

# 自律分散セッションステートマイグレーションによる需要変動 に応じたサーバ制御の検討

臼井 健<sup>†</sup> 地引 昌弘<sup>†</sup> 作元 雄輔<sup>‡</sup> 高野 知佐<sup>††</sup> 会田 雅樹<sup>‡</sup> 西永 望<sup>†</sup>

<sup>†</sup> 独立行政法人 情報通信研究機構 ネットワーク研究本部 〒184-8795 東京都小金井市貫井北町 4-2-1

<sup>‡</sup> 首都大学東京大学院 システムデザイン研究科 〒191-0065 東京都日野市旭ヶ丘 6-6

<sup>††</sup> 広島市立大学大学院 情報科学研究科 〒731-3149 広島市立安佐南区大塚東 3-4-1

E-mail: <sup>†</sup> ta-usui@nict.go.jp

あらまし サーバを省電化のために最適に集約する技術が注目されている。一般にサービスに対するアクセスは、オン・オフピーク時で、大きな隔たりがあり、需要に応じてサーバ台数を増減させ、省電力化やリソースの有効活用のため台数を最適化することが望ましい。筆者らは通信に関わる情報であるセッション情報を移設すること通信断を発生させずクライアントが接続するサーバを別の IP アドレスを持つサーバへ切り替えるセッションステートマイグレーション(以下、SSM)を提案している。SSMにより、通信断を発生させずサーバ台数を需要に応じて制御できるが、多数のサーバが存在する環境でサーバを制御する際、特定のサーバにセッション情報が集中することなく最適にサーバを集約、増設するかという課題がある。本稿では、多数のサーバを管理するネットワークで、局所的な情報共有だけにに基づく自律的判断で、SSMにより最適にサーバの集約や増設する方法を提案する。隣接するサーバがセッション情報数を自律分散的に交換し、動的にセッション情報を交換するためのグループを作成し、サーバ集約や、増設時に各サーバのセッション情報数を平滑化する手順を示す。また、自律分散制御により形成されたグループ内のセッション情報の平滑化について、提案方式の有効性をシミュレーションにより示す。

キーワード 自律分散, セッションステートマイグレーション, 需要変動, 新世代ネットワーク

## A Server Configuration with Autonomous Session State Migration to Demand Fluctuation

Takeshi Usui<sup>†</sup> Masahiro Jibiki<sup>†</sup> Yusuke Sakumoto<sup>‡</sup> Chisa Takano<sup>††</sup> Masaki Aida<sup>‡</sup> Nozomu Nishinaga<sup>†</sup>

<sup>†</sup> Network Research Headquarters, National Institute of Information and Communications Technology, 4-2-1, Nukui-Kitamachi, Koganei, Tokyo 184-8795, Japan

<sup>‡</sup> Graduate School of System Design, Tokyo Metropolitan University, Hino-shi, Tokyo 191-0065, Japan

<sup>††</sup> Graduate School of Information Sciences, Hiroshima City University, Hiroshima-shi, Hiroshima 731-3194, Japan

E-mail: <sup>†</sup> ta-usui@nict.go.jp

**Abstract** Optimal server consolidation is attracting attention for energy saving. The service access increases or decreases in number depending on the on-peak or off-peak. Optimally increasing or decreasing the number of servers to the demand fluctuation is desired for the energy saving and utilizing the resources. We have proposed session state migration architecture (SSM), which switches over the connection between the server and client without disrupting the communication by relocating session states from one server to another. The SSM enables us to increase or decrease the number of servers to the demand fluctuation without disrupting the communication. In the network managing a large amount of servers by the SSM, this paper proposes the method for the server configuration to increase or decrease the number of servers based on the autonomous decision from the locally shared information. In the proposed method, the number of session states is autonomously exchanged between the adjacent servers, and the groups for dynamically relocating the session states are created. This paper shows the procedures to consolidate the servers and to equalize the number of session states kept by the servers. A simulation shows the evenness of the number of session states among the groups created by the autonomous decentralized control.

