

IOC SURVEILLANCE SYSTEM FOR J-PARC MR CONTROL

Hiroyuki Nemoto^{1,A)}, Norihiko Kamikubota^{B)}, Noboru Yamamoto^{B)}, Shuei Yamada^{B)}, Susumu Yoshida^{C)}, Shigenobu Motohashi^{C)}, Takao Iitsuka^{C)}, Daisuke Takahashi^{C)}

^{A)} AC MOS INC

2713-7 Muramatsu, Tokai-mura, Naka-gun, Ibaraki, 319-1112

^{B)} J-PARC Center, KEK and JAEA

2-4 Shirakata Shirane, Tokai-mura, Ibaraki, Japan, 319-1195

^{C)} Kanto Information Service (KIS)

8-21 Bunkyouchou, Tsuchiura, Ibaraki, 300-0045

Abstract

J-PARC MR control system has been developed using EPICS toolkit. When we started construction for the MR control system, VME-bas computer was the default IOC (Input/Output Controller). Later, PLC-type CPU was introduced as a new type of IOC. Recently, Virtual IOC, an IOC on a KVM virtual machine, was added to our IOC types. In addition, the number of IOCs in total is increasing year by year. In order to realize easy management and surveillance of IOCs, we introduced PCMON library developed in EPICS community. Based on PCMON, we added new improvements to fit our environment. This article gives information in detail.

J-PARC MR 加速器制御システムにおける IOC (Input/Output Controller) 統合管理

1. はじめに

J-PARC MR 加速器制御システム^[1]は、EPICS (Experimental Physics and Industrial Control System) を採用して開発が行われている^[2]。機器制御計算機を、EPICS では IOC (Input Output Controller : 出力コントローラ) と呼ぶ。

J-PARC MR 加速器制御システムにおける当初の想定では、IOC として用いるのは VME bus 計算機のみであった。しかし、microIOC や PLC ハードウェアを用いた IOC が次第に導入され、近年では Blade Server 上に作成された仮想 OS を用いる Virtual IOC も導入された。そのため、IOC の台数・種類が増加している。

これらの IOC は、制御グループ内に機器グループごとの担当者を置くことで管理していた。しかし、台数の増加に伴い一括して管理・監視する必要性が増した。そこで、IOC 監視アプリケーション (iocStatus、pcMonitor) を整備した。これは、IOC 情報の取得と一覧表示画面から構成される EPICS アプリケーションである。今回の報告では、増加する IOC とその管理について報告する。

2. IOC

2.1 IOC 概要

IOC は、制御端末との通信をネットワーク上から EPICS 独自の通信プロトコルである CA (channel

access) を通じて行う。制御対象機器の制御は、各種 bus (Ethernet、GPIB 等) を通じて行う。IOC 上にて EPICS IOC シェル (iocsh) を実行することで制御プログラムが起動する。これにより、制御対象機器の制御と CA が開始される (図 1 参照)。

J-PARC MR 加速器制御システムでは、IOC に HDD 等ディスクを搭載しない。ネットワーク経由で Linux OS を起動し、主ファイルサーバ内の共通ディスクをマウントする。そして、共通ディスクに置かれた EPICS アプリケーションを起動することで、IOC として動作する。

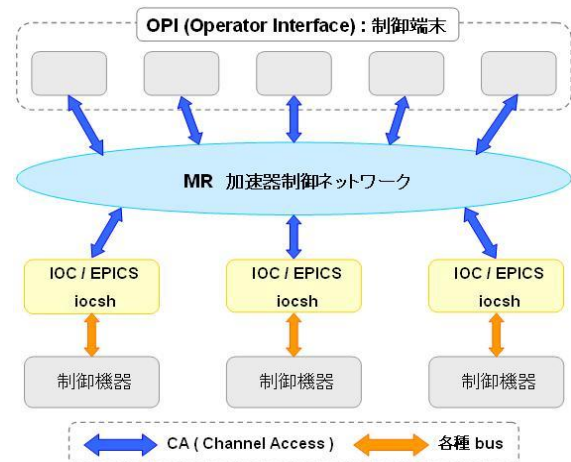


図 1 : IOC と通信関係図

¹ E-mail: hiroyuki@post.j-parc.jp

H24年7月時点において、J-PARC MR 加速器制御システムでは4種類、145台のIOCが稼動している。近年ではVME IOCは微増に留まるが、PLC IOCとVirtual IOCの増加が著しい(表1参照)。

