2 グリッドを実現するグリッドミドルウェア基盤

鶴澤 武士　国立情報学研究所

グリッドミドルウェア

グリッド環境においては，たとえばユーザが既存のアプリケーションプログラムを実行して結果を得るという，ローカルな計算機上で実行する場合には OS 以外には特別なサービスが動作している必要がないような単純な機能を実現するだけでも，遠隔地にある計算資源をユーザが安全に利用するためのセキュリティ技術，ユーザが利用可能な計算資源群を探し出す機能，その中からユーザの目的に合うようにジョブに計算資源を割り当てるスケジューリングを行う機能，割り当てられた計算資源でのジョブ実行の管理を行う機能，などの支援が必要である．また，ユーザがグリッド環境でのアプリケーション管理やジョブの実行・監視を行うための利用環境も重要な役割を持っている．このようにグリッド環境においてユーザに対してある機能を提供するために動作するサービスやソフトウェア群をグリッドミドルウェアという．本稿では特に科学技術計算を行うためのグリッド環境の実現に必要となるミドルウェアについて解説する．

グリッド上のセキュリティ技術

グリッド環境では複数の組織に属するユーザが，資源を共有して相互に連携しながらある目的を達成するための仮想的な組織（VO）を形成し，その VO に属することで共有する計算資源等を利用することができる．VO に属する資源やユーザは一般には別々の実組織に属しているので，安全な利用のためにはユーザと資源は相互に適切に認証されなければならない．また，認証の結果に基づきユーザがどのように資源を利用できるかなどの認可の判断を行う機能が必要である．

■ プロキシ証明書とシングルサインオン，権限の委譲

ユーザがグリッド上の資源を利用する際には，資源を利用しようとするユーザの身分を資源側が認証するとともに，その資源が利用したい資源であることをユーザ側が認証する必要がある（相互認証）．公開鍵証明書を用いてこの認証を行う際には暗号化して格納されたユーザの秘密鍵を復号化するパスフレーズを入力する必要がある．相互認証はグリッド上の資源にアクセスするたびに行う必要があるため，毎回パスフレーズの入力を要求されないことが望ましい．公開鍵の仕組みを応用し，パスフレーズの入力を 1 回だけにするシングルサインオンの技術として，Grid Security Infrastructure（GSI）[1] がある．GSI ではユーザはグリッド環境を利用する際にユーザ証明書を元にプロキシ証明書を作成する．このときパスフレーズの入力を求められるが，一度プロキシ証明書を作成すると，プロキシ証明書の有効期限内ではユーザは再びパスフレーズの入力をせずにグリッド環境の資源にアクセスすることが可能となる．また，グリッド環境では複数のサービスが連携してユーザから依頼された処理を実行するため，あるサービスがユーザの権限で他のサービスに処理を依頼する場合が生じる．たとえば，ユーザがスケジューリング機能を持ったサービス経由でジョブを実行する場合，ユーザからジョブ実行依頼を受け取ったスケジューリングサービスは，ユーザの権限でジョブ実行サービスにジョブ実行の依頼を発行することになる．この場合，スケジューリングサービスがユーザの身分に成り代わってジョブ実行サービスにアクセスし相互認証することになるが，それを実現するにはスケジューリングサービスがユーザ名義のプロキシ証明書と秘密鍵を持っている必要があり，このときスケジューリングサービスはユーザの権限を委譲されているという．GSI はこのような権限の委譲を行う機能も提供する．

サイエンスグリッドを実現するミドルウェアの 1 つである NAREGI ミドルウェア [2] においても，GSI，シングルサインオン，権限の委譲の機能を利用することでミドルウェアを構成するサービス群が連携し動作している．以降，同様にサイエンスグリッドのためのミドルウェア技術の実例として NAREGI ミドルウェアの関連コンポーネントを紹介する．

