Revisiting communication performance models for computational clusters

Alexey Lastovetsky, Vladimir Rychkov, and
Maureen O'Flynn
School of Computer Science and Informatics
University College Dublin
Dublin, Ireland
{alexey.lastovetsky, vladimir.rychkov, maureen.oflynn}@ucd.ie

Abstract—**In this paper, we analyze restrictions of traditional models affecting the accuracy of analytical prediction of the execution time of collective communication operations. In particular, we show that the constant and variable contributions of processors and network are not fully separated in these models. Full separation of the contributions that have different nature and arise from different sources will lead to more intuitive and accurate models, but the parameters of such models cannot be estimated from only the point-to-point experiments, which are usually used for traditional models. We are making the point that all the traditional models are designed so that their parameters can be estimated from a set of point-to-point communication experiments. In this paper, we demonstrate that the more intuitive models allow for much more accurate analytical prediction of the execution time of collective communication operations on both homogeneous and heterogeneous clusters. We present in detail one such a point-to-point model and how it can be used for prediction of the execution time of scatter and gather. We describe a set of communication experiments sufficient for accurate estimation of its parameters, and we conclude with presentation of experimental results demonstrating that the model much more accurately predicts the execution time of collective operations than traditional models.**

Computational cluster, MPI, communication performance model, analytical prediction of execution time, estimation of parameters

I. INTRODUCTION

Analytical communication performance models play an important role in optimization of parallel applications on computational clusters. Traditional communication models, such as the Hockney model [6], LogP [4], LogGP [1], and PLogP [7], are often used for estimation of the execution time of different algorithms of MPI collective communication operations on homogeneous clusters. For example, Chan et al. [3] and Thakur et al. [15] applied the Hockney model to compare the communication cost of different algorithms of the same collective operation in order to choose the fastest one for different message sizes and numbers of processors. Pjesivac-Grbovic et al. [14] showed that the estimations provided by the traditional models might differ from the observed communication execution times and result in non-optimal switch between algorithms.

In the case of heterogeneous clusters, there is another application of communication performance models to the optimization of MPI collective operations. Namely, the performance of a collective operation can be improved by the optimal mapping of heterogeneous processors to the nodes of the communication tree of the operation. Traditional communication performance models are usually homogeneous, with parameters having the same values for all processors and links. Therefore, they provide the same prediction for any mapping. Heterogeneous communication models do distinguish the contributions of different links and processors and hence may be used for this purpose. Hatta et al. [5] built optimal communication trees for collective operations with help of a simple heterogeneous extension of the Hockney model.

The accuracy of the analytical prediction of communication execution time depends on the choice of such a communication performance model that is the most appropriate to the targeted platform and allows for easy and natural expression of different algorithms of collective operations. Our target platform is a homogeneous or heterogeneous cluster with a single switch. An ideal intuitive communication performance model for this platform should have the following features:

- It is based on the point-to-point parameters that reflect ***constant*** and ***variable*** contributions of ***processors*** and ***network***. Full separation of the contributions that have a different nature and arise from different sources will lead to more intuitive analytical expressions of the communication execution time. In traditional communication performance models, the constant and variable contributions of processors and network are not fully separated.
- The execution time of any collective communication operation can be presented by a combination of ***maximums*** (parallel part) and ***sums*** (sequential part) of the point-to-point parameters. The formula of the execution time can include extra parameters that reflect the irregular behaviour of the collective operation and that are found empirically for a particular platform. Traditional models do not include such empirical parameters.
- There is a set of communication experiments that allows for the accurate estimation of the parameters.

Traditional models are designed so that their parameters can be estimated from the point-to-point communication experiments. The attempts to separate the contributions lead to a model whose parameters cannot be estimated from only the point-to-point experiments.

In this paper, we present a modification of the advanced communication performance model, LMO [8, 9], that fully separates the constant and variable contributions of processors and network. We suggest an approach to the design of the communication experiments required to estimate the parameters of the elaborated models and describe a set of communication experiments for the LMO model. We conclude with experimental results demonstrating that the LMO model much more accurately predicts the execution time of collective operations than traditional models.

This paper is organized as follows. In Section II, we discuss traditional communication performance models. In Sections III and IV, we describe the modification of the LMO model and the design of communication experiments required to estimate its parameters. In Section V, we compare the predictions provided by traditional and advanced models.

II. TRADITIONAL COMMUNICATION PERFORMANCE MODELS

In this section, we analyze the limitations of traditional communication performance models, preventing them from accurate estimation of the execution time of collective communication operations on computational clusters with a single swich.

Usually, 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 collective operations is expressed as a combination of the point-to-point parameters and predicted for different message sizes and numbers of processors. For homogeneous clusters, the point-to-point parameters are found statistically from the measurements of the execution time of communications between any two processors. Typical experiments include sending and receiving messages of different sizes, with the communication execution time being measured on one side.

Traditional communication performance models can be applied to heterogeneous clusters 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. Another way is the heterogeneous extension of traditional models, when different pairs of heterogeneous processors are characterized by different parameters. The small number of parameters is an obvious advantage of the first approach. It allows the expression of the execution time of any communication operation by a simple compact formula, which is independent on the processors involved in the operation. While simpler in use, the homogeneous models are less accurate than the heterogeneous ones. When some processors or links in the heterogeneous cluster significantly differ in performance, predictions based on the homogeneous models may become quite inaccurate. The number of communication experiments required for the accurate estimation of both homogeneous and heterogeneous models will be of the same order, $O(n^2)$, where n is a number of processors in the cluster.

