

M2M サービスシステムにおける ロングポーリング自律負荷分散方式

北上真二^{†, †††} 金子洋介[†] 菅沼拓夫^{††, †††}

機器をインターネットに接続し、様々な遠隔サービスを提供する M2M (Machine-to-Machine) サービスシステムにおいて、安全性と即時性を同時に満足させる通信プロトコルとして、ロングポーリング通信方式が有効である。ロングポーリング通信方式は、機器とサーバ間の通信セッションを一定期間維持させるため、大規模な M2M サービスシステムにおいて、負荷分散のための中間サーバやメタサーバを導入した場合は、そのサーバ自身が性能ボトルネックになるという課題がある。本稿は、ロングポーリング通信方式を適用した M2M サービスシステムにおいて、中間サーバやメタサーバを導入することなく、機器とサーバのみで負荷分散を実現する自律負荷分散方式を提案する。本方式は、ロングポーリング通信方式の特性を活用し、サーバの負荷分散と負荷平準化を実現する。本研究では、提案方式のシミュレーション評価を実施し、提案方式が有効であることを示すと同時に、適切な動作パラメータの条件を導出した。

A Method of Autonomic Load Balancing for Long Polling on M2M Service System

Shinji Kitagami^{†, †††} Yosuke Kaneko[†]
and Takuo Suganuma^{††, †††}

M2M (Machine-to-Machine) service system has started to prevail in which devices are connected directly to the internet and various services are offered. In the M2M service system, the long polling communication method is effective as a communication protocol that satisfies requirements of both security and immediacy simultaneously. It sustains a communication session among devices and servers for a definite period. For this reason, when a medium (or meta) server is introduced into a large-scale M2M service system for load balancing, it might cause a performance bottleneck. In this paper, we propose an autonomic load balancing method to achieve load balancing and load leveling using only devices and servers, without introducing a medium (or meta) server. This study estimates the proposed method by simulation, demonstrates its effectiveness, and derives the conditions of proper operation parameters.

1. はじめに

空調機器や加工機等、ネットワーク機能を有した設備機器や生産機器をインターネットに接続し、様々な遠隔サービスを提供する M2M (Machine-to-Machine) サービスシステムが注目されている[1]-[4]。この M2M サービスシステムにおいては、単機能のセンサから高機能の機器に至るまでの多種多様な機器の低コスト接続性、多数の機器に対する管理容易性が課題となっている[2]。また、機器への実装コストを最小限にしつつ、同時に、外部からの不正操作を防止する安全性と、制御が必要となるタイミングでアプリケーションサーバから機器への接続を行う即時性が求められる[3][4]。

このような M2M サービスシステムのネットワークに対する要件を満足する通信プロトコルとして、ロングポーリング通信方式が提案されている [5][6]。この通信方式では、アプリケーションサーバに対して、機器が要求の有無を問い合わせるためのポーリングを行う。該当機器への要求がなかった場合には、サーバは機器に直ちに応答を返さず、要求が発生するまで応答を待機させる。

ロングポーリング通信方式では、機器をイニシエータとするアウトバウンド通信により、サーバに対する要求の間合せと、その応答による要求の受け取りを行うため、インターネットから機器への不正アクセスを防御できる。また、サーバは、待機させたポーリングの応答で機器への要求を渡すため、サーバから機器への即時性も満足する。しかし、ロングポーリング通信方式は、サーバで通信セッションを待機させるため、サーバのメモリや CPU 等のリソースを多く消費し、サーバの負荷分散が課題である[7][8]。

サーバの負荷分散技術としては、ローカル負荷分散とグローバル負荷分散がよく知られている[9]。しかし、多数のセンサや機器が接続され、高頻度のデータ収集と制御が行われる M2M サービスシステムにおいては、負荷分散のための中間サーバやメタサーバが性能ボトルネックになる[10]。従って、大規模な M2M サービスシステムにおいて、ロングポーリング通信方式を採用するためには、中間サーバやメタサーバを導入せずに、機器とサーバのみで負荷分散を行う自律負荷分散方式の確立が重要となる。

本稿では、機器とサーバのみで負荷分散と負荷平準化を実現するロングポーリング自律負荷分散方式を提案する。本方式は、ロングポーリングのタイムアウト時に、サーバが、CPU 使用率とメモリ消費量を基に負荷指数を算出する。その負荷指数が閾値を超えた場合は、その負荷状態を機器に通知する。その情報に基づいて、機器が次に

[†] 三菱電機株式会社 情報技術総合研究所
Information Technology R&D Center, Mitsubishi Electric Corporation

