

## SPring-8/SACLA 加速器ログデータベース利用環境の構築

### ASSEMBLING AN USER ENVIRONMENT OF ACCELERATOR LOG DATABASE AT SPRING8/SACLA

岡田謙介<sup>#, A)</sup>, 福井達<sup>B)</sup>, 丸山俊之<sup>B)</sup>

Kensuke Okada<sup>#, A)</sup>, Toru Fukui<sup>B)</sup>, Toshiyuki Maruyama<sup>B)</sup>

<sup>A)</sup> JASRI

<sup>B)</sup> RIKEN

#### Abstract

A database system is in operation at SPring-8/SACLA to record time series data such as environmental measurements, outputs of sensors synchronized with the accelerator cycle. The data stored are available for the current machine status, checking a correlation between measurements, and so on. In this presentation, we report on the user-side environment. The access to the database directly related to the accelerator operation needs a special care, so that the control applications are basically limited to compiled executables linking C/C++ DB access libraries. For monitoring or analysis purpose, a web-cgi service has long been provided, and a Restful API has been launched recently. Since the write permission of the web-cgi is limited to administrators to protect DB contents, the option of the user-side modification is limited. The Restful API supposed to solve this dilemma, and to be used even in an accelerator operation at tuning stages. A signal registration procedure is also shown. A web interface is developed to manage a series of registration processes including syntax checks, transferring request files over the firewall, and registration to the test/production database. This environment is to be the baseline of the new synchrotron facility built in Tohoku and future SPring8-II facility.

#### 1. はじめに

SPring-8/SACLA 加速器の制御系は、建設当時の MADOCA 制御系[1]のコンセプトを引き継ぎ、操作対象に対して英語第5文型メッセージの命令を送ること、中央のデータベースで一元的にログデータを収集することなどの特徴がある。技術面では切り替えを進め、例えばデータベースエンジンは、SACLA の繰り返し周波数 60Hz に対応するためもあって初期の RDB (SAP ASE[2]) は、大量の書き込みに適したカラム型 NoSQL (Apache Cassandra[3]) に置き換えた[4,5]。この Apache Cassandra の運用側面については過去の報告を参照されたい[6]。

一方、ログデータ活用のユーザーインターフェース (UI) については膨大な運用用の GUI プログラムの蓄積があったこともあり手を入れずに、極力後方互換性を重視し、先に制御システム基幹部分の更新を進めてきた。しかし、近年 Python に代表される便利なスクリプト言語の台頭、ゲームやスマートフォンに慣れた目からくる GUI への要求から、UI についても整理を進めている。

本発表では我々のログデータベース利用環境の全体コンセプトと提供している各インターフェースについて述べる。長期的サポートの観点から、メンテナンス性と整合性を重視している。

#### 2. 利用シーン

加速器の各所に設置したデータ収集サーバーから送られる、環境測定値、入射器のパルス周期に同期した機器の読み出し値などは、時系列データとして中央のデータベースに集約される。このデータの利用目的として、こ

こでは以下のものに分類する。

- **operation(運転)**: 現在の状態を機器の設定に反映させ、設定パラメータをデータベースに保存するなどの加速器の運転に関わる操作。
- **monitoring(モニタリング)**: 継続的な監視や、機器のドリフトの確認。
- **analysis(解析)**: データ間の相関解析や過去データの確認。運転条件毎に切り出して可視化など。

これらはネットワークのゾーンで分割し、特にユーザー利用運転の質に直結し慎重に取り扱う必要がある「運転」については Control (制御) 内でのみ許可している (Fig. 1)。

