

解説

SPring-8 蓄積リング制御システム

田中 良太郎

SPring-8, (財)高輝度光科学研究センター・放射光研究所*

Control System of the SPring-8 Storage Ring

Ryotaro TANAKA

Accelerator Division, SPring-8

Control system of the storage ring is designed adopting the so-called Standard Model concept. The VMEbus is used as the front-end control system and widely distributed along with a 1436 m circumference of the storage ring. The VMEbus is controlled by a board computer with a RISC processor which is connected to the FDDI backbone network through the FDDI-Ethernet switching hub. CPU boards are operated by using LynxOS based HP-RT real-time operating system. The remote I/O system (RIO) interconnected by optical fibers is used as the field bus. The operator console system consists of UNIX workstations with human oriented interface built by using Motif based GUI builder. The control software is designed based on the event-driven client/server scheme. The remote procedure call (RPC) is used for communication between machine control applications over the network. Accelerator operation history and equipment parameters are stored on the standard database by using a RDBMS.

1. はじめに

大型放射光施設 (SPring-8) は世界最大級の放射光研究施設であり, 挿入光源を主とした短波長の高輝度光源により, 幅広い分野の精密な研究を推進することが目的である。SPring-8 には約30 m の長直線部, 1000 m におよぶ長尺ビームラインなどの特徴もあり, 総数61本にのぼるビームラインの利用が期待されている。施設は現在, 兵庫県西播磨の播磨科学公園都市に建設中であり (図1), 1997年2月に予定されている蓄積リングのコミッショニングに向けて建設が加速されている¹⁾。図2に示すように, この施設の加速器は,

線型加速器, シンクロトロン, そして蓄積リングの3つの加速器で構成される。

前段の線型加速器で発生された電子または陽電子は主線型加速器によって1 GeV に加速され, シンクロトロンで8 GeV まで加速された後, 蓄積リングに入射される。蓄積リングに入射された電子または陽電子は周長約1500 m の円周軌道上に周回蓄積される。蓄積リングでは円周に沿って配置してある偏向電磁石と挿入光源機器により放射光が発生され, ビームラインを通して実験ホールに導かれ実験に供用される。蓄積リングから短波長の高輝度放射光を安定に発生させるには, 蓄

* SPring-8, (財)高輝度光科学研究センター・放射光研究所
〒678-12 兵庫県赤穂郡上郡町 SPring-8
TEL 07915-8-0868 FAX 07915-8-0850
e-mail tanakar@sp8sun.spring8.or.jp



Figure 1. A bird's-eye view of the SPring-8 facility. The facility is under construction.

積リングは低エミッタンスの良質な電子または陽電子を安定に長時間蓄積する必要がある。

加速器制御系の目的は構成する各種機器を適切に制御し、高輝度かつ安定な放射光ビームを得ることである。大型放射光施設の加速器制御系は、線型加速器制御系、シンクロトロン制御系、蓄積リング制御系、ビームライン制御系、中央制御系、の5系で構成される。

施設は建設中のこともあって詳細が未定の部分もある。この解説ではすでに決定されているものと、予定されているものについて述べてあるので、場合によっては実際のシステムと相違する可能性がありうるが、この点をご了解願いたい。また、筆者は蓄積リング制御系と中央制御系を担当しているために、今回の解説はこれらの制御系について限らせていただき、線型加速器制御系、シンクロトロン制御系、ビームライン制御系の詳細については今回は範囲外とさせていただきます。

2. 蓄積リング制御系

2.1 制御系設計方針

SPring-8の制御系設計方針は以下のようになっている。

- 1) 分散処理系を高速のネットワークで結ぶ。
- 2) 線型加速器、シンクロトロン加速器と蓄積リングの各制御系は、加速器の建設時期の違いに影響されないように、お互いに緩やかに結合する。
- 3) すべての加速器を一箇所で運転できるようにし、これを少数のオペレータで行う。
- 4) できるだけ業界標準のハードウェアとソフトウェアを用いて建設する。
- 5) アプリケーション・プログラムを効率良く作成するために、CASE ツールを用いる。

