CONSTRUCTION OF NEW DATA ARCHIVE SYSTEM IN RIKEN RI BEAM FACTORY

Misaki Komiyama[1,A)], Akito Uchiyama [B)], Nobuhisa Fukunishi[A)]
[A)] RIKEN Nishina Center for Accelerator-Based Science
2-1 Hirosawa, Wako, Saitama, 351-0198
[B)] SHI Accelerator Service, Ltd.
1-17-6 Osaki, Shinagawa-ku, Tokyo, 141-0032

Abstract

RIKEN RI Beam Factory (RIBF) has two components, one is controlled by EPICS and the other is controlled by a non-EPICS system. At present, two kinds of data archive system have been in operation. One is an EPICS application and the other is taking data from non-EPICS systems. In order to unify the two applications into a single system, we have started to develop a new system since October, 2009. One of the requirements for this RIBF Control data Archive System (RIBFCAS) is that it routinely collects more than 3000 data from 21 EPICS Input/Output Controllers (IOCs) at every 1 to 60 seconds, depending on the type of equipment. An ability to unify the data from non-EPICS system is also required. To fulfill these requirements, a Java-based system is constructed, in which Java Channel Access Light Library (JCAL) developed by J-PARC control group is adopted in order to acquire large amounts of data as mentioned above. The main advantage of JCAL is that it is based on single threaded architecture for thread safety and user thread can be multi-threaded. The RIBFCAS hardware consists of an application server, a database server and a client-PC. The client application is executed on the Adobe AIR runtime. At the moment, we succeeded in getting about 3000 data from 21 EPICS IOCs at every 10 seconds, and validation tests are proceeding. Unification of the data archive system for non-EPICS systems is now in progress and it is scheduled to be completed in 2011.

RIBF制御系データアーカイブシステムの構築

1. はじめに

理研仁科加速器研究センターのRIビームファクトリー（RIBF）制御系は大きく二つのグループに分けることができる。図1にその概念図を示す。

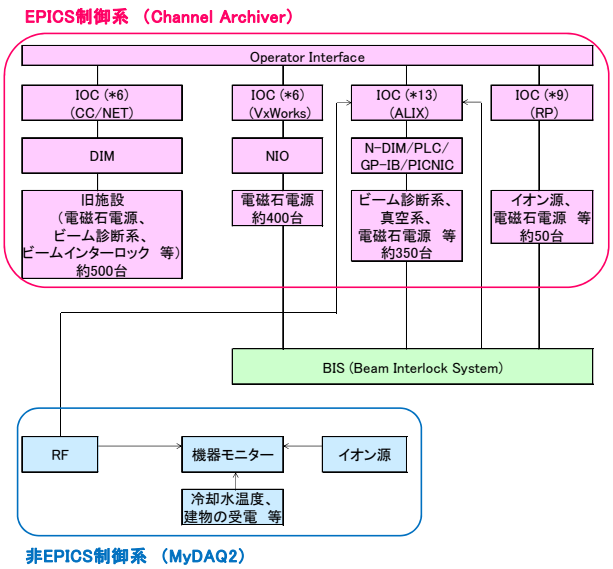

図1: RIBF制御系概念図

第一はExperimental Physics and Industrial Control System（EPICS）をベースに構築された部分で、加速器及びビーム輸送系の電磁石電源、ビーム診断系、真空系の制御などがこれに含まれる[1]。EPICS制御系ではInput/Output Controller (IOC)と呼ばれるオペレータインターフェースと各機器のコントローラを結ぶ計算機上でそれぞれの機器を制御するプログラムを実際に実行している。運用中のIOCとしては、Linux搭載の小型組込みボードであるPC Enginesの ALIX、東陽テクニカ製のCAMACのLinux搭載ネットワーククレートコントローラであるCC/NET、ソフトリアルタイムLinuxを搭載した横河電機製のPLC-CPUであるF3RP61-2L[2]及びVMEがある。また、それらが制御するコントローラとしては、理研で開発されたDevice Interface Module (DIM)及びNetwork-DIM (N-DIM) [3]、日立造船製のNetwork IO (NIO)、PLC、GP-IB及びトライステート製のPICNICがある。RIBF制御系を構成する機器の大部分がEPICS制御系に属している。

