Parallelizing Protein Docking Code

to Accelerate Drug Discovery

Brad Burkman

Louisiana School for Math, Science, and the Arts

30 July 2014

www.lsmsa.edu

NAK-ə-təsh

Parallelizing Protein Docking Code

to Accelerate Drug Discovery

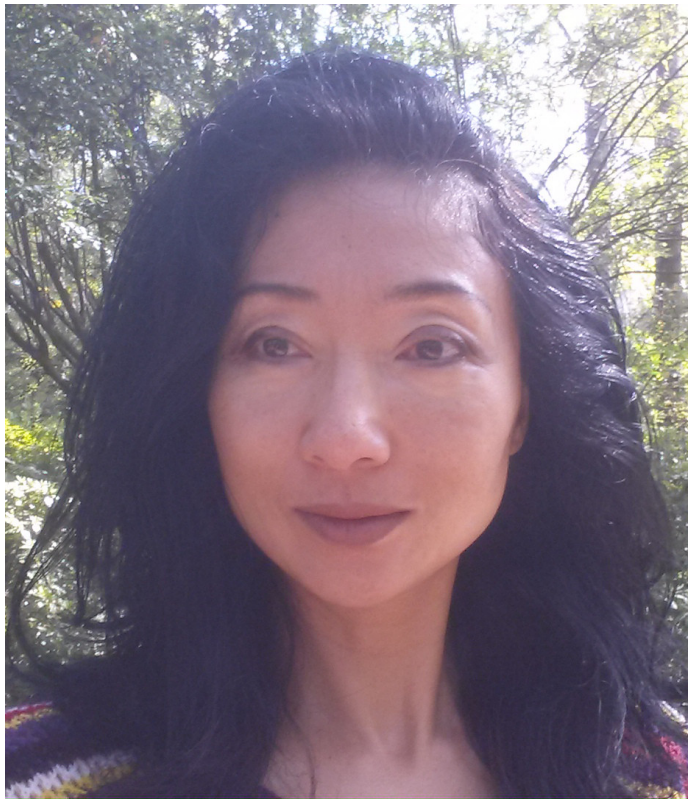
Brad Burkman

Louisiana School for Math, Science, and the Arts

30 July 2014

Mentors

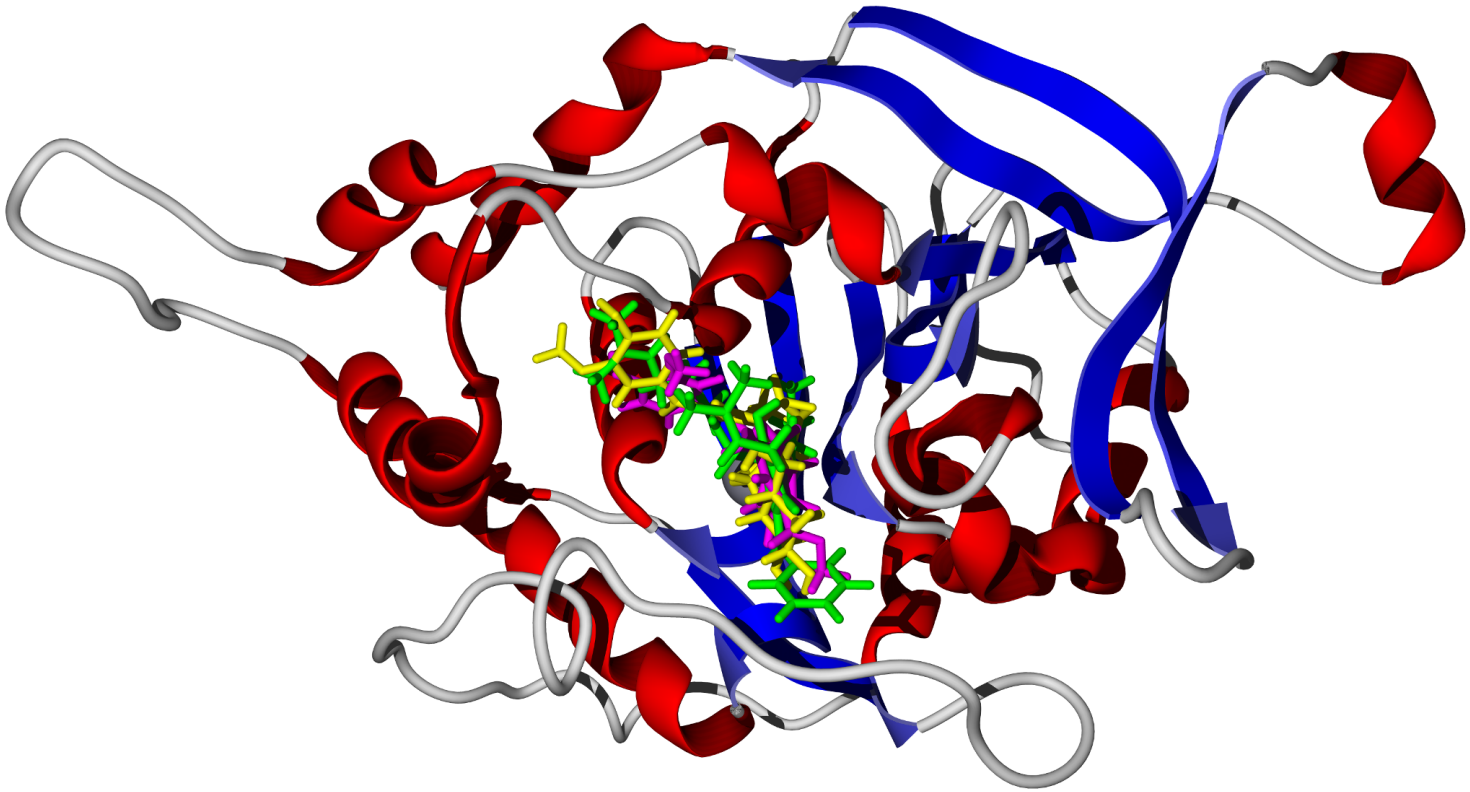
Michal Brylinski



Wei Feinstein

The Problem: Protein-Ligand Interactions

- Target Protein
  (human enzyme, receptor, regulatory protein)
- Drug candidates ("ligands")
- Other proteins (side effects)



The Problem: Protein-Ligand Interactions

×	Thousands of Drug Candidates
Thousands of Proteins	Millions of Interactions to Test

Only possible computationally, not experimentally.

Layers of the Problem

- Identifying proteins to target
- Molecular dynamics
- Fast Fourier Transforms
- 1000 lines of C code

Layers of the Problem

- ~~Identifying proteins to target~~