2.2 制御系概略

SPring-8の制御系は、すでに述べられた制御

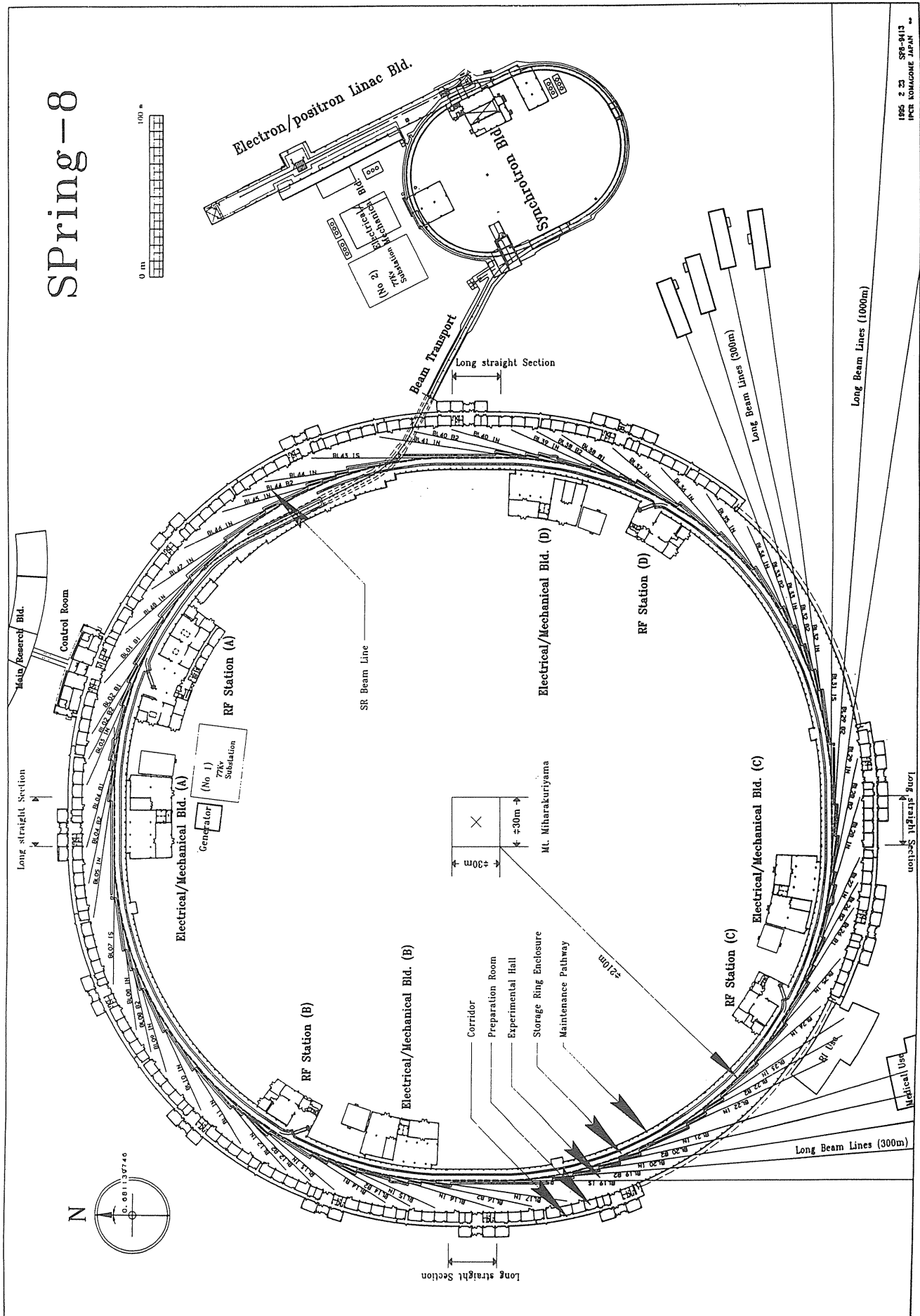


Figure 2. The schematic view of the accelerator complex of the Spring-8.

Control room

L,Sy control system

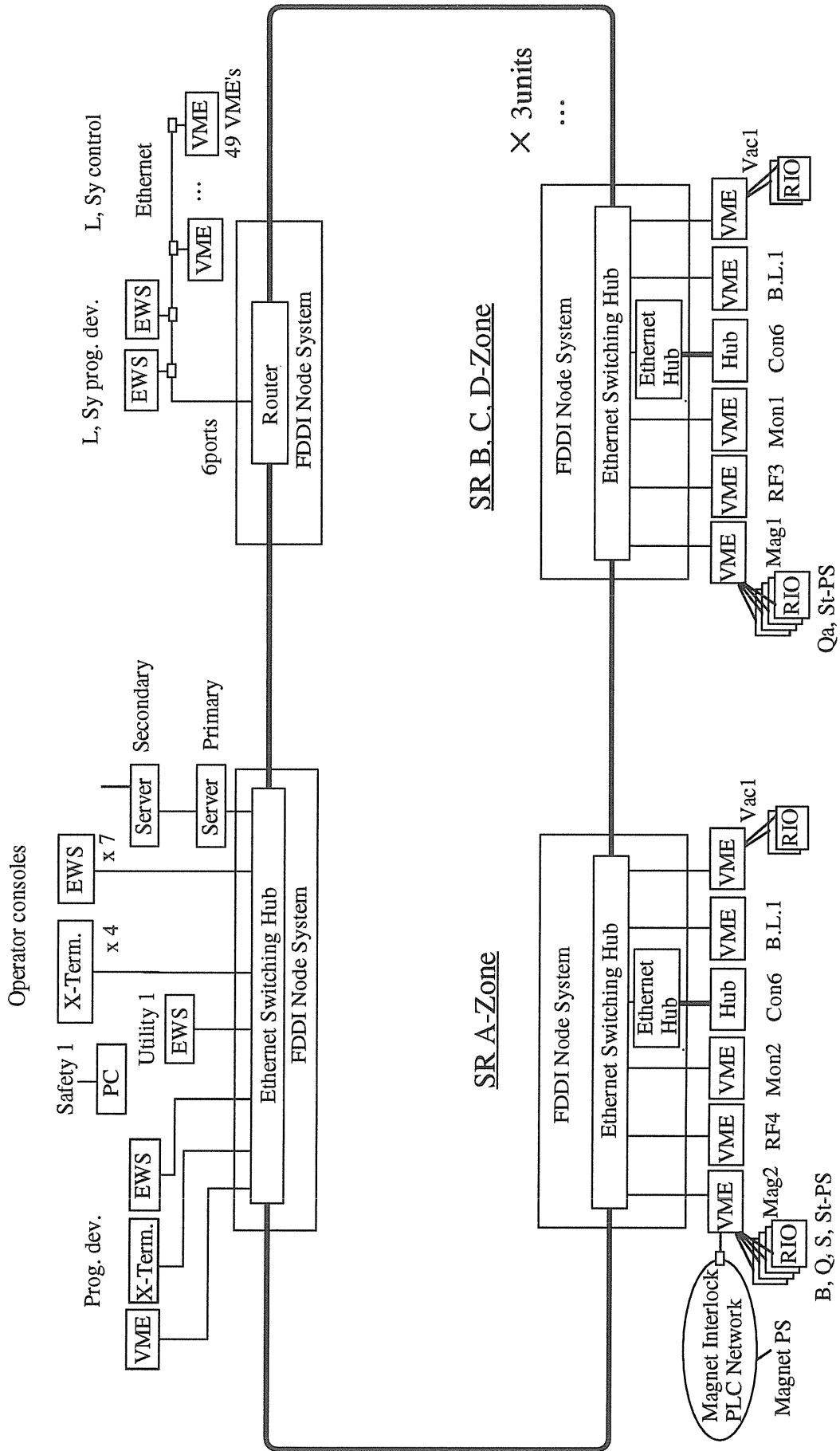


Figure 3. The architecture of the Spring-8 network and control system.

