

LL-004

Web サービスを適用した遠隔可視化システムの実現 Realization of the Remote Visualization System Based on Web Service

高垣 元[†]
Gen Takagaki

長尾 光悦[‡]
Mitsuyoshi Nagao

奥野 拓[†]
Taku Okuno

大谷 真[†]
Makoto Oya

1. はじめに

近年のインターネット技術の発達の一つの結果として、情報・資源などの分散化と共有が以前よりも容易となった。このことを背景に、物理的に離れた複数の計算機リソースをインターネットによって接続し利用することで、大規模な演算処理を遠隔利用可能とする研究が盛んに行われている。この結果、スーパーコンピュータなどによる大規模な演算処理も、より低レベルの計算機リソースからインターネットを通して行うことが可能となってきた。

しかし、演算結果が大規模になるほどに、人間には結果を理解することが困難なものとなっていく。演算結果が大規模である場合には、可視化(グラフィカルな表示)が直感的な理解を助ける有効な方法であるが、可視化するデータが大規模であるほど可視化処理にも大規模な演算が必要となる。そのため、可視化処理もインターネットを通して行うことが求められている。

インターネットを通して可視化処理を実現するときの課題は、以下である。

- プラットフォーム非依存
様々な計算機間をつなぐということを考えたとき、プラットフォーム非依存であるという点は非常に重要である。クライアントとして利用される多くのパーソナルコンピュータのOSはWindowsである。その他にもLinuxやUnixは多くの研究者が利用している。そのような中で特定のプラットフォームに依存しない仕様であることが重要である。
- ファイアウォールに対する透過性
インターネットにおいては、セキュリティ確保のためファイアウォールが存在する。そのため、独自プロトコルを用いて通信を行うと、ブロックされてしまう可能性が高い。

本研究では、Web サービスを用いることで以上の課題を解決し、ネットワークを通して簡単に利用することが可能な遠隔可視化システムを提案する。

2. Web サービス

Web サービスとは、インターネット標準の各種 Web プロトコルを利用してアクセス可能なプログラブルなアプリケーション・コンポーネントのことである。文書の構造化機能と実用性の両面を追求して開発されたマークアップ言語であるXMLベースのテクノロジーであり、仕様が標準化されているためプラットフォームに依存しない。

[†]北海道大学情報科学研究科

[‡]北海道大学情報基盤センター

Web サービスでは、プロトコルとしてはSOAP[2]を使用する。このSOAPは、下位プロトコルとしてHTTPなどを使用し、メッセージをやり取りすることで、リモート・マシン上のサービス・ルーチン呼び出すことを可能にする。HTTP上でSOAPメッセージの交換を行うことにより、ファイアウォールを越えて通信を行うことが可能である。これは、一般のファイアウォールではHTTP以外のポートがブロックされていてもHTTPのポートはブロックされていることは極めてまれなためである。

SOAPは応用性の高い規格であるが、RPC向けの基本的な部分が規定されているのみである。そのため、実用的なWebサービスに基づくアプリケーションを実現する際には、メッセージの改竄防止等のセキュリティ機能や長時間に及ぶ処理が必要なアプリケーションとの連携において有効である非同期通信等のSOAP自体には規定されていない機能を補完する必要がある。以上を踏まえ、サービス指向型のアプリケーションを実現するために開発されたミドルウェアがOpenSOAP[1]である。

本システムの開発では以下のような有効性があるため、OpenSOAPを利用することとした。

- オープンソース
オープンソースで提供されているため、本研究におけるシステムを開発する上で必要な部分を独自に実現し付加することが可能である
- セキュリティ機能
一般に、インターネットでは、SSLのようなトランスポート層の暗号化技術を用いることによりセキュリティを確保することが多いが、これに加えOpenSOAPではサーバ間でのメッセージ転送時など、メッセージ自体の暗号化及び認証が可能であり、SOAPメッセージの任意の特定部分を認証・暗号化できる。
- 非同期通信機能
RPCのような同期型通信だけでなく、非同期型通信も可能とするため、リクエストメッセージのキューイングやレスポンスメッセージのスプリーング機能を実装している。本システムにおいて実行する可視化処理は大規模な演算になる場合が多いと考えられる。このような場合には、一度接続を切ることができる非同期通信が必須である。
- C言語
OpenSOAPのコア部分はC言語を中心として開発されており、JavaやPerlで開発されているSOAP実装と比べれば、メッセージ処理のためのオーバーヘッドが小さい。

