SuperKEKB における pvAccess RPC を利用した Web 上のデータ可視化

WEB-BASED DATA VISUALIZATION USING EPICS PVACCESS RPC AT SUPERKEKB

佐々木信哉[#, A)]
Shinya Sasaki [#, A)]
[A)] High Energy Accelerator Research Organization (KEK)

Abstract

In order to visualize computer infrastructure metrics on the web, Grafana is used at SuperKEKB. As accelerator data is also required to be visualized on Grafana, pvAccess datasource plugin and HTTP / pvAccess API gateway are implemented. This system allows to visualize the data from pvAccess RPC servers on the web. PvAccess is the next generation communication protocol on EPICS 7 and supports the integration of all data into microservices. CSS alarm data and abort timestamps data are provided by pvAccess RPC servers and displayed on Grafana at SuperKEKB. This paper presents advantages of using pvAccess as an interface between services and the data visualization system architecture.

1. はじめに

SuperKEKB[1]では加速器制御に用いている計算機やネットワークスイッチの監視情報を、Zabbix[2]によって収集している。収集したデータは Grafana[3]を利用して可視化しており、監視対象機器の定期的な状況確認や、トラブル時の迅速な状況確認に役立っている。一方で、加速器制御に関わるアーカイブデータやアラームのログデータなどに関しては、それぞれ専用の Web ページを用意して表示、もしくは Web ベースではないソフトウェアで表示を行っている。

これらの加速器制御のデータに関しても Grafana などのソフトウェアを利用して、Web 上で可視化することが出来れば、収集したデータ利用の利便性を向上させることができる。また、専用の Web ページを作成する必要もなくなるため、開発の効率も向上させることができる。そのため、EPICS 7[4]で提供される pvAccess RPC を通じて、各サービスからデータを取得し、Grafana 上でデータを可視化するシステムを構築した。

本稿ではサービス間のインターフェースに pvAccess を利用する利点、構築したシステムの構成とその詳細について報告する。

2. Grafana

Grafana は収集されたデータを、グラフなどによって可視化することを可能とするオープンソースのツールである。ユーザーは Web サーバーとして動作する Grafana に Web ブラウザ経由で接続し、ダッシュボードの作成や閲覧を行うことが出来る。

可視化するデータは、様々なストレージバックエンドから取得することが可能である。Grafana ではそれらのストレージバックエンドをデータソースと呼んでいる。Graphite や InfluxDB などの時系列データベースの他、MySQL や PostgreSQL といったリレーショナルデータベースもデータソースとして公式にサポートしている。

Grafana ではパネルと呼ばれる単位で可視化ブロックを構成し、1 つ以上のパネルが配置されたパネルの集合体をダッシュボードと呼んでいる。パネルにはグラフや表など、複数の種類があり、データの表示方法によって選択することができる。ダッシュボードに配置されたパネルは、それぞれ一つのデータソースと結びついており、パネルごとのクエリ編集画面で、可視化したいデータを取得することができる。

また、Grafana はプラグインによって機能を拡張することが可能である。コミュニティが開発したプラグインを利用することが出来るほか、独自に開発したプラグインによってデータソースやパネルの種類を追加することが出来る。SuperKEKB では Zabbix プラグイン[5]を Grafana に適用し、Zabbix で収集したデータの可視化に利用している。

加速器分野での Grafana の利用も報告されており、加速器や物理実験、計算機インフラなどの監視に利用した例が報告されている[6]。

3. EPICS 7

EPICS 7 は、EPICS V3 と EPICS V4[7]を一つにまとめた EPICS の新メジャーバージョンとして、2017 年 12 月にリリースされた。EPICS 7 では EPICS V3 の機能はそのままに、EPICS V4 で開発された pvData、pvAccess によって、新たなデータ表現と通信プロトコルを利用することが出来るようになった[4]。これにより、従来は I / O Controller (IOC) とレコードを基礎として構成していたシステムの他に、任意の構造化データをネットワーク上でやり取りするサービスを実装することも可能になった。

