Lattice-Gas Automata Fluids on Parallel Supercomputers

Jeffrey Yepez
US Air Force, Phillips Laboratory, Hanscom Field, Massachusetts
Guy P. Seeley
Radex Corporation, Bedford, Massachusetts
Norman H. Margolus
MIT Laboratory for Computer Science, Cambridge, Massachusetts

November 23, 1993

Abstract

A condensed history and theoretical development of lattice-gas automata in the Boltzmann limit is presented. This is provided as background to set up the context for understanding the implementation of the lattice-gas method on two parallel supercomputers: the MIT cellular automata machine CAM-8 and the Connection Machine CM-5. The macroscopic limit of two-dimensional fluids is tested by simulating the Rayleigh-Bénard convective instability, Kelvin-Helmholtz shear instability, and the Von Karman vortex shedding instability. Performance of the two machines in terms of both site update rate and maximum problem size are comparable. The CAM-8, being a low-cost desktop machine, demonstrates the potential of special-purpose digital hardware.

Contents

1 Introduction **3**

2 Some Historical Notes **5**

3 Why lattice-gases? **8**

4 Lattice-Gas Automata **9**
4.1 Some Preliminaries about the Local Dyanamics 10
4.2 Coarse-Grained Dynamics 11
4.3 Triangular Lattice: B=6 12
4.4 Single Particle Multispeed Fermi-Dirac Distribution Function . . 14
4.5 Hexagonal Lattice 17

5 The Cellular Automata Machine CAM-8 **19**

6 The Connection Machine CM-5 **20**

7 Gallery of Computational Results **23**
7.1 Rayleigh-Bénard Convection on the CAM-8 23
7.2 Kelvin-Helmholtz Instability on the CAM-8 24
7.3 Von Karman Streets on the CM-5 25

8 Discussion **25**

9 Acknowledgements **27**

List of Figures

1 Illustration on a hexagonal lattice of the two-step collision and streaming process required to complete a single time step: (a) initial configuration, (b) collision by permutation, and (c) streaming of particles. 38
2 (a) Lattice vector label convention; (b) Hexagonal lattice convention with lattice directions $a = 3$ up and $a = 6$ down. Coordinates above the lattice nodes are (i, j) memory array indices. 38
3 MIT Laboratory for Computer Science cellular automata machine CAM-8. This 8 module prototype can evolve a D-dimensional cellular space with 32 million sites where each site has 16 bits of data with a site update rate of 200 million per second. 39
4 (a) A single processing node, with DRAM site data flowing through an SRAM lookup table and back into DRAM. (b) Spatial array of CAM-8 nodes, with nearest-neighbor (mesh) interconnect (1 wire/bit-slice in each direction). . . 39
5 CM-5 Node: SPARCCPU, 32 Mbytes of memory and 4 Vector processing units. 40
6 Node Memory Layout. 40
7 Thermo 13-bit CAM-8 Experiment: Rayleigh-Bénard convection cells at the critical Rayleigh number. Lattice Size: 2048×1024. Time Average: 100. Spatial Average: 64×64. Mass Density Fraction=1/5. Data presented at 50,000 time steps. 41
8 Momentum and vorticity map of two-dimensional shear instability on the CAM-8. Lattice size of 4096×2048 with toroidal boundary conditions. Spacetime averaging over 128x128 blocks for 50 time steps. FHP collisions with spectators and a rest particle. Data presented at time steps 0, 10000, and 30000 with Galilean velocity shift. 42
9 FHP-II CM-5 Experiment: Von Karman Streets Lattice Size: 4096×2048. Time Average: None. Spatial Average: 128x128. Mass Density Fraction=1/7. Data presented at 32,000 time steps. 43
10 Performance runs on a 256-node CM-5 for an FHP hexagonal lattice embedded into a 3D mesh. Performance significantly suffers by communication overhead for small lattice sizes. 43

1 Introduction

Lattice-gas automata dynamics is a discrete form of molecular dynamics. In molecular dynamics one simulates a many-body system of particles with continuous interaction potentials [1]. The particles have continuous positions and momenta. In lattice-gas dynamics the particles' positions and momenta are discrete and motion is constrained to a spacetime crystallographic lattice.

One may also view lattice-gas automata dynamics as an extension of cellular automata, popularized in the physics community by Stephen Wolfram [2, 3]. An elementary treatment of the cellular automata subject is presented by Tommaso Toffoli and Norman Margolus in their book on cellular automata machines [4]. Corresponding to the cellular automata paradigm, lattice-gas automata are ideally suited for massively parallel processing. In lattice-gas automata models, each possible momentum state at a given position is represented by a single digital bit. Therefore, a Pauli exclusion principle is enforced where there can be no more than a single particle per momentum state. As a particle in state α at some lattice site of the crystallographic space "hops" into state β, say at a neighboring site, a digital bit is moved from α and into β. So in lattice-gas dynamics one simulates a system of Boolean particles where the data streaming corresponds to spatial translation and the data permutation corresponds to collisional interactions. This computational analog of particle dynamics offers an exciting alternative to, and not simply an approximation of, the usual partial differential equations[5]. So the lattice-gas methodology has an intrinsic value beyond finite difference schemes.

The lattice gas approach has been extended to a lattice-based Krook-Bhatnager-Gross approximation of the Boltzmann equation [6, 7, 8, 9], which we call the lattice-Boltzmann equation. In place of the exactly computable dynamics of Boolean particles, one focuses on a statistical regime where a particle has a probability of occupying a given momentum state. Moreover, many particles can occupy the same momentum state at the same position. The approach offers both theoretical and computational advantages. An important theoretical

advantage is that one may capture the essential physics of the complex system by stating no more than the system's equilibrium distribution. Computationally it has the advantage of reducing noisy fluctuations in the system at the expense of discarding information concerning particle-particle correlations. Lattice-gas automata in the Boltzmann approximation have become one of the most important contributions by the lattice-gas community to high-performance computational physics [10]. The lattice-Boltzmann equation, usually implemented within one hundred lines of code on a massively parallel processor [1], allows the researcher to efficiently model complex systems, providing a straightforward particle-based metaphor to computation. Yet it relies on expensive floating point calculations and therefore is most suited for massively parallel machines such as the Connection Machine-5 (CM-5). In this paper, we use the lattice-Boltzmann equation only for theoretical analysis—all of our simulations are based on lattice gases.

We have implemented a general lattice-gas automaton on two parallel architectures: the experimental MIT CAM-8 and the Thinking Machines Corporation CM-5. Very briefly, a CAM-8 node has 2^{22} 16-bit sites (8 Mbytes DRAM) and a double buffered look-up table (2 Mbits SRAM), with a clock speed of 25 MHz. There are no processors, only lookup tables. The nodes are connected in a 3D mesh. A CM-5 node has 32 Mbytes of DRAM, four vector units at 16 MHz, and a SPARC processor at 32 MHz. A fat-tree network connects the nodes. The architectures of both machines will be discussed in more detail below.

The presentation here is limited to two-dimensional fluids. Three-dimensional hydrodynamics, immiscible fluids [11, 12, 13, 14], multiphase systems [15, 16, 17, 18], reaction-diffusion systems, magnetohydrodynamics [19], and flow through porous media [20, 21] are also subjects of active research. We have implemented hydrodynamic and thermohydrodynamic lattice-gases on the CAM-8, the first lattice-gas experiment conducted on the prototype machine. Results of the CAM-8 and CM-5 simulations of the Rayleigh-Bénard convection instability, Kelvin-Helmholtz shear instability, and Von-Karman vortex shedding instability are presented.

Our main findings are the following. The CAM-8 delivers 25 million site

updates per second per module for a 16 bit lattice gas. The CM-5 in principle can simulate larger lattices due to its much larger memory size (16Gbytes on a 512-node partition). In practice the typical problem sizes were almost identical on a 128-node partition of the CM-5 and on an 8-module CAM-8 prototype: 4096×2048 2D lattices on both machines. The CM-5 can deliver about 1 million site updates per second per node for a 16 bit lattice gas. For the CM-5 interprocessor communication cost is small relative to the computation involved in the data streaming and site update when the size of the lattice is large enough. This suggests that the relatively low communication bandwidth imposes no serious degradation on the delivered performance. We also find that the delivered performance increases as the lattice size increases, see appendix 9.

The best way to report performance of each machine for lattice-gas simulation is to give the site update rate. The best rate achieved on the thermohydrodynamic lattice-gas was 191 million site updates per second on an 8-module CAM-8 and 110 million on on a 256-node partition of the CM-5. For a simpler problem, the FHP lattice-gas, a 256-node partition of the CM-5 attained a site update rate of 550 million updates per second on a 32K by 2K lattice. The best rate obtainable on the 8-module CAM-8 for the same simple problem is 382 million site updates per second. The CAM-8 prototype is a desktop machine (see figure 3) containing approximately the same amount of digital logic and memory as is found in a common workstation. The efficiency of the CAM-8 comes at a severe price, in terms of its specialization to a certain class of scientific computations, albeit its applicability is widening [22].

2 Some Historical Notes

Let us briefly review some of the historical developments of the lattice-gas subject. An overview of the lattice-gas subject has also been given by Boghosian [23]. Simple implementations of a discrete molecular dynamics on a square lattice were investigated in the early 1970's by the French, in particular Yves Pomeau and coworkers [24]. By the late 1970's, cellular automata research was

underway at the Information Mechanics Group at MIT on reversible computation by Edward Fredkin, Tommaso Toffoli, and Norman Margolus[25, 26, 27]. The idea of building special-purpose machines to simulate physics-like models on a fine-grained space [28, 27] originated there and today still remains a strength of that group. A good review of the kind of cellular automata modeling done in the early 1980's is given by Gérard Vichniac [29]. During this time, Stephen Wolfram visited the Information Mechanics Group and was stimulated by their work. In 1983 Wolfram popularized cellular automata as a simple mathematical model to investigate self-organization in statistical mechanics[2, 30].

