RC-008

ボランティアコンピューティングの高効率化ためのクライアントレベルスケジューリング

Client-Level Task Scheduling for Effective Volunteer Computing

村田善智* 遠藤聡明† 滝沢寛之‡ 小林広明*
Yoshitomo Murata Toshiaki Endo Hiroyuki Takizawa Hiroaki Kobayashi

1 諸言

計算機科学の発達による民生用コンピュータの高性能化と低価格化により，高性能な個人用コンピュータやゲーム機器が一般家庭に広く普及している．しかし，それらの計算資源は常に利用されているわけではなく，稼働時間の大半は遊休状態となっている．そこで，これらの遊休状態にある計算資源の処理能力を集約して大規模科学技術計算を行う，ボランティアコンピューティングが提案されている．SETI@home [3] や Folding@home [4] などのボランティアコンピューティングプロジェクトでは，膨大な数の遊休計算資源を活用することで数百 TFlops を超える処理性能を実現している．

ボランティアコンピューティングでは，プロジェクトに対して計算資源を提供する者をユーザと呼び，ユーザによって提供された計算機をクライアントと呼ぶ．計算資源の提供はユーザの有志によって行われるものであり，各クライアントはプロジェクトから配布されたタスクの実行に責任を持つわけではない．このため，クライアントはボランティアコンピューティングプロジェクトへの自由な参加・離脱や，タスクの計算の中断を行うことが可能である．一方，サーバがクライアントでのタスク実行のタイミングを指示したり，実行中のタスクを管理したりすることが出来ない．この制約により，ボランティアコンピューティングではタスクの計算時間を保証することが出来ない．このため，計算効率の低下が生じ，高効率な分散計算の実現が困難であるという問題がある．このようなボランティアコンピューティングの問題を解決するためには，プロジェクトサーバにおけるスケジューリングの改善だけではなく，クライアントでのタスクの実行を管理する適切なスケジューリングを同時に実現する必要がある．

本報告の目的は，ボランティアコンピューティングにおける計算資源の利用効率を高めるための，クライアントにおけるタスクスケジューリング手法を確立することである．提案する手法は，各タスクが必要とする計算時間と締め切り時間の関係を考慮したタスク選択を実現する．さらに本報告では，シミュレーションによる性能評価から，本手法によりボランティアコンピューティングにおけるクライアントの利用効率を改善できることを明らかにする．

以下，第 2 節では，ボランティアコンピューティングを実現するデファクトスタンダードな基盤ミドルウェアである BOINC について説明し，複数プロジェクトに参加する場合にクライアントが行うスケジューリングの問題点について述べる．第 3 節では，締め切り時間を考慮したタスクスケジューリング手法を提案する．第 4 節では，シミュレータを用いた提案手法の有効性の評価を行う．第 5 節は，結言と今後の課題について述べる．

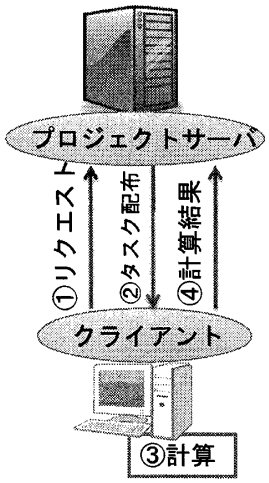


図 1 BOINC 概略

2 Berkeley Open Infrastructure for Network Computing

2.1 BOINC プラットフォーム

BOINC [2] は，現在最も一般的に利用されているボランティアコンピューティングミドルウェアである．BOINC の概略を図 1 に示す．BOINC はプロジェクトを管理しタスクを配布するプロジェクトサーバと，サーバからタスクを受け取り計算を行う多数の計算クライアントから構成される．BOINC を用いたボランティアコンピューティングの処理手順を以下に示す．

1. クライアントがプロジェクトサーバにタスクのリクエストを送信
2. プロジェクトサーバがクライアントにタスクを配布
3. クライアントが遊休計算資源を用いてタスクを計算
4. クライアントがプロジェクトサーバに計算結果を提出

ボランティアコンピューティングによる分散計算では，クライアントマシンのシャットダウンなどによるタスクの実行中

* 東北大学サイバーサイエンスセンター
† 東北大学工学部機械知能・航空工学科 (現在，NTT コムウェア所属)
‡ 東北大学大学院情報科学研究科

断によって，計算結果がいつまでもプロジェクトサーバに提出されなくなる，タスク消失の可能性がある．そこで BOINC では，各タスクに計算結果提出の締め切りが定められている．締め切り時間までに計算結果がプロジェクトサーバへ提出されない場合，プロジェクトサーバはそのタスクの処理が失敗したとみなし，そのタスクを他のクライアントへ再配布する．これにより，特定のタスクがいつまでも完了しない問題を解決している．

