様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２８ 年　 ５ 月 ３０ 日現在

機関番号：１６２０１
研究種目：基盤研究(C)（一般）
研究期間：2013～2015
課題番号：２５３３００８２

研究課題名（和文）クラウド環境で効率的に運用できるキャッシュサーバを用いたＷｅｂシステムの開発

研究課題名（英文）Development of Web system with efficient operation using cache server on Cloud environment

研究代表者
最所　圭三（SAISHO, Keizo）

香川大学・工学部・教授

研究者番号：５０１７０４８６

交付決定額（研究期間全体）：（直接経費）　　3,000,000円

研究成果の概要（和文）：Webサーバへのアクセス集中により十分な応答ができないとき，クラウド上で提供されるキャッシュサーバをアクセス量に応じて用いてサービスを提供するWebシステムを構築するための機能の開発を行った．具体的には，Webサーバの負荷量を監視する負荷監視機能，アクセスの振り分け先を設定する振分先設定機能，キャッシュサーバの起動・停止を行うキャッシュサーバ管理機能を開発した．これらの機能をソフトウェアロードバランサやDNSと組み合わせることで目的のシステムを構築できる．
さらに，キャッシュサーバでのサービスの質を向上させるために，キャッシュサーバからのアクセスをWebサーバで優先的に処理する機能も開発した．

研究成果の概要（英文）：Functions are developed for constructing Web system that provides service using cache servers supplied on the Cloud when access concentration on the Web server causes insufficient response. The number of cache servers is decided depending on the quantity of access.
Specifically, load monitoring function to monitor load of servers, destination setting function to set up destination servers of accesses and cache server management function to boot up and shut down cache servers are developed. By incorporating these functions with software load balancer or DNS, target Web system can be constructed.
Moreover, in order to improve service on cache servers, priority access mechanism for Web server which processes access from them prior to others is also developed.

研究分野：情報学

キーワード：Webサーバ　負荷分散　キャッシュサーバ　ロードバランサ　DNS　負荷予測　優先アクセス　クラウド

１．研究開始当初の背景

Web サーバへのアクセス量は常に変動しており，テレビで取り上げられるなどのイベントが発生すると急激に増加する．これに対応するために，同一のサービスを行うミラーサーバを用いて Web サーバのサービス能力を向上させる方法が一般的である．しかし，この方法では，ピークに合わせてミラーサーバを用意してしまうと，通常時はほとんどがアイドル状態になりコストパフォーマンスが悪くなる．一方，コストパフォーマンスを下げないために少なめに用意すると，ピーク時に十分にサービスができずユーザにストレスを感じさせるようになるという問題がある．この問題は，サーバの処理能力に起因するのであり，ネットワーク環境が向上しても解決できない．一方で，Web サーバのサービス能力を向上させるのではなく，Web サーバへのアクセス量を減らす方法として，キャッシュサーバの一種であるプロクシサーバを用いる方法がある．プロクシサーバは Web サーバとクライアントの中間に位置し，クライアントはプロクシサーバを経由して Web サーバにアクセスする．プロクシサーバでは，Web サーバにアクセスして得られたリソースをキャッシュしておき，キャッシュされたリソースへのアクセスがあれば，Web サーバにアクセスしないでキャッシュしたリソースをクライアントに返す．しかし，プロクシサーバを用いるかどうかはクライアントに依存するので，確実に Web サーバの負荷を軽減する保証はない．

２．研究の目的

Web サーバとキャッシュサーバを連携することにより，アクセス量に応じたサービス能力を最適なキャッシュサーバ数で提供する Web システムの構築を目的とする．

図 1 に示すように，キャッシュサーバがクラウドで提供され，それらを動的に参加・離脱させることにより，コストパフォーマンスの高い Web システムの開発を目指す．提案するシステムでは，通常は図 1 の上部に示すように，Web サーバのみでサービスを行っている．Web サーバへのアクセスが増加し自分で処理できなくなると，図 1 の下部に示すようにキャッシュサーバを立ち上げ，クライアントからのアクセスをキャッシュサーバに向くようにする．逆にアクセスが減少した場合は，キャッシュサーバにアクセスが向かないようにして停止させる．キャッシュサーバが不要になれば，図 1 の上部の状態に戻る．