After visiting the MIT Information Mechanics Group in 1983 and seeing a TM-gas simulation on the CAM-5 machine of Toffoli and Margolus[5, 28], Pomeau realized the potential for simulating large fluid systems and much new interest and activity in the field emerged. A race began to theoretically prove that a hydrodynamic limit emerges from simple lattice-gas automata. The intense interest was not stirred as much by the subject of hydrodynamics itself, but instead by the possibility of a simple cellular space-time model capturing such complex natural behavior in an exact way. In 1985 Wolfram completed the first hydrodynamics simulations on a triangular lattice [31] on the Connection Machine—at this time, lattice-gases were one of the most important applications for the bit oriented single instruction multiple data Connection Machine. By 1986 Frisch, Hasslacher, and Pomeau had reported the existence of an isotropic two-dimensional lattice-gas on the triangular lattice [32]. In the same year Wolfram completed the most detailed treatment of the basic theory including novel symmetry considerations and introduced the Boltzmann approximation. Frisch *et al.* found the minimal lattice symmetry needed to recover isotropic flow in the continuum limit is a triangular lattice with a particle possessing six momentum states. Their model is now referred to as the FHP-model or hexagonal lattice-gas model. Accompanying the seminal 1986 FHP paper was a paper by Margolus, Toffoli, and Vichniac on cellular-automata supercomputers for fluid-dynamics modeling [33]. The contribution of Margolus *et al.* complements the theoretical work of Frisch *et al.*, pointing out that with dedicated computa-

tional hardware the lattice-gas model potentially gains a unique advantage over traditional methods of physical modeling.

By 1987 the lattice-gas methodology was extended to model three-dimensional flows. The minimal lattice found by Frisch et el. [34] was the face centered hypercubic (fchc) lattice. The fchc lattice with 24-nearest neighbors is projected onto three dimensions in a simple fashion by limiting the depth of the fourth dimension of the simulation volume to one lattice link. Research is still underway on finding optimal collisions to minimize the viscosity of the fluid [35], however this task has proven very difficult. The reason for this difficulty is that the fchc lattice-gas has 2^{24} or 16.7 million input configurations. In practice, all possible collisions are not included in a simulation because of the large demand for local memory needed to pre-store all the necessary collisional events in table look-up format—an efficient format for implementing complex interactions. To ease memory loads, lattice isometries are exploited to reduce the size of look-up tables [36].

The hope of modeling very high Reynolds number flows by lattice-gas automata methods has not yet been realized with models that do not violate semi-detailed balance. However, lattice-gas models in the Boltzmann approximation have shown considerably more success in achieving high Reynolds number flows[2].

Recently the first prototype of the next generation cellular automata machine, CAM-8, has been constructed [37]. The current 8-module CAM-8 prototype, with a site density of 2^{22} 16-bit sites per module, has a total of 32 million sites. Within the next few years a large CAM-8 sponsored by the US Air Force will be constructed with at least 10^9 sites and will have a computational rate of approximately 12.5 billion site updates per second. This site update rate is about two orders of magnitude faster than that achievable with current parallel computers such as the Connection Machine, CM-5. In §7 we present some simulation results on the CAM-8 prototype and the CM-5.

3 Why lattice-gases?

There are many reasons for studying lattice-gases, both practical and theoretical. Some commonly cited reasons are oriented towards computer science and issues related to massively parallel processing. There are also appealing reasons related to modeling physical systems with complex boundary conditions. The lattice-gas' attributes include: 1) bit efficiency; 2) inherent simplicity; 3)logic density; and 4) exact computability.

Firstly, lattice-gas automata allow for high bit efficiency. A single digital bit is used to represent a particle. Unlike in floating point calculations where there exist uncontrolled round-off errors in the least significant bits, in lattice-gases all bits have equal weight, or to quote Frisch, there is "bit democracy." Consequently, the efficiency with which bits are used may be higher for lattice-gases.

Secondly, lattice-gas automata possess an inherent simplicity. Just as simple models in statistical mechanics, such as the Ising model, shed light on equilibrium critical phenomena, so too do lattice-gas models shed light on dynamical phenomena[18]. Moreover, their inherent simplicity gives them pedagogical value since many properties of macroscopic systems can be understood through analytical expressions given very simple local rules. For example, lattice-gases are a simple way to understand details of fluid systems such as the dependence of the shear viscosity on particle collision rates. Computational fluid dynamics codes are complicated and intricate in their approximations. Lattice-gases are perhaps the simplest expression of Navier-Stokes flows and are easily implemented.

Thirdly, the combination of bit efficiency with the simplicity and locality of some lattice-gas rules allows—in principle—nearly ideal logic density. At the highest logic density that is physically possible, there is the interesting prospect of lattice-gas architectures built out of "quantum hardware." There is the expectation that in the future, computation will be achieved on quantum computers [38, 39, 40, 41]. As the fundamental computational element's size

reduces to nano-scale ranges its behavior is governed by quantum mechanics. Quantum mechanics requires unitary, and hence invertible, time evolution—the microscopic reversibility of the lattice-gas dynamics is important here. Even before quantum mechanics becomes a constraint, the reversibility of lattice gas dynamics may become a significant benefit, since at very high logic densities the dissipation of heat caused by irreversible computations will become an issue[42, 43].

Fourthly, lattice-gas automata are exactly computable. Richard Feynman [44] considered on a discrete spacetime lattice "the possibility that there is to be an *exact* simulation, that the computer will do *exactly* the same as nature", and that using computers in an exactly computable way may lead to new possibilities in our understanding of physics. Although lattice-gases cannot model quantum systems [44], they do model classical systems while keeping mass, momentum, and energy exactly conserved. Exact modeling is valuable, for example, in cases where multiparticle correlations are essential to the system's behavior. Lattice-gas simulations can verify theoretical predictions beyond the Boltzmann mean-field approximation of uncorrelated collisions: the phenomenon of long-time tails in the velocity autocorrelation function [45, 46, 47] has recently been observed in lattice gases [48, 49, 50].

4 Lattice-Gas Automata

We first define, in the usual way, what a lattice-gas cellular automaton is. Then we analytically treat the lattice-gas in the Boltzmann limit to show that one may use strictly deterministic local rules to obtain the correct macroscopic limit. We show in particular that a chiral system is adequate to obtain correct hydrodynamics. We then summarize the derivation of the equations of motion for a thermal lattice-gas. Finally, we discuss our conventions for embedding a hexagonal lattice in a square lattice. We show the streaming relations used here for local and nonlocal collisions.

4.1 Some Preliminaries about the Local Dyanamics

Variables used are the following

$$
\begin{aligned}
\textit{Mass Unit} &: m \\
\textit{Spatial Unit} &: l \\
\textit{Temporal Unit} &: \tau \\
\textit{Particle Speed} &: c = \frac{l}{\tau} \\
\textit{Sound Speed} &: c_s \\
\#\ \textit{Momentum Directions} &: B \\
\textit{Lattice Vectors} &: \hat{\mathbf{e}}_a \\
a &= 1, 2, \ldots, B \\
\textit{Particle Number Variable} &: n_a \\
\textit{Distribution Function} &: f_a \\
\textit{Collision Operator} &: \Omega_a \\
\textit{Jacobian Matrix} &: J_{ab} \\
\textit{Number Density} &: n \\
\textit{Mass Density} &: \rho = mn \\
\textit{Bulk Velocity} &: \mathbf{v} \\
\textit{Total Internal Energy} &: \varepsilon
\end{aligned}
$$

Consider a spacetime lattice with N spatial sites, unit cell size l, and time unit τ. Particles, with mass m, propagate on the lattice with speed $c = l/\tau$. The lattice vectors are denoted by $\hat{\mathbf{e}}_a$ where $a = 1, 2, \ldots, B$. A particle's state is completely specified at some time, t, by specifying its position on the lattice, $\mathbf{x}$, and its momentum, $\mathbf{p} = mc\hat{\mathbf{e}}_\mathbf{a}$. The particles obey Pauli exclusion since only one particle can occupy a single momentum state at a time. The total number of configurations per site is 2^B. The total number of single particle states available in the system is $N_{\text{total}} = BN$. With P particles in the system, we denote the filling fraction by $d = P/N_{\text{total}}$.

The number variable, denoted by $n_a(\mathbf{x}, t)$, has the value one if a particle exists at site $\mathbf{x}$ at time t in momentum state $mc\hat{\mathbf{e}}_\mathbf{a}$ and zero otherwise. The evolution of the lattice-gas can be written in terms of n_a as a two-part collision and streaming process. The collision part permutes the particles locally at each site

$$n'_a(\mathbf{x}, t) = n_a(\mathbf{x}, t) + \Omega_a(\vec{n}(\mathbf{x}, t)), \tag{1}$$

where Ω_a represents the collision operator and in general depends on all the particles at the site. The streaming part permutes the particles globally. A particle at position $\mathbf{x}$ "hops" to its neighboring site at $\mathbf{x} + l\hat{\mathbf{e}}_\mathbf{a}$ and then time is incremented by τ

$$n'_a(\mathbf{x} + l\hat{\mathbf{e}}_\mathbf{a}, t + \tau) = n_a(\mathbf{x}, t) + \Omega_a(\vec{n}(\mathbf{x}, t)). \tag{2}$$

Equation (2) is the lattice-gas cellular automaton equation of motion. Because the dynamics only permutes the occupation of states, the system is strictly reversible, see figure 1.

4.2 Coarse-Grained Dynamics

To simplify the theoretical analysis of the lattice-gas dynamics, it is convenient to work in the Boltzmann limit where a field point is obtained by a block average over the number variables. That is, we may define a single particle distribution function, $f_a = \langle n_a \rangle$.It should be understood that whenever the single particle distribution function is written, its subscripted index is taken modulo B

$$f_{a+b} = f_{\bmod B(a+b)}. \tag{3}$$

Using the Boltzmann molecular chaos assumption the averaged collision operator simplifies to $\langle \Omega_a \rangle = \Omega_a(\langle \vec{n} \rangle)$, and by a Taylor expansion (2) we obtain the lattice Boltzmann equation

$$\partial_t f_a + c e_{ai} \partial_i f_a = \Omega_a. \tag{4}$$

A careful treatment of this procedure is given by Frisch *et al.* [34]. A general collision operator is constructed as follows

$$\Omega_a = \sum_{\{\zeta_i\}} \alpha Q_a(\{\zeta_i\}), \tag{5}$$

where $\{\zeta_i\}$ is a set of occupied particle states and $\alpha = \pm 1$ is a scalar coefficient and where each term in the sum is written in factorized form as

$$Q_a(i_1, \ldots, i_k) = \frac{f_{a+i_1}}{1 - f_{a+i_1}} \cdots \frac{f_{a+i_k}}{1 - f_{a+i_k}} \prod_{j=1}^{B} (1 - f_{a+j}). \tag{6}$$

We expand the distribution function about its equilibrium value, $f^{\rm eq}$

$$f_a = f^{\rm eq} + \delta f_a \tag{7}$$

so that, to first order, we have

$$\Omega_a(f^{\rm eq}) = \sum_b \frac{\partial \Omega_a}{\partial f_b} \delta f_b. \tag{8}$$