一方，タスクの締め切り時間の決定は，ボランティアコンピューティングの計算効率に係わる重要な問題である．タスクの計算時間に対して締め切り時間が短く設定された場合，締め切り後にクライアントから提出された計算結果をサーバは受け取らないため，タスクの計算に費やされたそのクライアントの計算能力が無駄になる．その結果，ボランティアコンピューティングにより多数のクライアントを利用した計算を行ったとしても，実際に有効な計算を行えるクライアントの台数は少なくなり，計算資源の利用効率の低下が生じる．逆に，タスクの計算時間に対して締め切り時間が長く設定された場合には，クライアントのシャットダウンなどによるタスク消失の検出が遅れ，タスクの他クライアントへの再割り当て実行が適切に行われない問題がある．特に，ボランティアコンピューティングのようにクライアントが頻繁に参加離脱を繰り返す環境では，タスク再割り当てが適切に行えないことによって，計算性能の低下や各タスクの計算が完了するまでのターンアラウンドタイムの増加が発生する．また，ボランティアコンピューティング環境には様々な計算性能のクライアントが存在し，タスクの計算時間も一定ではない．そのため，あるクライアントでの計算時間を基に適切な締め切り時間を設定したとしても，違うクライアントにおいても同様に適切な締め切り時間であるとは限らない．

2.2 複数プロジェクト参加時のタスクスケジューリング

Anderson らは文献 [2] において，ボランティアコンピューティングでは 1 台のクライアントが同時に複数のプロジェクトへ参加することで，計算資源の利用効率を改善できると述べている．あるプロジェクトのサーバがメンテナンス等の理由によって停止している場合，クライアントはサーバからタスクを取得することができない．従って，1 つのプロジェクトにしか参加していないクライアントでは，タスクが取得できない間，ボランティアコンピューティングによる計算能力の有効活用を行うことが出来ない．一方，複数プロジェクトに参加しているクライアントでは，メンテナンス中でない他のプロジェクトからタスクを取得することで，計算を続行することが可能である．BOINC では，複数のプロジェクトのアプリケーションを動作させるための実装がなされており，サーバから送られてきたタスクの締め切り時間情報を参照し，複数のプロジェクトの中から締め切りの最も近いタスクを選択して実行する，Earliest Deadline First [6](以下，EDF) スケジューリングが採用されている．

BOINC のプロジェクトサーバは，クライアントが単一のプロジェクトに参加していると想定して，計算結果の締め切り時間を設定している．しかし実際には，クライアントは複数プロジェクトに参加しており，EDF スケジューリングによって自身の遊休計算能力を各プロジェクトに対して均等に割り当てている．このため，サーバが予想したタスクの計算時間と，実際にクライアントがタスク計算を完了させるのに必要な時間が大きく異なり，締め切りまでに計算を完了できないタスクが発生する．締め切りまでにタスクの計算を完了できない場合，それまでに費やしたクライアントの計算資源が無駄となり，ボランティアコンピューティングプロジェクト全体の計算効率が低下する．

一方，竹房らは文献 [8] において，グリッド環境における締め切り時間を考慮したスケジューリング手法について提案し，計算開始前に締め切り時間までにタスクの計算が完了するかを判断することが有効であると述べている．しかしこの手法では，締め切り時間までに計算完了するかの判断は計算開始時に 1 回だけしか行われず，またボランティアコンピューティングの様なタスク計算完了までの時間が動的に変化する環境は想定されていない．従って，計算能力に大きなばらつきが生じるボランティアコンピューティング環境においてクライアントの利用効率を高めるためには，タスクの進捗状況を常に監視し，締め切りまでに計算結果を返却可能かどうかをクライアントが動的に判断するスケジューリング手法を実現する必要がある．

3 締め切り時間を考慮したタスクスケジューリング

本節では，タスク完了時間予測に基づき，締め切りに間に合うタスクのみを計算するスケジューリング手法を提案する．BOINC において，締め切りを過ぎたタスクの計算を続けることは計算能力の浪費を意味する．計算能力の浪費を防ぐために，締め切りを過ぎた，あるいは過ぎることが予測されるタスクをスケジューリングの対象から除外する．これにより，タスクの実行中に締め切り時間を過ぎてしまうことを回避することが可能となる．また，EDF スケジューリングに締め切り時間の判定を導入することで，ボランティアコンピューティング環境における，クライアントでのタスク計算時間の動的な変動に対応したスケジューリングを実現する．

提案スケジューリング手法の処理手順を図 2 に示す．提案手法では，まず EDF 手法を用いて締め切りの近いタスクを選択する．次に，選択したタスクが締め切り前に完了するかどうかを，計算時間を予測することで判断する．完了する見込みが有ればそのタスクの実行を決定するが，完了する見込みが無ければそのタスクは実行しない．タスクが実行されない場合，再び EDF によるタスク選択に戻り，完了見込みがあるタスクが選択されるまで，同様の手順を繰り返す．タスクの選択が完了すると，タスクの計算が開始される．計算開始から一定時間経過後，EDF による実行タスクの切り替えが行われる．タスクがすでに完了していた場合は，その結果をプロジェクトサーバに返却し新たなタスクを要求する．タスクがまだ完了されていな