Let us start with a traditional model proposed by Hockney [6]. The parameters of the Hockney model combine the processor and network contributions. The execution time of point-to-point communication is expressed as $\alpha+\beta M$, where α is the latency (constant contributions from processors and network), β is the bandwidth (variable contributions from processors and network) and M is the message size. The Hockney parameters are estimated with help of series of the point-to-point communications in one of two ways:

- 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): $\left\{ i \underset{0}{\overset{0}{\rightleftarrows}} j \quad i \underset{M}{\overset{M}{\rightleftarrows}} j \right\}_{k=0}^{R}$, or
- A series of roundtrips with messages of different sizes (to perform a linear regression, which fits the execution time into a linear combination of the Hockney parameters and a message size): $\left\{ i \underset{M_k}{\overset{M_k}{\rightleftarrows}} j \right\}_{k=0}^{R}$.

We can extend the Hockney model for heterogeneous clusters and introduce different parameters α_{ij} and β_{ij} for different pairs of processors, which also combine the processor and network contributions. In order to estimate the parameters of both original and extended models for a heterogeneous cluster, the above communication experiments should be performed for each pair of processors.

Let us consider how these models can be used for estimation of the execution time of MPI collective communication operations, namely, for different algorithms of scatter. We start with a simple, linear, algorithm, when messages are sent in the flat tree. There are only two ways to model this operation with these models. The first option is to assume that all point-to-point communications between the root and destination processors are performed sequentially. In this case, the total execution time will be expressed as a sum of $n-1$ point-to-point execution times: $(n-1)(\alpha+\beta M)$ (homogeneous Hockney) or $\sum_{i=0,i\neq r}^{n-1}(\alpha_{ri}+\beta_{ri}M)$ (heterogeneous Hockney). The second option is to assume that the point-to-point communications are fully parallel. In that case, the predictions will be $\alpha+\beta M$ with the homogeneous Hockney models and $\max_{i=0,i\neq r}^{n-1}(\alpha_{ri}+\beta_{ri}M)$ with the heterogeneous one.

Unfortunately, both these assumptions do not accurately reflect the way the operation is executed on a switched cluster. On this platform, the linear scatter combines serial execution at the sending processor and parallel execution in the network and at the receiving nodes. The lack of parameters separating the contributions of the processors and the network in the Hockney model does not allow for expressing such effects. As a result, both homogeneous and heterogeneous sequential Hockney predictions of the linear scatter are pessimistic, while their parallel counterparts are too optimistic (see Fig. 1).

Because of the design of the Hockney model, the same formulas can be applied to the estimation of linear gather. They are not accurate either. The experimental results for linear gather are discussed in Section V.

Despite the fact that in the case of the linear scatter, the heterogeneous Hockney model appeared as inaccurate in prediction of its execution time as the homogeneous one, in general, heterogeneous extensions of traditional models can provide more accurate predictions of collective operations on heterogeneous platforms, at least, for algorithms with some inherent parallelism. Examples of such algorithms are the algorithms of scatter and gather based on binomial communication tress. In the binomial tree, the sub-trees of the same order represent non-overlapping sets of processors. Therefore, communications within the sub-trees can be performed in parallel.

The communication tree for scatter/gather and 16 participating processors is shown in Fig. 2. The nodes of the tree represent the processors. The arcs represent the logical communication links between the processors. Given 16 data blocks of the same size 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. With the use of the heterogeneous Hockney model, the execution time of the binomial algorithm of scatter/gather can be approximated by the following formula [3]:

$(\log_2 n)\alpha + (n-1)\beta M$, where M is a size of the receive (scatter) and send (gather) buffers. In each sub-tree, the largest messages $2^k M$ are sent/received first, with k starting with $\log_2(n-1)$. The formula includes parallel (constant contributions in sub-trees of the same order, $C_k, k = 1,...,4$) and sequential (accumulated variable contributions) parts.

In this formula, communications in sub-trees of the same order are assumed simultaneous, which is unrealistic in the case of a heterogeneous cluster. The communication execution times in two sub-trees of the same order may be different. Moreover, the communication execution time associated with each sub-tree will also depend on mapping of the processor of the cluster to the nodes of the binomial communication tree. The homogeneous Hockney model is not detailed enough to express these nuances. At the same time, the use of the heterogeneous Hockney model allows us to propose the following more accurate formula for binomial scatter/gather:

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

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 Fig. 2). $T_c(k-1)$ is the execution time of the sub-tree c of order $k-1$ from the set C_{k-1}. For the tree in Fig. 2, 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). Maximums and recursion correspond to parallel communications in the sub-trees of the same height.

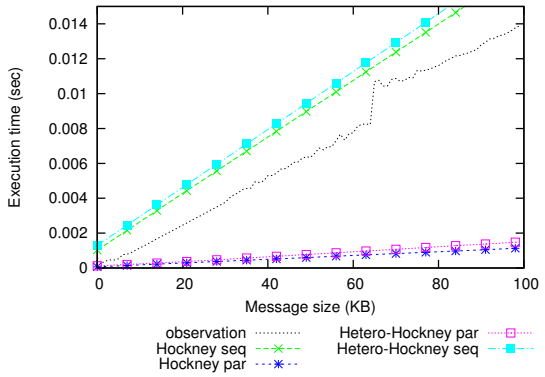

Figure 1. The prediction of the execution time of linear scatter on the 16-node heterogeneous cluster.

