A Software Tool for Accurate Estimation of Parameters of Heterogeneous Communication Models

Alexey Lastovetsky, Vladimir Rychkov, and Maureen O'Flynn

School of Computer Science and Informatics, University College Dublin, Belfield, Dublin 4, Ireland
{alexey.lastovetsky,vladimir.rychkov,maureen.oflynn}@ucd.ie
http://hcl.ucd.ie

Abstract. Analytical communication performance models play an important role in prediction of the execution time of parallel applications on computational clusters, especially on heterogeneous ones. Accurate estimation of the parameters of the models designed for heterogeneous clusters is a particularly challenging task due to the large number of parameters. In this paper, we present a set of communication experiments that allows us to get the accurate estimation of the parameters with minimal total execution time, and software that implements this solution. The experiments on heterogeneous cluster demonstrate the accuracy and efficiency of the proposed solution.

Keywords: Heterogeneous cluster, heterogeneous communication performance model, MPI, communication model estimation.

1 Introduction

Heterogeneous computational clusters have become a popular platform for parallel computing with MPI as their principle programming system. Unfortunately, many MPI-based applications that were originally designed for homogeneous platforms do not have the same performance on heterogeneous platforms and require optimization. The optimization process is typically based on the performance models of heterogeneous clusters, which are used for prediction of the execution time of different configurations of the application, including its computation and communication costs. The accuracy of the performance models is very influential in determining the efficiency of parallel applications. The optimization of communications is an important aspect of the optimization of parallel applications. The performance of MPI collective operations, the main constituent of MPI, may degrade on heterogeneous clusters. The implementation of MPI collective operations can be significantly improved, by taking the communication performance model of the executing platform into account.

Traditionally, communication performance models for high performance computing are analytical and built for homogeneous clusters. The basis of these

A. Lastovetsky et al. (Eds.): EuroPVM/MPI 2008, LNCS 5205, pp. 43–54, 2008.

models 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. 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. Thus, in this case, the heterogeneous cluster will be treated as homogeneous in terms of the performance of communication operations.

When some processors or links in the heterogeneous cluster significantly differ in performance, predictions based on the homogeneous communication model may become inaccurate. More accurate performance models would not average the point-to-point communication parameters. On the other hand, the taking into account the parameters for each pair of processors will make the total number of point-to-point parameters and the amount of time required to estimate them significantly larger. In [1], [2], 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. 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. At the same time, the design of communication experiments for accurate and efficient estimation of the parameters of this model is not a trivial task.

Usually, to estimate the point-to-point parameters, different variations of sending/receiving messages between two processors are used. As regards the heterogeneous model proposed in [1], [2], with point-to-point communications only, we cannot collect enough data to estimate the parameters, and therefore must conduct some additional independent experiments. We design these additional communication experiments as a combination of scatter and gather. The observation of scatter and gather on the clusters based on a switched network show that the execution time may be non-linear and non-deterministic, especially if the MPI software stack includes the TCP/IP layer. Therefore, in our design we take into account all the irregularities, which might make the estimation inaccurate, and carefully select the message size.

The statistical methods of finding the point-to-point parameters, normally used in the case of homogeneous communication models, will result in unacceptably large number of measurements if applied as they are to the 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 managed to reduce the number of measurements with the same accuracy as the exhaustive statistical analysis.

To the best of the authors' knowledge, there are no other publications describing heterogeneous communication performance models of computational clusters and the accurate estimation of the parameters of such models. In this paper, we present the software tool that automates the estimation of the heterogeneous communication performance model of clusters based on a switched network. The software tool can also be used in the high-level model-based optimization of MPI collective operations. This is particularly important for heterogeneous platforms where the users typically have neither authority nor knowledge for making changes in hardware or basic software settings.

This paper is organized as follows. In Section 2, related work on estimation of the parameters of communication performance models is discussed. In Section 3, we describe the point-to-point model of heterogeneous clusters based on a switched network and the design of communication experiments required to estimate its parameters. Section 4 presents the software tool for the estimation of the parameters of the heterogeneous communication performance model and the experimental results that demonstrate the accuracy and efficiency of the proposed solution.

2 Related Work

In this section, we discuss how the parameters of existing communication performance models are estimated. As all these models are built for homogeneous platforms, their parameters are the same for all processors and links. Therefore, to estimate them, it is sufficient to perform a set of communication experiments between any two processors.

