

## SuperKEKB における CSS Archiver 及び Channel Archiver の現状

### PRESENT STATUS OF CSS ARCHIVER AND CHANNEL ARCHIVER AT SUPERKEKB

廣瀬雅哉<sup>\*A)</sup>, 岩崎昌子<sup>#B)</sup>, 帯名崇<sup>B)</sup>, 佐々木信哉<sup>B)</sup>, 中村達郎<sup>B)</sup>, 中村卓也<sup>C)</sup>

Masaya Hirose<sup>\*A)</sup>, Masako Iwasaki<sup>#B)</sup>, Takashi Obina<sup>B)</sup>, Shinya Sasaki<sup>B)</sup>, Tatsuro Nakamura<sup>B)</sup>, Takuya Nakamura<sup>C)</sup>

<sup>A)</sup> Kanto Information Service Co.,Ltd.

<sup>B)</sup> High Energy Accelerator Research Organization (KEK)

<sup>C)</sup> Mitsubishi Electric System & Service Co.,Ltd.

#### Abstract

Test operation of SuperKEKB accelerator had been carried out up to June 2016 from February 2016 (Phase-1). In the Phase-1, the CSS Archiver and the Channel Archiver had been operated in parallel with KEKBLog, which is the main data archiving system. The number of archiving channels is expected to increase more than KEKB. Therefore, we have carried out measuring performance in the actual high-load environment during the Phase-1 operation in order to check the effectiveness of the CSS Archiver and the Channel Archiver. This paper describes the operation status of the CSS Archiver and the Channel Archiver.

#### 1. はじめに

高エネルギー加速器研究機構(KEK)では、KEKB 電子・陽電子ビーム衝突型加速器を用いた Belle 実験が行われてきた。現在、KEKB 加速器の更なる高輝度化を目的として、SuperKEKB 加速器の建設が進められており[1]、2016年2月から2016年6月には試験運転が行われた(Phase-1)。

KEKB 加速器では、加速器に関する各種データを蓄積するためのシステムとして、KEKB で独自に開発した KEKBLog[2]を運用し、運転用アプリケーションの開発には SAD[3]や Python、MEDM[4]が主に用いられてきた。SuperKEKB では、これらにあわせて、Control System Studio (CSS) [5]を用いた加速器運転用アプリケーションの開発を行い、さらに、CSS をベースとした加速器データ蓄積システムとして、CSS Archiver[6]の導入も試みている。ここで、CSS は、広く加速器・素粒子・原子核実験コミュニティにより開発、使用された実績を持つソフトウェアツールであり、CSS および CSS Archiver に対する、ユーザーコミュニティによるソフトウェアサポートを期待することができる。我々は、CSS Archiver を実際の SuperKEKB 加速器運転(Phase-1)で試験運用を行い、加速器運転による高負荷環境下でのパフォーマンスの測定や改善を行った。

なお、今回の運用では CSS Archiver を補助する目的で、Channel Archiver[7]も並行して運用した。CSS Archiver 及び Channel Archiver は KEK 入射器で既に運用されており[8]、次のようなことが分かっている。CSS Archiver はデータ量が多くなるにつれ、読出し速度の低下やデータベースサイズが大きく肥大化してしまうことが懸念されている。この点については Channel Archiver の方が優れているが、インデックスファイル破損による障害が度々発生してしまうといったことが懸念されている。

本稿では、これらの詳細について述べる。

#### 2. Channel Archiver 及び CSS Archiver

##### 2.1 Channel Archiver

Channel Archiver は Spallation Neutron Source (SNS) で C++を用いて開発された、Experimental Physics and Industrial Control System (EPICS) [9]のデータを記録するためのツールセットである。その中心となる Archive Engine は Channel Access (CA) を介して複数の CA Server (IOC) からデータを収集する。収集したデータは定期的に独自のバイナリ形式でローカルディスクファイル(データファイル)に書き出され、インデックスファイルによって関連付けられる。これらの履歴データは XML-RPC を実装した Data Server を起動することで、ネットワーク上の他計算機からも読出すことができる。

##### 2.2 CSS Archiver

CSS は、加速器のような大規模制御システムを操作・監視するために開発されたツール群である。これらのツールは Eclipse をベースに DESY で開発が開始され、現在では多くの研究機関が共同開発しており、Linux や Windows、Macintosh など複数のプラットフォームで動作可能としている。CSS Archiver は Channel Archiver の置き換えとして開発されたアーカイブシステムである。Figure 1 に示す通り、基本的な構成は同じであるが、収集したデータの保存先(バックエンドデータベース)に Oracle や MySQL、PostgreSQL といったリレーショナルデータベース(RDB)が採用されている。これにより、Archive Engine と RDB を別々の計算機上で運用させることも可能である。

\* kan-hiro@post.kek.jp

#Present affiliation is Osaka City University.

