Predicting the Performance of MPI Applications over Different Grid Architectures

[1][2]**Ahmed Badri Muslim Fanfakh**

[1]*University of Babylon, College of Sciences for Woman, Computer Department, Iraq.*

[2]*FEMTO-ST Institute, University of Franche-Comté, IUT de Belfort-Montbéliard, France*

afanfakh@gmail.com

Abstract

Nowadays, the high speed and accurate optimization algorithms are required. In most of the cases, researchers need a method to predict some criteria with acceptable accuracy to use it after in their algorithms. However, in the field of parallel computing, the execution time can be considered the most important criteria. Consequently, this paper presents a new model to predict the execution time of message passing interface applications execute over numerous grid scenarios. The model has ability to predict the execution time of the message passing applications running over any grid configuration in term of different number of nodes and their computing powers. The experiments are conducted over SimGrid simulator to simulate the grid configuration scenarios. The obtained results of comparing the real and the predicted execution time show a good accuracy. The average error ratio between the real and the predicted execution time for three benchmarks are 4.36%, 5.79% and 6.81%.

Keywords: Execution time prediction, Parallel computing, MPI, Grid.

1-Intoduction

Grid architecture consists of a set of clusters which are geographically distributed. Each cluster has a group of homogenous nodes which are not similar to the node of the other clusters in term of their speed. However, grid platform can be considered as a heterogenous architecture. The differences in the computing power in a heterogenous cluster leads to imbalanced workloads when execute the parallel message passing programs over that heterogeneous platform. Imbalanced workloads produce idle times that happen when the fast nodes waiting for the slowest one. Thus, idle times increase drastically the running time of the parallel application. One of the most popular metrics to evaluate the performance of the parallel program is the speed-up. It is the ratio between the sequential execution time and the parallel execution time of an application that solving the same problem [1]. The message passing interface (MPI) applications are commonly used parallel applications in the distributed environment. These applications are composed of

computation and communication times. Speed-up factor always affected by these times directly. Thus, any increase or decrease in these times are propositionally increase or decrease the seed-up measure. In the case of increasing number of nodes, the computation time is decreased and so the speed-up factor is increased. While, the increase in the number of nodes increases the communication time, which decreased the speed-up factors. Then, the speed-up factor has a nonlinear form when it is applied to the MPI applications. Therefore, the main goal of a wide parallel techniques and models is to reduce the execution time of the parallel applications. However, many optimization techniques depend on the execution time prediction methods which help researchers in the process of making decisions for each new state in the dynamic environment. The prediction methods can be implemented using many tools such as: statistic tools, AI algorithms, heuristic methods and analytical mathematical modeling. Some of these tools are costly in term of time complexity in case of a lot of iterations needed to predict the execution time of the parallel application.

In this work, new prediction model that predicts the performance of MPI applications when running it over heterogeneous grid platform. The model predicting the execution time for any grid platform composed of different number of nodes. The accuracy of this model is tested by executing NAS parallel benchmarks over distributed clusters each with different nodes in term of their hardware types and numbers.

Some statistical and analytical models have been proposed in the literature to predict the execution time of parallel applications (High Performance LINPAC) benchmark to achieve maximum possible performance. To predict the running time of HPL benchmark, authors introduced an analytical model in [2]. They proposed a semi-empirical model to predict the performance with less possible error ratio. Anther optimization model is developed to work on three overheads: sending and receiving messages, computational and proposed communication method as in [3]. The error ratio of the proposed prediction model was less than 5% when implementing the model over different clusters. In [4], researchers specify the argument of the HPL benchmark that can be used to predict both the power consumption and performance. Authors showed piecewise polynomial regression and ANN to predict the execution time in [5]. Researchers in [6] implement a machine learning approach using multilayer neural networks. The work in [7] presented a new regression-based method to predict the scalability of the parallel program. The last approach provided prediction error ratio between 6.2% and 17.3%. The work introduced in [8] proposed dynamic model to predict program performance by considering the size of the problem over a constant number of nodes. Authors in [9], proposed compound method that merge historical prediction method with profile base technique to schedule the underline applications. The performance of applications that solved a large scaled problems was predicted in[10]. Authors introduced a prediction framework to execute parallel programs using a small training set. In [11,12,13,14,15,16], researchers have been proposed analytical execution time prediction models for NAS MPI programs over heterogenous cluster and grid.

