

分類：情報提供型

オープングリッドサービスアーキテクチャ活用事例

エディタ：

I. Foster、Argonne & U. Chicago

D. Gannon、Indiana U.

H. Kishimoto、Fujitsu Labs

Jeffrin J. Von Reich、Hewlett Packard

2004年3月4日

オープングリッドサービスアーキテクチャの活用事例

本書の位置付け

本書は、オープングリッドサービスアーキテクチャ（OGSA）プラットフォームコンポーネントの定義のもとに利用されているグリッドの活用シナリオに関する情報を、コミュニティに提供することを目的としている。配布は無制限である。

本書について

分散システムの統合に向けた、広範かつ適用性や採用力に優れたフレームワークを持つオープングリッドサービスアーキテクチャ（OGSA）のビジョンを実現するには、E サイエンスや E ビジネスアプリケーションについて、多種多様なグリッド活用事例のシナリオを定義する必要がある。本書では、商用インフラとアプリケーション（商用データセンター、オンラインメディアやエンターテインメント、インターグリッド）、科学用インフラとアプリケーション（National Fusion Collaboratory、シビアストームモデリング、仮想組織グリッドポータル）、基礎グリッド技術（グリッド資源リセラー、サービスベースの分散クエリプロセッシング、ワークフロー、Grid Lite、インタラクティブグリッド）、そしてワーキンググループの活用事例（相互認可、Persistent Archive、資源利用サービス）を論題として取り上げている。本書に記載のグリッド活用事例リストは完全なものではない。また、活用事例は正式な要件に求められるような詳細なレベルで記述したものではない。

グローバルグリッドフォーラム

office@gridforum.org

www.ggf.org

著作権全文表示

Copyright © Global Grid Forum (2004) All Rights Reserved.

上記著作権表示とこの段落がすべての複製文書や派生的な研究に含まれている限りにおいて、いかなる種類の制限を課すことなく、一部または全部を対象に、この文書およびこの文書の翻訳物を複製し他者に供することができる。また、その内容に対するコメントや説明、あるいはその実装を支援する派生的な研究物を、準備、複製、発行、配布することができる。

しかしながら、この文書自体は、GGF 文書プロセスに定義された著作権の手続きに従わなければならない、あるいはそれを英語以外の言語に翻訳する必要がある場合において、グリッドの推薦内容を発展させる上で必要とされる場合を除き、著作権表示や GGF その他組織への照会情報を削除するなど、いかなる方法によっても変更することはできない。上記に規定する限定的な許可は永続的であり、GGF またはその後継者や譲受人によって無効となることはない。

この文書およびこの中に記載された情報は「現状有姿」を条件に提供される。グローバルグリッドフォーラムは、ここに含まれる情報の利用が、商業利用や特定の目的に合致することのいかなる権利や黙示的な保証も侵害しないという点において明示的、黙示的にすべての保証を拒否する。

知的所有権表示

GGF は、いかなる知的所有権、実装に関わるものとして主張され得るその他の権利、本書に述べられている技術の使用、あるいはこのような権利の行使が可能または不可能とされるライセンスの限度についても、その有効性や適用範囲に関してはいかなる立場もとら

ない。また、そうした権利を特定するために取り組んできた事実はない。出版のために利用できる権利の主張のコピーや利用されるライセンスのいかなる保証、実装者またはこの仕様のユーザによるそうした所有権の使用に対して、一般的なライセンスまたは許可を得るための手続きの結果は、GGF 事務局より入手することができる。

GGF は、すべての利害関係者に対して、この推薦内容を研究するために必要な技術をカバーできる著作権や特許、特許の応用、その他の所有権に対しての問い合わせに応じる。詳細は GGF エグゼクティブディレクタ宛に送ること（連絡先情報は GGF Web サイトを参照のこと）。

目次(訳文のページ番号)

1 はじめに	4
2 商用データセンター	5
3 シビアストームモデリング	16
4 オンラインメディアとエンターテイメント	22
5 National Fusion Collaboratory	35
6 OGSA および OGSA-DAI によるサービススペースの分散クエリプロセッシング	40
7 グリッドワークフロー	53
8 グリッド資源リセラー	59
9 インターグリッド	68
10 インタラクティブグリッド	76
11 Grid Lite	80
12 仮想組織のグリッドポータル	83
13 Persistent Archive	89
14 相互認可	99
15 資源利用サービス (RUS)	101
16 編者の連絡先	104

1 はじめに

OGSA-WG の憲章には次の一文が明記されている。

「OGSA プラットフォームコンポーネントの定義付けと優先付けを促進するための活用事例を作成および文書化するとともに、その選択の理由を裏付ける合理的な解釈を提示すること」

本書は、OGSA-WG 参加者または呼びかけに応じた外部の人々による活用事例シナリオを収録したもので、『オープングリッドサービスアーキテクチャプラットフォーム』の姉妹編となっている。本書では、指定されている OGSA の枠組みにそって活用事例の全調査内容を紹介している。また本書の他に、『オープングリッドサービスアーキテクチャ：Second Tier 活用事例』（文書 ID は GWD-I: draft-ggf-ogsa-tier2-usecase-2）もある。この文書で紹介した活用事例はいずれも完結したのではなく、未だ検討には至っていない。

OGSA-WG は活用事例文書をもとに、(a) 必要とされる重要なサービスの範囲を広範かつ比較的具体的に示し、(b) 多くのグリッドシステムおよびアプリケーションに不可欠なものと考えられるこうしたサービスの中核グループを特定し、(c) これら中核サービスに必要な機能性および中核サービス間の相互関係を上位から説明する。

これらの活用事例は、正式な要件を明示するという意図で定義されたものではないが、（また正式な要件に求められるような詳細なレベルの内容を含んでいないが、）定義プロセスに対して有益な情報をもたらすものである。将来この文書を改訂する際には、さらに多くの活用事例を紹介したいと考えている。

表 1：活用事例と本書への寄稿者

章	タイトル	寄稿者
---	------	-----

2	商用データセンター	Hiro Kishimoto、Andreas Savva、David Snelling
3	シビアストームモデリング	Dennis Gannon
4	オンラインメディアとエンターテインメント	Tan Lu、Boas Betzler
5	National Fusion Collaboratory	Kate Keahey
6	サービススペースの分散クエリプロセッシング	Nedim Alpdemir、Norman Paton
7	グリッドワークフロー	Takuya Araki
8	グリッド資源リセラー	Jon MacLaren、William Lee
9	インターグリッド	Jeffrin J. Von Reich
10	インタラクティブグリッド	Jeffrin J. Von Reich
11	Grid Lite	Jeffrin J. Von Reich
12	仮想組織グリッドポータル	Charles Severance
13	Persistent Archive	PA Working Group of GGF.
14	相互認可	Takuya Mori
15	資源利用サービス (RUS)	Bill Horn

2 商用データセンター

2.1 概要

多くの企業では、サーバや記憶装置といった IT 資源をデータセンターに移行することで、TCO (総所有コスト) の低減をはかっている。また、IT 資源や管理機能のアウトソーシングを実行または計画することにより、自社の中核ビジネスに専念しようとする企業も少なくない。この結果データセンターでは、サーバや記憶装置、ネットワークを含む数千もの IT 資源を管理する必要に迫られている。こうした資源の複雑な管理を簡素化し、可用性を高めるには、革新的なグリッドベースの資源管理ソフトウェアを必要とする。我々はこれを「商用グリッドシステム (CGS)」と呼んでいる。この活用事例で引用されている

グリッド技術または単に「グリッド」という言葉は、すべて CGS を指すものとする。また、CGS を実装するデータセンターを「商用データセンター（CDC）」と呼ぶことにする。

メインフレームが情報技術の中心だった時代には、IT システムインテグレータは、この単一で、堅牢、均質なプラットフォーム上に管理の容易な IT システムを構築することができた。ところが現代の IT システムインテグレータは、さまざまな OS やミドルウェア上で数十種類もの異なる API を使用しなければならない。いずれも（利用率向上のための）障害検出や応答、（パフォーマンス目標を達成するための）パフォーマンスのボトルネックの特定方法に一貫性がなく、結果的に QoS の保証にも一貫性を欠くことになる。CGS が提供するグリッドベースのメタ OS 機能なら、エンドツーエンドの QoS を実現することにより、IT システムインテグレータの負担を軽減することができる。

2.2 顧客

「グリッド管理者」は CDC の重要な参加者である。厳密に言えば、グリッド管理者は顧客ではなくプロバイダである。しかし彼らは、CDC のグリッドによって IT インフラの管理性向上のメリットが得られる。これが CGS の主要なモチベーションの 1 つとなっている。CDC のハードウェアやソフトウェアの管理は煩雑でコストがかかる作業であるため、管理者はプロビジョニングやモニタリング、チューニング、メンテナンス、エラー診断、障害回復といった IT インフラの主機能の自動化を要求するのである。

グリッド管理者に与えられた 1 つの要件は、IT インフラの利用率を高めることである。数名のアナリストの報告によれば、分散化した資源の実際の利用率は 20% に満たないケースが多いが、統合により 70% 以上に高めることができると伝えている。また、フェイルオーバーやプロビジョニングのために確保された資源、つまり生産に関わっていない資源も中にはある。こうした資源は、複数のシステム間で共有できるようにすることが大切で、共有が可能かどうかを物理的な配置のような 1 つの要因だけをもとに決定すべきではない。

グリッドは IT インフラの管理性を向上させる。これにより管理者の数は、たとえば数十名から 10 名以下に削減することができる。

「IT システムインテグレータ」は商用データセンターの顧客である。IT システムインテグレータは、異種のシステムを構築するという困難な問題を抱えている。たとえば、エンドツーエンドのパフォーマンス予測や保証、要求されている可用性レベル（99.99%など）の確実な達成、予想し得ないサービス需要（インターネットスパイクの問題）に対処するための追加的な資源のプロビジョニング、絶えず発生する変更作業（割引やその結果としてのアクセス負荷の変更、製品数、新規サービスなど）などがある。

IT システムインテグレータは、標準および QoS 対応のメタ OS 機能を提供する OGSA ベースのグリッドを活用することで、複雑な分散型および異種システムの構築を簡素化できるのではないかと期待している。

IT システムインテグレータはまた、グリッドを用いて容易にテストシステムを構築（VO の構築を通じて）することも可能である。

商用データセンターのもう一人の顧客は「IT 事業活動マネージャ」である。IT 事業活動マネージャは、たとえば、チケットを「エンドユーザ」に販売するチケットサービスを運営する。エンドユーザは CDC の参加者であるが、その直接の顧客ではない。彼らはチケットサービスの顧客なのである。

現在、CDC を利用する IT 事業活動マネージャはごく少数である。我々は将来的に何百名ものマネージャがそれぞれデータセンターを利用するものと見ている。

下の図は、上に述べた参加者の一部を表したものである。データセンターは現実の組織（RO）に、IT 事業活動は仮想組織（VO）に対応する。IT 事業活動マネージャは VO を組織し、VO が信頼性、拡張性、安全性に優れ、要求通りの QoS を達成するものと期待してサービスを運営する。一方グリッド管理者は RO を管理し、グリッドは彼らの作業負荷を軽減する。

