

量子化学計算における計算資源予測機能の改良

西川 武志[†] 伊藤 智[†]
長嶋 雲兵[†] 関口 智嗣[†]

これまでにグリッド技術を用いた量子化学アプリケーション Gaussian の利用環境として Quantum Chemistry Grid/Gaussian Portal (QC Grid/Gaussian Portal) の設計と構築を行ってきた。QC Grid ではメタスケジューラがジョブの要求する資源量を基底関数数の指数関数として推定した。今回は、分子中の原子数を説明変数に加えた新たな計算時間予測式とロバスト回帰分析によるパラメータ決定を導入することで、原子数に依存した実行計算時間不連続や繰返し計算の収束異常に由来する外れ値に対応できていなかった既存の実装の難点を克服したことを報告する。これにより新たなメタスケジューラは結果データベースから性能情報を抽出して自律的に予測機能へ反映することが可能となった。

Reformulation of Estimating Resource Requirements for Quantum Chemistry Calculation

TAKESHI NISHIKAWA,[†] SATOSHI ITOH,[†] UMPEI NAGASHIMA[†]
and SATOSHI SEKIGUCHI[†]

We have designed and developed the Quantum Chemistry Grid (QC Grid)/Gaussian Portal as a useful environment of the quantum chemistry application package by deploying the grid technology. At the QC Grid, the Meta-scheduler estimates the resource requirements of each job in a formula of an exponential function of the number of basis functions. The Meta-scheduler is improved by adding atomicity to the explanatory variable and reformulating estimation of calculation time. It enables to make accurate estimation though the discontinuity of the execution time exist which depends on the number of atoms in a molecule. It solves problems in former formulation by introducing the robust regression analysis at parameterizing of the estimating formula. The Meta-scheduler becomes possible by reformulating the calculation time estimation the autonomous reflection in the resource requirements prediction by extracting performance information from the results database.

1. はじめに

これまでにグリッド技術を用いた量子化学アプリケーション Gaussian¹⁾ の利用環境として Quantum Chemistry Grid/Gaussian Portal (QC Grid/Gaussian Portal) の設計と構築を行ってきた^{2),3)}。産業技術総合研究所先端情報計算センターでは分子軌道ポータルとして実際に運用し UNIX 版 Gaussian のキャラクタコマンドユーザインタフェースを隠蔽し Web インタフェースから Gaussian ジョブを実行可能であり、利用者を得てきた。

QC Grid/Gaussian Portal の構成要素は Web インタフェースを構築する Portal サーバ、専門家の知識

を蓄えた知識データベース、知識データベースを利用して利用者からの計算依頼内容を解析し必要となる計算機資源量 (CPU, メモリ, ディスク) を理論値と実測値から推定し、適切な計算資源配分を行うメタスケジューラ、計算結果を蓄積する結果データベース、スケジューラを備えた複数の計算ノードから構成されている (図 1)。

メタスケジューラは、利用者の理論性能が高ければ高速だろう、メモリをたくさん割り当てれば高速だろうという実際のアプリケーション性能とはかけ離れた思い込みによる計算資源の浪費を抑制する。これにより利用者は個々の計算実行環境の詳細を知らなくても計算資源を効率的に利用することが可能になった。しかしながらアプリケーションのバージョンアップにともなう資源要求量の変化、蓄積した結果データベースから性能情報を抽出して自律的にメタスケジューラの

[†] 産業技術総合研究所グリッド研究センター
Grid Technology Research Center, National Institute of
Advanced Industrial Science and Technology (AIST)

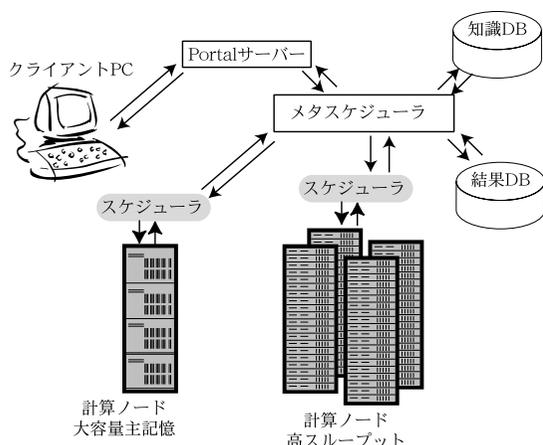


図1 QC Grid/Gaussian Portal の構成概要

Fig.1 Components of QC Grid/Gaussian Portal.

