5U-01

データセンタにおける制限時刻を考慮したCoflowスケジューラの提案

中村 三郎†　　相田 仁‡

東京大学大学院工学系研究科

1. はじめに

国際的にデジタルデータの量は飛躍的に増大しており，米国の調査機関によると2021年には約60兆GBにも達すると予想されている．大量のデータを単一のコンピュータで処理することは難しく，複数のマシンで分散して処理を行っている．分散処理では処理能力を足し算的に増やすことができる一方，ノード間でデータを送受信する必要があり，通信による遅延が全体のパフォーマンスを下げる要因となっている．実際，近年の研究では分散処理を行う際に生じる中間データの通信遅延が全体の完結時間の50%以上を占めることが明らかになっている[2].

分散処理の完結時間を決定するのは処理で用いられるフローのうち最も遅く完結するフローであり，すべてのフローの完結時間が重要なわけではない．したがってフローレベルで効率化を行う必要はなく，複数のフローをグループ化したCoflowという単位をスケジューリングして性能を上げる研究が行われている．Coflowスケジューラの性能評価は主に平均Coflow完結時間(平均CCT)と制限時刻満足率の二つで行われる．

本研究では平均CCTと制限時刻満足率の二つの向上が可能なVarysをベースとしつつ，制限時刻満足率をさらに改善する手法を提案し，フローレベルシミュレータを用いて評価を行った

2. Coflowスケジューラ Varys

Coflowとは2012年に考案された概念であり，分散処理での通信においてアプリケーションレベルで意味を持つ複数のフローのグループのことである[1]．分散処理全体の性能は一つ一つのフローで直接決まるわけではなく，それぞれの処理で用いるフローの中で最も遅いものや最も先のものによって決定される．したがってCoflowとしてグループ化し，Coflow内とCoflow間のスケジューリングとして分けることで，複雑さを軽減するとともにアプリケーションレベルで意味のあるスケジューリングが可能となる．

Coflowスケジューラの評価は主に平均CCTと制限時刻満足率の２つの観点から行われる．Coflow考案後は平均CCTの向上を目指す研究が主流であった．これは制限時刻が全てに設定されているわけではないなどの理由から平均CCTの向上が優先されたと考えられる．その中でも平均CCTだけでなく制限時刻満足率の向上もできるCoflowスケジューラとしてVarysがある[2].

本研究では主に制限時刻満足率の向上に注目しているため，制限時刻満足率の向上ができるVarysをベースとしつつ更なる性能向上を目指した．

Varysは2014年に発表されたCoflowスケジューラである[3]．平均CCTの向上か制限時刻満足率の向上を目的としてスケジューリングを行うことができる．

平均CCTの向上はSmallest-Effective-Bottleneck-First(SEBF)と呼ばれるヒューリスティックな優先度決定アルゴリズムとMinimum-Allocation-for-Desired-Duration(MADD)と呼ばれる割り当てアルゴリズムの二つが主な要因となっている．SEBFとはShortest-Coflow-Firstという最も短いCoflowを先に処理するアルゴリズムを発展させたものである．ネットワーク状況を考慮することでCoflowの終了時刻を予測して優先度を決定している．一方MADDとはその名の通り，必要最低限の割り当てだけを行うということである．最遅のフロー以外はどんなに早く終了しても意味がないため，最遅フローと同じ時間に終わるような帯域しか割り当てないということである．

制限時刻満足率の向上は主にアドミッション制御によって実現している．SEBFと同様に終了時刻を予想することで制限時刻に間に合うかどうかを判断し，間に合わなければ処理を一切行わない．これによりリソースを他のCoflowに集中させて制限時刻満足率の向上を果たしている．リソースの割り当ては平均CCTモードと同じMADDを用いている．

このようにVarysは平均CCTの向上と制限時刻満足率の向上のどちらかを実現している．しかし，後者は簡単なアドミッション制御のみしかしておらず，優先度の決定などでリソースを最適に使うようには設計されていない．

3. 提案手法

本研究ではVarysをベースとしつつ三つの変更を加えたスケジューラを提案する．

一つ目は処理中のCoflowを途中で停止させることでリソースを解放させるようにするということである．これは制限時刻に間に合わないCoflowを止めることでリソースの有効活用を目的としている．停止のタイミングはいくつか考えられるが，今回制限時刻を超えた段階での停止としている．ただし，他のCoflowが存在しない場合には停止させない．

二つ目は最初に割り当てた帯域を終了まで堅持するようにするということである．従来のVarysではCoflow到着時に割り当てた帯域を無制限に与え続けるのではなく，一定時間が経過した場合に割り当ての量を減らすようになっている．これは一つのCoflowにリソースが集中してしまうような事態を防いでいるが，制限時刻に間に合うかどうかの予測の精度を落とすことになるため，最初に割り当てた帯域を終了まで保証するようにした．

Proposal of a Coflow Scheduler Considering Deadline in Data Center
† Saburo Nakamura
‡ Hitoshi Aida
School of Engineering, The University of Tokyo

三つ目はアドミッション制御の改善である．従来は Coflow 到着時に空き帯域を全て利用できるとして制限時刻に間に合うかどうかで判断していた．しかし，使用されている帯域であっても Coflow の終了で将来的に使用可能になる可能性がある．そこで，制限時刻までに空く帯域も考慮して制限時刻に間に合うかどうか判断するよう変更した．

