Docker コンテナを利用した HPCI 仮想端末エミュレータ提供の検討

石井 宏治, 坂根 栄作, 合田 憲人

国立情報学研究所

k_ishii@nii.ac.jp

A Study on Deployment of HPCI Terminal Emulator using Docker Container

Koji Ishii, Eisaku Sakane, Kento Aida

National Institute of Informatics

概要

国立情報学研究所 (NII) は，革新的ハイパフォーマンス・コンピューティング・インフラ (HPCI) を構成する認証基盤の運用を担っている。その役割のなかで HPCI 資源にアクセスするための仮想端末エミュレータを主要なプラットフォームごとに提供している。本稿では，その維持管理において顕在化した問題点とその解決手法を報告する。

1 はじめに

革新的ハイパフォーマンス・コンピューティング・インフラ (High Performance Computing Infrastructure: HPCI) [1] は，全国の大学や研究機関（HPCI システム構成機関）に設置・資源提供されているスーパーコンピュータやストレージを連携し，産業界を含めた幅広い利用者層の多様なニーズに応える共用計算環境基盤を実現するものである。HPCI ではネットワーク上に分散した計算資源や Web 上のサービスに対して統一したアカウント情報で認証できる環境（シングルサインオン）を提供しており，国立情報学研究所 (NII) と HPCI システム構成機関は，X.509 公開鍵認証基盤に基づく Grid Security Infrastructure(GSI) [2] および SAML に基づく Shibboleth [3] を用いた認証基盤（HPCI 認証基盤）の運用を担当している。

HPCI 認証基盤は，HPCI 認証局ならびに証明書発行システムなどの表 1 に掲げる関連システム，利用者の認証情報を管理する Shibboleth Identity Provider (IdP) サーバ，スーパーコンピュータやストレージなどの計算資源へのログインノードとなる GSI-SSH サーバにより構成されている。図 1 は，HPCI 認証基盤の構成ならびに利用者が計算資源にシングルサインオンする際の手順を示している。Web 上のサービス利用時には IdP のアカウント情報による Shibboleth 認証が，計算資源の利用時には HPCI 認証局が発行する電子証明書を用いた GSI 認証が，それぞれ行われる。[4]

NII は，HPCI 連携サービス運営・作業部会 認証基盤サブワーキンググループ (Sub-WG) の一員として HPCI 認証局および関連システム等の運用を担当する他，HPCI 認証基盤に関連するソフトウェア開発や HPCI 運用事務局ヘルプデスクに対する運用支援業務なども行っている。[5]

これまで，HPCI の利用者が計算資源にシングルサインオンする際の推奨仮想端末ソフトウェアの一つとして，NII が保守する GSI-SSHTerm [6] を提供してきたが，当該ソフトウェアを維持管理をしていく上での問題点がいくつか浮上してきたため，これを機に GSI-SSH クライアントの提供体制を見直すこととした。

本稿では，Windows はもとより macOS，各種 Linux ディストリビューションなど，様々な環境下における使用が想定される HPCI の仮想端末エミュレータの提供と維持管理における問題点ならびに解決手法を報告する。

2 維持管理における問題点と課題

本節では，HPCI における仮想端末エミュレータの提供・維持管理にまつわる問題点と課題の整理を行う。

表 1　HPCI 認証基盤システム

システム名	主な役割
CA サーバ	電子証明書の発行および失効
RA サーバ	電子証明書発行および失効の申請に対する本人性を確認， 電子証明書の発行や失効を CA サーバへ要求
認証局リポジトリ	認証局に関する情報公開
証明書管理システム・証明書リポジトリ	発行済み電子証明書の管理， 電子証明書とローカルアカウントのマッピング情報管理
証明書発行システム・代理証明書リポジトリ	電子証明書発行のためのユーザインタフェース， 代理証明書の管理
DS サーバ	Shibboleth IdP 選択サービス

