【別添2】

Hi-sap: 安全で速いウェブサーバ
—世界中のウェブサイトを安全に—

1. 背景

近年，個人でウェブサイトを公開する人が増加している．サイトを設置する場所として，共有型ホスティングサービスに人気が集まっている．これは 1 台のサーバ計算機を多数の利用者で共有する形態のサービスであり，様々な機能を低コストで利用できるという特徴がある．

一方，ホスティングサービスで用いられているウェブサーバには，サーバ組み込みのモジュールとして提供されるプログラム（以下，組み込みモジュール）をセキュアかつ便利に利用できないという問題がある．Weblog や Wiki に代表される動的コンテンツを高速に配信する組み込みインタプリタを用いる場合，サーバを共有する別のユーザからファイルを盗視・改竄される危険性がある．また，ブラウザなどから HTTP 経由でサーバ上のファイルを操作する WebDAV を用いる場合，利用者がサーバにリモートログインしてファイルを直接操作することが困難になる．

これまでにファイル所有者の権限で動作するウェブサーバ，Harache を実現し，組み込みモジュールを安全かつ便利に利用できないという問題を解決した．しかし，Harache はサーバプロセスをセッション毎に終了させるため，組み込みインタプリタによる高速化を十分に活用できていないという問題がある．

2. 目的

本プロジェクトでは，共有型ホスティングサービスのような大規模環境[1]において組み込みモジュールをセキュアかつ便利に利用できないという問題に対し，セキュアかつ高性能なウェブサーバシステム，**Hi-sap**を提案する．

本システムでは，サーバに格納された多数のサイト群をサイトやコンテンツなどのパーティションに分割し，パーティション毎に異なるユーザ権限でサーバプロセスを実行する．これにより，組み込みインタプリタを用いた際の，サイトを共有する別の利用者からの盗視・改竄を防止することができ，かつ，WebDAV とファイルの直接編集を併用することが可能となる．このように，本システムを用いることで，組み込みモジュールをセキュアかつ便利に利用することが可能となる．

そして，パーティション毎に異なる権限で動作するサーバプロセスをプールして使い回すことにより，動的コンテンツの高速配信を実現する．また，ウェブサーバレベルのスケジューラ *Content Access Scheduler* を提案し，サーバ計算機当たりのサイト格納数に対するスケーラビリティを最大化する．

3. 開発の内容

3.1 設計

[1] サーバ当たりに数百から数千サイトを格納するサービスを対象とする．

UNIX 系 OS を対象とした本システムの設計方針は以下の通り.

- サーバ内の高セキュリティ：サーバプロセスをパーティション毎に異なるユーザ権限で実行，セキュアOSと連携
- 動的コンテンツの高速配信：異なるユーザ権限で動作するサーバプロセスをプール
- サーバ計算機当たりのパーティション数に対する高スケーラビリティ：サーバプロセスを動的に生成・終了

Harache とは異なり，サーバプロセスをプールして再利用するため，組み込みインタプリタによる高速化を十分に活用することが可能である.

また，サーバプロセスの動的生成・終了を司るウェブサーバレベルのスケジューラ，Content Access Scheduler を提案する. 基本方針は以下の通り.

- サーバプロセスをリクエストを契機として生成
- 高負荷状態に陥った場合，サーバプロセスを適宜終了

リクエストやコンテンツの特性に応じたスケジューリングを行なうことで，サーバ計算機当たりのパーティション数に対する高スケーラビリティを実現する.

3.2 実装

本システムを SELinux を有効にした Linux OS 上に実装した. 図 1 に示した通り，本システムはフロントエンドで動作する *dispatcher* (Apache 2.0.55 + *mod_hisap*)及び，それぞれ異なるユーザ権限で動作する 1000 個の *worker* (Apache 2.0.55)，worker の起動・終了を司るデーモン *hisapd* からなる.

dispatcher は閲覧者からのリクエストを受け取ると，リクエスト先のパーティションを担当する worker が起動していることを確認する. 起動している場合，担当 worker にリクエストをリバースプロキシする. 起動していない場合，hisapd に起動要請を行なう.

worker 停止の際の Content Access Scheduler のスケジューリングアルゴリズムは，ウェブサーバの性能に甚大な影響を及ぼすスラッシングに注目した. スラッシングが起きると判断する条件は以下の通り.

- スワップイン・アウトがともに発生していること
- メモリ利用率が99%を越えていること

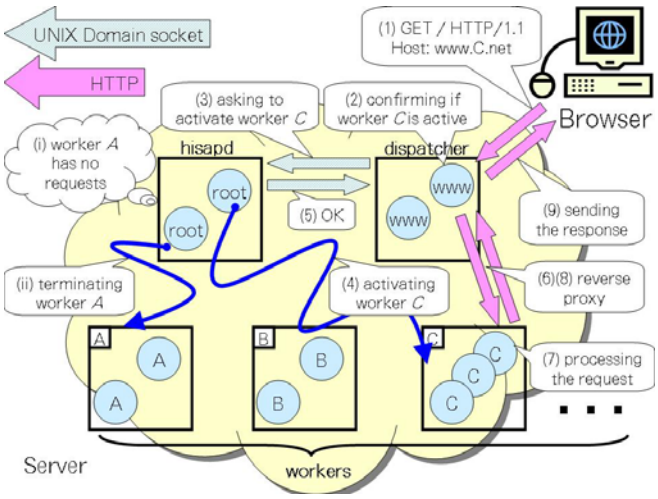

図 1: リクエスト処理手順

3.3 評価

実現したシステムに対して評価実験を行なった．基礎性能評価実験では，通常の Apache 及び *One-to-one*，suEXEC を有効にした Apache を比較として用いた．One-to-one は本システムから mod_hisap 及び hisapd を除いたものであり，システム起動時から全 worker が起動している．リクエスト頻度を変えながら各システム上に設置した 40KB 程度のデータを表示する PHP スクリプトに対してリクエストを送信した．実験結果を図 2 に示す．リバースプロキシによるオーバヘッドのため Apache と比較して平均 28.0%の性能低下が見られた．しかし，One-to-one と比較して平均 1.0%の性能低下にとどまり，mod_hisap 及び hisapd のオーバヘッドは極めて小さい．また，suEXEC と比較して平均 10.2 倍の性能を達成した．

