Asian Journal of Computer Science and Technology
ISSN: 2249-0701 (P) Vol.4 No.1, 2015, pp.8-13

DOI: https://doi.org/10.51983/ajcst-2015.4.1.1749

Review of the Parallel Programming and its Challenges on the Multicore Processors

Ramesh Singh Yadava
Computer Centre, Banaras Hindu University,Varanasi, India
E-mail: rsy@bhu.ac.in

(Received 5 January 2015; Revised 20 January 2015; Accepted 15 February 2015; Available online 23 February 2015)

Abstract **- In this paper, we have reviewed and discussed about the challenges, advantages, disadvantages of the parallel program, which occurs during the conversion of the sequential program into parallel program.The conversion of the existing sequential programs and algorithm into parallel have limitations like partitioning the task, sharing input and output data, dependencies of the output of one subtask to next subtask, synchronization of the output of the subprogram, and shifting the subprogram from the failure processor to the active processor, communication among the processors, load balancing, etc. The parallel programming routines, libraries and management software for parallel programming have been implemented, and these have several limitations, and these are also discussed.**

Keywords: **Parallization, Sequential, High Performance Computing(HPC), Parallel Programming**

I.INTRODUCTION

The parallelization of the program or software for the multicore processors is a very difficult task for programmers and algorithm developers. Some of the task is not possible to convert into parallel completely. This is not possible due to the lack of non-availability of the algorithm or subject expertise. Otherwise, the output of the program may be changed and incorrect. The parallelization of program may increase the space complexity, time complexity. But it may improve the throughput time of the program. So, the output of the parallel program can be obtained in a less time in comparison of the sequential program.

The most of the algorithms have been implemented with sequential programming, since in the past the parallel programming concept was not available to the developers. For converting the sequential program into parallel program, the algorithms have to be converted into parallel and implement it with parallel programming. For, converting the available algorithms into parallel, the subject expert is required to verify that the newly developed algorithm is working fine in the parallel programming. The researchers of the particular domain have to develop parallel algorithm for parallel programming, otherwise the output of the parallel algorithm may be change, or the output of the parallel program may not be correct.

The parallel programming can be performed with parallel compiler. The parallel compiler can compile the program efficiently, and execute it in a parallel mode. The parallel program can be implemented using loop parallelization. The compiler can be implemented using the shared memory compute cluster with limited features. The parallel compilers compile the program automatically in parallel. This compiler will not work with the distributed memory compute clusters [1]. The researchers have to work in this for improving the parallel compilation of the program. These compilers increase the performance of the program, but the problem occurs, when the threads of the program are running concurrently and accessing the same memory location at a time. The race condition may occur. For solving this problem, the thread access the data sequentially as per the loop used in the program[2].In this implementation the sequential processor work like a multiprocessor, but the performance of the program is not improved much as per our expectation.

The parallel programming libraries have been developed for implementing the parallel computing with the programming languages C/FORTRAN/C++. The implementation of the parallel programming with these high level languages are easy for the developers, since the developers are aware with these languages. The others parallel programming languages have also been developed like SISAL [3] and PCN [4]. The parallel programming languages routines, libraries and frame architecture have been developed with The MPI, PVM, LINDA, and P4System.

A. Challenges of Parallel Programming

The parallel programming has a several challenges, and these are discussed here in brief:-

The most of the programs have been developed sequential. The sequential programs are very lengthy. So, these programs have been developed with the use of several developers. These professional are well trained and experienced, and they are able to understand the problem for which they are developing the program. These are also available easily, however the parallel program developer are not available frequently. The parallel program developers are self learned. These are not the well trained professional developers. This is a large gap between parallel program and sequential program developers [5,6]. So, it is very difficult to convert the large sequential program into parallel program. So, that the professional training is required to the

parallel program developers, and this will make them expert, and they can able to write parallel program. This is one of the important problems with the parallel programming.

