IL NUOVO CIMENTO Vol. 32 C, N. 2 Marzo-Aprile 2009
DOI 10.1393/ncc/i2009-10379-6

Colloquia: CSFI 2008

Multiphase lattice Boltzmann on the Cell Broadband Engine

F. Belletti([1]), L. Biferale([2]), F. Mantovani([1]), S. F. Schifano([3]), F. Toschi([4])([5]) and R. Tripiccione([1])

([1]) *Dipartimento di Fisica and INFN, Università di Ferrara - Ferrara, Italy*
([2]) *Dipartimento di Fisica and INFN, Università di Tor Vergata - Rome, Italy*
([3]) *Dipartimento di Matematica and INFN, Università di Ferrara - Ferrara, Italy*
([4]) *Istituto per le Applicazioni del Calcolo, CNR - I-00161 Rome, Italy*
INFN, Sezione di Ferrara - I-44100 Ferrara, Italy
([5]) *Department of Physics and Department of Mathematics and Computer Science Eindhoven University of Technology - 5600 MB Eindhoven, The Netherlands and International Collaboration for Turbulence Research*

(ricevuto il 15 Maggio 2009; pubblicato online l'1 Settembre 2009)

Summary. — Computational experiments are one of the most used and flexible investigation tools in fluid dynamics. The Lattice Boltzmann Equation is a well established computational method particularly promising for multi-phase flows at micro and macro scales. Here we present preliminary results on performances of the LBE method on the Cell Broadband Engine platform.

PACS 47.11.-j – Computational methods in fluid dynamics.
PACS 47.11.Qr – Lattice gas.
PACS 47.27.ek – Direct numerical simulations.

1. – Introduction

Numerical methods for multi-phase and multi-component fluid dynamics are a research frontier in computational sciences. Amongst many possible approaches, the Lattice Boltzmann Equation method (LBE) [1, 2] presents several advantages. Here we just recall easy coding (it is an explicit methods), the flexibility in handling complex boundary conditions, the locality of communications (particularly important when porting the code to massively parallel supercomputers) and the flexibility in implementing different physical properties, such as the emerging of diffusive interfaces between different phases. The LBE is a mesoscopic method which, while retaining key features of the Boltzmann equation, solves fluid equations with the advantage of better efficiency with respect to molecular techniques, *e.g.* Molecular Dynamics.

In this paper, we report results on the performance of a multi-phase LBE method ported on the Cell Broadband Engine (CBE), `http://www.research.ibm.com/cell/`.

The interest in this platform comes from its impressive top performance, already exploited by the current top entry in the Top500 list (`http://www.top500.org`) that has a performance in excess of 1 PFlops. Cheaper application-driven massively parallel systems also plan to use the CBE. For instance, the QPACE project [3, 4] plans to develop a system with $\mathcal{O}(2048)$ computing nodes—each based on a CBE processor and 4 Gbyte memory—optimized for Lattice QCD (LQCD) simulations. A radically different approach to CBE-enabled scientific computing is the BOINC project (`http://boinc.berkeley.edu.`), that builds on the widespread diffusion of the CBE, which is the processor used by the PlayStation 3 (PS3) gaming platform. BOINC develops PS3GRID, a client enabling collaborative computing based on a distributed Molecular Dynamics software (`http://www.ps3grid.net/`). The performance and accuracy of a standard single-phase LBE on the CBE processor have been studied in [5]. In this paper, we focus on multi-phase versions of the LBE method and develop a different implementation approach.

2. – Details of the algorithm

The LBE equations that we consider are the ones proposed by Shan and Chen [6] to describe the dynamics of multi-phase flows. We focus on this algorithm because, as compared to the standard single-phase LBE [5], it allows to investigate a broader set of physical applications; we also expect that the larger ratio between floating point operations and memory accesses should make it possible to obtain better sustained performance on the CBE.

As usual, the LBE method evolves the distribution functions $f_i(\boldsymbol{x}, t)$, corresponding to the different (9 in 2D and 19 in 3D) populations of the LBE model that we consider. Physical quantities (*i.e.*, the density and velocity of the fluid at each point in space and time) are computed from the distribution functions $f_i(\boldsymbol{x}, t)$ taking into account the lattice velocities $\boldsymbol{c}_i$ of each population (see [1, 2, 6] for more details): $\rho(\boldsymbol{x}, t) = \sum_{i=0}^{19} f_i(\boldsymbol{x}, t)$ and $\rho \cdot \boldsymbol{v} = \sum_{i=0}^{19} \boldsymbol{c}_i f_i(\boldsymbol{x}, t)$. In the Shan and Chen approach [6, 7], density is not constant in space and time but rather it evolves describing the evolution of the liquid-vapour mixtures. At each time step the populations are updated by means of a relaxation towards the equilibrium distribution f^{eq} (this step is the floating point intensive part of the algorithm), $f_i'(\boldsymbol{x}, t) = f_i(\boldsymbol{x}, t) - \frac{1}{\tau}(f_i - f_i^{eq})$ where the equilibrium distribution $f_i^{eq}(\boldsymbol{x}, t)$ is a quadratic expression in $\boldsymbol{v}$, see [1] for details. The updated values of the populations are then streamed (this step only involves memory to memory copies and no floating point computations) in the direction of their velocities: $f_i(\boldsymbol{x}+\boldsymbol{c}_i, t+1) = f_i'(\boldsymbol{x}, t)$.

3. – LBE on the Cell processor

The CBE is a heterogeneous multi-core processor that includes a PowerPC Processing Element (PPE) and up to eight so-called Synergistic Processing Elements (SPEs). The PPE runs the operating system, while the SPEs are intended as compute-intensive processing elements. The SPEs can only access a small (256 kbytes) private memory, known as the local store (LS). Data is exchanged between main memory and LS via asynchronous direct memory accesses (DMA).