予測機能へ反映することが、課題として残されている。特に後者は、量子化学計算は基本的に非線形方程式の繰返し計算のため収束までの繰返し回数が一定でないこと、分子構造最適化ではさらに構造最適化の繰返し計算が加わり二重に繰返し計算の反復回数が不定になるため、時折、収束までに同じ規模の系に比べて1桁以上実行時間が増大する場合があります、このような外れ値の取扱いをどうするかが課題となる。特に遷移金属を含む分子ではこの問題が顕著である。今回は、原子数を説明変数に加えた新たな計算時間予測式とロバスト回帰分析によるパラメータ決定を導入することで既存の実装の難点、Gaussian 98 から 03 になり新しいアルゴリズムが導入され予測時間誤差が大きくなってしまったことを克服したことを報告する。これにより蓄積した結果データベースから性能情報を抽出して自律的にメタスケジューラの予測機能へ反映することについても対応可能となった。

2. 量子化学計算手法における計算時間予測の問題

量子化学計算プログラムは多体のシュレーディンガー方程式を非線形の一体問題に近似し自己無撞着場 (Self Consistent Field: SCF) 法 (別名 Hartree-Fock-Roothan 法, HF 法) により解くことを基本としている。詳細は専門書に譲る⁴⁾が、SCF 計算は

$$x = f(x) \quad (1)$$

という非線形方程式の解を、ある初期値から始め、その名のとおり、Self Consistent になるまで繰返し計算を行うことで求める。代表的な収束条件として全エネルギー変化が繰返し前後で一定値以下になるというものが、したがって収束回数はあらかじめ定まらない。

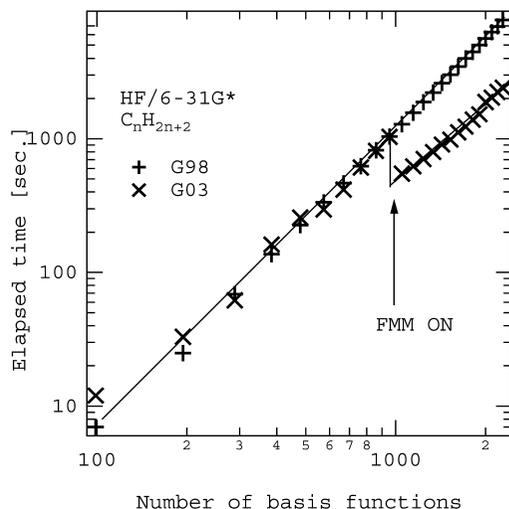


図2 脂肪族飽和炭化水素における HF/6-31G*計算の経過時間に及ぼす Fast Multipole Method の影響

Fig.2 Introduction of Fast Multipole Method effect on the time for HF/6-31G* calculation of Alkane.

い。金属を含む系では、金属を含まない系と比較して収束回数が増える傾向がある。また電子を奇数含む開殻系と呼ばれる場合も収束回数が増える傾向がある。分子構造最適化は SCF 計算によって求めた電子状態から分子の原子に働く力を求め、力の働く方向に原子を動かし、ある定められた収束条件に達するまでの繰返し計算を行う。3 原子以上からなる分子ではいわゆる N 体問題のため、解析解は存在せず、分子構造最適化の収束までの回数 (ステップ数) はあらかじめ定まらない。したがって構造最適化では電子状態の SCF 計算、N 体問題の求解と繰返し計算の二重構造となっている。したがって収束するまでの全計算時間予測は構造最適化の場合、非常に困難である。可能な限り計算資源の利用可能時間上限と予測される全経過時間を突き合わせるという効率の高いジョブスケジューリングのためには実行計算資源の実行時間制限と予測した全経過時間を比較する必要がある。しかしながらジョブの種別が構造最適化の場合は、利用可能時間上限が非常に長い、もしくは上限がないという計算資源を選択せざるをえなかった。