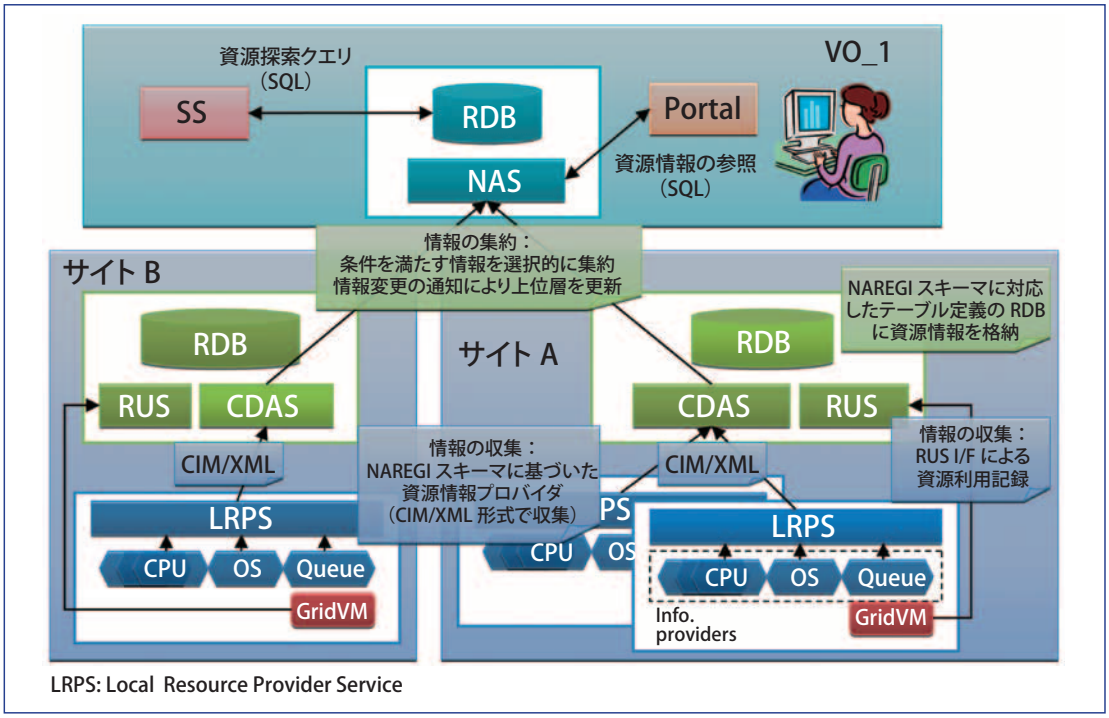


図 -1　NAREGI IS のサービス構成

■ プロキシ証明書の更新機能

グリッド環境上でもジョブの実行時間が数週間などの長時間ジョブを実行する要望もあり，その対応のため単純に長期間有効なプロキシ証明書を使うとプロキシ証明書が悪用されるリスクが高まる．そのためセキュリティ上望ましい短い有効期間のプロキシ証明書を用いて，より安全に長時間ジョブを実行するための機能が必要になる．この機能を実現するために，プロキシ証明書の有効期限を管理し無効化する前にプロキシ証明書の更新を行う手法がある．NAREGI ミドルウェアもこの手法を用いて，プロキシ証明書の有効期限が切れる前に新たなプロキシ証明書を発行し，更新を行う機能を提供している．

■ VO 管理

VO は複数の組織に属しているユーザがある目的を達成するために資源を共有する集合であり，VO の管理とは VO のメンバの管理とそのメンバに対する計算資源側での認可判断の仕組みを管理することである．ここでは Enabling Grids for E-sciencE（EGEE）プロジェクトによる Virtual Organization Membership Service（VOMS）[3] を用いた VO 管理の仕組みを紹介する．VOMS は VO のメンバとメンバの VO 内での Role や Capability 等の VO 属性を管理し，それらの属性の属性証明書の発行を行う．属性証明書はプロキシ証明書の拡張領域に格納可能である．属性証明書付きのプロキシ証明書を用いると，ユーザは自分が属する VO での VO 属性を計算資源側に伝達でき，計算資源側では伝達された VO 属性に従って認可判断を下すことができる．NAREGI ミドルウェアでも VOMS による VO 管理を行っている．計算機資源側のサービスはグリッド環境を構築するためのツールキットとして公開されているオープンソースソフトウェア Globus Toolkit 4（GT4）[4] を用いて実装され，GT4 の認可フレームワークを利用した VO 属性に基づく認可判断機構を組み込むことができるため，設定により計算資源側での細かい認可判断の実施も可能である．

情報サービス

■ 情報の収集

情報サービスはグリッド環境に含まれるさまざまな計算資源に関する情報収集を基本的な機能として持つ．グリッド環境では個々の計算資源により情報の表現方法が異なる可能性があり，収集した資源情報に対しても曖昧さのない共通理解が得られるような情報モデルが必要となる．NAREGI 分散情報サービス（以下，NAREGI IS）では抽象的かつ標準的な資源記述が可能となる Common Information Model（CIM）スキーマ [5] をベースとし，NAREGI ミドルウェア向けに拡張した情報モデル（NAREGI スキーマ）を採用している．

