超ロバスト並列計算

小柳義夫 須田礼仁 西田晃
情報理工学系研究科コンピュータ科学専攻

概要

サブプロジェクト「超ロバスト並列計算」では，ネットワークや計算機の構成・性能が不均質であったり動的に変動したりするような並列計算環境において，ネットワーク・計算機の故障や追加・負荷の変動などの外乱に対しロバストに対応し計算能力を効率よく引き出す手法の開発をめざしてきた．すなわち，数値アルゴリズムと並列化手法の両面から，性能（計算機資源の利用効率）のロバストネスを導くことが研究の目的である．本報告では平成 17 年度の成果の概要と今後の計画について略述する．

1 研究の背景・経過と全体計画

すでに万人にとって必須の道具となっている計算機とネットワークであるが，その機能と性能とは今後さらに新たな展開を見せるものと期待される．世界中の計算機がネットワークにつながることにより自律的に形成されている巨大情報システムは，そこで生じる情報量とともに総体的な情報処理能力も，従来のシステムでは考えられない莫大なものである．この巨大システムのもつ潜在能力を十全に引き出すことができれば，科学技術のみならず，人間と社会のさらなる進歩に寄与することが期待される．

しかしこのようなネットワーク計算環境は従来の超並列スーパーコンピュータとはまったく異なる構造をしている．従来の並列処理では均一な仕様の計算機を想定していたのに対し，汎用ネットワーク上では不均一な仕様の計算機が結合されている仮定のほうが自然である．また従来の並列処理では計算機とネットワークを占有して使っていたのに対し，汎用ネットワーク上では多くのユーザと共有する計算機がやはり共有のネットワークで接続されており，実効的な性能は他のユーザの影響などにより動的に変化する．さらに，システム的にまた地理的に離れた計算機を非常に多数結合しているため，動的な構成の変化や故障への対応についても考慮する必要がある．

すなわち，汎用ネットワーク上にすでに出現している新たな並列計算環境のポテンシャルを最大限に活かすためには，従来の占有型並列計算とは異なる，新たな並列化の手法が必要である．我々のサブプロジェクト「超ロバスト並列計算」では，並列プログラミングの概念と手法を根本的・全面的に刷新し，21 世紀の科学技術計算の発展を支える高性能並列計算の実現の礎となる並列化手法の総合的な開発を目指している．

実験環境の整備・既存研究の調査を中心に，本 COE の「大規模ディペンダブル情報基盤プロジェクト」の田浦研究室との情報交換などを行った平成 14 年度に続いて，平成 15 年度には不均一な計算環境に適応できるロバストな集団通信とデータ再分散の手法を開発し，簡易かつロバストに並列性能を引き出す手法である ERXPP の提案を行った．平成 16 年度には不均一な計算機・ネットワーク環境に適したデータマッピング（分割・割付手法）および計算機・ネットワークが故障しても計算を継続できる並列計算ソフトウェアシステムを開発した．本年度は，非一様な計算機で構成された並列計算機（ヘテロクラスタ）のための LU 分解の実装と，Multi-Master Divisible Load モデルに基づく効率的な再分散スケジューリングの手法の開発が主要な成果である．最終年度となる来年度はこれまでに開発してきた手法の集成とアプリケーションでの実証を中心に進める予定である．

2 本年度の成果

本節では平成 17 年度の主な研究成果について報告する．

まず，科学技術計算の基本計算の一つである LU 分解をヘテロクラスタに実装した研究について報告する．この研究の特徴は，負荷の均衡化に

は適しているが通信手順が複雑で不利だといわれてきた列ベース分割を効率的に用いている点にある．

次に，Multi-Master Divisible Load (MMDL) モデルによるデータ再分散の漸近最適スケジューリングについて報告する．最適化の難しい再分散問題に対して，理論上の限界に近い性能が得られる保証がある点が特徴である．

