

大域分散並列コンピューティング

田浦健次朗

情報理工学系研究科電子情報学専攻

1 はじめに

複数のサイトや管理ドメインにまたがったクラスタ群を、多数のユーザによって共有する環境で、効率的で、長時間にわたって高信頼に動作を続ける並列計算の方式について研究を行っている [1, 2, 5]. 本年度の成果の主なものを以下にまとめる.

2 耐遅延性の高い LU 分解

密行列の連立一次方程式の解法として代表的なものに LU 分解があり、そのための並列化手法も数多く研究されている. しかしながらそれらはほとんどの場合通信遅延が数十 μ 秒程度の LAN や クラスタを想定している. そのようなアルゴリズムは計算資源が広域に配置される環境では効率よく動作しない. 一方将来の並列計算プラットフォームとして、分散した計算資源の中から、必要なだけの資源を必要ときに即座に提供する、いわゆる Grid 環境が注目されている. そのような環境では、高遅延や低帯域を含んだ多様なネットワーク環境で頑健に高性能を達成するアルゴリズムが求められる. 本研究では LU 分解を対象として、アルゴリズムの高遅延化の研究を行った [1].

図 1 は通常の部分 pivoting による、逐次 LU 分解を擬似コードで示したものである. 外側のループ (k) の各繰り返しの最初に pivot 要素を見つける処理が必要であり、データが分散している際にはこの部分が遅延に敏感な処理になる.

図 2 中の左端の棒 (Partial) は、ノード間の遅延を人工的に増大させた場合の性能劣化を示している. 10ms 程度の遅延を加えると性能が 4 分の

```
for ( $k = 0; k < n; k++$ ) {
  finds pivot element  $a_{pk}$  in the  $k$ th column;
  exchanges the  $p$  th row and the  $k$ th row;
  for ( $i = k + 1; i < n; i++$ ) {
    for ( $j = k + 1; j < n; j++$ ) {
       $a_{ij} = a_{ij} - a_{ik}a_{kj}/a_{kk}$ ;
```

図 1: LU 分解と pivoting

一に低下している.

これに対し、我々が提案した Batched Pivoting 方式では、適当なバッチ化パラメータ d を設定することで、pivoting のための同期の回数をほぼ $1/d$ にすることができる. その際、必要な浮動小数点演算回数は dn^2 であり、LU 分解が必要とする $(2/3)n^3$ に比べて、通常の d (16 から 64 程度) では少ない. また、得られた結果の誤差は Partial Pivoting とほぼ同等であることが数値実験によって確かめられている.

再び図 2 に戻り、左端以外の 3 つの棒は Batched Pivoting においてバッチ化パラメータ d を 4, 16, 64 と変えて遅延への耐性を計測したものである. Partial Pivoting は Batched Pivoting において $d = 1$ としたものに相当し、 d を増やすことで遅延への耐性が大きく増していることが読み取れる. 図 3 は人工的な遅延のない環境で Partial Pivoting と台数効果を比較したものである. なお、BLAS ライブラリとして Goto (後藤) BLAS を用いており、Partial Pivoting の実装としては HPL をそのまま用いている.

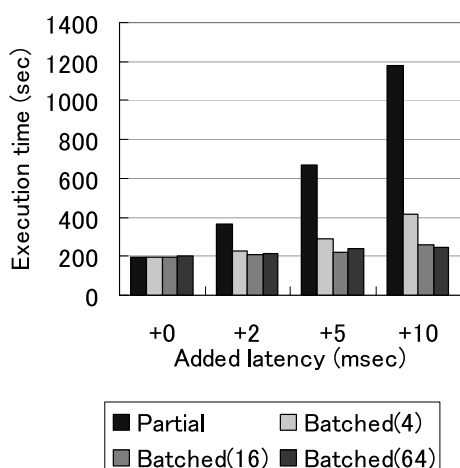


図 2: LU 分解の耐遅延性

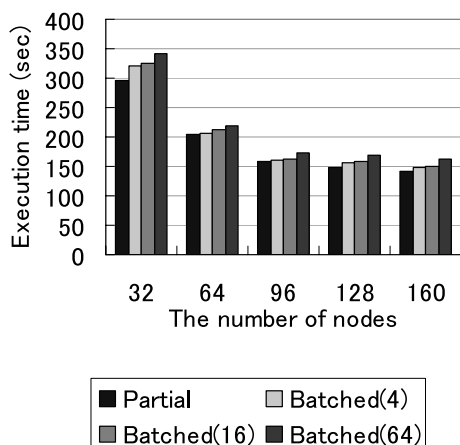


図 3: 台数効果

3 自律的計算におけるプロセス脱退 プロトコル

並列計算や分散計算において、参加プロセスの集合やプロセス間の接続を動的に変化(再構成)させながら計算を続行することは、資源の変化に適応した計算を行うための基礎である。たとえば広域分散環境や複数の利用者によって共有された環境においては、プロセス群をお互いに近くなるように配置したり、各時点で CPU/ネットワーク負荷がない、または少ないノードに配置したりする必要がある。並列アプリケーションが実行時に、

