

## SuperKEKB 加速器の制御システム CONTROL SYSTEM OF SUPERKEKB ACCELERATOR

杉村仁志 \*<sup>A</sup>、中村達郎 <sup>A</sup>、梶裕志 <sup>A</sup>、佐々木信哉 <sup>A</sup>、小田切淳一 <sup>A</sup>、秋山篤美 <sup>A</sup>、内藤孝 <sup>A</sup>、  
中村卓也 <sup>B</sup>、吉井兼治 <sup>B</sup>、芳藤直樹 <sup>C</sup>、飯塚祐一 <sup>C</sup>、廣瀬雅哉 <sup>D</sup>、浅野和哉 <sup>D</sup>

Hitoshi Sugimura\*<sup>A</sup>, Tatsuro Nakamura<sup>A</sup>, Hiroshi Kaji<sup>A</sup>, Shinya Sasaki<sup>A</sup>, Jun-ichi Odagiri<sup>A</sup>, Atsuyoshi Akiyama<sup>A</sup>,  
Takashi Naito<sup>A</sup>, Takuya Nakamura<sup>B</sup>, Kenji Yoshii<sup>B</sup>, Naoki Yoshifuji<sup>C</sup>, Yuuichi Iitsuka<sup>C</sup>,  
Masaya Hirose<sup>D</sup>, Kazuya Asano<sup>D</sup>

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

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

<sup>C</sup>East Japan Institute of Technology Co., Ltd.

<sup>D</sup>Kanto Information Service Co., Ltd.

### Abstract

The SuperKEKB accelerator inherits the control system of its predecessor KEKB accelerator. It uses a network distributed control system using EPICS. The operation interface uses script language such as Python, SAD. CSS is also used in each operation consoles. Control network has been redundantly equipped with 10 GbE and 1 GbE. IT infrastructures and EPICS processes are monitored by Zabbix and Elastic Stack, and visualization is performed using tools such as Grafana and Kibana. Data archive system is also being developed. In addition to KEKBLLog, the Archiver Appliance, which has been developed recently, is introduced as new archive system. Furthermore, the timing system is made it possible to handle quick timing calculation of bucket selection and the timing is transmitted by using 8-bit event code in optical communication.

### 1. はじめに

SuperKEKB 加速器は前身の KEKB 加速器の制御系を継承し、EPICS [1] を用いたネットワーク分散型の制御システムを利用している [2]。主に VME や PLC などといったフロントエンド機器を加速器内各所に設置し、IOC として用いることで制御対象の機器を制御する。一方で運転パネル (OPI) は Python や SAD [3] といったスクリプト言語を用いて、サーバー計算機上でプログラムを実行したり、CSS を用いて運転端末から操作するような運用を行っている。

制御ネットワークは 10 GbE と 1 GbE を冗長的に整備し、IT インフラの監視や EPICS 制御システムの監視を Zabbix、Elastic Stack で行い、Grafana、Kibana などのツールを利用して可視化を行っている。

データの保存には独自に開発を行った KEKBLLog [4] システムを主軸とし、専用の Cache Server を構築することで読み出しの高速化を図った [5]。他にも CSS アーカイブなども利用している。また、2018 年度からは EPICS コミュニティに普及が広がっている ArchiverAppliance を導入し、試験評価を進めている。

タイミング制御ではイベントタイミングシステムを用い、タイミングに識別子 (イベントコード) を付けることで数多くの設定パラメータを用いて運転している。SuperKEKB 主リングに入射するバケットを選択し、繰り返し 50 Hz でそのバケットへの入射タイミングを即座に計算を行う手法をとっている。これら一連の制御のためにメインリングと入射器との間でリフレクティブメモリ用のネットワークを敷設し、メモリの共有化を行うことで高速のデータ授受

