

Hadoop データセンタの運営における留保価格を用いた料金設定手法の検討

真鍋 優¹ 川原 純¹ 笠原 正治¹

概要：記憶容量の増大や計算性能の上昇といったハードウェアの発展に伴い、アプリケーションの取り扱うデータ量や計算量は日々増大している。そこで注目されているのが大規模なコンピュータクラスタにおいて処理を分散することで高速化を図る Hadoop 等の分散処理フレームワークである。Hadoop は元来少人数での使用を想定されているため、大規模な計算資源を多人数で共有するには計算資源の割当てを行う新しいスケジューラが必要である。加えて、データセンタ等において商業目的で Hadoop を運用するには、データセンタ管理者の収益を考慮した料金の設定手法が必要になる。本研究は利用者の入札に基づいたオークションにより計算資源の割当てを行うスケジューラを提案する。評価実験では、入札額と利用料金に合わせた計算資源の割当てに成功し、従来手法と比較するとデータセンタ管理者の得る収益が最小で 29%、最大で 297%増加した。

1. 研究背景

記憶容量の増大や計算性能の上昇といったハードウェアの発展に伴い、アプリケーションの取り扱うデータ量や計算量は日々増大している。そこで注目されているのが大規模クラスタにおいて処理を分散することで高速化を図る Hadoop [1] 等の分散処理フレームワークである。

Hadoop とは Google が開発した MapReduce [2] に影響を受け開発されたオープンソースソフトウェアである。MapReduce とはジョブを Map として細分化した上で各計算ノードで実行し、結果を Reduce としてまとめる分散処理フレームワークである。Hadoop バージョン 2 では Hadoop バージョン 1 のスケーラビリティを改善するため、YARN [3] が実装された。図 1 に YARN のイメージ図を示す。YARN において、Hadoop バージョン 1 においてクラスタとジョブの管理を行う jobtracker の機能は ResourceManager と ApplicationMaster に分割され、それぞれがクラスタの管理とジョブの管理を行う。YARN より導入された NodeManager では、コンテナと呼ばれる計算資源の単位において Map あるいは Reduce の処理を実行する。これらの変更により Hadoop バージョン 1 のボトルネックが解消され、スケーラビリティの向上を実現した。

YARN の導入により 4,000 ノードを超える大規模クラスタの管理を実現したことで、計算量の膨大な科学技術計算

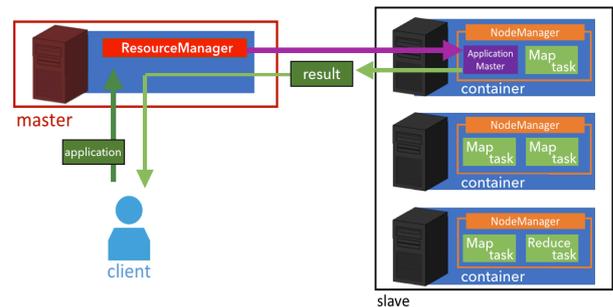


図 1 Hadoop2系 YARN イメージ図

やビッグデータの解析等のための運用が期待されている。しかし大規模なクラスタや膨大な量のデータセットを利用者が個人で用意することは容易ではなく、それらを複数人で共有することが求められる。本研究は利用者の利用料金とデータセンタ管理者の収益を考慮した、複数人で共有可能な Hadoop データセンタを構築することを目指す。

Hadoop を導入したクラスタで複数人の同時利用を可能とするために解決すべき主な課題として、動的な計算資源割当て手法の考案が挙げられる [4]。先行研究において、計算資源の割当てを利用者の申告する時間あたりの利用料金から決定する、Dynamic Priority parallel task scheduler [4] (以後 DP スケジューラ) なる手法が提案されている。しかし DP スケジューラでは、並列する利用者の数が少ないとき安価な利用料金で多くの計算資源を利用でき、データセンタ管理者の収益が減少するという問題がある。

本研究ではオークションメカニズムを導入した新しいス

¹ 奈良先端科学技術大学院大学
Nara Institute of Science and Technology

ケジューラを提案する。本提案手法は割当てる計算資源の量に合わせて下限の料金を設定することで、データセンタ管理者の収益の減少を防止する。評価実験において先行研究と利用料金の比較を行った結果、データセンタ管理者の得る収益は最小で 29%、最大で 297%増加した。

