汎用エッジコンピューティングのためのオンラインオーケストレーションメカニズム

代表研究者　Shao Xun　北見工業大学 情報通信系 助教
共同研究者　長谷川 剛　東北大学 電気通信研究所 教授

1 はじめに

1-1 モチベーション

近年、モバイル機器、通信、クラウドコンピューティングの急速な発展により、IoT(Internet of Thing)のサービスやアプリケーションが急増している。そのため、モバイル通信事業者にとっては、インターネットを介したモバイルトラフィックが増大し、大きな課題となっている。しかし、モバイル端末とクラウドデータセンターの距離が遠いため、モバイルユーザーは大きな遅延に悩まされ、満足なサービスを受けられないという問題があります。このような問題を解決するために、インターネットのエッジにサーバー資源を配置し、エッジサーバーでモバイルサービスを提供する方法が有望視されている。エッジコンピューティングにより、モバイルコンテンツプロバイダー、サービスプロバイダー、モバイルデバイスは、空間的に近い計算資源とストレージ資源を利用することができ、モバイルコアネットワークの混雑を緩和することができる。

エッジコンピューティングの最も興味深い用途の 1 つは、データのキャッシュです。ECP (Edge Cloud Provider) は、データがエッジクラウドのネットワークを通過する際に、人気のあるデータをキャッシュし、インターネットにリクエストを転送するのではなく、キャッシュされたデータでデータリクエストを処理することができる。タスクのオフロードも、エッジクラウドで盛んに研究されている分野である。モバイル端末では、ビデオ解析などの計算負荷の高いタスクをエッジサーバーにオフロードすることで、バッテリーを節約し、処理を高速化することができる。最近では，エッジコンピューティングを利用したオンラインゲーム、仮想現実、拡張現実など、より高度で複雑なエッジサービスが研究・開発されており [1] [2] [3] 、エッジコンピューティングが新しいアプリケーションを開発する大きな可能性を持っていることが明らかになっている。しかし、現在のところ、エッジコンピューティングの研究のほとんどは、サービスやアプリケーションに特化したものであり、汎用的なエッジコンピューティングに関する議論はほとんど行われていない。

1-2 私たちのビジョン

本研究では、アプリケーションに特化したエッジコンピューティングサービスではなく、汎用的なエッジコンピューティングシステムについて考察する。そして、そのビジネスモデルを分析し、主な問題点を明らかにし、解決策を提供する。図 1 に汎用的なエッジコンピューティングのイメージを示す。ECP は高速ネットワークで相互接続された複数の分散エッジデータ センタを運営している。近年、モバイルネットワークの進化に伴い、C- RAN の採用が大きなトレンドとなっている。本研究では、エッジデータセンターが C-RAN に配置されるケースを想定する。ECP は、分散エッジクラウドネットワークにより、分散クラウドプロバイダのように、モバイルサービスプロバイダやユーザに汎用的なエッジコンピューティングサービスを提供する。具体的には、ECP は、特定のモバイルサービス（サービス ID で識別）に関連付けられたモバイルデバイスがコンピューティングタスクを実行するための仮想マシン（VM）またはコンテナを備えた分離された環境を提供する。汎用的なエッジコンピューティングフレームワークは、高度にカスタマイズされたリクエストをサポートすることが想定されている。図 1 にその一例を示す。このリクエストでは、モバイルデバイスは、データ o1 を処理するためにタイプ 1 の 2 つの VM と、データ o1 および o4 を処理するためにタイプ 2 の 1 つの VM を必要とする。o1 と o4 は既にインターネット上に存在し、o4 はモバイルデバイスによってアップロードされることに注目されたい。このような要求を受け取った ECP は、要求を満たすためにどのように VM を提供するかを決定する必要がある。図 1 の例では、ECP は C-RAN 1 に 2 つのタイプ 1 の VM を、C-RAN 2 に 1 つのタイプ 2 の VM を組み立てている。o1 が既に C-RAN 1 に配置されているので、タイプ-1 VM はローカルキャッシュからデータをフェッチすることができるが、o1 と o4 が C-RAN 2 に配置されていないので、タイプ-2 VM は C-RAN 1 と 2 の間のリンクを介してデータ o1 と o4 をフェッチしなければならないことに注目されたい。このプロセスにより、ECP はクラウドプロバイダーのように VM を提供することで収益を得る。

このプロセスには、ECP、エッジサービスプロバイダー、モバイルユーザーの 3 種類のビジネスエンティティが関与し、それらの間で柔軟なビジネス関係を持つことが可能である。例えば、モバイルユーザはエッジサービスプロバイダーからモバイルサービスを申し込むことができ、エッジサービスプロバイダーは ECP と契約を結び、ECP がサービスに関連するリクエストを受け取ると、モバイルデバイスに VM を割り当て、エッジサービスプロバイダーに課金する。図 1 は、金銭の流れを示している。ECP とエッジサービスプロバイダーは、論理的かつ概念的なものであることに注意してください。実際には、モバイルネットワーク事業者が C-RAN のデータセンターを運営しているため、モバイルネットワーク事業者が ECP として機能する可能性があります。また、ECP は、モバイルユーザーに独自のサービスを提供するエッジサービスプロバイダーである可能性もある。

