

グリッド上のアプリケーション連携による問題解決システム

下坂 久司[†], 廣安 知之^{††}, 三木 光範^{††}

[†] 同志社大学大学院 ^{††} 同志社大学工学部

グリッド技術は近年, Web サービスを基盤としたサービスの連携に主眼をおいて標準化が図られている. 本論文では特に, アプリケーションの連携に注目し, グリッド上のアプリケーション連携の枠組みを提案する. アプリケーション所有者は提案する枠組みに基づき, 既存のアプリケーションを Web サービスとして表現することで, Notification を用いたアプリケーション実行の連鎖と, アプリケーション間のデータ交換を実現でき, エンドユーザは広域ネットワークを通じて任意にアプリケーション連携を設計することが可能となる. これにより, Web サービスとして表現された既存アプリケーションの連携をプラットフォームとする, 新しいシステムの構築が容易に行えることを期待できる.

Problem Solving System based on Applications Integration on the Grid

Hisashi SHIMOSAKA[†], Mitsunori MIKI^{††}, and Tomoyuki HIROYASU^{††}

[†] Graduate School of Engineering, Doshisha University

^{††} Knowledge Engineering Dept., Doshisha University

Recently, Grid technologies have been standardized with the emphasis on Web services integration. This paper especially focuses on applications integration. Then, the framework of applications integration on the Grid is proposed. By application holders expressing existing applications as the Web service using the proposed framework, the chain of application invocations using the notification framework and data exchange among applications can be realized. Therefore, an end-user is able to design arbitrary applications integration through the wide area network, thus, it is expected that new system which consists of existing applications integration can be developed easily.

1 はじめに

自動車や航空機設計などの複合領域にわたる問題解決においては, 構造解析や流体解析, 最適化, 可視化などの高性能アプリケーションをそれぞれ用意し, アプリケーション間の整合性を保ちながら統合して利用する必要がある. 一般にこのような問題解決を支援するシステムでは, アプリケーション所有者から提供される複数の既存アプリケーションをシステム上のサービスとして登録し, エンドユーザは登録されたアプリケーション情報をシステムから取得して, 任意にアプリケーション連携を設計することで問題解決を図る. 一方でアプリケーション実行には専用の装置やデータベース, 並列計算環境を必要とする場合があり, また計算環境に適したチューニングによってアプリケーション実行性能を高められることから, 広域ネットワークを利用したシステム構築が多く試みられている.

広域ネットワーク上のシステム構築には, 分

散した計算資源や情報資源を仮想的に統合して利用するための基盤技術であるグリッド^{1, 2)}の利用が, システムの実用性や開発コストを考慮した場合不可欠である. 最近では, Global Grid Forum(GGF)³⁾ において Open Grid Services Architecture(OGSA)^{2, 4)} が提案されており, Web サービスを基盤技術として, 資源間の連携に主眼を置いた標準仕様の策定が進められている.

このような背景から本研究ではアプリケーション連携に特に着目し, Web サービスを基盤としたグリッド上のアプリケーション連携の枠組みを提案する. アプリケーション所有者は提案する枠組みに従い, 既存アプリケーションをその実行環境とともに, 共通のインターフェース, 状態, 機能を有する Web サービスとして表現することで, エンドユーザは広域ネットワークを通じて任意にアプリケーション連携を設計できる. また本研究では, 提案する枠組みに基づいたグリッド上の問題解決システムを Application Igniting System と呼

び、その概要について述べる。

2 Web サービスと Notification

2.1 Web サービスとグリッド

近年、広域ネットワーク上に分散して配置された資源を、仮想的に統合して利用するための基盤技術であるグリッドが高い注目を集めている。特に最近では GGF において、ビジネス分野におけるグリッドの利用促進や、グリッドミドルウェア間の相互運用性の確保などを目的として、OGSA が提案されている。OGSA は状態管理技術を有する Web サービスを基盤技術とし、標準化されたメタ OS サービス群を定義する。サービスを記述するための仕様として、Open Grid Services Infrastructure(OGSI)⁵⁾ や Web Services Resource Framework / Web Services Notification (WSRF/WSN)^{6, 7)} があり、両仕様は本質的に同等の機能を提供する。本研究では既存アプリケーションを Web サービスとして表現し、状態としてアプリケーション実行状況を扱うことでアプリケーション連携を実現する。この際、Notification を用いてアプリケーション実行状況を他のサービスに通知し、処理の連鎖を実現する。

