SuperKEKB へのCSSアラームシステム導入における性能評価

EVALUATION OF THE CSS BASED ALARM SYSTEM FOR SUPERKEKB

中村卓也 *, A)、岩崎昌子 B)、帯名崇 B)、佐々木信哉 B)、浅野和哉 C)
Takuya Nakamura*, A), Masako Iwasaki B), Takashi Obina B), Shinya Sasaki B), Kazuya Asano C)
A)Mitsubishi Electric System and Service Co.,Ltd.,
B)High Energy Accelerator Research Organization (KEK),
C)Kanto Information Service Co., Ltd.,

Abstract

We have developed a new alarm system for the SuperKEKB accelerator operation based on CSS. In the past KEKB operation, we used the alarm monitoring program system based on the SAD script. For the stable operation under the 40 times higher luminosity environment, we have implemented the CSS alarm system, which is used in several accelerators. To apply the CSS based alarm system to SuperKEKB, here several 10 thousands points will be monitored, we did several performance tests to evaluate the stability under the SuperKEKB environment. In this paper, we report the evaluation tests of the CSS alarm system for SuperKEKB.

1． はじめに

高エネルギー加速器研究機構（KEK）では、KEKB 加速器を用いた B ファクトリー実験が行われてきた。現在、KEKB 加速器の更なる高輝度化を目的として、SuperKEKB 加速器の建設が進められている [1]。

SuperKEKB では、KEKB での電子・陽電子衝突頻度を約 40 倍高めることを目指しており、このような高輝度加速器の運転下において、安定して動作可能なアラームシステムの構築が重要である。KEKB では、主に SAD スクリプト [2] を利用したアラームシステムを運用してきたが、我々は SuperKEKB のための新しいアラームシステムとして、Control System Studio(CSS) [3] [4] を利用したアラームシステムの導入を検討している。ここで、CSS を用いたアラームシステムは、近年いくつかの加速器で導入されているが [5]、SuperKEKB はそれらの加速器に比べて、監視する点数が非常に多いという特徴がある。KEKB でアラームシステムが監視する点数は約 25000 点であったが、SuperKEKB ではさらに増加すると考えられ、そのような運転環境下で、CSS アラームシステムが正常に動作するかどうか、評価が必要となる。

本件では、SuperKEKB への CSS アラームシステム導入を検討するための、アラームシステム動作試験、およびその性能評価について報告する。

2． CSS アラームシステムの構成

CSS のアラームシステムは、いくつかのソフトウェアの組み合わせによって構成されている (Figure 1)。CSS ではアラームシステムの根幹をなすソフトウェアが用意されており、Alarm Server(アラーム監視サーバー)、Alarm Client(アラーム表示クライアント)、Alarm Config Tool(設定変更ツール)、JMS2RDB(アラーム履歴保存) といったソフトウェアがある。アラームシステムの各ソフトウェアは、JMS(Java Message Service) を通じて互いにメッセージの通信を行っており、その JMS のソフトウェアとして Apache ActiveMQ を使用している。また、アラームの設定や履歴情報の保存には RDBMS(Relational Database Management System) を利用している。CSS がサポートしている RDBMS には、Oracle [6]、MySQL [7]、PostgreSQL [8] があり、我々の環境では PostgreSQL を採用している。

CSS アラームシステムのメインのソフトウェアである Alarm Server は、EPICS レコードのアラーム情報をモニターして動作している。アラーム情報の変化を検知すると、その最新の値を RDBMS へと記録し、また、JMS を通じて他のソフトウェアへと情報を伝達する。JMS を通じたメッセージを受け取ると、Alarm Client ではアラーム情報の表示を行い、JMS2RDB ではアラームの履歴情報として RDBMS への記録を行う。Alarm Config Tool は、XML 形式で記述されたアラーム監視の設定ファイルをもとに、RDBMS に記録されているアラームの設定項目を一度に更新するソフトウェアである。そのため、大量の設定変更を行う際に便利に利用することができる。また数件程度の小規模な変更であれば、Alarm Client からオンラインで変更する事も可能である。

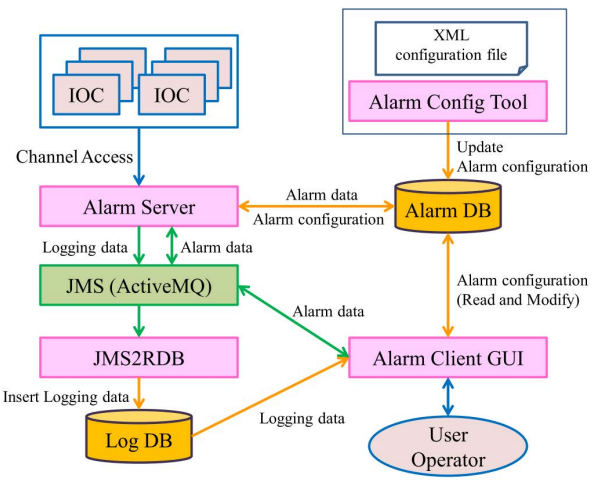

Figure 1: CSS alarm system structure.

