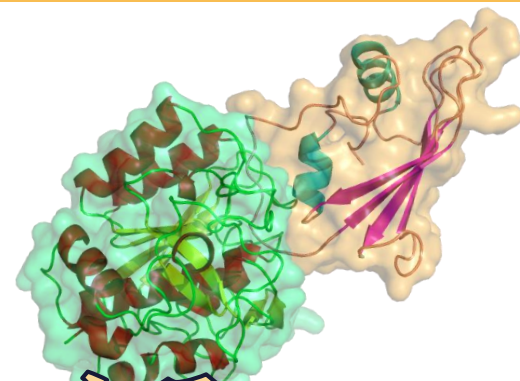




東京工業大学

Tokyo Institute of Technology



大規模GPUクラスタによる タンパク質ドッキング計算システム

大上 雅史^{1,2} 下田 雄大¹ 松崎 由理³ 石田 貴士¹ 秋山 泰^{1,3}

1. 東京工業大学 大学院情報理工学研究科 計算工学専攻

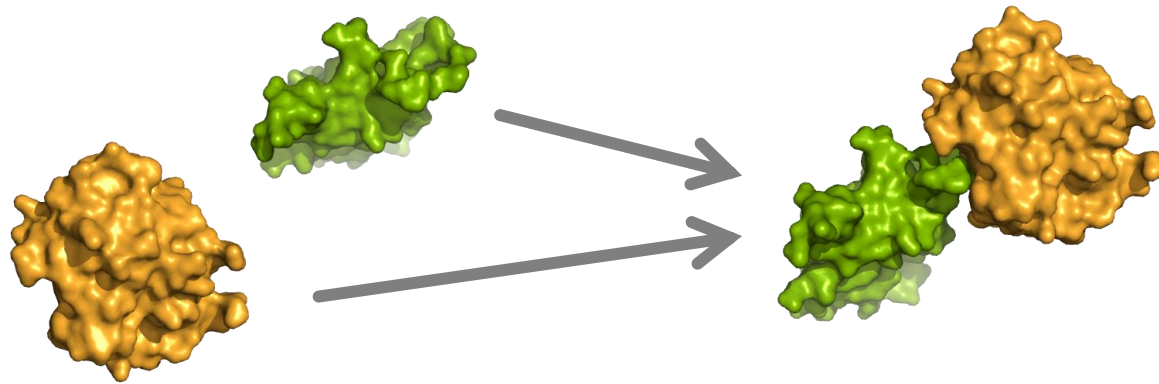
2. 日本学術振興会 特別研究員

3. 東京工業大学 情報生命博士教育院

タンパク質ドッキングとは

● タンパク質ドッキング

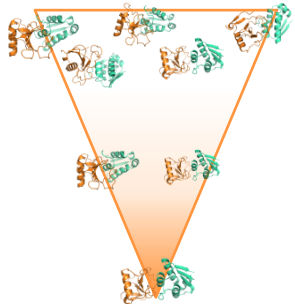
- 複合体の構造を単体の構造から予測する



- 最近のタンパク質ドッキングの基本的な戦略

1. 単体構造を剛体として並進と回転を許した探索によるサンプリング(**剛体ドッキング**)
2. 詳細なスコア関数によるリスコアリング
3. 主鎖や側鎖の構造を変化させてリファインメント

本発表の
対象



タンパク質剛体ドッキング

● 剛体ドッキング

- タンパク質を剛体とみなして計算
- Katchalskiの方法が良く用いられる

● Katchalskiの方法 ➤ Katchalski-Katzir E, *et al. PNAS*, **89**: 2195-2199 (1992)

- タンパク質をグリッドで表現し全空間並進探索を行う
- 積和の形の評価関数によってドッキング構造を評価
- 高速フーリエ変換を用いて計算を高速化
- リガンドタンパク質(L)を回転させて繰り返し計算(数千～数万パターン)

(↓実際は3次元)