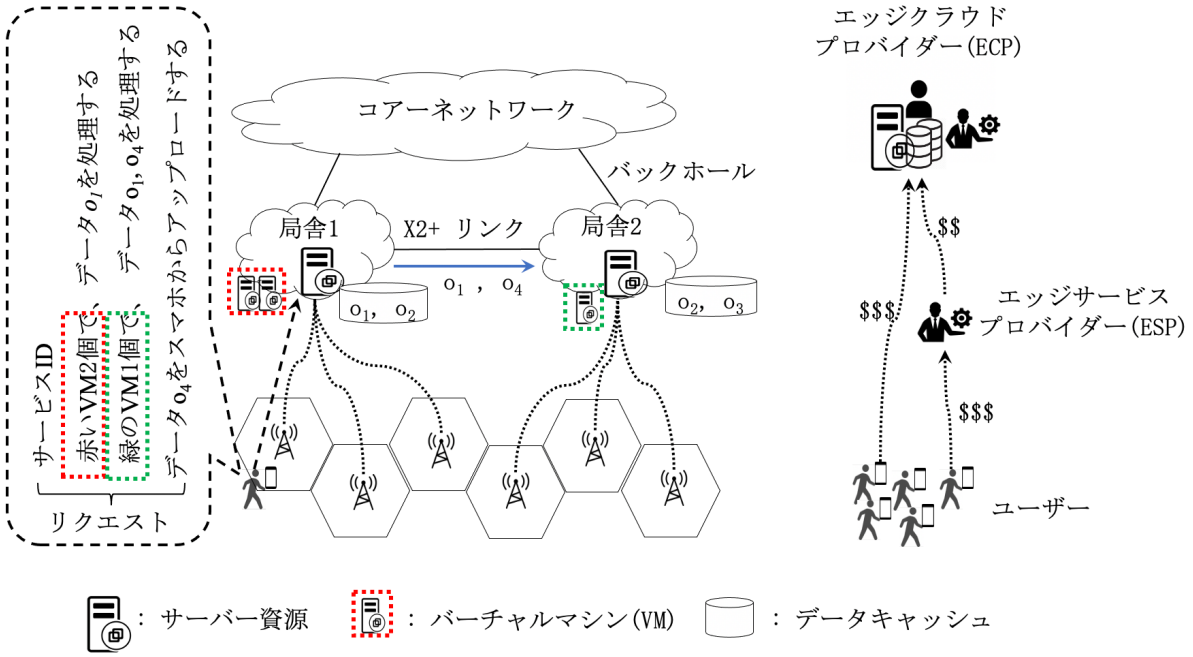

図 1: 汎用エッジコンピューティングのビジョン

1-3 オーケストレーションにおける挑戦的な課題

ECP の観点からは、VM の割り当てと公開データの配置に関して、複雑な共同最適化問題が存在する。リソース割り当て問題では、分散したエッジデータセンターの物理リソースが異種であるため、将来のシステム状態に関する統計的知識なしにリソース割り当ての決定を最適化することは困難である。長期的な収益を最大化するためには、各エッジデータセンターの各リソースの可用性を維持することが重要である。近接性だけを考慮してリソースを割り当てると、特定の要求シーケンスに直面する特定のエッジデータセンターのリソースが枯渇してしまう。また、収益だけでなく、クラウド間のデータ転送で発生するコストも無視することはできない。人気のあるデータを分散キャッシュに配置することで、クラウド間のデータ転送を減らすことができるが、リソース割り当てとデータ配置の決定には複雑な相互作用がある。たとえば、図 1 では、タイプ 2 の VM を C-RAN 1 で組み立てることで、o1 と o4 を取得するためのクラウド間データ転送コストを削減できるが、C-RAN 1 のリソース利用可能性の問題が発生する可能性もある。C-RAN 2 でタイプ 2 の VM を組み立てる場合、C-RAN 2 で o1 を先にキャッシュすると、クラウド間のデータ転送コストを節約できるが、キャッシュデバイスのサイズ制限のため、他のデータが放出されなければならない場合がある。将来の要求シーケンスを知ることなく、そのようなオンライン決定を行うためのアルゴリズムを設計することは困難である。たとえ将来の要求シーケンスがよく予測できたとしても、マルコフ決定過程のようなよく知られたオンライン学習アルゴリズムは、巨大な状態空間が与えられると、妥当な期間で結果を計算することができない。本稿では、資源割り当てとデータ配置相互作用を包括的に分析することにより、将来の要求シーケンスに関する事前知識を必要とせず、共同最適化問題を効率的に解くことができるオンライン機構を提案する。

2 システムモデルと問題策定

2-1 設定項目

ECP は、複数のエッジクラウドを分散して運用しつつ、高速ネットワークで相互接続することを想定している。各エッジクラウドのインデックスは i（j の場合もある）である。エッジクラウド i と j の間のレイテンシは $w_{i,j}$ と表記される。各エッジクラウドは、モバイルユーザーに特定のタイプの VM を組み立てるために使用できる複数種類の物理リソース（CPU、メモリ、ストレージなど）を持っている。また、複数種類のリソースを想定し、特定のリソースに r のインデックスを付ける。1 タイプ-k の VM を組み立てるためのリソース量 r を $g_{k,r}$ と表記する。ECP は、K 種類の VM を提供し、各 VM には k のインデックスが付けられる。ECP は、単位時間あたり k 種類の VM を提供することにより、p_k の収益を得る。エッジクラウドには、コンピューティングのためのリソースに加えて、クラウド間のデータ転送コストを削減し、モバイルユーザにより良いサービスを提供するために、人気のあるデータをローカルに配置できる低コストのキャッシングデバイスも存在する。データ o のサイズを s_o、エッジクラウド i のキャッシュのサイズを S_i と表記する。ECP は、運用コストを削減するためにキャッシュを補完的に運用し、モバイルサービスプロバイダとユーザには透過的である。リクエストには、サービス識別子、時間長、処理するデータリストに関連する VM のセット、アップロードするデータリストが含まれる。サービス識別子は、ECP が課金のために関連するモバイルサービスプロバイダを識別するためのものである。時間長は、VM の使用期間を指定するために使用される。要求されるデータは、インターネット上の既存のデータ（パブリックデータ）、または要求を開始したモバイルユーザからアップロードされたデータ（プライベートデータ）のいずれかである可能性がある。例えば、図 1 では、要求されたデータ o1 はパブリックデータ（すなわち、インターネット上で見つけることができる）であり、o4 はモバイルデバイスからアップロードされたデータである。タイプ k の VM に関連するリクエスト l のデータ集合を $O_{l,k}$ と呼ぶ。あるリクエスト l に対して、そのリクエストを満たすための様々な資源割り当て戦略が存在する。A^l を特定の実現可能な戦略として、バイナリ決定変数 $x_A^l \in \{0,1\}$ をそれに関連付ける。具体的には、x_A^l は A^l のとき 1、それ以外では 0 になる。一つの要求に対して、それを満たすために複数の資源配分戦略を採用する必要はない。形式的には、x_A^l に対して以下の制約がある。

$$\sum_{A \in A^l} x_A^l \leq 1, \quad x_A^l \in \{0,1\} \quad \forall l, A. \tag{1}$$

2-2 タイムスケール

この論文では、T で示される粗視化されたタイムスロットと t で示される細粒化されたタイムスロットを採用する。粗視化されたタイムスロットでは、時間平均の収益、時間平均のデータ伝送コスト、データの人気度など、長期的な特性を把握する。これらのリアルタイム性が求められない意思決定を、粗視化タイムスロットで行う。また、これらのタスクの他に、リアルタイム性が求められるタスクもある。例えば、リクエストが到着したとき、ECP はそのリクエストをあるタイムスロットの終了までバッファリングするのではなく、即座にリソースを割り当てなければならない。資源割り当てについては、各エッジクラウドにある各種資源の量に左右されるため、このような制約ごとに、特定資源のシャドープライスを評価するデュアル変数を導入する。具体的には、原始問題から双対問題を構成する。新しいリクエス l が到着すると、原始問題には新しい決定変数 x_A^l が導入され、双対問題には新しい制約が導入される。ある資源が割り当てられたとき、その資源に関連する双対変数（シャドープライス）を指数関数的に増加させ、1) その資源の希少性を反映し、2) 原始値の増分と双対値の増分の比を一定に保ち、どのリクエスト到着順序に対しても常に解が競合するようにする。しかし、要求プロセスが終了すると、割り当てられた資源が解放され、利用可能な資源量が増加するため、粗視化された各タイムスロットをさらに複数の細粒度のタイムスロットに離散化する。細粒度のタイムスロット t では、t の間に到着したリクエストのみを考慮し、終了したリクエストは無視する。t の終わりに、t の間に解放されたすべての資源を考慮して利用可能な資源量を更新し、t + 1 について新しい原始問題と双対問題のペアを開始する。これに対応して、利用可能資源量は t で添字され、各 t について、以下の制約を満たす必要がある。

$$\sum_{l:t \in t_l} \sum_{A \in A^l} N_{A,i,k}^l g_{r,k} x_A^l \leq c_{i,r,t} \quad \forall i, r, t, \tag{2}$$

ここで、t_l は l のリクエストに含まれる細粒度タイムスロットの集合、$g_{r,k}$ は各タイプ k VM を組み立てるために必要なリソース r の量、$N^l_{A,i,k}$ は割り当て戦略 A を採用した場合にエッジクラウド i に割り当てられたタイプ k VM の数である。この方法は、理論的最適解を近似するのに役立つ。本研究では、リクエストは任意の時間に到着し、我々の方法はタイムスロットの終わりまでリクエストをバッファリングするのではなく、即座に応答する。