図 1：商用データセンターの RO、VO および顧客

2.3 シナリオ

商用データセンターには次の 4 つのシナリオがある。

2.3.1 複数の社内システム

現在の社内システム、たとえば人事管理システムや財務会計、受注、CRM システムの大半は独立している。それぞれの社内システムは専用の IT 資源上で稼動するだけでなく、高い可用性や作業負荷の増大に備えて予備の IT 資源を確保している。作業負荷はそれぞれ異なり、ピーク時が必ずしも一致するとは限らないため、多くの IT 資源のアイドル状態が発生する。

グリッドは企業の IT 資源の大部分を管理し、必要な資源をそれぞれの社内システムにオンデマンドで提供できる。このため、各システムが必要としている予備資源を複数のシステムで共有できるようになり、IT 資源の利用率が向上する。また、より少ない IT 資源でより多くの社内システムを稼動することができる。

グリッドは、それぞれの社内システムのために事前に予約を行い、ハードウェアの割り当てや必要なソフトウェアやデータを配備し、要求されたアプリケーションを始動する。また、これらのすべての手順は自動化される。

グリッドはまた、フェイルオーバーやプロビジョニングといった自律管理機能を提供する。グリッドは多くの障害に自発的に対処する。

また、複数のリモートデータセンターでは、相互に連携してスケーラビリティと可用性を向上させることができる。地震や火災、テロ行為といった災害時でも、稼動を停止させないようにしなければならない。独立しつつもネットワーク化されたデータセンターなら、必要な物理的インフラの提供に使用することができる。

2.3.2 期間限定の商用キャンペーン

企業は、コンサートチケット販売や国際会議の参加登録、セールスプロモーションといった、期間限定のキャンペーンを計画することが少なくない。こうしたキャンペーンに使用する現行システムでは、需要のピーク時に対応できるように、余剰供給型の固定的な IT 資源を必要とする。このため、初期投資とメンテナンスには過大なコストが発生する。グリッドは必要な IT 資源をオンデマンドで提供し、利用ベースで課金する。

IT 事業活動マネージャはまた、スケーラビリティと可用性の点で、最も低コストのデータセンターを選択したり、複数のデータセンターを利用したりすることができる。

2.3.3 災害復旧

バンキングシステムや航空交通管制システムのような、公共の基幹インフラサービスを提供する IT システムでは、災害復旧機能が不可欠である。また、インターネットの普及により、定評ある Web サイト「Google」の例のように必要不可欠となったアプリケーションも数多くある。しかし、災害復旧機能は構築と運営に莫大なコストがかかる上、技術的にも高度な知識が要求される。

グリッドなら標準的な災害復旧の枠組みを、CDC 全体にわたって低コストでこのような IT 事業活動に提供することができる。

2.3.4 グローバルな負荷分散

地理的に散在する CDC は、高い作業負荷をシェアすることができ、複数のアプリケーションに対する高いスケーラビリティを提供できる。

2.4 関連資源

CDC は、サーバ、記憶装置、データ、ネットワークなどあらゆる種類の IT 資源を装備している。

グリッドは最低でも数千の資源管理が可能である。

2.5 OGSA プラットフォームの機能要件

これまで述べたシナリオについては、以下の機能が必要である。

1. 発見

最初に CDC の参加者は、使用するリファレンスを CDC に提出しなければならない。
最初のステップとして、1 つまたは複数のよく知られた発見サービスを用いる。

2. 認証、許可、アカウントिंग (AAA) [1]

顧客がジョブの要求を送信すると、CDC は顧客を認証し、送信された要求を許可する。
CDC はポリシーの識別も行う (SLA、セキュリティ、スケジューリング、ブローカリングポリシー)。グリッドは顧客が送信された要求を実行する権限があるかどうかをチェックする。

[1] この機能は OGSA プラットフォーム機能に追加する必要がある。

3. 事前予約 [2]

グリッドは、顧客の要求に基づいて要求の処理を開始する日時を登録する。 [3]

グリッドはジョブの仕様記述言語を要求が記述された言語に翻訳する。

[2] この機能は OGSA プラットフォーム機能に追加する必要がある。

[3] 「要求の処理」と「ジョブの処理」は異なる。事前予約の場合、要求の処理は将来の使用のために資源を予約しますが、ジョブの処理は予約時に実際に処理を実行する。

4. ブローカリング

グリッドは要求された時間間隔について最適な資源を探す (予約のための要求と仮定して)。資源に対するアクセスコントロールと配分量も適用される。予約が行われ、その参照が顧客に返される。

5. データ共有

ジョブの要求では必要なユーザデータも指定する (データベースやファイル)。組み合わせの選定にはデータのアクセシビリティも検討する必要がある。

6. プロビジョニング

予約の時間となる少し前に、グリッドはアプリケーションとユーザデータの配備を開始する。Java プログラムの場合、グリッドは指定された Java プログラム (jar ファイル) を

探してこれを予約資源に配備する。Java の配備機能は、すでにほとんどのホスティング環境で明確に定義およびサポートされている。

7. スケジューリング [4]

予約の時間が到来すると、グリッドはタスクを開始します。

[4] この機能は OGSA プラットフォーム機能に追加する必要がある。

8. メータリングとアカウントिंग

ジョブの実行中、メータリングサービスが資源の利用状況を追跡する。この情報はアカウントングサービスに渡される。

9. 障害対応 [5]

この活用事例では、ジョブがエラーに遭遇し、正常に完了できない場合、顧客は障害通知のみ受ければよいものと仮定している（障害対応手順は障害管理ポリシーに明記されている）。

[5] この機能は、[資料 : 1] で「フォールトトレラント」と呼ばれている。本書では、より汎用的な機能をカバーするため、「障害対応」と名称を変更している。

10. ポリシー

いくつかの属性はポリシーとして扱う必要がある。ブローカリングポリシーは顧客ごとの資源の利用配分を定義する。エラーおよびイベントポリシーは、プロビジョニングやフェイルオーバーといった自律管理を規定する。

11. セキュリティ

同一のデータセンター内での顧客の独立性は、重要な要件である。グリッドはアクセスコントロールだけでなくパフォーマンスの独立性も提供しなければならない。

「期間限定の商用キャンペーン」シナリオでは、上記の他に以下の機能を必要とする。

12. 仮想組織

顧客がジョブを要求すると、グリッドは IT 資源をジョブに提供するデータセンターに VO を構築する。グリッドは顧客の要求に基づいて、遠隔の CDC にある別のグリッドとネゴシエートし、CDC 全体にわたって VO を構築する。こうした VO は、必要なスケラビリティと可用性の達成に用いられる。

13. モニタリング

顧客は、遠隔のデータセンターで稼動しているアプリケーションの監視を要求できる。

14. 負荷分散

グリッドはジョブのパフォーマンスを監視し、負荷に応じた資源配分を調整することで、エンドユーザの要求をすべての資源に適正に配分する。

「災害復旧」シナリオでは、上記の他に以下の機能を必要とする。

15. 災害復旧

地震や火災等の災害により、データセンターが利用不能となった場合には、遠隔のバックアップ用データセンターがアプリケーションを引き継ぐ。

「グローバルな負荷分散」に述べたシナリオでは、他の追加的機能は必要ない。

2.6 OGSA プラットフォームサービスの利用度

以下は、前節の機能を提供する上で必要なサービスである。

1. 名前解決および発見サービス

このサービスは発見機能としてグリッドで使用する。

2. セキュリティサービス

このサービスは、OGSA AAA 機能に必要なものである。また、資源アクセスコントロールでもこのセキュリティサービスを必要とする。

3. 予約サービス

このサービスは事前予約に使用する。

4. ブローカリングサービス

このサービスは資源のブローカリングに使用する。

5. データ管理サービス

このサービスは、データセンター内およびデータセンター間のデータ共有に使用する。また、災害復旧にも使用される。

6. プロビジョニングおよび資源管理サービス

このサービスは、プロビジョニングおよびリモートサイトの VO 構築に使用する。

7. スケジューリングサービス

このサービスは優先ジョブスケジューリングに使用する。

8. メータリングおよびアカウントティングサービス

このサービスは、メータリングおよびアカウントティングに使用する。

9. 障害対応サービス

このサービスは障害対応に使用するもので、自律管理機能の 1 つである。災害復旧の際には、影響を受けた IT 業務アクティビティは、他のデータセンターに移行される。

10. ポリシーサービス [6]

このサービスは、ポリシー関連機能に使用する。

[6] [1] のポリシーサービスの説明は漠然としていて意味が明確ではない。

11. モニタリングサービス

このサービスは、モニタリング機能に使用する。

12. デプロイメントサービス

このサービスは、プロビジョニング機能に使用する。

2.7 セキュリティ要件

各商用 IT システム（VO に対応）は、相互に安全に独立していなければならない。競合企業が同じデータセンター（RO）内に同居している可能性があるためである。商用システムを開始する前に、VLAN または同等の技術を用いて VO を分割する必要がある。作業負荷が増えた時は、ネットワークの再構成を行って別のシステムに IT 資源（サーバなど）を移動するが、情報が漏れないよう注意しなければならない。

WS セキュリティが出発点となるが、商用グリッドシステムではいくつかの拡張も必要な場合がある。

VO は一箇所のデータセンターでも、複数のデータセンターに設けても構わない。ただし災害復旧や広域的な負荷分散の場合、VO は複数のデータセンターを使用するものとする。

2.8 パフォーマンス要件

サイエンスグリッドとは異なり、商用グリッドでは実行速度を最優先課題に据えない。代わりに、いくつかのサービス品質マトリックスを検討する必要がある。ベストエフォート型のスキームが必ずしも商用グリッド要件を満足するとは限らない。それぞれのジョブ要求は指定された日時に完了しなければならないため、事前の資源予約による期限スケジューリングがベースラインの条件となっているためである。通常、ジョブは事前に指定された一定期間実行され、一定のレベルのパフォーマンスを提供するものとされている。

インターネットスパイクの問題を回避するには、適応可能な資源配分（プロビジョニングなど）によって要求されたスループットの拡充をはかる。

各 IT システム管理者は、サービス品質保証契約（SLA）の要件を明示する。それぞれのジョブは、障害が発生すると、SLA に基づいて過負荷時の追加的資源や代替資源を要求する。要求をすべて満たすのが困難な場合は、SLA に基づいて優先度の低い要求が拒否

される。

2.9 活用事例の状況分析

上に述べたような単一または複数の問題を解決するため、すでになんらかの先端技術 [7] [8] や製品 [9] が市場に投入されている。こうした試みは独自のアプローチを採るもので、適用に限界がある。

