

# 集約Webトラフィックにおけるリソース配分モデルの提案

二宮 恵<sup>†</sup> 宇夫 陽次朗<sup>†</sup> 長 健二郎<sup>†</sup>

<sup>†</sup>株式会社 IJJ イノベーションインスティテュート

各 Web サービスは変動するトラフィックに対応するために、十分な余剰リソースを持たせてリソース設計を行う必要がある。しかしこの設計指針を、フラッシュクラウド (FC) のような、大規模ではあるが発生頻度の低いトラフィック変動に対して用いると、平均リソース利用効率が低下しサービスコストが上昇する。FC に対するオーバープロビジョニングを軽減するため、Web トラフィックを集約し、その統計多重効果によって全体として平滑化された Web トラフィックに対して、リソース設計を行うことを提案する。集約システムでは、Web トラフィック全体における個々の FC の影響が相対的に小さくなるため、システム全体における余剰リソースの比率を低下でき、リソース利用効率の向上が可能となる。本稿では、Web サービスの集約におけるトラフィック平滑化効果を検証するために、一般的な Web サービスのトラフィックのモデル化と集約化されたシステムのトラフィックのモデル化を行う。さらに本モデルを利用して集約に基づくリソース利用効率向上を示す。また、集約 Web トラフィックモデルに基づいて、ベンチマークツールを用いて集約 Web トラフィックの生成と評価手法の検証も行う。本モデルは、大量な Web サービスを収容するシステムのリソース設計指針を与えると同時に、効率の良い大規模な Web サービスシステムのアーキテクチャ設計に適用可能である。

## Resource Allocation Model for Aggregated Web Traffic

Megumi Ninomiya<sup>†</sup> Yojiro Uo<sup>†</sup> Kenjiro Cho<sup>†</sup>

<sup>†</sup>IJJ Innovation Institute Inc.

Each web service must be designed with a sufficient resource margin to accommodate various volumes of traffic. However, if a huge resource is spared to deal with a flash crowd, the average resource usage ratio decreases which results in a high service cost. We propose a new approach to solve this problem by consolidating multiple web services and aggregating web traffic to each service. The aggregated traffic is smoothed by a statistical multiplexing effect. Therefore, flash crowds have less influence on the aggregated service than each web service, and excessive over provisioning can be avoided. In this paper, traffic of a general web service and that of an aggregated service are modeled. Then the resource efficiency of both cases is compared using these models. The results show that resource efficiency is improved when web services are aggregated. The aggregated web traffic is generated based on this model. This model indicates how to design resource allocation for a huge system containing a large number of web services, and it will be helpful for the architecture design of an efficient large scale web service system.

### 1 はじめに

不特定多数のユーザがアクセスする Web サービスにおいては、個々のユーザはそれぞれ独立したタイミングでリクエストを行うため、それらのリクエストを受ける Web サービスのトラフィック (Web トラフィック) は一定ではなく揺らぎがある。

Web トラフィックの揺らぎは大きく分けて、周期的増減を繰り返す定常的な揺らぎと、短時間で不規則に発生する揺らぎがある。周期的揺らぎは一般

に利用者の生活パターンなどに由来する。不規則な変動の中には、通常のトラフィックの数倍から数十倍ものトラフィックを発生させる揺らぎもあり、この揺らぎが個々のサービスに与える影響は甚大である。このような大きな揺らぎはフラッシュクラウド (FC) と呼ばれている。FC の例として、2001 年 9 月 11 日の米国主要テレビ局 CNN [1]、MSNBC [2] の Web サイト、アメリカ大統領選挙戦の結果速報 [3]、オバマ大統領就任演説のストーリーミング中継 [4]、FIFA World Cup 2010 期間中の twitter [5]

