

計測・制御用簡易メッセージ配信システムの開発

高エネルギー加速器研究機構 小菅 隆、齊藤裕樹、伊藤健二

Abstract

最近ではネットワーク機器やパーソナルコンピュータ(以下 PC)の低価格化が進んでおり、小規模な計測・制御システムにおいても、複数台の PC を連携させた安価で効率的なシステムが構築可能である。複数の PC を連携させるためにはネットワークを利用したアプリケーションプログラム間の通信が不可欠であり、今回我々は、アプリケーションプログラム間のメッセージ配信が簡単に行えるシステム (STARS: Simple Transmission and Retrieval System) を開発した。ここでは、この STARS の詳細及び実際に高エネルギー加速器研究機構放射光研究施設ビームラインインターロック集中管理システム[1][2][3]に STARS を導入した結果について報告する。

1. はじめに

大規模な計測・制御システムの世界では、オブジェクト分散の技術を利用したシステム構築が盛んに行われているが、ネットワークを利用したシステム構築に際しては、小規模システムにおいてもプログラムの分散化は有益である。CORBA や DCOM の利用は小規模なシステムにおいても有効であるが、様々なオペレーティングシステム(以下 OS)や開発言語を利用する場合には、システム構築作業の大部分が CORBA、DCOM 導入のための作業となる事が予想される。これらの要求や問題点を考慮した結果、我々は簡単にアプリケーション間のメッセージ配信を行える小規模システム向けの通信機構が必要であるという結論に至った。

STARS におけるアプリケーション間通信は、全て TCP/IP Socket を利用したテキストベースのコマンドの送受により行われる。この事でシステムの扱いが非常に容易となり、それと同時に、開発言語や OS の選択の幅が広がる。また、デバッグを行う場合にも TELNET などのツールを利用することが可能である。さらに、コアの部分となるプログラムは Perl を使って開発されており、STARS は様々なプラットフォーム上で動作可能である。

2. STARS の詳細

STARS はユーザが作成するクライアントプログラム、システムの中心的存在となりメッセージの配信を行う TAK (Transferring Agent Kernel) サーバおよび他のシステムとの接続を行う場合や、複数の TAK サーバを接続する場合に使用する Bridge から構成される(図 1 参照)。TAK サーバは Perl によって記述されており、STARS 利用においては、あらかじめ TAK サーバを運用する PC に Perl (Version 5) がインストールされている必要がある。Perl は Linux、FreeBSD、Windows のような様々な OS で利用可能であり、TAK サーバも様々な OS 上で動作可能である。各クライアントプログラムは TCP/IP Socket を使用して TAK サーバに接続する。クライアントプログラムに関しては TCP/IP Socket を利用できれば OS や開発言語に特別な制限は無い。

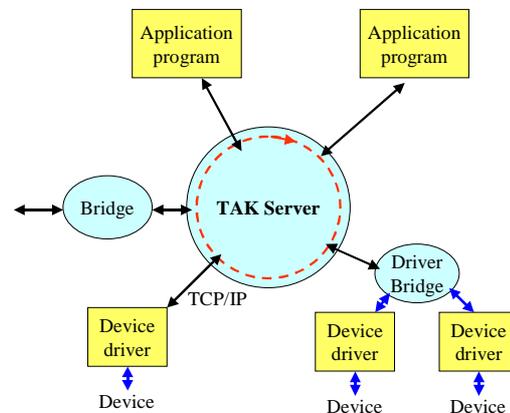


図 1. STARS の構成

2.1 クライアントの接続及びキーワードによる認証

クライアントプログラムが TAK サーバに接続する際の手順は次の通りである。

- サーバに TCP/IP Socket を使用して接続。
- サーバから任意の数値を示すキャラクターが送られて来る。
- サーバから送られてきた数値をもとにキーワードを決定。
- サーバに自ターミナル名とキーワードを送信。
- 接続が完了。

STARS はクライアントの接続に際して次のようなキーワードによる簡単な認証システムを有している。まず、TAK サーバに接続されるクライアントプログラムはあらかじめ TAK サーバ上に登録されていなければならない。また、TAK サーバは登録されたクライアントプログラム用のキーワード文字列の配列を持っており、クライアントが接続してくるとそのキーワード配列中の何番目のキーワードを使用するかクライアントに対して指定する。クライアントプログラムは接続の際この数値に対応したキーワードを送信しなければならない。

2.2 メッセージの転送

TAK サーバに認証され、接続された全ての STARS クライアントプログラムはデータ転送に関して同等な権利を有する。もし、アクセス制限などの機能が必要な場合は各クライアントで独自に用意する必要がある。

