

XML Web サービスのための分散 OS

大木 薫† 新城 靖† 佐藤 聡† 板野 肯三† 馬淵 充弘†

1. はじめに

XML Web サービスは、Web サービス(Web of service)ともよばれ、XML形式のデータ交換を行うことでソフトウェア・コンポーネントをネットワークを通じて利用可能にするものである。Web サービスの利点としては、使用する言語、プラットフォームから独立していることが上げられる。現在公開されている代表的な Web サービスとしては、Amazon Web Services、GoogleAPI および、Yahoo! Search Developer Network などがある。

現在の Web サービスの利用形態では、1 度に 1 つの Web サービスが利用されている。今後は単体で動作する形から発展し、複数の Web サービスが連携することが重要になっていくと思われる。本研究では、Web サービスを連携させるための分散 OS(Operating System)の要求要件を明らかにし、それを実装する。

2. Web サービス連携のための分散 OS に対する要件要求

2.1 ローカル OS における連携

ローカルの OS における連携と比較しながら、Web サービスを連携させるための分散 OS に必要とされる機能を明らかにする。ローカルの OS では、アプリケーションは他のアプリケーションとファイルを通して連携ができる。また、UNIX のパイプのようにファイルと互換性のあるプロセス間通信を用いて、他のアプリケーションの処理の結果を直接利用することができる。

現状の Web サービスで、複数の Web サービスのサーバを連携を連携させる場合、Web サービスのクライアントで、個々の Web サービスのサーバを呼び出す処理を記述しなければならない。通常、個々の Web サービスは、インタフェースやアクセス方法が異なる。現在、Web サービスではローカル OS のファイルのような、全てのサービスで利用

可能な標準的な基盤は存在しない。従来の Web サービスのクライアントは、特定の Web サービスのサーバや、非標準的なファイルを利用することしかできない。

RPC はもともと分散システムを構築するための技術であるが、ローカル OS においても、RPC は活用されている。そのような RPC の例として、Windows COM や、KDE における XML-RPC の利用が挙げられる。そのような RPC に対するアクセス制御はインターネットのような開かれた環境では、利用者認証を必要とするなどの問題があり、そのまま利用できない。

2.2 分散 OS における標準の定義

2.1 節で述べた問題を解決するために、次のような仕組みが必要である。

- (1) データ交換の標準を定義する。
- (2) サービスの起動方法の標準を定義する。
- (3) インターネットのような開かれた環境でも利用可能な利用者認証に頼らないアクセス制御方式を定義する。

本研究では、このような要件を満たす分散 OS を設計し、実装する。

3. XML Web サービスを連携させるための分散 OS

本分散 OS の構成を図 1 に示す。本分散 OS はグローバル・ファイルのサーバ、グローバル・パイプのサーバ、シェルから構成される。オブジェクトの保護にはカーナビリティを用いる。以下でこれらの要求について述べる。

3.1 グローバルファイルとグローバルパイプ

本分散 OS は Web サービスが利用できるファイル、およびディレクトリを提供する。そのようなファイル、およびディレクトリをそれぞれグローバル・ファイル、およびグローバルディレクトリと呼ぶ。グローバル・ファイルは、ストリームを使った読み出しや書き込みをするため、Java の

† 筑波大学システム情報工学研究科コンピュータサイエンス専攻

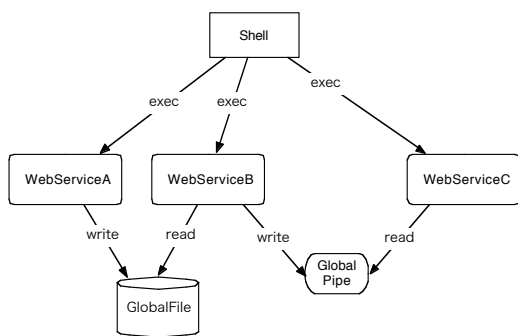


図 1:本分散 OS の構成

InputStream、および OutputStream と同様のインタフェースを持つ。

本分散 OS はファイル入出力と互換性のあるプロセス間通信を行うための機能として、グローバル・パイプを提供する。グローバル・パイプは、UNIX のパイプと類似の機能を Web サービスで提供する。このグローバル・パイプも、Java の InputStream、および OutputStream と同様のインタフェースを持つ。

グローバル・ファイル、およびグローバル・パイプにより、アプリケーションは動的に Web サービスのサーバを切り替えることが簡単になる。これらの機能により、本分散 OS は 2.2 節で述べた要件(1)を満たす。

3.2 シェル

シェルは利用者と対話することで、ファイルシステムの操作、Web サービスのサーバの呼び出し、およびパイプを用いた連携を行う機能を提供する。シェルが実行する Web サービスのサーバは全て同じインタフェースを持つ。具体的には、コマンドの実行を行う exec メソッドを持つ。このメソッドは、連想配列を引数および戻り値に持つ。この連想配列には、文字列、または、3.3 節で述べるケーパビリティを含めることができる。例えば、フィルタ形 Web サービスを利用する場合、引数と戻り値の連想配列の構成はそれぞれ以下ようになる。

引数 :

```
{stdin=inputCapability,
 stdout=outputCapability,
 arg1="en-jp"}
```

戻り値

```
{status="success"}
```

上の例において引数は stdin に入力元のグローバル・ファイル、またはグローバル・パイプのケーパビリティ、stdout に出力先のグローバル・ファイル、またはグローバル・パイプのケーパビリティ

を与えている。arg1 はフィルタ固有の引数である。戻り値には成功や失敗を示すステータスの要素がある。このようなインタフェースを用いることにより、統一された方法で Web サービスのサーバを実行することが可能になる。この仕組みにより、本分散 OS は 2.2 節で述べた要件(2)を満たす。

3.3 ケーパビリティによるアクセス制御

本分散 OS ではオブジェクトへのアクセス制御としてケーパビリティに基づくものを用いる。ケーパビリティには Web サービスのインタフェース情報である WSDL を用いている。ただし、内部の位置を示す URL に乱数を含ませ予測不可能にする[1]。

本分散 OS では、ファイルと互換性の無い Web サービスのオブジェクトについてもケーパビリティによるアクセス制御を行う。そのようなオブジェクトも 3.1 節で述べたグローバル・ディレクトリに登録可能とする。

ケーパビリティによるアクセス制御により、本分散 OS は 2.2 節で述べた要件(3)を満たす。

3.4 ファイル型の XML Web サービス

本分散 OS で、3.2 節で述べたシェルからアプリケーションとして実行される Web サービスの他に、ファイルとしてアクセス可能な Web サービスを提供する。これはローカル OS におけるプロセスファイルシステムと類似の機能である。このような方法は、固定されたキーワードによる検索エンジンや、天気予報などの Web サービスを利用する場合に有用である。

4. まとめ

本論文では XML Web サービスのための分散 OS について述べた。この分散 OS は Web サービスのためのファイルとパイプを提供する利用者との対話を行うシェルを提供する。アクセス制御のためにケーパビリティを用いている。

今後の課題は、分散 OS を利用するためのユーティリティプログラムを作成することである。また、スタックオブジェクトの生成による弱いケーパビリティの生成を行う機能、および Web サービスの同期を行うための機能を提供することを考えている。

参考文献

- [1] 新城 靖, 阿部 聡, 板野 肯三: “XML Web サービスのための大域的ファイル・サービスの提案”, 情報処理学会研究会報告 2004-05-96-03, pp.15-22 (2004).