

仮想化技術とクラウドコンピューティング

AWS 第1回

吉川 恭平 塩澤 康志 福島 康記 溝口 洸熙

情報学群実験第4 Group C5

July 24th, 2023

お品書き

1. 仮想化技術と仮想マシン
 - 仮想化と仮想マシン
 - 仮想化の歴史と展望
 - メリットとデメリット
2. 仮想化の方式
 - ホスト型
 - ハイパーバイザ型
 - コンテナ型
3. Docker (Option)
 - コンテナ立ち上げと利用
 - イメージ作成・取得
 - 実演
4. 準仮想化と完全仮想化
5. クラウドコンピューティング
 - クラウドとは
 - クラウドの分類
 - 代表的な IaaS
6. AWS
 - AWS とは
 - オンプレミスと AWS
 - レンタルサーバと AWS
 - AWS 導入例

1. 仮想化技術と仮想マシン

1. 仮想化技術と仮想マシン

- 仮想化と仮想マシン
- 仮想化の歴史と展望
- メリットとデメリット

2. 仮想化の方式

- ホスト型
- ハイパーバイザ型
- コンテナ型

3. Docker (Option)

- コンテナ立ち上げと利用
- イメージ作成・取得
- 実演

4. 準仮想化と完全仮想化

5. クラウドコンピューティング

- クラウドとは
- クラウドの分類
- 代表的な IaaS

6. AWS

- AWS とは
- オンプレミスと AWS
- レンタルサーバと AWS
- AWS 導入例

1.1 仮想化と仮想マシン

仮想化

“コンピューターの物理的資源を論理的に分割して、それぞれ独立並列した状態で利用できるようにすること。1台のサーバーで、複数の基本ソフトを独立並列に動作させるサーバー仮想化など。”[1]

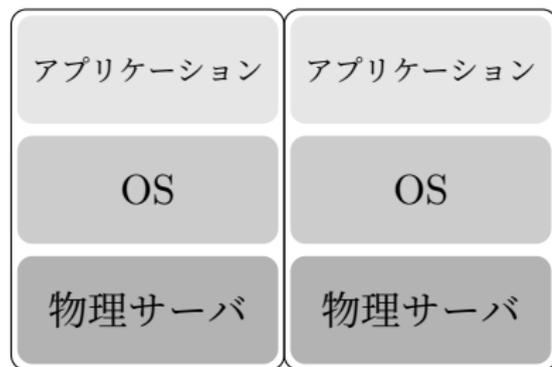
—▶ 1台の計算機で複数のOSやアプリケーションなどを並列に動作させること。

Virtual Machine（仮想機械，仮想マシン）

“あるコンピューターシステムの動作を、別システムで再現するソフトウェア。また、そのような動作環境。あるOSの動作を別のOS上で再現する場合など。バーチャルマシン。VM。” [1]