**Keyword** Autonomous, Session state migration, Demand fluctuation, New Generation Network

## 1. はじめに

インターネット上の様々なサービスが、事業者がデータセンタに設置したサーバより提供されている。サーバより企業のようなサービスが展開されており、設置されるサーバの品質保証の重要性はますます増加している。サービスの品質低下を抑える方法として、クライアントからのリクエストを予測し、同じサービスを提供するサーバを追加し、各サーバの負荷を軽減させる。

一般にサービスに対する需要は、オン・オフピーク時で大きく変動するため、省電力、リソースの有効活用のためオフピーク時にクライアントを収容しないサーバを停止する必要がある。事業者は、クライアントの需要に応じ、データセンタ内で稼働するサーバを集約し最小限にすることを検討している。

しかし、少数のクライアントがサーバにアクセスしている状況であっても、事業者はサーバを停止することはできない。なぜなら、多くのサービス（例えば、映像配信、電話のための呼制御、シンクライアントなど）ではクライアントが接続するサーバが変わると、サービスを利用する手続きを一から始める必要がある。サーバを停止することは、サービスの品質を低下させることに繋がる。また、映像配信を行っているサーバではクライアントは1,2時間アクセスし続けることがあり、そのサーバを停止するために事業者はクライアントが利用停止するまで待つ必要がある。

新世代ネットワーク[1]のサーバ制御のため、筆者らは、通信に関わる情報であるセッション情報を移設すること通信断を発生させずクライアントが接続するサーバを別のIPアドレスを持つサーバへ切り替えるセッションステートマイグレーション(以下、SSM)[2]を提案している。SSMによりクライアントを他のサーバへ接続させ、そのサーバを停止可能にする。SSMは、セッション情報単位で移設できるため、あるサーバから複数のサーバへセッション情報を分割して移設できる。近年注目されている仮想マシンの移設では、移設は仮想マシン単位であり、移設の単位の柔軟さがないため各サーバのリソースを使いきれない。本稿では、多数のサーバが存在する環境で、SSMによりサーバの集約、増設を行うことを想定する。

SSMにより多数のサーバを管理する場合、セッション情報をどのサーバからどのサーバへ移設するかを適切に決定することは困難である。なぜなら適切に行うためには、全てのサーバで収容しているクライアント数を把握し、各サーバへセッション情報の移設を命令する制御ノードが必要となるからである。また、クライアントの接続は動的に消滅するため、定期的にサーバに問い合わせ必要がある。サーバの数が大きくなればなるほどその制御ノードの負荷が大きくなる。

本稿では、制御ノードとなるロードバランサが保持する情報を最低限に抑え、各サーバが自律的にSSMを実行する、自律分散セッションステートマイグレーション(ASSM)を提案する。提案保方式では、ロードバランサが保持する情報として稼働しているサーバのリストのみに限定し、セッション情報の移動先のサーバ

は、全体最適化が望めないほどサーバが存在する環境で、局所的な情報共有だけに基づく自律的判断により決定する方法を提案する。

## 2. サーバ制御における課題

本節では、サーバ制御における課題を解決するための要求条件を述べる。また、既存のサーバ制御に関する研究を述べ、課題を説明する。

### 2.1. 要求条件

本節では、多数のサーバがいる環境でクライアントの需要に応じ、サーバの増設、集約を行っていくための3つの要求条件を示す。

(1)データセンタではサーバの負荷分散のためクライアントからのリクエストをロードバランサが受け付け、適切なサーバへ転送するのが一般的である。特定サーバへ負荷が集中させないために、ロードバランサは各サーバから収容しているクライアントの数を把握する必要がある。クライアントのアクセスの消滅は動的であるため、ロードバランサは定期的に情報を収集する必要がある。多数のサーバがいると負荷が大きくなる。ロードバランサに負荷を削減し、各クライアントからのアクセスを最適に分散する必要がある。(ロードバランサの負荷削減)

(2)クライアントからのリクエストが減少し各サーバのリソースの余裕が出てきた際、省電力化のためサーバを集約することが望まれる。サーバ集約を高速に達成するために、並行してサーバ集約を行う必要がある。しかし、並行してサーバ集約を実施するために、多数のサーバの中から最適にグループ分けし、セッション情報を動かすサーバやセッション情報を受け付けるサーバを決定していくことは困難である。集約され電源停止になるサーバがあるため、特定のサーバからセッション情報を移していくと決定することはできない。また、グループ分けしたサーバ間でセッション情報を移設し、特定のサーバにセッション情報が集中しないように各サーバが自律分散的にサーバを集約していく方式が必要である。(動的なサーバ集約)

