

マルチスレッド動作可能な PHP を導入した event 式サーバの構築 Construction of the Event Type Server with PHP Making Multi-thread Operation Possible

○古川快¹, 関根好文²*Kai Furukawa¹, Yoshifumi Sekine²

Abstract: This paper describes setting of Web server with uses thread. “prefork” is widely used all over the world. However, when many clients access a server, the “prefork” has the problem of using many memory and CPU. Then, we solving the problem by use thread based model “event”. It is one of the reasons an “occurrence” is not widely used all over the world that PHP is not thread safe. We solve the reason by rewriting the program to available thread. Through these methods, we construct of the event type server with PHP making multi thread operation possible.

1. まえがき

近年、インフラ整備が進み従来ホームページで主流だった文字を中心としたサービスはウェブログや SNS として利用者が増加傾向にある^[1]。これらはシステム提供者が情報提供者ではなく、利用者が情報提供者となっている。システムでも専門の人が構築するだけでなく個人でサーバを立ち上げる人が増えている。

個人で立ち上げる場合、UNIX 系 OS を使用することが多い。今回サーバの構築に使用するのはサーバの構築に適している^[2]CentOS を用いた。CentOS は Red Hat 社の Red Hat Enterprise Linux(以下 RHEL)をベースとして作られた OS で RHEL の無償公開されたソースコードから商標や商用の要素を含まない形で構成されている^[3]。そのため RHEL と互換がありながら無償で利用することができる^[4]。Web サーバとして構築するに当たり、Apache HTTP Server(以下 Apache)という使用率が高いソフトウェアを使用した。

Apache では HTTP によるリクエストを受け付けて処理を行う部分をモジュール化した Multi Processing Module(以下 MPM)というものがあり、最も使用されているものが prefork である。これは後方互換性があり、スレッドを使用していないため、スレッド非対応のソフトウェアを使用することができるためである。しかし、この方式はアクセスが増えるとサーバへの負担が大きくなる。そのため、多数のクライアントからのアクセスを処理する場合はそれに応じた大量のメモリと高速な CPU を用意しなければならないという問題がある。

本稿では MPM の 1 つである event での構築の前段階として prefork をマルチスレッド動作に変更した worker での PHP を導入したサーバの構築について検討を行った。

2. 本論

MPM は HTTP によるリクエストを受け付けて処理する部分をモジュール化したものである。CentOS を用いる場合は以下の 3 つの MPM を使用することができる。

(1)prefork

図 1 に prefork のイメージ図を示す。同図は前もって子プロセス(fork)を起動させておき、クエストに応じて子プロセスを割り当てる。リクエストがあつてから子プロセスを起動させると時間がかかるためこの手法を用いている。

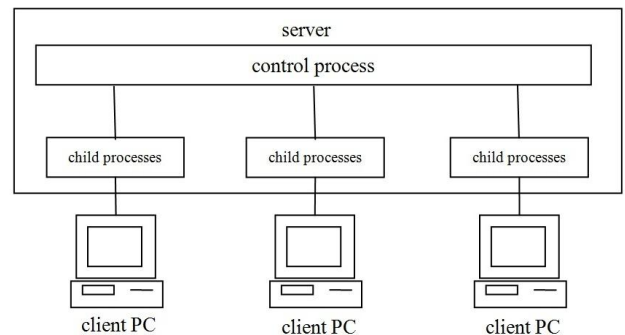


Figure 1. Prefork model.

(2)worker

図 2 に worker のイメージ図を示す。同図はリクエストに応じて必要な子プロセスを起動して、さらにプロセスがリクエスト数に応じてスレッドを起動し、リクエスト処理を行う。prefork に比べて子プロセスの数が少なく、代わりにスレッドを用いることによりメモリの節約ができる。

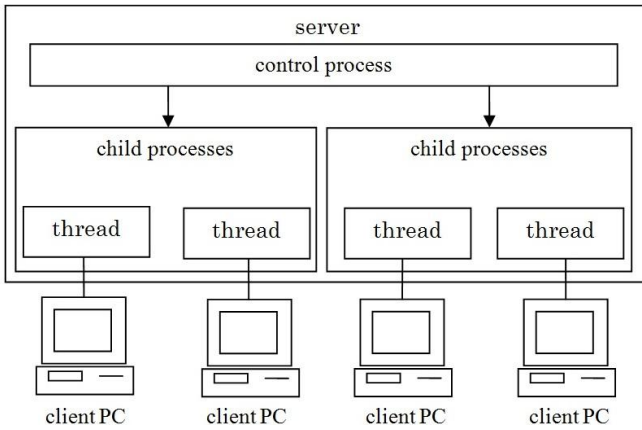


Figure 2. Worker model.