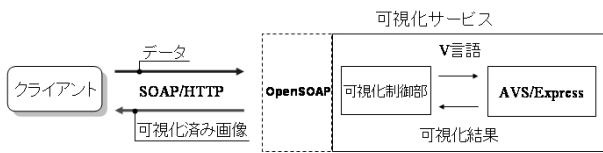


図 1: システムのイメージ

3. 遠隔可視化システム

3.1 システムの概要

本研究において開発したシステムのイメージを図 1 に示す。図中の「可視化制御部」が開発の中心部分である。

本システムでは、クライアントと可視化サービスは HTTP 上で SOAP を用いて通信を行っている。可視化サービスはクライアントからリクエストメッセージを受け取ると可視化アプリケーションに可視化処理を行わせ、出力された画像をレスポンスメッセージにのせてクライアントへ返す。

可視化アプリケーションとしては、今回は可視化アプリケーションとして標準的に利用されている AVS/Express を選択した。ただし、本システムは AVS/Express に限ったシステムというわけではなく、他のアプリケーションを利用可能にすることも考慮している。

AVS/Express とのインターフェースには、V 言語という言語を用いている。V 言語は AVS/Express のカーネルであるオブジェクトマネージャとのインターフェースとして用いられる言語である。可視化制御部が、クライアントから受け取ったリクエストメッセージに基づいて V 言語のコードを生成し、AVS/Express に処理させることで画像を生成する。

なお、本システムでは、SOAP ミドルウェアである OpenSOAP を用いることによって、非同期通信、セキュリティ機能を備えたサービスを実現している。

3.2 可視化サービス

可視化サービスの構成を図 2 に示す。可視化サービスは以下の 2 つの計算機で構成されている。

Web サービスインターフェースマシン

OpenSOAP サーバ及び可視化制御部が存在し、可視化サービスのインターフェースとなるマシン。

可視化実行マシン

可視化アプリケーションが存在し、実際に可視化処理を行うマシン。

可視化処理は一般に複雑な計算が必要であり、高速なコンピュータを必要とする。インターネット制御のためのマシンと分離することで、全体としての計算リソースの有効活用を図ることが目的である。また、サービスのエンドポイントを Web サービス側に置くことで、高速コンピュータへのハッカーの侵入を防ぐことも考察して

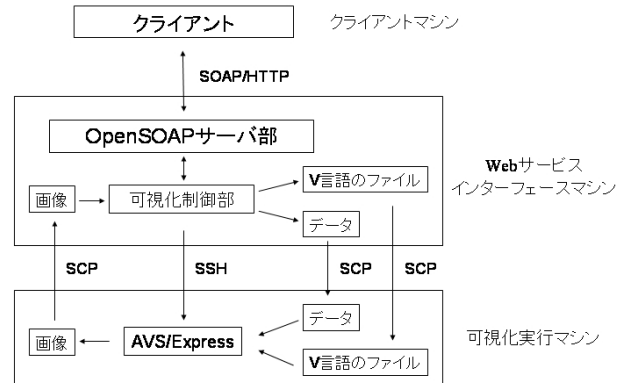


図 2: 可視化サービスの構成

いる。なお、2 つの計算機間は SSH で接続することによって、この間からの進入も防ぐようにした。

可視化サービスがリクエストメッセージを受け取り、可視化処理を行って、可視化結果 (画像) を含むレスポンスメッセージをクライアントに返す処理の流れは以下のようになる。

1. OpenSOAP サーバ部がクライアントからリクエストメッセージを受け取る。
2. OpenSOAP サーバ部が可視化制御部を起動し、リクエストメッセージを渡す。
3. 可視化制御部がリクエストに含まれているデータを Web サービスインターフェースマシンに保存する。
4. 可視化制御部がリクエストの内容にしたがって、V 言語のコードが書かれたファイルを Web サービスインターフェースマシンに作成する。
5. 可視化制御部がデータと V 言語のファイルを可視化処理を行う計算機に SCP で Web サービスインターフェースマシンから可視化実行マシンにコピーする。
6. 可視化制御部が SSH で AVS/Express を起動し V 言語のコードを実行させ、画像を生成する。
7. 可視化制御部が AVS/Express によって生成された画像を SCP で可視化実行マシンから Web サービスインターフェースマシンにコピーする。
8. 可視化制御部がレスポンスメッセージを生成し、OpenSOAP サーバに渡す。
9. OpenSOAP サーバ部がクライアントにレスポンスメッセージを送信する。