pvData は任意のデータ構造を定義する機能を提供するモジュールである。pvData によって任意の構造化データを生成することが可能となる[7]。

pvAccess は ChannelAccess[8]に代わる新たな通信プロトコルである。名前解決の方法など、いくつかのコンセプトは ChannelAccess から引き継いでいるが、pvData で生成された任意の構造化データを転送できるようになり、転送効率も改善されている。また、ChannelAccess において利用できた get、put、subscribe (monitor) に加えて、put-get、Remote Procedure Call (RPC) という新たなメッセージングパターンが pvAccess ではサポートされる。

shinya.sasaki@kek.jp

put-get はチャンネルに対して値を書き込んだ後、処理結果を呼び出し元に返す操作である。ChannelAccess においても、put with callback を利用することで、処理が完了したことを通知することは出来たが、その処理結果をレスポンスとして返すことは出来なかった。RPC は put-get と同様の操作であるが、リクエスト毎にやり取りするデータ構造を変更できることが put-get とは異なる[7]。

pvData が任意のデータ構造を定義することを可能にする一方で、EPICS 7 の Normative Types[9]は定義済みのデータ構造を提供する。例えば、スカラー値のためのデータ構造として NTScalar や NTScalarArray がある。基本的なデータ構造として Normative Types を利用することで、汎用的なクライアントを作成することが可能となる。そのため、一般的なアプリケーションは Normative Types を利用してデータを扱うことが推奨されている[10]。

Normative Types は、必須フィールド、オプションフィールド、任意の数の追加フィールドで構成される。必須フィールドとオプションフィールドは、Display Manager や Alarm Handler などの標準的なツールで使用するためのフィールドである。追加フィールドは特定用途のツールのために利用される[11]。

4. サービス間インターフェースとしての pvAccess

ここではサービス間のインターフェースとして pvAccess を利用する利点、欠点について説明する。

EPICS 7 では pvAccess RPC によって、リクエストに応じた任意の構造化データをやり取りすることが出来るようになった。これにより、IOC のリアルタイムデータや集約されたプロセスデータ、種々の構成データなどのデータをマイクロサービスに統合することが可能となった[4] 。例えば、レコード名とそのレコードに関する情報を提供する Directory Service や、マシンパラメータのスナップショットの保存・復元を行う Save Set Service など、実際に pvAccess RPC を利用したサービスが開発され、利用されている[4, 12]。

一方で、サービス間のインターフェースとして、一般的に広く利用されているのは HTTP である。HTTP は Representational State Transfer (REST) 形式のサービスと相性が良く、多くの言語やフレームワーク上で利用することが出来る。また、プロキシキャッシュやロードバランサなど、HTTP をサポートする多くのソフトウェアや機能を利用することもできる。

そのような HTTP と比較して、pvAccess をサービス間インターフェースとして利用する利点の一つは、加速器制御に利用される通信プロトコルと同じプロトコルで、サービスを利用できることである。通信プロトコルが同じであるため、ソフトウェア開発者は加速器制御用ソフトウェアの開発経験を活かしてサービスの開発を行うことが出来る。また、制御用ソフトウェアとの連携も取りやすくなる。例えば、サービスの処理結果を制御用ソフトウェアに反映させるような際に、プロトコルの変換をせずに行うことが出来る。

上記のような利点は、制御用通信プロトコルとして主に ChannelAccess を利用する SuperKEKB のような環境であっても恩恵が受けられる。その理由として、pvAccess が ChannelAccess のいくつかのコンセプトを引き継いでいるため、ChannelAccess を利用するソフトウェアの開発経験が pvAccess においても活かすことが出来ること。EPICS 7 のクライアントは pvAccess と ChannelAccess のどちらも利用することが出来るため、相互の連携も容易に可能であること。ChannelAccess と pvAccess が共に利用可能な CS-Studio[13]のような Display Manager も開発されていることなどが挙げられる。

その他の利点として、EPICS コミュニティが開発する、pvAccess をインターフェースとするサービスとも連携が取りやすいこと、pvAccess の monitor を利用して、イベントドリブンなサービスの連携も可能であることなどが挙げられる。

一方、pvAccess をサービス間インターフェースとして利用する欠点として、次のようなものが挙げられる。

- Grafana など HTTP で通信するサービスとは直接通信できず、API Gateway などを開発して通信する必要がある
- HTTP をサポートする多くのツールや機能を利用できない
- サポートされる言語が少ない
- HTTP と比較して、プロコトルレベルで定義されるメソッドが少ない