* nakataku@post.kek.jp

3． CSS アラームシステムの性能評価

PF-AR 加速器では、2011 年 10 月に CSS アラームシステムを導入し、これまで大きなトラブルもなく順調に運用を続けている。ここで PF-AR の CSS アラームシステムが監視するレコードの点数は 1150 点ほどであり、KEKB で監視していた点数の 25000 点と比べても、20 分の 1 以下の量となっている (Table 1)。また、SuperKEKB では監視点数のさらなる増加が予想されるため、CSS アラームシステムが数万点のレコードを監視可能かどうか、性能評価を行う必要がある。

Table 1: Number of Monitoring Points

KEKB		PF-AR (CSS Alarm)	
グループ	監視点数	グループ	監視点数
Linac	678	BM	10
Linac (RF)	137	BT	302
BT (positron)	560	Control	2
BT (electron)	675	ID	15
MG (LER)	8794	Kicker, Septum	73
MG (HER)	5497	Magnet	540
RF (LER)	1141	RF	197
RF (HER)	1656	Stopper	1
VAC (LER)	2190	Safety	14
VAC (HER)	1169	Vacuum	3
Operation	408		
BM	1243		
Safety	330		
合計	24478	合計	1157

3.1 性能評価の試験環境

KEKB 加速器の監視点数の 2 倍程度となる 50000 点を監視する試験環境を整備し、CSS アラームシステムで動作させる各ソフトウェアの実行時間について調査する。

性能評価試験で用いるレコードの構成は、周期的に値が変化する calc レコード 1 個と、その値によってアラーム状態が変化する calc レコード 1 個の、2 個で 1 セットとする。calc レコード 2 個のセットを 50000 セット用意し、合計 100000 レコードを扱うソフトウェア IOC を動作させる。また、アラームの発生状況については、40 秒ごとにアラーム状態となるレコードが 200 個、アラームが解除されるレコードが 200 個となるよう整備し、平均して 1 秒あたり 10 件のアラーム状態の変化が起きる構成とする。

このような試験環境を 3 種類の計算機に整備し、それぞれの計算機で性能評価を行った (Table 2)。計算機 1 は、PF-AR の CSS アラームシステムを運用している計算機と同等の性能を持つ計算機である。計算機 2 は、PC 上に VMware による仮想 OS 環境を構築した環境である。計算機 3 は、SuperKEKB 用の CSS アラームシステムの運用を予定している、近頃導入した計算機である。

Table 2: Specification of Linux Computers

	計算機 1	計算機 2	計算機 3
Type	Blade server	PC(VMware)	1U server
CPU	Intel Xeon E5450 3GHz	Intel Core 2 Duo 2.53GHz	Intel Xeon E5-2603 1.6GHz
# of cores	4	2	12 (6 x 2)
Memory	4GB	3.3GB	32GB
OS	CentOS 5.8	CentOS 6.5	CentOS 7.1
Disk Type	HDD	HDD	SSD
CSS source	3.1.2	3.1.2	3.1.2
Java	1.6.0_26	1.7.0_45	1.7.0_79
ActiveMQ	5.10.0	5.10.0	5.11.1
PostgreSQL	9.3.5	9.3.5	9.4.4

3.2 Alarm Config Tool

Alarm Config Tool を使ったアラーム設定の更新処理について、実行時間を測定する。50000 件のアラーム監視設定を記述した XML ファイルを元に、RDBMS のデータの更新を行う。この時、読み込ませる XML ファイルを調整し、設定の追加のみ行う場合、設定の削除のみ行う場合、設定の削除と追加を続けて行う場合の、3 種類のデータの更新方法について実行時間を測定した (Table 3)。

Table 3: Execution Time of Alarm Config Tool

	計算機 1	計算機 2	計算機 3
実行時間 (追加)	10 分 30 秒	4 分 30 秒	2 分 15 秒
実行時間 (削除)	4 分 30 秒	1 分	1 分
実行時間 (削除と追加)	14 分 30 秒	5 分 30 秒	3 分 10 秒

