

# トラフィック制御可能な Web サービスの実現

松井 健 金澤 典子 塚本 享治

東京工科大学メディア学部メディア学科

## 1 はじめに

Web サービスは、SOAP over HTTP[1]によってインターネットを介した疎結合のシステム統合を実現することが可能な技術であるとして注目を集めている。これにより、例えばチケット販売代理店と興行主のチケット販売などのプロセスを B2B のシステム統合によって効率化できる。

しかし、Web サービスは既存のシステム統合技術と比べ、処理負荷が多くかかる[2]。そのため、チケット販売のようなサービスリクエストが集中する際、Web サービスに輻輳が発生し可用性が低下する。

そこで本研究では、Web サービスをクラスタリングする際、早い段階での SOAP メッセージの流量を制御することで、Web サービスの輻輳問題を解決しようと考えた。これにより、SOAP メッセージが集中する Web サービス・エンジンへのメッセージ量を平滑化し、高可用性な Web サービスを提供できるようになる。

## 2 Web サービスのクラスタリング

Web サービスの高可用性を実現する方法として、Web サービス・エンジンを複数配備しサービスをクラスタリングする方法がある。この方法は、クラスタリングを拡張するたびに Web サービス・エンジンと Web サービスの実装プログラムを設置する必要がある。また、複数のサービス・エンジンからバックエンドシステムへの処理要求が発生し、バックエンドシステムに対する負荷が懸念される。

そこで我々は一つの Web サービス・エンジンにサービスを配備し、Web サービスを受け付ける窓口を中間ノードによってクラスタリングするという方法を考案した。この方法は、Web サービス・エンジンやサービスに依存することなくクラスタリングが行える。しかし、単にクラスタリングするだけでは、リクエストの増加に伴い Web サービス・エンジンやバックエンドシステムへの負荷を軽減できないという新たな問題が発生する。

この問題に対し、我々は Web サービス・エンジンへのサービスリクエスト量を安定させるトラフィック自己調節方式を考案した。

## 3 トラフィック自己調節中間ノード

トラフィック自己調節方式は、メッセージキュー、Pull 型メッセージ転送、メッセージ転送戦略、バッチ・メッセージを中間ノードに適用することによって実現する。

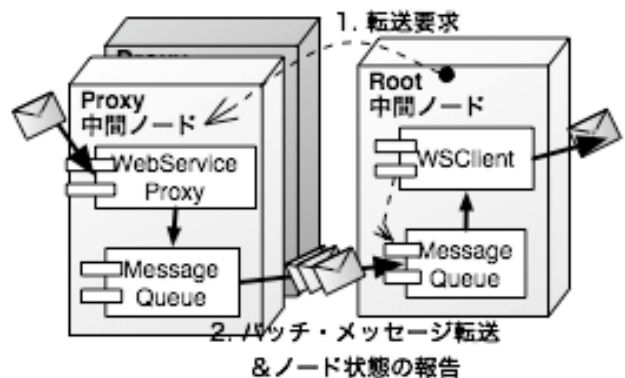


図 1 トラフィック自己調節中間ノード

中間ノードは、その立場によって Root ノードと Proxy ノードに分類できる(図 1)。Root ノードは、Web サービスと直接通信を行い、中間ノード全体を管理している。Proxy ノードは、Web サービスリクエストと直接通信する SOAP メッセージの受付窓口であり、Root ノードへメッセージを転送する。

### 3.1 SOAP メッセージキュー

中間ノードは、受け付けた SOAP メッセージを直ちに Web サービス・エンジンに転送するのではなく、親ノードの要求に応じて転送を行う。このため、中間ノードは転送要求が発生するまで SOAP メッセージを一時的に保持するものが SOAP メッセージキューである。そして FIFO (First-In First-Out) 方式に従って転送するメッセージを待たせている。

### 3.2 Pull 型メッセージ転送

通常、メッセージを転送する場合、転送元が自動的に転送先へメッセージを転送する「Push 型メッセージ転送」である。しかし、転送元が複数になると、転送経路に輻輳が発生する。この問題を回避するため、中間ノードのメッセージ転送には「Pull 型メッセージ転送」を採用した。Root ノードは、全ての子ノードに蓄積された SOAP メッセージを回収するため、それぞれの子ノードに対し、後述する