一方 OGSA は、オープンで拡張性がある包括的アーキテクチャであり、これらの問題の解決に適している。

現在我々は調査段階にある。調査が完了した後は OGSA ベースの CGS のプロトタイプ構築に取り組むと考えている。

[7] Océano Project、IBM. www.research.ibm.com/oceanoproject

[8] N1、Sun Microsystems. www.sun.com/software/solutions/n1

[9] Jareva、<http://www.jareva.com>

2.10 参考資料

1. Foster、I and Gannon、D. 「オーブングリッドサービスアーキテクチャプラットフォーム」(2003)。

www-unix.gridforum.org/mail_archive/ogsa-wg/doc00016.doc

2. Kishimoto、H.、Savva、A.、Snelling、D. OGSA 基本サービス：「商用グリッドシステムの要件」OGSA-WG 文書、2002 年 10 月 14 日

www-unix.gridforum.org/mail_archive/ogsa-wg/pdf00002.pdf

3 シビアストームモデリング

3.1 概要

気象学者と環境モデラーのコンソーシアムは、データモデリングとリアルタイム広域気象計測器および大規模シミュレーションを連携した、竜巻などの暴風域を正確に予測するグリッドの構築に取り組んでいる。これは極めて困難な課題であり、現行のストームシミュレーションの能力をはるかに超えるものである。現在の気象学者が言えるのは、シビアス

トーム（嵐や竜巻）の発生条件が整っているということだけであり、警告は実際の気象観測をもとに出されている。観測によってストームの進路を予測することは可能だが、既存の計算およびデータ分析機能では、ストームが特定の地点に出現することを確実に予測することまではできない。

3.2 顧客

主要な顧客は気象学者である。彼らは実際にグリッド資源を活用しなければならない。この仮想組織は広く分散しており、頻繁に移動する。二次顧客群としては、緊急対策管理者や災害復旧チーム、マスメディアが考えられる。

3.3 シナリオ

このシナリオの概略は以下のとおりである。ドップラーレーダーや衛星画像、圧力や温度、湿度検知計などの地上ベースのセンサーからのインストルメントデータストリームは、データマイニングエージェントが常にモニタリングし、危険なパターンが現れていないか観測する。危険なパターンが現れると、VOメンバーに通知され、大規模なシミュレーションを自動的に開始する。データマイニングツールが設定され、出力したシミュレーションをスキャンする。その結果は、刻々と変化する計測器からのデータストリームと比較される。また、データアーカイブを検索して類似パターンを探る。計測器の一部は自動的に再設定され、データストリームの精度を更新する。

ストームの変化にともない、追加的なシミュレーションを開始して予測の解析精度を高める。重要なイベントを検出すると、人手でプロセス全体をモニタリングし、一部のシミュレーションの方向性を選択することでプロセスを推進する（シミュレーションはアニメーションとしてビジュアライズ可能な出力ファイルを生成）。現場にいるその他のスタッフは、モバイル機器から、さらに多くのデータを入力する。予測は当局やメディアに伝えらる。

このシナリオはまだ実現には至っていない。グリッドのインフラが整っていないためである。今日数多くのさまざまなコンポーネントが存在するが、これらすべてを統合できるというわけではない。現在このグループでは、シミュレーションやデータマイニングのテス

ト、シミュレーションとデータストリームの統合に関する共同研究を中心に活動している。

3.4 関連資源

主要な関連資源は以下のとおりである。

- 複数の機関が参加するセンサーネットワーク
- 過去に発生したストームの履歴と計測記録のデータアーカイブ
- テラグリッド資源を含む計算資源

実現するサービスは

VO メンバーがシミュレーションやデータマイニングツール、データアーカイブ、センサーネットワークツールにアクセスするための統合グリッド

上に述べたシナリオを実行するグリッドサービス。最終的には自動化、自律型であること

3.5 OGSA プラットフォームの機能要件

3.5.1 基本機能

発見とブローカリング：天候が悪化した時は、非常に大量のシミュレーションとデータマイニングの連携作業が動的に始動する。これには、異なる規模の資源を特定する資源の発見およびブローカリングが必要である。

データ共有：非常に大規模な気象履歴データベース（レーダーデータその他の地上、空間ベースのデータを含む）に常にアクセスできなければならない。この情報は数百もの異なるデータベースを通じて配布される。刻々と変化する気象は、データマイニングサービスによる履歴情報に照らして追跡され、シミュレーションの境界条件の管理に使用される。

仮想組織：どの計測器やデータ、計算資源に誰がアクセスできるかは、極めて重要な要件である。

モニタリング：計算資源を継続的に維持するには、常に大規模シミュレーションをモニタリングする必要がある。計測器や計算/データグリッド全体を絶えずモニタリングしなけ

ればならない。

ポリシー：VO のメンバーによるデータベースや計測器、シミュレーションへのアクセスを管理するポリシーである。ポリシーは、シビアストームが予想された場合に通知すべき担当者も定義する。通知プロセスは自動的に実行される。

3.5.2 セキュリティ機能

複数のセキュリティインフラ：オンラインの計測器を管理できるセキュリティ管理機能である。

認証、許可、アカウントिंग：これらはすべて、VO の個人管理とそれぞれの権限の確立に欠かせない。

新規サービスの具体化：サービスの多くは一過性のシミュレーションやデータマイニングサービスである。これらはデータをモニタリングしているエージェントによって進行過程で具体化されなければならない。

3.5.3 資源管理機能

事前予約：スケジュール化されたデータ分析作業の多くが予約を必要とする。しかし、最も重要な作業は動的にスケジュールリングする必要がある。

スケジュールリング：動的スケジュールリングはこのシナリオに不可欠な要素である。時間通りに予測を完了するためにも、計算資源はオンデマンドで供給されなければならない。

負荷分散：1 つの資源がシミュレーションとデータマイニング作業で過負荷に陥った場合、別の計算エンジンを投入して負荷のバランスをとる必要がある。

通知 / メッセージング：この極めて動的なシナリオでは、通知およびメッセージングは非常に重要である。

完全なイベント駆動型機能である。

ロギング：前回のストームで何が起こったかを理解する上でロギング機能は不可欠である。パフォーマンスは後で最適化可能である。

ワークフロー管理：ワークフローは極めて動的で、イベント駆動型である。

システム属性

フォールトトレランス：優れたフォールトトレランスにより、リアルタイム予測はより確実になる。シビアストーム追跡中にグリッドが停止することは許されない。

災害復旧：極めて高速であること。このためには、すべてのコンピューテーションをミラーリングし、大幅に分散化する必要があるかもしれない。

自己回復機能：すべての分析/シミュレーション/予測シナリオでは、それぞれのエラーを修正できなければならない。

3.6 OGSA プラットフォームサービスの利用

必要なサービス：

名前解決および発見：シビアストームモデリングでは、メタデータの記述からデータ資源とデータカタログを発見できなければならない。これは発見の一部である。

サービスドメイン：サービスの集合体は慎重に統合する必要がある。資源ブローカは計算およびデータ記憶資源を保証し、ネットワーク帯域幅は必要な時にシミュレーションや分析で利用できることが条件である。したがってこうした異なるタイプのブローカは慎重に統合する必要がある。

セキュリティ：認証はすべての VO メンバーに必要である。しかしデータや計測器などの特定の資源にアクセスするユーザを管理するための入念な許可ポリシーも重要です。たと

えば、計測器へのアクセスを一部の VO メンバーのみに許可するといった管理が可能である。

ポリシー：ポリシーは基本的に計測器へのアクセスに関わる問題である。どのような条件下にレーダーを再配備できるかといった問題である。また、ポリシーは特定のサービス稼働システムで、公共の安全に不可欠なものとして「認知」されるものに対し、資源をプリエンブションするタイミングを決定する。

データ管理：データグリッドサービス：メタデータカタログ、ディレクトリおよびインデックスサービス、データアーカイブへのグリッド全体のアクセス、仮想データ管理。

メッセージング、キューイング、ロギング：必要な資源を必要な時に提供するため、資源ブローカはグリッド全体をモニタリングする必要がある。アプリケーションワークフローは極めて動的な「需要駆動型」の性格を持つことから、メッセージングとイベントシステムが必要である。また、エラーの原因を特定するにはロギングサービスが欠かせない。

イベント：イベントはこの活用事例の必須コンポーネントである。モニターは常時計測器のデータストリームをスキャンし、ストームの発生可能性を観測している。ストームを発見すると、イベントとメッセージ (pub/sub) システムが、シミュレーションその他のデータマイニングアプリケーションの開始に不可欠なワークフローシナリオをトリガする。

メータリングとアカウントティング：資源の利用はコストがかかる。したがって課金を実施し、その実施に必要な情報を提供しなければならない。

サービスの統合：ワークフローエンジンは、連携するシミュレーション/データマイニング/ビジュアライゼーション作業を統合する必要がある。ワークフローは極めて動的な性格を持っている。気象条件の変化といった外部のイベントは、作業の流れを変えることがある。作業時間が逼迫することもある。予測が時間通りに完了しない場合には、より多くの資源を割り当てる必要も出てくる。

管理：ソフトウェアの運用は重要な管理上の問題である。

プロビジョニングと資源管理：資源要件は極めて動的に変化する。緊急時には、計算機能や帯域幅、データなどの資源を大量に投入できることが必要である。

予約サービス：これは、プロビジョニングと資源管理を参照すること。

ブローカリングとスケジューリングサービス：計算およびデータ資源のブローカリングサービスが必要である。

スケジューリングと共同スケジューリングサービスが必要となる。

障害対応サービス：実時間に先行する予測を行うためには、システムの冗長構成で障害に対処しなければならない。

モニタリングサービス：グリッド全体にわたるモニタリング、メッセージング、イベントシステム、ロギングサービス。

3.7 セキュリティ要件

最も重要なセキュリティ要件は、許可されていないユーザに計測器管理へのアクセス権が付与された場合である。使い方を誤ると計測器に重大な損害を与える可能性がある。

3.8 パフォーマンス要件

パフォーマンスは活用事例の極めて重要な構成要素である。ストームの予測は実時間に先行する必要があるため、計算やネットワーク帯域資源を同時進行的に大量配分しなければならない可能性がある。1個のストームだけでも、数時間で100テラフロップの専用パフォーマンスが必要な場合もある。現状ではこれは不可能である。

3.9 活用事例の状況分析

現時点では、必要なサービスはいずれも実施されていない。しかし、計測器とデータネッ

トワークは利用されており、早くから数多くの特定の実験が行われていない。

3.10 参考資料

『 A Modeling Environment for Atmospheric Discovery 』、 the NCSA MEAD Expedition、参照サイトは <http://www.ncsa.uiuc.edu/expeditions/MEAD>

4 オンラインメディアとエンターテインメント

4.1 概要

エンターテインメントは、複数の参加者がこの体験を提供するために VO を形成する。最初のステップでは参加者の以下の役割に焦点を当てたいと思う。

- エンターテインメントコンテンツを獲得する消費者
- エンターテインメントコンテンツを供給するサービスプロバイダ
- エンターテインメントコンテンツを提供する発行者
- エンターテインメントコンテンツを獲得する開発者

それぞれの役割は複数の企業で構成されたり、エンターテインメントコンテンツも、異なるホスティング容量ニーズやライフサイクルから成る多種多様な形態（オンデマンドの映画やオンラインゲームなど）で構成されることもある。したがってこの活用事例では、主要な論点の 1 つとして作業負荷ニーズや現行のシステム構成に基づく動的な資源管理の簡素化に焦点を当てる。

