# Building the communication performance model of heterogeneous clusters based on a switched network

Alexey Lastovetsky[#1], Vladimir Rychkov[#]

[#] *School of Computer Science and Informatics, University College Dublin*
*Belfield, Dublin 4, Ireland*

[1]alexey.lastovetsky@ucd.ie

***Abstract*— Analytical communication performance models play an important role in prediction of the execution time of parallel applications on multiprocessors. Apart from designing such a model, accurate estimation of the values of its parameters is one of the main issues. This paper deals with a heterogeneous analytical communication model designed for prediction of MPI communications on heterogeneous clusters based on a switched network. Accurate estimation of the parameters of this model is a particularly challenging task due to a large number of the parameters. In this paper, we present a solution of the task based on a carefully designed set of communication experiments, which not only allows us to obtain the accurate estimation of the parameters but also tries to minimise the total execution time of the experiments. Experiments demonstrating the accuracy and efficiency of the proposed solution are also presented.**

## I. INTRODUCTION

A programming system for high performance computing on heterogeneous platforms, such as mpC [1], HeteroMPI [2], GridSolve [3], strongly relies on the performance model of the executing platform. The accuracy of the model very much determines the efficiency of applications. The model is used for prediction of the execution time of different configurations of the application, including their computation and communication costs, in order to find the optimal one. In this paper, we deal with modelling the performance of MPI communication operations on heterogeneous clusters bas ed on switched networks, which are arguably the most common platform for parallel computing.

Traditionally, communication performance models for high performance computing are analytical. Therefore, there are two main issues associated with such a model. The first issue is the design of the parameterized analytical model itself. The second issue is the efficient and accurate estimation of the parameters of the model for each particular platform from the targeted class. This paper mainly deals with the second issue.

Analytical predictive communication models are usually built for homogeneous clusters. The basis of that model is a point-to-point communication model characterized by a set of integral parameters, having the same value for each pair of processors. Collective operations are expressed as a combination of the point-to-point parameters, and the collective communication execution time is analytically predicted for different message sizes and numbers of processors involved. The core of this approach is the choice of such 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 the measurements of the execution time of communications between any two processors. When such a homogeneous communication model is applied to a cluster of heterogeneous processors, its point-to-point parameters are found by averaging values obtained for every pair of processors, and the averaged values are then used in modelling collective communication operations. Thus, in this case, the heterogeneous cluster will be treated as homogeneous in terms of the performance of communication operations.

When some processors 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 as they are directly determined by the performance characteristics of the processors. In such heterogeneous communication models, the total number of point-to-point parameters would be significantly larger. In [4], we proposed an analytical heterogeneous communication model designed for prediction of the execution time of MPI communications on heterogeneous clusters based on a switched network. Each processor contributes into the model a small number of point-to-point parameters, making the total number of the parameters proportional to the number of processors in the cluster. Experimental finding of the accurate values of the parameters for each particular switch-based heterogeneous cluster is not a trivial task. This paper presents the design of communication experiments that allow us to find these parameters.

The statistical methods of finding the point-to-point parameters, normally used in the case of homogeneous communication models, will result in unacceptably large amount of measurements if applied as they are to our heterogeneous communication model. Therefore, another issue that has to be addressed is the minimization of the number of measurements necessary to accurately find the point-to-point parameters. We avoid using the traditional statistical methods and perform a relatively small number of measurements for some particular message sizes, which gives us the same accuracy as the exhaustive statistical analysis.

This paper is organized as follows. In Section 2, related work in the area of communication performance modelling is discussed. In Section 3, we describe the point-to-point model

of heterogeneous clusters based on a switched network. In Section 4, we design communication experiments to measure the point-to-point parameters. In Section 5, experimental results demonstrating the accuracy and efficiency of the proposed solution are presented. Section 6 concludes the paper with a brief discussion of future work.

## II. RELATED WORKS

The Hockney model [5] of the execution time of point-to-point communication is $\alpha + \beta m$, where $\alpha$ is the latency, $\beta$ is the bandwidth and $m$ is the message size. The parameters of the Hockney model can be measured directly frompoint-to-point tests with help of linear regression.