(3)クライアントからのリクエストの増加が見込める場合、サーバを増設する。ロードバランサはラウンドロビン方式でリクエストを転送するので、特定のサーバに負荷が集中しないように各サーバが保持するセッション情報数を均一化しておく必要がある。しかし、多数のサーバが存在する場合、増設されたサーバを含め各サーバが収容するセッション情報数を高速に平滑化していくことが困難である。網全体でセッション情報の数の平滑化を図ると時間を多数のサーバが存在する場合時間を要する。サーバ集約時と同様にサーバ全体の中でグループ分けを実施し、各サーバが保持するセッション情報数を平滑化していくことが必要である。(動的な負荷平滑化)

### 2.2. 関連研究

需要変化に対応してサーバを管理する研究として、エージェントなどの制御ノード[3,4,5,6]を用いた方式と

Peer-to-Peer 方式[7]を対象にしたものがある。

Bernardo ら[3]は、サーバエージェントとの情報管理を行うロケーションサーバを設置し、クライアントの数に応じ、サービスを提供している。クラスタ内のサーバの負荷を制御ノードが監視することで、負荷に応じて自動的にサーバ増減する機構として、HotRod[4]、Muse[5]、Pinherira らによる研究[6]がある。これらの手法は、提案手法とは異なり、制御ノードにより各サーバの情報を監視するため、制御ノードの負荷が問題となる。また、クラスタを対象にしている場合、異なる IP アドレスを持つサーバを扱うことはできず、適応環境が限定される。

Peer-to-Peer 方式で各サーバ同士が情報を交換し、ミラーサーバの台数を自動的に増減する機構[7]が提案されている。この機構では、サーバを集約するためのセッション情報の移設や各サーバが保持するセッション情報の平滑化を考慮していない。そのため、効率的にサーバを集約できず、またクライアントからのアクセスが増えた時、提案手法より多数のサーバを増設することが考えられる。

### 3. 自律分散セッションステートマイグレーション

本節では 2 節の 3 つの技術要件を満たす ASSM の設計を示す。

#### 3.1. 概要

本稿では、図 1 のようにロードバランサがデータセンターの入り口に位置し、クライアントからのリクエストを処理する構成を想定する。ASSM では、局所的にサーバ同士が保持するセッション情報を交換させ、サーバ集約を目指す。多数のサーバが任意のサーバとセッション情報を交換する場合、各サーバで不特定多数のサーバとのメッセージ交換の正常性を確認する必要があり、運用コストが高くなる。本稿では、多数のサーバを容易に管理するため、各サーバは論理的に 1 次トラス型の隣接関係を形成するものとする。また、SIP や映像配信サーバの製品では、品質を維持するため、接続するクライアントの最大値を決める。本稿では、既存製品と同様にサーバが保持するセッション情報の最大値（収容可能上限）以下になるように、サーバを運用する。

事業者のオペレータが予め定めた間隔で定期的な各サーバが ASSM を実行し、稼動するサーバの削減、保持するセッション数の最適化を行う。クライアントからのリクエストが増えてきた際、サーバを増設する。ロードバランサは、到着するリクエストの到着間隔または過去の到着するリクエストのデータより、必要なサーバを推測するものとする。

#### 3.2. ロードバランサの機能

本節では、本稿で想定するロードバランサの機能を紹介する。ロードバランサは、転送するサーバのリストだけ把握し、ラウンドロビン方式でリクエストを各サーバへ転送する。ASSM により、各サーバ間で強調動作しサーバを集約していく。ロードバランサにサーバ集約のため特別な機能を要求しない。ロードバランサは、動的に停止されるサーバへ転送するのを防ぐために、リクエストが到着した際、サーバに受け付け可能かどうか問い合わせる。

