社団法人　電子情報通信学会
THE INSTITUTE OF ELECTRONICS,
INFORMATION AND COMMUNICATION ENGINEERS

信学技報
TECHNICAL REPORT OF IEICE.

大規模計算環境におけるユーザ満足度を考慮した資源管理へむけて

國府　理央† 　佐藤　仁† 　松岡　聡†,††

† 東京工業大学　〒 105–0123 東京都目黒区大岡山 1–21–2
†† 国立情報学研究所　〒 101–8430 東京都千代田区一ツ橋 2–1–2
E-mail: †rio@matsulab.is.titech.ac.jp, ††hitoshi.sato@gsic.titech.ac.jp, †††matsu@is.titech.ac.jp

あらまし　大規模計算環境上でアプリケーションを実行する際，ユーザは複雑な資源パラメタ設定を行う必要がある．しかし，ユーザは必ずしも専門家ではないため，自身で適切な資源選択を行うことは困難である．我々はこの問題を，ユーザの希望に応じた適切なパラメタ設定を提案するエキスパートシステムを構築することで解決できると考えているが，具体的にユーザがどのようにシステムを利用したいのかは明らかではない．そこで，実際のユーザのシステム利用希望パターンをアンケート調査により抽出し，これを利用してユーザの嗜好にあわせた資源選択提示を行うモデルをたてた．またモデルの検証により，実際にユーザの希望にあわせた資源選択提示ができることを確認した．
キーワード　大規模分散環境，最適化スケジューリング，資源管理

Towards Resource Management Considering User's Satisfaction in Large Distributed Computing Environments

Rio KOKUBU†, Hitoshi SATO†, and Satoshi MATSUOKA†,††

† Tokyo Institute of Technology　Oookayama 1–21–2, Meguro-ku, Tokyo, 105–0123 Japan
†† National Institute of Informatics　Hitotsubashi 2–1–2, Chiyoda-ku, 101–8430 Japan
E-mail: †rio@matsulab.is.titech.ac.jp, ††hitoshi.sato@gsic.titech.ac.jp, †††matsu@is.titech.ac.jp

Abstract　Users among the large distributed computing environment needs to configure the resource parameters to optimize their resource selection when they execute their application on the environment. As they are not all exparts, this is sometime hard for them. Although we believe this problem to be solved if there were an expart system that recommends the best resource selection for each users considering their requirement, how do users want to use the environment is not aparrent. To know this we first sampled the actual users' requirement patterns thorough the questionaire survey, and then, using this requirement pattern, we set up a model that suggest resource selection which meets users'demands. We confirmed the adequacy of the model thorough examination that it actually sugguest the resource selection meets users' demand.
Key words　Large distributed computing environments, optimized scheduling, resource management

1. はじめに

近年，大規模計算環境の利用がますます盛んになっている．従来は専門的な分野であった並列分散コンピューティングも，今では流体シミュレーションや DNA の配列計算など，様々な分野の科学技術計算に利用されている．今後とも，ますます利用者が増えていくであろう．

しかし，大学をはじめとする教育機関が所有しているような，TSUBAME [8]，T2K [9] といった大規模計算環境は，専門性が高く，その利用方法も複雑である．また主にバッチスケジューリングシステムが使われており，特に，利用する資源の選択については，使用する CPU コア数，メモリ容量，並列実行数といった多種多様な項目が設けられており，並列分散コンピューティングの専門家にとっては使い勝手がよいものの，大多数の一般ユーザにとってはどのように資源選択を行えばよいのかわからないという問題がある．

この問題について，ユーザがどのようにアプリケーションを実行したいのかという希望をもとに，そのユーザにとって最適となるような資源選択方法を自動で提示するようなエキスパートシステムがあればユーザにとってより大規模計算環境が使いやすくなるはずである．しかし，実際にユーザがどのような希望を持っており，どのように資源選択できれば満足なのかは明

らかでない．これらを明らかにするために，まず，ユーザの要望パターンを知るために実地でアンケート調査を行い，ユーザがどのように大規模計算資源を使用したいのかを調査した．その結果，数種類の利用希望パターンを抽出することができ，これらを利用して資源選択提案のモデル化を行うことができた．このモデルでは，ユーザが自分の希望がどの利用パターンに近いかを入力として，そのユーザがどのようにパラメタを選択すればよいのかを出力する．

