

ミドルウェア連携による計算・ネットワーク資源の 日米間グリッドコアロケーション実験

竹房 あつ子^{t1} 林 通秋^{t2} 築島 幸男^{t3}
岡本 修一^{t4} 柳田 誠也,^{t1,t5} 宮本 崇弘^{t2}
平野 章^{t3} 鮫島 康則,^{t4,t3} 中田 秀基^{t1}
谷口 篤,^{t4,t3} 工藤 知宏^{t1}

グリッドでは、計算資源に関する管理・利用技術は成熟しつつあるものの、ネットワークの帯域を資源として管理する技術は実験段階にあり、特に異なるネットワークドメインに属する拠点間の帯域を自動的に提供するの是非常に困難である。また、グリッドコアロケーションシステムは複数開発されているが、異なるシステム間で連携し、それぞれの管理ドメインの計算・ネットワーク資源を横断的に利用する試みはこれまでない。我々は、ネットワークの帯域を予約するためのインタフェース GNS-WSI を規定することを目的とした G-lambda プロジェクトを推進しており、複数ドメインに跨るネットワーク資源を確保することを考慮し、新たに GNS-WSI2 を規定した。G-lambda が前提とするアーキテクチャでは、グリッド資源スケジューラが複数のネットワークドメインおよび計算資源の資源管理マネージャと連携することにより、ドメインを超えた資源のコアロケーションが可能になる。また、本報告では、米国の Enlightened Computing プロジェクトと共同で行った、ネットワークと計算機資源の同時事前予約実験について述べる。本実験では、日米の複数ドメインに跨る計算資源およびネットワーク資源をミドルウェア連携により事前予約で確保し、動的に構築したグリッド環境上でアプリケーションを実行することに成功した。

An Experiment of Trans-Pacific Grid Computing and Network Resource Co-allocation in Corporation with Middlewares

ATSUKO TAKEFUSA,^{t1} MICHIAKI HAYASHI,^{t2} YUKIO TSUKISHIMA,^{t3}
SHUICHI OKAMOTO,^{t4} SEIYA YANAGITA,^{t1,t5} TAKAHIRO MIYAMOTO,^{t2}
AKIRA HIRANO,^{t3} YASUNORI SAMESHIMA,^{t4,t3} HIDEMOTO NAKADA,^{t1}
ATSUSHI TANIGUCHI,^{t4,t3} and TOMOHIRO KUDOH ^{t1}

While Grid technology for computing resource management is becoming mature, network resource management is still in an early phase of experiments. Especially, automated bandwidth provisioning between end points in different network domains is difficult. On the other hand, even though multiple Grid co-allocation systems have been developed and reported, no attempt have been made to allocate computing and network resources over multiple domains by inter-working of different management systems of the domains. We have been promoting the G-lambda project, in which we are defining an interface for bandwidth reservation. To realize provisioning of bandwidth between multiple domains, we defined a new version of the interface, GNS-WSI2. In the G-lambda architecture, inter-domain resource co-allocation is realized by inter-working between a Grid resource scheduler and resource managers of each network and computing domains. In this report, we introduce an experiment of co-allocation of bandwidth between Japan and the US and computing resources in Japan and the US. In cooperation with the US Enlightened Computing project, we successfully conducted the experiment using resources belong to multiple domains, and applications are executed on the dynamically co-allocated Grid environment.

^{t1} 産業技術総合研究所 National Institute of Advanced Industrial Science and Technology (AIST)

^{t2} 株式会社 KDDI 研究所 KDDI R&D Laboratories Inc.

^{t3} 日本電信電話株式会社 NTT Network Innovation Laboratories

^{t4} 情報通信研究機構 National Institute of Information and Communications Technology (NICT)

^{t5} 原所属：数理工研 SURIGIKEN Co., Ltd.

