# XML データ分割処理の並列化に関する研究

1150379　山本　佳祐　【 松崎研究室 】

## 1　はじめに

近年のデータの肥大化に伴い, XML データに対する並列処理の必要性が高まっている。木構造である XML データに対する並列処理において, 前処理として $m$-bridge [1] を用いた分割処理によって親子/兄弟関係を維持した分割をすることが有用である。しかし m-bridge を用いた分割処理を並列に行う手法は, まだ確立していない。

そこで本研究では $m$-bridge による分割処理を並列化したものを実装し, その実験結果を報告する。

## 2　$m$-bridge を用いた分割処理

分割対象の木構造のノード数を $n$ とする。木構造の任意のノード $v$ について, $W(v)$ を $v$ 以下のノード数, $C(v)$ を $v$ の子ノード $v'$ に対する $W(v')$ の最大値とする。ある定数 $m$ を用いて $m$-bridge を用いた分割処理では, 以下の条件式 (1) を満たすノード $v$ の直下で木を分割する。

$$\left\lfloor \frac{W(v)}{m} \right\rfloor \neq \left\lfloor \frac{C(v)}{m} \right\rfloor \tag{1}$$

条件式を満たすノードを探索するためには, 各ノード $v$ について $W(v), C(v)$ を求める必要がある。ここで, $C(v)$ を求めるには, すべての子ノード $v'$ について $W(v')$ を全て求める必要がある。そのため, $m$-bridge を用いた分割処理では以下の 3 つの処理を順に行う。

1. 全てのノード $v$ で $W(v)$ を求める。
2. 全てのノード $v$ で $C(v)$ を求める。
3. 条件式を満たすノードを探索し, その直下で分割する。

本研究では $m$-bridge を用いた分割処理の並列化を, 分割処理の各段階をそれぞれ並列化することによって行った。

## 3　木の並列縮約アルゴリズムを応用した累積計算

$m$-bridge を用いた分割処理において, あるノード $v$ の $W(v)$ を求めるには, その子ノード $v'$ の $W(v')$ の値が求まっている必要がある。このことは, $W(v)$ を求める処理が, 木の下方から上方へ値を求める処理であることを意味している。このような上向きの処理を並列に行う手法として, 木の並列縮約アルゴリズムがある。本研究では, 木の並列縮約アルゴリズムを応用した累積計算 [2] によって $W(v)$ を求める。

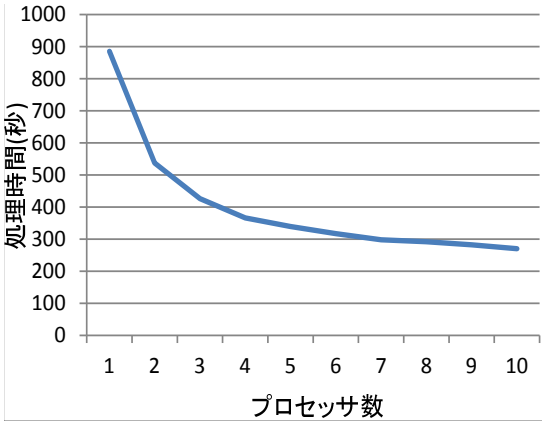


図 1　プロセッサの台数ごとの処理時間

具体的な方法としては, 下方から上方へと縮約される値を各ノードの値について記録することで $W(v)$ を求める。また $C(v)$ についても, 同様のアルゴリズムによって求めることが可能である。

その後, 全ノードから条件式を満たすノードを探索しマーク付けを行う。

データの分割とファイルへの書き出しは, XML データの文字列を並列縮約アルゴリズムを用いて結合し, その過程でマークが付けられたノードが出現した場合に新しいファイルへ書き出すことで行う。

## 4　実験結果

3.1GB の XML データに対して 10 台構成のクラスタを用いて, 並列に $m$-bridge を用いた分割処理の $W(v)$ を求める実験を行った。その結果を図 1 に示す。

図 1 では, 台数の増加に伴って処理時間が減少しており, 処理の並列化によって処理の高速化が実現できたことが分かる。

## 5　まとめ

本研究では $m$-bridge を用いた分割処理の並列化の方法を提案し, その成果を報告した。実験の結果より並列化による処理の高速化を確認した。

## 参考文献

[1] H. Gazit, G. L. Miller, S.-H. Teng. Optimal Tree Contraction in the EREW Model. In *Concurrent Computations*, pp. 139–156, 1988.

[2] K. Matsuzaki and R. Miyazaki: Parallel Tree Accumulations on MapReduce. In *7th International Symposium on High-level Parallel Programming and Applications*, pp. 31–50, 2014.