The second problem is transportability of the parallel program from one computing environment to other; as such the parallel program written for six core processor will not perform well on different number of core processor. So, the performance of the program may or may not be improved as per our computation speed on the execution of more core processors. The other problem may be compatibility of the program with other computing hardware and software.

The third problem is lack of tools and technique for debugging, tracing and testing the parallel program. The efficient testing and debugging tools are not available. These tools have to be developed for testing, debugging and tracing the parallel programs [7]. For this, a lot of the research work is required.

The fourth major problem is data dependence in the subprogram of the parallel programs. The output of a subprogram may be input for other program, and then this subprogram has to wait till the first is not completed. This will increase the throughput time of the program. The other problem in data dependencies may be parallel accessing of the data from a single sequential storage. So, the vector storage or several parallel storages are required to store and to retrieve the data parallel in the parallel programming.

The parallel programming for the multicore processors is a tough task, when all the cores have a shared cache memory. Otherwise, when each core of the processor has its own cache memory, then the computing speed of the parallel program will be faster, and it reduces the accessing time of the data, and the parallel programming work efficiently.

The algorithms for the mathematical and scientific computation, etc have been developed for the sequential programming. The algorithm developers have to develop the parallel algorithm for parallel programming, this is a very difficult task, since the parallel algorithm may change the execution steps than the sequential algorithm, and it may increase it complexities. So, the algorithm developers have to ignore the complexities of the algorithm on the cost of the faster throughput [8]. The parallel computing is difficult to implement for the mathematical computation, when one module depend upon other module. So, this implementation can be performed with using the asynchronous communication in the subprograms, and it may increase the computation time.

B.Parallel Programming Languages

There are several programming languages, libraries, routines and frame architectures. These have been developed and implemented for the parallel programming, which I have discussed here in brief.

The parallel programming languages libraries (MPI/PVM) have been developed. These libraries are used for the implementation of the parallel programming. In the, parallel program the communication, sharing the result, etc are required in the subprogram. The parallel programming (MPI/PVM) libraries provide these functions and routines for communication, sharing the result, etc as and when requires [9]. These libraries are implemented with using the programming language C/C++/FORTRAN.

The ParaSail (Parallel specification and implementation language) was developed for parallel programming in the multicore processors. Since, the multicore processors are the processor integrated in a single chip. So, the memory and communication channel can be same or dedicated to a particular processor. This language works well in the multicore processor for parallel programming[10].

The CUDA(compute unified device architecture) is a parallel programming platform and design for the parallel programming for the NVIDIA graphic processor. The CUDA provides a compute environment for developing the parallel program with C/OpenACC/C++/JAVA/Python, Direct Compute/MPI, etc. In this architecture the CPU work like a controller, and it allocates the job to the GPU(graphical processing units) for parallel processing. The computation performance of the GPU is very fast than CPU, and it improves the throughput time of the program [11].

The multithread programming can be used for implementing the parallel programming. The thread is subprogram, and it can be implemented using programming language JAVA/PHP/C++, etc [12, 13]. The multithread program executes parallel in the multicore processor cores.

The other parallel programming languages and framework like P4 system and Linda have also been developed, but these programming languages and framework do not get much popularity. So, these languages became absolute with time [12,13].

C. Performance of Parallel Computation on Multicore Processors

The sequential program is divided into subprograms for parallel programming. The parallel program can be executed on the multicore processor/grid computing/compute cluster/cloud computing server. Any one of these compute resources may be developed with using the multicore processors servers. The advantages of the multicore processor servers are that it consume less energy and provides more computing power than the single core processor servers. So, it is used in the computational architectures, where several compute nodes and large memory are needed for the computation.

The computing performance of the parallel program may be improved theoretically as per the Amdahl and Gustafson rules in the multicore processor. These rules are described in brief for showing the performance of computation in the parallel.