を可能にした。

### 2. サーバー計算機

#### 2.1 制御用サーバー計算機

制御用サーバー計算機的主要な役割は、IOC ソフトウェアの開発・管理と、上位層アプリケーションの実行である。1 台を除いて全て Linux を採用している。

- 第一世代サーバー：初期のサーバーの OS は、HP-UX であったが、その後主力は Linux に移行した。KEKB 時代に開発された古いアプリケーションの実行環境として、2 台のサーバー計算機 (Linux, HP-UX) が現役で使われている。
- 第二世代サーバー：現在はブレード・サーバーが主力であり、14 台の計算機が稼働している。計算機ごとに稼働させるサービスやアプリケーションをほぼ固定的に割り当てた運用を行っている。
- 第三世代サーバー：近年導入のサーバーは、ブレードの追加ではなく、ラックマウントの単体サーバーを設置している。
- 専用サーバー：このほかに試験用、テープバックアップ用などの専用用途の計算機も存在する。また、運転時のデータ収集を行うアーカイブ・ソフトウェアの運用のため、専用の計算機及びストレージも導入している。

#### 2.2 運転用サーバー計算機

制御用のサーバー計算機とは別に、SAD Cluster と呼ばれる一群の計算機 (SAD 計算機) も運転アプリケーション (主として SAD Script) の実行のために使われている。元来 SAD Cluster は加速器のモデ

\* hitoshi.sugimura@kek.jp

リング計算のためのオフライン計算機であったが、KEKB 加速器の運転開始と共に運転用アプリケーションの実行にも使われるようになった。歴史的には、HP-UX, Alpha/Tru64 UNIX, Linux, Macintosh という多種のプラットフォームが使われて来たが、現在は全て FreeBSD で構成され、運転用 4 台、オフライン専用 4 台が稼働している。SAD Cluster では NAS (SAD-NAS) を設置し、NFS でファイル・システムを共有している。SAD-NAS は制御用サーバー計算機群にもマウントしており、運転用・制御用の共通のメインストレージとなっている。

### 2.3 コンソール

中央制御室の運転用中央コンソールは、X 端末を採用している。現在は PC/Linux が使われているが、過去には、専用 X 端末、X エミュレーターを載せた PC/Windows や Macintosh, Macintosh/OSX など使われ、多様な変遷を辿って来ている。中央コンソールは基本的には X 端末として利用されるが、負荷分散のため、CSS など一部のアプリケーションはコンソール上でも実行される。中央制御室には運転用中央コンソールのほか、各グループが各種の専用端末を設置している。ローカル制御室にも開発・保守のため Mac Mini を X 端末として 1 台ずつ置いている。ローカル制御室のコンソールは、集中的な管理ができるよう、ネットワーク・ブートで起動するようになっている [6]。

## 3. ネットワーク

SuperKEKB 加速器制御ネットワークは L3 スイッチであるコアスイッチを中心にして構成される [7]。コアスイッチは Virtual Switching System (VSS) により 2 台の L3 スイッチが仮想的に 1 台のスイッチとして動作するため、片方の系統に障害が発生してもネットワークを構成し続けることが出来るようにしている。このコアスイッチに対して 39 台の L2 スイッチが接続する。必要に応じて、39 台の L2 スイッチの配下に更に L2 スイッチを接続して使用している。コアスイッチと L2 スイッチは 2 ポートで接続されており、2 つのポートは 2 台の L3 スイッチにそれぞれ別に接続する。接続に使用する 2 つのポートは、片方が 10 GbE、もう片方が 1 GbE の組み合わせと、両方とも 1 GbE の組み合わせがある。10 GbE と 1 GbE の組み合わせに対してはスパニングツリープロトコルで 1 GbE のポートが通常はブロックされることで冗長化を図っている。一方、両方とも 1 GbE の場合はリンクアグリゲーションを使用することで冗長化している。

Figure 1 に SuperKEKB 制御ネットワークの概念図を示す。SuperKEKB 加速器制御ネットワークと KEK オフィスネットワーク (機構内ネットワーク) はファイアウォールによって分離され、それぞれ独立に運用されている。ただし、一部の NAS に関してのみ両方のネットワークに接続して運用している。