The l.h.s. of (8) must vanish, since the particle distribution is non-changing under equilibrium conditions. The eigenvalues of the Jacobian of the collision operator,

$$J_{ab} = \frac{\partial \Omega_a}{\partial f_b}, \tag{9}$$

can be calculated and the number of these that vanish must equal the number of invariant quantities in the lattice-gas dynamics. Because of the finite-point group symmetry of the spatial lattice, the Jacobian matrix will be circulant, its elements can be specified by the difference of their indices, $J_{ab} = J_{a-b}$. This property of the Jacobian simplifies the solution of the eigenvalue equation

$$\sum_b J_{a-b} \xi_b^k = \lambda^k \xi_a^k, \tag{10}$$

where $k = 1, \ldots, B$. Let us make the ansatz that the eigenvectors have the following form

$$\xi_a^k = e^{2\pi i a k/B}. \tag{11}$$

Then inserting (11) into (10) and taking $m = a - b$, gives

$$\lambda^k = \sum_m J_m e^{2\pi i m k/B}. \tag{12}$$

4.3 Triangular Lattice: B=6

When implementing a lattice-gas on a parallel computer it is most convenient to use deterministic updating rules. This is important for several reasons. First of all, using deterministic rules, the lattice-gas possesses a strict time-reversal invariance. Therefore, it mimics the time-reversal invariance of natural physical laws. As a practical matter, the reversibility allows one to run the gas dynamics

forward to some state and then back to its initial state. This is a good way to check if the local rules are coded correctly. Second of all, the generation of random numbers typically takes times and using random bits increases the amount of state that the rule must deal with. Therefore, deterministic local rules are used here. Lattice-gas collisions can be catergorized with even or odd chirality. The method employed here uses even chirality collisions on even time steps and odd chirality collisions on odd time steps, thereby eliminating the need for a random coin toss. The validity of such a partitioning of the collisions must be justified. This may be done in a straight forward way using the results of §4.2. The simplest hydrodynamic example is a definite chirality hexagonal lattice-gas, almost identical to the usual FHP gas except no random coin toss is made. For a hexagonal lattice, $B = 6$, the eigenvectors of the Jacobian matrix, (11), are simply composed of 1 plus the three roots of -1

$$\xi_0 = (1,1,1,1,1,1) \tag{13}$$

$$\xi_1 = (\epsilon, \epsilon^*, -1, \epsilon, \epsilon^*, 1) \tag{14}$$

$$\xi_2 = (\epsilon^*, \epsilon, 1, \epsilon^*, \epsilon, 1) \tag{15}$$

$$\xi_3 = (-1, 1, -1, 1, -1, 1) \tag{16}$$

$$\xi_4 = (\epsilon, \epsilon^*, 1, \epsilon, \epsilon^*, 1) \tag{17}$$

$$\xi_5 = (\epsilon^*, \epsilon, -1, \epsilon^*, \epsilon, 1), \tag{18}$$

where $\epsilon = e^{i\frac{\pi}{3}}$. The collision operator that produces 2-body and 3-body symmetric collision is the following

$$\Omega_a = Q_a(1,4) - Q_a(0,3) + Q_a(1,3,5) - Q_a(0,2,4), \tag{19}$$

which is written in expanded form as

$$\begin{aligned}
\Omega_0 = & -\left(f_0\ (1-f_1)\ (1-f_2)\ f_3\ (1-f_4)\ (1-f_5)\right) + \\
& (1-f_0)\ f_1\ (1-f_2)\ (1-f_3)\ f_4\ (1-f_5) - \\
& f_0\ (1-f_1)\ f_2\ (1-f_3)\ f_4\ (1-f_5) + \\
& (1-f_0)\ f_1\ (1-f_2)\ f_3\ (1-f_4)\ f_5
\end{aligned}$$

Using (9) the Jacobian may be calculated[3]

$$\begin{aligned} J &= \mathrm{circ}[(1-f)^2 f, -(1-f)^2 f, (-1+f)^2 f^2, \\ &\quad (1-2f)\,(1-f)^2 f, (1-f)^2 f\,(-1+2f)\,, \\ &\quad -(1-f)^2 f^2] \end{aligned}$$

Using (12), the eigenvalues of J may be directly calculated

$$\begin{aligned} \lambda_0 &= 0 \\ \lambda_1 &= 0 \\ \lambda_2 &= 2\,\epsilon\,(1+\epsilon)\,(-1+f)^3 f \\ \lambda_3 &= -6\,(1-f)^2 f^2 \\ \lambda_4 &= 2\,\epsilon\,(1+\epsilon)\,(1-f)^3 f \\ \lambda_5 &= 0 \end{aligned}$$

There are only three zero eigenvalues, so the deterministic FHP-type lattice-gas model possesses only three invariants: the total mass and the two components of momentum. The methodology of successively switching between left and right-handed collision tables is therefore justified, at least in the Boltzmann limit. Switching between left and right-handed collision tables is done on the CAM-8 since there is no additional time or memory cost (per module) incurred in using multiple tables. However, since storing multiple lookup tables costs additional memory on the CM-5, in practice we use only a single collision table, and therefore, our CM-5 simulations use a chiral lattice-gas. In fact, the Von Karman street simulation in §7 presented in figure 9 is an example of a chiral lattice-gas.

4.4 Single Particle Multispeed Fermi-Dirac Distribution Function

It is essential to verify that in the macroscopic limit, the cellular automaton equation of motion (2) leads to Navier-Stokes hydrodynamics. To verify this, we begin with the most general form of the single particle distribution function, appropriate for even multispeed lattice-gases: the Fermi-Dirac distribution.

Fundamentally, this arises because the individual digital bits used to represent particles satisfy a Pauli-exclusion principle. Therefore, the distribution must be written as a function of the sum of scalar collison invariants, $\alpha + \beta e^{\sigma}_{ai} v_i + \gamma \varepsilon_\sigma$, implying the following form

$$f^{\sigma}_{a} = \frac{1}{1 + e^{\alpha + \beta e^{\sigma}_{ai} v_i + \gamma \varepsilon_\sigma}} \tag{20}$$

Using the identities in the appendix, an expansion to fourth order of (20) about zero velocity is the following

$$\begin{aligned} f^{\sigma}_{a} \quad = \quad & d_\sigma \\ & -d_\sigma(1-d_\sigma)\beta_1 e^{\sigma}_{ai}\frac{v_i}{c} \\ & -\frac{1}{2}d_\sigma(1-d_\sigma)(\alpha_2 + \gamma_2\varepsilon_\sigma)\frac{v^2}{c^2} \\ & +\frac{1}{2}d_\sigma(1-d_\sigma)(1-2d_\sigma)\beta_1^2 e^{\sigma}_{ai} e^{\sigma}_{aj}\frac{v_i v_j}{c^2} \\ & -\frac{1}{2}d_\sigma(1-d_\sigma)\beta_3 e^{\sigma}_{ai}\frac{v_i v^2}{c^3} \\ & +\frac{1}{2}d_\sigma(1-d_\sigma)(1-2d_\sigma)\beta_1(\alpha_2 + \gamma_2\varepsilon_\sigma) e^{\sigma}_{ai}\frac{v_i v^2}{c^3} \\ & -\frac{1}{6}d_\sigma(1-d_\sigma)(1-6d_\sigma + 6d_\sigma^2)\beta_1^3 e^{\sigma}_{ai} e^{\sigma}_{aj} e^{\sigma}_{ak}\frac{v_i v_j v_k}{c^3} \\ & +O(v^4) \end{aligned}$$

where $d_\sigma = f^{\sigma}_{a} \mid_{v=0}$. The coarse-grain averaged dynamics depend on the following dynamical variables.

Particle number density:

$$m \sum_{a,\sigma} f^{\sigma}_{a} = \rho, \tag{21}$$

Momentum density:

$$mc \sum_{a,\sigma} e^{\sigma}_{ai} f^{\sigma}_{a} = \rho v_i, \tag{22}$$

Moment density flux tensor:

$$mc^2 \sum_{a,\sigma} e^{\sigma}_{ai} e^{\sigma}_{aj} f^{\sigma}_{a} = \Pi_{ij}. \tag{23}$$

Total energy density, half the trace of the momentum flux tensor:

$$n\varepsilon = \frac{1}{2}\hat{\Pi}_{ii}. \tag{24}$$

Pressure tensor, $\hat{P}$:

$$\hat{P}_{ij} = m \sum_{a,\sigma} f_a^\sigma \left(ce_{ai}^\sigma - v_i\right)\left(ce_{aj}^\sigma - v_j\right). \tag{25}$$

Heat flux, $\mathbf{q}$:

$$q_i = m \sum_{a,\sigma} f_a^\sigma \left(ce_{ai}^\sigma - v_i\right)^2 \left(ce_{aj}^\sigma - v_j\right). \tag{26}$$

In equilibrium, the cellular automaton dynamical equation (4) reduces to

$$\partial_t f_a^\sigma + ce_{ai}^\sigma \partial_i f_a^\sigma = 0. \tag{27}$$

(27) implies 3 conservation equations. To obtain these equations, the identities for isotropic lattice vectors given in the appendix are necessary. Using (21) and (22) in (27) gives continuity (mass conservation):

$$\partial_t \rho + \partial_i(\rho v_i) = 0. \tag{28}$$

Using (22) and (23) in (27) gives the Navier-Stokes equation (momentum conservation):

$$\partial_t(\rho v_i) + \partial_j(\rho g v_i v_j) = -\partial_i p + \eta \partial^2 v_i. \tag{29}$$

Using (25) and (26) in (27) gives the heat equation (energy conservation):

$$\partial_t(n\varepsilon) + \partial_i(n\varepsilon v_i) + \frac{1}{2}\partial_i q_i + \partial_j(v_i P_{ij}) = 0, \tag{30}$$

where

$$p = (\gamma - 1)(n\varepsilon - \frac{1}{2}\rho g v^2) \tag{31}$$

and $\gamma = \frac{C_p}{C_v}$ or $\gamma - 1 = \frac{2}{D}$. Although the lattice-gas may in principle be comprised of an indefinite number of speeds, from (31) we see that the pressure depends upon the square of the bulk velocity, *i.e.* it is the difference of the total internal energy of the lattice-gas minus the bulk kinetic energy. In a single speed lattice-gas, this kind of velocity dependence is anomalous and is a well known deficiency of the lattice-gas. However, for a multispeed lattice-gas the existence of this term takes on a physical interpretation. For a classical ideal gas, the pressure is proportional to both the sound speed squared and the temperature

$$p = \rho c_s^2 = n k_B T. \tag{32}$$

Since $\rho = mn$, equating (31) with the ideal gas law (32) gives the total internal energy in terms of the bulk kinetic energy $\frac{1}{2}mv^2$ and the local particle thermal energy $k_B T$

$$\varepsilon = \frac{g}{2}mv^2 + \frac{k_B T}{\gamma - 1}. \tag{33}$$

The Navier-Stokes equation (29), the pressure (31), and the total energy (33) all explicitly have a factor g in them. In the Boltzmann limit, the factor g depends on the particle speeds, c_σ, and on the density distribution per speed, d_σ, by the following complicated expression

$$g = \frac{\sum_\sigma d_\sigma \sum_\sigma B_\sigma (\frac{c_\sigma}{c})^4 d_\sigma (1 - d_\sigma)(1 - 2d_\sigma)}{\left[\sum_\sigma B_\sigma (\frac{c_\sigma}{c})^2 d_\sigma (1 - d_\sigma)\right]^2}. \tag{34}$$

If $g = 1$, then a multispeed lattice-gas automata would exactly solve the ideal fluid equations where the physical interpretation of the total internal energy would then be that it partitions into a bulk motion term, or kinetic energy, and a fluctuating motion term, or random heat energy associated with a certain gas temperature. Therefore, the multispeed lattice-gas calculation, in the Boltzmann limit, would exactly agrees with classical kinetic gas theory, see the expression for the partial pressure of an electron gas given by Li and Wu [51]. The factor g approachs one only as the number of speed in the lattice-gas model becomes large. However, for a small number of speeds, a rescaling of the variables recovers the exact dynamics. A similar observation has been made by Teixeira [52] in his investigation of multispeed lattice-gas automata models.