情報モデルに基づいて情報を収集する機能の 1 つとして，資源情報プロバイダがある．NAREGI IS でも資源情報プロバイダ経由で OS 情報，プロセッサ情報，バッチキュー情報，アカウント情報，VO 情報，インストール済みソフトウェア情報などを収集している（**図 -1** 参照）．サービス状態関連情報など，外部から能動的に情報更新することが望ましい場合もあり，そのため NAREGI IS で

は外部からの情報登録用インタフェースを設けている．また，ジョブ実行時に使用した資源利用記録の収集も情報サービスの機能の 1 つであり，そのインタフェースは Open Grid Forum（OGF）では Resource Usage Service（RUS）仕様として策定されている．NAREGI IS も OGF RUS 準拠のインタフェースを実装している．

■ 情報の集約

広域に分散した資源から構成されるグリッド環境では収集した情報を効率的に集約し，管理する機能が必要である．そのための 1 つの方式として階層構造を持たせて情報を集約することが考えられる．NAREGI IS でもこの方式を採用しており，収集した情報の 1 次集約点であるセルドメインアグリゲータサービス（CDAS）を階層的に連結し，上位階層のノードアグリゲータサービス（NAS）に対して情報を集約する．このような階層構造を持ったとき，スケーラビリティの観点から上位階層には下位階層より特定の情報を抽出し保持できることが望ましく，また，下位階層の情報が更新されたときのみ，情報が上位階層に送られることが望ましい．NAREGI IS も与えられた条件を満たす情報のみを選択的に上位階層へ集約する機能および下位階層から情報の変更があったことを上位階層に通知し，上位階層に集約情報を更新させる機能を有している．

■ 情報の検索

収集・集約した資源情報は他のグリッドミドルウェアからさまざまな目的で参照されるが，必要な情報を容易に得るための検索機能が必要となる．NAREGI IS では資源情報は NAREGI スキーマに対応したテーブルを持つ RDB に格納されるため，SQL で検索を実行することができる．グリッド環境から情報サービスに対して検索を実行する場合は，不正なアクセスを防ぐために認証・認可の機能が必要だが，NAREGI IS は遠隔地の DB に対し Web サービス経由でのアクセスを可能とするミドルウェア OGSA-DAI（Open Grid Service Architecture-Data Access and Integration）の機能を用いて実装されており，OGSA-DAI に対する GT4 の認可フレームワークを用いた認可判断機構を組み込むことにより，ユーザ単位のアクセス制御のほか，検索式に対する条件を設定するなどの制御が可能となる．

スケジューリング機能

■ 資源探索

グリッド環境でのスケジューリングは，資源探索，システム選択，ジョブ実行の 3 段階に分けることができる[6]．資源探索においてはアカウントの有無などの利用権限による計算資源絞り込み，絞り込みのためのアプリケーション要件の定義とそのアプリケーション要件による絞り込みが行われ，その結果としてジョブを実行させるシステムの候補を取得する．これは OGSA Execution Management Services（EMS）に含まれる Candidate Set Generator（CSG）に相当する機能を実行したことになる．OGSA EMS ではジョブを実行させるシステムは Service Container（SC）と呼ばれる．絞り込みのためのアプリケーション要件はユーザが指定できるジョブに関するさまざまな要件の中から選び出される．NAREGI スーパースケジューラ（以下，NAREGI SS）は OGSA EMS に基づくサービス群を実装しており，CSG の実行時に，アプリケーション要件による絞り込みを実施するため NAREGI IS に対し検索を実行する（**図 -2** 参照）．ユーザは OGF の Job Submission Description Language（JSDL）仕様に基づいてジョブ記述を行うが，NAREGI SS の CSG ではあらかじめ用意された変換ルールに従って JSDL から絞り込み検索用の SQL を生成している．NAREGI ミドルウェアでは JSDL バージョン 1 仕様，OGF で定められたパラメータスイープジョブ用の JSDL 拡張，さらに 1 つの MPI ジョブの中で異なるタイプの資源を要求する MPMD（Multiple Program, Multiple Data）型のジョブに対応するための NAREGI 独自の JSDL 拡張に対応している．

■ システム選択

システム選択では資源探索によって選ばれた実行システム候補の中から，実行時間，費用，信頼性等の目的を達成するために最適なシステムが選択される．この段階ではシステムの混み具合や待ちキューの長さなど動的な情報を収集し，その情報を元に選択することなどが想定されている．システム選択は OGSA EMS の Execution Planning Service（EPS）に相当する機能を実行したことになるが，EPS ではジョブ実行のスケジュールを生成するだけで，スケジュールの実行・管理は Job Manager（JM）が行う．NAREGI SS では予約ジョブに対するスケジュール生成に加え，非予約ジョブに対するシステム選択処理をカスタマイズ可能とするために，資源選定サービス・インタフェースを定義し，1 つの実装として Random Selection Service（RSS）を提供している．