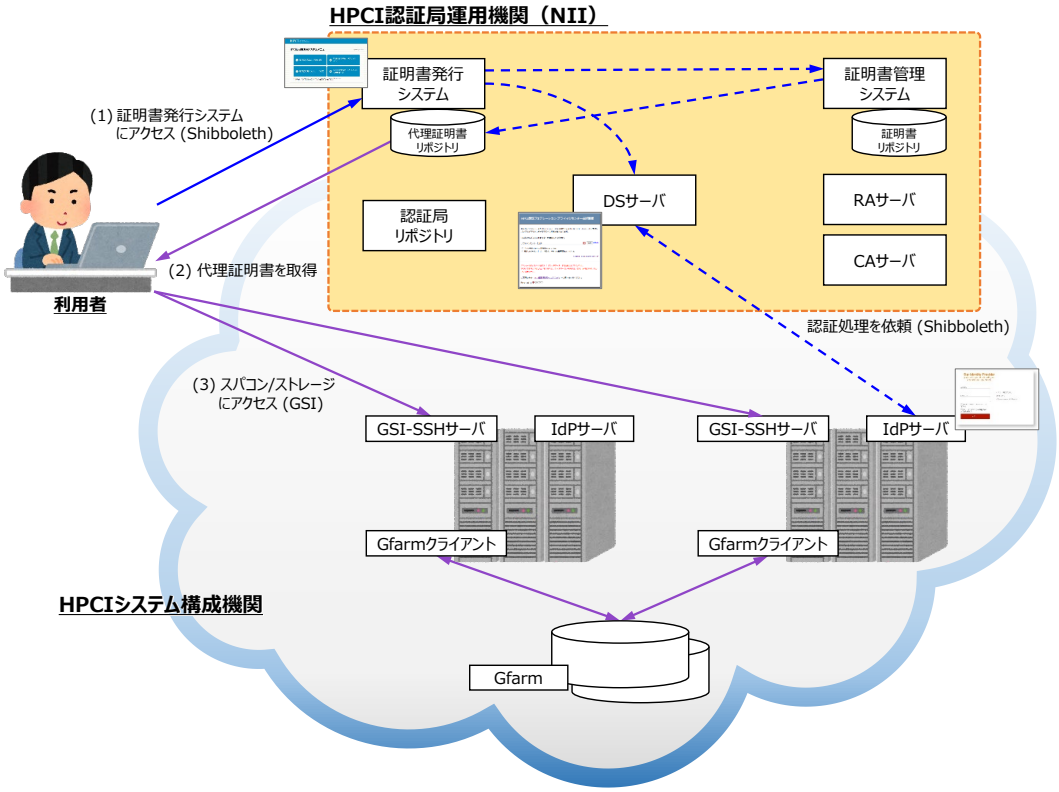

図 1　HPCI 認証基盤の構成とシングルサインオン手順

2.1　GSI-SSH クライアント

2019 年 9 月現在，HPCI において利用者が計算資源にシングルサインオンする際の推奨仮想端末ソフトウェアには GSI-OpenSSH Client と GSI-SSHTerm の二種類の GSI-SSH クライアントがあり，NII では必要に応じて，それぞれに対する独自の改修を行っている。

GSI は，米国アルゴンヌ国立研究所およびシカゴ大学を中心とする Globus Alliance によってオープンソースソフトウェアとして開発および保守されていたグリッドコンピューティング環境を構築するためのミドルウェア Globus Toolkit [8] においてセキュリティ機能を提供するもので，HPCI 認証基盤システムを構成する主要な認証・認可技術の一つである。GSI-SSH は，GSI を利用できるようにした SSH である。

2.1.1　GSI-OpenSSH

GSI-OpenSSH は，Globus Toolkit に含まれているコマンドラインインタフェースの GSI-SSH クライアントで、OpenSSH をベースに GSI 対応にしたものである。実装言語は C である。

Globus Alliance による Globus Toolkit のサポートは 2018 年末に終了となったため，NII では次期

HPCI システム更新までの HPCI 認証基盤の安全性を維持するためにソフトウェア保守管理体制を独自に組み，2017 年度第 4 四半期以降は随時の対応を行っている。NII 改修版 Globus Toolkit は，Red Hat Enterprise Linux (RHEL) 6, 7 および互換 OS のクライアント向けとして GSI-OpenSSH ならびに代理証明書リポジトリから代理証明書をダウンロードするための MyProxy を含む RPM パッケージを作成，NII 独自の公開リポジトリにて配布している。

NII 改修版とは独立に Globus Toolkit から派生して Grid Community Forum (GridCF) による保守が行われている Grid Community Toolkit (GCT) [9] が存在する。2019 年 9 月時点における GCT の提供状況は，RHEL 6, 7 および互換 OS ならびに Fedora 28, 29, 30 を対象とした RPM パッケージのみが正式版とされ，GSI-OpenSSH を含む内容で提供されている。その他の OS 向けについてはテスト段階やプレビュー版の扱いのパッケージが提供されているものの，それらの中には GSI-OpenSSH は含まれてはいない。

2.1.2 GSI-SSHTerm

GSI-SSHTerm は，イギリス NGS (National Grid Service) で開発されたグラフィカルユーザインタフェースの GSI-SSH クライアントで，Java で実装された SSHTerm をベースに GSI 対応にしたものである。

