2005－HPC－103（11）
2005／8／3

Application Igniting System: グリッド上のアプリケーション連携システム

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，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システムとして，Application Igniting System を提案する．提案システムでは，広域ネットワーク上において既存のアプリケーションを Web サービスとして表現し，Notification の枠組みを用いてこれらを連携する．またエンドユーザにより設計されたグリッド上のアプリケーション連携を，アプリケーションから呼び出して利用することが可能な仕組みも提供する．提案システムを用いることにより，Web サービスとして表現された既存アプリケーションをプラットフォームとする，新しいシステム構築を容易に行えることが期待できる．

Application Igniting System: Application Integration System on the Grid

Hisashi SHIMOSAKA †　Tomoyuki HIROYASU ††　Mitsunori MIKI ††

† Graduate School of Engineering, Doshisha University　†† Department of Knowledge Engineering, Doshisha University

Recently, Grid technologies have been standardized with the emphasis on services integration. This paper especially focuses on application integrations. Then, application integration system on the Grid, application igniting system, is proposed. In the proposed system, existing applications are expressed as the Web service on the wide area network and integrated by using the notification framework. The proposed system also provides the function for an application to use another application. Using the proposed system, it is expected that new system which consists of existing applications can be developed easily.

1　はじめに

複雑な問題の効率的な解決には，複数の高性能なアプリケーションを連携して利用することが不可欠である．例えば自動車や航空機設計などに代表される複合領域最適設計問題を考えた場合，構造解析や流体解析，最適化，可視化のための高性能な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．一般にこのような複合領域にわたる問題解決を支援するシステムにおいては，複数の既存アプリケーションをそれぞれ，システム上のサービスとして表現し，アプリケーションへの統一的なアクセス手段と連携方法を提供する．エンドユーザはアプリケーションの情報をシステムから取得し，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の実行には専用の装置やチューニングされた計算環境を必要とする場合があり，広域ネットワークを通じてシステムを構築することが有効であると考えられる．

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) において Open Grid Services Architecture(OGSA)[2, 3] 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，アプリケーションの連携に特に着目し，グリッド上のアプリケーシ

ョン連携システムとして Application Igniting System を提案する．提案システムではまず，異なる人や組織から提供される複数の既存アプリケーションを，広域ネットワーク上において共通のインターフェースと機能を持った Web サービスとして表現し，Notification の枠組みを用いて連携する．またエンドユーザにより設計されたグリッド上のアプリケーション連携を，アプリケーションから呼び出して利用することが可能な仕組みも提供する．提案システムを用いることで，Web サービスとして表現された既存アプリケーションをプラットフォームとする新たなシステム構築を容易に行えることが期待できる．

2 Web サービスと Notification

2.1 Web サービスとグリッド

近年，広域ネットワークを通じて計算資源や情報資源，人や組織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おり，すでに多数のグリッドミドルウェアが開発，利用されている．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 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)[4] や Web Services Resource Framework / Web Services Notification (WSRF/WSN) [5, 6] があり，両仕様は本質的に同等の機能を提供する．OGSA では Web サービスにおいて状態の取り扱いを必要としており，OGSI では Web サービス自身に状態を付加したグリッドサービスとして，WSRF では状態を持つリソースと状態を持たない Web サービスとの組み合わせにより，それぞれ状態管理を実現している．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の実行状況を扱う．アプリケーションの連携には，アプリケーションの実行状況を他のサービスに通知する仕組みが必要であり，これに Notification を用いる．

2.2 Notification

Notification はサービスの状態変化を，サービス利用者や他のサービスに通知するための枠組みである．Notification は，OGSI および WSN においていくつかの仕様が存在するが，最も単純な Notification の仕組みを Fig. 1 に示す．ここではまず，サービス A の状態の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，状態の変更が通知 (notify) される．これにより，サービス A の処理と処理結果をきっかけとして，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．

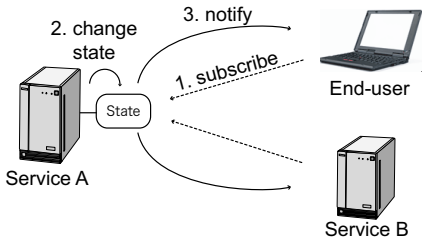

Fig. 1 Overview of the notification

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り，利点として次の 3 点が挙げられる．

- サービス構築時にメッセージの通知先を規定する必要がなく，登録によって通知先を動的に変更できる．
- ある 1 度の状態変化により，複数の通知先に同一メッセージを通知して，並列に処理の連鎖を実行できる．
- メッセージの通知元の情報をもとに，通知先では連鎖して実行する処理を変更できる．