系設計方針に基づいて設計し、建設されている。**図3**にSPring-8のネットワークと制御系の構成を示す。詳しくは後述するとして、ここでは概略を手短に見てみることにする。制御系はオペレータ・コンソールである上位ワークステーション系、被制御機器をコントロールする下位VMEシステム系、上位と下位を結ぶ基幹FDDI光ファイバー・ネットワークからなっており、分散処理型の制御構成となっている²⁾。

蓄積リングの制御系はできるだけ「フラット」なトポロジーになるように考えられているが、これはネットワークの管理、障害検知と障害復帰を容易にするためである。VMEシステムでコントロールする被制御機器の対象は、電磁石電源系、高周波系、真空系、ビームモニター系と挿入光源系である。挿入光源は蓄積リング中に設置されるためにその制御系はマシン制御系に接続されている。ビームライン制御に関して言えば、構成は蓄積リングの制御系と同様のシステムを採用するが、フロントエンド部と光学系はビームライン制御系に属しており、マシン制御系と取り合うことになっている。

VMEシステムを制御するCPUボードとして、RISCプロセッサを実装したヒューレット・パッカーカード(HP)社のHP9000/743rtを使用する。CPUボードのOSとしてHPのHP-RTリアルタイムOSを用いる。比較的単純で定型の制御のためのフィールド・バスとしては、諸外国の加速器制御系では、MIL1553, G64, BITBUSなどが用いられているが、ここではGP-IBと三菱電機製リモートI/Oシステム(RIO)が複数の副制御系で用いられている³⁾。加速器運転用のオペレータ・コンソールとしてHPのUNIXワークステーション(HP9000/700シリーズ)を使用することになっている。OSはHP-UXを用いている。

制御用ソフトウェアは全てC言語で書かれており、イベントドリブンで動作するクライアント/

サーバ構成になっている。オペレータ・コンソール上で動作する制御ソフトウェアと、CPUボード上で動作する制御ソフトウェア間の通信は、ネットワークをまたぐ遠隔通信であるリモート・プロシージャ・コール(RPC)で行われる。データベース・サーバは2CPUのHPJ200を主、副サーバ合わせて2台使用する予定である。データベース・サーバ上では、加速器運転の記録、各種制御機器のパラメータ、ビーム光学パラメータなどのデータの管理のために、市販のリレーショナル・データベース管理系(RDBMS)である「Sybase」を用いる。

3. フロントエンド制御系

3.1 VMEシステム

いわゆる「標準品」を選択するのは「拡張性」、「保守の容易さ」のためである。この結果としてSPring-8ではVMEシステムがフロントエンドの制御系として選択された(**図4**)。蓄積リングの全長は約1500mあり、全周にわたって設置されている被制御機器をコントロールするためには、加速器の周長に沿ってネットワークを敷設して制御系を分散させることが必要であるが、VMEシステムはこのような目的には適している。

前述のように、VMEシステムをコントロールするCPUボードはHP9000/743rtを使用するが、これはRISCプロセッサPA-RISC7100LCを実装し、77.7MIPSの十分なパフォーマンスを持っている。CPUボードには16MBのメモリと20MBのPCMCIAフラッシュROMカードが実装されており、Ethernet I/F、ハードディスクなどのためのSCSI-II I/Fなどを持つ。フラッシュROMカードはオペレーティングシステムのブート・デバイスとして使用することもできるので、我々はフラッシュROMカードをブート・デバイスとして用いている。勿論、ハードディスクを接続してブート・デバイスとして用いること

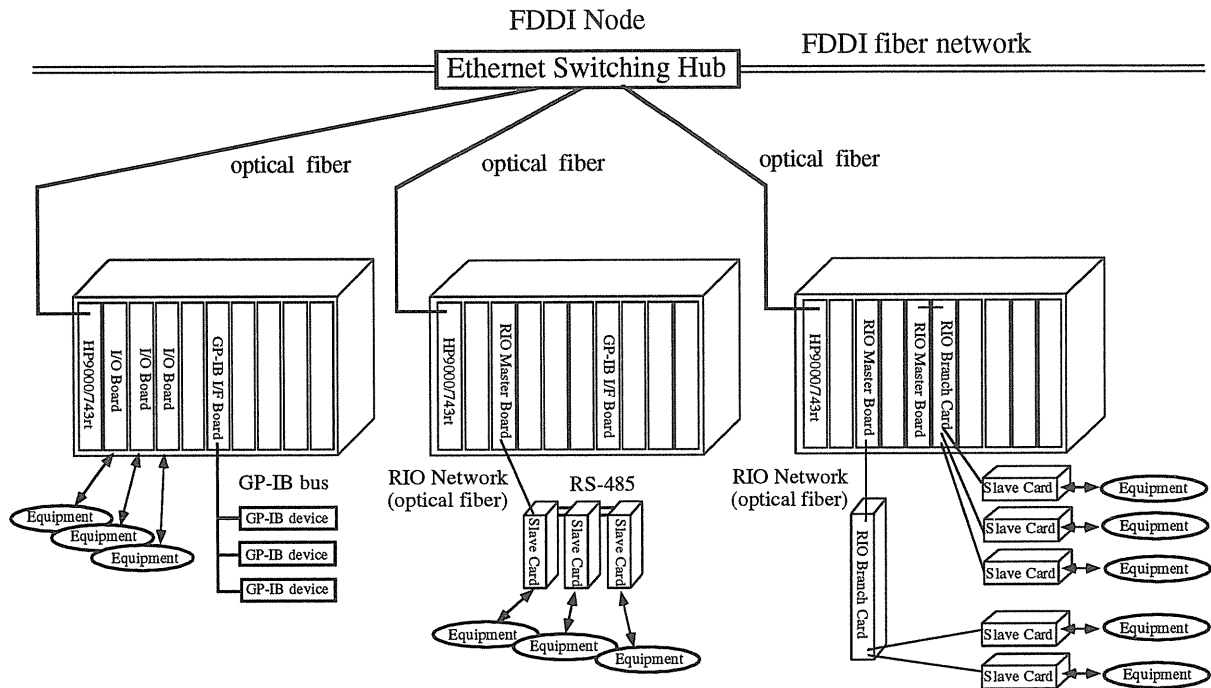


Figure 4. Examples of the VMEbus control system for the storage ring. Module configuration for RF system (left), RIO control system for beam position monitors (center), and RIO control system for magnet power supplies (right).