NGS による開発は既に終了しているが，GSI-SSHTerm のソースコードはオープンソースソフトウェアとして公開されており，NII ではこれまで必要に応じて独自の改修を行い，Web サイト上で ZIP 形式のファイルを公開・配布している。

NII 改修版 GSI-SSHTerm は，Globus Toolkit の主たる動作環境が UNIX であったために、Windows 環境向けとして提供することを目的としたものであるが，Java 実行環境上で動作することや導入の手軽さなどのことから，macOS や各種 Linux ディストリビューション上においても利用されている。

2.2 問題点と課題

昨今のソフトウェアあるいはプロトコルに対する脆弱性情報が次々と判明する状況を鑑みれば，機能改善・拡張が積極的に検討されており、かつ GSI が依存する暗号プロトコルである Transport Layer Security (TLS) の最新技術に追従していくことは，GSI-SSH クライアントの安全性を維持する上で必要不可欠である。TLS 1.0 および 1.1 を禁止するポリシが一般的になりつつある現状を踏まえれば、GSI-SSH の TLS 1.2 対応は焦眉の急である。GSI は GSSAPI の仕組みを利用している。GSI-SSHTerm については，GSSAPI そのものは Java 純正ライブラリで提供されているものの GSI は想定されていないために、オープンソースで開発されている暗号ライブラリからフォークして GSI に必要なメソッドを追加する、という手法を用いて GSI をサポートしている。そのような実装の事情から、TLS 1.2 に追従するにはベースとしている暗号ライブラリを更新し、その上で GSI に必要なメソッドを追加するなど改修の範囲が多岐に渡り，工数が膨大となることが容易に想定される。したがって、GSI-SSHTerm の TLS 1.2 対応は極めて困難と言わざるを得ない状況にあり、代替となる GSI-SSH クライアントの検討が課題となる。

GSI-SSHTerm に代わる GSI-SSH クライアントを検討するにあたり，以下に掲げる項目が要件として挙げられる。

1. 利用者の負担を考慮した導入の容易さはもとより極力少ない費用で入手および利用できること。
2. GSI-SSH クライアントを利用するローカル端末と計算資源との間でファイル転送できること。
3. 利用者が安心して利用できるように配布物の真正性を担保できること。
4. 維持管理の観点から、依存する暗号技術・プロトコルの最新版に容易に追従できること。

3 解決策の提案

3.1 Docker Desktop の利活用

Globus Alliance の Globus Toolkit では，Cygwin もしく MinGW と組み合わせて利用する Windows 向けのバイナリパッケージが提供されていたが，煩雑なセットアップ手順が必要となるため導入の難易度は高いと言える。しかしながら、C 言語で実装されたソースコードをそのまま流用するアプローチは有効であり、UNIX 向けの Globus Toolkit をどのように Windows 環境上に実現するかがポイントとなる。

ここでは，Windows 環境で利用可能な仮想化ソフトウェアとして Docker Desktop for Windows の活用を考える。Docker Desktop for Windows は，コンテナ型仮想化環境を提供する Docker [10] を Windows 環境（Windows 10 Pro, Windows 10 Enterprise, Windows 10 Education の各エディションのバージョン 1511 Build 10586 以降。いずれも 64 ビット版で Hyper-V を有効化しておく必要がある。）においても利用できるようするためのソフトウェアである。Docker

Desktop の導入には煩雑なセットアップ処理は必要なく，関連ツールも統合されているのため本ソフトウェアのインストールのみで Windows 環境における Docker の利用が実現されている。

Docker の仮想化環境を使用する際，読み込み専用のテンプレートである Docker イメージを入手した上で，実行用の Docker コンテナを起動する。Docker イメージは，Docker が動作する OS の種類に関わらず同一のイメージの利用が可能である。したがって，1 個の NII 改修版 GSI-OpenSSH からクライアントに特化した Docker イメージを作成すればよい。

また，macOS（Sierra 10.12 以降。）に対応する Docker Desktop for Mac も用意されており，Windows 環境と同様に扱うことができる。

3.2 Docker image for GSI-OpenSSH Client

本小節では，GSI-OpenSSH Client を基に構成される Docker image に関して，どういったイメージ構成にすべきか，またどのようにイメージを配布するかの検討を行う。

3.2.1 イメージ構成

Docker イメージの構成要素としての OS については前述のとおり CentOS 7 とし，Docker Hub の CentOS 公式リポジトリ上の CentOS 7 のうち，最新版となる Docker イメージをベースにして，NII で独自に改修する GSI-OpenSSH Client，MyProxy を含む必要最小限のパッケージを導入したイメージとして作成する。