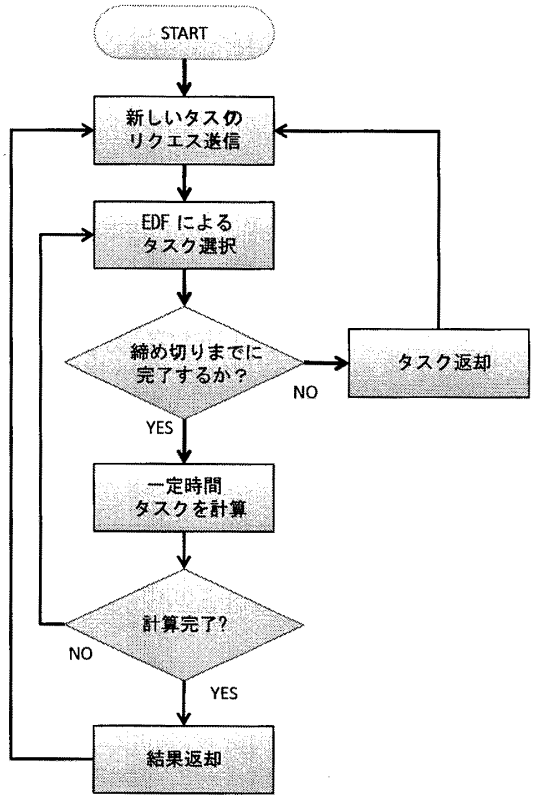


図 2　提案スケジューリング手法の処理手順

かった場合には，EDF によるタスク選択に戻り，同様の手順を繰り返す．

提案タスクスケジューリング手法により実行が中断されたタスクは，即座にプロジェクトサーバに返却される．プロジェクトサーバはタスクが返却されると，そのタスクの締め切りを再設定して他のクライアントに対して再配布する．提案手法では，締め切り時間まで待つことなく，より迅速にタスクの再配布を行うことができる．このため，タスク実行の平均ターンアラウンドタイムを短縮し，ボランティアコンピューティングの実行効率を高めることが可能である．

4　性能評価

本節では，BOINC クライアントのスケジューラに完了時間予測アルゴリズムを追加実装し，提案手法の有効性をシミュレーションによって評価する．シミュレーションには，BOINC のソースコードに付属するクライアントシミュレータを用いる．

4.1　評価指標

提案手法の有効性を評価するための評価指標として，タスク成功数，タスク再配布数，BOINC クライアントの計算効率，タスクの平均ターンアラウンドタイムを用いる．

タスク成功数は締め切りまでに完了したタスク数を示し，この指標が高い値を示すほど遊休計算資源を有効に活用できていることを表している．タスク再配布数は締め切りの超過や実行キャンセルによりサーバへ返却されたタスク数を示す．BOINC クライアントの計算効率は，ボランティアコンピューティングによってクライアントの遊休計算能力をどれだけ効率的に回収できたかを評価する．

BOINC クライアントの計算効率による評価を行うために，BOINC クライアントが稼働していた時間のうち，締め切り時間までに計算が完了したタスクに費やした計算時間，締め切り時間までに計算が完了しなかったタスクに費やした計算時間，そして BOINC クライアントが稼働していたにもかかわらず何もタスクを計算していなかった時間を調査する．BOINC クライアントの処理内容のうち，締め切り時間までに計算が完了したタスクに費やした計算時間の割合が大きいほど，ボランティアコンピューティングは計算資源のより高い利用効率を実現すると評価できる．また，BOINC クライアントが稼働していたにもかかわらずクライアントがタスクを計算していない時間には，プロジェクトサーバからのタスク配布待ち時間や，タスクを転送する際のネットワークオーバヘッドの時間などが含まれる．

ターンアラウンドタイムはタスクが配布されてから計算結果が提出されるまでの時間を示し，このターンアラウンドタイムが短いほど高い応答性能を実現するボランティアコンピューティング環境が構築可能であることを表している．

評価実験では，提案手法のほかに 2 種類のスケジューリング手法を比較対象として用いた．Round-Robin スケジューリングは，サーバから配布された順にタスクを選択する [5]．このスケジューリング手法は締め切りまでの時間を考慮しないため，締め切り時間が短い場合や，参加プロジェクト数が多い場合に，著しい性能低下が起こることが予想される．EDF スケジューリングは，現在の BOINC クライアントが行うスケジューリングであり，締め切りの近いタスクを選択して計算を行う．

4.2　シミュレーションパラメータ

BOINC プロジェクトで用いられるシミュレーションパラメータ [1] を基に，評価に用いるシミュレーションパラメータを決定する．表 1 に評価に用いるシミュレーションパラメータを示す．

プロジェクトサーバから配布されるタスクの計算に必要な時間は，平均 60 分，標準偏差 σ=10 の正規分布に従うものとする．スケジューリング間隔は，クライアントがスケジューリングを行う間隔を表す．CPU コア数は，1 台のクライアントが同時に計算できるタスク数を表す．シミュレーション時間は，シミュレーションを行った時間の長さを表す．タスク締め切り時間は，タスクがプロジェクトサーバから配布される際に設定される締め切りまでの時間を表す．参加プロジェクト数はクライアントが参加しているプロジェクトの数を表し，各プロジェクトが持つタスクの計算時間は全て平均 60 分とする．BOINC 稼働率は，シミュレーション時間のうち，クライアントマシンが BOINC クライアントを動作させている時間の割合を表す．BOINC クライアントはクライアントマシンが遊休状態のとき