もできる。

VMEボードとしては、アクロマグ社製のDI (Digital Input), DO (Digital Output), AI (Analog Input) ボード, 日立造船製のTTL DI/DIO ボード, マイクロクラフト社製のPTG (Pulse Train Generator) ボードを用いている。これらのボードは高周波加速系の制御に使われている。電磁石電源系, ビームモニター系, 真空系の制御にはフィールド・バスであるRIOシステムが使われている。一般的にはビームモニター系には高速性を重視したシステムが適しているが, 蓄積リングはDC的であり現状では対応できると考えている。しかしながら将来SPring-8が高度化した場合, フィードバックなどを想定したデータ収集系の高速化はオプションとして考えておく必要がある。RIO以外のフィールド・バスとしてGP-IBも使用されている。蓄積リング制御系では全部で28台のVMEシステムを使用する予定である。

3.2 リアルタイムOS

CPUボードをドライブするOSとしてHP-RTを使用しているが, その選択に当たってはまず, 1)リアルタイムOSであること, 2)オープン・システムであること, 3)POSIX (IEEE提唱) 準拠であることが考慮された。これらは, コンセプトにあるように「市場で標準的に用いられているもの」で, 機器制御であるために「リアルタイム処理ができる」ものを選択するということからきている。

HP-RTはLynxOSをベースとしてPA-RISC用に移植したものであり, リアルタイムUNIXと呼ぶこともできる。この点HP-RTは, UNIXユーザが比較的容易に理解することができるので, 開発する上で有利である。プログラム開発では, LynxOSはセルフ開発とクロス開発の両方であるために, LynxOSの上でプログラムを作成しテストできるが, HP-RTはクロス開発であるので開発用のホストコンピュータが必要となる。ク

ロス開発では、ホストコンピュータ上でサポートされている数々の有用なツールが使用できるという利点がある。LynxOS はインテル系、SPARC系などの複数のCPUに移植されて動作している。我々の場合、HP-RTはPA-RISC上でしか動作しないために、HP-RTを動作させるCPUボードの選択の幅は狭くなるという欠点があるが、HP社のサポート体制を信頼して採用することとした。新しいOSを採用する場合、企業のサポート体制が必要であり、開発期間、手間などを考えると欠くことはできない。現在、各種のVMEモジュール用のデバイスドライバが開発中である⁴⁾。

3.3 リモートI/Oバス

すでに述べたように、我々のVMEシステムではフィールドバスとして、光ファイバーで接続されたRIOシステムが用いられている。RIOシステムは光ケーブル接続であるために、電氣的にisolationされており、ノイズにも強いという特質がある。もともとは、幾つかの電磁石電源が数百ボルト程グラウンドから浮いているために用いることが決定されたものであるが、今ではビームモニター系、真空系などにも広く使われている。RIOシステムの構成は、1)デュアルポート・メモリーをもったマスター・モジュール、2)色々なタイプのスレーブカード、3)8ポートのブランチカードからなる。マスター・モジュールとスレーブカードの接続は、光ファイバーのスター型結合、またはRS485のツイストペア線にてバス結合される(図4を参照のこと)。

RIOシステムの特徴をあげると、1)光ファイバー結合、2)シリアル・リンク通信、3)マスター・モジュールあたり最大62枚のスレーブカード接続が可能、4)1Mbpsの通信速度、5)通信距離1Km以下まで、6)HDLCプロトコルを使用、7)バス結合とスター結合が選択可能、などである。現在のところ、7種類のスレーブカード

が使用可能であり(type-A, B, C, D, E, FとG)、使用目的に合わせて選択することができる。例えば、type-Bのカードは32bit DI, 32bit DOからなっている。

4. ネットワーク

4.1 ノード構成

図3にSPring-8のマシン・ネットワーク・システムが示されている。100Mbpsの通信速度をもつ光ファイバーFDDIをバックボーン・ネットワークとして、ここに6つのFDDIノードが設置されている。FDDIの全長は約4500mになり、光ファイバーは図5にあるようなエアブロン(Air Blown Fiber System)チューブの中に敷設されている。ABFSは多重パイプになっている。SPring-8では6つのパイプのものを用いており、マシン系はこのうちの1つを使用している。

ノード・システムは入射器のライナックとシンクロトロン共用で1つ、中央制御室に1つ、残り4つを蓄積リング用として用いている。入射器系ではDECnet, UDPなどのプロトコルを使用するためにノード・システムはマルチ・プロトコル・ルータ(3Com NETBuilder-II)が使用される。ルータからはEthernetが機器コントロール用VMEまで伸びている。

蓄積リングの4つのノード・システムはEthernet/FDDI switching hub(3Com LANPlex2500)からなっている。このswitching hubは各Ether-

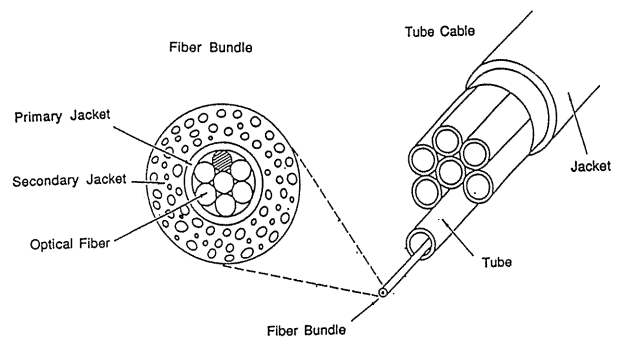


Figure 5. Tube configuration of the Air Blown Fiber System (ABFS) for FDDI fiber network.