The Hockney model [3] of the execution time of point-to-point communication is $\alpha + \beta m$, where α is the latency, β is the bandwidth and m is the message size. There are two ways to obtain a statistically reliable estimation of the Hockney parameters:

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

The LogP model [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 gap, g, is the minimum time between consecutive transmissions or receptions; it is the reciprocal value of the end-to-end bandwidth between two processors, so that the network bandwidth can be expressed as L/g. According to LogP, the time of point-to-point communication can be estimated by $L + 2o$. In [5], the estimation of the LogP

parameters is presented, with the sending, o_s, and receiving, o_r, overheads being distinguished. The set of communication experiments used for estimation of the LogP parameters is as follows:

- To estimate the sending overhead parameter, o_s, a small number of messages are sent consecutively in one direction. The averaged sending time measured on the sender side will approximate o_s.
- The receiving overhead, o_r, is found directly from the time of receiving a message in the roundtrip. In this experiment, after completion of the send operation, the sending processor waits for some time, sufficient for the reply to reach the receiving processor, and only then posts a receive operation. The execution time of the receive operation is assumed to approximate o_r.
- The latency is found from the execution time of the roundtrip with small message $L = RTT/2 - o_s - o_r$.
- To estimate the gap parameter, g, a large number of messages are sent consecutively in one direction. The gap is estimated as $g = T_n/n$, where n is a number of messages and T_n is the total execution time of this communication experiment measured on the sender processor. The number of messages is chosen to be large to ensure that the point-to-point communication time is dominated by the factor of bandwidth rather than latency. This experiment, also known as a saturation, reflects the nature of the gap parameter but takes a long time.

In contrast to the Hockney model, LogP is not designed for the communications with arbitrary messages, but there are some derivatives, such as the LogGP model [6], which takes into account the message size by introducing the gap per byte parameter, G. The point-to-point communication time is estimated by $L + 2o + (m-1)G$. The gap per byte, G, can be assessed in the same way as the gap parameter of the LogP model, saturating the link with large messages M, $G = g/M$.

In the PLogP (parameterized LogP) model [10], all parameters except for latency are piecewise linear functions of the message size, and the meaning of parameters slightly differs from LogP. The meaning of latency, L, is not intuitive; rather it is a constant that combines all fixed contribution factors such as copying to/from the network interfaces and the transfer over the network. The send, $o_s(m)$, and receive, $o_r(m)$, overheads are the times that the source and destination processors are busy for the duration of communication. They can be overlapped for sufficiently large messages. The gap, $g(m)$, is the minimum time between consecutive transmissions or receptions; 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 $g(m) \geq o_r(m)$. According to the PLogP model, the point-to-point execution time is equal to $L + g(m)$ for the message of m bytes. The estimation of the PLogP parameters includes the experiments which are similar to the LogP ones but performed for different message sizes. Although this model is adaptive in nature, because of the number and location of breaks of piecewise linear functions are determined while the model is being built, the total number of parameters may become too large.

There are two main approaches to modeling the performance of communication operations for heterogeneous clusters. The first one is to apply traditional homogeneous communication performance models to heterogeneous clusters. In this case, the parameters of the models are estimated for each pair of processors and the average values for all pairs are then used in modelling. The second approach is to use dedicated heterogeneous models, where different pairs of heterogeneous processors are characterized by different parameters. While simpler in use, the homogeneous models are less accurate. When some processors or links in the heterogeneous cluster significantly differ in performance, predictions based on the homogeneous models may become quite inaccurate. The number of communication experiments required for the accurate estimation of both homogeneous and heterogeneous models will be of the same order, $O(n^2)$.

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

The alternative approach is to use original point-to-point heterogeneous models that allow for easy and intuitive expression of the execution time of collective communication operations such as the LOM model [1], [2] designed for switched heterogeneous clusters. While easy and intuitive in use, these models encounter a new challenging problem. The problem is that the number of point-to-point parameters describing communication between a pair of processors becomes larger than the number of independent point-to-point communication experiments

traditionally used for estimation of the parameters. In this paper, we describe the set of communication experiments sufficient for the accurate and efficient estimation of the parameters and present the software tool that implements this approach.