(3)event

基本的には worker と同じだが、クライアントとの接続が継続しているか定期的に確認を行う Keep-Alive の手法に違いがある。Keep-Alive の処理をクライアントからの処理を行うスレッドとは別のスレッドに割り振って通信を処理することによりサーバへの負荷を軽減することができる。

PHP を導入するに当たり、PHP はマルチスレッドを使用した MPM のサポートを行っておらず、そのままの PHP ではスレッドを使用した event では起動することができない。そのため、設定用の RPM ファイルを書き換える必要がある。表 1 に今回の RPM の追加内容を示す。これにより、マルチスレッドを使用できる仕様の PHP を導入することができた。

Table 1. Additional content for RPM.

<pre>--with-tsm-pthreads \ --enable-maintainer-zts \</pre>
--

図 3 に worker と先に変更を行った PHP を用いて構築したサーバの接続テスト結果の一例を示す。このテストではリクエスト数を同時接続数の 10 倍としてアクセスを行い、1 つ当たりのリクエストを処理するのにかかる時間を測定した。この結果では同時接続数が 100 を超えたあたりから prefork の処理時間が大きく変化しており、300 を超えると worker の 2 倍ほどかかることからマルチスレッドを使用することによって大量にサーバへのアクセスがあった場合の処理能力が改善されたことを示している。

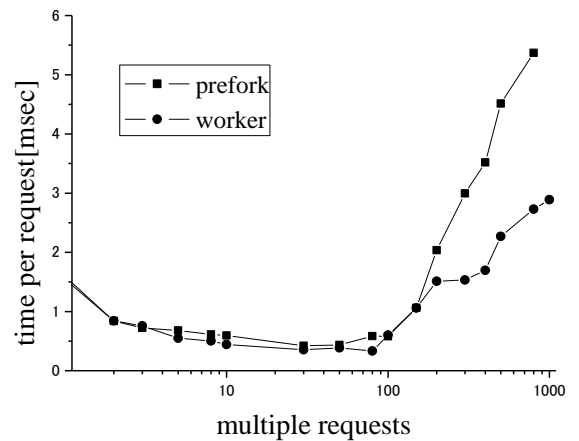


Figure 3. Stress test for access.

3. まとめ

今回、マルチスレッド環境で動作可能な PHP を導入し、event とほぼ同じ仕様である worker での構築を行い、接続テストにより同時接続数が 300 を超えると処理時間が半分程度にできた。その結果、マルチスレッド環境での構築を行うことにより、大量のアクセスがあった場合のアクセス処理にかかる時間の改善を行うことができることを示した。

今後は、Keep-Alive に改良を加えた event でのサーバの構築を行い、event, worker, prefork それぞれの環境下でのサーバへの負担や処理時間の変換を計測し、どの MPM を使用すると最もサーバへの負担軽減を行うことができるのか検証を行う予定である。

4. 参考文献

- [1] 福島 学, 濱田 大助, 末廣 一美, 高山 泰典, 松本 博樹, 兼松 大輔, 木野宮 孝樹, 杉尾 啓多:「Solaris を使用したサーバ構築方法について」, 日本文理紀要, 第 37 巻, 第 1 号, pp.43-53, 2009 年 3 月.
- [2] 平 初, 伊藤 幸夫, 上鍵 忠志, 中澤 直也, 面 和毅, 館林 綾ノ介, 高安 洋輝, 宇野 素史, 坂井 恵: 「CentOS サーバ構築バイブル 改訂第二版」, 毎日コミュニケーションズ, 2009 年.
- [3] 辻 秀典, 渡辺 高志, 鈴木 幸敏, できるシリーズ編集部:「できる PRO CentOS サーバ CentOS 5 対応」, 株式会社インプレスジャパン, 2009 年.
- [4] 中島 能和, 飛田 伸一郎:「Cent OS 徹底入門 第 2 版」, 翔泳社, 2009 年