the same time, the cost of the estimation can be significantly reduced if the heterogeneous cluster can simultaneously execute several independent communications involving non-overlapping sets of processors without degradation of their performance. In this case, the parallel execution of the non-overlapping communication experiments does not affect the experimental results and can be used for acceleration of the estimation procedure. Our primary target heterogeneous

platform, a heterogeneous cluster based on a single switch, is of that type. The optimization technique can be very efficient. For example, in our experiments on the 16-node heterogeneous cluster, the parallel estimation of the heterogeneous Hockney model with the confidence level 95% and relative error 2.5% took only 5 sec, while its serial estimation with the same accuracy took 16 sec. Both experiments give the same values of the parameters.

Design of communication experiments. The basis of any estimation method is the set of independent communication experiments used to obtain data sufficient to calculate the parameters of the analytical communication model of the heterogeneous cluster. Design of such a set is not an issue for a traditional homogeneous model. In this case, the parameters of the model are estimated for each pair of processors using the traditional point-to-point communication experiments as described in Section 2, and the average values for all pairs are then used in modeling.

The design of this set is also straightforward for a heterogeneous extension of a traditional homogeneous model. The heterogeneous Hockney model and the heterogeneous LogGP model are of this type. Such a heterogeneous model is obtained by straightforward extension of the base homogeneous model such that the communication parameters characterizing each pair of processors can be found from the set of point-to-point experiments described in Section 2.

The design of communication experiments becomes an issue for some original heterogeneous communication models, such as the LMO model. The traditional models use a small number of parameters to describe communication between any two processors. The number of these parameters and their use in the model are always defined in a way that allows for their accurate estimation with a set of point-to-point communication experiments between these two processors. The price to pay is that such a traditional point-to-point communication model is not intuitive. The meaning of its parameters is not clear. Different sources of the contribution to the execution time are artificially and non-intuitively mixed and spread over a smaller number of parameters. This makes the models difficult to use for accurate modeling of collective communications. For example, the Hockney model uses only two parameters to describe communication between two processors. The parameters accumulate contributions of the participating processors and the communication layer into the constant and variable delays respectively. In order to model, say, the linear (flat free) algorithm of scatter on a switched cluster in an intuitive way, we need separate expressions for the contribution of the root processor, the communication layer and each of the receiving processors. Otherwise, we cannot express the serialization of outgoing messages on the root processor followed by their parallel transmission over the communication layer and parallel processing on the receiving processors. The use of the Hockney model as it is results in either ignoring the serialization or ignoring the parallelization. In the former case the predictions will be too optimistic, while in the latter the predictions will be too pessimistic. In both cases, they are not accurate. While using more parameters, the LogGP model faces the same problem because it does not separate the contribution of the processors and the communication layer into the variable delay. The traditional way to cope with this problem is to use an additional (and non-intuitive) fitting parameter, which will make the overall model even less clear. While this approach can somehow work for homogeneous models, it becomes hardly applicable to heterogeneous models. The point is that a heterogeneous model would need multiple fitting parameters making it fully impractical.

The alternative approach is to use original point-to-point heterogeneous models, such as LMO, that allow for easy and intuitive expression of the execution time of collective communication operations. The analytical formula for scatter and gather on heterogeneous clusters provided by the LMO model will be discussed in Section 5. While easy and intuitive in use, these models encounter a new, challenging, problem. The problem is that the number of point-to-point parameters describing communication between a pair of processors becomes larger than the number of independent point-to-point communication experiments traditionally used for estimation of the parameters. In this paper, we address this challenge and propose an approach to design of the set of communication experiments sufficient for the accurate and efficient estimation of the parameters of such heterogeneous communication performance models.

4 Estimation of Parameters of Heterogeneous Communication Performance Models

In brief, in order to estimate the point-to-point parameters of a heterogeneous model when the total number of these parameters makes the point-to-point experiments insufficient, we propose to use additional independent communication experiments involving more than two processors. In more detail, the approach can be summarized as follows:

- As the point-to-point communication experiments do not provide sufficient data for the estimation of the parameters, some particular collective experiments between small numbers of processors (in our experiments, between three processors) are introduced. To make use of the results of these additional experiments, the heterogeneous point-to-point performance model is

extended by an analytical model of these particular collective operations with their execution time expressed via the point-to-point parameters.
- Then, a system of equations with the point-to-point parameters as unknowns and the execution times of the communication experiments as a right hand side is built and solved.
- Since more than two processors are participating in these additional experiments, the execution time should be measured by an appropriate timing method that provides a reasonable balance between the accuracy and efficiency. We propose to measure the execution time of the collective experiments on the sender side. This method has been proven fast and quite accurate for collective operations on a small number of processors.
- The additional collective communication experiments should be designed very carefully in order to avoid the irregularities in the execution time of the used collective operations. We suggest performing a preliminary test of the collective operations for different message sizes to identify the regions of irregularities and avoid the use of message sizes from these regions.
- For reliable estimation of the parameters, we use multiple repetitions of the experiments and the statistical analysis of their results.

As the efficiency of the estimation is an important issue, especially if the model is supposed to be estimated at runtime, we employ the following optimization techniques in the design of the experiments:

- If the target platform is a switched heterogeneous cluster, we will use parallel execution of the non-overlapping experiments.
- If we use, say, triplets of processors for the collective experiments, then a separate system of equations can be built and solved for each triplet. In any complete set of the collective experiments, all processors will participate in more than one triplet. Therefore, the value of some parameters can be found from different independent experiments and hence from different independent systems of equations. We propose to use these redundant values in the statistical analysis to reduce the number of repetitions of the computational experiments needed for reliable estimation of the parameters.

