

コンテナ技術を用いた教育情報システムの構築

175733E 宮平 賢

指導教員：河野 真治

Abstract

We renew our Educational Information System of the Faculty of Engineering of the University of the Ryukyus. The system future container system mainly, which is deiferente VM-based old system. Podman and Singularity is used as container engines, but we found building Podman container is vely slow in rootless mode. To improve the inconvenience of the container engine, we create ie-podman, witch is a wrapper of rootfull Podman. We also use Slurm as a task manager and a resource management and Ceph as file systems. As a result the system be comes open and convenient for student and teachers.

1 教育向けの情報システム

情報通信技術の普及に伴い学生が学ぶ学習環境が必要となる。その学習環境としてVMやコンテナにより、手軽に開発し試せる技術が普及している。だが、手元のPC上でVMやコンテナを立ち上げ、開発を行うことはできるが、VMやコンテナの使用には高性能PCや有料のクラウドサービスが必要になる場合がある。これらの負担をIT技術を学ぶ学生に負わせない新たな仕組みが必要である。

本研究では、本コースで提供するVM貸出サービスだけでなく、学科の汎用サーバのリソースを効率的に利用できる教育情報システムを提案する。教育情報システムには複数の汎用サーバと大容量ストレージサーバが存在する。複数のサーバを利用するにあたり、分散ストレージが必要となる。また、学習環境として利用されることから、複数の並列なアクセスに耐えられ、信頼性の高いファイルシステムが必要である。この要件を満たすストレージソフトウェアとしてCephを採用した。汎用サーバのリソースを効率的に利用するために、コンテナエンジンであるPodman, Singularity, ジョブスケジューラであるSlurmを採用した。これらのソフトウェアを合わせ教育情報システムの構築を行った。

2 Podman

PodmanはRedHat社が開発、提供するLinux上でOCIコンテナを開発、管理、実行するためのデーモンレスコンテナエンジンである[1]。PodmanはOCI準拠のコンテナランタイムに依存するため、前述したDockerなど他のコンテナエンジンと互換性を持つ。また、Podman CLIはDocker CLIと同じ機能を提供する。Podmanはコンテナとイメージストレージ、コンテナランタイムを介してLinuxカーネルと直接対話することで、デーモンレスで実行される。Podmanの制御下にあるコンテナは、特権ユーザ又は非特権ユーザのいずれかによって実行することができる。

3 Singularity

Singularity[2]とは、HPC環境向けに設計されたコンテナプラットフォームである。Singularityはマルチユーザに対応しており、コンテナ内での権限は実行ユーザの権限を引き継ぐため、ユーザに特別な権限の設定が必要ない。またデフォルトで、\$HOME, /tmp, /proc, /sys, /devがコンテナにマウントされ、サーバ上のGPUを簡単に利用できる。コンテナイメージはSingularity Image Format(以下、sif)と呼ばれる単一ファイルベースのため、アーカイブや共有が容易である。

4 Slurm

Slurm[3]はLinuxクラスタ向けのフォールトトレラント設計のジョブスケジューリングシステムである。Slurmには以下の3つの主要機能を提供する。

- 計算を実行するユーザに対してリソースへの排他的、非排他的なアクセスを割り当てる
- 割り当てられたノード上のジョブの開始、実行、モニタリングを行う
- 待機中のジョブキューを管理することにより、リソースの競合を解決する

5 Ceph

Cephは、RedHat社が開発、提供する分散ファイルシステムである。Cephは分散オブジェクトストレージであるRADOS(Reliable Autonomic Distributred Object Storage)がベースとなっている。RADOSはクラスタに保存されるデータの管理を待ち受け、保存オブジェクトへのアクセス方法としてObject Gateway, RADOS Block Device(以下、RBD), CephFSがある。Object GatewayはHTTP REST経由でクラスタに保存されるオブジェクトへ直接アクセスが

可能である。RBD はブロックデバイスとしてアクセスが可能で、libvirt を組み合わせて VM のディスクとして使用できる。また、RBD ドライバを搭載した OS にマップし ext4 や XFS などフォーマットして利用できる。CephFS は POSIX 互換のファイルシステムである。複数のアクセス方法を提供することで、用途に合わせ柔軟に変更することができる。

6 教育情報システムの構築

新システムは、VM ベースのシステムからコンテナベースへ移行する。新システムでも VM 貸出サービスを継続するが、新しく搭載される GPU を VM で利用するためには PCI パススルーなどの設定が必要となる。しかし、PCI パススルーでは、VM と GPU が 1 対 1 の関係になるため、GPU 希望する利用者全てに割り当てることができない。そのため、コンテナに移行することでリソースを効率よく利用し、管理を容易にする狙いがある。また、基幹サービスのデータを SSD 上に保存することで、サービスの高速化を図る。

