

SuperKEKB における Archiver Appliance の現状

PRESENT STATUS OF ARCHIVER APPLIANCE AT SUPERKEKB

廣瀬雅哉^{#, A)}, 梶裕志^{B)}

Masaya Hirose^{#, A)}, Hiroshi Kaji^{B)}

^{A)}Kanto Information Service Co., Ltd.

^{B)}KEK

Abstract

We develop the new archiver system with the Archiver Appliance for the SuperKEKB project. It records a huge amount of parameters in the accelerator operation. We prepare the cluster for the individual hardware groups. Therefore the system consists of the 9 clusters. The new archiver system works properly without any serious problem in the 2020 spring run. We confirm the usability of the Archiver Appliance.

1. はじめに

SuperKEKB 加速器[1]は、KEK つくばキャンパスに建設された電子陽電子衝突型加速器である。その運転制御システムは、主に Experimental Physics and Industrial Control System (EPICS) [2]を用いて構築されている。

SuperKEKB では、加速器に関する各種データをアーカイブするシステムとして KEK が独自に開発した KEKBLog[3]を運用している。KEKBLog は前プロジェクトである KEKB 加速器時代から継続して運用されており、SuperKEKB 運転開始時には SSD を用いたキャッシュサーバーを導入してデータ読み出し速度の向上を実現する[4]など、継続的に改善がなされている。その一方で、KEKBLog へのアーカイブ対象の追加や削除等の作業は管理者しか行えず、時々刻々と変化する運転方針や手法に迅速に対応するには限界がある。ユーザー自身が CSS Archiver[5]をサブアーカイブシステムとして運用している事例もある[6]が、CSS Archiver の性能は決して高くなく、主力アーカイブシステムとして導入することには疑義が残る[7]。そこで我々は、既に J-PARC などで運用されている EPICS Archiver Appliance(AA)[8][9]の導入を検討するため、2019年3月から6月の運転時に試験運用を行なった[10]。この試験運用で良好な結果が得られたため、2020年春期運転では本格的な運用を行なった。

本稿では、これらの現状について述べる。

2. EPICS Archiver Appliance の概要

AA は、SLAC で開発されているオープンソースのアーカイブシステムである。このシステムは、Engine、ETL、Data Retrieval、Management といった4つの Java サブレットで構成され、Apache Tomcat[11]上で実行をサポートしている。Engine サブレットは、Channel Access(CA)プロトコルによって EPICS IOC 上のチャンネルと接続して Process Variable(PV)を取得し、Google Protocol

Buffers[12]を用いたバイナリ形式で保存する。アーカイブデータの記憶領域は、「Short Term Store (STS)」「Medium Term Store (MTS)」「Long Term Store (LTS)」の3つに分けられ、これらの管理は ETL サブレットが担っている。直近のデータは STS に保存されるが、設定された期間が経過すると、MTS、LTS へと移動する仕組みとなっている。

アーカイブデータの読み出しは Data Retrieval サブレットが担っており、JSON や CSV、MATLAB などといったバイナリ以外のフォーマット取得することをサポートしている。

Management サブレットでは、ウェブインターフェースによる管理機能が実装されており、アーカイブ対象の追加や削除、メトリクス情報の確認等が行える。また、Business Process Logic(BPL)という WebAPI が実装されており、クライアントは HTTP 通信によって AA の管理を行うことも可能となっている。

3. 高負荷環境における動作試験

SuperKEKB では、1台のサーバに複数の AA が動作し、かつ1ノードあたりのアーカイブ対象チャンネル数が1万点以上となることが想定されている。

昨年の試験運用では、2つの AA でクラスターを構成し、特に問題なく動作することを確認した。そこで今回は、各ノードへ実運用に近い負荷を課す試験を実施した。

3.1 試験環境

試験環境は、AA や RDBMS を動作させるための IU サーバーと、IOC を動作させるための PC を用意し、これら2台が同じネットワークに接続される構成とした。OS は CentOS7.6 に統一し、AA は v0.0.1_SNAPSHOT_09-Oct-2018、RDBMS には MySQL 5.7 を採用した。

アーカイブ対象となる IOC は、チャンネル数 1,000 の IOC を複数用意することで、チャンネル総数 10,000 を実現した。

[#]kan-hiro@post.kek.jp

3.2 試験結果

今回の試験では、最大約 12 時間ほど動作させた後、ab コマンド[13]である 1 チャンネルの全アーカイブデータのリクエストを 1,000 回（同時接続数 100）実行してみたが、目立った性能低下は見受けられず、特に問題なく動作することが分かった。ネットワークトラフィック量においては、IOC が停止した際の CA サーチパケットで 250pps 前後となっていたが、運用上大きな影響は無いと考えられる。

