

# 分散メモリ並列計算機およびワークステーションクラスタを用いた 選択問題における通信操作の性能評価\*

17D-3

12B-3

牧野 浩之 渋沢 進†

茨城大学工学部情報工学科‡

## 1 はじめに

$N$  個の要素からなる系列  $S = \{s_0, s_1, \dots, s_{N-1}\}$  があり、 $1 \leq k \leq N$  なる  $k$  番目に小さい要素を見つけ出す問題を選択問題という [1, 2]。この問題は、ランキングを出す場合など、データベースや数理統計学などの分野において現れる。選択問題を並列化した並列選択アルゴリズムについては、計算量の観点から評価したものが多い [2]。しかしながら、並列プログラミング時に考慮すべき事項の 1 つである通信操作の観点から評価したものは少ない。

そこで本研究では、選択問題の実行時間を通信操作および通信時間の観点から評価する。また、プロセッサ間の計算量を出来るだけ均等化する手法を取り入れた場合の実行時間と、取り入れない場合の実行時間の比較についても行う。

## 2 並列選択

### 2.1 並列アルゴリズム

使用するプロセッサ数  $p$  およびランク  $k$  は与えられているものとする。本アルゴリズムでは、サーバの役割をするプロセッサとクライアントの役割をするプロセッサに分割する。サーバとなるプロセッサは 1 プロセッサ ( $P_0$ ) とし、残りのプロセッサ ( $P_1, \dots, P_{p-1}$ ) は全てクライアントとなる。

サーバで系列  $S$  を生成し、 $S$  を  $p$  個の部分系列  $S_i (0 \leq i \leq p-1)$  に分割する。その部分系列  $S_i$  を全プロセッサ  $P_i$  へ送る。本研究で設計した並列アルゴリズムを Algorithm 1 に示す。

#### Algorithm 1 para\_select( $S, k$ )

1. Decide the separator  $M$  on  $P_0$
2. Broadcast  $M$
3.  $P_i$  splits  $S_i$  into two subsets  $S_{iL}$  and  $S_{iG}$
4. Derive the reduction  $|S_{iL}|$   
/\*  $P_0$  has value  $\sum_{i=0}^{p-1} |S_{iL}|$  \*/
5. if  $k \leq \sum_{i=0}^{p-1} |S_{iL}|$  then
6. para\_select( $S_L, k$ )
- else
7. para\_select( $S_G, k - \sum_{i=0}^{p-1} |S_{iL}|$ )
- endif

end

\*Performance evaluation of communication operations for parallel selection using a distributed memory parallel computer and workstation cluster

†Hiroyuki Makino and Susumu Shibusawa

‡Department of Computer & Information Sciences, Faculty of Engineering, Ibaraki University

### 2.2 並列プログラムにおける負荷均等

本節では、並列プログラムにおいて、プロセッサ間で要素のやりとりを行う方法について述べる。2.1 節で示した並列アルゴリズムにおいては、各プロセッサが持っている要素数に差が生じる。そのため、プロセッサ間で計算時間に差が生じ、結果として並列処理の効率が落ちると考えられる。

そこで本研究では、各プロセッサが持つ要素数を出来るだけ均等にする手法を導入し、その効果を検証した。以下に負荷均等の概要を示す。

1. if  $P_0$  then
2. 各プロセッサが持つ要素数情報  $|S_i|$  を収集する
3.  $|S_i|$  を、要素数を基準にソートする  
/\*  $k$  番目の  $|S_i|$  は  $|S_i|_{(k)}$  とする \*/
4.  $|S_i|_{(k)}$  と  $|S_r|_{(p-k+1)}$  なる  $P_s, P_r$  がペアを組む
5. ペア情報を分配する
- endif
6. 各プロセッサは、ペアを組んだプロセッサと通信を行い、ペア間で要素数の均等化を行う。

本研究においては、送信プロセッサ  $P_s$  から受信プロセッサ  $P_r$  へ移動する要素数  $|S_{sr}|$  を (1) 式で表す。なお、 $|S_s|$  を  $P_s$  の要素数、 $|S_r|$  を  $P_r$  の要素数とする。

$$\begin{aligned} |S_{sr}| &= \lfloor (|S_s| - |S_r|) / 2 \rfloor && (|S_s| - |S_r| > t \text{ のとき}) \\ |S_{sr}| &= 0 && (|S_s| - |S_r| \leq t \text{ のとき}) \end{aligned} \quad (1)$$

$0 \leq t \leq 2$  のとき、負荷均等後に (2) 式が成り立つ。

$$0 \leq (|S_s| - |S_r|) \leq 1 \quad (2)$$

図 1 に  $p = 8, t = 2$  における要素の均等化の例を示す。

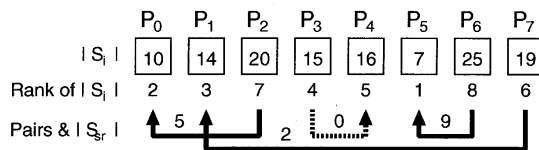


図 1: 負荷均等化の例 ( $p = 8, t = 2$ )