1. はじめに

グリッドコアロケーションシステムは、ユーザの要求に応じて広域に分散した資源を同時に確保し、1つのグリッド環境として提供する。グリッドを構成する資源は計算機、ネットワーク、ストレージ、実験装置など、多様であり、また各資源は異なる組織(ドメイン)から提供される。よって、分散する複数資源の同時確保を実現するには、各資源の資源マネージャが標準的なインタフェースで事前予約による資源提供することが求められる。

計算資源においては、従来からクラスタ型計算機を複数ユーザで有効に利用することを目的として、多くのキューイングスケジューラが開発されており、それ自体が事前予約機能を備えていたり、あるいはプラグインスケジューラにより事前予約を実現することができる。一方、ネットワーク資源では一般に電話や電子メール等による事前の交渉の後、オペレータによる手動設定で提供されており、サービスインタフェースからの自動的な資源提供は実験段階にある。さらに、拠点間のネットワークが複数のドメインに跨るような場合、その間の帯域を自動的に確保するにはさらに複雑な事前の調整が必要となる。

また、各資源スケジューラが提供する資源を統合してユーザに提供するグリッドコアロケーションシステムは複数開発されているが、異なるコアロケーションシステムが連携してそれぞれの管理ドメインの計算・ネットワーク資源を横断的に利用する試みはない。

本研究では、複数ドメインに跨るネットワーク資源を確保することを考慮し、G-lambda¹⁾プロジェクトで新たに GNS-WSI ver. 2(GNS-WSI2)を規定し、KDDI 研, NTT, 産総研が参照実装した²⁾。また、Enlightened Computing プロジェクト³⁾(以降、Enlightened)と共同で、日米間に跨る計算・ネットワーク資源をミドルウェア連携により事前予約で確保し、動的に構築したグリッド環境上でアプリケーションを実行する実証実験を行った。

G-lambda では、ネットワーク資源の事前予約を行うための標準ウェブサービスインタフェース GNS-WSI を規定している^{4),5)}。新たに規定した GNS-WSI2 では、標準化が進められている WSRF⁶⁾の採用と、上位コアロケータからの分散トランザクションに対応するため、2相コミットでの手続きを実現した。

実証実験では、G-lambda と Enlightened のグリッドコアロケーションシステムおよび資源マネージャを連携し、日米間に跨る資源を事前予約で同時確保し、動的に構築したグリッド環境上でアプリケーションを実行することに成功した。各ミドルウェアはそれぞれ異なるシステムアーキテクチャ、異なるインタフェースにより資源を提供している。各ミドルウェアを連携

して異なるドメインの資源を確保するため、ラッパモジュールを開発した。また、分散する各資源マネージャで確保した資源の状況を監視するため、予約資源モニタサービスを開発した。

2. GNS-WSI2 (GNS-WSI ver. 2)

G-lambda プロジェクト¹⁾は産総研, KDDI 研, NTT, および NICT の共同研究であり、2004 年 12 月から開始した。ネットワークオペレータの資源マネージャ(NRM)がグリッドミドルウェアなどのアプリケーションに対して、資源を事前予約で提供するための標準ウェブサービスインタフェース GNS-WSI(Grid Network Service - Web Services Interface)を確立することを目的としている^{4),5)}。

GNS-WSI ver. 1(GNS-WSI1)では、ネットワーク資源提供に必要なパラメータ群と予約資源の状態遷移を定義し、事前資源予約・修正・解放手続きをウェブサービスインタフェースで規定した。ウェブサービス自体はステートレスであるため、予約 ID を各オペレーションの引数として渡すことにより、NRM 側で各予約インスタンスを識別した。

新たに規定した GNS-WSI2 では、資源パラメータは GNS-WSI1 を継承しつつ、ステートフルなウェブサービスのために標準化が進められている WSRF (Web Service Resource Framework)⁶⁾を採用した。WSRF では、エンドポイントリファレンス (EPR) を用いて各サービスにおけるインスタンスを識別することができる。GNS-WSI2 では、EPR により各バスに対する予約情報を識別する。

また、GNS-WSI2 では 2 相コミットでの資源予約・修正・解放手続きを提供する。上位のコアロケータが分散する資源マネージャの管理する資源をトランザクションにより同時確保するためには、各資源マネージャによる 2 相コミットによる手続きのサポートが不可欠であるためである⁷⁾。

GNS-WSI2 を用いた Client と NRM の間での予約手続きの手順を以下に示す。Client はグリッド資源スケジューラなど、NRM の提供する資源を確保するものとなる。

