一斉公開型WWWサーバファームにおける動的サイト構成変更機構

知念賢一[†,☆] 藤部修平[†,☆☆] 西岡宗一郎[†]
山口英[†] 山本平一[†]

WWWサービスの社会的役割が大きくなるにつれて，WWWサーバの処理能力と耐故障性が求められつつある．そこで，1) 処理能力向上のためのクラスタによるWWWサイトの構築，2) 耐故障性を確保したデータセンタのような環境への設置，が実施されている．この2つの手法の延長には，多数のホストを集中設置して複数のWWWサイトが稼働する状況を想定したWWWサイト管理技術が必要となる．本論文では，そのような状況でのサイト構成変更技術，特にコンテンツの同期とサイト構成の一貫性を保ちつつ，動的に構成を変更する機構について述べる．また，この機構を実装したシステムにおいて，コンテンツの一貫性を維持したままサイト構成変更が数十ミリ秒程度で実現したことを示す．

A Dynamic Site Formation Mechanism for Synchronized Publication Type WWW Server Farm

Ken-ichi Chinen,[†] Shuhei Fujibe,[†,☆☆] Soichiro Nishioka,[†]
Suguru Yamaguchi[†] and Heiichi Yamamoto[†]

Content consistency is an important issue in the clustered WWW server site. Every WWW server has to provide same contents at any time. When the site employs dynamic formation, the issue is more important. Because the content consistency maintaining takes a lot of time. It often breaks down the advantage of dynamic formation which is quick expansion of the WWW site performance. Therefore, we developed a novel dynamic site formation mechanism with the content consistency maintaining. By introducing the stand-by state, our mechanism eliminates the lag of synchronization. This paper shows the design and implementation of the formation mechanism. Furthermore, using benchmarkings and field experiment, we report the evaluation of the WWW server farm that employs the formation mechanism.

1. 序　　論

大量のリクエストにさらされるWWW（World-Wide Web）サイトでは複数ホストを用いたクラスタ構成を採用することが一般的になりつつある．また，耐故障性や管理コストの削減等の点から，そのようなWWWサイトはホストを多数設置したデータセンタにおいてサイトを構築する傾向にある．このような背景から，多くのホスト群を用いて複数のWWWサイトを構築する技術が研究・開発され始めた．本論文ではこのようなWWWサーバホストを多数収容して，複数のWWWサイトを形成するホスト集合を**サーバファーム**（server farm）と呼ぶ．また，サーバファームを構成するWWWサーバホストを**要素サーバ**（element server）と呼ぶ．

クラスタによってサイトを構成する理由は，単体のWWWサーバホストでは到達できない性能を必要としたからである．したがって，このような形態では，WWWサーバホストの用いるソフトウェアやハードウェアの単体性能向上だけでなく，ある目標とする性能を引き出すための要素サーバの確保を焦点にしたサイト構築を考えねばならない．サーバファームでは利用した要素サーバの数が料金として支払われるため，所要要素サーバ数の見積りが重要となる．リクエストが処理できない事態を避けるために，多くのサイトではリクエストの多い時間に合わせて要素サーバを確保

† 奈良先端科学技術大学院大学
Nara Institute of Science and Technology
☆ 現在，北陸先端科学技術大学院大学
Presently with Japan Advanced Institute of Science and Technology
☆☆ 現在，株式会社セガ
Presently with SEGA Corporation

する．一方で，リクエストの多い時間は短く，確保した要素サーバの大部分は遊休化して経費を浪費する．

この事態を改善するためには，契約の単位時間を短くして，こまめに契約すればよい．そうすれば，サイト運用者は経費を節約し，ファーム運用者は機材を別の用途に利用できる．このような考えを押し進めて，随時必要な分だけ要素サーバを確保して，従量制課金ともいうべき契約形態への移行が望まれている．このような契約形態では，必要に応じてサイト構成を変更する技術が不可欠である．また，人手を介する契約手段を用いては迅速な構成変更は非常に困難なため，サイト構成を自動的に変更する機構が求められている．本論文では，これを**動的構成変更**と呼ぶ．本研究では，リクエストに応じて動的サイト構成変更を行う機構を設計した．

一方，複数の要素サーバを用いた WWW サイトでは，要素サーバ間のコンテンツの一貫性が必要である．たとえ，構成変更中でもコンテンツの一貫性を維持しなければならない．

これまで，我々は負荷分散装置を利用することでサイト構成の一貫性を維持し，通知を用いた WWW サーバクラスタにおけるコンテンツ一斉公開機構を開発してコンテンツ一貫性を保持してきた[1]．この 2 つを協調して適用することによって，動的なサイト構成変更が可能となる．本論文は，そのようなコンテンツ一斉公開をする WWW サーバクラスタを複数収容したサーバファームを想定した環境での動的なサイト構成機構の設計について述べる．