また，モデルの妥当性を検証するために，モデルを利用して予測した資源選択パターンと，ユーザが元々アンケート調査で答えた希望する資源選択パターンの比較を行ったところ，2 者の間に有意な差は認められず，十分妥当であると確認できた．

2. ユーザ満足度を考慮した資源管理

ユーザが真に望んでいるような資源選択を行うには，ユーザ自身が適切に資源のパラメタを設定する必要がある．ユーザが何を望んでいるかはそのユーザにしかわからないからである．しかし，大規模計算環境においては指定できるパラメタが CPU の周波数やメモリ容量，ストレージ容量など多岐に渡るだけではなく，アプリケーションの特性やシステムの混雑状況なども考慮すべきであったりなど，エキスパートユーザですらその時々に応じた最適なパラメタ設定をするのが困難である．実際，既存のシステムでは，どのようなパラメタ設定が可能なのかすら知らずに利用しているユーザも多い．

実行するアプリケーションがどのくらいメモリを使うのか，CPU 性能は何 GFLOPS あればよいのか，ストレージはどのくらい使うのか，といった，アプリケーションに合わせたチューニングを行うにはアプリケーションの内容に精通していなければならないが，アプリケーションを自分で書いたのではなく，単にツールとして利用している場合などは詳細なパラメタがわからない．結局，それが理由で最適なパラメタ設定がなされていないのが現状である．以上のように，パラメタ設定において，設定すべきパラメタ数が多すぎる，直感的に指定しづらい，という場合にはユーザが自分の希望通りの資源選択が行えないという問題がある．

このような問題を解決するため，より数が少なく，直感的に指定しやすいパラメタを新たに作成し，ユーザによるパラメタ設定を容易にすることを目的としている．つまり，エキスパートユーザの利用パターンを経験的に蓄積し，あまりアプリケーション内容やシステム利用方法に詳しくないユーザでもそれらを参考にして適切なパラメタ設定を行うことができるようになればユーザにとってより使いやすいシステムとなる．そのためには，資源選択を提案するシステム側が，あらかじめユーザの利用希望パターンを知っている必要がある．具体的には，まず実際のユーザの希望をアンケート調査し，調査結果の分析を行ってどのような希望のパターンが存在するのかを抽出する．次にその希望パターンを元に，ユーザがどの希望パターンに近いかを指定することにより最適なパラメタ設定を出力するようなモデルをたてる．

3. 関 連 研 究

資源選択の最適化については, これまで多くの研究がなされてきた. [1], [2], [4]〜[7], [10]

特にコストの最適化については Economic-based Scheduling として多く取り組まれ，ユーザおよび計算環境提供者の両者の立場からコストの問題を考える．

CSAR [7] では, 計算資源の利用コストが，需要量と供給量の変化のたびに両者の均衡がとれるように変動するようなモデルを採用している．

また消費者が支払いできる利用コストを提示し，提供者がそれを元に選出した消費者に資源を提供するような Auction モデルは，Spawn [6]，Popcorn [2] などで採用されている．

Condor [4], [5] では，遊休資源にジョブをマイグレーションすることにより，システム全体の稼働率が最大になり，かつユーザごとのスループットが最適化されるようなスケジューリングを行う．利用する資源の数の最適化を行うような研究には [10] がある．ユーザが投げたジョブをあらかじめ分析し，より少ない計算機資源でジョブを完了できる．

以上のスケジューリングシステムでは全て一元的な最適化が行われており，各ユーザの希望に対応できるような多元的な最適化を行っていない．

また，Nimrod [1] ではユーザが予算とジョブの実行完了時間の deadline を指定することができる．予算，deadline の 2 つのみではあるが，複数パラメタをユーザの希望に合わせて最適化しているが，実際の調査に基づかない資源選択手法であり，ユーザが真に希望している資源選択を行えるわけではない．

4. ユーザ意識調査

4.1 ユーザの希望調査

実際のユーザの希望にはどのようなパターンがあるのかを調べるため，東京工業大学所有のスーパーコンピュータ TSUBAME [8] のユーザを対象に，利用方法に関するアンケートを行った．アンケートは実行するアプリケーションの実行方法に関する全 20 問であり，25 人からの回答を得た．TSUBAME の利用の上位 7 割を占めるヘビーユーザの方や，TSUBAME を研究等で軽く利用している学生の方など，様々なタイプのユーザを対象とした．アンケート内容は図のようになっており，5 段階の指標で答える形式になっている．ただし (17)(18)(19) は「1：利用しない,3：場合に応じて,5：利用する」の 3 段階，(20) は「1：している,5：していない」の 2 段階のみとなっている．