レセプター
タンパク質


リガンド
タンパク質

➡

0	1	1	1	1	0
0	1	-5	-5	1	0
0	1	-5	1	1	0
0	1	-5	-5	1	0
0	1	1	1	1	0
0	0	0	0	0	0

0	0	0	0	0	0
0	0	1	1	0	0
0	1	1	1	1	0
0	0	1	1	0	0
0	0	0	0	0	0
0	0	0	0	0	0

$R(l, m, n)$
 $L(l, m, n)$

$$S(\alpha, \beta, \gamma) = \sum_{l, m, n} R(l, m, n) L(l + \alpha, m + \beta, n + \gamma)$$

畳込み和 $O(N^6)$

↓

$$S(\alpha, \beta, \gamma) = \text{IFFT} [\text{FFT}[R]^* \text{FFT}[L]]$$

FFTの利用 $O(N^3 \log N)$

タンパク質剛体ドッキングの応用事例

● 剛体ドッキング計算は様々な応用に用いられる

- アンサンブルドッキング

- ドッキング予測の戦略の1つ. 対象のペアに対してそれぞれ複数の構造のバリエーションを用意し, それらの組み合わせに対して計算
 - Grünberg R, et al. *Structure*, **12**: 2125-2136 (2004)
 - Król M, et al. *Proteins*, **68**: 159-169 (2007)

- タンパク質多量体ドッキング予測

- 2体の剛体ドッキングを多数行って多量体複合体構造を予測
 - Karaca E, Bonvin AMJJ. *Structure*, **19**: 555-565 (2011)

- タンパク質間相互作用予測, バーチャルスクリーニング

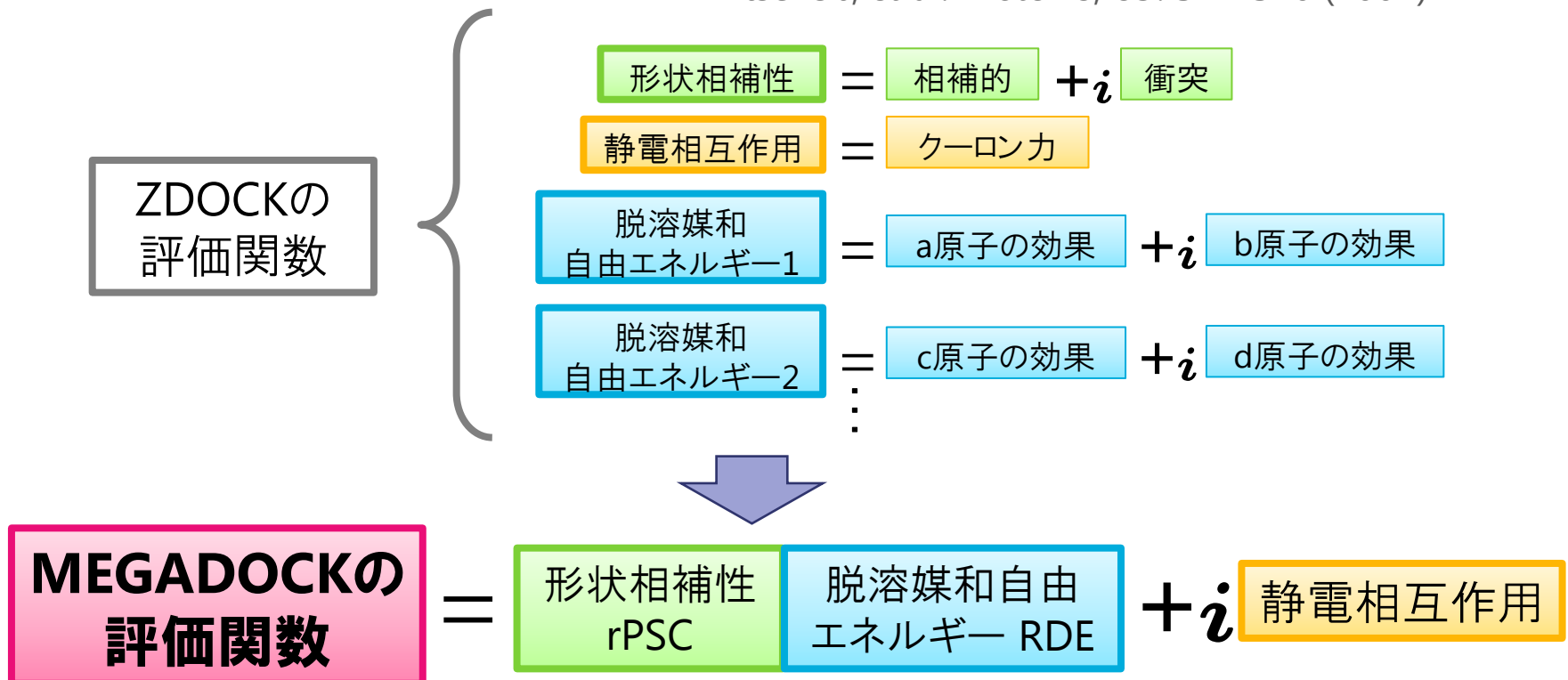
- 剛体ドッキングのスコアに基いて相互作用の有無を判別
 - Matsuzaki Y, et al. *J Bioinform Comput Biol*, **8**: 18 (2009)
 - Lopes A, et al. *PLOS Comput Biol*, **9**: e1003369 (2013)
 - Zhang C, et al. *Proteins* (early access) (2014)