The LogP model [6] 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, $P$. According to LogP, the time of point-to-point communication can be estimated by $L+2o$. The gap parameter is added for every message sent sequentially, so that the network bandwidth can be expressed as $L/g$. This model is extended for large messages by introducing the gap per byte, $G$, with the point-to-point communication time estimated by $L+2o+(m-1)G$ (the LogGP model [7]).

In the PLogP (parameterized LogP) model [8], some parameters are piecewise linear functions of the message size, the send and receive overheads are distinguished, and the meaning of the parameters slightly differs from LogP. The end-to-end latency, $L$, is a constant, combining all fixed contribution factors such as copying to/from network interfaces and the transfer over network. The send, $o_s(m)$, and receive, $o_r(m)$, overheads are the times the source and destination processors are busy during communication. 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: $g(m) \geq o_s(m)$ and $g(m) \geq o_r(m)$.

The authors of the PLogP model developed the logp_mpi library [8], which is widely used for measurement of the point-to-point parameters of LogP-based models. In the software package, two techniques of measurement are implemented for message-passing systems: direct and optimized. In both techniques the overheads $o_s(m)$ and $o_r(m)$ are measured directly from the time of sending/receiving the message of $m$ bytes in the roundtrips $i \underset{0}{\overset{M}{\rightleftarrows}} j$, consisting of a single sending of the message of $m$ bytes from processor $i$ to processor $j$ and a single zero reply, and $i \underset{M}{\overset{0}{\rightleftarrows}} j$, consisting of an empty send and non-empty reply. For each message size, these tests are initially run a small number of times. As long as the variance of measurements is too high, the amount of roundtrips is successively increased, staying actually sufficiently small (of the order of tens).

In the direct method, the gap values $g(m)$ are found from the execution time $s_n(m)$ of sending without reply a large number $n$ of messages of size $m$. As the execution time of sending $n$ messages in a row is $s(m_1,...,m_n) = g(m_1) + ... + g(m_n)$ on the sender side, the gap value is equal to $g(m) = s_n(m)/n$. The number of messages $n$ is obtained within the saturation process. The execution time $RTT_n(m)$ of a roundtrip consisting of $n$ sendings of the message of $m$ bytes and a single zero reply is measured for $n$ that is doubled until $\frac{RTT_{2n}(m)/2n - RTT_n(m)/n}{RTT_n(m)/n} \times 100\%$ changes less then 1%. The saturation ensures that the roundtrip time is dominated by bandwidth rather than latency. As a result, $n$ will be quite large (of the order of thousands and more) and the saturation will take the lion's share of the overall execution time. Thus, the direct technique of estimation of the parameter $g(m)$ is very expansive. Therefore, an indirect optimized method for estimation of the parameter has been proposed.

The optimized technique is based on the non-obvious assumption that the execution time of a single roundtrip is equal to $RTT(m) = RTT_1(m) = L + g(m) + L + g(0)$. This assumption allows us to replace the direct findings of the gap for each message size by the measurements of the execution time of the short roundtrips: $g(m) = RTT(m) - RTT(0) + g(0)$, with the only gap value of $g(0)$ found directly within the saturation process. The latency is found from a single roundtrip with empty messages: $L = RTT(0)/2 - g(0)$. As $RTT(m) = o + L_{LogP} + o + o + L_{LogP} + o$ (according to the LogP model) and $L_{LogP} = L + g(1) - o_s(1) - o_r(1)$ (see Table 1), the PLogP gap for $m$ bytes will be $g(m) = (o_s(m) + o_r(m)) + 2(g(1) - o_s(1) - o_r(1)) - (g(0) - o_s(0) - o_r(0))$, which is also non-obvious.

The assumption used in the optimized method for estimation of $g(m)$may not be accurate. For example, in our experiments on switch-based Ethernet clusters, we observed that the estimations of the gap obtained by the indirect method could be several times less than the (accurate) value obtained by the direct method. Moreover, the gap values found with the optimized technique are often less then send/receive overheads for small and medium messages, which contradicts the assumption that $g(m) \geq o_s(m)$ and $g(m) \geq o_r(m)$. Thus, the direct method is the only reliable way of the accurate estimation of the gap values. The main drawback of the direct technique is its high cost. Of course, this software can be efficiently used for estimation of the LogP/LogGP parameters, as they require the measurements only for three different

message sizes (0, 1, and $m$ bytes, where $m$ is sufficiently large) (Table 1).

