
グリッド環境利用のためのアプリケーション プログラミングインタフェースの研究開発

(独) 日本原子力研究開発機構

■ 執筆者Profile ■



鈴木 喜雄

2002年 (特) 日本原子力研究所入所
可視化研究担当
2004年 文部科学省研究振興局情報課
2005年 (独) 日本原子力研究開発機構
グリッド研究担当

■ 論文要旨 ■

「最先端・高性能汎用スーパーコンピュータの開発利用」プロジェクトにおいて、ITBL(IT Based Laboratory)プロジェクトの資産であるグリッドアプリケーションを次世代スパコンを頂点とする次世代ナショナルグリッドインフラに継承可能とし、グリッドアプリケーションの創生を醸成するため、グリッド環境利用のためのアプリケーションプログラミングインタフェース(API)を研究開発している。ITBL プロジェクトの中で開発された種々のグリッドアプリケーションのグリッド環境の利用形態を分析し、API の機能を体系的に定義することにより、API のプロトタイプを開発した。その結果、従来よりも簡便にグリッド環境を利用可能なシステムやツールを開発すること、システム及びツールの変更なく異なるグリッド環境を利用すること、グリッドミドルウェアの変更なく異なるグリッド環境を接続することが可能となった。

■ 論文目次 ■

1. はじめに	《 3》
1. 1 ITBL プロジェクト	
1. 2 ITBL基盤ソフトウェア	
2. グリッド環境利用の問題点・現状と課題	《 5》
3. 目的	《 6》
4. 共通クライアントAPI	《 6》
4. 1 ITBL プロジェクトにおけるアプリケーションの分類	
4. 2 共通クライアントAPIの策定	
5. 評価	《 10》
6. むすび	《 12》

■ 図表一覧 ■

図1 ITBL環境を利用しているアプリケーションの分類。	《 7》
図2 共通クライアントAPIの階層化。	《 8》
図3 システム起動画面.....	《 10》
図4 デスクトップツールによる可視化画像表示。	《 11》
図5 試験的に構築したITBLとNAREGIの相互接続環境。	《 12》
表1 ファイルコピーにおける時間計測結果。	《 13》
表2 ジョブ実行における時間計測結果。	《 13》

1. はじめに

平成 18 年度から開始された国家プロジェクト「最先端・高性能汎用スーパーコンピュータの開発利用」において、次世代スーパーコンピュータを頂点とし国内の大学や研究機関のコンピュータの資源などを統合的に利活用可能とする次世代ナショナルグリッドインフラの実現を目指し、グリッドミドルウェアの利活用技術を研究開発している。ここで、ITBL (Information Technology Based Laboratory) プロジェクト[1][2]の資産であるグリッド環境で動作するアプリケーション(グリッドアプリケーション)を、次世代ナショナルグリッドインフラに継承可能とし、グリッドアプリケーションの創生を醸成するため、ユーザーがグリッドミドルウェアの種別を意識せずにユーザーフレンドリーにグリッド環境を利用できるためのアプリケーションプログラミングインターフェース(API)の研究開発を進めている。このため、平成 13 年度から平成 17 年度までの 5 カ年間で国家プロジェクトとして実施してきた ITBL プロジェクトで開発されたグリッドアプリケーションに対して、それらが利用者端末(クライアント)からどのようにグリッド環境を利用してきたかを分析することにより、利用者端末からユーザーフレンドリーにグリッド環境を利用可能とし、かつ、異なるグリッド環境に対して共通的に利用可能とする API(共通クライアント API)の機能を体系的に定義し、プロトタイプを開発した。ここで、異なるグリッド環境として、ITBL 環境と次世代ナショナルグリッドインフラの中核を担うことが期待されている NAREGI 環境[3]を対象とした。

本 API を ITBL プロジェクトで開発されたグリッドアプリケーションの一つである組立構造解析に適用し、利用者端末上で動作するシステムを構築することにより、本解析に必要なとなる複数のスーパーコンピュータで実施しなければならない一連の処理を利用者端末上で制御することを可能とした。また、本 API を用いてスーパーコンピュータ上に置かれた可視化画像を利用者端末上でサムネイル表示可能とするツールを開発することにより、グリッド環境にあるスーパーコンピュータ上の可視化画像を簡便に閲覧することを可能とした。これらのシステムやツールは、それらを改変することなく異なるグリッド環境(ITBL 環境及び NAREGI 環境)で利用可能である。

更に本 API により、異なるグリッド環境間の接続を容易に行うことを可能とした。具体的には、ITBL 環境と NAREGI 環境の基盤となるミドルウェアを変更することなく、ITBL 環境と NAREGI 環境の相互接続環境を構築することに成功し、異なるグリッド環境の相互接続に対しても本 API が有効であることを実証した。本章の残りでは、ITBL プロジェクト及び ITBL 環境の基盤を担っている ITBL 基盤ソフトウェアについて概要を記述する。更に第 2 章では、グリッド環境利用の問題点・現状と課題について記述し、第 3 章では、本研究の目的について記述を行い、第 4 章で共通クライアント API について、第 5 章でその評価を記述する。最後に第 6 章でむすびを記述する。

