

J-PARC Main Ring Archive Web Viewer の現状と今後の展望

CURRENT STATUS AND FUTURE PROSPECT OF J-PARC MAIN RING ARCHIVE WEB VIEWER

土井幸之介^{#,A)}, 高橋大輔^{A)}, 山田秀衛^{B)}

Kounosuke Doi^{#,A)}, Daisuke Takahashi^{A)}, Shuei Yamada^{B)}

^{A)} Kanto Information Service

^{B)} High Energy Accelerator Research Organization (KEK) /J-PARC Center

Abstract

Since J-PARC Main Ring (MR) started beam operation in 2008, we have archived accelerator operation data using EPICS Channel Archiver. In the control group, a web interface is developed and used as one of means for utilizing the data collected in this system. In recent years we evaluated next generation archive systems replacing the Channel Archiver, and have been carrying out renovation of the Web interface accordingly. In this presentation, we will report, the current situation and future prospects concerning the migration of the archive system.

1. はじめに

J-PARC Main Ring(MR)の制御システムは、加速器を構成する機器の状態情報など 10 万点を超えるデータを取り扱っている。

2008 年にビーム運転を開始して以来、これら機器に関する各種データを蓄積、管理するシステムとして Channel Archiver、2015 年から CSS Archiver、2017 年から Archiver Appliance に移行してきた。そして、制御グループではこれらのシステムにて収集されたデータを利用するための手段の一つとして Web インターフェースを開発し運用してきた。

本発表では、その開発された Web インターフェースの運用状況、改修点、展望について報告する。

2. データアーカイブシステムの移行

2.1 Channel Archiver

Channel Archiver は EPICS 標準のデータアーカイブツールである。

データ収集サーバ上で Archive Engine と呼ばれるプロセスを稼働させることで、CA プロトコルを介して EPICS レコードの値を収集することができる。収集したデータはタイムスタンプ情報と共に独自のバイナリ形式でストレージへと保存される。

J-PARC MR における運用では、日々増加するデータの管理と頻発する index ファイルの破損が問題とされた。

特に index ファイルはデータを読み出すのに必要不可欠なものであり、このファイルを修復するまでの間は破損したデータの期間はデータアーカイブと閲覧ができない状態に陥ることが大きな問題とされた。これらの問題に対処するため Channel Archiver の統合管理ツールである Archive Daemon を導入した。Archive Daemon はチャンネルの接続状況の確認や Archive Engine の操作、自動再起動の設定ができる。そして、一定期間でデータの保存ディレクトリを切り替えることができるようになるため問

題の修復にかかる時間を短縮することができるようになった。しかし、アーカイブ登録を廃止された EPICS レコードの削除が困難なこと、index ファイルの修復に時間を要すること、更に Channel Archiver の開発が既に終了していることもあり、後継のシステムである CSS Archiver の運用を開始した。移行しなかった理由は後に記述する。

2.2 CSS Archiver

CSS Archiver は基本的に Channel Archiver と同様の機能を備えたデータアーカイブツールである。Channel Archiver との大きな違いは、収集したデータの保存に RDB を利用していることにある。これまではデータを引き出す際に独自のバイナリファイルから専用のコマンド、ツールを使用する必要があったが、CSS Archiver では SQL を使用することが可能となり、従来のアーカイブシステムよりもアプリケーション作成に対する汎用性が大きく向上した。しかし、Channel Archiver によってアーカイブされたレコードデータは独自のバイナリ形式で保存されているため CSS Archiver のデータ移行をするためには人手と時間が多く必要だった。そのため、移行自体は可能であったが CSS Archiver が導入されても Channel Archiver の運用は継続されることになった。

実際の運用上の問題は、ストレージ使用量の増加、データ取り出し速度の低下が挙げられる。

CSS Archiver によって RDB に保存されるデータは従来の Channel Archiver が保存するバイナリ形式よりもファイルサイズが大きく、同じ点数のレコードをアーカイブすると数倍のストレージ容量が必要となった。特に点数の多い配列レコードの保存効率に難があり、一部レコードは Channel Archiver から移行することが難しい状態にあった。また RDB の容量が増加すると共に検索範囲が拡大するため、データ取り出し速度の著しい低下が見られた。これらの問題は RDB をメンテナンスすることである程度対処が可能だが、長期間のデータを続けて閲覧したいといった要望が多かったため、データ取り出し効率の向上を目論み Archiver Appliance を導入した。

