5U-03

Webサービス連携における分散的なサービス発見

小関好尚† 松本倫子† 吉田紀彦†
†埼玉大学大学院理工学研究科

1 はじめに

近年 Web サービスでは，サービス開発環境の変化から Web サービス連携という新たな形態に注目が集まっている．Web サービス連携とは，複数のサービスを連携させ，1 つの大きな Web サービスとして動作させることである．

Web サービス連携を実現するためには 2 つの課題が存在する．まず 1 つ目に，連携するサービスの発見である．発見するサービスは，機能やサービス提供環境の状態の面で連携に適していなければならない．2 つ目に，発見したサービスの実行順序の決定である．これをスケジューリングと言い，目的の結果を得るために複数のサービスを正しく順序付けなければならない．現在では，これらの課題に対し，ブローカーなどを用いて中央集権的に解決するのが主流であり，具体的にはクライアントもしくはフロントエンド・サーバがその役割を担っている．

しかしながら，中央集権的なシステムでは，ブローカーが負荷集中点や単一故障点となってしまう問題が生じる．この問題を解決するためには，ブローカーを作らない分散的なサービス連携が望ましく，将来的にはその形態となることが予想される．

そこで本研究では，連携すべきサービスの情報を記述したスケジューリング情報を連携するサービスサーバへ転送し，サービスサーバ側でサービス発見と依頼を行う分散的な連携を提案する．そして，この提案について，シミュレーションによって有効性を示す．

2 関連研究

現在，分散的なサービス連携は研究段階にとどまっている．その研究の 1 つとして，分散ワークフローのサービス連携[1]がある．Duan らは，サービス出力データをサーバ間で直接送信する手段として RESTful なデータ送信手法を提案している．図 1 に示すように，クライアントはサービス出力データの URI を次に連携するサービスに伝える．その後，次に連携するサービスサーバは前のサービスサーバへ接続し，データを受信する．この手法により，サービスサーバ間でサービス出力データを直接送信することができるため，中央集権的にデータをやり取りするよりもデータ送信回数を抑えることができる．その結果，サービス出力データの転送時間を削減できる．しかしながらこの連携では，クライアントがブローカーとなり中央集権的にサービス発見を行うため，ブローカー部分の負荷集中，単一故障点となる問題の解決にはなっていない．

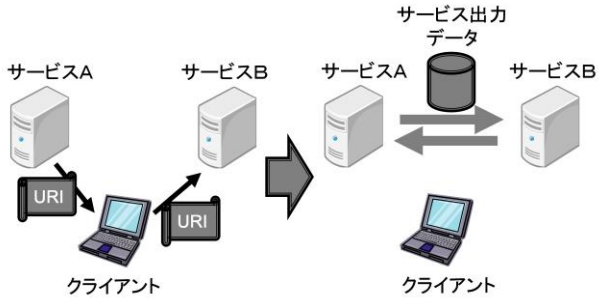

図 1:関連研究

3 提案手法

3.1 サービス発見

ユーザが望むサービスを見つけるためには，提供側サービスが持つ機能や提供環境の状態といったサービス情報を検索時に判断しなければならない．そこでサービス情報を構造化し，サービス間の比較を可能にするために，名前と属性のスキーマを作成し，名前スキーマの値にはサービス名を，属性スキーマとその値にはサービスの機能と状態を表現する．

サービスの検索方法は，サービスサーバの持つ名前と属性がクライアントの要求するサービス名，属性を満たした場合にヒットとする．複数のサービスがヒットした場合は，最も速くヒット応答が到着したサービスを採用する．これは，ヒット応答が速いサービスほどサービスサーバの混雑度が低いことが想定されるためである．

Distributed Service Discovery in Coordinating Web Services

Yoshihisa Koseki†, Noriko Matsumoto† and Norihiko Yoshida†
†Graduate School of Science and Engineering, Saitama University