等におけるクセス集中が報告されている。

Web サービスを提供するシステムでは、変動するリクエストに対してサービスを継続的に提供するために、トラフィックの揺らぎに対して十分な余剰リソースを持たせて設計する必要がある。しかし、FCのようなほとんど発生しない事象に対する揺らぎを考慮に入れると、リソースの利用率が大幅に低下し、結果としてサービスのコストがきわめて高くなってしまふ。

個々の Web サービスにおいては揺らぎをもつ Web トラフィックも、それらを集約していくと統計多重効果により揺らぎの影響が平滑化されることが知られている。FC のように影響の大きな揺らぎも、一定量以上の Web サービスを多重化することで、統計多重効果による平滑化が可能だと考えられる。

FC の影響を平滑化することが可能であれば、Web サービスを集約することにより、それぞれの FC による影響を排除もしくは無視できるようになる。集約型の Web サービスのリソース設計においては、余剰リソースの割合が削減可能となり、サービス全体のリソース利用効率が向上する。

FC を含めた大きな Web トラフィックの変動を平滑化するには、統計多重効果を得られるだけの十分に集約された Web トラフィックが必要になる。さらに、その集約効果を活用するためには、システムとして、十分な数の Web サービスを集約できるリソースが必要である。

既に大規模な Web サービスにおいては、ロードバランサー等によるトラフィックの負荷分散と、並列設置した多数のサーバにより大量のトラフィックを処理するようなシステム構成が用いられている。この大規模 Web サービスのシステム構成は、サーバリソースの拡張性も備えており、集約 Web サービスにおいても利用可能であると考えられる。今後、このようなリソース基盤を用いて、Web サービスの集約を可能にする集約型システムの技術開発が進められていくと予測される。

集約型システムのリソース利用効率を求めめるためには、統計多重効果が得られるだけの、Web トラフィックの集約数を明らかにする必要がある。そ

のため、本稿では Web トラフィックの特徴分析を行ったうえで、集約 Web トラフィックのモデル化とそのモデルに基づいた検証を行う。

また、集約 Web サービスは単体の Web サービスと異なり、そのサービス性能評価を行うためには集約 Web トラフィックを用いたベンチマークテストが必要である。個々の Web サービスのベンチマークにおいて、生成するトラフィック中に突発的なトラフィック変動を含ませることで、より現実的な性能評価が可能になることが Ningfang Mi らベンチマークテスト [6] によって明らかにされている。集約 Web トラフィック中の個々の Web トラフィックにおいても、Web トラフィックモデルに基づいて、FC のようなトラフィックを変動を含むトラフィックを生成することで、集約システムに対して、より現実的な性能評価が可能になると考えられる。そこで、ベンチマークツールを用いて、Web トラフィックモデルに基づいたトラフィックの生成を行い、集約システムにおける評価手法の検証を行う。

## 2 Web トラフィックの特徴分析

Web サービスに対するトラフィックモデルを作成するためにトラフィックの解析結果を示す。複数の Web サーバへのトラフィックモデルを解析した。ここでは、例として北陸先端科学技術大学院大学 (JAIST) の Web サーバへのトラフィックログを用いる。JAIST ではフリーソフトウェアの配布を行う国内最大規模の Web サービスを運用しており、非常に多数の利用者集合を持っているため、より一般的な解析が期待できる。

図 1 に、JAIST の Web サーバへの 2010 年 7 月のある 10 日間のリクエスト数の変化を示す。定常状態では平均 7000req/min 程度のリクエストが来ているが、day4 から day7 にかけてリクエスト数に大きな変動 (平均 20000req/min, 最大 50000req/min) が観測できる。これは Mozilla Firefox のバージョンアップが原因のアクセス集中である。広く使われているソフトウェアの自動バージョンアップ機構は、バージョンアップ版のファイルの存在によって集中的なアクセスを発生させるた