2.2 Notification

Notification は Web サービスの状態変化により駆動する、登録型の非同期メッセージ通知の枠組みである。Notification にはいくつかの仕様が存在するが、最も単純な Notification の仕組みを図 1 に示す。ここではまず、サービス A の状態変更通知を受けたいエンドユーザやサービス B はあらかじめ、サービス A に対してそれぞれ通知依頼 (subscribe) を行う。その後、なんらかの処理によりサービス A の状態に変更が加えられた際には、通知依頼のあったエンドユーザおよびサービス B に対して、サービス A から状態の詳細が記述されたメッセージとともに、状態変更が通知 (notify) される。これにより、サービス A の状態変化に基づいて、エンドユーザやサービス B において別の処理を連鎖的に実行することが可能となる。Notification の利点として次の 3 点が挙げられ、柔軟なアプリケーション連携の実現が可能となる。

- サービス構築時にメッセージ通知先を規定する必要がなく、登録によって動的に通知先を変更できる。
- 1 度の状態変化で、複数の登録元にメッセージを通知できる。
- メッセージ通知先では、通知されたメッセージのみならず、メッセージ通知元の情報を利

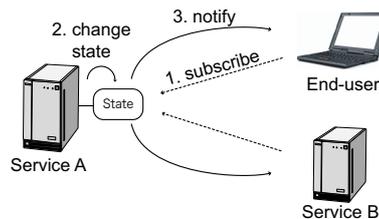


図 1: Overview of the notification

用した処理を実行できる。

3 グリッド上のアプリケーション連携

本研究では Web サービスを基盤としたグリッド上のアプリケーション連携の枠組みを提案する。アプリケーション所有者は提案する枠組みに従い、既存アプリケーションを Web サービスとして表現することで、エンドユーザは広域ネットワークを通じて任意にアプリケーション連携を設計することが可能となる。本節ではまず、提案するアプリケーション連携の枠組みについて述べ、その後、提案する枠組みに基づいた問題解決システムである Application Igniting System の概要について述べる。

3.1 アプリケーション連携の必要要件

一般にアプリケーション実行は、複数ファイルの入出力に置き換えることができ、アプリケーションのソースコードの変更を必要最小限とすることを考慮すると、アプリケーション間で入出力ファイルを交換することにより連携を行う方法が最も現実的である。一般的なアプリケーション連携システムでは、エンドユーザは各アプリケーションの概要、入出力ファイルの数、データ形式などの情報をシステムから取得し、アプリケーションの実行順序と、エンドユーザを含めたアプリケーション間のデータ交換を定義することにより、アプリケーション連携を設計する。ここでアプリケーション間のデータ交換を定義する際には、あるアプリケーションの出力ファイルを別のアプリケーションの入力ファイルとすることを指定するだけでなく、交換される入出力ファイルのデータ形式が異なる場合には、データ形式の変換方法も指定する必要がある。まとめると、エンドユーザは大きく次の 3 点を決定することによりアプリケーション連携を設計する。

1. アプリケーションの実行順序
2. アプリケーション間で交換される入出力ファイル

3. 交換される入出力ファイルのデータ形式の変換

ここで負荷分散を考慮した場合、集中管理によってアプリケーション実行とデータ交換を制御するのではなく、サービスとして表現されたアプリケーション間の分散管理による制御が望ましい。本研究では 2.2 節で述べた Notification の利点を踏まえ、各サービスの分散管理によるアプリケーション連携を実現する。

3.2 アプリケーションの実行順序

提案するアプリケーション連携の枠組みにおけるアプリケーション実行順序は、2.2 節で述べた Notification によるアプリケーション実行の連鎖により決定される。ここでアプリケーション連携全体の実行時間の短縮を考慮した場合、逐次的にアプリケーション実行を連鎖させるだけでなく、アプリケーションの並列実行をサポートする必要がある。例えば、あるサービス A においてアプリケーション実行後、サービス B および C において並列にアプリケーション実行を行えることが望ましい。この際サービス B および C の実行後に、同期をとってサービス D を実行するというように、アプリケーションの同期実行も同時に実現する必要がある。