各IOCは、使用場所や要求される処理能力により使い分けられている。以下、各IOCを紹介する。

表1：IOC稼動数内訳

| IOC 種別 | 台数(145) |
|-------------|---------|
| VME IOC | 84 |
| PLC IOC | 34 |
| Virtual IOC | 24 |
| microIOC | 3 |

2.2 VME bus 計算機

J-PARC MR 加速器制御システムにおける標準IOC(図2参照)。要求される処理能力に応じて4種類を使い分ける(表2参照)。

表2：VME bus 計算機機種

| 機種 | CPU | Memory |
|----------------|--------------------|----------|
| VMIC V7865 | Core Duo 2.0GHz | 3GB |
| VMIC V7807 | Pentium4M 1.8GHz | 1or1.5GB |
| VMIC V7700 | CeleronM 400MHz | 512MB |
| Sanritz SVA041 | LV-CeleronM 600MHz | 512MB |

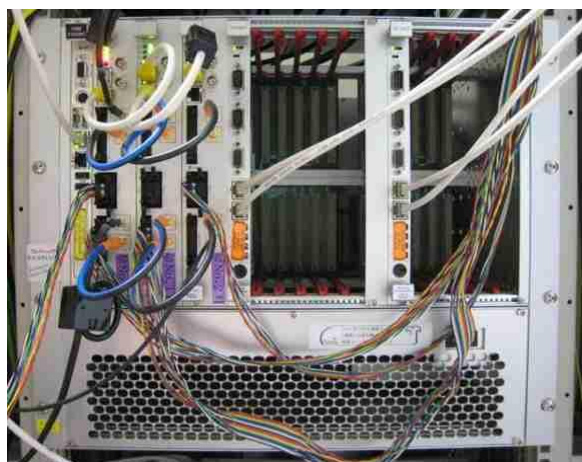


図2：VME bus 計算機

2.3 PLC

F3RP61(横河電機社製 PLC CPU)^[3]を使用。F3RP61はLinux対応CPUモジュールで、CF(Compact Flash)インタフェースを搭載する。CFにインストールしたLinux OS起動後、組み込みEPICSを動作させることができる。これにより、PLCでありながらIOCとして動作可能である(図3参照)。



図3：PLC(F3RP61)

2.4 Virtual IOC

KVMを用いてBlade ServerのホストOS上に仮想OSを作成し、仮想OS上にてLinux OSとEPICSを起動することでIOCとして動作する。これをVirtual IOC^[4]と呼称している。KVMはLinux上に仮想環境を作成するソフトウェアである。

仮想OSは、Blade Serverのメモリ容量の許す限り、複数台作成が可能である。そのため、VMEを複数用意する場合と比較して、コスト低減が可能である。

VME IOCほどの性能を必要としない監視系などが主な用途として挙げられる。

2.5 microIOC

microIOC^[5]は、筐体内に電源、各種bus(Ethernet等)、CF(Linux OS・EPICSアプリケーション内臓)を備える。そのため、ネットワーク環境や共通ディスクマウントの影響を受けることなく、独立したIOCとして動作可能である(図4参照)。

J-PARC MR 加速器制御システムでは、RS485(シリアル線)を持つ機器用に導入された。

microIOC背面のRS485コネクタとRS485インターフェースを持つ真空機器が直接結線されている。



(1) 正面



(2) 背面

図4：microIOC

3. IOC 監視アプリケーション(iocStatus、pcMonitor)

IOC 監視アプリケーション開発に際し、要件として下記が挙げられた。

- EPICS での開発
- IOC 詳細情報表示
- IOC timestamp 表示
- 一括表示画面

これらの要件について検討した結果、IOC 情報読み出し用、各 IOC 情報集約用、この二つのアプリケーションに分割して開発を行った。

3.1 iocStatus

各 IOC 情報の集約を行うアプリケーション。

IOC 監視用 IOC で実行。iocStatus で情報を集約後、一括表示画面に表示する。これにより、各 IOC 停止後も画面上的表示情報が保持される。

3.2 pcMonitor

IOC 情報読み出し用アプリケーション。各 IOC の st.cmd に設定することで、iocsh 実行中に情報収集を行う。PSI が開発した EPICS コミュニティ資産である PCMON:PC Monitoring Utility^[6]を利用し、機能拡張を加えて開発した。

