

変数アクセスコストを引力としてとらえたデータ分割手法とその視覚化

杉田 望代* 大音 真由美†
笹倉 万里子‡ 城 和貴*
moyo@ics.nara-wu.ac.jp

* 奈良女子大学 理学部 † 奈良女子大学 人間文化研究科 ‡ 岡山大学 工学部

概要

分散メモリ型環境において効率よく並列処理するには、「複数のメモリにデータをどのように分割・配置するか」というデータ分割問題を考えなければならない。本稿では、分散メモリ環境におけるデータ分割問題と自動並列化コンパイラの内部表現について述べた後、物理モデルである引力を応用したデータとタスクのクラスタ化支援とその視覚化の説明を行い、本視覚化を用いたデータ分割手法を提案する。さらに、そのプロトタイプを用いて、本手法の有効性を述べる。

A data partitioning method and its visualization assuming that variable access costs are described as gravity

Moyo Sugita* Mayumi Outo* Mariko Sasakura† Kazuki Joe*
* Nara Women's University † Okayama University

Abstract

To obtain effective parallel processing in distributed memory systems, facing to data partitioning problems, we should investigate how data and task are arranged and divided into distributed memories. In this paper, we describe a data partitioning problem in distributed memory systems and an intermediate representation of a parallelizing compiler, explain the support of clustering by gravity as a physical model, give an account of its visualization, and propose a data partitioning method with combination of the visualization. In addition, we report a prototype.

1 はじめに

今日、流体力学、原子物理学・天気予報・構造解析・分子科学・資源探査・画像処理などさまざまな分野で、大量の数値計算が必要とされている。というのも、実験では実現できないような現象を、シミュレーションによって解明できる場合があるからである。それらの解の精度や信頼性を確保するためには、膨大な計算量が要求される。それゆえに、計算機の高速化の要求が高まるのである。計算機の性能が向上すれば、より精度の高い計算が可能となり、計算結果の信頼性が高くなる。つまり、計算機の性能向上は、多くの分野での計算機利用を促し、各分野の発展の原動力となりうる。プロセッサの高速化の限

界が明らかになってきている現在、並列計算機による計算機性能の向上に期待がよせられている。

並列計算機では、各プロセッサは絶えずメモリアクセスを行い処理を進めていくため、メモリアクセスの効率は並列計算機自体の性能を左右する要因のひとつである。プロセッサとメモリの物理的結合の観点から並列計算機を分類すると、共有メモリ型並列計算機と分散メモリ型並列計算機の2種類に大別される。我々は、分散メモリ型並列計算機を対象計算機とする。分散メモリ型並列計算機の詳細については、第2章で述べる。

本稿では、分散メモリ環境におけるデータ分割について考察し、その手法と視覚化の提案を行う。以下第2章では、データ分割問題とデータ分割グラフ

について述べ、第3章ではデータ分割手法と視覚化の提案、さらに、プロトタイプを用いた本提案の検証を行う。

2 データ分割問題とデータ分割グラフ

2.1 分散メモリ環境

分散メモリ環境とは、多重化された各メモリモジュールをプロセッサと物理的に密接したローカルメモリとして配置されたものである。各メモリモジュールを各プロセッサのローカルメモリとして配置することによって、ローカルメモリに対し各プロセッサは相互結合網を介さずにメモリアクセスできる。そのため、多重化によるメモリ処理能力を高めるだけでなく、メモリアクセスの高速化をはかることができる。反面、他のプロセッサのローカルメモリ（リモートメモリ）に対するアクセス（リモートアクセス）は、相互結合網を介してアクセスするため極端に遅くなる。このため、分散メモリ環境ではプログラムや共有データの割り当てを注意深く行い、リモートアクセスを避けるデータ配置が必要である。

2.2 データ分割問題

リモートアクセスを避けるには、タスクをプロセッサに割り当てる際に、そのタスクがアクセスするデータをローカルメモリに配置することが理想的である。しかし、タスクとデータは多数存在し、それらの関係は非常に複雑で、理想的な配置を行うことは困難である。この問題は、代表的な N-P complete 問題である箱詰め問題に相当する。

そこで、データとタスクのアクセスコストを定義し、アクセスコストの大きいデータを優先的に、ローカルメモリに配置することを考える。従って、具体的なアクセスコスト値が必要になるが、プログラマがアクセスコスト値を推し量ることは困難であるため、自動並列化コンパイラの内部表現からアクセスコスト値を取り出す。