■ ジョブ実行

スケジューリングの最後の段階がジョブ実行であるが，この段階では，事前予約や，選択したシステムに対してファイルステージングや予約の実施などの実行準備を行って，ジョブを計算資源に対してサブミットする．その後，ジョブの進捗状況の監視を行い，ジョブの終了を検

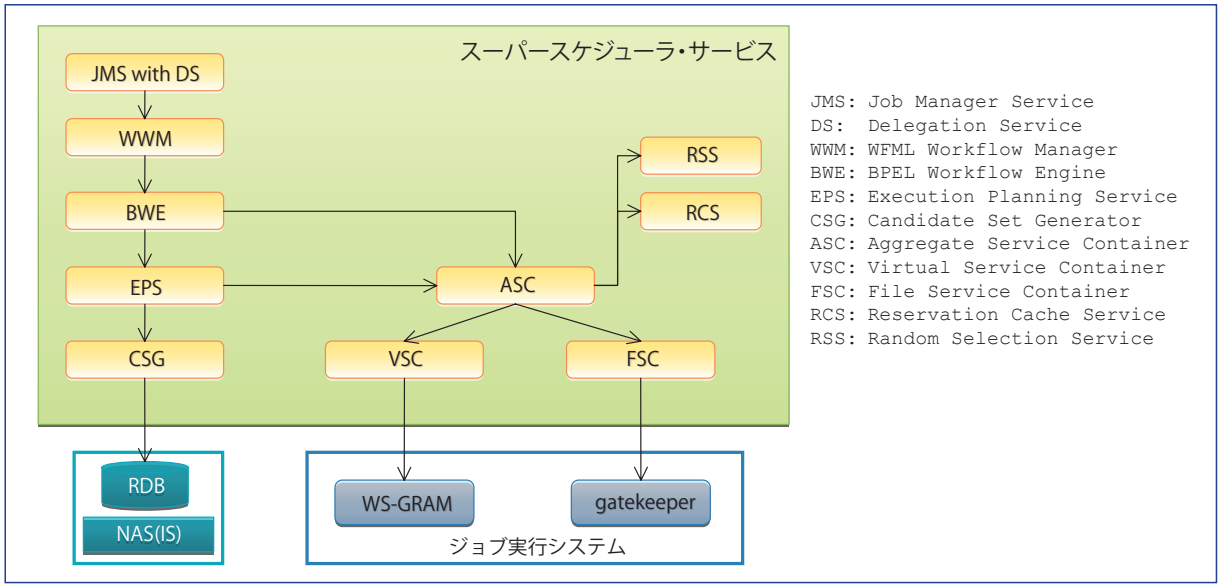


図 -2　NAREGI SS のサービス構成

知すると後処理を行う．ジョブ実行は OGSA EMS の JM の機能に相当する．NAREGI SS はこのジョブ実行段階に相当する処理においてユーザ権限でジョブ実行をジョブ実行サービスに依頼するための権限委譲の処理や長時間ジョブに対応するためのプロキシ証明書の更新，ファイルステージングなどのために SC 上へのワーキングディレクトリ作成などを行う．SC とのインタフェースとなる Virtual Service Container（VSC）で NAREGI SS がサポートする SC ごとの処理手順の違いを吸収している．また，NAREGI SS は単一サイトの資源の限界を超えるような規模のアプリケーションを実行するため，サイトをまたがった GridMPI の実行をサポートするための機能も有する．

ジョブ実行管理

■ プラットフォームの仮想化とジョブ実行管理機能

グリッド環境にはさまざまな機種の計算資源が混在しており，一般的にその利用方法は同じではないが，そのようなプラットフォームごとの差異は他のグリッドミドルウェアコンポーネントと連携する際に障害となる．したがってプラットフォームごとの利用方法を抽象化し，統一インタフェースを提供することはグリッド上でのジョブ実行環境として重要な機能である．さらに，グリッド環境からジョブ実行を受け付け，ジョブ実行状態を監視し，ジョブを制御できることが基本的な機能となる．

NREGI ミドルウェアでは NAREGI GridVM（以下，GridVM）がジョブ実行管理機能を提供する（**図 -3** 参照）．GridVM は OS とローカルスケジューラの複数の組合せをサポートするが，グリッド環境からはこれらの OS，ローカルスケジューラの違いは隠蔽されており，GridVM が提供する 2 つの Web サービスと，GT4 のジョブ管理機能 Web Services Grid Resource Allocation and Management（WS-GRAM）[4] を用いた統一インタフェースで資源予約，資源予約取り消し，予約情報取得，ジョブ投入，ジョブ強制終了，ジョブ状態取得，イベント通知の登録を行うことができる．ローカルスケジューラへのジョブ投入は JSDL に指定された内容に沿って，GridVM がローカルスケジューラ用のジョブ投入スクリプトを生成して実行する．

