# 3R-02　データマイニングツール dataFOREST を用いた異機種分散計算機環境におけるプロセッサ負荷予測

茂田 有己光[†], 林 拓也[†], 小出 洋[††],
鹿島 亨[†††], 筒井 宏明[†††], 笠原 博徳[†]
[†] 早稲田大学理工学部電気電子情報工学科, [††] 日本原子力研究所, [†††](株) 山武

## 1　はじめに

科学技術計算プログラムを異機種分散計算機環境上で並列・分散実行するためのメタスケジューリング手法 [1] の効果的な実現のためには, 各計算機のプロセッサ負荷やネットワーク負荷の予測値を用いてタスクの各計算機への割り当てを決める必要がある. このような分散計算機環境での動的負荷分散の代表的な研究としては Legion[2],Ninf[3] 等があり, 動的負荷予測の研究としては, 線形時系列モデルを用いて予測を行う研究 [4] や,Network Weather Service(NWS) における複数の予測モジュールのうち最も誤差の少ないモジュールを用いて予測を行おうとする研究 [5] 等がある. 本稿では, データマイニングツール dataFOREST を用いて, 位相論に基づいた TCBM(Topological Cased-Based Modeling)[6] でモデリングを行い, 日本原子力研究所計算科学技術推進センターに設置されている COMPACS 上の資源情報サーバ [7] によって収集されたプロセッサ負荷情報をもとに, プロセッサ負荷の予測を行う手法を提案し, 評価を行う.

## 2　メタスケジューリング

メタスケジューリング [1] は, ネットワークで接続された種類の異なる複数の計算機で構成された異機種分散計算機環境において, 単一の科学技術計算プログラムを各計算機に分散し, 計算所要時間を最小化しようとする技術である. 本研究では, 日本原子力研究所計算科学技術推進センターに設置されている COMPACS(COMplex PArallel Computer System, 日立 SR2201,SR2201 デスクサイドモデル,NEC SX4 等から構成される) 等の異機種分散計算機環境を対象例とする. メタスケジューリングの具体的な手順は, まずユーザが用意した逐次プログラムを Oscar マルチグレイン並列化コンパイラ [8] により BPA(Block of Pseudo Assignment statements),RB(Repetition Block),SB(Subroutine Block) の 3 種類のマクロタスクに分割し, それらのマクロタスク間の並列性を制御依存・データ依存を考慮した最早実行開始条件解析により抽出する. 次に, 分割されたマクロタスクを実行割り当て時点で最も速く処理できる計算機にスケジューリングするメタスケジューラルーチンをコンパイラが生成し, マクロタスク間に埋め込み並列化されたプログラムとして Stampi(異機種分散処理に MPI-2 を拡張したもの)[9] を用いたプログラムとして生成する. 生成されたプログラムは COMPACS 上の各計算機それぞれのネイティブコンパイラでコンパイルされ実行される. プログラムの実行時には, メタスケジューラが各タスクの実行時間が短くなるように, 計算機に自動的に割り当てる. このときメタスケジューラは,COMPACS 上の各計算機のプロセッサ負荷やネットワーク負荷を常時監視し, 負荷情報を収集している資源情報サーバ [7] から負荷予測値を受けとり, それを用いてマクロタスクの割り当てを行う. 本論文はこのプロセッサ負荷の予測を, データマイニングツール dataFOREST を用いて行う手法を提案する.

## 3　プロセッサ負荷予測手法

本章では, 今回予測に用いたデータマイニングツール dataFOREST の概要とその中核となるモデリング手法ならびに予測手法について述べる.

### 3.1　dataFOREST

dataFOREST は, 蓄積されたデータの集合から規則性や関係を利用して情報を抽出するデータマイニングツールであり, 要因間の関係が連続である情報を TCBM(Topological Case-Based Modeling)[6] によりモデリングする. この TCBM は, 事例ベースの枠組みを考慮した位相理論に基づくもので, 入出力関係の連続性が成り立つ一般的な対象に適用可能である。

### 3.2　TCBM

TCBM で扱う形式は

量子化入力値 : $X_1$,$X_2$,$\cdots$,$X_N$
事例データ数 : n
事例出力値 : Y

であり,

モデリングは, 事例出力値に対し寄与の高い順に入力値の選択を行い, ユーザーがモデルの誤差として許容できる範囲を示した出力許容誤差

$$\varepsilon = \frac{\text{モデルとして許容できる絶対誤差}}{Y_{max} - Y_{min}} \times 100[\%]$$

を用いて評価指標により入力空間の量子化を行うことにより行われる. ただし, ここで $\varepsilon$: 出力許容誤差 [%],$Y_{max}$: 事例出力値の最大値,$Y_{min}$: 事例出力値の最小値である.
評価指標には, 出力分布充足率と連続性条件充足率の二つがあり, 出力分布充足率は,

