

Java アプレットの性能評価

7N-6

青木裕司*、五十嵐史生*、野村立**、高橋岳宏**

三菱電機株式会社 *情報技術総合研究所、 **制御製作所

1.はじめに

WWW 技術の一般化が進み、オンライン監視システム等の産業向けシステムへの技術適用が望まれている。

産業向けシステムでは、接続クライアント数は一定という条件の下で、

- ・データの定期的な表示
- ・表示要求に対する応答性能の保証

などの要求を満たす必要がある。

テキスト表示やテーブル表示など HTML の機能範囲での性能は問題無いが、HTML に機能の無いグラフ表示や、図面表示などは生成処理が必要となり、上記の要求を満たすことができない可能性がある。WWW 技術の産業向けシステムへの適用の可否を評価するため、今回、グラフ生成について性能測定した。グラフの生成には、

・サーバ生成方式：

サーバ上の CGI プロセスによりグラフ描画し、GIF イメージファイルを生成する

・クライアント生成方式：

サーバ上の CGI プロセスによりデータを転送し、クライアントのブラウザ上でグラフを描画する

の二つの方式を比較した。

測定マシンの CPU 性能は以下の通りである。

- ・サーバ： Pentium100MHz×2
- ・クライアント：Pentium90MHz

Java applet performance evaluation

Yuji AOKI*, Fumio IGARASHI*,

Ritsu NOMURA**, Takehiro TAKAHASHI**

*Information Technology R&D Center,

**Power and Industrial Systems Center,

Mitsubishi Electric Corporation

2.サーバ生成方式とクライアント生成方式の比較

サーバ生成方式とクライアント生成方式について応答時間と CPU 負荷を測定し、比較した。

生成したグラフは 8 本のサインカーブであり、各グラフは約 900 個の点をプロットする。

サーバ生成方式の実現には GIF イメージファイルの生成に gd ライブラリを、クライアント生成方式の実現には Java アプレットを利用した。

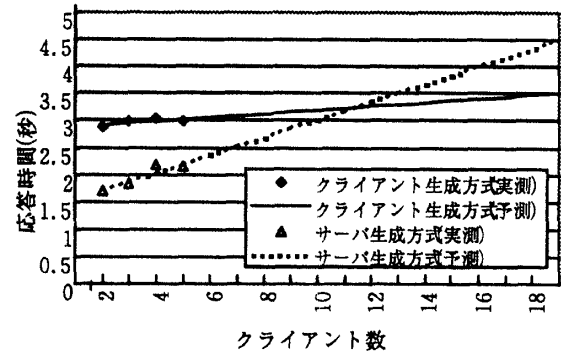


図 1 応答時間の比較

サーバ生成方式では、クライアント数の増加に伴う応答時間の増加が著しく、20 クライアント程度で 5 秒になると予想される。これに対し、クライアント生成方式では、応答時間の増加は比較的少なく、20 クライアントでも 4 秒以内の応答性能が確保できると予想される。

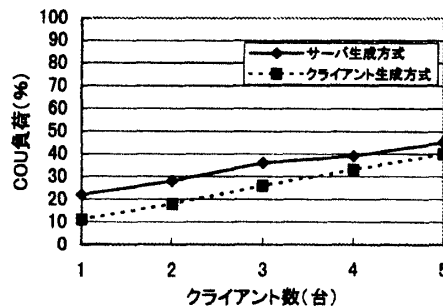


図 2 CPU 負荷の比較

この時のサーバの CPU 負荷は、図 2 に示すとおりである。サーバ生成方式に比べクライアント生成方式が若干低いが、ほぼ同じであり、約 10 クライアントで、CPU 性能の限界に達すると予想される。従って、クライアント生成方式で応答性能の確保ができて、サーバ CPU の負荷により、実際に処理できるクライアント数は制限されることになる。

3. データ転送方法の比較

前節で述べたように、クライアント生成方式により応答時間の保証は可能であるが、同時アクセスクライアント数の増加によるサーバの負荷増大が問題となる。このサーバの負荷はデータ転送に起因すると考えられる。前節のクライアント生成方式では、データ転送を行うために、CGI 機能によりデータ転送プロセスを起動している。図 2 で、サーバ生成方式とクライアント生成方式の CPU 負荷がほぼ同程度であることから、CGI によるプロセス起動のオーバーヘッドが大きいと推測される。従って、データ転送プロセスを常駐化し、起動時の負荷を減らすことで、データ転送の負荷軽減が可能であると予測される。データ転送プロセスを常駐させ、CPU 負荷を測定した。なお、前者を逐次起動方式、後者をプロセス常駐方式と呼ぶことにする。すなわち、

- ・ 逐次起動方式：

データ転送プロセスを CGI 機能により、表示要求毎に起動する

- ・ プロセス常駐方式：

データ転送プロセスをバックグラウンドプロセスとして常駐させておく

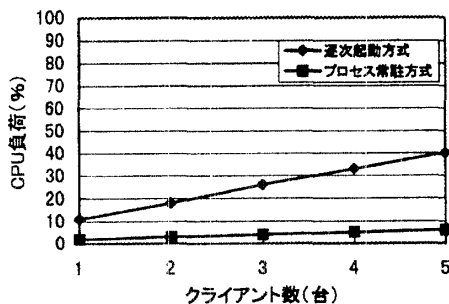


図 3 サーバ CPU 負荷の比較

両方式のサーバ CPU 負荷は図 3 の通りである。

逐次起動方式では、データ転送プログラムの起動負荷は 1 クライアントあたり 7% 程度であり、12 から 13 クライアントのアクセスで CPU 性能の限界に達することになる。これに対して、プロセス常驻方式では、1 クライアントあたりの負荷は 2% 程度であり、逐次起動方式に比べるとサーバ負荷の大幅な軽減が可能である。

データ転送量としては、逐次起動方式の場合、約 40KB、プロセス常驻方式の場合約 35KB であり、ネットワークの負荷としても、両方式に大きな差はない。

5. おわりに

産業向けシステムへの WWW 技術の適用において、

- ・ Java アプレットによるクライアント処理方式と
- ・ データ転送プロセスの常驻方式

により、サーバの負荷が軽減でき、表示要求に対する応答性能を確保したシステム構築が十分に可能なことがわかった。

産業向けシステムへのより有効な適用を行うためには、更新されるデータをオンラインで動的に表示する機能も求められ、今後はこのような機能の性能についても評価していく必要がある。

また、サーバの負荷軽減のために、WWW サーバソフトが CGI の代替機能として提供する、サーバで処理を行うインタフェースがあり、これらの機能が有効か評価していきたい。

参考文献

- [1] Thomas Boutell, "gd 1.2",
<http://www.boutell.com/gd/>