3 Heterogeneous Communication Performance Model and Its Estimation

The LOM model [1] includes both link-specific and processor-specific parameters. Like most of point-to-point communication models, its point-to-point parameters represent the communication time by a linear function of the message size. The execution time of sending a message of M bytes from processor i to processor j in a heterogeneous cluster $i \xrightarrow{M} j$ is estimated by $C_i + t_i M + C_j + t_j M + \frac{M}{\beta_{ij}}$, where C_i, C_j are the fixed processing delays; t_i, t_j are the delays of processing of a byte; 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 $\beta_{ij} = \beta_{ji}$.

To estimate the parameters of such a model, an approach with roundtrip point-to-point experiments is not enough. For a network consisting of n processors, there will be $2n + C_n^2$ unknowns: n fixed processing delays, n variable processing delays, and C_n^2 transmission rates. The execution time of the roundtrip, namely sending M_1 bytes and receiving M_2 bytes between nodes $i \xleftrightarrow[M_1]{M_2} j$, is equal to $T_{ij}(M_1, M_2) = (C_i + t_i M_1 + C_j + t_j M_1 + \frac{M_1}{\beta_{ij}}) + (C_i + t_i M_2 + C_j + t_j M_2 + \frac{M_2}{\beta_{ij}})$. The roundtrip experiments will give us only C_n^2 equations. Therefore, the first challenge we face is to find a set of experiments that gives a sufficient number of linearly independent linear equations, whose variables represent the unknown point-to-point parameters.

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

In order to find the rest $n + C_n^2$ parameters, we might use the roundtrips with non-empty message, but it would give us only C_n^2 linearly independent equations. Instead, we use the additional experiments, which include communications from one processor to two others and backward, and express the execution time of the communication experiments in terms of the heterogeneous point-to-point communication performance model. As will be shown below, the set of point-to-point and point-to-two communication experiments is enough to find the fixed processing delay and transmission rates, but there is one more important issue to be addressed. The point-to-two experiments are actually a particular combination of scatter and gather. The scatter and gather operations may have some

irregular behaviour on the clusters based on a switched network, especially if the MPI software stack includes the TCP/IP layer. Therefore, the message sizes for the additional experiments have to be carefully selected to avoid these irregularities.

We observed the leap in the execution time of scatter for large messages and the non-deterministic escalations of the execution time of gather for medium-sized messages (see Fig. 1). It prompted us introduce the particular threshold parameters to categorize the message size ranges where distinctly different behaviour of the collective MPI operations is observed, and to apply different formula for these regions to express the execution time with the heterogeneous point-to-point parameters.

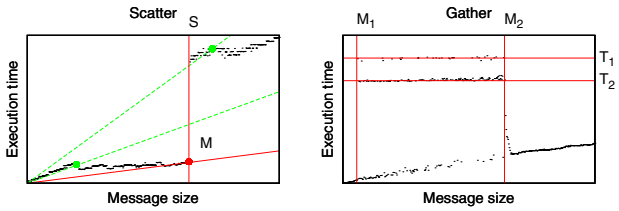

Fig. 1. The execution time of collective communications against the message size

The estimated time of scattering messages of size M from node 0 to nodes $1, ..., n$ is given by $n(C_0 + t_0M) + \max_{1\le i\le n}\{C_i + t_iM + \frac{M}{\beta_{0i}}\}$, if $M \le S$, and $n(C_0 + t_0M) + \sum_{1\le i\le n}\{C_i + t_iM + \frac{M}{\beta_{0i}}\}$, 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. This reflects the parallel communication for small messages and the serialized communication for large messages. The threshold parameter S corresponds to the leap in the execution time, separating small and large messages. It may vary for different combinations of clusters and MPI implementations.

For the gather operation, we separate 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 $n(C_0+t_0M)+\max_{1\le i\le n}\{C_i+t_iM+\frac{M}{\beta_{0i}}\}+\kappa_1M$, where $\kappa_1 = const$ is a fitting parameter for correction of the slope. For large messages, $M > M_2$, the execution time resumes a linear predictability with increasing message size. Hence, this part is similar in design but has a different slope of linearity that indicates greater values due to overheads: $n(C_0 + t_0M) + \sum_{1\le i\le n}\{C_i+t_iM+\frac{M}{\beta_{0i}}\}+\kappa_2M$. The additional parameter $\kappa_2 = const$ is a fitting constant for correction of the slope. For medium messages, $M_1 \le M \le M_2$, we observed a small number of discrete levels of escalation, that remain constant as the message size increases.