2.1 ヘテロクラスタにおける LU 分解

LU 分解は科学技術計算の高性能計算において特別の地位にある．第一に，それ自体として連立一次方程式の解法の最も重要なものであり，実用的に意味がある．第二に，LU 分解に含まれる部分演算の多くが他の行列演算に含まれたり類似したりしており，LU 分解を研究することにより，他の行列演算でも役立つ知見を得ることができる．第三に，LU 分解のアルゴリズムは単純さと複雑さが適度に混じりあっており，工夫をすることで計算機の性能を適切に引き出すことが可能である．このため，LU 分解は並列処理の研究テーマとして極めて適切であり，我々はヘテロクラスタにおける密行列演算の研究の端緒として LU 分解を選んだ．

2.1.1 列ベース分割と剰余サイクリック割当

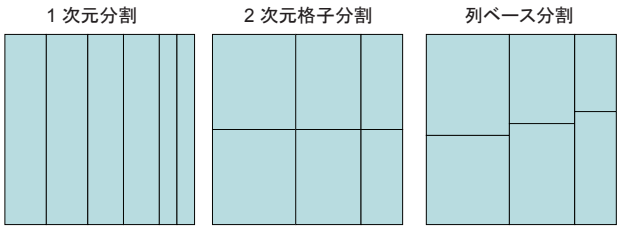

図 1 ヘテロクラスタ用の長方形の分割法

図 1 は行列のような長方形の領域を不均一な面積の長方形に分割する代表的な手法を示したものである．**1 次元分割**は領域の面積比を自由に設定できるが，プロセッサ数（領域数）が増えると長細い領域になってしまうのが欠点である．**2 次元格子分割**は領域の縦横比は抑えられるが，面積比の自由度が不十分で，プロセッサ数に制限があるなどの問題もある．**列ベース分割**は領域の縦横比，面積比，プロセッサ数のいずれも十分な自由度で制御できるが，各領域が隣接する領域の数が多く，通信回数が増えるのが弱点である．我々は列ベースに対する最適通信スケジューリングを提案しており，これを利用して列ベース分割に基づく並列 LU 分解を実装した．

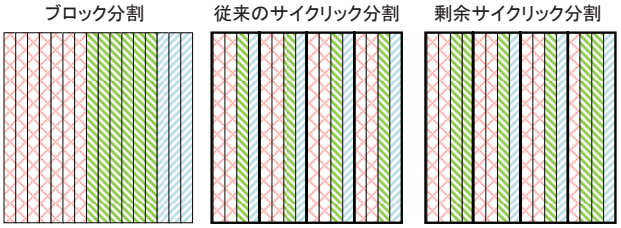

図 2 剰余サイクリック分割

図 2 は 1 次元分割のサイクリック割り当て法を示している．サイクリック割り当ては負荷の均衡化の一つの重要な手法であるが，テンプレートサイズが周回数に反比例して小さくなるため，負荷配分の自由度が減ってしまう．図 2 では，ブロック分割では 7:6:3 で分けられていたものが，サイクリックにするために 2:1:1 になってしまっている．剰余サイクリックの方法は，ブロック分割における割り当てブロックを周回数おきに拾うことにより，ブロック分割と同じ分割比を実現するものである．

2.2.2 結果

行列サイズ	2048	4096	6144
1 次元分割	**2.20**	3.26	2.32
2 次元格子分割	2.13	**3.75**	4.44
列ベース分割	2.13	3.41	**5.33**

表 1 ヘテロ（2.4GHz×4＋3.0GHz＋3.8GHz）クラスタによる LU 分解の性能（合計 Gflops）

表 1 は Xeon 2.4GHz×4 台＋3.0GHz×1 台＋3.8GHz×1 台をギガビットイーサで接続したヘテロクラスタに LU 分解を実装して得られた性能 (トータル Gflops) である．パラメタ類は行列サイズ・分割法ごとに最良だったものを使用している．

サイズが 2048 の場合は 1 次元分割が速い．これは枢軸選択の部分で通信が不要であることによるものと思われる．サイズが 4096 になると 2 次元格子分割が速い．1 次元分割では通信量が多く，パイプラインの深さが大きなオーバーヘッドになっているようである．その傾向はサイズ 6144 ではさらに顕著である．サイズ 4096 では列ベー