^{††} 東北大学 サイバーサイエンスセンター
Cyberscience Center, TOHOKU University

^{†††} 東北大学大学院 情報科学研究科
Graduate School of Information Sciences, TOHOKU University

接続するサーバを選択する。また、本方式は、サーバ間で負荷指数を交換することによって、サーバ負荷の平準化を実現する。本方式を採用することによって、接続する機器の台数に合わせてサーバを増やすだけで、システムをスケールアウトすることが可能となる。

本研究では、本提案方式による負荷分散のシミュレーションを行い、提案方式による自律負荷分散が有効であることを示す。また、本方式を最適に動作させるためのパラメータ条件を導出する。

2. 関連研究

インターネットにセンサや機器を接続する M2M サービスシステムにおいては、機器への実装コストを最小限にしつつ、ネットワークの安全性と即時性を確保する必要がある[3]。機器に通信サーバ機能を搭載し、アプリケーションサーバから直接アクセスするダイレクトアクセス通信方式は、サーバから機器に対するアクセスの即時性が確保される反面、外部からの不正操作を防止する安全性を確保するための機器実装にコストがかかるという課題がある。一方、機器からアプリケーションサーバに対して定期的に要求を問い合わせるポーリング通信方式は、外部から不正アクセスされないため、その安全性は確保されるが、即時性はポーリング間隔に依存する[6]。

そのため、安全性と即時性を同時に実現する通信方式として、ロングポーリング通信方式が提案されている[5][6]。ロングポーリング通信方式のプロトコルを図 1 に示す。ロングポーリング通信方式は、一般的なポーリング通信方式と同じく、機器からアプリケーションサーバに要求の有無を問い合わせるが、要求がなかった場合は、サーバ側で要求が発生するまで待機させる。このロングポーリング通信方式を採用した通信プロトコルとして Bayeux[5] がある。また、iopeNet[11]は、Bayeux を採用し、機器遠隔制御のための低コストの設備ネットワークを実現している。

しかし、ロングポーリング通信方式は、サーバで多くの通信セッションを待機させる必要があり、スケーラビリティ確保のための負荷分散が課題である[7]。すなわち、ロングポーリング通信方式を大規模な M2M サービスシステムに適用するためには、サーバの負荷分散が必要となる [8]。

サーバ負荷分散は、大規模なインターネットサービス・システムを運用する上で重要な課題となるため、従来から様々な研究に取り組まれている[9]。サーバ負荷分散としては、負荷分散装

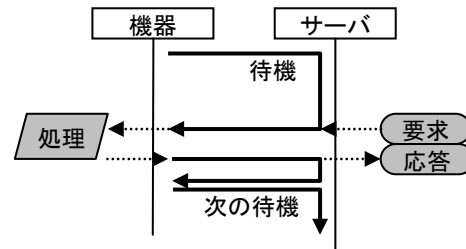
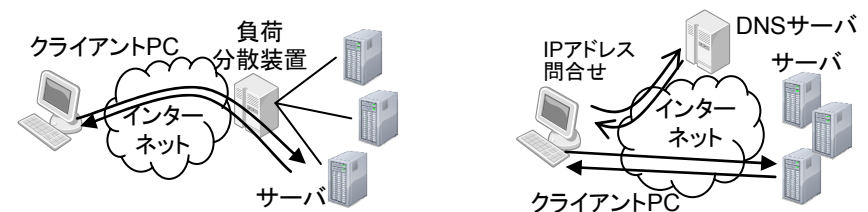


図 1 ロングポーリング通信方式

置によるローカル負荷分散方式や DNS サーバ等のメタサーバを用いたグローバル負荷分散方式が良く知られている。ローカル負荷分散方式とグローバル負荷分散方式の概念を図 2 に示す。ローカル負荷分散方式は、仮想的な IP アドレスを持つ負荷分散装置を設置し、クライアントからの要求をバックエンドのサーバ群に振り分ける。要求をサーバに振り分ける方式としては、サーバの負荷を監視し低負荷のサーバに振り分ける方式、要求のヘッダ情報に従って要求を振り分ける方式等が提案されている。しかし、M2M サービスシステムにおいて、多数のセンサや機器が常時接続されると、全ての通信が負荷分散装置を経由することになり、負荷分散装置がボトルネックになるという問題がある[10]。一方、グローバル負荷分散方式は、クライアントがサーバにアクセスする前に、クライアントがメタサーバに、アクセスすべきサーバを問い合わせる方式である。グローバル負荷分散方式として代表的な DNS ラウンドロビン方式は、インターネットで名前解決に広く用いられているネームサーバに、1 つのホスト名に対して複数の IP アドレスを登録しておき、クライアントからの要求に応じて、異なる IP アドレスを返すことにより、接続するサーバを振り分ける。サーバの負荷状態とは無関係に IP アドレスを返す DNS ラウンドロビン方式では効果的な負荷分散が実現できないため、重み付けラウンドロビン方式、ネームサーバにサーバ評価機構を組み込む方式等が提案されている。しかし、これらの方式は、サーバの負荷を外部から評価するため、負荷分散に遅延が生じるという課題がある[10]。