- (1) Client は資源予約サービスインスタンス(予約リソース)生成要求を NRM に送ると、NRM が予約リソースを生成し、そのリソースへの EPR を返す。予約リソースには要求した Client の資源予約に関するプロパティ情報が格納される。
- (2) (1) で受け取った EPR に対して、Client が資源予約要求を送ると、NRM はこの要求に対する情報を管理するコマンドリソースを生成し、その EPR を Client に返す。
- (3) (2) で受け取った EPR を用いて、現在の予約処理状況を調べる。NRM から Prepared が返ると、

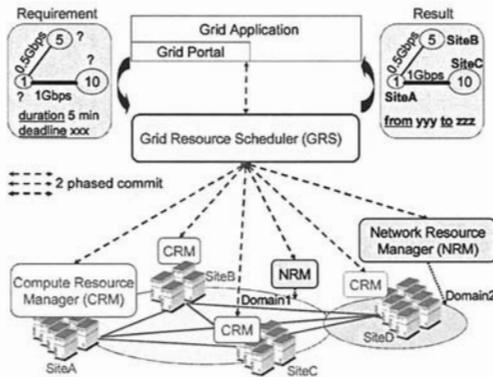


図1 グリッドコアロケーションシステムの概要

予約準備完了していることがわかる。

- (4) 予約を確定させるため、Client は (2) の EPR に対してコミット要求を送る。
- (5) (1) の EPR を用いて Client が NRM 予約状況を問い合わせ、Reserved (予約完了) であることを確認する。
- (1) は WSRF により、(3), (4) は 2 相コミットにより新たに発生した手続きである。

GNS-WSI2 の参照実装を、KDDI 研, NTT, 産総研が行った。産総研では、GNS-WSI2 のインタフェースモジュール GridARS-WSRF⁷⁾を開発した。GridARS-WSRF を用いることにより、ネットワーク資源提供者は煩雑な WSRF のコードを書くことなく、GNS-WSI2 の定義する 2 相コミット手続きにより資源を提供することができる。また、GridARS-WSRF は Globus Toolkit 4⁸⁾ の Java WS Core を用いて実装されており、予約情報の永続化と GSI による認証・認可も可能である。

3. ミドルウェア構成

本研究では、G-lambda と Enlightened が共同して、日米間に跨る計算・ネットワーク資源をミドルウェア連携により確保する実験を行った。

3.1 GL ミドルウェア構成

図 1 で示すように、G-lambda ではグローバル資源スケジューラ (GRS)、ネットワーク資源マネージャ (NRM)、計算資源マネージャ (CRM) の連携により、事前予約でユーザの要求する資源を同時確保する^{4),5)}。

GRS は、ユーザのクラスタ数、各クラスタのノード数、クラスタ間のネットワークの帯域と予約時間に関する要求情報をもとに、複数 CRM、複数 NRM と連携して、適宜分散する資源を割り当てる。

GRS-NRM 間では GNS-WSI2、GRS-CRM 間では JSDL⁹⁾ を事前予約のために拡張したもの、ユーザ-GRS 間ではこれらを集約したものをを用いる。いずれ

も WSRF をもとに 2 相コミットで手続きするため、GRS と複数 NRM、CRM 間で分散トランザクションが可能になる。また、階層的な 2 相コミット構造であるため、GRS も 1 つの資源マネージャとして他の管理ドメインに資源を提供することができる。

NRM は GNS-WSI2 を介して GRS からの要求に応じ、管理ドメイン内の拠点間の帯域を事前予約により提供する。NRM は GRS またはユーザに対して、拠点間の詳細なパス構成を隠蔽し、仮想的に一つのパスとして提供する。NRM ではドメイン内の各ラムダパスのスケジューリングと管理を行っており、ある予約の予約開始時刻になるとパスをアクティブ化する。

CRM もまた、GRS からの要求に応じ、管理下にある計算資源を事前予約により確保する。CRM には事前予約機能付きの既存キューイングスケジューラを用いる。予約時刻になると、あらかじめユーザにサブミットされていたジョブが確保された資源上で起動する。

実験では、GRS には産総研の開発する GridARS(Grid Advance Reservation-based System framework)⁷⁾、NRM には KDDI 研と NTT が開発する NRM、CRM は GridARS-WSRF、産総研の開発する PluS¹⁰⁾、SGE¹¹⁾ を用いている。PluS は既存キューイングスケジューラに事前予約機能を付加するプラグインスケジューラである。

