

## 論文

インターネットを利用したマルチクライアント型  
遠隔計測システム

内海富博\*・山口邦彦\*

## The Internet-Based Multi-Clients Remote Measurement System

Tomihiko Utsumi\*, Kunihiko Yamaguchi\*

## Abstract

The recent wide spread of the Internet have accelerated the development of distributed remote system for a variety of applications. In this paper, a Internet-based multi-clients remote measurement system having Client/Server architecture is developed. In the system, one or more clients can access to a server, and can plot a measurement data as soon as the clients receive it from the server. As Java applet allows more than one client to access to the server through Java compatible World Wide Web browser, the remote measurement operation in clients is independent of operating systems.

## 1. はじめに

近年、インターネットの急速な普及に伴い、これまで専用の回線や機器を用いていた遠隔システムが、インターネットへ接続可能な汎用機器を用いて低コストでシステム構築することが可能となっている。また、インターネットは時間や場所に制限されずに利用可能であるため、遠隔教育<sup>(1)</sup>、遠隔診断<sup>(2)(3)</sup>そして遠隔計測<sup>(4)(7)</sup>等の分野でインターネットを利用したシステム構築に関する事例が報告されている。

中野ら<sup>(1)</sup>は、Java 言語を利用して、インターネットに接続された一般的なパーソナルコンピュータ(PC)の WWW ブラウザ上で、遠く離れた実験装置を遠隔操作し物理実験を行うことを可能とするサーバクライアントシステムを報告している。同様の分散型データ収集システムは Grimaldi ら<sup>(4)</sup>と Kalaitzakis ら<sup>(5)</sup>によっても研究されている。また、月橋ら<sup>(2)</sup>は遠隔診断のためにデータの共有化が可能サーバクライアントシステムを Java 言語を用いて作成し、Hong ら<sup>(3)</sup>は同様のシステムを Active X を用いて作成している。Wieslaw と Michal<sup>(8)</sup>は Java 言語を用いて分散計測システム設計用のツールを作成した。一方、川原ら<sup>(6)</sup>と小山ら<sup>(7)</sup>は Microsoft Visual C++を用いて、計測器の制御をクライアント間直接通信で行い、計測後のデータ処理をサーバクライアント間で行う遠隔計測システムを構築した。

計測器の遠隔制御を行う場合、インターネット上にある PC 間は 1 対 1 で接続する必要がある。これは複数のクライ

アントが 1 台の計測器を同時に制御することを防ぐためである。しかしながら、場所に制限されずにあらゆる場所からインターネットを通じて、計測中のデータをリアルタイムに観測したい場合には、複数のクライアントの同時アクセスに対応する必要がある。筆者らは、複数のクライアントからの同時アクセスに対応するため、Java アプレットを用いて遠隔データ収集システムを構築した<sup>(9)</sup>。そこでは、計測データをサーバにファイルとして保存し、クライアントの Java アプレットからデータファイルを読み込む方式とした。しかしながら、長時間の計測によりデータ数が増えファイルサイズが大きくなると、クライアントがサーバからデータファイルを読み込み、グラフの描画を行う際に時間がかかってしまう。

そこで本研究では、更新されたデータのみをクライアントへ送信することによってグラフ描画の遅れを解消し、リアルタイムで計測データの取得を可能とした。また、計測器の制御を遠隔では行わず、サーバ側で計測を開始することで、複数のクライアントの同時アクセスに対応することを可能とした。そのため、一方のクライアントでは現在計測中のデータをモニタし、もう一方のクライアントでは、過去の計測データをモニタすることも可能である。また、計測したデータを処理する場合は、クライアント用 Java アプレットに処理機能を含めることで、サーバへの負担を軽減できる。本研究で構築したシステムを電気炉の温度変化の計測に適用し、システムの有用性について検討を加えた。また、これまでの報告例では、計測データ取得に関するリアルタイム性については検討されていないため、計測データ取得時間間隔の設定値と実測値からそのリアルタイム性について検討を行った。

2004年7月20日受理

\*秋田大学工学資源学部情報工学科, Department of Computer Science and Engineering, Faculty of Engineering and Resource Science, Akita University.