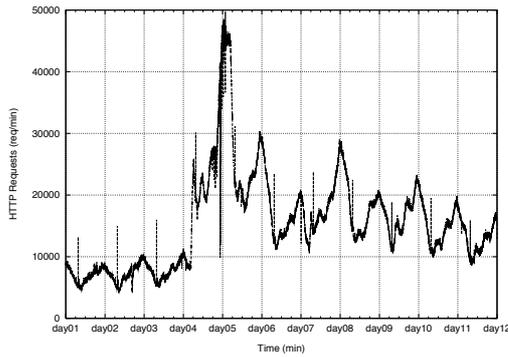


図1 リクエスト数の変動 (10日間)

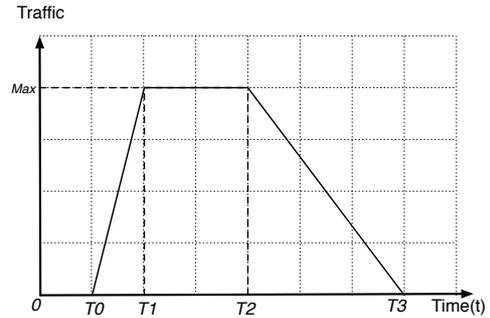


図2 Flash Crowd トラフィックモデル

め、FCの要因の一つである。

### 2.1 定常トラフィック

定常トラフィック状態であっても、短い時間間隔で常に変動する。図1においても、day1-4の間で一定の振幅でのアクセス変動が観測される。定常トラフィックにおけるトラフィックの変動は、急激なアクセス集中による変動とは異なり、その変動速度が緩やかで、またその変動は周期性を持っている。24時間周期や一週間周期の変動パターンは、利用者の生活パターンなどによって生成される変動であることが知られている。

### 2.2 集中トラフィック

図1のday4-day6に見られるような、定常トラフィックの変動の範囲を超えて発生する大きなアクセス集中によるトラフィック変動は、フラッシュクラウド (FC) と呼ばれている。FCの特徴は、ある外部要因 (トリガ) によって発生するため立ち上がり急峻であり、ある一定時間が経過するとFC発生前の状況に復帰することである。たとえば、テレビ放送などでそのリソースに対するURLが放送された場合や、大企業の新しい製品の発表資料の公開、利用者数が多い外部Webサイトからリンクされた場合などがトリガとなる。FCによりWebトラフィックは、通常の数倍から数十倍まで増加することもあり、それぞれのWebサービスにとってきわめてインパクトの大きな現象である。一方で外部要因がなければFCは発生しないため、通常は稀にしか起こらず、その発生は不規則である。

day4-6のアクセス集中においては、瞬間的にき

わめて大きなリクエスト数の増加が2度発生している。最初の増加でリクエスト数は1時間で約3倍に跳ね上がり、続く次の増加において、更に3時間で2倍に急増している。FCの期間全体でみると、リクエスト数の変動幅は5倍に及び、アクセス集中は約2日間続き、その後、アクセス集中前と同様に一日単位の周期的変動に戻っている。アクセス集中前と後で、リクエスト数が約2倍に増加しているのは、day4に実施された、Firefoxトラフィックの振り分けポリシー変更による影響と考えられる。

## 3 Webトラフィックモデル

Webトラフィックの特徴分析を反映する、Webトラフィックのモデルを示す。特徴分析の結果から、FCトラフィックと定常トラフィックの挙動は異なるため、モデル化は定常トラフィックとFCトラフィックに分割して行う。

### 3.1 定常トラフィックの分布モデル

定常トラフィックは小さな揺らぎとユーザーの活動パターンを要因とする中長期周期変動成分を持っている。モデルの単純化のために、これらの変動成分は、ある一定の分布の中に収まるものとする。ここでは、Webサービス*i*の定常トラフィック  $N_i$  を、中心値  $m$ 、分散  $s$  の正規分布内の値のシーケンスとして定義する。