2.3 データ分割グラフ (DPG)

現在、奈良女子大学、奈良先端技術大学院大学、和歌山大学では、Narafrase[1, 3] という自動並列化コンパイラを共同開発している。自動並列化コンパイラは中間表現という内部表現を持ち、これに対し解析・合成を行って並列化を行う。Narafrase は分散メモリ環境を対象としており、中間表現に制御フロー、制御依存、タスクの変数アクセス（頻度、データ依存を含む）、ループのネストレベルを同時に表現するデータ分割グラフ (DPG) を採用している。DPG は、2 種類の node と 3 種類のエッジで構成されている。2 種類の node は、C-node、D-node と呼ばれ、それぞれタスクと変数を表している。また、3 種類のエッジのうち 2 つは、C-node 間に存在し、制御フロー、制御依存エッジと呼ばれ、制御の流れ、制御依存を表している。残りのエッジは、データ・アクセス・エッジと呼ばれ、DPG 特有のもので、データ依存情報に加え、タスクの変数アクセス情報を表現している。DPG の詳細については、文献 [4] を参照していただきたい。我々は、DPG の変数アクセス情報を用いるデータ分割手法を提案する。

3 データ分割手法

3.1 本提案の目的と方針

人間の視覚認識能力が優れていることは、広く知られている。そこで、N-P complete 問題であるデータ分割問題を打開するために、人間の視覚認識能力を用いることを提案する。人間の判断を仰ぐためには、分割を示唆するためにアクセスコストの関係を分かりやすく提示しなければならない。そこで、人間が直感的に把握できるように、データとタスクを視覚化し、それらの位置関係によってアクセスコストを表現する。つまり、データとタスクの距離がアクセスコストの大きさを反映する。そこで、データとタスクの距離を調節する引力を本視覚化システムに応用する。ただし、我々が提案する引力は、物理モデルである引力を忠実に再現したものではなく、アクセスコストを表現するために応用していることに注意されたい。引力の大きさはアクセスコストに依存し、アクセスコストの大きいデータ・タスク間ほど、大きな引力が働く設定である。つまり、アクセスコストの大きいデータ・タスク間ほど互いに強く引き合い、近づき合う。そして、これらの移動が落

ち着いてきたところで、人間がクラスタの決定を行い、データ分割問題に近似解を導く。

3.2 DPGの視覚化

DPGからタスク、変数の大きさ、タスクの変数アクセス情報を抽出する。各タスクとデータを球と円盤で表現する。データの大きさに応じて、円盤の面積を調節する。これにより、データがメモリを占める相対的な大きさが示される。次に、アクセスのあるデータ・タスク間は線で結び、アクセスの種類を分類するために、Readアクセスエッジは緑、Writeアクセスエッジは赤で描画する。さらに、アクセスのあるデータ・タスク間にアクセスコストを反映した引力を定義する。次節において、我々が提案する引力の詳細について述べる。

次に、各データとタスクの配置についてであるが、1つの大きな球を用いてレイアウトする。データとタスクの間の距離はアクセスコストの大きさ表現するため、ある点から一定距離にある球面にデータを配置することにより、引力により各タスクがどのように中心からズレているかということで、引力つまりアクセスコストの局所性やバランスを把握できる。DPGの視覚化については、文献[2]を参照していただきたい。

3.3 データとタスクのクラスタ化支援のための物理モデル

まず、各タスクの Δ 時間後の座標を求め、次にデータの Δ 時間後の座標を求める。この処理を繰り返すことにより、引力に従ったデータとタスクのクラスタ化が行われる。各データ、タスクの Δ 時間後の座標は、それにアクセスするデータまたはタスク間の引力で決定される。各アクセスごとに以下の式を適用し、 Δ 時間後のデータとタスクの座標を決定する。以下には、データの Δ 時間後の x 座標を求める式を示すが、 y, z 座標やタスクにおいても同様の式を適用する。

$$F_x = F * \frac{X_d - X_c}{\sqrt{(X_i - X_c)^2 + (Y_i - Y_c)^2 + (Z_i - Z_c)^2}}$$

$$A_x = \frac{\sum_{k=1}^n F_x}{M}$$

$$X_{i+1} = A_x * \Delta^2 + X_i$$

ただし、 Δ :時間、 F :引力の大きさ、 F_x :引力の x 成分、 X_i :現在のデータの x 座標、 X_{i+1} : Δ 時間後の x 座標、 X_c :タスクの x 座標、 Y_c :タスクの y 座標、 Z_c :タスクの z 座標、 M :質量、 A_x :加速度の x 成分、 n :アクセス数を示す。