4. 評価

4.1 評価方法

本研究ではフローレベルシミュレータの CoflowSim[4] を用いて評価した．用いたデータセットは Facebook で 2010 年ごろに実行された Hive/MapReduce のログをもとに作られたベンチマークである[4]．3000 台のマシンが 150 のラックに入っており 10:1 のオーバーサブスクリプションである．合計の帯域は実際には 300Gbps であるが 100Gbps にスケールダウンしている．Coflow は 526 個であり一つのラックの中だけで完結しないように調整されている．また制限時刻については全ての帯域を利用できると仮定した際の予想最小 CCT に Uniform(1,2)を掛けたものを設定している．なお制限時刻の設定はすべての手法で同じである．

4.2 シミュレーション結果

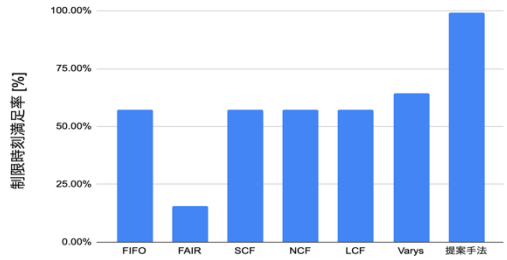

図 1．各手法での制限時刻満足率の比較

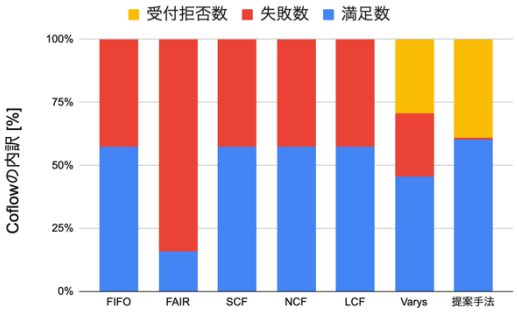

図 2．各手法での Coflow の内訳の比較

図 1 は各手法での制限時刻満足率を示したものである．なお SCF, NCF, LCF は Shortest / Narrowest / Lightest Coflow First を表している．なお， Vary，提案手法では Coflow の到着時に拒否したものは割合の計算に入れていない．提案手法では一度許可したものを途中で停止させているが，これは失敗したものとして割合の計算に含んでいる．

図 1 より提案手法では制限時刻満足率が大幅に改善しており，100%近くになっていることが分かる．Varys は他の手法よりも一段高い．

図 2 は Coflow の成功数，失敗数，受付許可数の内訳の比較である．なお提案手法の失敗は全て途中で停止させたものである．図 1 と比較すると Varys の成功数が LCF などの他の手法よりも低くなっていることがわかる．提案手法ではこの点が改善され，絶対数でも他の手法より高くなっていることがわかる．

図 3 は制限時刻を満足した Coflow の通信量の合計を各手法で比較したものである．Coflow のサイズはバラバラであるため，サイズの小さいものを優先すれば制限時刻満足率は高くなる．今回の結果から，FAIR はその逆で，

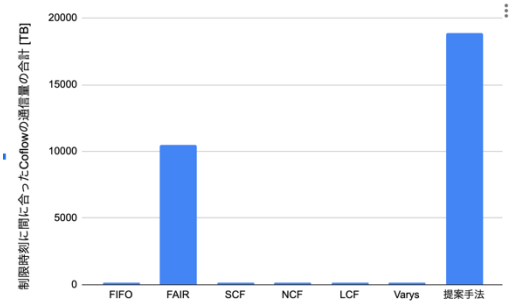

図 3．制限時刻に間に合った Coflow の通信量の合計の比較

満足率は低いもののサイズの大きい Coflow を成功させていることが分かる．また，提案手法はデータ量の面からも優れていることが分かる．全体のリソース量は変わらないことから，提案手法ではリソースを効率よく集中させて成功させているのに対し，他手法では失敗する Coflow にリソースを無駄に使っていることが分かる．

5. おわりに

本研究では Coflow スケジューラ Varys をベースとしつつ制限時刻満足率を改善させる新手法の提案を行い，フローレベルシミュレータで評価した．提案手法では制限時刻を満たす Coflow の割合と数の両方で優れた結果を示すことが分かった．また通信量の比較でも提案手法が優れており，リソースの選択と集中が効率よく行われていることが確認できた．

今後の課題としては，まず Coflow を途中で停止させるタイミングの改善があげられる．間に合わないと予測できた段階で停止させることで，リソースをより効率的に使うことができる．また，シミュレーション評価も単純なものだけであるため，他のデータセットの使用や，特殊な状況での性能評価をしていきたい．

参考文献

[1] M. Chowdhury and I. Stoica, “Coflow: a networking abstraction for cluster applications.” in HotNets, pp.31–36(2012).
[2] S. Wang, J. Zhang, T. Huang, J. Liu, T. Pan, and Y. Liu, “A survey of coflow scheduling schemes for data center networks,” IEEE Communications Magazine, vol. 56, no. 6, pp. 179–185, (2018).
[3] M. Chowdhury, Y. Zhong, and I. Stoica, “Efficient coflow scheduling with varys,” in ACM SIGCOMM Computer Communication Review, vol. 44, no. 4. ACM, pp.443–454(2014).
[4] M. Chowdhury, “Flow-level simulator for coflow scheduling used in varys,” https://github.com/coflow/coflowsim.