4.2 調査結果の分析

調査結果の分析の目的は 2 つある．1 つ目は観測可能な質問項目とその答えから，これまで観測不可能であった潜在的な利用方法の希望パターンを抽出することである．もう一つは，アンケートでしか収集し得ない，多くて細かい質問への回答をもとに，より少なくシンプルな質問項目を新たに設定し，それをインプット情報として複雑な資源選択パラメタを提案としてアウトプットするような数理モデルを構築することである．この数理モデルについては 4 章で述べる．この目的を満たす分析手

(1) 並列分散処理を始めて何年たつか. (年)
(2) アプリケーションが実行完了するまでの望ましい時間(hour).
ただし, キューでの待ち時間も含む.
(3) アプリケーションの実行において許容できる計算機利用金額(万円).
(4) アプリケーションが必要とするGPUのFLOPS値(GFLOPS).
(5) アプリケーションが必要とするメモリ容量(GB).
(6) アプリケーションの実行に許容できるメモリバンド幅(GB/sec).
(7) アプリケーションが必要とするGPUのFLOPS値(GFLOPS).
(8) アプリケーションが必要とするGPUのVRAM容量(MB).
(9) アプリケーションの実行に許容できるGPUの
メモリバンド幅(MB/sec).
(10)アプリケーションが必要とするストレージ容量(GB).
(11)アプリケーションの実行に許容できるストレージの
read I/Oバンド幅(GB/sec).
(12)アプリケーションの実行に許容できるストレージの
write I/Oバンド幅(GB/sec).
(13)アプリケーションの実行に許容できるネットワークバンド幅(Gbps).
(14)アプリケーションの実行に許容できるネットワーク遅延(msec).
(15)アプリケーションを実行するノード同士はどのような
ネットワークトポロジで接続されていることが望ましいか.
(16)アプリケーションを実行するのに望ましい並列数.
(17)ベストエフォートサービスを利用するか.
(18)性能保障サービスを利用するか.
(19)事前予約サービスを利用するか.
(20)アプリケーションに対して実行が途中で異常終了することを
考慮しているか.
(アプリケーションレベルチェックポイントを実装するなど)

図 1 質問項目

法として，因子分析を用いた．

4.2.1 因子分析

因子分析は，m 個の各個体における複数の観測変数 $\{x_{i1}, x_{i2}, \ldots, x_{in}\}$ に共通して影響をおよぼす因子 $\{f_{i1}, f_{i2}, \ldots, f_{ik}\}$ を抽出するための分析手法である $(i = 1, 2, \ldots, m)$．各変数が，共通因子によって表される部分，独立因子で表される部分からなる，という考え方に基づいている，変数の数を n 個，因子数を k 個と置いたとき，

$$\begin{aligned} x_{i1} &= a_{11}f_{i1} + a_{12}f_{i2} + \cdots + a_{1k}f_{ik} + \epsilon_{i1} \\ x_{i2} &= a_{21}f_{i1} + a_{22}f_{i2} + \cdots + a_{2k}f_{ik} + \epsilon_{i2} \\ &\cdots \\ x_{in} &= a_{n1}f_{i1} + a_{n2}f_{i2} + \cdots + a_{nk}f_{ik} + \epsilon_{in} \\ &(i = 1, 2, \ldots, m) \end{aligned} \tag{1}$$

を満たすような因子負荷行列 A および m 個の個体ごとの因子得点をまとめた因子得点行列 F，独自因子をまとめた独自因子行列 E を求めるのが目的である．ここで，各列に個体ごとのデータをまとめた $n \times m$ 行列を X とおくと，

$$X = \begin{pmatrix} x_{11} & x_{21} & \ldots & x_{m1} \\ x_{12} & x_{22} & \ldots & x_{m2} \\ \ldots & \ldots & \ldots & \ldots \\ x_{1n} & x_{2n} & \ldots & x_{mn} \end{pmatrix}$$

