Performance Modeling, Analysis, and Optimization of Cell-List Based Molecular Dynamics

Manaschai Kunaseth[1], **Rajiv K. Kalia**[1], **Aiichiro Nakano**[1], **Priya Vashishta**[1]
[1]Collaboratory for Advanced Computing and Simulations (CACS)
Department of Computer Science, Department of Physics, Department of Materials Science
University of Southern California, Los Angeles, CA 90089-0242, USA
(kunaseth, rkalia, anakano, priyav)@usc.edu

Abstract - *We have developed a performance prediction model for non-bonded interaction computations in molecular dynamics simulations, thereby predicting the optimal cell dimension in a linked-list cell method. The model expresses computation time in terms of the number and unit computation time of key operations. The model accurately estimates the number of operations during the simulations with the maximum standard error of 10.6% compared with actual measurements. Then, the unit computation times of the operations are obtained by bisquare regression. Analysis of this model reveals that the optimal cell dimension to minimize the computation time is determined by a trade-off between decreasing search space and increasing linked-list cell access for smaller cells. The model predicts the optimal cell dimension correctly for 80% of all tested cases, resulting in an average speedup of 10% and 52% for the cutoff radius of interaction, 6.6 and 10.0 Å, respectively.*

Keywords: Performance prediction model, Molecular dynamics simulation, Linked-list cell method, Performance optimization, Bisquare regression

1. Introduction

Molecular dynamics (MD) simulation is widely used to study material properties at the atomistic level. Large-scale MD simulations involving multibillion atoms are beginning to address broad problems [1-5], but increasingly large computing power is needed to encompass even larger spatiotemporal scales [6]. One of the widely used method for improving the performance and scalability of such large-scale MD simulations is the linked-list cell method, which employs a divide-and-conquer strategy to reduce the search space for atomic pairs within the cutoff radius R_c of the interatomic interaction [5, 7-9].

The conventional cell decomposition method uses cubic cell geometry with the dimension ~ R_c. However, this incurs considerable amount of unnecessary pair evaluations due to the excessive cell dimension. A commonly used method to decrease the redundant calculations is reducing the cell dimension, see *e.g.* Mattson *et al.* [7]. In particular, Yao *et al.* stated that the cell dimension of $R_c/2$ usually gives the best performance [8]. Our own experience, however, indicates that the optimal cell dimension varies considerably for different parameters such as R_c and the number of atoms. It is thus of great interest to systematically study and identify the factors that determine the optimal cell dimension.

To address this problem, we carry out a theoretical analysis to estimate the amount of non-bonded interaction computations, which dominate the computational time. Then, the platform-dependent computational cost associated with each key operation is determined by regression. Finally, we use the model to predict the performance as a function of the cell dimension, thus obtain the optimal cell dimension to minimize the computation time.

This paper is organized as follows. Section 2 presents the linked-list cell MD and reduced linked-list cell method, to which the proposed performance model is applied. Section 3 describes the theoretical analysis of simulation parameters related to the cell dimension selection. Section 4 presents the verification of the performance model including the fitting procedure of each estimated parameters. Section 5 evaluates the model against measured computation times. Conclusions are drawn in section 6.

2. Linked-list Cell Based Molecular Dynamics

Molecular dynamics simulation follows the phase-space trajectories of an N-atom system, where

force fields describing the atomic force laws between atoms are spatial derivatives of a potential energy function $E(\mathbf{r}^N)$ ($\mathbf{r}^N = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N\}$ is the positions of all atoms). Positions and velocities of all atoms are updated at each MD step by numerically integrating coupled ordinary differential equations. The dominant computation of MD simulation is the evaluation of $E(\mathbf{r}^N)$, which consists of two-body $E_2(\mathbf{r}_i,\mathbf{r}_j)$ and three-body $E_3(\mathbf{r}_i,\mathbf{r}_j,\mathbf{r}_k)$ terms in the program considered here [5].