In the next section, we apply this approach to the heterogeneous communication performance model LMO.

5 The LMO Model and Estimation of Its Parameters

In this section, we describe the heterogeneous communication performance model LMO (Lastovetsky et al. 2006) and the design of communication experiments required to estimate its parameters on heterogeneous clusters.

Like most of the point-to-point communication models, its point-to-point parameters represent the communication time by a linear function of the message size. The execution time of sending a message of M bytes from processor i to processor j in a heterogeneous cluster $i \xrightarrow{M} j$ is estimated by $C_i + Mt_i + C_j + Mt_j + \frac{M}{\beta_{ij}}$, where C_i, C_j are the fixed processing delays; t_i, t_j are the delays of processing of a byte; and β_{ij} is the transmission rate. The delay parameters, which are attributed to each processor, reflect the heterogeneity of the processors. The transmission rates correspond to each link and reflect the heterogeneity of communications; for networks with a single switch, it is realistic to assume $\beta_{ij} = \beta_{ji}$.

In terms of the Hockney model, $C_i + C_j = \alpha$ and $Mt_i + Mt_j + \frac{M}{\beta_{ij}} = \beta M$. In comparison with the Hockney model, ours reflects the heterogeneity of processors by introducing different fixed and variable delays. The parameters α of the Hockney model for $i \xrightarrow{M} j$, $j \xrightarrow{M} k$, and $k \xrightarrow{M} i$ point-to-point communications can be used to find fixed processing delays:

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} C_i \\ C_j \\ C_k \end{pmatrix} = \begin{pmatrix} \alpha_{ij} \\ \alpha_{jk} \\ \alpha_{ki} \end{pmatrix} \tag{4}$$

Unfortunately, the data are insufficient to determine variable processing delays and transmission rates, as we have $n + C_n^2$ unknowns but only C_n^2 equations.

In terms of the LogP/LogGP model, the sum of the fixed processing delays $C_i + C_j$ could be equal to $L + 2o$ or $L + 2o - G$, and $Mt_i + Mt_j + \frac{M}{\beta_{ij}} = MG$. Similarly to the Hockney model, the fixed processing delays could be found from every three point-to-point communications,

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} C_i \\ C_j \\ C_k \end{pmatrix} = \begin{pmatrix} (L + 2o - G)_{ij} \\ (L + 2o - G)_{jk} \\ (L + 2o - G)_{ki} \end{pmatrix} \tag{5}$$

but it is not sufficient to find the other parameters.

To estimate the parameters of such a model, an approach with roundtrip point-to-point experiments is not enough. For a network consisting of n processors, there will be $2n + C_n^2$ unknowns: n fixed processing delays, n variable processing delays, and C_n^2 transmission rates. The execution time of the roundtrip, namely sending M_1

bytes and receiving M_2 bytes between nodes $i \underset{M_2}{\overset{M_1}{\rightleftarrows}} j$, is equal to:

$$T_{ij}(M_1, M_2) = \left(C_i + M_1 t_i + C_j + M_1 t_j + \frac{M_1}{\beta_{ij}}\right) + \left(C_i + M_2 t_i + C_j + M_2 t_j + \frac{M_2}{\beta_{ij}}\right) \quad (6)$$

The roundtrip experiments will give us only C_n^2 equations. Therefore, the first challenge we face is to find a set of experiments that gives a sufficient number of independent linear equations, whose variables represent the unknown point-to-point parameters.

First, we measure the execution time of the roundtrips with empty messages between each pair of processors $i < j$ (C_n^2 experiments). The fixed processing delays can be found from $T_{ij}(0) = 2C_i + 2C_j$ solved for every three roundtrips $i \underset{0}{\overset{0}{\rightleftarrows}} j$, $j \underset{0}{\overset{0}{\rightleftarrows}} k$, $k \underset{0}{\overset{0}{\rightleftarrows}} i$ ($i < j < k$):

$$\begin{cases} T_{ij}(0) = 2C_i + 2C_j \\ T_{jk}(0) = 2C_j + 2C_k \\ T_{ki}(0) = 2C_k + 2C_i \end{cases} \quad (7)$$

Since C_i can be found from the systems for different j and k, it makes sense to take C_i as an average of $\frac{(n-1)(n-2)}{2} = C_{n-1}^2$ values obtained from all the different systems of equations.

In order to find the remaining $n + C_n^2$ parameters, we might use the roundtrips with non-empty message, but it would give us only C_n^2 linearly independent equations. Instead, we use the additional experiments, which include communications from one processor to two others and backward, and express the execution time of the communication experiments in terms of the heterogeneous point-to-point communication performance model. As will be shown below, the set of point-to-point and point-to-two communication experiments is enough to find the fixed processing delay and transmission rates, but there is one more important issue to be addressed. The point-to-two experiments are actually a particular combination of linear (flat tree) scatter and gather. The linear scatter and gather operations may have some irregular behavior on the clusters based on a switched network, especially if the MPI software stack includes the TCP/IP layer. Therefore, the message sizes for the additional experiments have to be carefully selected to avoid these irregularities.