授受されるデータは全てキャラクターベースで、デリミタは LF (0x0A) である。実際のデータ転送の方法としては、たとえば Term1 と名づけられたクライアントプログラムから Dev1 という名のクライアントプログラムにデータを転送する場合を例にすると図 2 に示す通りになる。はじめに Term1 は「Dev1 setdata 123」のようなコマンドを TAK サーバに送出する。ここで「Dev1」は宛先であり、「setdata 123」はメッセージである。その後、Term1 よりメッセージを受け取った TAK サーバは Dev1 に対して「Term1>Dev1 setdata 123」のような送り主を示す文字列を行頭に付加したメッセージを送信する。

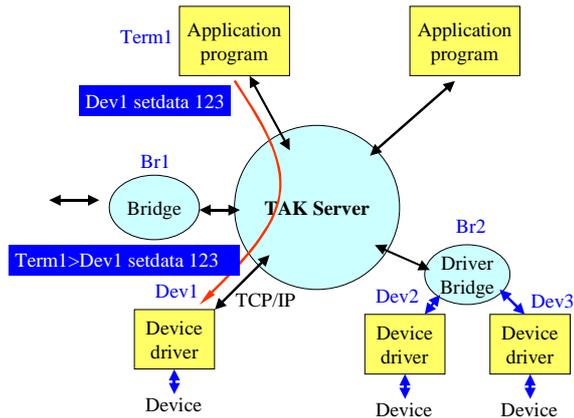


図 2. メッセージの転送

2.3 STARS のメッセージに関する規定

TAK サーバは、受け取ったメッセージを単に指定された宛て先に送信する。STARS ではこれらのメッセージに、ある程度取り決めを設けて効率的な動作を行うようにしている。実際には“@” (アットマーク) で始まるメッセージは何かのコマンドに対する回答であることを示し、“_” (アンダースコア) で始まるメッセージは何かのイベントが発生したことを示す。それ以外のメッセージはコマンドである。

2.4 イベント配信機能

前述の通り STARS において“_” から始まるメッセージはイベントメッセージであるが、TAK サーバはイベント配信先が不確定の場合や配信先が複数の場合に有用なイベント配信機能を有している。

イベントを発信するクライアントはあて先を System にしてイベントメッセージを送出する。その後 TAK サーバは、あらかじめ flgon コマンドにより配信の依頼を行っているクライアントに対してイベントメッセージを送出する。

3. パフォーマンス

ここでは STARS のデータ転送速度に関するテスト結果について報告する。テストの内容は、1つのクライアントプログラムを TAK サーバに接続し、そのクライアントプログラムから配信先を自分宛てに設定してメッセージを送信し受信するというものである。一度に送るメッセージについては約 90Bytes の場合と約 1.8kBytes の場合について行った。なお、速度を算出するために、この送受信を 10 万回繰り返しその時間を計り、速度を割り出した。表 1 にネットワーク (10Base-T) を介してデータを転送行った結果と、同一 PC 内でデータ転送を行った結果を示す。

TAK サーバを動作させるために使用した PC のスペックは「CPU: Intel Celeron 500MHz, Memory: 64Mbytes, OS: FreeBSD」であった。また、今回は残念ながら全く同じスペックの PC を準備することができなかった為、クライアント用の PC には「CPU: Intel Pentium III 500MHz, Memory: 256Mbytes, OS: Windows2000」を使用した。その為ネットワークを利用した場合とそうでない場合の転送速度の違いを正確に知ることはできなかったが、転送速度に関する概要を把握する事ができた。

テストの結果、ネットワークを介しても最高で約 349kBytes/秒のデータ転送速度が得られることが分かった。また、一度に送るデータの量が少ない場合、転送速度が低下 (但し、1つのメッセージを処理する速度は向上) する事が分かった。この結果から STARS においてある程度大きなデータを転送する際には、一度に転送するデータ量を大きくすれば良いことが分かる。

表 1. データの転送速度

	1つのメッセージを 処理する時間 (m秒)	データ転送速度 (Bytes/秒)
同一 PC 内 メッセージ長 93Bytes	0.870	106,896.552
同一 PC 内 メッセージ長 1893Bytes	2.120	892,924.528
10Base-T 経由 メッセージ長 93Bytes	1.600	58,125.000
10Base-T 経由 メッセージ長 1893Bytes	5.420	349,261.993