The remainder of the paper is organized as follows: Section 2 describes the execution time of MPI applications over grid. Section 3 presents the proposed prediction model for execution time of MPI applications. Section 4 shows the experimental results. The paper ends with a conclusion and future works in Section 5

2-The execution time of parallel application running over grid

This paper is interested in predicting the execution time of message passing interface (MPI) applications running over grid. MPI programs are portable applications that can be executed over any parallel hardware without changing one line in their code. Therefore, each program is consisted of two parts: computation and communication sections. Both these sections are important in the process of modeling the execution time of these applications. Moreover, grid is a heterogenous parallel platform which is composed from a number of computing clusters. Each node in a cluster is different from other nodes of other cluster in the computing power. While they are similar in the computing power with the nodes of the same cluster. Figure (1) demonstrates an example of grid that composed of three clusters.
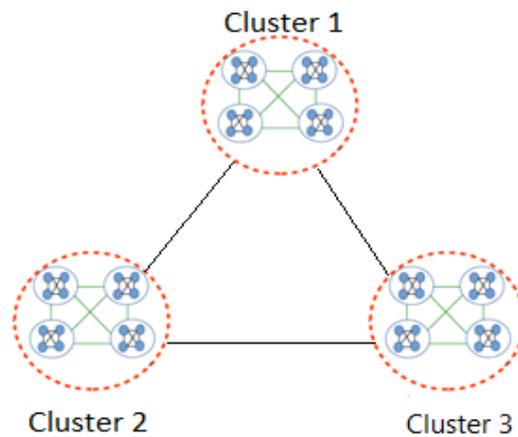
Figure (1): An example of a Grid composed of three clusters

The execution time of parallel tasks over a heterogenous grid results in different computation times according to the heterogeneity in the computing power. Whereas, the total execution time is similar due to synchronous barriers. Therefore, the fast tasks are waiting for the slowest task to finished its work as in figure (2). Thus, the execution time of the MPI program is the execution time of the slower task which is computed as in the equation (1).

$$T_{parallel} = \max_{\substack{i=1,2,\dots,N \\ j=1,2,\dots,M}} (T_{ij}) \qquad\qquad (1)$$

Where N is the number of clusters in the grid and M is the number of nodes in each cluster.
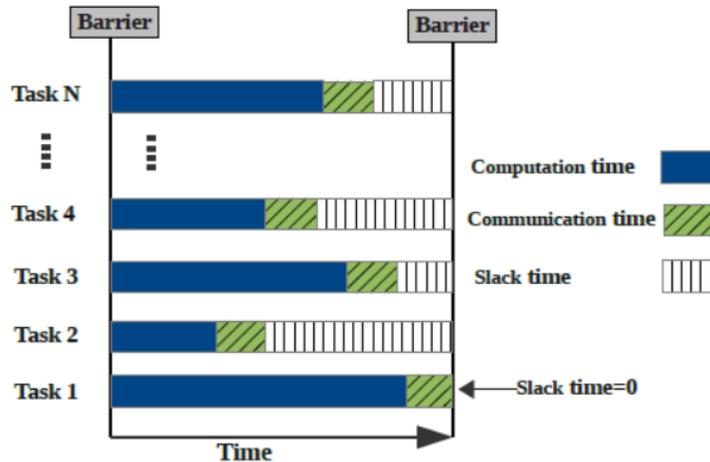
Figure (2): Parallel tasks execution over grid

3-The proposed prediction model for grid structure

The main goal of this section is to predict the execution time of any set of nodes in a grid when executing a parallel application over them. Mainly, the most important challenge in the predication process is when the underline environment is dynamic. Therefore, the interested environment that considered in this work is the different configurations of grid architectures. Grid infrastructure can be composed from N sites each one with M nodes as explained in section 2. However, grid's nodes are heterogenous in their computing power. Whereas, the difference in the number of nodes and/or nodes' types of the grid effects significantly on the running time of the executed parallel message passing program.