資源の状態に応じてそれを行うことが出来れば資源選択の柔軟性が高まり、資源の利用効率も向上する。

一方適応的な計算を正しく記述・実装することは容易ではない。特にプロセスの脱退を行うプロトコルは、競合状態によるメッセージの喪失やデッドロックなどの誤りを含みやすい。そのため実践的には、適応的計算は多くの場合、計算モデルやプロセス間の接続トポロジーを単純なものに限定して行われている。それに対し本研究では、プロセス間の接続トポロジーに制限がなく、全プロセスの任意の部分集合が脱退しようという一般的なモデルにおいて、プロセスが安全に脱退するプロトコルを提案した。本プロトコルでは、自分のプロセスに向かって飛行中のメッセージや、他のノードに転送すべきメッセージを残したまま、プロセスが脱退することがないことを保証する。

本プロトコルの特徴として以下があげられる。

並列性が高い: 中央の制御サーバなどを必要とせず、システム内の複数の場所で同時並行的に脱退を行うことができる。

局所性が高い: 通常、脱退するプロセスは自分の隣のプロセスとのみやり取りをすればよく、局所的な通信のみで脱退を行うことができる。

柔軟性が高い: 中央の制御サーバや、特定の制限されたトポロジーを仮定していない。したがってどんな結合トポロジーを持つアプリケーションへも自然に適用可能である。

本プロトコルの性能評価として、380 プロセス中の 120 プロセスを、ほぼ同時に脱退開始させ、全プロセスが脱退し終えるまでにかかった時間を計測した。比較として、脱退をすべて直列化するサーバを一台用いる方式との比較を行った。グラフを見てもわかるとおり、本プロトコルは、多数のプロセスが同時に脱退するような状況下で大幅な並列度の向上、それにとまなう性能向上を達成している。

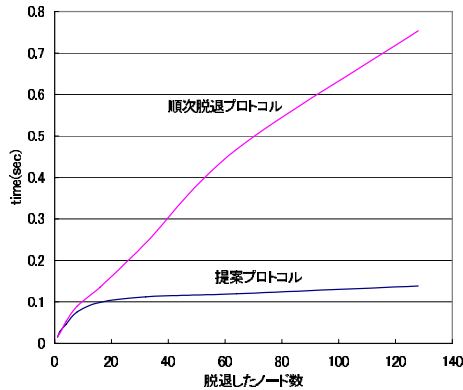


図 4: 脱退プロトコルの性能

4 CHLAC 特徴を用いた高精度 Gait 認識の高精度化と高速化

本 COE の中で複数分野にまたがる共同研究を推進する試みとして、100 時間ワークショップ [4] が開催された。100 時間ワークショップにおいて、当グループと、米澤グループ、大津・國吉グループ (実世界情報処理融合プロジェクト) との共同グループが結成され、本プロジェクトが行われた (大津・國吉グループから伊藤聡, 南里卓也, 下畠康幸, 白木孝義, 石黒勝彦, 深野亮, 米澤グループから金田憲二, 近山・田浦グループから堀田勇樹, 白井達也, 斎藤秀雄, 鴨志田良和が参加)。二つのサブグループに分かれて、高精度な認識処理を並列処理によって高速化するグループ [7] と、高精度な認識処理の実時間化を行うグループ [8, 6] に分かれて研究を行った。本報告では前者について述べる。

Gait 認識は動画像から、人物および動作 (走っている, 歩いている, など) を認識するタスクである。学習データとして多数の動画像と正解を与え、未知のテストデータに対して判定を行う。CHLAC 特徴は Gait 認識を高精度に行うために小林・大津ら [3] によって考案された特徴量であり、位置不変性を持つ、2 値化された差分画像に対して高精度を達成する、などの特徴を持つ。詳細は [3] を参照されたい。

高度な認識タスクに対して高精度化を達成する



図 5: Gait 認識タスク

には、特徴量や学習データの量を増やす必要があるが、それに伴い学習に必要な計算量は増加する。我々の研究グループが、これを並列処理によって解決すること、およびクラスタ環境を容易に利用できる環境を構築することなどを試みた。タスクとしては、図 5 のような階段を上るまたは下りる、そして走るまたは歩くの 4 通りの動作を 11 人の異なる人物が行うという、44 クラスの分類問題を行った。それぞれの動作を 6 回実行し、44966 枚の画像が用いられた。このタスクは人物の大きさが歩いている最中に変化するなど、動作認識タスクの中でも困難なものである。