他方はEPICS以外のシステムで制御されているグループで、RF、冷却水、一部のイオン源の制御などが各々独立したシステムとして存在する。

運転時各種パラメータの記録に関しては上記二つの部分に対してそれぞれ異なるデータアーカイブシステムを運用中である。EPICS制御系に対するアー

[1] E-mail: misaki@riken.jp

カイブシステムとして、EPICSコラボレーションが提供する「Channel Archiver」を用いて加速器の各種運転パラメータを記録している[4]。「Channel Archiver」ではデータはデータベースに格納される代わりに独自のファイル形式で保存されるため、記録したデータの閲覧には同じくEPICSコラボレーションが提供する「Archive Viewer」を利用している[4]。「Archive Viewer」は保存された任意の期間の任意のデータを簡単にチャート上に復元することが可能であるが、一方でそれらのデータをテキストファイルやExcelのような汎用性のあるファイル形式に保存するには別のアプリケーションを利用する必要があり、必要に応じてプログラムを作成して運用している。データをファイルベースで保存する代わりにデータベースに格納するように仕様を変更した「RDB Channel Archiver」が2010年にリリースされているが、RIBF制御系ではそれに追随するバージョンアップは行っていない。

一方、EPICSで制御されない各機器に対しては「MyDAQ2」を用いている。MyDAQ2はSPring-8コントロールグループにより開発されたシステムであり、各機器からネットワーク経由で取得されたデータはMySQLに格納され、webアプリケーションから容易に検索して閲覧することが可能である[5]。非EPICSシステムに関しては開発当初より各々独自の方式でデータを管理していたが、それらを統合して管理するためにMyDAQ2を導入した。

MyDAQ2導入時に他のシステムとの統一を実現するためにChannel Archiverを導入しなかった理由は、Channel ArchiverはEPICSアプリケーションとして他研究機関で開発されたものであり、非EPICSシステムのデータを取り扱えるように仕様を改変する事は難しいからである。一方、MyDAQ2の導入により、EPICS制御系のデータアーカイブシステムもChannel ArchiverからMyDAQ2に移行して全体を統一することも検討した。その結果、技術的にはMyDAQ2でEPICSのデータを扱うことは可能であるが、MyDAQ2はSPring-8の加速器施設全体ではなくそれぞれのビームラインの実験データの取得及び解析用に開発されたアプリケーションであり、データベースのテーブル構成が決まっているため、RIBFのEPICS制御系に含まれる加速器全体の大量のパラメータを扱う大規模データベースへの対応はそのままでは難しいという結論にとなった。

平常のデータ解析においては、効率的ではないにせよカバーする範囲の異なる二種類のデータアーカイブシステムが併存していることに大きな問題はない。しかし、例えば実験中にビームが不安定になり直ちに原因を解決しなければならない場合には、それが問題になることがある。加速器から取り出しているビームの強度に予期せぬ変化が観測された場合、その原因を特定するためのひとつの手段として、ビーム強度と異変の原因となり得るパラメータの相関を調べることが行われている。複数のパラメータの相関を調べる中で、例えばビーム強度と電磁石の温度の相関に注目することがあるが、現状ではそれを調べるためにはまずそれぞれが属するシステム上でデータを検索してチャートに表示し、フォーマットの異なる二つのチャートを見比べる必要があり、効率が良いとはいえない。もし統一されたアーカイブシステム上でデータを取得できればこの種の解析の効率は上がり、加速器の安定運転に少なからず貢献すると考えられる。更に統一されたシステムが、記録した過去のデータを呼び出して表示するだけではなく、リアルタイムにパラメータを表示する機能も併せ持たせれば、ビームの異変に気付いた時に直ちに原因となっている可能性のあるパラメータの相関を調べることができる。これは、様々な理由で現在EPICSで制御されていない機器をEPICS制御システムに組み込むことで解決できることであるが、運用中の施設においてその更新をすぐに行うことは難しい。そこで、RIBFの安定運転に必要不可欠なパラメータを必要に応じたサンプリングレートで全数取得し、データ表示は過去のデータの呼び出しだけではなくリアルタイムでも表示できるような機能を併せ持つデータアーカイブシステムを独自で開発することとした。

2. 新データアーカイブシステムの開発

2.1 システムの概要

2009年10月より、EPICSシステムと非EPICSシステムのデータを一元的に扱うRIBFデータアーカイブシステム（RIBFCAS）の開発をスタートした。システム概要は以下の通りである。

