

SPring-8/SACLA における加速器時系列データベースへの二次データ保存体系 A SCHEME TO STORE PROCESSED DATA INTO THE WELL-ORDERED DATABASE STORAGE FORMAT AT SPring-8/SACLA

岡田謙介^{#, A)}, 岩井瑛人^{A) B)}, 清道明男^{A)}, 鈴木伸司^{A)}
Kensuke Okada^{#, A)}, Eito Iwai^{A)}, Akio Kiyomichi^{A)}, Shinji Suzuki^{A)}
^{A)} JASRI ^{B)} RIKEN

Abstract

At SPring8/SACLA, a centralized time-series database system helps the accelerator performance improved. The system is in operation under a set of signal types for those the readout API and the web-based visualization are equipped. However there have been requests to include irregular type of signals such as measurements from outside of the accelerator control network or statistics from an application. By introducing an intermediate stage of in-memory database, the operations in the user side and the system side can be separated. It allows the database system to store variety of data points without special treatment and under an organized way. Data points from the SACLA beamline and from the SPring-8 bunch purity measurements are the first applications and the system is running stably for a year.

1. 背景

SPring-8/SACLA の加速器データ収集システムは、機器の制御コマンドと連携した信号フォーマットを規定し、原則 `get command` で取得できる値をデータベースに記録している[1,2]。蓄積したデータは `web` での簡単なグラフ表示環境の他、データベースアクセス `API` を利用して、加速器周りの環境の変動や設定の経緯を振り返り、運転調整計画の向上に役立っている。

ただし、これまではデータ収集の対象は加速器制御系に閉じており、SACLA の X 線強度などビームライン側の機器による測定結果は別のシステムで記録し、必要に応じて 60Hz の繰り返し周期に同期したイベント番号で 2 つのシステムを照合して、調整の経緯を確認してきた。近年、ビームライン機器の測定結果から指標を生成し自動的に加速器調整を実行する機械学習の手法を導入し、これまでの運転担当者のスキルのみ頼った調整から

抜け出しつつある[3]。この状況では、ビームライン機器からの指標と加速器設定値が短い時間単位で密に連携するため、同じ解析基盤で振り返ることの利便性が求められていた。

また機器制御の設計方針としてステータスを基本としているため、時間をかけて統計情報を利用する場合や、フィッティングパラメータを測定値とする場合、複雑な演算で測定値を加工する場合などとの相性が悪かった。このような場合にデータベースに記録する適切な解決策が必要だった。

2. 加速器データ収集インフラへの集約方針の検討

ここでユーザーをデータ収集インフラ管理者以外として定義し、どの測定値、設定値を記録し後で活用するかを把握している者とする。

規格外のデータをデータ収集インフラの一部に組み込み、読み出しの利便性を担保していく上で以下のように

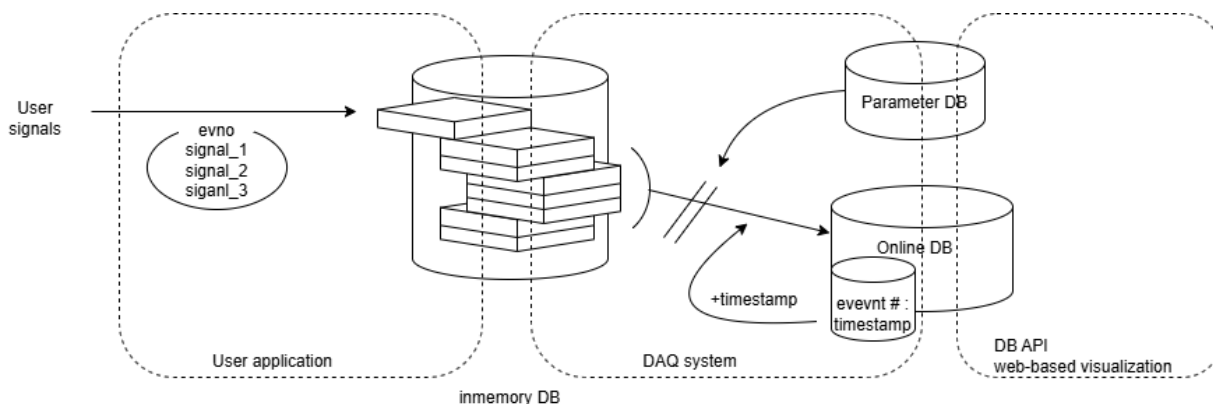


Figure 1: A DAQ concept for processed data. An in-memory DB mediates between a user and the DAQ system. The DAQ allows only signals registered in the parameter DB. The timestamp is attached to each point if necessary.