システムは学生や教授などが利用するため、マルチユーザで利用できるコンテナエンジンが必要となる。そこで、マルチユーザに対応している Podman と Singularity を採用する。Podman は独自の名前空間でコンテナ内で特権機能を利用できるため、特権が必要なサービスなどの実行に向いている。また、Singularity はコンテナ内で実行ユーザの権限を引き継ぐため、利用者が作成したプログラムの実行には向いている。だが、Podman の rootless では実行できない機能があり、Singularity ではイメージの Build がキャッシュされず低速である。そこで、Podman を wrapper した ie-podman を作成した。

新システムはコンテナベースに変更したことにより、利用者は汎用サーバのリソースを利用できる。そのため、複数の利用者が多くのリソースを要求するプログラムを実行した場合、リソース不足やリソースの競合が考えられる。そこで、汎用サーバのリソースを効率よく利用できるようにするため、ジョブスケジューラである Slurm により管理を行う。Slurm の利用方針として、最悪待ち時間を減らすのではなく、計算リソースの利用効率を上げることを重視する。

旧システムで使用したファイルシステムの GFS2 はロックマネージャの影響が大きく、GFS2 上の VM イメージにアクセスできなくなることがあった。新システムでは汎用サーバに SAS SSD が 5TB と旧システムより多く搭載されている。2 台のサーバに演習や研究用で利用する貸出 VM のイメージを保存し、残り 2 台には本コースで利用しているサービスを提供する VM を保存する。汎用サーバに保存することで、単一障害時の影響を小さくすることができる。また、Ceph は自己修復と自己管理機能を持つため、ユーザのホームディレクトリとして利用する。

7 教育情報システムの利用

構築した教育情報システムの管理方法、利用方法について述べる。

7.1 ie-podman の利用

ie-podman は Podman を wrap して複数ユーザで利用することができるコンテナ管理ツールである。Podman はマルチユーザに対応しているため、ie-podman を利用せずともコンテナの作成などを行える。だが、コンテナへの IP アドレスの割り当てには、root 権限が必要となるため rootless では実行できない。そこで、ie-podman では Podman のすべての機能を wrap するのではなく、rootless では実行できない機能を提供する。

新しいコンテナの作成は、Podman の run と同じように実行できる。run 時に `-gpu` オプションを指定することでコンテナ内に GPU を割り当てる。また、`-ip` オプションを指定することで、使用されていない IP アドレスが割り振られる。ie-podman を使用して、新しいコンテナの作成はソースコード 1 のように行う。

ソースコード 1: コンテナの作成

```
$ ie-podman run --ip --gpu [IMAGE_NAME]
```

Singularity から `sif` ファイルの作成は Podman と違いイメージの Build 時にレイヤーごとにキャッシュされない。そのため、Build 中にエラーが発生すると一から Build を再開する必要がある。そこで、ie-podman で作成したイメージを `sif` ファイルへ変換する機能を作成した。ie-podman でイメージを作成し、ソースコード 2 の操作を行うことで `sif` ファイルへ変換が行える。

ソースコード 2: イメージの `sif` 変換

```
$ ie-podman sif [IMAGE_NAME]
```

8 教育情報システムの評価

8.1 ie-podman の評価

Rootless の Podman、Singularity の不便な点を補うため、Podman の wrapper である ie-podman を作成した。ie-podman により特権が必要な機能も、利用者に特別な権限を与えることなく利用できるようになる。また、ie-podman は rootful の Podman を wrapper することにより、コンテナやイメージが SSD 上に保存される。そのため、rootless の Podman より高速化を図ることが可能になる。

そこで、ie-podman でのイメージの Build 速度の比較を行う。速度の比較を行うコンテナエンジンは、Docker、ie-podman、rootless の Podman である。図 1 はコンテナエンジンにおけるイメージの Build 速度である。

