Advanced Logistic Systems Vol. 10, No. 2 (2016) pp. 5–10.

APPROACH TO ACCELERATE ALGORITHMS TO SOLVE LOGISTIC PROBLEMS WITH GPGPU

JÓZSEF KONYHA[1]–TAMÁS BÁNYAI[2]

Abstract: Nowadays algorithms that were used to solve logistic problems are more and more difficult. Thanks to the globalization the networked multiple supply chains becomes even more complex. Moreover the data available from these networks growing to unprecedented levels. The fast decision making is business critical. The article is about the approach of use the GPU to accelerate algorithms to solve logistic problems faster. After some possible areas of this field are discussed we tried out the popular scientific GPU programming library named Torch. Finally test results of the measurements are presented.

Keywords: *parallel computing, complex problems, speedup, GPGPU*

1. INTRODUCTION

Global companies are managing not only single supply chain but also coordinating networking multiple supply chains. The design and operation of these networking supply chains requires design and control methods based on complex algorithms. The design and control processes of networking companies can be divided into four significant parts: purchasing, production, distribution and recycling. The aims of the design of these sub processes are the followings: lead time reduction; utilisation of capacities; cost reduction; increasing of flexibility and transparency; enhancement of quality of products and processes.

The logistic processes of companies have been influenced by the following trends: increasing number of product types; decreasing production depth; automation of materials handling; just in time and just in sequence application; increasing information (big data solutions); networking suppliers; customer oriented logistics; cooperation; process oriented reengineering; outsourcing of production and services; life cycle assessment; globalised value making chain; multimodal transportation and distribution; virtual enterprises, industry 4.0 [1, 2].

The planning of logistic systems is an iterative engineering process, including problem definition, problem analysis, choosing solution methods, solution, sale, installation, diagnostics, evaluation and revision. The planning tasks of logistic systems represent a wide range of engineering problems: description of material handling systems; design of loading unit building systems; facility location; routing; reliability analysis; production planning (scheduling); selection of material handling devices, etc. The complexity of these planning tasks led to the use of heuristics and metaheuristics because problems are NP-hard. Heuristics and metaheuristics are used in the following cases: no analytical methods are known to solve the problem; there are exact methods to solve the problem, but they cannot be used; heuristics and metaheuristics are more flexible than exact methods, especially form the point of view taking constraints into consideration [3]. Computing power is much needed in several complex

[1] research assistant, Research Institute of Applied Earth Sciences, University of Miskolc
konyha@afki.hu

[2] associate professor, Institute of Logistics, University of Miskolc
alttamas@uni-miskolc.hu
H-3515 Miskolc-Egyetemváros, Hungary

algorithms. Difficult algorithms for calculations are running on big databases and results required in short time. Varieties of methods exist to achieve results in short time. There are two opportunities for the speedup: develop faster algorithm or replace the hardware with a more powerful one. In the practice exist several ways to accelerate a process: algorithm optimization, parallel computing, distributed computing, data reduction, method simplification. Each of them can be expedient [11]. GPGPU (General-Purpose Computing on Graphics Processing Units) is a very promising trend of using parallel computing today. GPUs are sources of massively parallel computational power that can be used to solve large set of problems. The major manufacturers are increasingly support the use of general-purpose this hardware. A GPU can accelerate an algorithm if it computationally intensive and massively parallelizable. If the application cannot fulfil the criteria, it will run slower on a GPU [11, 12].

2. Literature review

The solution of logistics related engineering problems has a wide range of literature sources. The reliability analysis and risk management of logistics systems focuses on disruption risk management, operational risk control and logistics service risk [4, 5]. Multi stage stochastic network flow models can be used to model reliability and risk problems [6]. In the case of complex systems the optimisation algorithms need long computation time. The routing problems are in the most cases integrated problems like location-routing, inventory-routing, multi-echelon routing, routing problems with loading constraint [8]. Facility location problems means the geographical positioning of objects, like plants, warehouses, machines etc. The facility location problems are related with purchasing and sourcing, supply chain design and integration, process management and take social, economic and environmental aspects into consideration [9]. The used heuristic algorithms require long computation time. Scheduling problems represents a quite wide range of NP-hard problems with high computational complexity. The utilised heuristic methods include ant colony optimisation, differential evolution, electro magnetism, genetic algorithm, imperialist competitive algorithm, memetic algorithm, particle swarm optimisation, simulated annealing, tabu search, iterated greedy [10]. As this short review shows, the algorithms used to solve networked logistic systems are CPU intensive, so the GPGPU based optimisation is a reliable way to reduce computational time of complex algorithms.