We observed the leap in the execution time of linear scatter for large messages and the non-deterministic escalations of the execution time of linear gather for medium-sized messages. In Figure 3, typical graphs of the communication execution time are shown. These phenomena occur for any number of processors (even for three), with the thresholds constant and the values/frequencies of escalations dependent on number of processors. It prompted us to introduce the particular threshold parameters to categorize the message size ranges where distinctly different behavior of the collective MPI operations is observed, and to apply different formula for these regions to express the execution time with the heterogeneous point-to-point parameters. The scatter performance in Figure 3(a) is presented on a smaller scale to emphasize the leap in the communication execution time. The execution time of scatter is less than the execution time of gather; the linear parts of their graphs have different slopes.

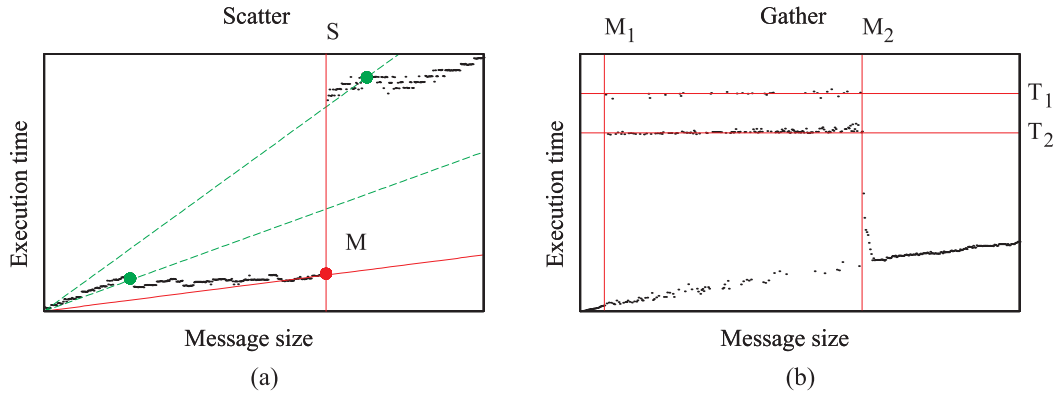

Fig. 3 The execution time of scatter (a) and gather (b) against the message size. The scales of the execution time axes are different.

Let us consider how the LMO model can be used to express the execution time of linear scatter and gather in an intuitive way. The execution time is expressed with the help of the point-to-point parameters of the LMO model (analytical part) and some extra parameters that reflect the irregularities observed on a switched cluster (empirical part). The estimated time of scattering messages of size M from node 0 to nodes 1, 2, …, n is given by:

$$\begin{cases} n(C_0 + t_0M) + \max\limits_{1 \le i \le n}\left\{C_i + t_iM + \dfrac{M}{\beta_{0i}}\right\} & M \le S \\ n(C_0 + t_0M) + \sum\limits_{i=1}^{n}\left(C_i + t_iM + \dfrac{M}{\beta_{0i}}\right) & M > S \end{cases} \quad (8)$$

where C_0, t_0, C_i, and t_i are the fixed and variable processing delays on the source node and destinations. The threshold parameter, S, corresponds to the leap in the execution time, separating small and large messages. It is a constant that may vary for different combinations of clusters and MPI implementations. The sequential part of this formula, $n(C_0 + t_0M)$, is related to the root processor, which consecutively processes the messages to be sent to the rest $n - 1$ processors. The maximum reflects the parallel transmissions followed by the parallel processing on the receivers. Therefore, this formula conforms with the features of network switches, which parallelize the messages addressed to different processors. The large messages, in contrast, are serialized, which is expressed by the sum of point-to-point contributions.

For the gather operation, we separate small, medium, and large messages by introducing parameters M_1 and M_2. Like S, these parameters depend on the particular cluster and MPI implementation. For example, on the 16-node heterogeneous cluster specified in Table 1, we observed $M_1 = 4$ KB, $M_2 = 65$ KB for LAM 7.1.3 and $M_1 = 3$ KB, $M_2 = 125$ KB for MPICH 1.2.7. For small messages, $M < M_1$, the execution time has a linear response to the increase of message size. Thus, the execution time for the many-to-one communication is estimated by:

$$n(C_0 + t_0M) + \max_{1 \le i \le n}\left\{C_i + t_iM + \frac{M}{\beta_{0i}}\right\} + \kappa_1 M \quad (9)$$

where $\kappa_1 = const$ is a fitting parameter for correction of the slope. For large messages, $M > M_2$, the execution time resumes a linear predictability with increasing message size. Hence, this part is similar in design but has a different slope of linearity that indicates greater values due to overheads:

$$(n-1)(C_0 + t_0M) + \sum_{i=1}^{n}\left(C_i + t_iM + \frac{M}{\beta_{0i}}\right) + \kappa_2 M \quad (10)$$

The additional parameter $\kappa_2 = const$ is a fitting constant for correction of the slope. For medium messages, $M_1 \le M \le M_1$, we observed a small number of discrete levels of escalation, that remain constant as the message size increases.

Thus, following the models of scatter and gather, in our experiments we gather zero-sized messages in order to avoid the non-deterministic escalations. For scatter, the message size M is taken as less than the value of the threshold parameter S. The wrong selection of the message size can make the estimation of the point-to-point parameters inaccurate, which is shown in Figure 3(a).