—▶ 仮想化するためのソフトウェアや動作環境。

1.1 仮想化と仮想マシン

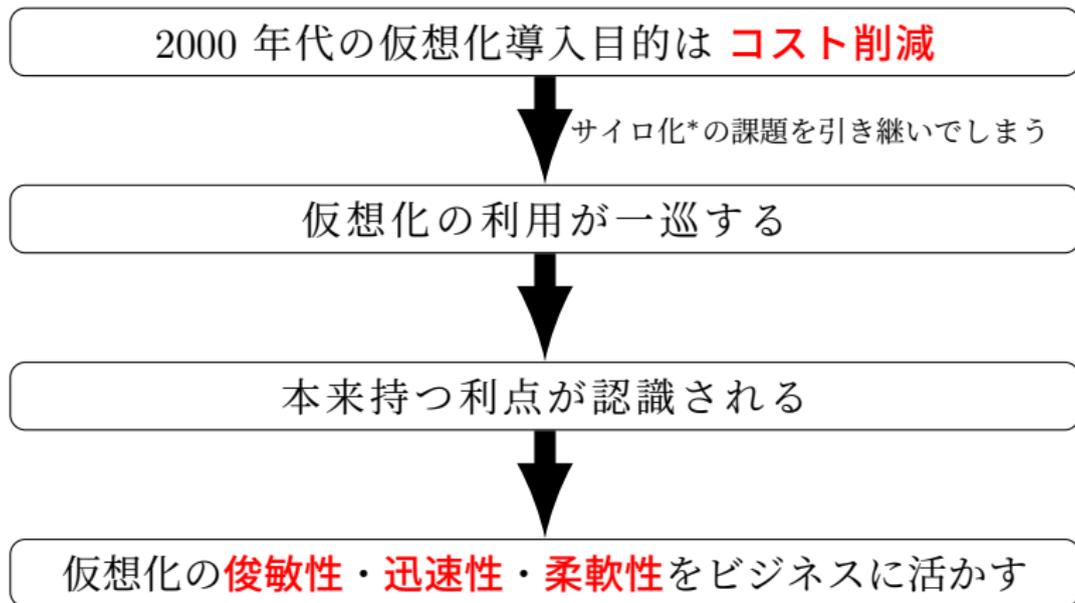


仮想化前



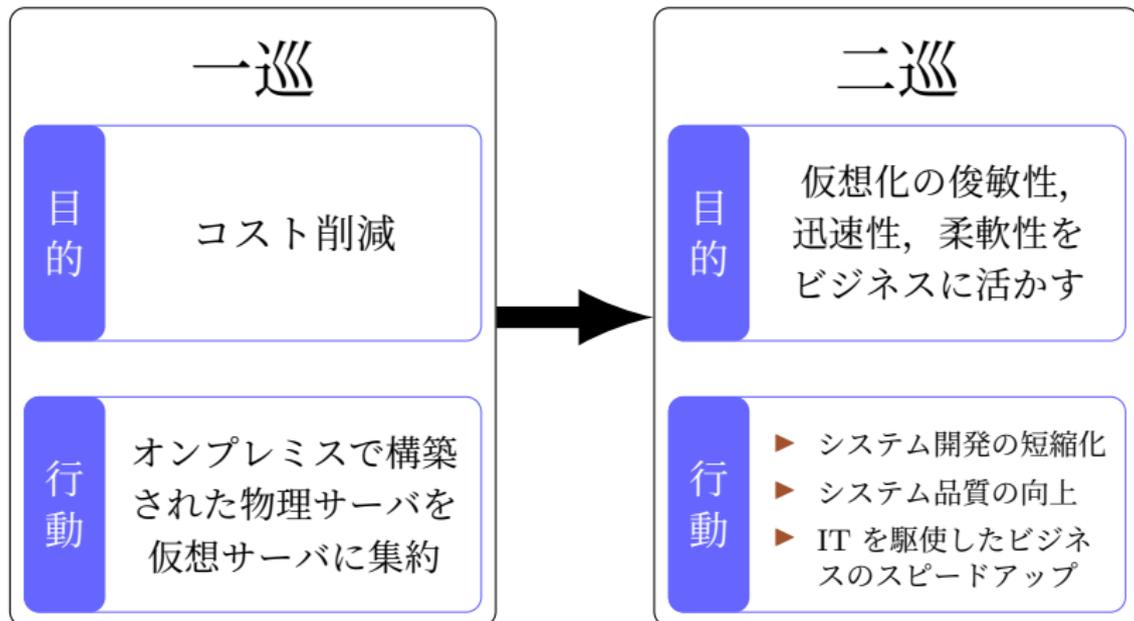
仮想化後

1.2 仮想化の歴史と展望



*複数のサーバが乱立し、他のシステムとの連携が取れない状態。

1.2 仮想化の歴史と展望



1.3 メリットとデメリット

✓ メリット

(仮想化導入の目的)

- ▶ 無駄なリソースの削減
- ▶ コスト削減
 - ⊕ ハードウェアを減らせる
- ▶ リソースを集約できる
 - ⊕ 管理が簡便になる

✗ デメリット

- ▶ 物理サーバに比べて性能が劣る.
- ▶ 物理層の故障による障害範囲が広がる.