文献 3) の量子化学計算資源予測機能の説明変数としては、基底関数数、手法、ジョブ種別に絞った。その理由は、予測に必要な知識や情報を保存するデータベース (知識データベース) の容量の節約のためであり、基底関数は原子数と比例関係にあり原子数の影響は基底関数数で十分取り込めると判断したためである。しかしながら前回対象とした Gaussian 98 からバージョンアップした Gaussian 03 ではクーロン相互作用

用の高速化手法, Fast Multipole Method⁵⁾ が導入され, 原子数 60 以上で自動的に適用されるようになった. このため Gaussian 03 では計算時間の分子を構成する原子数に依存した振舞い, 計算時間が FMM 適用となる原子数の前後で不連続となり (図 2), 基底関数は原子数と比例関係にあることをそのまま利用できなくなった.

3. 新たな計算時間予測式とロバスト回帰分析によるパラメータ決定

3.1 新たな計算時間予測式

新たな計算時間予測式では基底関数数に加えて原子数を説明変数として導入することから, 単純に原子数 (および電子数) に依存する項を採用するだけにとどまらず, より予測精度が高まるよう定式化の見直しを行った. 前回, 計算時間 t_{est} の予測式は, 定数項 d , 係数項 C , 基底関数数 N の α 乗からなる $t_{est} = d + CN^\alpha$ に基づいて, 一般的な量子化学アルゴリズムの理論に応じた予測式 (T: Theoretical)

$$t_{estT} = d_T + C_T N^{\alpha_T} \quad (2)$$

の係数を用意し, その係数を実測に応じて変更した予測式 (S: survey)

$$t_{estS} = d_S + C_S N^{\alpha_S} \quad (3)$$

を, 系の大きさ, 理論的導出の困難さ, 実測の有無等に応じて, 人手で任意に t_{estT} あるいは t_{estS} の各要素を組み合わせて構築した. 今回は単純に, パラメータ決定の際に人手による試行錯誤を除外するため,

$$t_{est} = \max(t_{estT}, t_{estS}) \quad (4)$$

により計算時間予測式を構築した.

ジョブ種別係数から閉殻 (Restricted) か開殻 (Unrestricted) かの係数 C_{RU} を独立させ (閉殻: 1, 開殻: 2), 係数定数項 C_{j0} と原子数 M の関数の積, $C'_j = C_{RU} C_{j0} C_j(M)$ として表すようにした. C_{j0} の決定に際しては, 全体の計算時間を最適化の回数で除したデータに基づくようにした. 最適化の全繰返し数は原子数 M の関数として表現可能なので $C_j(M)$ を最適化のアルゴリズムに応じて関数形を定めた. 予測式の骨格をなすエネルギー 1 点計算の予測式のパラメータ決定では SCF 計算の繰返し回数で計算時間を除したデータに基づくようにした. このとき新たに SCF アルゴリズムの種類ごとの係数 $C_{SCFkind}$ と SCF 繰返し係数 C_{SCFcyt} の積, C'_{SCF} を導入した. C_{SCFcyt} は SCF アルゴリズムの種類だけでなく, 分子を構成する原子の種類 (特に金属元素の有無) に依存したものを用意した. このように構造最適化の全ステップ数の原子数依存性や SCF 計算繰返し計算の影

響を柔軟に表現できる定式化を行った新しい計算時間予測式は次式のように表される.