### 3.2 Flash Crowds トラフィック分布モデル

FCのトラフィックシーケンスのモデル化として Ismail Ari らのグループによる定義 [7] を採用

する。Ari らによる FC のトラフィックモデルを図 2 に示す。図 2 では、FC は時刻  $T_0$  に始まり、短い時間 ( $T_1 - T_0$ ) で最大トラフィックに到達する。 $T_1$  から  $T_2$  まで最大トラフィックを維持し続け、 $T_3 - T_2$  時間かけて緩やかに減少して FC 以前の状態に復帰する。

Web サービス  $i$  におけるトラフィック分布の FC に起因する部分を  $F_i$  とする。FC は定常トラフィックとの対比によって規程されるため、定常トラフィックからの増加率を  $I_i$  とすると、 $F_i = I_i \times N_i$  となる。 $I_i$  は FC 発生時に図 2 のような形状の分布で、 $0:1$  の値を持つ分布関数である。

### 3.3 単体 Web サービスのトラフィック分布モデル

定常トラフィックと FC トラフィックはそれぞれ独立したトラフィックであるので、Web トラフィックの分布はの両トラフィック分布の和 (式 1) として表現できる。

$$T_i = N_i + I_i \times N_i \quad (1)$$

## 4 Web サービスの集約化とトラフィック分布モデル

単体の Web サービスのトラフィック分布モデルを用いて、複数の Web サービスを集約した場合のトラフィック分布モデルを定義する。

### 4.1 集約 Web トラフィック分布モデル

複数の Web トラフィックを集約した場合、集約後のトラフィックは個々の Web トラフィックの和となる。そのため、 $K$  サービスを集約した Web トラフィックの分布  $AggT_K$  は次のようになる。

$$AggT_K = \sum_{i=1}^K N_i + \sum_{i=1}^K I_i \times N_i$$

定常トラフィックは常に発生しているため、定常トラフィックの集約トラフィックは、集約されるサービスの数に対して線形に増加する。そのためトラフィックの分布も値が線形に増加し、上式の第一項は  $K \times N$  ( $N$  は各 Web サービスの平均分布) となる。一方で、FC トラフィックは時間的にもサー

ビス的にも離散的に発生し、かつその発生確率は非常に低いため、分布として定常トラフィックのような加算演算が成立しない。加算が発生するのは複数の Web サービスで同時に FC が発生した場合 (コリジョン) である。そのため第二項は  $K$  個のサービスの FC のコリジョンの発生確率の分布を  $C_K$  とした場合に  $N_i \times C_k$  となる。 $C_k$  の分布は各 Web サービスにおける FC の発生確率をパラメータとする分布関数であり、その増加率は集約されるサービスの数の増加より十分に緩やかとなる。理論上は、各 FC トラフィックの Max 値には上限が無く、また確率分布  $C_K$  において最大  $K$  個のコリジョンが発生する可能性もあることから、第二項の値には上限は無い。しかし、このような特殊なケースはきわめて稀であることと、物理的な制約により実際にはこのようなトラフィック増加は起こり得ないと考える。そこで第二項を  $K$  によって規程される定数  $F(K)$  とする。各 Web サービスにおける FC の発生確率を  $0.1\%$  程度だとすると、 $F(K)$  の分布は表 1 のようになる。

$$AggT_K = \sum_{i=1}^K N_i + F(K) \quad (2)$$

### 4.2 FC トラフィックの平滑化

式 2 より、式の第一項の増加率と式の第二項の増加率が異なることがわかる。また、FC の発生確率が十分に小さいことから、第二項の増加は第一項の増加と比べて非常に小さい。これらより、集約数が多くなり、 $\sum_{i=1}^K N_i \gg F(K)$  が成立する  $K$  であれば FC トラフィックの影響は相対的に小さくなる。このことから、適切な数の Web トラフィックを集約することによって、FC トラフィックの影響をキャンセルもしくは一定量以下になると想定できるようになる。

### 4.3 集約 Web におけるリソース配分

Web サービスの定常トラフィック  $N_i$  を処理するために必要とするシステムのリソース量を  $R_i$  とする。独立した  $K$  個の Web サービスを運用する場合、それぞれのシステムで FC 的な変動を考慮したりリソース設計をしなければならない。通常のトラ