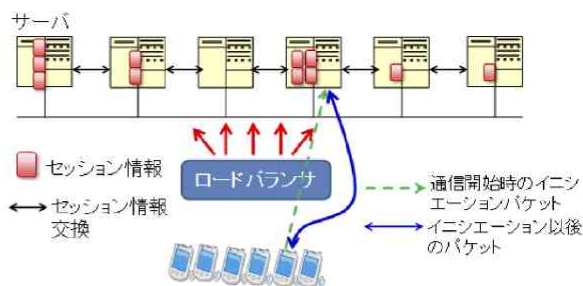


図 1 想定するネットワーク構成

クライアントは DNS によりサーバの IP アドレスを知るが、そのアドレスは共通の 1 つのアドレスで、ロードバランサにより各サーバへ転送される。SSM では通信開始後もサーバ・クライアント間で通信に利用するアドレスを変更することを可能にする。本稿では、その機能を利用し、サーバが通信に使用するアドレスをそのサーバ固有のアドレスに変更するものとする。図 1 に示すように、通信開始時のパケットはロードバランサを経由するが、以後のパケットはサーバ・クライアント間で直接やり取りされる。SSM を活用することで、ロードバランサを通過するトラフィックは軽減される。

#### 3.3. 動的なサーバ集約

本節では、サーバ集約のため、次の 2 つの Step を定期的に行い、ASSM を行う。実行間隔は事業者のポリシーに依存する。後述する自律分散制御により、多数のサーバの中でグループを形成する。自律分散制御の開始状態の S メトリックの初期値として、各サーバが保持するセッション情報数を用いる。

・ Step 1 隣接サーバと S メトリックを交換し自律分散制御により S メトリックの値を更新する。S メトリックの値に基づき 1 次トラスで形成サーバの中からグループ分けを実施し、セッション情報を移設し始めるサーバを決定する。

・ Step 2 Step 1 で決定したサーバからセッション情報を移設する。

以下の節でそれぞれの Step の手順を説明する。

##### 3.3.1. グループ分け

サーバ集約のための Step1 でのグループ分けの手順を説明する。一般に、自律分散的なシステムでは、サブシステムがそれぞれ局所的な動作を行うため、サブシステムの動作が適切に設計されていないと全体として破綻しうまく動作しない場合がある。多数のサーバが存在する環境で、各サーバが独自に SSM を実施していくと、オンピーク時に、特定のサーバにクライアントの接続が集中することも考えられる。その場合、別途無駄にサーバを増設する必要があり、サーバを制御する効果が小さくなる。特定のサーバに負荷が集中しないように適切にグループを決定する必要がある。

拡散方程式を利用した自律分散制御[8,9]を活用し、多数のサーバの中でグループ分けを実施し、SSM を実行していく制御方針を決定する。この自律分散制御の枠組みは、「システム全体の状態を結果的に望ましい方向に導くようにするために必要なサブシステムの動作の適切な設計」に関する処方箋を与える。ASSM では自律分散制御による処方箋を使い、セッション情報移設のためのグループ分けを実施する

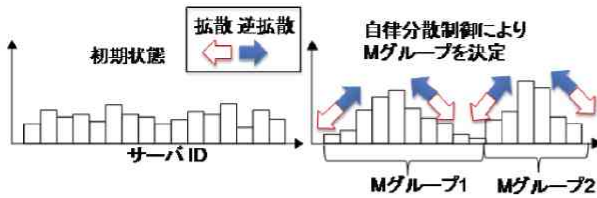


図2 自律分散制御を用いた M グループの決定方法概要

自律分散制御により、多数のサーバの中で各サーバが保持するセッション情報数を考慮し、グループ分けし、セッション情報を移設させるサーバ、受け付けるサーバを決定する指針を提供する。サーバの停止・追加があるため、グループ分けを決定する際、特定のサーバからセッション情報を移設、受け付けさせるように規則を予め決めることはできない。各サーバが保持するセッション情報数を考慮し、自律分散制御は、周囲の状況に適応した構造を生み出す。

自律分散制御により形成されるセッション情報を移設・交換するグループを M グループと定義し、M グループ形成のための手順を以下に示す。まず各サーバが隣接サーバと S メトリックを交換し、自律分散制御により各サーバが保持する S メトリックの値を変化させていく。ASSM では、各サーバは、ある一定間隔で保持しているセッション情報の数を S メトリックとして、隣接サーバと S メトリックの値を交換し、S メトリックの値を自律分散制御により更新していく。