[#] k.okada@spring8.or.jp

な方法が考えられる。

- A) 新たなデータ収集系、読み出し系をユーザー毎に作成する。
- B) 既存の収集系のデータベース書き込み部分を API 化してユーザーに提供する。制御端末上のアプリケーションとしてユーザーが演算結果を保存する。
- C) ユーザーとデータ収集系の中間に取り合い点を作成し、データ更新の取り決めを行う。データ保存形式は既存の収集系を踏襲する。

A は保存フォーマットを新たに規定し、書き込みと読み込みの API を専用を用意する。用途に応じた柔軟な対応が取れる利点があり、以前の制御系[4]ではこの方法をとっていたが、開発、検証のために導入に時間がかかること、案件毎の個別対応となりシステム変更が必要になった際に障害となることが問題である。

B は読み出し側の変更が必要なく、従来のインフラがそのまま使用できる。ただし、ユーザーがデータベースへの書き込みを直接行うことになり、システム管理者が把握できないデータ書き込みが増える危険や、最悪の場合はユーザーによるデータベース破壊の危険が考えられる。

C は、中間に取り合い点を設けることで既存の書き込みシステムとして管理でき、不適切な書き込みをはねられることに利点がある。

今回 C の方針を取ることを決定した。システムの概要を Fig. 1 に示す。取り合い点にはインメモリデータベースの Redis[5]を使用し、書き込み先のフォーマットは mdaq_sync format (signal_id, timestamp, evno, value の組 [2])とした。ユーザーは必要に応じて読み出し時に evno によって信号間のデータの関連付けを行う。

SACLA ビームライン (BL) データの取り扱いではもう一つ考慮が必要な点があった。BL データは 60Hz の周期でカウントアップする event number のみ記録されており、加速器側とは下位 32 ビット分で連携しているが、リセットする機会があるためデータベースのインデックスとして不具合がある。そこで event number と timestamp の紐づけを加速器側データベースの別の信号を参照して行う処理を加えた。Figure 1 では DAQ system 右下の部分に示した。

3. データ収集(DAQ)アルゴリズム

3.1 Redis format

Figure 2 にデータの取り合い点となる Redis のフォーマットを示す。これは SACLA の機械学習調整において目的関数となる BL 側データと機械学習アルゴリズム、機器設定 GUI の三者を結び付けているフォーマットを踏襲した。ユーザーは同じ event number を共有する複数信号データを Hash type で書き込み、String type の ready tag を更新する。Redis には型の概念はないが、取り決めとして実数値の信号は”_F”, 整数値の信号は”_I”で終わる変数名を付けることとした。”timestamp_F”は予約語で、

timestamp を設定する。最終的に timestamp は取得時刻としてデータベースに書き込まれる。

Figure 3 はデータ収集系で動作する複数 thread の動作を模式的に示した。watcher_pol thread が daq_cycle の定期間隔で Redis にアクセスし、前回処理済みの部分から ready tag または時刻参照データの決定分まで、daq_func thread が sync data format でデータベースに書き込む。timestamp が付与されない場合に対して timestamp_ref thread が event number と timestamp の関係を参照データを使って作成し shared memory を介して他の thread と情報を共有する。これも定期的 (tagtime_ref_interval) に確認し、カウントがリセットされた場合や、時刻のずれに対応する。

```

ready tag 127.0.0.1:6379[2]> get|latestTag_I|
"1574506543"

data      127.0.0.1:6379[2]> hgetall|shot_tag#1574506543
1) "photonEnergy_F"
2) "10.515565"
3) "pulseEnergy_F"
4) "361.298645"
5) "pulseEnergyBM2_F"
6) "9.032466"
7) "BLvalue_F"
8) "1.368436"
9) "timestamp_F"
10) "1655791493.600141"
11) "SCM3_charge_F"
12) "240472.578125"
13) "SCM3_meanX_F"
14) "209.869278"
    
```

Figure 2: Redis data format. Once the data are set for a tag, the user application updates the ready tag. (In this Fig., it is 1574506543.) “timestamp_F” is a reserved word.