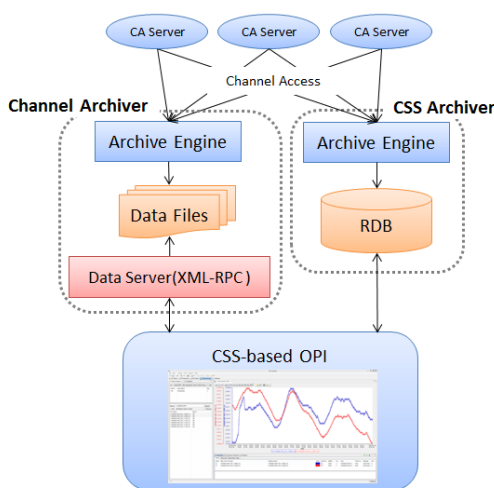


Figure 1: Channel Archiver and CSS Archiver overview.

### 3. 運用状況

#### 3.1 システム構成

はじめに、Channel Archiver 及び CSS Archiver の運用目的を以下に示す。

- 短期間のデータは CSS Archiver
- 長期間のデータは Channel Archiver

これらのアーカイブシステムは 1 台のサーバで並行運用した。そのサーバ構成を Table 1 に示す。Dell 製の 1U サーバで、ストレージは全 8 台ある HDD のうち 7 台で RAID-6 を構成し、残りの 1 台を予備ディスクとした。容量は 30 TB である。

Table 1: Specification of Server

	Dell PowerEdge R430
CPU	Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60GHz
OS	CentOS Linux release 7.1.1503 (Core)
Memory	32 GB
HDD	Dell Storage MD1400 6TB SAS Disk

Channel Archiver は 3.0.0 のバージョンを使用し、localhost 上の複数の Archive Engine のステータスを Web 上で管理できるツール (Archive Daemon) を用いて運用するように構築した。Archive Daemon はツールセットの 1 つであり、Figure 2 に示す通り、チャンネルの接続状況を確認できる他、Archive Engine の操作や自動再起動などの設定もできる。また、今後のデータ保守性やインデックスファイル破損による影響を考慮し、毎日にディレクトリ及びインデックスファイル、データファイルが作成されるように整備した。ただし、このままでは 1 日分のデータしか読み出すことが出来ないため、これらインデックスファイルを統合した master\_index ファイルを作成し、ユーザに提供している。master\_index ファイルは、イ

ンデックスファイルが更新される度に更新しなければならないが、リアルタイムに行うことは現実的に難しい。このため、非同期ではあるが、crontab を用いて定期的に更新するように整備した。

CSS Archiver は KEK 版 3.2.16.2 のバージョンを使用し、バックエンドデータベースには、これまでに利用実績のある PostgreSQL 9.4.5 を用いた。運用にあたっては、Figure 3 に示す通り、Channel Archiver 同様に複数の Archive Engine を Web 上で管理できるツール (Archive Reports) を導入した。Archive Reports[10]は SNS で JSP を用いて開発され、ステータスの閲覧の他、ユーザログインすればアーカイブチャンネルの追加・変更・削除を行うこともできる。

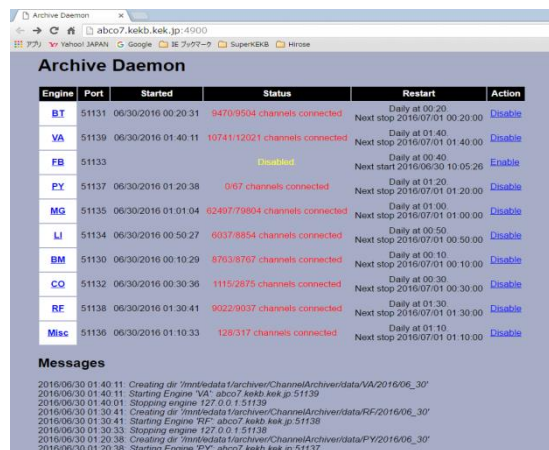


Figure 2: Archive Engines monitoring by the Archive Daemon.