図2に自律分散制御による M グループ決定方法の概要を示す。図2の左部が示すように、ある初期状態では各サーバは規則性のない S メトリックを保持するが、近接サーバと S メトリックを交換することで右部のように、M グループ、M エッジサーバを決定する。ある時刻  $t$  においてネットワーク内の各サーバ  $x$  が持つ S メトリックを  $q(x, t)$  とする。  $q(x, 0)$  として各サーバが保持するセッション情報数を用いる。各サーバの  $q(x, t)$  の総計は自律分散制御の計算中は一一定である。各サーバは隣接サーバの S メトリックの情報だけ交換し、  $q(x, t)$  の変化状況を計算していく。  $q(x, t)$  の分布の中で各サーバはピークを平滑化する拡散の働き（図2左部の斜め下向きの矢印）と、ピークを強調する逆拡散する働き（図2左部の斜め上向きの矢印）と釣り合うことで、分布  $q(x, t)$  の形状に何らかの構造を与える。以下に計算アルゴリズム概要を示す。

- 分布の初期値を  $q(x, 0)$  とする。
- 以下の微分方程式(1)~(3)に従い、隣接ノードの分布の値から各ノードの分布の値  $q(x, t)$  を時間発展させていく。  $c$  は分布の時間発展の速さ、  $\sigma^2$  は拡散の強さを表す。  $\kappa'$  は拡散係数である。

逆拡散を表す項

↓

拡散を表す項  
(通常の拡散方程式)

↓

$$\frac{\partial q(x, t)}{\partial t} = c \left( \frac{\partial f(x, t)}{\partial x} + \sigma^2 \frac{\partial^2}{\partial x^2} \right) q(x, t) \quad (1)$$

$$f(x, t) = -\frac{\partial}{\partial x} \Phi(x, t) \quad (2)$$

$$\Phi(x, t + dt) = -\left( q(x, t) - \kappa' \frac{\partial^2 q(x, t)}{\partial x^2} dt \right) \quad (3)$$

自律分散制御では、式(3)  $x$  の 2 階偏微分を、隣接サーバの保持する S メトリックの差分として計算する。各サーバは、  $a$  (自身と右隣りサーバとの差分)、  $b$  (自

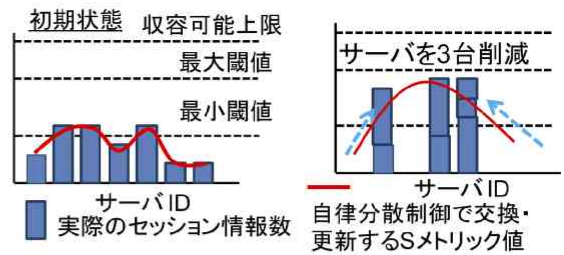


図3 セッション情報移設動作概要

身と左隣りサーバとの差分)を求め、  $a - b$  を計算する。一定回数 (定数  $t$ ) 各サーバは S メトリックを更新した後、隣接サーバ両方と比較して小さい S メトリックを持つサーバを、M エッジサーバと定義する。M エッジサーバとなったサーバはセッション情報を移設しないサーバに自身が M エッジサーバであることを通知し、その情報を受信したサーバは別の M グループの M エッジサーバとなる。

### 3.3.2. セッション情報の移設

各サーバはセッション情報を受け付ける際に、特定のサーバに過度に集中しないように次の二つの閾値を保持する。

- 最小閾値：收容セッション情報数がこの閾値より小さい場合は、セッション情報を他のサーバへ移設する。
- 最大閾値：この閾値より大きい場合は、他サーバからのセッション情報を受け付けない。

図3は単一の M グループ内のセッション情報の移設動作概要を示す。M エッジサーバに選定からセッション情報を移設していく。最小閾値以上、最大閾値以下のセッション情報を保持するように隣接サーバへセッション情報を移設していく。また、隣接サーバへ全セッション情報を移設すると最大閾値を超えてしまう場合、隣接のさらに隣接サーバへセッション情報を移設するものとする。

図3では、SSMにより7台のサーバが3台のサーバへ集約される例を示している。図3の曲線は、自律分散アルゴリズムにより計算した S メトリックを線で結んだものである。S メトリックにより、セッション情報を移設するサーバを決定し、最大閾値を超えないように SSM を実施している。セッション情報を收容しなくなったサーバは停止される前に隣接サーバへ、自身が隣接関係を形成していたサーバのアドレスを通知する。あるサーバが停止されても、その隣接となっていたサーバ間同士で新たに隣接関係が形成される。