2. 仮想化の方式

1. 仮想化技術と仮想マシン

- 仮想化と仮想マシン
- 仮想化の歴史と展望
- メリットとデメリット

2. 仮想化の方式

- ホスト型
- ハイパーバイザ型
- コンテナ型

3. Docker (Option)

- コンテナ立ち上げと利用
- イメージ作成・取得
- 実演

4. 準仮想化と完全仮想化

5. クラウドコンピューティング

- クラウドとは
- クラウドの分類
- 代表的な IaaS

6. AWS

- AWS とは
- オンプレミスと AWS
- レンタルサーバと AWS
- AWS 導入例

- └ 仮想化の方式
 - └ ホスト型

2.1 ホスト型

ハードウェアの中の OS 上に、土台となる仮想ソフトウェアをインストールし、仮想化ソフトウェアで仮想マシンを稼働させる。



2.1 ホスト型

ホスト型仮想化ソフトウェア 例

- ▶ VMware Workstation Player
- ▶ VMware Fusion
- ▶ Oracle VM Virtualbox



Virtualbox[3]

✓ メリット

- ▶ 既存マシンが利用できる.
- ▶ 仮想化に必要なソフトウェアが扱いやすい.

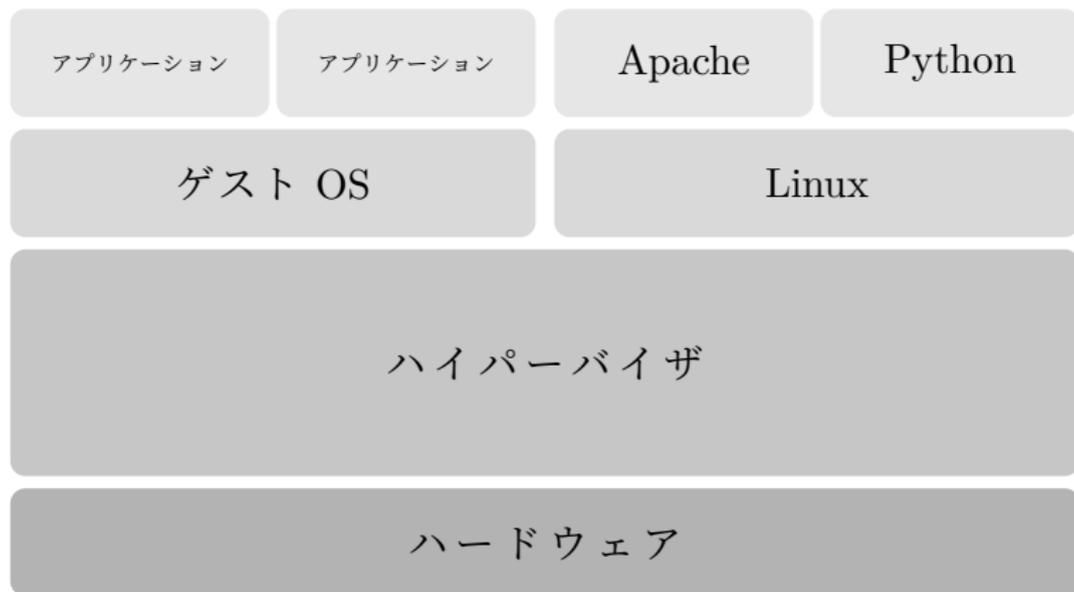
✗ デメリット

- ▶ ホスト OS を動作させるための物理リソースが必要.

[4]

