

KEK 電子・陽電子入射器における Archiver Appliance の運用状況

OPERATION STATUS OF ARCHIVER APPLIANCE IN KEK ELECTRON/POSITRON INJECTOR LINAC

佐武いつか^{#, A)}, 佐藤政則^{A, B)}, 佐々木信哉^{A)}, 廣瀬雅哉^{C)}, 工藤拓弥^{D)}, 草野史郎^{D)}, 王迪^{B)}
Itsuka Satake^{#, A)}, Masanori Satoh^{A, B)}, Shinya Sasaki^{A)}, Masaya Hirose^{C)}, Takuya Kudou^{D)}, Shiro Kusano^{D)}, Di Wang^{B)}

^{A)} High Energy Accelerator Organization (KEK), Accelerator Laboratory

^{B)} The Graduate University for Advanced Studies (SOKENDAI), Department of Accelerator Science

^{C)} Kanto Information Service Co., Ltd (KIS)

^{D)} Mitsubishi Electric System & Service Co., Ltd (MSC)

Abstract

In the KEK electron/positron injector linac, electron and positron beams are delivered to five different rings of SuperKEKB electron/positron/damping rings, PF, PF-AR. The control system of the injector linac is constructed using EPICS framework which is a distributed control system. Currently, we record approximately 100,000 data in the archive system. The number of PVs to be archived and the data size are increasing year by year. As the data collection software, CSS archiver has been in operation since 2011, and Archiver Appliance has been in operation since November 2019. Both of them are data logging tools based on EPICS control system framework. Archiver Appliance has advantages such as faster data reading and less disk consumption in comparison with CSS archiver. However, there are some problems such as an increase of Broadcast packet in the KEK injector control network and abnormal operation of CA search. For the research, we use Kibana, a data visualization tool that has already been used in SuperKEKB control system. In this paper we report minutely on the current operation status of Archiver Appliance in KEK injector linac.

1. はじめに

KEK 電子・陽電子入射器 (KEK 入射器) では、SuperKEKB、陽電子ダンピングリング、PF、PF-AR、の計 5 種類のリング加速器に対して、エネルギーや特性の大きく異なるビームを高速で切り替えて制御する同時入射が成功した (Fig. 1)。KEK 入射器では、SuperKEKB の目標性能を満足すべく、機器やソフトウェアのアップグレー

ドをおこなってきた [1, 2]。KEK 加速器の運転制御には、Experimental Physics and Industrial Control System (EPICS) [3] と呼ばれる制御システム構築のためのソフトウェアフレームワークを使用している。KEK 入射器では、現在約 11 万点のデータをアーカイブシステムで記録しており、アーカイブ対象の数とデータサイズは、今なお増加している。データ収集ソフトウェアとしては、従来から CSS archiver を運用しており、2019 年 11 月から Archiver Appliance (AA) [4] の運用を開始している。AA は、データ読み出しの高速化やディスク消費量の軽減などのメリットがある一方、導入当初から入射器制御ネットワークにおける Broadcast の増加や CA サーチの異常な動作などの問題がおきていた。SuperKEKB で既に運用されている Elastic Stack を用いた Broadcast パケットの監視システムを、入射器制御ネットワークにも導入し、調査を行った [5]。現在の運用状況も含めて、これらの調査および対処について詳細を報告する。

2. Archiver Appliance 運用状況

2.1 Archiver Appliance の概要

AA は、SLAC、BNL、MSU のコラボレーションによって開発されたアーカイバーである。データファイルを短期 (STS)、中期 (MTS)、長期 (LTS) データ保存用にステージを分けることで、ステージ間でデータを移動する組み込

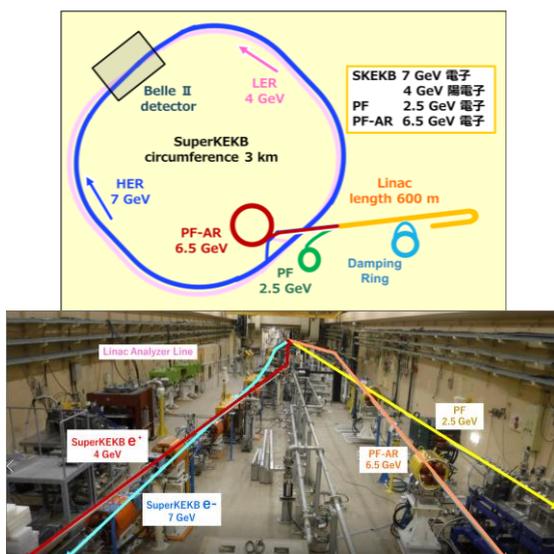


Figure 1: Schematic layout of KEK injector linac and related accelerators.

