

記事

[Toshihiko Minamoto](#) · 2020年12月22日 13m read

HealthShare の Apache HTTPD Web サーバー構成

HealthShare の理想的な Apache HTTPD Web サーバー構成に関するお問い合わせをよくいただいています。この記事では、真っ先に推奨される HealthShare 製品の Web サーバー構成について概要を説明します。

何よりもまず第一に、Apache HTTPD バージョン 2.4.x (64ビット) を使用することをお勧めします。2.2.x のような旧バージョンも使用できますが、HealthShare のパフォーマンスとスケーラビリティを確保するにはバージョン 2.2 はお勧めできません。

Apache の構成

NSD を使用しない Apache API モジュール

HealthShare では、NSD を使用しない Apache API モジュールをインストールオプションに指定する必要があります。動的にリンクされるモジュールのバージョンは、Apache のバージョンによって異なります。

- CSPa24.so (Apache バージョン 2.4.x)

Apache の httpd.conf での Caché Server Pages の構成は、このドキュメントで詳しく説明する HealthShare のインストールによって実行するのが最善です。ただし、この設定は手動で実行することもできます。詳細については、InterSystems のドキュメントに掲載されている Apache 構成ガイド「[推奨オプション：NSD を使用しない Apache API モジュール \(CSPa24.so\)](#)」を参照してください。

Apache マルチプロセッシングモジュール (MPM) の推奨事項

Apache Prefork MPM と Worker MPM

Apache HTTPD Web サーバーには、Prefork / Worker / Event の 3 つのマルチプロセッシングモジュール (MPM) が付属しています。MPM はマシンのネットワークポートへのバインド、リクエストの受理、およびリクエストを処理するように子プロセスに割り当てたりする役割を担います。Apache はデフォルトで通常はトランザクションが多い場合や同時接続による負荷が高い場合には適切に拡張できない Prefork MPM で構成されています。

HealthShare の本番システムでは、Apache MPM **Worker** を有効にしてパフォーマンスとスケーラビリティを確保する必要があります。Worker MPM が推奨される理由は次のとおりです。

- Prefork MPM はそれぞれ 1 つのスレッドを持つ複数の子プロセスを使用し、各プロセスが一度に 1 つの接続を処理します。Prefork を使用する場合は各プロセスが一度に 1 つのリクエストしか処理できないため、同時リクエストが発生した場合に支障が出ます。この場合、リクエストはサーバープロセスが解放されるまでキューに格納されます。また、スケーリングするには大量のメモリを消費する Prefork の子プロセスがさらに必要になります。
- Worker MPM は、それぞれ多数のスレッドを持つ複数の子プロセスを使用します。各スレッドが一度に 1 つの接続を処理することで同時実行性が大幅に向上し、メモリ要件が低くなります。Worker の場合はビジー状態の可能性があるシングルスレッドの Prefork

プロセスとは異なり、通常はリクエストの処理に使用できる空きスレッドがあるため、Prefork よりも高度に同時リクエストを処理できます。

Apache Worker MPM のパラメーター

Worker はスレッドを使用してリクエストを処理するため、Prefork プロセススペースのサーバーよりも少ないシステムリソースで多数のリクエストを処理できます。Worker MPM の制御に使用される最も重要なディレクティブは、各子プロセスが生成するスレッドの数を制御する **ThreadsPerChild** と起動可能なスレッドの最大数を制御する **MaxRequestWorkers** です。推奨される Worker MPM 共通ディレクティブの値を以下の表に詳しく記載しています。

推奨される Apache HTTPD Web サーバーのパラメーター	Apache Worker MPM のディレクティブ	推奨値
	MaxRequestWorkers	HealthShare の最大同時コンポーネントインターフェースの合計 * 注意: 構成時に
	MaxSpareThreads	250
	MinSpareThreads	75
	ServerLimit	MaxRequestWorkers
	StartServers	20

	ThreadsPerChild	25
--	-----------------	----

詳細については、関連する Apache バージョンのドキュメントを参照してください。

- [Apache 2.4 の MPM 共通ディレクティブ](#)

Apache 2.4 の Worker MPM 構成の例

このセクションでは、最大 500 人の TrakCare ユーザーが同時接続できるようにするために必要な RHEL7 の Apache 2.4 Web サーバー用に Worker MPM の構成について詳しく説明します。

1. まず、以下のコマンドを使用して MPM を確認します。

```
$ apache2l -v | grep MPM
Server MPM: prefork
```
2. 必要に応じて、構成ファイル /etc/httpd/conf.modules.d/00-mpm.conf を編集します。コメント文字 # の追加や削除を行い、Worker MPM モジュールのみがロードされるようにしてください。Worker MPM セクションを、以下と同じ順序で次の値に変更します。
3. Apache を再起動します。
4. Apache が正常に再起動したら、以下のコマンドを実行して worker プロセスを検証します。
次のような内容が表示され、httpd.worker プロセスを確認できます。

Apache の強化

Apache に必要なモジュール

公式の Apache 配布パッケージをインストールすると、デフォルトで特定の Apache モジュールセットが有効になります。Apache のこのデフォルト構成では、これらのモジュールが各 httpd プロセスにロードされます。次の理由により、HealthShare に必要のないすべてのモジュールを無効にすることをお勧めします。

- httpd デモンプロセスのメモリ使用量が削減されます。
- 不正なモジュールに起因するセグメンテーション違反の可能性が低下します。
- セキュリティの脆弱性が削減されます。

