令和2年度 卒業論文

コンテナ技術を用いた教育計算機システムの構築

琉球大学工学部工学科知能情報コース

175733E 氏名 宮平 賢

指導教員：河野 真治

目 次

第 1 章　はじめに **1**
1.1 システム管理チーム 1
1.2 論文の構成 1

第 2 章　技術概要 **2**
2.1 仮想化 2
2.1.1 ホスト型 2
2.1.2 ハイパーバイザー型 3
2.1.3 コンテナ型 3
2.2 KVM 4
2.3 Docker 4
2.3.1 Docker Registry 4
2.4 Podman 5
2.5 Singularity 5
2.6 Ceph 5
2.6.1 Ceph Monitor 7
2.6.2 Ceph OSD 7
2.6.3 Ceph Manager 7
2.6.4 Ceph Metadata Server 7
2.7 Ansible 7
2.8 Slurm 7
2.9 GitLab 7
2.10 rsnapshot 7
2.11 Akatsuki 7
2.12 ie-virsh 7

第 3 章　教育計算機システムの構築 **8**
3.1 旧システム 8
3.2 新システム 8
3.2.1 Ceph 8
3.2.2 Podman 8
3.2.3 ie-podman 8
3.2.4 Singularity 8
3.2.5 Slurm 8

第 4 章　教育計算機システムの管理 **9**
4.1　FileSystem の管理 9
4.2　Podman 9
4.2.1　ie-podman の利用方法 9
4.3　Singularity と Slurm を利用した演習 9

第 5 章　まとめ **10**
5.1　今後の課題 10

図 目 次

2.1 ホスト型 2
2.2 ハイパーバイザー型 3
2.3 コンテナ型 3
2.4 Docker 4
2.5 Podman 5
2.6 Ceph のアーキテクチャ 6

表 目 次

ソースコード目次

第1章　はじめに

1.1　システム管理チーム

1.2　論文の構成

第2章　技術概要

本章では, 本研究で使われる技術, 本コースで利用しているサービスについて概要を説明する。

2.1　仮想化

仮想化はコンピュータの CPU やメモリ, ディスクなどハードウェアのリソースを分割又は統合して, 仮想的なコンピュータやネットワーク環境を生成し提供する技術である。仮想化技術にはホストのどの部分から仮想化するかによってホスト型, ハイパーバイザー型, コンテナ型に分けることができる。

2.1.1　ホスト型

ホスト型の仮想化は, ホストとなる OS 上 (以下, ホスト OS) に仮想化ソフトウェアをインストールし, 仮想化ソフトウェア上で別の OS (以下, ゲスト OS) を稼働させる手法である (図 2.1)。仮想化ソフトウェアをホスト OS のアプリケーションの 1 つとして導入及び管理できるため, 手軽に仮想化を実現することができる。しかし, ゲスト OS の処理はホスト OS を経由しなければならないため, オーバーヘッドが大きくなる。

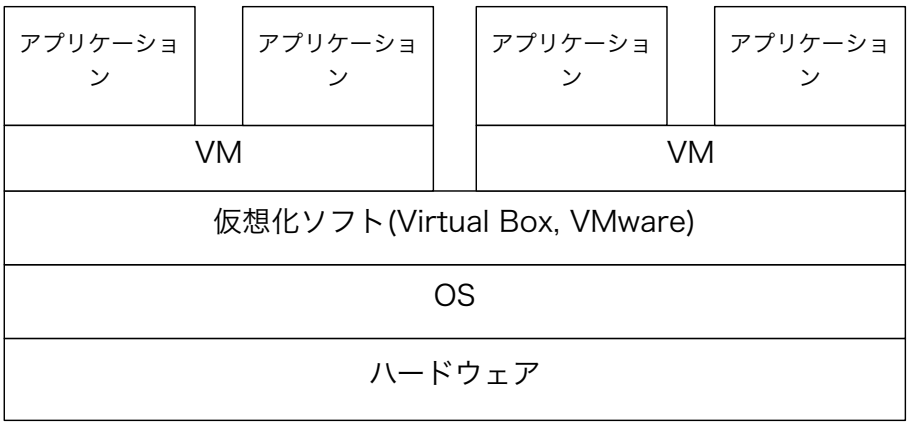

図 2.1: ホスト型

2.1.2　ハイパーバイザー型

ハイパーバイザー型の仮想化は, 仮想化システムを直接ハードウェアにインストールし, ハイパーバイザー上で複数のゲスト OS を稼働させる手法である (図 2.2)。ハイパーバイザーが直接ハードウェアを管理するため仮想化によるオーバーヘッドを小さくすることで, リソースを効率的に利用することができる。

