

# HP IceWall SSO + Keep-Alive

## はじめに

この技術レポートでは、Keep-Alive<sup>※1</sup>とは何か、を簡単にご紹介します。

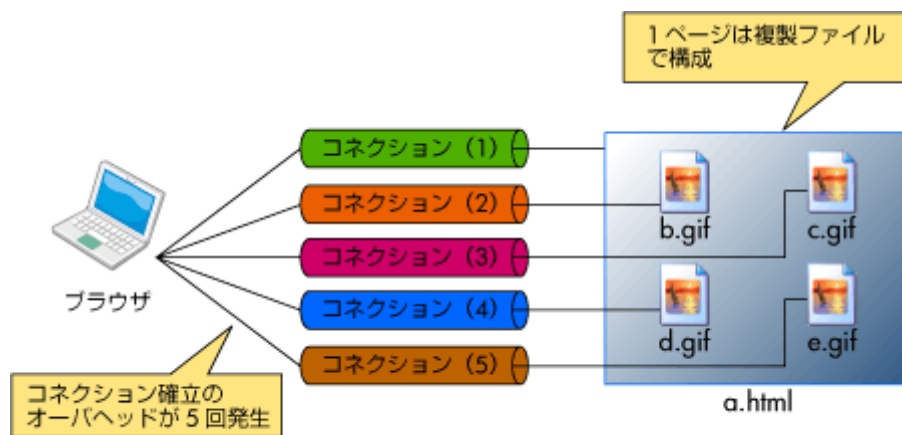
また、HP IceWall SSO(dfw, agent / mcrp)のメモリサイジングの考え方をご紹介します。

※1 持続的接続 (Persistent Connections) と呼ばれることもあります。

## Keep-Aliveとは？

初期のHTTPプロトコルでは、クライアントは1ファイルにつき1本の接続を確立して、ダウンロードを行うことが前提とされていました。

一般的に、WEBページは1枚のHTMLファイルと、複数の画像ファイル、スクリプトファイルなどで構成されることから、これはあまり効率がよいとは言えませんでした。

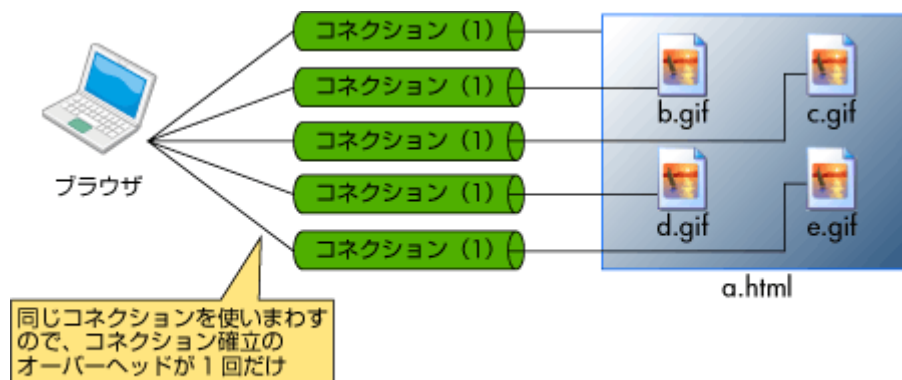


### WEB ページ取得イメージ (Keep-Alive なし)

実際にはブラウザは複数のファイルを並行して同時にダウンロードすることができます。  
(IEのデフォルト値ではHTTP1.1の場合は2コネクション)

この問題を解決するために、Keep-Alive と呼ばれる機能が組み込まれました。

クライアントは接続を維持し、同じ接続で複数のファイルをダウンロードできるようになったことから、効率が改善されました。



### WEB ページ取得イメージ (Keep-Alive あり)

## ApacheのKeep-Alive設定

HP IceWall SSOでも、動作プラットフォームであるApacheに設定を行うことで、Keep-Aliveのメリットを享受することができます。

Apacheの関連ディレクティブは以下の通りです。

- KeepAlive (On / Off)

「On」とすることで Keep-Alive を利用できます。

- KeepAliveTimeout (秒)

最終リクエストから設定秒数の経過で、コネクションを切断します。

- MaxKeepAliveRequests (回)

設定回数のリクエストに使用されると、コネクションを切断します。

- MaxClients (プロセス)

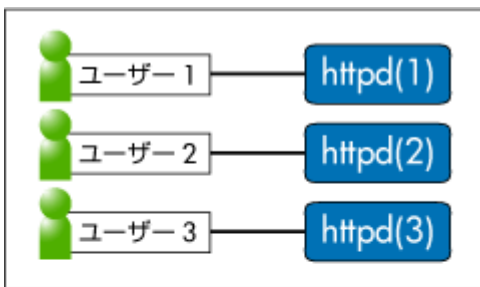
最大で、設定数のhttpdプロセスを起動<sup>※2</sup>することができます。

※2 HP-UX版 (worker MPM) では、ThreadsPerChildを1とすることがHP IceWall SSO製品で推奨されています。このため、HP-UX版でもMaxClientsの設定値を同時に処理可能なリクエストの最大数とみなしています。

