

## J-PARC ハドロンビームライン用データアーカイブシステムの開発(3)

### DEVELOPMENT OF DATA ARCHIVE SYSTEM FOR J-PARC HADRON BEAMLINE(3)

豊田晃久<sup>#</sup>, 青木和也, 上利恵三, 秋山裕信, 家入正治, 加藤洋二, 倉崎るり, 里嘉典, 澤田真也, 白壁義久, 高橋仁, 田中万博, 広瀬恵理奈, 皆川道文, 森野雄平, 山野井豊, 渡辺丈晃

Akihisa Toyoda<sup>#</sup>, Kazuya Aoki, Keizo Agari, Hironobu Akiyama, Masaharu Ieiri, Yohji Kato, Ruri Kurasaki, Yoshinori Sato, Shinya Sawada, Yoshihisa Shirakabe, Hitoshi Takahashi, Kazuhiro Tanaka, Erina Hirose, Michifumi Minakawa, Yuhei Morino, Yutaka Yamanoi, Hiroaki Watanabe

KEK

#### Abstract

As the data archive system of J-PARC hadron beamline, we have adopted the Channel Archiver since the start of operation in 2009 and RDB Channel Archiver from 2017. However, the Channel Archiver and the RDB Channel Archiver are not currently being maintained, it is necessary to shift to the next system. This time we will test and adopt the Archiver Appliance as a new system, and present performance comparison and future prospects.

#### 1. はじめに

J-PARC ハドロンビームラインでは、2009 年の運転開始以来制御フレームワークとして EPICS[1]を使用し、そのデータアーカイブシステムとしては Channel Archiver[2]を使用してきた。しかしこの Channel Archiver はメンテナンスされなくなったため、CSS(Control System Studio)に付属する RDB Channel Archiver[3]を稼働し、性能確認およびデータベースのリプリケーション化およびパラメータ最適化による性能向上を行い[4,5]、2017 年ごろから本格運用してきた。ところがこの RDB Channel Archiver もメンテナンスされなくなったため、今回 Archiver Appliance[6]を試用して性能確認およびクラスター化による性能向上を行った。

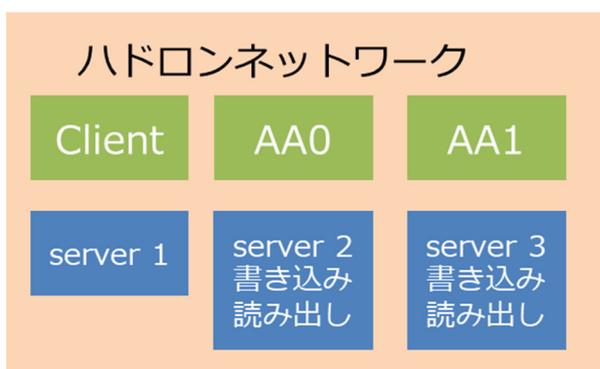


Figure 1: Archiver Appliance system framework.

#### 2. ハドロンアーカイブシステムの構成

ハドロンアーカイブシステムの EPICS channel 数は約 3400 で、サンプリング周期は 0.1 秒から 10 分、1 channel あたりのワード数は 1 から 30000 である。

Channel Archiver および RDB Channel Archiver のシ

<sup>#</sup> akihisa.toyoda@j-parc.jp

ステム構成は前回の報告[5]の内容と同じである。今回の報告する Archiver Appliance のシステム構成を Fig. 1 に示す。サーバーは CPU が Intel Xeon E5-2620 v3 2.4 GHz 6 コア 12 スレッドで、メモリが 128 GB、ハードディスクは SATA 7200 rpm 1 TB の 4 台から RAID10 構成である。AA0 は Archiver Appliance の 1 台目のサーバーで、AA1 は途中で追加したクラスター化のための 2 台目のサーバーである。ちなみに AA0, AA1 のいずれも EPICS GW は使用していない。またストレージについては、短期、中期、長期ともにハードディスクを使用している。Client は AA0 もしくは AA1 にアクセスしてデータを取得する。