このようなアプリケーション連携を実現するために、アプリケーションを表現する各サービスは、アプリケーション実行後に自動的に更新される「状態 S」と、他のサービスから状態変更通知を受け取ってアプリケーションを実行するインターフェースをそれぞれ保持する。後者のインターフェースは、複数の状態に通知予約を行っている場合、全てのサービスからの状態変更通知を待ち、同期をとってアプリケーション実行を行う。

サービス A から D のアプリケーション連携においてエンドユーザは、図 2(a) に示すように、実行したいアプリケーション実行順序の逆順に通知予約を行うことで、アプリケーション実行順序を決定する。これによりサービス A のアプリケーション実行後、図 2(b) に示すように、サービス A において状態 S が自動的に更新され、サービス B, C に状態変更が通知される。これにより両サービスにおいて、並列にアプリケーション実行が開始される。またサービス D では、サービス B, C の 2 つの状態 S に通知予約を行ったことから、両サービスの状態変更通知を待ち、同期をとってアプリケーション実行が開始される。

Notification をアプリケーション実行順序の決定に用いる利点は次の通りである。

- エンドユーザによりアプリケーション実行順序を任意に決定できる。

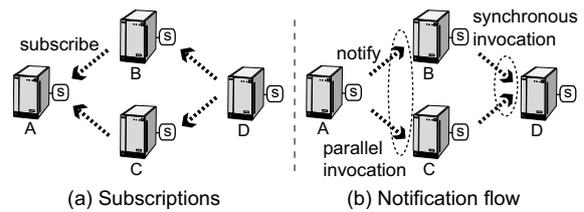


図 2: Chain of application invocations

- 複数サービスへのメッセージ通知により、アプリケーションの並列実行を実現できる。
- 通知予約先の情報を保持することで、アプリケーションの同期実行を実現できる。

3.3 アプリケーション間のデータ交換

提案するアプリケーション連携の枠組みにおけるアプリケーション間のデータ交換は大きく、交換される入出力ファイルをエンドユーザが指定する方法、データ形式の変換方法をエンドユーザが定義する 2 つによって実現する。交換される入出力ファイルのデータ形式が同一の場合には前者が、異なる場合は後者が用いられる。アプリケーション間のデータ交換は、エンドユーザの指示に従い、各サービスにおいてアプリケーション実行前に行われる。これにより複数サービスにおいてアプリケーション実行が並列に行われる際も、アプリケーション実行前に各サービスにおいてデータ交換を並列に実現できる。

3.3.1 交換される入出力ファイルの指定

あるサービス A の出力ファイルをそのまま、サービス B の入力ファイルとする場合、エンドユーザはあらかじめサービス B に対し、サービス A の出力ファイルを取得するよう指示する。これによりサービス B では、状態変更通知を受け取ってアプリケーションを実行する前に、入力ファイルの取得を行うことができる。

3.3.2 データ形式の変換方法の定義

データ形式の変換は一般に、出力ファイルから必要なデータの抽出と、抽出されたデータを用いた入力ファイルの生成により行われる。エンドユーザはあらかじめ、各サービスの入出力ファイルのデータ形式を参照し、データ抽出用と入力ファイル生成用の 2 つのアプリケーションを用意する。データ形式を変換してサービス A, B 間で入出力ファイルを交換する場合、3(a) に示すように、エンドユーザはサービス A に対してデータ抽出用の、サービス B に対して入力ファイル生成用のアプリケーションをそれぞれ送信し、サービス B に対してはさら