これらの応用では大量のタンパク質ペアを扱う場合が多く
膨大な数(数万～数百万ペア)のドッキング計算を必要とする

MEGADOCK

- MEGADOCK
 - Ohue M, et al. *LNBI*, **7632**: 178-187 (2012)
 - Ohue M, et al. *Protein Pept Lett*, **21**: 766-778 (2014)
 - 当研究室で開発している剛体ドッキングソフトウェア
 - 評価関数の改良により1 CPUコアでZDOCK*の約10倍高速化

* Mintseris J, et al. *Proteins*, **69**: 511-520 (2007)



最近の計算機

● 京 (理研)

- 世界第4位
(10.5 PetaFlops, 2014年6月)
- 705,024 CPUコア
(8 CPUコア × 88,128ノード)
- メモリ: 16GB/ノード



● TSUBAME 2.5 (東工大)

- 世界第13位
(2.8 PetaFlops, 2014年6月)
- 17,984 CPUコア
- 4,258 GPU (3 GPU/ノード)
- メモリ: 54GB/ノード



最近の計算機

● 京 (理研)

- 世界第4位
(10.5 PetaFlops, 2014年6月)
- 705,024 CPUコア
(8 CPUコア × 88,128ノード)
- メモリ: 16GB/ノード



● TSUBAME 2.5 (東工大)

- 世界第13位
(2.8 PetaFlops, 2014年6月)
- 17,984 CPUコア
- 4,258 GPU (3 GPU/ノード)
- メモリ: 54GB/ノード



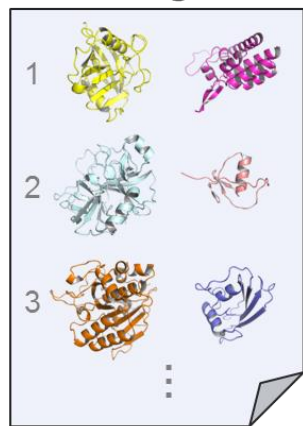
MEGADOCK-K

● スパコン(複数CPUノード)でMEGADOCK

➤ Matsuzaki Y, et al. *Source Code Biol Med*, 8: 18 (2013)

プロセス並列(MPI)

Docking List



Master Node

Worker Node

Worker Node

Worker Node

スレッド並列(OpenMP)

Receptor ボクセル化

Receptor FFT

Ligand 回転

Ligand ボクセル化

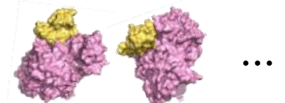
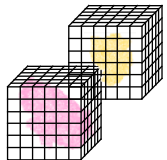
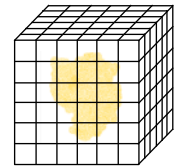
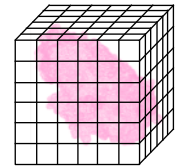
Ligand FFT

