KEK 入射器における CSS Archiver の現状

PRESENT STATUS OF CSS ARCHIVER AT KEK INJECTOR LINAC

工藤拓弥#, A), 草野史郎 A), 佐藤政則 B), 古川和朗 B)
Takuya Kudou #, A), Shiro Kusano A), Masanori Satoh B), Kazuro Furukawa B)
A) Mitsubishi Electric System & Service Co., Ltd.
B) KEK, Accelerator Laboratory/SOKENDAI, Department of Accelerator Science

Abstract

In the KEK injector linac, the device history information has been stored to the CSS-based Archiver system since 2011. Though PostgreSQL is used as the backend database in this system, its large database file size brings a slow readout speed performance. For this reason, we divide the database file into smaller size one corresponding to each month. In addition, we reorganize the tables in PostgreSQL database by using the pg_reorg without any locks. These measures bring a smaller database file size and the improvement of readout speed performance. In this paper, we present the status of CSS Archiver system used for the KEK injector linac operation in detail.

1. はじめに

KEK 入射器（以下、入射器）では、各制御機器の状態監視を目的として、設定値および出力値の変化情報履歴を蓄積している。KEKB 運転開始時には、機器グループごとに専用の履歴ソフトウェアを開発し、運用してきた。2004 年に、Spallation Neutron Source（SNS）で開発された Channel Archiver を導入した。Experimental Physics and Industrial Control System（EPICS）ツール群のひとつである Channel Archiver は、導入以降安定に運用されてきたが、インデックスファイル破損による障害が度々発生した。障害発生時には既に Channel Archiver の開発は終了しており、障害対処は困難であった。そこで、SuperKEKB 主リング制御システムにて導入予定である Control System Studio[1]（CSS）Archiver を導入し、2011 年に運用を開始した。本稿では、CSS Archiver の導入および運用状況について報告する。

2. CSS Archiver システム

CSS は、DESY で開発が開始され、現在、多くの研究機関が共同開発している制御システムツール群であり、Eclipse Platform 上で動作するものである。CSS には、加速器制御に要求される標準的なコンポーネントが豊富に用意されているため、短期間でのシステム開発が可能である。また、Linux[2]、Windows[2]、および Macintosh[2]など複数の OS 上で動作可能なため、様々なシステムへ容易に導入することができる。CSS は、前述のような利点から各研究所での利用が広まっており、SuperKEKB 主リング制御システムでも導入が予定されている。SuperKEKB 主リング運転開始以降は、入射器とのデータ相関を解析することも想定されるため、CSS のツールの一つである CSS Archiver を導入した。

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 およびそれらを記録するバックエンドデータベース部から構成されている。バックエンドデータベースは、Oracle[2]、MySQL[2]、および PostgreSQL[2] などのリレーショナルデータベース管理システム（RDBMS）から選択可能であるが、入射器では、電子ログブックシステム[3]での運用実績のあった PostgreSQL を採用している。CSS Archiver においては、記録対象の追加や削除などの変更を反映させるために Archive Engine の再起動が必要になる。入射器では、Archive Engine 再起動による履歴情報収集の停止時間を最小限に抑えるため、デバイスごとに Archive Engine を分けて運用している。SuperKEKB に向けた改造による制御機器の増加にともない、Archive Engine 数および記録対象情報点数も増加していった。Table 1 に示すとおり、2015 年 7 月現在、Archive Engine 数は 19、記録対象は約 55000 点にのぼる。

2011 年の運用開始から現在まで、約 270 億点の履歴情報を記録し、データベースのファイルサイズは 3.2 TB まで巨大化したが、比較的安定に運用をおこなっている。

Table 1: Current Operation Status

Operation date	2011/08 ~ 2015/07
Number of Archive Engine	19
Number of EPICS PVs	55597
Total number of archived data points	27.4 billion
Database file size	3.2 TB

3. CSS Archiver の運用状況

3.1 概要

CSS Archiver は、履歴情報収集系から表示ツールまで揃っているため、独自の開発をおこなわずに運

kudoh@post.kek.jp

用を開始することができた。しかしながら、運用期間の長期化にともない蓄積データサイズは巨大化し、読み出し速度の低下がみられるようになった。入射器では、利便性の向上、長期運用に向けたデータ管理、および読み出し速度の高速化を目的として、以下のような対策を実施している。

3.2 Viewer web アプリケーション

CSS Archiver の履歴データは、CSS ツール群の一部である Data Browser を用いて参照可能である。しかしながら、使用する各端末へ CSS をインストールする必要があるため利便性に乏しい。そこで、Adobe Flash[2]を用いた Viewer web アプリケーションを開発し、運用している。Adobe Flash を用いたアプリケーションは、単純な HTML で記述されたページと比して操作性および表現力に優れており、Adobe Flash Player[2]が動作する計算機環境であれば OS を問わず動作可能である。