In order to find variable processing delays, t_i, and transmission rates, β_{ij}, we measure the execution time of the C_n^2 experiments $i \xleftarrow[0]{M} j$ $(i < j)$, the roundtrips with empty replies, and the C_n^3 experiments $i \xleftarrow[0]{M} j, k$ $(i < j < k)$, where the source processor sends the messages of the same size to two processors and receives zero-sized messages from them. In $i \xleftarrow[0]{M} j, k$ communication, the contribution of the source node in the execution time will be $4C_i + 2Mt_i$. The total time of transmission and processing on the destinations will be equal to the maximal value among the destination processors $\max\left\{\left(2C_j + Mt_j + \frac{M}{\beta_{ij}}\right), \left(2C_k + Mt_k + \frac{M}{\beta_{ik}}\right)\right\}$. Thus, the execution time $T_i(M)$ of one-to-two communications with root i can be expressed by:

$$T_i(M) = 4C_i + 2Mt_i + \max\left\{\left(2C_j + Mt_j + \frac{M}{\beta_{ij}}\right), \left(2C_k + Mt_k + \frac{M}{\beta_{ik}}\right)\right\} \quad (11)$$

Let τ_j denote $2C_j + Mt_j + \frac{M}{\beta_{ij}}$. Removing the maximum and rewriting the equation, we get:

$$\begin{cases} 2t_i + t_j + \dfrac{1}{\beta_{ij}} = \dfrac{T_i(M) - 4C_i - 2C_j}{M}, & \tau_j > \tau_k \\ 2t_i + t_k + \dfrac{1}{\beta_{ik}} = \dfrac{T_i(M) - 4C_i - 2C_k}{M}, & \tau_k > \tau_j \end{cases} \quad (12)$$

Both alternatives, less the equations for the point-to-point roundtrips with empty reply $i \xleftarrow[0]{M} j$ and $i \xleftarrow[0]{M} k$, will give us the expression for the variable processing delay:

$$t_i = \begin{cases} \dfrac{T_i(M) - T_{ij}(M) - 2C_i}{M}, & \tau_j > \tau_k \\ \dfrac{T_i(M) - T_{ik}(M) - 2C_i}{M}, & \tau_k > \tau_j \end{cases} \quad (13)$$

where $T_{ij}(M)$ and $T_{ik}(M)$ are the execution times of the roundtrips. The inequalities can be simplified by adding $2C_i + Mt_i$ to both sides; the condition $\tau_j > \tau_k$ will become $T_{ij}(M) > T_{ik}(M)$. For the communications with other roots $j \xleftarrow[0]{M} i, k$, $k \xleftarrow[0]{M} i, j$, there will be similar expressions for t_j and t_k:

$$t_j = \begin{cases} \dfrac{T_j(M) - T_{ji}(M) - 2C_j}{M}, & T_{ji}(M) > T_{jk}(M) \\ \dfrac{T_j(M) - T_{jk}(M) - 2C_j}{M}, & T_{jk}(M) > T_{ji}(M) \end{cases} \quad (14)$$

$$t_k = \begin{cases} \dfrac{T_k(M) - T_{ki}(M) - 2C_k}{M}, & T_{ki}(M) > T_{kj}(M) \\ \dfrac{T_k(M) - T_{kj}(M) - 2C_k}{M}, & T_{kj}(M) > T_{ki}(M) \end{cases} \quad (15)$$

We assume $\beta_{ij} = \beta_{ji}$, therefore, $T_{ij}(M) > T_{ji}(M)$. All we need is to compare the values of $T_{ij}(M)$, $T_{jk}(M)$, $T_{ik}(M)$ and select the equations that satisfy the conditions. Then, the transmission rates can be expressed as $\dfrac{1}{\beta_{ij}} = \dfrac{T_{ij}(M) - 2C_i - 2C_j}{M} - t_i - t_j$. Thus, we have six equations with three conditions. For example, if $T_{ij}(M) > T_{ik}(M)$, $T_{ji}(M) > T_{jk}(M)$, $T_{ki}(M) > T_{kj}(M)$ then the system of equations will look as follows:

$$\begin{cases} t_i = \dfrac{T_i(M) - T_{ij}(M) - 2C_i}{M} \\ t_j = \dfrac{T_j(M) - T_{ji}(M) - 2C_j}{M} \\ t_k = \dfrac{T_k(M) - T_{ki}(M) - 2C_k}{M} \\ \dfrac{1}{\beta_{ij}} = \dfrac{T_{ij}(M) - 2C_i - 2C_j}{M} - t_i - t_j \\ \dfrac{1}{\beta_{jk}} = \dfrac{T_{jk}(M) - 2C_j - 2C_k}{M} - t_j - t_k \\ \dfrac{1}{\beta_{ki}} = \dfrac{T_{ki}(M) - 2C_k - 2C_i}{M} - t_k - t_i \end{cases} \quad (16)$$

If $i < j < k$, there will be $3C_n^3$ one-to-two experiments. The variable processing delays, t_i, can be obtained from C_{n-1}^2 different triplets, the processor i takes part in, and can be averaged. The transmission rates β_{ij} can be averaged from $n - 2$ values.

The design described in this section is optimal in terms of the execution time taken for estimation of the point-to-point parameters. The total execution time depends on:

- the number of measurements ($2C_n^2$ one-to-one and $3C_n^3$ one-to-two measurements),
- the execution time of every single measurement (fast roundtrips between 2 and 3 processors), and
- the complexity of calculations ($3C_n^3$ comparisons, $12C_n^3$ simple formulae for calculation of the values of the parameters of the model, and $2n + C_n^2$ averagings).