表 1　シミュレーションパラメータ

パラメータ名	値
平均タスク計算時間 [分]	60
スケジューリング間隔 [秒]	60
クライアント CPU コア数	2
シミュレーション時間 [分]	6,000
締め切り時間 [分]	60, 90, 120, 150, 180, 210, 240, 270, 300
参加プロジェクト数	1, 2, 3, 4, 5, 6, 7, 8
BOINC 稼働率 [%]	10, 20, 30, 40, 50, 60, 70, 80, 90, 100

に動作していることから，BOINC 稼働率はクライアントマシンが遊休状態にある割合も表している．各クライアントマシンは 1 分毎に BOINC 稼働率に従った動作状態の遷移を行い，BOINC クライアント動作時には全ての計算能力を BOINC のタスクに割り当てるものとする．

評価実験では，タスク締め切り時間，参加プロジェクト数，BOINC 稼働率をそれぞれ変化させ，各スケジューリング手法での性能変化を測定することで，提案手法の有効性を評価する．

4.3　評価結果

4.3.1　締め切り時間が性能に与える影響の評価

本節では，各スケジューラのスケジューリング能力に対してタスク計算の締め切り時間が与える影響について評価する．シミュレーションパラメータのうち締め切り時間を 90〜300 分の間で変化させ，参加プロジェクト数を 4，BOINC 稼働率を 80% として評価を行った．

締め切り時間を変化させた場合のタスク成功数を図 3 に示す．Round-Robin を用いた場合，締め切り時間が 120 分以下のでは成功タスク数は 0 であり，締め切り時間が長くなるにつれて緩やかにタスク成功数が増加する．EDF を用いた場合，締め切り時間が 180 分以上の時，Round-Robin に比べて成功タスク数が大きく向上している．しかし，締め切り時間が 150 分以下では Round-Robin と同様に成功したタスク数は少ない．一方，提案手法を用いる場合では，タスク締め切り時間が 150 分以下の時，タスク成功数に大きな向上がみられる．また，締め切り時間が 180 分以上の時でも，EDF を用いた場合と同様の高いタスク成功数を実現している．このことから，提案手法ではタスク締め切り時間に依存しない，高効率なボランティアコンピューティングを実現できる．

締め切り時間を変化させた場合のタスク再配布数を図 4 に示す．締め切り時間が 210 分以上の時，提案手法および EDF のタスク再配布数は 0 となる．これは，締め切り時間を考慮した

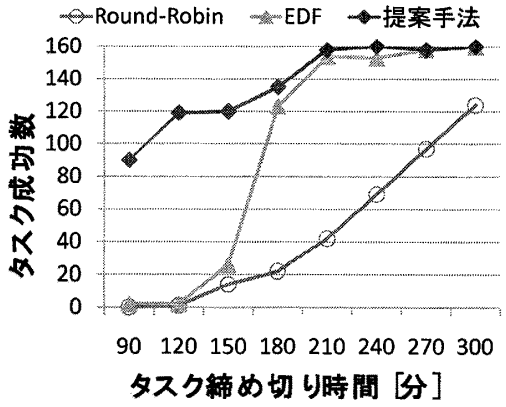


図 3　タスク締め切り時間とタスク成功数

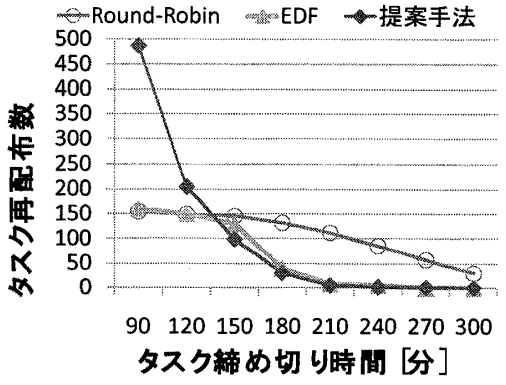
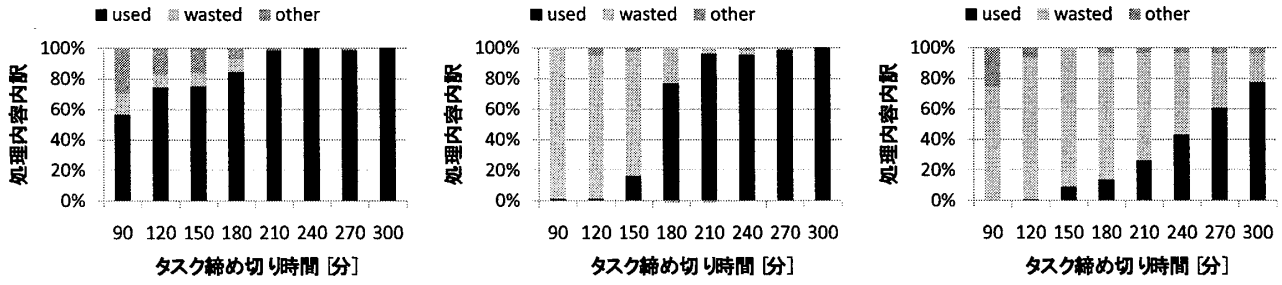