続く第 2 章では Hadoop クラスタ共有のための計算資源割当てを行うスケジューラや料金設定に関する先行研究について紹介する。第 3 章では本研究で提案する手法について説明し、第 4 章においてその評価のために行った実験について述べる。最後に本研究の結論と今後の課題を記す。

2. 先行研究

本章では Hadoop に標準で実装されているスケジューラと、複数人に対して動的な割当てを行うため考案された DP スケジューラ、そして本研究で用いるオークションメカニズムに関連する先行研究について述べる。

2.1 Hadoop の標準スケジューラと DP スケジューラ

Hadoop は元来少人数による利用が想定されており、それは計算資源の割当てを行うため Hadoop が標準で搭載している FIFO, Fair, Capacity という 3 種のスケジューラの仕様からも伺い知ることができる。FIFO スケジューラはアプリケーションを到着順に実行するスケジューラである。最初に実行されたアプリケーションに全ての計算資源を割当て、終了次第、次に実行されたアプリケーションに計算資源を割り当てる。Fair スケジューラは全利用者に割当てる計算資源の量を均等にすることを試みるスケジューラである。Fair スケジューラでは事前に利用者ごとに対応するプールとその優先度を設定する必要がある。利用者は設定されたプールにジョブを投入し、同プール内の利用者と同様になるよう計算資源が割当てられる。プールを複数の利用者で共有するため、FIFO とは異なり、計算資源を長時間保持している場合に割り込みを発生させることが可能となっている。Capacity スケジューラは事前に設定した割合だけ計算資源を割り当てるスケジューラである。例えば利用者 A に 30%、利用者 B に 70%と設定した場合、利用者 A が投入するアプリケーションはクラスタ全体の計算資源の 30%を、利用者 B は 70%を使用することができる。この割合はクラスタ全体の CPU 性能あるいはメモリ使用量を表している。これらのスケジューラは、複数のユーザの参加離脱が考えられる状況で必要となる動的な性質に欠けているという課題があり、複数人で利用するには新たなスケジューラが必要となる。

複数人で計算資源を共有するため開発されたのが DP スケジューラである。DP スケジューラの利用者は事前に各自が時間あたりに支払うことのできる金額である支出率を申告する。そしてそれぞれに割当てられる計算資源の量は、支出率が全体の何割を占めるかで決定する。例えば利

用者 A が支出率を 0.3、利用者 B が 0.7 と申告し利用者がその 2 名のみとすると、割り当てられる計算資源は利用者 A に 30%、利用者 B に 70%となる。計算資源はコンテナと呼ばれる単位に分割され、設定された割合に合わせた数が割当てられる。割当てられた複数のコンテナにおいて利用者はアプリケーションを実行し、一定間隔ごとに時間あたりに使用したコンテナ数に支出率を掛け合わせた分の料金を支払う。

DP スケジューラを利用することで、新規の利用者や、実行を終え離脱する利用者の設定をデータセンタ管理者がその都度行う必要がない。また、一定間隔ごとに計算資源の割当ての比率を計算することで、動的な利用が可能となっている。先行研究における評価実験において、DP スケジューラは各ユーザに割り当てる計算資源の割合を動的に変化させることと、最大 80 名のユーザによる並列利用が可能であると示されている。

DP スケジューラの仕様として利用できる計算資源の量を全体の支出率の割合で決定することから、利用者は全体の利用者が自身のみであると知った場合、支出率を極端に減少させることが考えられる。そのため、DP スケジューラを用いたデータセンタの運営において、収益がデータセンタ管理者の想定を下回る恐れがあることが課題となる。その原因となるのが、並列する利用者の数が少ないとき、利用者の計算資源に対する評価額を下回る申告でも多くの計算資源を利用可能になってしまう点である。そこで、商用データセンタを運営するには利用者が評価額を正しく申告するという性質を確立することと、データセンタ管理者の収益が保証される仕組みが必要であると考えた。本研究ではそのような料金設定手法の実現にオークションメカニズムを用いる。