4. 運用状況

4.1 システム構成

SuperKEKB では、2 台のサーバーでそれぞれ複数の AA を動作させ、かつクラスタリングする構成としている。そのサーバの構成を Table 1 に示す。HP 製の 1U サーバーで、ストレージは HDD 9 台による RAID-5 構成とした。

AA は v0.0.1_SNAPSHOT_09-Oct-2018 のバージョンを使用し、Java には OpenJDK 8、RDBMS には MySQL 5.7、ウェブコンテナには Apache Tomcat 7 を採用した。

Table 1: Specification of Servers

	HPE ProLiant DL380 Gen10
CPU	Intel Xeon Bronze 3104 1.7GHz
OS	CentOS 7.6
Memory	8GB × 6
Disk	6TB HDD × 9 (RAID-5)

AA は各ハードウェアグループごとに用意し、Table 2 のように割り当てた。これらのクラスターは結合させ、互いにデータを読み出せるようにしている。

ストレージは、各サーバー計算機に内蔵している大容量 HDD のみで STS/MTS/LTS を構成している。それぞれの保存期間は、STS を 24 時間、MTS を 2 日間、LTS を 1 年間に設定し、これら 3 つのデータ記憶領域は各ハードウェアグループごとに作成している。

4.2 チャンネル数及びデータサイズ

アーカイブ対象チャンネルは、KEKBLog とほとんど同じとしたが、既に入射器グループで AA が運用されている[14]ため、本システムでは入射器関係のカテゴリに含まれているチャンネルはアーカイブしていない。

ここで、アーカイブ対象チャンネル数及び 2020 年 2 月から 6 月までにおける履歴データサイズを Table 3 に示す。履歴データサイズにおいては、2016 年に同様のアーカイブ期間、チャンネル数で試験運用していた CSS Archiver が約 13TB であった[7]のに対し、AA は約 3.6TB であり、とても小さくなっている。

Table 2: System Architecture

Server Name	Appliance ID	JVM Max Heap Size
server1	injection	1GB
	vacuum	2GB
	magnet	2GB
	BT	2GB
	monitor	2GB
server2	users	1GB
	RF	2GB
	control	1GB
	others	1GB

Table 3: Channels and Size of Archived Data

Appliance ID	Number of Archived Channels	Data Volume
injection	281	150GB
vacuum	6,437	555GB
magnet	64,038	99GB
BT	13,790	760GB
monitor	12,163	447GB
users	20	32GB
RF	32,281	1,549GB
control	1,987	11GB
others	312	57GB
Total	131,309	3,660GB

4.3 ウェブインターフェースにおける機能制限

我々は 9 つの AA を結合させてクラスターを構成している。DB やアーカイブデータを保存するディレクトリはノードごとに用意しているが、動作に関する設定は個性が出ないように共通の設定ファイルを読み込むようにしている。

AA 標準仕様の各ノード用ウェブページは、他のノードのアーカイブチャンネルも操作できてしまう。各ハードウェアグループのユーザーが操作することを考えた場合、他のグループが管理するアーカイブチャンネルを操作できてしまうのは好ましくない。我々のシステムでは、ウェブページのプログラムを修正し、他のノードへの操作を制限することにした。

4.4 運転シフト専用クラスター

各ハードウェアグループ向けに構築したクラスターとは別に「運転シフト専用クラスター」を構築した。

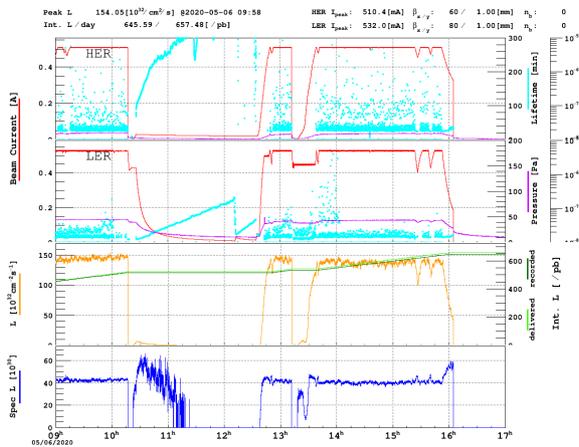


Figure 1: SuperKEKB Operation Summary.