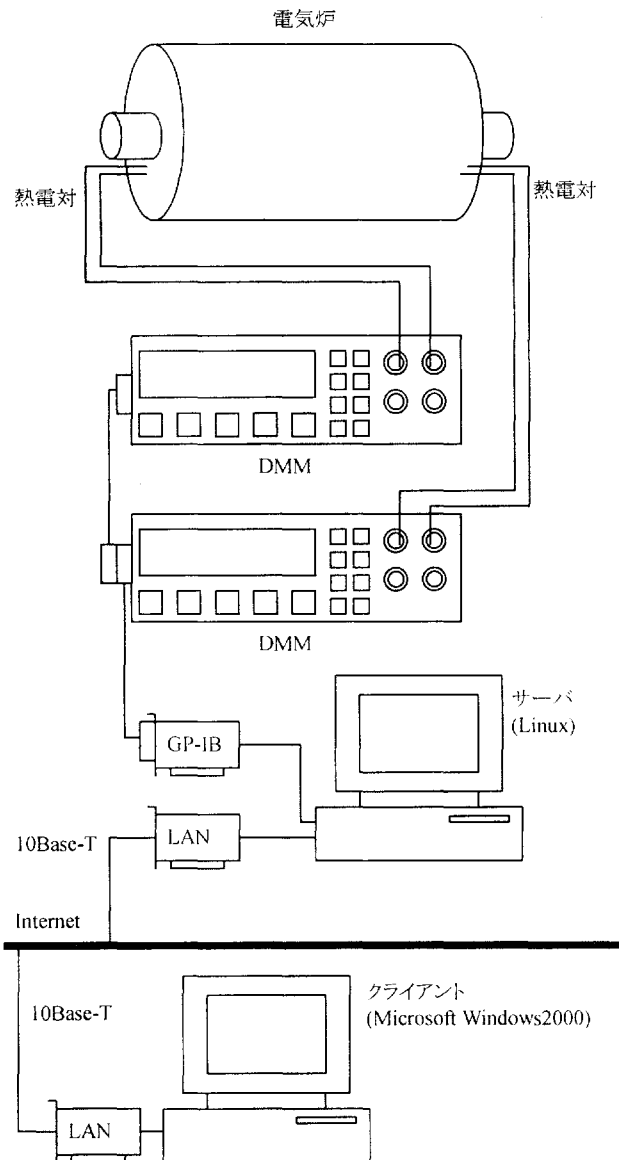


Fig.1 System configuration of the Internet-based Client/Server system for remote measurements.

## 2. システム概要

本研究で構築したシステムの構成を Fig.1 に、クライアント-サーバ間のデータ通信方法を Fig.2 にそれぞれ示す。本システムは、クライアントとサーバの二種類の PC で構成されており、サーバで計測を行う。サーバでは、はじめに WWW サーバと計測用サーバプログラムを起動する。計測用サーバプログラムは GP-IB インターフェースで接続したデジタルマルチメータ (DMM) を制御し、電気炉に設置した熱電対の熱起電力を読み込み、メモリ上に書き込むと共にデータファイルに保存する。さらに、クライアント用 Java アプレットからアクセスがあると、データ送信用スレッドを生

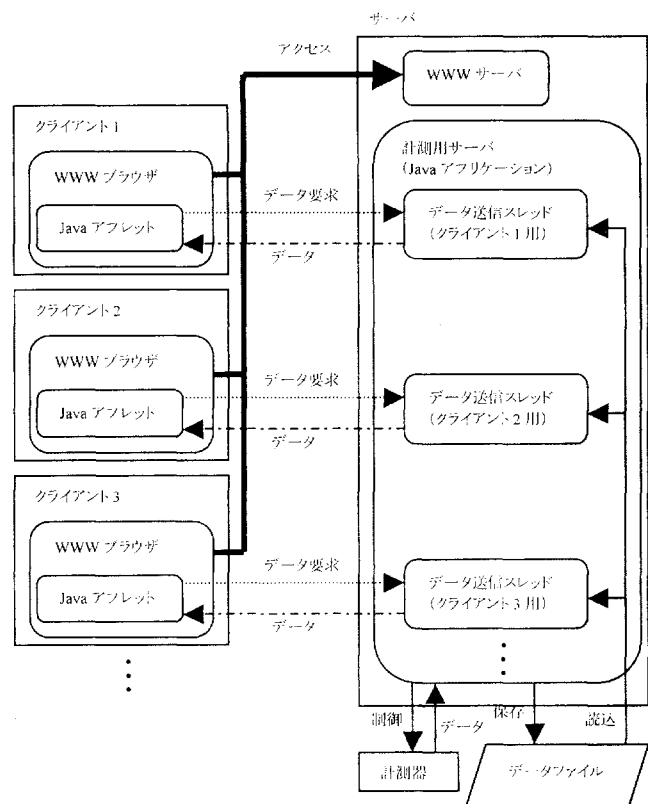


Fig.2 Method for data communications between server and client.

