

Parallelizing Protein Docking Code Brad Burkman Where? Mentor Problem

Next Step

Benefits

Thanks!

Questions

#### Parallelizing Protein Docking Code to Accelerate Drug Discovery

Brad Burkman

Louisiana School for Math, Science, and the Arts

30 July 2014

◆□▶ ◆帰▶ ◆ヨ▶ ◆ヨ▶ = ヨ = のへぐ



#### www.lsmsa.edu





#### NAK-ə-təsh



・ロト ・西ト ・ヨト ・ヨト ・ のへぐ



Parallelizing Protein Docking Code Brad Burkman

Where?

Mentor

Problem

Hardwar

Next Step

Benefits

Thanks!

Questions

#### Parallelizing Protein Docking Code to Accelerate Drug Discovery

Brad Burkman

Louisiana School for Math, Science, and the Arts

30 July 2014

◆□▶ ◆帰▶ ◆ヨ▶ ◆ヨ▶ = ヨ = のへぐ





Michal Brylinski



Wei Feinstein

▲口▶ ▲圖▶ ▲注▶ ▲注▶ -



# The Problem: Protein-Ligand Interactions

Protein Docking Code Brad Burkmai

Where?

Mentor

Problem

Hardwar

NEXL SLE

Benefits

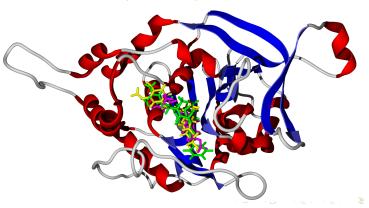
Thanks!

Questions

Target Protein

(human enzyme, receptor, regulatory protein)

- Drug candidates ("ligands")
- Other proteins (side effects)





# The Problem: Protein-Ligand Interactions

Parallelizing Protein		
Docking Code Brad Burkman	$\times$	Thousands of Drug Candidates
Where? Mentor	inds eins	Millions of
Problem	ousar Prote	Interactions
Hardware	ЪЧ	to Test
Next Steps	Γo	
Benefits		I

Only possible computationally, not experimentally.

◆□▶ ◆帰▶ ◆ヨ▶ ◆ヨ▶ = ヨ = のへぐ



Parallelizing Protein Docking Code

Brad Burkman

Where?

Mentor

Problem

Hardware Next Stej

Benefits

Thanks!

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



Parallelizing Protein Docking Code

Brad Burkman

Where?

Mentor

Problem

Hardware Next Stei

Benefits

Thanks!

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



Parallelizing Protein Docking Code

Brad Burkman

Where?

Mentor

Problem

Hardware Next Stei

Benefits

Thanks!

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



Parallelizing Protein Docking Code

Brad Burkman

Where?

Mentor

Problem

Hardware

in over over

Benefits

Thanks!

Questions

Identifying proteins to target

ヘロト 人間ト 人造ト 人造ト

- Molecular dynamics
  - Fast Fourier Transforms
- 1000 lines of C code



# ZDOCK Code Profile

Protein Docking Code Brad Burkman Where? Mentor Problem Hardware Next Steps Benefits

}

Questions

for ( i=0; i<3600; i++ ) { 14 Fast Fourier Transforms Each of  $72^3 = 373,248$  complex numbers

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

"Embarassingly Parallelizable" code



- Parallelizing Protein Docking Code Brad Burkman Where? Mentor Problem Hardware Next Steps Benefits
- Thanks!
- Questions



Intel Xeon E5 16 cores 22 GFLOPS/core 32 GB





- Parallelizing Protein Docking Code Brad Burkman Where? Mentor Problem Hardware Next Steps Benefits Thanks!
- Questions



Intel Xeon E5 16 cores 22 GFLOPS/core 32 GB



◆□▶ ◆帰▶ ◆ヨ▶ ◆ヨ▶ = ヨ = のへぐ

for ( i=0; i<3600; i++ ) {
 DoStuff(...);
}</pre>



Parallelizing Protein Docking Code Brad Burkman Where? Mentor Problem Hardware Next Steps Benefits

Questions



Intel Xeon E5 16 cores 22 GFLOPS/core 32 GB



◆□▶ ◆帰▶ ◆ヨ▶ ◆ヨ▶ = ヨ = のへぐ

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





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

Parallelizing Protein Docking Code Brad Burkman Where? Mentor Problem Hardware Next Steps Benefits



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



Parallelizing Protein Docking Code Brad Burkmar Where? Mentor Problem

Taruware

Repetite

Thanks!

Questions



NVIDIA K20 2496 cores 1.4 GFLOPS/core 5 GB

	. In 1949 we in 1949 w A statement of the theory of the theory of the theory of the theory of	
	~~~	
	. <u>A a constanta de constanta de</u>	
	THE CALL OF CALL AND A REAL AND A	
	R LER LE MER LER LER LER LER LER LER LER LER LER L	
	************************	
	THE REPORT OF THE REPORT OF THE PARTY OF T	
		-

◆□▶ ◆帰▶ ◆ヨ▶ ◆ヨ▶ = ヨ = のへぐ

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

Parallelizing Protein Docking Code

Brad Burkman

Where?

Mento

Problem

Hardware

Next Steps

Benefits

Thanks!

- 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



XSEDE

Extreme Science and Engineering Discovery Environment





# Questions?

#### bburkman@lsmsa.edu