$$t_{est} = d + C_s C'_{SCF} C_m C'_j N^{m+j} \quad (5)$$

t_{est} : 予測時間

d : 遅延時間

C_s : システム性能係数

C'_{SCF} : SCF アルゴリズム種別係数

$C'_{SCF} = C_{SCFkind} C_{SCFcyt}$

$C_{SCFkind}$: SCF アルゴリズム種別係数

C_{SCFcyt} : SCF 繰返し係数

C_m : 手法種別係数

C'_j : 新ジョブ種別係数

$C'_j = C_{RU} C_{j0} C_j(M)$

C_{RU} : Restricted, Unrestricted 係数

C_{j0} : ジョブ種別定数

$C_j(M)$: ジョブ種別原子数依存係数

N : 基底関数数

m : 手法種別累乗数

$+j$: ジョブ種別累乗増分

3.2 ロバスト回帰分析によるパラメータ決定

実測値に基づいて計算時間予測式に必要なパラメータを決定する際に以前は最小自乗法を用いていた. 最小自乗法で用いられる最自乗 (LMS) 基準はすべての誤差を均等な重みで扱っているため, 1 つの大きな例外値によって大きな影響を受けてしまう. このモデルから外れた値によって計算時間予測式のパラメータが大きな影響を受けないようにするため, 最小自乗法を適用する前に, あらかじめ外れ値を取り除く操作を人手で行っていた. QC Grid では蓄積した結果データベースから性能情報を抽出して知識データベースに収め, 自律的にメタスケジューラの予測機能へ反映することを目標に置いているが, パラメータ決定の際に人手がかかることが課題であった. 外れ値に影響されにくい, 頑強なパラメータ推定法としてロバスト回帰分析法⁷⁾を採用することでこの問題を解決した.

M-estimator は, 最もよく利用されるロバスト回帰分析法の 1 つであり, 最小自乗法で用いられる LMS 基準

$$LMS = \min \sum_i \epsilon_i^2 \quad (6)$$

の代わりに, それを少し変形した評価基準を用いる. ここで, ϵ_i はデータと推定値との誤差である. ここで, M-estimator では, これを例外値には小さな重み

を与えるように変形し,

$$M = \min \sum_i \rho(\epsilon_i) \quad (7)$$

を評価基準とする．ここで、 $\rho(x)$ は、 $x = 0$ で唯一の最小値を持つ正定値偶関数である．この基準は、もし $\rho(x) = x^2$ ならば、平均自乗誤差基準と同じになり、その意味で最小自乗法の拡張になっている．

関数 ρ によってモデルからずれたデータに対してどれくらいの重みが与えられるかが定まり、 $\Psi(x) = \frac{\partial \rho(x)}{\partial x}$ は、評価関数 (influence function) と呼ばれ、データがモデルからある程度離れるとその影響はほとんどなくなり、0 に近づくものが選ばれる．

M-estimator による推定のためのアルゴリズムは、式 (7) を最小化する最適化問題として定式化することができ、重み付き最小自乗問題となる．ただし、このアルゴリズムは、関数によっては必ずしも最適解に収束することは保証されないので、良い初期値から出発する必要がある．今回は出発初期値を既存の計算時間予測式の係数値を用いることで解消できる．

3.3 自律的パラメータ決定の実装

自律的パラメータ決定の実装では、結果を収めた結果データベースからあらかじめ資源予測式のパラメータ決定に必要な指標を抜き出して収めた知識データベースを用意し、資源予測に際し、どのような元素を含む、基底関数数がいくつ以上等の条件を設定して知識データベースから抽出したデータからパラメータをその場で決定して資源予測を行うことを目指した．実装にあたっては、現在、Web インタフェースからファイルアップロード後ジョブ投入完了を利用者まで通知する時間、計算予測式の定数項 d の遅延時間を 10 秒としているため、これを大きく増大させることのないよう 5 秒を切ることを目標とした．まず入力ファイル解析機能により Gaussian の入力ファイルから資源予測に必要な情報を抜き出し計算条件を設定する．知識データベースには SQL データベースシステムを利用し、入力ファイルで指定された計算条件に合致する SQL 文を発行して抽出する．

ロバスト回帰には統計計算言語 R⁶⁾ の MASS パッケージ⁷⁾ に含まれる rlm を用いた．rlm は線形回帰のみをサポートしているため、データベースから抽出した経過時間、原子数、基底関数数は対数を取り一時ファイルに保存し、R に読み込ませ、rlm の M-estimator によるロバスト線形回帰を実行し、係数を求め、計算時間を予測した．M-estimator の評価関数としては、既存の計算結果からの寄与が中央値から外れるに従って滑らかに小さくなるように図 3 の Tuckey bisquare

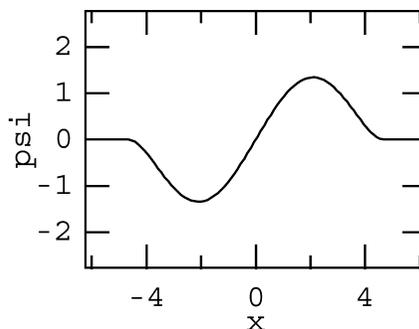


図 3 M-estimator の評価関数 Tuckey bisquare
Fig. 3 The influence functions for M-estimator, Tuckey bisquare.