As the parameters of our point-to-point model are found in a small number of experiments, they can be sensitive to the inaccuracies of measurement. Therefore, it makes sense to perform a series of the measurements for one-to-one and one-to-two experiments and to use the averaged execution times in the corresponding linear equations.

Minimization of the total execution time of the experiments is another issue that we address. The advantage of the proposed design is that these series do not have to be lengthy (typically, up to 10 in a series) because all the parameters have been already averaged with the process of their finding.

The procedure of the estimation of the point-to-point parameters is preceded by the estimation of the threshold parameters. These parameters are used to select the message size for the one-to-two communication experiments and to adjust the prediction of the execution time of collective operations. To estimate the threshold parameters, we use the scatter and gather benchmarks for different message sizes. The data rows for scatter and gather consist of the message sizes taken with some stride and the measured execution time $\{M^i, T^i\}$, $M^{i+1} = M^i + stride$. Typical data rows for heterogeneous clusters based on a switched network are shown in Figure 3. One can see that:

- The execution time of scatter can be approximated by the piecewise linear function with one break that correspond to the threshold parameter S to be found (Figure 3a).
- The execution time of gather has the regions of linearity for small, $M < M_1$, and large, $M > M_2$, messages (Figure 3b) and can also be approximated by the two linear functions.

To find the threshold parameters, we use the algorithm proposed by Bai and Perron (2003) and implemented in

the statistical package Strucchange (Zeileis et al. 2002). It considers the statistical linear models with multiple structural changes and uses dynamic programming to identify optimal partitions with different numbers of segments. The algorithm allows us to locate the break in the execution time of scatter, S, and the range of large messages for gather, M_2. Then we perform the linear regression of the execution time of gather on this range to estimate the slope correction parameter, κ_2, that is used to adjust the prediction of many-to-one execution time for large messages. The linear regression gives us two values c_0 and c_1: $T \approx c_0 + c_1 M, M > M_2$. The slope correction parameter, κ_2, is found as follows: $\kappa_2 = c_1 - \sum_{i=1}^{n-1}\left(t_i + \frac{1}{\beta_{0i}}\right)$.

To separate small and large messages for the gather data row, we use the following algorithm. First, we find the minimum message size for which the escalation of the execution time is observed $M_1 \approx M^k$, $k = \min\{i | T^{i+1}/T^i > 10\}$. The escalation is defined as the difference between the execution times, measured for two neighboring message sizes in the row, that has a 10 times scale of magnitude. Then, an additional gather benchmark is performed for the messages $M \le M^k$ taken with the stride two times less than in the previous benchmark. The new data row $\{M^i, T^i\}$, $M^{i+1} = M^i + stride/2$ is used in the next iteration of the algorithm to find more accurate value of M_1. This is repeated until the stride reaches 1 byte. After M_1 is found, the linear regression on the last data row is performed to obtain the linear parameters for the small messages, $T \approx c_0 + c_1 M$, $M < M_1$, and to calculate the slope $\kappa_1 = c_1 - \max_{i=1}^{n-1}\left\{t_i + \frac{1}{\beta_{0i}}\right\}$.

6 Experimental Evaluation

In this section, we present the experimental results demonstrating that the proposed technique allows us to estimate the parameters of the heterogeneous point-to-point communication model accurately and efficiently. By the example of parallel matrix multiplication, we show that the model-based optimization of collective operations can significantly improve the performance of MPI applications. We carried out the experiments with various MPI implementations and different clusters. This paper presents the experimental results obtained on the 16-node heterogeneous cluster described in Section 3.

We measured the execution time of point-to-point communications on a 16-node heterogeneous cluster and compared it with the predictions provided by the heterogeneous LogGP, PLogP, and LMO models. For the measurements, we used the MPIBlib, the MPI benchmarking library, which provides accurate and efficient benchmarking of MPI communication operations (Lastovetsky, O'Flynn, and Rychkov 2008). The LogGP and PLogP parameters were found for all pairs of processors with help of the logp_mpi library (Kielmann et al. 2000). In Figure 4, the results for one pair are shown. The linear predictions provided by the LogGP and LMO models are almost the same and acceptably accurate for small and medium sized messages. The PLogP point-to-point model is piecewise linear. It includes a large amount of empirical data stored in the functional parameters, and reflects the deviations of the execution time from the linear predictions.

All point-to-point models considered in this paper use a large number of measurements and very simple computations. Therefore, the measurements are the most time-consuming part in the finding of the parameters of these models. In Table 3, the number of measurements is estimated for each model and the time they take on the 16-node heterogeneous cluster is shown.

In Table 3, we compare the measurement costs of the point-to-point models of a heterogeneous cluster. For a cluster of n processors there will be C_n^2 single point-to-point communications. The parameters of the Hockney model for a single point-to-point communication are found by linear regression of k execution times of the roundtrips with different message sizes. Larger k provides a more accurate prediction. The execution time of each measurement depends on the message size. In our experiments, we used 10 message sizes ranging from 0 to 100 kb.

