

レギュラーサンプリングを用いる

3G-1

局所分布に着目した改良並列ソート*

山岸 秀規[†]東京大学大学院理学系研究科情報科学専攻[‡]

1 はじめに

本稿では、アーキテクチャに特化しない並列ソートアルゴリズムとして、レギュラーサンプリング [1] に基づく改良並列ソートアルゴリズムを提案する。

レギュラーサンプリングによる並列ソートでは、抽出するサンプル数を増やすことによりプロセッサ間の負荷バランスの均一性を向上させることができるが、過度にサンプル数を増やすことは通信量や演算量の増大を招き、性能の向上が望めない。提案する手法の目的は、抽出したサンプルの局所的な分布に着目することにより、通信量や演算量をほとんど増大させずに並列ソートの負荷バランスを向上させることである。

2 モデル

2.1 想定するマシンのアーキテクチャ

これまでに研究されてきた並列ソートアルゴリズムは、アーキテクチャに特化して、その性能を最大限に発揮させるものが多い [2]。レギュラーサンプリングによる並列ソートは、特定のアーキテクチャ上ではアーキテクチャに特化したアルゴリズムよりも劣る可能性があるが、広範なアーキテクチャにおいて良好な性能を示す [1]。

本稿で提案するアルゴリズムが対象として想定する計算機アーキテクチャは、レギュラーサンプリングによる並列ソートが対象として想定するアーキテクチャと同一であるが、以下では簡単のために、議論を p 台の PE (Processing Element) から成る MIMD 型並列計算機、特に分散メモリ型並列計算機に限定し、任意の 2PE 間の通信コストは等しいと仮定する。

2.2 並列ソートのモデル

以下、並列ソートを次のようにモデル化する。なお、PE 内におけるソートのモデルについては、 $i = 1$ の場合を考えればよい。

入力

長さ n の要素列 $\{A_i\} = a_i^1, a_i^2, \dots, a_i^n$ が PE i ($0 \leq i \leq p-1$) に入力される。すなわち全 PE に対し合わせて長さ np の要素列が与えられて

いる。与えられた任意の要素列中の任意の要素を含む集合 A に対し、全順序が与えられている。すなわち、

- 任意の $a, b, c \in A$ に対し、
 $a \leq a$
 $a \leq b$ かつ $b \leq a \iff a = b$
 $a \leq b$ かつ $b \leq c \implies a \leq c$
 の全てが成立する。

- 任意の $a, b \in A$ に対し、 $a \leq b$, $b \leq a$ の少なくともいずれか片方が成立する。

出力

要素列 $\{B_i\}$ が PE i ($0 \leq i \leq p-1$) に出力される。この要素列は以下の条件を満たす。

- 要素列 B_i の長さは必ずしも n ではないが、任意の出力列中の任意の要素を含む集合 B に対し、 $A = B$ 。かつ要素列の族 $\{A_i\}_{i=0,1,\dots,p-1}$, $\{B_i\}_{i=0,1,\dots,p-1}$ 中に複数回出現する要素の重複度は等しい。
- $0 \leq i \leq p-1$ なる i および $1 \leq (l, m) \leq n$ なる l, m に対し、 $l \leq m \iff b_i^l \leq b_i^m$
- $1 \leq i \leq p-1$ なる i および $1 \leq (l, m) \leq n$ なる l, m に対して $b_{i-1}^l \leq b_i^m$

3 並列ソートアルゴリズム

3.1 レギュラーサンプリングによる並列ソート

レギュラーサンプリングによる並列ソートは、以下の手順に従って行われる [1]。なお、最初に各 PE へ入力される要素数 n に対して、 $\omega = n/p$, $\rho = p/2$ とする。

1. 各 PE は、 n 個の要素を各 PE 内でソートし、第 1, $(1+\omega)$, $(1+2\omega)$, \dots , $(1+(p-1)\omega)$ 番目の要素をサンプルとして取り出す。
2. 各 PE で選んだサンプルを決められた PE へ集めて PE 内でソートし、ソート済の要素列から第 $(p+\rho)$, $(2p+\rho)$, \dots , $((p-1)p+\rho)$ 番目の要素を軸として取り出し、これを全 PE に Broadcast する。
3. 各 PE で軸をもとに要素列を p 個の部分列に分割し、 m 番目の部分列を PE $(m-1)$ へ送る。
4. 各 PE 内で受け取った部分列を改めてソートする。

*Extended Parallel Sorting Based on Regular Sampling and Local Distribution

[†]Hidenori YAMAGISHI

[‡]Department of Information Science, Graduate School of Science, the University of Tokyo

この方法に従えば、同一の値をもつ要素が全 PE にまたがって高々 n/p 個程度以内で、かつ $n > p^2$ であれば、4. の操作における各 PE の処理すべき要素数は $(n/p + p - 1)$ 以上 $(2n - n/p - p + 1)$ 以下となることが知られている [1].