変更時間を短縮するため，いくつかの工夫を施した．イ) コンテンツの保持機構はキャッシュ上に構築した．キャッシュであるから，全ホストへすべてのコンテンツをコピーする必要はない．したがって，コンテンツの整う時間が短く，ホスト上の所要記憶容量も小さくなる．ロ) サイトへのホスト追加時に，リクエストを振り分けない状態を設けて，その間にコンテンツを供給する．これには 2 つの理由がある．第 1 にコンテンツ一斉公開機構は一定の待機時間を必要とする．第 2 にサイトに追加された直後のホストはコンテンツを上流から転送するため，上流のホストにリクエストが集中する恐れがある．キャッシュを用いても，ある程度集中する可能性が残るため，リクエスト注入に緩衝を設けてリクエストの集中を緩和する．

以下，その動的なサイト構成変更機構の設計と実装，ベンチマークと運用による性能評価について述べる．この機構を用いた WWW サーバファームでは，サイト構成の一貫性だけでなく，コンテンツの一貫性も維持すること，構成変更が高速に処理されることを示す．

2. 関 連 研 究

Appleby ら[2] および吉村ら[3] は WWW サイトをバックエンドとフロントエンドに分け，それぞれにデータベースや WWW サーバを設置する複雑なサーバファームをモデル化している．複雑なサーバや広く一般的なサービスを前提としているため，OS のインストール等のためにリブートする等，サイト構成変更には数百秒の時間を要することが報告されている．

Appleby らや吉村らのどちらもコンテンツの同期の厳密性には触れていない．コンテンツの同期が重要なサイトでは本研究の機構が向いていると思われる．野上ら[4] は本研究に似たキャッシュを中心にしたシステム構成を発表しているが，コンテンツの同期については言及していない．

本論文で設計した機構は構成変更が高速に処理される．これは，本研究はデータベースや CGI-BIN 等のトランザクション処理を対象としていないために実現できたのではないかと，その適応範囲に疑問が残る．ここでは，その適応範囲について説明する．まず，多くの WWW サイトではトランザクション処理は用いられていないため，本機構は容易に導入可能である．そして，トランザクション処理をともなうサイトの多くではトランザクションはリクエスト全体の一部であり，クッキーや URI によってトランザクションを例外的に処理している．したがって，そのようなサイトでもトランザクションを例外的に処理するよう工夫すれば，本機構は導入可能である．以上のことから，多くのサイトで本機構が適応可能だと考える．一方，リクエストの大部分がトランザクションで占められ，かつ大量のリクエストを処理するサイトでは，コンテンツの同期が実現されている可能性は高く，本機構を導入する必要性は低いと考える．

一斉公開型のコンテンツ同期機構の詳細，そして他のコンテンツ同機機構との比較については，文献 1) を参照されたい．

3. 設計と実装

本章では，一斉公開型コンテンツ同期に適した動的サイト構成変更機構の設計について述べる．

3.1 システム構成

本研究で開発したサイト構成変更機構を導入したサーバファームは，多数の要素サーバホストとコンテンツ供給サーバホスト，負荷分散装置と負荷分散装置を制御するサーバ構成制御サーバホスト，そして，負

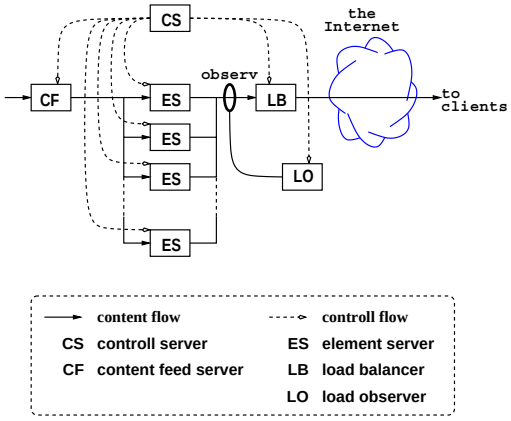

図 1 概念的システム構成
Fig. 1 A conceptual system structure.

荷を観測する負荷観測ホストで構成される（**図 1**）．

コンテンツ供給サーバはコンテンツ作成者から提供されたコンテンツを各要素サーバに供給する．負荷観測ホストは各要素サーバに到着するリクエストを観測してサーバ構成制御サーバに報告する．サーバ構成制御サーバホストは，負荷観測ホストの観測結果からサーバ構成変更の是非を判断して，負荷分散装置とコンテンツ供給サーバや要素サーバを制御する．

今回はサイトの処理能力として特に重要な，要素サーバの台数変更によるサイト規模の変更に焦点を当てる．すなわち，サイト規模の拡大と縮小を実現する．

3.2 要素サーバの状態：スタンバイの導入

要素サーバの追加と解放によってサイトの処理能力を変更する．したがって，原則的には要素サーバは無所属，いずれかのサイトに所属の 2 状態を持つ．

一斉公開方式では，コンテンツの要素サーバへの供給とクライアントへの公開は分離されており，個々の公開前に一時的な待機期間を必要とする．そして，動的に構成を変更する場合には，クライアントから要求されるコンテンツの大部分が要素サーバに保存されることが望ましい．これは，要素サーバが保持していないコンテンツをコンテンツ供給サーバに問い合わせてコンテンツ供給サーバに高負荷を与えることを避けるためである．また，サイトに新たに要素サーバを追加した際には，要素サーバ上のコンテンツを同期するための種々の処理が必要であり，瞬時にクライアントへコンテンツを提供しはじめることは難しい．そこで，サイトに要素サーバを追加する際に，一斉公開の準備のために中間状態を設ける．

以上のように，具体的には要素サーバは 3 つの状態を持つ．まず，どのサイトにも所属していない状態を**アイドル**（idle）と呼ぶ．コンテンツは提供しない

表 1 要素サーバの状態
Table 1 States of element servers.