成し、クライアントメモリ上のデータを送信する。データ送信用スレッドは、クライアント毎に生成されるため、複数のクライアントが同時にアクセスした場合にも対応することが可能である。また、サーバは最新のデータを 100000 個までメモリ上に記憶することができる。一方、クライアントは WWW ブラウザを用いて WWW サーバへアクセスし、サーバ上にあるクライアント用 Java アプレットをダウンロードする。Java アプレットはクライアントへダウンロードされた後、WWW ブラウザ上で実行される。その後、計測用サーバへ接続し、現在計測中のデータまたは過去に計測したデータファイルを取得し、グラフ描画を行う。また、現在計測中のデータを取得する場合、サーバが新しくデータを取得する毎にクライアントがデータを受信し、クライアントのグラフに追加描画する。

### 2.1 ハードウェア

本研究で用いた PC の仕様を Table 1 に示す。以下、サーバシステムおよびクライアントシステムについて説明する。

#### 2.1.1 サーバ

サーバでは、DMM の制御およびデータ取得のためのインターフェースとして Agilent Technologies 社製 HP82335

Table 1 Specification of Server and Client PCs

	サーバ	クライアント
CPU	AMD 5x86 166MHz	Intel PentiumIII 933 MHz
RAM	SIMM 32MB	RIMM 128MB
HDD	SCSI 1.2GB	IDE 10GB
OS	Linux	Microsoft Windows2000

Table 2 Software used in server.

OS	Linux (Slackware 3.5)
WWW サーバ	Apache ver. 1.3
開発環境	Java for Linux ver. 1.2 (Black Down Porting Team) GP-IB Library (The Linux GP-IB Package ver. 2.02)

GP-IB インターフェイスボードを使用した。データ転送速度は300KB/sである。DMMはKEITHLEY社製Model197(最大測定タイミング:3回/s)およびADVANTEST社製TR6851(最大測定タイミング:100回/s)を使用した。これらのDMMはGP-IBインターフェイスを備えているため、データを直接PCに取り込むことが可能である。本研究では、サーバを10Base-Tで学内LANに接続した。電気炉の温度計測には、クロメルーアルメル熱電対を使用した。

### 2.1.2 クライアント

クライアントは、Javaに対応したWWWブラウザが動作するPCでLANに接続してあれば良い。本研究では、OSとしてMicrosoft Windows 2000、WWWブラウザにはMicrosoft Internet Explorer 6を使用した。

### 2.2 ソフトウェア

Fig.3に本システムのソフトウェアの構成を示す。計測用サーバプログラムは、Javaアプリケーションであり、DMMの制御および取得データのデータファイルへの保存、データ送信用スレッドの生成を行う。クライアント用Javaアプレットは、クライアントのWWWブラウザ上で動作し、サーバのデータ送信用スレッドと通信を行い、データを受信する。

#### 2.2.1 サーバソフトウェア

Table 2にサーバで用いたソフトウェアを示す。OSにはGPL(GNU General Public License)に従うフリーソフトウェアであるLinux(Slackware 3.5)を、WWWサーバソフトウェアにはApache ver. 1.3を使用した。また、今回用いたGP-IBインターフェイスボードをLinuxで使用するため、Shroeterが開発し、The Linux Lab Projectで公開しているThe Linux GP-IB Package ver. 2.02を使用した。このパッケージにはデバイスドライバおよび開発用パッケージが含ま