複素畳込み

逆FFT

高評価値の構造を抽出

リガンド回転ループ(OpenMP)



最近の計算機

● 京 (理研)

- 世界第4位
(10.5 PetaFlops, 2014年6月)
- 705,024 CPUコア
(8 CPUコア × 88,128ノード)
- メモリ: 16GB/ノード



● TSUBAME 2.5 (東工大)

- 世界第13位
(2.8 PetaFlops, 2014年6月)
- 17,984 CPUコア
- 4,258 GPU (3 GPU/ノード)
- メモリ: 54GB/ノード

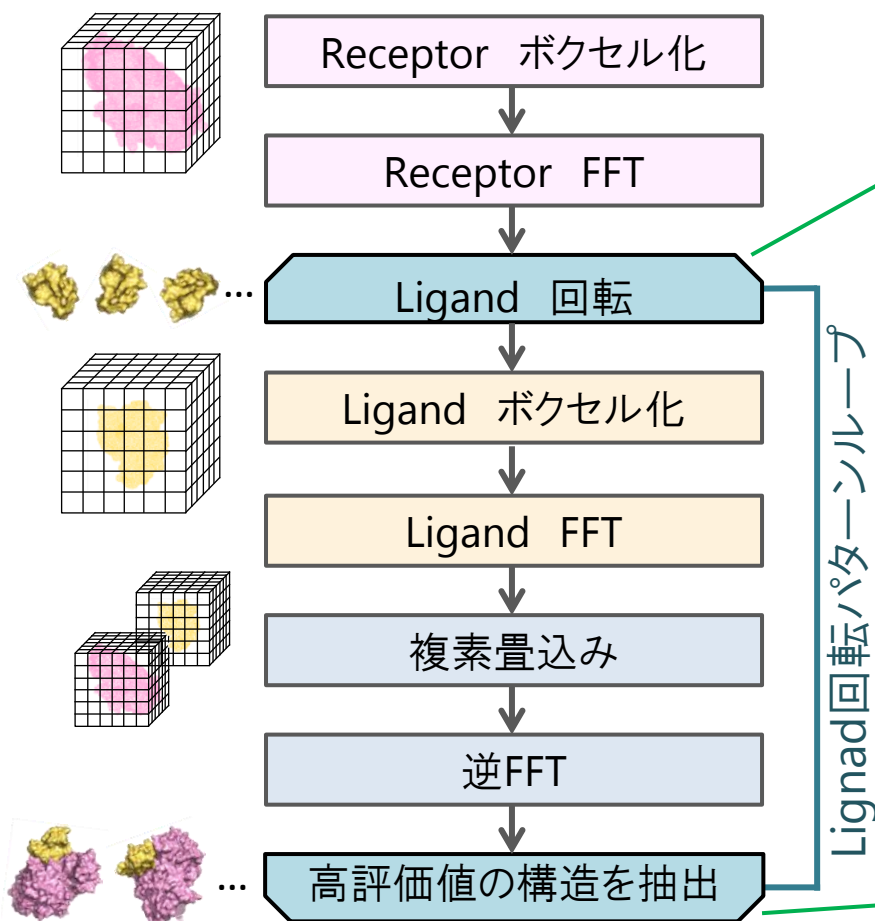


MEGADOCK-GPU

● GPU(単一ノード)でMEGADOCK

➤ Shimoda T, et al. ACM-BCB2013, 884-890 (2013)

ドッキングプロセス



GPU化プロセス



1. リガンドの回転とボクセル化

- リガンドの初期構造と回転座標のみを転送
- GPU上で回転計算
- 原子ごとの処理をGPUスレッドに割り当て

2. FFT, 逆FFT

- cuFFTライブラリを使用
- FFT基底の最適化

3. 複素畳込み

- 複素共役・乗算

4. 高評価値構造の抽出

- 並列リダクション

最近の計算機

● 京 (理研)

- 世界第4位 (10.5 PetaFlops, 2014年6月)
- 705,024 CPUコア

● TSUBAME 2.5 (東工大)

