parallab

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

Overview

- Introduction
- The framework PROTOMOL
- Components of PROTOMOL
- NAMD2 vs. PROTOMOL
- TTP
- Conclusions & Future work

- Newton's equation of motion

$$m_j r_j'' = F_j = \sum_{i=1, i \neq j}^{N} F_{ij}$$

$$F_{ij} = -\nabla u(r_{ij})$$

1. Compute/evaluate forces for all particles
2. Update positions & velocities

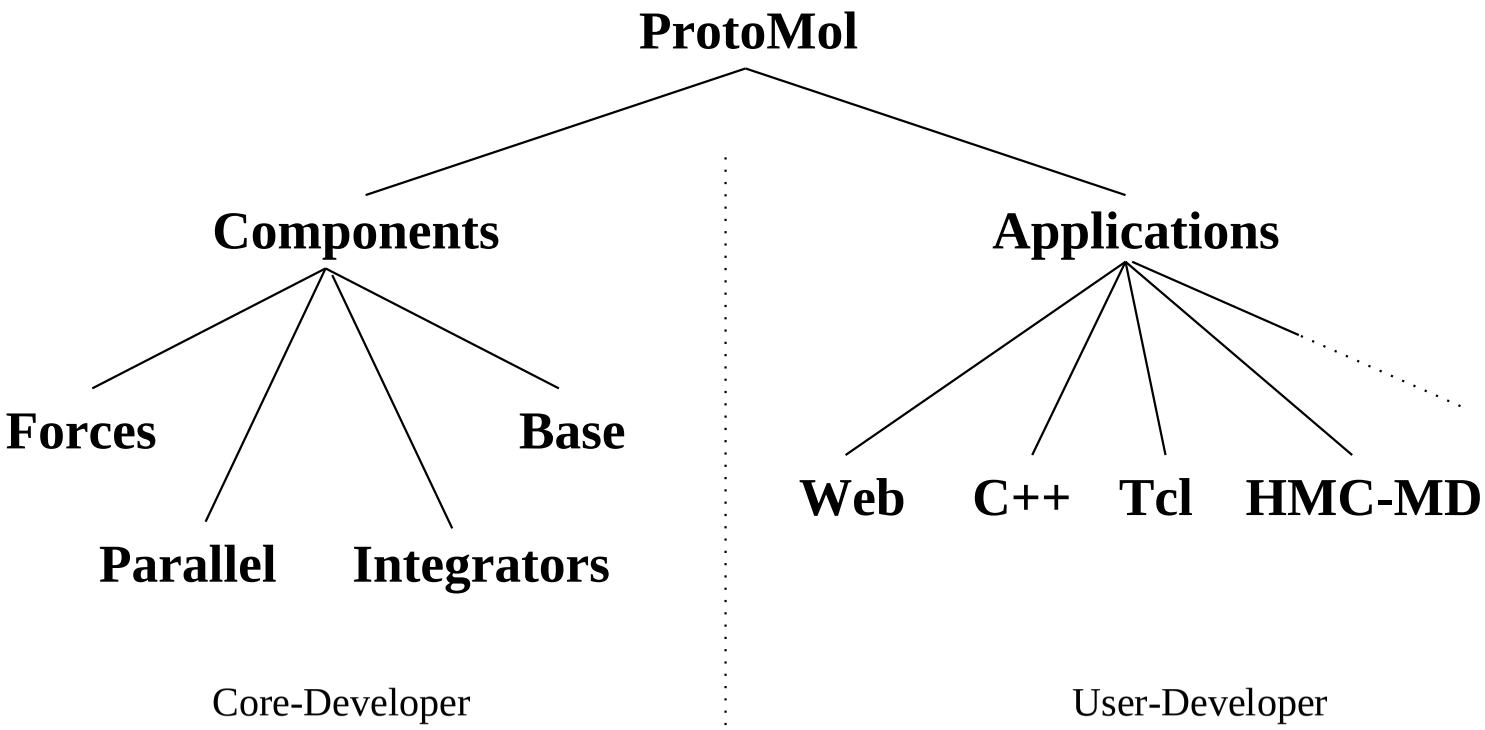
Why a new Molecular Dynamics Framework?

