GPU performance impact of the Durkin's radio propagation algorithm

Leonid Djinevski, Sonja Filiposka, *Member, IEEE*, Dimitar Trajanov, *Member, IEEE*, and Igor Mishkovski, *Member, IEEE*

***Abstract* — Network simulators are important tools when used by researchers. However, complex simulations for large-scale networks (especially if the scenarios are realistic, i.e. introducing terrain details) results in unreasonably long response from the simulator. High performance parallel resources like GPU devices are solution for obtaining results in reasonable time. Additionally, the latest GPU generations have architecture that enables the developer to configure different input parameters, in order to achieve better performance. In this paper we evaluate the performance impact of these configurations for various resolutions of terrains. The obtained results show that there is an impact on the performance, however it is not significant.**

***Keywords* — ad hoc networks, GPU, NS-2, simulation.**

I. Introduction

THE research in the area of ICT networks makes wide use of simulation tools before deploying real solutions. There are many tools (proprietary and open-source) that provide discrete event simulation. However, these simulators do not scale well for medium and large networks, thus their execution time is unreasonably long. This drawback is significantly expressed for wireless ad hoc network simulation, especially if terrain details are taken into consideration. The introduction of 3D terrains can have a great impact on the simulation duration since it includes additional geometrical operations, which are compute intensive.

Today's modern high-end graphic processors provide high performance at very small cost. The latest Kepler architecture of GPU processors from Nvidia achieve up to 3TFLOPS. Much research [1] (in many scientific areas) during the last few years has been done, where many examples document the huge execution throughput of general-purpose applications on GPU devices (GPGPU) [2]. Since the introduction of CUDA [3] and OpenCL [4], development of general purpose applications on graphic processors has accelerated. This is especially noticeable when analyzing the generations of the GPU architectures. Our interest in this paper is focused on the GPU memory hierarchy and other input parameters that are configurable. This results different configurations to impacts the performance of the GPU execution.

In recent work [5], we have developed a 3D terrain aware extension of the NS-2 network simulator [6], and ported the implementation for GPU execution [7]. We have further improved the performance of the extension by utilizing better terrain representations [8], like the Triangular Irregular Network (TIN) [9][10]. In this paper, we are presenting an analysis of the performance of the parallel extension executed on GPU, using different parameter inputs, in order to determine the optimal performance.

The paper is organized as follows: Section II presents a short overview of the GPU memory hierarchy. The Durkin's radio propagation algorithm is defined in Section III. The theoretical analysis was presented in Section IV, followed by the used testing methodology for the conducted experiments in Section V. The Results are described and discussed in Section VI. The conclusion of the paper is given in Section VII.

II. GPU memory hierarchy

The memory hierarchy of the GPU devices is placed as SIMD parallel machines. The latest generation of the Nvidia's Kepler architecture [11] is presented on Fig. 1. The memory hierarchy consists of 3 levels: the off-chip global memory, the on-chip Level 2 memory shared by the Streaming Multiprocessors (SM), and the on-chip Level 1 memory shared by the Scalar Processors (SP) within a single SM. Additionally, there are registers dedicated as private memory per single SP. Treads run per SP, and are organized in thread blocks. Thread blocks are executed per SM.

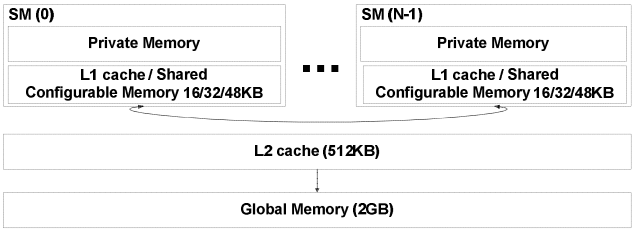

Fig. 1. GPU memory hierarchy.

Leonid Djinevski, FON University, av. Vojvodina, 1000 Skopje, Macedonia (tel: 389-2-2445619, e-mail: leonid.djinevski@fon.edu.mk)

Sonja Filiposka, Faculty of Computer Science and Engineering, ss. Cyril and Methodius University - Skopje, ul. Rugjer Boshkovikj 16, 1000 Skopje, Macedonia (tel: 389-2-3099153, e-mail: sonja.filiposka@finki.ukim.mk).

Dimitar Trajanov, Faculty of Computer Science and Engineering, ss. Cyril and Methodius University - Skopje, ul. Rugjer Boshkovikj 16, 1000 Skopje, Macedonia (tel: 389-2-3099153, e-mail: dimitar.trajanov@finki.ukim.mk)

Igor Mishkovski, Faculty of Computer Science and Engineering, ss. Cyril and Methodius University - Skopje, ul. Rugjer Boshkovikj 16, 1000 Skopje, Macedonia (tel: 389-2-3099153, e-mail: igor.mishkovski@finki.ukim.mk)

The focus of interest of this paper is the L1 cache memory because its size and set associativity which is configurable (16/32/48KB, with 4/4/6-way accordingly) during runtime. The used cache-line (cache block) size is also configurable (64B or 128B) during compilation.