None of the above point-to-point models reflects heterogeneity of the processors. Our point-to-point model [9] does reflect it in terms of node-specific fixed and variable processing delays. The parameters of those homogeneous models as well as the techniques of measurements cannot be used to build our heterogeneous communication performance model. If we follow the traditional approach, we will have to perform the same set of measurements for each pair of heterogeneous processors, which will require numerous measurements and take an unacceptably long time. In the paper, we propose a technique that allows us to find accurate estimations of the parameters of the heterogeneous point-to-point model with a relatively small number of measurements.

TABLE I
LOGP/LOGGP PARAMETERS EXPRESSED IN TERMS OF PLOGP

| LogP/LogGP | PLogP |
|---|---|
| $L$ | $L+g(1)-o_s(1)-o_r(1)$ |
| $o$ | $(o_s(1)+o_r(1))/2$ |
| $g$ | $g(1)$ |
| $G$ | $g(m)/m$, for a sufficiently large $m$ |
| $P$ | $P$ |

The models of collective communications are usually designed as a combination of the point-to-point communications. Thakur et al. [10] 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. [11] 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. [12] applied the Hockney, LogP/LogGP and PLogP models to different algorithms and topologies for barrier, broadcast, reduce, and all-to-all operations. They compared the predictions with measurements and presented the optimized collective operations based on the decision functions that switch between different algorithms, topologies, and message segment sizes.

Traditionally, the research on optimization of collective communications focuses on the analysis of such collective communication operations that allow for many different algorithms of implementation via point-to-point communications using different tree topologies. The goal is to find the optimal algorithm for each particular network configuration. The examples of such communication are broadcast and reduce. The scatter and gather operations, which are widely used in applications but not allow for that many implementations, have not received that much attention. The scatter and gather implementations are usually based on the flat tree topology (although recent versions of MPICH already use more efficient algorithms based on minimal spanning trees). In our research on performance of collective communications, we decided to start from modelling scatter and gather, which allowed us to focus on more fundamental properties of collective communications on a switched network rather than on different algorithms of their implementations.

In the work on communication performance, the problems of benchmarking the communication operations are often discussed. The execution time of communication operations can be measured:

- at a designated process,
- at the processes taking the longest time during the operation, and
- between the first process starting and the last process finishing the operation.

Worsh et al. [13] reported some drawbacks of the first two approaches. The synchronization (for example, barrier or zero-sized message passing) is necessary to ensure that all processes have finished the communication operations. It results in overlapping communication operations and increasing the communication time with the number of processes involved. They suggested an algorithm based on global time, which provides accurate measurement of the time between the first and last processes for any collective operations. It operates local times on processors, synchronizes the clocks, and averages the results obtained in the series of measurements.

As we only need to measure the execution time of point-to-point communication, we use the first approach, which requires a minimum of experiments, in combination with round trip messaging for synchronization.

## III. POINT-TO-POINT COMMUNICATION MODEL FOR HETEROGENEOUS CLUSTERS BASED ON A SWITCHED NETWORK

In this section, our point-to-point model presented in [9] is described and compared to other models. In heterogeneous clusters, the nodes have different characteristics. Therefore, it is appropriate to have different point-to-point parameters reflecting nodal contributions in the communication execution time. In clusters based on a switched network, each node can directly communicate to every other node via a single switch. This allows us to reduce the number of parameters for transmission delays.

Like most of point-to-point communication models, except for PLogP, our model is linear, representing 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; $\beta_{ij}$ is the transmission rate. Different parameters for nodal delays reflect the heterogeneity of the processors. 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 $\alpha$ of the Hockney model for $i \xrightarrow{M} j$, $j \xrightarrow{M} k$, $k \xrightarrow{M} i$ point-to-point communications can be used to find fixed processing delays:

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

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

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

but it is not sufficient to find the other parameters.

The meaning of the parameters and the assumption that the execution time of roundtrip be $RTT(m) = L + g(m) + L + g(0)$ makes it impossible to use the PLogP model in finding the parameters of our point-to-point model.

## IV. Design of Communication Experiments for Estimation of the Point-to-Point Parameters

In this section, the design of communication experiments for estimation of the point-to-point parameters of the heterogeneous communication model is presented. The proposed design addresses the following two challenges:

- Finding a set of experiments resulting in a sufficient number of linearly independent linear equations, whose variables represent the unknown point-to-point parameters.
- Minimization of the total execution time of the experiments.