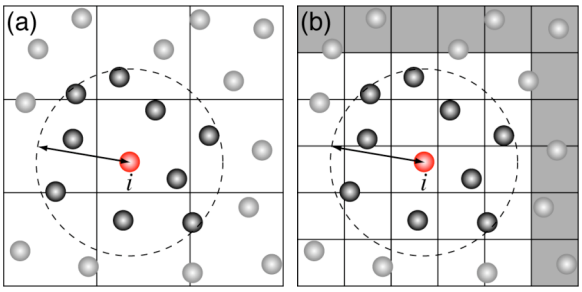

Figure 1: 2D schematic of the linked-list cell method. (a) Illustration of conventional linked-list cell method with cell dimension $R_s \geq R_c$. (b) 2D schematic of the reduced linked-list cell method. Atoms in the shaded region will be excluded from the force computation of center atom *i*. Only forces exerted by atoms within the cutoff radius (represented by a two-headed arrow) are computed.

Figure 1(a) shows a schematic of the computational kernel of MD, which employs a linked-list cell method to compute interatomic interactions in *O(N)* time. Periodic boundary condition is applied to the system in three Cartesian dimensions. Here, a simulation domain is divided into small rectangular cells, and the linked-list data structure is used to organize atomic data (*e.g.* coordinates, velocities and atom type) in each cell. By traversing the linked list, one retrieves the information of all atoms belonging to a cell, and thereby computes interatomic interactions. Thus, the cutoff radius R_c of the interatomic interaction usually determines the dimensions of the cells.

In order to explain the computational procedures in the linked-list cell method, let us define S, which is a set of all atomic pairs, (i, j), within the nearest-neighbor cells. The first phase of the computation, *i.e.* pair search, constructs a subset, S_0, of pairs (i, j) whose distance r_{ij} satisfies $r_{ij} \leq R_c$. Subsequently, second phase computes the forces between all atom pairs within S_0. Figure 2 shows the pseudocode for pair search and force computation.

```
Phase 1: Pair search
S0 = ∅
for ∀(i,j) ∈ S do
    if r_ij ≤ R_c do
        S0 = S0 ∪ (i,j)
    end
end
Phase 2: Force computation
for ∀(i,j) ∈ S0 do
    compute interaction force f_ij(r_ij)
end
```

Figure 2: Force computation algorithm

The reduced linked-list cell method attempts to reduce the search space S by decreasing the dimension R_s of the linked-list cell. To illustrate this, Fig. 1(b) shows the situation where R_s is reduced by half. The atoms in the shaded region are at least one cutoff distance away from the center atom i such that it can be safely excluded. Hence, less number of pairs needs to be evaluated, potentially resulting in the performance increase. However, this performance gain could be offset by the increased number of linked-list cells that need to be scanned (*e.g.*, from 9 to 25 cells in the 2D example) and the associated computational overhead for processing the increased number of linked lists.

The MD program used in this paper is parallelized based on spatial decomposition and message passing [5]. However, the focus of this paper is performance optimization within each compute node, and thus the number of atoms N hereafter denotes the number of atoms per node.

3. Performance Analysis of Non-Bonded Interactions

Computing non-bonded interactions usually consumes the largest portion of time in MD simulation [4]. In our performance model, we assume that the computational cost consists of three terms: (1) pair search where the number of search operations $P_s = |S|$, *i.e.* the cardinality of set S (phase 1 in Fig. 2); (2) force computation for which the number of force computation operations $P_c = |S_0|$ (phase 2 in Fig. 2); and (3) linked-list cells access overhead—P_l. Therefore, the non-bonded computation time in one MD step—T, is

$$T = T_s P_s + T_c P_c + T_m P_l, \quad (1)$$

where T_s, T_c, and T_m are the operation cost per P_s, P_c, and P_l, respectively.

Generally, the memory access cost (the third term in Eq. (1)) in the conventional cell decomposition method is ignored when compared to the pair-search and force-computation costs. However, in the reduced linked-list cell method, this overhead often becomes dominant, as we will see in the measurements in section 4.

In section 3.1, we derive P_s, P_c, and P_l for the conventional cell-decomposition method, while the extension of the model to the reduced cell method is derived in section 3.2. The operation costs T_s, T_c, and T_m are obtained from regression described in section 4.

3.1 Non-bonded interaction model of the conventional linked-list cell method

3.1.1 Pair search

The search space for non-bonded force computations is defined as the number of pairs that need to be evaluated in phase 1 of Fig. 2. For N atoms, the search space of an all-pair method is $O(N^2)$. In this section, we derive the approximation of the $O(N)$ search space in the linked-list cell method.