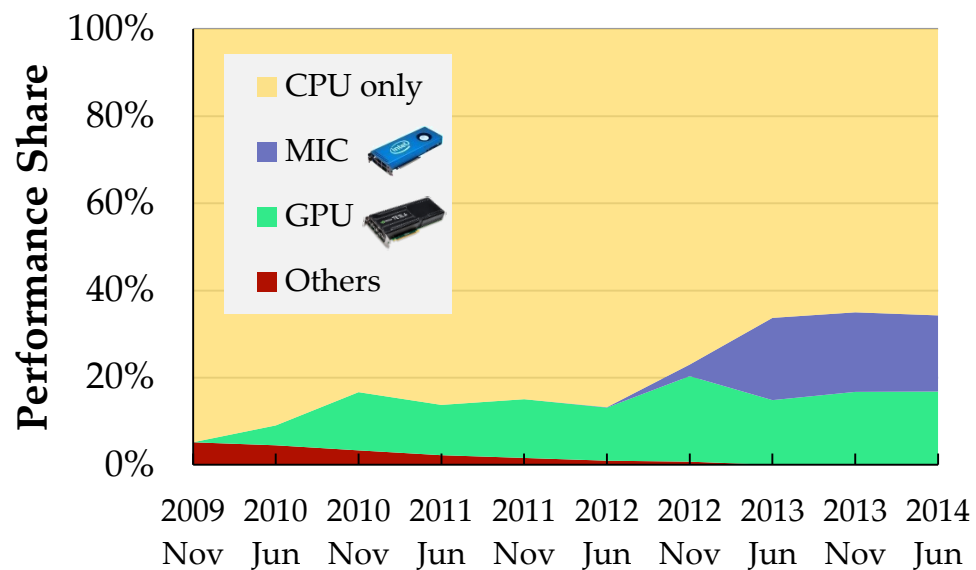
- 世界第13位 (2.8 PetaFlops, 2014年6月)
- 17,984 CPUコア, 4,258 GPU



ハードウェアアクセラレータを搭載した
並列計算機が増えつつある



ヘテロジニアスな計算機環境を
活用できるソフトウェアが求められる



top500.orgより. 全Flopsに対するハードウェアの割合.