3.2 提案する方法

上に示される程度の負荷バランスの均一性で十分なケースも考えられるが、実際にはより均等な負荷分散がなされれば、性能向上にもつながり、都合がよいと考えられる。本アルゴリズムでは 3.1 節の 2. の段階で集めたサンプルの局所的な分布状態を表す統計情報に着目し、より良好な負荷バランスの均一性を与える軸を決定する。具体的には一度レギュラーサンプリングで設定された軸をもとに以下の統計情報を計算し、軸を補正するという方法をとる。

1. 軸と同じ値をもつ要素の頻度 (frequency)
2. 軸を中心とする 3 次の標準化モーメント、歪度係数 (skewness)
3. 軸を中心とする 4 次の標準化モーメント、尖度係数 (kurtosis)

まず統計情報 1. に関し、その値を限度として、該当する軸の近傍の要素 (同じ値をもつ要素) を PE 間でずらして PE 間の要素数のばらつきを抑えることができるので、適用可能な場合にはいつも最大限に適用する。

次に統計情報 2. および 3. に関して、軸 a を中心とする m 次の標準化モーメント α_m は、次式で求められる。

$$\alpha_m = \frac{1}{\sigma^m} \sum (X - a)^m$$

ここで、 X はモーメントを計算する対象となる要素を意味する。また、 σ は標準偏差 (standard deviation) を表し、次式で求めることができる。式中、 n は計算の対象となる要素 X の総数を表す。

$$\sigma = \sqrt{\frac{1}{n} \sum (X - a)^2}$$

統計情報 2. は、各要素と軸との相対的な大小関係を 3 乗することにより、分布の偏りを符号として保存しつつ標準化 (標準偏差の 3 乗で割ってある) することで、絶対的な偏りの度合を表している。

統計情報 3. は、近傍に 4 次のオーダーに勝るだけの分布があるかどうか、近傍以外からの影響がどの程度あるかを標準化した量であり、直感的にはこの値が大きいほど分布は「尖っている」ことになる。「尖っている」分

布の近傍では、要素のわずかな大小差で、大幅に順位が変わってしまう。

これらを適用して、軸を動かした場合に全体としての程度の影響が出るかを推測することができれば、PE 間で通信を行なうことなく負荷バランスを向上させることが可能であり、性能向上につながると考えられる。

4 実装方法

今回の実装では、実際に要素交換を行なう前の段階で、まず負荷のギャップが 10% を超える PE があるかどうかを調べる。このような PE が検出された場合、該当する軸の近傍で統計情報 3. から判断される尖度が正規分布よりも緩やか ($\alpha_4 < 3$) であるかどうかを調べる。そして、以上の 2 つの条件が満たされる場合にのみ、統計情報 2. から判断される偏りを加えて抽出・整列済のサンプル中を幾分かずらして軸を選び直す。

負荷の偏りの許容限度は、アドホックに 10% がよいと判断した。正規分布よりも緩やかな分布のみを対象とするのは、あまりに急激な変位を伴う分布に対して補正をかけると、逆効果になるおそれがあるためである。

5 まとめ

レギュラーサンプリングに基づき、サンプルの局所的な分布情報に着目してプロセッサ間の負荷バランスの均一性を向上させる並列ソートアルゴリズムについて述べた。本手法により 3.1 節で示した純粋なレギュラーサンプリングよりも性能が向上することを期待している。

このアイデアに基づく並列ソートアルゴリズムを、分散メモリ型並列計算機 HITACHI SR2201 上に実装し、性能評価を行う。結果は講演当日に発表する予定である。

参考文献

- [1] Xiaobo Li, Paul Lu, Jonathan Schaeffer, John Shillington, Pok Sze Wong and Hanmao Shi: On the Versatility of Parallel Sorting by Regular Sampling, 1993.
- [2] Selim G.Akl: Parallel Sorting Algorithm, Academic Press, 1985.
- [3] Andrew Tridgell, Richard Brent and Brendan McKay: Parallel Integer Sorting, 1997.
- [4] 東京大学教養学部統計学教室: 統計学入門, 東京大学出版会, 1991.