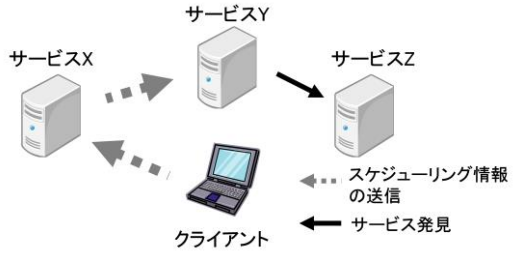

図 2:スケジューリング情報の転送例

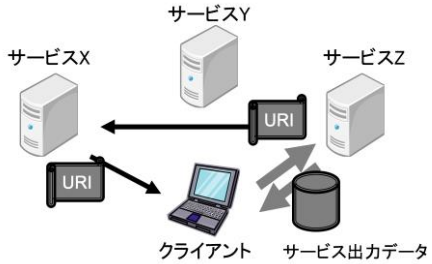

図 3:サービス結果の返送例

3.2 スケジューリング

本手法ではサービスサーバが次に連携すべきサービスを発見し，依頼する．この時，サービスサーバは次に連携するサービスの情報を知っておく必要がある．そのため，クライアントは連携するサービスの情報を順序付けて記述したスケジューリング情報を作成する．そして，図 2 のように，クライアントとサービスサーバがスケジューリング情報を次に連携するサービスサーバへ送信する．スケジューリング情報とは具体的に，サービスの情報とそのサービスの実行の順番付けから構成される．

3.3 クライアントへサービス結果の返送

クライアントへのサービス結果の返送は，クライアント側が NAT を用いることも想定し，図 3 のように初めに連携したサーバから送信する．また，ここでのサービス結果はサービス出力データではなく，データを指し示す URI とする．クライアントは受け取った URI に接続することでサービス出力データを取得する．これは，サービス出力データの送信回数を最小限にし，ネットワーク負荷を軽減するためである．

4 実験

本手法がクライアントの負荷分散とネットワークの負荷軽減に有効であることを確認するため，シミュレーション実験を行う．ここでは，従来の中央集権型，関連研究，本手法の比較を行う．

図 4 はサービス連携数が 5 の時のクライアントとサービスサーバの送信パケット数である．

本手法は従来の中央集権型，関連研究と比べ，送信パケット数がクライアントからほかのサービスサーバへと移っていることから，クライアントの負荷分散を確認できる．

図 5 はサービス連携数によるネットワークの全データ量である．本手法は関連研究とほぼ同じデータ量に抑えることができており，ネットワークの負荷軽減を確認できる．

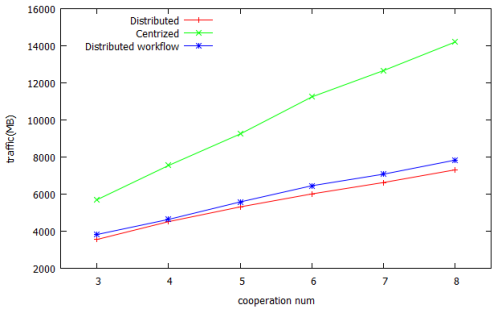

図 5:ネットワークの全データ量

5 まとめ

本研究では，連携すべきサービスの情報を記述したスケジューリング情報を連携するサービスサーバへ転送し，サービスサーバ側でサービス発見と依頼を行う分散的な連携を提案した．そして，シミュレーションにより，本手法においてクライアントの負荷分散，ネットワークの負荷軽減を確認した．

参考文献

[1] Kewei Duan, et al., “Composition of Engineering Web Services with Universal Distributed Data-Flows Framework based on ROA”, WS-REST, 2012.

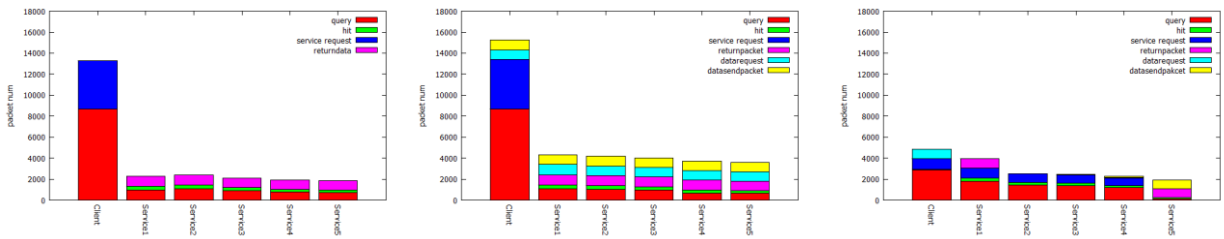

図 4:中央集権型（左図），関連研究（中央図），提案手法（右図）の送信パケット数