関数を採用した．

4. 結果評価

4.1 自律的パラメータ決定のシミュレーション

自律的パラメータ決定の実装が、実用的な時間で処理可能かどうかをシミュレーションを行って検証した．テスト環境には、ハードウェアとして 1.5 GHz PowerPC G4 を搭載し 1 GB 主記憶、80 GB ハードディスクドライブを搭載した PowerBook G4、ソフトウェアは Mac OS X 10.3 上の Perl5.8.1, MySQL 4.0.2, R 1.91 を用いた．60,390 件の結果を収めた結果データベースから抽出して知識データベースを作成した．知識データベースのサイズは約 3 MB となった．予測シミュレーション対象の入力ファイルには 60,390 件の結果に含まれない、平均 11,168 バイトのファイルサイズ、最小 293 バイト最大 4.8 MB の 50 種用意した．ランダムに 50 種のファイルを入力ファイルとして取り出し合計 500 回、自律的にパラメータ決定し計算時間を予測するのに必要な時間を測定した．

まず入力ファイルの解析時間は平均 0.16 秒であり、図 4 の大きな外れ値は 4.8 MB 弱のファイルであり、それに対する入力ファイル解析時間は 17 秒弱であり 1 MB に満たない入力ファイルでは 1 秒を超えることはなかった．所望計算条件のデータセットを SQL 文を発行して抽出し、経過時間、原子数、基底関数数の対数を取りファイルを一時ファイルに保存するのに要した平均時間は 0.49 秒、最長 0.88 秒であった．R をバッチモードで呼び出し MASS ライブラリを読み込み、一時ファイルからデータを読み込み、M-estimator を用いたロバスト回帰を実行してパラメータセットを決定するのに平均 2.04 秒、最長 2.89 秒を要した．全体の平均時間は 2.69 秒、最長時間は 19.0 秒 (最大入力ファイルのとき) であった．

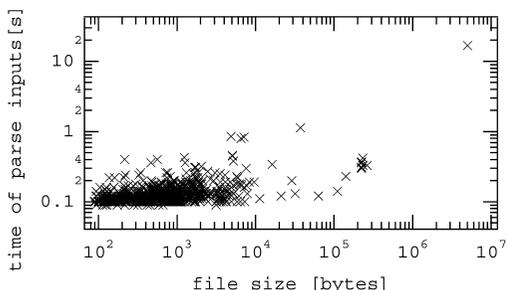


図4 自律的パラメータ決定のシミュレーション実行時間
Fig. 4 The simulation execution time of autonomous parameter definition.

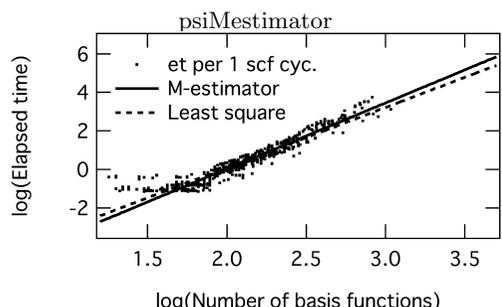


図5 新旧比較：SCF 1 サイクルあたりの経過時間の基底関数依存性
Fig. 5 The elapsed time per 1 SCF cycle dependence on number of basis functions.