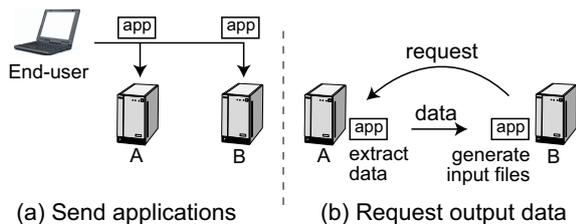


図 3: Translation of data structure

に、サービス A からデータを取得するよう指示する。これによりサービス B ではアプリケーション実行前に、3(b) に示すように、サービス A に対してデータ要求を行い、サービス A におけるデータ抽出、抽出したデータの返信、サービス B における入力ファイルの生成の 3 つの手順により、データ形式の変換が実現される。

3.4 Application Igniting System

Application Igniting System の概要を図 4 に示す。本システムにおけるアプリケーション連携は、3.2 節で述べたアプリケーション実行の連鎖と、3.3 節で述べたアプリケーションの入出力ファイルの交換により実現される。本システムではアプリケーション連携を補助することを目的に、以下の 3 つのサービスを用意している。

- Proxy サービス：エンドユーザからの全ての要求、ファイル転送を中継して実行する。プライベートネットワーク上のエンドユーザと、広域ネットワーク上のサービス群との橋渡しを行う。
- Index サービス：サービス群からアプリケーションの概要、入出力ファイルの数、データ形式などに関する情報を収集し、Proxy サービスを通じてエンドユーザに提供する。
- RFT サービス：高速で信頼性の高い第 3 者ファイル転送機能を提供する。サービス間のファイル転送は全てこのサービスを介して行われる。

4 まとめ

グリッド技術は近年、サービスの連携に主眼をおいた標準化が図られている。本研究ではアプリケーション連携に特に着目し、Web サービスを基盤としたグリッド上のアプリケーション連携の枠組みを提案した。アプリケーション所有者は提案する枠組みに従い、既存アプリケーションをその実行環境とともに、共通のインターフェース、状

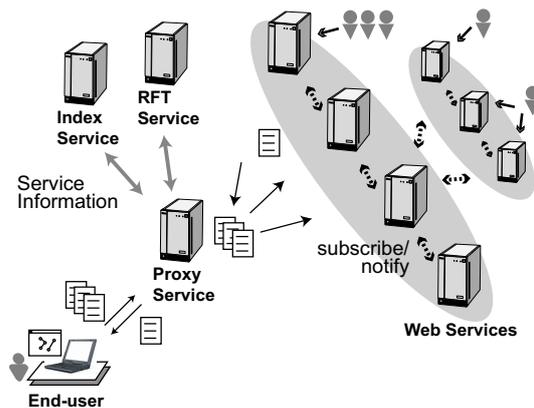


図 4: Overview of the Application Igniting System

態、機能を有する Web サービスとして表現することで、Notification を用いたアプリケーション実行の連鎖とアプリケーション間のデータ交換が実現でき、エンドユーザは広域ネットワークを通じて任意にアプリケーション連携の設計を行うことが可能となる。また提案する枠組みに基づいた問題解決システムである Application Igniting System について述べ、本システムを用いることにより既存アプリケーションの連携をプラットフォームとする新しいシステムの構築が容易に行えることが期待できる。

参考文献

- 1) Foster, I., Kesselman, C. and Tuecke, S., *The Anatomy of the Grid : Enabling Scalable Virtual Organizations*, International Journal of Supercomputer Applications, 2001.
- 2) Foster, I., Kesselman, C., Nick, J. and Tuecke, S., *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*, Globus Project, 2002.
- 3) Global Grid Forum: <http://www.gridforum.org/>
- 4) Foster, I. and et.al., *The Open Grid Services Architecture, Version 1.0*, Global Grid Forum OGSA-WG, GFD-I.030, 2005.
- 5) Tuecke, S. and et.al., *Open Grid Services Infrastructure (OGSI) Version 1.0*, Global Grid Forum OGSI-WG, GFD-R-P.15, 2003.
- 6) Czajkowski, K. and et.al., *The WS-Resource Framework, Version 1.0*, 2004, <http://www.oasis-open.org/committees/download.php/6796/ws-rsf.pdf>
- 7) Graham, S. and et.al., *Publish-Subscribe Notification for Web services, Version 1.0*, 2004, <http://www.oasis-open.org/committees/download.php/6661/WSNpubsub-1-0.pdf>