5D-06

コンテナ型仮想化機構を用いた大学向けハイブリッドクラウドの構築

田中 遼 †　　田胡 和哉 ‡
† 東京工科大学大学院 バイオ・情報メディア研究科 コンピュータサイエンス専攻
‡ 東京工科大学 コンピュータサイエンス学部

1 背景

近年，クラウド・コンピューティングを用いたサービスが数多く提供されている．その急激な広まりは教育機関にもおよび，本学においても学生が主体となって開発・構築・運用を行うプライベートクラウド施設「クラウドサービスセンター」を開設し，全学生・教員に対してサービスインを行った [1].

通常，大学等の教育機関でのオンプレミス型のインフラ運用は負荷の規模がある程度予測可能であり，その規模に合わせたインフラ環境の構築を行う事が可能である．しかし，教育機関では負荷のほとんどが授業中，および授業期間中に集中しており，その他の時間はリソースの余剰が発生してしまう点が問題となっている．

本稿では，本学プライベートクラウド施設のインフラ構築・運用より得た知見から，次世代の大学向けクラウド環境を考察し，開発した機構について述べる．

2 ハイブリッドクラウド機構 – Juneau

2.1 アプローチ

現状の大学プライベートクラウドの問題点を解消する為，軽量な仮想化技術として注目されているコンテナ型仮想化機構を用いてパブリッククラウドとプライベートクラウドを連携する事により，双方に負荷を分散するハイブリッドクラウド機構を開発した．

通常，異なるデータセンター間を連携させる上で最も問題となるのがセキュアなデータ共有の手法である．従来では，各データセンター間を拠点間 VPN 等で接続し，擬似的なローカルネットワークを作成する事によって，NFS 等のファイル共有システムを用いてデータ共有を行うケースが多い．しかし，従量課金が一般的であるパブリッククラウドとの連携を考えた場合，常にパケットが流れ続ける事によってコストが高くなってしまう上，パブリッククラウド側に拠点間 VPN 等を接続する設備が必要となる．また，動作させるアプリケーションの特性によっては常にデータを同期する必要がないケースも多い．本機構では，トンネリング技術とレイヤ型ファイルシステムを用いる事によって，擬似的なローカルネットワークを構築する事なく，任意のタイミングでの差分データの保存を可能とする．

本機構により，オンプレミスでは最小限のインフラ環境を確保し，負荷が増加する期間のみパブリッククラウドとの連携を行う事によって，高負荷なアプリケーションを低コストで提供可能となる．なお，本ハイブリッドクラウド機構を “Juneau” と呼称する．

2.2 設計

2.2.1 Docker

Docker とは，コンテナ型の仮想化機構である [2]. Docker では，ユーザランドのみを仮想化しており，ハイパーバイザ型の仮想化と異なり，ハードウェア仮想化等のオーバーヘッドが生じない為，起動・終了が非常に高速という特徴がある．また，ファイルシステムにレイヤ型方式である Aufs を採用しており，Dockerfile* および Dockerhub† と組み合わせる事によって，非常に高いポータビリティを実現している．

2.2.2 アーキテクチャ

Juneau のアーキテクチャを図 1 に示す．本機構はコアなプロセスを集約する “Juneau Master”，実際にコンテナを動作させる “Juneau Worker” の 2 つのサーバから構成され，これらは一対多の関係を持つ．

Juneau では，Worker とするサーバの動作条件をシンプルにする為，トンネリングによる通信経路の確保を行っており，グローバル IP アドレスを持ってさえいれば，特定のクラウドプラットフォームに依存せずに Worker を追加する事が可能となっている．また，データ共有の問題点を解決する為，コンテナのレイヤ型ファイルシステムに差分レイヤとしてユーザデータをコミットし，Master サーバに転送するという手法を用いている．これにより，常時ファイルの同期を取る必要がなく，任意のタイミングでデータのコミットを行う事が可能となり，Master サーバに転送されるデータ量もゴールデンイメージからの差分のみとなる．

Juneau は，主に次に示す機能によって構成される．

Building hybrid cloud using a container-based virtualization mechanism
†Ryo TANAKA　‡Kazuya TAGO
†Computer Science Major, Graduate School of Bionics,Computer and Media Sciences, Tokyo University of Technology
‡Department of Computer Science, Tokyo University of Technology
(†‡) Katakura-cho Hachiohji-shi Tokyo 192-0982, Japan

*コンテナイメージの構成情報を記述したテキストファイル
†作成したコンテナイメージを保存可能なレジストリ

Copyright ©2015 Information Processing Society of Japan.
All Rights Reserved.

