REMD toolkit : a composable software to generate parallelized simulation programs

Masakatsu Ito,[1] and Umpei Nagashima

Grid Technology Research Center, National Institute of Advanced Industrial Science and Technology, AIST Tsukuba Central 2, Tsukuba, 305–8568 JAPAN;

ABSTRACT

We developed a toolkit to generate a replica exchange method program which is suitable to solve the multiple-minima problem that prevents the accurate estimation of thermodynamical quantities. The toolkit was designed as a set of software components, so that any new variant of simulation program can be built by assembling suitable components. They are categorized according to three types of functions : parallelization of simulation programs, selection of sampling algorithms, and incorporation of an arbitrary force field implementation into the program.

The extensibility and efficiency of the toolkit was demonstrated by generating a new variant of replica exchange method program which implements CHARMM force field. It was shown that the replica exchange scheme not only reduces the total computational cost with the increase in the number of replicas but achieves almost linear–speedup with the number of CPUs.

Keywords: replica–exchange; molecular dynamics; object–oriented framework; parallel

1 Introduction

Recently, replica–exchange method (REM)[1] and replica–exchange molecular dynamics (REMD)[2] have gathered attention as ways to mitigate "subverted ergodicity" which presents challenges to Monte Carlo (MC) and molecular dynamics (MD) simulations in estimating thermodynamical properties. Because subverted ergodicity occurs at low temperature, REM and REMD take advantage of the ergodicity present at higher temperatures by exchanging the temperatures of molecular replicas.

Moreover these algorithms has theoretical composability where a simulation consists of multiple components such as molecular force field, statistical ensemble, and extended ensemble, so that a suitable combination of components can be chosen to solve a particular problem. In fact, various potential energy functions have been combined with REM, such as ab–initio molecular orbital (MO) method for the geometry optimization of a lithium cluster[3], knowledge–based scoring function for describing receptor–ligand binding[4], and ordinary molecular force fields such as AMBER[6] and CHARMM[7] for protein folding.

This composability, however, is not available in most simulation programs. Although there are state-of-the-art simulation packages that implement MO, MC, or MD algorithms, they are designed as monothilic applications rather than as composable software systems. Thus the applicability of REM and REMD simulations has been technically limited despite their advantages.

To resolve this technical problem, we have developed a C++ toolkit[5] designed as a set of software components (Figure 1). Each component has an object–oriented interface like a "socket" into which other components can plug, so that any new variant of simulation program can be built by assembling suitable components. Added to ordinary MC and MD simulations, our REMD toolkit can generate both REM and REMD algorithms to overcome the subverted ergodicity and to accelerate the estimation of thermodynamical properties.

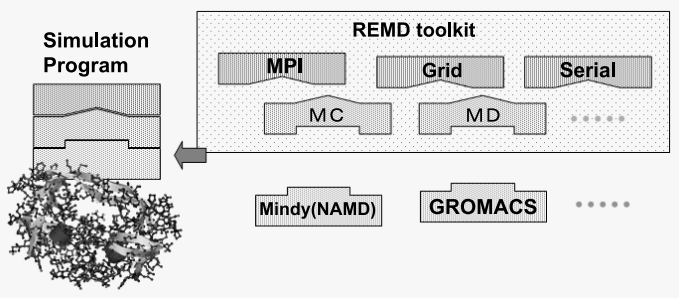

Figure 1: A toolkit was developed as a set of software components implementing parallelized REM and REMD algorithms and was then used as a software framework to generate variants of simulation programs by assembling the toolkit and force field components.

2 Object–oriented framework to customize and parallelize simulations

2.1 Component-based software design

The REMD toolkit is designed as an object–oriented framework to incorporate various implementation of force fields, so that REMD and REM simulations can be customized to suit different problems by combining with various force fields, such as a problem–specific modeling function, a general purpose force field, or an energy and force vector based on MO calculations.

[1]Current affiliation : Fujitsu Laboratories Ltd., 10-1 Morinosato–Wakamiya, Atsugi, 243–0197 JAPAN

A framework is not generally an executable, but rather helps one to generate various executables. One can customize it to a particular executable by creating problem–specific subclasses of abstract classes from it.

The REMD toolkit consists of three components to maintain extensibility; otherwise the dependency between classes prevents the toolkit from being added new functionality. Each component is related to the three important theoretical concepts in the simulation method: (1) extended ensemble component, (2) ensemble component, which is related to statistical ensembles except an extended ensemble, and (3) microscopic modeling component, which is related to models for calculating molecular potential energies and force vectors.