[#] z-doi@post.j-parc.jp

2.3 Archiver Appliance

Archiver Appliance は数百万のレコードをアーカイブすることを目的とした EPICS 制御システム用アーカイバであり、データ取得パフォーマンスに重点を置いている[1]。また、Channel Archiver における index ファイルや、CSS Archiver のようなエンジン/グループの概念も存在しない。レコードのアーカイブはパラメータごとに多段ストレージに分けて行っている[2]。Archiver Appliance は JSON / HTTP などの手段を用いて通信を行っている。また、CSV や SVG のデータのやり取りも可能である。

3. Web インターフェース

3.1 作成動機

制御グループでは、CSS Archiver、Archiver Appliance でアーカイブされたデータを Web ブラウザ上で OS や端末に依存せずに閲覧できるようにするために、それぞれ専用の Viewer を作成した。これら Viewer はレコード名と時間帯を指定することでアーカイブデータを読み出してグラフにプロットすることができる Web インターフェースである。

3.2 動作・開発環境

CSS Archive Viewer 及び Archive Appliance Viewer の開発環境、動作環境を Table 1、Table 2 にそれぞれ示す。

Table 1: CSS Archive Viewer

Development language	PHP, JavaScript
Operating environment	Apache2

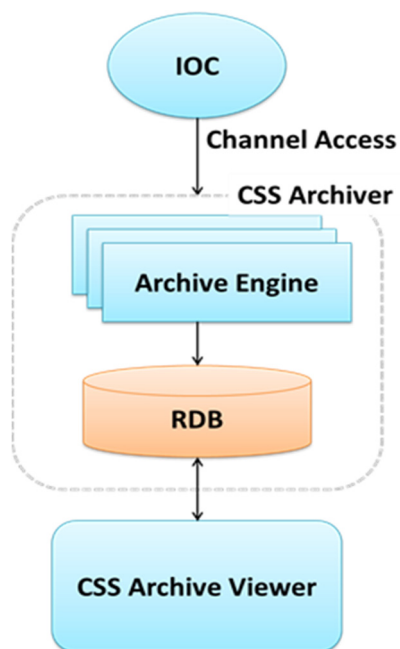


Table 2: Archive Appliance Viewer

Development language	PHP, JavaScript, Perl
Operating environment	Apache2

プロットにはグラフ描画ツール Gnuplot を使用している。CSS Archive Viewer は SQL にて取得したレコードデータを、Archive Appliance Viewer は Perl スクリプト (ArchiveDataClient-ap.pl)にて取得したレコードデータをもとに作成された plt ファイルを用いて Gnuplot は png ファイルを作成している。

データの検索・取得方法は Figure 1 に示すとおり CSS Archive Viewer は直接 RDB に接続してデータを取得して表示するのに対して、Archive Appliance Viewer は Archiver Appliance に保存されたデータをサーバから取得するために Perl スクリプト(ArchiveDataClient-ap.pl)を使用している。

ArchiveDataClient-ap.pl は、HTTP 通信により Data Retrieval にレコードのデータ取得要求をしている。

PHP だけではなく Perl も使用している理由は以下のようになっている。

- 単体のツールとしても ArchiveDataClient-ap.pl を使用できる
- 取得したデータを Gnuplot 向けに整形ができる

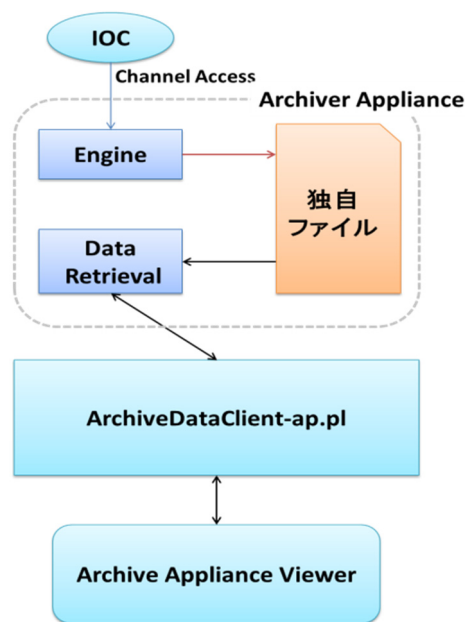


Figure 1: Backend.