また、大規模な M2M サービスシステムにおいては、多数のセンサを集約するセンサノードや設備機器等が数十万オーダーの規模で接続されることが想定されるため、その低コスト接続性と管理容易性が課題となる[2]。

従って、大規模な M2M サービスシステムを実現するためには、機器の実装や接続のコストを最小限にすると共に、負荷分散のための中間サーバやメタサーバ等を導入せず、機器とサーバのみで自律的に負荷分散を行う自律負荷分散方式の確立が重要である。



(a)ローカル負荷分散方式

(b)グローバル負荷分散方式

図 2 負荷分散方式

3. ロングポーリング自律負荷分散方式の提案

本章では、負荷分散のための中間サーバやメタサーバ等を導入せずに、機器とサーバのみでロングポーリング通信の負荷分散を実現する、M2M サービスシステムのための自律負荷分散方式を提案する。

3.1 基本アルゴリズム

提案方式の全体構成を図3に示す。提案方式は、サーバに実装される負荷分散機構と負荷平準化機構、および機器に実装されるサーバ選択機構から構成される。負荷分散機構は、サーバのCPU使用率やメモリ消費量から計算される負荷指数と、その上限を示す負荷指数閾値から、サーバの負荷状態を判断し、その結果を機器に通知する。サーバ選択機構は、サーバから通知された負荷情報を元に、次に接続するサーバを決定する。負荷平準化機構は、複数設置されたサーバ間で負荷指数を交換することにより、それぞれのサーバが目標とする負荷指数を決定し、それぞれのサーバの負荷を調整する。

提案方式の処理フローを図4に示す。以下、図中の(a)から(l)にそって、提案方式の基本アルゴリズムを説明する。

- (a) : 機器がサーバに接続する。最初に機器がサーバに接続する際は、ランダム方式やDNSラウンドロビン方式等により、接続するサーバを任意に選択する。
- (b) : サーバは、要求が発生するまで機器からの応答を待機させる。
- (c)~(d) : サーバにおいて、機器に対する要求が発生した場合は、サーバは、(a)の応答として、要求メッセージを機器に返す。
- (e) : 機器は、要求メッセージの内容に応じた処理を実行し、その結果をサーバに返す。
- (f)~(g) : サーバは、アプリケーションサーバに応答を渡すと同時に、機器に処理完了を通知する。
- (h) : 機器は、次の要求の有無を問い合わせるために同じサーバに接続を行う。
- (i) : 要求処理待ちにおいて機器接続タイムアウトが発生した場合は、サーバは、機器に対してタイムアウト通知を行う。
- (j) : サーバは、機器にタイムアウト通知を行う際に、負荷指数を基に自サーバの負荷状況を判断し、その結果を機器へのタイムアウト通知に付加する。
- (k) : サーバから高負荷であることが通知された場合は、機器は、初期接続時(a)と同様に、ランダム方式やDNSラウンドロビン方式等により、任意のサーバを選択して接続する。
- (l) : サーバが高負荷でない場合は、(a)に戻り、機器は同じサーバに接続する。

