GPGPU simulations of 2D lattice neutral models in ecology

Takeshi Oura[1] **and Kei Tokita**[2]

[1]Cybermedia Center and Graduate School of Science, Osaka University, Toyonaka, Osaka 560-0043, Japan

[2]Cybermedia Center, Graduate School of Science and Graduate School of Frontier Biosciences, Osaka University, Toyonaka, Osaka 560-0043, Japan

E-mail: [1]oura@cp.cmc.osaka-u.ac.jp, [2]tokita@cmc.osaka-u.ac.jp

Abstract. 2D lattice neutral models in ecology are studied using general-purpose computing on graphic processing units (GPGPU). Processing times of GPGPU and CPU simulations are compared for various system sizes and it is found out that larger the system size, faster the GPGPU version and the efficiency of GPGPU is maximally 1010 times higher. Ecological significance of the GPGPU simulations and the lattice neutral model is also reported.

1. Introduction

General-purpose computing on graphic processing units (GPGPU) [1] is one of the recent hot topics not only in computer sciences but also in simulation sciences like computational physics. It is still fresh in our memory that the N-body simulations on GPUs for astrophysics and turbulence by the DEGIMA GPGPU cluster system in Nagasaki University won Gordon Bell award in the Price/Performance category in 2009 and 2010 [2, 3], the GPU-accelerated TSUBAME 2.0 supercomputer at the Tokyo Institute of Technology ranked #5 on the TOP500 in 2010 [4] and the peta-scale phase filed simulation for dendritic solidification on TSUBAME won Gordon Bell award in the Special Achievement in Scalability and Time-to-Solution category in 2011 [5]. In the present study we demonstrate that GPGPU is also highly effective for simulations in biological physics and achieves maximally 1010-fold speedup compared to CPU simulations.

We here study a 2D lattice version of the Hubbell's neutral model of biodiversity[6, 7] which appropriately predicts species abundance distributions (SAD) while there are some controversy on the validity of its "neutral hypothesis". SAD has been thoroughly surveyed by field ecologists and also recently theoretically led by statistical mechanics for population dynamics with complex interspecies interactions [8]. One of the major problems of many studies on the neutral model is that they are mainly based on the mean-field theory [9] for essentially infinite dimensional systems though real ecosystems of sedentary species such as tropical rain forests and coral reefs originally intended by the neutral theory are essentially of 2D and finite length of migration of plants' seeds and coral larvae is important. Because mathematical analysis for 2D models has not progressed very much we here study 2D lattice neutral model using GPGPU.

2. Model

We consider an ecological model of L^2 individuals of species occupying $L \times L$ 2D lattice sites, being born, dying, migrating and evolving. Every lattice site is anytime occupied by an individual of some species, that is, no vacant site occurs and total number of individual is constant ($= L^2$) at all times, which is modeled on a "saturated" community like reefs with no space between corals and rain forests covered by tree crowns with no vacant space.

The model has five parameters: the death rate d, the birth rate b, the migration distance r, the speciation rate ν and the system size $L \times L$. The neutral theory assumes that the values of these parameters are constant and independent among species though they are essentially different among species. The migration area A is a set of the neighborhood of (i, j) and in the present study we assume the Moore neighborhood except the center ($A = M(i, j) = \{(i', j') : |i - i'| \leq r, |j - j'| \leq r, (i', j') \neq (i, j)\}$). The migration distance r is one of the essential parameter of the present model. The simulation starts from an initial state in which random natural numbers $s\,(s = 1, 2, 3, \cdots)$ are distributed onto $L \times L$ lattice sites. The number $s(i, j)$ on the site (i, j) denotes the species number. In one Monte Carlo step the following procedures are executed and repeated $L \times L$ times.

3. Algorithm

3.1. Original algorithm

(1) Choose randomly one site (i, j) and kill the individual on it with probability $d(\leq 1)$. Here we set $d = 1$ without loss of generality because d changes only the time scale of the dynamics.

(2) With probability $\nu(\leq 1)$ assign a new species $s' \notin \{s(i, j)|i, j = 1, 2, \cdots, L)\}$ onto (i, j). This process denotes evolution of a new species or migration of a new species from outside the system. In the present study we use a same value of $\nu = 0.001$.

(3) If the new species is not assigned in (2), choose one individual (s'') in the migration area A with probability in proportion to $b\rho$, where ρ is the density of species s'' in A, and copy s'' onto (i, j). The migration-distance r is one of the essential parameter of the present model. This process corresponds to fixation of pollens of plants or coral larvae from parental individuals in the migration area. We here set $b = 1$ without loss of generality because b is independent of species number under the neutral hypothesis.