To build prediction model for the running time of MPI applications over any grid configuration, some information must be gathered firstly. This information are the computation times *TCP* and the communication times *TCM* of an application. To gather these times correctly, the required parallel application must be running over a set of nodes that represent one nodes one per cluster. For example, if the grid composed from three cluster, then the application is running over three nodes each from one cluster. However, the number of nodes for all types equal to number of clusters which are denoted as *NCT*. When all the computation times are gathered from all nodes, the sequential computation time can be computed. The sequential execution time is computed by multiplying the computation time of the slower task by a number of nodes types *NCT* as follows:

$$Tcp\_seq = T_{cpslower} \cdot NCT \qquad (2)$$

Where $T_{cpslower}$ is the time of the computation gathered from slower task. .

In case of the application is supposed to be executed on a set of homogenous nodes in a grid of the number *N.M* nodes. Therefore, the execution time of the homogenous grid is calculated proportionally as follows:

$$T_{cp_{homo}} = \frac{Tcp\_seq}{N \cdot M} = \frac{T_{cpslower} \cdot NCT}{N \cdot M} \tag{3}$$

Accordingly, the ratio of change between the time of computations of parallel application executed over heterogenous grid composed from N cluster, each with M heterogenous nodes denoted as heterogenous scaling factor *HCF*. This factor is used to represent the change in these times as follows:

$$HCF_{ij} = \frac{T_{cpslower}}{T_{cp_i}} = \frac{\max\limits_{\substack{i=1,2,...,N \\ j=1,2,...,M}} (T_{cp_{ij}})}{T_{cp_{ij}}} \tag{4}$$

Where HCF_{ij} is the factor of heterogeneous computation of node j in cluster i.

These factors are used to predict the heterogenous computation times of a grid. Each node's type has its factor that represent its different with the homogenous ones. Therefore, the predicted computation times of each node in grid is computed by multiplying the homogenous computation time by the heterogeneous computing factor *HCF* as follows:

$$T_{cp_{predicted_{ij}}} = T_{cp_{homo}} \cdot HCF_{(t)ij} = \frac{T_{cpslower} \cdot NCT}{N \cdot M} \cdot HCF_{(t)ij} \tag{5}$$

Where i=1,2,...,N and j=1,2,...,M.

Therefore, relatively the predicted computation times are decreased or increased by a factor of *HCF*. As shown previously, the execution time of the MPI program executing over a heterogenous grid is the execution time of the slower task. Generally, the program consists of two sections: computation and communication times. However, the predicted execution time is the predicted computation time added to the predicted communication time. Thus, the predicted computation time of the slower task in grid is calculated as follows:

$$T_{cps_{predicted}} = \max\limits_{\substack{i=1,2,...,N \\ j=1,2,...,M}} T_{cp_{predicted_{ij}}} \tag{6}$$

The relation between the communication times and the number of nodes in a grid is propositional. The predicted communication time is computed relative with the gathered communication time of the running the application over NCT nodes in a grid with the new N.M grid's nodes. Therefore, the predicted communication time of an application is the communication time of the slower task as follows:

$$T_{cm_{predicted}} = \frac{T_{cm} \cdot N \cdot M}{NCT} \tag{7}$$

Therefore, the overall execution time of the parallel message passing application running over any grid architecture can be predicted by computing the total of the equations (7) and (8) as follows:

$$T_{predicted} = \max\limits_{\substack{i=1,2,...,N \\ j=1,2,...,M}} T_{cp_{predicted_{ij}}} + \frac{T_{cm} \cdot N \cdot M}{NCT} \tag{8}$$

The proposed model in the equation (8) is used to predict the running time of any MPI application over any grid platform that described in section (2).

4-The experimental results

In this section, both the experimental configuration and results show and explained for validating the proposed prediction model as in the next subsections:

4-1 Experiential setting

The major goal of this section is to present the tools, parameters and software used to test the proposed method of predicting the running time of parallel MPI applications. Heterogenous grid platform has been used in various configurations to test the ability of proposed model. To perform this setting easily, SimGrid simulators [17] was used as simulation tools to build different grid structures for each instance. Moreover, three parallel benchmark applications have used as a parallel application to evalute the ability of the new prediction model. The CG, MG, LU of NAS parallel benchmarks of NASA [18] were used. These applications are selected where they have different computation to communication ratios. While, the proposed model is used to predict the execution time of MPI application when running over any grid platform. Then multiple scenarios have been developed to validate the model accuracy. Table (1) shows six different grid scenarios.