According to the Amdahl[14], that a task cannot be completely possible to convert into parallel program. So, a rest of part of the program may be remained in the sequential. For this, Amdahl has assumed that the proportion 'P' of the task has been converted into parallel program, and the remaining part of the task (1-P) is in sequential, this program is executed on 'N' processors parallel and S_N is speedup of the converted program. So, the speedup of the parallel program will be:

$$S_N = \frac{1}{(1-P) + \frac{P}{N}}$$

Similarly Gustafson [15] has assumed that complete sequential program can be converted into parallel program, and he has assumed that a program is divided into 'P' subprogram, and these are executed on the 'N' processors and 'S' is speedup of the program. So, the parallel program speedup will be:

$$S = P + N(1-P)$$

In these equations, it has been assumed, that sequential program computing time is independent of the number of the processors.

The Amdahl has assumed that the complete program may not possible to convert into parallel. So, the throughput time of the program will not be linear, however Gustafson has assumed, that the complete process can be converted into parallel, so the compute time will vary linearly with the number of processors.

According to Amdahl and Gustafson rules the parallel program execution performance is not better than sequential program in the compute cluster or grid computing or in the cloud computing. But these computing resources have a large memory and computing power. So, the large data set can be executed on these.

The HPC(high performance computing) can be developed using single core processors or multicore processors. The multicore processors, HPC takes less energy and provides more computing power than single core processor. The multicore processors execute the program efficiently, since all the processor cores have cache memory and integrated on a single chip. So, the communication delay among the core of the processor is very small, and it can be ignored [16].

The parallel programming faces communication bottleneck and race condition. The communication overhead may occur for huge data and sharing the result of one processor to other processor. This can be reduced with redundant data storage and with high speed network switches and communication channel [17,18].

D. Types of Parallel Programming

The parallel programming can be implemented in two ways.

In the first way, it can be implemented with single program multiple dataset (SPMD), and next way, it can be implemented using multiple program multiple dataset (MPMD). In the SPMD programming, the same program code is executed on the different compute processor core with different dataset, and the result may be communicated in the compute processor cores, if it requires. This programming concept shares the memory, communication channel and storage. So, the communication overhead may be more and communication time may be increased. This programming is used in the tightly coupled compute cluster or multicore processor.

In the MPMD programming, the different programming codes are executed with different dataset on the different compute node or cores of the processor. This programming can be used in the compute cluster or cloud computing. This programming is used, where the data storage is parallel and able to provide data concurrently. This programming may be used in loosely coupled compute cluster as well as in tightly coupled compute cluster [17,19].

II. RELATED WORK

The several software have been developed for managing the parallel programming like program schedulers, parallel compiler and compute cluster managers. These may be installed and configured on the master node for managing the task parallel, load balancing and other work. The schedulers for the parallel programs schedule the subprogram to the particular processor shift the program to other processor, when the processor fails down or in ideal. The scheduler software may be PBS Pro/Maui with Moab/ Torque [20, 21], etc, in which the PBS pro and Torque are professional developed and supported software, however the Maui and Moab are open source software, so they have not a proper support.

The parallel programming can be used in the cloud computing. In the cloud the multiple dedicated mirrors of the data can be stored at different storage devices, and these can be accessed parallel for parallel programming. So, the computing performance can be improved in the cloud computing for the larger data [22].

The parallel programming in the grid computing is very difficult to implement, since the grid is situated at different geographical location and these are connected with high speed network. After all, the communication time between

the processors may be increased, and it cannot be negligible [23]. The programmer has to analyze the program before implementing in the grid compute cluster, otherwise the throughput time may be increased.

The Haskell has proposed and implemented a parallel programming. In this proposal, the Haskell has assumed a semi parallel program, in which the threads of the program executed parallel. The program threads cannot be parallel, since the time schedule for sharing the data cannot be possible to make accurate. Since in the Haskell program, the child thread is not controlled with its the parent thread for executing the thread parallel[24]. The program implemented with using the Haskell is not a complete parallel program, and it is difficult for the programmers to use it for the parallel programming.