Thus, following the model of scatter and gather, in our experiments we gather zero-sized messages in order to avoid the non-deterministic escalations. For

scatter, the message size M is taken less than the value of the threshold parameter S. The wrong selection of the message size can make the estimation of the point-to-point parameters inaccurate, which is shown in Fig. 1. c In order to find variable processing delays t_i and transmission rates β_{ij}, we measure the execution time of the C_n^2 experiments $i \xleftrightarrow[0]{M} j$ $(i < j)$, the roundtrips with empty replies, and the C_n^3 experiments $i \xleftrightarrow[0]{M} j, k$ $(i < j < k)$, where the source processor sends the messages of the same size to two processors and receives zero-sized messages from them. The execution time $T_i(M)$ of one-to-two communications with root i can be expressed by $T_i(M) = 4C_i + 2t_iM + \max(2C_j + t_jM + \frac{M}{\beta_{ij}}, 2C_k + t_kM + \frac{M}{\beta_{ik}})$. The execution times of these experiments are used in the following formula to get the values of the variable processing delays and then the values of transmission rates:

$$t_i = \begin{cases} \frac{T_i(M) - T_{ij}(M) - 2C_i}{M}, T_{ij}(M) > T_{ik}(M) \\ \frac{T_i(M) - T_{ik}(M) - 2C_i}{M}, T_{ik}(M) > T_{ij}(M) \end{cases} \qquad \frac{1}{\beta_{ij}} = \frac{T_{ij}(M) - 2C_i - 2C_j}{M} - t_i - t_j$$

As the parameters of our point-to-point model are found in a small number of experiments, they can be sensitive to the inaccuracies of measurement. Therefore, it makes sense to perform a series of the measurements for one-to-one and one-to-two experiments and to use the averaged execution times in the corresponding linear equations. Minimization of the total execution time of the experiments is another issue that we address. The advantage of the proposed design is that these series do not have to be lengthy (typically, up to ten in a series) because all the parameters have been already averaged with the process of their finding.

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

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

To find the threshold parameters, we use the algorithm proposed in [8]. It considers the statistical linear models with multiple structural changes and uses dynamic programming to identify optimal partitions with different numbers of segments. The algorithm allows us to locate the break in the execution time of scatter, S, and the range of large messages for gather, M_2.

Then we perform the linear regression of the execution time of gather on this range to estimate the slope correction parameter κ_2, that is used to adjust

the prediction of many-to-one execution time for large messages. The linear regression gives us two values c_0 and c_1: $T \approx c_0 + c_1 M$, $M > M_2$. The slope correction parameter κ_2 is found as follows: $\kappa_2 = c_1 - \sum_{i=1}^{n}(t_i + \frac{1}{\beta_{0i}})$. We find M_1 as $M_1 \approx M_k$, $k = min\{i : T^{i+1}/T^1 > 10\}$. The linear regression on the data row $\{M^i, T^i\}$, $i = 1, ..., k$ is performed to obtain the linear parameters for the small messages, $T \approx c_0 + c_1 M$, $M > M_1$, and to calculate the slope correction parameter $\kappa_1 = c_1 - max_{1 \le i \le n}\{t_i + \frac{1}{\beta_{0i}}\}$.

4 The Software Design and Experimental Results

To the best of the authors' knowledge, there are no available software tools for the estimation of heterogeneous communication models of computational clusters. In this section, we present such a software tool, and describe its features and design.

We design the software tool in the form of a library implemented on top of MPI. In addition to the library, the software tool provides a command line utility that can be used for one-time estimations. The utility uses the library to estimate the parameters of the heterogeneous communication performance model with the given accuracy and saves the data in a file that can be used later. One-estimation may be done during the installation of the software tool, or each time the parallel platform or MPI implementation has been changed. The estimation can also be performed in the user application at runtime, with the invocation of the library functions. The library consists of three modules:

1. The Measurement module is responsible for the measurement of the execution time of the communication experiments required to estimate the parameters of the heterogeneous model. It uses the MPIBlib benchmarking library [7], namely, the point-to-point, scatter and gather benchmarks. In addition, the Measurement module includes the function for measuring the execution time of the point-to-two communication experiments, $i \xleftarrow[0]{M} j, k$, required to find the variable processing delays and transmission rates. The point-to-point and point-to-two experiments are optimized for clusters with a single switch. As network switches are capable of forwarding packets between sources and destinations appropriately, several point-to-point or point-to-two communications can be run in parallel, with each process being involved in no more than one communication. This decreases the execution time the benchmark takes, giving quite accurate results.
2. The Model module provides the API, which allows the user to estimate the parameters of the heterogeneous communication performance model inside their application. This module uses the results of benchmarks provided by the Measurement module and the MPIBlib library, builds and solves the systems of equations described in the previous section. For estimation of the threshold parameters required to select the message size for point-to-two experiments, the **strucchange** library of the R statistical package is used [9]. It automates the detection of the structural changes in the linear regression models. The statistical analysis is performed with help of GSL

(GNU scientific library). More specifically, the parameters of the heterogeneous communication performance model are estimated within a confidence interval that indicates the reliability of estimation, which is implemented with help of GSL. For linear regression, the software tool uses GSL routines for performing least squares fits to experimental data.

3. The Optimization module provides a set of the optimized implementations of collective operations, such as scatter and gather, which use the parameters of the heterogeneous model [10].

To demonstrate the accuracy provided by the software tool, we compare the execution time of a single point-to-point communication observed for different message sizes with the predictions provided by the logp_mpi package [6] and by our software tool (Fig. 2). The logp_mpi package was used for the predictions of the PLogP and LogGP models. The experiments were carried out between two processors of the 16-node heterogeneous cluster, which has the following characteristics: 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, LAM 7.1.3. The PLogP model is more accurate but much more costly. The accuracy is due to the use of the functional parameters, each of which is approximated by a large number of scalar parameters. The linear predictions of LogGP and our point-to-point models are practically the same.

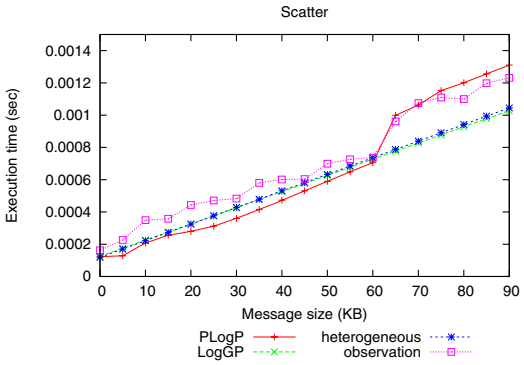

Fig. 2. The observed and predicted execution times of the point-to-point communication on the 16-node heterogeneous cluster

The point-to-point parameters estimated by the software tool are used in the analytical models of collective communication operations for prediction of their execution time. Therefore, the accuracy of estimation of these parameters can be validated by the comparison of the observed execution time of the collectives and the one predicted by the analytical models using the values of the point-to-point parameters estimated by the software tool. For the experiment, we use the linear scatter and gather, the analytical models of which are presented in Section 3. Fig. 3 shows the results of this experiment. One can see that the execution time of scatter is predicted with high accuracy. The same is true for gather, given

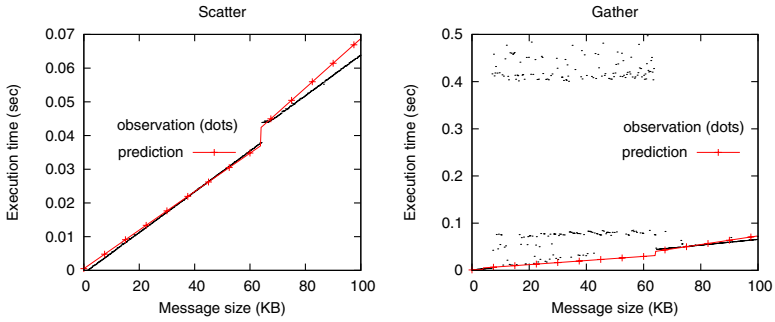

Fig. 3. The observed and predicted execution time of scatter and gather on the 16-node heterogeneous cluster

that the analytical model is not supposed to predict irregular escalations of the execution time for medium-sized messages.