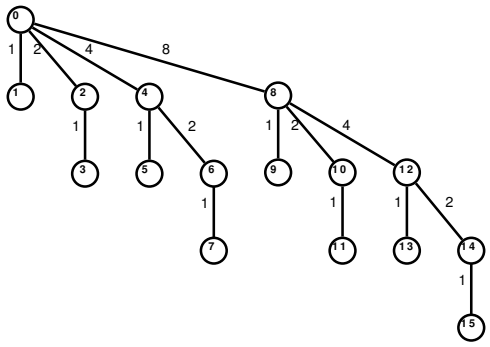

Figure 2. The binomial communication tree for scatter (gather) involving 16 processors. The nodes represent the processors. Each arc represents a logical communication link and is marked by the number of data block communication over this link.

For 8 participating processors with the root 0 the formula will look as follows:

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

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 in the case of 8 processors, it will be rewritten as:

$$\alpha + 4\beta M + \alpha + 2\beta M + \alpha + \beta M \approx \log_2 8\alpha + (8-1)\beta M \quad (3)$$

In Fig. 3, both homogeneous and heterogeneous Hockney predictions are compared with the observed execution time of the binomial scatter on the 16-node heterogeneous cluster specified in Table I. One can see that the heterogeneous Hockney model much better approximates the performance of the binomial scatter. At the same time, the example of linear scatter/gather reminds us that both heterogeneous and homogeneous Hockney models are quite restricted in their ability to accurately predict the execution time of arbitrary algorithms of collective communication operations. The main reason is that the Hockney model does not separate contributions of different nature in the execution time of a point-to-point operation non-intuitively combining them in a small number of point-to-point parameters.

This problem is not specific for the Hockney model; it is common for all traditional models. Let us consider some more elaborated traditional models such as LogP, LogGP and PLogP. The LogP model [4] 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; it reflects the constant contribution of network. The overhead, o, is the time period during which

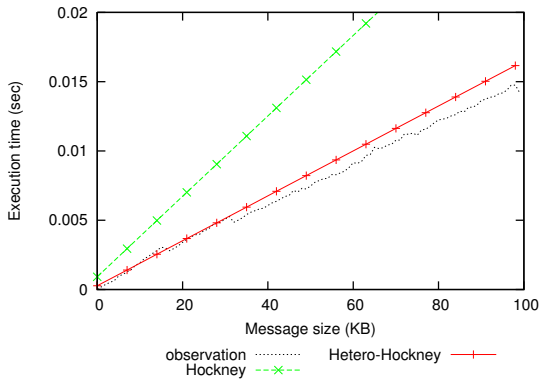

Figure 3. The prediction of the homogeneous and heterogeneous Hockney models vs the observation of the binomial scatter.

the processor is engaged in sending or receiving a message (a constant processor contribution). 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 $L+2o$. The LogP model assumes that a large message is decomposed to a series of short messages. In the formula for a series the gap parameter will be used: $L+2o+Mg$. Therefore, the gap can be attributed to the variable contributions of processors and network.

The LogGP model [1], an extension of LogP, 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 original gap parameter, g, is also used in the model to represent the delays between consecutive communications. For example, the execution time of m sendings of M bytes is estimated as follows: $L+2o+(M-1)G+(m-1)g$. Both gap parameters combine the contributions of processors and network. The gap is a constant parameter, while the gap per byte is variable.

In the PLogP (parameterized LogP) model [7], all parameters except for latency are piecewise linear functions of the message size, and the meaning of parameters differs slightly from LogP. The meaning of latency, L, is not intuitive; 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 (variable contributions of processors). They can be overlapped for sufficiently large messages. The gap, $g(M)$, is the minimum time between consecutive transmissions or receptions; it is the reciprocal value of the end-to-end bandwidth between two processors for messages of a given size M. The gap is assumed to cover the overheads ($g(M) \geq o_s(M)$, $g(M) \geq o_r(M)$) and represents mixed processor-network variable contributions. According to the PLogP model, the point-to-point execution time is equal to $L+g(M)$.

The parameters of LogP-based models are estimated by the following point-to-point experiments:

- Both the sending and receiving overheads are found directly from the time of sending and receiving a message. The o_s parameter is estimated in the roundtrip that consists of sending a message and receiving an empty reply: $i \underset{0}{\overset{M}{\rightleftarrows}} j$. In another roundtrip, $i \underset{M}{\overset{0}{\rightleftarrows}} j$, after completion of the send operation, the sending processor waits for some time, sufficient for the reply to reach its destination, and only then posts a receive operation.

The execution time of the receive operation approximates o_r.

- The latency is found as $L = RTT/2 - o_s - o_r$ from the execution time of the roundtrip with non-empty messages sent and received: $i \underset{M}{\overset{M}{\rightleftarrows}} j$.
- 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, $\sum_{x=0}^{s} i \underset{0}{\overset{\overbrace{M \ldots M}^{2^x}}{\rightleftarrows}} j$, reflects the nature of the gap parameter but takes a long time.

The estimation of the PLogP parameters will be the most time consuming because these experiments are performed for multiple message sizes, which are selected adaptively. For example, if the $g(M_k)$ is not consistent with the linearly extrapolated value based on $g(M_{k-2})$ and $g(M_{k-1})$, then another measurement is performed for the message size $M_k^{'} = (M_k + M_{k-1})/2$, and the $g(M_k^{'})$ is estimated.

The LogP-based models can be applied to heterogeneous clusters in the same way as the Hockney model. The parameters are found for all pairs of processors, with the above experiments being performed for each link. Then these parameters (heterogeneous version) or their average values (homogeneous version) will be used in modelling. However, there may be options how to build heterogeneous extensions of these models for heterogeneous clusters with single switch. For example, since the PLogP overheads, $o_s(M)$ and $o_r(M)$, correspond to the processor variable contributions, it is sensible to assume that they should be the same for all point-to-point communications the processor can be involved. This means that, in the heterogeneous extension of the PLogP model, the average processor overheads should be used (averaged from the values found in the experiments between all pairs included the given processor). On the other hand, the latency, L, and the gap, $g(M)$, parameters (which are connected with the overheads in the design of above communication experiment) represent both processor and network contributions and, therefore, cannot be averaged in this way. For this reason, it is not trivial and straightforward to extend the LogP-based models. This can be a subject of separate research.

