Efficient Execution of Smart City's Assets Through a Massive Parallel Computational Model

Muhammad Usman Ashraf(✉), Fathy Alboraei Eassa,
and Aiiad Ahmad Albeshri

Department of Computer Science, Faculty of Computing and IT,
King Abdulaziz University, Jeddah, Saudi Arabia
`m.usmanashraf@yahoo.com`, `fathy55@yahoo.com`,
`aaalbeshri@kau.edu.sa`

Abstract. Urban areas are at the forefront for upcoming wave of emerging technologies in building smart sustainable cities. These real time fashion based smart cities will be combined with physical infrastructures and services to improve the lives of its citizens. Transformation to smart city, innovation in performance increasing, data processing, planning and management will be required for smart city's assets. In order to achieve monolithic performance and data processing, an efficient parallel computational approach can bring us closer to the desired smartness in societies. In this study, we have proposed a massive parallel computational model to deal smart city's assets and maintain real time connectivity among them. This model will be considered as initiative for emerging exascale computing system which will be used to deal the applications in intelligent smart cities.

Keywords: Parallel computing · Hybrid programming · HPC
Smart cities

1 Introduction

Since last three decades, smart cities have been established and become operational in developed countries [1]. According to united nation report (UNR) [5], more than 50% people are living in urban areas and facing a wide range challenges to bring sustainability in their lives. The vision of smart cities is to improve the quality of lives in urban areas by using advance information communication technologies (ICT) [2], information processing technologies (IPT) [3] and internet of things (IoT) [4]. There are many definitions proposed for smart cities. According to Chourabi et al. [6] "*A smart sustainable city is an innovative city that uses information and communication technologies (ICTs) and other means to improve quality of life, efficiency of urban operation and services, and competitiveness, while ensuring that it meets the needs of present and future generations with respect to economic, social and environmental aspects*". According to the definition, these facts illustrates that smart cities are not only leveraging technologies but considered as a complex ecosystem where different

R. Mehmood et al. (Eds.): SCITA 2017, LNICST 224, pp. 44–51, 2018.
https://doi.org/10.1007/978-3-319-94180-6_6

stakeholders are tightly coupled each other. Figure 1 presents the key grouping technology aspects of a complex ecosystem for a smart society [7] where different players facilitate to others by providing solutions efficiently and maintain connectivity among them. The idea of smart cities is similar to the theory of ubiquitous computing which was proposed by 'Mark Weiser' co-founder of ubiquitous computing [24, 25].

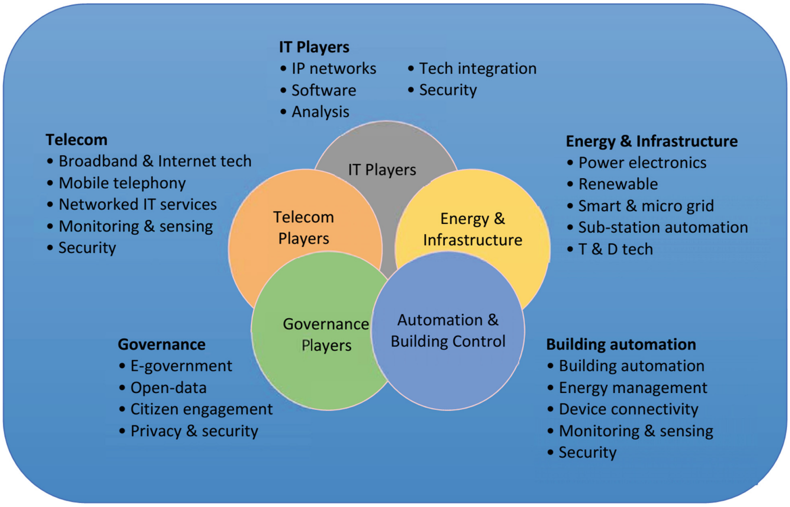

Fig. 1. Technology aspects of an ecosystem in smart city

This real-time data-driven based complex system requires a powerful computing system that can process data promptly and make the cities smarter.

In a sequel, advances in HPC play a vital role to bring smartness in urban areas [8]. Therefore, an effective usage of parallel computing can bring us closer to the desired smartness in intelligent cities [9]. For instance, usage of machine learning can discovered the solutions and make real time decisions. Similarly HPC parallel computing can be promising paradigm to deal smart city's assets.