3 Application Igniting System

本研究では，異なる人や組織から提供される複数の既存アプリケーションを，広域ネットワーク上において Web サービスとして表現し，エンドユーザが任意にそれらのサービスを連携して利用することのできる Application Igniting System を提案する．まず最初に提案システムの概要を述べ，そ

の後，アプリケーション連携の設計方法と仕組みについて述べる．またエンドユーザにより設計されたアプリケーション連携を，アプリケーションから呼び出して利用することを可能とする仕組みについても述べる．

3.1 Application Igniting System の概要

提案システムの概要を Fig. 2 に示す．提案システムではまず，異なる人や組織から提供される複数の既存アプリケーションを，広域ネットワーク上において共通のインターフェースと機能，状態を有する Web サービスとして表現し，エンドユーザはサービス群の情報を取得して，アプリケーション連携を設計する．そのため，エンドユーザはアプリケーションの実行ファイルを保持せずとも，アプリケーション連携を設計し，問題解決を図ることができる．提案システムにおけるアプリケーション連携は，2.2 節で述べた Notification と，アプリケーションの入出力をサービス間で交換することにより実現される．また提案システムにおける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 つのサービスが用意されている．

- 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の橋渡しを行う役割も担う．
- 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利用して行われる．

3.2 アプリケーション連携の設計方法と仕組み

3.2.1 アプリケーション連携システムの必要要件

一般にアプリケーションの実行は，複数の入力ファイルを必要とし，複数の出力ファイルを生成する．アプリケーション連携においては，アプリケーションのソースコードがなかったり，技術的にソースコードを改変することが困難な場合が多いため，アプリケーション間で入出力ファイルを

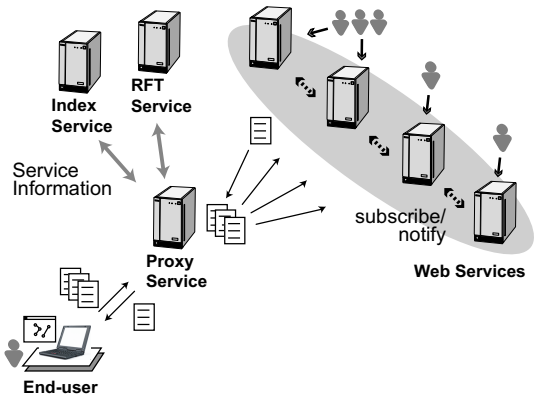

Fig. 2 Overview of the proposed system

交換することにより連携を行う方法が最も自然であり，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もとに，入力ファイルを用意し，該当するアプリケーションにあらかじめ送信しておく．その後，エンドユーザはアプリケーションを適切な順で実行し，目的を達成する．その際，アプリケーションはエンドユーザの指示に従い，実行前に指定されたアプリケーションの出力ファイルを入力ファイルとして取得する．アプリケーション間の入出力ファイルのデータ形式が異なる場合は，出力ファイルの一部分を抜き出し，その情報をもとに入力ファイルを作成しなおすことで，データ形式の変換を図る．まとめると，エンドユーザは大きく次の 4 点を決定してアプリケーション連携を設計し，アプリケーション連携システムではそれらの情報をもとに，エンドユーザが設計するアプリケーション連携を実現できる必要がある．

1. エンドユーザが用意する入力ファイル
2. アプリケーションの実行順序
3. アプリケーション間の入出力ファイル転送
4. 転送される入出力ファイルのデータ形式の変換

3.2.2 アプリケーション連携の設計

3.2.1 節で述べたアプリケーション連携システムの必要要件を満たすために，提案システム上の各

サービスはアプリケーションの実行状況を保持するための 1 つの状態と，他のサービスからの状態変化の通知によりアプリケーションを実行するためのインターフェース，データ変換のための仕組みを共通して有する．エンドユーザは次の 4 つの手順により，アプリケーション連携を設計する．

