



## 2009 年度下期未踏 IT 人材発掘・育成事業 採択案件評価書

### 1. 担当PM

加藤 和彦 PM(筑波大学 大学院システム情報工学研究科 教授)

### 2. 採択者氏名

チーフクリエイター: 関場 治朗(株式会社ユニークス 代表取締役 CTO)  
コクリエイター: なし

### 3. プロジェクト管理組織

株式会社オープンテクノロジーズ

### 4. 委託金支払額

5,500,000 円

### 5. テーマ名

省資源ホスティングのためのコンテナ最小化技術の開発

### 6. 関連Webサイト

<http://sfc-tools.sourceforge.net>

## 7. テーマ概要

本提案は、仮想サーバ1台あたりの計算機資源の占有量を抑えるための省資源ホスティング技術に関するものである。

省資源ホスティングには、ハードウェア仮想化の VM (Virtual Machine) や、OS (Operating System) 仮想化の VPS (Virtual Private Server) に対して、より小さな仮想化単位であるコンテナの利用が有効である。現在コンテナを利用したホスティング技術は、Sun の Solaris など商用 OS に実装されている。

ここで仮想サーバを、サービスを提供するものとみなすと、その実行段階を少なくとも3つに分類できる。サービスの開発ステージ、テストステージ、運用ステージである。

本提案は、運用ステージの仮想サーバを、計算機資源の占有量を可能な限り小さくしたコンテナの中で実行することを目的としている。そのために、

- (1) 差分保存型ファイルシステムを用いた開発ステージ用コンテナ管理機能
- (2) 必要ファイル検知を用いたテストステージ用コンテナ管理機能
- (3) 不要ファイルを除去したコンテナを実行する運用ステージ用コンテナ管理機能

の3機能を備えたコンテナステージングツールを開発する。

本提案により、1台の物理サーバに多くの運用ステージの仮想サーバを、少ない手間で集約できるようになる。

## 8. 採択理由

コンテナ方式による仮想化は、ゲスト OS 環境がホスト OS 環境と同じである制約は持つものの、少ない実行時オーバーヘッドで仮想化を実現できるという特色を有する。本提案は、コンテナ方式の仮想化システムにおいて、差管理技術を活用してコンテナサイズを最小化し、省資源の仮想ホスティング技術を開発しようとするものである。開発・テスト・運用の三ステージにおいて、統合的な管理ツールを開発する計画である。既に feasibility study を進め、提案する方式の有効性に関する実験を行い、有効性が期待できることを示している。これまでに大手 IT 企業にてオープンソースソフトウェアに関する豊富な開発経験を持ち、提案開発を成功に導ける可能性が十分に高いと期待できる。以上から採択と判定した。

## 9. 開発目標

本プロジェクトの実施目標は以下の通りである。

1. 開発ステージ用コンテナ管理機能  
既存コンテナ作成ツールを利用し、差分保存型ファイルシステムを用いた開発ステージ用のコンテナ管理機能を実装する。
2. テストステージ用コンテナ管理機能  
必要ファイル検知を用いたテストステージ用コンテナ管理機能を実装する。ファイル検知メソッドを用いたファイル抽出機能と、抽出ファイルのパッケージ化機能を含む。
3. 運用ステージ用コンテナ管理機能  
不要ファイルを除去したコンテナを実行させる運用ステージ用コンテナ管理機能を実装する。パッケージ化されたファイルシステムのデプロイ機能、および、既存のコンテナ作成ツールとの連携機能を含む。
4. 管理ツール  
一連のステージを統一的に管理するためのコンテナ管理機能を実装する。
5. 実証実験  
本システムを用いた実証実験と、システムのチューニング。

## 10. 進捗概要

概ね、当初の予定どおり進めることができた。プロジェクトの過程で、実装上、計画との相違が出たが、機能的な不足はない。

## 11. 成果

本プロジェクトではコンテナ型の仮想化として Linux Containers (lxc) をベースとし、省資源でホスティングを行うためのツール群を開発した。ツールは以下の各ステージ用のコンテナを管理する機能と各コンテナを統合的扱うツールである。ツールを使ってコンテナ内で動作させるアプリケーションに必要なファイルを抽出し、アプリケーションを動作させるための最小限のストレージを構築した。