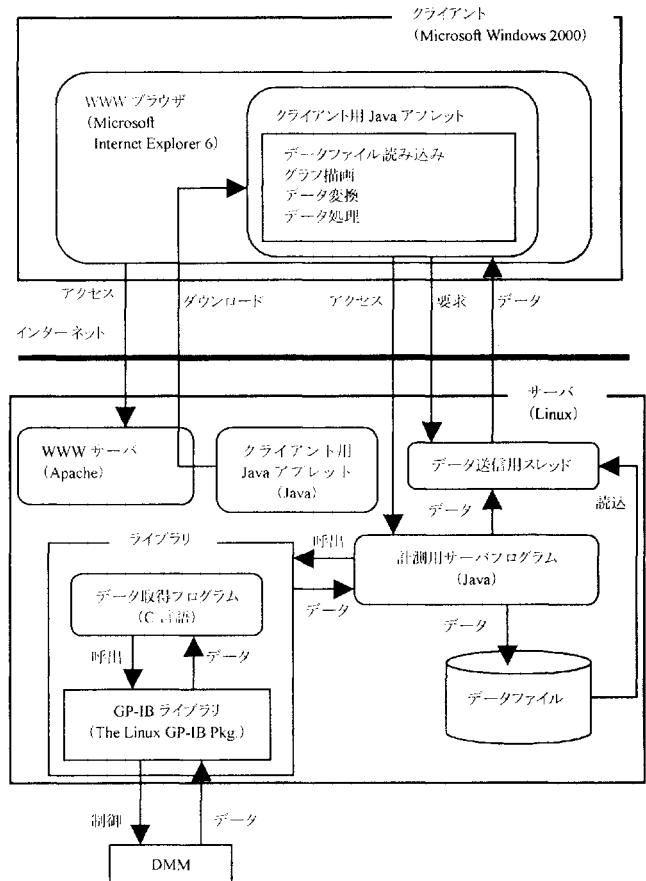


Fig.3 Software configuration for client/server system.

れている。このパッケージはC言語用であるため、本システムをJavaで構築する際には、直接使用することができない。そこで、最初にC言語でGP-IB制御を行う関数を作成し、ライブラリ化する。その後、Java Native InterfaceによりライブラリをリンクすることでJavaからGP-IBに接続したDMMを制御することができる。CコンパイラにはGPLに従うgcc ver. 2.7.2.3を使用した。また、計測用サーバプログラムはBlack Down Porting Teamが開発したJava for Linux ver. 1.2を用いて作成した。

計測用サーバプログラムは、設定したデータ取得時間間隔( $\tau_s$ )でDMMから計測データを取得し、順次データファイルへ保存する。データファイル名は年月日および時刻から自動的に生成し(例えば、2004年7月20日13時なら2004-07-20-13.dat)、1時間毎にデータファイル名を変えて計測データを保存する。DMMからデータ取得を行う処理のフローチャートをFig.4に示す。DMMからデータ取得する際には、データ取得直前のシステムクロックT1とデータ取得後のシステムクロックT2を比較し、データ取得前後のシステムクロックの差が $\tau_s$ 以上になるまでT2の取得を繰り返す。ここで、データを読み込む時間間隔は、DMMが

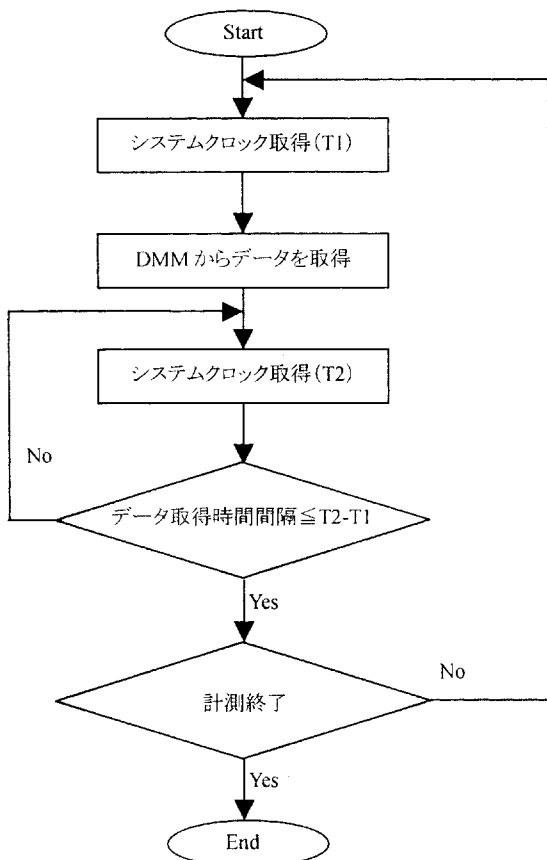


Fig.4 Flowchart for data acquisition.