[#] Itsuka.satake@kek.jp

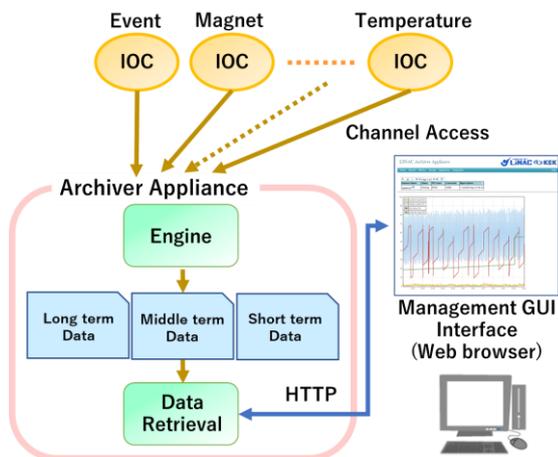


Figure 2: Schematic view of Archiver Appliance in the KEK injector linac.

Table 1: Software Environment of Archiver Appliance

Archiver Appliance	Nov 2019 Release
OS	CentOS 7.6.1810
Tomcat	apache-tomcat-9.0.36
Java	12.0.2 (OpenJDK 12.0.2)
Database	MySQL 5.7.25 (mysql-connector-java 8.0.15)

みプロセスにより、ストレージの高速化が実現でき、速い読み出しを可能にしている。また、異なる記憶媒体を使い分けることが可能である(Ramdisk, SSD, SAN など)。AA のアーカイブデータは、Google Protocol Buffers[6]を用いたバイナリファイルとして記録される。Figure 2 に、AA の KEK 入射器におけるシステム概要を示す。また、Table 1 にはソフトウェア環境を示す。

AA は 4 つのプロセス、すなわち、Engine、Retrieval、Mgmt、ETL から構成される。これらは、Apache Tomcat[7]を用いた Java Servlet であり、Servlet にはデータを収集する機能と、データを読み出す機能のものがある。前者は CSS Archiver と同様に Engine と呼ばれ、PV のデータを監視し、サンプリングして、値とタイムスタンプをディスクに書き込む。後者は、Retrieval と呼ばれる。Mgmt は、PV 追加削除やパラメータの変更をおこなう管理を担う。ETL は、STS から MTS へ、および MTS から LTS へのデータの移動を担っている。

クライアントおよび Retrieval 間の通信には HTTP を用いており、JSON や CSV などのフォーマットでのデータ取得が可能である。また、Web ブラウザを用いたシステム管理や監視機能をもつ管理インターフェイスや Business Process Logics (BPL) と呼ばれる API を利用したサンプルプログラム (Python 用いた処理スクリプト) が用意されている。

AA の Web ブラウザ上の管理画面からは、PV の追加、

停止および削除が可能である。また CSV ファイルのデータ出力などができる、Viewer 機能 (Fig. 3 (top)) も備わっている。Viewer としては、Google と個人や企業のコミュニティによって開発されている TypeScript ベースのオープンソースのフロントエンド Web アプリケーションフレームワークである Angular を用いており、Web ブラウザ上で表示可能である (Fig. 3 (bottom))。

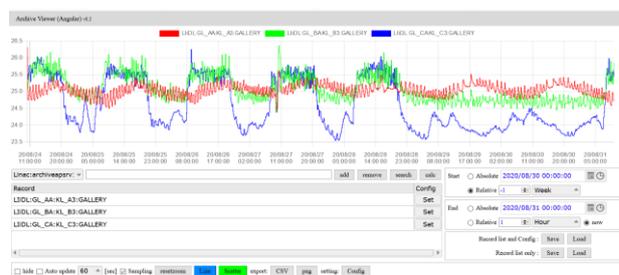
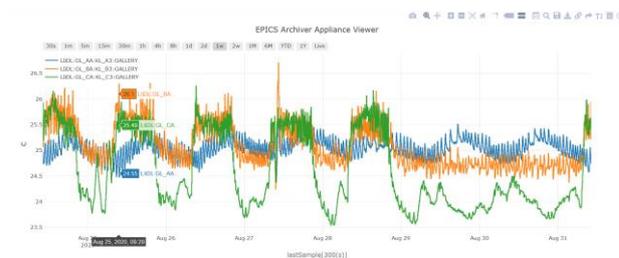


Figure 3: Screenshot of Archiver Appliance web viewer (top) and Archiver Appliance viewer using Angular (bottom).