Estimation of the PLogP parameters for each pair of processors includes:

- s experiments on saturating the link by empty messages, the ith experiment of which consists of 2^i sendings, and
- $2mr$ experiments on $i \underset{0}{\overset{M}{\rightleftarrows}} j$ and $i \underset{M}{\overset{0}{\rightleftarrows}} j$ roundtrips, where:
  - r is the number of roundtrips required to obtain more accurate send and receive overheads (the averaged execution time of the roundtrips $i \underset{0}{\overset{M}{\rightleftarrows}} j$ is also used for estimation of $g(M)$), and
  - m is the number of message sizes, necessary for accurate piecewise linear approximation of the execution time of point-to-point communication.

The numbers s, r, and m are found experimentally and can be different for different pairs of processors. In formulae in Table 3 that estimate the total number of measurements, we use the averaged values of s, r, and m. The saturation experiments take much more time than single roundtrips as they include up to 2^s sendings. The direct

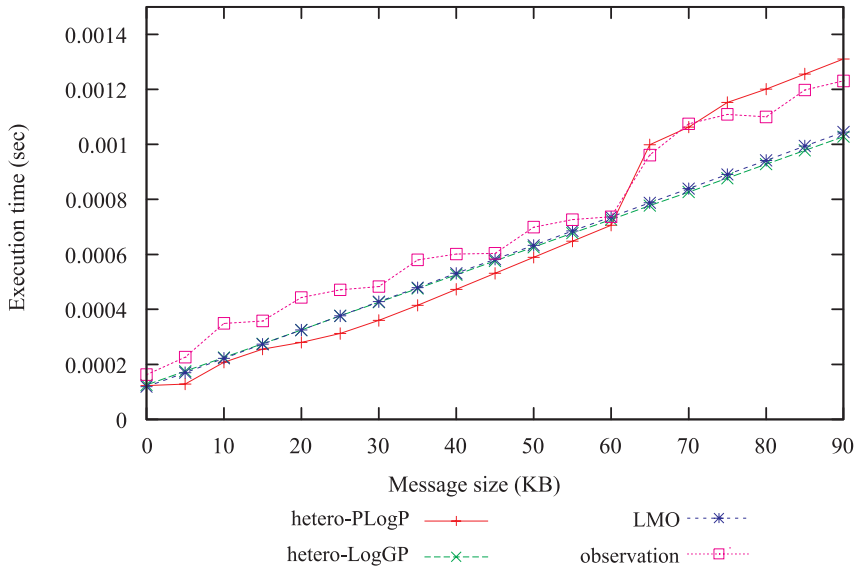

Fig. 4 Comparison of the predictions of the point-to-point models for a single point-to-point communication.

Table 3
Measurement costs of heterogeneous models on 16-node heterogeneous cluster.

Communication model	Number of measurements	Execution time, s
Hetero-Hockney	kC_n^2	0.17
Hetero-LogGP	$3sC_n^2 + 2rC_n^2$	–
Hetero-PLogP	$sC_n^2 + 2mrC_n^2$ or $msC_n^2 + 2mrC_n^2$ (direct)	63.11
LMO	$k_0C_n^2 + k_1C_n^2 + k_23C_n^3$	0.33

measurements of the gap for each message size require $(m-1)s$ more experiments.

The LogGP model requires three saturation processes with message of 0, 1, and M bytes to estimate the gaps per message/byte and two roundtrips with the message of 1 byte to estimate the send/receive overheads. The execution time of the estimation of the parameters is omitted because they were found via the PLogP parameters as described in Kielmann et al. (2000).

The accuracy in our heterogeneous communication point-to-point model is achieved by averaging the execution times in:

- a series of the k_0 measurements for each of C_n^2 empty roundtrips,
- a series of the k_1 measurements for each of C_n^2 non-empty roundtrips, and
- a series of the k_2 measurements for each of $3C_n^3$ one-to-two communications.

In our experiments, no more than 10 measurements in a series were needed to achieve the acceptable accuracy.

We showed that the LMO model provides the acceptable accuracy and can be efficiently estimated on heterogeneous clusters. In the next section we consider how this model can be applied to the optimization of collective communications in parallel applications.

7 Model-Based Optimization of Parallel Applications on Heterogeneous Clusters

The use of advanced intuitive heterogeneous communication models for optimization of communication operations has a potential to improve their performance (and hence the performance of the corresponding applica-