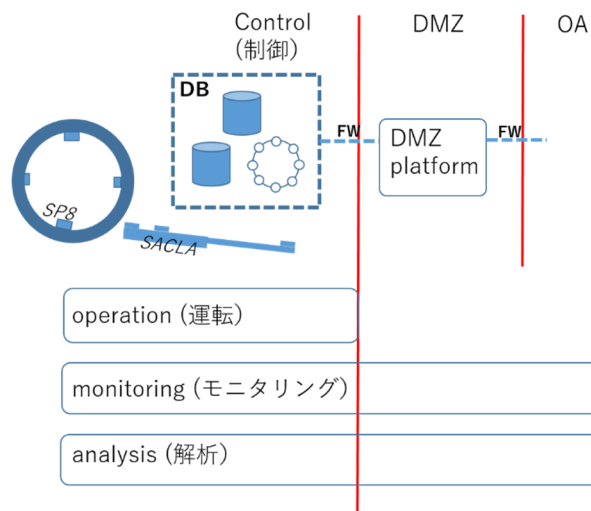


Figure 1: The concept of network zoning for the database access.

<sup>#</sup> k.okada@spring8.or.jp

### 3. アクセス方法

データの読み出しは、信号属性を保持するパラメータ DB に加え、参照期間によっては複数の場所を参照して組み合わせる手続きが必要なので、利用者にインターフェースを提供している。主に運転 GUI に利用する C/C++ ライブラリ、モニタリングや簡単な解析のための Web cgi、スクリプト言語などの利用を見込んだ RestAPI を提供している。

#### 3.1. C/C++ ライブラリ

データ収集系と対で、データ収納構造を意識させずに利用可能にしている。運転に利用する GUI は基本的にコンパイル言語(C/C++)で静的ライブラリとしてリンクして実行ファイルを作成する。GUI 環境については 4 章で触れる。

#### 3.2. Web cgi

現在値の確認や、期間を指定してトレンドプロット、簡単な相関プロットの作成と表示をウェブ上で提供している。実体は Python スクリプトでデータアクセス系を C/C++ ライブラリとは別に構成している。

歴史的な経緯で利用者の便利な接点として機器の状態変更、通知内容など、パラメータ DB 変更の機能も盛り込んできたため、スクリプトの変更権限は管理者に限定している。その結果、基本的なグラフ作成やデータのテキスト形式でのダウンロードは十分メンテナンスされて誰もが容易に行える一方、一般的な表示以外の要求には、管理者が特別なページを用意するか、利用者がテキストデータを自前で作成するかを選択しかない。

#### 3.3. RestAPI

データアクセス部分だけを切り出して安全に利用者の自由度を上げる目的で Restful API の提供を始めた。メンテナンスの二重化を避けるために C/C++ ライブラリを基に cython ラッパーを構築し Tornado web フレームでサービスを運用している (Fig.2)。これによりウェブスクレイピングなどを使って不安定なモニタリングを行っていた利用者も直接データを自分のサイドに流し込めるようになった。

また加速器調整等で運転条件を頻繁に変えたい場合に、Cython ラッパー (Fig.2 の mdaq\_api) を利用して Python スクリプトを活用するという需要はありそうだが、実行環境のバージョン管理の面での懸念があり、2021 年現在そのチャンネルはまだ開放していない。

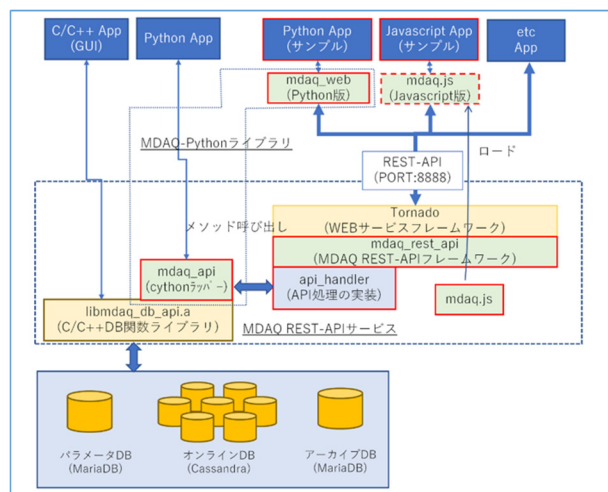


Figure 2: RestAPI frame.

