

仮想化 Archiver Appliance の導入と運用

○佐武 いつか^{a)}、榎本 嘉範^{a)}、佐藤 政則^{a)}、草野 史郎^{b)}、工藤 拓弥^{b)}

^{a)} 高エネルギー加速器研究機構、^{b)} 三菱電機システムサービス株式会社

1. はじめに

KEK 電子陽電子入射器(KEK 入射器)^[1]では、電子および陽電子ビームを SuperKEKB 電子/陽電子、陽電子ダンピングリング、PF、PF-AR の異なる 5 つのリングに供給している。現在も安定した運転と更なるビーム性能向上のため、様々な技術研究開発を続けている。運転に関わる様々な機器や周辺環境の情報を記録するため、膨大な運転データを収集・保存できるデータ収集システムの導入は不可欠である。KEK 入射器では、データ収集ソフトウェアとして Archiver Appliance(AA)^[2]を運用しており、17 万点以上のデータを効率的に収集・保存する仕組みを構築している^[3]。

こうした運用のなかで、他施設から本システムの導入を求める声が寄せられた。これを受け、より柔軟かつ迅速にシステムを展開できるよう、仮想化環境で動作する AA の整備を進めた。簡易な設定で導入できる仕組みを構築し、実際に導入先の施設で安定稼働することを確認できた。本研究会では、仮想化 AA の導入および実際の運用結果について詳細を報告する。

2. 仮想化 Archiver Appliance の概要

2.1 Archiver Appliance の概要

KEK 入射器では、Experimental Physics and Industrial Control System (EPICS)と呼ばれる制御システムソフトウェアフレームワークを使用し、加速器の運転制御をおこなっている。AA は、EPICS のデータ (Process Variable, PV) を長期間にわたって記録し、検索・可視化できるアーカイブシステムである。

AA は、クラスタ化による負荷分散に対応しており、複数のノードでデータ収集や読み出しなどの処理を分担している。短期、中期、長期データ保存用に異なる記憶媒体を使い分けることが可能で (Ramdisk, SSD, SAN など)、効率的なデータ管理を実現している。また、ブラウザを用いたシステム管

理・監視機能をもつ管理インターフェイスや Python を用いた処理スクリプトなどが用意されている。

AA は、Apache Tomcat 上で動作する Java Servlet として実装されており、EPICS PV のデータ収集・保存・検索を担う。データ収集をおこなう Servlet (Engine) は PV の値とタイムスタンプを記録し、Google Protocol Buffers を用いたバイナリファイルとして記録される。データ読み出し用 Servlet (retrieval) は HTTP 経由でクライアントの要求に応じたデータを提供する。JSON、CSV、MAT などのフォーマットでデータ取得することも可能である。データを表示するクライアントには、CSS ツール群の一部である Data Browser (バージョン 4.3 以降) や Java Archive Viewer が利用可能である。

2.2 仮想マシンの概要

仮想マシンのセットアップには、VMware Workstation Pro を利用した。VMware Workstation Pro は、Windows および Linux 上で動作する高機能なデスクトップ仮想化ソフトウェアである。複数の仮想マシンを同時に実行でき、開発、テスト、シミュレーションなど幅広い用途に対応している。高度なネットワーク設定 (NAT、ブリッジ、ホストオンリーなど) を備えており、柔軟な仮想ネットワークの構築が可能である。また、スナップショットやクローン機能により、仮想マシンの状態を保存・復元・複製できるため、環境構築やトラブルシューティングを効率的におこなうことができる。

3. 仮想化 Archiver Appliance のセットアップ

3.1 インストールおよび設定

VMware Workstation Pro17 を用いて、Rocky Linux の仮想マシンを構築・デプロイした。下記の手順に従い、仮想化 AA を構築した。

1. VMware Workstation Pro のインストール
2. 仮想マシンの作成 (OS の種類とそのバージョン、ディスク容量とメモリサイズの指定)
3. 選択した iso ファイル取得

4. 仮想マシンへ OS のインストール
5. 仮想マシン上で各種ソフトウェアのダウンロード、展開
6. AA のインストール
7. サービスの自動起動設定

3.2 ログファイルに関する設定

ログレベルを 警告 (WARN) に設定し、重要な情報のみを記録するようにした。また、日ごとにログファイルを作成する設定から、ファイルサイズに基づくローテーション方式へ変更した。日単位のログの場合、生成量によってファイルサイズが大きく異なり、管理が煩雑になる。一方で、ファイルサイズに基づくローテーションを採用することで、ログの分割が均一になり、一定期間のログを効率的に処理できる。1つのログファイルが 10 MB に達すると新しいファイルを作成し、最大5ファイルを保持する。さらに、タイムスタンプはアプリケーションの起動後の通算時間(ミリ秒)で記録されていたが、標準的な日時フォーマット (IS08601) を採用し、ログの解析や監視がしやすくなる。

3.3 JVM オプションの変更

Java8 以降、ヒープ外のメモリ領域として使用されていた「Permanent」領域が削除され、新たに Native メモリ領域として「Metaspace」が導入された。本システムにおいても、この仕様変更に対応するため、sampleStartup.sh 内の JVM オプションを MaxPermSize から MaxMetaspaceSize に変更した。

3.4 MySQL 接続の問題

本システムでは MySQL への接続に JDBC を利用しているが、デフォルトの設定では接続が確立できない問題が発生した。この問題への対応として、JDBC 接続設定において TLSv1.2 を明示的に指定することで、正常に MySQL へ接続できることを確認した。

3.5 ユーザーによる設定

ユーザーがおこなう作業は以下の3つである。VMware Workstation Pro で別の環境に仮想マシンを移動した場合、.lck (ロック) ファイルを削除する必要がある。これは VMware が仮想マシンを実行している際に作成するロックファイルであり、同じ仮想マシンが複数同時に起動することを防ぐために使われる。