Table (1): Grid configuration scenarios

Scenario name	Number of clusters	Total Number of nodes	Gflops of each node in cluster			
			Cluster 1	Cluster 2	Cluster 3	Cluster 4
Grid 4*2	4	8	40	50	60	70
Grid 4*4	4	16				
Grid 4*8	4	32				
Grid 3*9	3	27	35	45	55	-
Grid 3*12	3	36				
Grid 3*16	3	48				

According to the above table, each cluster has a computing power measured in Gflops which are different from the other clusters.

4-2 Experiment evaluations

To evaluate the results of the proposed execution time prediction model equation (8), three parallel MPI benchmarks program of NAS were used [18]. These programs solve different problems of size class B, where all benchmarks have different classes. Each benchmark is executed over all six grid scenarios that explained in table (1). The predicted execution time of the proposed model equation (8) is measured and compared with real execution time for each scenario of all benchmark. Figures 3, 4 and 5 show the comparison of results of the predicted execution time with real one in seconds. Each MPI program gives different results due to the difference in the granularity ratio that it has, where granularity is the ratio between the computation to the communication times. Moreover, each grid scenario has a different number of nodes which my difference in their computing power with other grid

scenarios. Therefore, these different grid scenarios with the MPI benchmarks executed over them give a good test cases to verify the proposed prediction model. The average percentage error between the real and the predicted execution time were measured for all three benchmarks. The average percentage error for CG, MG and LU benchmarks are 4.36%, 5.79% and 6.81% respectively.
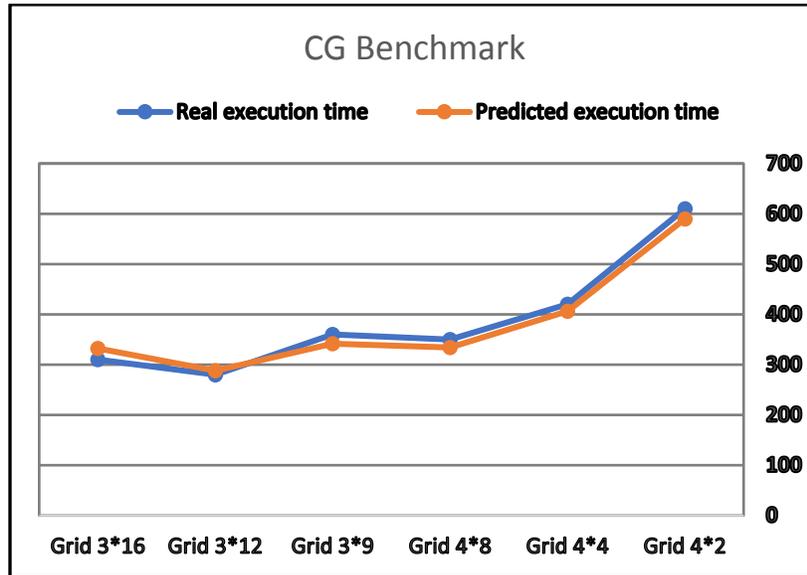

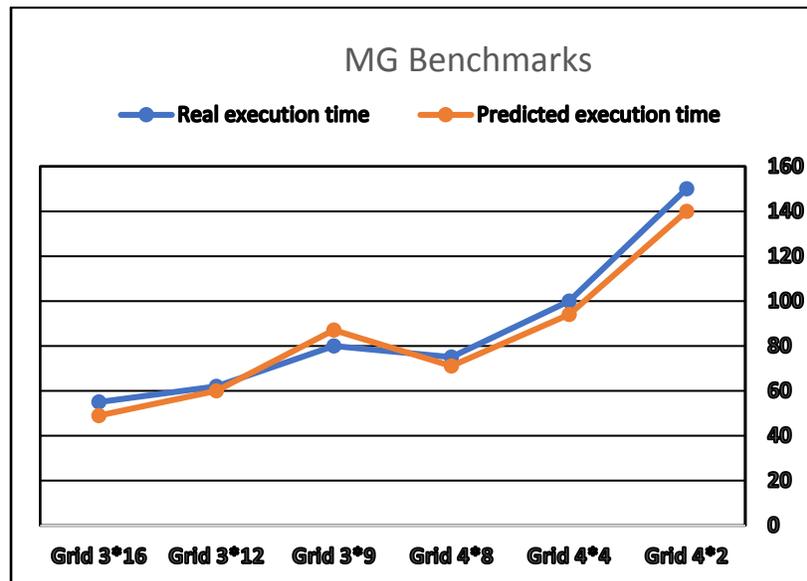**Figure (3): The predicted and real execution times of CG benchmarks**