■ アクセス制御

各サイトの計算資源をグリッド環境でどのように利用させるかは，各サイトのポリシーによって決定できる必要があり，グリッド環境からの操作に対して，計算資源の管理者が設定可能な認証・認可の仕組みが機能しなければならない．

GridVM では GT4 が提供する grid-mapfile を利用した認可判断機構により，grid-mapfile にエントリされていないユーザの利用を拒否できるほか，GridVM 独自の機能によるファイルアクセス保護機能や資源利用量制御機能によりきめ細かい制御も可能であり，サイトのポリシーに応じて設定することができる．

■ コアロケーション

科学技術計算の中には，複数のサイトが有する資源を使って実行することが必要な大規模計算や，複数のアーキテクチャ上でそれぞれに適したジョブをお互いに連携しながら実行する連成計算が存在する．各サイトに分割されたジョブは通信しながら実行を行うが，その場合各サイトのサブジョブの実行は同期して行わないと余分な通信待ちが発生し，効率的な実行が行えない．そのた

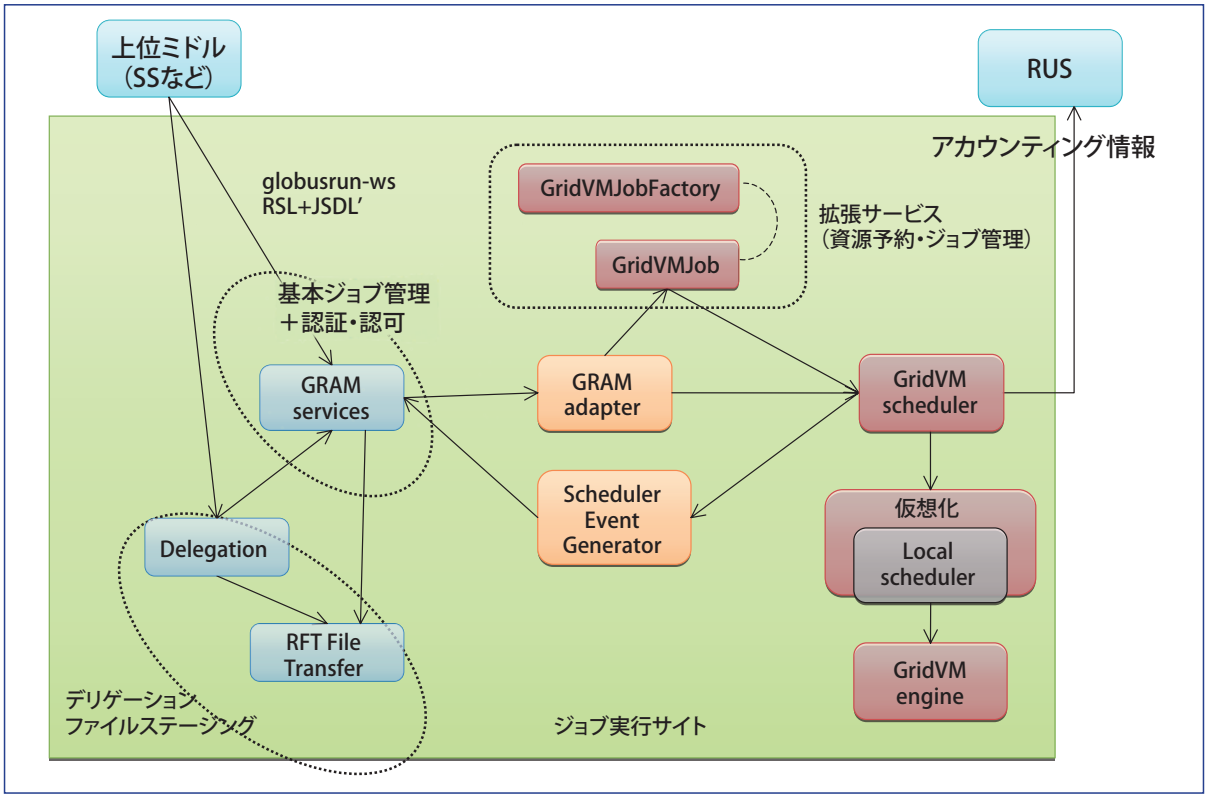


図-3　GridVM のサービス構成

め，各サイトの資源を同時に確保するコアロケーションが必要となる．GridVM は事前予約をサポートしており，NAREGI SS と連携して，各サイトの GridVM に対して同時刻に資源を使えるように事前予約を行うことでコアロケーションを実現する．