3.2 EL ミドルウェア構成

Enlightened Computing プロジェクトは 2005 年に設立され、MCNC, LSU, NCSU, RENCI, Cisco, AT&T, Calient が参加している。アプリケーションからの動的な事前予約要求に対応した、光コントロールプレーンとグリッドミドルウェアの統合を目的とした学際的な共同研究を NSF の資金を得て行っている。Enlightened のゴールは、ネットワークに接続された地理的に分散して配置された高性能計算装置や科学的な観測装置を、動的、適応的かつ最適に連携させて利用する手法の確立にある。

Enlightened では、HARC (Highly-Available Robust Co-scheduler)¹²⁾ と呼ばれるグリッドコアロケーションシステムを用いて分散する資源マネージャの管理する資源を確保する。HARC では、Acceptor と呼ばれるコアロケータと資源マネージャ間で Paxos コミットプロトコル¹³⁾を用いている。Paxos コミットでは、コアロケータとして複数の Acceptor を配備し、Acceptor 側の耐故障性を高めている。各資源スケジューラは HARC の資源マネージャの下に配備する。実験では、HARC の NRM には Enlightened プロジェクトで開発するネットワーク資源スケジューラ、HARC の CRM には MAUI と TORQUE を用いている。

GL ミドルウェアと HARC との主な違いは以下のとおりである。

- (1) G-lambda は WSRF(SOAP)、HARC は REST (XML over HTTP) がベースとなる。

- (2) HARCの Acceptorにはスケジューリング機能はなく、上位のコーディネータが行う。
- (3) HARCでは異なる資源も授受するテキスト内で一緒に指定することができるが、G-lambdaでは資源ごとに異なる WSDL を用意し、異なるスタブライブラリで各資源に関する手続きを行う。

4. 複数ドメインでの異種ミドルウェア連携

一般に、ネットワークは複数の管理ドメインから構成され、また、各プロジェクトごとに採用しているミドルウェアも異なる。よって、複数ドメインに跨る拠点間の帯域確保と、ミドルウェア連携の課題がある。

4.1 複数ドメインに跨る拠点間の帯域の確保

ネットワークの管理ドメインが複数ある場合、ドメインごとに独立の NRM が存在する。異なるドメインに属する拠点間の帯域予約をする場合、以下のような方法が考えられる。

- (1) 下位ネットワーク制御層での連携
- (2) ネットワーク資源マネージャ間での連携
- (3) 上位スケジューラ(ユーザ)での連携

(1) ではユーザは複数ドメインを意識する必要がなく、制御層のプロトコルによるトラフィックエンジニアリングが可能だが、現状では制御層では事前予約をサポートされおらず、ドメイン間の密接な連携が要求されるために競合事業者間では困難である。(2) は(1) 同様にユーザからは複数ドメインであることが隠蔽されるが、要求を受け付けた NRM は自ドメインに都合の良い経路を設定する可能性がある。(3) はユーザが詳細なネットワーク構成に関する知識を持つ必要があるが、ユーザ側でドメイン間接続を制御できる、下位層での連携は必要ないという利点がある。

本実験では、(3) の方法を採用した。複数のドメインに跨る拠点間の帯域を確保する場合、GRS が拠点間のネットワークドメイン情報を事前に取得し、関係するドメインの NRM に対して、計算資源とともにトランザクションで同時に事前予約手続きを行う。

4.2 ラッパモジュールによるミドルウェア連携

3.2 節で述べたように、G-lambda と Enlightened で採用しているコアロケーションシステムのアーキテクチャ、インタフェースが異なる。よって、図 2 に示すように、以下のラッパを経由して異なるプロジェクトの資源を確保するようにした。これにより、いずれのコアロケーションシステムも自システムと同じインタフェースにより異なるプロジェクトの資源を確保できるようにした。

GL→EL GNS-WSI ラッパ (NRM) GNS-WSI2 を介して HARC の Acceptor 経由で Enlightened のネットワーク資源を確保する。本ラッパは Enlightened が開発した。