一方，キャッシュサーバ上のキャッシュの更新が滞ってしまうと，クライアントに対するキャッシュサーバからの応答が遅くなり，サービスの品質が落ちてしまう．このため，キャッシュサーバからのアクセスを優先して処理する機能を Web サーバに追加することで，キャッシュサーバでの応答を高速化し，サービスの品質を向上させる．

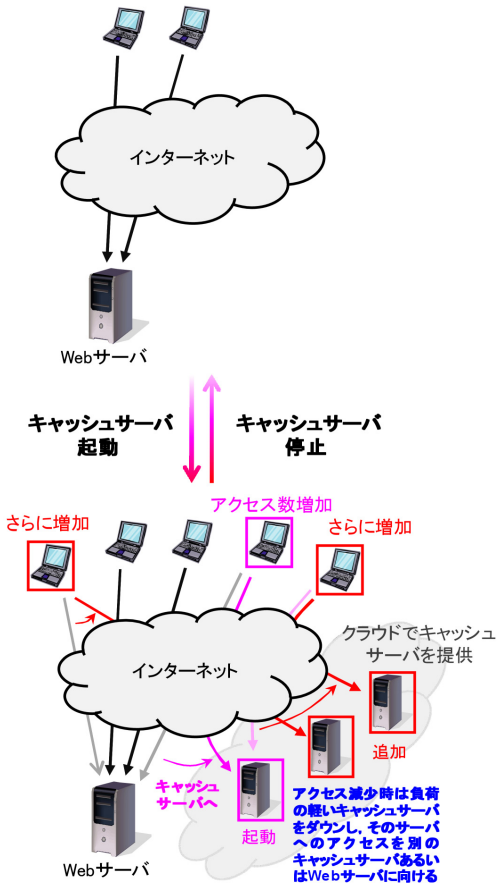

図 1．提案システムの概要

３．研究の方法

本研究で購入した実験用のホストサーバとネットワークスイッチを用いて仮想的なクラウド環境を構築し，その上で Web サーバ，キャッシュサーバを仮想マシンとして動作させる．これらのサーバを用いて以下に示す機能の開発と評価を行う．

Web サーバ側の機能

a. キャッシュサーバ数調整機能：キャッシュサーバの負荷状況を把握およびキャッシュサーバを起動・停止を指示する．
b. アクセスの振分機能：キャッシュサーバへのアクセスの振り分けを行う．
c. キャッシュ更新優先機構：キャッシュサーバからのアクセスを優先する．

キャッシュサーバ側の機能

a. 負荷状況通知機能：Web サーバ側の機能の a に対応する．
b. 過剰アクセスの振分機能：負荷バランスのために過剰アクセスを他のサーバに回す．
c. キャッシュ管理機能：キャッシュを破棄や更新を行う．

これらの機能を Web サーバプログラムである Apache と既存のソフトウェアロードバランサを用いて実装する．Web サーバの a と b の機能を実現するために，既存のソフトウェアロードバランサに，サーバの負荷量を監視する負荷監視機能，アクセスの振分先を設定する振分先設定機能，キャッシュサーバ

の起動・停止を行うキャッシュサーバ管理機能を追加する（図 2）．図中では，提案システムの Web サーバをリソースのオリジナルを保持していることからオリジンサーバと記述している．DNS を用いたアクセスの振分機能の開発も行う．キャッシュサーバの b の機能については Web サーバ側のアクセスの振分機能でカバーする．

Web サーバの c の機能については，本研究以前に開発していた NAP-Web と呼ぶ機構に機能追加することで実現する．キャッシュサーバの c の機能については，検討段階で留まっている．