ストレージの最小化を行うため、コンテナ内で提供するサービスを一度動作させる。この時コンテナのファイルシステムを監視する。監視されたコンテナでは、動作時に利用したファイルが検知できる。検知したファイルのみを最終的なサービスを提供するコンテナで利用することによりコンテナのストレージを最小化することができる。

本プロジェクトでは、提供するサービスを開発するための開発ステージコンテナ、開発

するサービスを動作させ必要なファイルを抽出するテストステージコンテナ、抽出したファイル郡で開発したサービスを動作させる運用ステージコンテナの三つのコンテナを通してストレージの最小化を行う。これら各コンテナの管理機能と統一的にコンテナを管理するツールを作成した。

従来のコンテナでアプリケーションを動かしたい場合、開発環境をそのまま動作させていたが、本来 unnecessary なファイルが大量にありディスクを浪費していた。本プロジェクトの成果を使い、最終的にコンテナで動作させたいアプリケーションをテストステージコンテナ上で実際に動作させることにより、そのアプリケーションが利用しているファイルを検出する事ができる。

この検出したファイルのみを利用するコンテナを作成することで、 unnecessary なファイルを削除し最小限のディスク消費量でアプリケーションを動作させることが可能となった。

従来のコンテナでは、アプリケーションを動かすためには開発環境をまるごと必要としていた。この為物理的に別のマシンでコンテナを動かす場合もその環境を移動する必要があった。また、コンテナのルートファイルシステムの設定を環境の移動の都度、適切なパスに変更する必要があったが、本プロジェクトの成果を利用すれば、自動ファイル検出で検出されたファイルとコンテナの設定を一つのパッケージにまとめる事ができる。まとめたファイルと設定を利用すれば、パッケージをインストールしたどここの環境でも同じようにコンテナを再現できる。また、コンテナが利用しているファイルは最小限になっているので、移動の際にも大きな負荷をかけることがない。

(参考)関連 URL

sfc-tools <http://sfc-tools.sourceforge.net/>

プロジェクトを進める過程で、実装上、以下にあげる計画との相違が出たが、いずれも、予定していた機能が含まれている。扱いが容易にできるようになっている分、当初の予定よりもよいものになった。

#### ライブラリの実装方式

計画書では liblxc を使い shared object としてライブラリを実装することを計画していたが、最終的な実装は ruby のライブラリとして実装し、スクリプトからそのライブラリを利用する方式に変更した。これは、ruby でツール群を実装する場合、shared library の実装と、ruby での実装に大差がないと判断したためである。機能としては計画通りの機能が実装されている。

## 運用ステージコンテナの実現方式

計画書では運用ステージコンテナは stackable ファイルシステムを利用せず、抽出したファイルツリーをそのままルートファイルシステムとして利用する計画であった。最終的な実装は、運用ステージにおいても stackable ファイルシステムを利用して、同一の運用ステージのコンテナで複数のインスタンスを起動可能にした。これにより、運用ステージでコンテナが追記/変更したファイルとベースとなるファイルが明確に分離され、ベースファイルの update が容易になった。

## 12. プロジェクト評価

Linux のコンテナ機能は、基本機能しかなく、それを活用するための支援環境は乏しく、広く使われているとは言えない現状である。また、今回の未踏の提案において、OS レベルのソフトウェア開発提案は、非常に少なくなっていることもあり、貴重なソフトウェア開発案件であった。OS レベルのソフトウェア開発は、地味ではあるが、コンピュータシステム利用を高度化する上で重要であり、また広く使われる様になった場合は、その社会貢献度は非常に大きい。本プロジェクトは、Linux のコンテナ機能を簡便に使うために本質的といえる開発を当初の予定通り行っており、今後の発展が大いに楽しみである。

## 13. 今後の課題

当初予定した開発は行えたが、利用実績を積み、有用性を実証していくのはこれからである。開発したシステムでは、コンテナに含むべきファイルを事前に検出し、それらを格納したコンテナを作るが、実行時に事前検出できなかったファイルにアクセスした場合にはアクセスエラーが起きる。実際の運用ではそのようなエラーは問題とならない可能性はあるが、それが問題ではないということは、実際の使用経験、使用実績を積んでいく中で明らかになっていくことである。開発したソフトウェアが国内外で広く利用され、その有用性が検証されると共に、利用経験を基に、さらなる改善が継続的に行われていくことを期待する。