2.2 Parallelization of simulations by a replica–exchange scheme

Figure 2 shows an extended ensemble component which performs a replica–exchange scheme in a parallelized simulation. An *ExtendedEnsemble* is located in a master simulator object, and statistical ensemble objects are distributed among worker simulator objects. An actual simulation is realized by performing the following two steps alternately.

1. Each instance of a subclass of *AbstEnsemble* performs simultaneously and independently an MC or MD calculation for given steps and sends the resultant energy value to the *ExtendedEnsemble* object.
2. *ExtendedEnsemble* collects energies, exchanges neighboring temperatures stochastically and sends back new temperatures to the ensemble objects.

AbstEnsemble is the abstract class for statistical ensembles whose subclasses, *StepEnsemble* and *DynamicalEnsemble*, implement MC and MD algorithms, respectively. The replica–exchange scheme can be customized by selecting these two subclasses. A *StepEnsemble* object is instantiated for an REM simulation, whereas a *DynamicalEnsemble* object is instantiated for an REMD simulation.

The message between ensemble objects and *ExtendedEnsemble* is transferred through the *StatManager* object, which also manages the estimation of thermodynamical properties. To reduce the overhead in the parallelized simulation, the amount of communication among objects is minimized by exchanging the temperatures instead of the configurations.

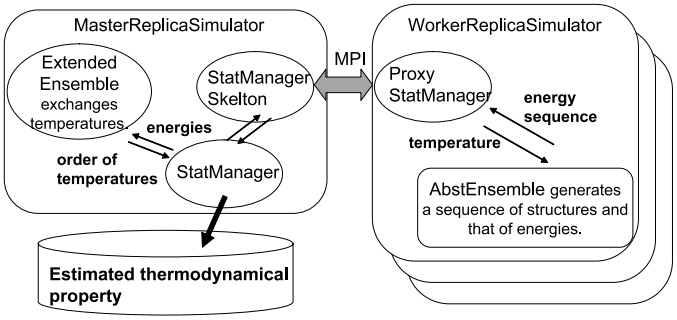

Figure 2: The composite structure of instance objects created in an extended ensemble component

2.3 Incorporation of an arbitrary force field

The microscopic modeling component is responsible for the implementation of force fields for MD calculation (and potential energy functions for MC calculation), and thus contains the code for calculating the energy and gradient vector of a force field. This component emphasizes an aspect of software frameworks that variants of simulation programs can be created from the framework. REMD simulation programs are generated by implementing force-field-specific subclasses from *AbstDynamicalWalker*, whereas REM simulation programs are generated by creating subclasses that implement an energy function for MC calculation.

The following code is the interface defined by *AbstDynamicalWalker*, whose subclass should be implemented.

```
class AbstDynamicalWalker {
 public:
  AbstDynamicalWalker();
  virtual ~AbstDynamicalWalker() {}

  virtual double map(); // potential energy

  virtual
  void walk(AbstThermostatPropagator& prop);
  // generate new molecular structure

  // other methods and instance variables
};
```

The constructor of the subclass should be implemented to initialize instance variables representing a molecular structure. *map()* method should calculate potential energy and force vectors, and *walk()* method should integrate equations of motion to generate new structure.

A force–field–specific subclass can be derived either by writing all the code to create a new microscopic model, or by using part of an existing simulation package.

3 Efficiency of the REMD toolkit with a molecular force field

The efficiency of the REMD toolkit was examined for a system of decaalanine $(Ala)_{10}$. To use a realistic force field such as CHARMM[7] and AMBER[6], customized simulation programs were built by plugging the sequential version of NAMD[8] (Mindy) into the REMD toolkit. To adapt the toolkit to the NAMD package, a force–field–specific subclass was derived from the *AbstDynamicalWalker* class. The subclass itself does not implement the CHARMM force field, but delegates the force field calculation to NAMD objects. This extension of the force field implementation was quite easy and simply equipped into the toolkit.

The force field parameters of CHARMM19 were used in the simulation. We used 32 replicas, and their temperatures were exponentially distributed as given by

$$T_i = T_L \times \left(\frac{T_H}{T_L}\right)^{\frac{i}{N-1}}, i = 0, \ldots, N-1 \qquad (1)$$

where $T_L = 174.12\,\mathrm{K}$ and $T_H = 800\,\mathrm{K}$. Before taking the data for analysis, 5×10^5 steps for each replica were taken for the relaxation where the replica–exchange was tried every 50 steps. Then, 2×10^7 steps were taken for each replica and a replica–exchange was tried every 200 steps. The MD time step size was set to 0.3 fs.