1. エンドユーザが用意した入力ファイルの送信
2. サービス間の通知依頼 (subscribe) による，アプリケーション実行順序の決定
3. アプリケーション実行前に取得する，他のアプリケーションの出力ファイルの指定
4. データ形式を変換するためのアプリケーションの転送

Fig. 3 に，エンドユーザが 3 つのサービス A，B，C の連携を設計する例について示す．全てのサービスは 1 つの入力ファイルから 1 つの出力ファイルを生成する．またサービス B の入力ファイルはサービス A の出力ファイルを，サービス C の入力ファイルはサービス B の出力ファイルを利用する．そのためサービスの実行順序は A，B，C となる．さらにサービス B の入力ファイルとサービス A の出力ファイルのデータ形式は同じとし，サービス C の入力ファイルとサービス B の出力ファイルのデータ形式は異なるものと仮定する．

Fig. 3 の連携においてエンドユーザはまず，サービス A に対して入力ファイルを送信する (Fig. 3(a))．次にサービス B はサービス A に対して，サービス C はサービス B に対して通知依頼 (subscribe) を行うよう，エンドユーザは各サービスに指示する (Fig. 3(b))．ここで各サービスは，指定されたサービスに対して，アプリケーションの実行状況を保持するための状態について，変更の通知を依頼する．その後エンドユーザはサービス B に対して，アプリケーション実行前にサービス A の出力ファイルを取得するよう指示する (Fig. 3(c))．一方でサービス B からサービス C へのファイル転送にはデータ形式の変換が必要なため，サービス B に出力ファイルから必要なデータを抜き出すためのアプリケーションを，サービス C に抽出されたデータから入力ファイルを生成するためのアプリケーションをそれぞれ送信しておき，さらにサービス C に対してアプリケーション実行前に，サービス B からアプリケーション実行結果を取得するよう指示する (Fig. 3(d))．

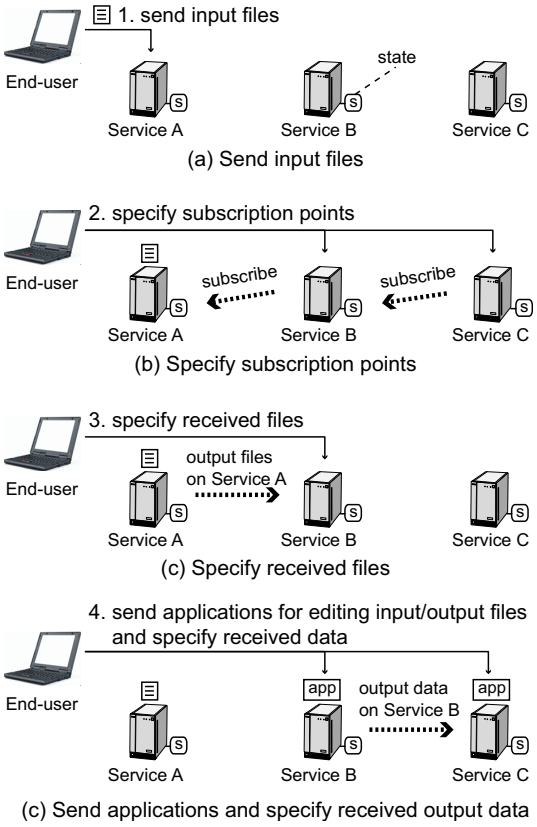

Fig. 3　Design of applications integration on the proposed system

3.2.3　アプリケーション連携の仕組み

3.2.2 節におけるアプリケーション連携の設計例において，エンドユーザがサービス A のアプリケーション実行を指示した際の，アプリケーション連携の仕組みについて Fig. 4 に示す．

ここではまず，サービス A のアプリケーション実行により出力ファイルの生成と，状態変化の通知 (notify) がサービス B に対して行われる (Fig. 4(a))．これによりサービス B が連鎖的に実行される．サービス B はアプリケーション実行前にエンドユーザの指示に従い，サービス A の出力ファイルを入力ファイルとして取得する．その後アプリケーションを実行して出力ファイルを生成し，状態変化の通知 (notify) をサービス C に対して行う (Fig. 4(b))．サービス C はサービス B の状態変化の通知を受け，アプリケーション実行前にサービ