エンターテインメントコンテンツのライフタイムでは、コンテンツの配布に関わる参加者が変わる場合がある。企業のライフタイムの場合、取り扱うべきエンターテインメントコンテンツが変わることもある。したがって、この活用事例のもう 1 つの主要な論点として、動的かつオープンなコラボレーションを可能にする標準インタフェースの提供に焦点を当てる。

4.2 顧客とニーズ

エンターテインメント体験には、これを提供するインフラに関して独自の要件を持つ 2 つの大きなカテゴリ、すなわち獲得とインタラクション（対話性）がある。コンテンツの獲得

(ビデオオンデマンドなど)は、あまりユーザとの対話を必要としない。しかしオンラインゲームのような別のコンテンツでは、多くのユーザインタラクションを必要とし、こうしたコンテンツの応答時間の保証が非常に重要な鍵となる。

オンラインエンターテイメントは、この数ヶ月間で大幅に採用が増えたが、コンテンツやビジネスモデル、インフラの分野では、依然として未成熟な段階にある。オンラインコンテンツの提供が増えれば、競合企業の差別化が重視されるようになる。ヘビーユーザを対象とした利用ベースの料金や購読モデルなど、新たな商機も登場すると考えられる。商取引はエンターテイメントに関連付けられ、あるいはユーザ体験固有のものとなる。

これは新しい分野であるだけに、コンテンツ開発者には分散ネットワーク向けのプログラミング能力が不足している。コンテンツの配布にバックエンドのデータセンターをどう利用すべきかについて、標準的なアーキテクチャもベストプラクティスさえも存在しない。今日最も一般的な方法は、それぞれのゲームタイトルについてストープパイプ式のソリューションを設計し、各ソリューションを個別に管理することである。結果的に、それぞれのゲームに導入されたインフラとコンポーネントは再利用されない。また、こうしたストープパイプソリューションは一定レベルの作業負荷(10,000 同時ユーザなど)を想定して設計されているため、この初期の想定量を超える拡張には、大幅な設計変更が必要となる。このため、今日のデータセンターは、供給過剰かサービス停止に追い込まれるほど逼迫しているかのいずれかに陥っている。さらに困ったことに、一度設計したゲームについては、その存続期間を知る手段がない。つまり、この種のゲームについては、わずか数日(ベータテスト環境の場合)、あるいは数年(Everquest など)しかデータセンターを必要としない可能性もあるということだ。

4.3 シナリオ

このシナリオは、ユーザ、サービスプロバイダ、発行者、開発者の4人の参加者で構成されている。ゲームプレイヤーなどのユーザは、ポータルにアクセスし、既知の利用者として認証される。

この許可を得た後、消費者は自分のアカウントで対話、あるいはオンラインゲームなどの提供されたエンターテイメント体験を獲得することができる。中には相互に連携して運営

する複数のサービスプロバイダもいる。たとえば、ネットワークサービスプロバイダは帯域を、容量ホスティングプロバイダはサーバや記憶装置資源を、そしてアプリケーションサービスプロバイダはオンラインゲームエンジンや標準的な CRM、ヘルプデスクアプリケーションや課金アプリケーションといった一般的なサービスを提供します。コンテンツプロバイダまたはスタジオは、ユーザが体験するメディアコンテンツやアートワーク、ゲームプレイを提供する。インテグレータや発行者はともに連携して商品をユーザに告知する。下図は各参加者の相互関係を簡単に表したものである。参加者間のインタラクションは変更される可能性があり、エンターテインメントコンテンツも同様である。したがって、動的な発見や提供されたインフラおよびサービスのインタラクションを可能にするだけでなく、資源配分を自動的に管理できることが重要な要件となる。

-(オリジナル図版参照)

下表は参加者それぞれの主な活動を表している。

ユーザ	発行者	スタジオ	xSP
-----	-----	------	-----

アカウントの登録（xSPにて）			アカウントの作成
	ユーザ購読サービスの作成		
		ホスティング環境の購入と購読	発行者向けビジネス提案の作成（オンデマンド環境）
			ゲーム環境の購読提供（予約/スケジューリング、購入含む）
	ユーザ購読サービスの削除		提案の削除
コンテンツ/ゲームの購読（発行者による）	許可の作成		許可情報の取得
許可/認証			許可/認証
コンテンツの検索			利用可能なコンテンツの発行
M&Eセッションの作成			
コンテンツの取得/使用			オンデマンドホスティング環境の構築（プロビジョニング、フェイルオーバー、作業負荷管理）
			資源の監視
			物理資源の追加
			新機能/サービスの追加
			機能/サービスのアップグレード

			環境オンデマンド提案の 削除
			サーバプールから物理資 源を削除
			資源 / サービスの削除
			負荷分散
			エラーのキャプチャ、問 題の特定、フェイルオー バ、回復
	メータリング 要件の定義		メータ利用
クライアントパッチ / PTF の適用			
			E コマースの統合
	課金およびレ ーティングパ ッケージに基 づく課金レコ ードの作成		
			課金およびレーティング パッケージに基づく課金 レコードの作成
	プレイヤーに 利用料金を請 求 (月、時間 単位など)		発行者に利用料金 / 利用 面積を請求

4.4 資源とサービス

オンラインエンターテイメントのデータセンターは、実現可能な分散環境では少なくとも以下のコンポーネントで構成される。

- 分散サーバ

- ネットワーク化された記憶装置

- セキュアネットワーク（マルチレベルのファイアウォールを含む）

- プレイヤーコンソール

オンラインエンターテイメントビジネスには少なくとも以下の機能がある。

- セキュリティサービス（認証／許可、識別マッピング他）

- フィナンシャルサービス（課金、レーティング、アカウントینگ他）

- 契約／決済サービス

- 顧客関係サービス（ユーザの活動のロギングとデータマイニング）

- 管理サービス（キャパシティ管理、作業負荷管理）

- メディア／エンターテイメント固有のサービス（マルチモーダル入力など）

第 2 節で特定した問題を解決するには、オンラインエンターテイメント環境をホスティングするインフラで以下の要件を満たす必要がある。

標準的なプラグイン可能なコンポーネントによる動的な構成（課金サービスや顧客関係サービスなど）。

- セキュアで信頼性に優れていること。

オンデマンド容量（作業負荷に応じた自動拡張）、新規サービスのアグリゲーション／選択、ノウハウを必要とする他企業との統合。

今日オンラインゲーム業界には、大きな信用上の障壁があり、発行者は資源やコンポーネントの共有に積極的ではない。この信用障壁を克服するには、コンポーネントは業界標準のインタフェースをベースとし、動的に交換できるものでなければならない（広範なプロバイダの中からコンポーネントを選定できる柔軟性など）。

- 斬新な商用ビジネスモデルを実現できること。

- オンラインゲームのアプリケーションニーズに適用できること。

次節では、さらに具体的な機能要件を、いくつかの事例を交えて紹介する。

4.5 機能的要件

4.5.1 発見

OGSA サービスは、実行時およびセットアップ時の両面で発見能力を持たなければならない。たとえば、ゲーム開発者は一連のレンダリングエンジンを探し出し、エンドユーザの画面解像度や接続帯域に基づいて特定のエンジンを採用する必要がある。

OGSA の発見では、マスキングが必要、つまり、数あるサービスの中で、ユーザの許可やサービスレベルに基づいて発見し得ないようなサービスを提示できなければならない。企業にはそれぞれ異なる信用水準がある。共同開発契約を締結している企業に対してはソフトウェアスタックのすべてのコンポーネントを公開するが、他社に対しては非公開とする企業もあるだろう。

4.5.2 新規サービスの具体化

新しいサービスの具体化が必要となる場合もある。たとえば、オンラインゲームに新たに 2,000 ユーザが参加する場合、こうした追加的プレイヤーに対応するには新たにゲームサーバを供給しなければならない。新しいサーバを供給するには、必要なサービスを具体化する必要があるが、これには導入とスケジューリング/ディスパッチングの 2 つの側面がある。導入は、必要なファイルやデータをサーバに移送する作業である。スケジューリング/ディスパッチングの例としては、1) 一定時間のサーバ資源の予約 (AI ロジックの実行に 2 時間予約するなど)、2) 実行順序と予約を満たせるかどうかの決定、3) 指定された時間が到来したら適切なプロセスをディスパッチする、などが挙げらる。

4.5.3 サービスレベル管理

オンラインエンターテイメントの管理で最も重要なサービスレベルの 1 つは、応答時間である。

たとえば、最初のユーザのゲームには 50ms の応答時間を保証し、RPG には 100ms を保証する場合が考えらる。

4.5.4 メータリングとアカウントティング

各消費者と各プロバイダに関して、資源の利用状況をログに記録する必要がある。この情報は、利用度に基づくユーザへの請求だけでなく、プロバイダが料金設定を決定するためのコスト分析にも使用される。

4.5.5 モニタリング

資源またはサービスのオーナーは、これらの資源やサービスのユーザがその状態情報を用いて利用状況を管理できるように、特定の状態を明らかにする必要がある。

4.5.6 ポリシー

ポリシーは、インフラのどのレベルにも存在すると言える。資源の監視および管理の仕方を規定する下位のポリシーから、課金などの業務プロセスの管理方法を規定する上位のポリシーまでさまざまである。上位のポリシーは、場合によっては下位のポリシーに分割されることもある。

4.5.7 ポリシーおよび機能要件に基づくサービスのグルーピング/アグリゲーション
オンラインゲームの場合、ゲーム開発者はネットワークのプログラミングやレーティング、課金、E コマースの統合など、多くの領域で知識や経験が不足している。このため、既存のサービスを用いてサービスを構成することが鍵となる。オンラインゲーム開発者が必要とする構成方法は、大きく分けて 2 種類ある。選択とアグリゲーションである。選択は、同じ操作インタフェースを持つ多くのサービスの中から、特定のサービスを選択する作業である（最高速の MP3 エンコーダを選択する場合など）。アグリゲーションはサービス間で機能フロー（ワークフロー）を組み合わせる作業である。たとえば、アカウントティングサービスのアウトプットをレーティングサービスに投入して課金記録を作成することが考えられる。アグリゲーションサービスに必要なもう 1 つの基本機能は、データやインタフェースのシンタックスや意味を変換することである。

4.5.8 セキュリティ

極めて柔軟な環境では、資源は長期にわたって複数のコンテンツタイトルに使用される。このためコンテンツプロバイダ側では、動的な環境が、一貫性あるユーザ体験の提供とい

う目標の妨げとならないよう信用を確立しておかなければならない。それぞれのコンテンツ商品が適度に独立性を保っていることも重要である。このレベルの独立性は、インフラ内のセキュリティ機能で確保すべきものである。

また、次のようなセキュリティ関連のサービスも必要である。

シングルサインオンのサポート。一人のプレイヤーが M&E 環境で複数の組織を横断することも考えられる。たとえば Everquest のプレイヤーは、e-bay で Everquest のキャラクターを購入し、pay-pal のアカウントで代金決済を行うかも知れない。シングルサインオン機能をサポートするには、ゲーム開発者は、サードパーティの認証および許可サービスや、認証マッピングサービスなどを利用することができる。

デジタル権限管理および鍵管理

侵入検知および保護機能

4.5.9 認定

信用組織が特定のサービスで意義のある活動を行っていることを証明するものである。たとえば、ある企業は、Yahoo ショッピングが認定した E コマースサービスのみを利用する。