サービス集約数	$F(K)$ の中央値	$F(K)$ の最大値	最大値 / $K$
1	0.01	10	10.0
2	0.02	10	5.0
100	1.24	20	0.20
1000	7.5	85	0.08

表1 コリジョンの発生分布 (発生確率=0.1%) (例)

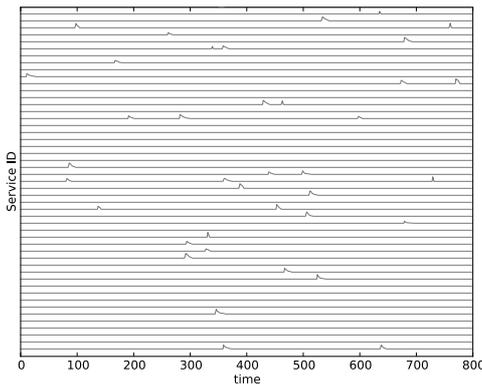


図3 各 Web トラフィックにおける FC (50 サービス分)

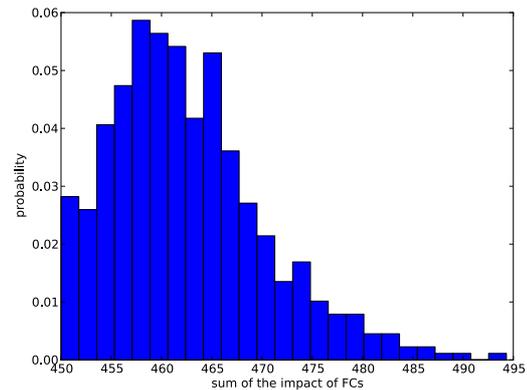


図4 FC トラフィックのインパクト分布

フィックに関する安全係数を  $S(S > 2)$  とした場合、 $K$  個の独立 Web システムの全体のリソース量は  $K \times (R_i \times S)$  となる。全体のリソースの総計は  $K \times S$  倍となる。

一方で、集約することで FC の影響を小さくできる状態においては、 $F(K)$  が集約数  $K$  の増加より変動が小さいことから、トラフィック  $F(K)$  が必要とするリソース量は全体のリソース量に対して十分に小さい。 $K$  個の集約 Web トラフィックが必要とするリソース量は、 $K \times R_i + \max(F(K)) \times R_i$  である。

このことから、集約 Web トラフィックに対してリソース設計を行うことで、個々にリソース設計を行うよりも、全体としてリソース量を削減でき、リソースの利用効率が大幅に向上する可能性がある。

#### 4.4 集約 Web トラフィック分布モデルの検証

集約 Web トラフィック分布モデルの仮定を数値シミュレーションを用いて検証した。図3に

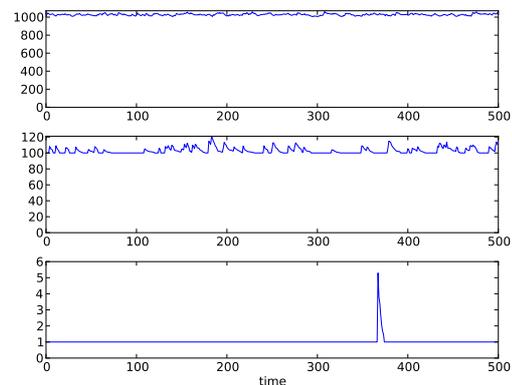


図5 FC の発生と集約によるインパクトの軽減効果 (集約数は上:1000, 中:100, 下:1)

450Web サービスにおける FC の発生パターン例を示す。横軸が時間軸で縦軸に 450 の Web サービス毎の FC の発生状況と、その定常トラフィックからの増加率 (インパクト) を表現している。トラフィック生成のパラメータは [FC の発生確率 0.1%, FC 発生時の最大インパクト 10.0, 試行時間