A compiler was developed for compiling the program in parallel. This compiler creates the concurrent parallel program with some conditions [25]. The compiler compiles the loops and conditional statement in the parallel for executing the program parallel. But this implementation is very difficult for analyzing and tracing the flow of the program.

A YuruBackup was implemented using incremental backup in the cloud for decreasing the data backup size in the storage. This backup was proposed and implemented for storing the fingerprint of the users in parallel at the different storing places with the help of the master node. These parallel stored data can be retrieved parallel at the slave nodes in the read mode for parallel accessing [26]. This backup technique makes efficient to access the data from the storage parallel and save the storage space. But this backup technique has disadvantage, that a particular data location cannot able to locate in the storage and it takes long time to search the particular data in the storage. So, that an efficient parallel programming is required for searching the data in this.

The 'R' parallel programming languages for the statistical computing has been developed in the year 2008. This language can be implemented in the parallel computing environment like compute cluster/ the multicore processor system/grid computing. Using this language, the statistical packages have been developed. This language can also be used with MPI/C/C++/FORTRAN. The important software packages like Snow and Rmpi have been developed using this language. These packages are used in the compute cluster. A package has also been developed for the grid computing, and it is available for the use, and other packages are under development. This programming language will be one of the important parallel programming language in the future [20].

III. DISCUSSION

There are several challenges in the parallel programming, in which some of the important challenges are disused here [27,28,29].

a) One of the important challenges of the parallel programming is communication in the subprogram. The communication in the subprogram may be different type. One of the important communications occurs, when the input of a subprogram is an output of other subprogram. For this purpose the synchronization is needed in the subprograms. This kind of implementation is very difficult for the developers. This is managed with using the messages passing and program scheduling. This challenge becomes more difficult, when the parallel subprogram executes on the muticore processor, where the same memory and communication channel are used with all the core of the processor for sharing and storing the data.

b) The sequential program is divided into subprogram. The subprogram of the program is called granularity of the programs. These subprogram are executed on the core of processor parallel. These subprogram are managed with a program scheduler in such a way, that all the core of the processor can be utilize maximum, for minimize the computing time, and this may improve the computing performance of the parallel program. It become difficult, when the subprogram is executed faster and other subprogram executed slower, then synchronization problem occur in the output of the program. For this program manager is needed for managing the subprogram efficiently. SO, the research work is required for developing this kind of software.

c) Sharing the data from the storage, in the parallel programming is very difficult, since the most of the storage devices are sequential. For avoiding the race condition the data should be stored in a mirror/ redundant ways, from where the data can be accessed parallel. A research work is needed for developing such a parallel program, which it may work properly in the parallel computing, and it can also support to access the data from different storage or from same storage parallel and it will decrease the communication time.

d) The communication bottleneck occur in the parallel programming, when it is implemented in the multicore processor, since the same communication channel is used for all the processor to access the data and result. This can increase the computation time and degrade the performance of the multicore processor. For reducing the communication time in the multicore

processor the cache memory is included with each core in the modern processors, and it decreases the communication time.

e) The implementation of the synchronization in the parallel subprogram is very difficult for the developers. The developers don't know the computation time each subprogram, which is required for matching the subprogram sequence. So, it is very difficult task to make the subprogram synchronize in the parallel programming. For this a lot of research work is required during the development of parallel algorithm.

f) There are no any tools available for debugging the parallel program. For it, the researchers have to work with developers for developing the software tools. This software tools can debug and trace the bug in the program and make it easy for the developers. It is a difficult task for the system programmer to develop a parallel program debugging and tracing tools.

g) The error may occur in the parallel program during the compilation/execution of the program. The correction of the parallel program is very difficult, since the mostly used parallel programming languages (MPI/PVM) only provide libraries functions, and it is very difficult to correct the error in these libraries function. So, a lot of research work is required in this area for developing a tool, which can trace the error and recover it.

h) The parallel programs are collection of subprogram and these subprograms include parallel program libraries and routines. The intelligent parallel program editors are required, which it can automatically include the parallel program libraries and routines.

i) The parallel programs are platform dependent. So, the developers have to develop a parallel program in such a way, that it may transport on any computing environment without changing the code of the program.