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 most accurate way to measure the execution time of MPI point-to-point communications is the measurement of round trip messaging. The execution time of sending $M_1$ bytes and receiving $M_2$ bytes between nodes $i \xrightleftarrows[M_2]{M_1} j$ is equal to

$$T_{ij}(M_1, M_2) = (C_i + M_1 t_i + C_j + M_1 t_j + \frac{M_1}{\beta_{ij}}) + (C_i + M_2 t_i + C_j + M_2 t_j + \frac{M_2}{\beta_{ij}}).$$

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 $i \xrightleftarrows[0]{0} j$, $j \xrightleftarrows[0]{0} k$, $k \xrightleftarrows[0]{0} i$ ($i < j < k$):

$$\begin{pmatrix} 2 & 2 & 0 \\ 0 & 2 & 2 \\ 2 & 0 & 2 \end{pmatrix} \begin{pmatrix} C_i \\ C_j \\ C_k \end{pmatrix} = \begin{pmatrix} T_{ij}(0) \\ T_{jk}(0) \\ T_{ki}(0) \end{pmatrix}.$$

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 $\frac{(n-1)(n-2)}{2}$ values obtained from all the different systems of equations.

The $C_n^2$ experiments $i \xrightleftarrows[0]{M} j$ ($i < j$) give us the same number of the following equations: $t_i + t_j + \frac{1}{\beta_{ij}} = \frac{T_{ij}(M) - 2C_i - 2C_j}{M}$. To find $n$ variable processing delays $t_i$ and $C_n^2$ transmission rates $\beta_{ij}$ we need at least $n$ more independent equations. We obtain the equations from the communications $i \xrightleftarrows[0]{M} j,k$, where the source processor sends the messages of the same size to two processors and receives zero-sized messages from them The design of these additional experiments takes into account non-linear behaviour of MPI one-to-many and many-to-one communications that we observed on switched networks [4], [9]. Namely, the execution time of many-to-one gather-like communications can non-deterministically escalate for a particular range of medium message sizes. Therefore, we chose to gather zero-sized messages in order to avoid the non-deterministic escalations. We also observed the leap in the execution time of one-to-many scatter-like operations for large messages. Therefore, $M$ is taken small enough. In this case, the source node does not wait until the current message has reached the destination node and can start sending next message. In this 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 $\max(2C_j + Mt_j + \frac{M}{\beta_{ij}}, 2C_k + Mt_k + \frac{M}{\beta_{ik}})$. Thus, the

execution time $T_i(M)$ of one-to-two communications with root $i$ can be expressed by $T_i(M) = 4C_i + 2Mt_i + \max(2C_j + Mt_j + \frac{M}{\beta_{ij}}, 2C_k + Mt_k + \frac{M}{\beta_{ik}})$.

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

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

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

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

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 turn into $T_{ij}(M) > T_{ik}(M)$. For the communications with other roots $j \xleftrightarrow[0]{M} i,k$, $k \xleftrightarrow[0]{M} i,j$, there will be similar expressions for $t_j$ and $t_k$:

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

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

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 $\frac{1}{\beta_{ij}} = \frac{T_{ij}(M) - 2C_i - 2C_j}{M} - t_i - t_j$. 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:

$$\begin{cases} t_i = \frac{T_i(M) - T_{ij}(M) - 2C_i}{M} \\ t_j = \frac{T_j(M) - T_{ji}(M) - 2C_j}{M} \\ t_k = \frac{T_k(M) - T_{ki}(M) - 2C_k}{M} \end{cases}$$

$$\frac{1}{\beta_{ij}} = \frac{T_{ij}(M) - 2C_i - 2C_j}{M} - t_i - t_j$$

$$\frac{1}{\beta_{jk}} = \frac{T_{jk}(M) - 2C_j - 2C_k}{M} - t_j - t_k$$

$$\frac{1}{\beta_{ki}} = \frac{T_{ki}(M) - 2C_k - 2C_i}{M} - t_k - t_i.$$

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

This approach can also be extended to the communications $i \xleftrightarrow[0]{M} j_1,...,j_k$ ($j_1 < ... < j_k$). This will require $(1+k)C_n^{1+k}$ experiments to perform, $(k+1)(k-1)$ inequalities to check, and $1 + k + C_{k+1}^2$ equations to solve. We also considered some other communication experiments for estimation of the point-to-point parameters that require multiple processors in a single communication, much more measurements, and complicated calculations.