Additionally, we are interested in the shared memory which is coupled with the L1 cache memory configurations. Thus for 16/32/48KB of L1 cache configurations, the shared memory is configured as 48/32/16KB respectively. The L1 cache memory and the shared memory form the 64KB Level 1 memory.

III. DURKIN'S RADIO PROPAGATION ALGORITHM

In this section we present the Durkin's algorithm [5], which makes use of diffraction and shadowing effects. The classical Fresnel solution is used for obtaining the diffraction loss, which is described by the following equations (1), (2) and (3):

$$G_d(dB) = 20\log|F(v)|. \quad (1)$$

$$v = h\sqrt{\frac{2(d_1+d_2)}{\lambda d_1 d_2}}. \quad (2)$$

where $F(v)$ is the Fresnel integral, which is a function of the Fresnel-Kirchoff diffraction parameter v defined in (2). The approximation of (1) is given by:

$$\begin{aligned} G_d(dB) &= 0, && v \le -1 \\ G_d(dB) &= 20\log(0.5 - 0.62v), && -1 \le v \le 0 \\ G_d(dB) &= 20\log(0.5 * e^{-0.95v}), && 0 \le v \le 1 \\ G_d(dB) &= 20\log(0.4 - 0.62v), && 1 \le v \le 2.4 \\ G_d(dB) &= 20\log(\frac{0.225}{v}), && v > 2.4 \end{aligned} \quad (3)$$

Based on the conditions: if there is Line Of Sight (LOS), whether first Fresnel zone clearance is achieved or there is inadequate first Fresnel zone clearance, the durkin's algorithm using the diffraction parameter v can determine the path loss of for a given transmitter/receiver (TR) pair [12][13].

IV. THEORETICAL ANALYSIS

Our parallel implementation for GPU execution uses the shared memory. The access of the shared memory is very fast (almost as fast as the private memory), thus the GPU performance increase. The size of the shared memory is set to be a factor of the threadblock in order to efficiently utilize the GPU resources. Since the size of a threadblock is configurable at runtime, performance difference is expected for different threadblock sizes.

According with the principle of cache locality [14], the larger the cache size is, the better is the performance of the execution. There are many papers that present results that show discrepant performance for different size configurations of the L1 cache memory.

Authors in [15] state that there is not specific procedure for choosing the optimal configuration, i.e., it is best to conduct try and error approach.

H1: We set a hypothesis that for the largest L1 cache configuration, and the largest threadblock, will achieve best performance.

V. TESTING METHODOLOGY

The used technology is described in this section. Series of experiments were performed in order to determine the optimal performance using different cache configurations. The testing environment is consisted of Intel i7-3770 CPU@3.40Hz, with 32GB of RAM at 1.60GHz and Nvidia GeForce GTX 680 GPU. The experiments were compiled using the Nvidia's nvcc compiler using the CUDA 5.0 toolkit. The operating system Ubuntu 12.04 LTS is running on the described hardware infrastructure.

A. Experiments

We have conducted three experiments for each size configuration of the L1 cache memory. Four Series of test cases were performed for block sizes of 64, 128, 256 and 512. For each of the series, different resolution of the terrain details were varied, starting form 1000 triangles, up to 12000 triangles. Terrain resolution under 1000 triangles does not resemble the terrain, thus more triangles provides more realistic scenario.

B. Test data

All test cases are iterated 11 times, and the average execution value of the GPU kernel was adopted, excluding the first iteration. We evaluate the performance by the inverse execution time, which is proportional to the speed of each test case. We denote I_{16} as the inverse execution time for the L1 configuration of 16KB, I_{32} and I_{48} as the inverse execution times for the L1 configurations of 32KB and 48KB respectively. In order to evaluate the performance impact of different cache configurations, we define *relative speedup* indicators. The relative speedup of the 32KB compared to the 16KB configuration of the L1 cache size is defined in (4). Relations (5) and (6) define the relative speedup of the 48KB compared to 32KB and 16KB respectively.

$$S_{32R16} = [\frac{I_{32}}{I_{16}}]. \quad (4)$$

$$S_{48R32} = [\frac{I_{48}}{I_{32}}]. \quad (5)$$

$$S_{48R16} = [\frac{I_{48}}{I_{16}}]. \quad (6)$$

VI. RESULTS

The obtained results from the experiments that were performed according to the testing methodology described in Section V are presented in this section.