net のポートに10 Mbps のフルのバンド幅を与えることができる。Switching hub は蓄積リングの保守通路に沿って設置され、switching hub の各 Ethernet ポートからは機器制御用 VME の CPU ボードまで1対1に光ケーブルが伸びている。光ケーブルは保守通路ケーブルラック上を通過して敷設される。機器制御用 VME には電磁石電源用、RF 用などがあるが挿入光源制御用の VME もこの中に含まれている。光ケーブルを使用するのは電氣的ノイズに強いからである。加速器制御機器を現場にて保守するために、switching hub から Ethernet の マルチポート・ハブ (3Com LinkBuilder-II) を経由して、周長に沿って設置されたメンテナンス用の情報コンセントまで Ethernet が敷設される。

中央制御室には switching hub が1つ設置され、switching hub の各 Ethernet ポートとオペレータコンソールの UNIX ワークステーションはメタル Ethernet ケーブルで1対1に接続される。中央制御室にはデータベース・サーバが2台設置される予定であるが、主データベース・サーバでは頻繁なオンライン・トランザクションが予想されるので FDDI ノードに直接に接続される。副データベース・サーバは主サーバのバックアップとして、また、マシン系以外のユーザへの加速器情報の公開用サーバとして用いられる予定なので、ネットワーク保安上 Fire wall としてのゲートウェイ機能を持たせることも考えている。

4.2 通信プロトコル

蓄積リングの通信プロトコルとして TCP/IP を用いる。また、ネットワーク管理のために Simple Network Management Protocol (SNMP) を用いる。TCP/IP を選択したのは信頼性を重視したためである。機器制御の観点から見ると、確実な制御を行うためには信頼性は重要なポイントである。パフォーマンスに関しては TCP/IP は最速ではないが、蓄積リングの制御にとっては十

分であり、ネットワークも比較的単純な構成となっている。ネットワークのプロトコル監視、障害検知などのために「HP OpenView」を用いてマシン系のネットワークを監視する予定である。

5. 運転用コンソールとインタフェース

5.1 オペレータ・コンソール

SPring-8 の線型加速器、シンクロトロン加速器と蓄積リングの3つの加速器は、蓄積リングに隣接する中央制御室から制御されることになっており、オペレータはワークステーション、X 端末などのコンソールを用いてこれらの加速器を制御する。

オペレータ・コンソールとして蓄積リング運転用に7台の UNIX ワークステーション HP9000/712/60などを用いることになっている。ワークステーションはそれぞれ128MB のメインメモリーと2GB のディスクを持っている。後述するように、オペレータ・コンソール上では X11/Motif ベースの GUI (Graphical User Interface) がインタフェースとして使用される。ちなみに、X11 と Motif ウィンドウ・マネージャが持つメモリーリークなどの問題があるために、連続運転を考えたときに一般的にはメインメモリーはできるだけ大きなほうがよい。入射器運転用には4台の X 端末を用いることになっている。これらのコンソールは FDDI-Ethernet switching hub で FDDI バックボーンに接続される。また、安全系とユーティリティ系のモニター用にワークステーションと PC が使用される予定である。

オペレータ・コンソールとモニター用 PC などは中央制御室に設置されるが、中央制御室ではワークステーション、PC などのうち本体部分が騒音発生源となりうる。そこで、本体部分を隣接する部屋に隔離設置することで、特に静粛性を重視した中央制御室を実現することになっている。また、設置される什器、遮光についても、人間工学的に最適化された静かで開放的な中央制御室にな

る予定である。

5.2 オペレータ・インタフェイス

オペレータ・コンソール上では X11/Motif ベースの GUI がインタフェイスとして使用されるので、オペレータはマウスなどで GUI の widget を直接操作することによって指令を送ることができる。我々は、GUI を作成するにあたって市販の GUI ビルダーの「X-Mate」(フジ・データ・システム製)を用いている。X-Mate は X11 プロトコルを用いており、look & feel は Motif に準拠している。編集用のエディターは操作が容易で、Motif のプログラミングの知識を必要とせず、画面上に見たままのイメージで (WYSIWYG で) ウィジェットが作成できる。ユーザのアプリケーションは X-Mate のコールバック・ルーティンに埋め込むようになっている。蓄積リング制御では、使用頻度の高い定型のグラフなどのテンプレートは制御グループから提供されるので、機器グループ、運転グループなどは look & feel の統一されたテンプレートを用いて、必要な GUI の作り込みを行うことができる。

前述したように、X11 と Motif ウィンドウマネージャが持つメモリーリークなどの問題があるために、連続運転を行う場合は注意が必要である。例えばウィンドウの拡大/縮小を行ったときに使用できないメモリー領域が発生する (メモリーリーク)。X-Mate は Xlib の上で走る自身のウィンドウマネージャを持っており、mwm のような Motif のウィンドウ・マネージャがなくても動作することができるので、ミッションクリティカルなところでも使用できる。もちろん我々の場合、どちらでも良いように十分なメモリーを用意している。

6. 機器制御

6.1 方法論

加速器を構成する機器を制御したい場合、どの

「機器 (オブジェクト)」に、「何 (設定値)」を、「どうしたいか (動作)」を指定しなければならない。一般的には、オブジェクトに制御指令を送る場合、該当する制御系を指し示す固有のアドレスとして、CPU 名と、VME バス上のアドレスなどが必要である。これらはハードウェアの構成に強く依存したものになるが、機器をどのように制御するかという加速器の運転手順に無関係に決まっていることが多い。

我々のシステムでは、機器を指定するのにこのようなアドレスを直接に用いて指定するのではなく、制御対象である「機器 (オブジェクト)」を意識した制御を行う。このため、アプリケーション・プログラムのレベルでは、できるだけハードウェアの構成に依存しないようなソフトウェアの構造とすることを目指す。ちなみにここでいう「機器」とは VME のモジュールではなく、マグネット電源、クライストロン電源などを指す。VME のモジュールは別に「デバイス」という呼び方を使う。