Let us consider how these models can be used for estimation of the execution time of linear scatter/gather. Similarly to the Hockney model, the execution time of both operations can be approximated by the same formulas: $L + 2o + (n-1)(M-1)G + (n-2)g$ (LogGP), $L + (n-1)g(M)$ (PLogP) [2], where M is a block size. Pjesivac-Grbovic et al. [14] demonstrated that the analytical prediction of the execution time of collectives provided by these models was not accurate. We do not know how to express in an intuitive way the execution time of these operations with the heterogeneous parameters of the LogP-based models.

The elaboration of communication performance models in order to separate the constant and variable contributions of processors and network can lead to more accurate prediction of the communication execution time. In [8, 9], we proposed an analytical heterogeneous communication performance model, LMO, designed for both homogeneous and heterogeneous clusters based on a switched network. The model includes the parameters that reflect the contributions of both links and processors to the communication execution time, and allows us to represent the aspects of heterogeneity for both links and processors. The LMO model provides more intuitive and accurate expression of the execution time of MPI collective operations. The parameters of this model cannot be estimated from the point-to-point experiments only. As a solution of this problem, we proposed to introduce additional collective communication experiments involving more than two processors [11]. These experiments are designed to give us sufficient data in order to build and solve simple systems of equations to find the point-to-point parameters. It separates the variable contributions of processors and network. At the same time, the constant parameters of the model still combine the fixed delays sourced from processors and the network.

In this paper, we present an extension of the LMO model that fully separates the constant contributions of processors and network. We suggest an approach to the design of the communication experiments required to estimate the point-to-point parameters of the elaborated models and describe a set of communication experiments for the LMO model. We conclude with experimental results demonstrating that the model much more accurately predicts the execution time of collective operations than traditional models.

III. LMO, A HETEROGENEOUS COMMUNICATION PERFORMANCE MODEL

The original LMO model [8, 9] is based on five parameters characterizing the point-to-point communication: $(C_i, t_i) \xrightarrow{(\beta_{ij})} (C_j, t_j)$. 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 is estimated by $C_i + C_j + M(t_i + \frac{1}{\beta_{ij}} + t_j)$, where:

- C_i, C_j are the fixed processing delays;
- t_i, t_j are the delays of processing of a byte;
- 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 $\beta_{ij} = \beta_{ji}$. One can see that the parameters describing the fixed delays combine the constant contributions of both the processors and the network. In this paper, we present an extended model that distinguishes between these contributions.

The extended model includes additional point-to-point parameters, latency L_{ij} (fixed network delays). The point-to-point communication is described by six parameters: $(C_i, t_i) \xrightarrow{(L_{ij}, \beta_{ij})} (C_j, t_j)$ and the execution time is estimated as $C_i + L_{ij} + C_j + M(t_i + \frac{1}{\beta_{ij}} + t_j)$. This model provides more flexibility to express the execution time of collectives. Namely, the formulas for collectives can include the fixed processor delays and latencies in different combinations of maximums and sums, which will reflect, for example, the cases when the processor delays are serialized, while transmission is performed in parallel. In terms of the Hockney model, the parameters can be expressed as follows: $C_i + L_{ij} + C_j = \alpha_{ij}^H$, $t_i + \frac{1}{\beta_{ij}} + t_j = \beta_{ij}^H$, which means that we distinguish between the processor and network contributions in the constant and variable parts of the point-to-point execution time.

Let us consider how this model can be used to express the execution time of MPI collective operations, for example, linear scatter and gather. The formulas are intuitive, including combinations of sums and maximums of the point-to-point parameters. The expression for linear scatter is the following:

$$(n-1)(C_r + Mt_r) + \max_{i=0, i\neq r}^{n-1}(L_{ri} + \frac{M}{\beta_{ri}} + C_i + Mt_i) \qquad (4)$$

The sequential part of this formula, $(n-1)(C_r + Mt_r)$, 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 to the features of network switches, which parallelize the messages addressed to different processors.

The execution time of linear gather is expressed with help of the point-to-point parameters of the LMO model (analytical part) and some extra parameters that reflect the irregularities observed in the execution time of collective operations on switched clusters (empirical part):

$$(n-1)(C_r + Mt_r) + \begin{cases} \max_{i=0, i\neq r}^{n-1}(L_{ri} + \frac{M}{\beta_{ri}} + C_i + Mt_i) & M < M_1 \\ \sum_{i=0, i\neq r}^{n-1}(L_{ri} + \frac{M}{\beta_{ri}} + C_i + Mt_i) & M > M_2 \end{cases} \qquad (5)$$

This formula also reflects the serialization of the processing on the root, which receives all messages. The extra threshold parameters, M_1 and M_2, are found from the observations of the execution time of linear gather and categorize the message sizes for which the performance of linear gather differs. For the small (less than M_1) and large (larger than M_2) messages the execution time increases linearly, while for the medium size messages the non-linear and non-deterministic escalations of the execution time are observed [10]. These parameters depend on the particular cluster and MPI implementation. For example, on the 16-node heterogeneous cluster specified in Table I, we observed $M_1 = 4KB$, $M_2 = 65KB$ for LAM 7.1.3 and $M_1 = 3KB$, $M_2 = 125KB$ for MPICH 1.2.7. The maximum reflects the parallel processing delays on the processors and parallel transmissions supported by network switch. However, the sending of large messages to one destination is serialized and is hence expressed by the sum of the point-to-point parameters.