加速器建設、および加速器メンテナンス時の利用を目的として、無線 LAN システムが整備されてい

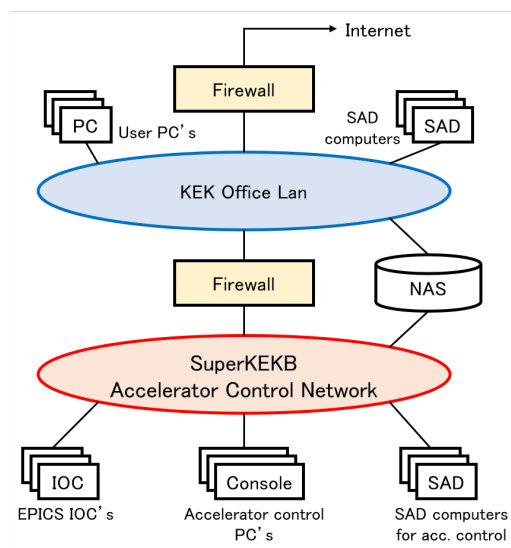


Figure 1: Schematic view of SuperKEKB network.

る。周長 3 km の SuperKEKB 主リングトンネル、入射器トンネル、および PF-AR 加速器トンネル内全域において無線 LAN を使用することが出来る。主リングアーク部には漏洩同軸ケーブル (LCX) アンテナを使用している。LCX アンテナは、20D 型、長さ 125 m、結合損失 65 dB、伝送損失 9 (dB/100 m) のものを 16 本、合計 2000 m 相当使用している。主リングアーク部には、電源供給設備がほとんどないため、地上部電源棟に設置された PoE モデムから無線 LAN 機器へ給電を行っている。主リング直線部には 16 本、入射器トンネルには 10 本のコリニアアンテナ (6 dBi) を設置している。また、トンネル内の放射線レベルを考慮して、LCX アンテナおよびコリニアアンテナは、1 MGy 以上の放射線耐性を有する物を選定した。無線 LAN システムを構成する合計 70 台のアクセスポイントは、1 台のアクセスポイントコントローラーで、管理を行っている。

## 4. 監視システム

SuperKEKB では大きく分けて 2 つの監視システムを利用している。一つは Zabbix を用いたメトリクス監視、もう一つは Elastic Stack を用いたログ監視である。メトリクス監視では、Zabbix を利用してメトリクスの収集および障害発生時のアラートの送信を行っている。2019 年 7 月現在、23 台の計算機と、88 台のネットワークスイッチに対して監視を行っている。収集したメトリクスはオープンソースのツールである Grafana 上で可視化している。また、EPICS PV 値をメトリクスとして Zabbix に送信する CA クライアントアプリケーションを開発し、IT インフラのメトリクスと同様に EPICS PV 値を監視することを可能とした。SuperKEKB では IOC の CPU 使用率やメモリ使用率、CA クライアントの接続数などを監視するために、このアプリケーションを利用している。Figure 2 が web 上で表示される Zabbix を用いたサーバーの負荷を示したスナップショットである。

ログ監視では Elastic Stack のプロダクト群である Logstash、Elasticsearch、Kibana を使用して、ログの収集・分析・可視化を行っている。SuperKEKB では監視対象の計算機上で 10 分ごとに実行した ps コマンドの結果を監視しているほか、caSnooper コマンドの出力を利用した CA サーチの頻度の監視、casw コマンドの出力を利用した CA Beacon Anomaly の監視を行っている。また、Wireshark とそのプラグインである cashark を利用して、SuperKEKB 加速器制御ネットワーク内のブロードキャストパケットの監視も Elastic Stack で行っている。Wireshark で CA プロトコルを解析するためのプラグインである cashark を利用しているため、ネットワーク上でどのような CA のパケットがやり取りされているかを確認することが可能となっている。