1. 1 ITBL プロジェクト

日本原子力研究開発機構は、物質・材料研究機構、防災科学技術研究所、理化学研究所、宇宙航空研究開発機構、科学技術振興機構とともに、平成 13 年度から平成 17 年度までの 5 カ年間、e-Japan 重点計画に位置付けられた ITBL (Information Technology Based Laboratory) プロジェクトを推進してきた。本プロジェクトでは、研究開発の IT

化を進めることにより、先端的な科学技術の各分野において技術革新を実現することを目的に、大容量ネットワーク上での研究機関のスーパーコンピュータなどの共用化、複雑で高度なシミュレーションや遠隔地との共同研究を可能とする仮想研究環境の構築とその普及促進を進めてきた。日本原子力研究開発機構システム計算科学センターは、ITBL プロジェクトにおいて基盤を担うソフトウェア (ITBL 基盤ソフトウェア) の研究開発を実施した。本ソフトウェアにより、12 の研究機関が所有する 27 台の計算機 (20 機種、約 57TFlops) を含むグリッド環境を実現し、構築された ITBL 環境の利用者はプロジェクト終了時点で 90 機関から 681 名に至った。

1. 2 ITBL 基盤ソフトウェア

ITBL 基盤ソフトウェアは、以下に示すとおり、認証及び通信基盤、基本サービス、アプリケーションサービスの 3 層構造を持つ。

(1) 認証及び通信基盤

ITBL 環境の基盤として、電子化された利用者証明書を用いて利用者の認証を行う認証基盤と、(1-2)に記載している高安全な通信を可能とする通信基盤を整備した。

(1-1) 認証基盤

ITBL 環境へのアクセスの際に、電子化された利用者証明書を用いて利用者認証を行う。ここで、利用者証明書の ID と計算機の ID との対応付けにより、ITBL 環境のすべてのコンピュータへのシングルサインオン (ユーザが一度認証を受けるだけで、許可されているすべての機能を利用できるようになるシステム) を実現している。(ただし、利用者は、各機関のポリシーに基づき、それぞれの計算機の ID を事前に取得しておく必要がある。)

(1-2) 通信基盤

ITBL では、インターネットを介して各大学や研究機関 (サイトと呼ぶ) の計算機を接続することを前提としている。そこで、できるだけ安全な通信を実現するため、ファイアウォールの設定として、Web で利用している HTTPS のポートしか利用する必要なく、Web の通信プロトコル (HTTPS プロトコル) により暗号化通信が可能となるよう通信基盤を構築した。本通信基盤を STARPC 通信基盤 [4][5] と呼ぶ。このような通信を実現する手段として、利用者端末とサイトを構成するバックエンド計算機群の仲介的役割を果たすサーバ (ITBL サーバ) を導入することとした。また、ITBL サーバは、ITBL 運用に関わる必要な諸情報を可能な限り吸収し、バックエンド計算機への負荷・影響をできるだけ最小化している。ITBL 環境に不正侵入された場合の「踏み台」や「データ漏洩」に対するリスクを軽減するため、ITBL サーバの機能を物理的に異なる 3 つの計算機「フロントサーバ」「中継サーバ」「データサーバ」に分散している。フロントサーバは、利用者端末と ITBL 環境との仲介、利用者認証やアクセス制御、データサーバ上のデータへのアクセスを担っている。中継サーバは、フロントサーバとサイト内の LAN に接続されているバックエンド計算機間の通信を仲介する。データサーバは、フロントサーバが行う利用者認証のためのデータなど、ITBL 環境の運用に必要な管理データを格納している。

(2) 基本サービス

下記(3)のアプリケーション開発サービスを実現する上で基本となる機能として、プロセス管理機能、システム情報モニタリング機能、タスクスケジュール機能、データ管理機能を整備した。

(2-1) プロセス管理機能

サービスプログラムの起動、終了や、プログラムの実行状態の確認、ジョブ投入の実制御を行う機能。

(2-2) システム情報モニタリング機能

ネットワーク資源情報の取得、計算機のシステム情報や利用状況についての情報取得などを行う機能。

(2-3) タスクスケジューリング機能

システム情報モニタリング機能で取得した情報をもとに、利用者の計算機選択ポリシーに基づいた計算機の決定や、ジョブの実制御を行う機能。

(2-4) データ管理機能

ファイル情報の管理、ファイルのコピーや削除、名称変更などの実作業を行う機能。

(3) アプリケーション開発サービス

上記(1)と(2)をベースに、利用者がアプリケーション開発時に実際に利用できる機能として、分散プログラミング機能、可視化機能、コミュニティ機能を整備した。

(3-1) 分散プログラミング機能

ファイルのコピー・削除・名称変更、ディレクトリの作成・削除・名称変更に対して、一つの計算機内でも計算機間でも共通な操作が行える機能(ファイル操作機能)、プログラム実行におけるプログラム間のデータ依存関係や実行制御関係を視覚的に操作できるワークフロー機能(TME: Task Mapping Editor)[6][7]、資源情報(サイトごとの計算機の利用者数、サイト内のネットワーク性能や実行計算機の情報、実行計算機のバッチキュー情報やスケジュール情報など)を取得・表示する機能(RIS: Resource Information Service)などを有する。これらの機能は Web ブラウザ上で利用可能である。更に、異機種計算機間でメッセージパッシング通信を実現するライブラリ(STAMPI: Seamless Thinking Aid Message Passing Interface)[8][9]や、ファイルの編集、コンパイル、プログラム実行、Xアプリケーション実行などを支援する機能も有する。