図 4　タスク締め切り時間とタスク再配布数

実行タスク選択により，クライアントが取得したタスクを確実に締め切り時間までに完了させることが出来るためである．一方，締め切り時間が 120 分以下の場合，提案手法は EDF と比較して非常に大きなタスク再配布数となる．これは，提案手法がタスクの計算が締め切り時間までに間に合わないと判断し，プロジェクトサーバへのタスクの返却を頻繁に行うためである．この頻繁なタスク返却と再配布により，タスクの返却・再配布の通信に求められるオーバーヘッドが増大し，ボランティアコンピューティング全体で性能が低下する可能性がある．

次に，クライアントにおける処理内容の内訳を図 5 に示す．図の *used* は締め切り時間までに計算が完了したタスクに費やした計算時間を表し，*wasted* は締め切り時間までに計算が完了しなかったタスクに費やした計算時間を表す．また *other* は，クライアントにおいてタスクの計算が行われていなかった時間を表す．EDF および Round-Robin を用いた場合，締め切り時間が 120 分以下の時の BOINC クライアントの計算効率は 0% である．それに対して，提案手法を用いた場合，締め切り時間が 120 分以下の時でも 50% 以上の計算効率を実現しており，ボランティアコンピューティングによる計算の能力の回収効率が高いことが確認できる．また，締め切り時間までに計算が完了しないタスクに費やす計算能力の浪費に関しても，提案



(a) 提案手法　(b) EDF　(c) Round-Robin

図 5　タスク締め切り時間が変化した時の BOINC クライアントの処理内訳

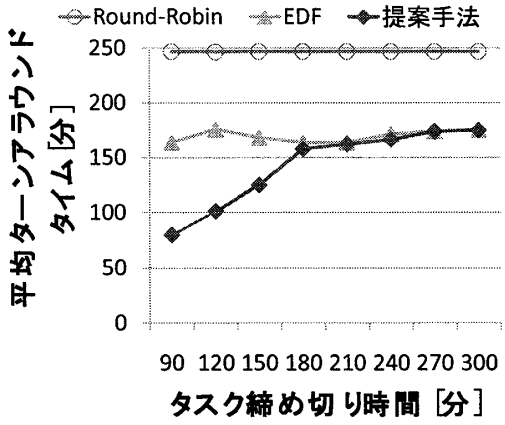


図 6　タスク締め切り時間と平均ターンアラウンドタイム

手法は EDF や Round-Robin と比較して，締め切り時間に間に合わず無駄に失われていた CPU サイクルを最大で 9 割程度削減することができた．一方，提案手法を用いた場合，他の 2 手法に比べてタスク計算を行っていない時間が多い．これは，上記のタスク再配布数に関する評価で述べたように提案手法が頻繁にタスクの取得と返却を繰り返すため，プロジェクトサーバからタスクを割り当てられていない時間が生じているためである．頻繁なタスクの取得と返却による，クライアントの計算不可能状態の発生は，今後の解決すべき課題の 1 つである．

図 6 に，締め切り時間を変化させた場合の平均ターンアラウンドタイムの結果を示す．Round-Robin を用いた場合，平均ターンアラウンドタイムは，締め切り時間に関係なく常に一定である．これは評価で利用した全てのタスクの平均計算時間が一定であり，これらのタスクを締め切り時間に関係なく順次実行しているためである．提案手法と EDF 手法を用いた場合，締め切り時間が 180 分以上の時は両者は同じ結果を示すが，締め切り時間が 180 分未満では提案手法のほうがより短い平均ターンアラウンドタイムを示している．特に，提案手法を用いた場合，どの締め切り時間においても，平均ターンアラウンドタイムは必ず締め切り時間より短くなる．これは，提案手法が締め切り時間内に完了するタスクのみを実行し，締め切り時間が過ぎたタスクの計算結果をサーバに提出することがないためである．また，締め切り時間が 180 分以上では，提案手法を用いた場合でも EDF を用いた場合でも，平均ターンアラウンドタイムは一定となっている．これは，提案手法によって締め切りに間に合わないと判断されるタスクの数が減少し，EDF 手法によるタスクスケジューリングの影響のみが現れるためである．

以上の結果より，提案手法は，設定するタスク締め切り時間に依存せず，常に高性能なボランティアコンピューティングを実現できるスケジューリング手法であるといえる．

4.3.2　参加プロジェクト数が性能に与える影響の評価

本節では，各スケジューラのスケジューリング能力に対してクライアントが参加するプロジェクト数が与える影響について評価する．シミュレーションパラメータのうち参加プロジェクト数を 1〜8 の間で変化させ，締め切り時間を 240 分，BOINC 稼働率を 80% とする．

参加プロジェクト数を変化させた場合のタスク成功数を図 7 に示す．実験結果から，参加プロジェクト数が多くなるにつれて全体的にタスク成功数は減少する．これは，参加プロジェクト数が増えるほど，プロジェクト 1 つあたりに割り当てられるクライアントの計算能力が減少し，締め切り時間までに計算が完了するタスクが減少するためである．提案手法は，タスクが締め切り時間までに完了するかを判断するため，他の手法に比べてタスク成功数の減少が低く抑えられている．特に参加プロジェクト数を 6 以上のとき，提案手法と EDF のタスク成功数に大きな差が確認できる．