GL→EL CRM ラッパ GridARS の CRM インタ

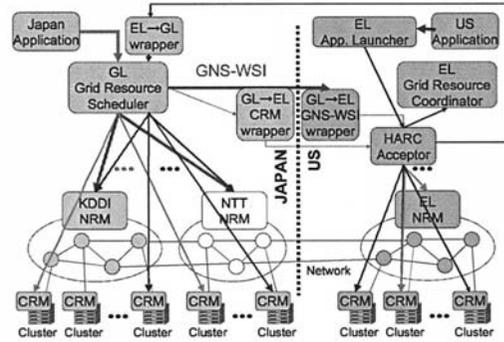


図 2 異種グリッドコアロケーションミドルウェア連携

フェースを介して HARC Acceptor 経由で Enlightened の計算資源を確保する。

EL→GL ラッパ (CRM/NRM) HARC 資源マネージャインタフェースおよび G-lambda の GRS を介して G-lambda のネットワーク・計算資源を確保する。

GL→EL GNS-WSI ラッパ および **GL→EL CRM ラッパ** では、GridARS コアロケーションシステムが提供する GridARS-WSRF インタフェースモジュールを用いている。EL→GL ラッパ (CRM/NRM) では、GridARS のシェルクライアントインタフェース⁷⁾を用いて GRS 下の資源を確保する。

4.3 予約資源モニタサービス

ラッパモジュールにより、EL, GL ミドルウェアを介して日米間に跨る異なるドメイン上の資源が確保可能となった。しかしながら、計算資源については既存キューイングシステムの実績から、予約時刻に資源が利用可能になることが期待できるが、なんらかのトラブルが発生する可能性がある。また、ネットワークの帯域をオンデマンドに提供する試みは実験段階にあり、計算資源の場合ほど安定しているとは言えない。従って、分散資源の予約状況を可視化することは、実証実験環境を構築する上で大きな助けとなる。よって、予約資源モニタサービスを構築した。

iGrid2005 での実証実験^{4),5)}では、ネットワーク資源のみを監視し、GMPLS のコントロールバケットを監視することにより予約状況を可視化した。しかしながら、本実験ではネットワークドメインが複数存在し、ネットワークの構成も複雑になるため、上記の方法では対処できない。また、計算資源の監視も必要である。よって、コアロケータ、NRM、CRM と連携して予約状況を収集する。

予約資源モニタサービスはビューアとアグリゲータからなる。EL, GL ミドルウェアでは、予約した資源の状態(予約済み/利用可能/予約時間終了/エラー)は各資源マネージャが、各ユーザの要求と資源予約のマッピング情報はコアロケータ側が管理している。アグリゲータが以下の各予約要求に関する XML 形式の

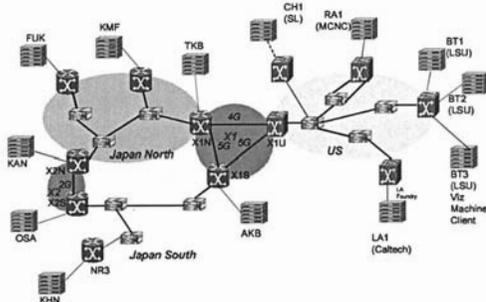


図3 日米間資源予約実証実験環境

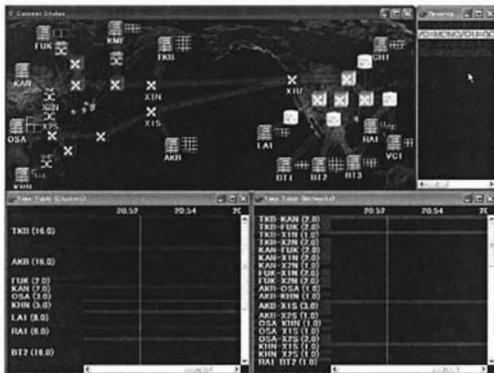


図4 予約資源モニターサービスビューア画面

情報を EL, GL ミドルウェアから収集して資源情報とマッピング情報をマージし、その情報をもとにビューアで出力する。

コアローケータ コアローケータ名 (EL/GL), CRM サイト名, NRM 名, 他のコアローケータ名, 各 RM, コアローケータでの資源予約 ID

NRM 開始・終了時刻, 端点情報, 帯域, 予約状況

CRM 開始・終了時刻, CPU 数, 予約状況

ビューアとアグリゲータは Java で実装した。アグリゲータには軽量な Java 組み込み HTTP サーバである OOWeb¹⁴⁾ を使い、ビューアとアグリゲータ間の通信は HTTP で行った。