4.5.10 ライフサイクル/変更管理

導入および稼働中のサービスに対する影響を最小限に留めてサービスをアップグレードまたは廃止するものである。これは必要なサービスを供給し、選択ルールやワークフローを変更することによって現在稼働している環境を動的に変えるワークフローで達成することができる。

4.5.11 障害管理

OGSI ソフト状態管理は、活動機能の実装方法の 1 つです。資源計装は、資源の稼働状況に関する追加的な情報を提供する。

ロギングサービスは、資源のパフォーマンスに関する履歴の追跡だけでなく、障害の発見や回復アクションのトリガにも有用です。たとえば、ソフトウェアが原因でゲームサーバのパフォーマンスが低下した場合、パッチを適用する場合などが考えられる。

4.5.12 プロビジョニング管理

M&E 業界のオンラインゲームを例にとると、オンラインゲームの作業負荷は、均一な正弦波に極めて近いものだが、典型的なサーバファームの利用率は依然として 20%程度に過ぎない。データセンターのプロバイダにとっては供給量がピーク時の作業負荷を超えないことが理想的だが、代わりに、予測と反響度をもとに必要なサービス品質保証契約を満たすだけの容量を採用することもできる。

(オリジナル図版参照)

4.5.13 作業負荷管理

オンラインゲームを例にとると、作業負荷の総量はそのままゲームサーバを利用する同時プレイヤー数の結果に比例する。ゲームの世界で 200 平方マイルをカバーするゲームサーバ A があり、このエリア内で戦闘が起こって多くのプレイヤーが殺到すれば、サーバの作業負荷は増大する。また、プレイヤーが他のエリアからこのエリアに流入するにつれて、他のサーバの作業負荷は減少する。したがってサーバ A の作業負荷が一定のしきい値を超えた時は、負荷分散ルーチンをトリガして資源（サーバなど）の再配分をはかる必要がある。言い換えれば、容量に余裕のあるサーバ間で作業負荷を分散し直すということである。

4.5.14 アプリケーション固有（マルチモーダル入力など）のサービス

音声認識エンジンなどの、ドメイン固有のサービスが必要なケースもある。

4.6 OGSA サービスマッピング

(オリジナル図版参照)

4.6.1 セキュリティ要件

消費者、サービスプロバイダ、開発者、発行者はそれぞれのセキュリティアイデンティティとコンテキストを有している（他の実体との関係など）。従来より企業環境に備わっていたすべてのセキュリティ機能についても、プライバシーや否認を含め、取り組まなけれ

ばならない。

4.6.2 パフォーマンス要件

バックエンドのサービインフラでは、同時利用者数とコンテンツ量の増加に応じて拡張できることが重要である。スケーラビリティのもう 1 つの側面として、単一のデータセンターが提供するコンテンツピースまたはゲームタイトルの数量がある。新しいタイトルもまた、プレイヤーごとにより多くの計算、ネットワーク、記憶装置資源が必要となるだろう。

4.6.3 状況分析

上に述べたような単一または複数の問題を解決するため、すでにいくつかの先端技術や製品が市場に投入されている。こうした試みは独自のアプローチを採るもので、適用に限界がある。

一方 OGSA は、オープンで拡張性がある包括的アーキテクチャであり、これらの問題の解決に利用できる。

現在我々は構想の検証段階にあり、検証が完了した後は OGSA ベースの商用グリッドシステムのプロトタイプ化に取り組みたいと考えている。

4.7 参考資料

5 National Fusion Collaboratory

5.1 概要

National Fusion Collaboratory (NFC) プロジェクト [参考資料 : 2] では、核融合研究のための仮想組織を定義し、このコミュニティで開発および実行するソフトウェアのニーズに取り組んでいる。これまでは、一般に開発者がソフトウェアを標準的な一連のプラットフォームに移植し、コミュニティユーザはこのソフトウェアを自分のマシンにインストールして使用方法を採っていた。

しかしユーザやプロバイダにとって、このプロセスは複雑なものである。バイナリとその従属物の複雑なインストールプロセスの洗礼を受けたユーザは、今度は新バージョンが登場する度にソフトウェアをメンテナンスしなければならない。科学向けのコードは概して何十年にもわたって開発および改善され続けるため、最新のモデリングやシミュレーショ

ン技法を反映させるには頻りにアップデートしなければならないほどシステムが複雑化しているという事実が、いっそうこのプロセスを困難なものにしている。

プロバイダから見れば、一部のプラットフォームに限定してソフトウェアをサポートする場合でも、多大なコストと手間がかかる。また、使い慣れていないプラットフォーム上でコミュニティソフトウェアのメンテナンスやデバッグを行うことは、問題の修正はもとより、手直しのために莫大な手間がかかることを意味する。

こうした背景もあって、核融合コミュニティは最近アプリケーションサービスプロバイダ（ASP）モデルの採用を決定した。これは「ネットワークサービスモデル」とも呼ばれている。ネットワークサービスモデルでは、ソフトウェアと一連の普及型のプラットフォームは、サービスプロバイダが提供または契約し、リモートでクライアントがアクセスできるようにする。サービスプロバイダは一連の適正なソフトウェアバージョンを維持管理するだけでなく、デバッグを行ったり、あるいはクライアントが目的の機能を満たすよう稼働状況を管理したりする。具体的には、ソフトウェア稼働の最大効率化、所定の時間枠内での稼働、一定の精度を持つ結果の生成などが含まれる（詳細は次節を参照）。クライアントはこれらの目標を設定し、リモートでコードを実行するため、メンテナンスコストを一掃できる。この共有方法は核融合コミュニティにとって初めてのもののだが、ソフトウェアやハードウェア資源の共有促進や、研究者がソフトウェア実装に関する細々した知識に悩まされることなく物理学に専念できることなどにより、急速に普及が進んでいる。

5.2 顧客

この活用事例の顧客は核融合科学者である。上に述べたサービスプロバイダは、一連の使い慣れたプラットフォーム上でサービスを提供することにより、メンテナンスコストを削減する方法を模索する。一方サービスクライアントは、ソフトウェアをリモートから実行して所定の目標を満たす方法、とくに核実験中に所定の時間枠内で実行できる能力を探求する。この環境では2つの主要な問題が浮上する。信用と管理の問題である。

信用問題では、自分のソフトウェアは必要な時に優先的に動くか、ソフトウェア/ハードウェア資源プロバイダとどのように契約すればよいのか、この契約の履行でどのような保

証が得られるのか、といった疑問に直面する。またプロバイダ側では、このサービスの導入による安全性の確保と、動的に変化するユーザコミュニティへの対処をどう保証すればよいか、という問題もある。

この環境で信頼性あるサービスを提供するにはどうすべきか、クライアントのニーズにどう応えるか、といった疑問に対処する管理上の問題。こうした問題はいずれも、国内だけでなく後の段階では最終的に数百、数千ユーザから成る国際的な広域導入の問題として取り組む必要がある。

以下の内容はクライアントとプロバイダの具体的なニーズをまとめたものである。

核融合実験中の QoS ベースの実行：パルスモードで稼動する磁気核融合実験では、15～20 分毎に最大 10 秒持続するプラズマを生成する。また実験ごとに複数のパルスを使用する。次のプラズマパルスへの変更は、前回のプラズマパルスから得た測定値（数百メガバイトのデータ）を、パルス間のおよそ 15 分以内に分析して決定する。このオペレーションモードは、リモート資源で動作するソフトウェアを使ってより短時間で分析とシミュレーションを実行できれば、それだけ効率的に行うことができるが、これはソフトウェアの実行時間が保証されている場合に限る。現在の機能を考えれば、「パルス間」分析に新しいソフトウェアを採用するには、通常、実験中に現場で専用に稼動する新しいクラスタを購入しなければならない。長期的に見ると、明らかにこのオペレーションモードでは拡張できない。所定の時間枠内で実行を保証するエンドツーエンドの QoS が提供可能という条件であれば、リモート資源でソフトウェアを実行できる能力には意義がある。たとえば、エンドツーエンドの QoS を入出力データ転送と実行時間に組み合わせ、ユーザの全 QoS 要件を満たすようにこの QoS ベースのワークフローの実行を保証する場合などが考えられる。

可用性契約：他の多くの科学コミュニティと同様、核融合コミュニティの研究の多くは、主要な会議での研究成果の発表に間に合わせる必要に迫られている。この点において、今のところ導入されている資源が不足しているようには見えない。しかし、こうしたイベントの前には、利用が急増して資源が一部のユーザの要求を満たせないところまで逼迫する

ものと予想される。この問題は、前もってサービスの可用性契約を結んでおき、需要が高まった場合にはそれに見合う可用性を要求するという仕組みを利用することで解決できるものとする。

利用ポリシー：上に述べたクライアントニーズには、いずれも利用ポリシーの詳細と、仮想組織だけでなくサービス/資源プロバイダ側で実施するための仕組みを必要とする。たとえばサービスプロバイダは、特定のソフトウェアを実行する権利を有するグループやユーザ、使用可能な資源、可用性に関する契約、サービスの実行管理などを明らかにしなければならない。また、このような利用ポリシーは、基本となる資源管理システムで適切に行使する必要がある。

柔軟な権限代理：クライアントのプログラムをシームレスにメンテナンスするには、サーバに対する柔軟な権限と代理ポリシーが必要である。たとえば、実行中に予期しない障害が発生した場合、サービスプロバイダはプログラムの診断やデバッグ、再起動が必要になることもある。プログラムはセキュアなデータベースにアクセスする場合もあるため、こうしたアクションを採るには、サービスプロバイダはこの利用パターンを再現できる権限を取得しなければならない。しかしクライアントと同等の役割を担うことは必ずしも合理的な選択肢とは言えない。サービスプロバイダがあまりに多くの権限を得ることになるため、クライアントは同意に消極的になりがちである。

コミュニティによる許可：クライアントは、個々のサービスプロバイダよりも、コミュニティによって許可されたコミュニティサービスを利用したいと考える。たとえば、ハードウェア資源（クライアントが認知していない場合もある）上でコードを実行する場合、その資源のアカウントを取得しなくても利用できるようにしなければならない。その代わりにクライアントがコミュニティによる認証情報さえ提示すれば、プログラムを始動できるような仕組みを設ける必要がある。

5.3 シナリオ

上に述べた実験的シナリオの場合、NFC サイト（クライアントのサイト）の 1 つに属する科学者は、実験中、T 時間（一般には約 10 分）以内に別の NFC サイト（サービスプ

ロバイダのサイト) にリモートで実行コードインストールし、メンテナンスする必要がある。非常に簡便な実行の場合、サービスプロバイダのサイトでは以下の内容を利用することができる。準備できたアプリケーション入力用の実験データをダウンロードするためのスクリプト、アプリケーションの適切な「短期」構成、T 時間未満の実行能力(いくつかのアプリケーションは精度/時間トレードオフを反映する複数の設定で利用可能)、クライアントに結果を配布するスクリプト、これらのアクションの順序や QoS との関係を記述した実行計画またはワークフローなど。要求されている QoS (このケースでは T 時間以内) で確実にコードを実行するには、クライアントサイトの科学者はアプリケーションサーバと契約する。この結果、実験を行っている間(通常は一日)は必要に応じていつでも T 時間以内でのコードの実行が保証される。当日中に要求される実行はごくわずかであり、サービスプロバイダ資源を他のクライアントと共有する必要があることから、資源配分は過剰供給に陥ることなく、他のソフトウェアが時間重視のアプリケーションと資源を共有して、状況に応じていつでもプリエンプトできるようにすることが大切である。