For the uniform cell decomposition method with cubic cell dimension for all cells, the volume of each cell is

$$V_c = {R_s}^3, \tag{2}$$

where R_s is the dimension of the linked-list cell such that $R_s \geq R_c$. If the number density of atoms in the system is ρ atom/Å^3, the average number of atoms in one linked-list cell is

$$N_c = \rho {R_s}^3. \tag{3}$$

The number of possible non-bonded pairs between atoms in two cells is then

$$P_{cc} = \frac{1}{2} N_c \left(N_c - 1\right) \approx \frac{1}{2} {N_c}^2 = \frac{1}{2} \rho^2 {R_s}^6. \tag{4}$$

Let C_{ijk} be a linked-list cell with cell indices i, j, and k in the x, y, and z directions, respectively. To compute forces exerted to all atoms in C_{ijk}, all atom-pair combinations between C_{ijk} and its nearest neighbor cells $C'_{i'j'k'}$ need to be evaluated, where

$$S_{neighbor}(C_{ijk}) = \{C'_{i'j'k'} \,|\, (i-1 \leq i' \leq i+1) \wedge (j-1 \leq j' \leq j+1) \wedge (k-1 \leq k' \leq k+1)\} \tag{5}$$

is the set of neighbor cells. Since the conventional cell decomposition method evaluates all atomic pairs between C_{ijk} and its 26+1 neighbor cells (including itself), the number of pairs to be evaluated per one linked-list cell is

$$P_s(C) = \frac{1}{2} \rho^2 {R_s}^6 N_{neighbor}, \tag{6}$$

where $N_{\text{neighbor}} = |S_{\text{neighbor}}| = 27$. Let L_x, L_y, and L_z be the number of cells in the x, y and z directions, respectively, then the total number of atom pairs to be evaluated in one MD step is

$$P_s = \sum_{i=1}^{L_x} \sum_{j=1}^{L_y} \sum_{k=1}^{L_z} P_s(C_{ijk}) = \frac{1}{2} \rho^2 {R_s}^6 L_x L_y L_z N_{neighbor}. \tag{7}$$

3.1.2 Force computation

After the interaction pairs have been evaluated, only the pairs that satisfy the evaluation criterion (*i.e.*, $r_{ij} \leq R_c$) undergo force computation. The number of force computations can be estimated as follows. First, we count the number of atoms that are located inside the sphere with radius R_c around each atom. The volume of such sphere is

$$V_{fc} = \frac{4}{3} \pi {R_c}^3, \tag{8}$$

and the number of atoms that reside within V_{fc} is

$$N_{fc} = \frac{4}{3} \pi \rho {R_c}^3. \tag{9}$$

Let N denote the number of atoms in the system. From Eq. (3), N can be expressed as

$$N = \rho {R_s}^3 L_x L_y L_z. \tag{10}$$

Combining Eqs. (9) and (10), the number of force computation operations—P_c is

$$P_c = N_{fc} N = \frac{4}{3} \pi \rho^2 {R_s}^3 {R_c}^3 L_x L_y L_z. \tag{11}$$

The number of pair-force computations in Eq. (11) can be halved by applying Newton's 3rd law (*i.e.*, the forces acting on a pair of atoms are equal and in the opposite directions) such that

$$P_c = \frac{2}{3} \pi \rho^2 {R_s}^3 {R_c}^3 L_x L_y L_z. \tag{12}$$

3.1.3 Linked-list cell access

Linked-list cell access cost refers to the overhead introduced by the cell decomposition scheme. To compute the forces on atoms in cell C, N_{neighbor} cells

need to be traversed. Therefore, the number of linked-list cell accesses is

$$P_l = L_x L_y L_z N_{neighbor}. \quad (13)$$

Combining Eqs. (7), (12) and (13), the prediction model for non-bonded pair computation is given by

$$\begin{aligned} T = & T_s\left(\frac{1}{2}\rho^2 R_s{}^6 L_x L_y L_z N_{neighbor}\right) \\ & + T_c\left(\frac{2}{3}\pi\rho^2 R_s{}^3 R_c{}^3 L_x L_y L_z\right) \\ & + T_m\left(L_x L_y L_z N_{neighbor}\right) \end{aligned}. \quad (14)$$

In the next subsection, we derive a corresponding model for the reduced cell method to be used to obtain the optimal cell dimension.