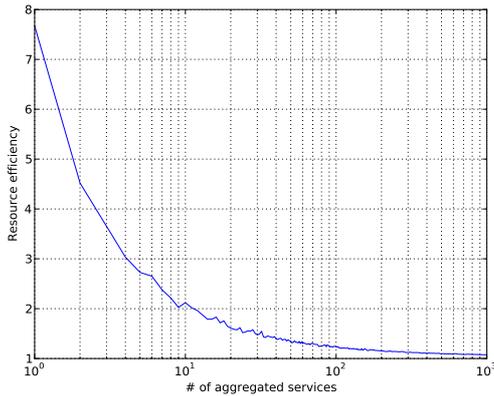


図6 集約 Web のリソース利用効率

600ticks]である。それぞれの水平方向の色つきの部分が図2の形状となっている。図3からは離散的にFCが発生し、まれにコリジョンが発生していることがわかる。このシミュレーション結果を集約した際のヒストグラム分布を図4に示す。450 サービスが集約された場合でも、その分布の中央値は462, 最大値は492程度で抑えられていることがわかる。

さらに集約数  $K$  に応じて時系列のトラフィックパターンがどのように変化するかを数値シミュレーションした。図5に、それぞれ集約数 1, 100, 1000 における、集約された Web トラフィックを示す。トラフィック生成のパラメータは [FC の発生確率 0.1%, FC 発生時の最大インパクト 10.0, 試行時間 500ticks] である。集約数が増えることで、Web トラフィックにおける FC の発生回数は増えるものの、FC トラフィックが全トラフィックに及ぼす影響がきわめて小さくなり、用意しなければならないリソース総数を削減できることがわかる。

そのため、集約数が増加するに従って1つのサービスあたりに用意しなければならないリソースの値は減少する。FC のインパクトを最大で 10 とした場合の各サービスあたり用意しなければならないリソースの最大値の期待値を図6に示す。集約数が1の場合は10倍程度必要であるが集約数が4で4倍程度、10で2倍程度と急速に小さくなっていくことがわかる。

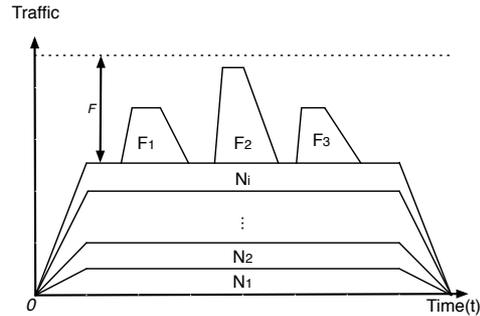


図7 集約 Web トラフィック生成パターン

## 5 集約 Web トラフィックの生成

集約システムの性能評価を行うためには、集約 Web トラフィックを生成する必要がある。そこで、実際のベンチマークツールを用いて個々の Web トラフィックを生成し、集約システムに対して、集約 Web トラフィックの生成を行う。

### 5.1 集約 Web トラフィックの生成手法

集約 Web トラフィックモデルから、定常トラフィックと FC トラフィックに分けて、集約 Web トラフィックを生成する。

図7において、 $N_1, N_2, \dots, N_i$  と  $F_1, F_2, F_3, \dots$  は、個々の Web トラフィックにおける定常トラフィックと FC トラフィックである。これらのトラフィックは、対象とする Web サービスや継続時間トラフィック量が異なるため、それぞれ個別に生成する。

各 Web における FC の発生確率とその FC の最大値は、事前に適切な値を与えるものとし、各 FC は、その発生確率に従って生成し、FC の規模、継続時間、FC 毎に変えて生成する。

### 5.2 ベンチマークツール

SPECweb2009 [8] は、Web サーバの性能評価を行うための標準的なベンチマークツールである。SPECweb を用いることで、Web サーバに対して標準的なトラフィックの生成が可能である。すぐに利用可能なコンテンツを内包していること、そのコンテンツに対して一般的なトラフィックの生成が可