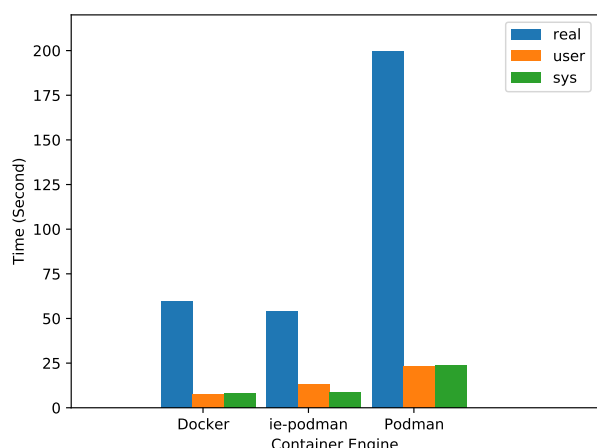


図 1: 書き込み速度の比較

Rootless の Podman はコンテナイメージをユーザのホームディレクトリに保存する。また、rootless では重複排除をサポートしていない VFS ストレージに制限される。Rootless の Podman は独自の名前空間内で特権機能を利用できるようにするため、rootfull と比べ経由する関数が多くなる。そのため、rootless では rootfull と比べ syscall が多く呼ばれることにより、他と比べイメージの Build 速度が遅くなっているのではないかと考える。

8.2 ファイルシステムの評価

旧システムの VM 保存場所として利用していた GFS2、ユーザのホームディレクトリとして利用していた NFS との速度比較を行う。ベンチマークには dd コマンドを使用する。データの変換方法に fdatasync を指定することで、書き込み終了の直前に sync を 1 回要求するため、実際の動作に近い動作で測定が可能である。図 2 は CephFS、Ceph RBD、GFS2、NFS におけるファイルサイズに対する書き込み速度である。

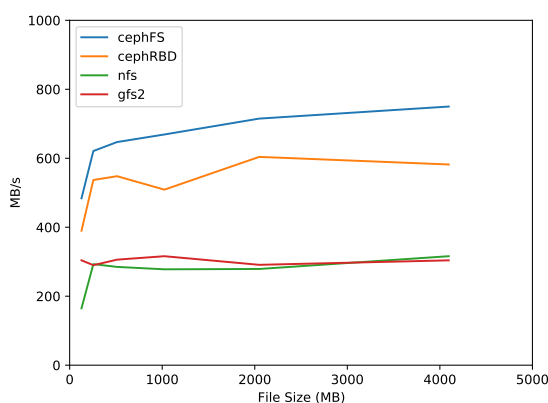


図 2: 書き込み速度の比較

旧システムでは、パッケージ等のアップデートがされておらず、Kernel の更新もされていなかった。Kernel は I/O に関する多くの機能を提供するため、GFS2 の書き込みより、Ceph が高速になったのではないかと考えられる。また、新システムでは、サーバのマシンスペックの他にネットワーク機器も更新されたことも要因だと考えられる。

9 まとめ

今年度のシステム更新で教育情報システムの構築を行い、VM ベースからコンテナベースへの移行ができた。また、学生が利用できる学習環境として VM 貸出サービスだけでなく、コンテナ環境も利用できるようにしたことにより、学生が自由にサーバのリソースを利用できるようになった。コンテナ環境として採用した Podman、Singularity の不便な点を補うために作成した ie-podman の評価を行った。新しく採用した Ceph と、旧システムのファイルシステムとして利用された GFS2、NFS との書き込み速度の比較を行い、速度向上がみられた。

10 今後の課題

旧システムの VM 貸出サービスは講義等で告知されたりしたが、実際にはあまり周知されておらず利用も少なかった。これは、システム管理チームからの利用方法について周知等が少なかったことも原因として挙げられる。本研究で構築した教育情報システムは、VM からコンテナまで利用できる。だが、利用は主に CLI から操作を行い、プログラムの実行には Slurm を利用する。VM 貸出サービスの変更や、コンテナ環境の利用方法についてまとめる必要がある。また、Slurm の Job の投下方法や必要なリソースの要求方法などをまとめ、定期的な周知を行う必要がある。

ie-podman で使用するネットワーク構成はプレフィックス長が 24 であるため、最大 254 個の IP アドレスしか割り当てできない。そのため、コンテナを削除せず停止のままでは、割り当て可能な IP アドレスが枯渇する。そこで、ie-podman が利用するネットワーク構成の変更を行う、もしくはコンテナが停止のまま数日経つ場合に ie-podman から自動削除する必要がある。

参考文献

- [1] Podman, <https://podman.io/>, 2021/1/4.
- [2] Singularity, <https://sylabs.io/singularity/>, 2021/1/8.
- [3] Slurm, <https://slurm.schedmd.com/overview.html>, 2021/1/14.
- [4] Ceph, <https://docs.ceph.com/en/latest/>, 2021/1/12.