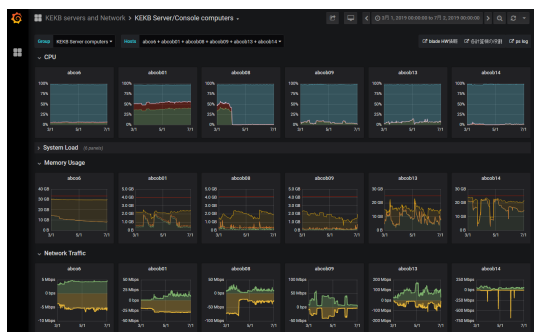


Figure 2: Snapshot of monitoring of server load with Zabbix.

## 5. IOC

IOC は主に VME ベースのものとして PLC ベースのものを多用している (Fig. 3, 4)。VME は KEKB 加速器の頃から使われてきており、CPU ボードには主に MVME5500 を使用し、OS には VxWorks を採用している。PLC ベースの IOC には、横河電機製 FA-M3 シリーズの PLC を採用し、CPU には Linux を載せられる F3RP61 を使用して EPICS を組み込んでいる。PLC の構成として二つの方式があり、ラダープログラム用の CPU を別に持ち、それと通信することで EPICS から間接的に制御を行う方式の IOC がある。また、処理能力を向上させた CPU である F3RP71 を近年試用しており、一部では CPU の置き換えを始めている。このほかに LLRF 用に開発した  $\mu$ TCA ベースの IOC も使用している。組込みシステムとして、Windows を組込んだオシロスコープに EPICS を搭載したものを KEKB 加速器から引き続き使用している。Windows ベースの IOC を採用する場合、OS のサポート期間が短いという難点があり、特に組込み機器で使う場合に OS の更新が困難な場合も多く、これらの保守は今後の課題と言える。



Figure 3: Typical example of VME IOC.

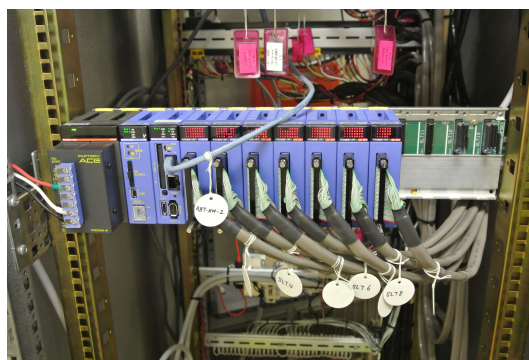


Figure 4: Typical example of PLC IOC.

## 6. フィールドバス

IOC と制御対象の機器とを結ぶフィールドバスは Table 1 のように多様な種類を使っている。ARCNET

Table 1: Fieldbus Used in SuperKEKB

フィールドバス	主な使用場所
ARCNET	電磁石電源
VXI-MXI	BPM
CAMAC	RF
GPIB	随所
RS-232C	随所
Ethernet	随所

は KEKB 加速器で多数導入された小型電源に対し低コストのインターフェースを実現するために導入された。SuperKEKB 加速器では ARCNET の VME インターフェイスボードの更新を行ない、電源側の組込みコントローラーやハブの更新を順次進めている [8]。かつては GPIB や RS-232C で接続されていた計測・制御機器も近年では Ethernet インターフェースを備えるのが標準的になっており、機器の更新につれて Ethernet の利用が増えて行っている。CAMAC のハードウェアは TRISTAN 加速器以来 30 年以上も使用されて来ているが、現在は RF 制御にのみ残されている。これも LLRF システムの更新に伴う形

