International Journals of Advanced Research in Computer Science and Software Engineering
ISSN: 2277-128X (Volume-8, Issue-10)

Research Article **October 2018**

Many-Task Computing

Matthew N. O. Sadiku, Philip O. Adebo, and **Sarhan M. Musa**
Roy G. Perry College of Engineering, Prairie View A&M University, Prairie View, TX77446, United States
Email: sadiku@ieee.org; philip.adebo@gmail.com; smmusa @pvamu.edu

Abstract: Many-task Computing (MTC) provides an efficient way to execute thousands of independent tasks on a cluster. The goal in MTC is to execute all the tasks in the most efficient way across a given resources. MTC is a practical computing paradigm that is widely used for scientific applications such as image processing, Monte Carlo simulations, data parallelism, and informatics. This paper provides a brief introduction to many-task computing.

***Key Words:* many-task computing**

I. INTRODUCTION

Computing plays a crucial role in how we explore the world. Many-task computing (MTC) is an emerging programming model that aims to bridge the gap between two computing paradigms: high-throughput computing (HTC) and high-performance computing (HPC). Several MTC applications do not neatly fit the stereotypes of HPC or HTC applications. MTC is different from HTC in the emphasis of using large number of computing resources over short periods of time to accomplish many computational tasks.

MTC emphasizes running many computational tasks over a short period of time. MTC is the execution of multiple tasks on one particular parallel platform at the same time. It has been well supported on multiple parallel systems such as cluster, grids, cloud, and supercomputers [1].

MTC provides a general framework for executing a large number of independent processes on high throughput parallel machines. It addresses many of the HPC shortcomings at extreme scales such as reliability and programmability. The number of tasks, quantity of computing, and volumes of data may be very large. The rapid developments of large-scale processing techniques such as supercomputer, grid, and cloud have provided strong support for MTC to get satisfactory results within a short timeframe [2]. Emerging petascale computing systems incorporate high-speed and low-latency interconnects designed to support tightly coupled parallel computations. A computing problem is represented as a MTC job, which could contain millions of tasks. The tasks may be small or large, uniprocessor or multiprocessor, compute intensive or data-intensive, dependent or independent, static or dynamic, homogeneous or heterogeneous, loosely coupled or tightly coupled. Job is a large collection of relatively small tasks submitted by a user. A job is classified as a MTC job when it uses a large number of resources over a short period of time. It is classified a HTC job if it is executed over a period of months [3]. Different users have different deadline requirements; more important jobs need to be prioritized and completed before deadline. The problem space can be partitioned into four main categories as shown in Figure 1 [4].

II. APPLICATIONS

MTC applications are often data intensive, as each job requires at least one input file and one output file, and can sometimes involve many files per job. Some applications that are a better fit for MTC than HTC or HPC are presented here [4,5]:

- *Astronomy*: This application is both compute intensive and data intensive, and has been run as both a HTC and a HPC, but found its scalability to be limited when run under HTC or HPC. For example, the Sloan Digital Sky Survey (SDSS) has datasets that exceed 10 terabytes in size.
- *Economic Modeling*: A good MTC candidate is Macro Analysis of Refinery Systems (MARS), which studies economic model sensitivity to various parameters. This application is challenging as the parameter space is extremely large, which can produce millions of individual tasks.
- *Pharmaceutical Domain:* The economic and health benefits of speeding drug development by rapidly screening for promising compounds and eliminating costly dead-ends is significant in terms of both resources and human life. This application is challenging as there many tasks, each task has a wide range of execution times with little to no prior knowledge about its execution time.

- *Bioinformatics*: In bioinformatics, Basic Local Alignment Search Tool (BLAST) search enables one to compare a query sequence with a library or database of sequences, and identify library sequences that resemble the query sequence above a certain threshold. Although the BLAST codes have been implemented in both HTC and HPC, they are often both data and compute intensive. Another example of MTC application is Genome Alternative Splicing in bioinformatics.

Other applications include chemistry, biometrics, data mining, data analytics, astrophysics, medical imaging, climate modeling, economics, and neuroscience. These MTC applications often consist of a very large number of data-intensive tasks with relatively short execution times.

III. CHALLENGES

Although MTC is a new paradigm, there are still many important issues to find adequate strategies for different types of scheduling and execution. Scheduling and executing MTC activities on huge clusters may also suffer with churn events, poor load balancing, and usability issues [6]. When hundreds of thousands of tasks are continually submitted to the local resource managers, there is the possibility that some tasks are missed and not executed. Also, there is always a chance that some task will fail [7].