2-3 収益

この研究では、特定の期間の収益ではなく、長期的な ECP 収益の最大化を目指す。長期的な収益は、時間平均の定式化で示す。具体的には、粗い粒度のタイムスロット T から得られる収益を R（T ）とすると、以下のようになる：

$$R(T) = \sum_{l:T\le\tau^l<T+1} L^l \sum_{A\in A^l}\sum_k p_k \sum_i N^l_{A,i,k} x^l_A, \tag{3}$$

タイムスロットごとに得られる収益を考えると、ECP の時間平均収益を次のように定義することができる：

$$\overline{R(T)} = \lim_{\mathcal{T}\to\infty} (1/\mathcal{T}) \sum_{T=0}^{\mathcal{T}-1} R(T). \tag{4}$$

2-4 データ伝送コスト

クラウド間のデータ輸送は、遅延による QoS（Quality of Service）の低下を避けられず、さらに悪いことに、ECP が大量なデータを輸送するために限られたクラウド間リンクを使用することはコスト高となる。この論文では、クラウド間のデータ転送コストは、次の 2 つの理由で発生する。1）ECP がモバイルユーザーの VM を、モバイルデバイスが関連するエッジクラウドにプロビジョニングしない場合、ECP はユーザーのアップロードデータを、VM がプロビジョニングされる他のエッジクラウドにクラウド間リンクで輸送しなければならない。2）あるユーザーの VM がローカルキャッシュで処理すべきデータを見つけられない場合、VM は隣接するエッジクラウドからデータを取り出そうとし、その際にクラウド間データ伝送コストが発生する。また、エッジクラウド i にキャッシュされたデータをS_i、各タイプ k の VM に関連する l が必要とするデータセットを$O_{l,k}$ と表記する。このようなデータ要件を与え、i が j から o のために取得する量を$f_{i,j,o}$とすると、粗粒度タイムスロット T におけるデータ伝送コストは次のように定式化できる：

$$C(T) = \sum_{l:T\le t^l<T+1}\sum_{A\in A^l}\left(\sum_i N^l_{A,i,k}\sum_{o\in\mathbb{O}_{l,k}-\mathbb{S}_i}\sum_{j:o\in\mathbb{S}_j} w_{i,j} f_{i,j,o}\right) x^l_A. \tag{5}$$

拠点間の単位トラフィックの伝送コストが一定であることを考えると、オブジェクト o の要求は常に最も低い $w_{i,j}$ を持つ j に向けられるので、分数ルーティング $f_{i,j,o}$ は余分なものとなる、つまり(5)は(6)になる：

$$C(T) = \sum_{l:T\le t^l<T+1}\sum_{A\in A^l}\left(\sum_i N^l_{A,i,k}\sum_{o\in\mathbb{O}_{l,k}-\mathbb{S}_i} w_{i,j^*} s_o\right) x^l_A. \tag{6}$$

ここで、

$$j^* = \operatorname{argmin}\{w_{i,j} : o\in\mathbb{S}_j\}.$$

輸送コストは避けられないものであり、あるレベル以下に抑える必要がある。しかし、輸送コスト制約は資源容量制約と異なり、資源容量は絶対に超えてはいけないが、輸送コスト制約については、ある時間帯の違反が許容される。具体的には、時間平均のデータ伝送コストを下式のように定義する：

$$\overline{C(T)} = \lim_{\mathcal{T}\to\infty} (1/\mathcal{T}) \sum_{T=0}^{\mathcal{T}-1} C(T), \tag{7}$$

それに、以下の制約を満たせることにより、

$$\overline{C(T)} \leq \mathcal{C}, \tag{8}$$

オリジナル問題の実行可能領域を拡張し、リクエストのダイナミクスをよりよく活用することができる。

2-5 問題策定

以上のモデルをまとめると、収益最大化問題は次のように定義できる：

$$\begin{aligned} &\max \overline{R(T)} \\ &s.t.\ (1),\ (2), and\ (8). \end{aligned} \tag{9}$$

上記の問題定義には、計算機資源割り当て問題とパブリックデータ配置問題という 2 つのサブ問題があることに注目されたい。この 2 つの問題は、時間的・空間的に異なる特徴を持つ。時間的側面では、リクエストが任意の時間に到着する可能性があるため、ECP は直ちに計算機資源の割り当てを決定しなければならないが、データの人気分布が比較的ゆっくりと変化するため、比較的長い期間を観測してデータ配置を決定しなければならず、頻繁に配置し直してもヒット率の向上には役立たない。空間的な面では、エッジクラウド間の VM マイグレーションが複雑であるため、計算資源配置の決定がなされれば、将来にわたって変更されない。一方、データ配置では、ヒット率を向上させるためにデータを再配置し、データ輸送コストを削減する必要がある。次章では、コンピューティングリソース割り当てとパブリックデータ配置の最適化問題を同時に解決するアプローチを説明する。

3 システムモデルと問題策定

本節では、計算資源割り当て問題とデータ配置問題を同時に解決する我々のアプローチを紹介する。本アプローチは、計算資源配置のための細粒度タイムスロットとパブリックデータ配置のための粗粒度タイムスロットのハイブリッドタイムスケールで動作する。粗視化されたタイムスケールでは、制約(8)が時間平均の定式化を持つことに着目し、目的関数と制約違反のトレードオフを可能にする。このため、次の粗視化タイムスロットで「消費」できる輸送コストの「予算」を表す長さの仮想待ち行列を導入し、ドリフトプラスペナルティ法を用いて長期収益を最適化し、仮想待ち行列を安定化させる。具体的には、粗視化された各タイムスロットの最初に、前のタイムスロットからの輸送コストを蓄積し、仮想待ち行列のバックログを更新し、仮想待ち行列の加重収益と「ドリフト」の和を最適化する計算資源割り当てを決定する。

このドリフトプラスペナルティベースの手法は、将来のシステムの状態について仮定することなく、最適に近い解を提供できるため、近年注目されている手法である。しかし、この方法は通常、バッファリングと決定、つまり、リクエストを一定時間バッファリングし、その後、バッファリングされたリクエストをどのように処理するかを決定する方法で動作する。このため、バニラドリフトプラスペナルティ方式は遅延耐性タスクには有効であるが、リアルタイム問題には対応できない。本研究では、計算機資源割り当てのリアルタイム要求に対応するために、双対問題によるオンラインアルゴリズムを導入することにより、ドリフトプラスペナルティ法の欠点を克服する。双対問題によるアルゴリズムの元形式は、有限時間内にタスクが終了するシナリオには適用できないため、双対問題によるアルゴリズムから得られる結果を近似するために、粗粒のタイムスロットをさらに細粒のタイムスロットに離散化する。

リクエストが到着すると、そのリクエストにリアルタイムで応答する。各粗粒度タイムスロットの終了時に、過去のタイムスロットにおけるデータの人気度変化を反映し、異なるエッジクラウドにデータを再配置する。この方法により、計算機資源割り当ての決定の即時効果を長期的に反映させることができ、長期的なデータ配置の決定は比較的長期にわたって計算機資源割り当てを導くことができる。

3-1 オンライン共同最適化のフレームワーク

実際の輸送コストとあらかじめ設定された上限値との差を表すために、仮想待ち行列を導入することで、待ち行列安定化技術により制約条件（8）を満足させることができる。具体的には、時間スロット T の待ち行列の残りを Q(T) とし、そのダイナミクスを次のように定義する:

$$Q(T+1) = \max\{Q(T) + C(T) - \mathcal{C}, 0\} \tag{10}$$

ここで、$\mathcal{C}$ は予め定められた時間平均輸送コストの上限値である。仮想待ち行列が安定であるということは、制約条件(8)が満たされていることを意味する。このことは、以下のレンマからわかる:

Lemma 1. $\lim_{T\to\infty} Q(T)/T = 0$ *implies* $\overline{C(T)} \leq \mathcal{C}$.