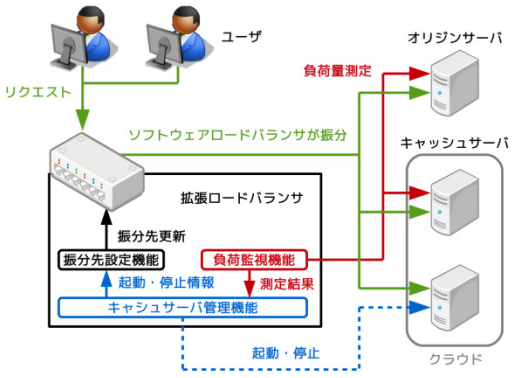

図 2．提案システムの構成

4．研究成果

(1) 負荷監視機能に関する成果

Web サーバの負荷量として適切なパラメタとその取得方法について検討し，それらを実験的に決定した．

本研究で用いるサーバプログラムである Apache では，複数の Web サーバプロセスを起動し，それらに 1 つずつリクエストを順に処理させる prefork モードがあり，リクエスト量に応じて使用するプロセス数を増減させている．負荷量を示すパラメタとして，一般的に用いられる CPU 使用率と Web サーバが用いる最大プロセス数（MaxClients）に対する現在の稼働しているプロセス数の割合（稼働率）を検討の対象とした．取得方法として，Web サーバ上で負荷監視用の外部プログラムを用いる方法と，サーバプロセスの状態を Apache が起動してからの平均 CPU 使用率を提供する server-status ページを用いる方法を検討の対象とした．図 3 と図 4 に，MaxClients を 200 とし，徐々に同時アクセス数を増やし，上限に達すると終了する実験を行った結果を示す．実験結果から，

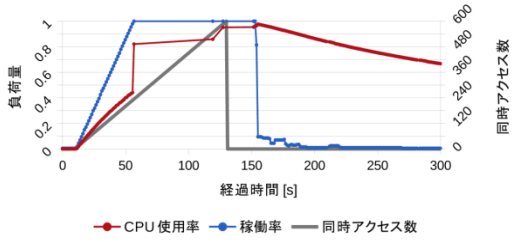

図 3．server-status による測定結果

server-status ページを用いる方法の稼働率が同時リクエスト数に追従して増加していることが分かったので，server-status ページを用いて得られる稼働率をサーバの負荷量として採用することにした．

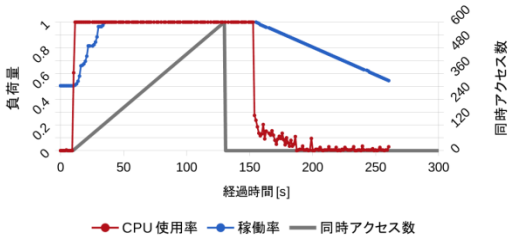

図 4．外部プログラムによる測定結果

(2) 振分先設定機能に関する成果

ロードバランサを用い，キャッシュサーバへの振り分けそのものはロードバランサの機能をそのまま用い，振分先の設定を行う機能を開発した．ロードバランサとして，L4 レベルのソフトウェアロードバランサである IPVS を用いた．

負荷監視機能で収集したオリジンサーバおよび起動中のキャッシュサーバの負荷量をリアルタイムで収集し，その平均値（平均負荷量）が上限（Th_{high}）を上回れば新たなキャッシュサーバへの振り分けを行うように設定し，逆に下限（Th_{low}）を下回ればキャッシュサーバを 1 つ選び，振り分けを止めるように設定する．この機能を実装し，オリジンサーバと 2 台のキャッシュサーバ実験を行った．この実験を行う段階では，キャッシュサーバ管理機構が未実装だったため，全て起動している状態で実験を行った．さらに，これ以降の実験でも同様であるが，キャッシュ管理機構も未実装なためミラーサーバで代用した．各サーバの MaxClients を 200，Th_{high} を 0.8，Th_{low} を 0.2 に設定し，同時実行数を段階的に増やし，そのあと段階的に減らす実験を行った．実験結果を図 5 に示す．

図中の①と②の時点で 1 台目と 2 台目のキャッシュサーバが振分先に追加されたことによって平均負荷量が下がっている．また，③の時点で 2 台のキャッシュサーバへの振り分けが連続して止められた．この結果から，負荷監視機能で負荷量を把握できており，負荷量に基づいてキャッシュサーバへの振り分けが行われていることを確認できた．