と表され，因子負荷行列 $A(n \times k)$ は

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1k} \\ a_{21} & a_{22} & \ldots & a_{2k} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \ldots & a_{nk} \end{pmatrix}$$

因子得点行列 $F(k \times m)$ は

$$F = \begin{pmatrix} f_{11} & f_{21} & \ldots & f_{m1} \\ f_{12} & f_{22} & \ldots & f_{m2} \\ \ldots & \ldots & \ldots & \ldots \\ f_{1k} & f_{2k} & \ldots & f_{mk} \end{pmatrix}$$

独自因子行列 $E(n \times m)$ は

$$E = \begin{pmatrix} e_{11} & e_{21} & \ldots & e_{m1} \\ e_{12} & e_{22} & \ldots & e_{m2} \\ \ldots & \ldots & \ldots & \ldots \\ e_{1n} & e_{2n} & \ldots & e_{mn} \end{pmatrix}$$

とそれぞれ表される．これらを用いて，全データを共通因子と独自因子で表現する因子モデルは

$$X = AF + E \tag{2}$$

で定義される．

因子分析の流れは次のようになっている．

まず，A の初期解を求める．初期解の求め方には主因子法，最尤法，最小二乗法などがある．主因子法では第一因子から順次因子寄与 (因子の説明力) が大きくなるように，最尤法では共通因子と独自因子が正規分布に従うと仮定してその仮定に基づく尤度関数が最大となるように，最小二乗法ではデータから計算された標本共分散行列とモデルから計算された共分散行列の差の二乗が最小となるように，それぞれ解を求める．

次に，因子数を決定する．因子数の決定基準には，カイザー・ガットマン基準，スクリープロット基準，適合度基準などがある．カイザー・ガットマン基準では，各因子がどの程度各変数を説明できているかという固有値が 1.0 以上のものを採用し，スクリープロット基準では縦軸に固有値，横軸に因子数をとったプロット図で，固有値が急激に下がる一歩手前の因子数を採用する．適合度基準では,「その因子数でのモデルが元データと適合する」という帰無仮説のもとにカイ二乗検定を行い，有意水準 5% としたときの p 値の妥当性から判断する．

また，初期解を求めた後は，因子の解釈をしやすくするために因子軸を回転する必要があり，バリマックス回転やプロマックス回転がある．前者は因子同士の相関を許さない直交回転，後者は因子同士の相関を想定する斜交回転の代表例である．それぞれ，回転後の因子に相関がある点，ない点が異なる．

こうして求められた因子負荷行列をもとに，どの因子がどの変数にどの程度の影響を与えているのかを見て因子の意味を解釈する．

最終的には因子負荷行列の係数を元に各個体の各因子に対する因子得点を求める．

初期解の求め方，因子数の決定方法，分析時に全ての方法を試し，最適なものを総合的な基準で選択した．

4.2.2 分析結果

質問項目や因子数を変えて様々なパターンでの分析を行った．次の 3 パターンでの結果を示す．まず，20 個の全質問項目を残

したパターン (a)，二つ目は，各変数の関連性をより精密に判断するため，因子空間からずれてしまうノイズ変数を除いたパターン (b)，三つ目は，GPU の 3 項目に強い相関が見られたので，残りの変数の関連性をより詳しく知るためにこの 3 項目を除いたパターン (c) である．

a） 全 20 項目の質問での分析結果

まずはもとの 20 問全ての質問項目を変数として分析を行った．因子負荷行列を図 1 に示す．初期解抽出は最尤法，因子数はカイザー基準による 5 因子モデル，回転はバリマックス法を用いた．因子負荷行列の各行には質問項目，各列には因子が配置されている．各要素は因子の質問項目に対する影響の大きさを表しており，値が大きいほど影響が大きい．正の値ならば因子の値が大きくなるほど質問項目の値が増え，負の値ならば質問項目の値が減る．