3.2 Non-bonded interaction model of the reduced linked-list cell method

3.2.1 Derivation

In this subsection, we derive the number of searched pairs P_s', that of pair-force computations P_c', and that of linked-list accesses P_l', when the cell dimension R_s is reduced to

$$R_s' = \frac{R_s}{\mu}, \quad (15)$$

where $\mu \in \mathbf{N}$ is a cell dimension reduction factor. The cell dimension reduction also modifies the number of linked-list cells such that

$$\begin{aligned} L_x' &= \mu L_x \\ L_y' &= \mu L_y \\ L_z' &= \mu L_z \end{aligned}. \quad (16)$$

In the reduced cell method, the 1st-nearest neighbor cells are not sufficient for interaction evaluations, and instead up to the μ^{th}-nearest neighbor cells need to be evaluated. Therefore, $N_{neighbor}$ can be calculated as

$$N_{neighbor} = (2\mu + 1)^3. \quad (17)$$

Substituting R_s', L_x', L_y', L_z', and $N_{neighbor}$ to Eqs. (7), (12), and (13), respectively, yield

$$\begin{aligned} P_s' &= \frac{1}{2}\rho^2 R_s'^6 L_x' L_y' L_z' N_{neighbor} \\ &= \frac{1}{2}\rho^2\left(\frac{R_s}{\mu}\right)^6 \mu^3 L_x L_y L_z (2\mu+1)^3, \\ &= \frac{(2\mu+1)^3}{2\mu^3}\rho^2 R_s{}^6 L_x L_y L_z \end{aligned} \quad (18)$$

$$\begin{aligned} P_c' &= \frac{2}{3}\pi\rho^2 R_s'^3 R_c{}^3 L_x' L_y' L_z' \\ &= \frac{2}{3}\pi\rho^2\left(\frac{R_s}{\mu}\right)^3 R_c{}^3 \mu^3 L_x L_y L_z, \\ &= \frac{2}{3}\pi\rho^2 R_s{}^3 R_c{}^3 L_x L_y L_z = P_c \end{aligned} \quad (19)$$

$$\begin{aligned} P_l' &= L_x' L_y' L_z' N_{neighbor} \\ &= \mu^3 (2\mu+1)^3 L_x L_y L_z \end{aligned}. \quad (20)$$

Equation (19) indicates that P_c' is independent of the reduction factor μ. This is consistent with the requirement that the choice of cell dimension should not affect the simulation results as noted by Heinz *et al.* [10].

3.2.2 Asymptotic analysis of the number of key operations

In this section, we analyze the relation of the reduction factor μ to P_s', P_c', and P_l'. This can be done by the expansion of Eqs. (18) and (20) in the power of μ:

$$P_s' = \left(\frac{1}{2}\mu^{-3} + 3\mu^{-2} + 6\mu^{-1} + 4\right)\rho^2 R_s{}^6 L_x L_y L_z, \quad (21)$$

$$P_l' = \left(8\mu^6 + 12\mu^5 + 6\mu^4 + \mu^3\right) L_x L_y L_z, \quad (22)$$

while P_c' is independent of μ. Equations (21) and (22) provide asymptotic behaviors of P_s' and P_l' with respect to μ as summarized in Table 1. (The constant term of P_s', which is irrelevant for the determination of optimal μ, is omitted in Table 1.) Since P_s' and P_l' are descending and ascending functions of μ, respectively, the optimal μ value that minimizes the computation time is determined by their trade-off. The asymptotic analysis reveals that the increase of the number of linked-list cell accesses P_l' with μ is significantly larger than the reduction of search space P_s', while P_c' is not affected by μ. This implies that asymptotically, the overhead cost from linked-list access overwhelms the benefit due to the reduced the search space for large μ. This agrees with finding by Mattson *et al.* [7] and Yao *et al.* [8] such that the computation time is minimal when μ is small.

From Eq. (1), the costs of pair search T_s and that of linked-list access T_m are essential parameters to quantify the trade-off between P_s and P_l, so that the optimal value of μ can be determined as described in section 4.

Table 1: Asymptotic complexity of parameter P_s', P_c', and P_l' over μ.