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 averaging of $2n + C_n^2$ values).

As the parameters of our point-to-point model are found in a small number of experiments, they can be sensitive to the inaccuracy of measurements. 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. One advantage of our design is that these series do not have to be long (typically, up to ten in a series) because all the parameters have been already averaged within the procedure of their finding.

## V. EXPERIMENTAL RESULTS

In this section, we present the experimental results demonstrating that the proposed technique allows us to accurately estimate the parameters of the heterogeneous point-to-point communication model. We also demonstrate that the analytical models of one-to-many and many-to-one communications built from the heterogeneous point-to-point communication model, parameters of which are obtained this way, are in good agreement with the experiments. We show that our design of experiments takes less time then traditional approaches.

We carried out the experiments with various MPI implementations and different clusters. This paper presents the experimental results obtained on the following platforms:

- **LAM-Fast:** 8 x Sun Ultra 5/10, Fast Ethernet, LAM 7.1.3
- **OpenMPI-Giga:** 11 x Intel Xeon 2.8/3.4/3.6, 2 x P4 3.2/3.4, 1 x Celeron 2.9, 2 x AMD Opteron 1.8, Gigabit Ethernet, Open MPI 1.1.4

The execution time of a single point-to-point communication measured for different message sizes is compared with the predictions of the LogGP, PLogP and our point-to-point models on OpenMPI-Giga cluster (Fig. 1). The PLogP model is piecewise linear. It includes a lot of empirical data in the functional parameters, and reflects the deviations of the execution time from the linear predictions. The linear predictions of LogGP and our point-to-point models are almost the same.

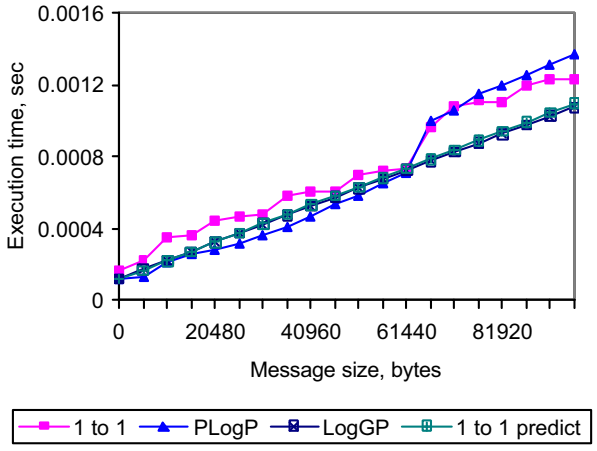



Fig. 1 Comparison of the predictions of the point-to-point models on OpenMPI-Giga cluster

All point-to-point models considered in this paper use a lot 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 2, the number of measurements is estimated for each model and the time they take on LAM-Fast and OpenMPI clusters is shown. LogGP and PLogP parameters are found with help of the logp_mpi library [8].

In Table 2, 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 $i$-th experiment of which consists of $2^i$ sendings, and
- $2mr$ experiments on $i \xleftarrow[0]{M} j$ and $i \xleftarrow[M]{0} j$ roundtrips, where:
  - $r$ is the number of roundtrips required to obtain more accurate send and receive overheads (the averaged execution time of the roundtrips $i \xleftarrow[0]{M} j$ 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 2 that estimate the total number of measurements, we use the averaged values of $s$, $r$ and $m$. The saturation experiments take much more time then single roundtrips as they include up to $2^s$ sendings. The direct 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 gap values and two roundtrips with the message of 1 byte to estimate the values of the send/receive overheads.