	factor1	factor2	factor3	factor4	factor5
並列処理経験年数	0.014	0.541	0.046	0.103	0.496
許容できる実行時間	0.066	−0.087	−0.017	0.980	0.153
許容できるコスト	0.688	0.179	−0.097	0.130	0.106
CPUのFLOPS	0.397	0.159	−0.037	−0.480	0.536
メモリ容量	0.006	0.160	0.114	−0.095	0.705
メモリバンド幅	0.312	−0.138	0.064	−0.447	0.394
GPUのFLOPS	0.447	0.756	0.376	0.020	0.097
GPUのVRAM容量	0.616	0.685	0.266	0.068	−0.006
GPUのVRAMバンド幅	0.491	0.579	0.515	−0.031	0.054
ストレージ容量	0.665	0.047	−0.051	0.028	0.068
ストレージread I/Oバンド幅	0.801	−0.170	0.008	−0.270	−0.159
ストレージwrite I/Oバンド幅	0.829	−0.043	0.382	−0.036	−0.247
ネットワークバンド幅	0.295	0.079	0.842	−0.033	0.438
ネットワーク遅延	−0.147	0.224	0.534	0.043	−0.020
ネットワークトポロジ	0.108	0.050	−0.027	−0.103	−0.512
並列数	0.386	0.093	−0.172	−0.265	0.239
BES	0.035	−0.060	−0.661	−0.302	−0.017
SLA	0.089	0.426	0.222	0.426	−0.006
HPC	−0.147	0.510	0.009	−0.003	0.027
障害対策	−0.023	0.105	0.210	0.411	−0.031

図 2　質問 20 項目での因子負荷行列

b） 因子空間からずれる質問項目を除いての分析結果

次に，いくつかの質問項目の見直しを行った．まず，共通性 (各質問項目の行の因子負荷量の二乗和．質問項目の因子空間へのあてはまりのよさを示す．) が 0.3 未満だったもの：(15) ネットワークトポロジ，(19)HPC キュー，(20) 障害対策の項目を除いた．次に，アプリケーションの実行パターンには直接関係ないと思われる (1) 並列計算の経験年数の項目，また，回答に偏りが見られた (14) ネットワーク遅延を除いた．さらに，本研究ではインプット情報としてではなくアウトプットすることを目的としている (17)，(18) の利用キューに関する質問も除いた．

結果を図 2 に示す．初期解抽出は最尤法，因子数はカイザー基準による 6 因子モデル，回転はバリマックス法を用いた．6 つの因子はそれぞれ次のようなアプリケーション実行パターンを表していると考えられる．

- 因子 1：GPU を利用し，大量のデータを処理する．ストレージの容量や I/O 幅を必要とし，特に書き込み性能を重視する．データの通信量も多い．
- 因子 2：計算量が多く，高速な計算のために CPU 性能やメモリ容量，メモリバンド幅を必要とする．データ通信量も多い．
- 因子 3：大量のデータを用いるため，ストレージの容量，読み書き性能を多く必要とする．
- 因子 4：実行を速く完了させるために，高性能 CPU を並列にたくさん使って計算を行う．データの読み込みが多い．
- 因子 5：GPU を使い，データを大量に読み書きして処理する．比較的大規模のため，時間やコストが多くかかると想定される．
- 因子 6：データの読み書きはあまり多くないが，計算を早く終わらせるために高性能の CPU を用いる．

	factor1	factor2	factor3	factor4	factor5	factor6
許容できる実行時間	−0.033	0.078	0.115	−0.675	0.185	−0.444
許容できるコスト	0.265	0.109	0.104	0.028	0.948	0.039
CPUのFLOPS	0.116	0.476	0.267	0.323	0.086	0.668
メモリ容量	0.079	0.976	−0.129	0.088	0.004	0.110
メモリバンド幅	−0.041	0.439	0.239	0.525	0.109	0.136
GPUのFLOPS	0.962	0.169	0.102	0.088	0.017	−0.044
GPUのVRAM容量	0.897	−0.033	0.181	−0.008	0.310	0.078
GPUのVRAMバンド幅	0.937	0.146	0.107	0.063	0.183	0.095
ストレージ容量	0.217	0.025	0.959	0.010	0.086	0.140
ストレージread I/Oバンド幅	0.155	−0.108	0.617	0.597	0.459	−0.101
ストレージwrite I/Oバンド幅	0.460	−0.023	0.462	0.159	0.512	−0.091
ネットワークバンド幅	0.573	0.684	0.100	0.037	0.098	0.005
並列数	0.066	0.113	0.044	0.620	0.077	0.004