上記の内、プロトコルレベルで定義されるメソッドに関して説明する。HTTP では、そのリクエストがリソースに対してどのような処理を行うかが HTTP メソッドによって決まる。このことは HTTP のプロコトルレベルで定義される。それに対して、pvAccess では get や put が定義されているが、HTTP メソッドの POST や DELETE に対応するようなリクエストは定義されていない。そのため、例えば pvAccess においてリソースの削除を行うような処理を行う際には、リソースの取得とは異なる URI を指定せざるをえない。したがって、作成、削除が行われるリソースに対して pvAccess を用いる場合に、URI 設計が難しくなることが予想される。

5. システム構成

5.1 全体構成

Grafana 上で独自のデータソースのデータを可視化するために構築したシステムの全体図を Fig. 1 に示す。な

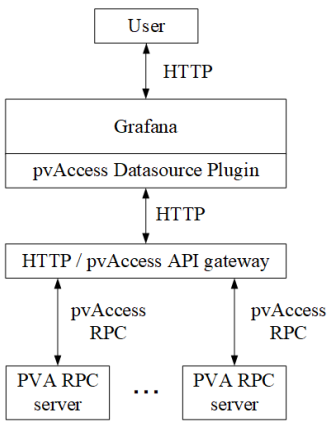

Figure 1: System diagram of data visualization system.

お、ここではユーザー(Web ブラウザ)からの HTTP リクエストが Grafana サーバー経由でデータソースへ送信される Server access mode で動作していることとしている。また、厳密には pvAccess datasource plugin は javascript として Grafana サーバーから Web ブラウザに読み出され実行されるが、ここでは Grafana サーバーの一部として組み込まれていることとしている。

Figure 1 に示す通り、本システムは pvAccess datasource plugin と HTTP / pvAccess API gateway、pvAccess RPC servers の 3 要素からなる。サービス間のインターフェースとして pvAccess RPC を採用した理由は、4 章で述べた、制御用プロトコルと同じプロトコルを利用できる利点や、EPICS コミュニティで開発されるサービスとの連携性の利点があるためである。

各要素間でやり取りされるメッセージの例を Fig. 2 に示す。可視化するデータをユーザーが取得し、表示するまでの流れは次の通りである。

1. Grafana の pvAccess datasouce plugin から、可視化したいデータ取得のリクエストを HTTP / pvAccess API gateway に送信する
2. API gateway が受け取った HTTP リクエストを pvAccess RPC のリクエストに変換して送信する
3. pvAccess RPC server が、リクエストに応じたレスポンスを API gateway に送り返す
4. API gateway が受け取ったレスポンスを HTTP レスポンスに変換し、ユーザーまで送り返す
5. ユーザーが受け取ったレスポンスをもとに、可視化されたデータが Web ブラウザ上に表示される

本システムにおいて、可視化されるデータは pvAccess RPC server によって提供される。そのため、可視化したいデータソースの追加は、所定のインターフェースを持つ pvAccess RPC server を追加することで可能となる。

5.2 pvAccess datasource plugin

pvAccess datasource plugin は pvAccess RPC server のデータを Grafana 上で表示するためのデータソース plugin である。Grafana が公式に提供しているサンプルプラグインである Simple JSON Datasource[14]を参考にして、General pvAccess Datasource[15]として開発を行った。

開発したプラグインの特徴は次の通りである。

- データソースの設定画面から RPC のために必要なチャンネル名や引数名などが設定可能
- RPC の引数は任意の数と名前が設定可能
- 時系列データと表データの 2 種類を利用可能

General pvAccess Datasource はデータ取得のために必要となる情報を HTTP で送信し、データの取得を行う。Figure 2 でも示される通り、HTTP で送信する情報は主に、取得するデータの期間や取得したいデータごとのパラメータ、pvAccess RPC のチャンネル名や引数名である。

5.3 HTTP / pvAccess API gateway

HTTP / pvAccess API gateway は、Grafana から送信される HTTP リクエストを pvAccess RPC のリクエストに変換し、Grafana と pvAccess RPC server を中継する役割を担う。gfhttpva[16]として開発した API gateway は General pvAccess Datasource と共に使用することを前提としている。

gfhttpva は Python で実装した HTTP サーバーである。Web フレームワークとして Flask[17]を使用している。また、Python から pvAccess を使用するために、pvAccess の Python API を提供する pvaPy[18]を使用している。