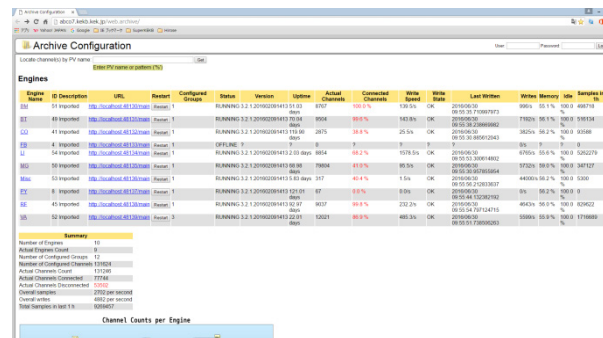


Figure 3: Archive Engines monitoring by the Archive Reports.

#### 3.2 PostgreSQL チューニング

運用計算機のリソースを考慮し、メモリ及びチェックポイント関連の一般的なチューニングを実施した。この他、読出し速度を向上させる目的で、PostgreSQL のパーティショニング機能を用いて、履歴データテーブルを月単位で分割し、各テーブルに対してインデックスを作成している。また、CSS Data Browser では読出した履歴データを間引きして表示する機能があるが、間引き処理をクライアント側で行っているため、データ転送量が多くなってしまふ。そこで、データ転送量を抑える目的で間引

き処理を行うストアドファンクションを導入した。

### 3.3 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。アーカイブ対象チャンネルの変更を反映させるには Archive Engine の再起動が必要となる。再起動による停止時間をできる限り少なくする目的で、KEKBlog と同様にカテゴリ毎に Archive Engine を分けるようにした。

ここで、アーカイブ対象チャンネル数及び履歴データサイズの推移を Table 2 に示す。約 947 億点の履歴データを記録し、CSS Archiver では約 13 TB、Channel Archiver は約 3.3 TB のデータサイズとなった。ここで、CSS Archiver は sample 及び array\_val の 2 つのテーブルに記録するため、これらのテーブルを合計したもの、Channel Archiver は du コマンドでインデックスファイル及びデータファイルの合計を出力したものである。Table 2 が示す通り、CSS Archiver のデータサイズは Channel Archiver の約 4 倍となってしまうほど肥大化した。なお、データサイズの増え方に違いが出ているのは、Channel Archiver で接続出来ているチャンネルが CSS Archiver では接続できなかったことが主な原因である。

Table 2: Channels and Size of Archived Data

	Number of Archived Channels	CSS Archiver	Channel Archiver
2016.01	146,218	1.3 TB	144 GB
2016.02	149,259	2.1 TB	499 GB
2016.03	149,128	2.4 TB	648 GB
2016.04	155,328	2.3 TB	742 GB
2016.05	155,353	2.4 TB	717 GB
2016.06	131,563	2.5 TB	553 GB
Total		13 TB	3.3 TB

### 3.4 履歴データ読出し速度

履歴データは CSS Data Browser などのクライアントプログラムによってグラフとして表示することができ、このプロセスは、サーバ側での処理、データ転送、クライアント側での処理の大きく 3 つに分けることができる。特にサーバ側での処理に時間がかかってしまうと、CSS Archiver は SQL の実行時間を、Channel Archiver は ArchiveExport コマンドの実行時間を測定した。いずれも時間範囲とチャンネル名を指定し、その結果を出力するものである。SQL の実行時間は EXPLAIN ANALYZE コマンド、ArchiveExport の実行時間は time コマンドで算出したものとしている。今回は 1Hz でプロセスする真空のコールドカソードゲージの値をモニタするチャンネルをサンプルとして読出し速度を測定した結果を Table 3 に示す。

今回はデータの開始日時を固定し、そこから日数を増やして測定を行った。履歴データの点数に多少の違いはあるものの、実行時間に大きな差が見られた。なお、

今回のチューニングでは Table 3 の結果となったが、チューニング内容によってパフォーマンスは変わってくるため、今回は異なるチューニングを実施したうえでの読出し速度の測定は課題の一つとして考えている。

Table 3: Reading Speed of Archived Data

	CSS Archiver		Channel Archiver	
	Time(sec)	Samples	Time(sec)	Samples
1 day	59.6	70,567	1.36	70,427
2 days	57.6	146,812	1.96	146,587
7 days	140.68	410,921	4.12	410,643
30 days	592.52	1,793,305	25.61	1,826,366