同一量子に存在する出力値の分布$< \varepsilon$

を満たす事例の全事例に対する比率を表す評価指数で, この値が低い場合はモデルデータにノイズが多く含まれていたり入力変数の選択が良くないことが想定されるので, ノイズ処理や入力変数の選択の見直しを行うべきであることを示す. また連続性条件充足率は,

ある事例の出力値 − 周りにある事例の出力値の平均$< \varepsilon$

を満たす事例の全事例に対する比率を表す評価指数で, この値が低い場合はモデルデータの計測間隔が荒い, モデルデータにノイズが含まれている, 連続的に変化する対象でないことなどが想定されるので, 計測間隔やノイズ処理の見直し, 連続的に変化する対象であるかの再検討などが必要だとわかる.

予測値の推定は, 入力された事例と類似した事例を検索し, 同一メッシュに含まれる事例が存在すればその事例を推定値とし, 同じメッシュに同一事例が存在しなければ, 検索範囲を拡大し, 周りのメッシュの値を平均したものを推定値とすることで行う.

新たな事例の学習は, 再モデリングをせずに事例ベースを部分的に改訂することにより逐次行う. 部分的改訂の方法は, 同一事例が存在した場合は

$$Y_{new} = \frac{kY_{old} + Y}{k+1}$$

---

* CPU Load Prediction on Heterogeneous Distributed Computing Environments Using dataFOREST Data-Mining Tool
Yukimitsu MODA[†], Takuya HAYASHI[†], Hiroshi KOIDE[††], Toru KASHIMA[†††], Hiroaki TSUTSUI[†††], Hironori KASAHARA[†]
[†] Department of Electrical, Electronics and Computer Engineering, Waseda University
[††] Japan Atomic Energy Research Institute
[†††] Yamatake Corporation

として, 同一事例が存在しない場合は既存の事例に新事例を追加することにより行われる. ただし, ここで $Y_{new}$: 更新後の事例出力値,$Y_{old}$: 更新前の事例出力値,k: 同一メッシュに存在するデータ数,Y: 新しく入力された事例出力値である.

## 4 プロセッサ負荷予測の性能評価

本章では,dataFOREST を用いて前章で述べた予測手法を用いたプロセッサの負荷の予測を行った時の性能について述べる. ただしプロセッサ負荷情報は, 他に負荷がない状態での処理時間を 1 として, 各時点で何倍の処理時間を要するかという値を示している.

本論文では, ある時刻より後 300 分の予測を行うために, 資源情報サーバにより管理されているその時刻より前の負荷情報のうち,3 台の並列計算機の 1 分間隔で 96 時間 48 分にわたる各プロセッサ (SR2201 のプロセッサ 1 から 16,SR2201 デスクサイドモデルのプロセッサ 1 から 8,SX4 のプロセッサ 1 から 3) の負荷情報を用い,dataFOREST により事例出力値 SR2201 のプロセッサ 1 に対して寄与の高い 15 種の入力変数を予測時刻より 1 分前から 10 分前の負荷情報から選択することでモデルを作成し,SR2201 のプロセッサ 1 の負荷をオンライン学習による予測モデルの実行時修正を行わない手法と学習を行う手法それぞれで予測する. ただし,3.2 節で述べたように, モデル作成に使用する計測した各プロセッサの CPU 負荷データにノイズが多く含まれている場合は予測誤差が大きくなる可能性があるため, 資源情報サーバが計測した負荷をそのまま入力データとしたもの (リミットカットなし) と, ノイズ除去のためプロセッサ負荷が 3.5 以上のモデルデータを 3.5 としたもの (リミットカットあり) の両方の場合について評価を行った.

プロセッサの負荷予測値とプロセッサ負荷の実測値のリミットカットなしの場合の学習なし (図 1(a))/学習あり (図 1(b)) をそれぞれ示す. 図 1 を見るとわかるように実測値と予想値の波形が酷似しており, 予測が適切に行われていることがわかる. ただし, ノイズに対する予測がされるため予測値に高周波成分が多く含まれ, 学習有無に関わらず相対誤差 (= (絶対平均誤差/実測値の平均値) × 100[%]) が 9.50[%] となり学習による相対誤差の改善が見られない. また, プロセッサの負荷予測値とプロセッサ負荷の実測値のリミットカットありの場合の学習なし (図 2(a))/学習あり (図 2(b)) をそれぞれ示す. 図 2 では, ノイズに対する予測が行われないため高周波成分が比較的少なく, 学習を行わない場合の相対誤差が 8.16[%] であるのに比べ, 学習を行ったことにより相対誤差が 8.14[%] となる. よって, リミットカットを行った場合の方がリミットカットなしの場合より予測値の相対誤差が小さいことから, リミットカットにより入力事例のノイズ部分を除去することで, 予測の精度を向上できることがわかる.