Operation count	Scaling with reduction factor μ
P_s'	$O\left(\mu^{-3}\right)$
P_c'	$O(1)$
P_l'	$O\left(\mu^{6}\right)$

4. Model Verification

As derived in section 3, the computation time for non-bonded interactions in the reduced linked-list cell method is given by

$$\begin{aligned} T &= T_s P_s' + T_c P_c' + T_m P_l' \\ &= T_s\left(\frac{(2\mu+1)^3}{2\mu^3}\rho^2 {R_s}^6 L_x L_y L_z\right) \\ &+ T_c\left(\frac{2}{3}\pi\rho^2 {R_s}^3 {R_c}^3 L_x L_y L_z\right) \\ &+ T_m\left(\mu^3(2\mu+1)^3 L_x L_y L_z\right) \end{aligned}. \quad (23)$$

In section 4.1, we validate the accuracy of the operation-counts models—P_s', P_c', and P_l'—by comparing them with measured values. In section 4.2, we fit the P_s', P_c', and P_l' derived from the estimation, Eq. (23), to obtain the operation costs—T_s, T_c, and T_m.

4.1 Validation of the operation counts model

MD simulations of silica at temperature 300 K involving up to 192,000 atoms are performed on an Intel Core-i7 2.66 GHz machine, using the cell reduction factor μ ranging from 1 to 4. The cutoff radius R_c of this dataset is set to 5.5 Å with atom density ρ = 0.0654 atom/Å^3, and the linked-list cell size R_s = 5.728 Å.

Under this condition, the actual values of P_s', P_c', and P_l' measured during the simulations are compared with those obtained from the model; see Table 2. The relative error of P_s' and P_l' is less than 4% in all cases, confirming the applicability of our model for these parameters. For P_c', the relative error is ~ 10% for the largest number of atoms (N = 192,000), while it is ~ 8% for the smallest system (N = 1,536). Nevertheless, the maximum error of 10% is acceptable for the subsequent analysis, especially in the light of the highly dynamic nature of MD simulations.

Table 2: Accuracy of the operation counts model.

Number of atoms	Reduction factor, μ	Relative error (%)		
		P_s'	P_c'	P_l'
1,536	1	1.67	8.01	3.60
	2	1.31	8.01	0.70
	3	0.96	8.01	0.19
	4	0.95	8.01	0.04
12,288	1	0.79	-0.51	3.50
	2	0.76	-0.51	0.60
	3	0.49	-0.51	0.09
	4	0.39	-0.51	-0.06
41,472	1	-0.24	-2.28	2.68
	2	-0.21	-2.28	-0.20
	3	-0.40	-2.28	-0.70
	4	-0.31	-2.28	-0.85
98,304	1	-1.43	-3.29	1.67
	2	-1.35	-3.29	-1.18
	3	-1.42	-3.29	-1.67
	4	-1.59	-3.29	-1.83
192,000	1	-3.50	-10.59	-0.28
	2	-3.29	-10.59	-3.08
	3	-3.39	-10.59	-3.57
	4	-3.38	-10.59	-3.71

4.2 Parameter Fitting

In section 4.1, we confirm that Eq. (23) accurately approximates the operation counts. In this section, we use the prediction from the model and the running time from the execution of actual MD simulation to find the cost parameters T_s, T_c, and T_m. Here, we need to use more datasets in order to get better statistics for the fitting. Since R_c is one of the most important variables, we here concentrate on the effect of R_c on the performance.

We run two additional sets of MD simulations using the same setting as in section 4.1 except for R_c, which is set here to 6.6 and 10 Å. Using the measured timings, we perform least square regression to obtain T_s, T_c, and T_m from Eq. (23). However, the results obtained from the least square regression show very large error up to 93% for estimated T_c; see Table 3. This may be explained by the two orders-of-magnitude difference between the largest and smallest dataset sizes (1,536 and 192,000 atoms), which causes the fitting residuals of larger datasets to dominate the significance of smaller datasets. To alleviate this artifact, a more robust regression algorithms, Huber regression and bisquare-weighted regression, are used instead of the least square

regression [11, 12]. Table 3 shows that the estimated operation costs from Huber regression have smaller standard errors (at most 40.8%), while those from bisquare-weighted regression have the least errors (at most 22.5%). Therefore, we use the estimated costs from bisquare-weighted regression to predict the performance and thereby obtain the optimal reduction factor μ in the next section.