2.2 ハイパーバイザ型

ハイパーバイザとは「仮想化のための OS」のようなもの。ホスト OS を必要としない、仮想化ソフトウェア。



2.2 ハイパーバイザ型

ハイパーバイザ 例

- ▶ VMware ESXi
- ▶ Linux KVM
- ▶ Microsoft Hyper-V

✓ メリット

- ▶ ホスト型に比べて、システム全体の観点から見てリソースの使用効率がよい。
- ▶ 物理サーバに比べて、運用にかかるコストを削減できる。

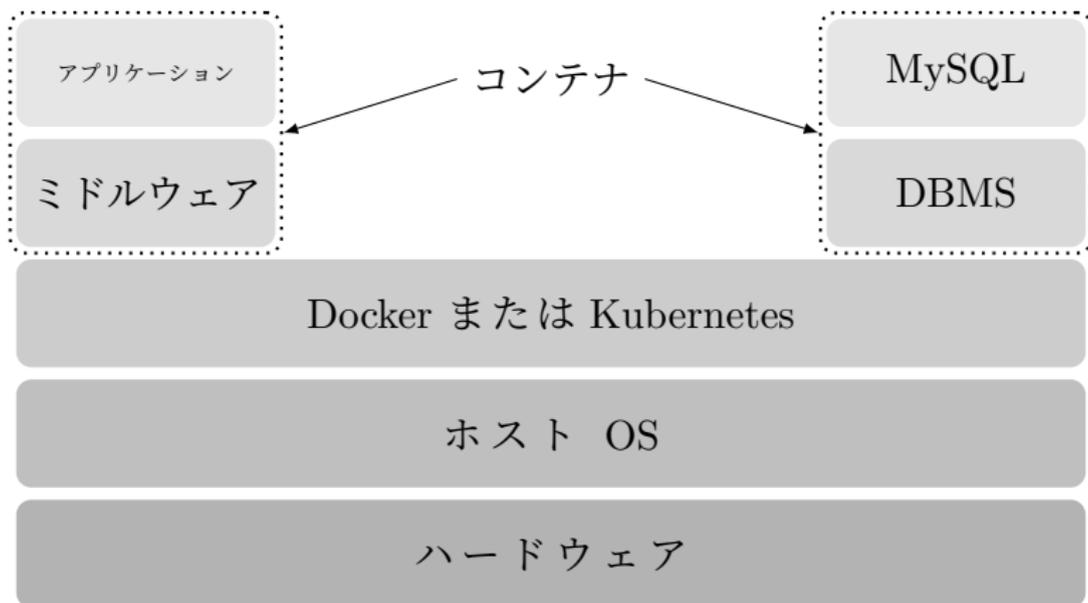
✗ デメリット

- ▶ 物理サーバに比べて、性能が劣る。
- ▶ 物理サーバに比べて、障害の範囲は大きくなる（ことがある。）

- └ 仮想化の方式
 - └ コンテナ型

2.3 コンテナ型

“アプリケーションを実行するための領域（ユーザ空間）を複数に分割して利用するもの”[4].



2.3 コンテナ型

ミドルウェア

システムソフトウェアは、基本ソフトウェア（OS）とミドルウェアに分類される。ミドルウェアはOSとアプリケーションソフトウェアの中立ちをする。 [5]

ミドルウェアの例

データベースを管理する、データベースを操作する基本ソフトウェア（DBMS*）がある。

*Data Base Management System

2.3 コンテナ型

✓ メリット

- ▶ バージョン依存が激しいもの (Python, JavaScript) などでも指定した環境を再現しやすい.
(Dockerfile)
- ▶ 環境構築が簡単.
- ▶ ゲスト OS が不要なので、アプリケーションの起動や処理が高速.

✗ デメリット

- ▶ 同一基盤上で異なる OS を動かせない.
- ▶ ホスト OS で障害が生じると、すべてのコンテナに影響が出る.

3. Docker (Option)

1. 仮想化技術と仮想マシン

- 仮想化と仮想マシン
- 仮想化の歴史と展望
- メリットとデメリット

2. 仮想化の方式

- ホスト型
- ハイパーバイザ型
- コンテナ型

3. Docker (Option)

- コンテナ立ち上げと利用
- イメージ作成・取得
- 実演

4. 準仮想化と完全仮想化

5. クラウドコンピューティング

- クラウドとは
- クラウドの分類
- 代表的な IaaS

6. AWS

- AWS とは
- オンプレミスと AWS
- レンタルサーバと AWS
- AWS 導入例

3.1 コンテナ立ち上げと利用

▶ コンテナを利用する.

1. イメージを作成 (build) または, 取得 (docker pull) する.
イメージの作成には Dockerfile を書く.
2. コンテナを作成して起動 (run) する.
3. コンテナ内でコマンドを実行 (exec) する.

▶ コンテナ利用後.

1. コンテナを停止 (stop) する.
2. コンテナを削除 (rm) する.
3. イメージを削除 (rmi) する.

3.2 イメージ作成・取得

▶ Dockerfile

```
1 FROM mysql:latest
```

```
$ docker build ./ -t mysql
```

▶ レジストリからイメージを取得 (Pull) する.

```
$ docker pull mysql:latest
```

3.3 実演



[6]