スはまだ遅いが，通信回数の多さが原因と思われる．サイズ 6144 では負荷が均衡している列ベースが最速になる．

このように，図 1 の 3 種類の分割法はどれかが他よりもよいというのではなく，状況によって適切に使い分けるべきであることが分かる．今回大規模問題では列ベースが最速であったことは意義深い．しかし，HPL では同じ構成で 9.77Gflops 出ており，実装上の改良をしないことには決定的な評価には至らないので，今後努力したい．

2.2 Multi-Master Divisible Load モデルによる漸近最適な再分散スケジューリングアルゴリズム

データ再分散は並列処理の基本問題の一つである．従来手法では，再分散のための通信とその後の計算とは分けて扱われることが多かった．これに対し，図 3 に示すように，通信と計算を同時にスケジューリングしたり，通信を分割したりすることにより，makespan（スケジュール長）をさらに短くすることが可能である．しかし，このように再分散方法の可能性を広げると，スケジューリングの最適化は非常に困難となるため，これまで知見らしい知見が得られていなかった．

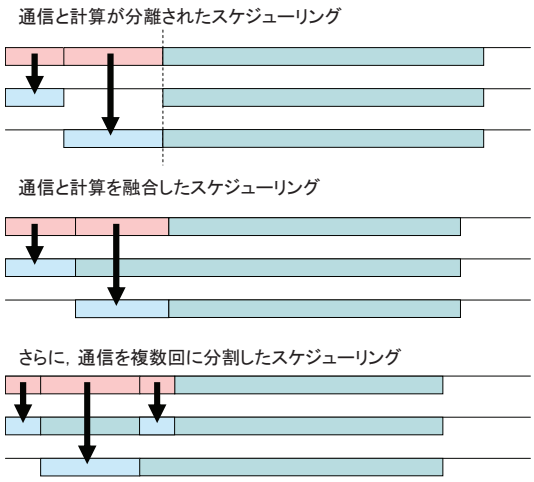

図 3 再分散スケジューリングとその改良

これに対して我々は，Multi-Master Divisible Load (MMDL) モデルを用いて，漸近最適なスケジューリングを極めて高速に求めるアルゴリズムを導いてきた．今回それにさらに改良を加えた．

2.2.1 漸近最適スケジューリングの改良

今回の改良の要点は 2 点である．第 1 点は，プロローグとエピローグの扱いの改良である．従来手法では定常ラウンドのみがスケジューリング対象で，プロローグはその通信部分，エピローグはその計算部分として定義されていた(図 4 上)．しかし改良手法ではプロローグにも計算を導入して定常ラウンドと同等の扱いとし，あまった計算をエピローグにまわすことにより，図 4 下のようにプロセッサの終了時刻を揃えた．

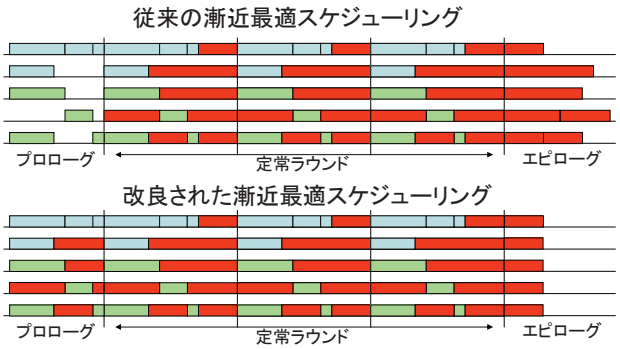

図 4 プロローグ・エピローグの扱いの改良

さらに，ラウンドサイズできるだけ大きく取ることによりラウンド数を削減した．その結果，ラウンドサイズは図 5 に示すようにラウンドごとに異なることになった．Yang-Casanova によるラウンドサイズ調整の提案によく似た結果が，理論的な最適化の結果得られたという点が興味深い．