次に Th_{low} の最適値について検討した．

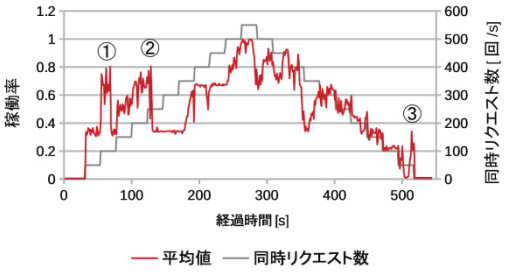

図 5．Th_{low} を固定にした実験結果

振分先を n から $n-1$ に減らしたときの減少した場合の平均稼働率は，$n/(n-1)$ 倍になると考えられる．従って，現時点の平均稼働率を $n/(n-1)$ 倍にしても Th_{high} よりも小さければ，振分先を 1 つ減らしても平均稼働率は理論上 Th_{high} 以下になる．この考えに基づき Th_{low} を以下の式で決定することにした．

$$Th_{low} = Th_{high} \times \frac{n-1}{n} - m$$

m は稼働率の揺らぎに過剰に反応しないためのマージンである．

結果は省略するが，キャッシュサーバ台数を 9 台にしたことを除き，前の実験と同じ条件で行った．その結果，平均稼働率に応じて振り分ける台数を増減できたが，稼働率の揺らぎが原因で同時アクセス数が変わらなくてもキャッシュサーバへの振り分けの追加と削除が繰り返される現象が発生した．

これを解決するために，平均稼働率の過去 n 秒間の平均を用いることにした．n の値を大きくすれば揺らぎの影響を小さくなるが負荷の増減に対する感度が悪くなり，小さくすれば揺らぎの影響を十分に小さくすることができないので，スケールアウト（振分先の追加）する場合とスケールイン（振分先の削除）する場合で n の値を変えることにした．

- スケールアウト(SO)：負荷の急激な増加を素早く反映するために n を小さくする．
- スケールイン(SI)：不必要な停止と起動を押さえるために n を大きくする．

この機能を実装し，先の実験と同じ条件で行った実験結果を図 6 と図 7 に示す．SO と SI の時の n の値を図 6 で 2 と 4，図 7 で 4 と 8 に設定している．図 6 では振り分けの開始，停止の繰り返しが発生しているが，図 7 では発生していない．しかし，スケールアウト時

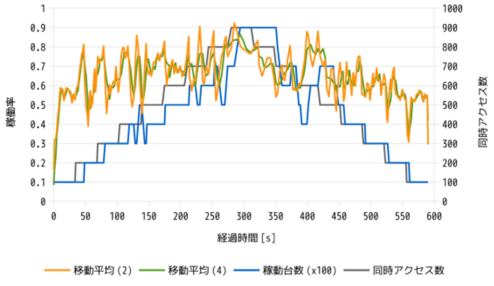

図 6. スケールイン(2)，スケールアウト(4)

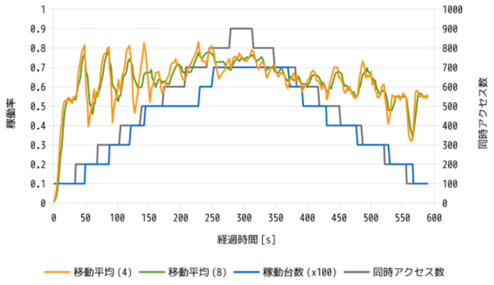

図 7. スケールイン(4)，スケールアウト(8)

の応答性が図 7 の方が少し悪くなった．

図 8 に SO と SI の時の値を 4 と 8 に設定したとき（Moving Average (MA)）と振分先数を固定にしたとき（n Servers (nS)，n は振分先の数）のクライアントへの応答時間を示す．同時アクセス数が一定になった時の MA の応答時間は 7S とほぼ同じ値になっている．応答時間の最大値も 7S の最大値とほぼ同じ値になっている．一方，5S では途中から応答が返らなくなっており，振分先が 5 では不十分だったことが分かる．