(3-2) 可視化機能

商用可視化ソフトウェア AVS/Express[10]と連携して、Web 上で可視化結果を表示・操作できる機能、ITBL 環境に存在する任意のコンピュータ上のファイルにアクセス可能なりモートデータ読み込み機能、及び、可視化処理を並列に実行可能な機能を有する。本可視化機能を AVS/ITBL[11]と呼んでいる。

(3-3) コミュニティ機能

コミュニティのメンバー間でデータを共有できるキャビネット、コミュニティのメンバー間の連絡・意見交換を行うことが可能な会議室、映像・音声・ドキュメントを利用して多地点とのオンラインミーティングを実現するビデオミーティングなど、研究コミュニティを形成するための機能を有する。

2. グリッド環境利用の問題点・現状と課題

上記 ITBL 基盤ソフトウェアの節にて記述されているような機能を利用する形態として、利用者端末を用いてポータルサーバなどにアクセスし、Web ブラウザなど決められたインタフェースを用いて PC クラスタやスーパーコンピュータなどのバックエンド計算機を利用

するという方法が用いられてきた。例えば、NAREGI 環境や初期の ITBL 環境などでは、利用者端末において Web ブラウザを起動し、グリッドミドルウェアが用意しているツールを利用するという形態をとっている。そのため、ユーザーは、グリッドミドルウェアが提供するインタフェースやコマンドを利用するという方法でしかグリッド環境にある計算資源を利用することができず、必ずしもユーザーフレンドリーにグリッド環境を利用することができない。更に、利用者端末の資源利用は想定されておらず、バックエンド計算機の計算資源を水平分散的に利用することしかできない。ユーザーフレンドリーにグリッド環境を利用可能とする試みとして、利用者端末においてユーザーが自由にグリッド環境を利用できるクライアント API の研究開発を進めてきた。クライアント API は、ユーザーの利用者端末上に構築したアプリケーションからグリッド環境の計算資源を利用することを可能とする。また、実行計算機を水平分散的に利用する従来の方法に加えて、利用者端末の計算資源との垂直分散的な利用も可能とすることで、利用者端末とバックエンド計算機群の計算資源を併せて垂直水平分散的に利用することが可能である。したがって、クライアント API は、利用者端末のリソースとバックエンド計算機群のリソース連携、負荷の大きさに依存した利用者端末とバックエンド計算機群への処理分散、利用者端末上に構築したアプリケーションへの様々な機能の実装など、自由度の高いグリッドアプリケーションの開発を可能とするポテンシャルを有している。クライアント API の他の研究例として、Commodity Grid Kits (CoG Kits) [12] や Simple API for Grid Applications (SAGA) [13] などがある。CoG Kits は、米国で開発されてきたグリッドミドルウェアである Globus [14] を隠蔽し、より使用しやすいインタフェースをユーザーに提供するために開発されたツールキットであり、米国を中心に様々なアプリケーションで利用されている。また、SAGA は、アプリケーションを簡便にグリッド化するため、機能及びその API の集合体の標準を規定することを目的に、Open Grid Forum (OGF) にて検討が進められておりサンプル実装が提供されている。

しかしながら、SAGA を除く種々のグリッドミドルウェアに実装されてきたクライアント API は、アプリケーションがどのようにグリッド環境を利用しているかという観点からトップダウン的に検討されてきたというよりは、グリッドミドルウェアが既に有している機能をベースにボトムアップ的に検討されてきた。SAGA は、トップダウン的に検討が進められており、今後期待される API ではあるが、現時点ではサンプル実装が提供されるに留まっている。したがって、グリッド環境を十分ユーザーフレンドリーに利用可能とする API は必ずしも実現していない。

また、これまで世界中で様々なグリッドミドルウェアが開発されてきており、グリッド環境が構築されてきているが、グリッド環境ごとにその利用形態が異なるため、ユーザーは異なるグリッド環境を利用しようとする場合、グリッドミドルウェアもしくは各グリッド環境が提供するクライアント API ごとにグリッドアプリケーションをカスタマイズしなければならない。

更に、このような世界中の異なるグリッド環境をシームレスに利用可能とするアプローチとして、異なるグリッド環境を相互接続する試みがなされてきているが、グリッド環境ごとにグリッドミドルウェアの仕様が異なるため、接続相手ごとに異なる手段を用いて開発を行わなければならない。

3. 目的

ITBL プロジェクトが終了した翌年度、すなわち、平成 18 年度から国家プロジェクトとして「最先端・高性能汎用スーパーコンピュータの開発利用」プロジェクトが開始された。本プロジェクトは、平成 24 年度までの 7 カ年で実施される計画となっており、世界最先端・最高性能の次世代スーパーコンピュータの開発・整備及び利用技術の開発・普及を目標に推進されている。当センターは、第 1 章で記述したとおり、ITBL 環境のアプリケーション資産を、次世代スパコンを頂点とする次世代ナショナルグリッドインフラに継承可能とし、グリッドアプリケーションの創生を醸成するため、ユーザーがグリッドミドルウェアの種別を意識せずにユーザーフレンドリーにグリッド環境を利用できることを目的として、API の研究開発を実施している。ここで、ITBL プロジェクトでの知見を元に、第 2 章で記述した課題を解決する観点から、よりユーザーフレンドリーにグリッド環境を利用できるアプリケーションを構築可能な API として、共通クライアント API の実現を目指している。