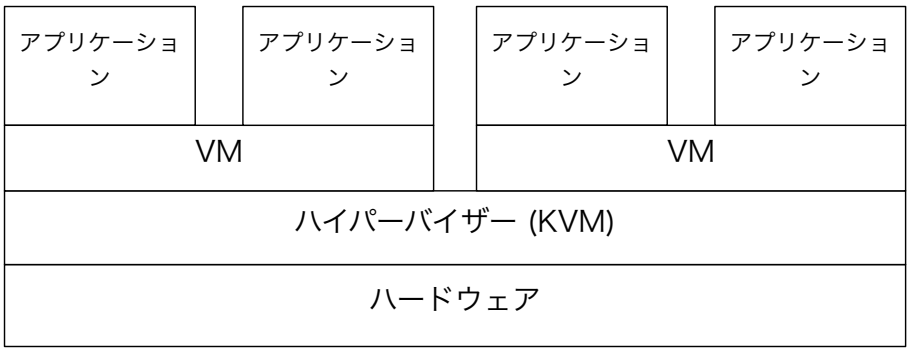

図 2.2: ハイパーバイザー型

2.1.3　コンテナ型

コンテナ型の仮想化は, OS レベルの仮想化技術を利用して複数のコンテナと呼ばれる独立空間を形成し, 独立空間でアプリケーションをそれぞれ構築することができる手法である (図 2.3)。各コンテンはオペレーティングシステムカーネルによって独立したプロセスとして実行される。前述のホスト型やハイパーバイザー型と比べ, コンテナはゲスト OS を起動することなくアプリケーションを実行することができるため, リソース効率が良く処理が軽量である。

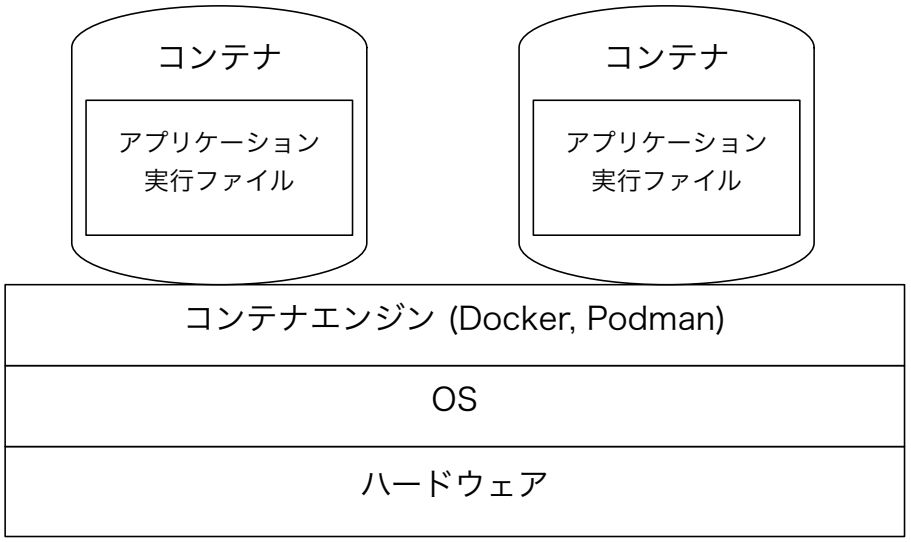

図 2.3: コンテナ型

2.2 KVM

KVM (Kernel-based Virtual Machine)[1] は Linux カーネル 2.6.20 以降に標準搭載されているハイパーバイザーである。KVM は Intel VT 及び AMD-V を含む x86 ハードウェア上の完全仮想化をサポートしている。KVM はハイパーバイザーと各仮想マシン間のレイヤーとして Virtio API を使用して, 仮想マシンに準仮想化デバイスを提供する。これにより, 仮想化によるオーバーヘッドを少なくできる。

2.3 Docker

Docker[2] は Docker 社が開発, 提供する Linux 上で動作する隔離された Linux コンテナをデプロイ, 実行するアプリケーションである。Docker はコンテナを実行するだけでなく, コンテナイメージの作成や共有する仕組みも提供している。Docker コマンドを処理するには Docker daemon と呼ばれるデーモンプロセスを実行する必要がある。この Docker deamon は Docker で行う処理を一箇所で実施する (図 2.4)。