4. ビームラインインターロック集中管理システムでの利用

高エネルギー加速器研究機構放射光研究施設 2.5GeV Ring には現在 20 を超すビームラインがあり、各々のビームラインには、放射線安全、ビームライン真空の保持及びビームライン機器の保護を目的としたビームライン・インターロックシステム（以下 BLIS）が設置されている。各 BLIS はビームラインインターロック集中管理システム（以下、集中管理システム）に接続され、集中管理システムは全ての BLIS を統合的に監視・制御している。集中管理システムは各 BLIS との信号の授受を行うインターフェース部と数台の PC から構成されており、COACK（Component Oriented Accelerator Control Kernel）[4][5][6]と呼ばれる加速器用に開発された制御システムにより管理・運転されている。今回、我々は各ビームラインとの信号の授受を行うインターフェース部に STARS を利用した。

4.1 インターフェース部の構成

集中管理システムインターフェース部のハードウェア及びソフトウェアの構成は図 3 に示す通りである。光ファイバーを通して送られてくる各 BLIS からの運転状態を示す信号は、シリアル/パラレルコンバータを介してインターフェース用 PLC（Programmable Logic Controller）に、またインターフェース用 PLC からの制御信号は分配器と光ファイバーを通して各 BLIS に送出される。インターフェース用 PLC とインターフェース用 PC の間は RS-232C により接続されており、インターフェース用 PC 上では STARS が動作している。今回作成したクライアントプログラムは、PC のシリアルポートを通して PLC と直接通信を行うインターフェースドライバー、変化情報を記録するためのログ記録クライアントである。

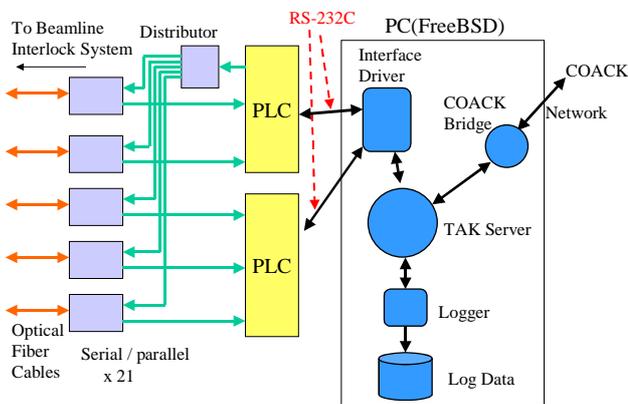


図 3. 集中管理システムインターフェース部の構成

4.2 COACK Bridge

STARS と COACK とのインターフェースの働きをする部分が COACK Bridge である。COACK Bridge は STRS 内のクライアントプログラムからのメッセージを XML（Extensible Markup Language）メッセージに変換し COACK に送出する。また、COACK から送られてくる XML メッセージを解析し STARS 用のメッセージに変換、指定されたクライアントに向けて送出する。この仕組みにより COACK 側では全ての STARS クライアントプログラムを単に COACK のノードとして使用することができる。

4.3 導入結果

STARS は 2000 年 10 月から集中管理システムに利用されているが、これまで非常に安定に動作している。また、ログデータの記録も良好で、ビームラインに異常が発生した際の対処に有効な情報を提供することができた。

5. まとめ及び今後の予定

今回我々は以上のようなアプリケーションプログラム間のメッセージ配信が簡単に行えるシステム STARS を開発した。また、ビームラインインターロック集中管理システムに利用し、良好な結果を得ることができた。現在 STARS は一部のビームライン制御システムに導入されることが予定されており、様々なシステムに対応するためクライアントプログラムの開発が続行されている。

参考文献

- [1] 小菅隆, 斉藤裕樹, 伊藤健二, “放射光ビームライン・インターロック集中管理システム”, 核融合科学研究所技術研究会報告 (1991) 172
- [2] 斉藤裕樹, 小菅隆, 伊藤健二 LAN を用いたインターロックシステムの監視 核融合科学研究所技術研究会報告 (1994) 228
- [3] 小菅隆, 斉藤裕樹, 伊藤健二, “放射光ビームライン・インターロックシステムとネットワーク” KEK Proceedings 95-14 (1996) 23
- [4] I. Abe, et al., "COACK-II PROJECT ON ACCELERATOR CONTROL KERNEL DEVELOPMENT", ICALEPCS'99, Trieste, 1999
- [5] T. Kosuge, et al., "COACK APPLICATION FOR THE BEAMLINE INTERLOCK SYSTEM AT THE PHOTON FACTORY", PCaPAC2000, Hamburg, 2000
- [6] I. Abe, et al., "Recent status on COACK project", PCaPAC2000, Hamburg, 2000