In the literature the number of GPGPU articles is growing. GPU acceleration is widely used. As the *Figure 1* shows a massive performance difference started about 2003. In 2012 the fastest available GPU was seven times faster than the fastest CPU. Scientist and engineers have to take advantage of the potential of GPU if they want to stay in race [13].

Theodore Kim and Ming C. Lin used GPUs to simulate the growth of ice crystals. They experienced a factor of 9 speedup, making possible the interactive simulation. They used Bolzmann Methods for gas and fluid flow simulations [14]. Signal and image processing are well parallelizable processes. The authors of [15] are accelerated the processing of CT and MR images. The performance speedup factors were ~5 for CT and ~3 for MR images. They used 3D rendering systems to produce realistic, high quality output images [15]. M. Pietro'n, A. Byrski and M. Kisiel-Dorohinicki implemented a hybrid system for solving the difficult black-box Golomb Ruler problem. Their goal was to find heuristics for solving Golomb Ruler problem that can be accelerated in GPGPUs and compare them with highly optimized counterparts implemented in CPU. They GPGPU implementation 10 times faster than the fastest multicore CPU one [16]. One article from 2015 is about the use of GPUs in embedded systems. The authors studied the performance and programmability of a GPU in embedded environment with different programming frameworks. The different frameworks they tested has proven to be very well suited for data processing or other parallel tasks [17]. In numerous article the authors trying to accelerate different algorithms with GPGPU in several areas of life. As these some emphasized example shows, the GPU suitable to accelerate lot of algorithm successfully.

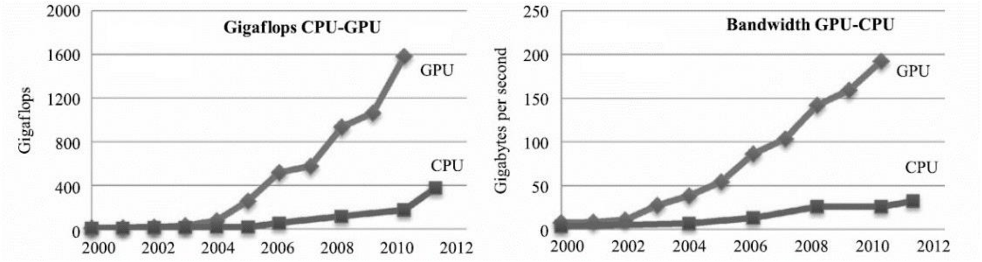

Figure 1. Historical performance comparison of GPU and CPU [13]

3. GPGPU PROGRAMMING WITH CUTORCH

GPU-accelerated computing is the use of a graphics processing unit. This hardware has a massive computational power which can be turned into a general-purpose computing. Today a general desktop CPU have 2–8 cores, in contrast a modern GPU have hundreds of cores. The newly released GEFORCE GTX 1080 with PASCAL architecture has 2580 CUDA cores [18]. A typical architecture of the GPU and the CPU can be seen in *Figure 2.*

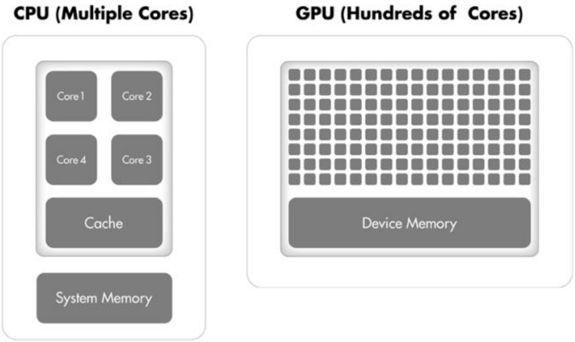

Figure 2. Difference in the architecture of CPU and GPU [12]

The major manufacturers offering high level programming APIs to their devices. It was designed to hide the hardware. One of the biggest manufacturers, the nVidia created their own solution named CUDA (Compute Unified Device Architecture). AMD is one other major manufacturer for graphic devices. They also have a solution for GPGPU programming API named CTM (Close to Metal). Hereinafter this article is about CUDA only.

Torch is a scientific computing framework based on LuaJIT. Lua is a powerful programming language. It supports procedural programming, functional programming, object-oriented programming, and data-driven programming. In this case just-in-time (JIT) compilation means compilation done during execution of a program. The biggest advantage of the JIT compilation is a real-time architecture targeting. It can be very important because of the wide variety of graphical devices. Torch have a huge number of scientific libraries for machine learning, computer vision, signal processing, parallel processing, image, video, audio and networking among others. CUDA can be used in torch with the cutorch library that provides CUDA based methods.