本研究の概要

● 背景

- タンパク質ドッキングの大量実行が必要
- GPUスーパーコンピュータの台頭

● 目的

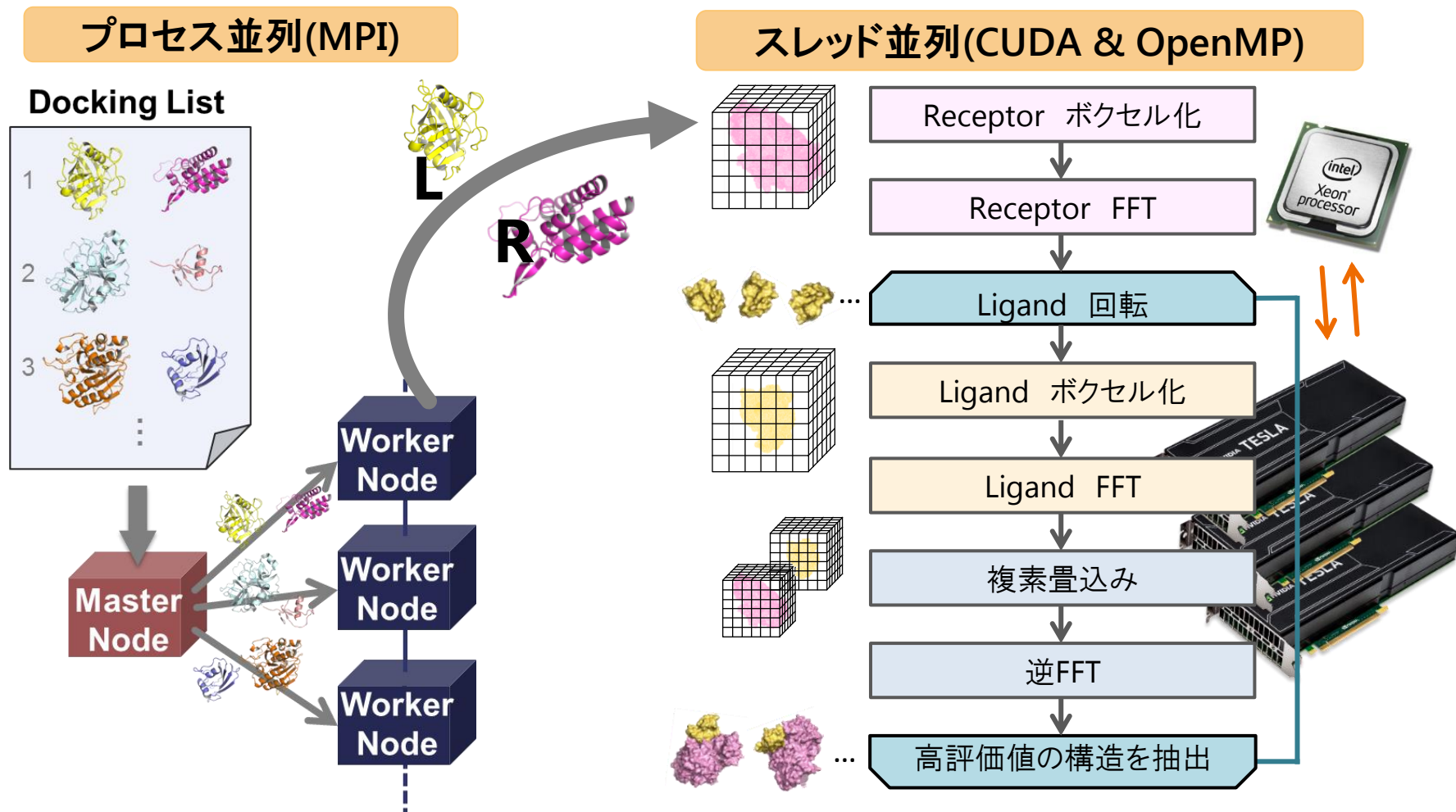
- 複数のGPUノード間で実行可能なMEGADOCKの開発

● 方針

1. 単一のGPUノード上での並列化
 - Shimoda T, et al. *ACM-BCB2013*, 884-890 (2013)
2. 複数のCPUノード間での並列化
 - Matsuzaki Y, et al. *Source Code Biol Med*, **8**: 18 (2013)
3. 複数のGPUノード間での並列化
 - Ohue M, et al. (submitted)

複数GPUノードの利用

● GPUクラスタ向け並列化



性能測定

● 計算機環境



- TSUBAME 2.5 Thinノード (~420ノード*利用) *一般ユーザの同時利用可能最大数

CPU	Intel Xeon X5670 (2.93 GHz) (6 cores) × 2
Memory	54 GB
GPU	NVIDIA Tesla K20X (GK110) × 3
GPU Memory	6 GB / GPU
OS	SUSE Linux Enterprise Server 11 SP1
Compiler	Intel C++ Compiler 14.0.2.144
FFT library (CPU)	FFTW 3.2.2
CUDA	CUDA 5.5
FFT library (GPU)	cuFFT 5.5