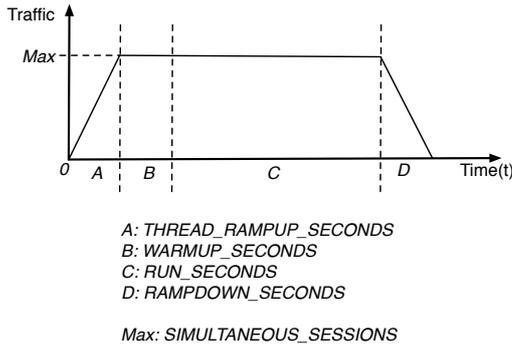


図8 SPECwebトラフィック生成パターン

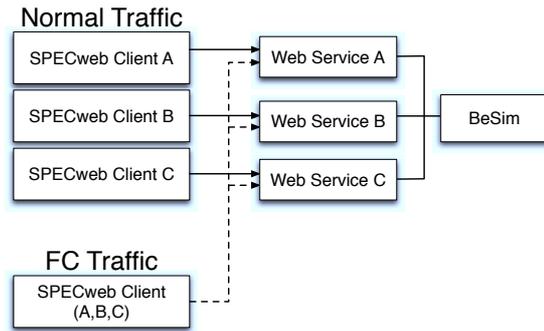


図9 実験構成

能なことから、本実験ではSPECwebを用いることとした。

SPECwebはコンテンツごとに、異なるトラフィックパターンを持っており、Bankはネット銀行、Ecommerceはネットショッピング、Supportはダウンロードサービスの3種類のトラフィックを生成することが出来る。コンテンツは全てdynamicコンテンツを含んでおり、dynamicコンテンツは疑似DB接続をサポートするBeSimプログラムと連携して、それぞれのリクエストの振る舞いが均一にならないようにトラフィックを生成する。

実験においては、この3種類のコンテンツの中から集約Webトラフィックにおいて最も可能性が高いと考えられる、ダウンロードサービス(Support)のコンテンツを用いることとした。

5.3 SPECwebのトラフィックパターン

SPECwebのトラフィック生成パターンは図8のようになり、設定パラメータによってA, B, C, Dの各期間の長さ、Max値を変更することが出来る。

5.4 テストシナリオ

3つのWebサービスに対して、SPECweb2009を用いて集約Webトラフィックを生成を行う。

実験構成は図9のように、定常トラフィックとFCトラフィックを生成するために計4台のSPECweb Client、SPECwebにおいてDBをシミュレートするBeSimを1台、Webサーバ、Webサーバを3台用意した。

今回のテストでは、3つの定常トラフィックと3

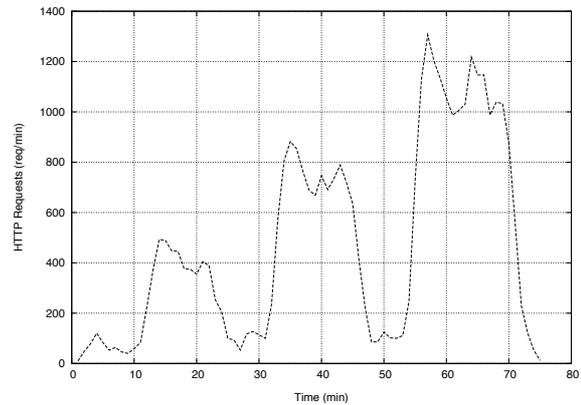


図10 集約Webトラフィック生成

つのFCトラフィックを生成した。定常トラフィックは全て同時に開始し、その後FCトラフィックをF1, F2, F3の順番に生成した。

本来であれば、集約Webトラフィックの生成手法に従い、FCの発生はランダムであるべきが、今回は試行時間の都合により各FCの間は一定時間空け、各FCの継続時間と規模は表2に従って生成した。

