

確率モデルに基づく CDN を応用した障害耐性サーバーの開発

1. 背景

これまで、障害対策や負荷分散を行うには、高価な装置を導入するしかなく、それができない場合、何の対策も立てられない状況であった。この状況を改善し、安価に障害対策や負荷分散を行えるようにすることを目指す。同時に、これまではやむを得ず高価な機器を導入していたが、実際にはそれほど高度な機能を必要としない場合に対するソリューションを与えることを目指す。

2. 目的

弱結合分散環境を容易に構築できるソフトウェアを提供することにより、高価なロードバランサーなどのハードウェアを導入せずにサーバーの障害耐性を高め、負荷分散を可能にする。

このような、弱結合分散環境構築機能を有するソフトウェアを

- ・ GUI による操作が可能な Windows 用統合サーバー
- ・ Apache 用モジュール

の形式で作成する。

3. 開発の内容

Windows 版・Apache 用モジュール共に、以下の機能を有している。

● ファイルを同期する機能

ファイルを一定時間ごとに同期する。既に送信済みのファイルについては、ファイルが更新されない限り再度送信されることはないので、ネットワークにかかる負荷は最小限度に抑えられる。同期済みのファイル名はログに保存される。

● 暗号化機能

同期の際にネット上にデータが流れるが、機密を守りたいファイルについては AES 暗号化により暗号化されて送られる。なお、ファイル名やパーミッションなどに関しては常に暗号化されている。

● DNS 更新機能

Failover（障害対応）においては、同期後に、同期に成功したサーバーの IP アドレスを DNS に登録し、DNS に記されている IP アドレス情報は稼動しているサーバーのものであることが保証されているようにする。ちなみにこのような手続きをとらなくても、クライアント側で IP アドレスを順にチェックして、稼動しているサーバーを選択するようになっているので、本機能がなくとも Failover 自体は可能である。

負荷分散においては、リクエスト数が予め指定した値になったときに DNS 情報を更新してラウンドロビンになるようにする。その結果、クライアントからのリクエストが分散される。負荷分散を停止する閾値は、確率モデルに基づき、危険率 1%で負荷分散を開始する閾値に戻らないと判断できる値に自動的に決定される。

DNS 情報の更新には、さまざまな DNS に柔軟に対応できるように、外部プログラムを呼び出す形式をとる。

これらに加えて Windows 版では

- Web サーバーとしての基本機能
- CGI の呼び出し機能
- バーチャルサーバー機能
- アクセス制限機能

を実装している。

Windows 版にのみ実装した機能は、Apache においては既に存在するものであり、改めて実装する必要はない。

Windows 版は、GUI による設定が可能で、簡単に使えることを目指している。Apache 版は、既存の Apache のブランド力を活用しつつ、必要な機能をモジュールとして提供することにより、本格的使用に耐えうるものとする。同時に、多くの OS 上での動作を可能にし、他のさまざまなモジュールを用いた拡張性をも実現している。

構造は図 1 及び図 2 のようになっている。Windows 版の作成においては、Delphi と C++ Builder を使い、Socket を使いやすくするコンポーネントや HTTP 制御など、あらかじめ C++ Builder に備わっている機能を活用して、工数の削減を図ると同時に安定した動作を実現している。今回実装した部分が赤い太線で囲まれている。