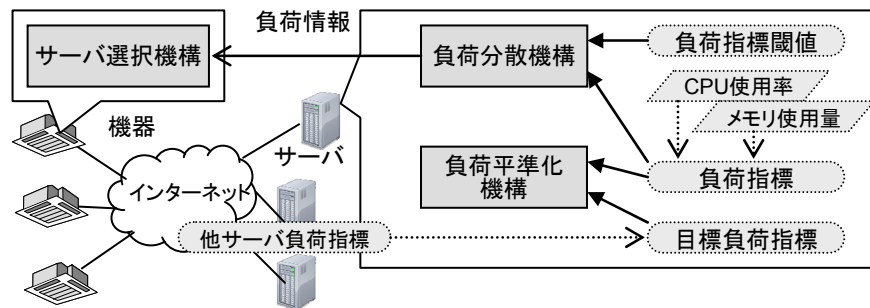


図3 ロングポーリング自律負荷分散方式の全体構成

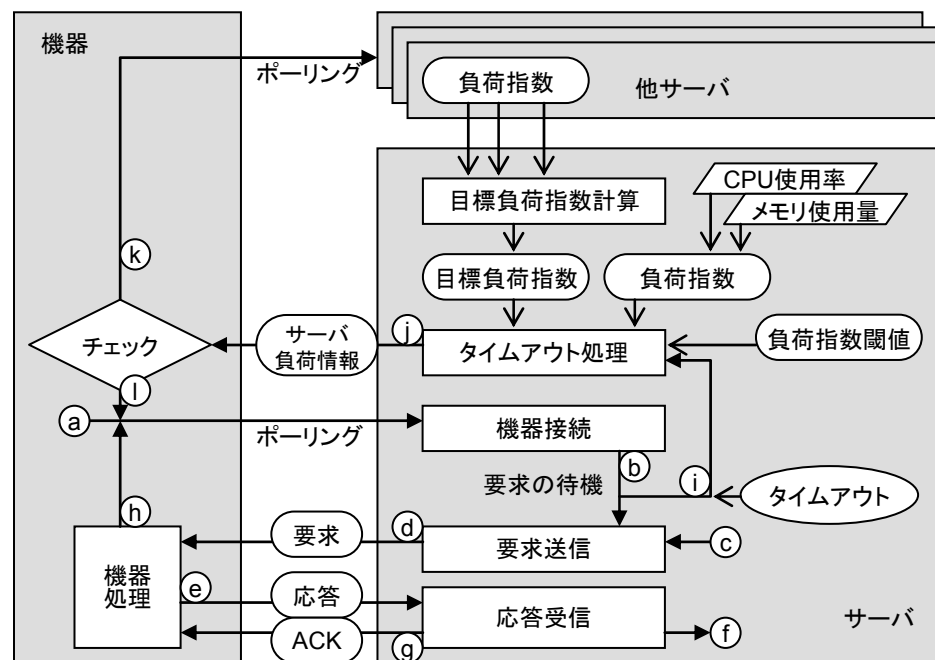


図4 ロングポーリング自律負荷分散方式の処理フロー

なお、(k)の再接続処理において、同じサーバには再接続しないというサーバ選択方式も考えられるが、機器側の実装を最小限にするために、単純なサーバ選択方式とした。

3.2 負荷分散機構

負荷分散機構は、サーバの CPU 使用率と使用量メモリを監視し、それぞれに重みを掛けて合計した値を負荷指数とする。また、各サーバには負荷指数の閾値を設定しておき、負荷指数がこの閾値を超えた場合に、サーバが高負荷であると判断する。

本提案方式のアルゴリズムでは、負荷分散するサーバの台数を N とすると、1 台のサーバ高負荷状態が続いた場合、機器接続のタイムアウト時間毎に、そのサーバへの機器接続数が平均的に $1/N$ になることが期待できる。図 5 に、負荷分散機構によって、高負荷のサーバの負荷が軽減されるイメージを示す。負荷の最大値を L_{max} とすると、機器接続タイムアウト時間において L_{max}/N に近づくように、負荷指数が下がっていく。ただし、負荷指数が負荷指数閾値以下になった時点で、負荷分散処理は中断され、負荷指数は一定となる。

3.3 負荷平準化機構

3.2 節で述べた負荷分散機構は、サーバの負荷が高くなった場合に、そのサーバに接続されている機器を他のサーバへの接続に切り替えることによって、該当サーバの負荷分散を図る。従って、サーバの負荷が低くなった場合であっても、そのサーバに接続される機器の台数は増えず、システム全体として負荷が偏った状態になる。

そこで、この負荷の偏りを是正するために、本方式では、負荷平準化機構を導入する。図 6 に、負荷平準化機構による負荷平準化のイメージを示す。負荷平準化機構は、サーバ間で定期的に現在の負荷指数を交換し、それぞれの負荷指数の平均値を目標負荷指数とする。ここで、サーバ間で負荷指数を交換する間隔を「負荷平準化同期間隔」と呼ぶ。現在の負荷指数が目標負荷指数を上回った場合は、他に負荷の低いサーバが