5.5 実験結果

実験の結果を図10に示す。

表2より、F1, F2, F3のMAX: SIMULTANEOUS\_SESSIONSは、定常トラフィックの10倍、20倍、30倍となるように設定されており、結果からもFCトラフィックが正しく生成されている。C: RUN\_SECONDSについても、300秒、450秒、600秒とピークの継続時間が延びている。

パラメータ	定常トラフィック			FC トラフィック		
	N1	N2	N3	F1	F2	F3
A: THREAD_RAMPUP_SECONDS	300			150	150	150
B: WARMUP_SECONDS	300			150	150	150
C: RUN_SECONDS	3800			300	450	600
D: RAMPDOWN_SECONDS	300			300	300	300
Max: SIMULTANEOUS_SESSIONS	5			50	100	150

表2 SPECwebのパラメータ設定

このことより、これらのパラメータを変更してFCを発生させることで、よりモデルに近いトラフィックの生成が可能になることが確認できた。

今回の実験では数台規模の実験環境でトラフィック生成を行ったが、Web サービス集約数に合わせて図9のSPECweb Clientの台数を増設することで、より大規模な集約Webトラフィック生成が可能である。

## 6 おわりに

本研究では、Webトラフィックを集約することで、統計多重効果により、個々のFCトラフィックの影響を相対的に小さくし、トラフィック全体として平滑化可能とすることで、集約型システムにおいてリソース利用効率の高いリソース配分モデルを提案した。

集約Webサービスにおける、トラフィック平滑化効果を検証するために、Webトラフィックの特徴分析とトラフィックのモデル化、及び集約Webにおけるトラフィック分布のモデル化を行った。そして、このトラフィック分布モデルに基づいて、数値シミュレーションを行った。その結果から、集約数を増やしていくことで、集約WebトラフィックにおいてFCトラフィックの影響がきわめて小さくなることを明らかにした。

このことから、集約Webトラフィックに対してリソース設計を行うことで、システム全体での余剰リソースを減らし、リソース利用効率が向上することを示せた。

更に、モデルに基づいて集約Webトラフィックの生成手法を提案し、実際にベンチマークを用いて

モデルに基づいたトラフィックの生成が可能であることを示した。

本研究で提案したモデルとベンチマーク手法を用いることで、今後、集約型Webサービスにおけるアーキテクチャの評価が可能になると考えられる。

## 謝辞

本研究にあたりWebサーバのログを提供頂いた北陸先端科学技術大学院大学(JAIST)の宇多仁先生に感謝する。本研究はNEDOのグリーンITプロジェクト(エネルギー利用最適化データセンタ基盤技術の研究開発)の一環として行われた。

## 参考文献

- [1] W. LeFebvre. Cnn.com: Facing a world crisis. Invited Talk, USENIX Annual Technical Conference, June 2002. CNN Internet Technologies.
- [2] V. N. Padmanabhan and K. Kunwadee. The case for cooperative networking. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*, pp. 178–190, Cambridge, MA, USA, March 2002. Springer-Verlag.
- [3] CNET News. Election spurs record traffic to news sites.
- [4] Xconomy. Obama inauguration breaks streaming-media records.
- [5] twitter blog. Big goals, big game, big records.
- [6] N. Mi, G. Casale, L. Cherkasova, and E. Smirni. Injecting realistic burstiness to a

traditional client-server benchmark. In *ICAC '09: Proceedings of the 6th international conference on Autonomic computing*, pp. 149–158, New York, NY, USA, 2009. ACM.

- [7] I. Ari, B. Hong, E. L. Miller, S. A. Brandt, and D. D. E. Long. Modeling, analysis and simulation of flash crowds on the internet. Technical Report UCSC-CRL-03-15, Storage Systems Research Center Jack Baskin School of Engineering University of California, Santa Cruz Santa Cruz, CA, USA, February 2004.
- [8] SPECweb2009. Specweb2009 home page, 2009. <http://www.spec.org/web2009/>.