TABLE II
COMPARISON OF THE MEASUREMENT COSTS OF THE POINT-TO-POINT MODELS ON DIFFERENT PARALLEL PLATFORMS

| Comm. Model | Number of measurements | LAM-Fast time, sec | OpenMPI-Giga time, sec |
|---|---|---|---|
| **Hockney** | $kC_n^2$ | 0.28384 | 0.17326 |
| **LogGP** | $3sC_n^2 + 2rC_n^2$ | – | – |
| **PLogP** | $sC_n^2 + 2mrC_n^2$<br>$msC_n^2 + 2mrC_n^2$ | 334.048066 | 63.110291 |
| **p2p** | $k_0C_n^2 + k_1C_n^2 + k_2 3C_n^3$ | 2.147378 | 0.332256 |

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

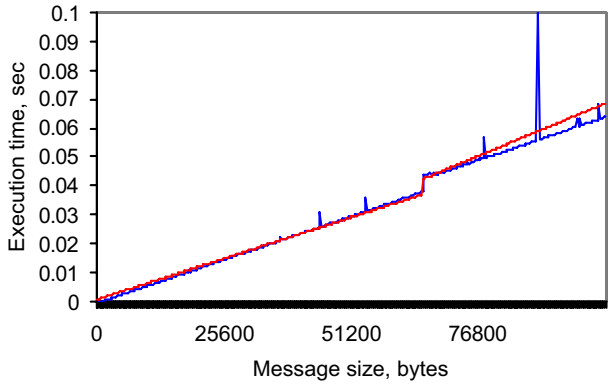
In our experiments, no more then ten measurements in a series were needed to achieve the acceptable accuracy.

Our point-to-point model was designed to serve as a basis for modelling collective operations on heterogeneous clusters based on a switched network. In next experiments, we check the accuracy of two such models given their point-to-point parameters are estimated with the procedure proposed in the paper.

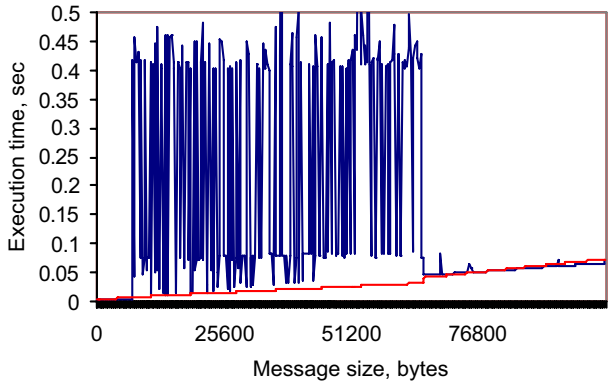
Collective communication operations can be implemented by using a wide range of algorithms taking into account such factors as topology, number of processors, message sizes. Here, we consider two collective operations, scatter and gather, and straightforward algorithms of their implementation based on flat tree topology. Namely, these operations are implemented as follows: the root executes $n$ send (or receive) operations and each process executes a receive (or a send). The models of these operations are built upon the heterogeneous point-to-point model and presented in [4]. Along with the point-to-point parameters, the models include a small number of parameters that reflect some features specific to each particular collective operation that are found empirically. They can be either a constant or a function of a number of processor and a message size and can also depend on the parallel platform and MPI implementation. The finding of the additional parameters is out the scope of this paper.

The one-to-many model [4] reflects the leap in the execution time and categorizes the small and large messages. Parameter $S$ is a message size threshold, separating small and large messages. It is different for different combinations of clusters and MPI implementations. The estimated time of scattering messages of size $M$ from node 0 to nodes $1,2,...,n$ is given by $C_0 + t_0 \times n \times M + \max_{1 \le i \le n} \left\{ C_i + t_i M + \frac{M}{\beta_{0i}} \right\}$, if $M \le S$, and $C_0 + t_0 \times n \times M + \sum_{i=1}^{n} (C_i + t_i M + \frac{M}{\beta_{0i}})$, if $M > S$, where $C_0$, $t_0$, $C_i$, $t_i$ are the fixed and variable processing delays on the source node and destinations. The one-to-many model displays parallel communication for small messages and a serialized communication for large messages. Fig. 2, a shows the prediction and the observation of the execution time of one-to-many communications for different message sizes.

The many-to-one model [4] differentiates small, medium and large messages by introducing parameters $M_1$ and $M_2$. 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 involving $n$ processors ($n \le N$, where $N$ is the cluster size) is estimated by $n(C_0 + t_o M) + \max_{1<i\le n} \left\{ C_i + t_i M + \frac{M}{\beta_{0i}} \right\} + \kappa_1 M$, 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 for increasing message size. Hence, this part of the model has the same design but a different slope of linearity and greater value due to overheads: $n(C_0 + t_0 M) + \sum_{i=1}^{n} (C_i + t_i M + \frac{M}{\beta_{0i}}) + \kappa_2 M$. The additional parameter $\kappa_2 = const$ is a fitting constant for correction of the slope.