2.2 従来の CSS Archiver との比較

2011 年から運用しているデータ収集ソフトウェアは、CSS archiver で構築されている [8]。CSS [9,10] とは、DESY で開発が開始された、加速器などの大規模制御システムを操作や監視するための制御システムツール群である。CSS にはデータ収集ツールや GUI、アラームシステムなどの様々な機能がある。

CSS Archiver は、EPICS 通信プロトコルである Channel Access を介して情報を収集する Archive Engine とリレーショナルデータベース管理システムである PostgreSQL で構成されている。Archive Engine は EPICS PV のデータを収集し、サンプリングした PV の値とタイムスタンプを PostgreSQL に記録している。データベース上にある情報の表示には、Adobe Flash Player を用いた Viewer を使用している。2020 年に Adobe Flash の配布が終了するため、AA と同様に Angular を用いた Viewer に変更している。

Table 2: Data Size of Archiver System

Archiver name	Data size/year
Archiver Appliance	~1.8 TB
CSS Archiver	~7.1 TB

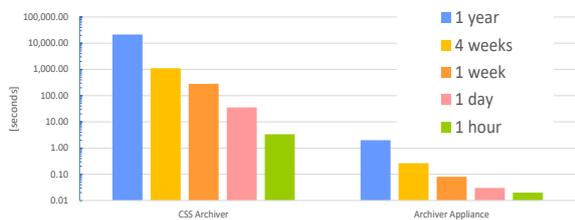


Figure 4: Benchmark of data retrieval from archive system.

KEK 入射器では、機器の追加などにより CSS Archiver のデータベースサイズが年々増大し、PostgreSQL からの読み出し速度の低下を懸念していた。このため、読み出し速度が速く、ディスク消費量の小さい AA の運用を始めた。各アーカイバーにおける 1 年分のデータサイズを比較したところ、AA は CSS Archiver に比べて約 1/4 のディスク消費量であった (Table 2)。また、読み出し速度の高速化も明らかであった (Fig. 4)。例えば、従来の CSS Archiver ではデータ表示に約 6 時間を要するものが、AA では数秒程度に短縮化された。

3. Broadcast に関する問題

AA 導入直後には、フィールド情報をもたない一部の PV (LabVIEW を用いた IOC など) に対して、フィールド情報を読みにくい現象により、broadcast が増加していたことがあった。これは、フィールド情報を読まない設定に変更することで対処した [11]。

しかし、その後もなお CSS Archiver と比較すると、接続できなくなった PV への CA サーチ (主な broadcast パケット) が多く、broadcast パケット数・パケットサイズが多いことがわかった。

これは、AA の CA サーチ間隔が短いことと、入射器制御ネットワーク内の beacon anomaly が多いことが原因だと考えられる。通常、beacon anomaly を出すのは IOC 再起動時などである。CA サーチ間隔が徐々に長くなっている中、beacon anomaly を受けると、CA サーチが最も短いサーチ間隔へ戻ることになる。そのため、最長の 30 秒間隔になる前に、何度も繰り返し CA サーチをとばすことになる。これが broadcast が急増してしまう主な原因だと思われる。caRepeater は、beacon anomaly を監視しており、再接続した PV をいち早く認識し、アーカイブを再開するために機能している。以上の問題から、下記の対処をおこなった。

- i. caRepeater を停止する。
- ii. 定期的に PV の接続状態を監視し、接続が復旧した PV に対してアーカイブを再開するプログラムを動作させる (cron で実行する)。

2 つ目の対処により、caRepeater を止めたデメリットを補うことができる。プログラムでは、paused/resume をそれぞれ 10/30 分ごとにおこない、極力接続できない PV へのアクセスを減らしている。現在、caRepeater を止めたことによる不具合は見ら

れていないため、上記の状態で運用することとした。

AA におけるデフォルトの CA サーチ間隔最長 30 秒は変更できない仕様であったが、AA の 2019 年 11 月バージョンでは、最長 300 秒で変更もできる仕様となった。