- Molecular dynamics
- Fast Fourier Transforms
- 1000 lines of C code

Layers of the Problem

- ~~Identifying proteins to target~~
- ~~Molecular dynamics~~
- Fast Fourier Transforms
- 1000 lines of C code

Layers of the Problem

- ~~Identifying proteins to target~~
- ~~Molecular dynamics~~
- ~~Fast Fourier Transforms~~
- 1000 lines of C code

ZDOCK Code Profile


```
for ( i=0; i<3600; i++ ) {
```

14 Fast Fourier Transforms

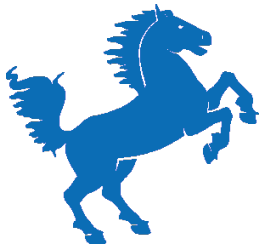
Each of $72^3 = 373,248$ complex numbers

```
}
```

Section of Code	Runtime (Seconds)	Runtime (Proportion)
Before Loop	1.94	0.57 %
Loop	340.38	**99.32** %
After Loop	0.39	0.11 %

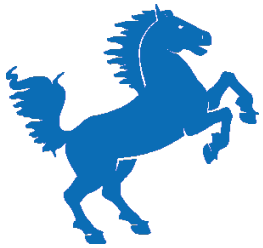
"Embarassingly Parallelizable" code

The Hardware: CPU



Intel Xeon E5
16 cores
22 GFLOPS/core
32 GB

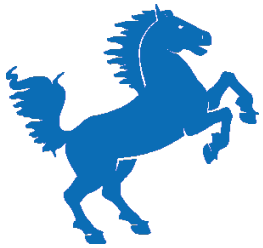
The Hardware: CPU



Intel Xeon E5
16 cores
22 GFLOPS/core
32 GB

```
for ( i=0; i<3600; i++ ) {
    DoStuff(...);
}
```