3.4 クラスタ化支援

我々が提案する物理モデルを適用すると、各データとタスクはアクセスコストに従って近づき合う。これらの動きをシステム全体として観測すると、データとタスクのクラスタ化が始まっている。このクラスタ化はアクセスコストを反映しており、視覚認識能力をもちいたクラスタ決定の支援となる。

次に、物理モデル適用後の視覚化の一例として、図1を挙げる。データとタスクの配置を観測すると、直感的に2つのクラスタの存在を確認することができる。また、2つのクラスタ間にはアクセスを表現する線が存在し、これらのクラスタで分割を行った場合のリモートアクセスを示している。

4 結論

本稿では、分散メモリ環境におけるデータ分割問題を、新たなる視点で考察するために、データとタスクのアクセス情報を視覚化した。これにより、データとタスクの関係を直感的に把握できるようになった。さらに、当該視覚化システム上で物理モデルである引力を応用し、クラスタ化を行うことで、人間の視覚認識能力を用いたデータ分割の支援を行った。

今後、本視覚化システムの改良について検討している。本手法の引力のみによる移動では、ユーザがクラスタを決定する配置にすることが困難である。そこで、本システムに斥力など新たなる手法を加え、分割を示唆する配置(例: 図2)を目指す。また、決定されたクラスタをコンパイラにフィードバックする機能を備えていない。そこで、本ユーザの判断を自動並列化コンパイラに反映する機能を実現させたい。実装に関しては、ループボディのみにとどまっており、本視覚化システムがプログラム全体を扱えるような改善も加えたい。



図 1: 視覚化の例

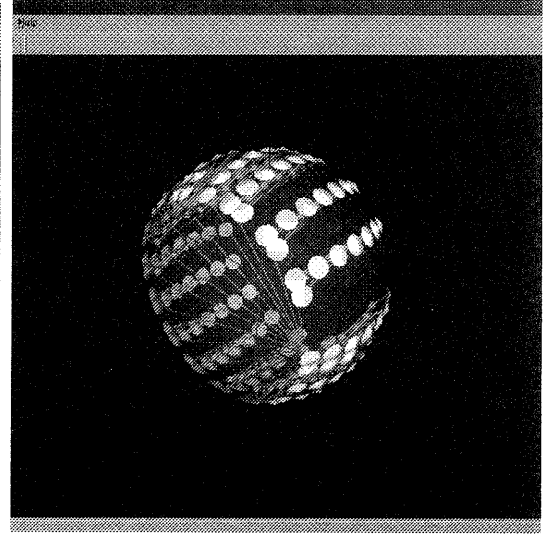


図 2: クラスタ化の例

参考文献

- [1] K.Kambe, T.Nakanishi, K.Joe, Y.Kunieda, F.Kako: *An Implementation of Loop Transformations with a Universal Intermediate Representation Interface Library*, PDPTA'99, pp.1905-1911 (1999).
- [2] 杉田望代、笹倉万里子、城和貴: データ分割グラフの3次元視覚化, 情報処理学会研究報告、MPS-26-6, pp.21-24 (1999).
- [3] 羽田昌代、神戸和子、中西恒夫、城和貴: 自動並列化コンパイラの統一的中間表現とインターフェースを用いたコード変換の実装, 情報処理学会研究報告、ARC-134-26, pp.151-156 (1999).
- [4] T.Nakanishi, K.Joe, H.Saito, C.Polychronopoulos, A.Fukuda, K.Araki: *The Data Partitioning Graph: Extending Data and Control Dependencies for Data Partitioning*, LCPC, pp.170-185 (1994).
- [5] T.Nakanishi, K.Joe, H.Saito, A.Fukuda, K.Araki: *The CDP² Algorithm: A Combined Data and Program Partitioning Algorithm on the Data Partitioning Graph*, LCPC, pp.170-185 (1994).
- [6] M.Sasakura, K.Joe, Y.Kunieda, and K.Araki: *NaraView: An Interactive 3D Visualization System for Parallelization of Programs*, Int'l J. of Parallel Programming, Plenum Pub., Vol. 27, No. 2, pp.111-129 (1999)
- [7] *OpenGL Reference Manual*, Addison-Wesley, 2nd edition (1995).