In Figure 2-4 we present the speed of the Durkin's algorithm for 16KB, 32KB and 48KB of L1 cache size configurations, where the x-axis N represents the resolution of the terrain detail, and *block 64/128/256/512* stands for the threadblock size accordingly. The best performance is not the test case with the largest threadblock (512), but the smallest block of (64) elements, which validates the conclusions stated in [15].

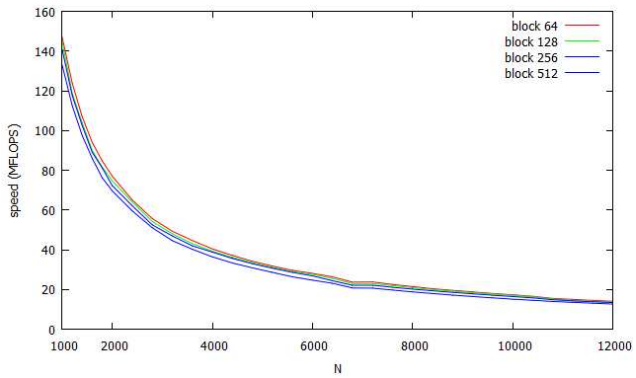

Fig. 2. Speed for 16KB of L1 cache size configuration.

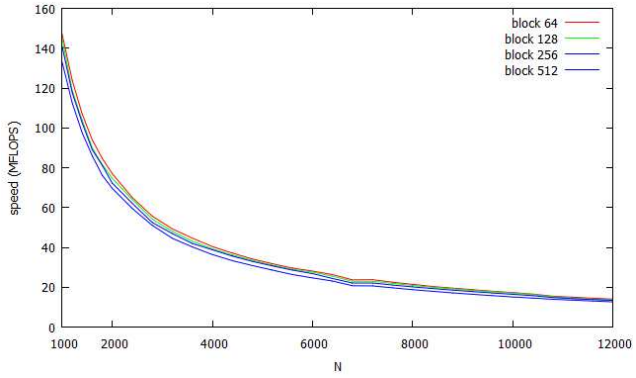

Fig. 3. Speed for 32KB of L1 cache size configuration.

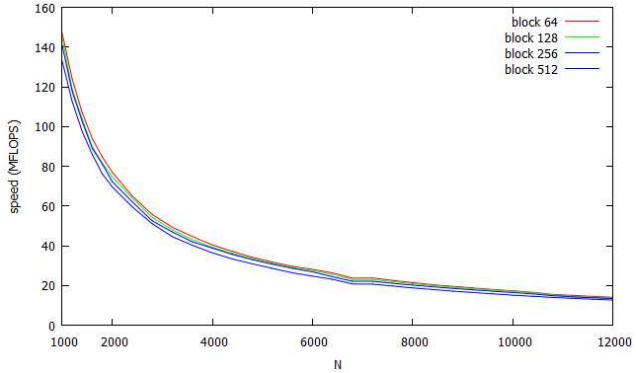

Fig. 4. Speed for 48KB of L1 cache size configuration.

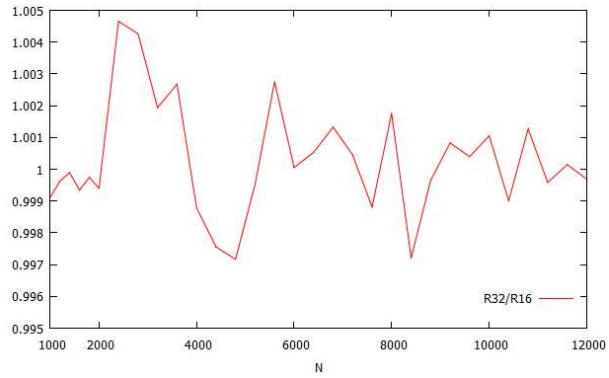

Fig. 5. Relative speedup S_{32R16}.

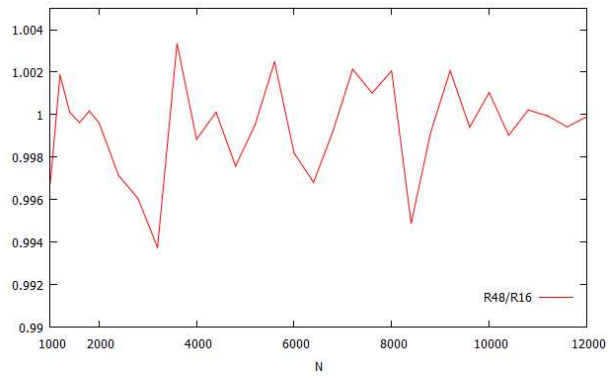

Fig. 6. Relative speedup S_{48R16}.