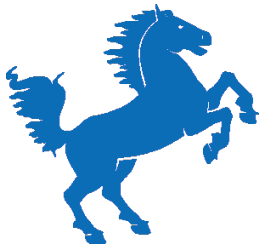
The Hardware: CPU



Intel Xeon E5
16 cores
22 GFLOPS/core
32 GB

```
#pragma omp parallel for
for ( i=0; i<3600; i++ ) {
    DoStuff(...);
}
```

The Hardware: CPU

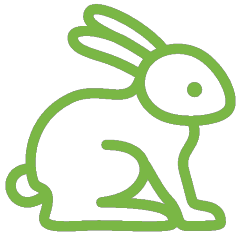


Intel Xeon E5
16 cores
22 GFLOPS/core
32 GB

10x speedup!

```
#pragma omp parallel for
for ( i=0; i<3600; i++ ) {
    DoStuff(...);
}
```

The Hardware: MIC

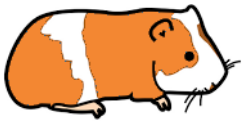


Intel Xeon Phi
60 cores
16 GFLOPS/core
8 GB

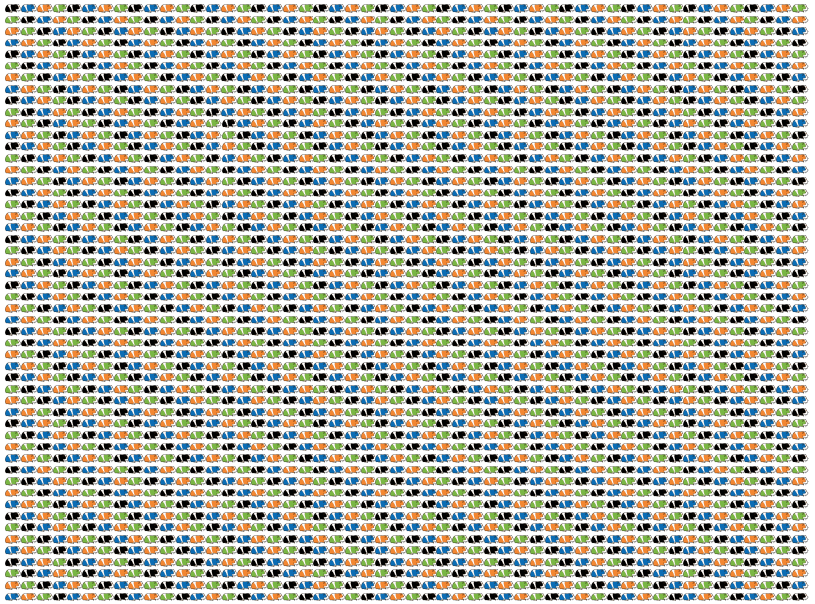
Challenges

- FFT library for the MIC not yet fully developed
- Efficient division of the problem may not be feasible

The Hardware: GPU



NVIDIA K20
2496 cores
1.4 GFLOPS/core
5 GB



5x speedup!

- One CPU core
- Offload FFT to GPU

Next Steps

- Use all 16 CPU cores and GPU together
- Use MIC effectively
- User-friendly packaging
- Write paper

Benefits from the Summer

- Compelling real-world application
- Fast Fourier Transforms
- Work to give to students
- Outreach sections of NSF proposals

Acknowledgements

People

Michal Brylinski and
Wei Feinstein

Doug James and
Carlos Rosales (TACC)

Nancy Wilkins-Diehr (SDSC)

Juana Moreno

Institutions

LA-SiGMA
Louisiana Alliance for Simulation-Guided Materials Applications

XSEDE
Extreme Science and Engineering Discovery Environment

TACC

NSF

Questions? bburkman@lsmsa.edu