2.2 オークション理論

本研究において用いるオークションメカニズムとはオークションの仕組みを取り入れた計算資源割当て手法であり、データセンタの利用者および管理者の利得を増加させるためオークション理論の導入を検討している。オークション理論とは社会全体の利得である社会的余剰を最大化することを目的とした資源分配の理論であり、昨今現実の問題に適用されることが増加し注目を集めている。基本的に、オークションでは売り手が公開した単一の財、あるいは複数の財に対して買い手は自らの評価額を入札し、最大の評価額を入札した買い手が財を落札する。オークション理論を導入した仕組みを設計するにあたって、不正が行われないうことと、買い手にとって評価額を正しく入札することが最適であるように設計することが求められる。この特徴により利用者の虚偽の申告を防止することが可能となる。

オークション理論はデータセンタでの利用にも応用できると期待されており、関連する先行研究として [5] や [6] が

ある。[5]は商用のクラウドサービスにおける料金設定に関する内容となっている。昨今、Amazon EC2 [7]のスポットインスタンス [8]等、仮想マシンをオークションによる料金設定に基づいて提供するサービスが存在する。しかし、現状ではオークションによって決定される仮想マシンの利用料金は、仮想マシンの性能の違いを十分に考慮できていない。そのため市場の要求に適切に応えるためには、仮想マシンに対する利用者の要求・評価を正しく把握し、複数の財を取り扱う組み合わせオークションによって異なる種類・場所の仮想マシンをまとめて取り扱う必要がある。[5]ではデータセンタや仮想マシンの種類の違い、利用者の予算等、多くの制約を含んだオークションメカニズムを設計している。[6]は電力を効率的に送電するスマートグリッドにおいて、データセンタへの送電量にデマンドレスポンスを導入することを目標とした研究である。デマンドレスポンスとは、電気料金の高騰のような状況の変化に応じて需要者側から送電量を変化させることである。デマンドレスポンスは消費電力を削減し、電力の効率的運用を行う優れた手法であるが、データセンタにおいて電力の減少は可用性の面から避けるべきと考えられ、導入が難しいとされている。[6]ではその課題の解決のためにオークション理論を導入し、データセンタとスマートグリッドの社会的余剰を最大化することを試みた。

商用のデータセンタにおいて計算資源を提供するにあたり、データセンタ管理者の利益を保証するためには割当てる計算資源の量に合わせた利用料金の下限を設けることが有効であると考えられる。財の料金の下限を用いた複数財のオークションメカニズムとして Leveled Division Set protocol [9](以後 LDS プロトコル)が挙げられる。LDS プロトコルでは売り手が事前に財の全ての組み合わせに売値の下限である留保価格を設定する。一方買い手もそれらの財の組み合わせ全てに入札を行い、オークションを行う。オークションは組み合わせた財の数が最も多いものから行い、その組み合わせの留保価格を上回る入札があった場合、落札となる。落札した際の支払額は一般化 Vickrey オークション [10]により決定される。一般化 Vickrey オークションとは、最高額の入札を行った買い手が2番目に高い入札額で落札する Vickrey オークション [11]を複数財に適用した手法である。一般化 Vickrey オークションにおいて、落札した買い手の支払額は、落札した買い手が参加しない場合の社会的余剰から落札した買い手が参加した場合の他の買い手の社会的余剰を引いた額となる。これらの値は自身の評価額に左右されることはないため、買い手は自身の評価額通りの入札を行うことが最適となる。LDS プロトコルではそのような一般化 Vickrey オークションの特徴を満たしつつ、一般化 Vickrey オークションの課題であった架空名義による入札を解決することに成功した [9]。

3. 提案手法

本研究では、DP スケジューラで考案された利用者ごとの支出率に基づいた計算資源割当て手法と、LDS プロトコルで考案された財のセットと留保価格を用いたメカニズムを組み合わせた手法を提案する。計算資源の割当ては DP スケジューラと同様に利用者の入札する時間あたりの支出率に基づいて行う。異なるのは割当てる計算資源の量を入札額である支出率の単純な比率にせず、利用者間のオークションによって決定する点である。

オークションは一定間隔ごとに行われ、その都度割当てる計算資源の量を各利用者の入札に基づいて決定する。データセンタ管理者は計算資源を N 等分に分割し、 $1/N$ の倍数の計算資源を利用者に提供する。また、データセンタ管理者は $i = 1, 2, \dots, N$ に対し、 i/N の計算資源それぞれに留保価格 R_i を設定し、利用者に提示する。これは i/N の計算資源を利用するために必要な価格の下限を意味する。各利用者はデータセンタで実行するジョブを持つ。利用者はデータセンタ管理者が提示した留保価格と自身の予算から利用を希望する計算資源の最大量を決定する。利用者 j の希望する計算資源の最大量を M_j とする。利用者 j は $i = 1, 2, \dots, M_j$ に対し、 i/N の計算資源それぞれに入札を行う。この入札額は利用者が支払うことを希望する金額以下であり、留保価格 R_i 以上である必要がある。

