同時アクセス数制御機構におけるクライアント識別機構の開発

13T245　利根 大樹 （最所研究室）

特定のサービスに対して，同時アクセス数制御機構おけるクライアント単位でし識別アクセスの可否を行う機構において，特定のサービスのクライアント情報を渡す機能，複数サービスへの振分を可能とする機能の実装について述べる．さらに，同時アクセス数制御機構における HTTPS や様々な HTTP メソッドの利用可否の検証について述べる．

1　はじめに

当研究室では，ある特定のサービス (特定サービス) を安定的に提供するため，IP アドレスとクライアント単位のフィルタリングを組み合わせた同時アクセス数制御機構の開発を行っている．この機構により，同時アクセス数に上限を設け，上限以上のアクセスを許可しないことでサーバの負荷を軽減し，安定したサービスの提供が可能となる．ファイアウォールにより不特定多数の IP アドレスからのアクセスを防ぐことができ，DoS 攻撃にも対応できる．さらにクライアント単位でフィルタリングも行うことで，NAT 環境などからのアクセスでも正しくフィルタリングできる．

先行研究 [1][2] では単一のサービスを対象とした実装が行われた．しかし先行研究では，特定サービスではユーザの認証時のアカウント名がわからず，クライアントに応じたサービスの提供をしたい場合，特定サービス毎にクライアントを識別する仕組みの用意が必要となる．また本研究では複数の特定サービスへの対応も目指している．本稿では，クライアント単位でフィルタリングするクライアント識別機構における，特定サービスサーバへクライアントの識別子を渡す機能と複数の特定サービスへの振り分けを行う機能の，設計と実装について述べる．また先行研究での実装部分において発見されたバグの修正について述べる．さらに HTTPS や様々な HTTP メソッドの同時アクセス数制御機構における利用可否の検証についても述べる．

2　サービス利用までの流れ

認証からサービス利用までの流れを図 1 に示す．特定サービスサーバ (SS サーバ) へのアクセスは常に，IP フィルタリングサーバ (IPF サーバ)，クライアント識別サーバ (CI サーバ) を経由する．認証サーバ (Auth サーバ) は，ユーザ認証機能を提供する．IPF サーバは IP アドレスによるフィルタリングを行う．CI サーバでは，クライアント単位でフィルタリングを行う．SS サーバへのアクセスは以下の手順で行う．

クライアントは Auth サーバ上にて，ユーザ認証を行う (1)．認証が成功すると，IPF サーバに，同時アクセス数に基づいてアクセス可能かを問合せる (2)(3)．

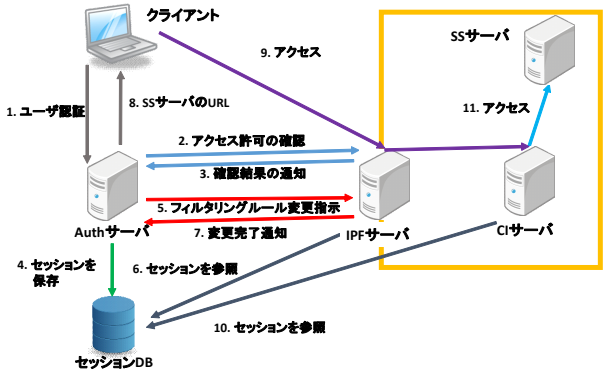

図 1: 同時アクセス数制御機構の構成

アクセス可能であれば，セッション情報を登録し (4)，IPF サーバにフィルタリングルール変更を指示する (5)．IPF サーバはセッション DB を参照し (6) フィルタリングルールを変更してアクセスを許可した後，Auth サーバへ変更完了通知を出す (7)．その後，クライアントは SS サーバへアクセスするための URL を受け取り (8) アクセスする (9)．IPF サーバでは，IP アドレスによるフィルタリングを行う．IPF サーバを通過すると，CI サーバはセッション DB を参照し (10)，アクセスの許可されたクライアントかどうかを識別する．これにより，NAT 環境やプロキシ経由の同一の IP アドレスを持つクライアントからのアクセスを識別する．アクセスの許可されたクライアントからであれば，SS サーバにアクセスし結果をクライアントに返す (11)．許可されていないクライアントの場合は，認証させるため Auth サーバにリダイレクトする．次回以降のアクセスでは，手順 (9) 以降を繰り返す．

3　SS サーバへのクライアント識別子の通知

特定サービスにおいて，クライアント毎に異なる情報やサービスを提供したいといった要求が考えられる．そのために特定サービスにはクライアントを識別する仕組みが必要となる．そこで本機構において，クライアントが認証時に利用したアカウント名をクライアント識別子として，クライアントのアクセスと同時に SS サーバへ渡すように実装した．クライアントからのリ