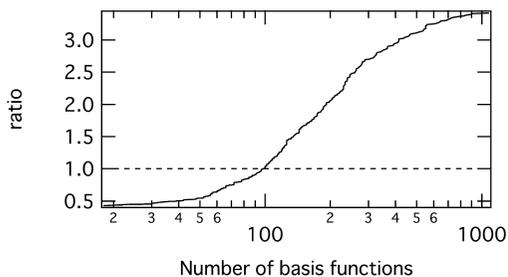


図6 新旧比較：最小自乗法による予測時間と M-estimator による予測時間の比の基底関数依存性
Fig. 6 Drawing a comparison in the estimation time dependence on number of basis functions between LMS and M-estimator.

4.2 計算時間予測式の比較

既存の 60,390 件の結果から抽出した知識データベースの全データを用いてパラメータを決定した SCF1 サイクルあたりの経過時間の基底関数依存性を最小自乗法と M-estimator でパラメータ化した様子を図 5 に示す。式 (5) の m : 手法種別累乗数は最小自乗法では 3.11, M-estimator では 2.70 が得られた。図 6 では新旧計算時間予測法の予測値の比を示す。代表として

表 1 SCF エネルギー 1 点計算結果から M-estimator を用いて決定したパラメータ

Table 1 The coefficients which were determined by M-estimator from the results of SCF single point energy calculation.

係数	値
d	10
$C_{SCFkind}$	1.00
C_{SCFcyc}	13.0
C_m	2.70
C_{j0}	1.00
$C_j(M)$	1.00
m	1.78e-06
$+j$	0

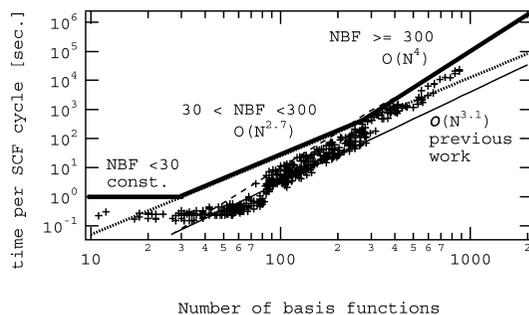


図7 SCF 計算 1 回あたりの経過時間に基づいた予測式
Fig. 7 Time estimation equation which were determined by the elapsed time of single SCF calculation.

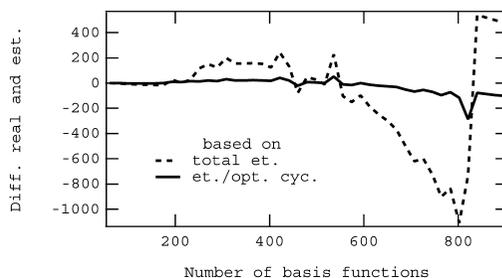


図8 全経過時間に基づいた予測式および構造最適化サイクルあたりの時間に基づいた予測式、それぞれの予測値と実経過時間との差
Fig. 8 The difference between the estimation which were determined by total elapsed time and the estimation which were determined by single SCF calculation time.

Gaussian 03 における既定の SCF (HF) エネルギー 1 点計算の場合に M-estimator を用いて実測値から推定したパラメータを表 1 に示し、それと式 (4) により理論値と組み合わせた予測式と実測値をプロットしたグラフを図 7 に示す。

図 8 にシミュレーションで利用した入力ファイル群の構造最適化計算を行った際の、従来の全経過時間に基づいた予測式と新たな計算時間予測式、構造最適化

サイクルあたりの時間に基づいた予測式によるそれぞれの予測計算時間と実際の経過時間との差の基底関数数の関数として示す。

5. 考 察

4.1 節の結果から自律的パラメータ決定のシミュレーション、従来の静的計算時間予測式を用いた場合と今回の自律的パラメータ決定実装との資源予測機能自体の実行時間の差は、ロバスト回帰を実行する時間とほぼ等しくなり、その時間は平均で 2 秒程度である。最大入力ファイルの場合の資源予測機能実行時間差は平均 1.97 秒となり入力ファイルの解析に時間の多数を占めロバスト回帰実行時間は 1 割程度である。したがって計算機性能としてノートブックパソコンを用い、それほど高速ではない CPU やディスクを用いた環境であっても、利用者からのジョブ投入時に計算時間予測係数を定めても問題とはなりえない時間内で処理が完了する可能性が示せた。