4.5 Hexagonal Lattice

In a hexagonal lattice there are six lattice vectors which we enumerate by the following convention

$$\hat{\mathbf{e}}_{\mathbf{a}} = -\left(\sin\frac{\pi a}{3}, \cos\frac{\pi a}{3}\right), \tag{35}$$

where $a = 1, 2, \ldots, 6$. The spatial coordinates of the lattice sites may be expressed as follows

$$\mathbf{x}_{ij} = \left(\frac{\sqrt{3}}{2}j, i - \frac{1}{2}(j \bmod 2)\right) \tag{36}$$

where i and j are rectilinear indices which specify the data memory array location used to store the lattice-gas site data. For a multispeed lattice-gas it is necessary to shift data more than one lattice length. Let $s = (j \bmod 2)(r \bmod 2)$. Given a particle at site (i, j), it may be shifted to a site r lattice units away to a remote site (i', j') by the following mapping

$$(i', j')_1 = \left(i + \frac{r+1}{2} - s, j - r\right) \tag{37}$$

$$(i', j')_2 = \left(i - \frac{r}{2} - s, j - r\right) \tag{38}$$

$$(i', j')_3 = (i - r, j) \tag{39}$$

$$(i', j')_4 = \left(i + \frac{r+1}{2} - s, j + r\right) \tag{40}$$

$$(i', j')_5 = \left(i - \frac{r}{2} - s, j + r\right) \tag{41}$$

$$(i', j')_6 = (i + r, j) \tag{42}$$

where $(i', j')_a$ denotes the shifted site, *i.e.* $(i, j) \rightarrow (i', j')$ with a shift along vector $\vec{\mathbf{r}} = r\hat{\mathbf{e}}_a$ and where division by 2 is considered integer division.

These streaming relations are useful for implementing a lattice gas in a structured language such as the C-language. Our implementation on the CM-5 in the C-language and DPEAC use these relations for all address computations. In these streaming relations, the modulus operator is base 2 because a two-dimensional hexagonal lattice embedded into a square three-dimensional mesh is pleated.

The simplest way to see this embedding is to define $z = (j \bmod 2)$. Therefore the third dimension along the z-axis is narrow, only one lattice distance wide. Half of the lattice sites are at $z = 0$ and the other half are at $z = 1$. This divides the hexagonal lattice into two sublattices that we refer to as *pleat 0* and *pleat 1*. Table 1 lists the components of the data translation vectors, or kicks, for each stream direction, $a = 1, 2, \ldots, 6$, for both pleats. This kick table was used for the CAMForth implementation on the CAM-8 and the C* implementation on the CM-5. This is equivalent our general streaming relations for the case when $r = 1$. The usefulness of this kind of embedding is that if the data for each one of the sublattices is rendered for display, it can be drawn in simple raster

form and fluid structures will appear correctly, *i.e.* a sound pulse will appear circular.

5 The Cellular Automata Machine CAM-8

The cellular automata machine CAM-8 architecture devised by Norman Margolus of the MIT Laboratory for Computer Science [37, 22] is the latest in a line of cellular automata machines developed by the Information Mechanics Group at MIT [28, 4, 33]. It is optimized for performing lattice-gas simulations. The CAM-8 architecture itself is a simple abstraction of lattice gas dynamics. Lattice gas data streaming and collisions are directly implemented in the architecture. The communication network is a cartesian three-dimensional mesh. Crystallographic lattice geometries can be directly embedded into the CAM-8. Each site of the lattice has a certain number of bits (a multiple of 16) which we refer to as a "cell". Each bit of the cell, or equivalently each bit plane of the lattice, can be translated through the lattice in any arbitrary direction. The translation vectors for the bit planes are termed "kicks". The specification of the x,y, and z components of the kicks for each bit plane (or hyperplane) exactly defines the lattice. An interesting property of the architecture is that the kicks can be changed during the simulation. Therefore, the data movement in the CAM-8 can be quite general. Once the kicks are specified, the coding of the lattice-gas streaming is completed. In effect, the kicks determine all the global permutations of the data.

Local permutations of data occur within the cells. These permutations are the computational metaphor for physical collisions between particles[4]. All local permutations are implemented in look-up tables. That is, all possible physical events with a certain input configuration and a certain output configuration are precomputed and stored in SRAM, for fast table look-up. The width of the CAM-8 look-up tables are limited to 16-bits, or 64K entries. This is a reasonable width satisfying the opposing considerations of model complexity versus memory size limitations for the SRAM. Site permutations of data wider than 16-bits

must be implemented in several successive table look-up passes. Since the look-up tables are double buffered, a scan of the space can be performed while a new look-up table is loaded for the next scan.

Figure 4 is a schematic diagram of a CAM-8 system. On the left is a single hardware module—the elementary "chunk" of the architecture. On the right is an indefinitely extendable array of modules (drawn for convenience as two-dimensional, the array in normally three-dimensional). A uniform spatial calculation is divided up evenly among these modules, with each module simulating a volume of up to millions of fine-grained spatial sites in a sequential fashion. In the diagram, the solid lines between modules indicate a local *mesh* interconnection. These wires are used for spatial data movements. There is also a tree network (not shown) connecting all modules to the front-end host, a SPARC workstation with a custom SBus interface card, that controls the CAM-8. It downloads a bit-mapped pattern as the initial condition for the simulations. It also sends a "step-list" to the CAM-8 to specify the sequence of kicks and scans that evolve the lattice-gas in time. One can view the lattice-gas simulation in real-time since a custom video module captures site data for display on a VGA monitor, a useful feature for lattice-gas algorithm development, test and evaluation. The CAM-8 has built-in 25-bit event counters so that measurements can be done in real-time without slowing the lattice-gas evolution. We have used this feature to do real-time coarse-grain block averaging of the lattice-gas number variables and to compute the components of the momentum vectors for each block. The amount of coarse-grained data is sufficiently small that it can be transferred back to the front-end host for graphical display as an evolving flow field within an X-window. See figures 7, 8 and 9 for example flow fields.

6 The Connection Machine CM-5

The Thinking Machines Corporations's CM-5 is a massively parallel computer that contains up to 16384 processing nodes[53][5]. Figure 5 shows a processing node consisting of a SPARC CPU, 32 Mbytes of memory and 4 Vector processing

units. These processing nodes are all connected via a "fat-tree" communications net that allows fast inter-node communication. These processing nodes are controlled by a front-end host computer which is a modified SUN workstation. The SPARC processor on each node issues instructions to the vector units and performs most bookkeeping tasks while the vector units perform arithmetic and logical operations on the data. Each vector unit has a peak rate of 32 million 64-bit ops (floating point or integer) for a combined total of 128 Mops/node. Each node's memory is divided into 8 Mbyte banks, one for each vector unit. Each vector unit has it's own independent 128 Mbyte/sec path to memory for a combined memory bandwidth of 512 Mbyte/sec for each node. The CM-5 at the Army High Performance Computing Research Center in Minneapolis, Minnesota contains 544 nodes for a total of 16 Gb of memory and 64 Gops of peak processing speed.

We have also implemented a lattice gas simulator on the CM-5 in a multiple instruction multiple data (MIMD) style. The CMMD message passing library is used for inter-node communication and host-node interaction. In order to get the highest possible performance we explicitly manipulate the vector units on each node using their assembler language known as DPEAC. To ease the burden of hand coding the vector units a macro package known as GCC/DPEAC is used. This package uses features available in the GNU C compiler to issue assembler language instructions from ANSI C and simplifies matters considerably.

We partition the problem space into equally sized rectangular units. Each processing node is responsible for updating one of these rectangular units. This partitioning allows one to send a small number of long messages to connect the space together. Inter-node communication is only necessary along one of the axes of the problem space. Since the inter-node communications network is optimized for long message lengths we expect that this partitioning will make the effective use of available communications bandwidth. Within a processing node, each of the 4 vector units is responsible for updating it's quarter of the space. Communication between each vector unit's 8 Mbyte bank of memory is mediated by the SPARC processor.

There are two distinct phases of a lattice gas update cycle. The first phase is the collision phase where particles can interact. The second phase involves streaming of the bits to their new locations, consistent with their velocity and the lattice on which the simulation is being performed. In most lattice gas models all collisions can happen concurrently and all sites can stream their data concurrently as well. The collision phase can be handled via look up tables (LUT's) for 16 bit sites. The LUT is attractive in that it can be an extremely simple and fast update mechanism.