PCMON は、Linux OS の /proc 以下を参照し、IOC 情報を取得する。そのため、機能拡張を容易に行えるという利点があり、開発のベースとして採用した。

スキャン間隔は、timestamp : 1 秒、詳細情報 : 10 秒。PID 等の固定値は起動時のみ読み込む。

開発により追加した機能として下記が挙げられる。

① ネットワークトラフィック

IOC の Ethernet port トラフィックを表示する。/proc/net/dev より、各 Ethernet port トラフィックを取得する。

② PID・PPID

iocsh の PPID (Parent Process ID)、PID (Process ID) を表示する。プログラム内で getppid()、getpid() を実行し、PPID、PID を取得する

③ CA Client 数

IOC に接続されている CA Client 数を表示する。/proc/net/tcp から CA Client に対応する local_address をカウントして接続数を取得する。

④ timestamp 誤差アラーム

各 IOC では timestamp record を実行し、timestamp を取得している。これを用いて、監視用 IOC を基準に各 IOC の timestamp 誤差を計測する。誤差が発生した場合、IOC_Status 画面上でアラーム発報を行う。発報時の警告色は、誤差が前後 5 秒以上であれば黄、30 秒以上であれば赤とした。

3.3 GUI

画面開発には、EPICS 標準 GUI ツールである MEDM (Motif Editor and Display Manager)を使用した。

① IOC_Status Main 画面

機器グループごとに所有 IOC が全台起動中かシグナルにより確認可能。IOC_Status 画面表示ボタンを配置 (図 5 参照)。

ALL IOC シグナルと機器グループシグナルが、IOC 稼働状況に応じて発報。全 IOC 起動時は緑、一台以上停止時は赤。

② IOC_Status 画面

IOC 一括表示画面。IOC hostname、timestamp、hert beat・CPU・Memory シグナル等、動作状態を把握する際の判断材料となる値を配置した。表示は機器グループごとに分類し、さらに IOC 種別での分類も行う (図 5 参照)。

