

高トラフィックを想定した Web アプリケーションにおける 性能向上のためのアプローチ

宮本啓生[†] 鶴飼ひろみ[†] 平澤茂樹[†]

(株)日立製作所 システム開発研究所[†]

1. はじめに

近年のインターネットの拡大によって、検索エンジンや電子商取引など、様々なサービスが世の中に数多く輩出されている。それに伴い、現在では Web サービスを提供する技術も進歩し、Servlet や JSP (JavaServer Page) を始めとした新しい技術によって、Web サービスが提供されている。しかし、一方では、いまだ CGI (Common Gateway Interface) を利用した Web サービスも数多く存在している。一般的に CGI は、HTTP (Hyper Text Transfer Protocol) 要求をサーバに受信するたびにプロセスを起動するため、レスポンスが悪いと言われている。しかし、サービスを提供するアプリケーションに合わせたサーバ環境を構築することにより、レスポンスを向上させ、既存のサーバ構成でも高トラフィックに対応できる可能性がある。本報告は、CGI によって Web サービスを提供する高トラフィックを想定した評価モデルにおいて、レスポンス向上のための検討を行うものである。評価に当たり、高速な CGI を提供する TP1/Web^[1]を利用して環境の構築を行った。

2. TP1/Web の概要

TP1/Web は、処理プロセスを常駐化させておくことによって、要求ごとに行われる CGI の起動時に必要なローディング処理を小さくし、高速 CGI を実現しているミドルウェアである。処理プロセスを常駐化するとき、DB への接続や業務プログラムの初期化処理を行い、また、初期化処理と同時に HTML (Hyper Text Markup Language) をサーバのメモリにロードしておく、CGI 実行時の I/O を減らすことでローディング時間を小さくし、実行速度を高速化している。TP1/Web マネージャによって、同処理を行う CGI のプロセスを複数管理することで、大量の要求を受け付けることが可能である。その他、CGI の要求に応じて動的に HTML を生成する機能も搭載しており、電子商取引などの顧客毎に表示項目を変更するサービスに有効である。

当ミドルウェアを適応したサービスが多数存在しており、今後、継続してサービスを提供していく上で、アクセスの増加が見込まれることから、評価モデルに TP1/Web を用いた。

3. 評価モデル

3.1 モデル構成

評価モデルの構成を図 1 に示す。負荷分散装置は多数のクライアントから受けた要求を各 Web サーバに振り分ける。要求を受け取った Web サーバは、要求

“Approach for the improvement in execution by the web application supposing high traffic”

Hiroki Miyamoto[†], Hiromi Ukai[†],

Shigeki Hirasawa[†]

[†] Hitachi, Ltd. Systems Development Laboratory

に対する処理をバックエンドサーバに依頼し、クライアントへ応答を返す。以上の処理が 2 台の Web サーバに対して定期的に 200 件/秒発生し、最繁忙時には 500 件/秒の高トラフィックが発生すると想定した。

このような CGI で提供する Web サービスシステムに対して、先にあげた高トラフィックに対応した環境を設定することを目的とし、次のような計測と評価を行った。

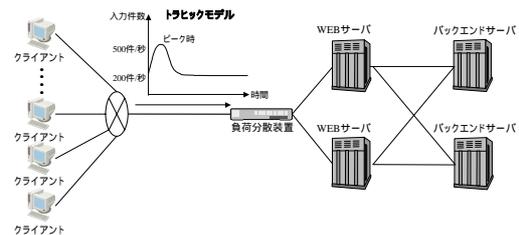


図 1 評価モデル

3.2 計測環境と検証方法

計測では、アプリケーションに適合した環境を設定するために、TP1/Web のサービスプロセス数をパラメータに取り、プロセス数とクライアントでのレスポンス時間との関係調べる。

まず、任意のプロセス数を設定し、仮想クライアントである LoadRunner^[2]から 200 件/秒の高トラフィックを数秒間発生させて、LoadRunner 上でのレスポンス時間を計測する。通常、CGI における Web サービスは、Web サーバ 1 台当たり 30 ~ 50 件/秒を処理するものとして構成されているが、ここでは、その倍の 100 件/秒が定常的なトラフィックとして設定する。トラフィックに対して CGI が処理しきれない場合には Apache^[3]と同機能を持つ Hitachi Web Server (以下、HWS) の子プロセスから CGI への接続エラーとしてシステムログに出力されるため、システムログを監視しながら、10 秒間のトラフィックを発生させた。1 サーバ 1000 件のレスポンス時間から平均を取り、再度プロセス数を変更して同様の作業を繰り返す。

クライアントの要求は LoadRunner から直接負荷分散装置である ServerIronXL に 100BASE-T ネットワーク経由で送信され、2 台ある Web サーバに振り分けられる。TP1/Web のサービスプロセス数を Web サーバ 1 台当たり 15 ~ 40 プロセスの間で変化させて検証し (図 1)、エラーがなく、高速で処理が可能なプロセス数を特定する。システム構成の詳細は表 1 とする。

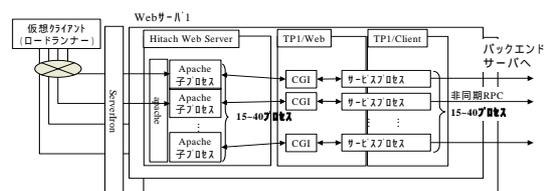


図 2 Web サーバの構成

クライアント	LoadRunner 7.0.2()
Web サーバ	UNIX サーバ Hitachi Web Server TP1/Web OpenTP1/Client
ネットワーク	100BASE-T