スケーラビリティ評価実験では，One-to-one を比較として用いた．パーティション数を変えながら各システム上の PHP スクリプトに対してリクエストを送信した．実験結果を図 3 に示す．本システムは，One-to-one と比較して終始高いスループットを示し，パーティション数増加に伴うスループットの低下が小さい．一方，One-to-one はパーティション数 600 でメモリ不足により OS がハングアップしてしまい，実験が継続できなかった．

以上の評価実験より，本システムが高性能及び高スケーラビリティを達成していることを確認した．

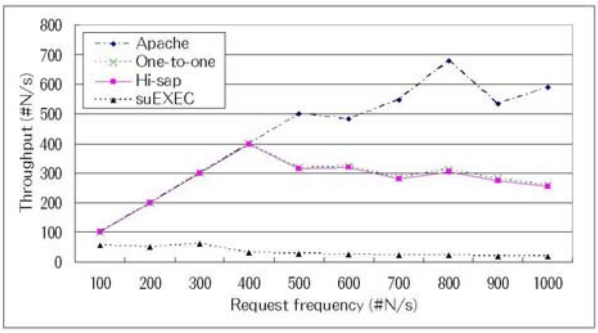

図 2: 基礎性能評価実験

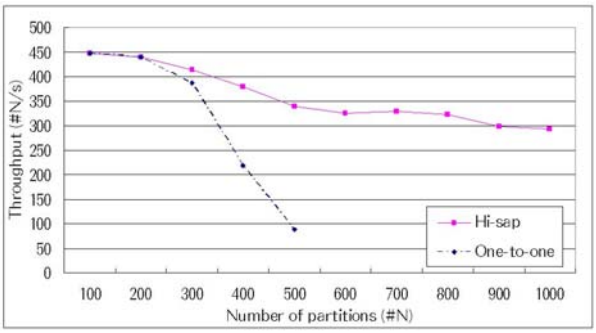

図 3: スケーラビリティ評価実験

4. 従来の技術（または機能）との相違

表 1 にウェブサーバ実現方式の比較を示す．通常のウェブサーバはサーバ内のセキュリティが低い．suEXEC と POSIX ACL を併用することで高セキュリティを達成するが，動的コンテンツの処理性能が低い．サンドボックス及び仮想計算機毎にパーティションを割り当てることで高セキュリティを達成するが，サーバ計算機当たりのサイト格納数に対するスケーラビリティが低い．PHP には言語処理系レベルでサーバ内部のセキュリティを高めるセーフモードという機構が存在するが，言語に依存するため汎用性に欠ける．Apache バージョン 2.0 には perchild という MPM (Multi-Processing Module)が存在し，サイト毎に異なるユーザ・グループ権限でウェブサーバを実行することができ，高セキュリティを実現できる可能性があるが，perchild が安定して動作するとの報告は無く，最新の安定版であるバージョン 2.2 からは削除された．

一方，本システムは全ての項目で評価が高く，万能である．まず，パーティション毎に異なるユーザ権限でサーバプロセスを実行した上でセキュア OS と連携するため，極めて高

いセキュリティを達成する．また，組み込みインタプリタによる高速化を十分活用できるため，動的コンテンツの処理性能が高い．そして，Content Access Scheduler を用いた worker の動的な起動・終了を行なうため，サーバ計算機当たりのサイト格納数に対するスケーラビリティが高い．また，本システムは処理系に依存しない汎用的なセキュリティ機構を提供する．

表 1: ウェブサーバシステム実現方式の比較

	サーバ内部のセキュリティ	動的コンテンツの処理性能	スケーラビリティ	汎用性
通常のウェブサーバ	×	◎	○	○
suEXEC & POSIX ACL	○	×	○	○
サンドボックス／仮想計算機	◎	◎	▲／×	▲
PHP セーフモード	○	◎	○	×
Apache perchild MPM	○	―	▲	○
One-to-one	○	○	▲	○
Harache	○	▲	○	○
Hi-sap	◎	○	○	○

5. 期待される効果

本システムを共有型ホスティングサービスに適用することにより，ウェブサイトを安全かつ高速に運用することが可能となる．また，これまでサーバ外部からの攻撃に比べて軽視されてきたサーバ内部のセキュリティについて広く認識されることが期待できる．

6. 普及（または活用）の見通し

就職に伴い，開発の主体は所属研究室（電気通信大学 情報工学科 中山研究室）の後輩に引き継ぐ予定だが，今後も開発チームの一員として **Hi-sap** の普及に貢献したいと考えている．成果は順次オープンソースソフトウェアとして公開する．共有型ホスティングサービス向けウェブサーバシステムの世界標準を目指したい．

7. 開発者名（所属）

原 大輔
（電気通信大学 大学院電気通信学研究科 情報工学専攻 博士前期課程）

（参考）開発者 URL

http://www.hi-sap.net/, http://chess.cs.uec.ac.jp/~hara-d/