HealthShare に推奨される Apache モジュールの詳細を以下の表に記載しています。
以下に掲載されていないモジュールは無効にすることができます。

モジュール名	説明
alias	ホストファイルシステム内のさまざまな場所をドキュメントツリーにマッピングする機能と、URL リダイレクト機能を提供します。
authz_host	クライアントのホスト名、IP アドレスに基づいてアクセス制御を行います。
dir	末尾のスラッシュを補完してリダイレクトする機能と、ディレクトリのインデックスファイルを扱う機能を提供します。
headers	HTTP リクエストとレスポンスヘッダーを制御および変更するためのモジュールです。
log_config	サーバーに対して発行されるリクエストのログを記録します。
mime	リクエストされたファイル名の拡張子をファイルの振る舞いと内容に関連付けます。

モジュール名	説明
negotiation	クライアントの特性に応じて最適なドキュメントの内容を選択する機能を提供します。
setenvif	リクエストの特徴に基づいて環境変数を設定できるようにします。
status	サーバーのステータスとパフォーマンスに関する統計を表示します。

モジュールを無効にする

セキュリティの脆弱性を減らすように構成を強化するには、不要なモジュールを無効にする必要があります。

Web サーバーのセキュリティポリシーを管理するのはお客様の役割です。

少なくとも、以下のモジュールを無効にする必要があります。

モジュール名	説明
asis	独自の HTTP ヘッダーを含むファイルを送信する機能を提供します。
autoindex	index.html ファイルが存在しない場合にディレクトリインデックスを生成し、ディレクトリをリスト表示する機能を提供します。
env	CGI スクリプトと SSI ページに渡される環境変数を変更する機能を提供します。
cgi actions	cgi - CGI スクリプトを実行する機能を提供します。 メディアタイプまたはリクエストメソッドに基づいて CGI スクリプトを実行し、リクエストに応じてアクションをトリガーする機能を提供します。
include	サーバーで解析された HTML ドキュメント（サーバーサイドインクルード）を使用可能にするモジュールです。
filter version	リクエストの高度なフィルタリング機能を提供します。 IfVersion を使用して構成ファイル内でバージョン情報を処理できるようにします。
userdir	リクエストをユーザー専用のディレクトリにマッピングする機能を提供します。具体的には、URL の <code>username</code> がサーバー内のディレクトリに変換されます。

Apache SSL/TLS

転送中のデータを保護し、機密性を確保して認証を行うため、InterSystems は HealthShare

サーバーとクライアント間のすべての TCP/IP 通信を SSL/TLS

で暗号化し、さらにはユーザーのブラウザクライアントと提案されたアーキテクチャの Web

サーバーレイヤー間のすべての通信に HTTPS を使用することを推奨しています。所属する組織に固有のセキュリティ要件に準拠していることを確認するには、必ず所属する組織のセキュリティポリシーを参照してください。

SSL/TLS 証明書を提供および管理するのはお客様の役割です。

SSL 証明書を使用している場合は、`sslmodule (modssl.so)` を追加してください。

追加の Apache 強化パラメーター

Apache 構成をさらに強化するには、`httpd.conf` で次の変更を行ってください。

- 潜在的なクロスサイトトレーシングの問題を防ぐため、`TraceEnable` をオフにする必要があります。
- Web サーバーのバージョンが表示されないようにするため、`ServerSignature` をオフにする必要があります。

補足的な Apache 構成パラメーター

Keep-Alive

Apache の Keep-Alive 設定は、新しいリクエストのたびに新しい接続をオープンせず、同じ TCP 接続を使用して HTTP 通信を行えるようにするものです。Keep-Alive はクライアントとサーバー間に持続的な接続を維持します。Keep-Alive オプションを有効にすると、ネットワークの輻輳が軽減され、後続リクエストの遅延が少なくなり、同時接続のオープンに起因する CPU 消費が低下し、パフォーマンスが向上します。Keep-Alive はデフォルトでオンになっており、HTTP v1.1 標準では当然オンにすることが求められています。

ただし、Keep-Alive を有効にする場合は注意が必要です。Internet Explorer は古いバージョンでの既知のタイムアウトの問題を回避するため、IE10 以降でなければなりません。また、ファイアウォール、ロードバランサー、プロキシなどの仲介者が「持続的なTCP接続」に干渉し、予期せず接続が閉じられてしまう可能性もあります。

Keep-Alive を有効にする場合は、Keep-Alive のタイムアウト値も設定する必要があります。Apache デフォルトの Keep-Alive タイムアウト値は非常に小さく、壊れた AJAX (hypercotent などの) リクエストに関連して問題が発生する可能性があるため、ほとんどの構成では大きくする必要があります。

これらの問題は、サーバーの Keep-Alive タイムアウト値がクライアントのタイムアウト値よりも大きくなるように調整することで回避できます。つまり、サーバーではなくクライアントがタイムアウトして接続を閉じるようにすべきです。ブラウザが当然オープンしていると考えている接続を（特に POST する場合に）使用しようとする、問題が発生します（ほとんどの場合は IE で発生し、他のブラウザではそれほど発生しません）。

HealthShare Web サーバーに推奨される KeepAlive と KeepAliveTimeout の値については、以下を参照してください。

Apache で KeepAlive を有効にするには、httpd.conf で以下のように変更を行ってください。

CSP Gateway

CSP Gateway の KeepAlive パラメーターについては、各リクエストの HTTP レスポンスヘッダーによって KeepAlive のステータスが決まるため、この値はデフォルトの「No Action」のままにしてください。

[#Red Hat Enterprise Linux \(RHEL\) #SOAP #インターシステムズビジネスソリューションとアーキテクチャ #システム管理 #パフォーマンス #HealthShare](#)

ソースURL:<https://jp.community.intersystems.com/post/healthshare-%E3%81%AE-apache-httpd-web-%E3%82%B5%E3%83%BC%E3%83%90%E3%83%BC%E6%A7%8B%E6%88%90>