● データセット

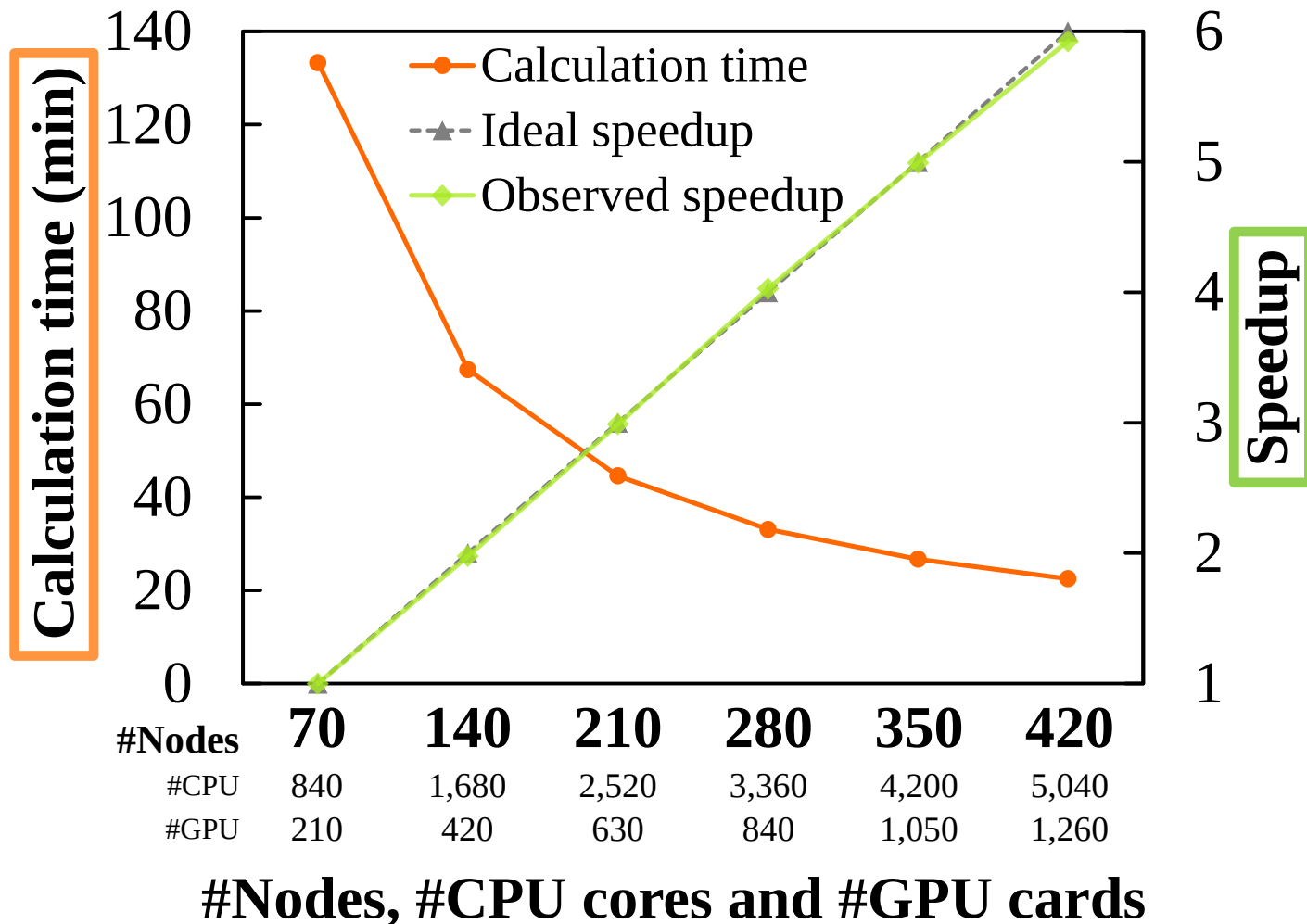
- ZLAB docking benchmark 4.0

➤ Hwang H, et al. *Proteins*, **78**: 3111-3114 (2010)