以上の結果から，振分先設定機能が最適な割当数を設定できることを確認できた．

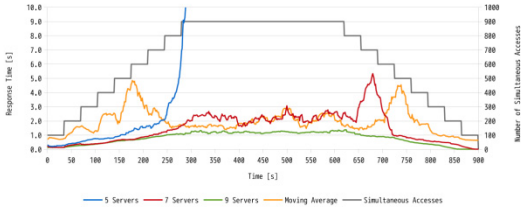

図 8. 応答時間の比較

(3) キャッシュサーバ管理機能に関する成果

キャッシュサーバ管理機能は負荷監視機能から負荷量を受け取り，負荷量が Th_{high} を超えていればキャッシュサーバを起動し，Th_{low} を下回っていれば停止を行う機能である．起動・停止を行った場合は，振分先設定機能にそのことを伝える．本研究では，キャッシュサーバの起動・停止を行うために libvirt と呼ばれる仮想サーバ管理用のライブラリを用いて実装した．キャッシュサーバの起動を開始してから実際にサービスが可能となるまでの遅延や振り分けを停止してから処理中のリクエストを全て処理するまでの遅延が存在するため，これらを考慮して振分先の登録・削除を行う必要がある．実験により，これらの遅延を調査したところ，構築した仮装クラウド環境では両方とも 30 秒であった．この結果から，キャッシュサーバの起動開始から 30 秒後に振分先設定機能に起動したことを伝え，停止するキャッシュサーバを振分先設定機能に伝えてから 30 秒後に停止するようにした．

これらの機能を組み込み，振分先設定機能の評価実験と同様の実験を行った．図 9 に結果を示す．キャッシュサーバが少ない間は起動したキャッシュサーバがサービス可能になる遅延の影響で平均稼働率が Th_{high} を超

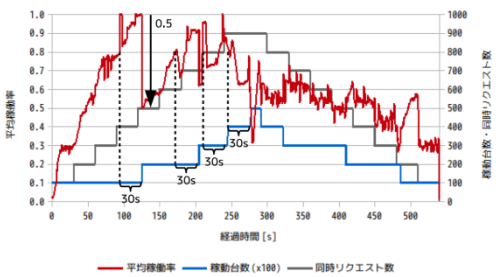

図 9. キャッシュサーバ管理機能の適用結果

える期間が長くなっているが，それ以外では平均稼働率を十分に下げることができている．さらに，キャッシュサーバの起動台数が期待した数より少なくなっていた．この結果を受け，平均稼働率が急激に上っているときに複数のキャッシュサーバを同時に起動するようにしたが，多少の改善はあったものの，根本的な解決にはなっていなかった．

この問題を解決するために，現在の平均稼働率と 10 秒前の平均稼働率を用いて 30 秒後の平均稼働率を予測し，その値が Th_{high} を超えていればキャッシュサーバを起動するようにした．しかし，キャッシュサーバがサービス開始直後に停止される現象が現れた．そこで，スケールアウト後に一定時間スケールインしないようにした．これら改良を行ったキャッシュサーバ管理機能を用いて行った実験の結果を図 10 に示す．平均稼働率は Th_{high} 以下で推移するようになったが，スケールインで支障が出た．この問題は残ったが，平均稼働率を予測することが有効であることを確認できた．

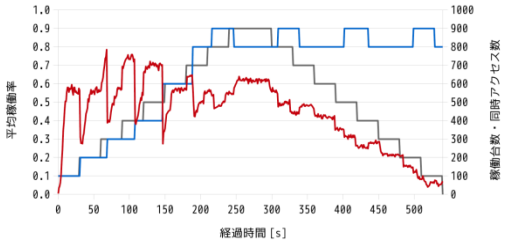

図 10. 予測機能の効果

(4) DNS を用いた振分機構に関する成果

ロードバランサを用いる方法では，ロードバランサそのものがボトルネックになることが考えられる．そこで，DNS ラウンドロビンを用いて負荷を分散する機構を開発することにした．