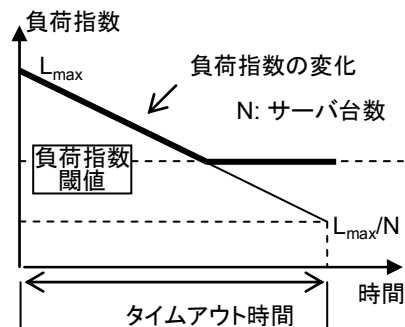


図 5 負荷分散

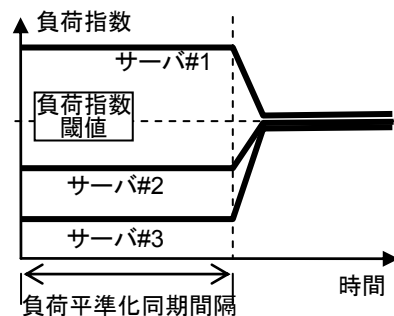


図 6 負荷平準化

存在することを意味するため、負荷平準化機構は、機器に対して接続サーバの再選択を指示する。

4. シミュレーション評価

M2M サービスシステムをモデル化して、提案方式によるサーバ負荷分散のシミュレーションを実施することにより、提案方式の有効性を評価した。

4.1 シミュレーションモデル

提案方式を評価のためのシミュレーションモデルを図 7 に示す。接続する機器の台数は 3,000 台とし、3 台のサーバで負荷分散を行う。シミュレーション開始時点では、各サーバに接続されている機器の台数及びそのタイムアウト時間の残りはランダムとする。サーバ#1 に与える過負荷の形状を図 8 に示す。なお、機器接続タイムアウト時間、サーバの負荷指数閾値、及び負荷平準化同期間隔については、それぞれの実験によって変化させるものとする。なお、各サーバの負荷指数は、機器 1 台が接続された時の負荷指数が 1 になるものとした。

4.2 負荷分散収束性の評価

負荷分散の収束性については、機器がサーバをランダムに選択する従来方式との比較、および機器接続タイムアウト時間をパラメータとした負荷分散の動作について評価を行った。

4.2.1 従来方式

提案方式を従来方式と比較するために、機器がランダムに接続サーバを選択する従来方式のシミュレーションを行った。このシミュレーションでは、機器接続がタイムアウトになった時に、機器が次に接続するサーバをランダムに選択するものとした。従来方式のシミュレーション結果を図 9 に示す。従来方式はサーバの負荷状態に係わらず接続先のサーバがランダムに決定されるため、再接続時においても機器の 1/3 が高負荷のサーバに接続されることになる。その結果、サーバ#1 の負荷が他のサーバに

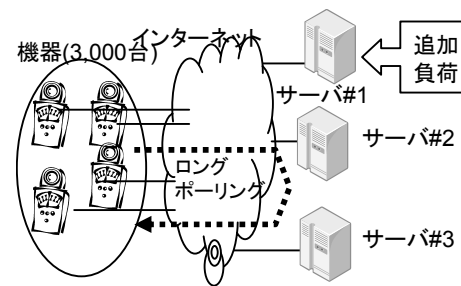


図 7 シミュレーションモデル

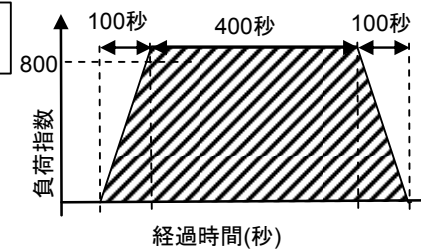


図 8 追加負荷パターン

分散されず、サーバ#1 の高負荷状態が継続している。

4.2.2 提案方式による負荷分散

提案方式の負荷分散機構を動作させた場合のシミュレーションを実施し、負荷分散の収束性を確認した。

図 10 は、機器接続タイムアウト時間を 60 秒、各サーバの負荷指数閾値を 1,400 にした場合の負荷指数の変化および機器接続切り替え回数の変化(上段グラフ)と、各サーバに接続されている機器の数の変化(下段グラフ)のシミュレーション結果である。この結果は、提案方式の負荷分散機構を動作させると、サーバ#1 の負荷指数が閾値(1,400)に達した時点から、サーバ#1 の接続機器数が減少し、他のサーバに負荷が分散されることを示している。この負荷分散処理が行われている約 50 秒間に、他のサーバへの接続切り替えが最大 20 回/秒の頻度で発生している。なお、サーバ#1 の高負荷状態が解消された 700 秒後の時点で、サーバ#1 の負荷指数は 600 まで低下するが、他のサーバよりも負荷指数が低い状態が続き、サーバの負荷の偏りが発生していることが分かる。