- Existing (highly optimized) frameworks are not easy to extend with new algorithms
- No real encapsulation of data distribution & parallelization
- No fully customizable integrators and forces

The framework PROTOMOL I

- Object-oriented component framework, C++, 40'000 LOC
- Integrator hierarchy, suited for multiple time stepping
- Generic forces and algorithms
- Switching functions to smooth/split forces/potentials
- Interface to standard I/O formats (PDB, PSF, PAR, XYZ, DCD, CHARMM 19/28, ...) and applications
- Components for parallelization & data distribution

The framework PROTOMOL II



Incremental Parallelization

- Implementing sequential version of a new algorithm, for verification
- Parallel and sequential part go hand in hand
- Gaining from already parallelized components
- Extending to parallel version

PROTOMOL Components I: Integrators

MTS Integrators:

- Verlet-I/r-RESPA
- Mollified Impulse Method
- Hybrid Monte Carlo

STS Integrators:

- Leapfrog (mid/endpoint)
- Nosé-Hoover
- Langevin BBK
- SHAKE

PROTOMOL Components II: Forces/Potentials

Forces/Potentials:

- Angle
- Bond
- Dihedral
- Haptic
- Improper
- LennardJones
- Coulomb
- FullEwald (Coulomb)
- PMEwald (Coulomb)

Algorithms:

- Cutoff
- SimpleFull
- Full
- Long

Switching Functions & Boundary Conditions

Switching Functions:

- C1
- C2
- Shift
- Complement
- Range

Boundary Conditions:

- Normal
- Periodic
- Normal Vacuum

NAMD2 vs. PROTOMOL I

+ Intuitive simulation configuration

  Easy to use MTS integrator definition/strategy (integrator definition language)

+ Forces associated to integrators

+ Force algorithms associated to forces/potentials

+ Customizable switching functions

+ Easy to extend the framework

– Does not scale as well as NAMD2 ($>$ 16 CPU's)

NAMD2 vs. PROTOMOL II

```
structure       equil298K_01.psf

parameters      equil298K_01.par
exclude         1-3
1-4scaling      1.0

switching       on
switchdist      0.1
cutoff          6.5
margin          0
stepspercycle   1

molly off

timestep 1.0

outputenergies  2000
outputtiming    2000
binaryoutput    no

coordinates     equil298K_01.pdb

outputname      equil298K_01.namd.output

temperature     300
numsteps        2000
```

```
initial_t 0.0
nsteps 2000
cell_size 6.5
outputf 2000
init_temp 300

boundaryConditions Normal
cellManager Cubic

Integrator {
  level 0 Leapfrog {
        timestep 1.0
    force Bond, Angle
    force Coulomb
          -algorithm NonbondedCutoff
          -switchingFunction Shift
          -cutoff 6.5
          -switchon 0.1
    force LennardJones
          -algorithm NonbondedCutoff
          -switchingFunction FSWC1
          -cutoff 6.5
          -switchon 0.1
  }
}
```

Example: MTS Integrator

```
initial_t 0.0
nsteps 100
cell_size 6.5
outputf 1
init_temp 300

boundaryConditions Normal
cellManager Cubic

Integrator {

  level 2 HybridMC {
        cyclelength 20
        warmupcycles 10
        temperature 300
  }

  level 1 Impulse {
        cyclelength 5
    force Coulomb
          -algorithm NonbondedSimpleFull
          -
switchingFunction ComplementSWC1
          -cutoff 6.5
  }

  level 0 PLeapfrog {
        timestep .5
    force Bond, Angle
    force Coulomb
          -algorithm NonbondedSimpleFull
          -switchingFunction SWC1
          -cutoff 6.5
    force LennardJones
          -algorithm NonbondedCutoff
          -switchingFunction FSWC1
          -cutoff 6.5
          -switchon 0.1
  }

}
```

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

- User support
- Contributing new features for TTP
- Implementation support
- Parallelization support

Conclusions

- We provide a flexible component framework
- Generic programming to customize components, without performance penalties
- Contains common potential/forces & integrators
- CHARMM 19 and 28 force fields
- Starting with a sequential implementation
- Easy to extend to a parallel implementation
- Hiding parallelization in few objects

Future Work

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