These are the main challenges, in which the research and development is required. This will be performed with collaboration with researchers and developers.

IV. FUTURE WORK

The conversion of the sequential program into parallel program is very difficult for the developers. Since the most of the algorithms are sequential. For converting the sequential algorithm into parallel, the subject expert is required. The subject expert helps to develop the parallel algorithm. This algorithm is implemented in the parallel programming, in which, some of the problem can be converted and some cannot possible to convert. It depends on the nature and limitation of the problem. The mathematical problem, where one function depends on other, then it is difficult to convert, but when the modules are independents, it can be easily converted into parallel programming.

A lot of research work is required in the parallel programming implementation from the sequential programming. The researchers and developers have to work with each other in team for developing the parallel programming for utilizing the computing power of the compute processor cores and sever for better performance in computing time.

The parallel programmers have to be trained properly, so that they can able to develop the professional developed software of the applications, and the developed application may work better than the existing sequential program.

Future of the parallel programming is very bright, so, the researcher, algorithm developer and program developer have to work together for developing the parallel program. The parallel programming will be advantage for the researcher for computing the large and complex problem in a very small time, wherever the sequential program takes long time to compute the same problem and researchers have to wait. Sometime the result of problem may get after the incident such like forecasting of the weather, natural disaster, etc.

V. CONCLUSIONS

The major challenges and research work required in the parallel programming have been discussed in this paper. These challenges have several limitations in the available software and hardware technologies. For minimizing the challenges, the researchers and developers have to work together for developing the research software for the researchers. So, the researcher can get the research result efficiently with parallel programming.

With the technologies and software, the algorithm developers have to implement the parallel algorithms. So, that developer can easily able to implement it. But some problem cannot be possible to convert into parallel, these problem have to be convert in such a way that the computation time of these problem can also be minimized. For this purpose the expert of that area is required for developing the parallel algorithm and the researchers have to motivate for developing the parallel algorithm for the exiting problems and new problems.

REFERENCES

[1] L. Moura E Silvay and R. Buyya, Parallel Programming Models and Paradigms, http://www.buyya.com/cluster/v2chap1.pdf, Vol. 1, Chapter 1, pp 1-27,1999.

[2] Choi, Jong-Deok, Manish Gupta, Mauricio J. Serrano, Vugranam C. Sreedhar, and Samuel P. Midkiff, "Stack allocation and

synchronization optimizations for Java using escape analysis," ACM Transactions on Programming Languages and Systems (TOPLAS) 25, no. 6 (2003), pp. 876-910.

[3] J. Feo, D. Cann, and R. Oldehoeft, "A Report on the SISAL Language Project," Journal of Parallel and Distributed Computing, vol 10, pages 349-366, 1990.

[4] I. Foster and S. Tuecke. Parallel Programming with PCN. Technical Report ANL-91/32, Argonne National Laboratory, Argonne, December 1991.

[5] B.L.Massingill, T.G. Mattson, and B.A. Sanders, "Reengineering for Parallelism: an entry point into PLPP for legacy applications," Concurrency and Computation: Practice and Experience, Vol. 19, p. 503-529, 2007.

[6] G.Vélez, and Horacio, "A New Industry-Centred Module on Structured Parallel Programming," School of Computing , Robert Gordon University, Postgraduate Certificate in Higher Education Learning and Teaching , 2010.

[7] Meade, Anne, Jim Buckley, and J. J. Collins, "Challenges of evolving sequential to parallel code: an exploratory review," In Proceedings of the 12th International Workshop on Principles of Software Evolution and the 7th annual ERCIM Workshop on Software Evolution, pp. 1-5. ACM, 2011.