図 11 のグラフは、機器接続タイムアウト時間をパラメータとして変化させた時の、サーバ#1 の負荷指数の変化を示している。これは、機器接続タイムアウト時間が短い場合は、負荷指数は閾値内で収まっているが、機器接続タイムアウト時間が長くなると、一時的に負荷指数が閾値を超えることを示している。特に、機器接続タイムアウト時間が 240 秒の場合は、負荷指数が 1,700 近くまで上昇し、負荷指数閾値まで収束するまでに約 140 秒を要することが分かる。

4.2.3 負荷平準化の評価

図 10 に示した通り、サーバ#1 の高負荷状態が解消された後に、その負荷指数が他サーバより低い状態となっている。このサーバ負荷の偏りを解消するための負荷平準化機構の効果を評価するために、負荷分散と負荷平準化を組み合わせたシミュレーションを実施した。ここでは、負荷平準化同期間隔をパラメータとして、サーバ#1 が高負荷になった時の各サーバの負荷指数の変化を確認した。

負荷分散機構と負荷平準化機構を同時に動作させた場合のシミュレーション結果を図 12 に示す。ここで、グラフ(a)(b)(c)は、それぞれ負荷平準化同期間隔が 30 秒、90 秒、240 秒の場合を示している。負荷平準化同期間隔が 30 秒の場合(図 12(a))は、サーバ#1 の負荷指数が閾値(1,400)に達した後、約 1,300 まで低下し、他サーバと同程度の負荷指数となっている。また、サーバ#1 の高負荷状態が解消された後も、負荷指数が低い状態のままとはならず、すべてのサーバの負荷指数が、元の状態に戻ることが分かる。負荷平準化同期間隔が 90 秒の場合(図 12(b))は、同期間隔が 30 秒と同じ傾向を示すが、サーバ#1 の負荷指数が、その閾値を越えて最大 1,600 まで上昇している。負荷平準化同期間隔が 240 秒の場合(図 12(c))は、サーバ#1 の負荷指数が閾値を超えることがなく、サーバの再選択の回数も少なくなる。これらの結果は、

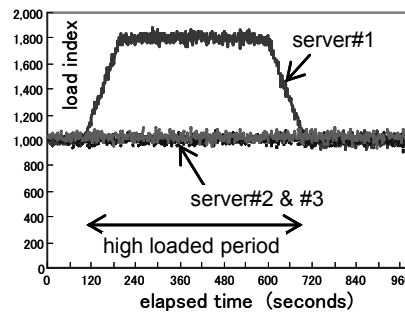


図 9 負荷分散なし

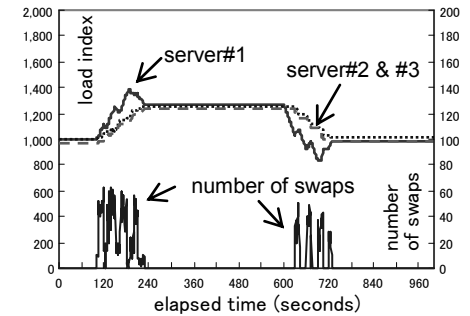


図 12(a) 負荷分散と負荷平準化
(平準化間隔=30 秒)

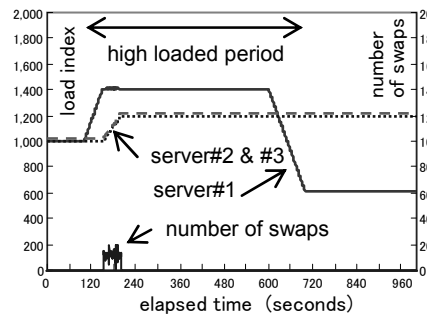


図 10 負荷分散のみ

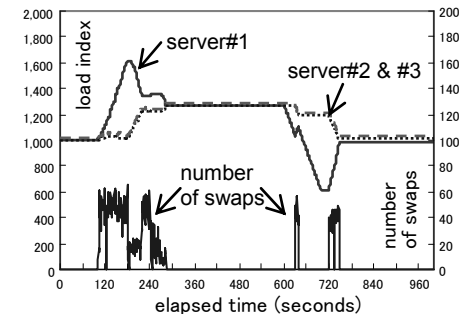


図 12(b) 負荷分散と負荷平準化
(平準化間隔=90 秒)