(1) データベースはオープンソースのものを利用する。
(2) EPICS制御系に関しては、IOCからEPICSの通信プロトコルであるChannel Access (CA)で直接データを取得するプログラムをJavaベースで新規作成する。
(3) 非EPICS制御系の部分に関しては複数存在するシステムのそれぞれに対してインターフェースを開発するのではなく、従来通りMyDAQ2を運用することとし、MyDAQ2との連携を通じてデータを取得する。
(4) クライアントプログラムは過去のデータを表示する機能とともに、リアルタイムでデータを表示する機能も持つこととする。
(5) RIBFCASの運用方針として、過去5年間程度のデータを保持することとする。直近の過去1～2年分のデータに関してはクライアントPCから随時検索可能とし、それ以前のデータはバックアップから取得する構造とする。

(1)に関しては、RIBFCASのデータベースは大規模になるため、それに適したデータベースを選定することが重要である。オープンソースという要求を満たし、更にデータの検索スピードを上げるための仕組みであるパーティショニングの機能を持つPostgreSQLを採用することとした。

(2)に関し、扱うデータは電磁石電源やビームモニター機器や真空機器などを制御している22台の

IOCから得られる約3000点のパラメータであり、それらを機器の種類に応じて1秒から60秒間隔で取得する。RIBFCASでは大量のデータを取り扱う為、データ収集速度が速く、確実に収集できるCAのライブラリを選択することが極めて重要である。これらCAライブラリの選択に関しては後述する。

(3)のMyDAQ2との連携の実現に関しては、RIBFCASの中にMyDAQ2用のテーブルを準備してそこに必要なデータを書き込むのではなく、MyDAQ2で提供されているXMLでデータを返すCGIを参考に、MyDAQ2からデータを取得し、RIBFCASで扱えるXMLの形式でデータを送信するCGIを作成する、という方針で実現する。

(4)に関しては、クライアントアプリケーションはAdobe AIRで作成することとした。その利点として、作成されたクライアントアプリケーションはAdobe AIRランタイムが動作可能な環境であればプラットフォームの種類を選ばず、複数のプラットフォーム上で動作する。

2.2 CA ライブラリの選定

CAライブラリの候補として、Java Channel Access (JCA)、Java Channel Access Light Library (JCAL)及びJCAのAPI経由でJCALを使用するために作成されたアダプタであるJCA-JCALの三つに関してデータ収集速度の試験を行った。JCAはEPICSコラボレーションにより提供されているライブラリであり、JCALはJ-PARCコントロールグループにより開発されたライブラリである[6]。試験はJavaで実装されたデータ収集プログラムの一プロセスが対応するIOCの全パラメータに対してサンプリングレート1秒でアクセスしてデータ取得を試み、全データの取得に連続して5回成功もしくは5回失敗するまでに要する時間を測定するというものである。各IOCは各々表1に示される約200点の制御パラメータを持っている。IOCの種類に関する速度依存性を確認する為に3種類のIOCを対象に試験を行った。JCAはスレッドセーフな設計ではないために、プログラムはシングルスレッドで実装した。一方、JCALはJCAの問題点を克服するべく開発されたものであり、スレッドセーフな実装になっているため、プログラムはマルチスレッドで実装した。結果を表1に示す。

表1: EPICSのJavaライブラリの性能評価

IOC	パラメータ数	JCA 実行時間（ミリ秒）	JCA 平均時間（ミリ秒）	JCA	JCA-JCAL 実行時間（ミリ秒）	JCA-JCAL 平均時間（ミリ秒）	JCA-JCAL	JCAL 実行時間（ミリ秒）	JCAL 平均時間（ミリ秒）	JCAL
CC/NET	187	49830	43	失敗	5102	2	成功	5035	2	成功
ALIX	198	46986	42	成功	5104	2	成功	5042	2	成功
VME	192	45540	42	成功	5101	1	成功	5037	2	成功

試験の結果、

- JCAL及びJCA-JCALは独自のコアプログラムを用いているためにJCAと比較して実行時間が飛躍的に短縮されており、データのサンプリングレートとして設定した1秒以内に190程度のデータをすべて処理し、1データの取得に要する時間も2ミリ秒程度になった。両者に有意な差は認められなかった。
- JCAの実行速度が遅い理由は以下のように考えている。JCAライブラリのAPIからIOCへの接続を呼び出しても実際にプログラムが接続の状態になるにはある程度の時間を要するという現象が見られた。チャンネル接続のAPIを呼び出した直後に任意のチャンネルにデータ取得を要求すると未接続のステータスが返され、データの取得に失敗するのである。これを回避するためには、接続後40ミリ秒、IOCがCC/NET(CAMAC)の場合には100ミリ秒を要するためにデータ取得時間が長くなった。