The above Monte Carlo steps are iterated until the system leads to a stationary state at a time which depends on the parameters. We use periodic boundary conditions. Simple implementation by pseudocode of the original algorithm is given in Algorithm 1.

Algorithm 1 Original algorithm

```
for time t < end do
  i ⇐ integer random number generated by U(L)
  j ⇐ integer random number generated by U(L)
  x ⇐ random number generated by U(1)
  if x < ν then
    s(i, j) ⇐ new species
  else
    i' ⇐ integer random number generated by U(i − r, i + r)
    j' ⇐ integer random number generated by U(j − r, j + r)
    s(i, j) ⇐ s(i', j')
  end if
end for
```

1	**2**	1	2	1	2	1	2
3	**4**	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4
1	2	1	2	1	2	1	2
3	4	3	4	3	4	3	4

(a) $r = 1$, $L = 8$

1	**2**	**3**	1	2	3
4	**5**	**6**	4	5	6
7	**8**	**9**	7	8	9
1	2	3	1	2	3
4	5	6	4	5	6
7	8	9	7	8	9

(b) $r = 2$, $L = 6$

1	**2**	**3**	**4**	1	2	3	4
5	**6**	**7**	**8**	5	6	7	8
9	**a**	**b**	**c**	9	a	b	c
d	**e**	**f**	**0**	d	e	f	0
1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
9	a	b	c	9	a	b	c
d	e	f	0	d	e	f	0

(c) $r = 3$, $L = 8$

Table 1. Sites with same number updated at the same time.

3.2. Modified algorithm for parallel processing

We modified the algorithm for parallel processing using the checker-board decomposition[10]. Sites which have no influence on each other are able to be updated at the same time. These sites are calculated in parallel. Table 1 is a scheme to construct such sites. For example, in the case of Table 1(a), the sites with same number ("1", "2", "3" and "4") are updated in parallel and the one Monte Calro step (MCS) is processed in four parallel steps for "1", "2", "3" and "4" in principle. Sites processed at the same time are at the interval of r. The smaller r, the larger the number of sites is, the better performance is. As the number of the groups updated in parallel A is $(r+1)^2$ shown in Table 1, the number of the sites which can be updated in parallel is $L^2/(r+1)^2$. If the number of available cores is sufficiently lager than the number of parallel tasks, processing times are proportional to $(r+1)^2$. Simple implementation by pseudocode of the modified algorithm for the parallel processing is given in Algorithm 2.

Algorithm 2 Modified algorithm

```
for time t < end do
  for set ∈ {set'|{i, j|each s(i, j) have no influence }} do
    for each (i, j) ∈ set in parallel do
      x ⇐ random number generated by U(1)
      if x < ν then
        s(i, j) ⇐ new species
      else
        i' ⇐ integer random number generated by U(i − r, i + r)
        j' ⇐ integer random number generated by U(j − r, j + r)
        s(i, j) ⇐ s(i', j')
      end if
    end for
  end for
end for
```

4. Results

4.1. Settings

We compared processing times of simulations by GPUs and multi-core environments of CPUs. For GPGPU the calculation is executed by nVidia Tesla C2050 and C2050×2(+openMP). As a multi-CPUs environment we use Intel Xeon W3350 4 cores for openMP calculation and Xeon 5160 4∼100 cores for MPI. One node has 2 Xeon 5160 (= 4 cores). In the following processing times by Xeon W3350 1 core is used as a baseline for the performance.

4.2. System-size (L^2) dependency of processing times

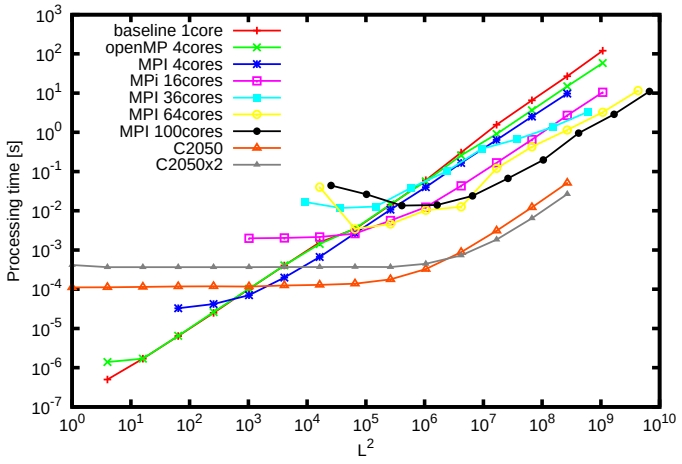

Figure 1. Processing times per 1MCS vs. system size L^2 ($r = 1$)