We have distributed the LUTs throughout the machine, indeed each vector unit has it's own copy of the LUT. Figure 6 shows the memory layout on each node. During the collision phase each vector unit fetches all the sites in it's partition of the problem space and runs them through it's copy of the LUT. Since each vector unit has it's own independent 128 Mbyte/sec data path to a bank of memory, this operation can be performed extremely rapidly. With this high degree of parallelism the LUT operation consumes a small fraction of the time necesary to update the space. As the number of bits of site data grows beyond 16 [64K entries] the LUT's begin to consume too much memory. For models that involve larger quantities of site data (*i.e.* # bits > 20) other methods involving LUT compression/decompression need to be used for the collision phase.[54]

The streaming phase is more complex. The approach taken here is to hold the address of each bit's destination in a pre-computed table. These tables may be computed in ordinary C, which is advantageous for changing from one model to another. Additionally, potentially complex addressing calculations are performed only once, during initialization. Before a site can be updated, a communication phase must take place so that each site can access all it's neighbors. Communication must take place across node and vector unit boundaries. The communication is done so that every site has access to all it's neighbor values on one vector unit's 8 Mbyte bank of memory. To update a particular site each vector unit first loads all the neighbors of the site to be updated from the appropriate areas. Then the relevant bits from each neighbor site are extracted

sequentially to build up the new site value. The new site value is then written to memory.

After implementing a lattice-gas simulator with the above considerations in mind we find that we can achieve update rates on the order of 550 Msites/sec on a 256 node partition of the CM-5 for the FHP gas model. This timing was done using a 32K $\times$ 2K lattice. We find that the longer the system is across each node the greater the performance realized. This is due to the fact that long system sizes across each node increase the fraction of sites in the interior of each vector unit that do not need to communicate with sites on adjacent vector units or processing nodes.

7 Gallery of Computational Results

7.1 Rayleigh-Bénard Convection on the CAM-8

A well known fluid instability of a thermohydrodynamic system is Rayleigh-Bénard convection[55, 56]. Rayleigh-Bénard convection is popular because one can observe the onset of order and then the transition to chaos in the flow patterns [57, 58].

Our implementation of the two-speed hexgonal lattice-gas with a rest particle, includes gravitational forcing, free-slip and no-slip boundaries which may be oriented horizontally, vertically, or inclined $\pm 60°$, and heating and cooling sites in order to model temperature controlled boundary surfaces. This has been encoded within the site data space of 16-bits per site for simple implentation on the CAM-8. The ability of encoding such complex dynamics within 16-bits is one of the remarkable aspects of the lattice-gas formalism in terms of efficient memory use affording us the ability to do flash updating from prestored collision tables. 98% of the 2^{16} entries in the collision tables are used (*i.e.*, not the identity) in this model. Similar lattice gas models have been implemented by Burges and Zaleski [59] and by Chen *et al.* [9] and Ernst and Das[60].

To optimize the collision frequency between the fast and slow particles we have chosen their momenta to be of unit value. That is, the slow particles have

unit mass, $m_1 = 1$ and the fast particles have half the mass, $m_2 = 1/2$. In this way, $p_1 = p_2 = 1$ and their energies are $E_1 = 1$ and $E_2 = 2$. With this convention we have the usual FHP-type collisions [32] between the different speed particles while conserving mass, momentum, and energy. These include head-on 2-body collisions, three-body collisions, collisions with spectators, etc. Grosfils, Boon and Lallemand have introduced a three-speed thermohydrodynamic gas with speeds $1, \sqrt{3}, 2$ and a rest particle [61]. With this 19-bit model, efficient collisional mixing can occur with all particle having the same unit mass. Since the particles may now carry different units of energy, in addition to the equation of continuity and Euler's equation, in this system we have an energy transport equation.

7.2 Kelvin-Helmholtz Instability on the CAM-8

Another well known fluid instability is the Kelvin-Helmholtz shear instability. Figure 8 shows the a simulation of a shear instability on a hexagonal lattice 4096×2048 in size with toroidal boundary conditions. A momentum map is overlayed on a vorticity map. Clockwise vorticity is shaded red and counter-clockwise vorticity is shaded blue. The initial conditions for the simulation are very uniform, see Figure 8a. A gas density is chosen, in this case approximately 1/7 filling, and two horizontal regions are set with uniform, but opposing flow directions. That is, the majority of the fluid, the background region, is set with a uniform flow velocity of approximately 0.4 c_s (Mach 0.4) flowing to the right. A narrow stripe 256 sites wide is set in the center of the space flowing to the left at -0.4 c_s. No sinusoidal perturbation is given to the counter-flow narrow stripe region as in previous lattice-gas simulations[62]. No external forcing is applied during the simulation run. The only perturbation is caused by minor fluctations produced by the random number generator when producing a uniform fluid density. After approximately 10,000 time steps, the narrow horizontal center stripe forms a sinusoidal pattern. The sinsoid's amplitude grows, form a wave that eventually breaks into several counter-rotating vortices and the two flow regions begin to substantially mix. Figure 8 shows the initial state of the fluid

and then the resulting states at 10,000 and 30,000 time steps. By $t = 30,000$ the formation of a wave is apparent. Eventually after 400,000 time steps, the fluid attains a uniform flow to the right after the system has equilibrated, exactly conserving the momentum in the initial configuration.

7.3 Von Karman Streets on the CM-5

Figure 9 shows a simulation of vortex shedding from a flat plate after 32,000 time steps on a hexagonal lattice 4096×2048 in size. A momentum map is overlayed on a vorticity map. Clockwise vorticity is shaded red and counter-clockwise vorticity is shaded blue. A flat plate obstacle is placed in a channel of fluid with a flow directed towards the right of the figure. The fluid flow is forced by completely reconstructing the fluid's velocity distribution at each time step in a "forcing strip" at the left of the channel. This forcing method prevents sound waves and other disturbances from propagating around the torus in the flow direction and overwhelming the flow behavior. The boundary conditions are effectively cylindrical.

The flow is started from a random distribution of particles at the appropriate density with a net velocity close to that of the steady state flow. Since this is not a true equilibrium starting condition, some transient behavior appears in the form a sound pulse that propagates down the channel. This pulse is absorbed by the forcing strip. After 2000 time steps the system is equilibrated with no transient phenomena visible. This equilibration time is very short compared with the time necessary for vortex development. The cylindrical boundary condition appears to work extremely well, allowing one to run the simulation to long times.

8 Discussion

We have presented a theoretical description of lattice-gas automata in a discrete D-dimensional space. Attention was focused on two-dimensional fluids for numerical simulation. We have implemented some lattice-gas fluids on a

new prototype cellular automata machine, the CAM-8. Identical models on the CM-5 were also implemented. To illustrate lattice-gas dynamics in the macroscopic limit, several fluid instabilities were tested: (1) the Rayleigh-Bénard convective instability just after the critical Rayleigh number has been reached in the system by suitable gravitational forcing and temperature gradient; (2) the Kelvin-Helmhotz shear instability; and (3) the Von Karman vortex shedding instability.

The strength of the CAM-8 in these simulations is that it is optimized for fine-grained spatial calculations. It can handle many lookup-tables because of its double buffering. It can perform data streaming by spatial data shifts without slowing down the simulation. For a multispeed system or a lattice-gas with long-range interactions, large shifts are necessary. The interaction neighborhood on the CAM-8 need not be local: data can be shifted to the nearest neighbor or a distance thousands of sites away with the same computational effort.

The strengths of the CM-5 in these simulations are its software, size, and flexibility. CM-5 applications can be coded largely in standard programming languages such as C and FORTRAN-90. CM-5 machines offer the possibility of simulating enormous problem spaces due to their much larger memories. In our implementation, streaming is done in the CM-5 by address maps preloaded into memory. Although this uses a lot of memory, the flexiblity of precomputed addressing tables simplifies the implementation of complex lattice geometries.

The main theoretical and computational points of this paper are:

1. Deterministic microscopic lattice-gas dynamics produce the correct Navier-Stokes hydrodynamics in the macroscopic limit where no randomness is used in the local rules. Treating individual digital bits as multispeed fermionic particles allows one to simulate fluid systems where mass, momentum, and energy are exactly conserved and where the dynamics has a time-reversal invariance.

2. A workstation-scale CAM-8 prototype is an efficient, cost-effective platform for lattice gas problems that rivals the capabilities of extant parallel supercomputers.

3. The lattice gas simulation method may be directly ported to a variety

of parallel computer architectures: the facilities provided by most parallel supercomputers are suitable for efficiently running lattice gas applications. Communication costs for running lattice-gas automata simulations decrease as the problem being run increases in size and can essentially be neglected for large systems.

In closing, it is interesting that the massively parallel Connection Machine 2, to date has achieved some of the best update rates for lattice-gases, for example, 700-750 million sites per second update rate for the FHP lattice-gas[63, 23]. Building a multiple instruction multiple data Connection Machine 5, although more focused on general messaging, has improved upon its predecessor's performance (for the FHP lattice-gas, currently the CM-5 could exceed 2 billion updates per second on a 1024-node partition). Although machines like the CM-5 can solve a wider class of computational problems, we believe that effort spent on exploring locally interacting automata models on fine-grained architectures will lead to new practical methods for accurately modeling physics.

9 Acknowledgements

JY would like to thank Dr. Bruce Boghosian for several discussions, particularly those concerning latttice-gas invariants. Thanks expressed also to Donald Grantham and Dr. Robert McClatchey of Phillips Laboratory's Atmospheric Sciences Division for their support of the lattice-gas basic research initiative at our laboratory. Finally, thanks go to Dr. Marc Jacobs, Air Force Office of Scientific Research, who promptly arranged for supercomputer time on the Connection Machine that has proved very valuable to our progress.

Research time on the Connection Machine has been supported by, or in part by the Army Research Office, contract number DAAL03-89-C-0038 with the University of Minnesota Army High Performance Computing Research Center at Minneapolis, Minnesota. Our CM-5 code for lattice gas simulation will be deposited there for use by interested parties.

Identities for Isotropic Lattice Tensors: $\prod_{k=1}^{2n} \hat{\mathbf{e}}_k^\sigma$

Our momentum states are $\hat{\mathbf{e}}_a^\sigma$, where $a = 1, 2, \ldots, B_\sigma$, and where σ denotes the particle speeds. The momentum state-space per particle speed has cardinality B_σ. Lattice summations over odd powers of $\hat{\mathbf{e}}$ must vanish by symmetry. The following identities, listed up to the fourth moment, hold for arbitrary values of B_σ and spatial dimension D [64]

$$\sum_a e_{ai}^\sigma = 0 \tag{43}$$

$$\sum_a e_{ai}^\sigma e_{aj}^\sigma = \frac{B_\sigma}{D}\delta_{ij} \tag{44}$$

$$\sum_a e_{ai}^\sigma e_{aj}^\sigma e_{ak}^\sigma = 0 \tag{45}$$

$$\sum_a e_{ai}^\sigma e_{aj}^\sigma e_{ak}^\sigma e_{al}^\sigma = \frac{B_\sigma}{D(D+2)}(\delta_{ij}\delta_{kl} + \delta_{ik}\delta_{jl} + \delta_{il}\delta_{jk}). \tag{46}$$

C* Implementation

A parallel version of the C-language developed by Thinking Machines Corporation is the C* language[65]. This is a well developed language in spirit very close to its predecessor – it is as concise as the C-language itself. It offers many useful constructs making the coding of algorithms for parallel data very efficient. As typical of most parallel languages, an array operation is handled in a single instruction — for the most part programming loops do not appear in the code. Most parallel computation is achieved by data movement. The geometry of the problem is specified at the onset by defining your the data structure's *shape*. This is usually a D-dimensional array with a certain size in each dimension. The shape definition defines all the needed communication topology for the compiler. It is possible to declare boolean shapes in C*. We have used this feature to encode each bit plane of the lattice-gas. This is a convenient feature of the language making efficient use of memory. Normally, in a lattice-gas code, one must extract and insert individual bits at the lattice sites. The option of working directly with boolean arrays has therefore simplified the coding effort substantially. If individual elements of a parallel array must be accessed, C*

uses the syntax of parallel left indexing. Right indexing of arrays is reserved for its usual C-language meaning. We use right indexed arrays to represent the individual bit planes of the lattice gas.