で徐々に減らしていく予定であるが、まだしばらくは維持が必要なため、外注で行っていた CAMAC モジュールの定期保守の内製化を進めている。

## 7. 上位層アプリケーション

SuperKEKB 加速器で特徴的なのは、スクリプト言語を用いたアプリケーション開発が盛んに行なわれていることである。最も多く使われているのは SAD に組み込まれているスクリプト言語である SAD Script である。これに加えて Python も多く使われている。いずれの言語も EPICS の標準通信プロトコルである Channel Access をサポートしており、GUI 作成には Tkinter を利用することで制御・運転のためのプログラム開発が可能になっている。ルミノシティ・フロンティアを目指す SupreKEKB 加速器では、運転ツールの絶え間ない改良が不可欠であり、インタープリタで動くスクリプト言語を用いることで迅速な開発を可能にしている。スクリプト言語以外では、EPICS の標準的なツールもいくつか採用している。CSS は広く使われており、特に BOY によって作られた操作パネルは多用されている。SuperKEKB 加速器のアラーム・システムは、CSS をベースに GUI 部分を Python で独自に開発したものを使っている。データ・アーカイブ・システムは、KEKB 加速器で独自開発した KEKBLog を引き続き採用している。KEKBLog の弱点だったデータ読み出し速度については、キャッシュサーバーの構築により大きく向上した。また、一部のサブシステムでは CSS Archiver も使われている。近年普及を見せている Archiver Appliance の試用も始めており、CSS Archiver の後継として期待される。運転ログは電子化されている。KEKB 加速器では Zope をベースに Zlog [9] と呼ばれる運転ログシステムを開発したが、SuperKEKB 加速器でも引き続き採用している。制御システムと連携させることでデータの自動入力などの支援機能があり、効率的なオペレーションを支えている。

## 8. アボートトリガーシステム

アボート要求信号を集約してアボートキッカーへトリガーを送信する役割を持つアボートトリガーシステムは、SuperKEKB のために新システムに更新された [10]。Figure 5 はアボートトリガーシステムの概念図である。2019 年 7 月現在、164 のアボート要求信号をこのシステムで収集している。システムは、要求信号を電気信号から光信号に変換する部分と、光信号を集約する部分とに分けられる。電気信号 (TTL もしくは接点出力) として出力されるアボート要求信号は光信号に変換されて集約される。要求信号を出力する機器に近いところで光信号に変換することで、信号へのノイズを減らし、高速かつ安定に信号を収集することが可能となっている。光信号に変換された要求信号は、8 入力 1 出力の集約モジュールによって集約される。集約モジュールは数珠つなぎに接続されていて、最終的に信号が 1 つに集約される。アボート要求信号は加速器リングに沿って点在する 8 箇所のローカル制御室で部分的に集約されて、

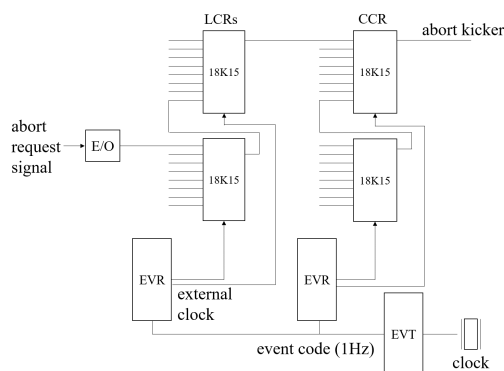


Figure 5: Schematic view of abort trigger system.

その後中央制御室で再び集約される。ローカル制御室から中央制御室までの最長距離は約 2 km であり、その伝送時間は約 10  $\mu$ s である。システムの応答時間はこの伝送時間がほとんどを占めている。このシステムはタイムスタンプ機能を備えており、それぞれの集約モジュールがアボート要求信号を受信した時間を 100 ns の分解能で記録することが可能となっている [11]。記録されたタイムスタンプは、アボートが発生した原因や過程を究明することに役立っている。