調査には Elastic Stack の一部であるデータ可視化ツールの Kibana と、ネットワーク情報を可視化している cacti を使用した。今回 KEK 入射器に導入した Elastic Stack は、ネットワークのパケットやプロセス情報を監視している。Elastic Stack のコアプロダクト群である、Logstash、Elasticsearch、Kibana は様々な

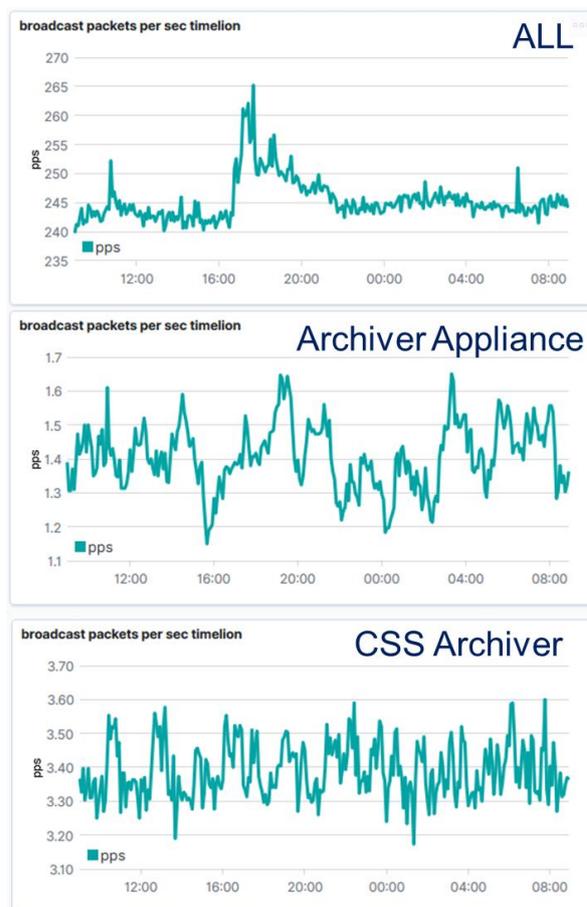


Figure 5: Broadcast packets of the KEK injector linac control network.

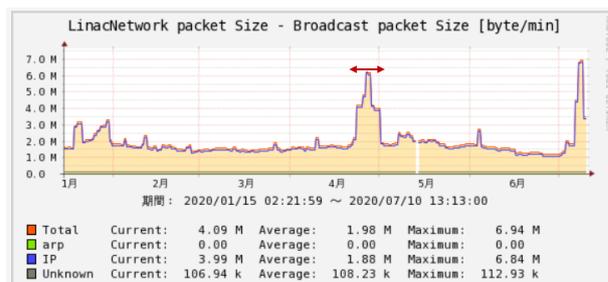


Figure 6: Broadcast packet size of the KEK injector linac control network.

ソースからデータを取得でき、リアルタイムに検索、分析、可視化することができる。ネットワークのプロセス情報に加えて、CA Packets や caSnooper などの Channel Access に関する情報も得ることができる。Figure 5 に、Kibana による broadcast パケット数を示す。1秒あたりのパケット数は、CSS Archiver と比較すると AA は半分ほどである。Figure 6 には、cacti による入射器ネットワークでの broadcast パケットサイズを示す。矢印で示した期間は caRepeater が動作しており、停止させることで約 4 Mbyte/min 減らすことができた。

4. 再接続後の CA サーチ問題

通常は PV の接続が切断された後、その PV へ CA サーチが発行されることになる。この PV が再接続されると、アーカイブが再開し CA サーチが消える。AA ではしばしば、一部の PV において再接続されても CA サーチが消えない問題が起こることがわかっている。再接続以降は、正常にデータがアーカイブされており、PV の status についても、Being Archived 状態に変化している。これらの問題は、Kibana の CA パケット情報を監視して調査した(Fig.7)。この症状が起こる PV は、特定の機器に限らないが、比較的発生頻度の高い PV グループがあることがわかった。これらの現象に対して、4通りの対処を試した。

- i. paused/resume をおこなう。
 - ii. paused および delete をおこない、再登録する。
 - iii. cleanUpAnyImmortalChannels コマンドを実行する。
 - iv. AA を再起動する。
- i, ii では、CA サーチが消えないままで効果がなかった。iii では、"/cleanUpAnyImmortalChannels" という Engine の BPL が実装されている。本来削除されるべきチャンネル(CA サーチが成功したチャンネルなど)が残っている場合があり、これを削除するための BPL で、今回の現象