この試験について、計算機 1 と計算機 2 について計算機の性能を単純に比較すると、計算機 1 の方が高い性能であるにもかかわらず、実行時間は計算機 2 の方が速くなるという結果となった。Alarm Config Tool による RDBMS のデータ更新は、一度設定を全て消去してから改めて登録を行う方法となっており、この時に大量のデータ更新処理が行われる。そのためディスクの書き込み速度や RDBMS の処理速度などによって、Alarm Config Tool の実行時間が左右されるものと思われる。

今回の試験について、計算機 1 ではアラーム設定の変更に 14 分 30 秒もの時間がかかっており、加速器の運転中に設定を変更するというような、短時間での設定変更要求があった場合に対応することができない。より高速にアラーム設定変更を行える手段について、検討する必要がある。

3.3 ソフトウェア IOC 起動時間

ソフトウェア IOC は、CSS アラームシステムにとって必ずしも必要なソフトウェアではないが、柔軟性のあるアラーム判定を行うために必要となることがある。例えば、元のレコードのアラーム判定と異なるアラーム判定を行う場合や、1 つのレコードに対して値ごとに複数のアラーム判定を行う場合、複数のレコードの値を総合

してアラーム判定を行う場合など、ソフトウェア処理を行うレコードを持つ IOC が必要となる。今回は 100000 件のレコードを持つソフトウェア IOC について、起動にかかる時間を測定した (Table 4)。また計算機 1 では、100000 件のレコードのロードについて 10000 件ごとにレコードの定義ファイルを分割して用意し、10000 件ごとのロードにかかる時間の測定も行った (Table 5)。

Table 4: Start-up Time of Software IOC

	計算機 1	計算機 2	計算機 3
起動時間	10 分 15 秒	10 分 30 秒	4 分 10 秒

Table 5: Load Time Every 10000 Records

レコード数	時間	レコード数	時間
1～10000	1 秒	50001～60000	63 秒
10001～20000	9 秒	60001～70000	75 秒
20001～30000	25 秒	70001～80000	87 秒
30001～40000	38 秒	80001～90000	98 秒
40001～50000	51 秒	90001～100000	110 秒
		IOC Initialize	58 秒
		合計	10 分 15 秒

PF-AR 加速器では、監視しているほぼ全てのレコードのアラーム判定を一つのソフトウェア IOC で行っているが、1000 件程度では大きな負荷ではないため問題にはならない。今回の試験により、数万件のレコードを運用する場合には大きな負荷となってしまう事がわかった。一つのソフトウェア IOC にて大量のアラーム判定レコードを運用するよりも、ハードウェアを制御する各 IOC にアラーム判定用レコードを整備する事が望ましい。

3.4 Alarm Server、Alarm Client 起動時間

Alarm Server、及び Alarm Client の起動にかかる時間について測定を行う。Table 6 に示す通り、Alarm Server については 1 分以内に起動が完了し、Alarm Client についても 20 秒から 40 秒程度で起動が完了しており、運用の面では特に大きな問題はならないと思われる。

Table 6: Start-up Time of Alarm Server

	計算機 1	計算機 2	計算機 3
Alarm Server			
RDBMS からの設定読み込み	5 秒	12 秒	3 秒
レコードへの接続	12 秒	40 秒	5 秒
合計	17 秒	52 秒	8 秒
Alarm Client			
CSS 起動時間	5 秒	8 秒	5 秒
Alarm GUI 起動時間	15 秒	25 秒	15 秒
合計	20 秒	33 秒	20 秒

3.5 アラームログの書き込み時間

レコードのアラーム状態が変化すると、JMS2RDB によってアラームのログが RDBMS に書き込まれる。今回の試験環境では、40 秒ごとに 400 件のアラーム状態の変化が起こるよう整備しており、その時のログの書き込みにかかる時間を測定する。また、大量のアラーム状態変化が起きた時の、ログの書き込み時間についても測定する。大量のアラームを同時に発生させるために、性能評価用のソフトウェア IOC の停止操作を行う。このソフトウェア IOC の停止を行うと、監視対象のレコードが無くなるため、Alarm Server は INVALID Alarm であると判定する。この時に同時に 50000 件のアラーム状態の変化が発生するため、その際のログの書き込み時間について測定する。これら 400 件のログの書き込み、及び 50000 件のログの書き込みについて、Table 7 に測定結果を示す。