(a)




(b)

Fig. 2 Modelling scatter and gather on LAM-Fast cluster

For medium messages, $M_1 \le M \le M_1$, we observed a small number of discrete levels of escalation, remaining constant as the message size increases. The model describes the probability of escalation to each of the levels as a function of message size and the number of processors involved in the

operation. If no escalation occurs, the linear model used for small massages will accurately predict the execution time. The prediction of many-to-one communications for different message sizes is shown in Fig. 2, b. A line for small messages continues even for medium sized messages, but the software implemented our communication performance model also provides the determining the levels of escalations and their probabilities.

## VI. CONCLUSION

This paper has described the point-to-point communication performance model of heterogeneous clusters based on a switched network and proposed the efficient technique for accurate estimation of its parameters. This technique includes a relatively small number of measurements of the execution time of one-to-one and one-to-two roundtrip 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 has been validated experimentally. Our future work includes development of a software package implementing the proposed technique.

## ACKNOWLEDGMENT

The work was supported by the Science Foundation Ireland.

## REFERENCES

[1] A. Lastovetsky, "Adaptive parallel computing on heterogeneous networks with mpC," *Parallel Computing*, 2002, vol. 28, pp. 1369-1407.

[2] A. Lastovetsky, R. Reddy, "HeteroMPI: Towards a message-passing library for heterogeneous networks of computers," *Journal of Parallel and Distributed Computing*, 2006, vol. 66, pp. 197-220.

[3] D. Arnold, H. Casanova, J. Dongarra, "Innovation of the NetSolve Grid Computing System," *Concurrency: Practice and Experience*, 2002, vol. 14, pp. 1457-1479.

[4] A. Lastovetsky, M. O'Flynn, "A Performance Model of Many-to-One Collective Communications for Parallel Computing," in *Proceedings of the 21st International Parallel and Distributed Processing Symposium (IPDPS 2007)*, 2007.

[5] R. Hockney, "The communication challenge for MPP: Intel Paragon and Meiko CS-2," *Parallel Computing*, 1994, vol. 20, pp. 389-398.

[6] D. Culler, R. Karp, D. Patterson, A. Sahay, K.E. Schauser, E. Santos, R. Subramonian, T. von Eicken, "LogP: Towards a realistic model of parallel computation," in *Proceedings of the fourth ACM SIGPLAN symposium on Principles and practice of parallel programming*, 1993, pp. 1-12.

[7] A. Alexandrov, M.F. Ionescu, K.E. Schauser, C. Scheiman, "LogGP: Incorporating long messages into the LogP model," in *Proceedings of the seventh annual ACM symposium on Parallel algorithms and architectures*, 1995, pp. 95-105.

[8] T. Kielmann, H. Bal, K. Verstoep, "Fast measurement of LogP parameters for message passing platforms," J.D.P. Rolim, Ed., *IPDPS Workshops*, ser. Lecture Notes in Computer Science, Cancun, Mexico: Springer-Verlag, 2000, vol. 1800, pp. 1176-1183.

[9] A. Lastovetsky, I. Mkwawa, M. O'Flynn, "An Accurate Communication Model of a Heterogeneous Cluster Based on a Switch-Enabled Ethernet Network," in *Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS 2006)*, 2006, pp. 15-20.

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

[11] T. Kielmann, R. F. H. Hofman, H. Bal, A. Plaat, R. A. F. Bhoedjang, "MagPIe: MPI's collective communication operations for clustered wide area systems," in *Proceedings of PPoPP*, 1999, pp. 131–140.

[12] J. Pjesivac-Grbovic, T. Angskun, G. Bosilca, G. E. Fagg, E. Gabriel, J. J. Dongarra, "Performance Analysis of MPI Collective Operations," in *Proceedings of 19th International Parallel and Distributed Processing Symposium (IPDPS 2005)*, 2005.

[13] T. Worsch, R. Reussner, W. Augustin, "On Benchmarking Collective MPI Operations," *Recent Advances in Parallel Virtual Machine and Message Passing Interface: 9th European PVM/MPI Users' Group Meeting*, ser. Lecture Notes in Computer Science, Berlin / Heidelberg Springer, 2002, vol. 2474, pp. 271-279.