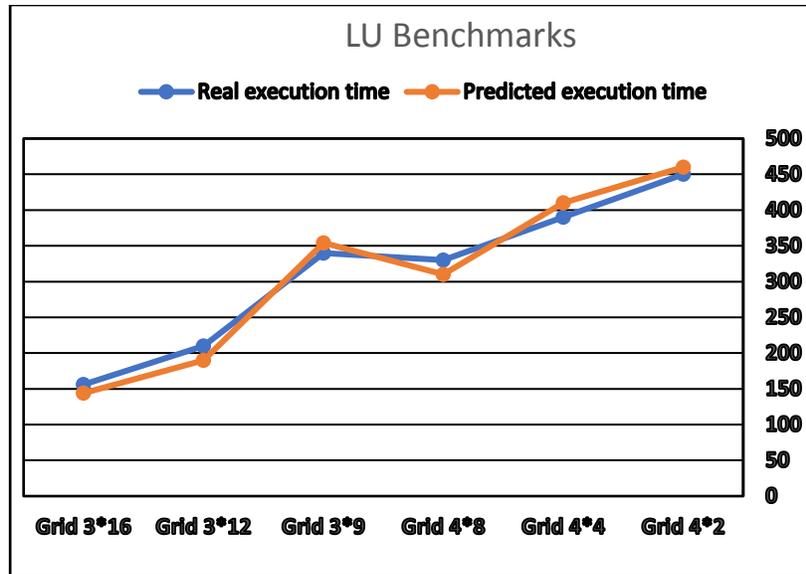**Figure (4): The predicted and real execution times of M G benchmarks**

Figure (5): The predicted and real execution times of LU benchmarks

5-Conclusion and future works

One of the most complex problems in the scientific area is the prediction process when the working environment is changeable. Therefore, this paper deals with a method that predicting the running time of MPI applications running over different heterogenous grid configurations. Three NAS parallel benchmarks have been executed over different grid scenarios to test the proposed execution time prediction model. SimGrid simulator used to simulate these grid scenarios. The results of the proposed new model give the average percentage errors 4.36%, 5.79% and 6.81% for CG, MG and LU benchmarks respectively.   These results represent various grid scenarios in term of different number of nodes and their types. According to the results obtained from the proposed model, in future, it is interested to use this model as guided tool in one of the optimization algorithms to enhance the design of grid structure. Moreover, the consumed energy of a grid can be easily predicted depending on proposed model.

Conflict of Interests.
There are non-conflicts of interest.

References

1. V. Rajaraman and R. A. M. M. C. SIVA, Parallel Computers Architecture and Programming. PHI Learning Pvt. Ltd., 2016.
2. C.-Y. Chou, H.-Y. Chang, S.-T. Wang, K.-C. Huang, and C.-Y. Shen, "An improved model for predicting HPL performance," in International Conference on Grid and Pervasive Computing, 2007, pp. 158–168.

3. Xu, Z., & Hwang, K.: Modeling communication overhead: MPI and MPL performance on the IBM SP2. IEEE Parallel & Distributed Technology, Systems & Applications, Vol. 4, No. 1, pp. 9-24, 1996.

4. B. Subramaniam and W. Feng, "Statistical power and performance modeling for optimizing the energy efficiency of scientific computing," in Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing, 2010, pp. 139–146.

5. B. C. Lee, D. M. Brooks, B. R. de Supinski, M. Schulz, K. Singh, and S. A. McKee, "Methods of inference and learning for performance modeling of parallel applications," in Proceedings of the 12th ACM SIGPLAN symposium on Principles and practice of parallel programming, 2007, pp. 249–258.

6. K. Singh, E. İpek, S. A. McKee, B. R. de Supinski, M. Schulz, and R. Caruana, "Predicting parallel application performance via machine learning approaches," Concurr. Comput. Pract. Exp., vol. 19, no. 17, pp. 2219–2235, 2007.