表 1 システム構成

3.3 計測結果

100 件/秒・1Web サーバの高トラフィックを両Web サーバにかけ、プロセス数と平均レスポンス時間の関係を計測した。結果を図 3 に示す。20 プロセスのときに、レスポンスが最も早くなり、処理件数も多い。15 プロセスでレスポンス時間が大きくなった理由として、CPU の処理する件数が減ったため、サービスプロセス自身の処理は早くなるが、CGI 起動の時間はほぼ固定であるため、起動できるプロセス数が少なくなった分、平均レスポンス時間が遅くなったものと考えられる。他にプロセス数を 100 以上にした場合も計測を行ったが、平均レスポンス時間が増す結果となった。

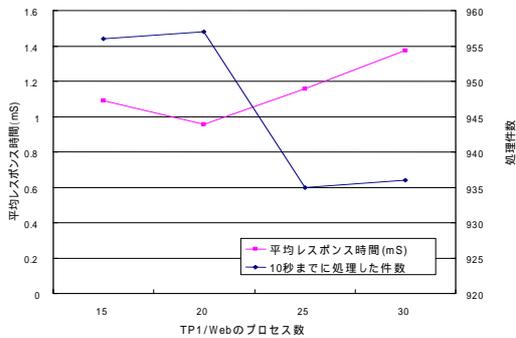


図 3 TP1/Web のプロセス数と平均レスポンス時間、処理件数の関係

4. 評価

3.3 節の結果から TP1/Web のプロセス数を 20 に設定し、評価モデルで想定した高トラフィックをかけ、このシステムへの影響を調べる。入力是最初の 1 秒間に 1 サーバ当たり 250 件の要求が入り、その後 100 件/秒が 9 秒継続する。図 4 がその結果である。10 秒間 1000 件のときと比べて、1 秒多く時間がかかっている。10 秒 1000 件のグラフの 1 ~ 10 秒までの出力件数から、このシステムの限界性能は 95.7 件/秒であり、処理し切れなかった要求が累積していつてしまう。図 5 は入力した毎秒の要求に対してのレスポンス時間の内訳である。1 秒目の繰越分は徐々に処理されていくものの、毎秒約 5 件ずつの繰越が発生しているため、10 秒 1000 件より更に後ろへ伸びていく結果となった。このときの CPU 使用率は、図 6 にあるようにユーザプロセスとシステムで 100% になっている。95.7 件/秒の性能から更に向上させることを考えた場合、システムの処理が CPU 利用の大半を占めていることから、これは I/O などの処理であると考えられるため、プログラムの起動時に I/O が発生しないように、プログラムをスワップアウトさせないようにすればよい。

LoadRunner は MERCURY INTERACTIVE 社の登録商標です。

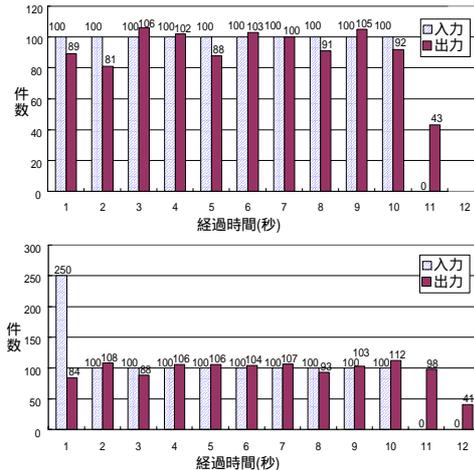


図 4 100 件/秒を 10 秒間かけた時の毎秒の処理件数 (上)、ピークを含むモデルの毎秒の処理件数 (下)

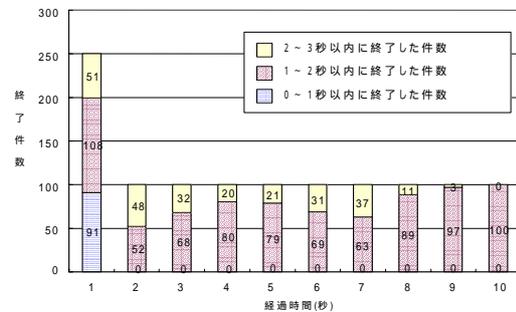


図 5 ピークを含むモデルのレスポンス時間内訳

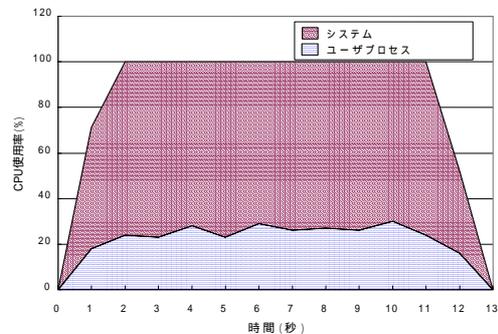


図 6 CPU 使用率

5. おわりに

TP1/Web の常駐プロセス数をチューニングすることによって、CGI の処理件数が 1 サーバ当たり 50 件/秒程度であったのに対して、95.7 件/秒まで向上させることが可能であることがわかった。

アプリケーションやトラフィックモデルによって、設定値は変化するものと考えられるため、具体的なシステム案件に対しても適合性を検証したい。

参考文献

- [1] <http://www.hitachi.co.jp/Prod/comp/soft1/0pentp1/product/index.html>
- [2] <http://www-svca.mercuryinteractive.com/products/loadrunner/>
- [3] <http://www.apache.org/>