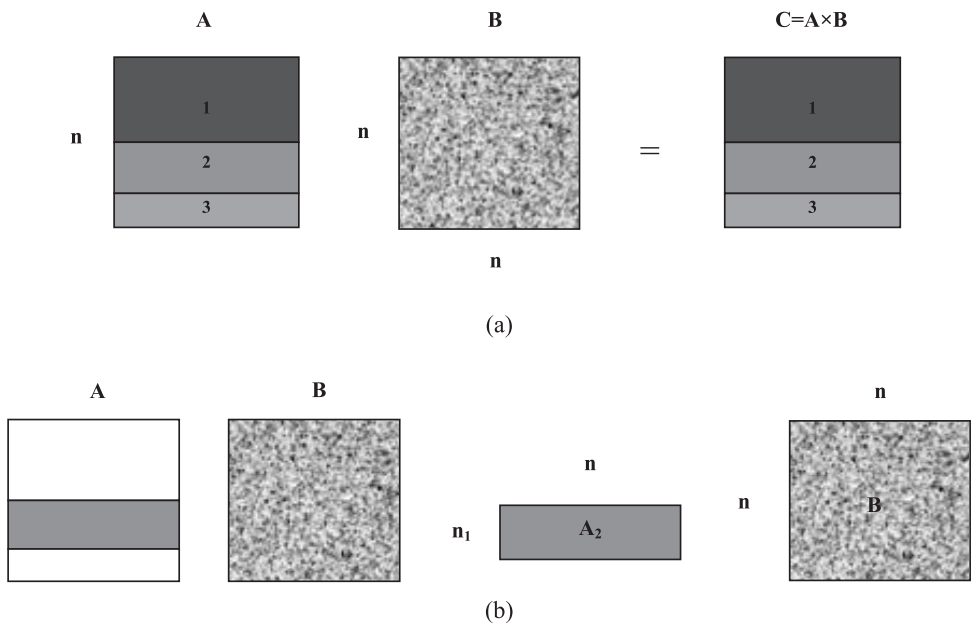

Fig. 5 (a) Matrix operation $C = A \times B$ with matrices A, B, and C. Matrices A and C are horizontally sliced such that the number of elements in the slice is proportional to the speed of the processor. (b) Serial matrix multiplication $A_2 \times B$ of dense matrix A_2 of size $n_1 \times n$ and dense matrix B of size $n \times n$ to estimate the absolute speed of processor 2.

tions) on heterogeneous computational clusters. However, utilization of this potential is conditioned by the ability to accurately and efficiently estimate the parameters of these models. Given the parameters are estimated inaccurately, implementation of the optimization algorithms in applications running on heterogeneous clusters will not have a positive effect on their performance. In this paper, we have proposed an efficient solution to the problem of accurate estimation of the parameters of the advanced heterogeneous communication performance models. This gives us an opportunity to apply the models to the optimization of MPI collective operations on heterogeneous clusters.

In this section, we show that the use of the heterogeneous communication models can significantly improve the performance of parallel applications on heterogeneous clusters. As a sample application, we use parallel matrix multiplication, a simple but important linear algebra kernel representative for many real-life scientific applications. We use the LMO heterogeneous communication performance model for optimization of the MPI broadcast, scatterv, and gatherv collective operations used in this application. The analytical prediction of the execution time with this model is used for the optimal mapping of the heterogeneous processors to the nodes of the corresponding communication trees. We compare the performance of the parallel application using the optimized versions of these collective operations against the one using their native MPI implementations.

Our sample application is based on the master–slave paradigm. The master process distributes the data between the slave processes, coordinates the computations, and collects the result. This type of parallel application is often used in practice, for example, in processing of a large amount of image data collected from the hyperspectral sensors on airborne/satellite platforms (Plaza et al. 2006). Our application multiplies two dense square matrices, $C = A \times B$, and employs a simple heterogeneous parallel algorithm based on one-dimensional matrix partitioning (see, for example, Lastovetsky and Reddy 2007). As shown in Figure 5, the matrices A and C are horizontally sliced such that the number of elements in a slice is proportional to the speed of the processor owning the slice. All the processors contain all the elements of matrix B. We assume a one process per processor configuration. With this one-dimensional partitioning, our application performs: (1) irregular scatter of matrix A, (2) broadcast of matrix B, (3) parallel multiplications, and (4) irregular gather of matrix C.

We ran this application on the heterogeneous cluster (see Section 3 for specifications) in two modes. The first uses native MPI collective operations. The second uses

Table 4
Matrix multiplication on 16-node heterogeneous cluster.

Matrix size, $N \times N$	With native collectives, s	With optimized collectives, s
1000	0.80	0.65
2000	5.51	5.02
3000	16.61	11.05
4000	74.29	64.79
5000	221.00	188.72
6000	494.24	394.65

our implementation of these operations optimized for the cluster. The optimization is based on the LMO model of the cluster, whose parameters were estimated using the presented techniques. The results of these experiments are given in Table 4. One can see that the performance of the application was improved by up to 20% as a result of the optimization of communications.

We are not going into detail of the optimization algorithms, which is a subject of another independent research paper. The main idea is to use non-flat communication trees and to perform their optimal mapping based on the analytical predictions of the LMO model. The native scatterv and gatherv are traditionally implemented using a flat tree (in all known MPI implementations). They are exposed to the escalations of the execution time similarly to their regular counterparts, scatter and gather. The use of the non-flat trees allowed us to eliminate these escalations. The optimal mapping of these trees allowed us to significantly reduce the execution time of these operations (up to 40% compared with their native counterparts). The native broadcast is implemented by the binomial algorithm. Our version uses the LMO model for optimal mapping of the nodes of the broadcast communication tree onto the processors of the cluster.

8 Conclusion

In this paper, we have described an efficient technique for accurate estimation of parameters of the point-to-point communication performance model of heterogeneous clusters based on a switched network. This technique includes a relatively small number of measurements of the execution time of one-to-one and one-to-two round-trip communications for some particular message sizes and solution of simple systems of linear equations. The accuracy of estimation is achieved by careful selection of message sizes and averaging the values of the parameters. The efficiency and accuracy of the proposed technique were validated experimentally. The performance gains due to the use of the model for optimization of MPI collective operations were demonstrated for the parallel matrix multiplication application.

Acknowledgments

This work is supported by the Science Foundation Ireland and in part by the IBM Dublin CAS.