Table 7: Execution Time of Logging Message Writing

	計算機 1	計算機 2	計算機 3
ログ書き込み (400 件)	3.37 秒	1.99 秒	0.52 秒
ログ書き込み (50000 件)	8 分	9 分 40 秒	1 分

400 件のログの書き込みについて数秒程度の時間がかかっており、1 秒あたりの書き込み可能件数は 100 件から 200 件程度となる。過去の KEKB 運転時のアラームの発生状況を調査すると、ビームのアボートが起きた際に一度に 30 件程度のアラームが発生していた。今回の試験結果にあるように、1 秒あたり 100 件以上処理できる性能であれば問題ない。

大量のログの書き込みについては、全てのログの書き込みが完了するまでに長い時間を要する事が分かった。大量のアラームが発生した直後にログを確認しても、一部の情報しか書き込まれていない状況となる。アラーム発生時の全てのログを確認したい場合には、書き込み完了まで長い時間を待たなくてはならない。

KEKB 運転時のアラームの発生状況を調査すると、加速器の保守日などに数秒間のうちに数千件のアラームが発生することがしばしばあった。そのような場合に、数分間ログが確認できない状況になる事は問題であるため、ログの書き込み速度を改善する方法について検討する必要がある。

4. 問題点の対策

4.1 アラーム設定の RDBMS 直接編集

Alarm Config Tool によるアラーム設定の変更時に、実行時間が長くかかる問題について検討する。

Alarm Config Tool を使用せずに、直接 RDBMS を編集する方法について調査を行った。Alarm Client にはアラームの設定を 1 件ずつ変更する機能があるが、その時に実行される SQL 文を解析した。その解析結果から RDBMS のデータを直接編集する SQL を整備し、50000 件のアラーム設定の変更処理を実行した。Alarm Config Tool では 14 分 30 秒かかっていた処理が、RDBMS の直接編集により 7 分で完了する事を確認した。RDBMS を直接編集するツールを整備するなどして、加速器の運

転中の設定変更など、短時間での設定変更の要求に対応できるものと思われる。

4.2 RDBMS (PostgreSQL) の調整

PostgreSQL のパラメータの調整によって、データの登録にかかる実行時間を速くする事ができないか調査した。通常、PostgreSQL では、ログ先行書き込み (WAL) というトランザクションのログを残す手法が設定されている [9]。この WAL のファイルが実際にディスクに書き込まれるまで待つかどうか、synchronous_commit というパラメータで調整する事ができる。

このパラメータは初期値では ON になっており、PostgreSQL へのデータ登録時に WAL のファイルがディスクに書き込まれるのを待ってから、クライアントにデータ登録完了の報告を返すよう動作する。このパラメータを OFF に設定すると、WAL のファイルの書き込み完了を待たずに PostgreSQL のデータ登録が完了となり、データ登録にかかる実行時間を短縮することができる。ただし、トランザクションのログ保存の確実性が低下することを考慮する必要がある。計算機 1 と計算機 3 の環境で、synchronous_commit を OFF した場合のデータ登録作業の実行時間を測定した。

Table 8: Comparison of an Execution Time by Synchronous_commit Parameter Setting

計算機 1		
synchronous_commit	ON	OFF
アラーム設定 (削除と追加)	14 分 30 秒	2 分 45 秒
ログ書き込み (50000 件)	8 分	1 分
計算機 3		
synchronous_commit	ON	OFF
アラーム設定 (削除と追加)	3 分 10 秒	2 分 50 秒
ログ書き込み (50000 件)	1 分	50 秒

Table 8 に示した通り、計算機 1 では synchronous_commit を OFF にすることで、実行時間を大幅に短縮できる事を確認した。計算機 3 については、実行時間は速くなったものの大きな差は見られなかった。この要因の一つとして、計算機 3 が使用するディスクが SSD であるため、元々ディスクの書き込み性能が高いことが考えられる。

ここで、synchronous_commit を OFF にした場合に、データ損失の危険があることを検討する必要がある。synchronous_commit が OFF の状態でデータベースがクラッシュした場合に、WAL のファイルの書き込みが完了しているかどうか確実性が無いため、データベースをリカバリーする際に直近 (初期値では 600 ミリ秒) のデータの書き込みが保証されなくなるという危険性がある。そのため、データ損失の危険性と、書き込み性能のどちらを優先するかを検討した上で、パラメータを調整する必要がある。