被制御「機器」を論理化して、ハードウェアに強く依存した物理層から分離し (「デバイスの抽象化」)、論理的な「機器」として制御の対象にする。これをソフトウェア上で実現するために、ソフトウェアを設計するにあたり「機器中心の制御」の考え方を強く反映させる。つまり、加速器運転のアプリケーション・ソフトウェアを、「オブジェクト (機器)」に対して「メッセージ (指令)」を伝達するというソフトウェア構造の枠組みの上に構築する。

6.2 Equipment Manager

機器を「抽象化」することによって、最上位の加速器運転のアプリケーション・ソフトウェアのハードウェア構成からの独立性が良くなる。これによって、ソフトウェアの拡張性、柔軟性が増し、ハードウェアを変更したときアプリケーション・ソフトウェアの変更は最小限ですむという利

点が生じる。そこで、HP-RT上で動いている加速器運転のアプリケーション・ソフトウェアに「デバイスの抽象化」という性格を明確に持たせ、それをより一般的なフレームワークとして構築すべく、「Equipment Manager」という概念を導入し、EMのサーバプロセスを誕生させることにした⁵⁾。

Equipment Managerは遠隔のCPUボード上で動作しており、上位からの抽象化された命令を受け取り、解釈してVME上の「物理デバイス」との対応づけを行い、デバイス・ドライバを呼び出して命令を末端の機器に伝える。EMは機器からの応答(バイナリ)を受取り、これを論理的な値へと抽象化して上位に返す。たとえば電磁石の電流値を設定する場合、「123A」のような電流値は、EMの中でVMEモジュールへの設定値(「バイナリ値」)に変換される。

EMは抽象化された機器名に対応する物理アドレスのテーブル(Device Configuration Table)を持ち、論理層と物理層の対応づけを行っている。上位の制御シーケンスはアドレスに関する知識をもつ必要はない。EMは命令を受け取り処理を行い結果を返す。一つの命令の処理が終了するまで次の命令は受け付けられない。またEM経由以外で機器にアクセスすることは禁止される。EMに対する命令は最大255文字の文字列で構成される。

EMを作成しようとするアプリケーション・プログラマーは、制御グループから提供されるEMのフレームワーク(コア・ソフトウェア)に自身のコードを埋め込み、希望するEMを作り込んで行く。

6.3 機器制御ソフトウェア

図6に蓄積リング制御用ソフトウェアの構成を示す。制御用ソフトウェアはイベントドリブンのクライアント/サーバ構造であり、全てC言語で書かれている。加速器運転のアプリケーションソ

フトウェアを容易に、かつ、モジュール性良く作成するために、ソフトウェアのインタフェース関数(Application Program Interface)はできるだけ数少なく、簡単な書式のもので作成する。また、APIは将来のミドルウェアの高度化に対応できるように「Plug & Play」ができるように配慮されている。

加速器運転員はヒューマン・マシン・インタフェースのGUIを用いて機器制御指令を発行する。この指令は人間が読めるような「メッセージ」でかかっている。発行されたメッセージはUNIX System Vの「メッセージ」でプロセス間通信によって「メッセージ・サーバ(MS)」に送られる。MSは命令中にあるオブジェクトの種類にしたがって、該当する「アクセス・サーバ(AS)」にメッセージを配信する。例えば、オブジェクトが電磁石電源なら、電磁石のAccess Serverに配信する。これは現実の郵便のシステムに似ている。ここまでが同一のワークステーション上で動作する。

ASは「指令(メッセージ)」の内容からこれを実行する「EM」を割り出し、該当する遠隔のCPU上で動作中の「EM」に対してRPC(リモートプロシージャコール)にて「指令」を伝達する。これは現実の電話のシステムに似ている。EMは受け取った指令を解釈し、デバイスドライバを呼び出して該当するVMEモジュールに対して指令を送る。EMは指令を完了すると結果をASあてに送り返す。もし、指令が機器の設定値の読み取りの場合であれば、同じ経路を逆にたどり、読み取り値をASあてに送り返し、MSを経由して、指令を発行したヒューマン・マシン・インタフェースのGUIまで送られる。

抽象化では一個の機器がハードウェアレベルでは複数に対応していてもよい。また、「設定値」と「動作」もハードウェア構成からなるべく独立するよう単なるビット列ではなく、物理単位を持った値として扱えるように抽象化を行う。この「機