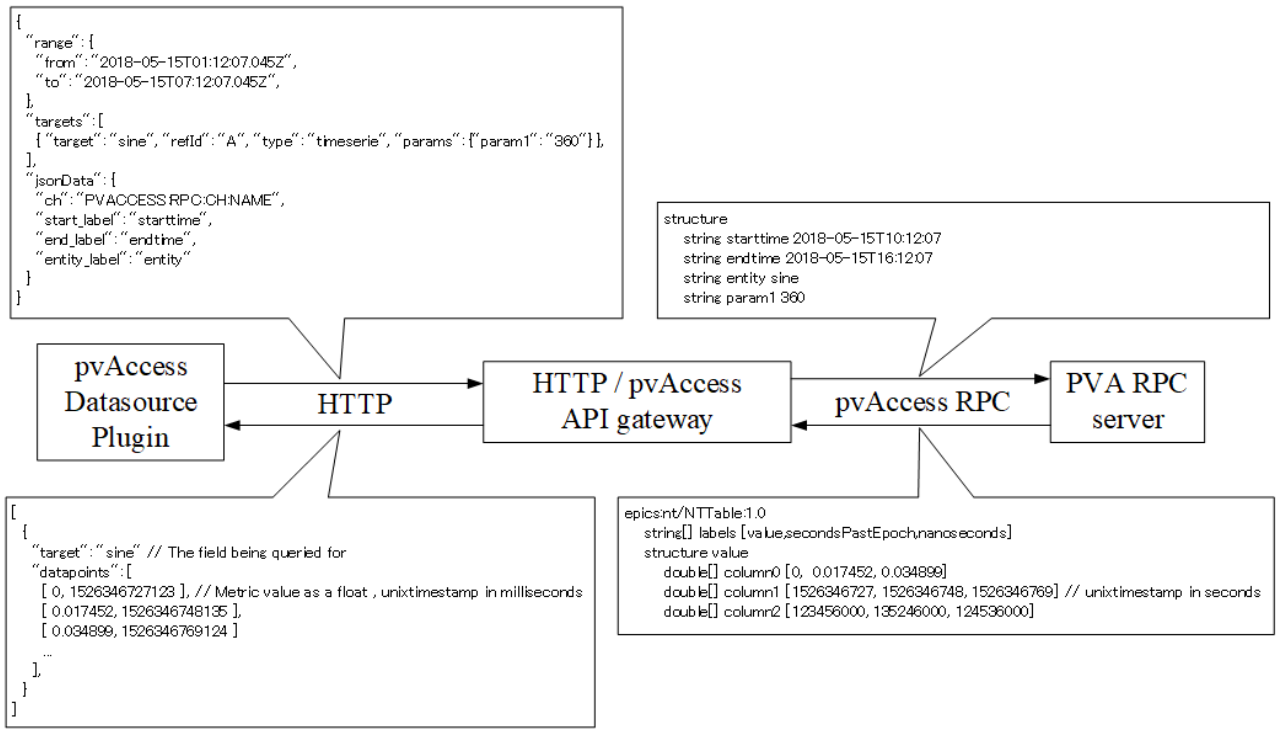

Figure 2: Message flow in data visualization system.

5.4 pvAccess RPC server

本システムでは pvAccess RPC server が可視化するデータを提供する。RPC の引数やその戻り値は API gateway である gfhttpva によって定められる。

gfhttpva は RPC server に対して、任意の数、名前の引数を利用することが出来るが、以下の 3 つに対応する引数は必ず RPC server に渡される。ただし、引数名は任意のものを使用できる。

- starttime: データ取得範囲の開始時間
- endtime: データ取得範囲の終了時間
- entity: どのデータを取得するかを示す文字列

RPC の戻り値は、そのデータが時系列データであっても、表データであっても Normative type の NTTable を使用する。戻り値が時系列データであれば、value、secondsPastEpoch、nanoseconds の 3 つの列をもつ NTTable を返す必要がある。戻り値が表データであれば任意の数と名前の列を NTTable に含めることが出来る。