### 3.4. 動的な負荷平滑化

本節では、クライアントからのリクエストが増加してきた際、サーバを増設し各サーバを收容するクライアント数を平滑化するため、次の4つの手順を実行する。

- Step 1 M グループに、サーバを追加する。
- Step 2 M グループ内で追加したサーバが隣接関係に加わる。
- Step 3 M グループ内で各サーバが保持するセッション情報数を平滑化のため、自律分散制御により S メトリックを交換し、値を更新する。
- Step 4 各サーバは、交換した S メトリックの値をもとに、保持するセッション情報数との差分を隣接に移設するか、隣接サーバからセッション情報を受け取り、

Mグループ内での各サーバが保持するセッション情報数の平滑化を実現する。

Step2で、増設されたサーバはまず、LANに新規に追加されたことを示すマルチキャストメッセージを送信する。Mトップサーバはそのメッセージに回答を返し、新規に増設されたサーバはそのMトップサーバと隣接関係を形成する。複数のMトップサーバが回答を返すことが考えられるがその場合、速く回答を返したMトップサーバのグループに属する。

Step3で、各サーバはMグループ内の各サーバが収容するセッション情報数を平滑化するためSメトリック値を更新する。この際、自律分散制御をMグループ内のみで実施する。ここでの自律分散制御は各サーバが保持するセッション情報数の平滑化のみを目指すので、Sメトリック値の拡散のみを実施する。Step4で、各サーバは得られたSメトリック値の差分の数だけセッション情報を隣接サーバへ移設し、または隣接サーバから受け取り、Mグループ内のセッション情報数の平滑化を実現する。

Mグループ内ではセッション情報数は平滑化されるが、Mグループ間でも保持する各サーバが保持するセッション情報数は平滑化されていることが望まれる。特定のMグループ内で各サーバが保持するセッション情報数が多いと、ロードバランスはラウンドロビン方式でリクエストを転送するだけであるため、特定のMグループ内のみにはサーバを増設する必要が出てくる。他のMグループには、リクエストを十分に収容できるサーバが存在するため、不効率なサーバ増設となる。このような不効率な増設を防ぐため、平均セッション情報数のばらつきは少なくする必要がある。

## 4. 平滑化効果の評価

本節では、サーバを効率的に増設するため自律分散制御に形成されたMグループ間で各サーバが保持するセッション情報数がどの程度平滑化されているか、静的にMグループを決定した場合と比較し、評価する。Mグループ毎で各サーバのセッション情報数が平滑される期待値（すなわち各サーバが保持する平均セッション情報数）がMグループ間での標準偏差を検証する。

### 4.1. 評価方法

サーバ台数200、2000のネットワークで、各サーバが保持するセッション情報数として、指数分布( $\lambda=1$ )に基づく数値を乱数で与え、100を乗じ、小数点以下を棄却した値を与えた。図4、図5は200、2000台の各サーバが保持するセッション情報数を表示したものである。X軸は、各サーバのサーバIDを指し、Y軸は保持するセッション情報数である。各サーバは自律分散制御の初期値として、このセッション情報数を使う。

サーバ台数200、2000のネットワークで自律分散制御を一定時間実施し、Mグループを形成する。静的にMグループを形成した場合と比較し、各Mグループの平均セッション情報数に対して、Mグループ間の分散値を評価する。自律分散制御により形成されたMグループ数と同数のグループを、静的に形成されるものとして、各Mグループ間のサーバ数が同等でMグループを作成した。自律分散制御と静的にMグループを決定した際、各Mグループ内で平滑されたセッション情報数の分散を調査する。また、 $c$ は1、 $\sigma^2$ は0.4、 $\kappa'$ は0.1、 $t$ は90として自律分散制御を実施した。

