RIBF 制御系高信頼化への死活監視システム実装の試み

AN ATTEMPT TO IMPLEMENT ALIVE MONITORING SYSTEM FOR RELIABLE EPICS-BASED RIBF CONTROL SYSTEM

内山暁仁#, A), 込山美咲 A)
Akito Uchiyama #, A), Misaki Komiyama A)
A) RIKEN Nishina Center

Abstract

The modern control system model for large accelerator consisted of three-layer structure that is operator interface, middle layer by framework (EPICS, MADOCA, TANGO, and etc.), and device interface layer. In case of RIBF control system, EPICS is utilized as the framework for the construction and many network-based devices are utilized with Ethernet as the device interface layer. On the other hand, in the three-layer structure, it is not easy to investigate the case of a problem from the operator interface, when the control system has failure in the device interface layer. In order to achieve the higher reliable RIBF control system, the problem should be detected as soon as possible. Therefore, we have developed and introduced the alive monitoring system for the EPICS IOCs and the network-based devices. The information for the EPICS IOCs and the network-devices is needed to implement the alive monitoring system, and it is obtained from EPICS PV management system developed by RIKEN Nishina Center in 2015. In actual use, we have confirmed that this alive monitoring system is useful method to detect the trouble status.

1. はじめに

現在における大型実験施設向けの制御モデルは、オペレータインターフェースとデバイスインターフェース間にフレームワーク (EPICS[1], MADOCA[2], TANGO[3] 等) から成る中間層を挟む三層構造で実現されている。一方、三層構造は障害時の見通しが良くない場合がある。近年、VME や CompactPCI といった CPU とデバイス間の通信（デバイスインターフェース層）がフィールドバスになっているシステムだけでなく、Ethernet をデバイスインターフェース層とみなしネットワークベースのデバイスを用いて、システム実装する事例も多い[4]。RIBF 制御系では主に EPICS を用いて制御システムを構築しており、横河電機 FA-M3 や理研で開発された N-DIM[5]といったネットワークベースのデバイスを多く採用している[6]。

RIBF 運転時では仮に制御システムに何らかの不具合が起きたとしても、これらデバイスを含めた原因究明を迅速に行う事がビーム可用性の観点からも重要である。例えば障害時の原因が EPICS IOC (Input/Output Controller) にあるのか、ネットワークにあるのか等、三層構造ではオペレータの観点から迅速に判断することは簡単ではない。したがって EPICS を用いた RIBF 制御系における障害時の原因究明を迅速に判断可能にさせる事ができれば、システムの高信頼化に繋がると考え、死活監視システムの開発、運用を始めた。

2. 検討すべき障害

EPICS ベースの制御システムにおいてオペレータの観点から、何らかの不具合が起きた時以下の対応が考えられる。デバイスに不具合が起きたケースでは、この時 EPICS Channel Access (CA) プロトコルは正常にソケットオープンできる。つまり CA クライアントは一見正常で、問題の現状を把握するためには EPICS IOC からのメッセージ出力をモニタ、若しくは Ping やポートスキャン (nmap 等) を用いてデバイスの状況を確認する事が必要になる。次に IOC プロセスが落ちているケースを考える。IOC と直接 CA 接続しているオペレータインターフェースをモニタする事で該当レコードに CA クライアントが接続できていないことは容易に確認できる。しかし RIBF 制御系の IOC 数は60を超えているので、該当レコードが属している IOC 名を瞬時に知ることは簡単ではない。また別々の IOC に属しているレコードを複数用いて EPICS Sequencer[7]を実装した場合には、CA 接続が切れている事をオペレータインターフェースで知る事ができないケースもある (Fig. 1 参照)。したがって結局 Sequencer のソースを確認し当該レコードに対応する IOC に対して、Ping や CA 用ポート (RIBF 制御系では標準 5064 番ポートを使用) の状況把握が必要になる。

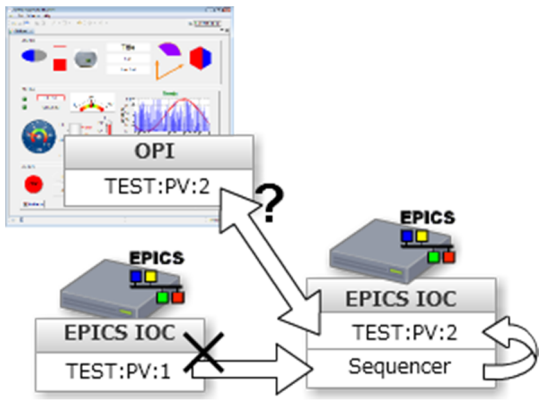

Figure 1: Example of the trouble case in EPICS Sequencer.