次に各利用者に割当てる計算資源の量を決定するメカニズムについて述べる。最初に全ての計算資源量に対して、留保価格より低い入札を行った利用者を拒否する。拒否されなかった利用者に対して $1/N$ の計算資源が割当てられる。ここでは拒否されなかった利用者の数を x とする。次に残りの $1 - x/N$ の計算資源の割当てを試みる。 $1 - x/N + 1/N$ の計算資源に対する各利用者の入札を確認する ($+1/N$ は最初に割り当てた計算資源の分である)。留保価格以上の入札が1つだけ行われていた場合、その入札を行った利用者に $1 - x/N$ の計算資源を割当てる。留保価格以上の入札が複数行われていた場合、それらの中で最大の入札を行った利用者に $1 - x/N$ の計算資源を割当てる。留保価格以上の入札が行われていなかった場合、 $1/N$ を減らした $1 - x/N$ の計算資源の割当てを試みる。 $1 - x/N$ の計算資源に対する各利用者の入札を確認する。留保価格以上の入札が1つだけ行われていた場合、その入札を行った利用者に $1 - x/N - 1/N$ の計算資源を割当てる。留保価格以上の入札が複数行われていた場合、それらの中で最大の入札を行った利用者に $1 - x/N - 1/N$ の計算資源を割当てる。このように計算資源を $1/N$ ずつ減らしていき、留保価格以上の入札が1つでも行われていた場合、その計算資源の量に対して最大の入札を行った利用者に計算資源を割当てる。計算資源の割当てが行われた場合、計算資源

割合	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
留保価格	0.7	1.4	2.1	2.8	3.5	4.2	4.9	5.6	6.3	7.0
ユーザA	0.7	1.4	2.1	2.8	3.5	0	0	0	0	0
ユーザB	0.8	1.6	2.4	3.2	4.0	4.8	0	0	0	0
ユーザC	0.9	1.8	2.7	3.6	4.5	5.4	6.3	0	0	0

図 2 提案手法一例

に残りがあればその割当てを試み、残りがなければ終了する。最後まで入札を確認した後、割当てが行われていない計算資源は未使用とする。

本方式では留保価格を上回る入札を行った利用者には、最低の単位の計算資源を割当ててことを保証する。今回は簡単のため 100% の計算資源を 10% ずつ区切っており、その場合は 10% を保証することとなる。10% ごとに区切ることで、各利用者に 10% を保証することとなり、並列で利用できる利用者の最大数が 10 名と制限されてしまうが、これを 1% ずつ区切ることで 100 名、また、小数点以下まで設定することでそれ以上の利用者が同時に利用することが可能である。利用者は落札した際の入札額を、一定時間ごとに確認する利用コンテナ数分だけ利用料金として支払う。

本手法の例として、留保価格と各利用者の入札の様子を図 2 に示す。図 2 の例において、最小の単位である 10% に対する入札では全ての入札が留保価格を超えているため、全ての利用者に 10% の計算資源を割当てて。残りの 70% の割当てを試みる。70 + 10% に対する入札において留保価格を上回る入札は存在しないため、次に 60% の割当てを試みる。60 + 10% に対する入札においてユーザ C の入札が留保価格を上回る。このとき留保価格を上回る入札を行ったのはユーザ C のみであるため、利用者 C に 60% の計算資源が割当てられる。続いて 10% の計算資源の割当てを試みる。10 + 10% の入札において留保価格を上回る利用者はユーザ C を除いた 2 名となる。2 名の入札を確認すると、ユーザ B の入札がユーザ A のそれを上回るため、10% の計算資源をユーザ B に割当てて。その結果、ユーザ A に 10%、ユーザ B に 20%、ユーザ C に 70% が割当てられる。

4. 評価実験

本章では、提案手法を用いた複数人への計算資源の割当てにより、平行して実行されるジョブの計算時間への影響やデータセンタ管理者が得る収益の増減を確認するための評価実験について述べる。