■アカウンティング情報のレポート

ユーザおよび管理者にとって，ジョブがグリッド環境内のどこでどのように実行されたかの情報を得られることは運用上重要な機能である．GridVM はジョブの終了時，そのジョブが消費した資源量を NAREGI IS に登録する．アカウンティング情報は OGF RUS 仕様に準拠したインタフェースを用いて登録され，ユーザ ID，VO 名，ジョブ ID などをキーにして検索・参照できる．

利用環境

■グリッドポータル

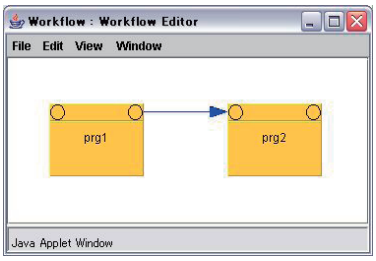
サイエンスグリッドのユーザは一般には計算機の専門家ではなく，研究の手段として計算機を利用しているため，グリッド特有の技術（証明書の取得・登録，プロキシ証明書の作成，ジョブ記述のための書式など）の詳細に関してはなるべく意識せずに，グリッド環境へのサインオンからアプリケーションの実行，実行状態の監視，アプリケーション登録・配備などのアプリケーション実行環境の整備などが行える利用環境が提供されることが望ましい．そのような目的のためには，Web インタフェースを用いたグリッドポータルが有用である．

NAREGI ミドルウェアでは利用者がグリッド環境を使用するためのインタフェースとして，Web ブラウザでアクセスする NAREGI Portal（以下，Portal）を提供する．Portal 上の各種ツール経由でグリッド環境へのサインオン，アプリケーション登録からジョブ実行までを行えるようになっている．

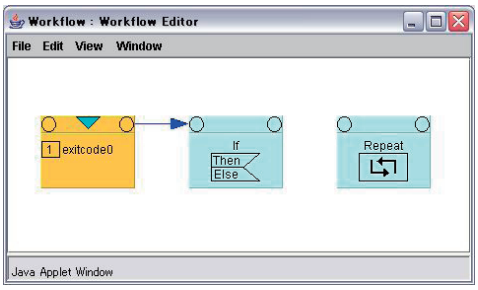
グリッド環境を利用するためにはユーザ証明書の取得と登録が最初に必要となるが，Portal ではユーザが認証局から発行されたユーザ証明書を取得し，適切な場所へ登録する処理までを行う画面を提供しており，ユーザに証明書の保管に関して意識させない．グリッド環境へサインオンする際にもユーザは自分の属する VO やその VO 内での Group，Role を指定するだけでよく，プロキシ証明書の作成，格納に関する詳細は隠蔽されている．

■アプリケーション実行支援機能

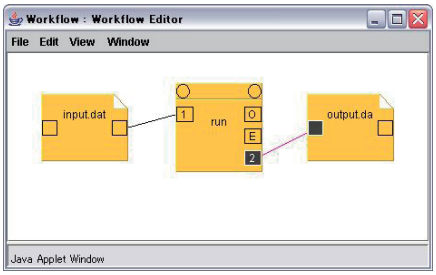
ユーザは個々のアプリケーションを単独で実行するだけではなく，前処理や後処理を含む複数のアプリケーションを連携させながら一連の計算を行わせることも多いが，グリッド環境では，個々のアプリケーションが実行される計算機は，それぞれのアプリケーションに対する資源要件に応じてスケジューリング機能により自動的に割り当てられるため，それらの一連の計算が必ずしも同じ場所で実行されるとは限らない．また，グリッド環境



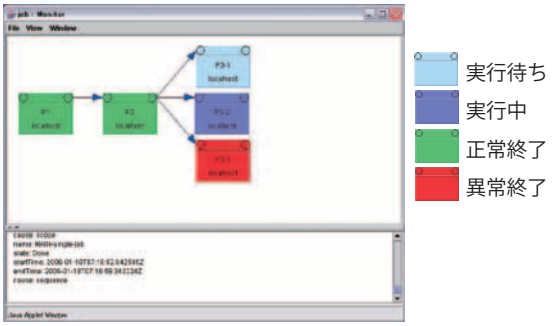
プログラムアイコン間の依存関係を定義した例．先行ジョブが終了するのを待って後続ジョブが実行される．