ロードバランサ用の振分先設定機能を，DNS に対し，キャッシュサーバの起動（停止）の際にはそのキャッシュサーバの IP アドレスを DNS サーバに追加（削除）するように修正することで開発した．DNS サーバで行った変更が，DNS キャッシュが原因でクライアントには遅れて反映されることがある．この時間を考慮して DNS サーバへの追加・削除，キャッシュサーバの起動・停止を行うようにした．オリジンサーバと 5 台のキャッシュサーバに対し，最大同時アクセス数を 500，Th_{high} を 0.8 にしてロードバランサを用いたときと同様の実験を行った．図 11 に結果を示す．平均稼働率が 0.8 の時にキャッシュサーバを起動しているが，サービス可能になってから DNS サーバに登録するため，平均稼働率は 1 まで上がり(①)，そのあと減少している(②)．同時アクセス数の減少とともに平均稼働率も減少した(③)．ロードバランサを用いる方法よりも平均稼働率の変動は大きくなっているが，同時アクセス数に応じてキャッシュサーバの起動・停止が行われ，それらにアクセスを振り分けることができている．

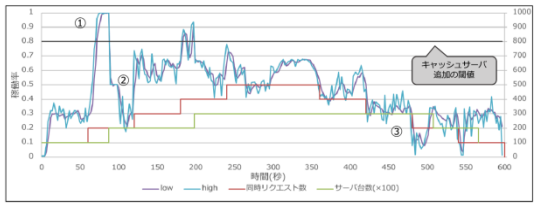

図 11. DNS を用いたアクセス振分機構

(5) 優先アクセス機構に関する成果

本研究以前に開発していた NAP-Web と呼ぶ過負荷時のユーザの不満を軽減する Web システムを拡張することで優先アクセス機構を実装した．

図 12 に開発した機構の概要を示す．スケジューラは，キャッシュサーバからのアクセスなら右側の優先アクセス機構に，それ以外は左側のオリジナルの NAP-Web の機構に渡す．優先アクセス機構では Pri_Run_Ready キューの数のアクセスを実際に処理し，それ以上は Pre_Wait で待たせる．NAP_Web の機構では Run_Ready キューの数のアクセスを実際に処理し，それ以外は Wait，Re_Access で待たせるか，アクセス可能となる時刻を記録したチケットをクライアントに返すとともに Next_Wait に記録する．実際に処理されるリクエスト数は Pri_Run_Ready と Run_Ready のキューの個数で決まるため，その比率を変えることのより，優先アクセスの処理に利用できる Web サーバの計算機資源の割合を変えることができる．

図 13 は，Run_Ready のキューの個数を 30 にして通常アクセスを大量に行いながら，1，5，10 の 3 つのパターンの同時優先アクセスを連続して行った実験結果を示す．通常アクセスと優先アクセスのスループット（処

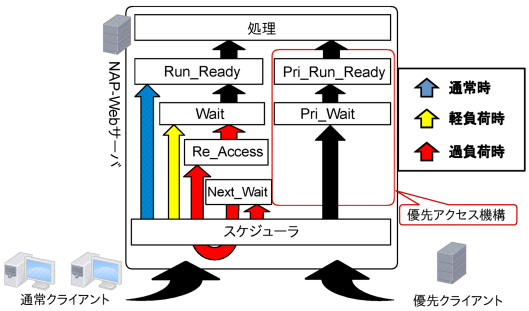

図 12. 優先アクセス機構

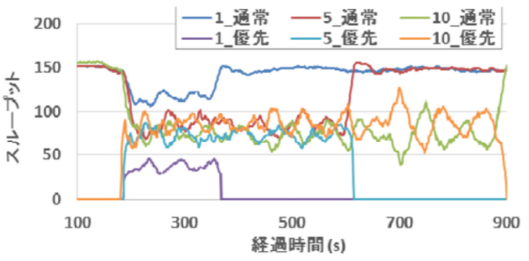

図 13. 優先／通常アクセスのスループット

理量）の比率は Run_Ready のキューの数と同時優先アクセス数の比率には比例していないが，同時優先アクセス数が増えるほど優先アクセスのスループットが増加している．

次に，Run_Ready のキューの個数を 20，Pri_Run_Ready のキューの個数を 2 に設定して，再アクセスが生じるほどの通常アクセスを行いながら，5，10，20 の 3 つのパターンで同時優先アクセスを繰り返す実験を行った．図 14 に優先アクセスの応答時間を示す．通常アクセス側の応答時間は 4 秒であったが，優先アクセスの応答時間はそれよりもはるかに短く，優先アクセスの同時アクセス数でのみ決定されていることが分かる．この結果は優先アクセス機構の有効性を示している．