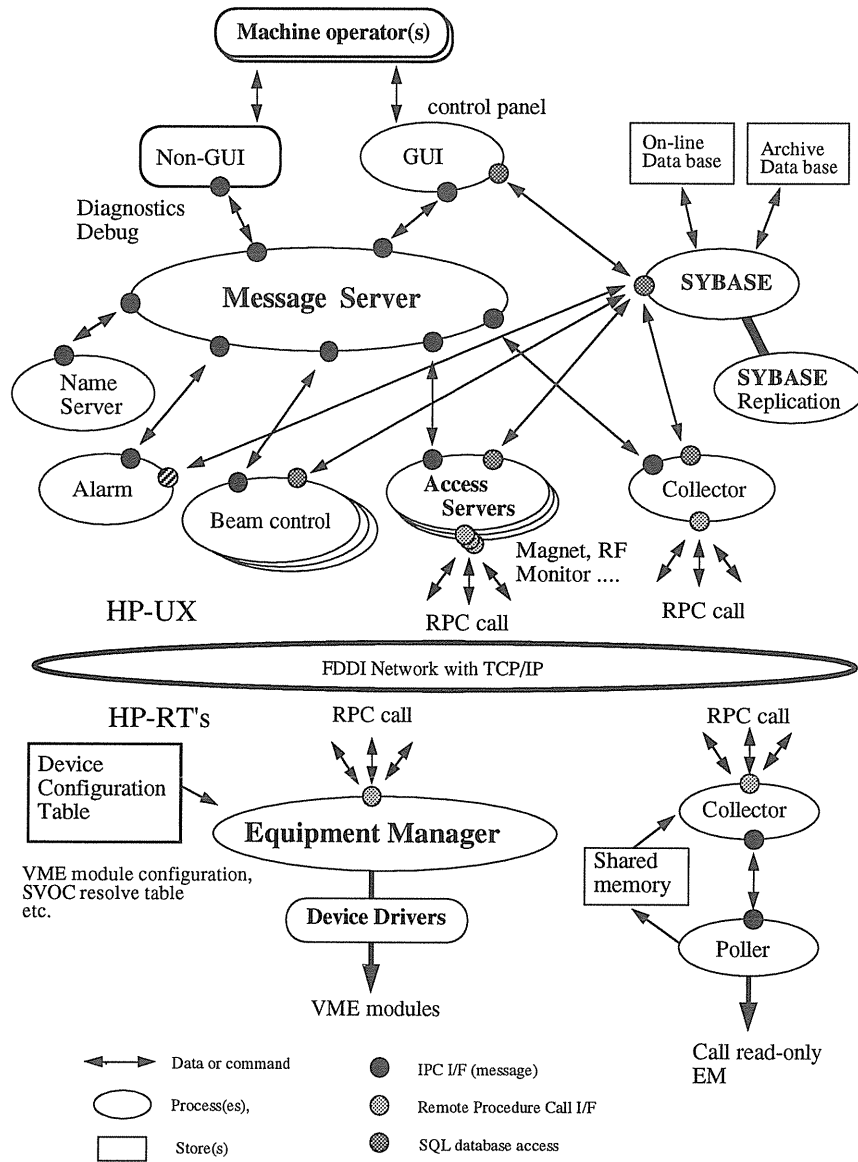


Figure 6. The client/server software structure of the storage-ring control system.

「器の抽象化」の利点は制御のシーケンスがハードウェア構成とある程度独立に記述できることである。たとえば、ハードウェア構成のある程度の変更を行った場合、制御シーケンスを変えずに、加速器運転のアプリケーションソフトウェアを最小限変更するだけでよいようなシステムを実現することができる。これはソフトウェアの再利用性、拡張性を考えたときに大きな利点となる。

7. 制御指令

7.1 メッセージ

加速器運転オペレータは機器に対する制御指令

(メッセージ) を指令書式にしたがって作成し、これを最大255文字の文字列データとしてMSのインタフェース関数を呼ぶことで送信する。実際にはGUIを使って操作するので、指令を送るたびにメッセージを毎回手で作成することはない。

メッセージの書式は英語の文章に似ており、S/V/O/Cの形式をとる。ここで、Sは命令の送信者で、メッセージを送信したプロセスのID、プログラム名、ホスト名などが書かれており、送信者のIDを表わし機器に対するアクセス権の審査にも使われる。Vは機器に対する指令動作で、Oは目的の機器名、Cは設定値や読み取り値など

の機器が持っている属性となっている。

Oは全系にわたってユニークであり、機器制御という立場から機器をクラス化して付けられている。このようなメッセージによる制御を行うことによって、機器の増加などに容易に対応できソフトウェアの拡張性がある。SVOCの書式はsyn-tacticではなく、できるだけsemanticになるように考えられている。

7.2 メッセージ・ミドルウェア

加速器運転オペレータから発行された制御メッセージを、何らかの方法でプロセス間通信(IPC)を行い、プロセス間で伝達しなければならない。これを行うのに、「メッセージ指向ミドルウェア」という概念をベースにして、以下に述べるようなメッセージ伝達機構をUNIX+C言語の範囲内で作成する。

オペレータのメッセージはSVOCの書式で表わされるが、このメッセージは同一ワークステーション内ではMSによってUNIX System Vのメッセージ機構を用いて配達される。MSに送られて来るメッセージはアクセス権、構文などをチェックされ、正しい文法のもの宛先に配達されて、トランザクションは記録される。MSを使用するプロセスはまず始めに、自身が使用するキューをMSから割り当ててもらわなければならない。MSは割り当てたキューをClient Registration Table (CRT) に記録するが、CRTは動作中のクライアントのモニターとしても使うことができる。MSは各ワークステーションで1つのみ動作し、今のところMS同志は直接には通信しないが、将来的にはデータベースを介して通信することは有り得る。

ネットワークを介して遠隔に通信するクライアント/サーバ間の通信手段として、Remote Procedure Call (RPC) を使用することはすでに述べた。RPCを採用することによって、ローレベルでのソケット通信の複雑な手順を知ることなしに、ク

ライアント/サーバの枠組みを比較的容易に構築することができる。RPCの実装としては、SUNのOpen Network Computing (ONC) RPCと、HP/ApolloのNCS/RPCをベースにしたOSFのDistributed Computing Environment (DCE) RPCがあるが、我々はONC/RPCを採用した。これはONC/RPCはNFSのベースになっており、NFSをサポートするワークステーションで幅広く用いられているからである。

8. データベース系

8.1 サーバ

データベース・サーバとして2台のワークステーションを用いる。副サーバにはPARISC7200を2CPUで実装したHPJ200を使用する。構成は、主メモリ256MB、DATテープシステム、CD-ROM ジュークボックス、RAID5の13GB ディスクアレイからなる。主サーバの機種は選定中であるが、少なくとも副サーバと同等かそれ以上のものを使用する予定である。主サーバ上には8.3に述べるような複数のデータベースが存在するので、頻繁なオンライン・トランザクションが予想されるためにFDDIポートに直接に接続される予定である。

8.2 データベース管理系

加速器のデータベースには、加速器運転の記録、各種制御機器のパラメータ、ビーム・オプションなどの種々のデータがある。これらのデータを効率良く管理し、加速器の運転、機器制御、ビーム解析を行うプロセスから統一的な方法でアクセスできるように、市販のリレーショナル・データベース管理系を用いてデータを管理する。我々は比較テストの結果、管理系に「Sybase SQL System 10」を採用した(実際にはより高速のバージョン11を用いる)。