A/D 変換しサーバのメモリへ書き込むまでの時間 ( $\Delta\tau$ ) を考慮しなければならない。計測データをリアルタイムに取得するためには、Fig.5 に示すように  $\tau_s$  を  $\Delta\tau$  より長く設定する必要がある。もし、 $\Delta\tau$  よりも  $\tau_s$  が短い場合は、DMM からサーバへデータを取り込む処理を終えるまで待機し、データ取得後次の処理へ移るため、実際のデータ取得時間間隔は  $\tau_s$  よりも長くなる。

### 2.2.2 クライアントソフトウェア

クライアントでは、OSとしてMicrosoft Windows2000を使用した。また、クライアント用 Java アプレットの作成には、Sun Microsystems 社の Java2 SDK Standard Edition ver. 1.4.2\_01 を、Java 対応の WWW ブラウザとして Internet Explorer 6 を使用した。

クライアントがサーバから受信するデータは DMM から取得した電圧値であるため、今回の計測のように温度を測る場合は電圧値から温度へ変換する必要がある。そのため、クライアント用 Java アプレットに電圧から温度へ変換する処理を用意し、クライアント側でデータ処理することとした。このことにより、必要なデータ処理をクライアント PC で行うため、サーバへの負担は軽減される。

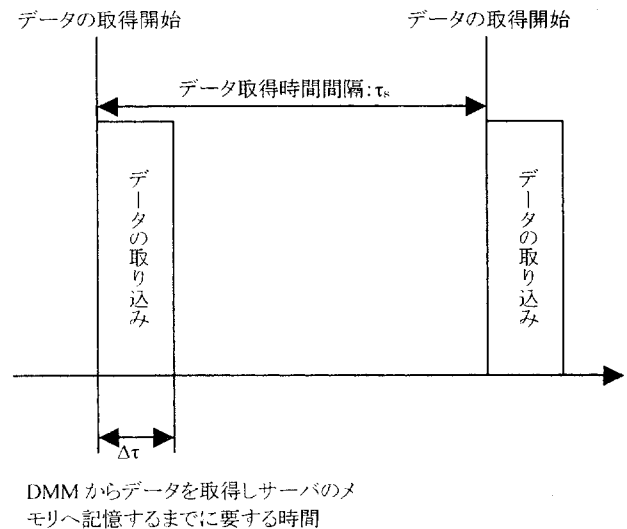


Fig.5 Data acquisition time between two successive events.

### 3. 実験結果および考察

はじめに、設定した通りのデータ取得時間間隔  $\tau_s$  でデータが取得できているか確認するため、 $\tau_s$  を 5s, 1s, 0s と変え、データ取得時間間隔の実測値  $\tau_e$  の  $\tau_s$  依存性を調べた。ここで、 $\tau_s=0s$  では DMM がデータ取得後、直ちに次のデータ取得を行う。Fig.1 において電気炉の電源を入れず温度が一定の状態ではデータ数が 2000 個まで測定を行った。  $\tau_s$  を変えて測定した  $\tau_e$  のグラフを Fig.6(a), (b), (c) に示す。2 台の DMM を用いて順番に 1 台ずつデータ取得を行っているため、測定タイミングは遅い方の DMM のタイミングになる。Fig.6(c) より  $\tau_s$  が 0s の場合、DMM (Model 197) のデータ取得タイミングが 3 回/s であるため約 300ms でデータ取得が可能であるものの、実際の  $\tau_e$  は約 300ms を中心に 100~500ms の間で変動しており、必ずしも DMM の最大データ取得タイミングでデータを取得することができない。これは OS のタイムシェアリング処理によってバックグラウンドで起動している他の処理にリソースが奪われるため、他の処理の動作状況がデータ取得処理に影響を及ぼしたためと思われる。一方、 $\tau_s$  が 1s, 5s では設定したデータ取得時間間隔より  $\tau_e$  の時間変動が小さい。その上  $\tau_s$  から大きくなれば生じる回数も、 $\tau_s=1s$  でデータ 1000 個あたりに 1 個、5s でデータ 2000 個あたりに 1 個と少ない。したがって本研究では  $\tau_s$  が概ね 1s 以上であればデータをリアルタイムに計測することが十分可能であることがわかる。また、データ数 10000 個、1 データあたり 80Byte のとき、クライアントがサーバへデータ要求を行ってから、データの受信が完了するまでの時間を計測した結果、1 データあたりの平均値は約 4ms であった。よって同一 LAN 内の使用では 10Base-T に