The LMO model allows us to build the formulas that accurately reflect different aspects of communications in the algorithms of collective operations that are performed on our target platform, switched clusters. The accuracy of the prediction with the LMO model will be demonstrated in Section V.

IV. Estimation of the LMO point-to-point parameters

In contrast to the traditional communication performance models, the LMO model requires more complicated communication experiments to estimate its parameters. The answers to the questions as to why these experiments involve more than two processors and how they are designed are given in [11]. In this paper, we propose an elaborated approach to the estimation of the parameters of the advanced communication performance models such as LMO. Then we apply this approach to the extended LMO model and present a modified set of experiments required to estimate its parameters. This 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

[12], which 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 is proved 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 perform multiple repetitions of the experiments and statistical analysis of their results.

The cost of the accurate estimation of a communication model of the heterogeneous cluster can be quite significant as it typically involves multiple repetitions of the same communication experiments between different subsets of the processors and statistical processing of their results for a reliable approximation of the parameters. 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:

- The cost of the estimation can be significantly reduced if we simultaneously execute several independent communications involving non-overlapped sets of processors without degradation of their performance. On clusters based on a single switch, 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. This 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.
- 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 additional collective experiments, some processors will participate in more than one triplet. Therefore, the value of some parameters can be found independently from different independent experiments. 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.

We applied this approach to estimation of the parameters of the extended LMO model. The modified set of communication experiments is similar to one that was proposed in [11]. In addition to roundtrips, it includes the parallel communications between three processors $i \underset{N}{\overset{M}{\rightleftarrows}} j,k$, which consist of the sending of M bytes from the processor i to the processors j,k and the receiving of the N byte replies. The execution time of this communication experiment can be represented as a sum of the execution times of linear scatter and gather, $T_{scatter}(M)+T_{gather}(N)$. This is because we assume that the execution time of the copying of the root's block on the root processor is negligibly small.

The constant parameters are estimated from the roundtrips and one-to-two communications with empty message as in (6). The expressions for the roundtrips (7) can be used to simplify the formula for the one-to-two communication. The solution of the system of equations is shown in (8).

$$\begin{cases} T_{ij}(0) = 2(C_i + L_{ij} + C_j) & i \underset{0}{\overset{0}{\rightleftarrows}} j \\ T_{jk}(0) = 2(C_j + L_{jk} + C_k) & j \underset{0}{\overset{0}{\rightleftarrows}} k \\ T_{ik}(0) = 2(C_i + L_{ik} + C_k) & i \underset{0}{\overset{0}{\rightleftarrows}} k \\ T_{ijk}(0) = 2(2C_i + \max\limits_{x=j,k}(L_{ix} + C_x)) & i \underset{0}{\overset{0}{\rightleftarrows}} j,k \\ T_{jik}(0) = 2(2C_j + \max\limits_{x=i,k}(L_{jx} + C_x)) & j \underset{0}{\overset{0}{\rightleftarrows}} i,k \\ T_{kij}(0) = 2(2C_k + \max\limits_{x=i,j}(L_{kx} + C_x)) & k \underset{0}{\overset{0}{\rightleftarrows}} i,j \end{cases} \quad (6)$$

$$T_{ijk}(0) = 2(2C_i + \max_{x=j,k}(L_{ix} + C_x)) = 2C_i + \max_{x=j,k} T_{ix}(0) \quad (7)$$

$$\begin{cases} C_i = (T_{ijk}(0) - \max\limits_{x=j,k} T_{ix}(0))/2 & L_{ij} = T_{ij}(0)/2 - C_i - C_j \\ C_j = (T_{jik}(0) - \max\limits_{x=i,k} T_{jx}(0))/2 & L_{jk} = T_{jk}(0)/2 - C_j - C_k \\ C_k = (T_{kij}(0) - \max\limits_{x=i,j} T_{kx}(0))/2 & L_{ik} = T_{ik}(0)/2 - C_i - C_k \end{cases} \quad (8)$$

The variable parameters are found with help of the same communication experiments but with non-empty messages. Due to the irregularities of linear scatter and gather observed on switched clusters, the size of messages should be carefully selected in the one-to-two experiments [11]. We send the messages of medium size to avoid a possible leap in the execution time of scatter observed for LAM and Open MPI, and receive empty replies to eliminate the escalations in the execution time of gather. We build the system of equations (9).

Having replaced some items by the point-to-point execution time, we obtain the expression (10) of the execution time of one-to-two communication. The variable processor delays and transmission rates are found as in (11).

$$\begin{cases} T_{ij}(M) = 2(C_i + L_{ij} + C_j + M(t_i + \frac{1}{\beta_{ij}} + t_j)) & i \overset{M}{\underset{M}{\rightleftarrows}} j \\ T_{jk}(M) = 2(C_j + L_{jk} + C_k + M(t_j + \frac{1}{\beta_{jk}} + t_k)) & j \overset{M}{\underset{M}{\rightleftarrows}} k \\ T_{ik}(M) = 2(C_i + L_{ik} + C_k + M(t_i + \frac{1}{\beta_{ik}} + t_k)) & i \overset{M}{\underset{M}{\rightleftarrows}} k \\ T_{ijk}(M) = 2(2C_i + Mt_i) + \max\limits_{x=j,k}(2(L_{ix} + C_x) + M(\frac{1}{\beta_{ix}} + t_x)) & i \overset{M}{\underset{0}{\rightleftarrows}} j,k \\ T_{jik}(M) = 2(2C_j + Mt_j) + \max\limits_{x=i,k}(2(L_{jx} + C_x) + M(\frac{1}{\beta_{jx}} + t_x)) & j \overset{M}{\underset{0}{\rightleftarrows}} i,k \\ T_{kij}(M) = 2(2C_k + Mt_k) + \max\limits_{x=i,j}(2(L_{kx} + C_x) + M(\frac{1}{\beta_{kx}} + t_x)) & k \overset{M}{\underset{0}{\rightleftarrows}} i,j \end{cases} \quad (9)$$