4. 共通クライアント API

ITBL プロジェクトにおいて得られたアプリケーションがどのようにグリッドを利用するかという知見から、共通クライアント API の研究開発を実施し、これまでに、ITBL 環境及び NAREGI 環境で共通的に利用可能なプロトタイプを構築した。

4. 1 ITBL プロジェクトにおけるアプリケーションの分類

ここでまず、API として必要となる機能分析を行うため、ITBL プロジェクトにおけるアプリケーションがどのように ITBL 環境を利用しているかを分析し、図 1 のとおり分類した。

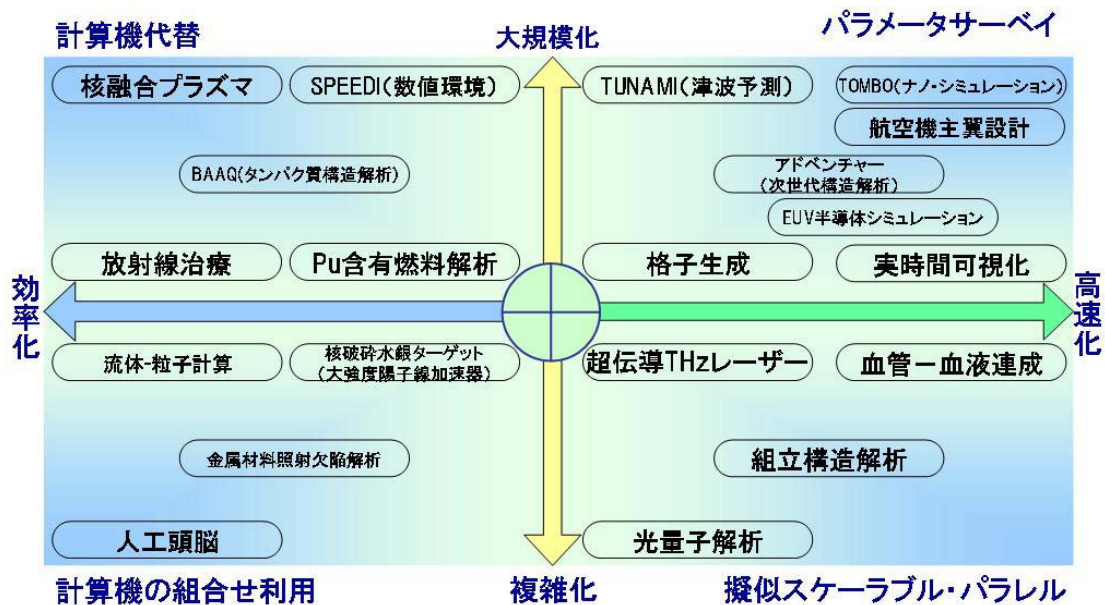


図 1 : ITBL 環境を利用しているアプリケーションの分類。

ここで示されている通り、アプリケーションによる ITBL 環境の利用は、「パラメータサーベイ」、「計算機代替」、「計算機の組み合わせ利用」、「擬似スケーラブル・パラレル」の4つに大別できることを見出した。

(1)パラメータサーベイ

結果のパラメータ依存性を調べるために、異なる複数のパラメータを入力とした複数のシミュレーションを実施する。

(2)計算機代替

計算機に異常が生じた場合などに、代替となる計算機を利用する。

(3)計算機の組み合わせ利用

連成シミュレーションに代表されるような、異なる複数の分野のシミュレーションを融合して行う場合に、スカラ計算機やベクトル計算機など、それぞれの分野に対して最適な計算機を組み合わせ利用する。

(4)擬似スケーラブル・パラレル

一台の計算機ではメモリなどが不足する場合に、複数の計算機を一台の大きな計算機に見立てて利用する。

4. 2 共通クライアント API の策定

このように、アプリケーションが ITBL 環境をどのように利用するかという観点から、グリッド環境において必要となる共通クライアント API を図2のとおり策定した。ここで、グリッド基盤として記載されているレイヤーは、これまで世界中で開発されてきたグリッドミドルウェアとそれが有する API を意味している。グリッドミドルウェアは、既存の基盤と Globus をベースにした基盤に大別される。これらいくつかのグリッドミドルウェアに対しては、なんらかの形で API の開発が進められてきているが、我々が策定した API は、これらグリッドミドルウェアの上位に位置づけられるものであり、様々なグリッドミドルウェアの API に対する共通化を図るものである。

level	layer	機能								
high	分野別機能	構造	振動	流体	熱	電磁場	材料			
	処理別機能	計算機組合せ	計算機代替	擬似スケーラブル・パラレル	パラメータサーベイ					
middle	手法別機能	FEM	FVM	FDM	Particle	MD	Monte Carlo			
	統合機能	異機種活用		メタ・コンピューティング	スループット	分散処理				
	個別機能	ワークフロー	可視化	コンパイラ	ターミナル	グリッド関係				
low	基本機能	ジョブ実行			ファイル操作		資源情報管理			
グリッド基盤		認証			通信					
		GRAM	DMS	GASS	GridFTP	...	ジョブ実行	ファイル操作	資源情報	...
		Globus ベース				既存システム				
	NAREGI	EGEE	TeraGrid	...	ITBL	UNICORE	AVAKI	DIET	...	