Usually, the statistically reliable estimation is achieved by averaging the results of numerous repetitions of the same experiment. The software tool has an addition level of the averaging of the experimental results. Namely, each individual experiment produces multiple estimates of the same parameter that are also averaged. Consider, for example, the experiment estimating the fixed processing delays. When the execution time of the empty roundtrips between all pairs of processors has been measured, the fixed processing delay of a processor can be found in an identical manner from C_{n-1}^2 systems of equations, one for each of the C_{n-1}^2 triplets of the processors that include this processor. Therefore, the first approximation of the fixed processing delay will be calculated by averaging these C_{n-1}^2 values. For more accurate estimation, this communication experiment can be then repeated several times, giving several estimates of the fixed processing delay which can be further averaged. As a result, the number of the repetitions will be much smaller.

In total, the following series of repetitions are performed:

- a series of the k_0 repetitions for the experiment including C_n^2 empty roundtrips,
- a series of the k_1 repetitions for the experiment including C_n^2 one-to-one communications, and
- a series of the k_2 repetitions for the experiment including $3C_n^3$ one-to-two communications.

In our experiments on the 16-node heterogeneous cluster, no more than ten repetitions in a series were needed to achieve the acceptable accuracy. The estimation of the parameters took just fractions of a second, which allows us to use the library for the runtime estimation in user applications.

5 Conclusion

This paper has described the software tool for accurate estimation of parameters of the heterogeneous communication performance model. The software tool

implements the efficient technique that requires 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 the solution of simple systems of linear equations. The accuracy of estimation is achieved by averaging the values of the parameters, and careful selection of message sizes. The fast and reliable MPI benchmarking of point-to-point and collective operations also support efficiency and accuracy of the software tool. The software tool is freely available at `http://hcl.ucd.ie/project/CPM`

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

References

1. Lastovetsky, A., Mkwawa, I., O'Flynn, M.: An Accurate Communication Model of a Heterogeneous Cluster Based on a Switch-Enabled Ethernet Network. In: Proc. of ICPADS 2006, vol. 2, pp. 15–20. IEEE Computer Society Press, Los Alamitos (2006)
2. Lastovetsky, A., O'Flynn, M.: A Performance Model of Many-to-One Collective Communications for Parallel Computing. In: Proceedings of IPDPS 2007 (2007)
3. Hockney, R.: The communication challenge for MPP: Intel Paragon and Meiko CS-2. Parallel Computing 20, 389–398 (1994)
4. Culler, D., Karp, R., Patterson, D., Sahay, A., Schauser, K., Santos, E., Subramonian, R., von Eicken, T.: LogP: Towards a realistic model of parallel computation. In: Proceedings of PPoPP 1993, pp. 1–12. ACM, New York (1993)
5. Culler, D., Liu, L., Martin, R., Yoshikawa, C.: LogP Performance Assessment of Fast Network Interfaces. IEEE Micro. 16(1), 35–47 (1996)
6. Alexandrov, A., Ionescu, M., Schauser, K., Scheiman, C.: LogGP: Incorporating long messages into the LogP model. In: Proc. of SPAA 1995, pp. 95–105. ACM, New York (1995)
7. Kielmann, T., Bal, H., Verstoep, K.: Fast measurement of LogP parameters for message passing platforms. In: Rolim, J. (ed.) IPDPS-WS 2000. LNCS, vol. 1800, pp. 1176–1183. Springer, Heidelberg (2000)
8. Lastovetsky, A., Rychkov, V., O'Flynn, M.: MPIBlib: Benchmarking MPI Communications for Parallel Computing on Homogeneous and Heterogeneous Clusters. In: Lastovetsky, A., Kechadi, T., Dongarra, J. (eds.) EuroPVM/MPI 2008. LNCS, vol. 5205. Springer, Heidelberg (2008)
9. Bai, J., Perron, P.: Computation and Analysis of Multiple Structural Change Models. J. of Applied Econometrics 18, 1–22 (2003)
10. Zeileis, A., Leisch, F., Hornik, K., Kleiber, C.: Strucchange: An R package for testing for structural change in linear regression models. J. of Statistical Software 7(2), 1–38 (2002)
11. Lastovetsky, A., O'Flynn, M., Rychkov, V.: Optimization of Collective Communications in HeteroMPI. In: Cappello, F., Herault, T., Dongarra, J. (eds.) PVM/MPI 2007. LNCS, vol. 4757, pp. 135–143. Springer, Heidelberg (2007)