クライアントが契約に基づく実行を要求すると、サービスプロバイダはプログラムを起動してこれを監視し、必要に応じて特定のアクションの障害を選択的に回復する。また、プログラムの重要度に応じて、サービスプロバイダはこの供給を増やしたり反復実行することもある。

このシナリオは、問題となっているサービスによってはいっそう複雑になる。実行時間またはクライアントが体験する他の QoS 面はエンドツーエンドであることが必要であり、言い換えれば、サービスプロバイダはアプリケーションの実行だけでなく、データベースへのアクセスやデータの転送その他のアクティビティの責任も負う。転送時間(複製)前のデータの可用性は動的になるため、このケースでは活用できないという点に注意することが大切である。

同様に、国内的(場合によっては国際的)には、導入データの転送は重要な要素になると考えるが、現状では管理の確かさを保証することはできない。また、QoS ベースの実行は、大規模な施設の大規模な核融合研究所だけでなく、小規模な施設の小規模な核融合研究所で利用できることが重要である。

時間は別として、核融合コードは時間を重視しない実行モードを選択できる代わりに結果の精度が要求されたり、一定の時間量よりは、ある期限までに完成させることを条件に時間要件を緩和することもある。シナリオの詳細については [参考資料 : 3] を参照すること。

5.4 関連資源

主要な関連資源は以下のとおりである。

1. サービスプロバイダサイトにおけるハードウェア資源。スーパーコンピュータから単一のワークステーションまでさまざまである。
2. クライアントサイトで稼動するマシン。
3. 核融合サイト間のネットワーク（サービスプロバイダサイトとクライアントサイト）。これらは広く分散しており、場合によっては世界的に分散することもある。

提供されるサービスは、主にサービスの実行に関するものであり、将来的には実験的ハードウェアサービス（実験装置など）に関わる可能性がある。

5.5 OGSA プラットフォームの機能要件

この活用事例では、[1] に述べたように以下の OGSA 機能を使用している。

1. 発見 クライアントは、利用するネットワークサービスを前もって探しておく必要がある。サービスブローカは利用可能なハードウェアとソフトウェアを特定しなければならない。
2. ワークフロー管理 核融合グリッドネットワークサービスは、複数のコンポーネントによるワークフローである（リモート実行、入出力データの転送など）。
3. サービスタスクのスケジューリング サービスプロバイダ（またはその代行業務を行うブローカ）は、クライアントが要求する実行上の制約要件を満たせるように資源をスケ

ジューリングしなければならない。スケジューリングは事前予約の形態を取る場合もある。

4. 災害復旧 サービスプロバイダ（またはその代行業務を行うブローカ）は、クライアントのエンドツーエンドの制約要件を満たせるように努めなければならないため、障害の回避に役立つ適応力が必要な場合もある。

5. ブローカリング サービスブローカは、クライアントが要求する実行に適したソフトウェアとプラットフォームを特定する。

6. 負荷分散 サービスプロバイダ資源をより効率的に使用するには、負荷分散が必要となることもある。

7. フォールトトレランス 時間重視の実行機能を提供するには、信頼性の高いソリューションが必要である。

8. 移送管理 信頼性の高い移送管理は、このアプリケーションで必要なエンドツーエンドの QoS 獲得に不可欠である。

9. レガシアプリケーション管理 レガシの問題に対処するグリッドの可能性を実現することは、このプロジェクト最大の動機であった。

10. ブローカリングを容易にするサービス この機能は、サービスブローカが要求された制約条件を満たすワークフローを構築し、実行する上で、不可欠なものである。

11. アプリケーションおよびネットワークレベルのファイアウォール これは核融合の事例の中では長期化している問題である。とくに多種多様な異なるポリシーの取り扱いや国際サイトでのとくに厳しい制限が、問題をいっそう困難にしている。

12. 契約ベースの対話 このプロジェクトでは、クライアントとサービスプロバイダ（必ずしも人的なものとは限らない）間で契約を取り決め、この契約を上位のエンドユーザ構

造に組み込むことのできる契約ベースの対話力が必要である。

13. 許可と利用ポリシー 上に述べたように利用ポリシーの内容決定と実行するための仕組みを必要とする。

5.6 OGSA プラットフォームサービスの利用

以下のサービスは、前節で述べた機能を提供するために必要なものである。

1. 名前解決および発見サービス
2. セキュリティサービス
3. プロビジョニングと資源管理サービス
4. メータリングとアカウントティングサービス
5. ポリシーサービス
6. メッセージングとロギング
7. モニタリングサービス
8. メータリングとアカウントティング
9. 管理
10. サービスの統合

5.7 セキュリティ要件

サーバサイトでは、資源の利用についてだけでなく、特定のソフトウェア（またはアプリケーションサービス）の利用に関する許可を付与する能力が求められる。VO 固有の許可ポリシーは中央で、資源固有のポリシーは資源オーナーが管理する必要がある。

また、アプリケーションサービスプロバイダは不正な動作をするアプリケーションのデバッグに必要なユーザの権限のサブセットを受け持つ能力が求められる。これは、アプリケーションが、このプログラムを開始したユーザの権限に基づいて実験データベースにアクセスするためである。アプリケーションプロバイダは、ユーザに対して透過的にそのプログラムをデバッグし、再送信できることも少なくはない。

5.8 パフォーマンス要件

サービスをリアルタイムで提供できることも必須条件である。もう一つ重要なことは、他の QoS 要件（精度に関するアプリケーション固有の概念）を満たす能力である。

5.9 活用事例の状況分析

必要な機能のいくつかは、すでに Globus によって提供されている。これはコミュニティが核融合サービスを導入し、順調に運用しているという事実からも明らかである。現在、実施課題や契約ベースの対話、スケジューリング、これらをサポートする適応技術に関する問題の調査が進められている。また、セキュリティモデルの変更や、ファイアウォールのような導入問題の解決を進展させることも必要である。

5.10 参考資料

1. Foster, I. and Gannon, D. 「オーブングリッドサービスアーキテクチャプラットフォーム」

www.unix.gridforum.org/mail_archive/ogsa-wg/doc00016.doc

2. Keahey, K., Fredian, T., Peng, Q., Schissel, D.P., Thompson, M., Foster, I., Greenwald, M. and McCune, D. 「Computational Grids in Action : the National Fusion Collaboratory」

『Future Generation Computing Systems』（近日発行）,18（8）. 1005-1015.

3. Keahey, K. and Motawi, K. 『Taming of the Grid : Virtual Application Services』、Argonne National Laboratory、Mathematics and Computer Science Division Technical Memorandum ANL/MCS-TM-262、（2003）

6 OGSA および OGSA-DAI によるサービスベースの分散クエリプロセッシング

6.1 概要

サービスベースの分散クエリプロセッサは、1 つまたは複数の既存のサービスを通じて宣言型言語で表されたクエリの評価をサポートする。これらのサービスは OGSA-DAI プロジェクト（www.ogsa-dai.org）で提供されるようなデータベースサービスを含むものと考えられるが、他のコンピューティングサービスを含む可能性もある。したがってサービスベースの分散クエリプロセッサは、サービスの統合をサポートし、ワークフロー言語の

ようなサービス統合向けインフラに対しては補完的なものと見なすことができる。グリッドの設定では、分散クエリプロセッシングは、要求の予想資源要件に基づいてオンデマンドで計算資源を発見および使用する機能から、利点を享受できる。

グリッド上の分散クエリプロセッサは、これ自体をサービスと見なすことができ、ここではグリッド分散クエリサービス (GDQS) と呼ぶことにする。

原則として GDQS は、構造化データコレクションの統合や分析を行うあらゆるグリッドアプリケーションに適用することができる。GDQS が関わる典型的な活用事例には、アプリケーションの領域に関係なく、いくつかの基本的なフェーズが存在する。また、中にはユーザに対して透過的なフェーズもあれば、ユーザとの対話を必要とするフェーズもある。しかしいずれも、グリッドのソフトウェアインフラの特定の要件を含んでいる。次節では概要を述べ、それぞれのフェーズについては 1.3 節で詳しく説明する。

ファクトリの発見およびサービスインスタンス作成フェーズ。ユーザは GDSR (Grid Data Service Registry) に問い合わせることで、GDQS ファクトリを発見しなくてはならない。適切なレジストリの知識や合理的な検索条件を使えるかどうかはユーザ次第である。ファクトリが見つかったら、インスタンスが作成される。

資源発見フェーズ。GDQS は、クエリプランを最適かつ効率的にスケジューリングできるように、利用可能なグリッドノードの計算機能に関するメタデータを取得する必要がある。このフェーズはユーザに対して透過的である。

GDQS セットアップフェーズ。ユーザは、複数のデータソースや分析サービスにアクセスするために、GDQS インスタンスを準備しなければならない。これにはファクトリハンドルや統合されるデータソースをまとめる OGSA-DAI サービス向けの適切な構成ドキュメントの提供、分析に使用するサービスの WSDL URL の提供などがある。GDQS は、この情報を使用してデータソースやサービスの WSDL コンテンツのデータベーススキーマをインポートすることにより、送信されたクエリを処理 (コンパイルと最適化) できるようにする。

クエリ（要求）送信フェーズ。ユーザは OQL（Object Query Language）でクエリを定式化し、これを GDQS に送信する。

クエリ実行および結果配布フェーズ。クエリが送信されると、GDQS は、資源発見および GDQS セットアップフェーズで収集した情報を考慮しながら、利用可能なグリッドの計算資源を駆使し、クエリのコンパイル、最適化、スケジューリング、実行へと進む。次にこの結果は、OGSA-DAI グリッドデータサービス（GDS）ポートタイプインタラクションセマンティクス [OGSA-DAI を参照] で許可された対話パターンでユーザのテーマに配布される。

6.2 顧客

SB-DQP では、商用または科学的な背景のあるユーザが対象として考えられる。この利用パターンには基本的な特徴として、分散および異種の資源のデータを、サービスとして提供される分析機能に統合したいというニーズがある。たとえば、分散クエリプロセッシングは、多くの分散構造データストアが存在するバイオインフォマティクスに相応しい技術と考えられており、個々の分析でこうしたいくつかのストアや分析ツールにアクセスしなければならない場合も多々ある。バイオインフォマティクスには、重要な（著しい可変容量の）構造化データストアと、こうしたストアから抽出可能なデータに適用できる多くの分析ツールが、数百存在する。現在多くのバイオインフォマティシャンが一連の未接続（あるいは大半は手動で接続した）のアクティビティを適用することで、データと分析の統合をはかっている。こうした未接続のアクティビティを最適な方法で結ぶ標準のクエリ言語を使用する宣言型インタフェースは、バイオインフォマティクスのコミュニティではとくに注目されている。