#### 3. 試験内容とその結果

データは 2017 年および 2018 年の運転時データを利用した。以下の試験内容を実施した。

- Archiver Appliance の運転時で負荷がかかった状態でのデータ取得速度の測定
- Archiver Appliance のクラスター化による性能測定および性能向上。
- 従来のアーカイブとの比較。

##### 3.1 データ取得速度の測定方法

データ取得速度の測定方法としては前回の方法を踏襲した[5]。具体的な測定方法を以下に再掲する。

- 低負荷のデータ取得対象として、1 spill(5.2 秒)に 1 回更新する(1 point/spill)1 ch のレコード(1 ch/point)を選んだ。これを Scaler データと呼ぶ。
- 中負荷のデータ取得対象として、1 spill に 1 回更新する(1 point/spill)64 ch のレコード(64 ch/point)を選んだ。これを Array データと呼ぶ。
- 高負荷のデータ取得対象として、1 spill に 1 回更新する(1 point/spill)3000 ch のレコード(3000 ch/point)を選んだ。これを Large Array データと呼ぶ。

データ取得クライアントとしては、Channel Archiver と RDB Channel Archiver については前回と同じ python-

xmlrpc および python-MySQL ライブラリを使用した。Archiver Appliance については python URLlib2 ライブラリを使用した。

Figure 2 にデータ取得速度測定のためのフィッティングの様子を示す。横軸 60000 point が約 4 日に相当する。青丸が Channel Archiver、赤丸が RDB Channel Archiver、緑丸が Archiver Appliance のデータである。それぞれ 0 点を通る 1 次関数でフィットした。

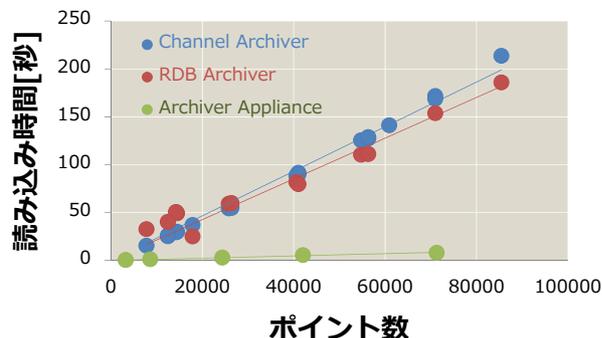


Figure 2: Example of fitting for measuring data readout speed.

### 3.2 Archiver Appliance の性能試験

まず初めに Archiver Appliance をスタンドアロンで構成した。具体的には Fig. 1 の AA0(Archiver Appliance 0)を構成した。その上で運転中の負荷の有無による性能差を測定したのが Table 1 である。これにより運転中で書き込み負荷が大きい状況でも Archiver Appliance のデータ取得性能に影響はないことが分かる。参考までに AA0 の HDD 書き込みスループットについて、運転中は約 400 kBytes/s、運転停止中は約 20 kBytes/s である。

Table 1: Accelerator Operation Load Dependence of Data Acquisition Speed of the Archiver Appliance

データの種類	運転中のデータ取得速度(ch/秒)	運転停止中のデータ取得速度(ch/秒)
Scaler	$3.2 \times 10^4$	$3.2 \times 10^4$
Array	$5.4 \times 10^5$	$5.4 \times 10^5$
Large Array	$1.3 \times 10^6$	$1.3 \times 10^6$

### 3.3 クラスタ化による Archiver Appliance の性能向上

続けてクラスタ化によって Archiver Appliance の性能向上を図った。具体的には Fig. 1 の AA1(Archiver Appliance 1)を新たに追加し、AA0 から 100 ch 程度移行した。その際にストレージ消費率がおおよそ AA0:AA1=2:1 になるようにチャンネルを選んでいる。

Table 2 に AA0 からデータを取得した場合と AA1 からデータを取得した場合のデータ取得速度の違いを測定した結果を示す。ちなみにデータを取得した対象のチャンネルは AA0 に記録されている。どちらの

Appliance から取得してもデータを取得速度に差がないので、AA0 と AA1 の間の通信は主要なオーバーヘッドになっていないことが分かる。

Table 2: Archiver Appliance Dependence of Data Acquisition Speed of the Archiver Appliance