4. CPU VS GPU PERFORMANCE COMPARISON WITH CUTORCH

Our aim was to develop and test a simple algorithm to see how torch works on the two different architectures, on CPU and GPU. To use the GPU in torch the cutorch library required. Cutorch provides a CUDA backend for torch.

A simple matrix multiplication was chosen for the tests. It is a computationally intensive and massively parallel calculation. It is a good candidate to take advantage of the architecture of a GPU.

The tests were run on Intel i5-6600 CPU, nVidia GTX 750 GPU and nVidia GTX 960 GPU. Table I. shows the selected hardware unit details. There are significant differences between the numbers of integrated cores in the units.

Table 1
Details of the tested processing units

Unit type	Manufacturer	Name	No. of cores	Core frequency
CPU	Intel	i5-6600	4	3300 MHz
GPU	nVidia	GTX 750 2GB	640	1020 MHz
GPU	nVidia	GTX 960 4GB	1024	1241 MHz

For measurements the popular Linux distribution named Fedora was used. The algorithm was tested with different matrix sizes and the calculation time was recorded and compared. The matrix multiplication results can be seen on the *Figure 3.* As the diagram shows, in case of a small matrix the performance difference between the two different architecture of processing units was minimal. As the size of the matrix was increases the gap is growing.

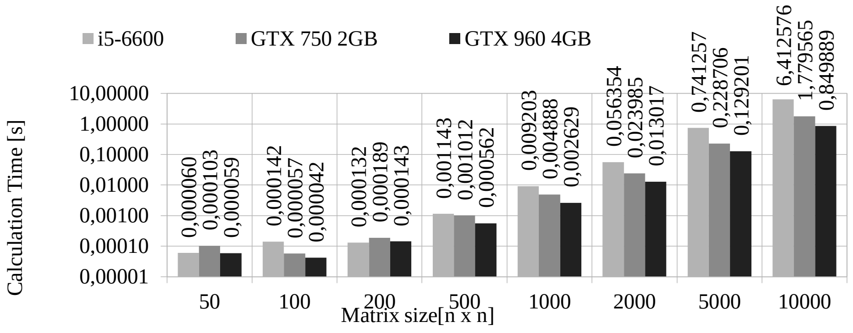

Figure 3. Matrix multiplication results on different processing units

Figure 4 can be divided into two parts. The difference between the processing times on GPU and CPU is fluctuate up to 200 x 200 dimensions according to the saved time (CPU$_{\text{time}}$ – GPU$_{\text{time}}$) / CPU$_{\text{time}}$.

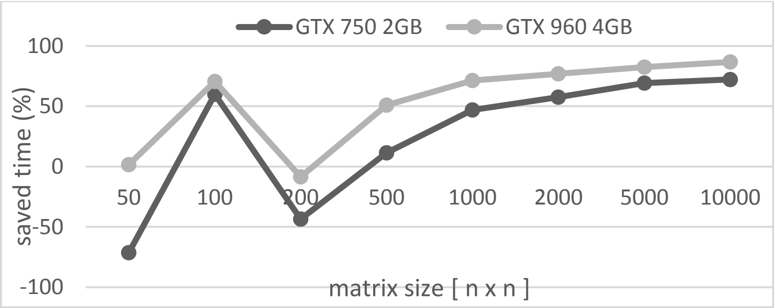

Figure 4. Saved time

For larger matrix sizes the computational time on the CPU increases rapidly. The saved time percentage for larger matrix multiplication is growing.

5. Summary

Companies in logistic sector are making efforts to a fast, data-driven effective business decision making. Today logistic networks are growing. There are a lot of complex algorithms to design and optimize them. The amount of data we have today is far beyond the processing power of conventional systems.

Scientists and engineers are trying to accelerate their algorithms in many ways. One of them is parallel programming. We discussed the differences between the CPU and the GPU. GPGPU acceleration is a powerful way to speedup algorithms which are computationally intensive. A simple performance test was carried out and the results are presented. We calculated the saved time also. Future research direction is the optimization of large scale inventory management systems [20–22] especially in the case of demand changes and other manufacturing and service related demand driven supply chains.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 691942. This research was partially carried out in the framework of the Center of Excellence of Mechatronics and Logistics at the University of Miskolc.

References

[1] Qin, J.–Liu, Y.–Grosvenor, R.: A Categorical Framework of Manufacturing for Industry 4.0 and Beyond. *Procedia CIRP*, 52 (2016), 173–178.

