

Regional SNS
クラウド環境対応作業
結果報告書

2012年2月29日

株式会社ネットワーク応用通信研究所

目次

1. はじめに.....	1
2. 問題点と対策内容.....	1
2.1. Rack のバージョンに伴う Content-Length の異常.....	1
2.2. ImageMagick のバージョンが想定バージョンと異なっていた.....	1
2.3. バイナリファイルのダウンロードが MOGOK 環境では動作しなかった.....	1
2.4. テンポラリディレクトリが書き込み禁止になっていた.....	2
2.5. MOGOK β 版の制限事項に対する対応.....	2
3. まとめ.....	3
4. 報告者所感.....	4

1. はじめに

本ドキュメントでは、OpalSNS（以下、本システムと呼称）を MOGOK β 版の PaaS 環境において利用可能にするために行った対応内容を報告します。

2. 問題点と対策内容

本システムを MOGOK β 版環境で動作させるに当たって遭遇した問題点と、それに対して行った対策内容を以下に示します。

2.1. Rack のバージョンに伴う Content-Length の異常

問題点

MOGOK β 版においてアプリケーションサーバの Rack バージョンが 1.2.x であり、レスポンス時に Content-Length が 0 になる場合があります、本システムの一部の画面（主にフォームを用いる画面）で、HTML が完全にサーバから受け取れない問題が発生しました。

対策内容

MOGOK 側の Rack バージョンは将来的に対策済みバージョンにアップデートされることが予想されるため、一時的に本問題を本システム側のみで解消するためのモジュールを作成し、本システム導入時に選択的に適用／不適用を決められるように設定して対応しました。

2.2. ImageMagick のバージョンが想定バージョンと異なっていた

問題点

MOGOK β 版の ImageMagick が、本システムが依存しているバージョンと異なっているため、MOGOK β 版環境において使用するライブラリを、RMagick が MOGOK のデプロイ時に導入に失敗してしまう

対策内容

ImageMagick のバージョンは、アプリケーション実行環境のベースとなる OS の標準パッケージのバージョンであるため、根本的対応が難しいと判断し、本システムで使用する画像処理ライブラリを RMagick から MiniMagick に変更し、それに伴い必要なソースコードの修正を実施して対応しました。

2.3. バイナリファイルのダウンロードが MOGOK 環境では動作しなかった

問題点

MOGOK のアプリケーション実行環境では、アプリケーションサーバとフロントエンドの Web サーバが別マシンに分かれている（これは一般的な運用環境でもある）。

しかし、本システムのファイルダウンロード機構は Rails の「`send_file`」メソッドを使用し、アプリケーションサーバのファイルシステム内に存在するファイルを直接フロントサーバからクライアントへ送る方法を探っていた。

このため、ファイルをクライアントへ転送するよう依頼を受けたフロントサーバは当該ファイルを自身のローカルファイルシステム内に見つけることが出来ず、ファイルのダウンロードに失敗する問題が発生した。

対策内容

`send_file` を使わずファイルをダウンロードできるように、本システムのファイルダウンロード機構のソースコードを改修し、MOGOK 環境でも問題無くダウンロード可能になるよう修正しました。

2.4. テンポラリディレクトリが書き込み禁止になっていた

問題点

現状の MOGOK β 版ではアプリケーション実行環境のテンポラリディレクトリ (`/tmp`) が **Read Only** になっており、このディレクトリに一時ファイルを格納している本システムの動作に影響があった。

対策内容

本システム側でアプリケーション実行環境の環境変数を修正し、書き込み可能なディレクトリをテンポラリディレクトリにするよう修正を実施しました。

2.5. MOGOK β 版の制限事項に対する対応

問題点

MOGOK β 版では、アプリケーションのログ情報が Rails 標準のままでは出力されないため、Rails 側に数行の対策コードを追加する必要があった。

対策内容

MOGOK β 版固有の制限事項（ログを標準出力に書き出す必要がある）に対応するため、本システムの環境設定ファイルを修正し、以下を追記して対応しました。

```
config.logger = Logger.new(STDOUT)
$stdout.sync = true
```

3. まとめ

MOGOK のβ版固有（将来的なリリースで解消されることが予想される）の問題が多く、ほとんどは将来の MOGOK 製品版では解消されるものと思われますが、バイナリファイルのダウンロードの問題や ImageMagick ライブラリのバージョン問題などは、製品版や他の PaaS でも発生し得る問題であり、ソフトウェア側で最初から考慮して実装しなければならない項目も多いと考えられます。

本システムを MOGOK β版環境に適用する過程では、これら PaaS 側のアプリケーション実行環境についても、アプリケーション開発者側がある程度把握していないと、ローカル環境でテストしていたのでは気付かない（顕在化しない）問題に遭遇する可能性が予想以上に高いことが分かりました。

今回の改修では、「将来の MOGOK 側の環境変更があっても、柔軟に対応できること」を意識して修正作業を行いました。

PaaS 側でパッケージの更新などを行い、元々の問題が解消された場合も、本システム側のソースコードの修正は発生しないか、極力小さな変更で済むよう、修正内容の部品化・局所化を徹底しています。

本改修作業によって、OpalSNS (RegionalSNS) については、ソースコードレベルで PaaS 環境に対する対応が十分に進んだものと考えます。

4. 報告者所感

PaaS 環境におけるアプリケーション開発は、アプリケーション実行環境側のソフトウェア構成が予め用意されており、サーバ構築の手間が無い反面、アプリケーション実行環境側のソフトウェア構成を任意にカスタマイズし難いという問題が存在します。

このため、アプリケーション実行環境も自前で作成するタイプの開発作業とは異なり、定められた実行環境に適合するようにアプリケーション側の設計・実装を行わねばならず、既存のアプリケーションを PaaS 環境に適合させる作業は、通常の開発とは異なったノウハウが必要となり、予想以上に難しい問題に直面しました。

特に、

- アプリケーション実行環境（OS 側）に組み込まれているアプリケーションのバージョン確認
- アプリケーションサーバとフロントサーバの物理構成
- 通信できるポートの把握

については、その重要度を痛感しました。

今回適用を試みた **Regional SNS** は、ソースコードの量的にも、また依存している外部パッケージ・サービスの種類のにも、PaaS 環境に適合を試みる上で、一般的な知見を得るに適した規模でした。

今回本システムを **MOGOK β** 版環境に適合させる過程で得られたノウハウは、一般的なアプリケーションを **MOGOK** 環境に適合させる上でも十分に参考になり得る情報と考えます。