## 3 測定について

### 3.1 測定環境

本研究では、並列計算機 SR2201 およびワークステーションクラスタ環境を用いて測定を行った。

**並列計算機 SR2201:** 並列計算機 SR2201 は、分散メモリ型で構成されている。プロセッサ間通信は、ノード間を結ぶクロスバネットワークを通して行

われる。なお、本研究で使用したSR2201の最大プロセッサ数は12である。

**ワークステーションクラスタ:** 本研究では、32台のワークステーション(WS)をEthernetで接続した環境を用いた。

### 3.2 測定条件

本研究では、使用するプロセッサ数を一定にして全体の要素数を変化させる方法、および要素数を一定にして使用プロセッサ数を変化させる方法を用いて測定した。なお、本研究では同一のプログラムを10回実行し、10回測定して得られたデータのうち、実行時間が最も短いデータを測定値として採用した。

## 4 測定結果および考察

### 4.1 要素数を変化させた場合

並列計算機を用いて、 $p=8, t=2$ にして要素数 $N$ を1万から1000万要素まで変化させた場合の実行時間 $T_{exec}$ および通信時間 $T_{comm}$ 、データの再配置にかかった時間 $T_{place}$ を図2に示す。ここで、 $T_{place}$ は要素そのものに対する通信にかかった時間を表し、 $T_{comm}$ の内数とする。また、計算時間 $T_{comp}$ を(3)式で表す。

$$T_{comp} = T_{exec} - T_{comm} \quad (3)$$

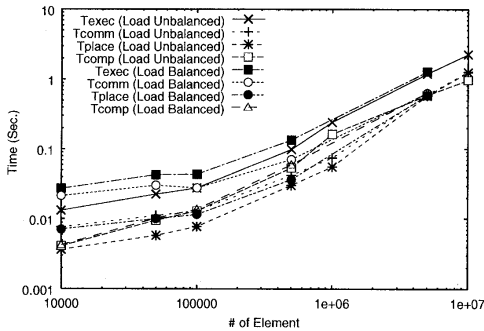


図2: 要素数を変化させた場合の時間(並列計算機)

並列計算機およびクラスタ環境においても、 $T_{comm}$ の多くは $T_{place}$ で占められていた。データの再配置に要する処理を考察すると、プログラム開始直後に発生する要素の分配がデータ再配置時間の多くを占めていた。このため、要素数が多くなると、再配置に占める通信時間の割合が増し、並列計算機において要素数が1000万の場合には、その割合は98%であった。

次に、本研究で導入したデータの再配置に伴うプロセッサ間の負荷均等化の効果について検討する。 $t=2$ のとき、プロセッサペア間で要素数の差が2以上であれば、必ずプロセッサ間で要素の送受信が行われる。すなわち、1回の負荷均等処理につき、最大で $p/2$ 回の通信が起きる。そのため、実行時間全体に対する通信処理の割合が増加し、実行時間に対して通信時間の割合が大きい環境においては、負荷均等を行った場合での実行時間は、負荷均等を行わなかった場合の実行時間より長くなった。これより、 $t$ の値を大きくする、すなわち送受信が行われる回数を減らすことにより、負

荷均等の効果が期待できると考えられる。

### 4.2 使用プロセッサ数を変化させた場合

次に、要素数 $N$ を100万、 $t=2$ にして使用プロセッサ数を変化させた場合の $T_{exec}$ および $T_{comm}$ 、 $T_{place}$ について、クラスタ環境での結果を図3に示す。

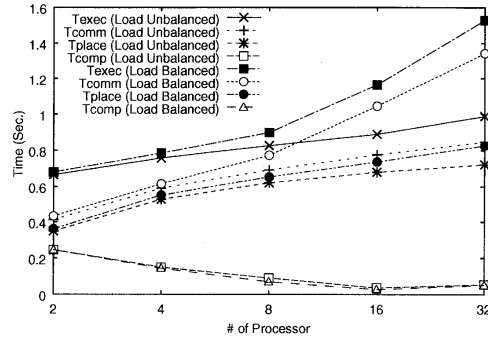


図3: プロセッサ数を変化させた場合の時間(クラスタ)

並列計算機およびクラスタ環境とも、プロセッサ数の増加に従って各プロセッサにおける計算時間は減少していた。これは、各プロセッサに割り当てられている要素数が減少しているからである。一方、通信時間に関しては、並列計算機環境では概ね一定であるのに対して、クラスタ環境では、通信時間はプロセッサ数が増加するにつれて単調増加している。また、実行時間に占める通信時間の割合は、並列計算機では最大36%であるのに対して、クラスタ環境においては最大88%となっていた。このため、クラスタ環境での実行時間は、通信時間の影響を大きく受け、プロセッサ数の増加に対して単調増加している。これより、クラスタ環境では通信経路の品質によって性能が向上すると考えられる。

## 5 おわりに

本研究では、並列計算機SR2201およびWSクラスタ環境を用いて、並列選択問題における通信操作の性能評価を行った。その結果、クラスタ環境における実行時間は通信操作の影響を大きく受けていた。すなわち、クラスタ環境での分散処理の効果は、プロセッサ間を結ぶネットワークの品質によって期待できると考えられる。また、各プロセッサでの計算時間を均等化させるために行った負荷均等処理では、プロセッサ間での要素の送受信が多数発生するため、結果として通信量が増加することが分かった。

今後の課題としては、データの再配置に発生する通信操作についての解析や、異機種環境における実験などが挙げられる。

## 参考文献

- [1] S. G. Akl. *The Design and Analysis of Parallel Algorithms*. Prentice-Hall, 1989.
- [2] I. Al-furiah, S. Aluru, S. Goil, and S. Lanka. Practical algorithms for selection on coarse-grained parallel computers. *IEEE Trans. on Parallel and Distrib. Sys.*, Vol. 8, No. 8, pp. 813-824, 1997.