図 3　モデルからずれる変数を除いての因子負荷行列

c） GPU の項目を除いての分析結果

さらに，GPU に関する項目 (7)，(8)，(9) を除き，これらの質問項目に対して「GPU を利用しない」と答えた回答のみを対象に再度分析を行った．

結果を図 3 に示す．初期解抽出は最尤法，因子数はカイザー基準による 5 因子モデル，回転はバリマックス法を用いた．5 つの因子はそれぞれ次のようなアプリケーション実行パターンを表していると考えられる．

- 因子 1：計算量が多く，高性能の CPU をたくさん用いて速く実行を完了させる．データの高速な読み書きも必要とされ，メモリ容量やメモリバンド幅も多く使用する．
- 因子 2：長期的にデータの読み書きを行い続ける．実行時間，コストともに多めにかかると想定される．
- 因子 3：データを大量に読み込んで処理し結果を出力する．比較的小規模で，実行時間もそれほど長くならない，
- 因子 4：大量のデータを用いるため，ストレージの容量，読み書き性能を多く必要とする．
- 因子 5：中間結果を適宜書き込む．データの通信量が多い．

	factor1	factor2	factor3	factor4	factor5
許容できる実行時間	−0.326	0.337	−0.481	0.088	0.221
許容できるコスト	0.054	0.993	−0.006	0.054	0.047
CPUのFLOPS	0.725	−0.096	0.112	0.149	0.087
メモリ容量	0.564	0.460	−0.464	−0.035	0.338
メモリバンド幅	0.650	−0.112	0.489	0.169	0.160
ストレージ容量	0.212	0.009	0.128	0.962	−0.095
ストレージread I/Oバンド幅	0.292	0.212	0.808	0.460	−0.032
ストレージwrite I/Oバンド幅	−0.065	0.241	0.136	0.314	0.373
ネットワークバンド幅	0.225	0.006	−0.165	−0.163	0.944
並列数	0.932	0.215	0.158	0.047	0.008

図 4　GPU 項目を除いての因子負荷行列

5. 因子を利用した数理モデルと考察

5.1 因子を利用する数理モデル

前章で抽出された因子を利用し，ユーザからの入力に対して最適な資源選択を提案するような数理モデルについて検証する．この検証では，因子の抽出方法は前章の (b) のケース：モデルからずれる質問項目を除去したものを用いた．因子分析においては，3 章の (2) 式の左辺から右辺を求めることを行ったが，逆に右辺から左辺を求めることを考える．すなわち，因子負荷行列 A と，因子得点行列 F，独自因子行列 E から，データ行列 X を求められる．

$$AF + E = X \tag{3}$$

この場合，事前のアンケート分析により A，E は既知であるため，あとは因子得点 F を指定することにより，ユーザの資源利用に関する詳細な希望 X の予測が可能である．つまり，入力としてユーザがどの因子に重きをおくかという情報 (F) を受け取り，それを元に，そのユーザが希望するであろう細かな資源選択情報 (X) を出力することができる．これが因子を利用して資源選択の提示を行うモデルとなる．

5.2 モデルの妥当性の検証

モデルがどれだけ正しくユーザが望むような資源選択を提示できるかを検証する．モデル作成時に，分析に取り入れたデータおよび取り入れなかったデータのそれぞれに対し，モデルにより正しく因子得点から元データの予測ができるかどうかを検証する．

検証方法としては，質問項目数を n としたとき，元データの回答とモデルによる予測回答をそれぞれ表す n 次元ベクトル $x = \{x_1, x_2, \ldots, x_n\}$ と，$y = \{y_1, y_2, \ldots, y_n\}$ のカイ二乗距離 d_{chi} を求める．x と y のカイ二乗距離を，

$$d_{chi} = \sqrt{\sum_{i=1}^{n} \frac{(x_i - y_i)^2}{p_i}}$$

ただし，

$$p_i = \frac{|x_i| + |y_i|}{N}$$

$$(N = \sum_{i=1}^{n} (|x_i| + |y_i|))$$

と定義する．

5.2.1 モデル作成に用いたデータによる検証

各ユーザにおける (1) 実際の回答と，(2) モデルから予測された回答を表す n 次元ベクトルのカイ二乗距離を計測した (n = 質問項目数)．この距離を全 25 回答について求めたものを図 4 に示す．なお，比較のため，最悪のケースとなるダミー回答のペアを以下のように 2 組作成し，これらの距離を調べた．