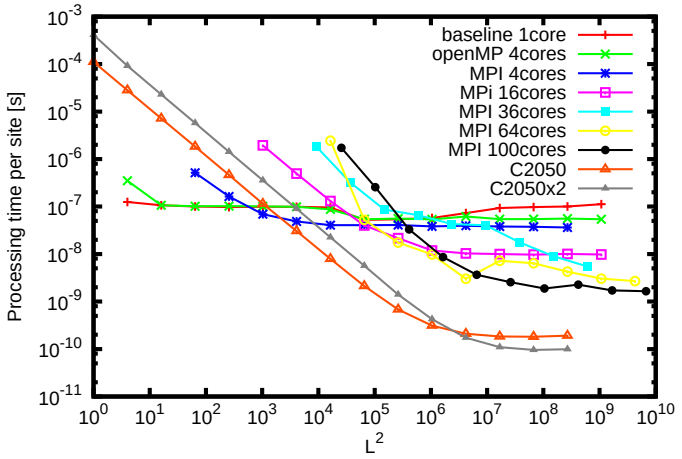

Figure 2. Processing times per site vs. system size L^2 ($r = 1$)

Processing times at various system size L^2 are mesuared to clarify size dependency. Figure 1 shows observed processing times per 1 MCS by GPUs and multi-core environments of CPUs. In wide range of L^2 C2050 and C2050×2 are faster than multi-CPUs. In MPI calculations the

more cores used, the worse processing times at small L. In addition system-size (L) dependency of MPI processing with higher degree of parallel (> 36 cores) fluctuates in comparison with C2050's. Figure 2 represents processing times per sites. At $L^2 > 10^7$ effects of overhead on C2050 is ignorable. In contrast effect of overhead on MPI is significant. One weak point of C2050 is that C2050 can calculate up to $L \sim 16384$ because C2050's memory size is only 2 GByte. Peak performance of GPUs and CPUs is summarized in Table 2. From Figure 2 and Table 2 for the system $L \times L = 16384 \times 16384 = 268435456$ the GPGPU calculation is 523 and 1010 times more efficient than the CPUs for C2050 and C2050×2 respectively.

	Processing times per site [s]	Scale factor $\left(\frac{\text{Baseline time}}{\text{Processing time}}\right)$
Xeon W3350 1core (baseline)	1×10^{-7}	1
Xeon W3350×4cores (openMP)	5.42×10^{-8}	1.84
Xeon 5160	1.92×10^{-7}	0.52
Xeon 5160×4cores (MPI)	3.63×10^{-8}	2.756
Xeon 5160×16cores (MPI)	1.07×10^{-8}	9.35
Xeon 5160×36cores (MPI)	6.08×10^{-9}	16.45
Xeon 5160×64cores (MPI)	3.71×10^{-9}	27.06
Xeon 5160×100cores (MPI)	1.65×10^{-9}	44.38
Tesla C2050	1.91×10^{-10}	523
Tesla C2050×2	9.9×10^{-11}	1010

Table 2. Peak performance of the simulations by single core, multicore (MPI) and many cores of GPGPU.

4.3. Migration-distance (r) dependency of processing times

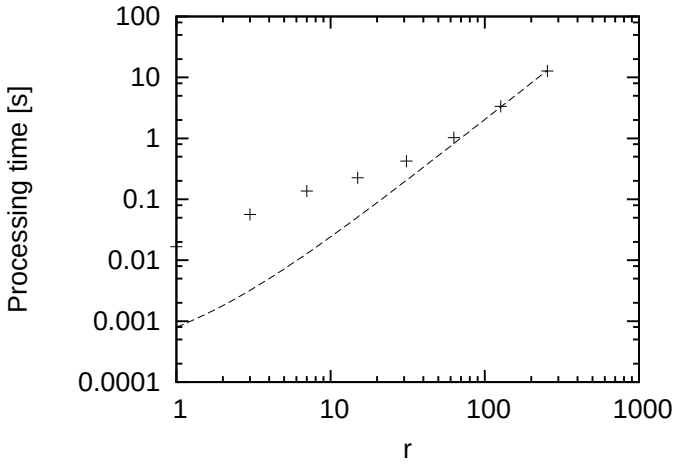

Figure 3. Processing times per 1MCS vs. migration distance r ($L = 16384$)

In Figure 3 cross symbols represent processing times for various r at 1MCS. Dashed line is proportional to $(r+1)^2$. As noted previously, processing times are proportional to $(r+1)^2$. But at small r the number of sites processed simultaneously are large, thus processing times increase against $(r+1)^2$. The larger r, processing times close to $(r+1)^2$ asymptotically. At $r > 60$ processing times are coincided with $(r+1)^2$.