参加プロジェクト数を変化させた場合のタスク再配布数を図 8 に示す．参加プロジェクト数の増減に関わらず，提案手法でのタスク再配布数は常に他の手法よりも低い値となった．また，参加プロジェクト数を増やした場合にその傾向は顕著であり，複数プロジェクトに参加した場合に提案手法が効率的なスケジューリングを行えることができる．

参加プロジェクト数を変化させた場合のクライアントの処理内訳を図 9 に示す．実験結果より，EDF を用いた場合と Round-Robin を用いた場合，参加プロジェクト数の増加に従い，締め切り時間に間に合わないタスクを実行することによる計算能力の浪費が増大している．一方提案手法は，参加プロ

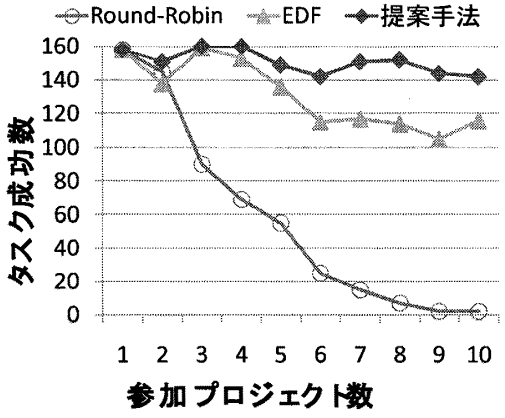


図 7　参加プロジェクト数とタスク成功数

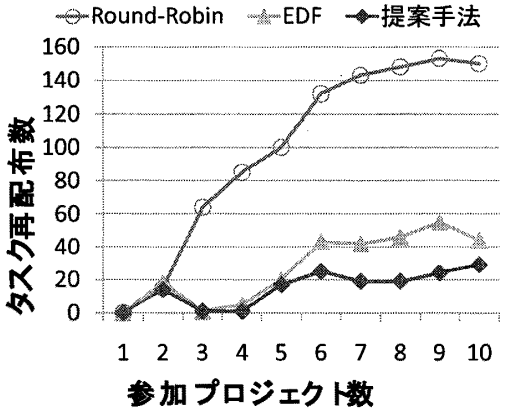


図 8　参加プロジェクト数とタスク再配布数

ジェクト数に関わらず計算資源の有効利用率が最も高くなり，締め切り時間に間に合わないタスクの計算による計算能力の浪費が生じにくいことが確認できる．参加プロジェクト数が多くなるにつれて提案手法と他の手法の差が顕著であることが確認できる．

参加プロジェクト数を変化させた場合の平均ターンアラウンドタイムを図 10 に示す．全てのスケジューラにおいて，参加プロジェクト数が増加するとターンアラウンドタイムも増加する傾向が見られた．しかし，提案手法ではその増加量が他の手法よりも低く抑えられていることが図 10 より確認できる．

以上の結果より，多数のプロジェクトに参加するときのスケジューリングとして提案手法が有効であることが示された．

4.3.3　BOINC 稼働率が性能に与える影響の評価

本節では，各スケジューラのスケジューリング能力に対して BOINC クライアントの稼働率が与える影響について評価する．シミュレーションパラメータのうち BOINC 稼働率を 10〜100% の間で変化させ，締め切り時間を 240 分，参加プロジェクト数を 4 として評価を行った．