To examine the convergence of the energy distribution in the REMD simulation, the heat capacity $C_v(T)$ of $(\mathrm{Ala})_{10}$ is plotted in Figure 3 at several simulation steps. The convergence is considerably slow. The shape of the $C_v(T)$ curve changes until the total steps exceed 5.1×10^8, after which the curve remains relatively unchanged. Therefore, we used the curve at the final step to estimate the error of other simulations as described in the following subsection.

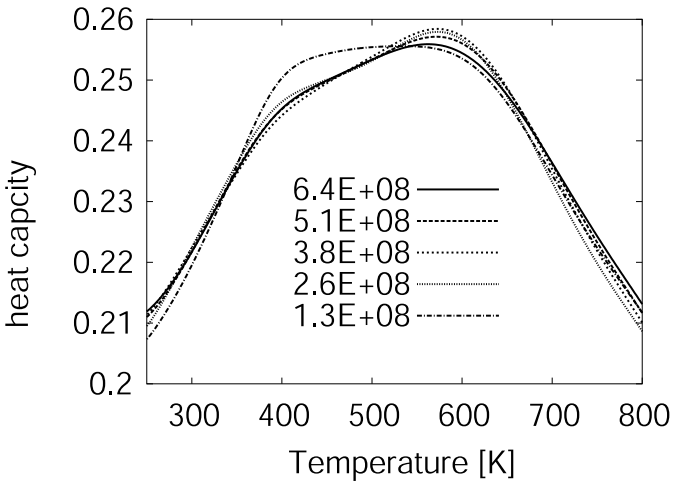

Figure 3: Heat capacity as a function of temperature, $C_v(T)$, obtained from the REMD simulation of 32 replicas ($N = 32$). Different lines correspond to different total number of sampled conformations.

The temperature region below 500 K shows relatively fast convergence; in this region, the curve remains relatively unchanged after 2.6×10^8 steps. This region is not affected by the multi–minima problem in which a molecule, such as a peptide, at lower temperature can easily get trapped in the local minima of the potential energy surface. Two regions, however, show comparatively large deviations until the simulation steps reach 5.1×10^8, possibly due to the slow diffusion over the configurational space. One region is located at the peak of $C_v(T)$ around 580 K. This peak might be related to the helix–coil transition reported in several numerical studies [9]–[11]. This pseudocritical behavior implies an extremely long relaxation time of peptide dynamics around the critical temperature. This slow relaxation might prevent the decaalanine from approaching equilibrium, thus slowing the convergence of the heat capacity near the critical temperature. The other region is located near a higher temperature, 700 K. Because the frequency of the structure exchange trial at the highest temperature is half that at lower temperatures, the replica of decaalanine does not rapidly equilibrate. Because the simulation therefore cannot fully explore the configurational space, estimation of the heat capacity remained incorrect at the earlier simulation steps.

3.1 Performance evaluation

To estimate the dependence of the required steps on N, we further performed 7 simulations, one each for $N = 3$, 4, 6, 8, 11, 16, and 23, where N was based on an exponential increase.

After the preliminary run, we performed each simulation until the total number of simulation steps reached 10^8 (or 30 ns), and thus the number of simulation steps for each replica was $10^8/N$. The other simulation conditions such as the MD time step size were the same as those of the simulation in the previous subsection. The temperatures of the replicas were exponentially distributed as given by Eq.1.

Thus, the number of replicas, N, affects the effectiveness of the simulation. Figure 4 shows the total number of simulation steps when the heat capacity error $\delta C_v(N_{\mathrm{conf}})$ first becomes smaller than $0.01\mathrm{kcal/mol \cdot K}$. Although the simulations for $N = 3$ and 4 were not able to reduce the error to that value, for the other simulations ($N = 6, 8, 11, 16, 23$), the number of required conformations tended to decrease with increasing N. The number of total required conformations steeply decreased with increasing N from 6 to 11, but then remained relatively constant with further increase in N. Despite this saturation at N above 11, parallelization can further reduce the simulation steps required for estimating the thermodynamical quantities. Because the simulation consisting of N replicas can be easily performed in parallel by N CPUs, the number of required steps can be divided further by N.