全質問項目について，(1) 全回答群のうち最も高い値のものを選ぶ，(2) 全質問項目について実際の解答群のうちから最も低い値のものを選ぶ，(3) モデルから予測された回答群のうち最も低い値のものを選ぶ,(4) モデルから予測された回答群のうち最も高い値のものを選ぶ．(1) と (3) のカイ二乗距離は 36.0,(2) と (4) のカイ二乗距離は 38.8 となった．

一方，実際の 25 回答のうちでは，最も元回答と予測回答の距離が開いたケースでは 9.6，最も近かったケースでは 2.7 であった．この最悪のケースと最良のケースの元データおよび予測回答 (標準化値) を図 5 に示す．

最良ケースでは，モデルによる予測回答がほぼ元データの回答に添っており，最悪のケースでも，そこまでずれが大きくないことがわかる．

5.2.2 モデル作成に用いなかったデータによる検証

カイ二乗距離による検証で，因子から逆向きに元データをほぼ予測できることは確かめられた．ただしこれらのデータはモデル作成に用いたデータであるため，モデルによく添うのはある意味当然であるといえる．そこで，モデル作成に分析に取り入れなかったデータをモデルに通し，予測されたデータが元のデータに添っているかどうかを検証する．

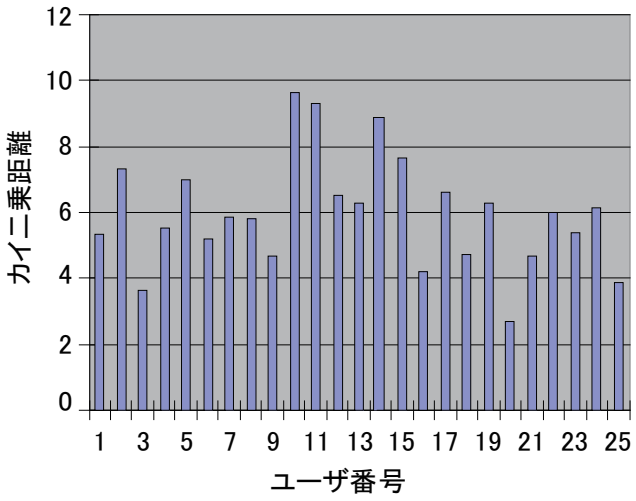

図 5 ユーザごとの元回答とモデルによる予測回答とのカイ二乗距離

5.3 TSUBAME における利用シナリオ

この数理モデルの利用シナリオを，TSUBAME での利用を例に説明する．TSUBAME では，課金対象となるバッチキューへのジョブ投入時に，資源選択に関して下表のようなオプション指定が可能である．

ユーザが投入するアプリケーションをどの利用パターンで実行したいかを，どの因子に重きを置くかという形式で指定することにより入力する．例えば，因子が次の 5 つあるようなモデル：「GPU を利用する」,「計算量が多い」,「データ通信が多い」,「ストレージ利用が多い」,「早く実行を終わらせる」においては，ユーザがどのパターンに何割ずつ近いかを,「5:2:1:2:0」のように指定する．この入力情報から，モデルが，指定するべき資源選択オプションを出力する．