$$T_{ijk}(M) = 2(2C_i + Mt_i) + \max_{x=j,k}(2(L_{ix} + C_x) + M(\frac{1}{\beta_{ix}} + t_x)) = \\ = 2C_i + Mt_i + \max_{x=j,k}(T_{ix}(0) + T_{ix}(M))/2 \quad (10)$$

$$\begin{cases} t_i = (T_{ijk}(M) - \max\limits_{x=j,k}(T_{ix}(0) + T_{ix}(M))/2 - 2C_i)/M \\ t_j = (T_{jik}(M) - \max\limits_{x=i,k}(T_{jx}(0) + T_{jx}(M))/2 - 2C_j)/M \\ t_k = (T_{kij}(M) - \max\limits_{x=i,j}(T_{kx}(0) + T_{kx}(M))/2 - 2C_k)/M \\ \frac{1}{\beta_{ij}} = (T_{ij}(M)/2 - C_i - L_{ij} - C_j)/M - t_i - t_j \\ \frac{1}{\beta_{jk}} = (T_{jk}(M)/2 - C_j - L_{jk} - C_k)/M - t_j - t_k \\ \frac{1}{\beta_{ik}} = (T_{ik}(M)/2 - C_i - L_{ik} - C_k)/M - t_i - t_k \end{cases} \quad (11)$$

The set of experiments includes C_n^2 roundtrips and $3C_n^3$ one-to-two communications. The processing delays, C_i and t_i, can be obtained from C_{n-1}^2 different triplets, the processor i takes part in, and can be averaged; the latencies, L_{ij}, and the transmission rates, β_{ij}, can be averaged from $n-2$ values:

$$\overline{C_i} = \frac{\sum_{j,k \neq i} C_i}{C_n^2} \quad \overline{t_i} = \frac{\sum_{j,k \neq i} t_i}{C_n^2} \quad \overline{L_{ij}} = \frac{\sum_{k \neq i,j} L_{ij}}{n-2} \quad \overline{\beta_{ij}} = \frac{\sum_{k \neq i,j} \beta_{ij}}{n-2} \quad (12)$$

The execution time of the estimation of the parameters depends on:

- 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 $2(n+C_n^2)$ averagings).

As the parameters of our point-to-point model are found from a small number of experiments, they can be sensitive to 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 ten in a series) because all the parameters have already been averaged during the process of their finding. Another optimization is related to the target platform, that is a switched cluster. All communication experiments are performed in parallel on non-overlapped pairs or triplets of processors. As network switches provide forwarding packets between sources and destinations without contentions, the parallel execution does not affect the accuracy of the estimation.

V. EXPERIMENTAL RESULTS

In this section, we present the experimental results for the 16-node heterogeneous cluster with a single Ethernet switch and LAM 7.1.3 specified in Table I. We developed the software tool that provides the estimation of parameters of the LMO model and the heterogeneous extensions of the Hockney, PLogP and LogGP models [13]. Using the models' parameters, this tool predicts the execution time of different algorithms of collective communication operations and provides the optimized model-based algorithms. In this section, we present the experimental results of modelling of scatter and gather. The communication execution time was measured with help of the MPIBlib benchmarking library [12] with the confidence level 95% and the relative error 2.5%.

The expressions of the execution time of linear scatter and gather are summarized in Table II. Only the LMO model provides two different formulas for scatter and gather, reflecting steeper slope in the execution time of gather observed for large messages. Only the LMO model reflects the irregular behaviour of linear gather. On computational clusters with TCP/IP layer (including the cluster specified above), we observed a leap in the execution time of linear scatter (see Fig. 4, observation, 64KB). In the previous version of the LMO model, we included the extra parameter that reflects this leap. However, for larger messages, these leaps regularly repeated, converging to the line with the same slope. We could have included multiple empirical parameters to the LMO model and have presented the execution time of scatter as a piecewise linear function, but due to not very

TABLE I. SPECIFICATION OF THE 16-NODE HETEROGENEOUS CLUSTER

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

TABLE II. THE PREDICTION OF THE EXECUTION TIME OF LINEAR SCATTER AND GATHER

Model	Linear scatter prediction	Linear gather prediction
Hetero-Hockney	$\sum_{i=0,i\neq r}^{n-1}(\alpha_{ri}+\beta_{ri}M)$	
LogGP	$L+2o+(n-1)(M-1)G+(n-2)g$	
PLogP	$L+(n-1)g(M)$	
LMO	$(n-1)(C_r+Mt_r)+\max_{i=0,i\neq r}^{n-1}(L_{ri}+C_i+M(\frac{1}{\beta_{ri}}+t_i))$	$(n-1)(C_r+Mt_r)+\begin{cases}\max_{i=0,i\neq r}^{n-1}(L_{ri}+C_i+M(\frac{1}{\beta_{ri}}+t_i)) & M<M_1\\ \sum_{i=0,i\neq r}^{n-1}(L_{ri}+C_i+M(\frac{1}{\beta_{ri}}+t_i)) & M>M_2\end{cases}$

significant values of the leaps and for simplicity, we considered only the linear model, which satisfactorily approximates the observed execution time of the native (linear) LAM scatter. The PLogP prediction provides the same accuracy for medium size messages and also reflects the leap in the execution time, after which it diverges from the observations. The estimations of other traditional models are inaccurate.