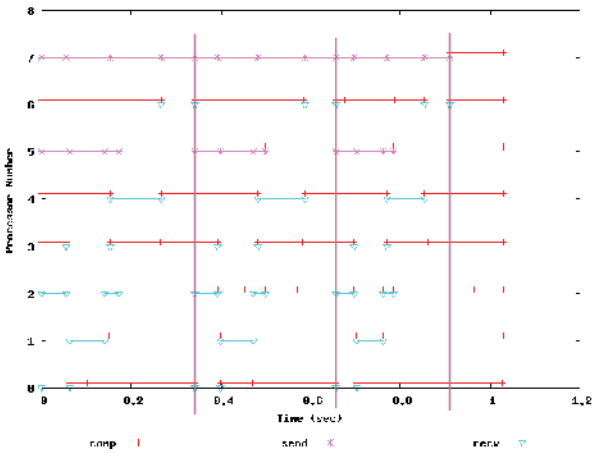

図 5 改良手法によるスケジューリング例

2.2.2 結果

表 2 はプロセッサ性能や初期割り当ての異なるいくつかの場合に対して，従来の漸近最適スケジューリングと今回の改良スケジューリングを比較したものである．いずれもラウンド数が減少し，

makespan が短くなり，理論的下限に非常に近い値を実現できるようになった．

例	従来手法		改良手法		make span 下限
	round 数	make span	round 数	make span	
A	40	1.0410	10	1.0054	1.0000
B	22	0.7364	1	0.7201	0.7197
C	38	1.1421	10	1.1107	1.1053
D	46	1.0788	5	1.0427	1.0406

表 2 漸近最適スケジューリングの改良の効果

3 まとめ

本報告では超ロバスト並列処理プロジェクトの平成 17 年度の研究開発の成果について報告した．2 節で報告したもの以外に，MMDL の実機への応用で必要となる性能評価とパラメタフィッティングの研究も行っている．

これまでの 4 年間の研究により，不均一な計算機・ネットワークで構築された並列計算環境における並列処理手法について充実した知見が得られたと考えている．しかし，これまで得られた知見だけでは，汎用ネットワーク上に現出している並列環境のバリエーションの広さに自由自在に適応するというのには程遠いということも認めなければならない．

問題を難しくしている要因は多数あるが，大きなものとしては，(1) ネットワークのトポロジー，(2) レイテンシの大きさ，(3) 不安定性や故障の可能性，が挙げられる．(1) ネットワークのトポロジーに関しては，適切に対応しようとするとそのための計算量が爆発してしまう点が難しさの本質である．近似により計算量を下げることは原理的には可能であるが，実用的に適切な近似が何かということから難しい問題である．(2) レイテンシは，大きくても確定的に出るのであれば個別の対応が可能であるが，汎用ネットワークで生じるような予期できないレイテンシに対応するのが難しい．(3) 故障については，適用するために相当なオーバーヘッドがかかるため，それをぎりぎり最低限に抑えることが必要であり，そのためにはシステムからアルゴリズムまで徹底した最適化が要求される点で大変である．

本プロジェクトでは (1) と (3) の課題についてはすでに取り組み，一定の成果を挙げてきたのは昨年度までに報告したとおりであるが，それらの成果は問題を解決するのに十分とはいえない．残された時間でこれらの課題の本質的な解決を見出すことは現実的ではないが，研究を進めなければならない課題であり，新たな研究態勢を整えるところから取り組みたいと考えている．

主な外部発表

[1] Y. Hourai, A. Nishida, and Y. Oyanagi, “Network-aware Data Mapping on Parallel Molecular Dynamics,” Proceedings of 11th International Conference on Parallel and Distributed Systems (ICPADS2005), pp. 126－132.

[2] 富さやか，須田礼仁，「Multi-master divisible load モデルに対する漸近最適スケジューリングの評価」，情報処理学会研究報告，2005(57)，pp. 51－56.

[3] 三森慶卓，須田礼仁，「ヘテロ型クラスタのための 2 次元列ベース分割における通信スケジューリングと分割の最適化」，情報処理学会研究報告，2005(57)，pp. 57－62.

[4] 富さやか，須田礼仁，「Multi-master divisible load の漸近最適スケジューリングの実機への実装」，情報処理学会研究報告，2005(81)，pp. 37－42.

[5] 三森慶卓，須田礼仁，「２次元列ベース分割によるヘテロ型クラスタのための LU 分解」，情報処理学会研究報告，2005(81)，pp. 127－132.

[6] 須田礼仁，富さやか，「有限サイズの multi-master divisible load 問題に対する再分散スケジューリングアルゴリズム」，情報処理学会研究報告，2006(20)，pp. 109－114.