③ G_PCMON_status 画面

各 IOC の詳細情報を表示する (図 6 参照)。

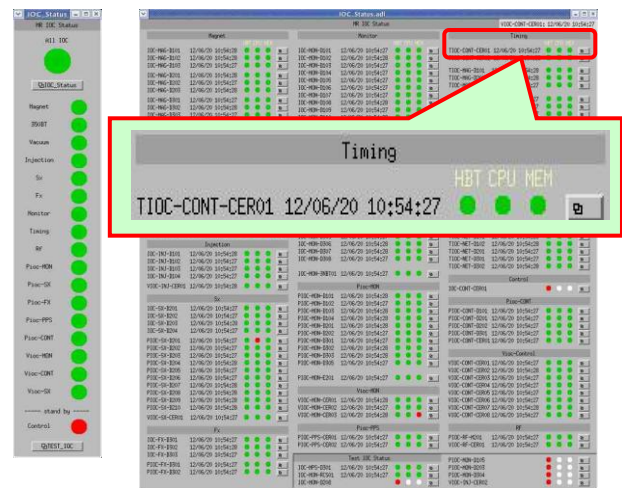


図 5 : IOC_Status

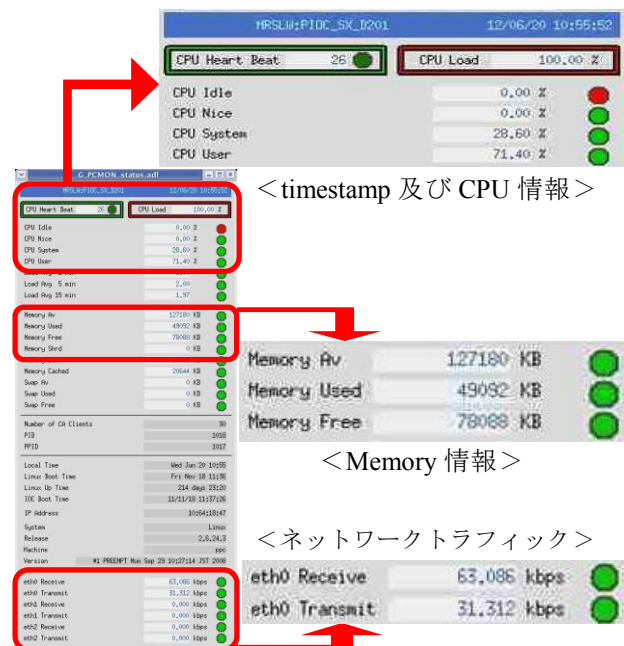


図 6 : G_PCMON_status

4. 考察

CPU・Memory 使用率、ネットワークトラフィック等、IOC 稼働中に変動する情報を準リアルタイムで確認可能になった。これにより、VME IOC の上位機種への変更による使用率緩和、専用ネットワークポートを用意することによるトラフィック低減など、負荷軽減が容易になった。

IOC 起動後の timestamp 誤差確認が可能になった。新規 IOC 起動時や長期停止後の再起動時、稀に timestamp 誤差を生じることがあった。しかし、画面上から確認できることにより、速やかな調査や再設定が可能になった。

IOC の突発的な停止や負荷上昇を、画面上から確認可能になった。これにより、該当 IOC の timestamp を元にして、関連データの履歴を調査する (MR 制御システムの Channel Archiver^[7]) ことで原因追求できるようになった。

IOC 監視アプリケーションの活用事例として、VME bus 計算機ボタン電池液漏れに伴う IOC 停止が挙げられる。

突然の IOC 停止を iocStatus 上で確認したため、停止時刻付近の Channel Archiver データを参照、負荷の増大等なく突然停止していた (図7参照)。次に、IOC へネットワーク経由でアクセスしたが応答無いため、VME bus 計算機を直接確認した。パネル上の LED 点灯確認後、電源から取り外すと基板上的ボタン電池が液漏れを起こしていた (図8参照)。代替機を用意し差し替え後、再度調査を行った。

その後、VME bus 計算機の掃除とボタン電池交換により、正常に動作した。

5. まとめ

今回開発した iocStatus、pcMonitor により、複数 IOC の同時監視が実現した。

IOC 総数と種類を把握しやすくなった。

一覧表示画面により、管理側・ユーザー側双方において IOC 稼働状況の確認が容易に行えるようになった。

IOC 停止や異常を画面上から確認できることで、原因解明やその間の代替機の用意など、速やかな対処が可能になった。

参考文献

- [1] N.Kamikubota, et al., "J-PARC Control toward Future Reliable Operation", Proceedings of the ICALEPCS 2011, Grenoble, France Oct.2011 p.378
- [2] EPICS web site "<http://www.aps.anl.gov/epics/>"
- [3] 小田切 他, "F3RP61 を利用した組み込み EPICS の加速器制御への応用", 第 6 回加速器学会年会 (東海村)、2009 年 8 月 5-7 日、p.435
- [4] N.Kamikubota, et al., "Virtual IO Controllers at J-PARC MR using Xen", Proceedings of the ICALEPCS 2011, Grenoble, France Oct.2011 p.1165
- [5] microIOC web site "<http://www.microioc.com/>"
- [6] PCMON website "<http://epics.web.psi.ch/software/pcmon>"
- [7] N.Kamikubota, et al., "Data Archiver System for J-PARC Main Ring", Proceedings of the IPAC'10, Kyoto Japan May 23-28, 2010 p.2680

Archiver XML-RPC Interface (MR)

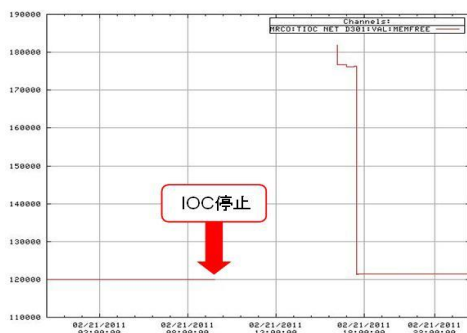


図 7 : Channel Archiver



図 8 : ボタン電池液漏れ