According to HPC studies, trend is going to be changed and projected to introduce a new powerful computing system called "Exascale" which will be able to achieve ExaFlops number of calculation per second till 2020 [10, 20]. Exascale will be thousands fold increase in current Petascale system. Though Exascale is facing some tremendous challenges due to strict limitation in power consumption and increase in resources [11]. Consequently, instead of continue traditional way of increasing resources, parallel computing can be an integral part to make it possible to achieve massive performance in the system [12]. In current study, we emphasized how to achieve massive parallelism to deal digital city's assets in an efficient way. Previously, parallelism was achieved through homogenous (CPU cores) systems but now tread has been changed where data computation is performed with collaboration of CPUs and accelerated GPU devices [13, 21]. These accelerated GPU devices are very powerful that compute data in parallel using all integrated cores on it [14]. In order to program such heterogeneous computing system, a new parallel programming model is required that can achieve massive parallelism (combining coarse-grain, fine-grain and GPU computation) through inter-node and intra-node computation over a distributed cluster system [15]. Therefore we have proposed a new tri-level hybrid parallel computing model which is implementable on distributed cluster systems. Furthermore, rest of the paper is organized in such a way that, Sect. 2 describes the advances in HPC and its motivations for building smart cities. Section 3 contained the proposed tri-level hybrid framework and its work flow. Moreover, Sect. 4 depicted the impact of proposed model on smart city's assets.

2 Related Work

Parallel computing is an integral part for smart cities to provide efficient and innovative services for citizens. Therefore the pioneers in HPC started different projects to facilitate smart cities like 'Smart Japan' which was invented to catalyze its nation into sustainable growth. They planned a national strategy to promote their cities by connecting all manners of things through an intelligent ICT [31]. Keeping focus on international economics competitiveness, they are hoping to develop the first Exascale computing system in 2020 to facilitate smart cities. China also committed for their five years plan (2016–2020) to cultivating Smart Cities with smart buildings, new urbanization and smart grid initiatives. They already invested $13 billion to develop 277 city with smart features. Moreover, Government of Tianjin Binha China has reserved $2.5 billion to develop Tianhe III which will be used to promote smart societies [32, 33]. In the sequel, united state announced 160 million in 2015 for new smart cities development to deliver smart services. In september 2016, US government announced $80 million for smart societies initiative. In order to efficient processing of these smart services, they committed to deliver a powerful exascale computing system at the end of current decade which will be able to deliver 1 ExaFlops calculation per second by 20 MW power consumption [34]. Likewise, United Kingdom launched new forums for smart cities development named "Ministerial Smart Cities (MSC) and Future Cities Catapult (FCC)" [35]. The primary goal of these forums was to solve facing challenges during providing intelligent services in smart cities. Carry on promoting smart city's assets, national strategy and ICT plans in Singapore established their public & private sectors and branded as world's first smart city [36]. Singapore national strategy (SNS) planned to deliver a smart cities by addressing isolated challenges including traffic management, street lights, water management, weather forecasting, tracking and gene permutations etc. Leading to a rapid development in smart cities and efficient processing requirement, we have proposed a massive parallel computational model which will be applicable for future exascale computing systems.

3 Massive Parallel Computing Model

This section contained the proposed massive parallel computational model which is tri-level hybrid of MPI, OpenMP and CUDA. This model was developed specifically for attaining massive parallelism in heterogeneous cluster systems that can be achieved through inter-node, intra-node and accelerated GPU computation. However massive parallelism was in three levels of granularity including coarse-grain, fine-grain and finer-grain parallelism. Figure 2 depicts three levels of granularity from different computing environments.