6. まとめと今後の課題

大規模計算環境において，ユーザの満足度を考慮するような

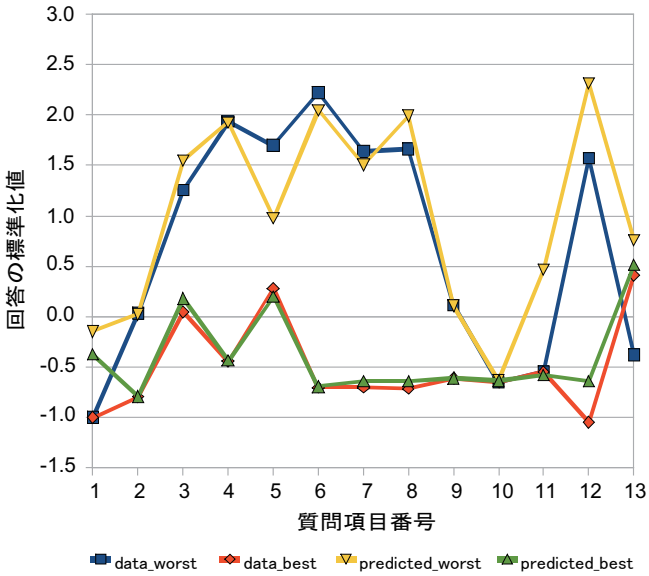

図 6　最悪ケースと最良ケースにおける元回答と予測回答とのずれ

オプション	引数	概要
-smp	<cpus>	ジョブに必要なCPU数を指定. 同一ノード内でのみ実行
-mpi	<cpus>	MPIでの実行およびジョブに 必要なCPU数を指定
-linda	<cpus>	Gaussian Lindaで実行する CPU数を指定(8の倍数)
-ddl	<cpus>	GAMESSを実行するCPU数
-proc	<cpus>	平行実行に必要なCPU数を指定. 同一ノード内でのみ実行
-mem	<memory size>	各プロセスのメモリサイズを指定. 単位はGbytes
-rank0mem	<memory size>	Rank0の使用するメモリサイズ
-stripe	<count>	Lustre FSの組み方を指定する
-q	<Queue Name>	SLAまたはBESのキュー グループの実行時間
-pl	<Premium Level>	使用するプレミアレベルを指定する
-rt	<run time(min)>	実行する時間ユニットを指定する
-et	<Executable Period>	ジョブを実行するキューを指定

図 7　TSUBAME で指定可能な資源に関するオプション

資源選択にむけて，ユーザへのアンケート調査とその分析により利用希望パターンの抽出を行った．また，利用パターンを元に資源選択を提案する数理モデルをたて，検証を行い，ユーザの希望に添うような資源選択が提示できることを確認した．

今後の課題としてはまず，モデルの精度を上げるために，大規模なアンケート調査を行っていきたい．また，実システム環境における混雑状況やハードウェアの制限状況なども考慮に入れられるようなモデルを構築していきたい．将来的には，アプリケーションを投入する際にアンケートに答えてもらい回答を自動でデータベースに取り入れ再分析し続けるようなエキスパートシステムの構築を行っていきたい．また今回の論文ではユーザが実際に満足できるかどうかを，もともとユーザが回答したデータを利用することでしか評価していないため，実際にユーザに使ってもらい，満足できるかどうかを評価実験していきたい．

文　献

[1] R. Buyya, D. Abramson, and J. Giddy, *Nimrod/G: An architecture of a resource management and scheduling system in a global computational grid,*, in Proceedings of the 4th International Conference on High Performance Computing in the Asia-Pacific Region, vol. 1, 2000, pp. 283-289

[2] O. Regev and N. Nisan, *The POPCORN market. online markets for computational resources,*, Decision Support Systems, vol. 28, no. 1-2, pp. 177?189, 2000.

[3] H. Casanova, A. Legrand, and M. Quinson, *SimGrid: a Generic Framework for Large-Scale Distributed Experiments,*, in 10th IEEE International Conference on Computer Modeling and Simulation, 2008.

[4] J. Basney, and M. Livny, *Deploying a High Throughput Computing Cluster,*, High Performance Cluster Computing, R.Buyya(editor).Vol. 1, Chapter 5, Prentice Hall PTR, May 1999.

[5] M. Litzkow, M.Livny, and M.Mutka, *Condor -A Hunter of Idle Workstations,*, Proceedings of the 8th International Conference of Distributed Computing Systems(ICDCS 1988), January 1988, San Jose, CA, IEEE CS Press, USA, 1988.

[6] C. Waldspurger, T. Hogg, B. Huberman, J. Kephart, and W. Stornetta, *Spawn: A Distributed Computational Economy,*, IEEE Transactions on Software Engineering, Vol. 18, No. 2, pp 103—117, IEEE CS Press, USA, February 1992.

[7] J. Brooke, M. Foster, S. Pickles, K. Taylor, and T. Hewitt, *Mini-Grids: Effective Test-beds for Grid Application,*, Proceedings of the First IEEE/ACM International Workshop on Grid Computing (GRID 2000), Dec. 17, 2000, Bangalore, India, Springer Verlag Press, Germany, 2000.

[8] *TSUBAME Grid Cluster,*
http://www.gsic.titech.ac.jp/ ccwww/

[9] *T2K Open Supercomputer,*
http://www.open-supercomputer.org/

[10] R.Huang, H.Casanova, A.Chien, *Automatic Resource Specification Generation for Resource Selection,*, in 10th IEEE International Conference on Computer Modeling and Simulation, 2007.