We have implemented a two-dimensional hexagonal lattice embedded into a three-dimensional mesh. This implementation is equivalent to our implemention in CAMForth on the CAM-8 and GCC/DPEAC on the CM-5. Streaming of pleat 0 and pleat 1 are coded separately. We give a C* code fragment for this embedding. Note that the comments to the right of the lines correspond exactly to the kick components listed in Table 1. With a few C* lines of code one can completely implement hexagonal lattice-gas streaming. We have used the C* command *to-torus-dim(destination pointer, source, axis, distance)* to shift the bit planes with toroidal boundary conditions. This is an efficient communication routine for sending data in a regular fashion using grid communications. The partitioning of the space between processors is handled completely by the C* compiler. We will see how efficiently the compiler does this partitioning in the discussion to follow.

We have tested our C* implementation for different situations. Given a certain lattice size, for example 1024×2048, with have found the performance of the CM-5 to vary linearly with the number of processing nodes. This linear variation is expected so long as the lattice size is sufficiently large. To determine a reasonable lattice size, we have performed repeated simulations with different lattice sizes but with a fixed number of processors. The results obtained for a fixed 256-node partition of the CM-5 is given in figure 10, in which we plotted simulation site update rates for lattice sizes $64 \times 128, 128 \times 256, \ldots, 8192 \times 16384$. For small lattice sizes, the performance is very poor, on the order of a million site updates per seconds. This is because the streaming is limited by processor to processor communication bandwidth. As the lattice size increases, the number of sites interior to the node grows and the number of sites on the partition boundary deceases. Consequently, the site update rate continuously improves with larger lattices. The update rate asymtotically approaches about 25 million site updates per second. This is equivalent to approximately 100,000

sites updates per processing node. This is roughly the maximum update rate achievable on a SPARCstation 1. A full 512-node partition has a peak rate of about 50 million site updates per second, which is about one quarter the speed of the CAM-8.

References

[1] A. Rahman. Correlations in the motion of atoms in liquid argon. *Physical Review*, 136(2A):405–410, 1964.

[2] Stephen Wolfram. Statistical mechanics of cellular automata. *Reviews of Modern Physics*, 55(3):601–644, 1983.

[3] Stephen Wolfram. Universality and complexity in cellular automata. *Physica*, 10D:1–35, 1984.

[4] Tommaso Toffoli and Norman Margolus. *Cellular Automata Machines*. MIT Press Series in Scientific Computation. The MIT Press, 1987.

[5] Tommaso Toffoli. Cellular automata as an alternative to (rather than an approximation of) differential equations in modeling physics. *Physica*, 10D:117–127, 1984.

[6] Richard L. Liboff. *Introduction to the Theory of Kinetic Equations*. Robert E. Krieger Publishing Company, 1979.

[7] J.P. Rivet and U. Frisch. Lattice gas automata in the boltzmann approximation. *Comptes Rendus*, 302(II):p. 267, 1986. In French. Translation appears in Lattice Gas Methods for Partial Differential Equations, SFI SISOC, Eds. Doolen et al., Addison-Wesley Publishing Co., 1990.

[8] Guy R. McNamara and Gianluigi Zanetti. Use of the boltzmann equation to simulate lattice-gas automata. *Physical Review Letters*, 61(20):2332–2335, 1988.

[9] Shiyi Chen, Hudong Chen, Gary D. Doolen, Semion Gutman, and Minxu Lee. A lattice gas model for thermohydrodynamics. *Journal of Statistical Physics*, 62(5/6):1121–1151, 1991.

[10] George Em Karniadakis and Steven A. Orszag. Nodes, modes and flow codes. *Physics Today*, 46(3):34–42, 1993.

[11] Daniel H. Rothman. Negative-viscosity lattice gases. *Journal of Statistical Physics*, 56(3/4):517–524, 1989.

[12] S. Chen, G.D. Doolean, K. Eggert, D. Grunau, and E.Y. Loh. Local lattice-gas model for immiscible fluids. *Physical Review A*, 43(12):7053–7056, 1991.

[13] Andrew K. Gustensen and Daniel H. Rothman. A lattice-gas model for three immiscible fluids. *Physica*, D(47):47–52, 1991.

[14] Andrew K. Gustensen and Daniel H. Rothman. A galilean-invariant immiscible lattice gas. *Physica*, D(47):53–63, 1991.

[15] Hudong Chen, Shiyi Chen, Gary D. Doolen, Y.C. Lee, and H.A. Rose. Multithermodynamic phase lattice-gas automata incorporating interparticle potentials. *Physical Review A*, 40(5):2850–2853, 1989. Rapid Communications.

[16] Cécile Appert and Stéphane Zaleski. Lattice gas with a liquid-gas transition. *Physical Review Letters*, 64:1–4, 1990.

[17] Cécile Appert, Daniel Rothman, and Stéphane Zaleski. A liquid-gas model on a lattice. In Gary D. Doolean, editor, *Lattice Gas Methods: Theory, Applications, and Hardware*, pages 85–96. Special Issues of Physica D, MIT/North Holland, 1991.

[18] Jeffrey Yepez. A reversible lattice-gas with long-range interactions coupled to a heat bath. In Gary D. Doolean, editor, *Proceedings of the Pattern Formation and Lattice-Gas Automata Conference*. Fields Institute, American Mathematical Society, 1993. To appear.

[19] Shiyi Chen, Hudong Chen, Daniel Martínez, and William Matthaeus. Lattice boltzmann model for simulation of magnetohydrodynamics. *Physical Review Letters*, 67(27):3776–3779, 1991.

[20] Daniel Rothman. Macroscopic laws for immiscible two-phase flow in porous media: results from numerical experiments. *Journal of Geophysics Research*, 95:8663, 1990.

[21] Shiyi Chen, Karen Diemer, Gary D. Doolen, and Kenneth Eggert. Lattice gas automata for flow through porous media. In Gary D. Doolean, editor, *Lattice Gas Methods: Theory, Applications, and Hardware*, pages 72–84. Special Issues of Physica D, MIT/North Holland, 1991.

[22] Norman Margolus. Cam-8: a computer architecture based on cellular automata. In Gary D. Doolean, editor, *Proceedings of the Pattern Formation and Lattice-Gas Automata Conference*. Fields Institute, American Mathematical Society, 1993. To appear.

[23] Bruce M. Boghosian. Lattice gases illustrate the power of cellular automata in physics. *Computers in Physics*, 5(6), Nov/Dec 1991.

[24] J. Hardy, O. de Pazzis, and Y. Pomeau. Molecular dynamics of a classical lattice gas: Transport properties and time correlation functions. *Physical Review A*, 13(5):1949–1961, 1976.

[25] Tommaso Toffoli. Computation and construction universality of reversible cellular automata. *Journal of Computer and System Sciences*, 15(2):213–231, 1977.

[26] Edward Fredkin and Tommaso Toffoli. Conservative logic. *International Journal of Theoretical Physics*, 21(3/4):219–253, 1982.

[27] Norman Margolus. Physics-like models of computation. *Physica*, 10D:81–95, 1984.

[28] Tommaso Toffoli. Cam: A high-performance cellular-automaton machine. *Physica*, 10D:195–204, 1984. A demonstration TM-gas experiment was part of the CAMForth software distribution.

[29] Gerard Y. Vichniac. Simulating physics with cellular automata. *Physica*, 10D:96–116, 1984.

[30] Norman H. Packard and Stephen Wolfram. Two-dimensional cellular automata. *Journal of Statistical Physics*, 38(5/6):901–946, 1985.

[31] Stephen Wolfram and James B. Salem. Thermodynamics and hydrodynamics with cellular automata. In Stephen Wolfram, editor, *Theory and Applications of Cellular Automata*, pages 362–365. World Scientific, 1986. Submitted November 1985.

[32] Uriel Frisch, Brosl Hasslacher, and Yves Pomeau. Lattice-gas automata for the navier-stokes equation. *Physical Review Letters*, 56(14):1505–1508, 1986.

[33] Norman Margolus, Tommaso Toffoli, and Gérand Vichniac. Cellular-automata supercomputers for fluid-dynamics modeling. *Physical Review Letters*, 56(16):1694–1696, 1986.

[34] Uriel Frisch, Dominique d'Humières, Brosl Hasslacher, Pierre Lallemand, Yves Pomeau, and Jean-Pierre Rivet. Lattice gas hydrodynamics in two and three dimensions. *Complex Systems*, 1:649–707, 1987.

[35] Michel Hénon. Viscosity of a lattice gas. In Gary D. Doolean, editor, *Lattice Gas Methods for Partial Differential Equations*, pages 179–207. Santa Fe Institute, Addison-Wesley Publishing Company, 1990.

[36] J.A. Somers and R.C. Rem. The construction of efficient collision tables for fluid flow computations with cellular automata. In P. Manneville, N. Boccara, G.Y. Vichniac, and R. Bidaux, editors, *Cellular Automata and Modeling of Complex Physical Systems*, pages 161–177. Springer-Verlag, February 1989. Proceedings of the Winter School, Les Houches, France.

[37] Norman Margolus and Tommaso Toffoli. Cellular automata machines. In Gary D. Doolean, editor, *Lattice Gas Methods for Partial Differential Equations*, pages 219–249. Santa Fe Institute, Addison-Wesley Publishing Company, 1990. The first 8-module CAM-8 prototype was operational in the fall of 1992.

[38] Norman Margolus. Parallel quantum computation. In W.H. Zurek, editor, *Complexity, Entropy, and the Physics of Information,SFI Studies in the Sciences of Complexity, vol. VIII*, pages 273–287. Addison-Wesley, 1990.