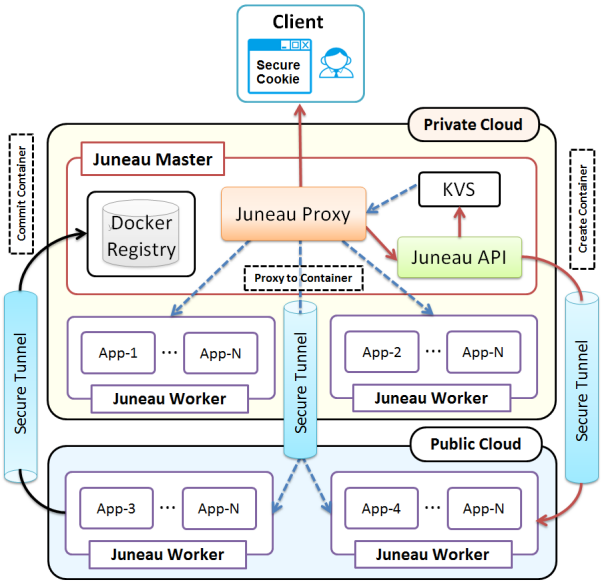

図 1: Juneau のアーキテクチャ

- Juneau Master
  - Docker Registry : コンテナイメージを保存するリポジトリ
  - Juneau Proxy : クライアントからのアクセスを受け付け，コンテナへプロキシする動的リバースプロキシ
  - Juneau API : スケジューラを内包し，コンテナの起動等を行う API
- Juneau Worker
  - Docker Daemon : 実際にコンテナを動作させるデーモン

Juneau Proxy はユーザからのアクセスを受けると Secure Cookie からユーザを判定し，当該ユーザのコンテナが動作していない場合は Juneau API を呼び出し，コンテナの起動を依頼する．Juneau API はコンテナの起動要求を受け，内部のスケジューラからコンテナを実行するホストを決定し，Docker Remote API を用いてコンテナの起動を行い，コンテナへの接続情報を呼び出し元に通知する．Juneau Proxy は通知された接続情報を元にアクセスのプロキシ先を切り替える事で，起動したコンテナにアクセスをプロキシする．コンテナ起動後は，常に当該ユーザのコンテナにプロキシが行われる．

データの保存は Juneau API にコンテナの保存要求を送信する事で行われる．保存要求を受けると，当該ユーザのコンテナを一時的に停止状態とし，ファイルシステムの差分レイヤを作成し，再び起動状態とする．その後，取得した差分レイヤをリバーストンネリングを用いて Juneau Master 上の Docker Registry に転送する事でデータの永続化が行われる．

表 1: Web IDE の評価実験結果

項目	結果 (平均)
動作速度	4.33 (5 段階 – 5 が最高)
コンテナの初回起動時間	4697ms
コンテナの保存処理時間	3755ms

3 ウェブベース統合開発環境における使用例

本機構の使用例として，ウェブブラウザ上でプログラミングが可能となる OSS のウェブベース統合開発環境 (Web IDE) を中心としたサブプロジェクトを立ち上げ，東京工科大学での運用を開始した．

そこで，所属研究室にて被験者 12 人に対して評価実験を行った結果，表 1 に示す通りとなった．今回の実験では，AWS[3] の東京リージョン上に構築された Worker で測定を行ったが，初回アクセス時のコンテナ起動時間の内訳はそのほとんどがデータ転送における時間であった．これは実装上，コンテナ自体はすでに立ち上がっている状態であるが，データ転送を待つ実装となっている為である．従って，データ転送をジョブキューにエンキューするような方式を取る事で，更なる高速化が図れると考察される．

本機構を用いて AWS と連携を行う事によって，オンプレミスでは最小限のリソースで学生にプログラミング環境を提供する事が可能となった．また，コンテナ側で使用可能なリソース量を制限する事により，一人のユーザが負荷の高いプログラムを実行した場合も，他のユーザ環境に影響を与えずに運用する事が可能となった．

4 課題と展望

現在の実装では，実際にコンテナが実行されるサーバは単純なラウンドロビン方式，もしくは動作コンテナ数によって決定される．しかし，この実装はサーバのスペックやリソース消費を考慮していない為，今後機械学習を用いたスケジューラを拡充していき，サーバ負荷によって適切なコンテナ実行先を動的に決定する方式を実装予定である．また，更なる高速化を図る為，分散ファイルシステムの開発を行っている．

参考文献

[1] ITmedia Enterprise – “学生が開発した「自作クラウド」を全学生 8000 人向けに本格稼働” http://www.itmedia.co.jp/enterprise/articles/1312/20/news017.html (2015/01/08)

[2] Docker, Inc – “Docker” https://www.docker.com (2015/01/08)

[3] Amazon Web Services, Inc – “Amazon Web Services” http://aws.amazon.com (2015/01/08)

Copyright ©2015 Information Processing Society of Japan.
All Rights Reserved.