バックエンドデータベースと Viewer アプリケーション間の通信には、ActionScript Message Format と呼ばれるバイナリフォーマットを基盤とした amfphp[4]を用いている。これにより、軽量かつ高速に通信をおこなうことが可能となり、アプリケーションのパフォーマンス向上に大きく寄与している。

本アプリケーションは、複数のデータを同時にグラフ化することが可能である。また、マウス操作によるグラフの拡大ができるため、複数データ間の相関解析を容易におこなうことができる。データ読み出し期間は、絶対日時指定に加え相対日時指定も可能となっており、自動更新機能と組み合わせることで、最新の履歴情報を自動で表示することができる。これにより、機器の状態監視が容易におこなえるようになった。Figure 1 は、本アプリケーションの表示例を示している。

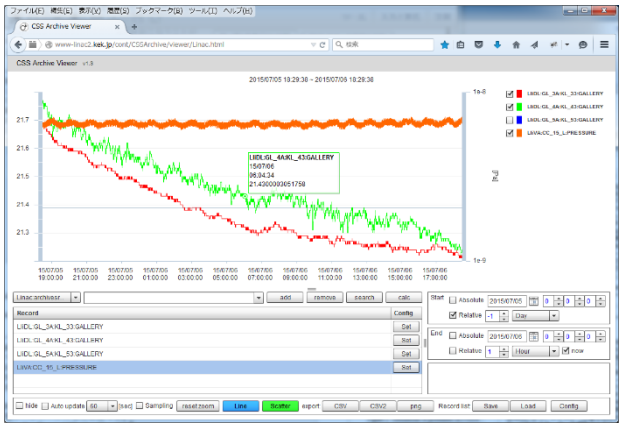

Figure 1: Viewer web application.

3.3 データベースのパーティショニング

本システムのデータベースサイズは、運用開始から半年間で 362 GB にまで肥大化した。今後の連続運用により、さらにデータベースサイズは肥大化し、一つのディスク上に格納できなくなることが予測された。そのため、本システムでは 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ている。PostgreSQL は、テーブルごとに格納先を指定できるテーブルスペースと呼ばれる機能を有しており、この機能を用いて分割した履歴データテーブルを複数のディスク上に分散して格納することが可能である。パーティショニング機能は、分割した複数のテーブルを一つのテーブルとして見せることができる。このため、パーティショニング導入による参照側の変更は不要であった。また、パーティショニング導入による読み出し速度の高速化も期待されたが、速度面での向上は見られなかった。

3.4 ローカルディスクを用いた専用サーバ計算機

バックエンドデータベースに使用している PostgreSQL などの一般的な RDBMS では、ディスクアクセスが読み出しおよび書き込み速度高速化のボトルネックになる。メモリ上に全てを展開できる小規模データベースとは異なり、本システムのような巨大データベースにおいては、ディスクアクセスが発生することは避けられない。そこで、高信頼性かつ低アクセス速度のネットワークディスクを用いるサーバ計算機および低信頼性かつ高アクセス速度のローカルディスクを用いたサーバ計算機の 2 台を用いた並列運用をおこなっている。Table 2 に示すとおり、ローカルディスクを使用するサーバ計算機は、ネットワークディスク上のデータ読み出しと比較して、約 10 倍高速に読み出しをおこなうことが可能であり、通常の読み出しにはこちらを使用している。

Table 2: Read Speed of 1-Hour History Data

Network disk	0.835 sec
Local disk	0.091 sec

3.5 データベースチューニング

バックエンドデータベースに使用している PostgreSQL は、チューニング未実施の状態に於いても一定の性能を確保することができる。しかしながら、サーバ計算機の性能を最大限活用するためのチューニングをおこなうことにより、読み出しおよび書き込み速度を大きく向上させることが可能である。本システムでは、メモリ関連の一般的なチューニングに加え、同期コミットの無効化を実施している。通常、PostgreSQL はデータの永続性を保証するために、ディスクへの書き込みが完了するまで待つ同期書き込みと呼ばれる機能を使用するが、同期コミットの無効化により、ディスクへの書き込みが非同期になる。これにより、データ損失のリスクは微増するが処理速度は大きく向上する。本システム稼動中のサーバ計算機においてディスク入出力待ちが頻発し、処理速度が低下していたが、同期コミットの無効化によりディスク入出力待ちを大きく低減することができた。これにより、読み出しおよび書き込み速度を大きく向上することができた。

