Tuning Vector and Parallel Performance for Molecular Dynamics Codes on the Cray X1

L. Bolikowski, W. Burakiewicz, M. Cytowski, M. Fronczak,
M. Kozakiewicz, Ł. Ligowski, M. Łopuszyński, F. Rakowski,
W. Rudnicki, J. Słowińska, K. Wawruch
ICM UW, Poland

CUG 2005, Albuquerque

What is ICM?

Why Molecular Dynamics, and how?

Tuning results

Summary

Questions...

Interdisciplinary Centre for Mathematical and Computational Modelling

High Performance Scientific Computing

the largest Polish supercomputing centre

state licenses for scientific software

Interdisciplinary research in computational sciences

modelling in biomolecular, material, atmospheric, environmental sciences

visual modelling and data processing

Network services for the Polish scientific community and for general public

weather forecasts

Virtual Library, academic television, popular science events

Why Molecular Dynamics?

60% of ICM users use these codes

Our resources

One year ago – just me and Łukasz Bolikowski, with some help

Summer HPC school (15 students)

Currently team of 7 developers

MD codes choice

CHARMM, DFTB, CPMD, VASP, Siesta, Gromos

Chemistry at HARward Macromolecular Mechanics (CHARMM)

macromolecular simulations, including energy minimization,

molecular dynamics and Monte Carlo simulations.

huge and complicated Fortran 77 code with MPI

Initial status

huge (3xx K lines), but important code

does not compile on X1

Development

versions c29b1 and c31b1 ported

some bugs fixed (mostly memory allocation)

partial vectorization

CHARMM current status

6% of the code responsible for 90% of computation time

- We have selected that 6% of code for tuning

We have vectorized 20% of selected code

Parallel scalability is poor

- Problem of MPI

CHARMM performance

1x SSP X1 vs. Opteron 246

Test	Opteron	X1 initially	X1 currently
MD, 1k steps Cytochrome in water counterions, Ewald	40 m	84 m	33 m
MD, 15k steps GTP in Water, Verlet	55 m	204 m	??? m
MD, 1k steps Cytochrome in water counterions, CPT	38 m	231 m	??? m

Self-Consistent Charge Density-Functional Tight-Binding (SCC-DFTB)

developed by prof Th. Frauenchaim at University of Paderborn

Fortran Code (15 K lines)

Initial performance

Cray X1 MSP: **336 Mflops**

Development

main loop vectorization (85% of time)

general loops' redesign to longer vector length

better data alignment for memory access

Results?

1x MSP X1 vs. Opteron 246

Opteron	X1 initially	X1 currently
618.79 s	470.87ß	**48.29 s**

3 213 Mflops

currently 65% of computation time is Linear Algebra

Poor parallel scalability (MPI – 58% of time)

Car-Parrinello Molecular Dynamics (CPMD)

- Fortran Code (150 K lines)
- MPI communication

Changes

- porting
- adjusting compiler directives
- minor loop redesign

Results?

X1e vs. Opteron 246

Test	Opteron	X1 SSP	X1 MSP
Diborane	334 s	194 s	119 s
32 water molecules	3 014 s	784 s	389 s
Silicon super cell (64 atoms)	240 s	82 s	59 s
120 atoms of carbon	19 032 s	3 092 s	1 089 s

1 000 – 2 000 Mflops in SSP mode

Again, poor parallel scalability

Vienna Ab-Initio Simulation Package (VASP)

developed by prof Juergen Hafner and Juergen Furthmueller

at Vienna University

Fortran code with MPI

initial SSP performance: **100 – 300 Mflops** (PC: 270 Mflops)

Development

porting (basing on port of previous version)

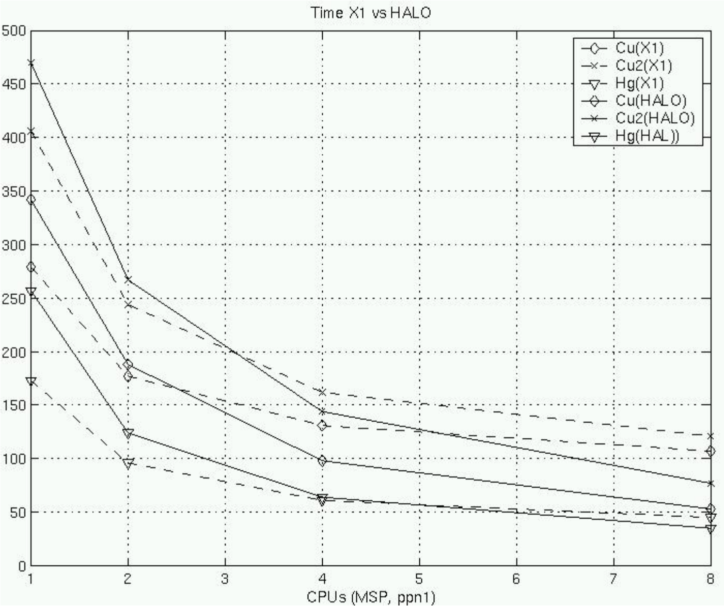
optimization of 3x main loops (90% of computation time)

VASP on Cray X1 vs. Opteron 246

Test set	X1 SSP	X1 MSP	Opteron
Cu	346 s	279 s	366 s
Cu2	481 s	406 s	470 s
Hg	201 s	173 s	257 s
HgIspin	389 s	339 s	465 s
HgLreal	375 s	358 s	527 s
AlNiNi	1 440 s	1 165 s	1 560 s
Ni	269 s	213 s	183 s

Performance rose to 800 – 1 555 Mflops per SSP

Parallel scalability?


Time X1 vs HALO
500
450
400
350
300
250
200
150
100
50
0
1
2
3
4
5
6
7
8
CPUs (MSP, ppn1)
Cu(X1)
Cu2(X1)
Hg(X1)
Cu(HALO)
Cu2(HALO)
Hg(HAL))

Gromos

Fortran code with PFortran/CAF parallelization (60 K lines)

vector version for Cray Y-MP of evaluation of nonbonded interactions between atom pairs

Development

we have focused on PROMD

loop redesign

Performance

Cray X1 SSP: 540 Mflops (80% of Opteron performance)

reasonable scalability

Spanish Initiative for Electronic Simulations with Thousands of Atoms (Siesta)

- Fortran 90 code with MPI
- ab-initio package

Current status

- poor performance: **200 – 430 Mflops** per SSP
- Opteron is usually faster
- poor parallel scalability
- optimizations also improve Opteron run times

Summary

Generally, it is possible to obtain reasonable performance

MPI problem

Where do you want to go *tomorrow*?