仮想待ち行列を安定させるために、リアプノフ関数を次のように定義する：

$$L(Q(T)) = \frac{1}{2}Q^2(T)$$

そして、1-スロット ドリフトは

$$\Delta_1(Q(T)) = L(Q(T+1)) - L(Q(T))$$

ドリフトプラスペナルティ法[4]に従い、粗視化された各時間スロットにおいて、以下の項の下限を最大化することにより、時間平均収益 R（T）を一定の最適性ギャップ内に収めると同時に仮想待ち行列を安定化させることができる：

$$VR(T) - \Delta_1(Q(T)) \tag{11}$$

ここで、V は収益とデータ伝送コスト制約違反のトレードオフを制御するための予め決められた非負のパラメータである。$C^2(T)$は C^{max2} によって非終端的に上界されると仮定し、上記の項の下界は、仮想待ち行列のダイナミクスから得ることができる。具体的には:

$$\begin{aligned} & L(Q(T+1)) - L(Q(T)) \\ &= \frac{1}{2}\left(Q^2(T+1) - Q^2(T)\right) \\ &= \frac{1}{2}\left(\max\{Q(T) + C(T) - \mathcal{C}, 0\}^2 - Q^2(T)\right) \\ &\leq Q(T)(C(T) - \mathcal{C}) + B, \end{aligned} \tag{12}$$

ここで、B = $\max\{C^{max2}, C^2\}/2$。上記の不等式の両辺に-1 を掛け、VR(T)を足すと、(11)の下界が以下のように得られる：

$$VR(T) - \Delta_1(Q(T)) \geq VR(T) - Q(T)(C(T) - \mathcal{C}) - B. \tag{13}$$

粗視化された各時間スロットにおいて、制約条件(2)と(1)を用いて(13)の右辺を貪欲に最大化し、長期的には全体最適に近似して仮想待ち行列を安定に維持する。

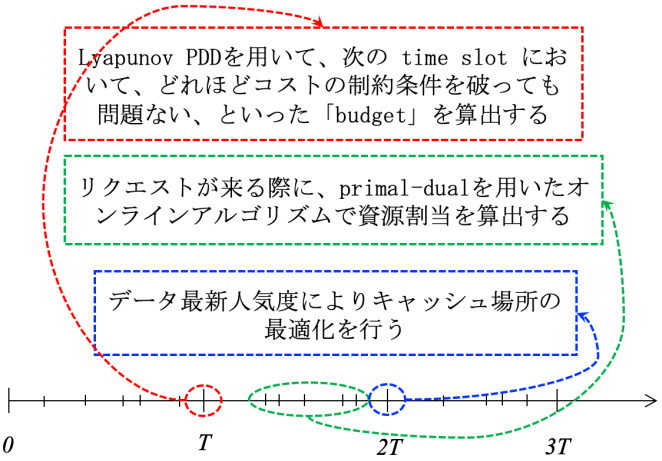

図 2: オンライン共同最適化のフレームワーク

3-2 オンラインコンピューティング資源配分

詳細は文献[5]の 3.2 節に参照してください。

3-3 パブリックデータ配置

詳細は文献[5]の 3.3 節に参照してください。

4 性能分析

本節では、提案手法のによる ECP の収益の長期性能保証を提示する。

Theorem 1. *Let $\widehat{R_N(z)}$ denote the optimal objective function value with the N-slot look-ahead problem in the zth time frame. Suppose a period of ZN coarse-grained time slots, where Z is a constant. We have*

$$\frac{1}{ZN}\sum_{T=0}^{ZN-1} R(T) \geq \left(1-\frac{1}{e}\right)\left(\frac{1}{Z}\sum_{z=0}^{Z-1}\widehat{R_N(z)} - \frac{BN}{V}\right).$$

証明は文献[5]の 4 節に参照してください。

5 シミュレーションによる性能評価

基本的な設定：提案手法を評価するために、MATLAB を用いて離散イベントシミュレータを開発した。本節では、シミュレーションによる評価結果を示す。ここでは、5 つのエッジクラウドを運用する中規模な ECP を考える。各クラウドはモバイルタスクコンピューティングのための 3 種類のリソース（例：CPU、メモリ、ストレージなど）を持っている。研究の汎用性を保つため、資源は特定しない。リソースは分割可能であり、各リソースの初期量は各拠点で 5,000 であると仮定する。資源は、2 種類の VM に似せて使用することができる。タイプ 1 の VM は各リソースを 10、20、30 個必要とし、タイプ 2 の VM は 30、20、10 個必要とする。タイプ 1 の VM の価格は単位時間あたり 10、タイプ 2 の VM の価格は 20 である。VM を組み立てるのに必要なリソースの他に、各エッジクラウドには人気データを透過的にキャッシュできるキャッシングデバイスがあると仮定する。各キャッシングデバイスのサイズは、各エッジデータセンターで同じです。特に図示しないが、総キャッシュサイズはユニバーサルコンテンツの 40%のサイズとなる。本研究では、インターネットから取得できるデータ（パブリックデータ）だけでなく、モバイルユーザがアップロードしたデー

タ（プライベートデータ）も考慮する。特に図示しないが、プライベートデータの総量はパブリックデータの 2 倍とした。エッジクラウドは、高速ネットワークで相互接続されている。ローカルキャッシュ、隣接キャッシュ、リモートクラウドからのデータ取り込みのエンドツーエンド遅延は、それぞれ［5, 10］(ms)、［20, 50］(ms)、［100, 200］(ms) の範囲の一様分布に従ってランダムに割り当てられる。粗視化タイムスロットと細粒化タイムスロットで時間を離散化し、各粗視化タイムスロットは 500 個の細粒化タイムスロットを含む。各 VM の寿命は、細粒度のタイムスロットの範囲[1, 5]に一様に分布すると仮定する。データの人気度は、指数 0.6 の Zipf 分布に従う。本節の残りの部分では、提案手法をいくつかのベンチマーク手法と比較し、パラメータの影響について議論する。

VM リクエストのダイナミクスを用いた評価こ：のシミュレーションでは、リクエストはポアソン到着に従うものとし、さらに、期待到着速度は 25 の細かいタイムスロットごとに 0 から 50 までランダムに変化するものとした。タイプ 1 とタイプ 2 の VM は等しい確率で要求される。図 3 はリクエストのダイナミクスを示す。