[39] Seth Lloyd. A technologically feasible quantum computer. Complex System Group T-13, Los Alamos National Laborarory, Preprint.

[40] Michael Biafore. Cellular automata for nanometer-scale computation. *Physica D*, 70(3/4), 1994.

[41] Detlef Heitmann and Jorg P. Kotthaus. The spectroscopy of quantum dot arrays. *Physics Today*, 46(6):56–63, 1993.

[42] Charles H. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 6:525–532, 1979.

[43] Charles H. Bennett. Thermodynamics of computation—a review. *International Journal of Theoretical Physics*, 21:219–253, 1982.

[44] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467–488, 1982.

[45] B.J. Alder and T.E. Wainwright. Velocity autocorrelations for hard spheres. *Physical Review Letters*, 18(23):988–990, 1967.

[46] Yves Pomeau. A new kinetic theory for a dense classical gas. *Physics Letters*, 27A(9):601–602, 1968.

[47] M.H. Ernst, E.H. Hauge, and J.M.M van Leeuwen. Asymptotic time behavior of correlation functions. *Physical Review Letters*, 25(18):1254–1256, 1970.

[48] T.R. Kirkpatrick and M.H. Ernst. Kinetic theory for lattice-gas cellular automata. *Physical Review A*, 44(12):8051–8061, 1991.

[49] R. Brito and M.H. Ernst. Lattice gases in slab geometries. *Physical Review A*, 44(12):8384–8687, 1991.

[50] R. Brito and M.H. Ernst. Ring kinetic theory for tagged-particle problems in lattice gases. *Physical Review A*, 46(2):875–887, 1992.

[51] Chihwen Li and Chwan-Hwa Wu. A particle-in-cell fluid model for radio frequency glow discharges. *Computers in Physics*, 7(3):363–375, 1993.

[52] Christopher Teixeira. *Continuum Limit Of Lattice Gas Fluid Dynamics*. PhD thesis, Massachusetts Institute of Technology, Department of Nuclear Engineering, 1992. Kim Molvig Thesis Supervisor.

[53] TMC. *CM5 Technical Summary*. Thinking Machines Corporation, Cambridge, Massachusetts USA, 1992.

[54] Michel Hénon. Implementation of the fchc lattice gas model on the connection machine. *Journal of Statistical Physics*, 68(3/4):353–377, 1992.

[55] Christiane Normand and Yves Pomeau. Convective instability: A physicist's approach. *Reviews of Modern Physics*, 49(3):581–624, 1977.

[56] E.G.D. Cohen. The kinetic theory of fluids–an introduction. *Physics Today*, pages 64–73, January 1984.

[57] Giorgio Careri. *Order And Disorder In Matter*. The Benjamin/Cummings Publishing Company, 1984.

[58] Gregory L. Baker and Jerry P. Gollub. *Chaotic Dynamics an introduction*. Cambridge University Press, 1990.

[59] Christopher Burges and Stéphane Zaleski. Buoyant mixtures of cellular automaton gases. *Complex Systems*, 1:31–50, 1987.

[60] M.H. Ernst and Shankar P. Das. Thermal cellular automata fluids. *Journal of Statistical Physics*, 66(1/2):465–483, 1992.

[61] P. Grosfils, J.P. Boon, and P. Lallemand. 19-bit thermal lattice-gas automaton. *Journal of Statistical Physics*, 1993. To appear.

[62] Tsutomu Shimomura, Gary D. Doolen, Brosl Hasslacher, and Castor Fu. Calculations using lattice gas techniques. In Gary D. Doolean, editor, *Lattice Gas Methods for Partial Differential Equations*, pages 3–9. Santa Fe Institute, Addison-Wesley Publishing Company, 1990.

[63] S. Chen, G.D. Doolean, K. Eggert, D. Grunau, and E.Y. Loh. Lattice gas simulations of one and two-phase fluid flows using the connection machine-2. In A.S. Alves, editor, *Discrete Models of Fluid Dynamics*, pages 232–248. World Scientific, September 1990. Series on Advances in Mathematics for Applied Sciences, Vol. 2.

[64] Stephen Wolfram. Cellular automaton fluids 1: Basic theory. *Journal of Statistical Physics*, 45(3/4):471–526, 1986.

[65] TMC. *C* Programming Guide*. Thinking Machines Corporation, Cambridge, Massachusetts USA, 1990. Version 6.0.

Table 1: Streaming for 2D Hex Lattice Embedded a into 3D Mesh

Direction	**x**	**y**	**z**
PLEAT 0			
1	0	0	1
2	0	-1	1
3	0	-1	0
4	1	-1	1
5	1	0	1
6	0	1	0
PLEAT 1			
1	-1	1	-1
2	-1	0	-1
3	0	-1	0
4	0	0	-1
5	0	1	-1
6	0	1	0

[1] We have written lattice-Boltzmann code in the parallel C-Star language on the Connection Machine-5.

[2] Lattice Boltzmann simulation for three dimensional flows with Reynolds numbers of about 50,000 were presented in June 1993 at the International Conference on Pattern Formation and Lattice-Gas Automata sponsored by the Fields Institute, Waterloo, Canada.

[3] Actually, the most straight foward way to calculate the matrix elements of J is to write a Mathematica code.

[4] Locally, the CAM-8 is not limited to performing only permutations, it can do general mappings. However, since we are interested in only particle conserving reversible dynamics, permutations are sufficient.

[5] Currently the largest CM-5 resides at Los Alamos National Laboratory with 1024 Nodes.

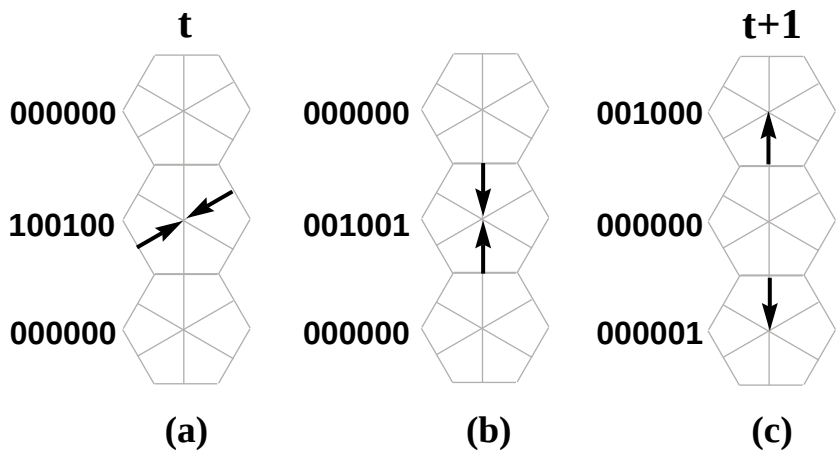

Figure 1: Illustration on a hexagonal lattice of the two-step collision and streaming process required to complete a single time step: (a) initial configuration, (b) collision by permutation, and (c) streaming of particles.

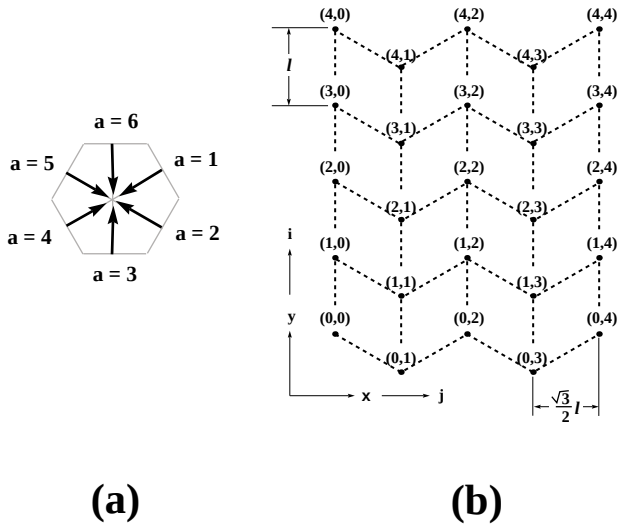

Figure 2: (a) Lattice vector label convention; (b) Hexagonal lattice convention with lattice directions $a = 3$ up and $a = 6$ down. Coordinates above the lattice nodes are (i, j) memory array indices.

Figure 3: MIT Laboratory for Computer Science cellular automata machine CAM-8. This 8 module prototype can evolve a D-dimensional cellular space with 32 million sites where each site has 16 bits of data with a site update rate of 200 million per second.

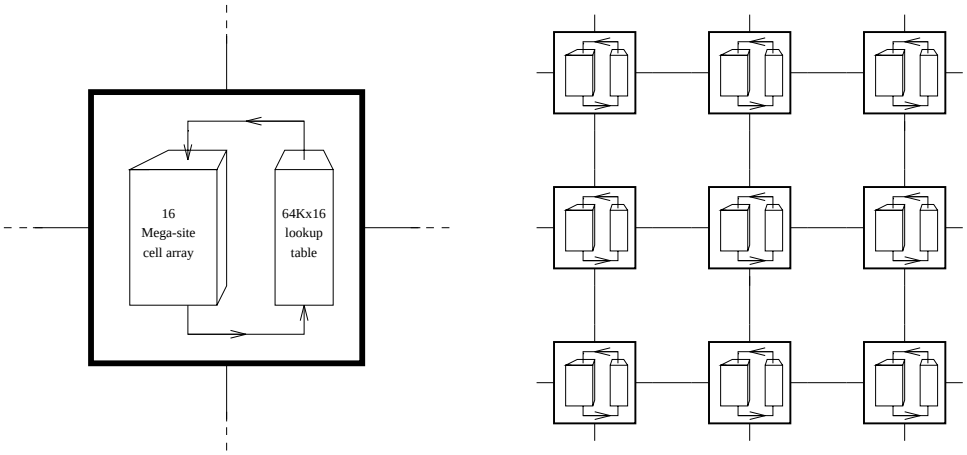

Figure 4: (a) A single processing node, with DRAM site data flowing through an SRAM lookup table and back into DRAM. (b) Spatial array of CAM-8 nodes, with nearest-neighbor (mesh) interconnect (1 wire/bit-slice in each direction).

