Protomol: An Object-oriented Component Molecular Dynamics Framework

T. Matthey, University of Bergen, Norway

Protomol is a collaboration work with J. A. Izaguirre, CSE, University of Notre Dame, USA.

NSF Grant BIO-0083653

Molecular Dynamics Framework
An Object-oriented Component

paralab
Overview

Introduction

The framework PROTOCOL: An Object-oriented Component Molecular Dynamics Framework

NAMD2 vs. PROTOCOL

Components of PROTOCOL

The framework PROTOCOL

Introduction

Conclusions & Future Work

TTP
1. Compute/evaluate forces for all particles

\[ \Delta T \frac{d}{dt} n \Delta - = \mathcal{F} \frac{\Delta}{\mathcal{T}} \]

2. Update positions & velocities

\[ \mathcal{H} \frac{\mathcal{E}}{\mathcal{F}} \leq 1 \Rightarrow \mathcal{H} = \frac{\mathcal{E}}{\mathcal{F}} \]

\[ m \frac{\mathcal{E}}{\mathcal{F}} \mathcal{H} = \frac{\mathcal{E}}{\mathcal{F}} \mathcal{H} = \frac{\mathcal{E}}{\mathcal{F}} \mathcal{H} \]

Newton's equation of motion •

Molecular Dynamics
Why a new Molecular Dynamics Framework?

No fully customizable integrators and forces

No real encapsulation of data distribution & parallelization

To extend with new algorithms (existing (highly optimized) frameworks are not easy to extend with new algorithms)

No real encapsulation of data distribution & parallelization
Object-oriented component framework, C++, 40,000 LOC

- Components for parallelization & data distribution
- DCD, CHARMM 19/28, ... and applications
- Interface to standard I/O formats (PDB, PSE, PAR, XYZ)
- Switching functions to smooth/split forces/potentials
- Generic forces and algorithms

- Integrator hierarchy, suited for multiple time stepping
- Integrator-oriented component framework, C++, 40,000 LOC

Protomol: An Object-oriented Component Molecular Dynamics Framework

© TTP, June 2001, T. Mathey
The framework ProtoMol II
Incremental Parallelization

- Extending to parallel version
- Gaining from already parallelized components
- Parallel and sequential part go hand in hand for verification
- Implementing sequential version of a new algorithm
SHAKE •
Langevin BBK •
Nose-Hoover •
(Lang/endpoint)
Leapfrog •

STS Integrators:

Hybrid Monte Carlo •
Mollied Impulse Method •
Verlet-I/IV-RESPA •

MTS Integrators:
Components II: Forcens/Potentials

Forcens/Potentials:
- Angle
- Bond
- Dihedral
- Haptic
- Improper
- Lennard-Jones
- PMEwald (Coulomb)
- Simple PMEwald (Coulomb)
- Simple Full
- Full
- Long
- Cutoff

Algorithms:
- Simple Full
- Full
- Long
- TTP, June 2001, T. Matthey
Switching Functions & Boundary Conditions

Switching Functions:
- $C_1$
- $C_2$
- Shift
- Complement
- Range
- Normal Vacuum
- Periodic
- Normal

Boundary Conditions:
- Normal Vacuum
- Periodic
- Normal

Switching Functions:
- $C_1$
- $C_2$
- Shift
- Complement
- Range
NAMD2 vs. Paratomi

- Does not scale as well as NAMD2 (< 16 CPUs)
- Easy to extend the framework
- Customizable switching functions
- Force algorithms associated to forces/potentials
- Forces associated to integrators
- Easy to use MTS integrator definition/strategy (iterate)
- Intuitive simulation configuration
Example: MTS Integrator

```
Integrator
      
level0PLeapfrog
                     
forceLennardJones
cutoff 6.5
-algorithm NonbondedCutoff
-switchingFunction FSWC1
-cutoff 6.5
-switchingFunction SMCT

forceBond, Angle
timestep 0.5

level 0 PLeapfrog

forceCommand

level 1 Impulse

forceCommand

temperature 5

warmupcycles 10

cumulative 20

level 1 Impulse

Integrator

boundaryConditions Normal

timestep 0.5

output 1

cellSize 6.5

nsteps 100

initial 0.0
```
TTP as one of the first major users
- Producing and publishing results with PROTOMOL
- Contributing new integrators/forces/algorithms
- Application developer/end user
- Use of PROTOMOL together with other software
Parallelization support

Implementation support

Contributing new features for TTP

User support
Conclusions

- We provide a flexible component framework

- Generic programming to customize components, with performance penalties

- Contains common potential/forces & integrators

- Starting with a sequential implementation

- Easy to extend to a parallel implementation

- Hiding parallelization in few objects

- CHARMM 19 and 28 force fields

PROTOCOL: An Object-oriented Component Molecular Dynamics Framework

T.T.P., June 2001, T. Matthey

PROTOCOL: An Object-oriented Component Molecular Dynamics Framework

T.T.P., June 2001, T. Matthey
Future Work

- Adding contributions from TTP
- GUI
- Providing components for generic fast electrostatic forces
- Multi-threading
- Generic load balancing algorithm
- Generic communication approaches