3.3 機能

3.3.1 CSS Archive Viewer

CSS Archive Viewer は基本的な機能として以下を有している。

- 指定した期間のデータ点数が 2000 点を超えていた場合データ点数を 2000 点に間引きして取得する
- ワイルドカード、Engine ごとにレコード名を絞り込むことができる
- 指定したレコードのアーカイブ開始日時・最終更新日時の閲覧

Figure 2 に表示画面を示す。

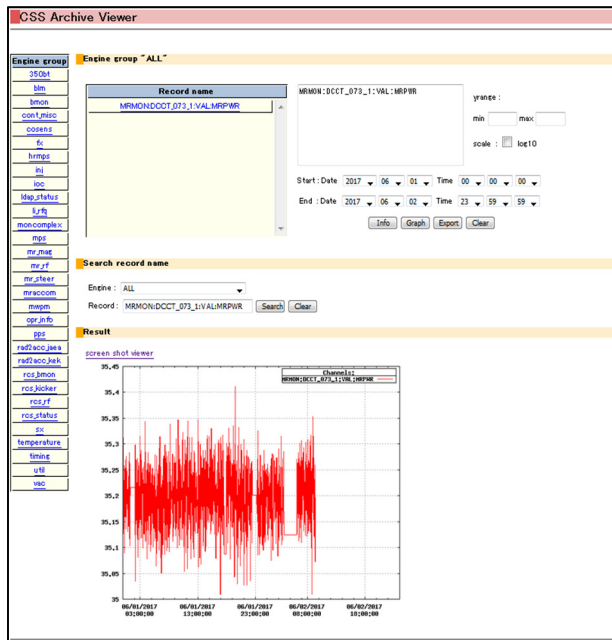


Figure 2: CSS Archive Viewer.

3.3.2 Archive Appliance Viewer

Archive Appliance Viewer の基本的な機能は CSS Archive Viewer と変わらないがプロットする際の条件が増えたことにより柔軟な検索ができるようになった。また、CSS Archive Viewer にはレコードの絞り込みの条件に Engine を指定できたが Archive Appliance Viewer には CSS Archiver のような Engine の概念が存在しないためそれらに関する検索機能は存在しない。

プロットの条件に増えた項目を以下に示す。

- 取得するデータ数
- 取得するデータの間引き方

Figure 3 に表示画面を示す。

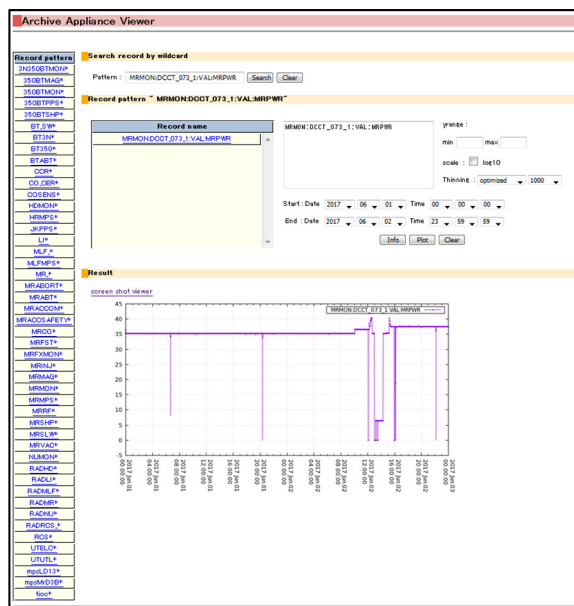


Figure 3: Archive Appliance Viewer.

4. まとめと今後の展望

4.1 現在の問題点

現在、Archiver Appliance に 2017 年 1 月以前のデータを Channel Archiver から順次変換しているが、まだ全ては変換できていない。そのため、Archive Appliance Viewer だけでは 2017 年 1 月以前のレコード全ての閲覧ができない。CSS Archiver への新規登録を停止した現在でも 2017 年 1 月以前のデータを閲覧するために CSS Archive Viewer は運用されている。

Archiver Appliance、Archiver Appliance Viewer 自体は運転開始後、特に問題は発生していない。

4.2 改善された点

過去には、CSS Archiver は 2.2 で述べたように、データ量が多くなるにつれてデータの読み出しに時間がかかるため CSS Archive Viewer のプロットにも時間がかかってしまう問題があった。しかし、Archiver Appliance は RDB をデータの保存領域に使用していないためその点は心配する必要がなく、Archive Appliance Viewer を使用すれば短時間でデータの呼び出し、プロットが可能になった。

4.3 今後の展望

Archive Appliance Viewer 以外のその他インターフェースとの連携を考えた機能の追加を行う。

参考文献

- [1] The EPICS Archiver Appliance;
https://slacmshankar.github.io/epicsarchiver_docs/details.html
- [2] THE EPICS ARCHIVER APPLIANCE;
<https://accelconf.web.cern.ch/AccelConf/ICALEPCS2015/papers/wepgf030.pdf>