3.3 可視化アプリケーションの Web サービス化

可視化アプリケーション AVS/Express を Web サービスとして利用可能とするために、クライアントから送られてきたリクエストメッセージ (SOAP メッセージ) を解析し、そのメッセージにしたがった可視化処理を AVS/Express に行わせる部分が可視化制御部である。

リクエストメッセージから V 言語のコードを生成する部分については、以下の仕様とした。

- クライアントは利用可能な可視化手法の中から可視化手法を選択し、可視化のパラメータを指定する。このようにすることで、クライアントは可視化サービスが V 言語を用いて AVS/Express を利用していることを意識することなく、可視化サービスを利用することができる。また、利用するアプリケーションが AVS/Express でなく他のアプリケーションである場合でも同様なインターフェースで利用可能である。
- 可視化のパラメータには可能な限りデフォルトの値を決めておく。
クライアントが指定しなければならないパラメータの数を減らすことができ、クライアントが可視化サービスをより簡便に利用することが可能となる。

利用頻度の高い標準的なものとして、以下の可視化手法をサポートした。

- 鳥瞰図 <Surfplot>
- 断面 <Isosurface>
- 流線 <Streamline>

これらの可視化手法はすべて、可視化処理をする数値データと可視化のパラメータを入力として、画像を出力する仕様である。画像のフォーマットとしては、JPEG、PostScript 等のいくつかのフォーマットの中から選択可能となっている。

3.4 メッセージフォーマット

本システムのクライアントと可視化サービス間の通信において利用されるメッセージのフォーマットの設計を行った。可視化サービスに実装された数種類の可視化機能を扱うために、クライアントから可視化サービスに渡されるリクエストメッセージと、可視化サービスからクライアントへ渡されるレスポンスメッセージの 2 種類のメッセージのフォーマットを設計した。

3.4.1 リクエストメッセージ

リクエストメッセージの形式は WS-Attachments に準拠した DIME 形式のメッセージである。可視化処理をする数値データが添付データとして送られる。数値データを添付データとして送ることでパフォーマンスが向上する。

リクエストメッセージには可視化処理を行うために必要な情報が含まれている。具体的な例として、鳥瞰図を

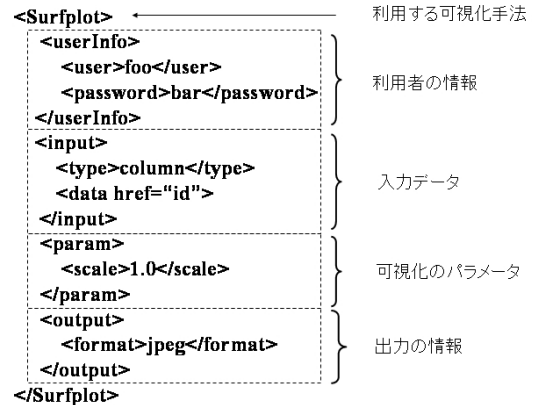


図 3: リクエストメッセージの例 (SOAP Body 内のみ)

可視化手法として選択した場合のリクエストメッセージの一部を図 3 に示す。このほかに数値データが添付されて送られる。

図 3 のメッセージには以下の情報が含まれている。

1. 利用する可視化手法 <Surfplot>
利用する可視化手法を指定。
2. 利用者の情報 <userInfo>
ユーザ名、パスワードを指定。
3. 入力データ <input>
添付されて送信されている可視化処理をする数値データを識別するための ID とその数値データの形式を指定。
4. 可視化のパラメータ <param>
倍率 (scale) を指定。
5. 出力の情報 <output>
出力する画像のフォーマット (JPEG) を指定。

3.4.2 レスポンスメッセージ

可視化サービスが要求された可視化処理を正常に実行し、画像を生成できている場合は、画像データの含まれたレスポンスメッセージをクライアントへ送信する。リクエストメッセージのパラメータが不適切であるなどのエラーが起きた場合には、フォルトメッセージがレスポンスメッセージとしてクライアントへ送信される。

なお、バイナリデータである画像データはテキストデータである SOAP メッセージにはそのままでは含めることはできない。本システムでは、このことを解決するためにレスポンスメッセージの形式もリクエストメッセージと同じ WS-Attachments に準拠した DIME 形式のメッセージとし、画像データは添付して送ることとした。

```
<SurfplotResponse>
  <image href="id" /> ← 画像データのID
</SurfplotResponse>
```