7. B. J. Barnes, B. Rountree, D. K. Lowenthal, J. Reeves, B. De Supinski, and M. Schulz, "A regression-based approach to scalability prediction," in Proceedings of the 22nd annual international conference on Supercomputing, 2008, pp. 368–377.

8. J. L. Hennessy and D. A. Patterson, Computer architecture: a quantitative approach. Elsevier, 2011.

9. B. Miegemolle and T. Monteil, "Hybrid Method to Predict Execution Time of Parallel Applications.," in CSC, 2008, pp. 224–230.

10. A. Jayakumar, P. Murali, and S. Vadhiyar, "Matching application signatures for performance predictions using a single execution," in 2015 IEEE International Parallel and Distributed Processing Symposium, 2015, pp. 1161–1170.

11. J. C. Charr, R. Couturier, A. Fanfakh, and A. Giersch, "Dynamic frequency scaling for energy consumption reduction in synchronous distributed applications," in 2014 IEEE International Symposium on Parallel and Distributed Processing with Applications, 2014, pp. 225–230.

12. J.-C. Charr, R. Couturier, A. Fanfakh, and A. Giersch, "Energy consumption reduction with DVFS for message passing iterative applications on heterogeneous architectures," in 2015 IEEE International Parallel and Distributed Processing Symposium Workshop, 2015, pp. 922–931.

13. A. Fanfakh, J.-C. Charr, R. Couturier, and A. Giersch, "Optimizing the energy consumption of message passing applications with iterations executed over grids," J. Comput. Sci., vol. 17, pp. 562–575, 2016.

14. A. B. M. Fanfakhri, A. Y. Yousif, and E. Alwan, "Multi-objective Optimization of Grid Computing for Performance, Energy and Cost," Kurdistan J. Appl. Res., vol. 2, no. 3, pp. 74–79, 2017.

15. A. Fanfakh, J.-C. Charr, R. Couturier, and A. Giersch, "CPUs Energy Consumption Reduction for Asynchronous Parallel Methods Running over Grids," in 2016 IEEE Intl Conference on Computational Science and Engineering (CSE) and IEEE Intl

Conference on Embedded and Ubiquitous Computing (EUC) and 15th Intl Symposium on Distributed Computing and Applications for Business Engineering (DCABES), 2016, pp. 205–212.

16. S. K. Idrees and A. B. M. Fanfakh, "Performance and Energy Consumption Prediction of Randomly Selected Nodes in Heterogeneous Cluster," in International Conference on New Trends in Information and Communications Technology Applications, 2018, pp. 21–34.

17. H. Casanova, A. Legrand, and M. Quinson, "Simgrid: A generic framework for large-scale distributed experiments," in Tenth International Conference on Computer Modeling and Simulation (uksim 2008), 2008, pp. 126–131.

18. N. A. S. P. Benchmarks and M. Versions, "NASA Advanced Supercomputing Division," *NASA Ames Res. Center, CA, USA*, 2003.

الخلاصة

في الوقت الحاضر خوارزميات التحسين عالية السرعة تكون مطلوبة. في معظم الحالات، يحتاج الباحثــون إلى طريقة للتنبؤ ببعض المعايير بدقة مقبولة لاستخدامها في خوارزمياتهم. ومع ذلك، في مجال الحوسبة المتوازية يمكن اعتبار وقت التنفيذ من أهم المعايير. لذا، يعرض هذا البحث نموذجًا جديدًا للتنبؤ بالوقت للتنفيذ لتطبيقات المتوازيــة الموزعة المنفذة على العديد من سيناريوهات الشبكة. حيث يمتلك النموذج المقترح القدرة علــى التنبؤ بوقت تنفيذ التطبيقات المتوازية التي تعمل عبر أي تكوين للشبكة من حيث عدد العقد المختلفة وقوى الحوسبة الخاصة بها.

لقد تم تنفيذ التجارب على المحاكي سمكرد الذي يمتلك خاصية السهولة في بناء نماذج شبكية متعدد ومختلفة. نتائج الاختبارات بين اوقات التنفيذ الاصلية والاوقات المتنبئة بينت دقة تجيريبية جيده. معدل الخطأ النسبي بين وقت التنفيذ الاصلي والمتنبأ لثلاث برامج معيارية تكون هي 4.36٪، 5.79٪ و 6.81٪.