4.4. Rank abundance relations

While we saw the advantage of GPGPU against multi-core environments in the previous section, we here consider the biological significance of the present study. In the context of the study of the neutral model one of the most important observable is the rank abundance relations (RAR) (the relations between abundance and the specie's rank, which is essentially equivalent to SAD). The migration-distance (r) dependency of RAR is depicted in Figure 4, which clearly displays that migration-distance completely changes RAR. In the figure lower-ranked species with small abundance much less than 6000 are omitted. This migration-distance (r) dependency was essentially not clarified by the mean field theory of the neutral model. As the theoretical treatment of the spatial explicit model has not been progressed, the present study verifies that the large scale GPGPU simulation is effective. As the derivative of the inverse function of RAR is SAD, we observe that for larger r larger width of SAD, which means that abundant and rare species coexist in the system.

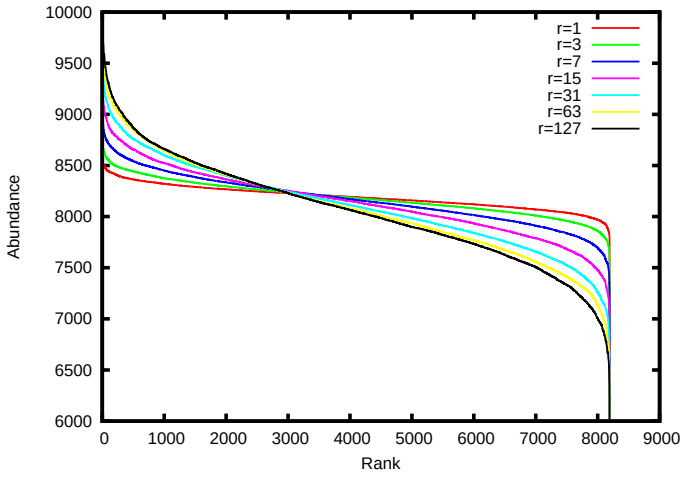

Figure 4. Rank abundance relations for various r ($L = 8192$).

5. Summary

Simulations are executed for various system sizes and each processing time of simulations using GPGPU (nVIDIA Tesla C2050) and CPU (Intel Xeon W3530 2.80GHz and Intel Xeon 5160 3.00 GHz) are compared. This scale factor 1010 is even more than the number of the CUDA cores ($= 448 \times 2$) of Tesla C2050, which means that the simulation code for the lattice neutral model is highly optimized for parallel computation using GPGPU. In the calculations of the present model the performance of one board of C2050 is comparative to a PC cluster with hundreds of cores. In this sense, GPGPU provides much better cost performance. As the present model is a variant of a lattice model with mesoscopic interactions often studied in physics, the scalability and the high cost-performance of the GPGPU simulation gives a promise for other studies of computational physics.

6. Acknowledgements

The authors thank Mr. Sugahara for illuminating discussions. This work was supported in part by Global COE Program (Core Research and Engineering of Advanced Materials—Interdisciplinary Education Center for Materials Science), MEXT, Japan. This work was also supported in part by the initiative-based project E0-5 "Creation and Sustainable Governance of

New Commons through Formation of Integrated Local Environmental Knowledge ", Research Institute for Humanity and Nature (RIHN). K.T. acknowledges support by grants-in-aid from MEXT, Japan (Grants NO.24570099). MPI calculations are carried out on the PC clusters at Cybermedia Center, Osaka University.

References

[1] *GPGPU*, Wikipedia, `http://en.wikipedia.org/wiki/GPGPU`
[2] Hamada T *et al.* 2009 *ACM Gordon Bell prize price/performance, SC09*, Portland, USA, November 2009
[3] Hamada T and Nitadori K 2010 *ACM Gordon Bell prize honorable mention price/performance, SC10*, New Orleans, USA, November 2010
[4] *TOP500* 2010 `http://www.top500.org/lists/2010/11`
[5] Shimokawabe T *et al.* 2011 *ACM Gordon Bell prize special achievements in scalability and time-to-solution, SC11*, Seattle, USA, November 2011
[6] Hubbell S P 2001 *The Unified Neutral Theory of Biodiversity and Biogeography* (Princeton: Princeton University Press)
[7] Rosindell J, Hubbell S P and Etienne R S 2011 *TREE* **26** 340–348
[8] Tokita K 2004 *Phys. Rev. Lett.* **93** 178102
[9] Etienne R and Alonso D 2007 *J. Stat. Phys.* **128** 485–510
[10] Landau D P and Binder K 2000 *A Guide to Monte Carlo Simulations in Statistical Physics* (Cambridge University Press)