## 9. イベントタイミングシステム

SuperKEKB 入射器はビームパルスを 50 Hz の頻度で 4 リング (SuperKEKB (HER/LER), PF, PF-AR) に供給している。各リングへは異なるエネルギー、電荷量をもつ電子や陽電子を供給するため、入射器は入射モードごとに独立の制御パラメータセットを持ち、これを 50 Hz の頻度で切り替える運転をしている。このような運転のために、SuperKEKB 加速器ではどのパラメータセットを用いるかをイベントコードを用いて識別するシステムを導入した。このシステムはイベントコードを RF クロックに同期させて各制御機器に送信し、クロックに同期したタイミングで信号を出力することが可能であり、イベントタイミングシステムと呼んでいる [12]。次にどのビームモードの入射を行うかの設定をイベント送信モジュール「Event Generator (EVG)」から行い、入射器やダンピングリング、メインリング各所に設置されたイベント受信モジュール「Event Receiver (EVR)」を用いてイベントコードに応じたタイミングを出力する。イベント送信のタイミングはメインリングのどの RF バケットに入射するかで設定される。メインリングの RF バケット数は 5120 あるため、電子入射では 5120 のタイミングパターンが用いられる。陽電子入射はダンピングリングを通じて行われるため、ダンピングリングの RF バケット数 230 も考慮し、全部で 117760 (5120 $\times$ 23) になる。この中から 20 ms $\pm$ 1 ms の範囲を満たすタイミングを選び、イベントを発生させている [13]。Figure 6 にタイミングシステムの概要図を示す。イベントタイミングは入射器に設置したメインタイミングステーションから入

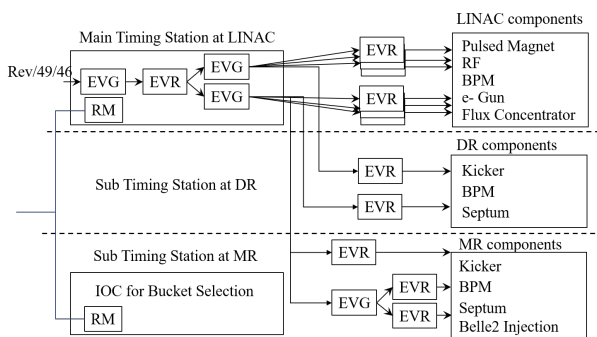


Figure 6: Schematic view of timing system.

射器、ダンピングリング、メインリングのそれぞれの制御機器にタイミング信号を分配する。さらに、入射バケットの選択とタイミングの計算のためにメインリングに専用のIOCを設け、計算した値を分散共有メモリ(レフレクティブメモリ)を介してメインタイミングステーションに送るというシステムを組んでいる。これにより、メインタイミングステーションでの負荷を抑え、50 Hz ごとのタイミング値の更新を素早く行うことができるようにしている。

また、イベントタイミングシステムではタイミング以外にもビームゲート信号を送ったり、ショット番号、ビームモード、RF位相などの情報も送ることができる。これらを1本の光ファイバーを利用して通信できるため、制御機器間でのやりとりを簡素化させることができた。

## 10. まとめ

SuperKEKB 加速器の制御システムはKEKB 加速器の制御システムを継承しながら、アボートトリガーシステムの実装、サーバー計算機の管理および監視、ネットワーク監視、ソフトウェアの整備、新規アーカイブシステムの導入、タイミングシステムの多様化など、様々なシステムを改良したり、新たなシステムの導入も行うことで安定的な長期運転を目指している。

## 参考文献

- [1] <https://epics-controls.org/>
- [2] M. Iwasaki *et al.*, “The Construction Status of the SuperKEKB Control System”, in Proc ICALEPCS’15, Melbourne, Australia, October 2015, paper MOB3O04; <https://accelconf.web.cern.ch/AccelConf/ICALEPCS2015/papers/mob3o04.pdf>
- [3] <http://acc-physics.kek.jp/SAD/>
- [4] T.T. Nakamura *et al.*, “Data Archiving System in KEKB Accelerators Control System”, in Proc ICALEPCS’05, Geneva, October 2005, paper PO1.077-7;