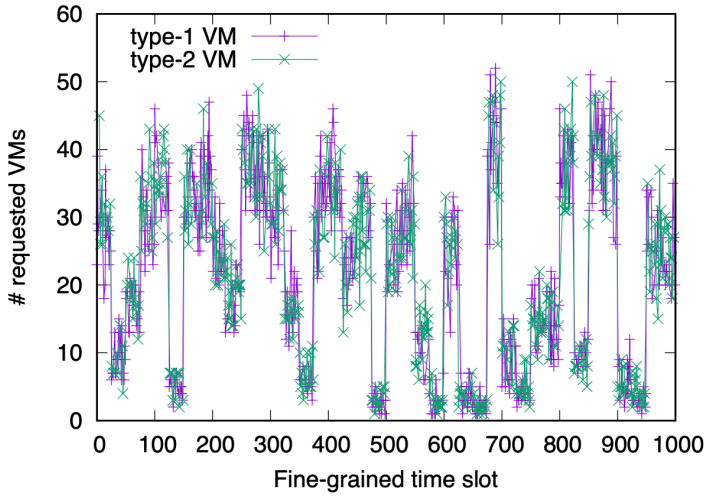

図 3: リクエストのダイナミクス

シミュレーションの間、データの人気度は固定とした。ベンチマークとして、MyopicCoop (Cooperative Caching) と MyopicNoCoop (Non-Cooperative Caching) の 2 つの方式を実装しました。Myopic Resource Allocation とは、ECP が要求を受けたときに、必要な VM を最小の伝送コストで組み立てる方法である。協調的キャッシングでは、ECP は節 3-3 で述べられた方法によりキャッシングの判断を行い、非協力的キャッシングでは、各キャッシュ装置が独立して最も人気の高いデータをキャッシュする。時間平均伝送コスト制約 L=35000、ドリフトプラスペナルティ関連パラメータ V=100000 とした。図 4 は、3 つの手法の伝送コストを示している。提案手法は、長期的な伝送コスト制約を満たしつつ、Myopic 手法よりもはるかに優れたパフォーマンスを示すことが分かる。

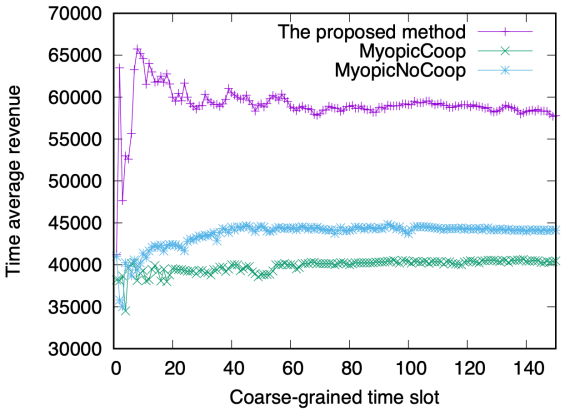

図 4：時間平均収益

図 5 は、提案手法の利点である、輸送コスト制約の時間平均表現をうまく利用していることを部分的に示している。粗視化された時間帯では、伝送コストが L を大きく超えても、要求を受け入れることができる。このようにすることで、連続するタイムスロットにおいて、伝送コストの相対的な重みが収入に対して増加する。直感的には、輸送コスト制約の時間平均表現が与えられたとき、提案手法はサンプル近視眼的手法に比べてより大きな実行可能領域を持つことになる。

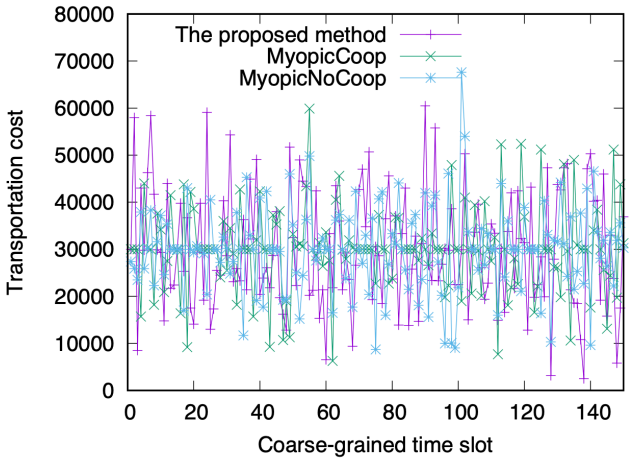

図 5：データ転送コスト

また、システムの安定性を調べるために、シミュレーション期間中の仮想待ち行列の長さを確認したところ、図 6 が示すように、時間の経過とともに 0 に近づいていくことが確認された。

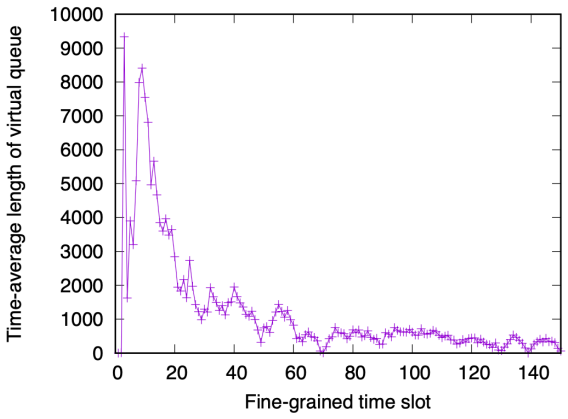

図 6：仮想待ち行列の長さ

データ人気度のダイナミクスによる評価：本研究では、VM リクエストだけでなく、データの人気度にも依存するダイナミクスを考慮する。データの人気度は時間とともに変化するが、我々の知る限り、人気度の変化を正式に定義し測定する有効な方法は存在しない。本実験では、各データの人気度変化を直接測定することを避け、代わりに 「人気度推定誤差率」という概念を導入する方法を提案する。推定誤差率は平均値 0.3 のポアソン乱数であり、トラフィックの上位 50%に寄与するデータがキャッシュされない確率を意味する。図 7 に推定誤差の動態を示す。

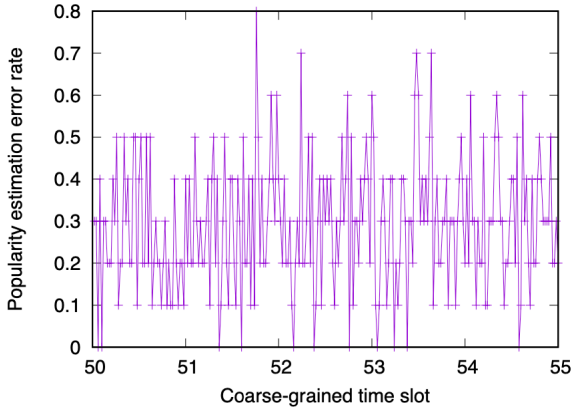

図 7: データ人気度の推定誤差

本実験では、データ人気度ダイナミクスの影響に注目し、要求到達率を細粒度タイムスロットあたり 25 に固定した。また、伝送コスト L を 35000、ドリフトプラスペナルティ関連パラメータ V を 100000 に設定した。先の実験と同様に、時間平均の収益を図 8 で、伝送コストを図 9 で、仮想待ち行列の長さを図 10 で比較します。図 8 では、提案方式が最も良い性能を発揮していることがわかる。

