3. VIRTUAL AND PHYSICAL MACHINES WITH MEGA-PROCESSOR CHIPS – THEORETICAL AND EXPERIMENTAL RESULTS

PÉTER SZOLGAY, Professor; ANDRÁS KISS, Assistant Professor; ZOLTÁN NAGY, Associate Professor, ISTVÁN REGULY, Assistant Professor
Graduate students: ENDRE LÁSZLÓ

Many core computers are the current approach to the solution of computationally intensive problems. At the present day, many core may mean 1000 but soon few 100000 or even more processing elements will be available on a desktop machine.

New ideas and new methods are required in algorithm development for these types of architectures. Using the cellular architecture of processors and memories new kind of parallel algorithms have to be developed. For the same Physical multi-cellular machine architecture different Virtual Machines can be defined, the latter is serving for algorithmic and software development. However, all virtual machines are to be uniquely mapped to a specific Physical Machine (a chip or chip set, etc.). Hence all computing element has a geometrical position on the Physical Machine and a geometric or topographic address on the Virtual Machine. In this new scenario of computer science and computer engineering computational complexity, computing power is a multi-parameter vector, any algorithm solving a problem will have a speed-power-area-bandwidth-accuracy metric. These parameters and this metric should be handled simultaneously and optimized.

Some new principles are summarized below.

First of all, we do not have efficient compilers though many examples are solved daily. In some of our problems we have used the CNN Universal Machine as an intermediate platform for compiling algorithms. Namely, we have developed an efficient CNN Universal Machine simulator on the FPGA or GPU and if an algorithm is described by as a cellular wave computing algorithm defined on a CNN Universal Machine, the compiling problem is solved.

We emphasize that due to the topographic nature of the Physical machine, an optimal *partitioning and placement is also needed* on the virtual machine, or on the intermediate "compiler prototype" (like the CNN Universal Machine). A Field Programmable Gate Array (FPGA) based framework has been described to accelerate simulation of a tough topographic problem, a nonlinear wave problem that is a complex physical spatio-temporal phenomenon, such as fluid dynamics (the hard test-problem since the invention of the digital computer). We have studied this problem since a while to develop a prototype virtual machine platform for them.

i) In the course of the design of an arithmetic unit which consists of locally controlled groups of floating point units both *partitioning and placement* aspects have to be considered. To solve this problem a framework has been elaborated in which the partitioning is based on an initial floor-plan of the vertices of the layered data-flow graph of the algorithm solving a given PDE or another problem. We have formulated this problem as a method for *implementing data-flow graphs on FPGAs*. The algorithm can minimize the number of cut nets and guarantee deadlock-free partitions.

ii) In case of topographic problems, *complicated geometries require unstructured spatial discretization, which results in irregular memory access patterns* severely limiting computing performance. *Data locality* is improved by mesh node renumbering technique, which results in predictable memory access pattern. Additionally, *storing a small window of node data in the on-chip memory* of the FPGA can increase data reuse and *decrease memory bandwidth requirements*.

iii) As one example, in numerical solution of fluid flows the generation of the floating-point data path and control structure of the arithmetic unit containing dozens of operators is a very challenging task when the goal is high operating frequency. Efficiency and use of the framework is described by a case study solving the Euler equations on an unstructured mesh using finite volume technique. On the currently available largest FPGA the generated architecture contains three processing elements working in parallel providing two orders on magnitude (90 times) speedup compared to a high performance microprocessor.

iv) Numerical algorithms can be implemented by using fixed –point or Floating point arithmetic with different precision. The selection of the *optimal data precision* to the different steps of an algorithm is *NP complete*. As an example, the solution of the advection equation was analyzed by using first and second order discretization methods. For this example if the resources are on an FPGA, and power dissipation or processing speed is prescribed, we have developed and algorithm for determining an optimal bit width.

v) A Fermi NVIDIA GPU based implementation of a CNN Universal Machine was also developed, and was analyzed and compared to a multicore, multithreaded CPU. The basic goal of this project is to get a clear picture about the capabilities and limits of GPUs.

vi) Key algorithms are the different Kalman filter implementations in trajectory estimation of the intruder airplane. To increase the speed of computation of unscented nonlinear Kalman filter was implemented on Xilinx Spartan 6 LX45 FPGA. The speed of this implementation is approximately 1000 iteration /s. It would be important if we want to follow multiple targets.