図2：共通クライアント API の階層化。

このような共通クライアント API として、高レベル、中レベル、低レベルの3つのレベルに API を階層化している。このような階層化により、上位の API 構築に際して、下位の API を用いて容易に構築可能となる。

(1) 高レベル API

高レベル API は、処理別機能、分野別機能により構成することとした。処理別機能は、ITBL 環境を利用しているアプリケーションの分類に基づき策定した。更に、分野別機能は、ITBL 環境を利用しているアプリケーションを分野という観点から見たときの分類となっている。これら高レベル API を用いることで各アプリケーションを容易に構築できるよう実装を計画している。

(2) 中レベル API

中レベル API は、個別機能、統合機能、手法別機能により構成することとした。個別機能には、ワークフロー、可視化、コンパイラ、ターミナル、グリッド連係、ジョブ実行、ファイル操作、資源情報管理が含まれる。ここで、グリッド連係 API は、異なる複数のグリッドミドルウェアが混在する環境においても利用者が利便性を損なわないための相互利用を実現する API である。このような API により、世界中に混在する様々なグリッドミドルウェアの利活用を、アプリケーションが個別に対応しなくともよい環境を提供できると考えている。ワークフロー、ファイル操作、資源情報管理は、ITBL クライアント API を拡張したものである。統合機能としては、高レベル API において処理別機能として策定している API を効率的に実現する機能として、機種活用、メタ・コンピューティング、スループット、分散処理を位置づけている。更に、手法別機能としては、高レベル API において分野別機能として策定している API を実現する機能として、有限要素法、有限体積法、有限差分法、粒子法、分子動力学法、モンテカルロ法などを位置づけている。

(3) 低レベル API

低レベル API には、認証、通信の API を位置づけている。認証及び通信は、ITBL のクライアント API を拡張したものである。低レベル API のうち、図2において緑枠で囲った API について ITBL 基盤ソフトウェア及び NAREGI ミドルウェアに対してプロトタイプ(認証 API(5)、通信 API(123)、ジョブ実行 API(6)、ファイル操作 API(14)、資源情報管理 API(2) (括弧内は API の数))を構築し、評価を実施した。実現している API の機能概要を以下に示す。

(3-1) 認証 API

利用者端末上に構築したアプリケーションから、グリッド環境の計算機群のリソースを利用する際の認証を可能とする。

(3-2) 通信 API

すべての機能の基盤となる。利用者端末上のアプリケーションからグリッド環境への通信路を確立し、通信を行う。利用者端末とバックエンド計算機及びバックエンド計算機間において、ストリーム通信と RPC 通信を行うことができる。

(3-3) ジョブ実行 API

プログラム実行におけるプログラム間のデータ依存関係や実行制御関係を視覚的に操作できるワークフロー機能の実行を可能とする。ITBL 環境では TME スクリプトを、NAREGI 環境では WFML を実行する。実行のキャンセルや実行状態の取得を行う API も有する。

(3-4) ファイル操作 API

利用者端末及びグリッド環境の計算機上のファイル管理、ディレクトリ表示、ファイルコピー、リモートファイル I/Oなどを可能とする API を有する。

(3-5) 資源情報管理 API

資源情報を取得する機能を実行可能とする。計算機のキュー情報(キューの名称、稼動状態、ジョブ数など)を取得する。計算機情報の一覧、キュー情報の一覧を取得可能とする API を有する。

5. 評価

現在、ITBL プロジェクトにて当センターが実施したアプリケーションの一つである組立構造解析(図1参照)を制御しているシステムに対して、共通クライアント API を適用した。本システムでは、共通クライアント API のプロトタイプに関して、認証、通信、ジョブ実行、ファイル操作、資源情報管理の主要な API が用いられている。組立構造解析は、複数の部品の組立物として構成される原子力施設の振動に対する安全性をより厳密に示すことを目指して実施されており[15]、このよう組立構造解析を利用者端末上からコントロールすることを目的にシステムを構築した。共通クライアント API の利用により、メッシュデータの読み込み、複数の部品に対する構造解析の各計算機での実行、計算結果の利用者端末への転送、可視化処理、結果画像の利用者端末上での表示、といった一連の処理の制御を利用者端末上から可能とするシステムを実現した。本システムは、その修正の必要なく ITBL 環境及び NAREGI 環境で動作可能である。図3にシステムの起動画面を示す。

