

# ヘテロジニアスな分散処理環境における地理情報処理のための処理時間予測の検討

五大開発株式会社	正会員	○荒木 光一
石川工業高等専門学校	正会員	新保 泰輝
五大開発株式会社	非会員	柏 大吾
五大開発株式会社	非会員	藤原 大祐
五大開発株式会社	正会員	鱈 洋一

## 1. はじめに

多くの地理情報処理は対象とする地域を分割した図郭単位で行われるため、高性能な GIS では複数のコンピュータによる分散処理を行うことで高速化を図っている. 文献[1]では分散処理を利用した GIS において、各図郭 (データ) の処理時間が異なる地理情報処理を各コンピュータの処理時間が平準化するように、各図郭を各コンピュータに手動で割り当てることで高速化を実現している.

しかしながら、このような平準化を自動で行うためには各図郭の処理時間の情報が必要である. また、組織内で遊休しているコンピュータを用いる場合、各コンピュータのスペックが異なるため、ヘテロジニアス(異種混在)な分散処理環境となる. そのため、各コンピュータのスペックを考慮して各コンピュータへの割り当てを行う必要がある. これらのことから、ヘテロジニアスな分散処理環境を対象とした地理情報処理の処理時間予測手法が必要である. そこで、本稿ではその基礎検討として、ヘテロジニアスな分散処理環境における地理情報処理時間の予測手法を提案し、その予測精度を評価する.

## 2. システムモデル

図-1 に、本稿で想定するヘテロジニアスな分散処理環境のシステムモデルを示す. 図中の計算ノードは地理情報処理を行うコンピュータであり、各計算ノードのスペックは異なる. また、図中のファイルサーバには地理情報処理を行う実行ファイルと、地形データなどの入力データが保存されている. 計算ノードの処理結果もファイルサーバに保存される. 各計算ノードとファイルサーバはネットワークで接続されている.

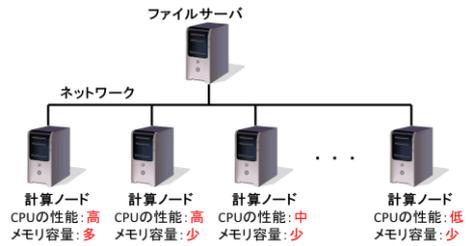


図-1 システムモデル

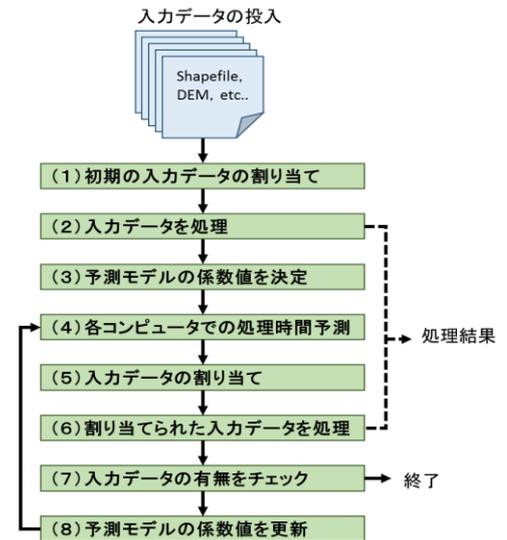


図-2 提案する処理時間予測手法のフロー

## 3. 処理時間予測手法

本稿では、地理情報処理時間の予測手法として計算ノード毎に予測モデル式を構築する手法を提案する. 本提案手法では、各計算ノードに全入力データを一度に割り当てず、適度に分割して割り当てている. 割り当てた処理が完了する度に、各計算ノードの予測モデル式の係数値は実処理時間と入力データを参照して更新される.

図-2 に、提案する処理時間予測手法のフローを示す. (1)処理時間が短い入力データを各計算ノードに適度に割り当てる. (2)各計算ノードで処理を行い、実処理時間を取得する. (3)と(4)では実処理時間と入力データを元に Levenberg-Marquardt 法 (LM 法) で各予測モデル式の係数値を決定し、未処理の入力データの処理時間を予測する. (5)と(6)では予測処理時間を参照して処理時間が平準化するように、適度な入力データを割り当てる. 各計算ノードで処理を行い、各入力データの実処理時間を取得する. (7)は全入力データの処理が完了したかを確認する. (8)これまでの実処理時間を元に LM 法で予測モデル式の係数値を更新する.