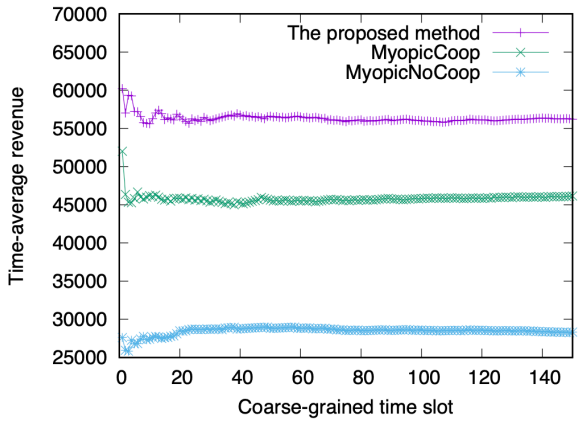

図 8: 時間平均収益

Fig. 9 はその理由を示している。提案手法は時間平均制約式をよりよく利用しており、myopic 手法ではそれができないからである。

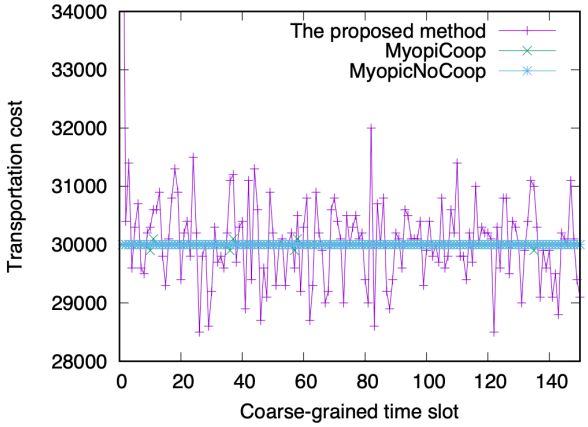

図 9: データ転送コスト

図 10 は、仮想待ち行列の長さが時間の経過とともに 0 に近づくことを示しており、システムの安定性を示している。

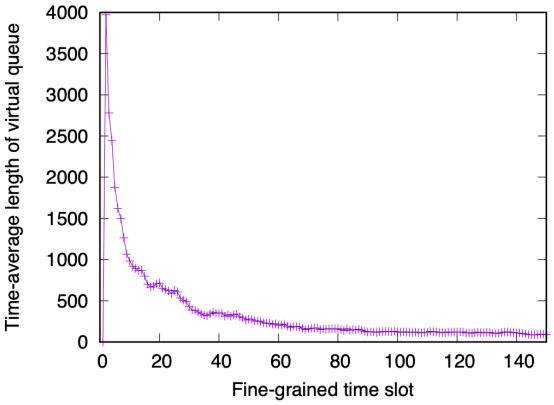

図 10：仮想待ち行列の長さ

総キャッシュサイズの影響を評価：キャッシュサイズは、システムに深く影響する。キャッシュサイズが大きいと、より多くのデータをキャッシュすることができるため、転送コストが小さくなります。また、数学的には、キャッシュサイズが大きいほど、最適化問題の実行可能領域が大きくなる。本実験では、キャッシュサイズと汎用データサイズの比を 0.1、0.5、0.9、時間平均輸送制約 L=35000、ドリフトプラスペナルティ関連パラメータ V=100000 とし、粗視化 150 タイムスロットのシミュレーションを実施した。図 11、図 12、図 13 はそれぞれ時間平均の収益、伝送コスト、仮想待ち行列長を示している。図 11 は、キャッシュサイズが時間平均収益に与える影響を直感的に示している。図 11 は、時間平均伝送コストが 3 つのキャッシュサイズのすべてで満足することを示し、図 13 はシステムの安定性を示唆する。

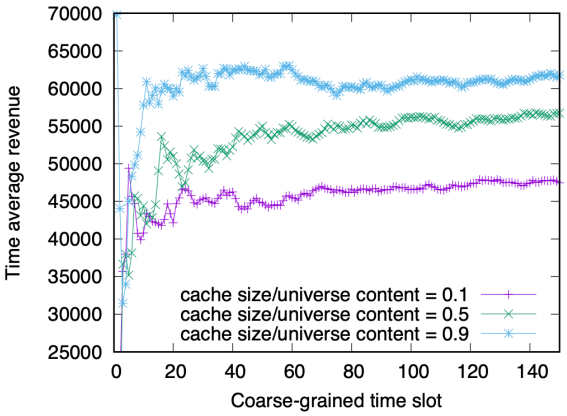

図 11：時間平均収益

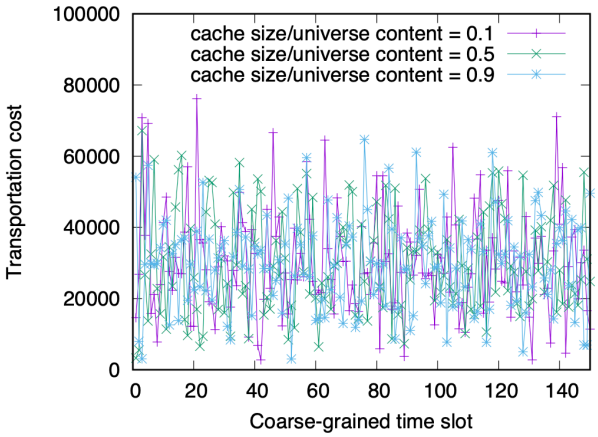

図 12：データ転送コスト

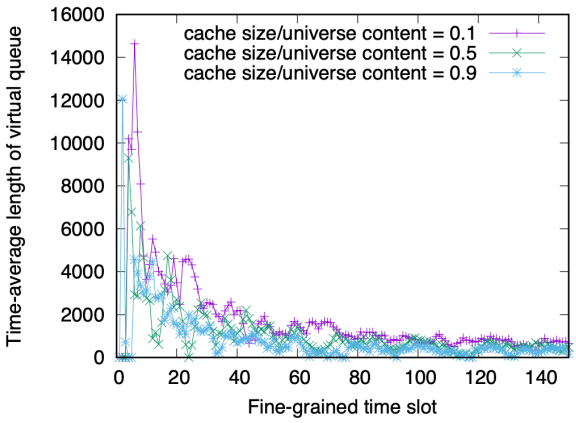

図 13：仮想待ち行列の長さ

データソースの影響を評価する：キャッシュサイズの他に、プライベートデータ量とパブリックデータ量の比率も、システムの性能を左右する重要な要素である。この比率は、エッジアプリケーションの構成を反映している。プライベートデータの例としては、モバイル機器からアップロードされた動画をさらに処理することが挙げられる。このようなプライベートデータは他人と共有できないので、キャッシュする意味はありません。パブリックデータの例としては、OTT（Over-The-Top）プロバイダーからの短いビデオがある。パブリックデータが多いと、キャッシングデバイスの使用率が上がる。本シミュレーションでは、プライベートデータとパブリックデータの量の比率を 0.5、2.0、3.5、時間平均輸送制約 L=35000、ドリフト＋ペナルティ関連パラメータ V=100000 とし、粗視化 150 タイムスロットのシミュレーションを実施した。図 14、図 15、図 16 は時間平均の収益、伝送コスト、時間平均の仮想待ち行列長を示したものである。パブリックデータが多いほど、キャッシュできるデータ量が多くなり、最適化問題の実行可能領域が広がるため、キャッシュサイズに差がある場合と同様の結果になっている。