状態名	サイト所属	コンテンツ供給	リクエスト流入
アイドル	no	no	no
スタンバイ	yes	yes	no
アクティブ	yes	yes	yes

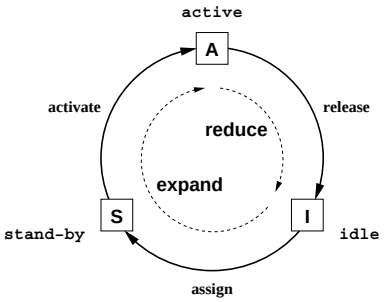

図 2 要素サーバの状態遷移
Fig. 2 The state transition of element server.

ため，コンテンツの供給を受けず，クライアントのリクエストは流入しない．コンテンツを提供する状態を**アクティブ**（active）と呼ぶ．特定のサイトに所属し，コンテンツの供給を受けつつ，クライアントのリクエストが流入される．そのリクエストに応じてコンテンツを提供する．そして，その中間状態を**スタンバイ**（stand-by）と呼ぶ．前述のようにスタンバイはコンテンツ同期のための状態で，サイトに所属してコンテンツの供給を受けるが，リクエストは流入されない．**表 1** にこれらの状態をまとめる．要素サーバはアイドル，スタンバイ，アクティブの順に状態遷移する（**図 2**）．ただし，この手法はスタンバイ状態滞留時間が短い場合には，コンテンツの同期が間に合わない恐れがある．

3.3 サイト構成変更とその判断

サイトを拡大する際には，スタンバイ要素サーバをアクティブにすると同時に新たにアイドル要素サーバをスタンバイとしてサイトに追加する．状態遷移はメモリ操作で実現されることから，その所要時間は数マイクロ秒程度である．また，先のスタンバイ状態滞留時間が短い際の欠点を補うため，サイトごとにスタンバイ要素サーバを常時用意して，スタンバイ状態滞留時間を長時間確保する．

図 3 に要素サーバ 6 台で，アクティブとスタンバイがそれぞれ 3 台と 2 台の場合の拡大と縮小の例を示した．アクティブ 3 台とスタンバイ 2 台の構成が中央に示されている．拡大する際（右に移行）にはスタン

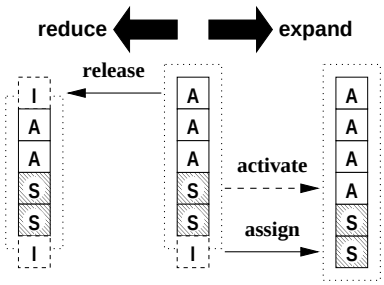

図 3 サイトの拡大と縮小
Fig. 3 The expantion and reduction of WWW site size.

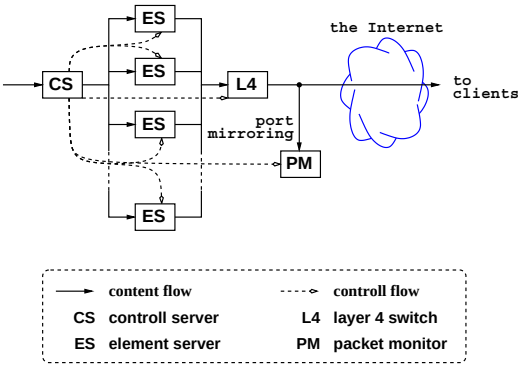

図 4 実装上のシステム構成
Fig. 4 A topology in actual system.

バイ 1 台をアクティブに，アイドル 1 台をスタンバイとして追加する．逆に，縮小する際（左に移行）にはアクティブ 1 台をアイドルにして解放する．

機材を循環的に使用して使用頻度を均一化するために，追加時にはアイドル状態が最も長い要素サーバをスタンバイ状態にする．逆に，解放時には最もアクティブ状態の長い要素サーバを解放対象とする．

