HMMER の OpenMP による並列化

池田 晃* 松井 藤五郎* 大和田 勇人*

東京理科大学 理工学部 経営工学科*

1 はじめに

HMMER[1] は遠縁なタンパク質に有効な相同性検索ツールとして開発された．HMMER を利用して複数のドメインにおいて検索精度を高めたツールが MDHMMER[2] である．

MDHMMER は検索クエリとして各ドメインのアミノ酸配列群を与えると，HMMER による相同性検索によってドメインごとの類似度が求められ，全てのドメインの類似度から結合 E-value という統合的な評価値が計算される．結合 E-value で昇順にソートし，ドメインの位置情報も考慮に入れた上で，最終的な検索結果が出力される．

MDHMMER ではマルチドメインタンパク質の各ドメインに対して処理を行うため，ドメインの数が増えるにしたがって時間を要してしまう．そこで我々は実行時間を短縮するために，OpenMP[3] を用いて MDHMMER を並列化し，PC クラスタ上にシステムを構築した．これを **PMDHMMER** と呼ぶ．

本論文では PMDHMMER の並列化手法について解説し，最後に評価実験を行なうものとする．

2 PMDHMMER

PMDHMMER では MDHMMER を実行する際に最も時間を要する相同性検索部分の並列化を試みた．各ノードが実行する逐次の hmmsearch プログラムに対して OpenMP を用いて並列化を行った．

ドメインをデータベース内の膨大なタンパク質と照合する際，各ノードにおいて各 CPU コアが同数のタンパク質を検索するように分担させた．この一回ごとの計算量はほぼ同一であると考えられるため，同数ずつ割り当てた場合に待ち時間が発生する可能性は低いと言える．

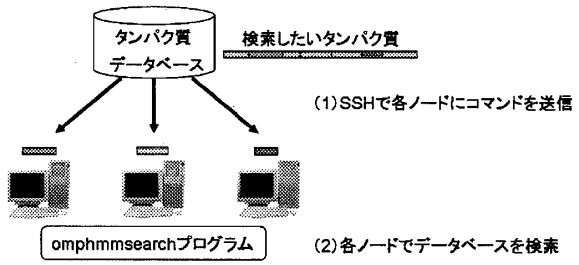

図 1 ドメイン並列の処理の流れ

また，クラスタのノードに処理を分担させる並列化手法として各ドメインを丸ごと各ノードに担当させるドメイン並列と，あらかじめ分割したデータベースを各ノードが担当するデータベース並列の二種類がある．それぞれの手法については各節で詳述していく．

2.1 ドメインごとの並列化

ドメインごとの並列処理の流れを図 1 に示す．ドメインごとの並列化は単に各ドメインの処理を各ノードに割り振るだけなので実装が容易である．しかしながら，ドメインとノードの数が一致していない場合は全ノードを活用できないために非効率である．また各ドメインの長さが不均一な場合には待ち時間が生じてしまうなど欠点も多い．条件が全て揃っている場合には最も有効な手法である反面，その他の場合には計算資源の有効活用といった観点から考えると不適当である．

2.2 データベースの分割による並列化

データベース並列の手順を図 2 に示す．データベースの並列化は先にデータベースを分割しておかなければならないため，ドメインごとの場合に比べてオーバーヘッドが大きくなる．一方で，データベースをノードの数に応じてほぼ均等に分割することが可能であるため，各ノードが担当する計算量もほぼ同一になり，計算資源の有効活用を図れるという利点がある．通信コストなどが余計にかかってしまう分，ドメイ

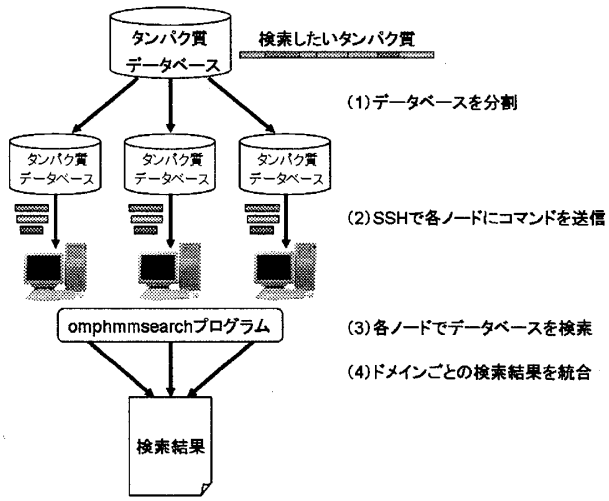

図 2 データベース並列の処理の流れ

Parallelizing HMMER with OpenMP

Hikaru IKEDA*, Tohgoroh MATSUI*, and Hayato OHWADA*

Department of Industrial Administration, Faculty of Science and Technology, Tokyo University of Science*