並列処理のために、本 COE の設備である IBM BladeCenter クラスタ (2.4/2.8GHz Xeon 192 ノード, 384 CPU) を用いた。認識率の結果は図 6 に示すとおりである。抽出する特徴量の次元を 251 にした単純な処理 (3 pixel × 3 pixel × 3 frame のパターンを用いてひとつの特徴量を抽出する) では、認識率が 85% なのに対して、より多くの特徴量を用いることで最大で 98% の認識率を達成している (誤りを 1/7 程度に軽減)。もっとも精度のよい結果は、学習に 3 時間程度を要しており、1 CPU であれば 119 時間程度を要していたものと概算される。

5 MEDLINE アブストラクトのからの大規模知識抽出

辻井グループとの共同で、深い自然言語処理を大規模なデータに対して適用し、正確な知識抽



図 6: 認識率

出を行った。本年度は、辻井グループが研究を進めている HPSG 構文解析器を、オンラインで利用可能な生物医学論文のアブストラクト集全件に適用した。この処理には、本 COE の設備である IBM BladeCenter クラスタ (2.4/2.8GHz Xeon 192 ノード, 384 CPU) と、本学新領域創成科学研究科 近山研究室の設備である Appro クラスタ (2.4/2.8GHz Xeon 65 ノード, 130 CPU) を統合して用いた。統合のために我々が開発した並列処理ツール GXP を用いている。

MEDLINE は、医学・生物学に関する論文のほぼすべてを網羅した文献データベースであり、約 1500 万件の論文、そのうち約 700 万件に対するアブストラクトを収納し、アブストラクト中の文数は合計で約 7000 万文になる。HPSG は単一化に基づいた文法枠組みであり、文中の動詞の意味上の述語などを解析する。HPSG に基づいた構文解析を行うと、受動態をはじめとした、同じ意味に対するさまざまな表記を認識することができる。一方で構文解析にかかる計算量が大きいという問題点を持っている。

本プロジェクトでは、辻井グループが開発してきた高速・高精度な HPSG 構文解析器、それを生物医学テキスト用に機械学習を用いて適応した構文解析器 Enju を用いて、MEDLINE アブストラクト全件の構文解析を行った。1 文あたり約 1 秒という仮定の下では約 2 年かかる処理を、約 10 日間で終了した。ここで得られた構文解析の結果からたんぱく質間相互作用の抽出が行われ、インタラクティブな情報抽出システム MEDUSA が開

発されている。

参考文献

- [1] Toshio Endo and Kenjiro Taura. Highly latency tolerant gaussian elimination. In *Proceedings of IEEE/ACM International Workshop on Grid Computing (Grid2005)*, pages 91–98, 2005.
- [2] Yuuki Horita, Kenjiro Taura, and Takashi Chikayama. A scalable and efficient self-organizing failure detector for grid applications. In *Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing (Grid2005)*, pages 202–210, 2005.
- [3] Takumi Kobayashi and Nobuyuki Otsu. Action and simultaneous multiple-person identification using cubic higher order local auto-correlation. In *International Conference on Pattern Recognition*, pages 741–744, 2004.
- [4] Mihoko Otake, Ryo Fukano, Shinji Sako, Masao Sugi, Kiyoshi Kotani, Junya Hayashi, Hiroshi Noguchi, Ryuichi Yoneda, Kenjiro Taura, Nobuyuki Otsu, and Tomomasa Sato. Autonomous Collaborative Environment for Project Based Learning. In *Intelligent Autonomous Systems 9 T. Arai et al. (Eds.)*, pages 756–763. IOS Press, 2006.
- [5] Hideo Saito, Kenjiro Taura, and Takashi Chikayama. Collective operations for wide-area message passing systems using adaptive spanning trees. *International Journal of High Performance Computing and Networking (IJHPCN)*, 4(3), 2005.
- [6] 白木 孝義, 石黒 勝彦, 深野 亮, 鴨志田 良和, 白井 達也, 斎藤 秀雄, 田浦 健次朗, 大武 美保子, 佐藤 知正, and 大津 展之. CHLAC 特徴と Grid コンピューティングを併用したリアルタイム動作認識. *電子情報通信学会技術研究報告*, 105(615):97–102, 2006.
- [7] 伊藤 聡, 堀田 勇樹, 金田 憲二, 南里 卓也, 下畠 康幸, 田浦 健次朗, 大武 美保子, 佐藤 知正, and 大津 展之. 大規模クラスタを用いた高精度な Gait 認識. *電子情報通信学会技術研究報告*, 105(615):91–96, 2006.
- [8] 鴨志田 良和, 田浦 健次朗, and 近山 隆. 冗長性を用いた低遅延並列実行時間動作認識システム. In *先進的計算基盤システムシンポジウム (SACISIS2006)*, 大阪, May 2006. (to appear).