### 4. GUI 環境

#### 4.1. 概要と経緯

加速器運転には、主に C/C++ library を静的リンクした GUI を利用している。デバッグや臨機応変な対応には難があるが、トレードオフとして堅実な運用を目的としている。

GUI の開発環境は SPring-8 立ち上げの際、X-Mate (フジ・データシステム (株)) [7] を選択し、以来いくつものアプリケーション、コーディングのノウハウも蓄積されていたため、OS の入れ替えにも対応させて長年運用してきた。しかし例えばタブ表示方式などもなく、現代のユーザーインターフェースの感覚から時代遅れの感が強くなってきたため、今回オペレーション端末の OS を SLES11 から SLES15 に移行するタイミングで、X-Mate からオープンソース版 Qt5 [8] への乗り替えを進めている。

#### 4.2. Qt プラグインまたはライブラリ

X-mate の開発に慣れた利用者の初期の円滑な移行のために次のいくつかの Qt プラグインまたは関数ライブラリを作成した。

- 子プロセスとのメッセージのやりとり
- 帳票
- グラフプラットフォーム
- 基本図形配置プラグイン
- スレッドセーフプログラミングサンプル

#### 4.3. 移行の手順

新しく入射器を作成した NewSUBARU [9,10] では 2020 年夏から 2021 年春までの長いシャットダウン期間が利用できたため、オペレーション端末として SLES15 のみを導入し、必要な GUI は Qt の GUI に全て移行した。

SPring-8/SACLA については、運転が継続するので、新規 GUI から少しずつ Qt ベースで作成し、しばらく混在させる方針である。

### 5. 信号登録

収集データを管理下に置くために、収集信号の素性

をパラメータDBに登録している。基本的に新しい機器を設置するタイミングで必要なので年に数回の長期停止期間に各機器担当から申請を受け付け、以下の手順で行っている。

- 申請内容の入ったファイルを受付
- 申請内容のチェック。(名前の重複、現実的でない収集周期、意図していないであろう変更など。)
- 仮登録(仮環境への登録)
- 本番環境への登録

これまで一項目毎の申請者とのメールのやり取りを行い、仮登録の手順を異なるネットワークゾーンの本番環境へコピーする手間がかかり、また登録スクリプトの操作にも煩雑な部分があったので、停止期間のたびの登録作業はかなりの量となっていた。

この度、手順を整理し、申請をウェブ上で管理することで作業の簡素化を図った(Fig. 3)。申請者はチェックまでを済ませ登録管理者に連絡する。後の登録手順はウェブ上のボタン操作で可能となる。

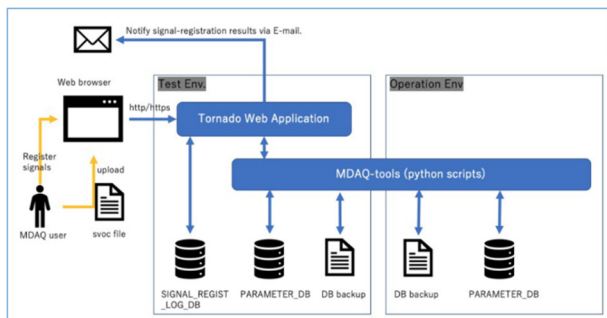


Figure 3: Signal registration frame.

## 6. 画面キャプチャ集

上記で述べた内容の画面キャプチャをいくつか紹介する。

Figure 4,5 は Web cgi のインターフェースを示す。Figure 6 は RestAPI のためのヘルプ画面で、図中左カラムに Python, Javascript のサンプルへのリンクがあり、利用者は自分の環境ですぐにカスタマイズを行うことができる。Figure 7 は信号登録 Web の画面キャプチャで左の画面で申請とチェックを通したところで右の画面の申請リストに追加され、仮登録、本登録の進捗が確認できる。