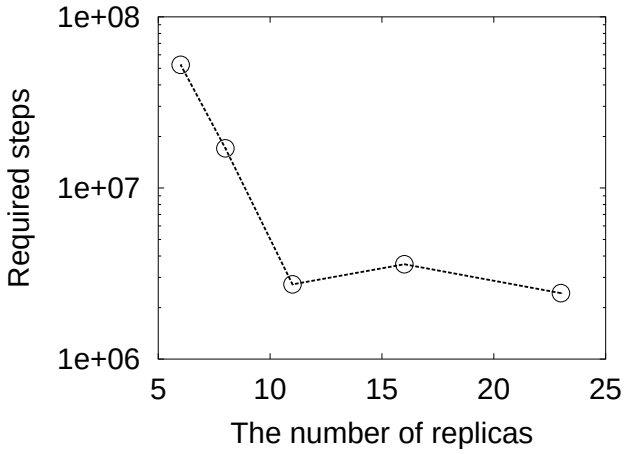

Figure 4: The number of simulation steps is plotted as a function of the number of replicas. The number of steps is required to decrease the heat capacity error less than $0.01\mathrm{kcal/mol \cdot K}$.

To examine the parallel efficiency of the REMD toolkit, an REMD simulation consisting of 32 replicas was performed using different numbers of CPUs. We used a PC Linux cluster consisting of 32 nodes, each having dual Pentium III 1400MHz and 2.3GB memory. The number of CPUs was increased from 1 to 32 (i.e., $1, 2, 4, \ldots$ and 32), so that the number of replicas placed on each CPU was correspondingly decreased from 8 to 1. The rate increased almost linearly

as the number of CPUs was increased; the rate of the parallelized simulation with 8 CPUs was 32×0.97 times larger than that of the serial simulation. REM and REMD algorithms are known to be suitable for parallelization, because the communication cost between replicas is quite small compared with the computational cost for the force field calculation in each replica.

4 Summary and Future Direction

We developed a C++ toolkit to parallelize molecular simulations by using a replica–exchange scheme, which enables the simulation to generate the canonical distribution at any temperature, even at low or transition temperatures where MC and MD simulations fail. The toolkit is designed as an object–oriented framework consisting of software components, so that the replica–exchange scheme can be customized to generate various simulation algorithms,

Simulations consisting of different numbers of replicas N were performed until the heat capacity error becomes smaller than $0.01\text{kcal/mol} \cdot \text{K}$, and it was found that the total number of simulation steps drastically decreased with increasing N from 6 to 11. Although the number remained relatively constant when N was further increased, parallelization can further reduce the computational time. The sampling rate of the parallelized simulation was found to increase almost linearly with the number of CPUs, up to N. This implies that computational time for the parallelized simulation can be reduced in proportion to $1/N$.

Added to Mindy (the serial version of NAMD[8]), we have adapted the toolkit to GROMACS[12] package which implements efficient algorithms for force field calculation. This extension is now applying to more realistic systems such as a receptor–inhibitor complex.

The REMD toolkit is being released under GNU General Public License.

Acknowledgments This research is supported by "Research and Development for Applying Advanced Computational Science and Technology" of Japan Science and Technology Corporation.

REFERENCES

[1] K. Hukushima, K. Nemoto, J. Phys. Soc. Jpn. **65**, 1604 (1996).
[2] Y. Sugita and Y. Okamoto, Chem. Phys. Lett. **314**, 141 (1999).
[3] Ishikawa, Y.; Sugita, Y.; Nishikawa, T.; Okamoto, Y. Chem Phys Lett 2001, 333, 199.
[4] Vekhivker, G. M. et. al Chem Phys Lett 2001, 337, 181.
[5] M. Ito, T. Nishikawa, and U. Nagashima, J. Comp. Chem. , submitted.
[6] Weiner, P.K. Kollman, P. A. J Comp Chem 1981, 2, 287.
[7] Brooks, B. R.; Bruccoleri, R. E.; Olafson, B. D.; States, D. J.; Swaminathan, S.; Karplus, M. J Comp Chem 1983, 4, 187.
[8] K. Schulten et al. J. Comp. Phys. **151**, 283 (1999).
[9] Hansmann, U.H.; Okamoto, Y. J Chem Phys 1999, 110, 1267; J Chem Phys. 1999, 111, 1339.
[10] Alves, N.A.; Hansmann, U.H. Phys Rev Lett 2000, 84, 1836.
[11] Mitsutake, A.; Okamoto, Y. J Chem Phys 2000, 112, 10638.
[12] Berendsen, H.J.C., van der Spoel, D. and van Drunen, R., Comp. Phys. Comm. **91**, 45 (1995).