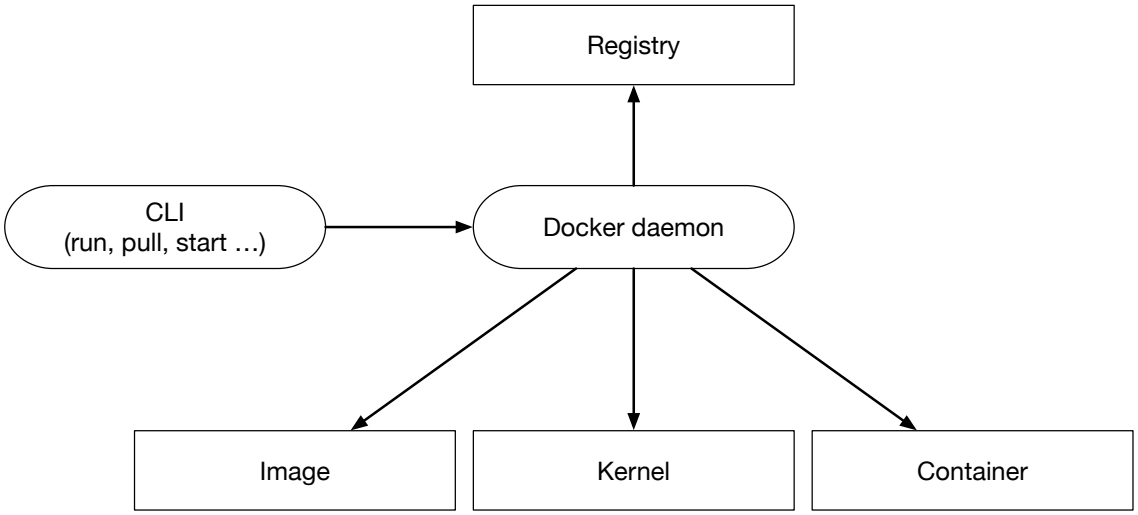

図 2.4: Docker

2.3.1 Docker Registry

Docker Registry は Dcoker イメージを保存, 配布できるサーバサイドアプリケーションである [3]。以下の場合に利用される。

- イメージの保存場所を厳密に管理する
- イメージを配布するパイプラインを全て所有する
- イメージの保存と配布を社内や学内の開発ワークフローに密に統合する

2.4 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 [9]。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 Linxu 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 (図 2.5)。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。

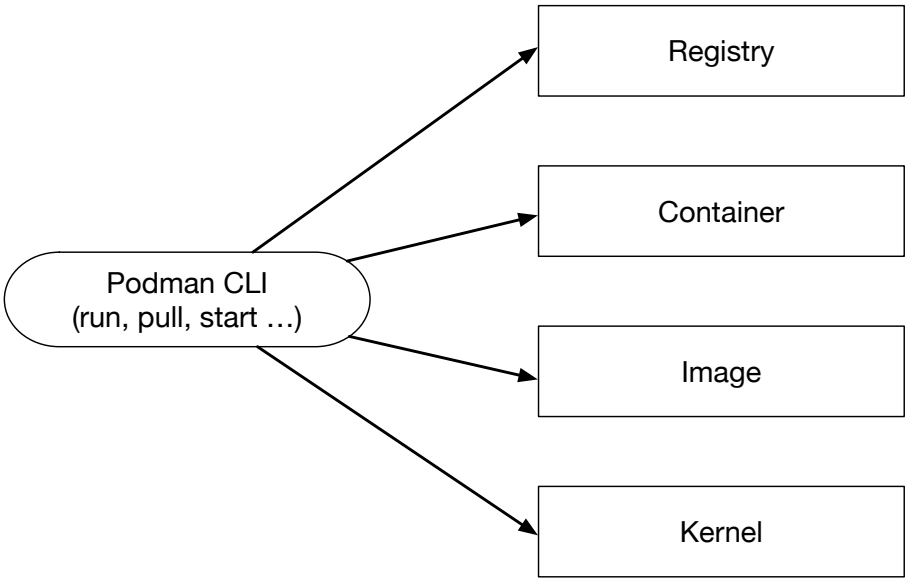

図 2.5: Podman

2.5 Singularity

Singularity[4] 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。

2.6 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 (図 2.6)。オブジェクトストレージはデータをオブジェクトという単位でやり取りをするストレージシステムである。複数のストレージを束ねて利用できるオブジェクトストレージが分散オブジェクトストレージである。RAODS では, Object Storage Daemon (OSD) にデータ格納する。オブジェクトの配置には, クラスタマップを元に Controlled Replication Under Scalable Hashing (CRUSH) アルゴリズムによりオブジェクトの格納先を選択する。配置の計算に必要とする情報はごくわずかであるため, Ceph クラスタ内のすべてのノードは保存されている位置を計算できる。そのため, データの読み書きが効率化される。また, CRUSH はデータをクラスタ内のすべてのノードに均等に分散しようとする。