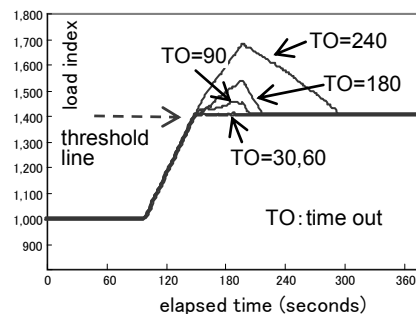


図 11 負荷分散時のオーバーシュート

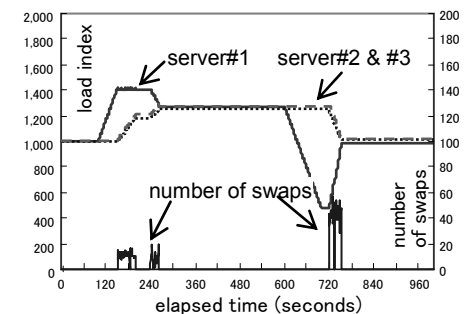


図 12(c) 負荷分散と負荷平準化
(平準化間隔=240 秒)

提案方式の負荷平準化機構を動作させることにより、サーバの負荷の偏りが解消されることを示している。ただし、負荷平準化同期間隔の値によっては、高負荷となったサーバの負荷指数が閾値を超える場合があることが分かった。

5. 考察

5.1 負荷分散

図9に示した通り、機器が接続するサーバをランダムに選択する従来の負荷分散方式では、高負荷となったサーバの負荷は、追加負荷パターンと同じ形状で高負荷状態が継続する。これは、DNS ラウンドロビン方式による負荷分散を適用した場合も同様の傾向を示すと考えてよい。これに対して、提案方式の負荷分散機構を導入した場合は、図10に示した通り、サーバの負荷指数が閾値を超えた時点で、機器の接続が他のサーバに分散され、サーバ#1の負荷指数は閾値を超えないレベルで安定した。これは、負荷分散のための中間サーバやメタサーバを導入することなく、サーバの負荷分散が実現できていることを意味する。

なお、図11に示した通り、本提案方式では、機器接続タイムアウト時間が大きい場合は、一時的に、負荷指数が閾値を超える。機器の総数を M (台)、サーバ台数を N (台)、機器接続タイムアウト時間を T_x (秒) とすると、サーバ#1の負荷指数が閾値を超えた段階で、1秒間に M/T_x 台の機器において接続サーバの再選択が実行される。今回のシミュレーションでは、サーバの選択はランダムに行うものとしたため、 T_x 秒後に $M*(N-1)/N$ 台の機器が別のサーバに再接続される。すなわち、 $(M/N)*(N-1)/(N*T_x)$ 台/秒の割合で、機器がサーバ#1以外のサーバに再接続されることになる。従って、サーバ#1に与えた追加負荷の増加傾きを δ_L (負荷指数/秒) とすると、式(1)に示す条件の場合に、サーバの負荷指数が一時的に閾値を超えてしまうことになる。

$$T_x > (M * (N - 1)) / (N^2 * \delta_L) \quad \dots\dots(1)$$

今回のシミュレーションでは、 $M=3,000$ 、 $N=3$ 、 $\delta_L=800/100=8$ であったため、機器接続タイムアウト時間 T_x が83秒以上の場合に、サーバ#1の負荷指数が一時的に閾値を超える。これは、図11のグラフで示したシミュレーション結果と一致する。

以上のことから、本提案の負荷分散方式を適用する場合は、機器接続タイムアウト時間を、想定される負荷変動の立ち上がり特性に合わせて設定する必要がある。ただし、機器接続タイムアウト時間を短くしすぎると、負荷分散の収束性は高まるが、単位時間あたりの機器からの接続回数が増えることになり、それが負荷増大を招く。

5.2 負荷平準化

図12に示した通り、サーバの過負荷状態が解消された後、サーバの負荷指数に偏りが発生した場合に、本提案の負荷平準化機構が、その負荷偏りを是正することができている。すなわち、この結果は、本提案方式は、負荷を分散するだけでなく、負荷の平準化にも対応できることを示している。

負荷平準化同期間隔の設定値の影響については、負荷平準化同期間隔の値によっては、負荷指数が増大するケースがある。これは、負荷分散機構と負荷平準化機構が競合するためである。たとえば、負荷平準化同期間隔が90秒の場合(図12(b))は、シミュレーション開始から90秒後に、負荷平準化機構が目標負荷指数を設定する。この時点では、全てのサーバの負荷指数は1,000であるため、目標負荷指数も1,000となる。次に、100秒が経過した時点で、サーバ#1の負荷が上がり始め、160秒付近で閾値1,400を超えるため、負荷分散機構がサーバ#1の負荷を他のサーバに分散する。サーバ#1の負荷が分散されたサーバ#2と#3の負荷指数は上昇して1,000を超えるが、その時点で目標負荷指数は1,000であるため、負荷分散機構が機能し、その負荷を他のサーバに分散する。よって、180秒後に次の目標負荷指数が再設定されるまでは、負荷分散機構によるサーバ#1からサーバ#2と#3への負荷移行と、負荷平準化機構によるサーバ#2と#3からサーバ#1への負荷移行が競合していると考えられる。