サイト規模の拡大と縮小の判断には，2 つの閾値を用いるダブルウォータマーク方式を採用する．1 台あたりの処理能力の限界から，拡大と縮小のためのリクエストレートの閾値を定める．それぞれを**拡大閾値**，**縮小閾値**と呼ぶ．サイトにリクエストが到着するレート（以下，**到着レート**）が，サイトを構成する要素サーバホストの拡大閾値の合計以上になると，サイト規模の拡大として，サイトへ新たにホストを追加する．逆に，到着レートが，サイトを構成するホストの縮小閾値の合計未満になるとホストをサイトから除去する．

n 台のホストで構成されるサイト判断関数 d は，リクエストレート r を引数とする以下の関数で表現できる．戻り値 1 は拡大，-1 は縮小を判断したことを意味する．0 は変更なしを示す．H_i と L_i は各ホストの拡大閾値と縮小閾値である．

$$d(r) = \begin{cases} 1 & \left(r \geq \sum_{i=1}^{n} H_i\right) \\ 0 & \\ -1 & \left(r < \sum_{i=1}^{n} L_i\right) \end{cases} \qquad (1)$$

3.4 キャッシュ効果

一斉公開型 WWW サーバクラスタはコンテンツ同期を目的とし，キャッシュは負荷分散のために設けている．一般的な WWW キャッシュでは再利用度の高いコンテンツを蓄えてヒット率を向上させ，キャッシュの効果である応答時間の短縮，トラヒックの軽減を図る．しかし，新鮮度の視点で見た場合は，コンテンツを長時間保持することは最新でないコンテンツを保持する可能性が高くなる．したがって，一斉公開型ではキャッシュのコンテンツ保持時間は短くして，積極的にコンテンツを破棄あるいは更新確認を行う．同様に，解放された要素サーバのコンテンツは，将来その要素サーバがあるサイトへ追加された時点では，新鮮度が低くなっており，すべて破棄される．よって，要素サーバの解放時にキャッシュ内のコンテンツを解放する必要はない．

文献 5) 等の報告によると，WWW コンテンツは Zipf の法則に沿った傾向を持つ．すなわち，一部のリクエスト頻度の高いコンテンツに関するリクエストが，全体のリクエストの大部分を占める．そして，前述のようにキャッシュのコンテンツは積極的に破棄される．したがって，本方式は小さなキャッシュ領域でも効果を期待できる．

3.5 リクエスト加速度

本論文ではリクエストレート変化の指標として，単位時間あたりのレートの変化を加速度として定める．

$$\text{加速度} = \frac{\text{リクエストレートの変化量}}{\text{時間}} \qquad (2)$$

そして，リクエストレートを req/s，そしてリクエストレート加速度を req/s^2 と表記する．

3.6 実　　装

我々はコンテンツ供給プログラム dnotify [1)] に負荷分散装置を制御する機構を加え，制御サーバの役割を与えた．すなわち，実装上はコンテンツ供給サーバと制御サーバが同じとなった（**図 4**）．当初の構成図（図 1）より構成要素を減らすことができた．この dnotify に対する文献 1) からの変更は約 7,100 行程度である．

負荷分散装置は L4 スイッチを採用した．現在の実装は Extreme Networks 社製 Summit1i と Foundry Networks 社製 ServerIronXL に対応している．dno-

tify は HTTP および TELNET を扱えるため，他の L4 スイッチへの対応も比較的容易である．

要素サーバ上の WWW サーバプログラムは chamomile[6] を採用した．また，所属サイトの変更に合わせて，コンテンツを切り替えて chamomile を再起動する補助プログラムを作成した．

負荷観測は ENMA[7] を採用した．ENMA はパケットモニタによって WWW サーバの負荷を観測するプログラムである．本研究では，ENMA に要素サーバごとの集計機能を付加した．この変更は約 1,900 行程度である．

一般に到着レートには揺らぎがあるため，計測値を直接用いてサイト規模変更の方針を判断すると，むやみに規模を変更して著しいオーバヘッドが生じたり，発散してしまったりする可能性がある．そこで，リクエストレートは移動平均を用いて算出した．その移動平均によるリクエストレートから規模変更の判断を行う．このリクエストレートはコンテンツやリクエスト列の傾向に影響を受けると思われるが，これまで算出方法に関する研究は報告されておらず，適した算出方法は不明である．よって，移動平均の幅は可変とし，経験的な値 30 秒をその既定値とした．

本機構ではサイトごとにスタンバイ要素サーバホストをつねに必要とする．余分な資源を減らすため，スタンバイ要素サーバホストの台数を極力少ない方が望ましい．そこで，スタンバイ状態の要素サーバはサイトごとに 1 台とした．各要素サーバのスタンバイ状態での最低待機時間は可変とし，既定値を 30 秒に定めた．したがって，既定値を用いたサイト拡大は最短で 30 秒に 1 台（0.033 台/秒）となる．これは，要素サーバが 50，100，1000 req/s のリクエストレートを処理できる場合，それぞれ 1.65，3.33，33.3 req/s^2 に相当し，多くのケースに十分対応可能である．また，先に述べた，スタンバイ状態の滞留時間が短い場合の欠点の回避策としても大きく期待できる．

なお，メンテンナンスや障害時には，特定の要素サーバを取り除いたり，自動的に構成変更機構を停止・再開したりすることが必要となる．そのため，それらの操作を行うための WWW による GUI も別に実装した．