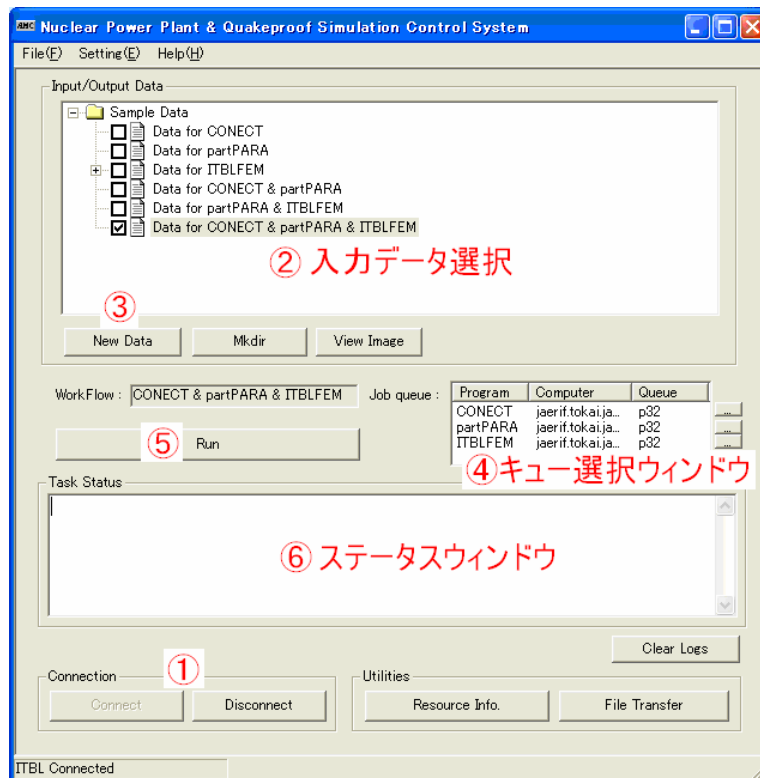


図3：システム起動画面。

画面は、①ITBL 接続ボタン、②入力データ選択ウィンドウ、③入力データの新規登録ボタン、④ジョブキュー選択ウィンドウ、⑤実行ボタン、⑥ステータスウィンドウからなる。まず①ITBL への接続を行い、②入力データを選択する。新たにデータを登録する際は③New data ボタンを押し、入力ファイルのあるディレクトリの指定を行う。入力データが決定されると、計算に使用されるプログラムが決定するので、そのプログラムを実行するキューの選択を④ジョブキュー選択ウィンドウにおいて行う。以上で計算条件の設定が終わり、⑤実行ボタンを押し。その後のジョブの状況は⑥ステータスウィンドウに表示される。実行されたプログラムが終了すると、計算結果を可視化するウィンドウが新たに表示される。

また、共通クライアント API の利用により、バックエンド計算機上の可視化画像を利用者端末上でサムネイル表示可能とするデスクトップツールの開発を行った。本ツールは、windows のエクスプローラライクなインタフェースを有しており、バックエンド計算機群と利用者端末上のファイルを統合的に扱うことが可能である。すなわち、これら計算機間でのファイルコピーや移動、各計算機上のディレクトリ作成や一覧表示、ファイルの削除などを行うことなども可能である。図4にデスクトップツールのイメージを示す。本ツールについても、共通クライアント API の適用により、利用者端末からユーザーフレンドリーにグリッド環境のファイル操作や可視化を可能とした。本ツールも、修正の必要なく ITBL 環境及び NAREGI 環境で動作する。

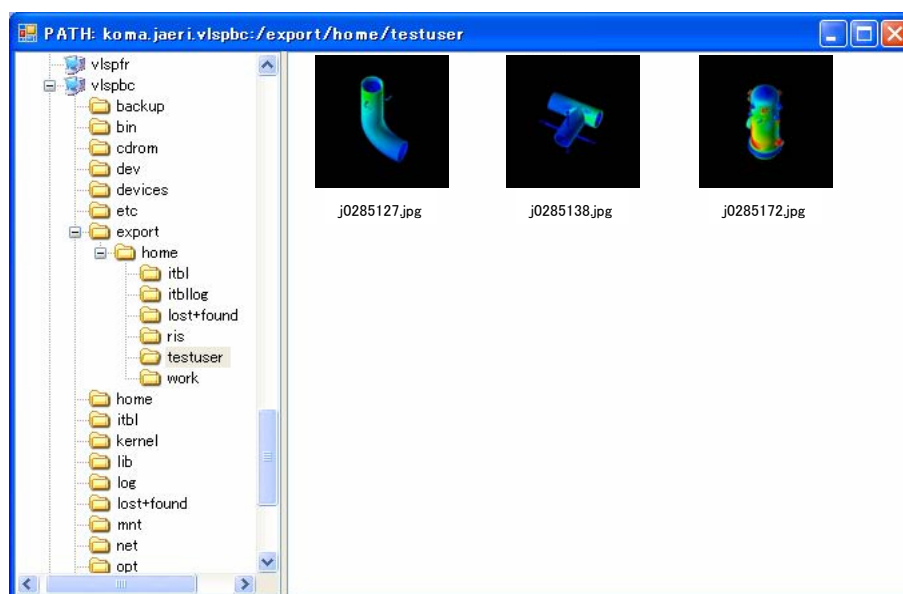


図4：デスクトップツールによる可視化画像表示。

更に、共通クライアント API を用いて、ITBL 環境と NAREGI 環境の相互接続システムを設計し、本設計を検証するための検証プログラムを試作した。共通クライアント API を用いることで、アプリケーション開発と同様のメリットを得ることができる。すなわち、他方のグリッドミドルウェアが変更された場合にも、共通クライアント API をコールしている側のグリッドミドルウェアを変更する必要がない。更に、共通クライアント API を実装しているグリッドミドルウェアに対しては、同じシステムを用いて接続を行うことが可能