IV. CONCLUSION

Many-task computing (MTC) denotes high-performance computations comprising multiple distinct activities, coupled via file system operations. It is a widely used computing paradigm which is utilized in large distributed system. Python is an excellent way to implement and manage many-task computing. It is much more accessible and easier to learn than a traditional HPC language. Many-task computing is also being explored at the petascale level.

REFERENCES

[1] J. Johnson et al., “Understanding the costs of many-task computing workloads on Intel Xeon Phi coprocessors,” https://pdfs.semanticscholar.org/c20b/5f49106c3ab191557a840ab814b43ed36410.pdf

[2] S. Chen et al., “An application-level priority scheduling for many-task computing in multi-user heterogeneous environment,” Proceeding *of the International Conference on High Performance Computing & Simulation*, October 2013.

[3] M. Lunacek, J. Braden, and T. Hauser, “The scaling of many-task computing approaches in Python on cluster supercomputers,” *Proceedings of the IEEE International Conference on Cluster Computing*, Sept. 2013.

[4] I. Raicu, I. T. Foster, and Y. Zhao, “Many-task computing for grids and supercomputers,” *Workshop on Many-Task Computing on Grids and Supercomputers*, 2008.

[5] I. Raicu,”Many-task computing: Bridging the gap between high-throughput computing and high-performance computing,” Doctoral Dissertation, The University of Chicago, March 2009.

[6] J. Dias et al., “Improving many-task computing in scientific workflows using P2P techniques,” 3rd *Workshop on Many-Task Computing on Grids and Supercomputers*, New Orleans, LA, 2010.

[7] S. Yue et al., “Dynamic DAG scheduling for many-task computing of distributed eco-hydrological model,” *The Journal of Supercomputing*, 2017, pp 1–23.

AUTHORS

Matthew N.O. Sadiku is a professor in the Department of Electrical and Computer Engineering at Prairie View A&M University, Prairie View, Texas. He is the author of several books and papers. His areas of research interest include computational electromagnetics and computer networks. He is a fellow of IEEE.

Philip O. Adebo is an instructor at Texas Southern University. He is currently working towards a PhD in Electrical and Computer Engineering Department, Prairie View A&M University with emphasis on power systems. His research interests include power systems, renewable energy, microgrids, smart-grid systems, restructuring power system, and optimization of power systems.

Sarhan M. Musa is a professor in the Department of Engineering Technology at Prairie View A&M University, Texas. He has been the director of Prairie View Networking Academy, Texas, since 2004. He is an LTD Sprint and Boeing Welliver Fellow.

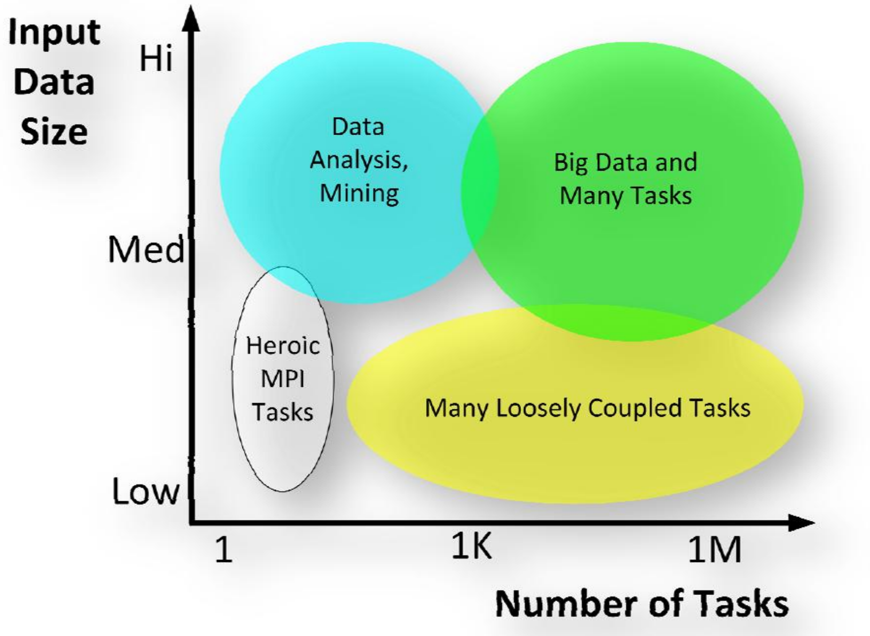

Figure 1 Problem types with respect to data size and number of tasks [34].