Author Biographies

Alexey Lastovetsky received a Ph.D. degree from the Moscow Aviation Institute in 1986, and a Doctor of Science degree from the Russian Academy of Sciences in 1997. His main research interests include algorithms, models, and programming tools for high performance heterogeneous computing. He is the author of mpC, the first parallel programming language for heterogeneous networks of computers. He designed HeteroMPI, an extension of MPI for heterogeneous parallel computing, and SmartNetSolve/SmartGridSolve, an extension of NetSolve/GridSolve aimed at higher performance of scientific computing on global networks. He has also made contributions to heterogeneous data distribution algorithms and modeling the performance of processors in heterogeneous environments. He has published over 90 technical papers in refereed journals, edited books, and international conferences. He authored the monographs "Parallel computing on heterogeneous networks" (Wiley, 2003) and "High performance heterogeneous computing" (with Jack Dongarra, Wiley, 2009). He is currently a senior lecturer in the School of Computer Science and Informatics at the University College Dublin, National University of Ireland. At UCD, he also created and leads the Heterogeneous Computing Laboratory.

Vladimir Rychkov received his Ph.D. degree from the Russian Academy of Sciences in 2005. His main research interests include the design of algorithms and tools for parallel and distributed computing systems, mathematical modeling and numerical methods, computer aided design, and engineering.

References

Alexandrov, A., Ionescu, M., Schauser, K., and Scheiman, C. (1997). LogGP: incorporating long messages into the LogP model for parallel computation, *J. Parallel and Distributed Computing*, **44**(1): 71–79.

Arnold, D., Casanova, H., and Dongarra, J. (2002). Innovation of the NetSolve grid computing system, *Concurrency and Computation: Practice and Experience*, **14**(13–15): 1457–1479.

Bai, J. and Perron, P. (2003). Computation and analysis of multiple structural change models, *J. Appl. Econometrics*, **18**(1): 1–22.

Bhat, P., Prasanna, V., and Raghavendra, C. (2003). Efficient collective communication in distributed heterogeneous systems, *J. Parallel and Distributed Computing*, **63**(3): 251–263.

Culler, D., Karp, R., Patterson, D., Sahay, A., Schauser, K., Santos, E., Subramonian, R., and Eicken, T. von. (1993). LogP: Towards a realistic model of parallel computation. In *Proc. of PPoPP 1993*, pp. 1–12. New York: ACM.

Culler, D., Liu, L., Martin, R., and Yoshikawa, C. (1996). LogP performance assessment of fast network interfaces, *IEEE Micro*, **16**(1): 35–43.

Hatta, J. and Shibusawa, S. (2000). Scheduling algorithms for efficient gather operations in distributed heterogeneous systems. In *Proc. of WPP 2000*, pp. 173–180. Los Alamitos, CA: IEEE Computer Society.

Hockney, R. (1994). The communication challenge for MPP: Intel Paragon and Meiko CS-2, *Parallel Comput.*, **20**(3): 389–398.

Kielmann, T., Bal, H., and Verstoep, K. (2000). Fast measurement of LogP parameters for message passing platforms. In *Proc. of IPDPS 2000*, pp.1176–1183. Berlin: Springer.

Kielmann, T., Hofman, R., Bal, H., Plaat, A., and Bhoedjang, R. (1999). MagPIe: MPI's collective communication operations for clustered wide area systems. In *Proc. of PPoPP 1999*, pp. 131–140. New York: ACM.

Lastovetsky, A. (2002). Adaptive parallel computing on heterogeneous networks with mpC, *Parallel Comput.*, **28**(10): 1369–1407.

Lastovetsky, A. and O'Flynn, M. (2007). A performance model of many-to-one collective communications for parallel computing. In *Proc. of IPDPS 2007*, pp. 1–8. Los Alamitos, CA: IEEE Computer Society.

Lastovetsky, A. and Reddy, R. (2006). HeteroMPI: Towards a message-passing library for heterogeneous networks of computers, *J. Parallel and Distributed Computing*, **66**(2): 197–220.

Lastovetsky, A. and Reddy, R. (2007). Data partitioning with a functional performance model of heterogeneous processors, *Int. J. High Performance Computing Applications*, **21**(1): 76–90.

Lastovetsky, A., Mkwawa, I., and O'Flynn, M. (2006). An accurate communication model of a heterogeneous cluster based on a switch-enabled Ethernet network. In *Proc. of ICPADS 2006*, pp. 15–20. Los Alamitos, CA: IEEE Computer Society.

Lastovetsky, A., O'Flynn, M., and Rychkov, V. (2008). MPIBlib: Benchmarking MPI communications for parallel computing on homogeneous and heterogeneous clusters. In *Proc. of EuroPVM/MPI 2008*, pp. 227–238. Berlin: Springer.

Pjesivac-Grbovic, J., Angskun, T., Bosilca, G., Fagg, G., Gabriel, E., and Dongarra, J. (2007). Performance analysis of MPI collective operations, *Cluster Computing*, **10**(2): 127–143.

Plaza, A., Valencia, D., Plaza J., and Martinez, P. (2006). Commodity cluster-based parallel processing of hyperspectral imagery, *J. Parallel and Distributed Computing*, **66**(3): 345–358.

Thakur, R., Rabenseifner, R., and Gropp, W. (2005). Optimization of collective communication operations in MPICH, *Int. J. High Performance Computing Applications*, **19**(1): 49–66.

Zeileis, A., Leisch, F., Hornik, K., and Kleiber, C. (2002). Strucchange: An R package for testing for structural change in linear regression models, *J. Statistical Software*, **7**(2): 1–38.