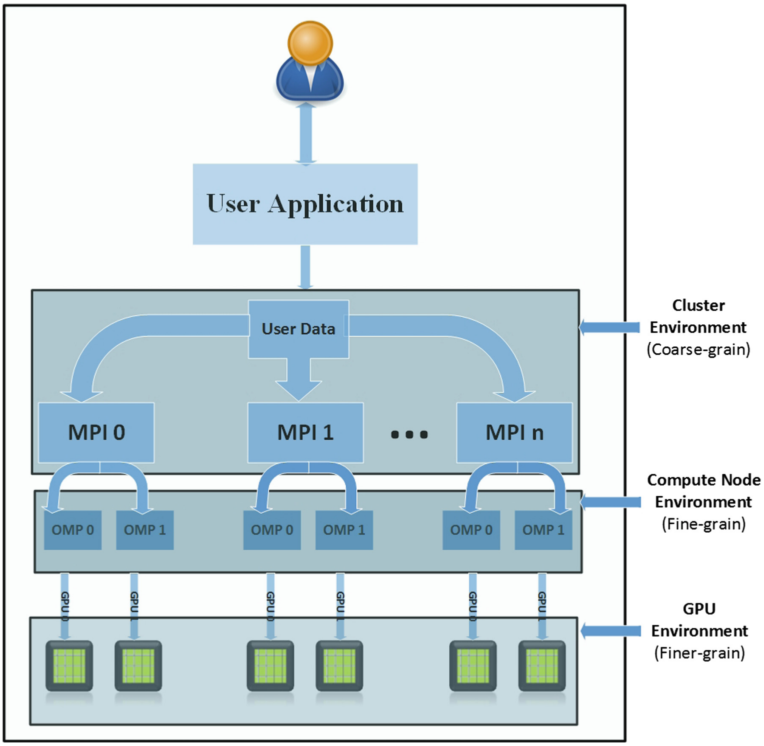

Fig. 2. Tri-level hybrid parallel computational model

In proposed massive parallel computational model, we implemented 'divide and conquer strategy algorithm' [17] that reduced 'n' problems into 'm' sub-problems and scattered over processors using different parallel programming libraries. Once processing of 'm' sub-problems was completed, the original solution for 'n' problem was obtained by gathering the solutions of 'm' sub-problems. Following this strategy, user interacts with user application through any native programming language (C/C++ or FORTRAN) where parallel computing statements are defined to be executed and broadcasted to parallel computing world. A detailed description of three parallel computing environments is given as follows:

3.1 Inter-node Computational

The first level of parallelism in our proposed model was achieved through inter-node communication. In parallel computing, our systems are not much intelligent, however programmer needs to inform the system that which statement will be executed in parallel. Once parallel computing statements are confirmed then broadcasted over all connected nodes in the system. Before scattering data over nodes, some necessary information of computer is gathered like number of connected nodes, CPU processors, available CPU threads per node, number of accelerated GPU devices and some other system configuration parameters. Based on these parameters, programmer analyze and distribute over connected system nodes using a standardized SIMD based message passing interface (MPI) library [16]. Generally in MPI, blocking (synchronous) and non-blocking (asynchronous) two prevalent mechanisms are being used to transferred and gathered from processors. Normally, blocking mechanisms is used when a strong synchronization is required due to dependency in data. In this case, the resources are reserved using some pre-defined MPI waiting statements until the processing is completed. Unlike, in non-blocking data is scattered and gathered asynchronously because there is no dependency in data [18, 19]. Resource are free immediately after data processing. Therefore, sending and receiving MPI mechanism depends on nature of data. As we are just distributing data over the processors which consequently provides coarse-grain parallelism at this level. Once data is transferred over connected nodes, it entered in second level of parallelism described in following section.

3.2 Intra-node Computational

Intra-node computation is second level of parallelism in proposed model which was thread level parallel computation within the computer. Intra-node computation was performed by parallelizing available CPU threads in a system. Now a days, many programming models are available to parallelize CPU threads like OpenMP [22], OpenACC [23] etc. We can choose any of them on the bases of model features and application requirements. Due to more optimized and advance features, ordinarily OpenMP is considered as the best model to parallelize code through CPU threads. Within outer region of OpenMP, we can define multiple OpenMP parallel sections as well as multiple directives to make fine granularity. Again, here programmer must be assured about the statements which are going to define in OpenMP parallel regions and sub sections. Once, OpenMP thread level computation is completed, we can start serial code execution at outside of OpenMP scope. According to our model, we start the third level of parallelism within the OpenMP regions and make run-time library calls.

3.3 Accelerated GPU Computational

Accelerated GPU computing is the third level of parallelism in our proposed model. Accelerated GPU devices are being used prevalently for massive parallel computation by consuming a small amount of energy. After completing second level parallelism over CPU cores, data is transferred on GPU devices to achieve finer granularity. At this stage, multiple GPU kernels are defined to which are programmed for GPU computation. Based on installed GPU structure and configuration, different parameters are defined and passed along with data through kernel call. In order to define a generic kernels, we can define 'template' class as a generic datatype, however after execution, kernel will return data with similar datatype as received. Many programming models are available now to program GPU devices including CUDA [27], OpenACC [26], OpenCL [28] and OpenMP [29] etc. Due to space limitations, we recommend kan et al. [30] for reader to find the best model for GPU programming. Based on optimization and advance features, CUDA is considered as best model for GPU programming yet. However, CUDA kernels return the results from GPU devices to Host CPU cores and complete third level of parallelism. Using divide and conquer strategy, processed data is returned to previous levels and eventually gathered by MPI master process.