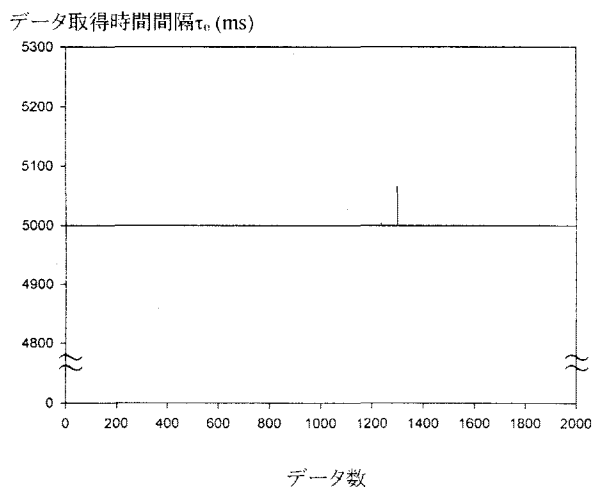
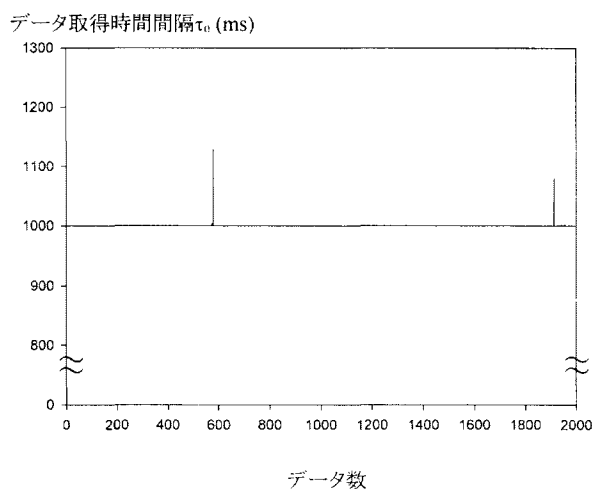
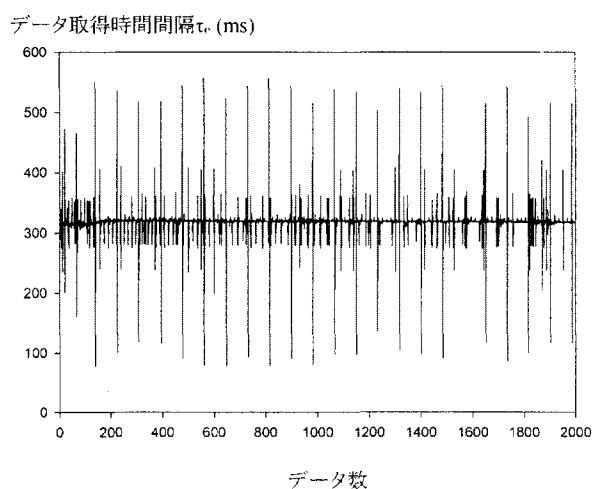
(a) データ取得時間間隔 $\tau_s$  : 5s(b) データ取得時間間隔 $\tau_s$  : 1s(c) データ取得時間間隔 $\tau_s$  : 0s

Fig.6 Measured acquisition time  $\tau_c$  for (a)  $\tau_s=5s$ , (b)  $\tau_s=1s$ , (c)  $\tau_s=0$ , where  $\tau_s$  is the value of setted acquisition time between two successive events.