次に、VMware Workstation Pro の GUI を用いてネットワーク設定をおこなう。IP アドレス、ネットマスク、

およびゲートウェイの情報を適切に設定し、仮想マシンのネットワーク環境を構成する。設定完了後、変更を適用するために AA サービスを再起動する。

最後に、データの保存先を変更する。必要に応じて NAS をマウントする。AA サービスを停止した状態で、スタートスクリプト内でデータ保存先のパスを修正し、適切なディレクトリを指定する。設定の適用後、AA サービスを再起動し、変更が反映されたことを確認する。

4. 運用状況および障害対応

4.1 本システムの運用状況

2024年9月より、仮想化 AA によるアーカイブを開始した。図1に、仮想化 AA のシステム概要を示す。

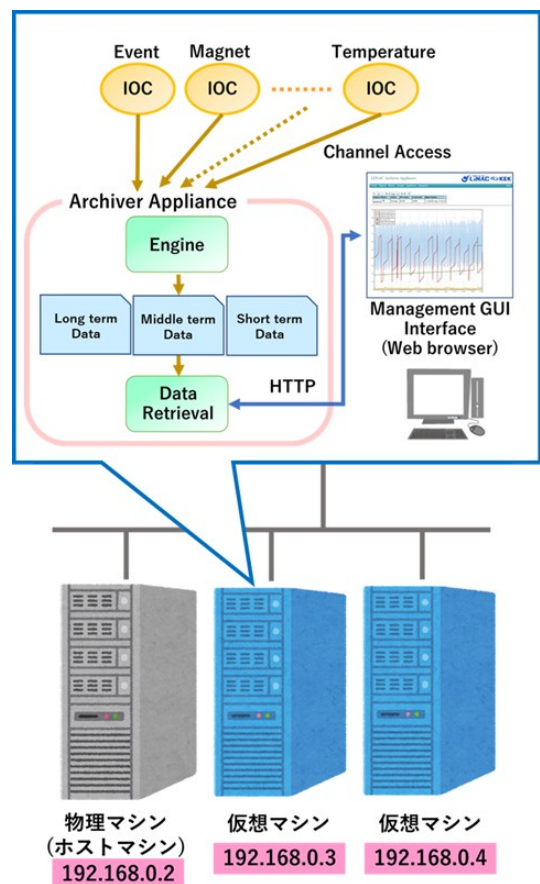


図1 仮想化 AA の概要図

仮想ネットワークの設定として、ブリッジ接続を選択した。これは、仮想マシンを物理ネットワークに直接接続するためのネットワーク設定方法である。この接続方法では、仮想マシンがホストマシンと同じネットワーク上のデバイスとして振る舞い、

物理ネットワーク上の他のデバイスと直接通信できる。仮想マシンは、ホストマシンのネットワークインターフェースカード (NIC) を通じて物理ネットワークに接続され、仮想マシンに対して個別の IP アドレスが割り当てられるため、外部のサーバーや他のデバイスと同様にアクセスできる。実際に運用を開始した本システムの構成および保存データについて、表 1 に示す。また、ソフトウェア環境を表 2 に示す。

表 1 データおよびメモリ構成

データ数	849
データサイズ	約 8 GB
メモリ (仮想マシン)	8 GB
メモリ (ホストマシン)	16 GB

表 2 システム構成ソフトウェア

AA	ver. 1.1.0
OS	Rocky Linux 9.4
Tomcat	apache-tomcat-9.0.93
Java	ver. 12.0.2 (OpenJDK 12.0.2)
Database	MySQL 8.0.36 (mysql-connector-java 8.0.15)

4.2 発生した障害内容と対処

AA のサービスを起動後、約 48 時間でサービスが停止する不具合が起きた。サービスを再起動しても、約 48 時間経過すると再び停止してしまうことがわかった。

AA の負荷が高まり、Java ヒープサイズとして割り当てていた 4GB を使い切る状況が発生していた。これにより、物理メモリが逼迫し、スワップが発生して OS 全体のパフォーマンスが低下し、最終的にプロセスが応答不能となる問題が生じていたとみられる。

スタートスクリプト内で Java ヒープサイズの割り当てを 4GB から 1GB に下げる対処を実施した。

ヒープサイズを 1GB に調整したことで、メモリ不足やスワップの発生を回避でき、AA のプロセスが過

負荷で停止する問題が解消されたと考えられる。

5. まとめと今後の展望

仮想化環境で動作する AA の整備をおこなった。簡易な設定で導入できる仕組みを構築し、実際に導入先の施設で安定稼働することを確認でき、その有効性を実証した。現在も継続してデータをアーカイブしており、数か月の間安定して運用できている。今後は、ソフトウェアのアップデートを簡易に実施できる仕組みの構築や、Viewer 機能の整備を進める予定である。これにより、運用環境の利便性を向上させ、より効率的かつ柔軟なシステム運用を実現することを目指す。

参考文献

- [1] H. Ego *et al.*, “Upgrade status of KEK electron/positron injector LINAC for improvement on beam injection to SuperKEKB”, Proceedings of the 21st Annual Meeting of Particle Accelerator Society of Japan, Yamagata, Japan, Jul. 31-Aug. 3, 2024, THP023.
- [2] The EPICS Archiver Appliance;
https://slacmshankar.github.io/epicsarchiver_docs/
- [3] I. Satake *et al.*, “OPERATION STATUS OF ARCHIVER APPLIANCE IN KEK ELECTRON/POSITRON INJECTOR LINAC”, in Proceedings of the 17th Annual Meeting of Particle Accelerator Society of Japan, Sep. 2-4, 2020, pp. 735-738.