5. 実運転レコードでの動作試験

これまで試験用のレコードを使って負荷試験を行っていたが、実際の機器制御用のレコードを監視し、CSS アラームシステムの動作試験を行う。

現在、真空機器制御のレコード 4139 点について CSS アラームシステムでの監視を開始している。真空グループの作業により圧力が上昇する事があり、その時の CSS アラームシステムの動作について確認した。真空ゲージの値をモニターするレコードには、圧力の値によって、正常、軽度警告、重度警告の状態が切り替わるよう、アラームのしきい値を設定している。真空の作業により圧力が上昇した時に、真空ゲージごとに軽度警告や重度警告のアラームが発生した。この時、CSS BOY で作成された制御パネルの圧力の表示には、軽度警告の値は黄色で、重度警告の値は赤色で表示されていた (Figure 2)。CSS アラームの表示プログラムでも対象のレコードが黄色と赤色で表示され、CSS BOY の制御パネルの表示と一致しており、CSS アラームシステムが正しく動作している事を確認した (Figure 3)。

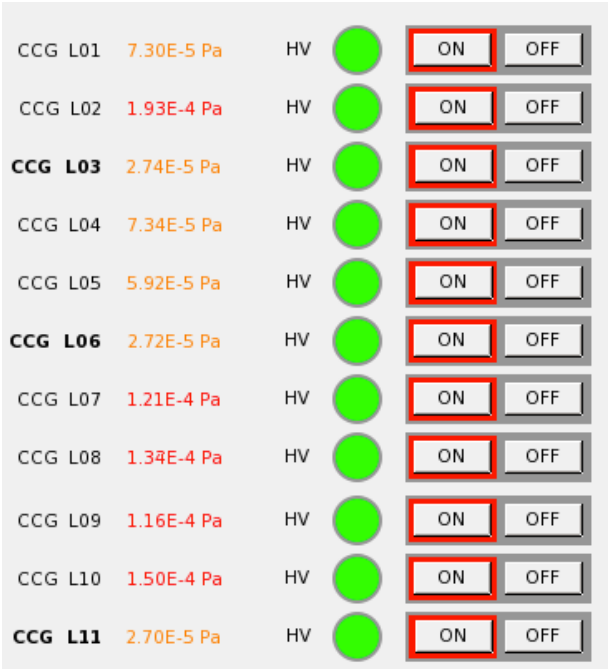

Figure 2: CSS BOY control panel.

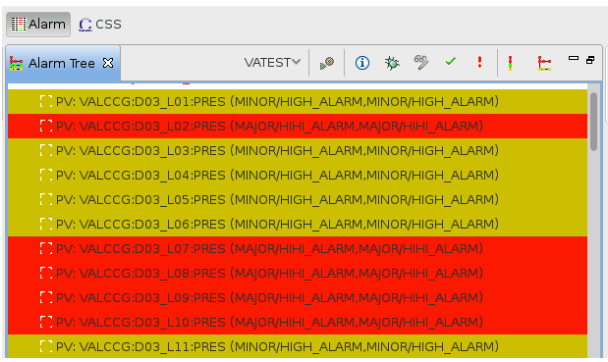

Figure 3: CSS alarm client, alarm tree panel.

6． まとめ

SuperKEKB 用のアラームシステムとして CSS アラームシステムの導入を検討し、安定した運用が可能であるか性能の評価を行った。KEKB 加速器での監視点数の 2 倍にあたる、50000 点のレコード監視について動作試験を行い、正常な運用が実現可能である事を確認した。今後は SuperKEKB 用 CSS アラームシステムへの実運転レコードの追加を進めていき、SuperKEKB 加速器の運転開始に備えたい。

参考文献

[1] N.Ohuchi, et al., “Construction Status of SuperKEKB”, Proc. of IPAC14, pp. 1877-1879 (2014); http://accelconf.web.cern.ch/AccelConf/IPAC2014/papers/weoca01.pdf

[2] SAD program http://acc-physics.kek.jp/SAD/

[3] Control System Studio http://cs-studio.sourceforge.net/

[4] Control System Studio (CSS) at KEK http://www-linac.kek.jp/cont/epics/css/

[5] 中村卓也, 他, “PF-AR 加速器制御におけるアラームシステムの更新”, Proceedings of the 9th Annual Meeting of Particle Accelerator Society of Japan, Osaka, Aug. 8-11, 2012.

[6] Oracle Database http://www.oracle.com/jp/database/overview/

[7] MySQL https://www-jp.mysql.com/

[8] PostgreSQL http://www.postgresql.org/

[9] PostgreSQL ログ先行書き込み (WAL) https://www.postgresql.jp/document/9.4/html/runtime-config-wal.html