が明らかになった。上記に基づき、RIBFCASではJCALをCAライブラリとして採用することとした。

2.3 RIBFCAS の構成

これら検討結果をベースに構築されたRIBFCASの構成を図2に示す。

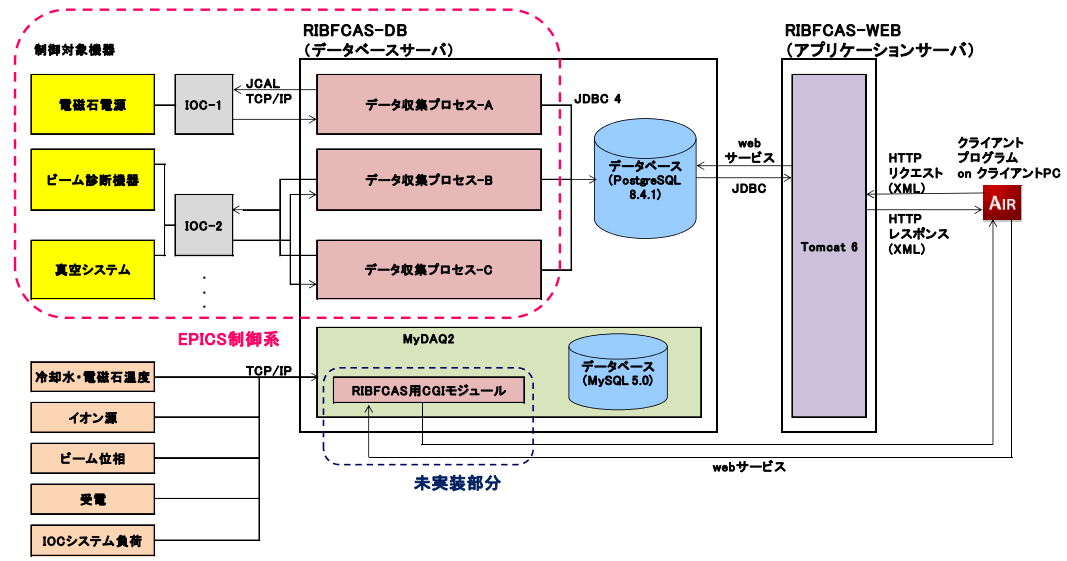

図2: RIBFCAS構成図

RIBFCASのハードウェア構成及びそれぞれの役割は以下の通りとした。

- データベースサーバ：EPICS IOCからデータを取得するJCALベースのプログラムを実行する。併せてデータベースを管理する。データベース管理システム（DBMS）はPostgreSQL 8.4.1を採用する。現在はMyDAQ2もこのサーバ計算機上で運用しており、MyDAQ2のDBMSであるMySQLの管理も行っている。
- アプリケーションサーバ： Javaが走るwebアプリケーションサーバであるTomcatを実装した。クライアントPCからのリクエストでデータベースにアクセスし、取得したデータをXMLに処理してクライアントに返す。
- クライアントPC：Adobe AIRで実装したクライアントアプリケーションを実行する。

採用したハードウェアの仕様を表2に示す。なお、RIBFCAS-DBサーバの能力はRIBFCASの長期安定運転には不十分なので、アップグレードを検討中である。

表2: RIBFCASのサーバ計算機

	RIBFCAS-WEBサーバ	RIBFCAS-WEBサーバ
CPU	Intel Xeon (Quad Core) 2.33 GHz	Intel Core2Duo 2.20 GHz
Memory	16 GB	1 GB
HDD	120 GB (RAID10, SAS)	80 GB
OS	CentOS 5.2 (32 bit)	CentOS 5.3 (64 bit)