このクラスターはシングルノードで運用し、報告資料の作成に必要な PV のみをアーカイブする。ストレージは短期間の保持を前提としており、STS には RAM を採用して直近のデータ読み出し速度の高速化を図っている。

SuperKEKB では、ROOT[15]を用いて Figure 1 のような運転時のシフトレポート図を作成し、シフト引継ぎ時の報告などに利用できるように整備を進めている。

5. 運用上の課題

5.1 CA サーチャパケットの送出量の低減

EPICS Archiver Appliance では、Channel Access (CA) ライブラリとして JCA が使われている。JCA は、EPICS 標準の CA ライブラリ (libca) とは若干異なる実装となっており、CA サーチャに関する挙動に違いがあることが分かっている[16]。

例えば、マグネット関係の EPICS IOC が再起動された時で約 450pps 前後の CA サーチャパケットが発生していた。同時期の KEKBlog が約 100pps 前後であったため、倍以上の CA サーチャパケットが送出されている。

これまでの運用では、この事象に関連した重大な問題は確認されていないが、今後アーカイブ対象が増加する可能性は十分考えられるため、ネットワークに大きな負荷を与えてしまうことが懸念される。

5.2 接続済み PV への CA サーチャパケットの発生

アーカイブが開始された PV であるにも関わらず、その PV に対する CA サーチャパケットが送出され続けてしまう現象が発生した。今回の運用では 1 度だけ発生した問題であり、アーカイブには影響はなかった。この問題はアプライアンスを再起動することで解決することが分かっているが、根本的な原因は理解されておらず、早期解決が難しい。

6. まとめと今後の展開

SuperKEKB 加速器の後継アーカイブシステムとして、我々は AA によるアーカイブシステムを構築し、運用を行った。運用上の課題がいくつか確認されたが、実際の運転環境下においても概ね順調に動作することが確認できた。

今後は、より安定した運用を実現するために、サーバーリソースの拡張や監視体制の強化など、運用環境のさらなる整備を進めていく。

参考文献

- [1] Y. Ohnishi *et al.*, “Accelerator design at SuperKEKB”, Prog. Theor. Exp. Phys., 2013, 03A011
- [2] EPICS;
<http://www.aps.anl.gov/epics/>
- [3] T. T. Nakamura *et al.*, “Data Archiving System in KEKB Accelerators Control System”, Proceedings of the 10th ICALEPCS, Geneva, Oct. 10-14 2005;
https://accelconf.web.cern.ch/accelconf/ica5/proceedings/pdf/P1_077.pdf
- [4] A. Morita *et al.*, “Improvement of data archive reading speed for SuperKEKB”, Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan, Chiba, Japan, Aug. 8-10, 2016, TUP093.
https://accelconf.web.cern.ch/accelconf/ica5/proceedings/pdf/P1_077.pdf
- [5] K. Kasemir, “RDB Channel Archiver”, March 2010;
<https://ics-web.sns.ornl.gov/css/docs/RDBChannelArchiver.doc>
- [6] K. Aoki *et al.*, Abstracts of CSSJ Conference, Vol.92(2015)p.85.
- [7] M. Hirose *et al.*, “Present status of CSS Archiver and Channel Archiver at SuperKEKB”, Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan, Chiba, Japan, Aug. 8-10, 2016, TUP091.
- [8] The EPICS Archiver Appliance;
http://slacmshankar.github.io/epicsarchiver_docs/
- [9] S. Yamada *et al.*, “Deployment of archiver appliance at J-PARC main ring”, Proceedings of the 14th Annual Meeting of Particle Accelerator Society of Japan, Sapporo, Japan, Aug. 1-3, 2017, WEP100;
https://accelconf.web.cern.ch/accelconf/ica5/proceedings/pdf/P1_077.pdf
- [10] H. Kaji *et al.*, “Application of EPICS Archiver Appliance at SuperKEKB”, Proceedings of the 16th Annual Meeting of Particle Accelerator Society of Japan, Kyoto, Japan, Jul. 31-Aug.3, 2019, THPH004.
- [11] Apache Tomcat;
<http://tomcat.apache.org/>
- [12] Google Protocol Buffers;
<https://developers.google.com/protocol-buffers/>
- [13] <https://httpd.apache.org/docs/2.4/programs/ab.html>
- [14] I. Satake *et al.*, “Introduction of archiver appliance in KEK electron positron injector LINAC”, Proceedings of the 16th Annual Meeting of Particle Accelerator Society of Japan, Kyoto, Japan, Jul. 31- Aug.3, 2019, FRPH002.
- [15] <https://root.cern>
- [16] M. Hirose *et al.*, “Evaluation of the CSS V4 Alarm System for SuperKEKB”, Proceedings of the 14th Annual Meeting of Particle Accelerator Society of Japan, Sapporo, Japan, Aug. 1-3, 2017, TUP093;
https://accelconf.web.cern.ch/accelconf/ica5/proceedings/pdf/P1_077.pdf