### SR/XFEL MDAQ(DMZ) ver 1.6.09

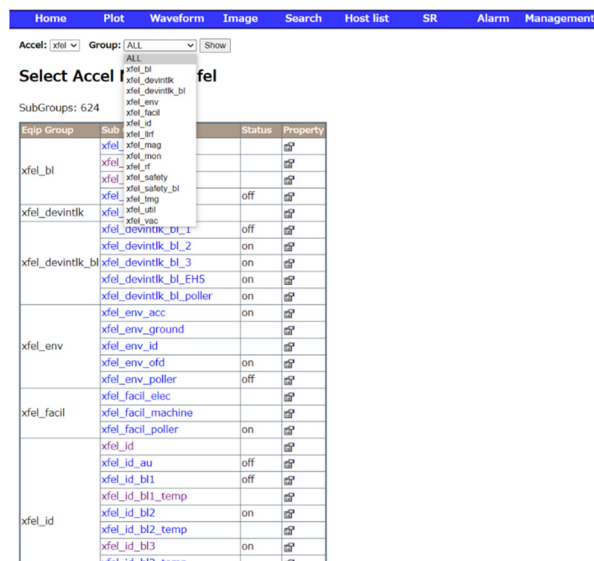


Figure 4: The front page of Web cgi. The step from Group to Sub-group leads to a signal list.

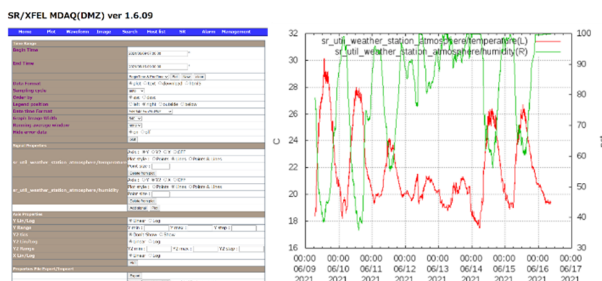


Figure 5: Multi-signal plot example of Web-cgi. The plot can be set to log or X-Y in the page.

MDAQ REST-API ver.1.0.1

Method	Path	Content type	Description
GET	/api/v1/accel	text/json	加速器の情報を取得する。
GET	/api/v1/acceldata	text/json	配列データを取得する。
GET	/api/v1/paramset	text/json	現在のパラメータセットを取得する。
GET	/api/v1/param	text/json	機器の情報を取得する。
GET	/api/v1/name	text/json	機器グループの情報を取得する。
GET	/api/v1/host	text/json	ホスト情報を取得する。
GET	/api/v1/metadata	text/json	画像メタ情報を取得する。
GET	/api/v1/mdaqdata	text/json	MDAQプロセス情報の取得。
GET	/api/v1/scanidata	text/json	指定したPOL/SYNC信号の時系列点データを取得する。
GET	/api/v1/runid	text/json	RUN情報を取得する。
GET	/api/v1/signal	text/json	信号情報を取得する。
GET	/api/v1/statuscurrent	text/json	パンチカレントを取得する。
GET	/api/v1/statusdedicated	text/json	パンチフェーズを取得する。
GET	/api/v1/status	text/json	CODデータを取得する。
GET	/api/v1/statuslist	text/json	CODデータの情報リストを取得する。
GET	/api/v1/statusset	text/json	保存されたパラメータセットを取得する。
GET	/api/v1/subgroup	text/json	機器グループの情報を取得する。
GET	/api/v1/wformmeta	text/json	波形メタ情報を取得する。

Figure 6: The help page of Rest-API. It has links to sample pages of Python and Javascript. Users can start their customization based on the sample.

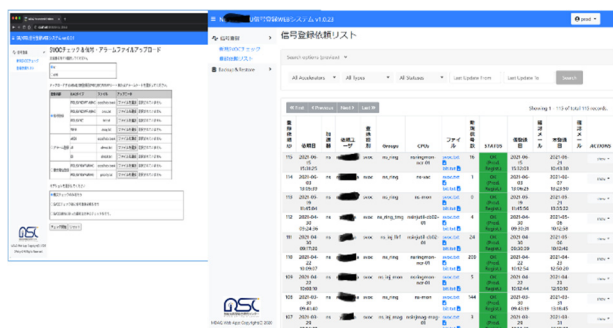


Figure 7: Sample pages from the signal registration web. (Left) A page for uploading request files. (Right) Once the request passed the check, the request queue is created. The manager can proceed the test registration to the real one by clicking the action button.