条件分岐アイコンとループ実行アイコンの例．先行ジョブのexit codeによる分岐と繰り返し実行が設定できる．



データアイコンとプログラムアイコンを用いて入出力関係を定義した例．データアイコンにはファイルのロケーションが設定されており，プログラム実行の前後にステージインおよびステージアウトが実行される．



モニタ表示の例．

図 -4　NAREGI WFT の概要

では，アプリケーションを単独で実行する場合であっても，通常入力ファイルはジョブ実行前に実行マシンへ転送され，ジョブ終了後，結果ファイルは実行マシンから外部へ転送される．複数のアプリケーションを連携させる場合には，スケジューリング機能が割り当てた計算機間でこれらの入出力ファイルの転送が発生する．グリッド環境において，このようなアプリケーション間の依存関係を定義し，その依存関係に従って実行制御を行うことや異なる場所で実行されるアプリケーション間での入出力ファイルの受け渡しを行うための機能としてワークフローを用いることができる．

NAREGI ミドルウェアではワークフローを用いてアプリケーションを記述し，実行制御および実行状態の監視を行うためのツールとして NAREGI ワークフローツール（以下，NAREGI WFT）を提供している（**図 -4** 参照）．NAREGI WFT ではプログラムやデータを GUI 画面上のアイコンとして表現し，それらの間の依存関係を定義することにより，NAREGI SS と連携して実行順序の制御を行わせることができる．NAREGI WFT では単一プロセスのプログラムを実行する際に用いるプログラムアイコンのほかに NAREGI ミドルウェアがサポートしている GridMPI ジョブ，連成ジョブ，バルクジョブに対応したアイコン，およびループや条件分岐を行うためのフロー制御アイコンなどが用意されており，それらのアイコンを組み合わせて，より複雑な処理を行うワークフローを作成することも可能である．作成したワークフローの実行および監視は NAREGI WFT の画面上から行うことができる．

WFT の画面上でアイコンのプロパティとして表形式で入力した内容から JSDL 形式のジョブ記述が自動的に生成されるため，JSDL に関する知識をユーザが持っていることを前提とせずに利用することができる．JSDL に関する知識を持っているユーザがより詳細な指定をする場合には直接 JSDL 形式で編集を行うことも可能である．

ユーザが作成したアプリケーションのワークフローに含まれる個々のジョブに対して JSDL が対応づけられており，その記述内容に従って適切なジョブ実行環境を個別に割り当てられる．そのため実行時にならないとファイルの転送先，転送元を具体化することができないが，NAREGI WFT と NAREGI SS とが連携してホスト名の決定と必要なファイルの転送を行うため，ユーザはそれぞれのジョブのプロパティとして設定した入力ファイル名，出力ファイル名だけを意識すればよい．ワークフローの実行が何らかの原因で失敗した場合など向けに，デバッグ実行用の画面が用意されており，ブレークポイントの設定やステップ実行によってワークフローのデバッグを行うことが可能となっている．