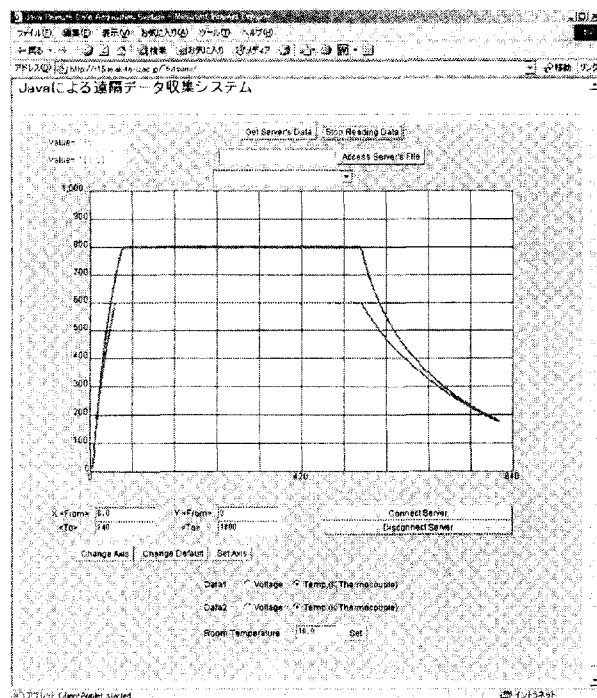


Fig.7 Experimental results on time dependence of furnace temperature.

において、 $\tau_s$ が1s以上であればシステムとしてのリアルタイム性は保証できる。

本研究で構築したシステムを電気炉の温度変化の計測に適用した。測定条件として、電気炉の高温側を800℃、低温側を600℃に設定し、 $\tau_s$ を1sとして計測を行った。はじめに、サーバ側の計測用サーバプログラムを実行し、データの取得を開始する。次に、クライアントPCからWWWブラウザを使用し、サーバへアクセスを行うことで計測を行う。840分間(データ数:50400個)の測定を行い、データファイルが1時間毎に計14個生成され、データが保存されていることを確認した。Fig.7に計測結果を示す。500分を過ぎた時点で、電気炉の電源を切ったため途中から温度が下がっていることがわかる。また、複数のクライアントの接続に対しても、同時に6クライアントまでの動作を確認した。データの表示に関しては、電圧、温度どちらでも表示することが可能であるため、計測対象や計測機器を変えることで汎用的に使用することが可能である。

#### 4. むすび

本論文では、Javaを用いて複数のクライアントの同時アクセスに対応した遠隔計測システムの構築を行った。実際に電気炉の温度変化の計測に適用したところ、十分に対応できることを確認した。また、複数のクライアントのデータを取得し、もう一方で過去のデータを取得できる。さらに、

データ取得時間間隔は機器の性能に依存しているため、データ取得時間間隔 $\tau_s$ を機器に合わせて変えることで、リアルタイムにデータを取得できることを明らかにした。本システムでは、データ取得時間間隔を 1s 以上に設定することで、リアルタイムに計測データを取得することが可能である。

#### 参考文献

- (1) 中野裕司, 岩橋克聰, 柏本史郎, 松尾進, 森昌弘 (1998): インターネットを利用した物理実験教育システム, 大学の物理教育, 2号, pp. 44-48.
- (2) 月橋こずえ, 中村亨弥, 木竜徹, 斎藤義明 (1998): Javaによる遠隔診断のためのサーバクライアントシステム, 信学論(D-II), Vol. J81, No. 10, pp. 2494-2497.
- (3) Hong Zeng, Ding-Yu Fei, Cai-Ting Fu, Kenneth A. Kraft (2003): Internet(WWW) based system of ultrasonic image processing tools for remote image analysis, *Computer Methods and Programs in Biomedicine*, Vol.71, Issue 3, pp.235-241.
- (4) D. Grimaldi, L. Nigro, F. Pupo (1998): Java-based distributed measurement systems, *IEEE Trans. Instrum. Meas.*, Vol.47, No.1, pp.100-103.
- (5) Kostas Kalaitzakis, Eftichois Koutroulis, Vassilios Vlachos (2003): Development of a data acquisition system for remote monitoring of renewable energy systems, *Measurement*, Vol.34, Issue 2, pp.75-83.
- (6) 川原勝広, 三宅基史, 勝山豊 (1999): Web環境でクライアント間直接通信を利用した遠隔測定システム, 信学論(B), Vol. J82-B, No.10, pp.1942-1944.
- (7) 小山長規, 勝山豊 (2001): WWW環境での遠隔測定/計算システム, 信学論(B), Vol. J84-B, No. 2, pp.303-305.
- (8) Wieslaw Winiecki, Michal Karkowski (2002): A New Java-Based Software Environment for Distributed Measureing Systems Design, *IEEE Trans. Instrum. Meas.*, Vol.51, No.6, pp. 1340-1346.
- (9) 内海富博, 神戸祐史, 山口邦彦, 三浦茂 (2001): Javaを用いた遠隔データ収集システム, 秋田大学工学資源学部研究報告, 22号, pp. 61-65.