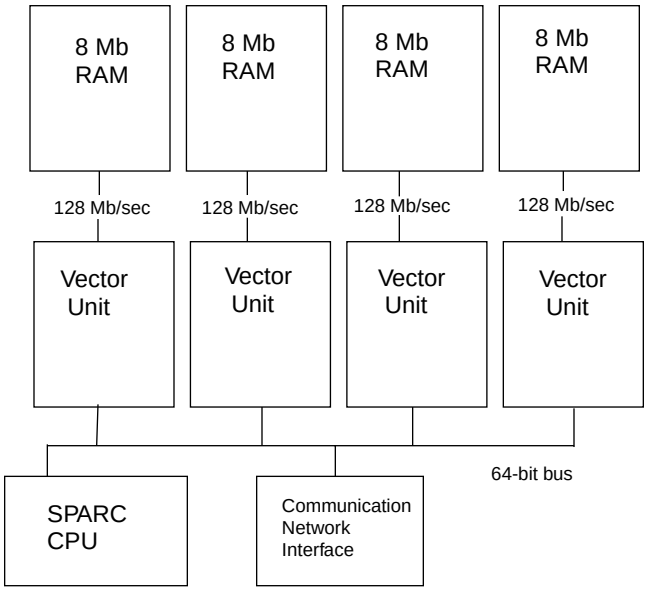

Figure 5: CM-5 Node: SPARCCPU, 32 Mbytes of memory and 4 Vector processing units.

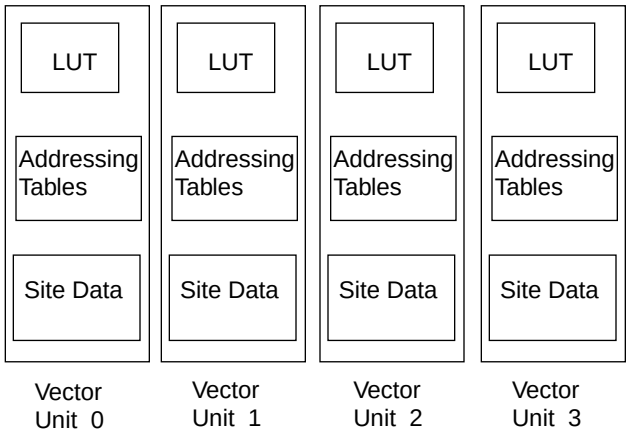

Figure 6: Node Memory Layout.

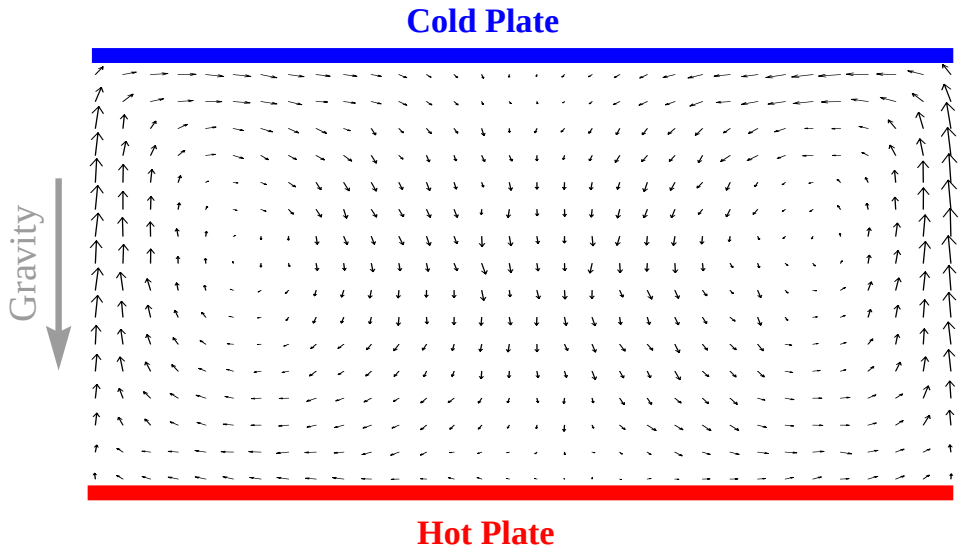

Figure 7: Thermo 13-bit CAM-8 Experiment: Rayleigh-Bénard convection cells at the critical Rayleigh number. Lattice Size: 2048×1024. Time Average: 100. Spatial Average: 64×64. Mass Density Fraction=1/5. Data presented at 50,000 time steps.

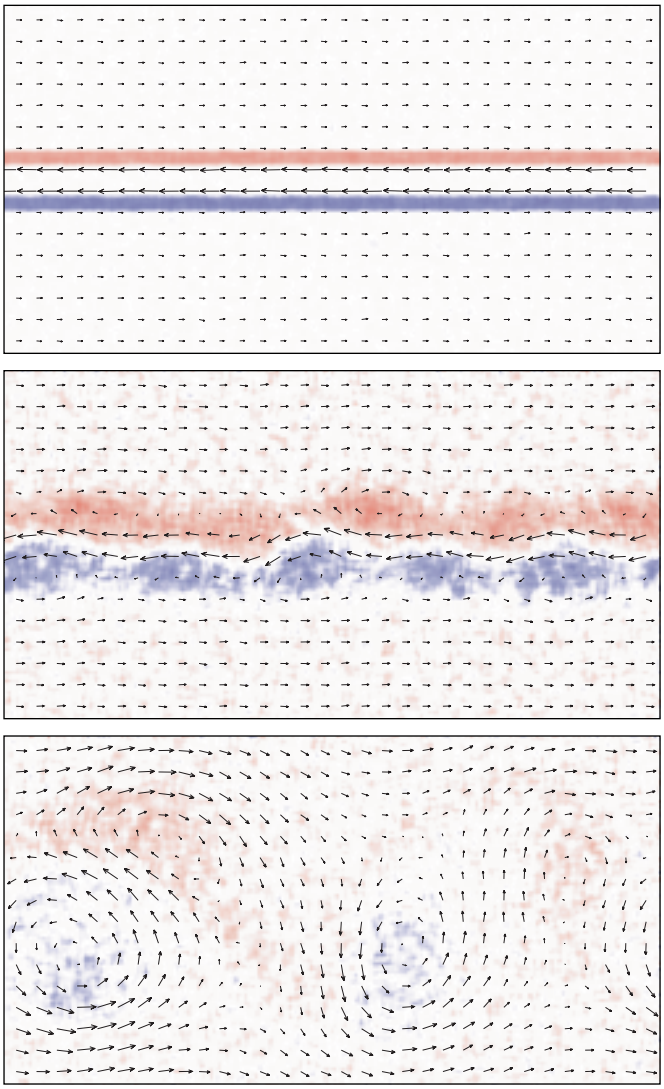

Figure 8: Momentum and vorticity map of two-dimensional shear instability on the CAM-8. Lattice size of 4096×2048 with toroidal boundary conditions. Spacetime averaging over 128x128 blocks for 50 time steps. FHP collisions with spectators and a rest particle. Data presented at time steps 0, 10000, and 30000 with Galilean velocity shift.

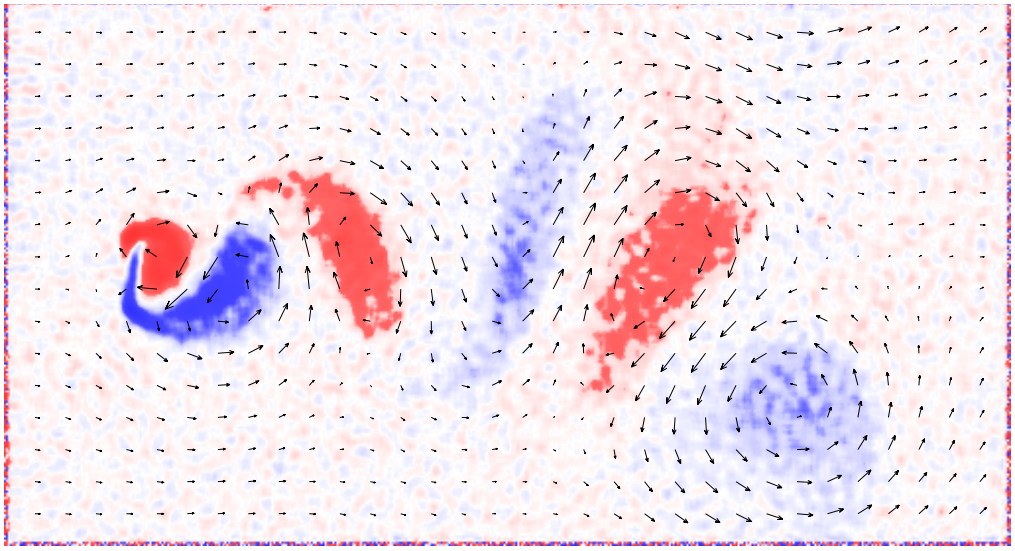

Figure 9: FHP-II CM-5 Experiment: Von Karman Streets Lattice Size: 4096×2048. Time Average: None. Spatial Average: 128x128. Mass Density Fraction=1/7. Data presented at 32,000 time steps.

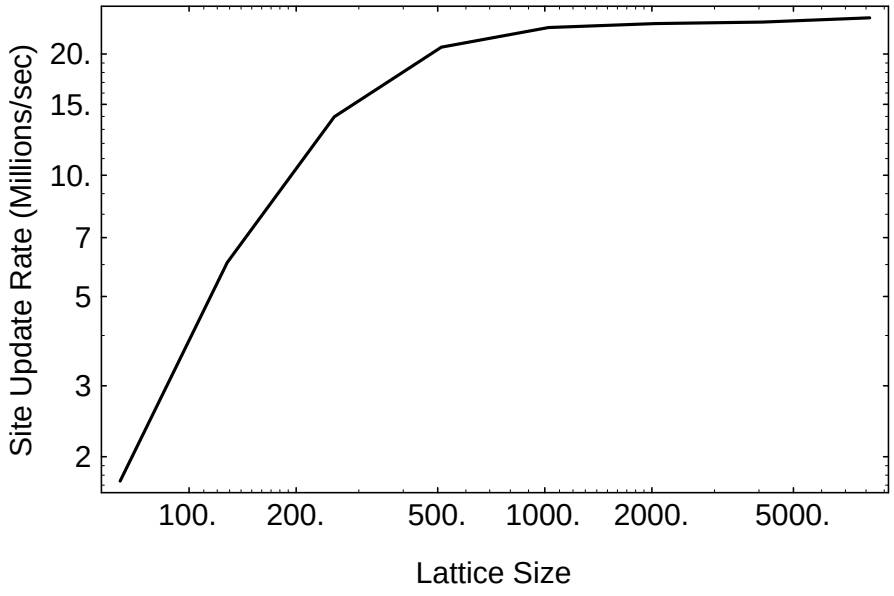

Figure 10: Performance runs on a 256-node CM-5 for an FHP hexagonal lattice embedded into a 3D mesh. Performance significantly suffers by communication overhead for small lattice sizes.