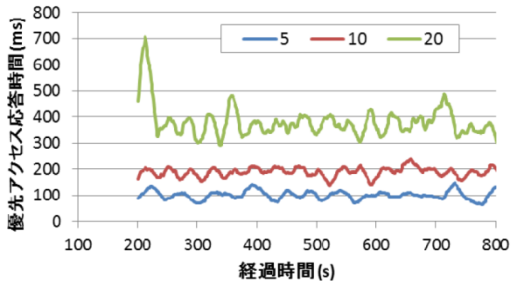

図 14．優先アクセスの応答時間

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計 2 件）

① A.Horiuchi, K.Saisho, Prototyping and Evaluation of Virtual Cache Server Management Function for Distributed Web System , Proc. the 2015 International Conference on Computational Science and Computational Intelligence (CSCI'15), 査読有，2015，324-329

② A.Horiuchi, K.Saisho, Development of Scaling Mechanism for Distributed Web System, Proc. 16th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD 2015). 査読有, 2015, 283-288

〔学会発表〕（計 14 件）

① 溝渕久哲，最所圭三，優先アクセス機構を持つ Web サーバにおける優先アクセス管理機構，2016 年電子情報通信学会総合大会，2016，九州大学伊都キャンパス（福岡県・福岡市）

② 堀内晨彦，最所圭三，負荷予測に基づく分散 Web システムにおけるキャッシュサーバ管理機能の改良，2016 年電子情報通信学会総合大会，2016，九州大学伊都キャンパス（福岡県・福岡市）

③ 溝渕久哲，最所圭三，優先アクセス機構を持つ Web サーバの開発およびその評価，第 14 回情報科学技術フォーラム（FIT2015），2015，愛媛大学（愛媛県・松山市）

④ 堀内晨彦，最所圭三，分散 Web システムにおけるキャッシュサーバ管理機構の試作と評価，第 14 回情報科学技術フォーラム（FIT2015），2015，愛媛大学（愛媛県・松山市）

⑤ 堀内晨彦，最所圭三，移動平均による分散 Web システムにおける振分アルゴリズムの改良，情報処理学会第 77 回全国大会 2015，京都大学（京都府・京都市）

⑥ 大川昌寛，最所圭三，ファイアウォールを用いた Web サーバのための同時アクセス数を動的に制御するアクセス制御機構の設計，情報処理学会第 77 回全国大会，2015，京都大学（京都府・京都市）

⑦ 堀内晨彦，最所圭三，分散 Web システムにおけるキャッシュサーバへの振分アルゴリズムの改良，平成 26 年度電気関係学会四国支部連合，2014，徳島大学（徳島県・徳島市）

⑧ 溝渕久哲，曽川直也，最所圭三，NAP-Web における優先アクセス機構の設計と評価，平成 26 年度電気関係学会四国支部連合大会，2014，徳島大学（徳島県・徳島市）

⑨ 蔵本幸司，堀内晨彦，最所圭三，DNS を用いたアクセス振り分け機構の提案，平成 26 年度電気関係学会四国支部連合大会，2014，徳島大学（徳島県・徳島市）

⑩ 堀内晨彦，最所圭三，クラウドに適した Web システムにおけるキャッシュサーバの負荷監視および負荷分散，電子情報通信学会情報ネットワーク研究会，2014，香川大学（香川県・高松市）

⑪ 堀内晨彦，最所圭三，クラウドに適した Web システムの負荷監視機能の改善について，情報処理学会第 76 回全国大会，2014，東京電機大学東京千住キャンパス（東京都・足立区）

⑫ 堀内晨彦，小笹光来，最所圭三，クラウドに適した Web システムの負荷監視機能の改善について，平成 25 年度電気関係学会四国支部連合大会，2013，徳島大学（徳島県・徳島市）

〔その他〕

ホームページ等

https://air.eng.kagawa-u.ac.jp/doku.php?id=wiki:theme

６．研究組織

(1)研究代表者

最所 圭三（SAISHO, Keizo）

香川大学・工学部・教授

研究者番号：５０１７０４８６