キーワード 地理情報処理, GIS, ヘテロジニアス, 分散処理, 処理時間予測

連絡先 〒921-8051 石川県金沢市黒田1丁目35番地 五大開発株式会社 TEL076-240-6588

表-1 評価環境

コンピュータ	OS	CPU	メモリ容量(GB)
ファイルサーバ	CentOS 6.7 64bit 版	Intel Core i7-3820 (3.60GHz)	64
計算ノード 1	Windows 7 Pro 64bit 版	Intel Core i7-6700K (4.00GHz)	8
計算ノード 2	Windows 7 Pro 64bit 版	Intel Core i7-6700K (4.00GHz)	8
計算ノード 3	Windows 7 Pro 64bit 版	Intel Core i7-920 (2.67GHz)	8
計算ノード 4	Windows 10 Home 64bit 版	Intel Core i5-5200U (2.2GHz)	8
計算ノード 5	Windows 7 Pro 64bit 版	Intel Core i7-5820K (3.30GHz)	64

4. 予測精度の評価

予測精度評価として、等高線(端点と標高)を入力とする傾斜区分図作成プログラムに対して 500 個の Shapefile を用いて評価した。このプログラムは制約付きドロネー三角形分割後、三角形の傾斜を算出するものである。Visual C++ 2012 で実装し、分散処理のライブラリには MSMPI v6 を用いた。表-1 に評価環境を示す。ファイルサーバと各計算ノードは 100Mbps のイーサネット接続した。

処理時間予測手法で指定する処理時間予測モデル式は、図-2(2) で取得した実処理時間を参考に  $y=ax^2$  とした。ここで、 $x$  は Shapefile のファイルサイズであり、 $a$  は係数値である。プログラムの起動には MSMPI の affinity と priority 4 のオプションを付属した。

(1) 単体計算ノードでの予測精度の評価

単体計算ノードでの予測精度評価として、ファイルサーバと計算ノード 1 を用いて検討した。500 個の Shapefile から無作為に選んだ 1 個の Shapefile に対する予測処理時間と実処理時間の誤差で評価した。図-2(1)(5) で 100 個の Shapefile を計算ノードに割り当て、それらの実処理時間を元に図-2(3)(8) で  $y=ax^2$  の  $a$  を決定・更新する。その後、追加の Shapefile を入力し、図-2(4) から (8) を繰り返した。表-2 に、計算ノード 1 での予測精度の比較を示す。サンプル数  $n$  は予測モデル式の係数値を決定・更新するために用いられる実処理時間のデータ数であり、計算済みのデータ群から無作為に抽出した実処理時間の数である。サンプル数 500 の場合、全ての実処理時間を用いた評価となる。表-2 より、サンプル数が増加するにつれて、予測処理時間は実処理時間に近くなり、誤差は低下している。したがって、図-2(4) から (8) の繰り返しによって予測精度は向上したといえる。

(2) ヘテロジニアスな分散処理環境での予測精度の評価

表-3 にヘテロジニアスな分散処理環境での予測精度の比較を示す。比較対象(表中の 20 サンプル)は、実処理時間が短い 100 個の Shapefile を各計算ノードに 20 個割り当て、その実処理時間から LM 法と  $y=ax^2$  で予測したものである。平均誤差は各計算ノードに割り当てられた Shapefile の予測処理時間と実処理時間の誤差の平均である。計算ノード 1, 3 から 5 では、提案した処理時間予測手法は予測モデル式の係数値を逐次更新することで平均誤差を低減できた。一方、計算ノード 2 では平均誤差が増加した。これは、計算ノード 2 に割り当てた 20 サンプル(予測手法:20 サンプル)は、図らずも、予測モデル式の係数値を適切に決めるものであったことが原因である。

5. おわりに

本研究では、ヘテロジニアスな分散処理環境における処理時間予測手法を提案した。処理時間が短い 20 サンプルのみで予測した手法と比較して、提案予測手法は処理時間の平均誤差を低減できた。

今後は、複数の予測モデル式の中から精度の高いモデル式を自動選択する手法と、回帰分析と相関係数により処理時間に影響しているパラメータを判断する手法について研究し、より精度と汎用性を高める予定である。

参考文献

[1] N. Stojanovic and D. Stojanovic, "High-performance computing in GIS: techniques and applications", International Journal of Reasoning-based Intelligent Systems, Vol.5, No.1, pp.42-49, 2013.

表-2 計算ノード 1 における予測精度の比較

サンプル数 $n$	予測処理時間 (sec)	実処理時間 (sec)	誤差 (%)
100	168	342	50.98
200	118		65.62
300	146		57.47
400	282		17.83
500	316		7.78

表-3 分散処理環境における予測精度の比較

コンピュータ	予測手法	平均誤差(%)
計算ノード 1	20 サンプル	338.74
	提案予測手法	94.85
計算ノード 2	20 サンプル	14.74
	提案予測手法	45.54
計算ノード 3	20 サンプル	72.65
	提案予測手法	50.05
計算ノード 4	20 サンプル	1693.06
	提案予測手法	473.57
計算ノード 5	20 サンプル	41.35
	提案予測手法	30.76