## 7. 今後の見通し

SPring-8/SACLA の運転は、2021 年に全面的に SACLA 入射に切り替わったため、これまで Linac/Syncro の制御に割り当てていたオペレーション端末の OS を SLES15 に切り替え、Qt GUI 利用環境の使い勝手を高める。RestAPI のサービスは始めたところだが、すでにいくつか利用問い合わせが来ており、展開が期待できる。一方 Web cgi はデータ閲覧とパラメータ設定部分が混在し管理者権限でメンテナンスを続け、RestAPI との住み分けを整理していきたい。

新しく運用を始めた信号登録 web はウェブ上の操作で完結できる一方、これまで申請者の意図を聞き取って細かくケアしていた部分が希薄になるので、今後の運用でミスが起こらないよう特に気を付けていく必要がある。

ところで、このところ定点観測的な測定の枠に収まらず、コンディションを頻繁に変更するような類の測定がいくつかでている。測定条件を記録しておいて、後で条件の切り替わり等を検出してデータを分類する一般的なサービスに昇華できないかと案を練っているところである。

## 8. まとめ

SPring-8/SACLA 及び、同じサイト内の SCSS+、NewSUBARU の制御系では、測定データを中央のデータベースに集約して誰もが加速器の状態把握をすることができる。これは SPring-8 建設当時からの方針を引き継いだ特徴である。SACLA からの SPring-8 入射計画をきっかけとして、データ収集規模の拡大に適応すべくデータベースエンジンの置き換えを行ってきた[11,12]が、今回は利用環境についての更新について報告した。建設の進んでいる東北放射光施設でも同じコンセプトを移植することになっているが、リモートサイトで維持していくにあたって見落としがないか注意深く確認していく。

## 参考文献

- [1] R.Tanaka *et al.*, “The first operation of control system at the SPring-8 storage ring”, Proceedings of ICALEPCS 1997, Beijing, China, (1997).
- [2] SAP ASE, <https://www.sap.com/products/sybase-ase.html>
- [3] Apache Cassandra; <https://cassandra.apache.org>

- [4] T. Fukui *et al.*, “Status of the Control System for the SACLA/SPring-8 Accelerator Complex”, in Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17), Barcelona, Spain, Oct. 2017, pp. 1995-1999.
- [5] K.Okada *et al.*, “Database Scheme for Unified Operation of SACLA/SPring-8”, in Proc. 16th Int. Conf. on Accelerator and Large Experimental Physics Control Systems (ICALEPCS'17), Barcelona, Spain, Oct. 2017, pp.201-205.
- [6] K.Okada *et al.*, “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 回日本加速器学会年会(2020)プロシーディングス THPP23.
- [7] <https://www.fdsnet.co.jp/products/x-mate/>
- [8] <https://www.qt.io/>
- [9] S.Hashimoto *et al.*, “ニュースパル 1.0GeV 新入射器の建設とコミショニング運転“第 18 回日本加速器学会年会(2021).
- [10] S.Hashimoto *et al.*, “ニュースパル放射光施設の現状“第 18 回日本加速器学会年会(2021).
- [11] H. Maesaka *et al.*, “On-Demand Beam Route and RF Parameter Switching System for Time-Sharing of a Linac for X-ray Free-Electron Laser as an Injector to a 4th-Generation Synchrotron Radiation Source”, in Proc. 10th Int. Particle Accelerator Conf. (IPAC'19), Melbourne, Australia, May 2019, pp. 3427-3430.
- [12] T.Fukui *et al.*, “The Design of the Control System for the SACLA/SPring-8 Accelerator Complex to Use the Liinac of SACLA for a Full-Energy Injector of Spring-8”, in Proc. 10th Int. Particle Accelerator Conf. (IPAC'19), Melbourne, Australia, May 2019, pp.2529-2531.