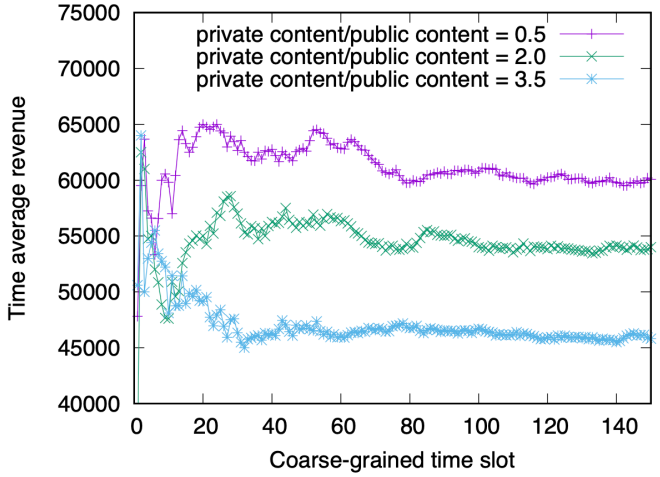

図 14：時間平均収益

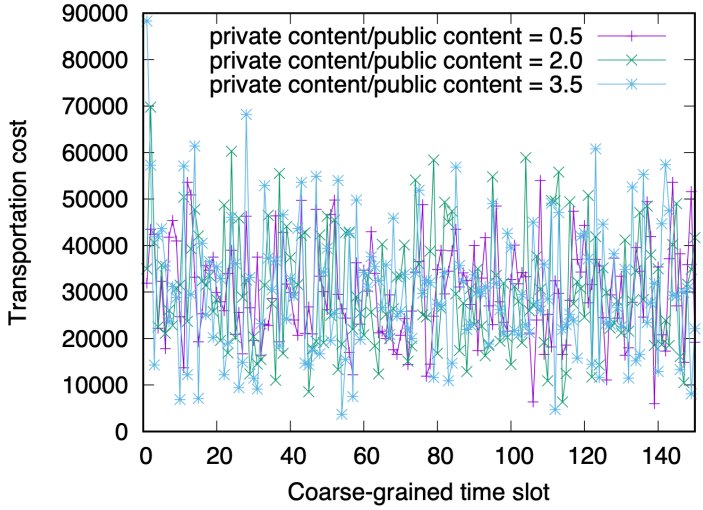

図 15：データ転送コスト

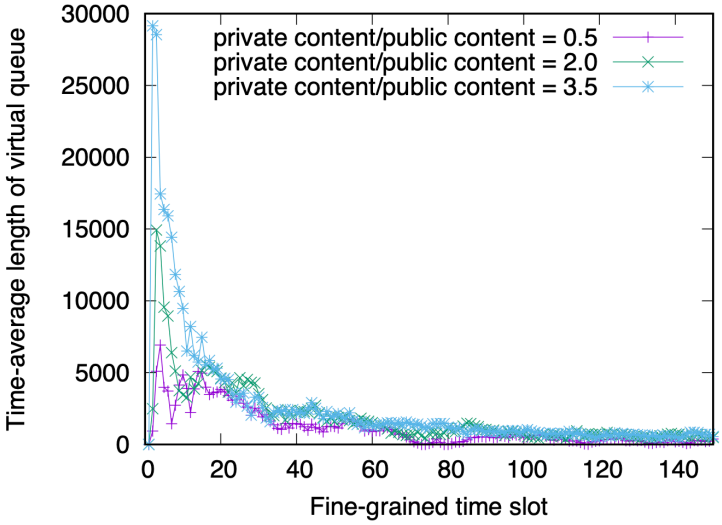

図 16：仮想待ち行列の長さ

N-look-ahead アルゴリズムとの比較：本提案手法が競合比のある最適解を達成すること、つまり理論的最適解との性能差は一定であることを証明したことになる。理論的な最適解を得るためには膨大な計算資源が必要となるため、本実験では N スロット先読みアルゴリズムを実装し、最適解を近似的に求めることとした。このアルゴリズムは、N 個の粗視化されたタイムスロットによる未来が完全に予測できると仮定する。N が無限大に近づけば、N-look-ahead アルゴリズムが理論的な最適解となることは明らかである。図 17、図 18、図 19 は N=5 での比較結果である。

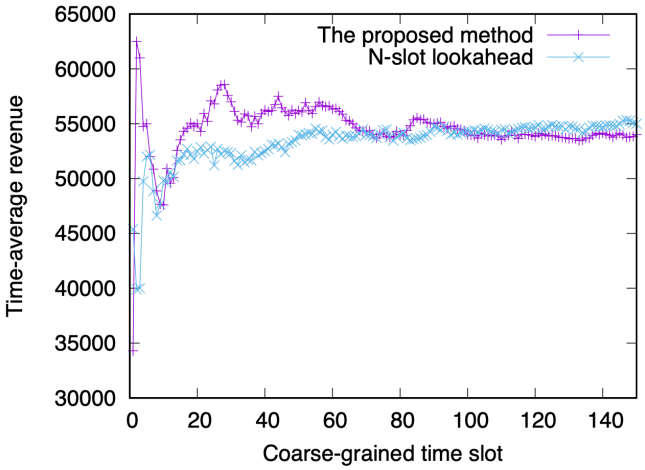

図 17：時間平均収益

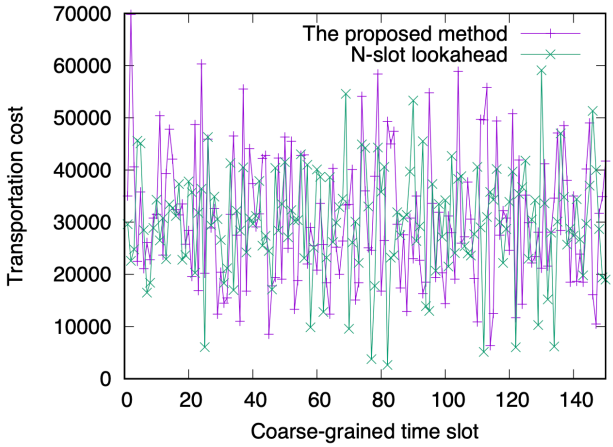

図 18：データ転送コスト

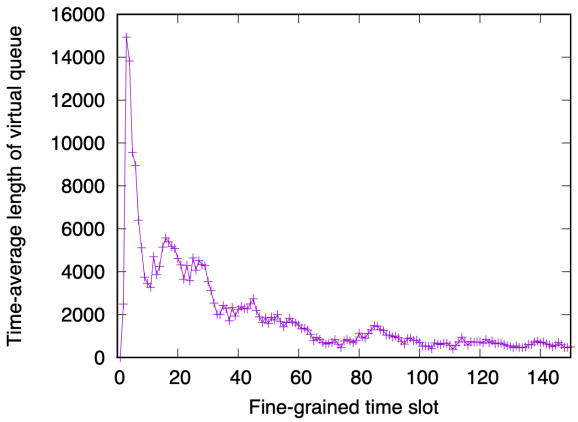

図 19：仮想待ち行列の長さ

V の影響の評価：本アルゴリズムでは、収益とコストをトレードオフするためのパラメータはドリフトプラスペナルティパラメータ V の 1 つだけである。この実験では、時間平均コスト制約 L を 30000 から 36000 まで、V を 60000 から 180000 まで変化させ、1 回のシミュレーションを 150 粗視化した時間スロット分、 1000 回行った。図 20、21、22 は V の影響を示している。明らかに、V は図 20、21 の収益とデータ伝送コスト制約の違反との間のトレードオフを制御している。長期的なデータ伝送コスト制約 C がある種の期待値である場合、V を小さくすれば制約を維持しやすくなり、V を大きくすれば制約に違反することになる。しかし、通常、V が大きい方が小さい方よりも ECP に多くの収益をもたらす。これは、主に図 22 に見られるリクエストの受け入れ率による。この事実は、ECP が収益とデータ伝送コストとの間のより良いトレードオフを行うために、実際に C と V の両方を決定するために重要である。