Table 3: Estimated operation costs from various regression algorithms.

Regression algorithm	Operation cost	Estimated cost (ns)	Standard error (%)
Least square	T_s	9.76	40.2
	T_c	34.83	93.1
	T_m	10.44	7.6
Bisquare	T_s	6.48	22.2
c = 1.865	T_c	52.89	22.5
	T_m	11.80	2.5
Huber	T_s	10.77	14.9
c = 1.345	T_c	32.56	40.8
	T_m	10.81	3.0

5. Performance Optimization Results

Here, we use the estimated operations counts and costs in the previous section to predict the performance of the simulation when the cell size is reduced. The optimal reduction factor μ* is then determined as the one that minimizes the predicted computation time:

$$\mu^* = \arg\min\left(T(\mu)\right). \quad (24)$$

To study the computation time as a function of μ from the prediction, we compare the predicted and measured performances of the system with R_c = 5.5, 6.6, 10.0 Å, with the number of atoms ranging from 1,536, 12,288, 41,472, 98,304, to 192,000. Figure 3 shows that the prediction captures the essential behavior of the non-bonded computation time in the actual simulations as a function of μ. This result agrees with the asymptotic characteristics of μ in Table 1, where the linked-list cell access cost dominates for larger μ.

Table 4 compares μ* obtained from the prediction, Eq. (23), with those obtained from actual measurements. The μ* for R_c = 5.5 Å from actual simulation dataset are all 1, similar to the predicted values. Namely, reducing the cell dimension does not improve the performance in this case. On the other hand, the predicted μ* of the system with R_c = 6.6 and 10.0 Å is 2 in almost all cases with the average speedup of 1.10 and 1.52, respectively. This result indicates that the reduction of the cell dimension becomes more beneficial for larger R_c. In such cases, the pair search space is larger and the associated search cost P_s' becomes more important relative to the list-processing overhead P_l'. Figure 4 shows the speedup achieved from our optimization compared to the actual optimal speedup.

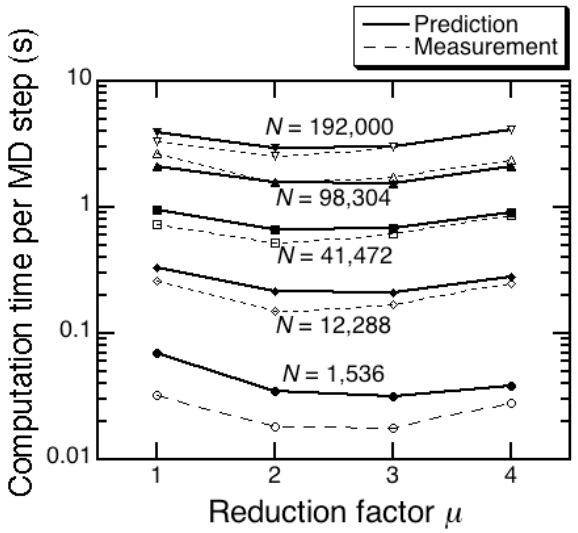

Figure 3: The running time from the prediction and the measurement for R_c = 10 Å. The numerals are the number of atoms.

Table 4: The predicted versus measured values of the optimal μ with their corresponding speedup compared to the case of no cell dimension reduction.

R_c (Å)	N	μ*		Speedup	
		Prediction	Actual	Prediction	Actual
5.5	1,536	1	1	1.00	1.00
	12,288	1	1	1.00	1.00
	41,472	1	1	1.00	1.00
	98,304	1	1	1.00	1.00
	192,000	1	1	1.00	1.00
6.6	1,536	2	2	1.12	1.12
	12,288	2	2	1.28	1.28
	41,472	2	1	0.92	1.00
	98,304	2	2	1.15	1.15
	192,000	2	2	1.02	1.02
10.0	1,536	3	3	1.85	1.85
	12,288	3	2	1.54	1.74
	41,472	2	2	1.41	1.41
	98,304	3	2	1.54	1.70
	192,000	2	2	1.28	1.28