3.6 データベースの再編成

バックエンドデータベースに使用しているPostgreSQL は、追記型アーキテクチャを採用しているため、データ更新を頻繁におこなうことにより実データサイズ以上にデータベースファイルサイズが肥大化する。これは、断片化と呼ばれる現象で有り、速度性能の著しい低下をもたらす。CSS Archiver は、履歴データの追記のみをおこなうため、データの更新は発生せずデータ自体は肥大化しない。しかしながら、読み出し速度高速化のために使用しているインデックスは、履歴データが追記されるごとに更新を繰り返し、サイズが増加し続ける。PostgreSQL の CLUSTER コマンドにてデータを再編成することにより、これを解消することが可能であるが、対象のテーブルがロックされてしまうため、システムの一時的な停止が要求される。

このため、本システムではテーブルのロックを必要としないデータ再編成ツールである pg_reorg[5]を使用している。これを用いることにより、運用中にシステムを停止することなく、データの再編成を定期的におこなうことが可能となった。Table 3 に示すとおり、定期的なデータ再編成の導入により、インデックスサイズが半減し、読み出し速度は導入前と比べ約 3 倍高速になった。

Table 3: Result of 1-Month Data Reorganization

	Before using pg_reorg	After using pg_reorg
Data size	14 GB	14GB
Index size	12 GB	665 MB
Read speed of 1 day history data	2.237 sec	0.744 sec

4. 課題

4.1 PostgreSQL 以外のデータベースの検討

本システムのバックエンドデータベースに使用している PostgreSQL を始めとした RDBMS は、データの一貫性を保証するように設計および最適化されている。そのため、本システムのようなデータ追記が主である巨大システムでは、満足な速度性能を達成することが一般的に困難である。

速度性能の改善を目的として、バックエンドデータベースに Apache Cassandra[6]を用いる Cassandra Archiver for CSS[7]の導入を検討している。Apache Cassandra は、Facebook 社にて開発が開始された大規模データ格納用のデータベース管理システムである。個々を識別する情報とそれに対応する値を格納するキー・バリュー型と呼ばれる単純な仕組みを採用しているため、RDBMS と比較して高速な動作が期待できる。また、データを複数の計算機で分散して格納する機能を有しており、サーバ計算機の台数を増やすにつれ性能や耐障害性を向上させることが可能である。Cassandra Archiver for CSS の運用試験をおこなったところ、Table 4 に示すとおりバックエンドデータベースに PostgreSQL を用いる場合に比べ、データベースサイズが約 1/10 程度に大きく減少することが確認できた。現在も運用試験を継続しており、読み出し速度試験、長期安定試験を実施する予定である。

Table 4: Database File Size of 1-Day Data

Backend database	Database file size during 1 day operation
PostgreSQL	16 GB
Apache Cassandra	1.65 GB

4.2 データ欠落問題

本システムは、履歴データ最終記録日時や接続の成否などの情報を、記録対象ごとに参照することが可能である。この情報の最終記録日時ではデータ収集を続けているように見えるが、実際にはバックエンドデータベースにデータが記録されていないという不具合が度々発生した。不具合が発生していたサーバ計算機にて、メモリ不足によるスワップ使用が頻発していたため、より搭載メモリの多いサーバで Archive Engine を動作させるように変更したところ不具合が発生しなくなった。問題調査のために構築した試験環境では症状が再現せず、明確な原因は特定できていないが、サーバ計算機のメモリ不足に起因してデータ欠落が発生したと推測される。

5. まとめ

入射器では、バックエンドデータベースに RDBMS を使用する CSS Archiver を導入し、2011 年に運用を開始した。運用を重ねるにつれ、データベースサイズは大きく肥大化し、読み出し速度の低下が見られた。このため、データベースのパーティショニング化、運用中のデータベース再編成などの対策を実施することにより、データベースファイルサイズの縮小および読み出し速度の向上を実現できた。現在、バックエンドデータベースに Apache Cassandra を使用する Cassandra Archiver for CSS の試験中であり、満足な試験結果が得られれば、実運用に使用する予定である。今後も、新システムの試験および現システムの改良を重ね、安定した入射器ビーム運転に貢献していきたい。

参考文献

[1] http://controlsystemstudio.org/.
[2] 本論文中で使われているシステム・製品名は、一般に各社の商標または登録商標です。
[3] T. Kudou et al., “Upgrade of Electronic Logbook System at KEK Injector Linac”, Proceedings of the 8th Annual Meeting of Particle Accelerator Society of Japan, Tsukuba, Aug. 1-3, 2011.
[4] http://www.silexlabs.org/amfphp/.
[5] https://github.com/ossc-db/pg_reorg/.
[6] http://cassandra.apache.org/.
[7] http://oss.aquenos.com/epics/cassandra-archiver/.