次に 4.2 節、図 6 から従来の最小自乗法によってパラメータを定めた計算時間予測式ではロバスト回帰によるパラメータ決定と比較して、基底関数数の小さな系の影響から、基底関数の大きな系での実行時間を過小評価してしまっている傾向が読み取れる。200 基底関数から従来法ではロバスト回帰に対して半分、1000 基底関数では 3 分の 1 の実行時間と見積もっている。Itanium 2 1.3 GHz 3 MB キャッシュ 8 GB メモリ機 1 CPU において Gaussian 03 の 1000 基底関数での HF における SCF1 サイクルの実時間はおおむね 1,000 秒台であり、60,390 件の結果において SCF の繰返し数の平均が 13 回である。すなわち従来の予測ではエネルギー 1 点計算でも 1000 基底関数では 3 時間半かかる計算を 1 時間強と見積もってしまう、短い時間制限の計算資源を選択し、計算が中断されてしまう可能性が新予測法より大きい。新予測式では見積りが甘くジョブ中断となってしまうことが従来法より大幅に削減できることが期待できる。さらに Gaussian では SCF の繰返しの回数や構造最適化のステップ数の上限は入力ファイルで指定しない限り既定値が適用され、入力ファイルで指定する場合は通常規定値を上回る値が指定される。入力ファイルからこれらの情報を得て既定値もしくは利用者設定値に SCF 繰返し数や構造最適化のステップ数を新たな計算時間予測式に代入することで、平均的な実行時間予測以外に、実行時間上限値の予測が可能になる。平均的な実行時間と実行時間上限値を利用してメタスケジューラが計算資源の選択を行えば過大評価も抑制可能となり、資源利

用効率が向上すると期待できる。

構造最適化計算の実行時間予測においても新たな計算時間予測式は図 8 に示すように 600 基底関数を超える領域で、従来の全経過時間のみに基づく、原子数をあらわに評価していない予測式よりも良好な結果を示している。今回予測式の評価のためのシミュレーションに用いた入力ファイル群において 1 原子あたりの基底関数数は 9.33 であり、600 基底関数を超える領域は原子数にして 64、すなわち FMM が有効になる領域に重なる。

以上のように新たな計算時間予測式は、原子数を説明変数として取り入れるとともに繰返し計算の影響を考慮できるよう改良したため、前回の課題であった原子数によって不連続に計算時間の挙動が変化することに対応可能である。また M-estimator を用いたロバスト回帰を用いることで、自律的に予測に必要なパラメータを決定することが可能になり、利用者からのジョブ投入時、その場で最適なパラメータを構成可能で、実用的である可能性が示せた。

6. ま と め

原子数を説明変数に加えた新たな計算時間予測式とロバスト回帰分析によるパラメータ決定を導入することで、原子数に依存した実行計算時間不連続や繰返し計算に由来する収束異常の場合の外れ値に対応できていなかった既存の実装の難点を克服した。原子数を説明変数に加えることとロバスト回帰分析によるパラメータ決定を導入することで、副次的にメタスケジューラが結果データベースから性能情報を抽出して自律的に予測機能へ反映することが可能となった。

今後の課題としてグリッド環境上の分散した複数の拠点で保持されているデータベースを利用して自律的にメタスケジューラが計算資源の最適配分を試みることがある。今回は結果データベースから知識データベースへのデータの抽出は事前に済ませており、またメタスケジューラと知識データベースは同じ計算機上で稼働している。グリッド環境上ではどうしても他のサイトからの情報伝達には時間がかかってしまう。実用上問題がない応答時間で利用者にサービスを提供可能にするにはどうしたらよいか課題として残される。

謝辞 本研究の一部は、独立行政法人新エネルギー・産業技術総合開発機構平成 16 年度産業技術研究助成事業の一環として委託を受け実施している「量子化学グリッド ASP 実証実験」の成果である。