BOINC 稼働率を変化させた場合のタスク成功数を図 11 に示す．EDF と Round-Robin を用いた場合，BOINC 稼働率が

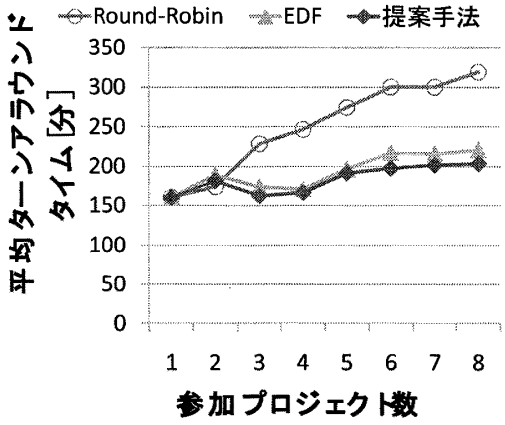


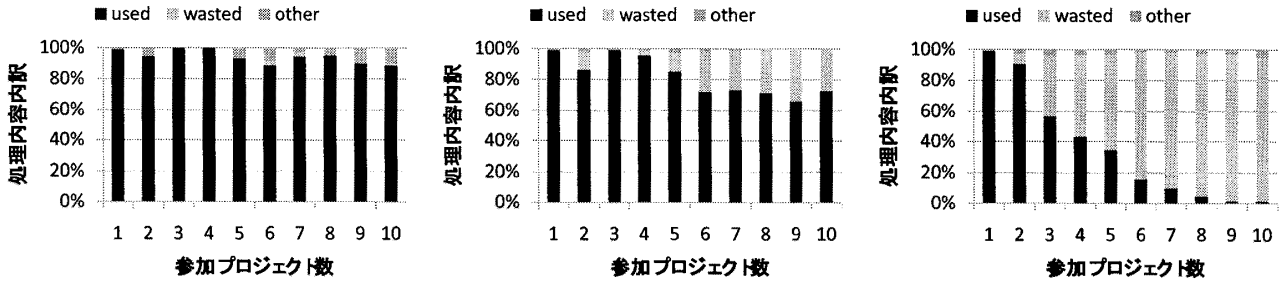
図 10　参加プロジェクト数と平均ターンアラウンドタイム

30% 以下の時では，ほとんどのタスク計算が完了していない．一方，20% 以上の BOINC 稼働率において，提案手法では他の 2 手法よりも高いタスク成功数を示している．しかし，図 3, 7 ほどの提案手法によるタスク成功数の大きな増加は確認できない．これは，BOINC 稼働率が低い場合，BOINC クライアントが連続で稼働している時間がタスクの締め切り時間よりも短いためである．特に BOINC 稼働率が 10% では，ほぼ全てのタスクの計算において，計算が完了する前にクライアントが停止状態となり，そのまま締め切り時間を超過して計算失敗となっている．

BOINC 稼働率を変化させた場合のタスク再配布数を図 12 に示す．この図から，BOINC 稼働率が低い場合における提案手法のタスク再配布数が他の手法よりも多くなっていることがわかる．しかし，BOINC 稼働率が低い場合にタスク成功数も多いことから，締め切り時間が短い場合にタスク再配布数が増加した現象と同じ事が起きていると推測される．BOINC 稼働率が 30% 以下の場合は，提案手法によるタスク再配布数が多い傾向が続くが，BOINC 稼働率が 40% 以上の場合にはこの傾向が逆転し，提案手法によるタスク再配布数は低く抑えられている．

BOINC 稼働率を変化させた場合のクライアントの処理内訳を図 13 に示す．EDF と Round-Robin を用いた場合，BOINC 稼働率が 60% 以上でなければ計算資源の利用効率が 50 %以上にならないのに対し，提案手法を用いた場合ではでは BOINC 稼働率が 20 %の時点で 50 %以上の計算資源の利用効率を実現している．また，BOINC 稼働率が 20 %以上の場合でも，提案手法は常に最大の計算資源の利用効率を示している．一方，締め切り時間を変化させたときの評価と同様に，提案手法ではタスクを計算していない時間の割合が，他の手法に比べて大きい．これは，締め切り時間が短い場合と同様に提案手法がプロジェクトサーバからのタスクの取得と返却を頻繁に行うためで，プロジェクトサーバからのタスク取得時間が非常に大きな問題であることが確認できる．

BOINC 稼働率を変化させた場合の平均ターンアラウンドタ



(a) 提案手法　(b) EDF　(c) Round-Robin

図 9　参加プロジェクト数が変化した時の BOINC クライアントの処理内訳

イムを図 14 に示す．BOINC 稼働率が低くなるほど，Round-Robin 手法や EDF 手法での平均ターンアラウンドタイムは増加するが，提案手法では平均ターンアラウンドタイムはほぼ一定である．Round-Robin 手法や EDF 手法を用いた場合，BOINC 稼働率の低下により単位時間当たりにタスクに割り当てられる計算能力は低下する．その結果，タスクの計算時間が増加し，特に BOINC の稼働率が 10〜20% の低い場合，ターンアラウンドタームの大きな増加を引き起こしている．一方，提案手法は締め切り時間に間に合うタスクのみを実行し，その結果をプロジェクトサーバに返却するため，他の手法の様なターンアラウンドタイムの増加は生じない．

BOINC 稼働率が低い場合，クライアントによるタスクの計算時間が増加するため，締め切り時間に基づくタスクの再配布が必要である．しかし，Round-Robin 手法や EDF 手法では，タスクに設定された締め切り時間になるまで，タスクの他クライアントへの再割り当てを行うことが出来ない．一方，提案手法ではタスクが完了できないと判断されると，タスクの締め切り時間前だとしても直ちにタスクの他クライアントへの再割り当てを行うことが可能である．実際，シミュレーション結果から，提案手法によるタスク再配布数の増加と，ターンアラウンドタイムの減少が確認できる．以上のことから，BOINC クライアントの稼働率が低いときの提案手法の有効性が明らかである．

5　結言

本報告では，クライアントが複数プロジェクトに参加し，タスクを締め切りまでに計算しきれない状況を考慮し，タスクを選択実行するスケジューリングを提案した．またシミュレーションによる評価結果より，本手法によって締め切りに間に合わないタスクに計算資源を浪費することを防ぎ，締め切りに間に合って計算が完了するタスク数を増加させることが可能であることを示した．

今後は，タスクの返却と再リクエストがネットワークやサーバにかける負荷を評価する．タスク返却によるネットワークの負荷を評価することで現実的なタスク返却の頻度を明らかにし，高性能ボランティアコンピューティングシステムを構築するための指針を明らかにする．また，スケジューラによって返却と判断されたタスクをサーバに直接返却するのではなく，クライアント間で直接タスクの移動を行うことを検討する [7]．これにより，サーバへの負荷集中を回避し，より大規模なボランティコンピューティング環境の実現を目指す．