- [http://accelconf.web.cern.ch/AccelConf/ica05/proceedings/pdf/P1\\_077.pdf](http://accelconf.web.cern.ch/AccelConf/ica05/proceedings/pdf/P1_077.pdf)
- [5] A. Morita, “SuperKEKB におけるデータアーカイブの読み出し高速化”, Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan, Chiba, Japan, August 8–10, 2016, pp. 1150–1153; [https://www.pasj.jp/web\\_publish/pasj2016/proceedings/PDF/TUP0/TUP093.pdf](https://www.pasj.jp/web_publish/pasj2016/proceedings/PDF/TUP0/TUP093.pdf)
- [6] T. Aoyama *et al.*, “KEKB サーバー計算機へのネットワークブートと集中制御の応用”, Proceedings of the 5th Annual Meeting of Particle Accelerator Society of Japan, Higashi-hiroshima, Japan, August 6–8, 2008, pp. 640–642; [https://www.pasj.jp/web\\_publish/pasj5\\_lam33/contents/PDF/TP/TP010.pdf](https://www.pasj.jp/web_publish/pasj5_lam33/contents/PDF/TP/TP010.pdf)
- [7] M. Iwasaki *et al.*, “Design and Status of the SuperKEKB Accelerator Control Network System”, Proceedings of the 11th Annual Meeting of Particle Accelerator Society of Japan, Aomori, Japan, August 9–11, 2014, pp. 759–763; [https://www.pasj.jp/web\\_publish/pasj2014/proceedings/PDF/SAP0/SAP093.pdf](https://www.pasj.jp/web_publish/pasj2014/proceedings/PDF/SAP0/SAP093.pdf)
- [8] T.T. Nakamura *et al.*, “SuperKEKB のための電磁石電源制御システムの改造”, Proceedings of the 13th Annual Meeting of Particle Accelerator Society of Japan, Chiba, Japan, August 8–10, 2016, pp. 1154–1158; [https://www.pasj.jp/web\\_publish/pasj2016/proceedings/PDF/TUP0/TUP094.pdf](https://www.pasj.jp/web_publish/pasj2016/proceedings/PDF/TUP0/TUP094.pdf)
- [9] K. Yoshii *et al.*, “Web-Based Electronic Operation Log System – Zlog System”, in Proc ICALEPCS’07, Knoxville, October 2007, paper WOAB04; <http://accelconf.web.cern.ch/AccelConf/ica07/PAPERS/WOAB04.PDF>
- [10] S. Sasaki *et al.*, “Upgrade of Abort Trigger System for SuperKEKB”, in Proc. ICALEPCS’15, Melbourne, Australia, October 2015, paper MOPGF141; <https://accelconf.web.cern.ch/AccelConf/ICALEPCS2015/papers/mopgf141.pdf>
- [11] S. Sasaki *et al.*, “Development of time stamp recording system for SuperKEKB abort trigger system”, Proceedings of the 14th Annual Meeting of Particle Accelerator Society of Japan, Sapporo, Japan, August 1–3, 2017, pp. 610–612; [https://www.pasj.jp/web\\_publish/pasj2017/proceedings/PDF/TUP0/TUP096.pdf](https://www.pasj.jp/web_publish/pasj2017/proceedings/PDF/TUP0/TUP096.pdf)
- [12] H. Kaji *et al.*, “Installation and Commissioning of New Event Timing System for SuperKEKB”, Proceedings of the 12th Annual Meeting of Particle Accelerator Society of Japan, Fukui, Japan, August 5–7, 2015, pp. 223–227; [https://www.pasj.jp/web\\_publish/pasj2015/proceedings/PDF/FROL/FROL15.pdf](https://www.pasj.jp/web_publish/pasj2015/proceedings/PDF/FROL/FROL15.pdf)
- [13] H. Kaji *et al.*, “Bucket Selection system for SuperKEKB”, Proceedings of the 12th Annual Meeting of Particle Accelerator Society of Japan, Fukui, Japan, August 5–7, 2015, pp. 1278–1281; [https://www.pasj.jp/web\\_publish/pasj2015/proceedings/PDF/THP1/THP100.pdf](https://www.pasj.jp/web_publish/pasj2015/proceedings/PDF/THP1/THP100.pdf)