## 4.2. 評価結果

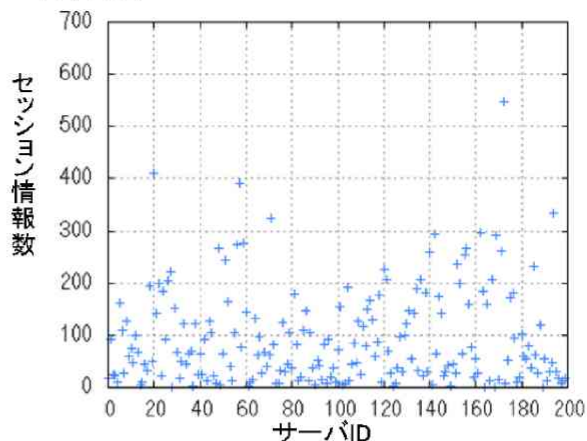


図4 200台のサーバが保持するセッション情報数

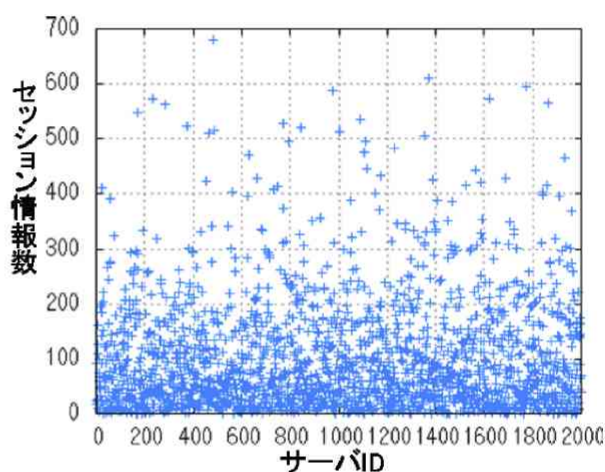


図5 2000台のサーバが保持するセッション情報数

図6、7は200台、2000台のサーバが自律分散制御により更新したSメトリックの値を示す。図4、5のセッション情報数と比較すると、極端に大きな値をとるSメトリックは存在せず、一方で極端に小さい値も存在しない。図6に記したように、200台のサーバの環境では、3つのMグループが形成された。比較対象として、サーバID1から67、68から134、135から200で静的にMグループを形成される場合を考える。

表1に200台のサーバを3つのグループに分けた場合で、提案手法と静的にグループ分けした場合で、グループ内のサーバが保持するセッション情報数の平均値（平均セッション情報数と定義）を示す。クライアントからのリクエストが増加してきた状況を考えて、提案方式は静的にグループを形成する場合より、9台少ないサーバでクライアントを収容できる。

表2に提案方式、静的な方法でMグループを作成した場合で、200、2000台のサーバで、各Mグループが保持する平均セッション情報数の標準偏差を示す。2000台のサーバが存在する環境では自律分散制御により、42個のMグループが形成された。Mグループ内に48台ずつサーバが属する形で、静的に42個のMグループを作成し、提案方式と平均セッション情報数の標準偏差を比較した。

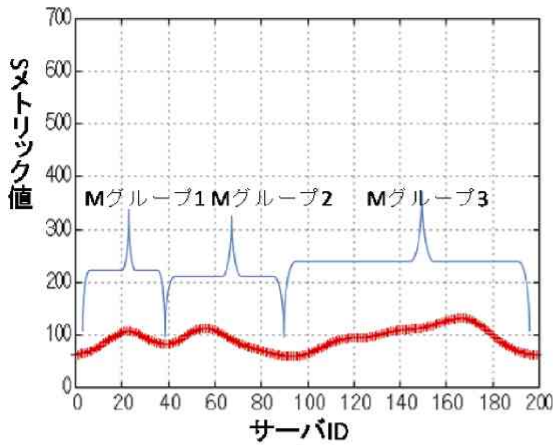


図 6 自律分散制御後の 200 台のサーバが保持する S メトリック

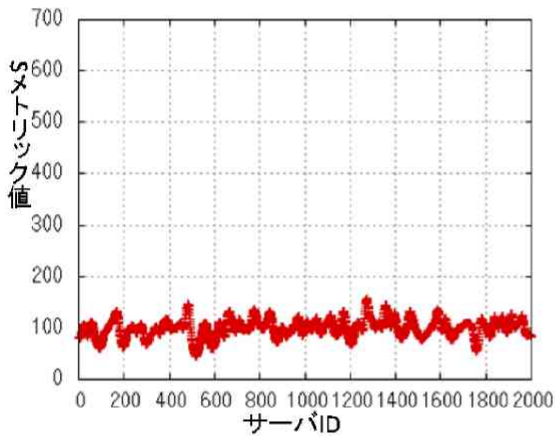


図 7 自律分散制御後の 2000 台のサーバが保持する S メトリック

表 1 200 台のサーバで、各 M グループ内で平滑されたセッション情報数

	グループ内の ID の範囲	グループの平均セッション情報数
提案手法	1-40	87.3
	41-94	89.2
	95-200	95.6
静的なグループ分け	1-67	94.5
	68-134	74.1
	135-200	104.6