配布するイメージについては，そのライフサイクルも検討する必要があるが，Docker イメージの更新契機としては，ベースとなる CentOS イメージのマイナーバージョン毎のリリース時もしくは NII 改修版 GSI-OpenSSH Client，MyProxy および関連パッケージの新バージョンリリース時が挙げられる。

3.2.2 イメージの配布方法

Docker イメージを公開・配布する方法には，いくつかの選択肢がある。

まず挙げられるのは，レジストリを利用する方法である。レジストリは，Docker イメージの保管と提供を行う場所であり，Docker イメージの提供元単位のリポジトリに分けられる。Docker イメージの登録および取得は Docker が提供するツールに統合されており，例えば利用者が Docker イメージをリポジトリから取得すると，Docker 内にイメージを取込む処理までが一括して行われる。

レジストリを使用する以外の方法では，Docker イメージを Web サイト上で公開・配布する方法も考えられる。イメージ提供者は，公開する Docker イメージを tar ボール化し，Web サイト上に掲載する。利用者は，Web サイトからダウンロードしたイメージを Docker に取込んで使用する。

■Docker Hub Docker Hub [11] は，Docker 社が提供するレジストリのクラウドサービスで，様々な提供元によるリポジトリにおいてコンテナ構築に使えるイメージが公開されている。前述のとおり Docker image for GSI-OpenSSH Client のベースイメージとして利用する CentOS 7 の公式イメージも Docker Hub 内にある CentOS のリポジトリにて提供されている。CentOS のリポジトリは，Docker 社の審査を通過した Docker Official Images として登録されており，誰もがイメージを登録できる公開レジストリである Docker Hub のなかであっても信頼できるリポジトリとして扱われている。

■プライベート・レジストリ 独自に構築したレジストリをプライベート・レジストリと呼ぶ。プライベート・レジストリを構築する方法としては，Docker 社が DockerHub で自ら配布する Registry コンテナを利用して独自のレジストリを構築する方法，Amazon 社提供のクラウドサービス Amazon Elastic Container Registry の利用などが挙げられる。

■Docker Content Trust Docker Hub に限らずプライベート・レジストリを使用する場合も含めて，Docker イメージをリポジトリから取得する際は Docker 内にイメージを取込む手順までが一括して行われる。この時，Docker Content Trust を使用するように設定している場合は Docker イメージの提供者の検証が行われ，署名済みかつ提供者検証済みであるイメージのみコンテナとして利用することができる。Docker Content Trust による検証の流れは次のとおりである。提供者がイメージをレジストリへ送信する前に，Docker はイメージ提供者の秘密鍵を使ってイメージに署名を行う。レジストリから Docker イメージを取得する際，Docker はイメージ提供者の公開鍵を使い，イメージ提供者が作成したものであるかを確認する。Docker Content Trust を使用するように設定していれば，レジストリから取得したイメージに署名が無い場合や提供者が検証できない場合はイメージをレジストリから入手したとしてもコンテナとして利用することはできない。

■tar ボール化イメージの検証 tar ボール化した Docker イメージについては，利用者によるダウン

ロードと Docker への取込みは個別に行う必要があるため，Docker Content Trust によるイメージ提供者の検証を適用することもできないが，ファイルのハッシュ値を別途公開しておき，Docker イメージと比較することでファイルの真正性を確認することは可能である。Windows 7 以降では，標準機能として用意されている CertUtil コマンドを使用することでファイルのハッシュ値を確認することができ，対応するハッシュアルゴリズムは MD2，MD4，MD5，SHA-1，SHA256，SHA384，SHA512 である。

3.2.3 配布方法の選定

Docker イメージの公開先として，Docker Hub，プライベート・レジストリならびに Web サイトのそれぞれを HCPI 認証基盤における事情を考慮しながら比較する。

Docker Hub を利用する場合，HCPI 認証基盤のリポジトリを作成した上で Docker から push することで Docker イメージの登録が完了し公開される。しかしながら，Docker Hub は公開レジストリであるため，誰もが自由にレジストリを作成できることから，配布物の性格を考慮すると Docker Official Images としての認定を Docker 社から受けておくことが望ましいが，審査および対応のための時間が必要となる。

プライベート・レジストリにて配布する場合，独自に構築したレジストリは新規の公開サイトとなるため，運用・維持管理における検討も新たに必要となるなど，Docker イメージ公開に至るまでには時間を要することが懸念される。