#a-uchi@riken.jp

3. システムデザイン

3.1 監視機能

大型実験施設制御システムにおける監視システムの先行研究として SPring-8 では FL-net に特化した監視システムの開発を行っている[8]。また EPICS IOC の死活監視としては Argonne National Laboratory (ANL)で Nagios ベースの監視システムが開発されている[9]。一方、デバイスのプロトコル、EPICS IOC どちらかだけではなく、双方の死活監視を簡便な設定で行う事ができるという事が本システムの特徴である。実装される監視機能は、以下である。

・IOC への Ping を利用したネットワーク死活監視。
・IOC へのポートチェックによる CA の死活監視。
・IOC への caMonitor を利用した CA の死活監視
・デバイスに対する Ping を用いたネットワーク死活監視。
・デバイスに対するポートチェックによるサービスの死活監視。

EPICS IOC やデバイスに対する Ping やポートスキャンを用いた死活監視は監視プログラムを polling させ実装する。また caMonitor は都度ソケットをオープンさせずとも、CA コネクションの接続を維持し続けコネクション切断時は必ずイベントが発生する。よって、そのイベントをキャプチャする事で IOC へのネットワーク経路に関する障害も監視可能であると考えた。システムの概要を Fig. 2 に示す。

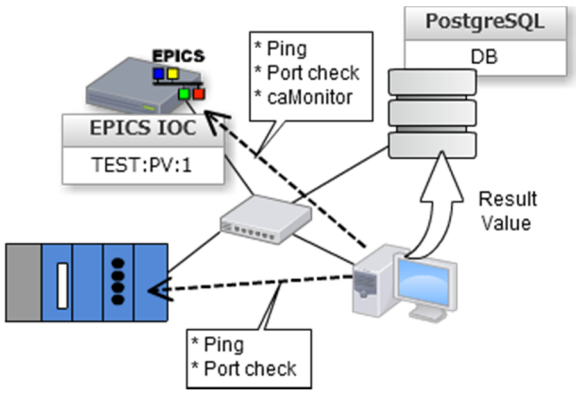

Figure 2: The behavior of alive monitoring system.

3.2 監視対象の情報

2.の議論より EPICS IOC だけでなく、ネットワークベースのデバイス全てにおいて死活監視が必要である。一方、システム構築するにあたり、監視対象の IOC とデバイスについての情報をゼロから作成していくと漏れがでる可能性がある。なぜなら複数人で開発を行える環境であり、デバイス追加に伴う EPICS データベースの変更がしばしば行われるからである。例えば 3.1 の先行研究で議論した ANL で開発された Nagios ベースの死活監視システムにおいても、管理システム IRMIS (Integrated Relational Model of Installed Systems)[10] と連携して監視対象の情報を取得している。よって、本システムでは監視対象の情報取得に昨年仁科センターで開発された EPICS PV (Process Variable) 管理システムを利用する事とした。

3.3 EPICS PV 管理システム

EPICS 管理システムの概要は、EPICS スタートアップスクリプトファイルの構文解釈をする事により実際運用している EPICS データバースで定義されているレコード名、フィールド名、ネットワークデバイスの情報 (IP アドレスと使用ポート) を抜き出し、自動で PostgreSQL ベースのデータベースに格納、管理する仕組みを持つ[11]。現在 RIBF 制御系で運用されている EPICS IOC の情報は全てこの管理システムによってデータベースに格納されている。よって EPICS PV 管理システムを使用する事で監視対象に必要な情報は漏れなく取得可能である。

4. システム概要

4.1 システムバックエンド

Ping とポートチェックの polling アクセスを用いた死活監視は、1 分おきに IOC とネットワークデバイスの状態を確認し、その結果は PostgreSQL ベースのデータベースに格納する。具体的なシステムの振る舞いは以下の通りである。管理システムから取得したポート番号を元にソケットの状態を確認する。この時管理システムから取得するポート番号の情報は IOC であれば 5064 が返り、ネットワークベースのデバイスであればそのデバイスで利用しているプロトコルに基づいたポート番号が返る (例えば、横河電機 FA-M3 シリーズであれば 12289 番ポート)。次に管理システムから取得したポート番号を元に TCP で状態を確認し、オープンされていれば、”1”をテーブルに挿入する。また指定ポートがクロースされていれば、Ping で再度状態をチェックしオンラインであれば”3”、オフラインであれば”0”をテーブルに挿入する。

caMonitor を用いた CA の死活監視では IOC が持つレコード全てをモニタするのではなく、主なレコード 10 個から 20 個程度をモニタしている。監視プログラムの振る舞いは、コネクションが確立した時にそのレコードの種類に応じた整数 (例えば bi レコードであれば”3”)がテーブルに挿入され、コネクションが切断されたら”-1”がテーブルに挿入される。