である。現在、本プログラムを、ITBL 環境としては実運用している環境を、NAREGI 環境としては国立情報学研究所に構築いただいた試験環境に適用することにより、ITBL-NAREGI 相互接続環境の構築に成功した。図 5 に構築した環境の概要を示す。NAREGI 環境としては、国立情報学研究所において、テスト環境としてバックエンド計算機に Linux 計算機を用意していただいた。本環境を用いることで、ITBL 環境から NAREGI 環境に対して、ファイル操作、ジョブ実行、資源情報管理が行えること(図 5 ①)、NAREGI 環境から ITBL 環境に対して、ジョブ実行が行えること(図 5 ②)を確認した。ここで、両環境間の通信は、HTTPS 通信であり、インターネット上を安全に通信できる。

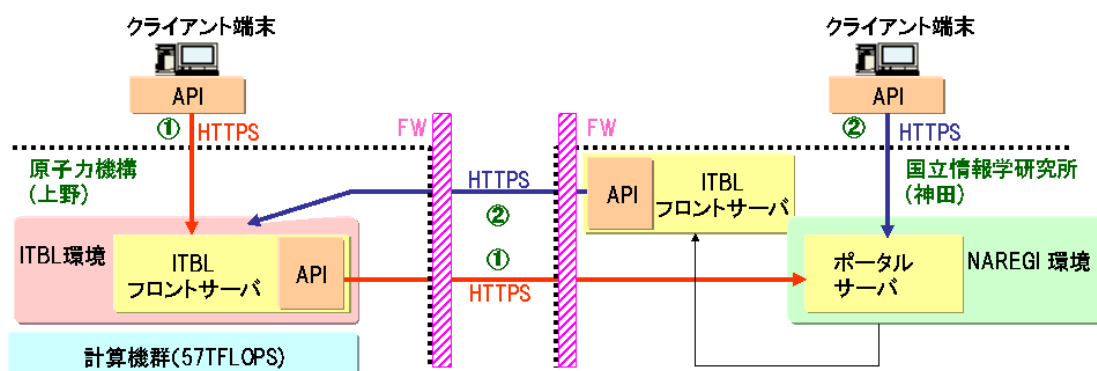


図 5 : 試験的に構築した ITBL と NAREGI の相互接続環境。

共通クライアント API 及び ITBL-NAREGI 相互接続環境に対して定量的な評価を実施するため、本環境を用いてファイル操作及びジョブ実行の性能評価を実施した。ファイル操作については、ITBL フロントサーバに接続している利用者端末と、ITBL 環境の p690 (IBM 社製) 及び NAREGI 環境の Linux 計算機間でそれぞれファイルコピーを行い要した時間を計測した。また、ジョブ実行については、ITBL フロントサーバに接続している利用者端末から、各環境の計算機へジョブを投入し終了するまでの時間を計測した。実行したジョブはホスト名を取得する単純なもの (/bin/hostname) である。表 1、表 2 にそれぞれ結果を示す。

表 1 から、NAREGI 環境にある Linux 計算機から、ITBL フロントサーバに接続している利用者端末へファイルをコピーするのに要する時間と、ITBL 環境にある p690 からファイルをコピーするのに要する時間にそれほど大きな差はなく、また、ITBL フロントサーバに接続している利用者端末から NAREGI 環境にある Linux 計算機へファイルをコピーするのに要する時間と、ITBL 環境にある p690 へファイルをコピーするのに要する時間にもそれほど大きな差はないことが分かる。したがって、相互接続環境に対しても従来の ITBL 環境と遜色なくファイルコピー可能であることが確認できた。また、表 2 では、ITBL フロントサーバに接続している利用者端末から、NAREGI 環境にある Linux 計算機へのジョブ投入は、ITBL 環境にある p690 へのジョブ投入よりも非常に時間を要していることが分かる。この原因としては、NAREGI 環境においてジョブ投入に時間を要すること、かつ、相互接続環境構築のために作成した試作プログラムにて時間を要していることの二つの原因がある。同様のジョブを NAREGI 環境のみで実行した場合には、約 33.1 秒要していることから、試作プログラムにおいて、残り 14.5 秒 (50.2-33.1-2.6) を要していることになる。本プログラムにおけるボトルネックについての詳細な検討は今後の課題である。しかしながら、一般

的なジョブはここで用いたような単純なジョブではなく、数分から数時間のオーダーであるため、ジョブ投入に数十秒の時間を要したとしても重大な問題にはならないと考えられる。

表 1 : ファイルコピーにおける時間計測結果。

ファイルサイズ	1MByte	10MByte	100MByte
Linux 計算機(NAREGI)→利用者端末	13.5(s)	46.6(s)	364.6(s)
利用者端末→Linux 計算機(NAREGI)	8.5(s)	36.7(s)	318.5(s)
p690(ITBL)→利用者端末	4.8(s)	34.2(s)	331.7(s)
利用者端末→p690(ITBL)	4.4(s)	31.4(s)	300.1(s)

表 2 : ジョブ実行における時間計測結果。

利用者端末→Linux 計算機(NAREGI)	50.2(s)
利用者端末→p690(ITBL)	2.6(s)

6. むすび

「最先端・高性能汎用スーパーコンピュータの開発利用」プロジェクトにおいて、共通クライアント API のプロトタイプを開発することにより、従来よりも簡便にグリッド環境を利用可能なシステムやツールを開発することが可能となった。また、システム及びツールの変更なく異なるグリッド環境を利用することが可能となった。更に、異なるグリッド間の相互接続に対しても共通クライアント API は、有用な役割を果たすことを示した。