クエストを CI サーバが受け取り，SS サーバへリクエストを送る際に，SS サーバへのリクエストヘッダにアカウント名を含めて送る．以上の機能を実装し，SS サーバからアカウント名を受け取れること，アカウント名を利用してクライアント毎に異なる情報の掲示が可能なことを確認した．

4　複数サービスへの振り分け

図 2 に，ユーザ認証から特定サービスを利用するまでにおいて，各サービスへの振り分けが行われる際の概略を示す．クライアントは gate.hoge.piyo の URL で Auth サーバにアクセスし，認証する (1)．認証が通れば，Auth サーバはクライアントへ hoge.piyo の URL を返しリダイレクトさせる (2)．クライアントは hoge.piyo へアクセスし (3)，クライアントからのアクセスは CI サーバで SS サーバ A へ振り分けられる (4)．同様にクライアントが gate.foo.bar で認証 (1) した場合は，foo.bar へリダイレクトされ (2)(3)，SS サーバ B へ振り分けられる (4)．次回以降のアクセスでは (3) と (4) の手順を繰り返す．ただしどちらのサービスにおいても，認証をせず SS サーバへアクセスしようとすると Auth サーバへリダイレクトされる．ここで述べたように実装し，アクセスの振り分けが可能なことを確認した．

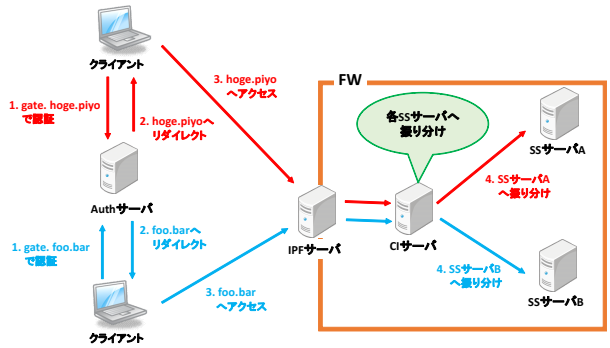

図 2: 複数サービスへの振り分け

5　アクセス拒否時の問題

先行研究で実装された CI サーバでのフィルタリングにおいて，アクセス拒否時でも SS サーバからの応答を返すバグが発見された．先行研究では，CI サーバからのリダイレクト先 URL を含めたレスポンスの確認や SS サーバのアクセスログの確認がされておらず，バグの発見ができなかったと思われる．このバグに対して修正を行い，CI サーバからのリダイレクト先 URL を含めたレスポンスと SS サーバのアクセスログからバグの修正ができていることを確認した．

6　HTTPS の利用可否

クライアントと CI サーバ間，CI サーバと SS サーバ間の 2 つの区間において，どちらか，あるいは両方で HTTPS を用いた場合の同時アクセス数制御機構における利用可否を調べた．結果を表 1 に示す．両区間で HTTPS を用いた場合は許可してはいけないアクセスを SS サーバへ通してしまったが，片方で HTTPS を用いた場合は正しくフィルタリングできた．表 1 より，図 1 に示した構成では，インターネットを介するクライアントと CI サーバ間を HTTPS で通信でき，セキュアな通信が実現できることが確認できた．

表 1: HTTPS の利用可否

クライアント − CI サーバ	CI サーバ − SS サーバ	利用可否
HTTP	HTTPS	○
HTTPS	HTTP	○
HTTPS	HTTPS	×

7　HTTP メソッドの利用可否

同時アクセス数制御機構において，GET，POST，HEAD，OPTIONS，PUT，DELETE，PATCH メソッドに関して利用可否を調べた．HTTP メソッドにはその他にも TRACE や CONNECT など存在するが，これらは検証するための環境の準備ができず調査できなかった．検証結果を表 2 に示す．表 2 より，検証したメソッドは全て利用可能であることがわかった．

表 2: HTTP メソッドの利用可否

メソッド	GET	POST	HEAD	OPTIONS
利用可否	○	○	○	○
メソッド	PUT	DELETE	PATCH	
利用可否	○	○	○	

8　今後の課題

ファイアウォールを用いた同時アクセス数制御機構の機能を，近藤氏と分担して開発を進めた．そのため，近藤氏の実装したファイアウォールを用いた機構 [3] との統合が必要である．

参考文献

[1] 大川昌寛, “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 回全国区大会講演論文集, 5X-4, pp.3-169 - 3-170, 2015 年.

[2] 杉本亮太, “ファイアウォールを用いた同時アクセス数制御機構の試作” 学士論文, 2016 年.

[3] 近藤裕基, “ファイアウォールを用いた同時アクセス数制御機構におけるアクセス権管理機能の実装” 学士論文, 2017 年.