4.2 クライアントシステム

PostgreSQL ベースのデータベースに格納された死活監視情報はユーザインターフェースから読み出す。ユーザインターフェースは Web アプリケーションによって実現されており、VMware 仮想サーバ[12]上にて Apache と PHP の組み合わせで実装されている。また Web アプリケーションは Web アプリケーションフレームワークである Bootstrap[13]を用いて構築した。開発された Web アプリケーションのスクリーンショットを Fig. 3 に示す。このユーザインターフェース上で CA とネットワークデバイスの現在の死活監視状況と過去ステータスがログとして確認する事ができる。

5. 障害検知の例

死活監視システムは 2015 年 9 月ごろより試験運用

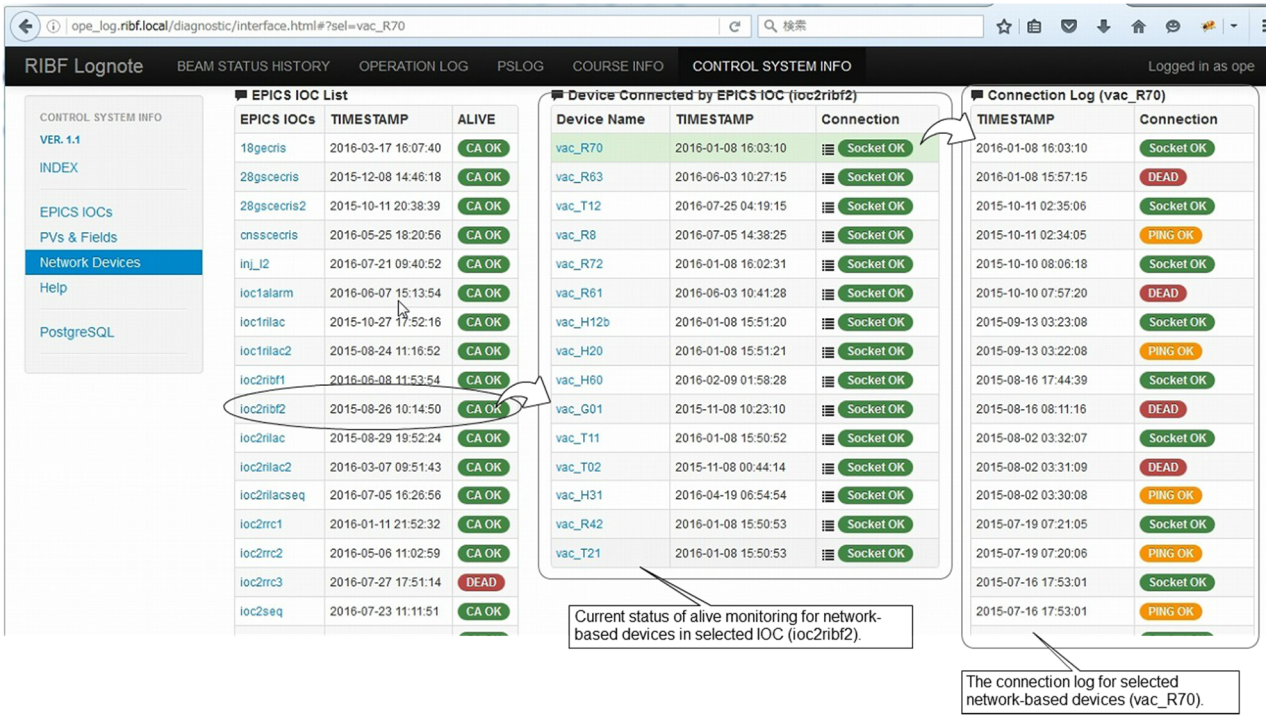

Figure 3: The screenshot of the user interface for alive monitoring system.

をしており IOC のプロセスが落ちた事やデバイス間ネットワークの不通といった単純な障害検知だけでなく、本システムを実装した事で原因が判明した障害の例もある。それらを以下に報告する。

5.1 ファラデーカップの挿入

ビーム輸送中にファラデーカップが勝手に挿入されるという事象が起きた。ファラデーカップを動作させるためのデバイスは N-DIM を用いており、我々はその障害を疑ったが原因の調査が必要である。原因調査でオペレーショナルログシステムと死活監視システムのログを付きあわせた所、当該 N-DIM の死活監視ステータスが”DEAD”から”Socket OK”に遷移したのと同じタイミングでファラデーカップが挿入されていたという事を確認した。N-DIM はその特徴として基盤に何らかの障害が起き、CPU 内部でリブートがかかった際安全方向、すなわちビーム照射を停止させるような動作をとる。よって今回の事象は基盤の障害でリブートがかかったためにネットワークから N-DIM が見えなくなり、ファラデーカップが挿入されたとログより判断できた。