4. 評　　価

ベンチマークと運用によって開発した動的サイト構成変更機構を評価した．

4.1 ベンチマーク

設計時のシステム構成に合わせて機材を設置した（**図 5**）．使用した機材とその設定パラメータをそれぞれ**表 2** と**表 3** に示した．各ホストは IBM PC 互換機を用い，負荷分散装置は Extreme Networks 社製 Summit1i である．そして，この節の実験ではコネクションとリクエストは 1 対 1 に対応している．

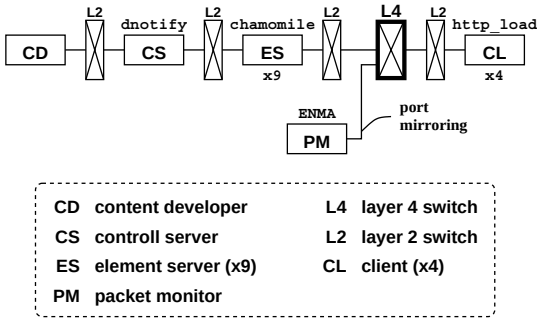

図 5 ベンチマーク実験システム構成
Fig. 5 A system formation in the benchmark testbed.

表 2 ベンチマーク実験機材
Table 2 The specifications of devices in the benchmark.

	OS	CPU GHz	MEM GB	台数
制御サーバ	RedHat Linux7.3	P3 1.0 ×2	1	1
要素サーバ	RedHat Linux8.0	Xeon 2.4	1	9
モニタリングサーバ	RedHat Linux8.0	Xeon 2.4	1	1
クライアント	RedHat Linux8.0	P3 0.866 ×2	0.5	4
負荷分散装置	Extreme Networks 社製 Summit1			1

P3 とは Pentium III を意味する．

表 3 ベンチマーク実験パラメータ
Table 3 The configuration parameters in the benchmark.

グループ	単一あるいは複数	
判断方式	過去 30 秒の移動平均	
処理限界値	100	100%
拡大閾値	70	70%
縮小閾値	30	30%

単位はコネクション到着率 [conn./s]

前述のように要素サーバ 1 台の処理能力が 100 req/s でスタンバイ待機時間が 30 秒であれば，この WWW サーバフォームは 3.33 req/s^2 までのリクエスト加速度を処理可能である．

4.1.1 単体サイト

クライアントでリクエストを発生させ，到着リクエストレートに応じてサイト構成が変化することを確認する．クライアントは http_load を用い，リクエスト発行レートを変化させるよう改造した．

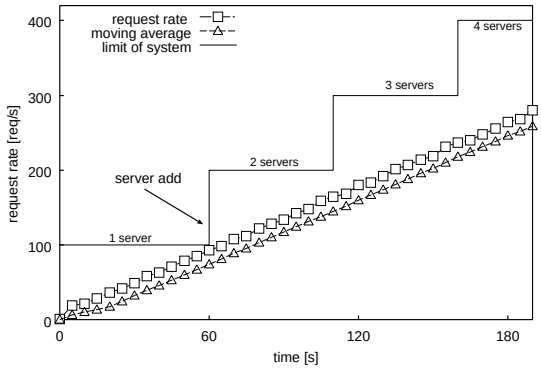

図 6 負荷追従特性（加速度 1.667 [req/s^2]）
Fig. 6 A relationship between the workload and the number of servers (acceleration 1.667 [req/s^2]).

比較的速い拡大の試行として，1 分に 1 台追加される状況を作り出した．6 秒おきにリクエストレートが 10 req/s 増加．すなわち，加速度 1.667 req/s^2 でリクエストを発行して，到着レートにサイト構成が追従することを確認した．この構成変更時にクラアントへのエラー転送や一貫性の喪失は起きなかった．

要素サーバ台数の経時変化を**図 6** に示す．設定に従い，サイトが 100 req/s ごとの処理限界に到達する前に要素サーバが追加されている．そして，その構成変更時間を**図 7** に示す．全体の 95%が 54 ミリ秒以内であり，迅速に構成変更が行われている．

4.1.2 複数サイト

目標とする複数サイトでのサイト構成変更の例として，リクエストレートの変動する 2 サイトの実験を行った．最大 6 台を必要とするリクエスト列 2 つを合成したリクエスト列を要素サーバ 9 台で構築したサーバファームに与えた．与える負荷のリクエスト加速度は 0.416 req/s^2 である．静的なサイト構成管理では，このシナリオには各サイト 6 台，計 12 台の要素サーバが必要となる．

動的なサイト構成管理では，2 つのサイト間でホストを共有して，到着レートに応じて必要なサイトへホストが移った（**図 8**）．すなわち，静的管理では要素サーバが 12 台必要だが，動的管理では計 9 台で同じシナリオを処理した．これは動的なサイト構成変更が，負荷の変動に柔軟に対応できることの確認となる．