#### ApacheのKeep-Alive実装の注意点

ApacheはKeep-Aliveによるコネクションを維持するために1 httpdプロセスが占有される仕様です。

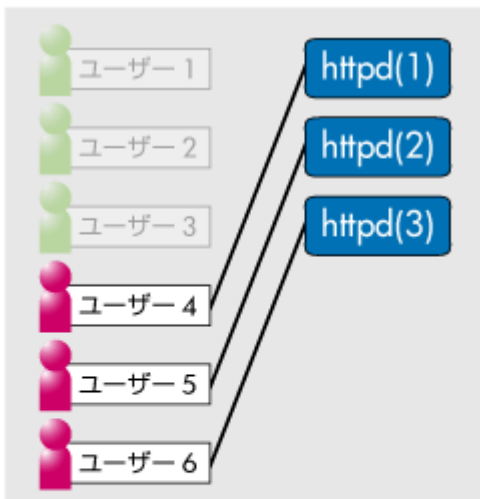
Keep-Alive未使用時に比べ、httpdプロセスの利用効率が低下し<sup>※3</sup>、より多くのプロセスが必要となります。



#### 初期状態

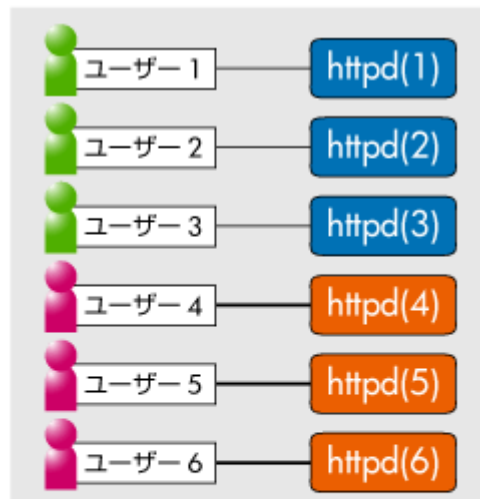
3 ユーザーがアクセス。  
1 リクエストを処理するのに、1 プロセスがアサインされる。

さらに 3 ユーザーがアクセスすると ...



#### Keep-Alive 未使用

リクエストの処理が終了したプロセスは解放され、新ユーザーのリクエストに再アサインされる。



#### Keep-Alive 使用

リクエスト処理終了後もコネクション維持のためプロセスは解放されない。新ユーザーのリクエストには新規プロセスが起動され、アサインされる。

httpdプロセスが不足すると、クライアントからのリクエストが受け付けられなくなってしまうため、MaxClientsには十分に大きな値の設定が必要です。

MaxClients値を大きくすると、メモリ使用量も増大するため、適切なメモリサイジングが必要になります。

※3 利用効率の低下は、MaxKeepAliveRequestsやKeepAliveTimeoutの設定値を小さくすることで、その度合いのある程度コントロールすることができます。B2Cサイトでは、KeepAliveTimeoutを3秒程度に設定しているケースがあります。

dfw、agent / mcrpをインストールしたhttpdの1プロセスあたりのメモリ使用量の指標<sup>※4</sup>は以下のとおりです。

RHEL5) dfw	0.9MB 程度
RHEL5) agent / mcrp	0.7MB 程度
HP-UX) dfw	2MB 程度
HP-UX) agent / mcrp	3MB 程度

この指標にMaxClientsを乗じることで、ベースとなる物理メモリ使用量が算出できます。

例) RHEL 5、agent / mcrp構成、MaxClients = 1000の場合  
- ベースとなるメモリ使用量 = 0.7MB × 1000 = 700MB

※4 OSインストール直後のhttpd.confにdfw、agent、mcrp設定、KeepAlive On設定を追加した状態での計測値であり、あくまでも目安となります。RHELはfreeコマンド、HP-UXはglanceコマンドで計測しています。

サイズの大きいファイルのアップロード、ダウンロード時には、ファイルサイズ分の作業用メモリが確保<sup>※5</sup>されます。

ピーク時におけるアップロード、ダウンロードの同時発生件数にファイルサイズを乗じることで、一時的なメモリ使用量が算出できます。

例) ピーク時には10MBのファイルダウンロードが同時に10件発生する場合  
- 一時的なメモリ使用量 = 10件 × 10MB = 100MB

ベースとなるメモリ使用量に一時的なメモリ使用量を加えた値が利用可能な物理メモリ量の合計を超えないようにします。

例) ベースメモリが700MB、一時メモリが100MBの場合  
- 700MB + 100MB = 800MB < 利用可能な物理メモリ量

※5 ファイルアップロード、ダウンロードにより一時的に確保された作業メモリは原則として処理終了後に解放されますが、MaxRequestPerChildに適切な値を設定することで、定期的にプロセスを再作成することを推奨します。

## おわりに

HP IceWall SSOにおいても、Keep-Aliveは性能改善に有効なソリューションです。メモリサイジングにご注意の上、ご利用を検討ください。