## 4. 課題

### 4.1 Channel Archiver

開発が既に終了しているため、保守面での期待ができないのが課題として挙げられる。今後、バグが見つかる可能性も考えられるが、そのサポートは見込めない。また、仕様により、master\_index が 2GB までしか作成できず、全ての履歴データを読出すことが出来なくなる問題が発生した。この他、本システムのように日毎にインデックスを作成する状況下では、履歴データが更新される度に master\_index を再作成する必要があるため、リアルタイムな読出しには向いていない。

### 4.2 CSS Archiver

今回の運用では、バージョンアップの検討とバックエンドデータベースの選択が課題として挙げられる。

まず、バージョンアップの検討であるが、採用した KEK 版 3.2.16.2 では CA 関連のトラブルが発生し、履歴データが欠落してしまう問題が発生した。発生したトラブルについては、原因が特定できなかったものもあるが、一部はバグとして既に報告されており、当該バージョンよりも新しいバージョンでは修正されている。開発は現在も続いているため、保守や改善が見込める。また、常に最新バージョンの情報を取得し、対応していく必要はあると考えている。

一方、今回特に課題となるのが、バックエンドデータベースの選択である。アーカイブするチャンネルの数は、KEKB 加速器運転時よりも増えることが見込まれるうえ、リソースは限られているため、データサイズは極力少ないのが望ましい。この改善案として、aquenos 社が開発した Apache Cassandra[11]をバックエンドデータベースとする Cassandra Archiver for CSS[12]を導入を検討することが考えられる。Cassandra Archiver は KEK 入射器で既に試験されており、RDB と比べて高速な動作及びデータベースサイズの減少が期待できる[8]。この他、1 か月以内といった短期間の履歴データ保存に特化し、それ以前の履歴データは削除するという運用方針とすることも改善案の一つとして考えられる。

## 5. まとめ

CSS Archiver が SuperKEKB 加速器のアーカイブシステムとして導入可能かどうかを検証するため、実際の高負荷環境でのパフォーマンス測定や改善を行ってきた。また、補助目的で並行運用した Channel Archiver についてもパフォーマンスを測定することができた。しかし、現段階では SuperKEKB 加速器のアーカイブシステムとして導入可能かどうかを判断するのは難しい状況である。今後、Phase-2 に向けて検討していく必要がある。

## 参考文献

- [1] K. Akai *et al.*, “Design Progress and Construction Status of SuperKEKB”, Proceedings of IPAC12, pp. 1822-1824 (2012);  
<http://accelconf.web.cern.ch/AccelConf/IPAC2012/papers/tuppr006.pdf>
- [2] 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/ica05/proceedings/pdf/P1\\_077.pdf](https://accelconf.web.cern.ch/accelconf/ica05/proceedings/pdf/P1_077.pdf)
- [3] <http://acc-physics.kek.jp/SAD/>
- [4] <http://www.aps.anl.gov/epics/extensions/medm/>
- [5] <http://controlsystemstudio.org/>
- [6] K. Kasemir, “RDB Channel Archiver”, March 2010;  
<https://ics-web.sns.ornl.gov/css/docs/RDBChannelArchiver.doc>
- [7] K.U. Kasemir and L.R. Dalesio, “Overview of the Experimental Physics and Industrial Control System (EPICS) Channel Archiver”, October 2001;  
<https://accelconf.web.cern.ch/accelconf/ica01/papers/THAP019.pdf>
- [8] T.Kudou *et al.*, “Present status of CSS Archiver at KEK Injector Linac”, Proceedings of the 12th Annual Meeting of Particle Accelerator Society of Japan, Tsuruga, August 5-7, 2015;  
[http://www.pasj.jp/web\\_publish/pasj2015/proceedings/PDF/WEP1/WEP113.pdf](http://www.pasj.jp/web_publish/pasj2015/proceedings/PDF/WEP1/WEP113.pdf)
- [9] <http://www.aps.anl.gov/epics/>
- [10] <https://ics-web.sns.ornl.gov/css/products.html>
- [11] <http://cassandra.apache.org/>
- [12] S. Marsching *et al.*, “Scalable archiving with the Cassandra Archiver for CSS”, Proceedings of ICALEPCS2013, San Francisco, CA, USA;  
<http://accelconf.web.cern.ch/AccelConf/ICALEPCS2013/papers/tuppc004.pdf>