4.2 運 用 実 験

現実的な負荷に対して実装したシステムが有効であることを示すために，高校野球インターネット中継において本システムを運用した☆．

図 9 にシステム構成，**表 4** に使用機材の仕様を示す．前節のベンチマーク実験とは負荷分散装置が異なり，

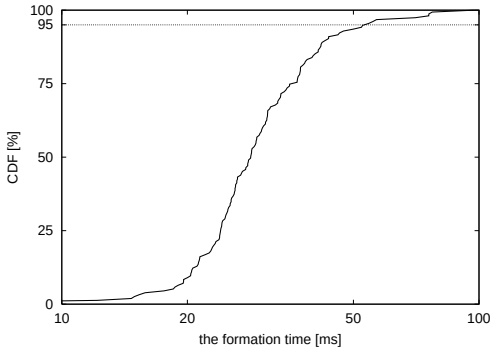

図 7 変更時間の累積分布
Fig. 7 The CDF of the formation time.

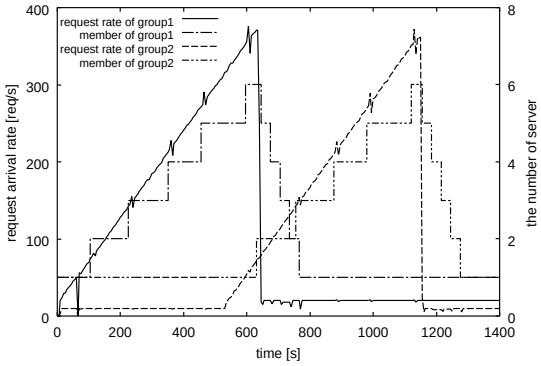

図 8 複数サイトでの台数変化
Fig. 8 The server migration in a two-sites scenario.

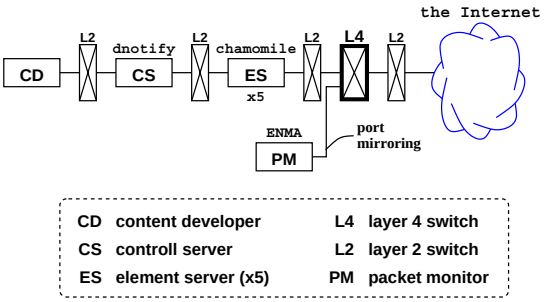

図 9 運用実験システム構成
Fig. 9 The topology of field experiment.

運用実験では Foundry Networks 社製 ServerIronXL を使用した．また，ServerIronXL は負荷情報を外部へ出力できるため，この実験では，負荷監視ホストを設けず，制御サーバは直接 ServerIron XL から負荷情報を得た．設定パラメータを**表 5** に示す．

また，実際の WWW ブラウザによってアクセスさ

☆ 中継イベントの経過，コンテンツ生成等を含めたシステム全体の概要は文献 8) で報告済み．

表 4 運用実験機材
Table 4 The specification of devices in the field experiment.

	OS	CPU (GHz)	MEM (GB)	台数
制御サーバ	Solaris 8	P3 1.13×2	1	1
要素サーバ	RedHat Linux 7.1	P3 1.13×2	1	5
モニタリング	（L4 スイッチの統計情報を利用）			
クライアント	特定不能			不明
負荷分散装置	Foundry Networks 社製 ServerIronXL			1

表 5 運用実験パラメータ
Table 5 The parameters in the field experiment.

グループ	単一	
判断方式	過去 30 秒の移動平均	
処理限界値	500	100%
拡大閾値	350	70%
縮小閾値	150	30%

単位はコネクション到着率 [conn./s]

れていることから，リクエストの転送はパーシステントコネクション（persistent connections）上で行われた．したがって，1 つのコネクションごとに複数リクエストが発行される．このことから，この実験では負荷情報はコネクションを用いた．コネクションはこれまでに議論されたリクエストの集合と見なせることから，リクエスト到着レートに応じたサイト構成変更技法がコネクション到着レートにも適応できる．この実験では，コネクション数の 3.91 倍がリクエスト数に相当した．すなわち，要素サーバ単体の処理限界は 1955 req/s，拡大・縮小閾値はそれぞれ 1369, 587 req/s に相当する．また，本サーバファームの許容加速度は 65.1 req/s^2 である．

図 10 は 1 日のリクエスト総数が最大となった 2002 年 8 月 19 日のリクエストレートとサーバ台数の経時変化を示す．リクエストレートの上昇・下降に従ってサーバ台数が増減し，かつ処理限界を超えずにサイトの処理能力を保持している．**図 11** と**図 12** は増加と減少の箇所をそれぞれ拡大したものである．ミクロな視点でも，限界を超えないようサイトの処理能力を制御している．

次に許容加速度と比較するため加速度を求める．ベンチマークと異なり，実際の運用サイトではリクエスト加速度は変化する．したがって，単純には求められないが，リクエストの移動平均の差分から加速度を近似する．先の 8 月 19 日のログからリクエストの過去 30 秒の移動平均をとり，差分をもって加速度を算出