評価に用いた相互接続環境のプログラムは、設計検証のために試作したプログラムであり、今後、機能拡充及び性能向上により、運用に耐えるシステムを構築していく計画である。また、現在実現している共通クライアント API はプロトタイプであり、機能拡充及び性能向上、更にアプリケーションによる実証を重ねることにより、運用版を開発していく計画である。現在、よりアプリケーション構築に適した高レベルのクライアント API の研究開発を進めているところであり、本開発により、グリッド環境をよりユーザーフレンドリーに利用可能とする利活用技術の構築を行っていく予定である。

謝辞

本研究開発を進めるにあたり、NAREGI 環境の利用などについて支援いただきました国立情報学研究所の皆様へ感謝いたします。本研究開発の一部は、文部科学省のプロジェクト「最先端・高性能汎用スーパーコンピュータの開発利用」のもとで実施されました。

参考文献

- [1] 山口勇吉, 武宮博: 仮想研究環境ITBL基盤ソフトウェアの開発, 日本数値流体力学会誌, Vol.9, No.3 (2001) pp.83-88.

- [2] Higuchi, K., Imamura, T., Suzuki, Y., Shimizu, F., Machida, M., Otani, T., Hasegawa, Y., Yamagishi, N., Kimura, K., Aoyagi, T., Nakajima, N., Fukuda, M. and Yagawa, G. "Grid Computing Supporting System on ITBL Project", Veidenbaum et al. (Eds.) 5th International Symposium on High Performance Computing (20 ~ 22 October 2003, Tokyo, Japan), LNCS2858 (2003) pp.245-257.
- [3] Matsuoka, S., Shinjo, S., Aoyagi, M., Sekiguchi, S., Usami, H. and Miura, K. "Japanese computational grid research project: NAREGI", Proceedings of the IEEE, Volume 93, Issue 3, March 2005 pp.522-533.
- [4] Takemiya, H. and Yamagishi, N. "Starpc: a library for Communication among Tools on a Parallel Computer Cluster - User's and Developer's Guide to Starpc -", JAERI-Data/Code 2000-005 (2000).
- [5] 武宮博, 山岸信寛: 並列計算機クラスタ上のツール間通信を支援するライブラリ:Starpc -Starpc利用及び開発手引書, JAERI-Data/Code 2000-006 (2000).
- [6] Imamura, T., Hasegawa, Y., Yamagishi, N. and Takemiya, H. "TME: A Distributed resource handling tool", Recent Advances in Computational Science & Engineering, International Conference on Scientific & Engineering Computation (IC-SEC) (3-5 December 2002, Raffles City Convention Centre, Singapore) (2002) pp.789-792.
- [7] Suzuki, Y., Matsumoto, N., Yamagishi, N., Higuchi, K., Otani, T., Nagai, H., Terada, H., Furuno, A., Chino M. and Kobayashi, T. "Development of Multiple Job Execution and Visualization System on ITBL System Infrastructure Software and Its Utilization for Parametric Studies in Environmental Modeling" Computational Science - ICCS 2003 Sloot et al. (Eds.) International Conference (2-4 June 2003, Melbourne, Australia and St. Petersburg, Russia, Proceedings, Part III), LNCS2659 (2003) pp.120-129
- [8] 今村俊幸, 小出洋, 武宮博: 異機種並列計算機間通信ライブラリ:Stampi利用手引書 第二版, JAERI-Data/Code 2000-002 (2000).
- [9] Imamura, T., Tsujita, Y., Koide, H. and Takemiya, H. "An Architecture of Stampi: MPI Library on a Cluster of Parallel Computers", In Recent Advances in Parallel Virtual Machine and Message Passing Interface, LNCS 1908, Springer (2000) pp.200-207.
- [1 0] Upson, C. et al. "The application visualization system: A computational environment for scientific visualization", IEEE Computer Graphics and Applications, Vol.9, no.4, (1989) pp.30-42.
- [1 1] Suzuki, Y., Sai, K., Matsumoto, N. and Hazama, O. "Visualization system on Information Technology Based Laboratory", IEEE Computer Graphics and Applications, Vol.23, No.2, March/April (2003) pp.32-39.
- [1 2] Laszewski, G., Gawor, J., Lane, P., Rehn, N., Russell, M., and Jackson, K., "Features of the Java Commodity Grid Kit", Concurrency and Computation: Practice and Experience, 14 (2002) pp1045-1055.
- [1 3] GGF. (2004) Simple API for Grid Applications Research Group. [Online].

Available: <http://forge.gridforum.org/projects/saga-rg/>

- [14] Foster, I. and Kesselman, C. "Globus: A metacomputing infrastructure toolkit", International Journal of Supercomputing Applications, Vol.11, No.2 (1997) pp.115-128.
- [15] 西田明美、松原仁、田栄、羽間収、鈴木喜雄、新谷文将、中島憲宏、谷正之、近藤誠: 原子カプラントのための3次元仮想振動台の構築 -組立構造解析法による巨大施設解析システムの提案-, 日本原子力学会和文論文誌, Vol.6, No.3 (2007) (印刷中).