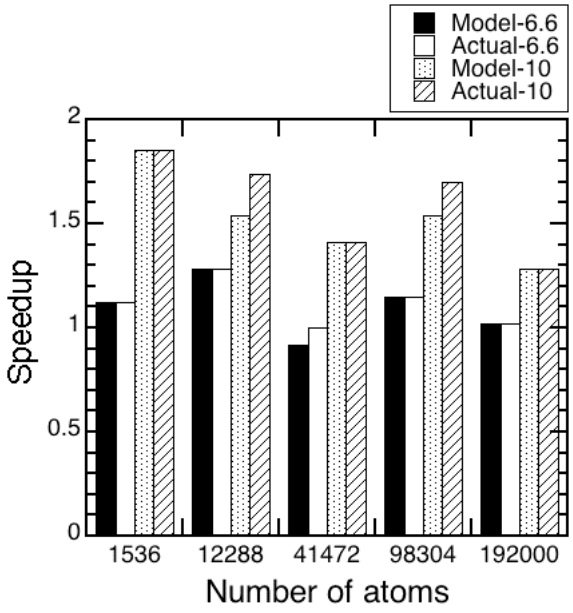

Figure 4: Comparison of speedup gained from predicted and actual μ^* for R_c = 6.6 and 10 Å.

6. Conclusions

We have developed a performance prediction model of non-bonded interaction computations of molecular dynamics simulations, which can be used to obtain the optimal cell dimension in the reduced linked-list cell algorithm. The model has been validated against measurement results. The model accurately predicts the optimal reduction factor, and average speedup of 1.10 and 1.52 are obtained for R_c = 6.6 and 10.0 Å datasets, respectively. Our analysis reveals that the optimal reduction factor is determined by a trade-off between decreasing search space and increasing linked-list cell access for smaller cells. Future research will explore the possibility of using the model to compare other MD algorithms. Other factors that affect the selection of the reduction factor will also be explored. This work was partially supported by NSF-ITR/PetaApps/EMT/CRI, DOE-SciDAC/BES/EFRC, and DTRA.

References

[1] J. C. Phillips, *et al.*, "NAMD: Biomolecular simulations on thousands of processors," in *Supercomputing*, Los Alamitos, CA, 2002.

[2] D. E. Shaw, "A fast, scalable method for the parallel evaluation of distance-limited pairwise particle interactions," *Journal of Computational Chemistry,* vol. 26, pp. 1318-1328, Oct 2005.

[3] K. J. Bowers, *et al.*, "Zonal methods for the parallel execution of range-limited N-body simulations," *Journal of Computational Physics,* vol. 221, pp. 303-329, Jan 20 2007.

[4] D. E. Shaw, *et al.*, "Anton, a special-purpose machine for molecular dynamics simulation," *Communications of the ACM,* vol. 51, pp. 91-97, Jul 2008.

[5] K. Nomura, *et al.*, "A metascalable computing framework for large spatiotemporal-scale atomistic simulations," *Proceedings of the 2009 International Parallel and Distributed Processing Symposium,* IEEE, 2009.

[6] The spatiotemporal scale covered by MD simulation on a sustained petaflops computer per day (*i.e.* petaflops•day of computing) is limited to $NT = 2.14$ (*e.g.* $N = 2.14$ billion atoms for $T = 1$ nanoseconds); see Ref. [5].

[7] W. Mattson and B. M. Rice, "Near-neighbor calculations using a modified cell-linked list method," *Computer Physics Communications,* vol. 119, pp. 135-148, Jun 1999.

[8] Z. H. Yao, *et al.*, "Improved neighbor list algorithm in molecular simulations using cell decomposition and data sorting method," *Computer Physics Communications,* vol. 161, pp. 27-35, Aug 1 2004.

[9] C. I. Rodrigues, *et al.*, "GPU acceleration of cutoff pair potentials for molecular modeling applications," *Proceedings of the 5th Conference on Computing Frontiers*, Ischia, Italy, 2008.

[10] T. N. Heinz and P. H. Hunenberger, "A fast pairlist-construction algorithm for molecular simulations under periodic boundary conditions," *Journal of Computational Chemistry,* vol. 25, pp. 1474-1486, Sep 2004.

[11] P. J. Rousseeuw and A. M. Leroy, *Robust regression and outlier detection*: John Wiley & Sons, Inc., 1987.

[12] P. J. Huber, "Robust Estimation of Location Parameter," *Annals of Mathematical Statistics,* vol. 35, pp. 73-101, 1964.