5. 日米間資源予約実証実験

G-lambda と Enlightened が共同で、日米間に跨る計算・ネットワーク資源をミドルウェア連携により事前予約で確保する実証実験を行った。

5.1 実証実験の概要

図3に実験環境を示す。日本側は NICT が運営する JGNII ネットワークテストベッド上に KDDI 研 (Japan North) と NTT (Japan South) が管理するドメインを構築し、米国側は Enlightened のドメイン

(US) で構成される。また、つくば、秋葉原、上福岡、大阪、けいはんな、金沢、福岡、ロサンゼルス、ノースカロライナ、ルイジアナ、カリフォルニアにあるクラスタを用いた。NRM は各ドメインごとに開発し、ドメイン間の接続本数を管理するために別途産総研の開発した NRM を用意した。また、CRM は各サイトごとに用意した。

実証実験は次の手順で行われた。

- (1) G-lambda 側から資源予約し、予約結果を予約資源モニターサービスのビューア上で確認する。
- (2) Enlightened 側から資源予約し、予約状況をビューア上で確認する。
- (3) 予約時刻になると確保した資源が利用可能となり、各アプリケーションを実行する。ビューア上に現在確保されている帯域、計算機の CPU 数が表示される。
- (4) 各アプリケーションの実行結果が表示される。アプリケーションプログラムには、Enlightened ではブラックホールのビジュアル化、G-lambda は並列科学技術計算を用いた⁷⁾。

実験時のビューア画面を図4に示す。上部の地図画面上に現在利用可能となっている帯域、CPU 数が表示される。上部右側には各予約を要求したユーザの DN (Distinguished Name) が出力される。下部の左側には CPU 数、右側には帯域に関する予約情報 (拠点名、予約時刻) が出力されている。

本実験により、複数 NRM に対して GRS がトランザクションで同時に事前資源予約手続きを行うことにより、GNS-WSI2 を介して複数ドメインに跨る帯域を提供できることを示した。また、異なるコアローケーションシステムの連携により、日米間に跨る計算・ネットワーク資源を確保し、動的に構築した環境上での並列アプリケーションの実行に成功した。

5.2 議論

ミドルウェア連携によるコアローケーション実験に当たり、4.2 節で述べたように各ミドルウェアを連携するためのラッパモジュールを構築した。しかしながら、ラッパモジュールの構築には他のコアローケーションシステムの内部構造を把握しなければならず、簡単に実装できるわけではない。また、複数仮想組織以上の連携では、N×N のラッパを実装しなければならない。よって、ネットワークの予約に関する標準的なインタフェースを規定することが、非常に重要な課題となる。

予約資源モニターサービスによるモニタリングは、非常に直感的であり、実験環境構築においてデバッグ作業の助けとなった。ただし、本サービスは予約資源が利用可能になったかどうかのみを示すものであり、実際には資源が利用可能になってもアプリケーションが実行されないことが度々あった。その理由としては、設定上の問題、高負荷等により実行ジョブが開始しない、あるサイトのジョブが立ち上がらないために MPI

アプリケーションの一斉起動に失敗する、などがある。このような場合、手作業で一つ一つ原因を究明しなければならず、利用者の観点からは、より詳細なモニタリング情報の可視化が必要である。

オンデマンドに帯域を提供するため、各計算機では制御用とデータ通信用の2つのIPアドレスを割り当てている。制御用は常に利用可能であるが、データ通信用は予約時間内にしかコネクティビティがないため、並列プログラムを実行するためのコンフィグレーション設定が複雑になる、帯域を確保していない拠点間での通信のため、ルーティング設定を変更するなどの作業が必要となる、などの問題がある。また、現状ではIPアドレスに対して要求された帯域を提供するため、特にSMP(対称型マルチプロセッサ)のように1つの計算ノード上で複数ユーザのジョブを同時に実行するような場合、要求したユーザのみに予約された帯域を割り当てることはできない。これらの問題を解決する技術が期待される。