6. 開発した pvAccess RPC server

6.1 pvAccess RPC servers の実装

開発した 3 つの pvAccess RPC server は、いずれも Python で実装を行った。また、pvAccess の API として pvaPy を使用している。

6.2 pvAccess RPC サンプルデータソース

pvAccess RPC サンプルデータソースは時系列データを返す RPC server のサンプルとして開発した[19]。

この RPC server は starttime、endtime として指定された時間範囲に対して、entity で指定されたデータを返す。また、param1 という引数で、取得するデータのパラメータを指定できる。

例えば、entity に sine、param1 に 360 を指定すると、指定された時間範囲で 0 度から 360 度まで変化する正弦波のデータが返される。Grafana 上に表示された pvAccess RPC サンプルデータのスクリーンショットを Fig. 3 に示す。

Figure 3: Screen shot of pvAccess RPC sample data. Entities are set to sine and cos.

6.3 CSS アラームデータソース

CSS アラームデータソースは、SuperKEKB で運用している CSS アラーム[20]の現在情報や履歴を取得するための RPC server として開発した[21]。この RPC server では時系列データは扱わず、表データのみを返却する。RPC のリクエストを受けると、情報が保存されている PostgreSQL サーバーに対して SQL を発行してデータを取得し、呼び出し元に返却する。

アラームの現在情報を取得する際は、引数の時間範囲は使用せず、entity のみを使用する。entity はアラームの登録グループでデータを絞る込むために使用される。

履歴情報の取得では、与えられた時間範囲で発生したアラームの状態変化を取得できる。entity は現在情報の取得と同様に、登録グループでデータを絞るこむために使用される。また、message 引数によってアラームメッセージの内容でデータを絞り込むことも可能である。

Grafana 上に表示されたアラームの現在情報のスクリーンショットを Fig. 4 に示す。

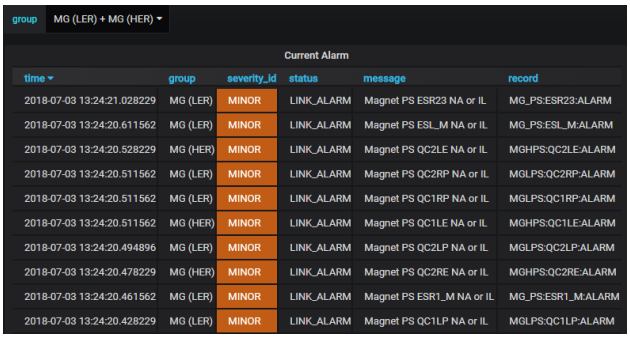

Figure 4: Screen shot of current CSS alarm data. Displayed groups are MG (LER) and MG (HER).

6.4 アボートタイムスタンプデータソース

アボートタイムスタンプデータソースは SuperKEKB のアボートタイムスタンプ[22]を表示するために開発した。この RPC server でも時系列データは扱わず、表データのみを返却する。RPC のリクエストを受けると、ファイルに保存されたタイムスタンプ情報から、必要となる情報を取得し、呼び出し元に返却する。LER、HER どちらのリングの情報を取得するのかどうかは、引数の entity によって決定される。

Grafana 上に表示されたアボートタイムスタンプデータのスクリーンショットを Fig. 5 に示す。

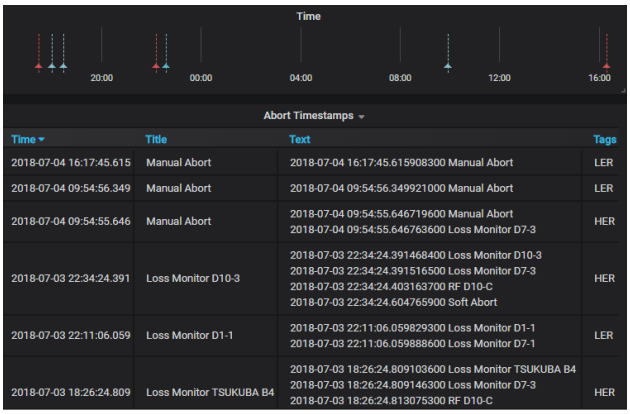

Figure 5: Screen shot of abort timestamps data.