バイオインフォマティクスに向けた GDQS の潜在的価値を例証する詳細なシナリオは、1.3 節で説明する。このシナリオでは、2 つの分散データ資源、GO（Gene Ontology）データベースと GIMS（Genome Information Management System）のデータを、BLAST という分析ツールに統合する。

6.3 シナリオ

以下の OQL クエリは、バイオインフォマティクによる GDQS との対話および 6.1 節で紹介するフェーズの通過につなげる方法を例証するシナリオを構築するための出発点となるものである。最初にクエリを説明し、次に以下を例証するシナリオを紹介する。

p. たんぱく質 ID の選択、たんぱく質の p、たんぱく質 Term の t から (p. 配列) プラス t、ここでは t. 用語 ID=「GO:0008372」および p. たんぱく質 ID=t. たんぱく質 ID

このクエリは、GO 用語の「GO:0008372」 (= 未知の細胞要素) の注釈を付けたそれぞれのたんぱく質について、それと類似したこれらのたんぱく質を返す。たんぱく質とたんぱく質 Term extent が 2 つのデータベースからそれぞれ取得されるものとする ([21] のように)。GIMS (Genome Information Management System) [img.cs.man.ac.uk/gims] と GO (Gene Ontology) [www.geneontology.org] はそれぞれ MySQL リレーショナルデータベース管理システムで (個別に) 稼動する。クエリはまた、BLAST 配列類似プログラム [www.ncbi.nlm.nih.gov/BLAST/] を呼び出す。これはたんぱく質の配列が与えられており、たんぱく質 ID と同様のスコアを含む一連の構造を返す。クエリは基本的に選択プロジェクト結合クエリだが、2 つのリレーショナルデータベースからデータを取得し、結合結果で外部のアプリケーションを呼び出す。分散環境を通じたこのクエリの処理に対するサービスベースのアプローチにより、オプティマイザーは複数のプロバイダから選択する (ほとんどの異種性は均一なインタフェースの後ろにカプセル化されるという意味において) ことができ、複数のオペレータのコピーを生成して類似性を活用することができる。たとえばサンプルのクエリでは、オプティマイザーは異なる GO および GIMS データベース、異なる BLAST サービス、クエリサブプランを評価する異なるノードの中から選択することができる。

6.3.1 サービス発見とインスタンスの作成

図 1 は、最初のフェーズにおける対話を表している。図の最初の対話は、GDQS ファクトリが初期設定の一環として自身を GDSRegistry に登録している事実を表す。クライアントは GridService::FindServiceData オペレーションを使ってレジストリに問い合わせを行い、適切な GDQS ファクトリ (GDSF) (対話 2) を発見する。次にクライアントは、

OGSA ファクトリポートタイプを使って GDQS のインスタンスを作成する。

(オリジナル図版参照)

図 1 サービス発見および GDQS インスタンスの作成

6.3.2 GDQS インスタンスのセットアップ

GDQS にとって、統合するデータソースのデータベーススキーマ情報の収集は不可欠である。図 2 は、GDQS がこの情報を取得するセットアップフェーズにおける対話を表したものである。クライアントは特定のデータソース（対話 2）の GDS ファクトリを発見し、このファクトリのハンドル（GSH : GDSF）を、ファクトリへの問い合わせ（対話 3）で取得した構成スクリプトとともに、インポートスキーマコール（対話 4）を介して GDQS インスタンスに渡す。また、作成される GDS のタイプを決定する構成ドキュメントを提供する必要もある。クライアントは、GDS ファクトリに問い合わせでサポートされている構成セットを調べ、最も利便性の高いものを選択する。次に GDQS インスタンスは、ファクトリハンドルとクライアントから提供された構成ドキュメントを使って GDS インスタンス（GDS1）を作成し（対話 5）、その GDS でまとめられたデータソースのデータベーススキーマを取得する（対話 6）。

(オリジナル図版参照)

図 2 データソースのスキーマ情報のインポート

6.3.3 計算資源メタデータの収集

また GDQS では、オブティマイザーがプラン各部の配分をできる限り効率的にスケジューリングできるように、利用可能なグリッドの計算資源に関する十分なデータを収集することも重要である。

現在の OGSA 参照実装はこのニーズを十分に満たしてはいないが、高水準のインデックスサービスを提供することで、計算資源メタデータの収集、キャッシュ、アグリゲーションを可能にしている。図 3 は、GDQS でグリッドの複数のノードから資源メタデータを収集するサービスベースのアーキテクチャを表したものである。このセットアップでは、インデックスサービスは、このサービスがバックエンドの情報プロバイダを使って導入されているシステムで動的な情報を収集する。GDQS は中央のインデックスサービスを、

メタデータのキャッシュおよびアグリゲーション用のサーバと見なし、他の分散インデックスサービスを(2)購読させる。リモートインデックスサービスは指定された間隔で通知メッセージを(3)送信する。このペイロードは、バックエンドの情報プロバイダによって決定されたフォーマットによる資源メタデータである。GDQS は(4) findServiceData コールを使い、そのサーバからの SDE として集約された情報を取得することができる。

インデックスサービスの階層は、仮想組織のインフラの一環として構築されていると考えられるかもしれない。組織の資源プールを構成するグリッドのノードを特定することは、GDQS の運用範囲を超えているためである。

(オリジナル図版参照)

図 3 計算資源メタデータの取得

6.3.4 クエリ(要求)の送信

このフェーズにおける対話(初期のクエリ送信は別として)のほとんどは、ユーザに対して透過的なサービス間の対話となっている。図 4 は、これらの対話を表したものである。関連するデータソースのスキーマをインポートした後、クライアントは GDS ポートタイプを通じてパフォーマンスコールを使い、(1)クエリを送信することができる。なお、クエリ送信のフォーマットと意味は OGSA-DAI フレームワークに準拠している。送信されたクエリはコンパイルされ、分散クエリ実行計画の中に最適化される。次に GDQS は、クエリオプティマイザーが作成する個々のクエリサブプラン(またはパーティション)をグリッド上の異なるノードで実行するための、一連の GQES (Grid Query Evaluator Services)を作成する。GQES インスタンスのスケジューリングはまた、収集したメタデータに基づいて最適な方法で計画される。GQES インスタンスが指定した実行ノード(これらはグリッドのあらゆる場所に存在し得る)上に作成されると、GDQS はそこに割り当てられた各プランのパーティションに(2)渡す。これにより DQP フレームワークは、均一なサービスベースのインタフェースがこの達成に必要な下位の複雑性の大半を隠している場合でも、(默示的に)並行評価の利点を楽しむことができる。最後に、GQES インスタンス(のいくつか)は他の GDS インスタンスと(3)対話してデータを取得し、この結果、GQES インスタンスに(4)通知を開始し、最後に GDT ポートタイプを経由

してクライアントに戻る。

(オリジナル図版参照)

図 4 : クエリの実行 概要

6.3.5 クエリの実行と結果の配布

6.3 節の最初に述べているサンプルクエリでは、クエリ送信によって図 5 のようなグリッドサービス (GS) 対話ダイアグラムが発生する。ストア、すなわち N1 と N2 をスキャンする GQES は異なるホストで具体化される。N2 の条件 (利用可能なメモリなど) は、GDQS がハッシュ結合を N2 に割り当てたことを証明するものとなっている。BLAST オペレーションコールの場合、GDQS は、2 つの GQES N3 と N4 を通じて並行化できる利点を得られた。GDQS は (1) 要求を受理するとこれを図 7 (d) の分散クエリ計画にコンパイルし、その各パーティションは 1 つまたは複数の実行ノードに割り当てられる。各実行ノードは、GDQS (2) で作成された GQES インスタンスに対応する。次に GDQS は、各プランパーティションを XML ドキュメントとして指定された GQES インスタンスに (3) ディスパッチする。それぞれの GQES インスタンスは、そのプランパーティションを受け取ると、評価を開始する。クエリの実行は反復モデルを使用したデータフロー計算であり、各オペレータは `fopen ()`、`next ()`、`close ()` インタフェースを実行する。パーティションを実行する GQES インスタンスのデータフローには、セマンティクスがストアにアクセスする必要のあるオペレータが含まれている。

それぞれの GQES インスタンス内では、`open ()` コールが `topmost` オペレータに到達すると、初期化手続きが開始される。このコールは、リーフオペレータに達するまで、すべてのレベルで親から子へとオペレータツリーを伝わる。次に他の GDS との対話が発生する。各 GDS のハンドルは GDQS によって、それを必要とする各 GQES インスタンスに渡された XML ドキュメントに埋め込まれる。たとえば、ノード N2 (図 5) では、`open ()` コールのストリームが一連のスキャンオペレータに到達すると、N2 GQES は N2 上の GDS インスタンスと対話を開始する。これによってデータは、GIMS データベースにアクセスする GDS のたんぱく質範囲から上流に流れる準備ができる。

(オリジナル図版参照)

図 5 : クエリの実行および結果の配布 詳細

データストアにおける破壊的な異種性の形態の多くは、標準の GDS インタフェースでカプセル化される。このため SB-DQP は、グリッドメタファーが具体化、すなわちクエリ評価が異種のデータと計算資源を通じて実行される能力を活かすが、異種性はユニバーサル GS インタフェースの後ろにカプセル化され、一貫性のある均質なサービス間の対話セマンティクスを生起する。

6.4 関連資源

GDQS では次のような計算資源の利用が期待される。(i) クエリ評価者サービスの実行、この中のいくつかは一つのクエリの評価で連携する場合もある。(ii) データを主要なソースから分析ツールまたはクエリのデータの結合や操作を行う評価者に移行。(iii) パフォーマンスまたは信頼性の中間結果を保持。こうした計算サービスは、いずれも複雑な要求の特定のニーズをサポートするため動的に区別し、割り当てられなければならない。

GDQS で使用するサービスの場合、次のような内容を含むものと考えられる。(i) サービスレジストリ。クエリがそれらを通じて評価される前にサービスの説明を GDQS にインポートしなければならないため。(ii) 構造化データアクセスサービス。セットアップコストを削減するには構造化ストアへの一貫性のあるアクセスが重要であるため。(iii) 柔軟な移送サービス。たとえばデータのストリーミングをサポートし、複数のサイトに並行して配布するなど。

6.5 OGSA プラットフォームの機能要件

発見とブローカリング。SB-DQP では、利用可能な計算資源やグリッドデータサービス (GDS)、分析サービス (AS) を発見できることが極めて重要である。クエリが定式化されるデータソースのデータベーススキーマをインポートするには、GDS の発見が必要である。分析サービスの発見は、これらのオペレーションに必要な、あるいはサポートするオペレーションおよびデータのタイプを特定し、クエリに組み込めるようにする上で必要である。ここでの重要な要件は、SB-DQP クライアント (ユーザ) および DQP サービスの両方でこうしたサービスに関するメタデータを発見し解釈できる均一なモデルである