RODOS 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 などで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。

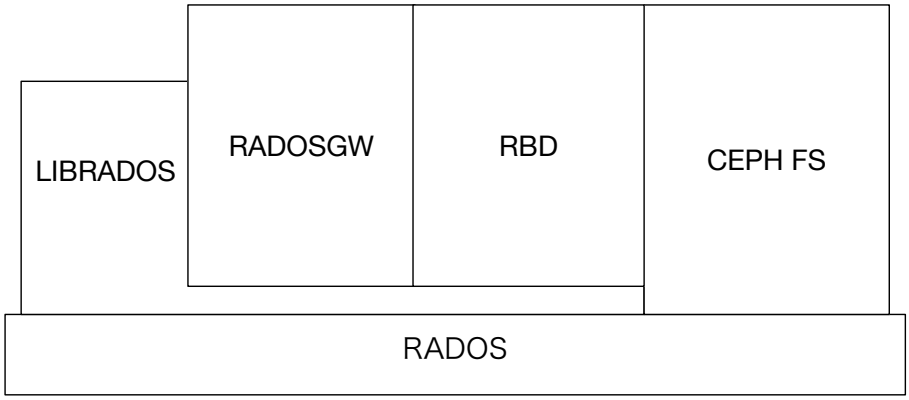

図 2.6: Ceph のアーキテクチャ

Ceph では, ノードとはクラスタを構成するサーバであり, ノードでは以下の 4 つのデーモンが実行できる。

- Ceph Monitor
- Ceph OSD
- Ceph Manager
- Ceph Metadata Server

2.6.1 Ceph Monitor

Ceph Monitor (以下, MON) ノードはクラスタのヘルス状態に関する情報, データ分散ルールを維持する。障害が発生した場合, クラスタ内の MON ノードで Paxos という合意アルゴリズムを使用して, どの情報が正しいかを多数決で決定する。そのため, 奇数個の MON ノードを設定する必要がある。

2.6.2 Ceph OSD

Ceph OSD (以下, OSD) は物理ストレージになる。このデーモンは 1 台の HDD などの物理ストレージに対して, 1 つのデーモンが動作する。OSD は MON と通信し, OSD デーモンの状態を提供する。

2.6.3 Ceph Manager

Ceph Manager (以下, MGR) ノードはクラスタ全体から状態情報を収集する。MGR は MON と共に動作し, 外部のモニタリングシステムや管理システムのインターフェースとして機能する。

2.6.4 Ceph Metadata Server

Ceph Metadata Server (以下, MDS) ノードは CephFS のメタデータを保存する。

2.7 Ansible

Ansible[6] は RedHat 社が開発, 提供するシステム構成, ソフトウェアの展開などを行う自動化ツールである。Ansible では一連の処理を YAML 形式のファイルで記述し, エージェントレスで SSH を介してタスクを実行する。そのため, インフラストラクチャをコードとして残すことができる。

2.8 Slurm

2.9 GitLab

2.10 rsnapshot

2.11 Akatsuki

2.12 ie-virsh

第3章　教育計算機システムの構築

3.1　旧システム

3.2　新システム

3.2.1　Ceph

3.2.2　Podman

3.2.3　ie-podman

3.2.4　Singularity

3.2.5　Slurm

第4章　教育計算機システムの管理

本章では, 構築した教育計算機システムの管理の方法, 利用方法について述べる。

4.1　FileSystem の管理

4.2　Podman

4.2.1　ie-podman の利用方法

4.3　Singularity と Slurm を利用した演習

第5章　まとめ

5.1　今後の課題

参考文献

[1] KVM，https://www.linux-kvm.org/，2021/1/8.

[2] Docker，https://www.docker.com/，2021/1/8.

[3] Docker Registry，https://docs.docker.com/registry/，2021/1/8.

[4] Singularity，https://sylabs.io/singularity/，2020/9/11.

[5] Ceph，https://docs.ceph.com/en/latest/，2021/1/12.

[6] Ansible，https://www.ansible.com/，2021/1/12.

[7] 平良 太貴 and 河野 真治，OS 授業向けマルチユーザ VM 環境の構築，研究報告システムソフトウェアとオペレーティング・システム (OS)(2014).

[8] 城戸翔太，安里悠矢，城間政司，長田智和，谷口祐治，”情報系学科における教育情報システムの構築及び運用管理に関する取り組み”，研究報告インターネットと運用技術 (IOT)(2016).

[9] Podman，https://podman.io/，2021/1/4.

謝辞

感謝します。

2021 年 2 月
宮平 賢