データベースサーバ上で運用されるデータ収集プログラムは以下の特長を持つものとした。

- データ取得プロセスは複数プロセスでマルチスレッド。将来のデータ取得対象機器の増加に対してプロセスの分割やスレッドの追加による柔軟な対応を実現する。
- 多数の機器をXMLで管理することにより、機器の増減をXMLのノードの追加及び削除を行うことだけで対応することが可能である。
- 整理された多層構造を実現し、各システムコンポーネントを独立させる（データベース、データ収集プロセス、サーバ内にあるデータベースからのデータ取得ロジック、サーバ内のデータ配信（Webサービス）、クライアント）。
- プロセスのステータスを監視し、異常が発生した場合は自動的にプロセスの再起動を行う。

RIBFCASは2010年度末に基幹部分が完成し、2011年度より試験運転を開始した。

3. 運用状況

RIBFCASのEPICS制御系に関する部分の試験運用は、RIBFの運転スケジュールに合わせ、2011年4月26日夕刻より開始した。EPICS部分は、22台のIOCに分散されている制御対象機器から約3000点のパラメータを9プロセスに分割して10秒間隔で取得しデータベースに格納する。対象パラメータとしては電磁石電源のDAC値、ADC値、インターロック信号を含むステータス、加速器及びビームダクトに取り付けられたバッフルスリットからのビーム電流値、真空度などが大部分を占めている。また、データ取得間隔10秒はシステムの性能を確認するために設定した値であり、加速器の周期の遅い安定性の解析の為には十分な早さであるが、実際の運用では機器の種類に応じて最適な値を設定する必要がある。尚、この段階ではまだMyDAQ2との連携部分は開発中のため実装されていない。

試験運用において、運転パラメータの継続的取得には基本的に成功した。データベースの容量は日々増大し、7月13日現在で約76GBのデータが蓄積された。この間、システムの継続運用の過程において、ひとつ問題点があった。それは、各プロセスに含まれるIOCの種類により、プロセスによってメモリの使用量が32bit OSのメモリ空間の使用制限の上限に達して収集プロセスが停止する、という現象が見られることである。経験的にCC/NETを対象としたプロセスはメモリ使用量が上限に達することはないが、それ以外のIOCを対象としたプロセスは、含まれるデータ数により使用メモリの増加の割合は異なるものの、プロセス停止が発生することが判明した。原因として、RIBFCASのデータ取得プログラムが微小なメモリリークを起こしている可能性があり、現在プログラムの見直しを行っている。しかし、調査には時間が必要である為、問題が解決するまでは各プロセスのメモリ使用量を監視し、メモリ使用量が上限に達する前にプロセスを再起動する、という方法で運用を行っている。

クライアントアプリケーションは任意のデータのリアルタイム表示と履歴の表示の両方に対応しており、チャート上に示されたデータはテキストファイルもしくは画像ファイルに記録することが可能である。また、チャートの縦軸のスケールはリニアスケールとログスケールの選択が可能である。現段階で任意のひとつのパラメータの24時間分のデータを検索するのに要する時間は約10秒であり、一時間分のデータの検索では約0.3秒であった。また、一度検索した24時間分のデータを再度検索するのに要した時間は約0.05秒であった。

4. 今後の予定

データ取得プログラムのメモリ使用量の問題は解決が必要なもののデータ取得は滞りなく実行されていることから、RIBFCASのEPICS制御系のデータを取り扱う部分はほぼ完成したと考えている。次なるステップとしてMyDAQ2との連携の実現、取得した莫大なデータの管理方法の検討を行っている。データ管理に関しては、データベースはテスト運用を開始して約2ヵ月半で既に76GBの容量を必要とした為、RIBFCASのデータ量として一年間に500GBを想定し、それに見合ったハードを検討中である。システムの完成は今年度中を目指している。

謝辞

RIBFCASの構築の実装は株式会社スウェイドックの江藤太一郎氏と大高康史氏が担当されました。

システムの設計においても数々の助言をいただき、
ここに感謝申し上げます。

参考文献

[1] M.Komiyama, et al., Proceedings of ICALEPCS2009, Kobe, Japan, 2009-10, p.275
[2] http://www.yokogawa.co.jp/rtos/Products/rtos-prdcpu9-ja.htm
[3] M.Fujimaki, et al., RIKEN Accel.Prog. Rep., 37(2004), p.279
[4] http://www.aps.anl.gov/epics/extensions/index.php
[5] T.Hirono et al., Proceedings of PCaPAC08, Ljubljana, Slovenia, 2008-10, p.55.
[6] H.Sako, et al., Proceedings of ICALEPCS2009, Kobe, Japan, 2009-10, p.842