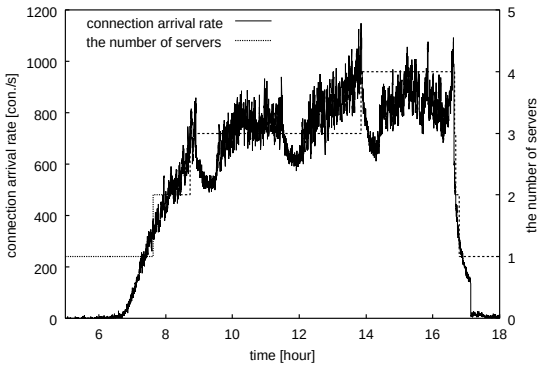

図 10 要素サーバ台数変化
Fig. 10 The transition of the number of element servers (macro view).

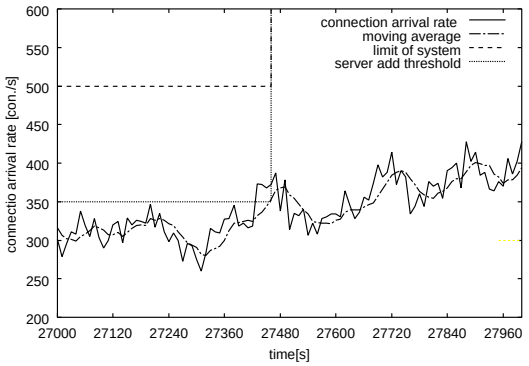

図 11 要素サーバ台数変化（ミクロ；拡大）
Fig. 11 The transition in server expantion (micro view).

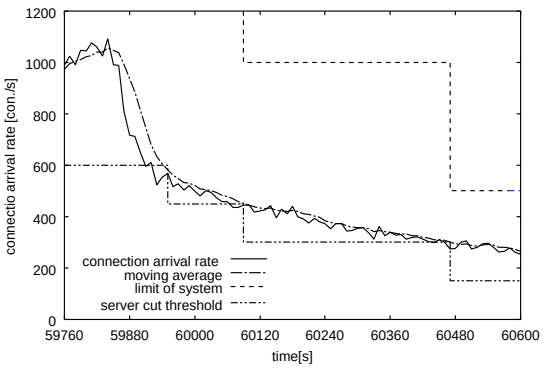

図 12 要素サーバ台数変化（ミクロ；縮小）
Fig. 12 The transition in server reduction (micro view).

した．**図 13** がその累積分布である．算出された加速度は設計限界 65.1 req/s^2 にほぼ収まる．99%tail 値は 23.4 req/s^2 である．このことから，この実験で到着したリクエスト列は設置したサーバファームの能力に収まる規模であったと考える．

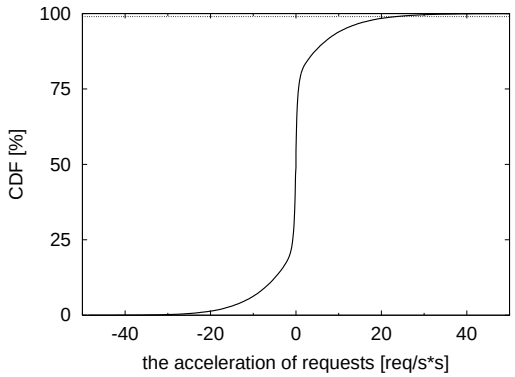

図 13 加速度の累積分布
Fig. 13 The CDF of request acceleration.

5. 今後の課題

本論文では，処理能力の向上を目標にサイト構成変更を要素サーバの台数変更にしぼって論じたが，危機回避の視点で考えると，負荷分散装置やコンテンツ供給サーバホストに関する構成変更も必要であろう．

本研究ではリクエストレートで規模変更を判断したが，システムが稼働中に算出した加速度を用いた規模変更判断の実現を検討すべきであろう．また，ダブルウォータマーク方式以外の制御も検討する必要がある．スタンバイ状態のサーバ台数をファームの状態やリクエストの傾向を用いて算出できるだろう．

5.1 open issues

2001 年 9 月 11 日の CNN の WWW サーバでは，リクエストが 7 分間で 2 倍になったと報告されている[9]．構成変更方式にかかわらず，突然の急激なリクエストレートの変化への対応策の検討が必要であろう．

動的なサイト構成変更についての研究は始まったばかりで，WWW サイト規模とレート変化の関連等，具体的な事例における記録に乏しい．今後は多くの事例における多種の記録を収集しなければならない．

6. 結　　論

本論文は，WWW サーバファーム上に構築された WWW サイトのサーバ台数を動的に変更する機構を紹介した．本機構は負荷に従うだけの単純なフィードバックにとどまらず，コンテンツの一貫性を保持する同期機構と共存し，かつ，その同期機構も制御する．