The LMO includes not only analytical but also empirical parameters. For small (less than 4KB) and large (more than 65KB) messages, the execution time of linear gather is represented as two lines with different slopes (Fig. 5). For medium size messages, the LMO model defines the most frequent values of escalations and their probability. The escalations are non-deterministic and reach 0.25 sec. The LMO model also shows the probability that the execution time will fit the linear model for small messages: the probability becomes less with the growth of message size. Therefore, only the LMO prediction reflects the irregularity in the execution time of the native (linear) LAM gather. As regards the intervals for small and large messages, traditional models better predict the execution time of gather rather than scatter.

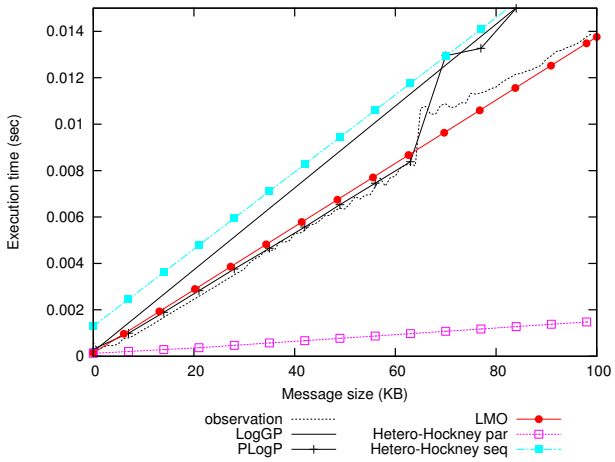

Figure 4. The prediction of the execution time of linear scatter on the 16-node heterogeneous cluster.

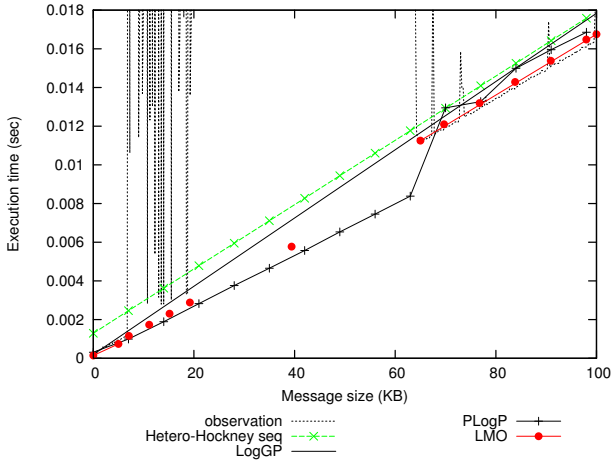

Figure 5. The prediction of the execution time of linear gather on the 16-node heterogeneous cluster.

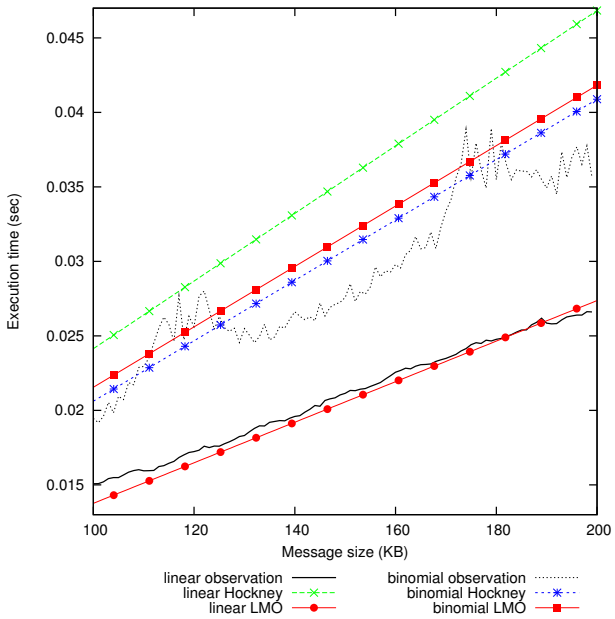

Figure 6. The performance of the linear and binomial algorithms of scatter vs the heterogeneous Hockney and LMO predictions.

Fig. 4 and Fig. 5 demonstrate that the intuitive prediction provided by the LMO model is more accurate than the predictions of traditional models. The accurate prediction of the execution time allows for the correct decision on switching between the algorithms of a collective communication operation. In Fig. 6, the predictions provided by the heterogeneous Hockney and LMO models are presented for the linear and binomial algorithms of scatter for messages $100KB < M < 200KB$. Similarly to [14], the Hockney model mispredicts that the binomial algorithm outperforms the linear one, switching in favour of the first, whereas the decision based on the LMO approximation will be correct. The accurate prediction can be a basis for the model-based optimization of collective operations. Fig. 7, shows the performance of a simple optimized version of gather that was implemented on top of its native counterpart by splitting the messages of medium size and performing a series of gathers in order to avoid the escalations. Using the empirical parameters of the LMO model for linear gather, we gained 10 times better performance [10].