Web サイト上で配布する場合，既に公開している NII 改修版 GSI-SSHTerm と同様に tar ボール化した Docker イメージを公開すれば良く，既存の Web サイトの仕組みが流用できるため，公開までに必要な時間は最も短くなる見込みである。

Docker image for GSI-OpenSSH Client を迅速に公開したい事情から，まずは Web サイト上での配布を行い，将来的には Docker 社の審査を受けた上で Docker Official Images として Docker Hub にて公開することを検討する。

4 評価

本節では，HPCI の利用者向けに Docker image for GSI-OpenSSH Client を提供することによって，2.2 節で掲げた要件を満たしているかの確認等を行う。

要件 1 については、Docker Desktop は無償にて提供されていること（Docker Hub からの入手，アカウント登録も無償。）、また、その導入は各 OS における一般的なインストール方法をサポートしていることから実現していると言える。

要件 2 は利用者のローカル端末から HPCI 計算資源へのデータ転送が出来ることを求める。これについては Docker Desktop はその実行環境のフォルダをコンテナにマウント可能であること、さらに GSI-OpenSSH からファイル転送プログラムである gsiscp あるいは gsisftp を利用することにより実現できる。

要件 3 については、差し当たり Docker Content Trust の仕組みを利用しないものの、パブリックドメインの証明書を利用した HTTPS プロトコルでダウンロードすることにより必要最小限の要件を満たしている。

要件 4 については、Docker image を通して C 言語によるソースコードをそのまま Windows, macOS に適用できることから、基本的に C 言語のソースコード管理に一本化でき、GSI-OpenSSH の依存する OpenSSH ならびに OpenSSL に追従することにより最新の暗号技術をサポートすることが可能である。

Docker image for GSI-OpenSSH Client は GSI-SSHTerm の代替として導入したものの、この手法は RHEL 以外の Linux ディストリビューション，例えば Ubuntu や OpenSUSE に対しても適用可能であり、当初の目標を超えたサポート拡充が期待できる。

これらのことなどから総合的に判断して，Docker image for GSI-OpenSSH Client の提供開始以後は GSI-SSHterm からの移行を推奨し，ある程度の期間をもって GSI-SSHTerm の保守を終了することを検討している。

5 おわりに

本稿では，マルチプラットフォーム対応を考慮した HPCI のエンドユーザ向けアプリケーションの提供と維持管理における問題点ならびに解決手法として Docker コンテナを利用する方法を報告した。今回検討した手法は HPCI に限らず汎用的な利用方法にも応用できるものでもあり，皆様の課題解決のお役に立てれば幸いである。

また，Docker image for GSI-OpenSSH Client についてはその汎用性の検証を行った上で，GSI を利用するコミュニティへも共有していく予定である。

参考文献

[1] 革新的ハイパフォーマンス・コンピューティング・インフラ (HPCI) の構築について, http://www.mext.go.jp/a_menu/kaihatu/jouhou/hpci/1307375.htm

[2] Von Welch, Frank Siebenlist, Ian Foster, John Bresnahan, Karl Czajkowski, Jarek Gawor, Carl Kesselman, Sam Meder, Laura Pearlman, Steven Tuecke, “Security for Grid Services”, in Proc. of the 12th IEEE International Symposium on High Performance Distributed Computing, 2003.

[3] R.L. Morgan, Scott Cantor, Steven Carmody, Walter Hoehn, Kenneth Klingenstein, “Federated Security: The Shibboleth Approach”, EDUCAUSE Quarterly, Vol.27, No.4, 2004.

[4] 合田 憲人, 坂根 栄作, 本山 一隆, 青木 道宏, 漆谷 重雄, “HPCI のためのネットワーク・認証基盤”, 大学 ICT 推進協議会 2013 年度年次大会論文集, 2013.

[5] 石井 宏治, 坂根 栄作, 合田 憲人, “HPCI 認証基盤の活動報告”, 大学 ICT 推進協議会平成 28 年度年次大会論文集, 2016.

[6] HPCI 向けソフトウェア, https://www.hpci.nii.ac.jp/software/index.html

[7] 石井 宏治, 坂根 栄作, 合田 憲人, “HPCI 認証基盤における Oracle Java SE サポート・ロードマップへの対策”, 大学 ICT 推進協議会平成 30 年度年次大会論文集, 2018.

[8] Globus Toolkit, http://toolkit.globus.org/toolkit/

[9] Grid Community Toolkit Official Documentation, https://gridcf.org/gct-docs/

[10] Enterprise Container Platform — Docker, https://www.docker.com/

[11] Docker Hub, https://hub.docker.com/