本機構は通知を用いた一斉公開機構を基盤とした．ホスト構成拡大時にコンテンツ同期中の新規要素サーバへはリクエストを注入しない．一斉公開ができる時間の確保と，コンテンツ供給サーバへのリクエストの集中を緩和するため，一定期間の待機の後，新規要素サーバにリクエストを注入する．これにより，一斉公開の特性を損なわずに動的に要素サーバを追加できる．

ベンチマークと運用実験を通して，その機構の設計と実装の有効性を評価した．本機構を実装したサーバファームでは，サイト構成変更の大部分は 54 ミリ秒以内に収まり，迅速に機能することを示した．

謝辞 運用実験では，多くの視聴者の方に利用していただいた．誌面ながらみなさまに深く感謝を申し上げる．そして，実験環境の構築には朝日放送株式会社，株式会社スマートコネクト，サイバー関西プロジェクト，WIDE project の協力を受けた．貴重な協力に感謝する．chamomile の運用および補助プログラム作成を支援していただいた開発者の河合栄治氏に感謝する．

参 考 文 献

1) 西馬一郎，河合栄治，知念賢一，山口　英，山本平一：通知によるコンテンツ一斉公開機構を用いた WWW クラスタシステム，情報処理学会論文誌，Vol.43, No.11, pp.3439–3447 (2002).
2) Appleby, K., Fakhouri, S., Goldszmidt, G., Fong, L., Kalantar, M., Krishnakumar, S., Pazel, D.P., Pershing, J. and Rochwerger, B.: Oceáno — SLA based management of a computing utility, *IEEE International Symposium on Integrated Network Management* (2001).
3) 吉村　裕，垂井俊明，庄内　亨，河辺　峻，杉江　衛：Web アクセス集中に対応したサーバ自動割当制御，電子情報通信学会論文誌，Vol.J85-D-I, No.9, pp.866–876 (2002).
4) 野上耕介，江頭　徹，桐葉佳明：Web サービスのアプリケーション複製による動的負荷分散制御方式，電子情報通信学会技術研究報告，Vol.2002, No.158, pp.57–62 (2002).
5) Breslau, L., Cao, P., Fan, L., Phillips, G. and Shenker, S.: Web Caching and Zipf-like Distributions: Evidence and Implications, *INFOCOM'99*, pp.126–134 (1999).
6) 河合栄治，白波瀬章，塚田清志，山口　英：商用 WWW サービスの IPv6 への現実的な移行手法，情報処理学会論文誌，Vol.44, No.3, pp.742–750 (2003).
7) 中村　豊，知念賢一，山口　英，砂原秀樹：ENMA: パケットモニタによる WWW サーバの性能解析システムの設計と実装，電子情報通信学会 D-1，Vol.83, No.3, pp.329–338 (2000).
8) 赤藤倫久，藤部修平，三野敦史，沖本忠久：サイバー関西プロジェクト甲子園 2002 (2)，*UNIX MAGAZINE*，pp.181–185, アスキー (2003).
9) LeFebvre, W.N.: CNN.com: Facing a World Crisis, Invited Talk on 2002 USENIX Annual

Technical Conference (2002).

(平成 15 年 4 月 24 日受付)
(平成 15 年 10 月 16 日採録)

知念 賢一（正会員）

1992 年琉球大学工学部卒業．1998 年奈良先端科学技術大学院大学情報科学研究科博士課程修了．同年同研究科助手．2003 年北陸先端科学技術大学院大学情報科学研究科助手．インターネット上のアプリケーション，特にサーバプログラムおよびその構築方法の研究開発に従事．博士（工学）．IEEE，電子情報通信学会，日本ソフトウェア科学会各会員．

藤部 修平

2001 年徳島大学工学部卒業．2003 年奈良先端科学技術大学院大学情報科学研究科博士前期課程修了．同年株式会社セガ入社．在学中，WWW サーバクラスタに関するサーバ構成変更，コンテンツ供給の研究に従事．修士（工学）．

西岡宗一郎

2002 年愛媛大学工学部卒業．同年奈良先端科学技術大学院大学情報科学研究科博士前期課程入学．現在在学中．WWW サーバクラスタ構築，ログ解析に従事．

山口 英（正会員）

1986 年大阪大学基礎工学部卒業．1990 年同大学大学院博士後期課程中退．1990 年同大学情報処理教育センター助手．1992 年奈良先端科学技術大学院大学情報科学センター助手，同助教授．1993 年同大学情報科学研究科助教授．2000 年同教授．インターネットアーキテクチャ，コンピュータセキュリティの研究に従事．工学博士．IEEE，電子情報通信学会各会員．WIDE Project ボードメンバ．

山本 平一

1963 年大阪大学工学部卒業．1965 年同大学大学院修士課程修了．同年日本電信電話公社電気通信研究所入所．1990 年〜1992 年 NTT 理事・無線システム研究所所長．1992 年奈良先端科学技術大学院大学情報科学研究科教授．1997 年〜1998 年（専任），2003 年（兼任）同大学副学長．工学博士．無線通信，移動体通信システム，通信システム，モバイルコンピューティングの研究に従事．電子情報通信学会フェロー．