4.1 評価実験概要

本提案手法によって DP スケジューラの課題である管理者の収益の低下を防ぐことができるかを検証し、商用データセンタでの運用が可能かどうかの評価を行うため実験を行った。本提案手法の比較対象として、DP スケジュー

ラと、DP スケジューラにおいて他の利用者が実行していない状況では支出率を最低額に変更するスケジューラ (DP-down スケジューラと呼ぶ)、の 2 種のスケジューラを用いて同様の実験を行った。本評価実験ではデータセンタに見立てた、3 台の仮想マシンから構成されるクラスタにおいて実施した。実行するジョブとして、Hadoop に標準で実装されている円周率計算 (pi) とワードカウント (wc) のサンプルプログラム、そして Web ページの重要度を決定するページランクプログラム [12](pr) を用いた。ワードカウントおよびページランクにおいて利用するデータは、intel によって開発されたビッグデータベンチマークスイートである HiBench [13] により生成している。本実験では最大の利用者数を 3 名とした。各利用者は同一の処理を 5 回ずつ実行し、それぞれ終了後に 5 分の間隔を設けている。つまり全体の利用者数は時間によって変動し、割当てられる計算量が変動することとなる。また、2 分間隔で使用しているコンテナ数分の支出率を支払う。

評価実験の環境を表 1、実験内容について表 2 に示す。表 1 にはそれぞれのマシンの OS と、メモリ容量、仮想コア数、Hadoop のバージョンを順に表している。表 2 には 1~4 までの実験における各利用者の実行するジョブと支出率を記載している。実験 No. 1 において、user 1, user 2, user 3 の全ての利用者は pi を実行した。同様に実験 No. 2 では wc を、実験 No. 3 では pr を実行している。実験 No. 4 では user 1, user 2, user 3 はそれぞれ pi, wc, pr を実行した。表 3 はそれぞれの利用者による計算資源の各割合に対する入札額と留保価格を示す。ユーザ 1, 2, 3 は最大で 40%, 60%, 70% をそれぞれ希望している。

表 1 クラスタ構成

OS	Memory[GB]	VCore	Hadoop ver.
CentOS 7.3.1611	8	8	2.7.1
CentOS 7.4.1708	8	8	2.7.1
MacOS X 10.9.5	8	8	2.7.1

表 2 各実験における利用者の実行ジョブと支出率 (S. Rate)

実験 No.	user 1		user 2		user 3	
	Job	S. Rate	Job	S. Rate	Job	S. Rate
1	pi	2	pi	3	pi	5
2	wc	2	wc	3	wc	5
3	pr	2	pr	3	pr	5
4	pi	2	wc	3	pr	5

表 3 入札額及び留保価格

割合	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
留保価格	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	10.0
user 1	1.0	2.0	3.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0
user 2	1.0	2.0	3.0	4.0	5.0	6.0	0.0	0.0	0.0	0.0
user 3	1.0	2.0	3.0	4.0	5.0	6.0	7.0	0.0	0.0	0.0

4.2 評価実験結果

DP スケジューラ, DP-down スケジューラ, 提案手法によるそれぞれの実験結果を図3, 図4, 図5に示す. 各図における折れ線グラフは利用者ごとのジョブ実行時間を, 棒グラフは支払った利用料金を表している. また, 図3, 図4, 図5において実験 No. は表2のものに対応している.