VI. Conclusion

In this paper, we have considered how to model collective communication operations on switched clusters in an intuitive way, and analyzed why the analytical prediction of traditional models may be inaccurate on this platform. The common problem of all traditional models is the combining of contributions of different nature. The intuitive models separate the constant and variable contributions of the processors and the network and provide the expression of the execution time of any collective communication operation as a combination of maximums and sums of the point-to-point parameters. By the example of the LMO model, designed for

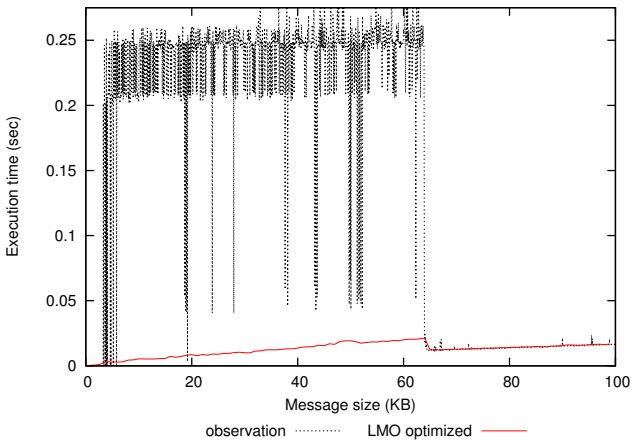

Figure 7. The LMO model-based optimization of linear gather on the 16-node heterogeneous cluster.

homogeneous and heterogeneous switched clusters, we showed that the full separation of these contributions leads to more intuitive and accurate predictions of the execution time of collective operations. In contrast to the traditional models, the parameters of intuitive models cannot be estimated from only the point-to-point experiments. In this paper, we described the efficient technique for accurate estimation of parameters of such model, which includes a relatively small number of point-to-point and collective communications and a solution of simple systems of linear equations. The accuracy of estimation was achieved by careful selection of message sizes and averaging the values of the parameters. The accuracy of the intuitive modelling of scatter and gather was validated experimentally.

Acknowledgment

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

References

[1] A. Alexandrov, M. Ionescu, K. Schauser, C. Scheiman, “LogGP: Incorporating long messages into the LogP model,”. Proc. SPAA 1995, ACM, 1995, pp. 95-105.

[2] L. Barchet-Estefanel, G. Mounie, “Fast Tuning of Intra-cluster Collective Communications,” Proc. EuroPVM/MPI 2004. LNCS, vol. 3241, Springer, 2004, pp. 28–35.

[3] E. Chan, M. Heimlich, A. Purakayastha, R. van de Geijn, “On optimizing collective communication,” Proc. Cluster 2004, IEEE Computer Society Press, 2004, pp. 145-155.

[4] D. Culler, R. Karp, D. Patterson, et al., “LogP: Towards a realistic model of parallel computation,” Proc. PPoPP 1993, ACM, 1993, pp. 1–12.

[5] J. Hatta, S. Shibusawa, “Scheduling algorithms for efficient gather operations in distributed heterogeneous systems,” Proc. WPP 2000, pp 173-180.

[6] R. Hockney, “The communication challenge for MPP: Intel Paragon and Meiko CS-2,” Parallel Computing, vol. 20, 1994, pp. 389–398.

[7] T. Kielmann, H. Bal, K. Verstoep, “Fast measurement of LogP parameters for message passing platforms,” Proc. IPDPS 2000. LNCS, vol. 1800, Springer, 2000, pp. 1176–1183.

[8] A. Lastovetsky, I. Mkwawa, M. O'Flynn, "An Accurate Communication Model of a Heterogeneous Cluster Based on a Switch-Enabled Ethernet Network," Proc. ICPADS 2006, vol. 2, IEEE Computer Society Press, 2006, pp. 15–20.

[9] A. Lastovetsky, M. O'Flynn, "A Performance Model of Many-to-One Collective Communications for Parallel Computing," Proc. IPDPS 2007, IEEE Computer Society Press, 2007, pp. 1-8.

[10] A. Lastovetsky, M. O'Flynn, V. Rychkov, "Optimization of Collective Communications in HeteroMPI," Proc. EuroPVM/MPI 2007. LNCS, vol. 4757, Springer, 2007, pp. 135–143.

[11] A. Lastovetsky, V. Rychkov, "Building the Communication Performance Model of Heterogeneous Clusters Based on a Switched Network," Proc. Cluster 2007, IEEE Computer Society Press, 2007, pp. 568-575.

[12] A. Lastovetsky, V. Rychkov, M. O'Flynn, "MPIBlib: Benchmarking MPI Communications for Parallel Computing on Homogeneous and Heterogeneous Clusters," Proc. EuroPVM/MPI 2008. LNCS, vol. 5205, Springer, 2008, pp. 227-238.

[13] A. Lastovetsky, V. Rychkov, M. O'Flynn, "A Software Tool for Accurate Estimation of Parameters of Heterogeneous Communication Models," Proc. EuroPVM/MPI 2008. LNCS, vol. 5205, Springer, 2008, pp. 43-54.

[14] J. Pjesivac-Grbovic, T. Angskun, G. Bosilca, G. Fagg, E. Gabriel, J. Dongarra, "Performance Analysis of MPI Collective Operations," Cluster Computing, vol. 10(2), 2007, pp. 127–143.

[15] R. Thakur, R. Rabenseifner, W. Gropp, "Optimization of Collective Communication Operations in MPICH," Intl J. of High Performance Computing Applications, vol. 19, 2005, pp. 49–66.

Alexey Lastovetsky received a PhD 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 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 into heterogeneous data distribution algorithms and modeling the performance of processors in heterogeneous environments. He 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 the PhD degree from the Russian Academy of Sciences in 2005. His main research interests include design of algorithms and tools for parallel and distributed computing systems, mathematical modeling and numerical methods, computer aided design and engineering.

Maureen O'Flynn is a PhD candidate in the School of Computer Science and Informatics at University College Dublin, National University of Ireland. Her research interests include communication performance models for parallel computing on heterogeneous platforms.