4. 準仮想化と完全仮想化

1. 仮想化技術と仮想マシン

- 仮想化と仮想マシン
- 仮想化の歴史と展望
- メリットとデメリット

2. 仮想化の方式

- ホスト型
- ハイパーバイザ型
- コンテナ型

3. Docker (Option)

- コンテナ立ち上げと利用
- イメージ作成・取得
- 実演

4. 準仮想化と完全仮想化

5. クラウドコンピューティング

- クラウドとは
- クラウドの分類
- 代表的な IaaS

6. AWS

- AWS とは
- オンプレミスと AWS
- レンタルサーバと AWS
- AWS 導入例

4. 準仮想化と完全仮想化

ホスト OS 型とハイパーバイザ型を比較すると、性能的にはハイパーバイザ型が有利。ハイパーバイザの中にも**完全仮想化**と**準仮想化**がある。

完全仮想化

ハードウェアも含めて、すべてを仮想化する方式。ハードウェアも論理ハードウェア。

準仮想化

ゲスト OS を少し改造して、ゲスト OS から直接ハードウェアを操作できる方式。処理速度が大きくなる。（理由は次頁。）

 [詳しい説明を飛ばす](#)

4. 準仮想化と完全仮想化（詳しく）

完全仮想化

ハイパーバイザ上で動作するゲスト OS の中身に何も変更を加えないで動作させる方式。ゲスト OS からハードウェアを操作する（特権命令の実行）は許されないので、ハイパーバイザで処理を変換する。

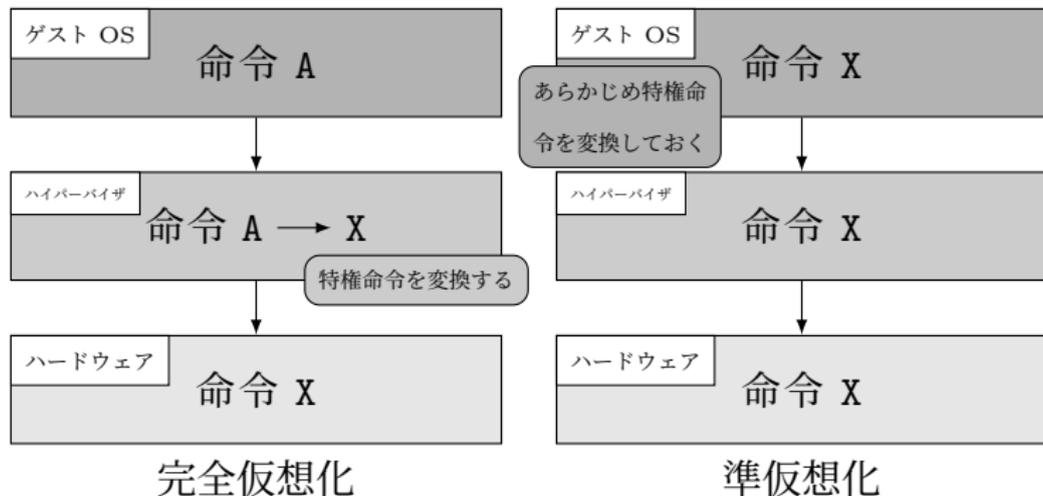
準仮想化

ゲスト OS に変更を加えて性能向上を図った方式。ゲスト OS からハードウェアを操作する部分（特権命令）を、ハイパーバイザに直接処理が渡るように変更を加える。ハイパーバイザでの命令変換時にかかるオーバーヘッドを削減する

4. 準仮想化と完全仮想化（詳しく）

命令 A：ゲスト OS からの特権命令*.

命令 X：ハイパーバイザからの特権命令.



*CPU のステータスを変更するといった、特別な権限を与えられたプログラムしか使用できない命令.

5. クラウドコンピューティング

1. 仮想化技術と仮想マシン

- 仮想化と仮想マシン
- 仮想化の歴史と展望
- メリットとデメリット

2. 仮想化の方式

- ホスト型
- ハイパーバイザ型
- コンテナ型

3. Docker (Option)

- コンテナ立ち上げと利用
- イメージ作成・取得
- 実演

4. 準仮想化と完全仮想化

5. クラウドコンピューティング

- クラウドとは
- クラウドの分類
- 代表的な IaaS

6. AWS

- AWS とは
- オンプレミスと AWS
- レンタルサーバと AWS
- AWS 導入例

5.1 クラウドとは

クラウドコンピューティング

クラウドコンピューティング（以下、クラウド）とは、コンピュータリソース尾の利用形態。コンピュータの計算リソースやストレージ領域、アプリケーションによる処理をネットワーク経由で提供する。 [8]

クラウドは以下の4つに分類されることが多い。

- ▶ IaaS (Infrastructure as a Service)
- ▶ PaaS (Platform as a Service)
- ▶ SaaS (Software as a Service)
- ▶ DaaS (Desktop as a Service)