メッセージ戦略に基づいた転送数で SOAP メッセージを転送するよう要求する。これにより、メッセージが集結する Root ノードへの輻輳を回避する。

### 3.3 メッセージ転送戦略

中間ノード間でメッセージ転送する際、メッセージの転送数をどのように決定するかが重要となる。このメッセージ転送戦略には、1)メッセージ未転送数平滑化戦略と 2)メッセージ転送数一定戦略の2つが考えられる。1)は、各 Proxy ノードのメッセージ処理状況に応じて、転送数を決定するメッセージ転送戦略である。一方 2)は、全ての Proxy ノードのメッセージ処理状況に関係なく、一定数のメッセージを転送する戦略である。この2つの戦略は、適用対象となる Web サービス固有の要件であり、中間ノードの設定によって変更できる。

### 3.4 バッチ・メッセージ

Proxy ノードは、Root ノードのメッセージ転送戦略に基づいて決定された複数のメッセージを SOAP メッセージキューから取り出し Root ノードへ転送する。ここでメッセージを転送する方法として、1)一通ずつ転送する方法と 2)まとめて転送する方法の2つが考えられるが、1)は、転送毎にネットワーク I/O 処理が発生するため 2)に比べ転送時間が長くなる。このような理由から、複数メッセージをまとめたバッチ・メッセージを作成しメッセージの転送を行うことにした。

## 4 本方式の評価と考察

Web サービスを、Web サービス・エンジンによるクラスタリングとトラフィック自己調節中間ノードによるクラスタリングを用いて構築し、リクエスト失敗数と Web サービスのレスポンスタイムを測定し比較した。クラスタリングする Web サービスはチケット販売サービスで、中間ノードはメッセージ未転送数平滑化戦略を取り入れている。

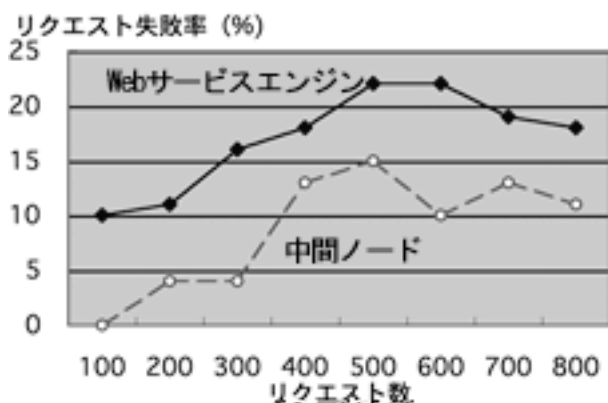


図1 Web サービスプロバイダへの負荷状況

図1は、リクエストのタイムアウトなどによってサービス呼び出しが失敗した割合とリクエスト数の関係を表している。この結果から、中間ノードによるクラスタリングの方が、高可用性な Web サービスであることがわかる。Web サービス・エンジンへの安定したトラフィックを実現することにより、サービスの可用性を高めることができたが、そのトレードオフとして、SOAP メッセージの処理が待たされるため、リクエスト数の増加に伴い、Web サービスのレスポンスタイムが増加してしまう(図2)。

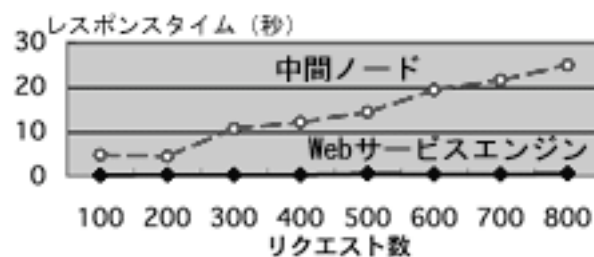


図2 Web サービスの平均レスポンスタイム

このレスポンスタイムの増加は、特に RPC 形式の Web サービスには大きく影響を与えてしまう。このことから中間ノードには、レスポンスタイム短縮を考慮したメッセージ転送戦略と Web サービスのレスポンス転送メカニズムが必要である。

## 5 おわりに

Web サービスのクラスタリングに、トラフィック自己調節中間ノードを利用することで、スケラブルで高可用性な Web サービスを提供することができるようになった。また、Web サービスプロバイダは、Web サービスへのクラスタリングが容易になった。

## 6 謝辞

本研究を進める上で貴重なアドバイスをして下さった橋本勉教授、棟上昭男教授に感謝します。また、本方式の評価時にサーバの提供と構築をしていただいた東京工科大学サークル XMLPro に感謝します。

## 7 参考文献

- [1] Don Box et al, "Simple Object Access Protocol (SOAP) 1.1", W3C Note, <http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>, (2000)
- [2] Dan Davis et al, "Latency Performance of SOAP Implementations", Department of Electrical and Computer Engineering, Center for Advanced Information Processing (CAIP), The State University of New Jersey, (2002)