4 Discussion

The urban question raised many challenges for current and emerging technologies to provide new data sources with greater spatial temporal solutions. Advances in HPC are becoming the reasons to deal such challenges to develop intelligent cities. Although, different approaches were proposed in past to deal HPC application but demand of innovative applications in smart cities imposes to rethink about introduce efficient computing technologies and models. These technologies should fulfil demand of future massive computational technology under strict constraints of energy consumption, number of cores in the system and budget as well. However, new adaptive, intelligent and energy efficient approaches are the fundamental demands of such HPC systems. In this context, we have proposed a novel massive parallel model in current study. The purpose of proposed model was to provide such platform where smart city's assets can be respond efficiently. This could be possible only through massive parallel computation in the system. This approach was tri-level hybrid of MPI, OpenMP and CUDA and provide massive parallelism through different levels. The major advantage of proposed model was that it is applicable for heterogeneous cluster systems where energy efficient accelerated GPU devices are installed in traditional CPU machines. Moreover, the proposed model was able combine all categories of parallelism including coarse-grain, fine-grain and finer-grain through inter-node, intra-node and GPU computation respectively. However, the proposed research can be an initiative HPC applications in smart cities.

5 Conclusion

Smart cities initiatives are becoming attention in marketplace to develop innovative services for the citizens through ubiquitous computing applications in urban spaces. Smart cities are required to deal a large number of citizens and intelligent devices that collect and share data in infinite ways. Such data is strictly belongs to the territory on which they are defined and perform computation through dispersed computing nodes.

In this way, parallel computing can play an integral part to compute smart city related applications. In current study, we have proposed a novel massive parallel computational model which is tri-level hybrid of MPI, OpenMP and CUDA. The proposed model is applicable for heterogamous cluster systems where a large number of nodes are connect along with accelerated GPU devices. Data computation is performed through three different parallel computing levels (coarse-grain, fine-grain and finer-grain) and complete computation efficiently. The proposed model will be considered as initiative for emerging exascale computing system. By future perspectives, tri-level hybrid model is required to implement at large level through different HPC innovative smart cities applications.

References

1. Balsamo, D., et al.: Wearable and autonomous computing for future smart cities: open challenges (2017)
2. Yaqoob, I., Hashem, I.A.T., Mehmood, Y., Gani, A., Mokhtar, S., Guizani, S.: Enabling communication technologies for smart cities. IEEE Commun. Mag. **55**(1), 112–120 (2017)
3. Lv, Z., et al.: Managing big city information based on WebVRGIS. IEEE Access **4**, 407–415 (2016)
4. Khare, P., Khare, A.: Internet of things for smart cities. In: Exploring the Convergence of Big Data and the Internet of Things, p. 96 (2017)
5. Urbanization UNFPA, 25 September 2017. http://www.unfpa.org/urbanization
6. Chourabi, H., et al.: Understanding smart cities: an integrative framework. In: Proceedings of 2012 45th Hawaii International Conference on System Science (HICSS), pp. 2289–2297. IEEE, January 2012
7. Allwinkle, S., Cruickshank, P.: Creating smart-er cities: an overview. J. Urban Technol. **18**(2), 1–16 (2011)
8. Klein, B., Koenig, R., Schmitt, G.: Managing urban resilience. Informatik-Spektrum **40**(1), 35–45 (2017)
9. Kettani, H.: Advances in high performance computing and their impact on smart cities. In: Proceedings of the 2017 International Conference on Smart Digital Environment, pp. 229–231. ACM, July 2017
10. Putman, W.M., Suarez, M.J.: GEOS Atmospheric Model: Challenges at Exascale 2017
11. Geist, A., Reed, D.A.: A survey of high-performance computing scaling challenges. Int. J. High Perform. Comput. Appl. **31**(1), 104–113 (2017)
12. Ferreira, K., et al.: Evaluating the viability of process replication reliability for exascale systems. In: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC), pp. 1–12. IEEE, November 2011
13. Mittal, S., Vetter, J.S.: A survey of CPU-GPU heterogeneous computing techniques. ACM Comput. Surv. (CSUR) **47**(4), 69 (2015)
14. Gruber, L., West, M.: GPU-accelerated Bayesian learning and forecasting in simultaneous graphical dynamic linear models. Bayesian Anal. **11**(1), 125–149 (2016)
15. Varbanescu, A.L., van Nieuwpoort, R.V., Hijma, P., Bal, H.E., Badia, R.M., Martorell, X.: Programming models for multicore and many-core computing systems. In: Programming Multi-core and Many-core Computing Systems, pp. 29–58 (2017)
16. Pajankar, A.: Message passing interface. In: Raspberry Pi Supercomputing and Scientific Programming, pp. 61–65. Apress, New York (2017)