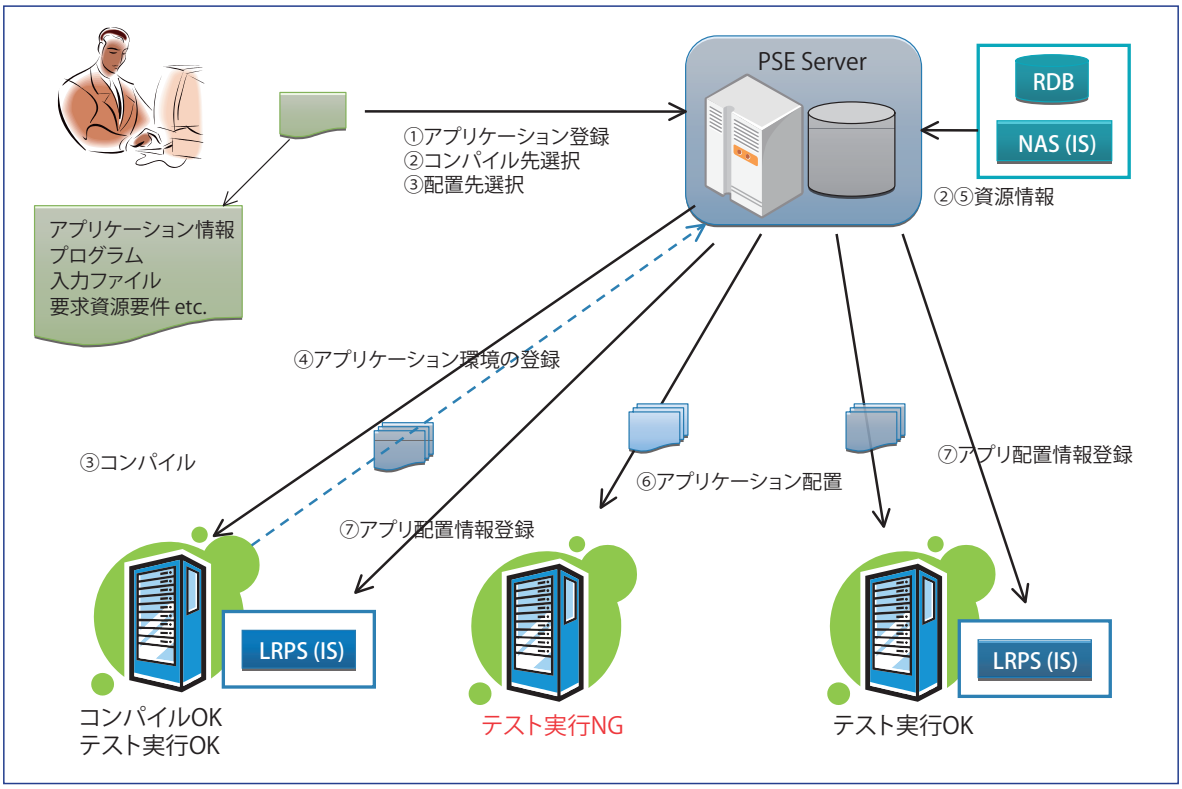


図 -5　NAREGI PSE の概要

■アプリケーション整備支援機能

サイエンスグリッドのユーザは既製のアプリケーションを用いるだけでなく，それぞれの研究のために自作のアプリケーションを開発し計算を実行することも多いが，そのような自作のアプリケーションをグリッド環境で実行する場合，ユーザが利用可能なすべての計算機でそれぞれの環境向けにコンパイル・インストールし，それぞれの計算機のインストール先ディレクトリ等の実行環境の管理を行うことは容易ではない．また，アプリケーションを改良して更新する際にも，インストールした計算機すべてについて人手でもれなく更新作業を行うことは相当な労力となる．このようにユーザ自身がアプリケーション開発を行ってグリッド環境で利用するような場合に，ユーザの作業を支援するツールの存在は不可欠である．NAREGI ミドルウェアではこのようなユーザ支援ツールの 1 つとして NAREGI 問題解決環境（以下，NAREGI PSE）を提供する（**図 -5** 参照）．NAREGI PSE ではユーザが作成したアプリケーションは Portal 上の画面を通してアプリケーションの登録を行い，そのアプリケーションに対して計算機アーキテクチャや OS，メモリ容量などのユーザが指定したリソース要件を満たす計算機向けにコンパイルし，その結果をリソース要件が満たされる各サイトに配備することができる．コンパイルおよび配備の際にはアプリケーションのテスト実行をさせることができるため，コンパイル，配備が正しく行われたことが確認できる．NAREGI PSE を通して配備したアプリケーションの実行パス等は NAREGI PSE によって管理されているので，前述の NAREGI WFT の画面上から目的のアプリケーションを登録時に指定したキーワード等で検索しインポートを実施することで，ユーザはアプリケーションの具体的な実行パス等を意識することなく，あらかじめ登録された状態で使うことができる．逆に NAREGI WFT で作成したワークフローをインポートして NAREGI PSE に登録することも可能となっている．また，登録済みのアプリケーションを更新する際も，NAREGI PSE によって配備先等が管理されているため確実にかつ容易に行うことができる．さらに，NAREGI PSE ではユーザが属する VO 内でアプリケーションを共有し，同じ VO に属する共同研究者がグリッド環境に配備したアプリケーションを利用することが可能である．

参考文献

1）Foster, I. et al. : A Security Architecture for Computational Grids., 5th ACM Conference on Computer and Communications Security (1998).
2）NAREGI ミドルウェア : http://middleware.naregi.org/Download/
3）Alfieri, R. el al. : VOMS, an Authorization System for Virtual Organizations, 1st European Across Grids Conference (2003).
4）Globus Toolkit : http://www.globus.org/toolkit/
5）Common Information Model（CIM）Standards : http://www.dmtf.org/standards/cim/
6）Schopf, J. M. : Ten Actions When SuperScheduling, http://www.ogf.org/documents/GFD.4.pdf (2001).

（平成 21 年 10 月 26 日受付）

鶴澤 武士
tsurusawa@nii.ac.jp
国立情報学研究所リサーチグリッド研究開発センター特任准教授．