5.2 クラウドの分類

▶ IaaS

仮想サーバやストレージ，ネットワークサービスなどをインターネット経由で提供する。

(AWS, Azure, GCP*, IBM Cloud など)

▶ PaaS

ミドルウェアをサービスとして提供する。OS とミドルウェアの管理はサービス提供者で行われ，ユーザはミドルウェアを直接操作できる。

(AWS, Azure, GCP など)

▶ SaaS

ソフトウェアをサービスとして提供する。

(Slack, Zoom, Google Calendar など)

*Google Cloud Platform

5.2 クラウドの分類

▶ DaaS

“クラウドサービスとして提供される仮想デスクトップ環境を組織として利用するしかけ。” [9]
(Azure Virtual Desktop など)

▶ VDI (Virtual Desktop Infrastructure)

“クライアント PC のデスクトップサーバ上に仮想化して集約し、稼働させる仕組み。” [10]

! DaaS は VDI の一種である。

5.3 代表的な IaaS

- ▶ AWS
- ▶ Microsoft Azure
- ▶ Google Cloud Platform
- ▶ さくらのクラウド（さくらインターネット）

6. AWS

1. 仮想化技術と仮想マシン

- 仮想化と仮想マシン
- 仮想化の歴史と展望
- メリットとデメリット

2. 仮想化の方式

- ホスト型
- ハイパーバイザ型
- コンテナ型

3. Docker (Option)

- コンテナ立ち上げと利用
- イメージ作成・取得
- 実演

4. 準仮想化と完全仮想化

5. クラウドコンピューティング

- クラウドとは
- クラウドの分類
- 代表的な IaaS

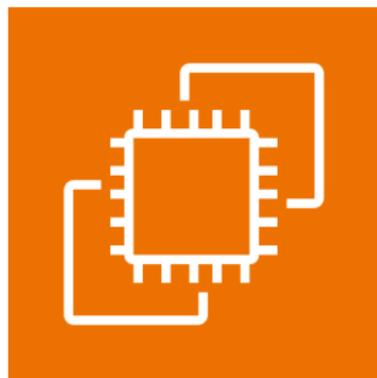
6. AWS

- AWS とは
- オンプレミスと AWS
- レンタルサーバと AWS
- AWS 導入例

6.1 AWS とは

AWS

Amazon Web Service の略称。Amazon が提供するクラウドサービスで、ネットワークを経由して仮想コンピュータやストレージなどのサービスを提供している。 [8]



Amazon Elastic Compute
Cloud (Amazon EC2)[11]



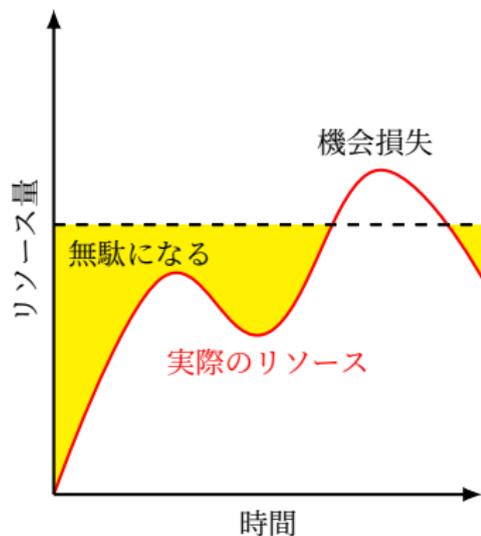
Amazon Simple Storage
Service (Amazon S3)[11]



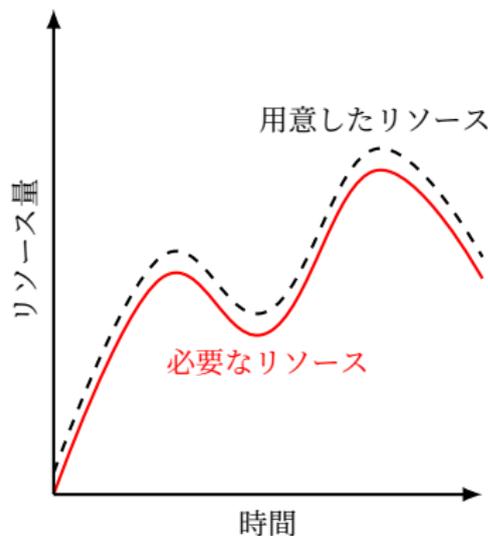
Amazon Relational Database
Service (Amazon RDS)[11]