表 1　各手法とドメイン数別の実行時間

名称	ドメイン数 2	3	4
MDHMMER	52.48	79.34	105.63
OpenMP-MDHMMER	**25.57**	**38.65**	**51.48**
pthread-MDHMMER	25.58	38.69	51.51
Domain-OpenMP-**PMDHMMER**	**12.94**	**13.18**	**13.17**
Domain-pthread-**PMDHMMER**	13.00	13.30	13.28
Database-OpenMP-**PMDHMMER**	14.00	14.28	14.47
Database-pthread-**PMDHMMER**	**13.96**	**14.19**	**14.22**
All4-Database-OpenMP-**PMDHMMER**	7.67	11.03	14.47
All4-Database-pthread-**PMDHMMER**	**7.61**	**10.94**	**14.22**

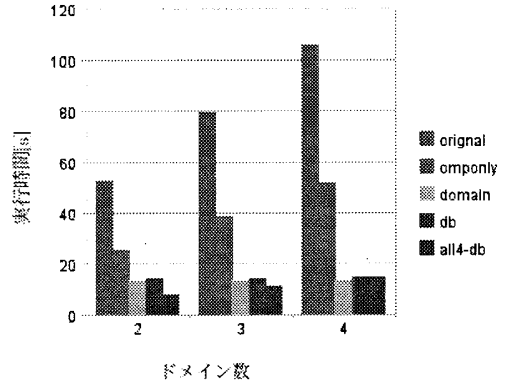

図 3　オリジナルと OpenMP 版の各手法におけるドメイン数別の実行時間

ンごとの並列化に比べてオーバーヘッドの影響が大きいものの，汎用性という点では一歩優れている手法である．

3　実験手法

3.1　シミュレーションデータセット

実験用に仮想のマルチドメインタンパク質を含むシミュレーションデータセットを作製した．データベースはランダムに生成される配列長 1000 のタンパク質 30000 セットにより構成されており，検索のクエリとなるドメイン配列群は各ドメインの配列長 20，ドメイン間の配列長を 30 とし，ドメイン間の領域については完全にランダムな配列に置き換えた．

3.2　実装環境

スイッチングハブを介して互いに Gigabit Ethernet でネットワーク接続し，全 4 台によるホモ PC クラスタを構築した．各ノードの CPU には Intel Core 2 Duo プロセッサを用いており，2 つの CPU コアを活用した共有メモリ並列化が可能である．また共通のデータ利用や出力結果の保存の利便性を考えて NFS（Network File System）を用いた．尚，ノードのうち 1 台が NFS サーバを兼ねている．

3.3　評価方法

omphmmsearch プログラムの実行から結果出力までの実時間の計測をそれぞれ 100 回ずつ行い，その平均値を各システムの所要時間とした．また，HMMER 付属の pthread 版も比較対象として用いた．本実験ではドメイン数やノード数などを変更して合計 27 通りの場合について計測を行った．

オリジナルの MDHMMER は常に 1 台で実行した時間を計測し，それ以外の提案手法についてはドメイン数とノード数を同じにして実験を行った．それに加えて全 4 台を活用したデータベース並列の実験も行い，他の手法と比較した．

4　結果

前章で作成したシミュレーションデータセットを用いた 27 通りの実験結果を表 1 に示す．全ての実験結果においてオリジナルの MDHMMER よりも実行時間が短縮された．

5　考察

全結果からオリジナルと OpenMP 版を抜粋してグラフ化したものが図 3 である．hmmsearch プログラムを OpenMP あるいは pthread によって並列化し，2 つの CPU コアに対して処理を分散させている．そのため全てのドメイン数において少なくとも実行時間が 2 分の 1 に短縮された．

またドメイン並列とデータベース並列を比較してみるとオーバーヘッドの差がそのまま表れた結果，ドメイン並列の場合に優位性が見られた．一方で，全ノードを活用した場合にはデータベース並列がよりよい結果となった．

そして OpenMP 版と pthread 版の比較では，各ノードに CPU コアと同じ数だけタスクを割り当てた場合には OpenMP 版のほうが若干優位性を持っているようだが，それ以上のタスクを分配した場合には pthread 版が有利になっている．

6　今後の展望

PMDHMMER は台数以上に MDHMMER の実行時間を短縮することができたことから，本論文で提案した OpenMP による HMMER の並列化は有効であると言える．ただし，OpenMP 版と pthread 版は一長一短であり，場合に応じて使い分けることが望ましい．今後は，ドメインごとに長さが異なるデータセットでも検証を行い，より汎用性が高いシステムを構築したい．

参考文献

[1] Sean Eddy. HMMER Profile Hidden Markov Models for Biological Sequence Analysis, Version 2.1.1, Washington University Department of Genetics, 1999.

[2] 瀬下真吾，賀屋秀隆，松井藤五郎，朽津和幸，大和田勇人. Multi-Domain HMMsearch:マルチドメインを持つ遠縁なタンパク質のための相同性検索ツール．第 5 回情報科学技術フォーラム (FIT-2006) 情報科学技術レターズ，LH-006, pp.153-156 (2006). 2006 年 9 月，福岡大学，福岡.

[3] OpenMP, http://www.openmp.org/blog/