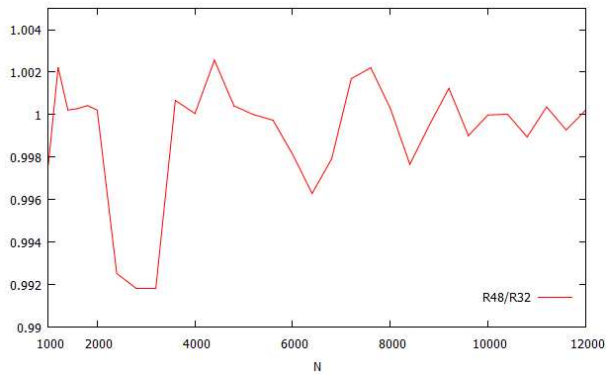

Fig. 7. Relative speedup S_{48R32}.

We present our alternative approach towards analyzing the results by evaluation of the relative speedup indicator for different size configurations of the L1 cache memory. Figures 5-7 present the behavior of the S_{32R16}, S_{48R16} and S_{48R32} indicators respectively. There is a trend of stabilizing the speedup indicator for larger number of triangles (resolution) of the terrain representation. This can be easily noticed at Fig. 7, where the 48KB configuration performance is converging to speed is converging to the 16KB configuration. However, there is no significant impact on the performance, and the obtained results are discrepant for smaller resolutions of the terrain representation.

VII. Conclusion

In this paper we presented the optimal implementation of the Durkin's 3D terrain aware radio propagation model for the NS-2 network simulator, by efficient use of Level 1 memory. We have conducted three experiments and series of test cases that evaluate the performance impact for different size configurations of the L1 cache memory, and threadblock size.

Our hypothesis regarding the configuration with the largest size of 48KB for the L1 cache memory, and the largest threadblock size to achieve better performance, was proven wrong in the first part. The results show that the relative speeds for 16KB and 48KB are very similar and very close to the speed achieved by 32KB L1 configuration. The main factor for the small impact on the performance is the large L2 cache memory, and the small L1 cache size which generates huge cache capacity misses.

Additionally, the results disprove the second part of our hypothesis, since the smallest threadblock achieves best performance.

Our extension runs in single precision, since the numerical data for Durkins algorithm are in the float range. This makes it very suitable for utilizing the resources of the GPU, although double precision support has improved in the Kepler architecture.

For future work, we plan to evaluate the performance of the Durkin's algorithm for terrain aware radio propagation extension, with utilizing novel GPU technologies, like dynamic parallelism. Additionally, we plan to analyze the performance on a multi-GPU infrastructure.

REFERENCES

[1] S. Che, M. Boyer, J. Meng, D. Tarjan, J.W. Sheaffer, and K. Skadron. (2008). A performance study of general-purpose applications on graphics processors using cuda. Journal of parallel and distributed computing, 68(10):1370–1380.

[2] Harris, M.J., "General Purpose Computation on GPUs", retrieved February 2013 from http://www.gpgpu.org/.

[3] NVIDIA CUDA, retrieved February 2010 from, http://developer.nvidia.com/object/cuda.html/.

[4] The OpenCL Specification, Version 1.0, document Revision 43, 2009, retrieved February 2010 from http://www.khronos.org/opencl/.

[5] Filiposka, S., Trajanov, D.: Terrain-aware three-dimensional radiopropagation model extension for ns-2. Simulation 87(1-2), 7{23 (2011).

[6] ns-2, network simulator (1989).

[7] Djinevski, L., Filiposka, S., Trajanov, D., Mishkovski, I.: Accelerating wireless network simulation in 3D terrain using GPUs. Tech. Rep. SoCD:16-11, University Ss Cyril and Methodius, Skopje, Macedonia, Faculty of Information Sciences and Computer Engineering (June 2012).

[8] Vuckovik, M., Trajanov, D., Filiposka, S.: Durkins propagation model based on triangular irregular network terrain. In: ICT Innovations 2010, pp. 333-341. Springer (2011)

[9] Unit 39-the tin model (2011), http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u39.html

[10] Zeiler, M.: Modeling our world, environmental systems research institute. Inc. Redlands, California (1999).

[11] NVIDIA, "Next generation cuda compute architecture: Kepler gk110," 2012.

[12] Edwards, R., Durkin, J. 1969. Computer Prediction of Service Area for VHF Mobile Radio Networks. Proceedings of the IEEE, Vol. 116, No. 9, pp. 1493-1500, 1969.

[13] Rappaport, T. S., 2002. Wireless Communications: Principles and Practice, Prentice Hall, New York.

[14] Grama, A., Karypis, G., Kumar, V., Gupta, A.: Introduction to Parallel Computing (2nd Edition). Addison Wesley, 2nd edn. (Jan 2003).

[15] Volkov, Vasily. "Better performance at lower occupancy." In Proceedings of the GPU Technology Conference, GTC, vol. 10. 2010.