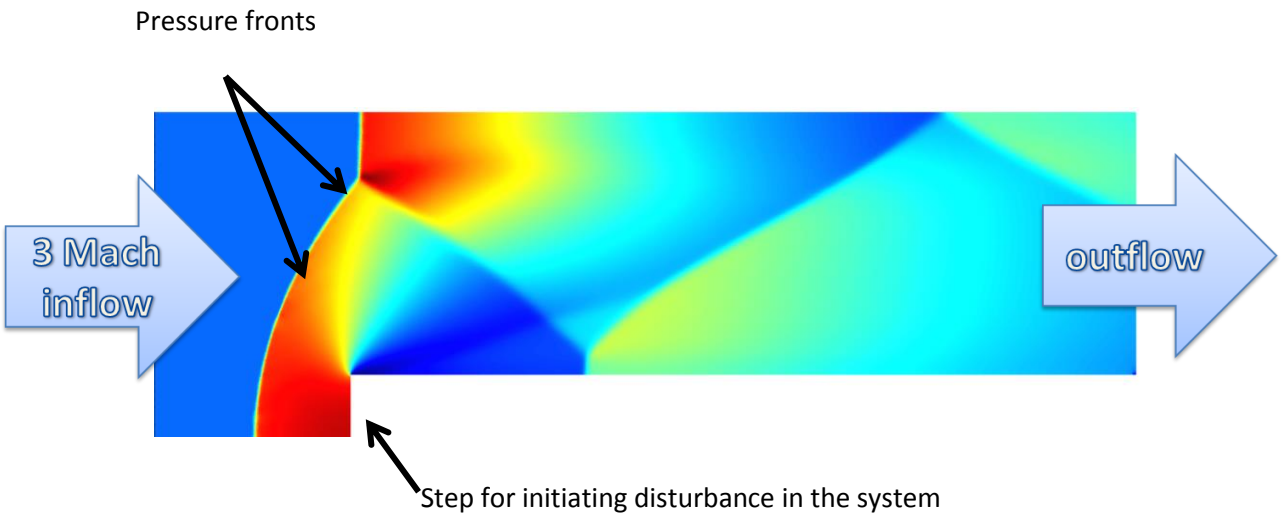

Fig. 1 *A two dimensional flow analysis was implemented on different array computing architectures*

PUBLICATIONS

[1] [Z. Nagy, P. Szolgay. Configurable Multi-Layer CNN-UM Emulator on FPGA, *IEEE Trans. on CAS I* Vol. 50, No. 6, pp. 774-778, 2003

[2] Z. Nagy, P. Szolgay. Solving Partial Differential Equations on Emulated Digital CNN-UM Architectures", *Functional Differential Equations,* Vol. 13*,* No. 1 pp. 61-87, 2006

[3] P. Kozma, P. Sonkoly and P. Szolgay. Seismic Wave Modeling on CNN-UM Architecture", *Functional Differential Equations, Vo1. 13,* No. 1, pp. 43-60, 2006

[4] Z. Nagy, Zs. Vörösházi,P. Szolgay. Emulated Digital CNN-UM Solution of Partial Differential Equations, *Int. J. CTA, Vol. 34, No. 4, pp. 445-470, 2006*

[5] Cs. Nemes, Z. Nagy and P. Szolgay. Automatic generation of locally controlled arithmetic unit via floorplan based partitioning, *Proc. of CNNA 2012*, Torino, 2012

[6] A Kiss, Z. Nagy, Á. Csík and P. Szolgay. Examining the accuracy and the precision of PDEs on FPGA, *Proc. of CNNA 2012*, Torino, 2012

[7] A. Hiba, Z. Nagy and M. Ruszinkó. Memory access optimization for computations on unstructured meshes, *Proc. of IEEE CNNA 2012*, Torino, 2012

[8] E. László, P. Szolgay, Z. Nagy. Analysis of a GPU based CNN Implementation, *Proc. of IEEE CNNA2012*, Torino, 2012

[9] Z. Nagy, Cs. Nemes, A. Hiba, A. Kiss, Á. Csik and P. Szolgay. Accelerating unstructured finite volume solution of 2-D Euler equations on FPGAs, *Proc. of 15th International Conference on Fluid Flow Technologies*, Budapest, 2012

[10] Cs. Nemes, Z. Nagy, A. Hiba, Á. Csík, , A. Kiss, M.Ruszinkó, P. Szolgay. Accelerating unstructured finite volume compuations on fieldprogrammabelgate arrays, *Concurrency and Computation: Practice and Experience, 2013. DOI:10.1002/cpe.3022*

[11] Cs. Nemes, G.Barcza, Z. Nagy, Ö.Legeza P. Szolgay. The density matrix renormalization group algorithm on kilo-processor architectures: implementation and trade-offs, *Computer Physiscs Communications, 2014,*

[12] E. László, Z.Nagy, M.B.Giles, I.Reguly J.Appleyard and P. Szolgay," Analysis of parallel processor architectures for the solution of the Balck-Scholes PDE, Proc. of IEEE ISCAS2015, Lisbon, 2015.

[13] A.Kiss, Z.Nagy, P.Szolgay, Gy.Csaba, X.S.Hu, W.Porod," Emulating massively parallel non-Boolean operators on FPGA", Proc. of IEEE ISCAS2015, Lisbon, 2015.