Reliability Modeling of MEMS devices on CUDA based HPC Setup

Rohit Pathak[ǂ] and Satyadhar Joshi[†]
[ǂ] Acropolis Institute of Technology & Research, Indore, Madhya Pradesh, India
[†] Shri Vaishnav Institute of Technology & Science, Indore, Madhya Pradesh, India
rohitpathak@ieee.org, satyadhar_joshi@ieee.org

Abstract*-In this paper we have reviewed the development in CUDA and the implementation of various distribution that exists in the reliability for MEMS based devices on a CUDA setup. The various distributions can be highly optimized so that the system can be simulated highly on CUDA. We have shown the type of distribution may vary from exponential to binomial to other that are being proposed recently. The Reliability modeling codes to calculate reliability function, failure rate function, Mean time to failure (MTTF) and Mean Residual time (MRT) are proposed for MEMS technology for these specific calculations needs to be performed for which CUDA plays a very important role. It is observed that High Performance Computing (HPC) can be used to optimize reliability calculation and help to accelerate research in reliability of MEMS. The three key abstractions of CUDA i.e. hierarchy of thread groups, shared memories, and barrier synchronization are exposed as a set of extensions to C language, which provides fine-grained data parallelism and thread parallelism, nested within coarse-grained data parallelism and task parallelism. The key is division of the computations of Reliability analysis into crude sub-problems that can be solved parallely in isolation independently, and then into finer pieces that can be executed in parallel with mutual cooperation among them. Allowing threads to solve each sub-problem cooperatively, this division of problem preserves expressivity of language. Each sub-problem is thus scheduled to be solved on any of the available processor cores allowing transparent scalability. Thus computations of Reliability analysis can be performed by using a compiled CUDA program that can execute on any number of GPU cores. During the programming we need not know the exact configuration and thus only the runtime system needs to know the physical processor count.

I. INTRODUCTION

There are various distributions needed in MEMS reliability computations, therefore there is a need to distribute and analyze the various distributions available for calculation of reliability. Also in recent times CUDA has been implemented to compute complex scientific computation where similarity type of operations can be highly optimized. Recently the various distribution used in reliability has been limited to exponential and simpler ones, but since the number of devices has increased there is a requirement of high computations power. CUDA can be used to harness the power of GPU's for computational intensive tasks [37, 38]. M. Harris et al. showed how the many core GPU of NVIDIA can be used for many computational intensive problems and tasks, he also discussed how CUDA is helping the advancement of research industry [29]. David et al. compared various computation platforms such as CPU, GPU, FPGA and massively parallel processors by performing various tests on each setups, and also estimated the power efficiency of each device [31], from which its clear that GPU's can deliver high amount of computational power.

GPU can be used to greatly boost up algebraic computations as shown by Vasily et al. [15], who provided some algorithmic optimizations which boosted the linear algebraic computations on NVIDIA GPU. GPU can also be used for general computations for games and physics simulation as shown by Marcelo et al. who proposed a new game loop architecture which used GPU in collaboration with CPU for mathematical computations [16]. CUDA can also boost up computations which require processing of large graphs having millions of vertices [17]. Numerical fluid mechanics computations such as Lattice-Boltzmann can be accelerated upto *eighteen* times using a *three* GPU setup [21]. Tölke et al. proposed an alternate approach for implementing 2-D Lattice Boltzmann on CUDA which resulted in an incredible speedup [23].

High performance molecular dynamics and simulations can be achieved in computational biology by using CUDA. One such implementation was shown by Weiguo Liu et al. and he showed how his algorithms can utilize the parallel power of GPU [28]. In computational biology, tissue morphology and pathology require huge computation power for 3D visualization and reconstruction of large sample tissue structure, GPU and CUDA offers such power by offering its highly parallel structure which can boost the registration of large sets of high resolution images by nearly three to six times [22]. Smith-waterman algorithm of computational biology can be accelerated nearly *twenty three* times using CUDA [20].

A large computation power is required by radio astronomy interferometers and is increasing quadratically with increase in array signals. Chris Harris et al. proposed to use GPU for accelerating such computations upto two times [24]. Various implementations of volumetric Mass-Spring-Damper systems were investigated in CUDA and the performances were also compared with the earlier systems utilizing OpenGL [25]. Quantum Computer simulation is one of the most resource demanding problem which requires high memory and

978-1-4244-4570-7/09/$25.00 ©2009 IEEE

[7] Lisec T., Huth C., Wagner B., "Dielectric Material Impact on Capacitive RF MEMS Reliability," *Proc. 34th European Microwave Conference (Amsterdam)*, 2004.