5.2 Channel Access プロトコルの瞬断

深夜オペレーション中に特定の EPICS IOC に関わらずオペレータインターフェース GUI の EPICS PV 値が白抜きになったが、数秒後に元に戻る、といった事象があった。複数 IOC が同時に障害を起こすといった事は考えにくく、多くのオペレータインターフェースが死活監視システムと同様、仮想サーバ上(Fig. 4 参照)で実装されているためオペレータインターフェース用サーバと状況が似ていると考え、死活監視システムのログをまず調査した。

polling アクセスを用いた IOC への死活監視ログでは異常が確認できなかったが、caMonitor を用いた死活監視では同時刻に全ての IOC と CA コネクションが切れていた事がログに残っていた。この様な事象はネットワークの問題もしくは仮想サーバ自体に不具合が起きていると考え、同時刻に何かイベントが無いかリストアップした所、仮想サーバ用イメージファイルのバックアップ開始時間と同時刻に障害を起こしている事がわかった。イメージファイルには OS だけでなくシステム全体が含まれているためファイルサイズも巨大になる。また共有ディスクには高負荷のログが残っていたため、バックアップによる仮想サーバの機能低下が障害を引き起こした原因であると判断し、バックアップ手法の変更をした。

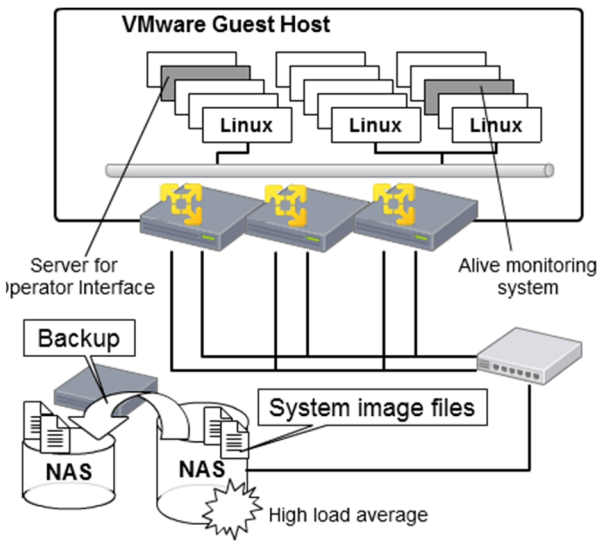

Figure 4: State of the trouble in the backup work.

6. まとめ

RIBF ではビーム可用性の観点から障害を迅速に検知する事が重要である。EPICS を用いた制御システムで障害を検知する死活監視システムの開発、実装を行った。昨年開発した EPICS PV 管理システムを利用する事で監視対象の情報を取得し、監視対象 IOC とデバイスを漏れなく死活監視する事ができた。本システムの実装後、複数の事例においてシステムの健全性診断に役に立っている。今後障害検知だけでなく、NetFlow[14]を用いてネットワークトラフィックをモニタし、制御システムの信頼性をさらに高めていきたいと考えている。

参考文献

[1] Dalesio, Leo R. *et al.*, Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 352.1 (1994): 179-184.
[2] T. Matsumoto *et al.*, Proceedings of the 11th Annual Meeting of Particle Accelerator Society of Japan, August 9-11, 2014, Aomori, Japan, P. 764-P. 767.
[3] Götz, A. *et al.* Proceedings of the ICALEPCS2003, Gyeongju, Korea, October (2003), P. 220-P. 222.
[4] J. Odagiri *et al.*, Proceedings of the 6th Annual Meeting of Particle Accelerator Society of Japan, August 5-7, 2009, Tokai, Ibaraki, Japan, P. 435-P. 437.
[5] M. Fujimaki *et al.*, RIKEN Accelerator Progress Report 37 (2004), p. 279.
[6] M. Komiyama *et al.*, RIKEN Accelerator Progress Report 40 (2007), p. 120.
[7] http://www-csr.bessy.de/control/SoftDist/sequencer/
[8] H. Kashima *et al.*, Proceedings of the 9th Annual Meeting of Particle Accelerator Society of Japan, August 8-11, 2012, Toyonaka, Osaka, Japan, P. 214-P. 217.
[9] D. E. R. Quock *et al.*, Proc. PCaPAC08, Ljubljana, Slovenia (2008), p. 19.
[10] D. A. Dohan *et al.*, Proceedings of ICALEPCS2007, Knoxville, Tennessee, USA (2007), p. 82.
[11] A. Uchiyama *et al.*, Proceedings of ICALEPCS2015, Melbourne, Australia (2015), No. WEPGF032.
[12] A. Uchiyama *et al.*, Proceedings of the 10th Annual Meeting of Particle Accelerator Society of Japan, August 3-5, 2013, Nagoya, Japan, P. 1109-P. 1112.
[13] http://getbootstrap.com/
[14] Claise, Benoit. "Cisco systems NetFlow services export version 9." (2004): https://tools.ietf.org/html/rfc3954