[8] W.-M. Hwu, "Three challenges in parallel programming," http://parallel.illinois.edu/blog/three-challenges-parallel-programming, March, 2012.

[9] MPICH2, http://www.mcs.anl.gov/research/projects/mpich2/, May 5, 2011.

[10] S. Tucker Taft, and SofCheck, Introduction to ParaSail, Sofcheck Software analysis and verification, O'relly OSCON oscon.com, 2011.

[11] CUDA C Programming Guide, PG-02829-001_v5.0, nvidia.com, May, 2014.

[12] M.Snir, S.Otto, S.Huss-Lederman, D. Walker , and J. Dongarra, MPI: The Complete Reference, The MIT Press, Cambridge, Massachusetts London, England, 2002.

[13] A.Geist, A. Beguelin, J. Dongarra, W.Jiang, R.Manchek and V.Sunderam, PVM: Parallel Virtual Machine A Users' Guide and Tutorial for Networked Parallel Computing, The MIT Press, 1994.

[14] G. Amdahl, "Princeton University - Amdahl's Law," https://www.princeton.edu/~achaney/tmve/wiki100k/docs/Amdahl_s_law.html,May, 2014

[15] J. L. Gustafson, http://en.wikipedia.org/wiki/Gustafson s_law, May,2014

[16] Li, Jianhua, Weibin Guo, and Hong Zheng. "An Undergraduate Parallel and Distributed Computing Course in Multi-Core Era," 9th International Conference for Young Computer Scientists, ICYCS 2008, Zhang Jia Jie, Hunan, China, , IEEE, pp.18-21, November, 2008.

[17] L. Ridgway Scott, T. Clark and B. Bagheri, "Lecture Notes in Computer Science," Springer, Volume 3515, pp 44-51, 2005.

[18] A.Geist, A. Beguelin, J.Dongarra, W. Jiang, R.Manchek and , V. Sunderam, PVM: Parallel Virtual Machine A Users Guide and Tutorial for Networked Parallel Computing, The MIT Press, 1994.

[19] S.Palaniappan, and C. Sook Ling, "Clinical Decision Support Using OLAP with data mining," International Journal of Computer Science and Network Security, Vol. 8, No. 9, pp.290-296, 2008.

[20] D. Jackson, Q. Snell, and M.Clement, Core Algorithms of the Maui Scheduler, D.G. Feitelson and L. Rudolph (Eds.): JSSPP, pp. 87–102, 2001.

[21] PBS, http://www.pbsworks.com/ , June 25, 2011

[22] Digipede, "Grid and Cluster Computing: Options for Improving Windows® Application Performance," ©Copyright Digipede Technologies, LLC,2011.

[23] D. Burford D, "Cloud Computing: A Brief Introduction," Land Enterprises Inc., http://www.ladenterprizes.com/pdf/CloudComputing.pdf, 2010.

[24] S.Peyton Jones and S. Singh, "A Tutorial on Parallel and Concurrent Programming in Haskell: Lecture Notes from Advanced Functional Programming," Summer School, Microsoft Research Cambridge, 2008

[25] Xu. Quanqing , Zhao. Liang, and Xiao. Mingzhong, "YuruBackup: A Space-Efficient and Highly Scalable Incremental Backup System in the Cloud," Int J Parallel Prog, Springer Science, October 2013.

[26] P. H. Madden, Parallel Computing: The Elephant in the Room, Computer Science Department , SUNY Binghamton, August 19, 2010

[27] http://www.intel.com/technology/itj/2007/v11i4/3-development/4-challenges.htm, December, 2013

[28] G. Barish, "Scalablead High-Performance Web Applications," May 24,2002.

[29] M. Wrinn, "Top 10 challenges in parallel computing," http://software.intel.com/en-us/blogs/2008/12/31/top-10-challenges-in-parallel-computing, May,2014.