- 176ペアの複合体構造
- 全対全の組み合わせ ($176 \times 176 = 30,976$ ペア) のドッキング計算

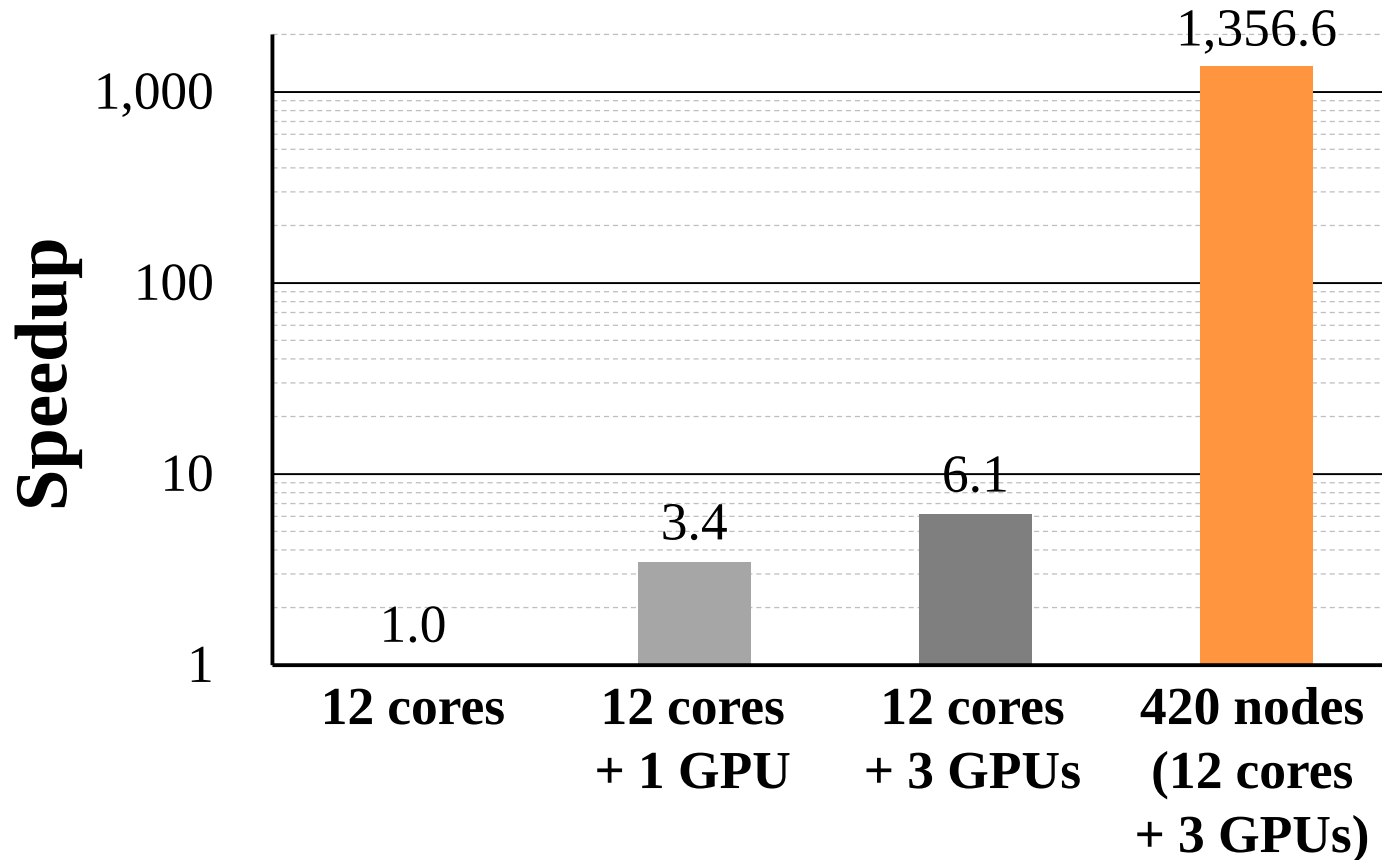
ノード数を変えたときの計算時間の推移

30,976ペアのドッキングを行ったときの計算時間



高速化率の比較

30,976ペアのドッキングを行ったときの計算時間



結論

● タンパク質ドッキングソフトウェアMEGADOCKの大規模GPUクラスタ向けの並列実装を行った

- 良好なスケーラビリティを確認
- 単一CPUノード利用時に比べて420ノードで1350倍の高速化
- 100万ペアのドッキング計算が約半日で完了
- オープンソースで公開中
<http://www.bi.cs.titech.ac.jp/megadock/>

● 今後の課題

- MIC搭載クラスタ向けの実装
- GPUとMICのクラスタ間の性能比較

本研究は、科研費(特別研究員奨励費23・8750, 26・30002)およびHPCIシステム利用研究課題(hp140173)の支援を受けて行われました。