Sybase SQL ServerにSQLを発行することで、オンラインまたはオフラインで動作している

プロセスから Sybase 上のデータにアクセスすることができる。SQL の発行は実際には RPC コールであり、ネットワーク接続された複数のワークステーション上の、複数のプロセスからデータをアクセスするような、「クライアント/サーバ」のソフトウェア構造に良くなじむ。Sybase を選択したのはそのオンライン・トランザクション性能が優れているからであり、また、データの「レプリケーション」機能も有用であるからである。

一般的に、大型加速器のような施設を動かすとき、マシン系外部からマシンの運転系が安易に操作できないように配慮する必要がある。しかしながら一方で、運転には関係していないがマシンのデータを必要とするユーザに対しては、マシンの情報を提供する必要がある。そこで、Sybase の「レプリケーション」機能を用いて主サーバのデータのコピーを副サーバに持たせ、これをマシン運転系以外のユーザへの公開用データとすることができる。これによって、主サーバの負荷は減り、副サーバに主サーバのバックアップとしての役割を持たせることもできる。

8.3 加速器データベース

蓄積リングのデータベースは3つのカテゴリーに分かれる。すなわち、「パラメータ・データベース」、「オンライン・データベース」、「アーカイブ・データベース」である。「パラメータ・データベース」は各種制御機器の物理的パラメータ、ビーム・オプティクスなどの恒常的なデータであり、テーブルは正規化されている。「オンライン・データベース」は加速器の運転状況、機器ステータスなどの現在のマシンの状態が保存されており、高速化を必要とするためにテーブルは正規化されていない。オンラインデータは機器グループごとにデータ収集の周期があり、データは秒単位から分単位で常にアップデートされている。「オンライン・データベース」には現在の機器異常を示す「アラーム」の状況も保存されている。

「オンライン・データベース」の保存時間はおおむね1時間くらいであり、この時間を経過すると「オンライン・データベース」は間引かれて、「アーカイブ・データベース」へと保存される。「アーカイブ・データベース」は加速器運転の記録（歴史的データ）として保存され、加速器の高度化などのスタディに用いられる。

8.4 データ収集

機器のデータはオペレータ用の GUI から、ワークショットの命令を発行して取得することもできるが、「オンライン・データベース」のような定型の収集のためにデータ収集専用の「Poller/Collector」系を用意する（図6参考のこと）。

Poller プロセスは CPU ボード上で動作しており、読み取り専用の EM に対してあらかじめ指示された命令群を発行し、管轄している機器からデータを取得し、データを共有メモリー上へ書き込む。共有メモリーはリングバッファになっており、メモリーの深さをある程度確保することで、分単位くらいの一定期間のデータを保存することができる。したがって必要な深さを確保しておけば、機器に障害が発生した場合に共有メモリーのダンプを行い、障害の解析を行うということに利用できる。Collector サーバプロセスは Poller と同じ CPU ボード上で動作しており、上位のワークステーションで動作する Collector クライアントプロセスからの定期的リクエストで、共有メモリーから最新のデータセットを読み込み、Collector クライアントへ渡す。Collector クライアントは、複数の Collector サーバから送られてきたデータをまとめて、タイムスタンプを押してオンライン・データベースへ書き込む。今のところ、オンライン・データベースのデータレートは凡そ 8 KByte/sec 程度を想定している。

データはこのような Poller/Collector 系の定時の収集/保存の他にも、オペレータのトリガーで任意のタイミングで保存できるようになっている

る。

8.5 データ操作

Sybase上に保存されているデータのアクセス(主に検索)は、Sybaseの「Client library」を用いて行う。Client libraryでは、サーバに対してRPCベースでSQL文を発行することによって、データ検索を行うことができる。実際には運転グループなどのマシン・アプリケーション作成者のために、Client libraryをラップした比較的容易なデータアクセス用の関数をいくつか用意する予定である。データベースの削除などは、特権を持った管理者のみができるように制限し、管理者がデータベースを保守するツールとしてSybaseの「Power Builder」を用いる予定である。

9. おわりに

SPring-8は現在のところ建設中であり、細部が決定されていないところもあって、やむなく今回の解説から省かせていただいたものもある。特に、放射光を利用する立場のユーザの方々には、ビームラインの制御系はどのようなものか大いに気になるであろう。また、線型加速器制御系、シンクロトロン制御系、蓄積リング制御系とをリンクさせ、一体化した中央制御システムを構

築するのは今後の作業として残っている。これらのものについては、いずれまた機会があれば触れてみたいと思っている。

蓄積リングの制御系を設計するに当たっては、放射光研究施設に限らず、先行する加速器研究所のいろいろな方々から御教示を頂いた。特に、A. Gotz/ESRF, C. Serre/CERN, W. McDowell/ANL, K. Cahill/FNAL, S. Howry/SLAC, H. Nishimura/LBLそしてJ. Urakawa/KEKの各氏をはじめ、上記の加速器研究所の多数のスタッフの方々から、有益なご助言とご支援を頂いたことをここに感謝します。

文献

- 1) M. Hara et al.: Proc. 4th European Particle Accelerator Conference, London, June (1994) p. 597.
- 2) T. Wada et al.: Proc. of the Int. Conf. on Accelerator and Large Experimental Physics Control Systems, Tsukuba, Japan, November (1991) p. 151. R. Tanaka et al.: submitted to Proceedings of ICALEPCS '95, Chicago, USA, (1995).
- 3) H. Takebe et al.: Proc. of the 4th European Particle Accelerator Conf., London, June (1994) p. 1827. H. Takebe et al.: submitted to Proceedings of ICALEPCS '95, Chicago, USA, (1995).
- 4) T. Masuda et al.: submitted to Proceedings of ICALEPCS'95, Chicago, USA, (1995).
- 5) A. Taketani et al.: submitted to Proceedings of ICALEPCS '95, Chicago, USA, (1995).