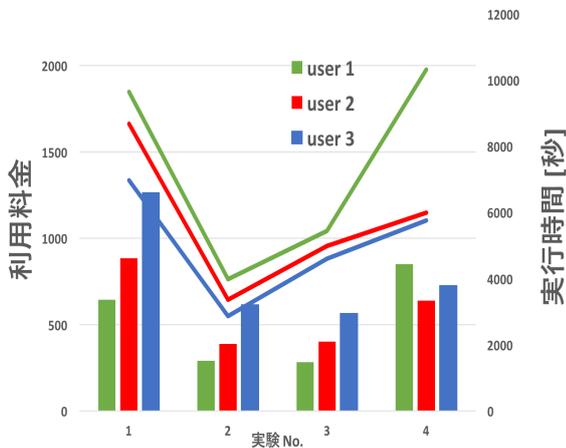


図3 DP スケジューラ実験結果

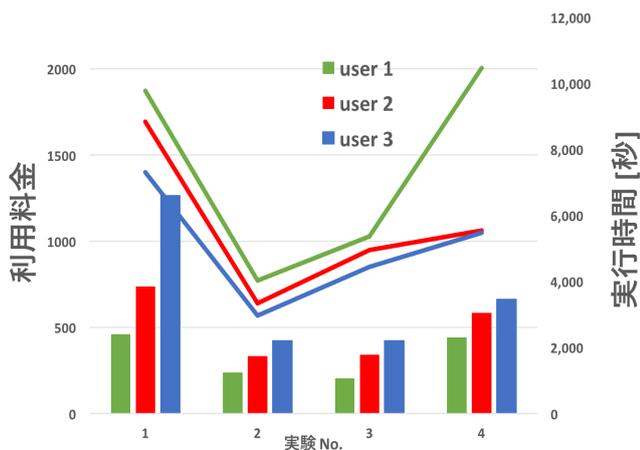


図4 DP-down スケジューラ実験結果

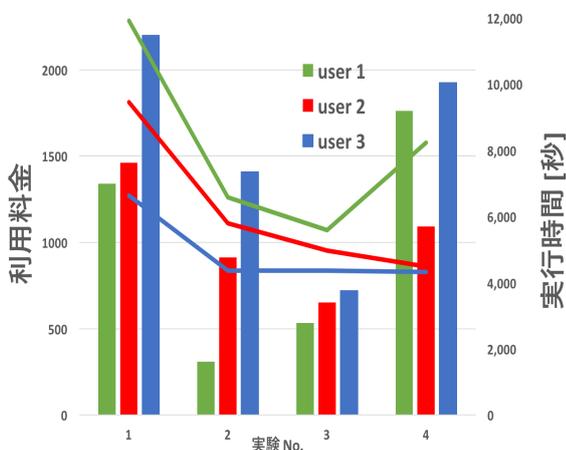


図5 提案手法実験結果

まず DP スケジューラと DP-down スケジューラの実験結果について述べる. 図3と図4において, 各実験の実行時間はほぼ同値となった. 実行時間の違いは最大では DP-down における実験 No. 3 において約7分増加したものであり, それ以外の結果ではおおよそ数秒から5分の間となっている. DP スケジューラと比較し, DP-down スケジューラにおける利用料金は全ての実験, 利用者において減少している. 各実験において3名の利用者の利用料金の合計は, 実験 No. 1 において約11%, 実験 No. 2 において約23%, 実験 No. 3 において約22%, 実験 No. 4 において約24%それぞれ減少しており, 全ての実験においてデータセンタ管理者の得る収益は減少した. 利用者ごとに利用料金を見ると, 実験 No. 4 における user 1 の利用料金の減少が著しい. 実行時間は他の2名の利用者の2倍近いにも関わらず, DP-down スケジューラにおいて約48%利用料金が DP スケジューラから減少した.

次に提案手法の実験結果である図5について述べる. 実行時間については, DP スケジューラおよび DP-down スケジューラと同様に, 希望する計算資源の量が多く, より大きな利用料金を申告した利用者の実行時間が全ての実験において小さくなっていることが確認できた. 提案手法において, 他の2種のスケジューラとは割当てる計算資源の割合が異なるため, それぞれの利用者の実行時間に差ができています. また, 提案手法では他の利用者が存在なくなり計算資源が余っている場合でも, 入札した値より多くの計算資源を利用することができないという特徴も実行時間の変動に影響している. 提案手法における利用料金について DP-down スケジューラの結果と比較すると, 実験 No. 1 から実験 No. 4 まで全ての実験において3名全ての利用者の利用料金が增加了. 利用料金の増加は, 最小で実験 No. 2 における user 1 の29%, 最大で実験 No. 4 における user 1 の297%という結果となった.

4.3 評価実験考察

DP スケジューラと DP-down スケジューラを比較することで, DP スケジューラの課題としていた収益の減少を確認できた. DP-down スケジューラで見られた利用料金の減少は絶え間なく利用者がジョブを実行していれば起こることはないが, 実際のデータセンタにおいて全利用者が同一のジョブを同じタイミングで実行するとは考えにくく, どうしても利用者が減少する状況は発生すると考えられる. このことから DP スケジューラをデータセンタで使用するためには収益を確保することのできる料金設定手法の考察が必要であると明らかとなった.