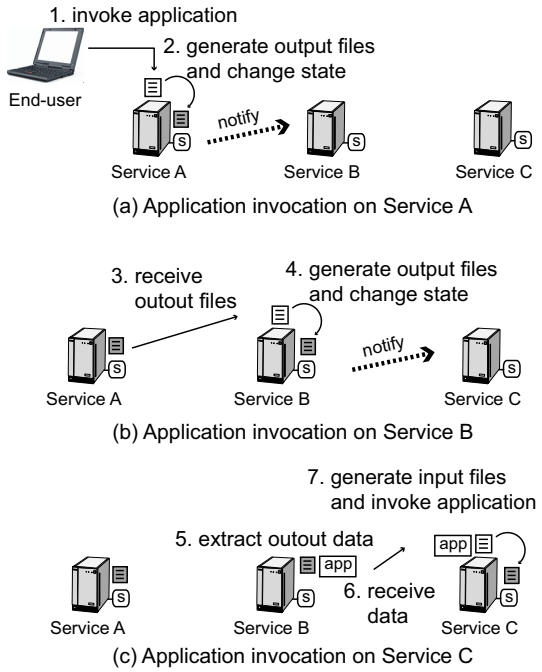

Fig. 4　Applications integration on the Grid

ス B に対してアプリケーション実行結果の取得を要求する．サービス B はエンドユーザから提供されたアプリケーションを利用して，出力ファイルからデータを抽出し，それをサービス C に返信する．サービス C では返信されたデータをもとに，同じくエンドユーザから提供されたアプリケーションを利用して入力ファイルを生成し，アプリケーションを実行する (Fig. 4(c))．アプリケーション実行後は，出力ファイルの生成と状態の更新が行われるため，エンドユーザはサービス C の状態変更の通知を受けることで，アプリケーション連携の終了を確認できる．

3.3　アプリケーション連携の呼び出しと利用

3.2 節ではエンドユーザが設計したアプリケーション連携を，エンドユーザ自身が利用する仕組みについて述べた．提案システムではさらに，アプリケーションから連携を呼び出し，利用するための仕組みを提供する．

3.3.1　アプリケーション連携呼び出しの必要要件

アプリケーション実行中に他のアプリケーションを呼び出し，利用することを前提としたアプリケーションは一般に，アプリケーション呼び出し部分を任意に変更できるよう汎用的に作成されており，次のように一般化して考えることができる．

1. 呼び出すアプリケーションの入力ファイルを書き出す．
2. 特定のコマンドもしくは API を実行し，他のアプリケーションを呼び出す．
3. コマンド実行によって得られた出力ファイルを読み込む．

そのためアプリケーション連携呼び出し部分においては，一時的に書き出された入力ファイルを利用し，エンドユーザが設計したアプリケーション連携を呼び出した後，アプリケーション連携の実行結果を指定された出力ファイルに書き出すという一連の機能が必要となる．

3.3.2　アプリケーション連携呼び出しの設計

3.3.1 節で述べたアプリケーション連携呼び出し部分の必要要件を満たすために，提案システムでは新たに，アプリケーション連携の入力状況を保持するための状態を用意し，さらに他のサービスの状態変化の通知を受け取り，アプリケーション連携結果を取得して利用するためのインターフェースを，提案システム上のサービスにそれぞれ付加する．またエンドユーザが設計したアプリケーション連携を，アプリケーションから呼び出すことを可能にする実行コマンドと API を提供する．

Fig. 5 に，エンドユーザが設計したアプリケーション連携を，アプリケーションから呼び出す際の設計について示す．呼び出すアプリケーション連携は 3.2 節で述べたものと同じであり，それをサービス D のアプリケーションから呼び出すことを想定する．Fig. 5 においてエンドユーザはまず，アプリケーション連携の最初のサービスであるサービス A からサービス D に対して，新しく用意されたアプリケーション連携の入力状況を保持するための状態について，更新の通知依頼 (subscribe) を行うよう指示する．一方でエンドユーザは，サービス D からアプリケーション連携の最後のサービスであるサービス C に対して，アプリケーションの実行状況を保持するための状態について，更新の通知依頼 (subscribe) を行うよう指示する．3.2 節と同様に，サービス間の入出力ファイルによる情報交換はあらかじめエンドユーザが指示する．