6. 関連研究

グリッドでネットワーク資源を考慮したコアロケーションはあまり行われていない。VIOLA¹⁵⁾ プロジェクトでは、本研究同様、グリッドで計算資源とネットワーク資源を事前予約で同時に確保し、提供することを目的としている。VIOLAではOGF(Open Grid Forum)で標準化がすすめられているWS-AgreementNegotiation仕様に基づくメタスケジューラをUNICORE¹⁶⁾ベースのグリッド環境で開発することを提案している。メタスケジューラからの計算・ネットワーク資源の確保は現状では独自のプロトコルにより行っている。また、グリッドでネットワーク資源の動的な確保も考慮したインターオペラビリティ実験は本実験のほかにはない。

7. まとめと今後の課題

本研究では、複数ドメインに跨るネットワーク資源を確保することを考慮し、G-lambdaプロジェクトで新たにGNS-WSI2を規定し、KDDI研、NTT、産総研が参照実装を行った。GNS-WSI2はWSRFベースであり、上位コアロケータからの分散トランザクションに対応するため、2相コミットでの資源予約・修正・解放手続きを可能にする。

また、G-lambdaとEnlightenedが共同で、日米間に跨る計算・ネットワーク資源をミドルウェア連携により事前予約で確保し、動的に構築したグリッド環境上でアプリケーションを実行する実証実験を行った。実証実験では、異なるアーキテクチャ、インタフェースのグリッドコアロケーションシステムと異なるネットワークドメインを管理するNRMおよびCRMを連携し、日米間に跨る資源を事前予約で同時確保する

ことに成功した。さらに、分散する各資源の状況を監視するため、予約資源モニタサービスも開発した。

謝辞 実証実験にあたり、テストベッド構築、ミドルウェア開発、アプリケーション実行に携わったEnlightened ComputingプロジェクトおよびG-lambdaプロジェクトの皆様には深く感謝いたします。

参考文献

- 1) G-lambda: <http://www.g-lambda.net/>.
- 2) Takefusa, A., Hayashi, M., Hirano, et al.: GNS-WSI2: Grid Network Service - Web Services Interface, version 2, OGF19 GHPN Research group (2007).
- 3) Enlightened Computing: <http://enlightenedcomputing.org/>.
- 4) Takefusa, A., Hayashi, M., Nagatsu, et al.: G-lambda: Coordination of a Grid Scheduler and Lambda Path Service over GMPLS, *Future Generation Computing Systems*, Vol. 22, pp.868-875 (2006).
- 5) 竹房あつ子, 林通秋, 長津尚英ほか: G-lambda: グリッドにおける計算資源と光バスネットワーク資源のコアロケーション, 情報処理学会研究報告 2006-HPC-105, pp.121-126 (2006).
- 6) OASIS Web Services Resource Framework (WSRF) TC: Web Services Resource 1.2 (WS-Resource) Committee Specification (2006).
- 7) 竹房あつ子, 中田秀基, 武宮博ほか: 異種の複数スケジューラで管理される資源を事前同時予約するグリッド高性能計算の実行環境, HPCS2007 論文集, pp.135-142 (2007).
- 8) Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems, *IFIP International Conference on Network and Parallel Computing*, Springer-Verlag LNCS 3779, pp.2-13 (2005).
- 9) Anjomshoaa, A., Brisard, F., Drescher, M., et al.: Job Submission Description Language (JSDL) Specification, Version 1.0 (2006).
- 10) 中田秀基, 竹房あつ子, 大久保克彦ほか: グローバルスケジューリングのための計算資源予約管理機構, HPCS2007 論文集, pp.127-134 (2007).
- 11) Grid Engine: <http://gridengine.sunsource.net/>.
- 12) HARC: <http://www.cct.lsu.edu/~maclaren/HARC/>.
- 13) Gray, J. and Lampton, L.: Consensus on Transaction Commit, Msr-tr-2003-96, Microsoft Research (2004).
- 14) OOWeb: <http://ooweb.sourceforge.net/>.
- 15) Barz, H.: Dynamic allocation of network resources in VIOLA, *VIOLA workshop* (2005).
- 16) UNICORE: <http://www.kfa-juelich.de/unicore/>.