表 2 200、2000 台のサーバで、各 M グループが保持する平均セッション情報数の標準偏差

	グループ作成方法	平均セッション情報数の標準偏差
サーバ 200 台	提案方式	3.4
	静的	12.7
サーバ 2000 台	提案方式	8.3
	静的	13.7

表 2 の結果より提案方式を用いると、静的に M グループを作成する場合に比べて、保持するセッション情報数の値にばらつきが少ないグループを作成できていることがわかる。標準偏差の値が大きく M グループ内の平均セッション情報数にばらつきがあると、特定の M グループにてクライアントからのリクエストを処理できなくなり、サーバを増設する必要が出てくる。表

2 の標準偏差の値より、提案方式のほうがサーバを増設することなく、クライアントからのリクエストを処理することがわかる。

提案方式では、サーバ 2000 台の環境は、200 台の環境と比べると、各 M グループの平均セッション情報数にばらつきがある。サーバ 2000 台の環境では、サーバ 200 台の環境に比べて、平滑された M グループを形成するためには、長時間 S メトリックを交換・更新する必要があるためである。サーバ台数に応じて計算時間 (変数  $t$ ) の最適化を考えていく必要がある。

## 5. おわりに

本稿では、自律分散制御によりセッション情報の移設先を決定しサーバ集約や保持するセッション情報数の平滑化をする ASSM というサーバ制御方式を提案した。多数のサーバの管理を容易にするため、サーバの接続を論理的に一次元トラス型の隣接関係を形成していると想定した。ASSM では、多数のサーバの中で各サーバが保持するセッション数を考慮したグループを自律分散的に形成し、そのグループ内でサーバ集約、セッション情報数の平滑を実施する。

サーバ 200 台、2000 台を想定したミュレーション試験を実施、自律分散的にグループを形成しセッション情報を移設していく提案方式のほうが、各グループのセッション情報数を平滑できることがわかった。今後は、自律分散制御のパラメータの最適化と、より多数のサーバが存在する環境でのサーバ集約、平滑化効果を評価していく。

## 文献

- [1] N. Nishinaga, "NICT New-Generation Network Vision and Five Network Targets," IEICE Transactions on Communications Vol. E93-B, no. 3, pp. 446-449 2010.
- [2] T. Usui, K. Nakauchi, Y. Shoji, Y. Kitatsuji, H. Yokota, N. Nishinaga, "Design of Session State Migration Middleware for Disruption-free Services," The 6th International Conference on Next Generation Mobile Applications, Services and Technologies (NGMAST) 2012
- [3] L. Bernardo and P. Pinto, "Scalable service deployment using mobile agent," 2<sup>nd</sup> Int'l Workshop on Mobile Agents (MA'98), Lecture Notes in Computer Science 1477, pp.261-272, 1998
- [4] IBM, "HotRod," <http://www-6.ibm.com/jp/autonomic/casestudies/hotrod.html>
- [5] J.S. Chase, D.C. Anderson, P.N. Thakar, A.Vahdat, and R.P.Doyle, "Managing energy and server resources in hosting centers," Symposium on Operating Systems Principles (SOSP'01), pp.103-116, 2001.
- [6] E.pinheiro, R.bianchini, E.V. Carrera, and T.L. Heath, "Dynamic cluster reconfiguration for power and performance," Compilers and Operating systems for Low Power, Kluwer Academic Publishers, 2002
- [7] 揚妻 匡邦, 河野 健二, 岩崎 英哉, 益田 隆司, "需要変化に動的に対応する伸縮自在サーバ群の基本機構," 電子情報通信学会論文誌, J88-D-I(4), 767-779, 2005-04-01
- [8] C. Takano, M. Aida, M. Murata, and M. Imase, "New framework of back diffusion-based autonomous decentralized control and its application to clustering scheme," IEEE Globecom 2010 Workshop on Network of the Future, December 10, 2010.
- [9] C. Takano, M. Aida, M. Murata and M. Imase, "Proposal for autonomous decentralized structure formation based on local interaction and back-diffusion potential," IEICE Transactions on Communications vol. E95-B, no. 5, pp. 1529-1538, 2012.