[8] Nan L., Souran M., "Reliability based MEMS system Modeling and Optimization," *Proc. 44th Annual International Physics Symposium on Reliability (San Jose)*, 2006.

[9] Guang-Ren S., Li-Nuan T., Tzu-Yuan C., Cheng Y.T., Lin T.K., Wensyang H., "Ni-P-CNTs Nanocomposite Film for MEMS Applications," *Proc. 4th IEEE Conference on Nanotechnology*, 2004.

[10] Guang-Ren S., Yu-Ting C., Li-Nuan T., "Synthesis and Characterization of Ni–P-CNT's Nanocomposite Film for MEMS Applications," *IEEE Transaction on Nanotechnology,* Vol. 4, No. 5, Sep. 2005.

[11] Li-Nuan T., Guang-Ren S., Cheng Y.T., Wensyang H., "Power and Reliability Improvement of an Electro-Thermal Microactuator Using Ni-Diamond Nanocomposite," *Proc. Electronic Components and Technology Conference*, 2004.

[12] Victor S., Alastair T., "MEMS Failure Analysis and Reliability," *Proc. of IPFA (Singapore)*, 2003.

[13] Rohit P., Satyadhar J., Salman Ahmed, "Wireless Sensor Network: Simulink Modeling and Reliability of Nano-Nodes," *Proc. 2nd International Workshop on Electron Devices and Semiconductor Technology, IEDST 2009*, 1-2 Jun. 2009, DOI=10.1109/EDST.2009.5166129.

[14] Rohit P., Satyadhar J., "Multi-Scale Modeling and Analysis of Nano-RFID systems on HPC setup," *Contemporary Computing, Communications in Computer and Information Science, Springer Berlin Heidelberg*, Vol. 40, DOI=10.1007/978-3-642-03547-0_61.

[15] Vasily V., James W.D., "Benchmarking GPUs to Tune Dense Linear Algebra," *Proc. ACM/IEEE conference on Supercomputing, Conference on High Performance Networking and Computing*, 2008.

[16] Marcelo P.M.Z., Esteban W.G.C., Aura C., Anselmo M., Regina C.P.L.T., Paulo A.P., Luis V., Bruno F., "A game loop architecture for the GPU used as a math coprocessor in real-time applications," *Computers in Entertainment (CIE), Media Arts, Best of SBGames 2007*, Vol. 6 , Issue 3, Oct. 2008.

[17] Pawan H., Narayanan P.J., "Accelerating Large Graph Algorithms on the GPU Using CUDA," *High Performance Computing – HiPC 2007, Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, Vol. 4873/2007, pp. 197-208, Jan. 22, 2008.

[18] John A.S., Sam S.S., Wen-Mei W.H., "MCUDA: An Efficient Implementation of CUDA Kernels for Multi-core CPUs Source," *Lecture Notes In Computer Science, Springer-Verlag Berlin, Heidelberg*, pp. 16–30, 31 Jul. – 2 Aug. 2008, DOI=10.1007/978-3-540-89740-8_2.

[19] Sain-Zee U., Sara B., Melvin L., Wen-mei H., "CUDA-lite: Reducing GPU Programming Complexity," *LCPC 2008*.

[20] Ali A., Gregory M.S., "Scalable and highly parallel implementation of Smith-Waterman on graphics processing unit using CUDA," *Cluster Computing, Springer Netherlands*, 11 Jun. 2009, DOI=10.1007/s10586-009-0089-8.

[21] Riegel E. , Indinger T., Adams N.A., "Implementation of a Lattice–Boltzmann method for numerical fluid mechanics using the nVIDIA CUDA technology," *Computer Science - Research and Development*, Springer Berlin / Heidelberg, Vol. 23, Nos. 3-4, pp. 241-247, Jun. 2009. DOI=10.1007/s00450-009-0087-3.

[22] Antonio R., Manuel U., Lee C., Kun H., "Non-rigid Registration for Large Sets of Microscopic Images on Graphics Processors," *Journal of Signal Processing Systems*, *Springer New York*, Vol. 55, Nos. 1-3, pp. 229-250, Apr. 2009, DOI=10.1007/s11265-008-0208-4.