3˙1. *Implementation details.* – We implemented in single precision the 2D and 3D version of the algorithms which have 9 and 19 populations, respectively (D2Q9 and D3Q19 in standard LBE notation). We tested the algorithm on an IBM QS22 blade system running at 3.2 GHz.

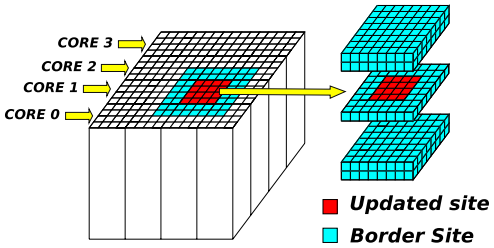

Fig. 1. – Data distribution on the SPEs. The domain is decomposed and assigned to different SPEs. Frames around computing volumes are loaded, in order to access neighbouring sites. Each SPE updates several columns of the lattice. Only a fraction of a column is stored in the LS; however, data is used in pipeline style so each lattice site is loaded onto the local store only once.

The algorithm for D3Q19 was implemented as follows (see also fig. 1). Each SPE loads from main memory to LS a sub-plane of the lattice (its size being limited by the LS size) loading also a frame around the volume boundaries extending for two lattice sites; in this way each SPE is able to perform both displace and collisions without further accesses to main memory. The whole lattice volume is divided in the several columns, and the sub-planes of each column are updated sequentially from the top to the bottom. At each update step, each SPE stores in its LS six different sub-planes: the sub-plane under update, four sub-planes making up the upper and lower boundaries, and one sub-plane used to implement data transfers from the LS to the main memory, in a double-buffering mode. Computing and data transfers are performed concurrently. The overhead of loading the frames is mitigated by a non negligible reuse of data in the LS, as each sub-plane is loaded only once but is used to update five adjacent sub-planes.

3˙2. *Performances*. – We now derive theoretical upper bounds on the expected performances and use these estimates to discuss our measured results. In our case, the performance obtainable from the CBE is limited by memory bandwidth. In order to update one grid site, we need to load (and store) in main memory $I = 4 \cdot 2 \cdot Q$ bytes (4 refers to the use of single-precision word size, 2 keeps into account load and store while Q is the number of LBE populations). For a D3Q19 LBE, updating each site implies $R \sim 500$ (about 100 for the D2Q9), so the peak obtainable (bandwidth limited) floating point performance is $R \cdot BW/I \simeq 75\,\text{Gflops}$, where $BW = 25.6\,\text{GB/s}$ is the memory bandwidth.

We summarize our performance results in table I. We measure some 20% of the expected peak performance. Analyses of our runs show that we lose about 50% of our running time synchronizing the SPEs. We are confident that we can improve substantially this step, so we expect to be able to reach $\simeq 40\%$ of peak performance. As shown in table I the performance of the 2D version is higher because memory access patterns are easier to implement. Furthermore, reasonably large 2D lattices (linear size up to ~ 1024 points) fully fit within the LS, so the above bounds do not apply; in this case extremely performing codes are possible. As remarked above, our codes are in single precision. The CBE perforance in double precision is one half that of single precision, so we reasonably expect that all our figures should be reduced by a factor two for a double-precision code.

TABLE I. – *Estimated and measured performances for LBE codes in 3D and 2D on the CBE (in single precision). See the text for details.*

Code		Performance (Gflops)
Processor peak		204.8
BW limited peak		$\simeq 75$
D3Q19	measured value	16.3
D3Q19	Improved synchronization	$\simeq 30$
D2Q9	arbitrary lattice size (measured)	27.0
D2Q9	small lattice size (measured)	105.0

4. – Conclusions

We ported a multi-phase Lattice Boltzmann method on the CBE processor. The goal of this exercise was to investigate the performances to be expected on massively parallel CBE based supercomputers, *e.g.* the QPACE machine [3]. Given the performances that we measured on the CBE and based on our previous experience in porting LBE methods on massively parallel systems like the APE series of machines [8], some estimates are readily made; using the full-size planned QPACE machine (2048 nodes) we may consider 3D lattices of linear size larger than 4096 running at approximately 60 Tflops (remember that inter-node communications are not a bottleneck for LBE on machines organized as a 3D grid of processors). This performance would imply a lattice update time in the order of 0.5 s.

* * *

We acknowledge useful discussions with M. BERNASCHI, G. DE FABRITIIS and S. SUCCI.

REFERENCES

[1] CHEN S. and DOOLEN G. D., *Annu. Rev. Fluid Mech.*, **30** (1998) 329.
[2] BENZI R., SUCCI S. and VERGASSOLA M., *Phys. Rep.*, **222** (1992) 145.
[3] GOLDRIAN G. *et al.*, *Comput. Sci. Eng.*, **10** (2008) 46.
[4] BAIER H. *et al.*, Status of the QPACE Project (2008).
[5] HARVEY M. J., DE FABRITIIS G. and GIUPPONI G., *Phys. Rev. E*, **78** (2008) 056702.
[6] SHAN X. W. and CHEN H. D., *Phys. Rev. E*, **47** (1993) 1815.
[7] SBRAGAGLIA M., BENZI R., BIFERALE L., SUCCI S., SUGIYAMA K. and TOSCHI F., *Phys. Rev. E*, **75** (2007) 026702.
[8] BARTOLONI A. *et al.*, *Int. J. Mod. Phys. C*, **4** (1993) 993.