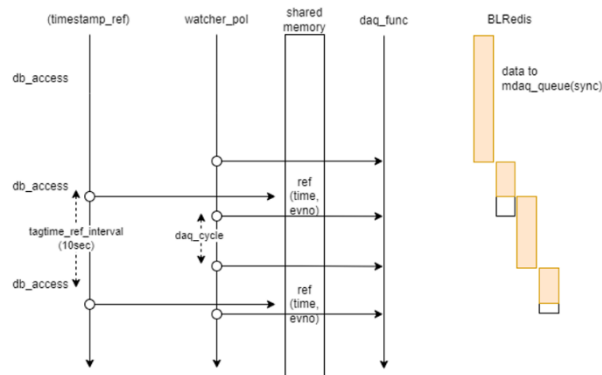


Figure 3: Threads in the DAQ system. “watcher_pol” reads the Redis periodically and “daq_func” writes the DB. “timestamp_ref” updates evno:timestamp LUT in the shared memory.

3.2 信号管理

Redis server のホスト情報や Redis 上の取り合いの信号名、時刻情報参照信号名の定義のためにパラメータデータベースに新たに管理テーブルを追加した。Figure 4 に信号管理関連部分の ER 図を示す。新たに

追加した部分は図の右下の3テーブルである。読み出し時は従来の SIG_INF と SIG_TABLE の2つの既存のテーブルを使用し、保存フォーマットは sync data format と変更なしのため、読み出し API の側の改造は不要である。信号登録手続きのため、信号登録用のシートに記入タブを追加し、信号登録 web[1]の改造を行い、通常信号登録申請のルーチンに乗せた。

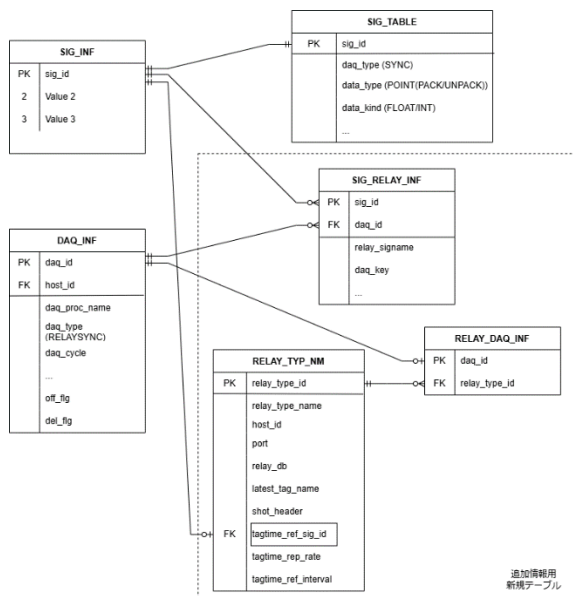


Figure 4: Signal information in the parameterDB. The three tables in the bottom-right corner are added for the DAQ system described in this paper.

3.3 インフラ保護の設計

ユーザーが安心して開発するためにも、既存のデータを破壊しないこと、無駄にストレージ容量を消費しないことをシステムとして担保することが重要である。

今回、インフラ保護の観点から次の条件を組み込んだ。

- A) Redis 上の信号名は事前の信号登録を済ませてある必要がある。書き込みプロセスは信号管理テーブルの Redis 信号名 (`{relay_signame}` in SIG_RELAY_INF)をもとに Redis を読み込む。これにより把握していない書き込みを防止できる。
- B) セットが完了したタグ番号 (`{latest_tag_name}` in RELAY_TYP_NM の値)を`#(Ltag)`として、際限なく過去までさかのぼることはせずに、`#(Ltag)-N`以降のタグを持つデータだけを対象にする。現在 `N=800` をハードコーディングしている。これは、意図せず誤ったタグ番号で Redis に書き込んでしまった場合の保護である。
- C) `timestamp_F` を現在時刻と比較し、規定の時間(現在は 30 分)以内のデータのみを処理する。この条件で、過去のデータの上書きを防ぐ。

4. 運用例

4.1 SACLA ビームライン機器のデータ収集(別ネットワークセグメントデータ収集)

SACLA 加速器調整は、利用実験のための X 線自由電子レーザーの質を上げるために行うが、X 線の計測機器のあるビームライン (BL) とは制御ネットワークセグメントを分けている。調整を手動で行っている間は、制御室に BL ネットワーク端末を別途設置し目視で状況把握する程度で十分だったが、機械学習で自動的に最適解を探るためには加速器側で BL データの取得が必須となる。そのための取り合いの仕組みとしてインメモリデータベースの Redis [5]が導入された。本システムの導入は調整過程の振り返りや、新しいロジックの導入検討にあたっての相関解析などの目的で、BL データへのアクセスを簡便にするためである。