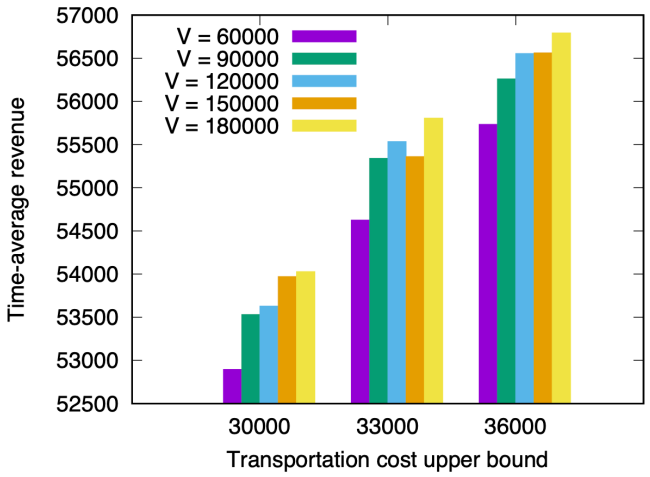

図 20：異なる L や V における時間平均収益

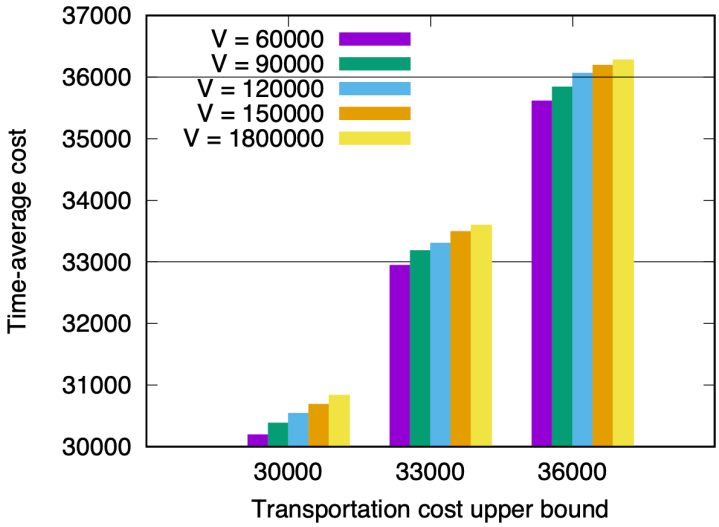

図 21：異なる L や V におけるデータ転送コスト

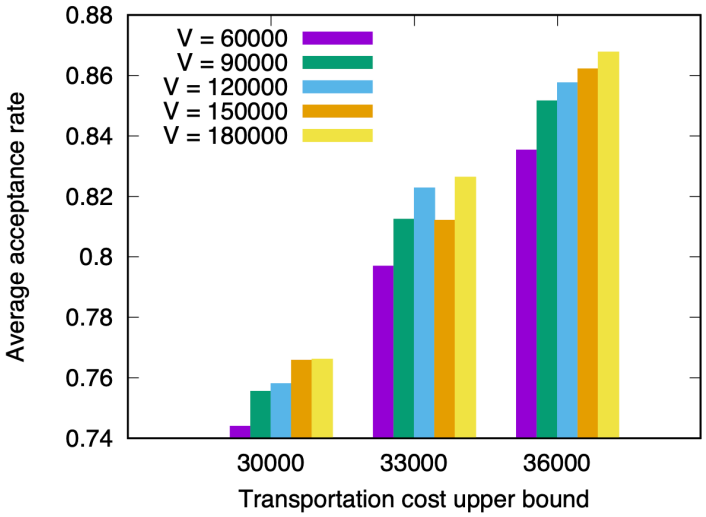

図 22: 異なる L や V におけるリクエストの受け入れ比率

6 むすび

エッジコンピューティングは、クラウドコンピューティングとモバイルコンピューティングの時代において、ますます重要性を増している。サービスやデータをエッジに移すことで、ネットワークプロバイダーはコアネットワークの負担を軽減し、モバイルユーザーは性能向上の恩恵を受け、サービスプロバイダーは新しいサービスやアプリケーションを開発する機会を得ることができる。しかし、現在、多くの研究がアプリケーションに特化したエッジコンピューティングシステムに集中しており、エッジコンピューティングの利点を十分に生かしきれていないのが現状である。本研究では、汎用エッジコンピューティングのビジョンを提示し、汎用エッジコンピューティング環境におけるアプリケーションとエコシステムを分析する。効率的な汎用エッジコンピューティングを実現するための主な課題として、異種環境における動的な要求に対する資源配分とデータ配置の相互関係を明らかにし、将来のシステム状態に関する仮定や知識なしに一定のギャップで最適化への近似値を得ることができる新しいオンラインのフレームワークとアルゴリズムを提案する。理論解析とシミュレーションの結果、提案手法により、エッジクラウドプロバイダは仮定やシステムの将来知識なしに、ほぼ最適な利益を獲得できることが示された。

【参考文献】

[1] W. Cai, R. Shea, C. Huang, K. Chen, J. Liu, V. C. M. Leung, and C. Hsu, “A Survey on Cloud Gaming: Future of Computer Games,” IEEE Access, Vol. 4, pp. 7605–7620, 2016.

[2] W. Cai, F. Chi, X. Wang, and V. C. M. Leung, “Toward Multiplayer Cooperative Cloud Gaming,” IEEE Cloud Computing, Vol. 5, No. 5, pp. 70–80, 2018.

[3] W. Cai, Y. Chi, C. Zhou, C. Zhu, and V. C. M. Leung, “Ubcgaming: Ubiquitous Cloud Gaming System,” IEEE Systems Journal, Vol. 12, No. 3, pp. 2483–2494, 2018.

[4] M. J. Neely, “Stochastic Network Optimization with Application to Communication and Queueing Systems,” Morgan and Claypool Publishers, 2010.

[5] X. Shao, G. Hasegawa, M. Dong, Z. Liu, H. Masui, and Y. Ji, “An Online Orchestration Mechanism for General-Purpose Edge Computing,” IEEE Transactions on Services Computing, DOI: 10.1109/TSC.2022.3164149, April, 2022.

〈発　表　資　料〉

題　　名	掲載誌・学会名等	発表年月
A Novel Neural Network Model for Demand Prediction of Bike-Sharing	Proceedings of EAI MONAMI 2020	2020年11月
Neural Networks with Improved Extreme Learning Machine for Demand Prediction of Bike-Sharing	Springer Mobile Networks and Applications	2021年3月
A Grid and Vehicle Density-Prediction-based Communication Scheme in Large-Scale Urban Environments	Proceedings of IEEE ICT-DM 2021	2021年12月
Collaborative Intelligence Enabled Routing in Green IoV: A Grid and Vehicle Density Prediction Based Protocol	IEEE Transactions on Green Communications and Networking	Accepted for publication (2022年6月)
An Online Orchestration Mechanism for General-Purpose Edge Computing	IEEE Transactions on Services Computing	2022年4月