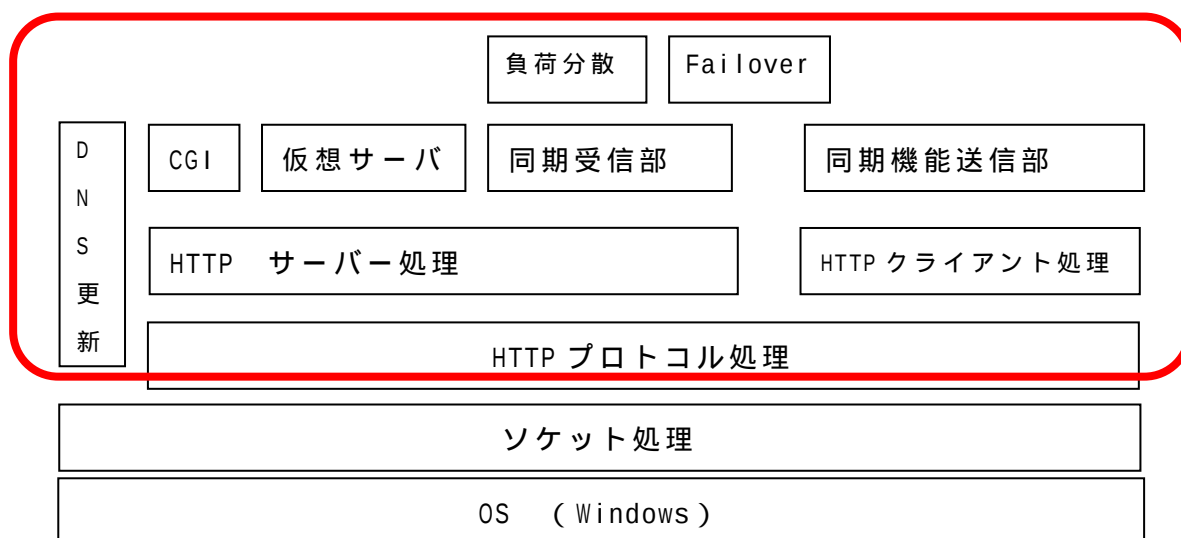


図 1 Windows 版 弱結合サーバー構成図

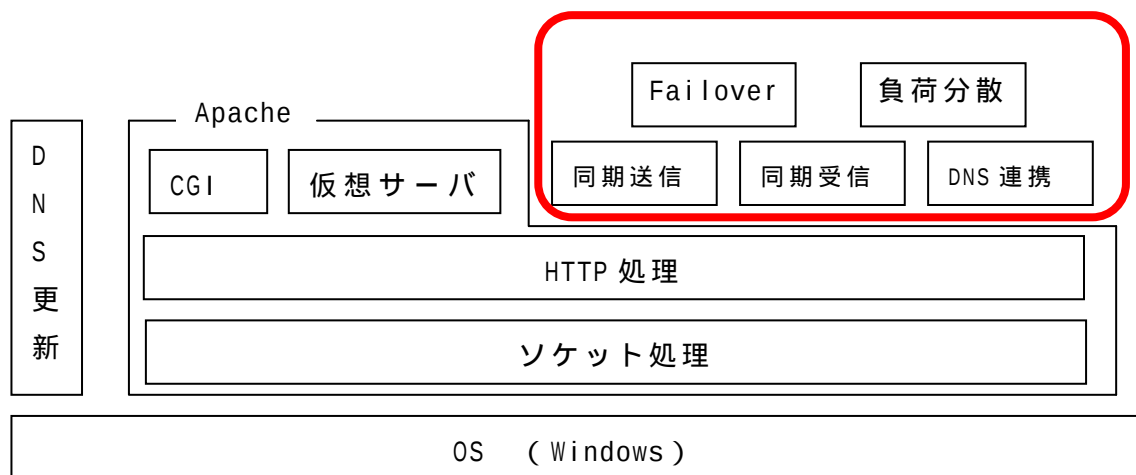


図 2 Apache モジュール構成図

共に太線の四角い枠で囲まれた箇所が今回の実装部分である。暗号化部分等は同期機能の送受信部に共に内蔵されているが、図が煩雑になるので省略している。

4. 従来の技術(または機能)との相違

現状では負荷分散、障害耐性をソフトウェア面で個々に行うものはいくつか挙げられるが、同期を動的に行い、それを利用して、動的に障害対策、負荷分散を行うといった手法はなく、われわれの開発独自のものといえる。また、負荷分散のタイミングを確率モデルに基づいて最適に決めているため、従来の製品よりも、より効果的な負荷分散が行える。