このような負荷分散と負荷平準化の競合を回避するためには、式(2)に示すように、負荷平準化同期間隔 T_v をシステムの過負荷が継続すると想定される時間以上に設定しておく必要がある。ここで、 D_L は想定される高負荷継続時間(秒)である。

$$T_v > D_L \quad \dots\dots(2)$$

以上で述べたように、本提案方式は、負荷分散のための中間サーバやメタサーバを導入せずに、機器とサーバのみで自律的な負荷分散と負荷平準化が実現できる。ただし、本方式を適切に動作させるためには、負荷の変化状況に合わせて、機器接続タイムアウト値と負荷平準化間隔を設定する必要があることが分かった。なお、負荷の変化を監視することにより、これらのパラメータを動的に変更することも可能である。

5.3 機器側の実装性

図4に示した通り、本方式において、機器側に実装するサーバ選択機構は、ロングポーリングのタイムアウト応答で、接続サーバが高負荷であることが通知された場合に、初期のサーバ選択方式に戻るといった単純な処理である。また、本方式を実装するに当たって、機器側にパラメータ設定等が不要であるため、本提案方式は、リソースが制限される機器にも実装が容易である。

6. まとめ

M2M サービスシステムのための自律負荷分散方式を提案した。本提案方式は、大規模な M2M サービスシステムの通信にロングポーリング通信方式を適用する場合に、機器とサーバのみでサーバの負荷分散と負荷平準化を実現する。本研究では、本提案方式のシミュレーションを実施し、本提案方式の負荷分散と負荷平準化が有効であることを実証した。また、提案方式の設定パラメータとなる機器接続タイムアウト時間と負荷平準化同期間隔の条件を導出した。今後は、サービスの特性と現在の負荷状況に基づき、ロングポーリングのタイムアウト時間等の負荷分散と平準化のためのパラメータを動的に変更することによって、更に負荷分散の自律性を高める研究に取り組む予定である。

参考文献

- [1] D.Boswarthick: "Machine 2 Machine -When the machines start talking-", Mobile World Congress, Barcelona, http://docbox.etsi.org/m2m/Open/Information/M2M_presentation.pdf (2010-2)
- [2] G.Wu, S.Talwar, K. Johnsson, N.Himayat, and K.D.Johnson: "M2M: From Mobile to Embedded Internet", Communications Magazine, IEEE Vol 49 Issue 4, pp36-43 (2011-4)
- [3] ETSI:"Machine-to-Machine communications (M2M); M2M service requirements",ETSI TS 102 689, <http://www.etsi.org/Website/Technologies/M2M.aspx> , v1.1.1(2010-8)
- [4] I.Cha, Y.Shah, A.Schmidt, A.Leicher, and M.Meyerstein;"Trust in M2M communication",IEEE Vehicular Technology Magazine 4 (3) p. 69-75 (2009-9)
- [5] A.Russell:"Bayeux Protocol -- Bayeux 1.0.0", <http://cometd.org/documentation/bayeux/spec> (2007)
- [6] D.Guinard, C.Floerkemeier, S.Sarma, Sanjay;"Cloud Computing, REST and Mashups to Simplify RFID Application Development and Deployment", Proceedings of the 2nd International Workshop on the Web of Things WoT 2011, pp.1-8 (2011-6)
- [7] IETF RFC 6202:"Known Issues and Best Practices for the Use of Long Polling and Streaming in Bidirectional HTTP", <http://trac.tools.ietf.org/html/rfc6202> (2011-4)
- [8] 北上真二: "遠隔機器制御におけるロングポーリング通信の負荷分散に関する考察", 平成 22 年度電気関係学会東海支部連合大会講演論文集 E1-7 (2010-8)
- [9] T.Bourke:"Server load balancing",O'Reilly & Associates,Inc., Sebastopol, CA(2001)
- [10] T.Tonouchi, and Y.Beppu:"Autonomic Management of Scalable Load-Balancing for Ubiquitous Networks",9th Asia-Pacific Network Operations and Management Symposium(2006-9)
- [11] 中尾, 佐藤, 天野:"低リソース・リアルタイム対応設備ネットワーク用通信プロトコル",パナソニック 電工技報, Vol.57 No.2(2009)