[2] Kolberg, D.–Zühlke, D.: Lean Automation enabled by Industry 4.0 Technologies. *IFAC-PapersOnLine*, Vol. 48, No. 3 (2015), 1870–1875.

[3] Marti, R.–Reinelt, G.: *The Linear Ordering Problem, Exact and Heuristic Methods in Combinatorial Optimization.* Springer-Verlag, Berlin–Heidelberg, 2011.

[4] CHOI, T. M.–CHIU, C. H.–CHAN, H. K.: Risk management of logistics systems. *Transportation Research Part E: Logistics and Transportation Review,* 90 (2016), 1–6.

[5] LAM, H. Y.–CHOY, K. L.–HO, G. T. S.–CHENG, S. W. Y.–LEE, C. K. M.: A knowledge-based logistics operations planning system for mitigating risk in warehouse order fulfilment. *International Journal of Production Economics,* 170 (2015), 763–779.

[6] ALEM, D.–CLARK, A.–MORENO, A.: Stochastic network models for logistics planning in disaster relief. *European Journal of Operational Research,* Vol. 255, No. 1 (2016), 187–206.

[7] CÔTÉ, J. F.–GUASTAROBA, G.–SPERANZA, M.: The value of integrating loading and routing. *European Journal of Operational Research,* Vol. 257, No. 1 (2017), 89–105.

[8] BELLE, J. V.–GERMAIN, B. S.–PHILIPS, J.–VALCKENAERS, P.–CATTRYSSE, D.: Cooperation between a Holonic Logistics Execution System and a Vehicle Routing Scheduling System. *IFAC Proceedings,* Vol. 46, No. 7 (2013), 41–46.

[9] CHEN, L.–OLHAGER, J.–TANG, O.: Manufacturing facility location and sustainability: A literature review and research agenda. *International Journal of Production Economics,* 149 (2014), 154–163.

[10] ALLAHVERDI, A.: A survey of scheduling problems with no-wait in process. *European Journal of Operational Research,* Vol. 255, No. 3 (2016), 665–686.

[11] LÍŠKA, M.–OČKAY, M.: General-Purpose Computing on Graphics Processing Units: New trends for computational acceleration. *Science & Military,* 2, 2007.

[12] REESE, J.–ZARANEK, S.: GPU Programming in MATLAB. *Mathworks Newsletters,* 2012.

[13] BRODTKORBA, A. R.–HAGENA, T. R.–SÆTRA, M. L.: Graphics processing unit (GPU) programming strategies and trends in GPU computing. *Journal of Parallel Distributed Computing,* 73 (2013), 4–13.

[14] KIM, T.–LIN, M. C.: Visual Simulation of Ice Crystal Growth. *Symposium on Computer Animation,* 2003.

[15] ZHUA, F.–GONZALES, D. R.–CARPENTER, T.–ATKINSON, M.–WARDLAW, J.: Parallel perfusion imaging processing using GPGPU. *Journal of Computer Methods and Programs in Biomedicine,* 108 (2012), 1012–1021.

[16] PIETRO'N, M.–BYRSKI, A.–KISIEL-DOROHINICKI, M.: GPGPU for Difficult Black-box Problems. *Proceedings of the ICCS 2015 International Conference on Computational Science,* Vol. 51 (2015), 1023–1032.

[17] FABER, P.–GRÖSSLINGER, A.: A Comparison of GPGPU Computing Frameworks on Embedded Systems. *Proceedings of International Federation of Automatic Control,* 2015, 240–245.

[18] NVIDIA, 2016. *GeForce GTX 1080 technical specifications.*
http://www.geforce.com/hardware/10series/geforce-gtx-1080

[19] FÜVESI, V.–KONYHA, J.: Review of machine learning toolboxes. *Műszaki Tudomány az Észak-Kelet Magyarországi Régióban,* 2016, 116–224.

[20] KORPONAI, J.–BÁNYAI, Á.–ILLÉS, B.: The effect of the demand changes on the inventories. In: SCHENK, Michael (ed.): *8th International Doctoral Student Workshop on Logistics.* 2015, 29–34.

[21] KORPONAI, J.–BÁNYAI, Á.–ILLÉS, B.: The effect of the supply accuracy and the demand-changes on the inventories and on the costs. *Advanced Logistic Systems. Theory and Practice,* Vol. 9, No. 1 (2015), 5–16.

[22] KORPONAI, J.–BÁNYAI, Á.–ILLÉS, B.: The effect of the supply accuracy on the inventories. *Production Systems and Information Engineering,* 7 (2015), 15–31.