図 4: レスポンスメッセージの例 (SOAP Body 内のみ)

図 4 に鳥瞰図で可視化処理を行った場合のレスポンスメッセージの一部を示す。<image> タグは添付されている画像データを識別するための ID を示すためのタグである。

4. 検証実験

構築した遠隔可視化システムの動作検証を行った。検証に用いた計算機リソースについて以下に示す。

Onyx 可視化サーバ SGI Onyx 300, 32CPU, 16GB メモリ, IRIX6.5

SOAP PC Dell 社製 PC, Intel(R) Celeron(R) CPU 1.80GHz, 4GB メモリ

Client PC Intel(R) Celeron(R) CPU 2.0GHz, 512MB メモリ, RedHat Linux 9.0

Onyx は, AVS/Express がインストールされており, 可視化実行マシンとして利用した。SOAP PC は, Web サービスインターフェースマシンとして利用した。Onyx 及び SOAP PC によって可視化サービスを実現している。なお, Onyx 及び SOAP PC は北海道大学情報基盤センターに存在する。Client PC は, 北海道大学大学院工学研究科オープンシステム工学講座における PC であり, SOAP PC に SOAP リクエストを送るクライアントとして利用した。

クライアントと可視化サービスは HINES (北海道大学情報ネットワークシステム) によって接続されている。クライアントはオープンシステム工学講座のファイアウォール内に存在し, 可視化サービスは情報基盤センターのファイアウォール内に存在している。

検証には, 鳥瞰図を可視化手法として用いた。鳥瞰図は 2 次元の格子状のデータを受け取り, 格子点上のデータ値にしたがって画像を生成する可視化手法である。

この鳥瞰図を可視化手法として, クライアントから HINES を通じて可視化サービスに SOAP リクエストを送り, サービスから画像を取得可能であるかを確認した。また, 可視化のパラメータを変更することで, 異なる可視化処理が可能であるか確認した。

図 5 は可視化処理を行った結果である。図 5 は同じ入力データであるが視点を変更した場合の出力結果である。入力データには以下の式で生成した 2 次元の格子状の座標データを用いた。

$$f(x, y) = 100 \sin\left(\frac{x}{100}\right) \cos\left(\frac{y}{100}\right)$$

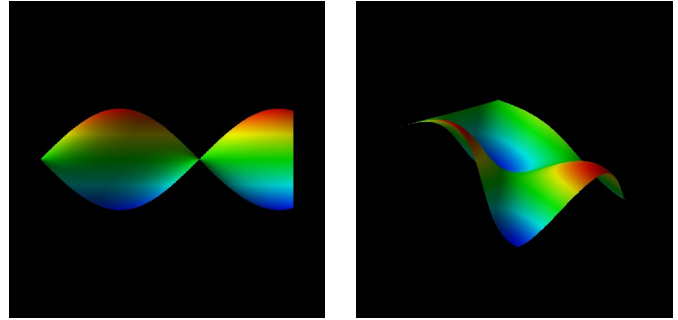


図 5: 可視化結果

$$0 \leq x, y < 500 \quad (x, y \in \mathbb{Z} : \text{整数})$$

クライアントから可視化サービスへの経路にはファイアウォールが存在しているが, ファイアウォールの設定を特別に変更することなく可視化処理を行うことが可能であることが確認された。

5. おわりに

本研究では, 可視化アプリケーションを遠隔地からであってもネットワークを通して利用することが可能な遠隔可視化システムを提案し, その動作検証を行った。本システムは Web サービスを用いているのでプラットフォームに非依存であり, 既存のセキュリティポリシーにも抵触しにくいものとなっている。このサービスを利用することで, プラットフォームやファイアウォールの存在を気にすることなく, ネットワークを通して簡単に可視化処理を行うことが可能になる。

今後の課題としては, 他のアプリケーションも Web サービスとして利用可能とし, また, それらと連携することでより広範なサービスを提供することで, システムの利便性を向上させることが挙げられる。

6. 謝辞

本研究はノーステック財団研究開発助成事業の補助を受けて行われた。

参考文献

- [1] “平成 12 年度 地域新生コンソーシアム研究開発事業 「即効型地域新生コンソーシアム研究開発」 「IT サービス事業共有基盤向けミドルウェアの開発」 成果報告書”, 2002
- [2] Don Box, David Ehnebuske, et al., “Simple Object Access Protocol (SOAP) 1.1”, W3C Note, 2000