17. Farzan, A., Nicolet, V.: Automated synthesis of divide and conquer parallelism. arXiv preprint arXiv:1701.08345 (2017)
18. Coti, C., et al.: Blocking vs. non-blocking coordinated checkpointing for large-scale fault tolerant MPI. In: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing, p. 127. ACM, November 2006
19. Shi, R., et al.: Designing efficient small message transfer mechanism for inter-node MPI communication on InfiniBand GPU clusters. In: Proceedings of 2014 21st International Conference on High Performance Computing (HiPC), pp. 1–10. IEEE, December 2014
20. Ashraf, M.U., Eassa, F.E.: Hybrid model based testing tool architecture for exascale computing system. Int. J. Comput. Sci. Secur. (IJCSS) **9**(5), 245 (2015)
21. Ashraf, M.U., Fouz, F., Eassa, F.A.: Empirical analysis of HPC using different programming models. Int. J. Mod. Educ. Comput. Sci. **8**(6), 27 (2016)
22. Chapman, B., Jost, G., Van Der Pas, R.: Using OpenMP: Portable Shared Memory Parallel Programming, vol. 10. MIT press, Cambridge (2008)
23. Kirk, D.B., Wen-Mei, W.H.: Programming Massively Parallel Processors: A Hands-on Approach. Morgan kaufmann, San Francisco (2016)
24. Ashraf, M.U., Khan, N.A.: Software engineering challenges for ubiquitous computing in various applications. In: Proceedings of 2013 11th International Conference on Frontiers of Information Technology (FIT), pp. 78–82. IEEE, December 2013
25. Weiser, M.: Ubiquitous computing. In: ACM Conference on Computer Science, p. 418, 1994 March
26. Wienke, S., Springer, P., Terboven, C., an Mey, D.: OpenACC — First Experiences with Real-World Applications. In: Kaklamanis, C., Papatheodorou, T., Spirakis, P.G. (eds.) Euro-Par 2012. LNCS, vol. 7484, pp. 859–870. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32820-6_85
27. Sanders, J., Kandrot, E.: CUDA by Example: An Introduction to General-Purpose GPU Programming, Portable Documents. Addison-Wesley Professional, Boston (2010)
28. Stone, J.E., Gohara, D., Shi, G.: OpenCL: a parallel programming standard for heterogeneous computing systems. Comput. Sci. Eng. **12**(3), 66–73 (2010)
29. Li, H.F., Liang, T.Y., Lin, Y.J.: An openMP programming toolkit for hybrid CPU/GPU clusters based on software unified memory. J. Inf. Sci. Eng. **32**(3), 517–539 (2016)
30. Kan, G., He, X., Ding, L., Li, J., Liang, K., Hong, Y.: A heterogeneous computing accelerated SCE-UA global optimization method using OpenMP, OpenCL, CUDA, and OpenACC. In: Water Science and Technology, p. wst2017322 (2017)
31. Obi, T., Ishmatova, D., Iwasaki, N.: Promoting ICT innovations for the ageing population in Japan. Int. J. Med. Inform. **82**(4), e47–e62 (2013)
32. Smart, E.C., Cooperation, G.C.: Comparative Study of Smart Cities in Europe and China. Current Chinese Economic Report Series. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-46867-8
33. Dongarra, J.: Sunway TaihuLight supercomputer makes its appearance. Nat. Sci. Rev. **3**(3), 265–266 (2016)
34. Simon, H.: Why we need exascale and why we won't get there by 2020. In: Optical Interconnects Conference, Santa Fe, New Mexico, May 2013
35. Moir, E., Moonen, T., Clark, G.: What Are Future Cities?: Origins, Meanings and Uses. PDF). Foresight Future of Cities Project and Future Cities Catapult (2014)
36. Nam, T., Pardo, T.A.: Conceptualizing smart city with dimensions of technology, people, and institutions. In: Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times, pp. 282–291. ACM, 2011 June