[23] Jonas T., "Implementation of a Lattice Boltzmann kernel using the Compute Unified Device Architecture developed by nVIDIA," *Computing and Visualization in Science*, *Springer Berlin / Heidelberg*, Jul. 24, 2008, DOI=10.1007/s00791-008-0120-2.

[24] Chris H., Karen H., Lister S.S., "GPU accelerated radio astronomy signal convolution," *Experimental Astronomy, Springer Netherlands*, Vol. 22, Nos. 1-2, pp. 129-141, Oct. 2008, DOI=10.1007/s10686-008-9114-9.

[25] Allan R., Jesper M., Thomas S.S., "Exploring Parallel Algorithms for Volumetric Mass-Spring-Damper Models in CUDA," *Lecture Notes in Computer Science, Springer Berlin / Heidelberg*, Vol. 5104/2008, pp. 49-58, 07 Jul. 2008, DOI=10.1007/978-3-540-70521-5_6.

[26] Eladio G., Sergio R., Maria A.T., Emilio L.Z., "Parallel Quantum Computer Simulation on the CUDA Architecture," *Lecture Notes in Computer Science, Springer Berlin / Heidelberg*, Vol. 5101/2008, pp. 700-709, 25 Jun. 2008, DOI=10.1007/978-3-540-69384-0_75.

[27] Eladio G., Sergio R., Maria A.T., Emilio L.Z., "Memory Locality Exploitation Strategies for FFT on the CUDA Architecture," *Lecture Notes in Computer Science, High Performance Computing for Computational Science - VECPAR 2008, Springer Berlin / Heidelberg*, Vol. 5336/2008, 16 Dec. 2008, DOI=10.1007/978-3-540-92859-1_39.

[28] Weiguo L., Bertil S., Gerrit V., Wolfgang M.W., "Molecular Dynamics Simulations on Commodity GPUs with CUDA," *Lecture Notes in Computer Science, High Performance Computing – HiPC 2007, Springer Berlin / Heidelberg*, pp. 185-196, 22 Jan. 2008, DOI=10.1007/978-3-540-77220-0_20.

[29] Mark H., "Many-core GPU computing with NVIDIA CUDA," *Proc. of the 22nd annual international conference on Supercomputing*, 2008.

[30] Tianyi D.H., Tarek S.A., "hiCUDA: a high-level directive-based language for GPU programming," *Proc. of 2nd Workshop on General Purpose Processing on Graphics Processing Units, ACM International Conference Proceeding Series*, Vol. 383, pp. 52-61, 2009.

[31] David B.T., Lee H., Wayne L., "A comparison of CPUs, GPUs, FPGAs, and massively parallel processor arrays for random number generation," *Proc. of the ACM/SIGDA international symposium on Field programmable gate arrays*, pp. 63-72, 2009.

[32] Shane R., Sara S.B. , Christopher I.R., Sam S.S., David B.K., Wen-mei W.H., "Optimization principles and application performance evaluation of a multithreaded GPU using CUDA," *Proc. 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming*, pp. 73-82, 2008.

[33] Seyong L., Seung-Jai M., Rudolf E., "OpenMP to GPGPU: a compiler framework for automatic translation and optimization," *Proc. 14th ACM SIGPLAN symposium on Principles and practice of parallel programming*, pp. 101-110, 2009.

[34] Shane R., Christopher I.R., Sam S.S., Sara S.B., Sain-Zee U., John A.S., Wen-mei W.H., "Program optimization space pruning for a multithreaded gpu," *Proc. 6th annual IEEE/ACM international symposium on Code generation and optimization*, pp. 195-204, 2008.

[35] Muthu M.B., Uday B., Sriram K., J.R., Atanas R., P.S., "A compiler framework for optimization of affine loop nests for gpgpus," *Proc. 22nd annual international conference on Supercomputing*, pp. 225-234, 2008.

[36] Christopher I.R., David J.H., John E.S., Klaus S., Wen-Mei W.H., "GPU acceleration of cutoff pair potentials for molecular modeling applications," *Proc. 5th conference on Computing frontiers*, pp. 273-282, 2008.

[37] John N., Ian B., Michael G., Kevin S., "Scalable Parallel Programming with CUDA," *Queue, Q focus: GPUs*, Vol. 6, Issue 2, pp. 40-53, Mar./Apr. 2008.

[38] Kayvon F., Mike H., "A closer look at GPUs," *Communications of the ACM,* Vol. 51, Issue 10, pp. 50-57, Oct. 2008.