参 考 文 献

- 1) Frisch, M.J., et al.: Gaussian 03, Revision C02, Gaussian, Inc., Pittsburgh PA (2003).
- 2) 西川武志, 長嶋雲兵, 関口智嗣: Quantum Chemistry GRID/Gaussian Portal の構築, 情報処理学会研究報告, 2002-HPC-90-8, pp.43-48 (2002).
- 3) 西川武志, 長嶋雲兵, 関口智嗣: Quantum Chemistry GRID/Gaussian Portal Phase 2, 情報処理学会研究報告, 2002-HPC-92-8, pp.43-48 (2002).
- 4) Foresman, J.B. and Frish, A.E., 田崎健三(訳): 電子構造論による化学の探求, 第2版, ガウシアン社(1998).
- 5) Millam, J.M. and Scuseria, G.E.: *J. Chem. Phys.*, Vol.106, p.5569 (1997).
- 6) Ihaka, R. and Gentleman, R.R.: A language for data analysis and graphics, *Journal of Computational and Graphical Statistics*, Vol.5, No.3, pp.299-314, (1996).
<http://www.r-project.org/>
- 7) Venables, W.N. and Ripley, B.D.: *Modern Applied Statistics with S, 4th Edition*, Springer (2002).
<http://www.stats.ox.ac.uk/pub/MASS>
- 8) 西川武志, 長嶋雲兵, 関口智嗣: Quantum Chemistry Grid/Benchmarking Portal の構築, ハイパフォーマンスコンピューティングと計算科学シンポジウム, HPCS2003 (2003).

(平成 16 年 10 月 4 日受付)

(平成 17 年 1 月 14 日採録)



西川 武志 (正会員)

1992 年慶應義塾大学理工学部計測工学科卒業。1998 年同大学大学院理工学研究科計測工学専攻博士課程修了, 博士(工学)。同年日本学術振興会未来開拓学術研究推進事業リサーチ・アソシエイト(分子科学研究所理論研究系)。2001 年 4 月より独立行政法人産業技術総合研究所。現在同所センターグリッド応用チーム研究員。化学物理, 計算機性能評価技術, グリッドアプリケーションサービスの研究開発に従事。日本物理学会, 日本化学会, フラーレン・ナノチューブ研究会各会員。



伊藤 智(正会員)

1982 年千葉大学理学部物理学卒業。1987 年筑波大学大学院物理学研究科博士課程修了。同年株式会社日立製作所に入社, 中央研究所に勤務。2002 年 5 月に同所を退職。2002 年 6 月に独立行政法人産業技術総合研究所に入所。2003 年よりグリッド応用チーム長(2004 年 10 月にビジネス応用チームに改名)。以来, グリッドのビジネス応用に関する研究に従事。日本物理学会, つくばサイエンスアカデミー各会員。



長嶋 雲兵(正会員)

1983 年北海道大学大学院理学研究科博士後期課程化学第二専攻修了。理学博士。同年岡崎国立共同研究機構分子科学研究所電子計算機センター助手。1992 年お茶の水女子大学理学部情報科学科助教授。1996 年同教授を経て, 1998 年通産省工業技術院物質工学工業技術研究所基礎部理論化学研究室長。1999 年同産業技術融合領域研究所計算科学研究グループ長, 2001 年 4 月独立行政法人産業技術総合研究所に改組。同所先端情報計算センター情報基盤研究開発室長。2002 年より同所グリッド研究センター総括研究員。筑波大学連携大学院大学教授。計算化学, 情報化学, 大規模数値計算, 広域分散並列処理の研究開発に従事。日本化学会, IEEE, 応用数理学会, 計算工学会, 化学工学会, 日本コンピュータ化学会各会員。



関口 智嗣(正会員)

1982 年東京大学理学部情報科学科卒業, 1984 年筑波大学大学院理工学研究科修了, 同年工業技術院電子技術総合研究所入所。以来, データ駆動型スーパーコンピュータ SIGMA-1 の開発, ネットワーク数値ライブラリ Ninf, クラスターコンピューティング, グリッドコンピューティング等に関する研究に従事。2001 年独立行政法人産業技術総合研究所に改組。2002 年 1 月より同所グリッド研究センター長。市村賞, 情報処理学会論文賞受賞。グリッド協議会会長。SIAM, IEEE, つくばサイエンスアカデミー各会員。