7. まとめ

Grafana の Datasource plugin として General pvAccess Datasource を、HTTP / pvAccess API gateway として gfhttpva を開発し、pvAccess RPC server が提供するデータを Grafana 上で可視化するためのシステムを構築した。これにより、pvAccess RPC server を追加することで、Grafana から利用できるデータを追加することが可能となった。また、サービス間のインターフェースとして pvAccess を採用したことにより、Grafana からの利用のみならず、加速器制御用ソフトウェアからも容易にサービスを利用することが可能となった。

SuperKEKB では CSS アラーム、アボートタイムスタンプの情報を提供する pvAccess RPC server を開発し、運用している。今後は、アーカイブデータなどのデータを提供する pvAccess RPC server を整備することを検討する。

参考文献

[1] Y. Ohnishi *et al*., "Accelerator design at SuperKEKB", Prog.Theor. Exp. Phys. (2013) 03A011; http://ptep.oxfordjournals.org/content/2013/3/03A011.full.pdf
[2] https://www.zabbix.com
[3] https://grafana.com
[4] L.R. Dalesio *et al*., "EPICS 7 Provides Major Enhancements to the EPICS Toolkit", in Proc. ICALEPCS'17, Barcelona, Spain, Oct. 2017, paper MOBPL01; https://doi.org/10.18429/JACoW-ICALEPCS2017-MOBPL01
[5] Zabbix plugin for Grafana; https://grafana.com/plugins/alexanderzobnin-zabbix-app
[6] Grafana at CERN; https://grafana.com/files/grafanacon_eu_2018/Borja_GrafanaCon_EU_2018.pdf
[7] T. Korhonen *et al*., " EPICS Version 4 Progress Report", in Proc. ICALEPCS'13, San Francisco, CA, USA, Oct. 2013, TUCOCB04; http://accelconf.web.cern.ch/AccelConf/ICALEPCS2013/papers/tucocb04.pdf
[8] Channel Access Protocol Specification; https://epics.anl.gov/base/R3-16/1-docs/CAproto/index.html
[9] EPICS V4 Normative Types; http://epics-pvdata.sourceforge.net/alpha/normativeTypes/normativeTypes.html
[10] EPICS V4 Overview and Architectures; http://epics-pvdata.sourceforge.net/arch.html
[11] EPICS normativeTypesCPP; http://epics-pvdata.sourceforge.net/docbuild/normativeTypesCPP/tip/documentation/ntCPP.html
[12] G. Shen *et al*., "NSLS II Middlelayer Services", in Proc. ICALEPCS'13, San Francisco, CA, USA, Oct. 2013, MOPPC155 ; http://accelconf.web.cern.ch/AccelConf/ICALEPCS2013/papers/moppc155.pdf
[13] http://controlsystemstudio.org
[14] Simple JSON Datasource; https://github.com/grafana/simple-json-datasource
[15] General pvAccess Datasource; https://github.com/sasaki77/generalpvaccess-datasource
[16] gfhttpva - grafana http / pvAccess API gateway; https://github.com/sasaki77/gfhttpva
[17] http://flask.pocoo.org
[18] S. Veseli, "PvaPy: Python API for EPICS PV Access", in Proc. ICALEPCS'15, Melbourne, Australia, October 2015, paper WEPGF116; http://jacow.org/icalepcs2015/papers/wepgf116.pdf
[19] pvAccess RPC sample for gfhttpva; https://github.com/sasaki77/grafana-pvarpc-sample
[20] T. Nakamura *et al*., "Operation Status of CSS Alarm System for SuperKEKB", in Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan, August 8-10, 2016, Chiba, Japan, TUP095; http://www.pasj.jp/web_publish/pasj2016/proceedings/PDF/TUP0/TUP095.pdf
[21] cssalmrpc; https://github.com/sasaki77/cssalmrpc
[22] S. Sasaki *et al*., "Development of time stamp recording system for SuperKEKB abort trigger system", in Proceedings of the 14th Annual Meeting of Particle Accelerator Society of Japan, August 1-3, 2017, Sapporo, Japan, TUP096; http://www.pasj.jp/web_publish/pasj2017/proceedings/PDF/TUP0/TUP096.pdf