6.2 オンプレミスと AWS

AWS はオンプレミスに比べてリソースを効率よく運用できる。[8]



オンプレミス



AWS の利用

6.3 レンタルサーバと AWS

レンタルサーバ

“1 台のサーバを複数のユーザが共用で利用する形態” [8]

レンタルサーバの利用形態

	共用サーバ	専用サーバ	仮装専用サーバ
利用形態	1 台の物理サーバを分割して利用	1 台の物理サーバを占有	1 台の物理サーバ上に立てた仮想サーバを利用
コスト	安い	高い	中間
カスタマイズ	ほぼ不可能	可能	可能
セキュリティ	コントロール不可	高くコントロール可能	高くコントロール可能
他のユーザの影響	強く受ける	受けない	多少受ける

引用 [8]

6.3 レンタルサーバと AWS

AWS EC2

EC2 は 1 台の物理マシン上に、複数の仮想コンピュータを立ち上げて使い、管理者権限 (root) を持ったアカウントを使用し、その仮想コンピュータ内のすべてを管理する。 [8]

「EC2」とレンタルサーバの「仮想専用サーバ」は似たサービス。しかし...

EC2 の特徴

- ▶ CPU やメモリを容易に変更できる。
- ▶ ディスクを動的に追加*できる。

これまでの
レンタルサーバにはない!

等々...

*インスタンスの再起動なしに変更すること。

6.4 AWS 導入例

- ▶ Cloud LaTeX（アカリク）
 - ▶ コンパイルなど（EC2）
 - ▶ プロジェクト，テンプレート管理（S3，MySQL）
- ▶ 株式会社任天堂
 - ▶ 『マリオカート ツアー』のDBに，「Amazon Aurora」（RDBMS）を採用。
- ▶ 株式会社東京海上日動火災保険
- ▶ JCB
- ▶ 盛岡市
- ▶ 浜松市
- ▶ SUBARU

Reference I

- [1] 三省堂編.
スーパー大辞林.
三省堂, 2020.
- [2] NTT コムウェア株式会社.
【前編】仮想化の歴史と展望.
https://www.nttcom.co.jp/comware_plus/img/201706_guest_interview.pdf, Last confirmed date: July 12th, 2023.
- [3] ORACLE.
Welcome to virtualbox.org!
<https://www.virtualbox.org>, Last confirmed date: July 7th, 2023.

Reference II

- [4] アイティーエム株式会社.
仮想化技術について解説.
<https://www.itmanage.co.jp/column/virtualization-server-integration/>, Last confirmed date: July 6th, 2023.
- [5] 國友義久.
IT の基礎 (ファーストステップ) .
ファーストステップ. 近代科学社, 2011.
- [6] Docker Inc.
Docker logos.
<https://www.docker.com/company/newsroom/media-resources/>, Last confirmed date: July 7th, 2023.

Reference III

- [7] 特定非営利活動法人 IT 整備士協会.
第 3 章 仮想化技術.
https://kasouka.jp/Chap03_1_1.pdf, Last confirmed date: July 10th, 2023.
- [8] 佐々木拓郎, 林晋一郎, 小西秀和, 佐藤瞬.
Amazon Web Services パターン別構築・運用ガイド: 一番大切な知識と技術が身につく.
SB クリエイティブ, 2015.
- [9] 大場みち子.
IS デジタル辞典 [DaaS].
一般社団法人情報処理学会. 情報システムと社会環境研究会編 <https://ipsj-is.jp/isdic/4297/>, Last confirmed date: July 7th, 2023.

Reference IV

- [10] 高橋尚子.
IS デジタル辞典 [VDI].
一般社団法人情報処理学会. 情報システムと社会環境研究会編 <https://ipsj-is.jp/isdic/4300/>, Last confirmed date: July 7th, 2023.
- [11] Amazon Web Services Inc.
AWS アーキテクチャアイコン.
<https://aws.amazon.com/jp/architecture/icons/>,
Last confirmed date: July 8th, 2023.

Reference V

- [12] Amazon Web Services Inc.
お客様のクラウド導入事例.
<https://aws.amazon.com/jp/solutions/case-studies/>, Last
confiremd date: July 12th 2023.