5. 期待される効果

同期機能、障害耐性機能、負荷分散機能を使い以下のような問題が改善される。

- ・ サーバーの耐性に不安がある。
- ・ ハードウェアによる障害耐性向上・負荷耐性向上が資金的に難しい。
- ・ より簡単にサーバーの耐性を向上したい。
- ・ 多くのサーバーを効率的に使用したい。

以下に負荷分散機能を使用した場合の効果を示す。

負荷分散が効果的に行われているかどうかを定量的に知るために、10 台のクライアントから 50KB のデータを毎秒 12 個取得する負荷を 1 台のサーバーにかけて、別のクライアントから、1MB～10MB までのサイズのデータを取得するのに要した時間を測定した。測定は、各データサイズに対して 50 回ずつ計 500 回行い、その平均値をとった。

同様の測定を、2 台で負荷分散したときと、3 台で負荷分散したとき、及び負荷をかけなかったときに行い、相互に比較した。

結果を図 3 に示す。2 台で負荷分散しただけで、負荷がないときとほぼ同様なパフォーマンスが得られているのがわかり、Web サーバーにかかる負荷を分散するという目的には十分な効果を発揮しているといえる。

取得に要した平均時間(ミリ秒)

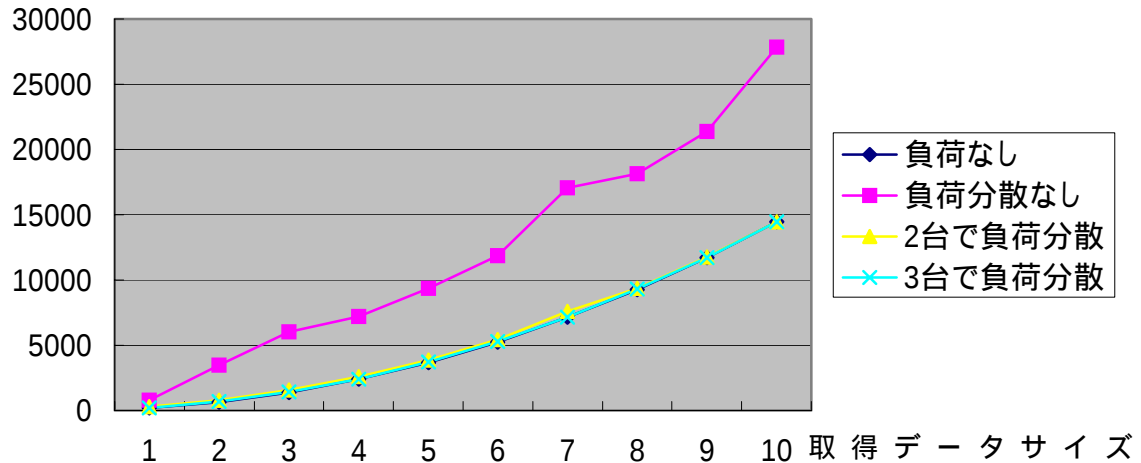


図 3 負荷分散の効果を測定した結果

6. 普及(または活用)の見通し

Apache のモジュールに関しては、オープンソースとして公開し、市場の活性化と発展ということを念頭におき開発を行ってきた。既に Apache 関係のサイトに登録しており、今後さまざまな人が刺激を受け発展していくようになるであろう。我々自身もさまざまなユーザーの意見を取り込み、このソフトの改善を進めて行きたい。

7. 開発者名(所属)

加藤崇(金沢工業大学 工学研究科 情報工学専攻)

石井充(金沢工業大学 情報工学科 講師)

佐々木和志(金沢工業大学 工学研究科 情報工学専攻)

西村美由紀(三谷産業株式会社 情報システム部 フィールドサポート課)

(参考)

<http://eagle-net.ne.jp/~blackstar/>

<http://eagle-net.ne.jp/~blackstar/module/>