実験 No. 4 に着目すると, 実行時間の差はほぼ無いにも関わらず, DP スケジューラでは他の利用者に対して最大だった user 1 の利用料金が DP-down スケジューラでは最小になった. このとき user 1 は他の2つのジョブと比べて

実行時間の長い円周率計算のジョブを行っている。このことから、計算量の大きいジョブは申告する料金を小さくすることが利用者にとって最適となり、たとえ優先度の高いジョブであったとしても評価通りの料金を申告しないということが起こり得る。提案手法ではこの点がある程度解決されており、計算量の大きいジョブを長時間実行することで利用料金が增大していることが確認できた。

5. 結論と今後の課題

本稿では複数人の同時利用が可能な Hadoop データセンタの構築を目指し、オークションメカニズムと留保価格を導入した手法の提案を検討した。評価実験では、提案手法によって利用者の入札の大小関係に合わせた実行時間となるように計算資源の割当てを行うことと、全体的に利用料金の増加を確認できた。今後の課題として、落札した計算資源に対する利用料金を入札額より安価にすることで利用者の参加を促すオークションメカニズムの考案や、留保価格の適切な設定が挙げられる。その際には利用者だけでなく、データセンタ管理者の収益やデータセンタの利用率も考慮し、全体の利得を増大させる必要がある。また、本提案手法においては落札した際の支払金額は本人の落札金額に依存するが、そのようなオークションは正直なオークション [14] ではない。正直なオークションとは、利用者が財に対して本人が考える価値通りに正直に入札することが最適となるメカニズムを導入したオークションである。収益を増大させるためには、正直なオークションメカニズムの設計が必要であり、今後の課題である。

謝辞

本研究の一部は、科研費基盤 (B) 15H04008 および NAIST Bigdata Project による支援を受けて実施している。

参考文献

- [1] Hadoop. The Apache Hadoop project available from <http://hadoop.apache.org/index.html> (accessed 2018-05-14).
- [2] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," *Communications of the ACM*, Volume 51, Issue 1, pp.107-113, 2008.
- [3] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, B. Saha, C. Curino, O. O' Malley, S. Radia, B. Reed, and E. Baldeschwieler. "Apache Hadoop YARN: Yet Another Resource Negotiator," *SOCC '13 Proceedings of the 4th annual Symposium on Cloud Computing Article No. 5*, 2013.
- [4] T. Sandholm and K. Lai, "Dynamic proportional share scheduling in Hadoop," *JSSPP 10 Proceedings of the 15th international conference on Job scheduling strategies for parallel processing*, pp.110-131, 2010.
- [5] W. Shi, L. Zhang, C. Wu, Z. Li and F. C. M. Lau, "An Online Auction Framework for Dynamic Resource Provisioning in Cloud Computing," *SIGMETRICS '14 The 2014 ACM international conference on Measurement and modeling of computer systems*, pp.71-83, 2014.
- [6] Z. Zhou, F. Liu and Z. Li, "When Smart Grid Meets Geo-distributed Cloud: An Auction Approach to Datacenter Demand Response," *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp.2650-2658, 2015.
- [7] Amazon Elastic Compute Cloud. available from <https://aws.amazon.com/ec2/> (accessed 2018-05-14).
- [8] Amazon EC2 Spot Instances. available from <https://aws.amazon.com/ec2/spot/> (accessed 2018-05-14).
- [9] M. Yokoo, Y. Sakurai and S. Matsubara, "Robust Combinatorial Auction Protocol against False-name Bids," *Artificial Intelligence*, Volume 130, Issue 2, pp.110-116, 2001.
- [10] H. R. Varian, "Economic Mechanism Design for Computerized Agents," *Proceedings of the First USENIX Workshop on Electronic Commerce*, 1995.
- [11] W. Vickrey, "Counterspeculation, auctions and competitive sealed tenders," *Journal of Finance*, Volume 16, Issue 1, pp.8-37, 1961.
- [12] P. Lawrence, B. Sergey, M. Rajeev and W. Terry, "The PageRank Citation Ranking: Bringing Order to the Web," 1999. available from <http://ilpubs.stanford.edu:8090/422/> (accessed 2018-05-14).
- [13] HiBench. available from <https://github.com/intel-hadoop/HiBench> (accessed 2018-05-14).
- [14] K. Deshmukh, A. V. Goldberg, J. D. Hartline and A. R. Karlin, "Truthful and Competitive Double Auctions," *Proceedings of the 10th Annual European Symposium on Algorithms*, pp.361-373, 2002.