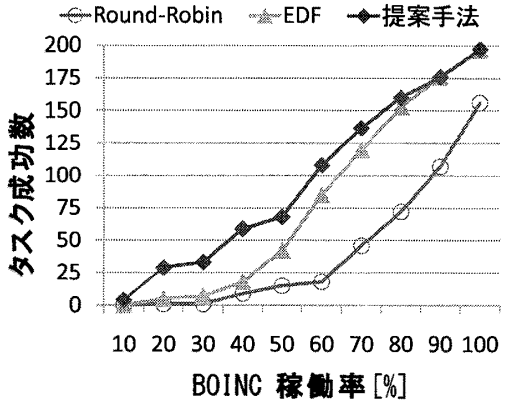


図 11　BOINC 稼働率とタスク成功数

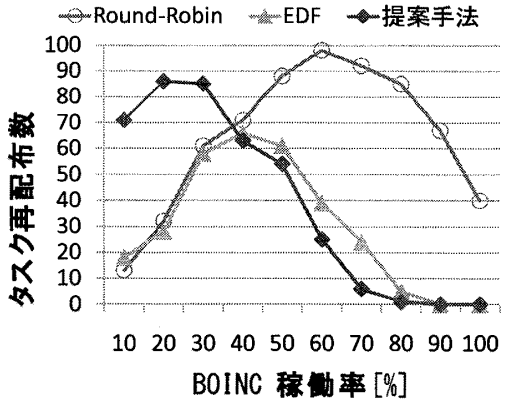
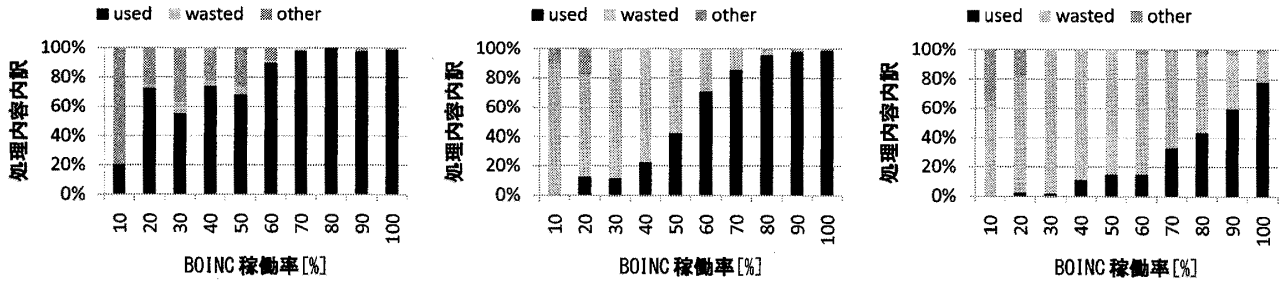


図 12　BOINC 稼働率とタスク再配布数

謝辞

本研究の一部は，文部科学省科研費特定領域研究 (18049003)，および総務省特定領域重点型研究開発



(a) 提案手法　(b) EDF　(c) Round-Robin

図 13　BOINC 稼働率が変化した時の BOINC クライアントの処理内訳

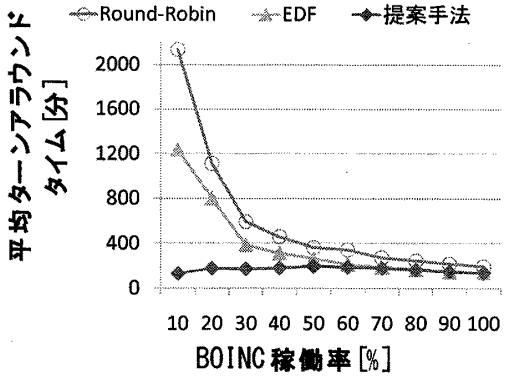


図 14　BOINC 稼働率と平均ターンアラウンドタイム

(061102002) の支援を受けている.

参考文献

[1] "SourceCode BOINC Trac". `http://boinc.berkeley.edu/trac/wiki/SourceCode`.

[2] David P. Anderson. Boinc: A system for public-resource computing and storage. In *5th IEEE/ACM International workshop on Grid Computing*, pp. 4–10, November 2004.

[3] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. Seti@home: An experiment in public-resource computing. *Communications of the ACM*, Vol. 45, No. 11, pp. 56–61, November 2002.

[4] Folding@home distributed computing. `http://folding.stanford.edu/`.

[5] J. Furnkranz. Round robin classification. *Machine Learning Research*, Vol. 2, pp. 721–747, Mar 2002.

[6] M. Kargahi and A. Movaghar. Non-preemptive earliest deadline first scheduling policy: a performance study. In *IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, pp. 201–208, Sept 2005.

[7] 村田善智, 稲葉勉, 滝沢寛之, 小林広明. 大規模計算環境における分散協調型負荷分散手法. 情報処理学会論文誌特集号 "新しいパラダイムの中での分散システム/インターネット運用・管理", Vol. 3, No. 49, pp. 1214–1228, 2008.

[8] 竹房あつ子, 松岡聡. Grid 計算環境におけるデッドラインスケジューリング手法の性能. 情報処理学会 電気通信処理学会 並列シンポジウム JSPP2001 論文集, pp. 263–270, 2006.