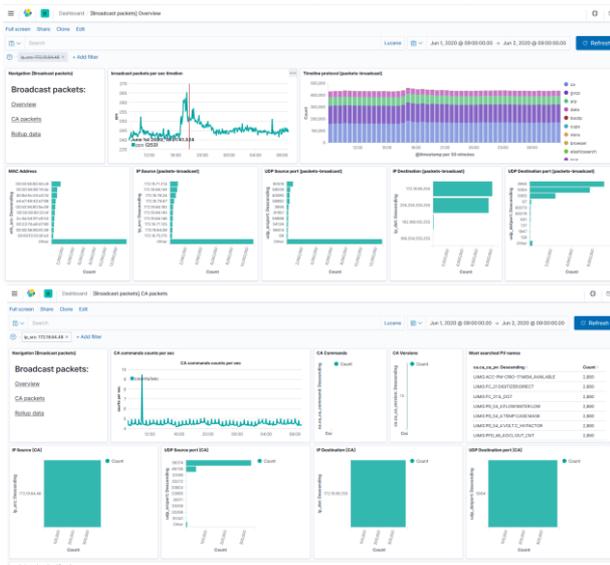


Figure 7: Broadcast packet (top) and CA packet data (bottom) of the KEK injector linac control network.

に似ているため実行を試みたが、該当しなかった。iv について、再起動後は残った CA サーチが消えた。

以上のことから、発生頻度の高い PV のグループをクラスタ化することとした。定期的に該当クラスタのみ再起動を実施することで、CA サーチが消えるよう対処する。いまだ原因は不明であるため、引き続き調査をおこなう予定である。

5. まとめと展望

AA は KEK 入射器の新たなアーカイブシステムとして、Angular を用いた Viewer とともに、高速なデータ読み出しとディスク消費量軽減を実現した。CSS Archiver は今後もバックアップとして運用する予定である。導入当初から、AA による broadcast が増加する問題が起きていたが、定期的に PV の接続状態を監視することで、現在は問題なく運用できている。一部の PV については CA サーチの異常な動作が見られるが、クラスタ化することで対処でき、Engine の負荷分散にも有効である。

今後はよりネットワークの監視を強化して、上記の問題解決と、さらなる操作性向上のためシステムの改良を続けていく。

参考文献

- [1] Mitsuo Akemoto *et al.*, Prog. Theor. Exp. Phys. (2013) 03A002.
- [2] K. Furukawa *et al.*, “Advanced acceleration mode switching for simultaneous top-up injection at KEK electron/positron injector linac”, in Proceedings of the 17th Annual Meeting of Particle Accelerator Society of Japan, Kyoto, Japan, Sep. 2-Sep. 4, 2020, WEOOP01.
- [3] EPICS;
<http://www.aps.anl.gov/epics/>
- [4] The EPICS Archiver Appliance;
https://slacmshankar.github.io/epicsarchiver_docs/
- [5] H. Sugimura *et al.*, “SuperKEKB 加速器の制御システム”, in Proceedings of the 16th Annual Meeting of Particle Accelerator Society of Japan, Kyoto, Japan, Jul. 31-Aug. 3, 2019, THOI06.
- [6] Google Protocol Buffers;
<https://developers.google.com/protocol-buffers/>
- [7] Apache Tomcat;
<http://tomcat.apache.org/>
- [8] T. Kudou *et al.*, “PRESENT STATUS OF CSS ARCHIVER AT KEK INJECTOR LINAC”, in Proceedings of the 12th Annual Meeting of Particle Accelerator Society of Japan, Tsuruga, Japan, Aug. 5-7, 2015, WEP113.
- [9] Control System Studio;
<http://cs-studio.sourceforge.net/>
- [10] Control System Studio (CSS) at KEK;
<http://www-linac.kek.jp/cont/epics/css/>
- [11] I. Satake *et al.*, “INTRODUCTION OF ARCHIVER APPLIANCE IN KEK ELECTRON POSITRON INJECTOR LINAC”, in Proceedings of the 16th Annual Meeting of Particle Accelerator Society of Japan, Kyoto, Japan, Jul. 31-Aug. 3, 2019, FRPH02.