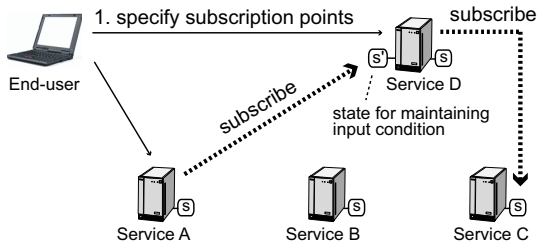

Fig. 5　Design for invoking applications integration

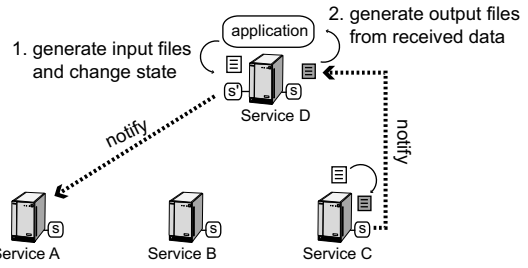

Fig. 6　Invocation of applications integration

3.3.3　アプリケーション連携呼び出しの仕組み

3.3.2 節におけるアプリケーション連携の設計において，サービス D のアプリケーションがアプリケーション連携の入力ファイルを書き出し，提案システムが提供する実行コマンド，もしくは API を実行した際の，アプリケーション連携の仕組みについて Fig. 6 に示す．提案システムが提供する実行コマンド，もしくは API が実行されることにより，アプリケーション連携の入力状況を保持するための状態がサービス D において更新される．これにより，サービス A に対して状態変更の通知 (notify) が行われ，サービス A の実行が開始される．このことは，エンドユーザが設計したアプリケーション連携の呼び出しが実現されたことを意味する．サービス A はエンドユーザの指示に従い，入力ファイルを用意してアプリケーションの実行を開始する．アプリケーション連携終了時には，サービス C においてアプリケーション実行状況を保持するための状態が変更され，サービス D に状態変更が通知される．サービス D はここで，エンドユーザの指示に従いアプリケーション連携結果の取得を行った後，それを出力ファイルとして書き出し，アプリケーションに通知する．アプリケーションは書き出された出力ファイルを読み込み，アプリケーション連携結果として利用する．

4　まとめと今後の課題

グリッド技術は近年，サービスの連携に主眼をおいて標準化が図られている．本研究ではアプリケーション連携に特に着目し，グリッド上のアプリケーション連携システムとして，Application Igniting System を提案した．提案システムではまず，異なる人や組織から提供される複数のアプリケーションを広域ネットワーク上において，共通のインターフェース，状態，機能を持つ Web サービスとして表現し，Notification の枠組みを用いて，それらを連携させる仕組みを提供する．さらには，エンドユーザが設計したアプリケーション連携を，アプリケーションから呼び出して利用することを可能とする仕組みもあわせて提供する．今後の課題としては，Globus Toolkit を用いた提案システムの実装と性能評価が挙げられる．

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) Foster,I., Kishimoto,H., Savva,A., Berry,D., Djaoui,A., Grimshaw,A., Horn,B., Maciel,F., Siebenlist,F., Subramaniam,R., Treadwell,J. and Von Reich,J., *The Open Grid Services Architecture, Version 1.0*, Global Grid Forum OGSA-WG, GFD-I.030, 2005.

4) Tuecke,S., Czajkowski,K., Foster,I., Frey,J., Graham,S., Kesselman,C., Maquire,T, Sandholm,T., Snelling,D and Vanderbilt,P., *Open Grid Services Infrastructure (OGSI) Version 1.0*, Global Grid Forum OGSI-WG, GFD-R-P.15, 2003.

5) Czajkowski,K., Ferguson,F.D., Foster,I., Frey,J., Graham,S., Sedukhin,I., Snelling,D., Tuecke,S. and Vambenepe,W., *The WS-Resource Framework, Version 1.0*, 2004, http://www.oasis-open.org/committees/download.php/6796/ws-wsrf.pdf

6) Graham,S., Niblett,P., Chappell,D., Lewis,A., Nagaratnam,N., Parikh,J., Patil,S., Samdarshi,S., Sedukhin,I., Snelling,D., Tuecke,S., Vambenepe,W. and Weihl,B., *Publish-Subscribe Notification for Web services, Version 1.0*, 2004, http://www.oasis-open.org/committees/download.php/6661/WSNpubsub-1-0.pdf