Table 1 に SACLA BL 信号の一部を示した。pulse energy は頻繁に目的関数に含まれる。実験の種類によってはエネルギーFWHM や X 線ビーム中心位置に重きを置くことがある。その他、機械学習の目的関数の試行段階に対応するため、`ch1~5` といった含みを持たせた信号名も作成した。スタディを経て、新たに意味のある信号名に移行し、定点観測としての意味を持たせることを期待している。

Table 1: SACLA Beamline Signals (a Part)

signal	evno	value
<code>_bl_2_photon/energy</code>	553265879	7.48628 keV
<code>_bl_2_pulse/energy</code>	553265879	365.438 μ J
<code>_bl_2_SCM3/meanx</code>	553265879	363.181 pixel
<code>_bl_2_SCM3/meany</code>	553265879	198.553 pixel
<code>_bl_2_NISpec/charge</code>	553265879	13797.3 au
<code>_bl_2_NISpec/fwhm</code>	553265879	31.4583 eV

4.2 バンチ純度モニターデータ収集(統計データ収集)

放射光利用用途によっては、背景事象との分離に放射光の時間構造が重要である。その際 SPring-8 ストレンジリングを周回する電子は 508MHz の RF バケットに収まるので、メインバンチに対して、理想的には空のサテライトバンチへの電子の混入が不純度となる。バンチ純度モニターは偏向磁石放射光の可視光を利用するモニターで、メインバンチからの強い光の時は光シャッターを閉じ、サテライトバンチの微弱光の時は光シャッターを開けることにより、光子計測の大きなダイナミックレンジ ($\sim 10^9$) を実現している[6]。一式の測定は一つのメインバンチアドレスに対する一つのサテライトアドレスの不純度として GUI で集計し統計データとして算出され、一式の処理が終わったところで同じ evno を割り振って予め登録した信号名で取り合い点の Redis に書き込む仕様となっている。Table 2 に記録信号の一部を示す。光子数計測値から消光比を踏まえて純度 (purity) を算出している。

トップアップ入射を行って電子ビームを保持している間、常時リング一周の不純度の時間経過を監視しており、

Bunch mode により、計測時間は異なる(203 bunch の A モードで約 20 時間、孤立バンチの H モードで 100 分など)が、一周分の計測が終わったところで、evno をもとに再集計してリング全体の純度状況の把握を行っている。

Table 2: Purity Signals (a Part)

signal	evno	value	note
_main/address	508306	1044	
_main/count	508306	1913 cnt	5 set
_satellite/address	508306	1045	
_satellite/count	508306	41 cnt	80 set
_satellite/purity	508306	2.05094e-09	
_trigger/count	508306	50000	cnt/set

5. まとめ

SPring-8/SACLA において、運転性能の向上のために加速器機器の測定値、設定値の時系列データを保存し、解析を行うための環境を整備してきた。しかし、これまで加速器制御ネットワーク外のビームライン機器のデータや、アプリケーションで統計処理を行った後のデータは信号登録と読み出し方法の規格から外れ、データベース環境の恩恵を受けられずにいた。今回インメモリデータベースのRedisを中間に設け、取り合い点のルール

を作成することで、様々なユーザーが安全に記録信号を追加できるようにした。SACLA のビームライン機器データの記録、SPring-8 のバンチ純度モニターの統計値の記録に適用し、安定に運用している。

参考文献

- [1] K. Okada, T. Maruyama, T. Fukui, “SPring-8/SACLA 加速器ログデータベース利用環境の構築” 第 18 回日本加速器学会年会(2021)プロシーディングス TUP039.
- [2] K. Okada, R. Fujihara, T. Maruyama, “加速器データログのための NoSQL データベース (Apache Cassandra) 安定運用” 第 17 回日本加速器学会 (2020) プロシーディングス THPP23.
- [3] E. Iwai *et al.*, “Spectral-brightness optimization of an X-ray free-electron laser by machine-learning-based tuning”, J. Synchrotron Rad. 30, 1048-1053, 2023.
- [4] A. Yamashita and M. Kago, “MADDOCA II Data Logging System Using NoSQL Database for SPRING-8”, in Proc. 15th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'15), Melbourne, Australia, Oct. 2015, pp. 648-651.
- [5] Redis - The Real-time Data Platform, <https://redis.io/>
- [6] K. Tamura and T. Aoki, “SINGLE BUNCH PURITY DURING SPRING-8 STORAGE RING TOP-UP OPERATION”, Proceedings of the 1st Annual Meeting of Particle Accelerator Society of Japan and the 29th Linear Accelerator Meeting in Japan (2004).