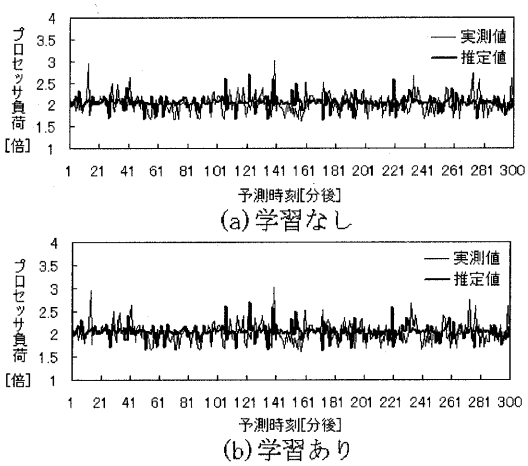


図 1: 計測した CPU 負荷を用いた予測の精度

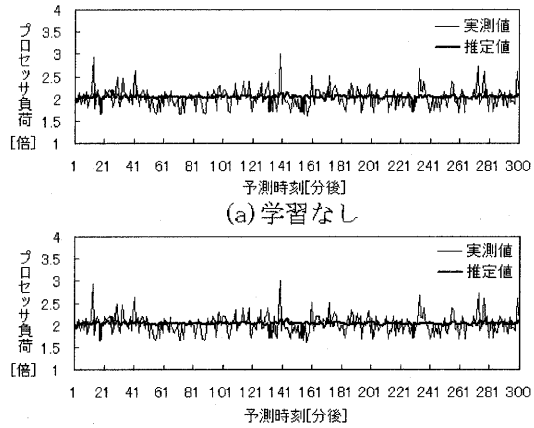


図 2: リミットカットした計測 CPU 負荷を用いた予測の精度

## 5 おわりに

本稿では, データマイニングツール dataFOREST を用いて異機種分散環境におけるプロセッサ負荷予測を行い, その性能評価を行った. その結果, 元事例そのものから予測した場合で真値からの相対誤差 9.5%, リミットカットによりノイズを除去した場合で相対誤差 8.14%の予測値が得られ, データマイニングを用いた負荷予測手法の有意性が確かめられた. 今後の課題としては,dataFOREST を用いてプロセッサ負荷だけではなくネットワーク負荷の予測を行うこと, またそれらの予測情報を用いてメタスケジューリングを行うことが挙げられる.

本研究の一部は NEDO アドバンスト並列コンパイラプロジェクトにより行われた.

## 参考文献

[1] 村杉明夫, 林拓也, 飛田高雄, 小出洋, 笠原博徳, “OSCAR マルチグレイン並列化コンパイラを用いたスーパーコンピュータクラスタのためのメタ・スケジューリング手法”, 情報処理学会第 58 回全国大会, 2D-06, 1999

[2] Andrew Grimshaw, Adam Ferrari, Fritz Knabe, Marty Humphrey, “Legion:An Operating System for Wide-Area Computing”, IEEE Computer 32:5, pp.29-37, 1999

[3] 中田秀基, 竹房あつ子, 松岡聡, 佐藤三久, 関口智嗣, “グローバルコンピューティングのためのスケジューリングフレームワーク”, 並列処理シンポジウム JSPP'99 論文集, pp.277-284, 1999

[4] Peter A.Dinda, David R.O'Hallaron, “An Evaluation of Linear Model for Host Load Prediction”, Proc. of the 8th IEEE Symposium on High-Performance Distributed Computing, 1999

[5] Rich Wolski, Neil Spring, Jim Hayas, “The Network Weather Service:A Distributed Resource Performance Forecasting Service for Metacomputing”, Journal of Future Generation Computing Systems, pp.757-768, 1998

[6] 筒井宏明, 黒崎淳, 佐藤友彦, “履歴データを事例として使用する非線型モデリング技術 TCBM:Topological Case Based Modeling”, 計測自動制御学会論文集 Vol.33, No.9, pp947-954, 1997

[7] 小出洋, 山岸信寛, 武宮博, 林拓也, 引田雅之, 笠原博徳, “メタスケジューリングのための資源情報”, 計算工学講演会論文集, Vol.5, No.2, pp.357-360, 2000

[8] 笠原博徳, “並列処理技術”, コロナ社, 1991

[9] T.Imamura, Y.Tsujita, H.Koide and H.Takemiya, “An Architecture of Stampi: MPI Library on a Cluster of Parallel Computers”, Proc. of 7th EuroPVM/MPI Conference, LNCS 1908, pp.200-207, 2000