データの種類	AA0からのデータ取得速度(ch/秒)	AA1からのデータ取得速度(ch/秒)
Scaler	$3.2 \times 10^4$	$3.2 \times 10^4$
Array	$5.4 \times 10^5$	$5.4 \times 10^5$
Large Array	$2.8 \times 10^6$	$2.8 \times 10^6$

Table 3 にクラスタ化したことによるデータ取得速度の向上を示している。低負荷および中負荷では有意なクラスタ化の効果はないが高負荷では約 2 倍の速度向上効果があることが分かる。

Table 3: Cluster Effect of Data Acquisition Speed of the Archiver Appliance

データの種類	クラスタ化前のデータ取得速度(ch/秒)	クラスタ後のデータ取得速度(ch/秒)
Scaler	$3.2 \times 10^4$	$3.2 \times 10^4$
Array	$5.4 \times 10^5$	$5.4 \times 10^5$
Large Array	$1.3 \times 10^6$	$2.8 \times 10^6$

### 3.4 従来のアーカイバとの比較

前節で性能向上の処置を行った Archiver Appliance と従来のアーカイバとの比較を行った結果を Table 4 に示す。Archiver Appliance はどのデータ種類においても Channel Archiver および RDB Channel Archiver のいずれよりも 6 から 40 倍程度高速で、かつ高負荷であるほどより高速にデータが取得できることが分かる。

Table 4: Summary of Data Acquisition Speed of the Channel Archiver, the RDB Channel Archiver, and the Archiver Appliance

データの種類	Channel Archiver のデータ取得速度(ch/秒)	RDB Channel Archiver のデータ取得速度(ch/秒)	Archiver Appliance のデータ取得速度(ch/秒)
Scaler	$1.4 \times 10^3$	$5.1 \times 10^3$	$3.2 \times 10^4$
Array	$2.7 \times 10^4$	$3.5 \times 10^4$	$5.4 \times 10^5$
Large Array	$6.6 \times 10^4$	$7.3 \times 10^4$	$2.8 \times 10^6$

最後にストレージ消費率の比較を表 5 に示す。RDB Channel Archiver はストレージ消費率が高い問題があったが、Archiver Appliance は Channel Archiver に比べても低いストレージ消費率となっていることが分かる。

Table 5: Comparison of the Storage Consumption Rates of the Channel Archiver, the RDB Channel Archiver, and the Archiver Appliance

	Channel Archiver	RDB Channel Archiver	Archiver Appliance
(GB/日)	1.86	32.7	1.69

#### 4. まとめ

今回は新しく Archiver Appliance をアーカイバとして試験採用し、性能向上および性能評価をおこなった。その結果、従来のアーカイバである Channel Archiver や RDB Channel Archiver と比べて、データ取得速度については 6-40 倍高速であり、かつストレージ消費率についてもより少ないことが確認できた。またデータ欠けなどの問題もないことが確認できていることから、次のランからは Archiver Appliance をハドロンアーカイブシステムとして正式に採用する予定である。過去のデータについては順次 Archiver Appliance に移行していく予定である。

#### 参考文献

- [1] EPICS Web page; <http://www.aps.anl.gov/epics/>
- [2] Channel Archiver Web page; <https://ics-web.sns.ornl.gov/kasemir/archiver/>
- [3] Control System Studio Web page; <http://controlsystemstudio.org>
- [4] A. Toyoda *et al.*, “DEVELOPMENT OF DATA ARCHIVE SYSTEM FOR J-PARC HADRON BEAMLINE”, Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan (2016) MOP091, pp 630-633.
- [5] A. Toyoda *et al.*, “DEVELOPMENT OF DATA ARCHIVE SYSTEM FOR J-PARC HADRON BEAMLINE(2)”, Proceedings of the 14th Annual Meeting of Particle Accelerator Society of Japan (2017) WEP099, pp 1140-1143.
- [6] Archiver Appliance Web page; [https://slacmshankar.github.io/epicsarchiver\\_docs/](https://slacmshankar.github.io/epicsarchiver_docs/)