こと、そしてこれらを計算資源に関する情報（ホスティング環境や CPU の処理速度、メモリ容量などのマシン機能）に関連付けられるということである。

メータリングとアカウントティング。SB-DQP は、多くの GDS その他のグリッドや Web サービスを使用する可能性がある。これらは、分散クエリサービスの課金コスト全体に影響を与える要素を個々に持っている。SB-DQP は、他の関連するデータソースやサービスで導入しているメータリングやアカウントティング、課金方式に統合できなければならない。可能ならば、ユーザの選好に基づいて最も利便性の高い中から選択できることも重要である。これは、こうしたシームレスな統合がインフラレベルでサポートされている場合のみ可能である。

データの共有と管理。データの共有と管理は SBDQP の基本要件である。これは 2 つのレベルで実行される。下位では、データソースへのアクセスにグリッドデータサービスを利用する。上位では取得したデータをデータフローアーキテクチャの方針に沿うように処理（結合、圧縮、分析など）する。しかし現状では、これによってスキーマの統合や一貫性の問題が解決されることはない。SB-DQP は、こうしたデータ管理機能を通じて、セマンティックデータモデル統合やトランスペアレントデータキャッシング、コンシステンシー管理としての利点を享受する。

モニタリング。SB-DQP はいくつかのコンテキストで監視を必要とする。第一に、統合するサービスの進捗状況をモニタリングする必要がある。進捗情報は Evaluator サービス（GQES）、GDS、分析サービスから収集しなければならない。第二に、クエリの対話は（大量のデータまたはネットワーク条件により）長時間にわたる可能性があるため、SB-DQP は資源の再割り当てや Evaluator サービスの再スケジュールで対応する必要がある。次に、計算資源をモニタリングして動的な情報を収集し、どうすれば変化する条件に適応できるかについて決定を出す材料としなければならない。

複数のセキュリティインフラ。ほとんどのケースでは、分散クエリは、異なるセキュリティポリシーやインフラによってそのアクセスが制限されている可能性のある複数のデータ資源へのアクセスを必要とする。SB-DQP では、クライアントの代わりに透過的に複数

の資源へのアクセス許可を取得できるインフラに依存することが不可欠である。

資源利用の最適化。SB-DQP は、分散サービス（データおよび計算の両方。OQL は外部機能呼び出しをサポートするため）を通じて宣言型 OQL クエリの効率的な実行プランの作成を担うクエリオプティマイザ（Polar*システム）を使用する。これにより SB-DQP は、黙示の並行による宣言型要求のシステムサポート最適化を提供する。移送管理。SB-DQP がデータフロー計算としてクエリを実行する際には、効率的なデータ転送が極めて重要になる。SOAP/HTTP による XML データのみの送信は、データ重視のアプリケーションにとって特別に便利というわけではない。ここでは、サービス間の対話に利用できる効率の高いものをはじめとする複数の転送プロトコルを実装することが極めて望ましいと言える。

フォールトトレランスと災害復旧。大量のデータを返す可能性のある長時間実行型のクエリにとっては、フォールトトレランスが何よりも重要である。

6.6 OGSA プラットフォームのサービス利用

名前解決と発見。豊富なクエリがレジストリに維持されているメタデータに送信できるようにする使いやすいインタフェースを介してグリッドサービスを発見することは、SB-DQP の可用性にとって重要である。SB-DQP のセットアップでは、関連するデータソースのスキーマのインポートに、グリッドデータサービスファクトリの発見が必要である。

サービスドメイン。SB-DQP はサービスドメインの好例と見ることができる。これは、ライフタイムにわたって複数のグリッドクエリ Evaluator サービスとその他の Web サービスを調整し、特定のコンテキストに統合するためである。

メッセージングとイベント。SB-DQP でイベントの通知を受ける必要のあるコンテキストがいくつか存在する場合がある。関連するデータソースのスキーマが変化すると、DQP では、クエリが新しいデータベーススキーマに対して有効なものであるか、これらの変化を認識したい。もう一つのコンテキストは、モニタリングの進行である。クエリの実行が進行中、SBDQP はクエリの各評価ノードにおける実行状態を示す通知メッセージを受ける必要がある。また、計算資源の可用性と状態に関する定期的な更新情報を受け取り、必要に応じてクエリ評価を再スケジュール化できるようにする必要がある。

トランザクション。分散トランザクションは、今のところ SB-DQP ではサポートしているが、将来インフラで提供されるトランザクションインタフェースから利点を得ることは間違いはない。

サービスの統合。SB-DQP は 2 つの意味でサービスの統合フレームワークを実現する。内部アーキテクチャが分散クエリプランを構築および実行する方法という意味と、サービスとして利用可能なデータや分析資源を通じてクエリを実行できるという意味の 2 つである。後者のサービス統合の形態は、ワークフロー言語のような他のインフラに対して補完的なものと見なすことができる。

6.7 セキュリティ要件

GDQS が直面するセキュリティの課題の特徴は、設定によって異なる可能性があるが、極めて要求は厳しいと言える。たとえば、一つのクエリは管理の異なるドメイン内にあるサービスで実行し、管理の異なるドメインにあるノードで実行する評価者を割り当てることにより、利点を得ることができる。クエリで操作されるデータにはプライバシーの問題もある。たとえば、要求者が個人が所有するデータを組織から手放したくないものの、公共ソースのデータとは結合したいと考える場合もある。このように、一つの企業、複数の企業、そしてすべての参加者のシナリオがすべて可能である。

6.8 パフォーマンス要件

分散クエリのパフォーマンスには多くの側面がある。クエリは宣言型であるため、その実行は計画されなければならない。クエリの計画では、クエリに適したサービスを使用する際のコストに関する包括的な情報にアクセスする必要がある。また、クエリの評価に利用できる計算資源に関する情報も必要である。

クエリ計画の異なるオペレーションでは、異なる転送形態が好まれる場合がある。たとえば、多くの分散クエリプロセッサはパイプライン並行方式をサポートするが、いくつかのオペレーションはブロッキング方式を採る。つまりバルクデータ配布に適していることになる。クエリの部分評価にどのオペレータを使用すべきかは、利用可能な計算資源の能力と負荷による。並行方式はクエリ評価のパフォーマンス向上に利用されることが多いのだ

が、グリッドのようなオープン環境では明らかにスケジューリングは問題となる。

6.9 活用事例の状況分析

6.1 節で述べているように、活用事例のそれぞれのフェーズにはインフラにより必要とされるサービス/機能に関する密接な関係がある。以下のリストは、各フェーズのこうした要件と、これらが現行の OGSA 参照実装をどの程度まで満たしているか特定しようとする試みである。

サービスの発見とインスタンスの作成。ここでの第一要件は、クエリをサービスレジストリに送信することによって、データソースの GDQS ファクトリと GDS ファクトリを発見する能力である。ここではサービスレジストリが、登録されるサービスに関する潜在的に豊富な情報を特定し、発行できる能力、そして定評ある（使いやすい）クエリ言語でこの豊富な情報を問い合わせる能力をサポートしなければならない。

既存の OGSA 参照実装は、サービスの説明に対して問い合わせる能力を十分にサポートしているとは言えない。GS 仕様の最新のドラフトで提案されているサービスグループのアイデアは、この点においてより複雑なサポートを提供する。

GDQS インスタンスのセットアップ。ここでの重要な要件の 1 つは、グリッドデータサービスは、確立された方法でまとめるデータベースのスキーマを提供できることが要求されるということ。言い換えると、GDQS は GDS インスタンスに問い合わせで特定のデータソースのスキーマを取得できなければならないということ。

サービスデータ要素は、確立された方法でこうした情報を提供するための一候補であることは明らかである。今のところ、SDE を介してこの情報を問い合わせる方法はサポートされていない。GDQS は、OGSA-DAI フレームワークに対するカスタムエクステンションによってデータベーススキーマを取得する。ただし、ここで言及している要件は、OGSA よりも OGSA-DAI プロジェクトに、より直接的に関係している。

計算資源メタデータの収集。ここでの適切な OGSA サービスは、中核的な OGSI の一部ではないものの、上位のサービスとして提供されるインデックスサービスである。インデックスサービスはグリッド資源メタデータ収集のための柔軟なアプローチを提供するよう

に見えますが、未解決の問題もいくつか残されている。SB-DQP では、複数のクラスのメタデータが相互に関連付けられ、一貫性のある方法で提供される必要がある。メタデータのクラスで必要なものは以下のとおりである。

CPU の処理能力、メモリ容量、ディスクの空き容量といったグリッドノード（グリッドユーザコミュニティに計算資源を提供するマシン）の機能

一連のグリッドノード間のネットワーク接続における、通信負荷に関する動的（リアルタイム）な情報

提供するサービスという意味でのグリッドノードの特性。たとえば、特定のグリッドノードがグリッドデータサービスファクトリまたはグリッドクエリ評価ファクトリのどちらを提供するのかについての情報など。

今のところ、このようなクラスのメタデータの収集および関連付けを行う一貫した方法はない。

クエリ（要求）送信。インフラサービスの使用に関するクエリ送信の潜在的な重要性は、GDQS がサブクエリを実行するために任意の数のグリッドノード上に動的に GQES のインスタンスを作成しなければならないこと。今のところ、これはファクトリがその特定のノード上にすでに導入されている場合のみ、ノードにグリッドサービスインスタンスを作成できる。これはクエリオプティマイザーがグリッドノードの限定的なセット（GQES ファクターが存在する場所のみ）しか検討できないよう制限する。理想的には、ファクトリコードを動的にホスティング環境に送信し、これを導入することによって、GQES インスタンスのスケジューリングのためにすべてのグリッドノードを検討できるようにするのがよいだろう。

クエリの実行と結果の配布。ここでの第一要件は、効率的な転送方式に束ねる能力である。今のところこれは、SOAP/HTTP による XML のみがシームレスにサポートされている。Globus Grid FTP API にアクセスを提供する信頼性の高いファイル転送サービスは、シームレスにサービスインタフェースと統合するようには見えない。必要なことは、サービス間の対話レベルでの効率的なデータ転送を直接サポートすることである。

6.10 参考資料

1. M.N. Alpdemir, A. Mukherjee, N.W. Paton, P. Watson, A.A.A. Fernandes, J. Smith, T.Gounaris

「グリッド分散クエリサービス (GDQS) の設計」OGSA-DAI 設計ドキュメント (2002年12月2日)

2. J. Smith, A. Gounaris, P. Watson, N.W. Paton, A.A.A. Fernandes, R. Sakellariou

「グリッドの分散クエリプロセッシング」Proc. 3rd Int. Workshop on Grid Computing, J.Sterbenz, O.Takada, C.Tschudin, B.Plattner (eds.), Springer-Verlag, 279-290, (2002)

GWD-I (draft-ggf-ogsa-usecase-2.0)

7 グリッドワークフロー

7.1 概要

ワークフローは、既存のサービス (UNIX システムの Shell スクリプトなど) を結合することによって新しいサービスを構築する便利な方法として注目を浴びている。ワークフローの定義をワークフローエンジンに登録することにより、新しいグリッドサービスを構築、利用することが可能である。定義はワークフローエンジンによって解釈され、定義に指定されたとおりに他の複数のグリッドサービスを呼び出す。

7.2 顧客

ワークフローは、グリッドサービスのユーザとプロバイダの両者で使用される。ワークフローは以下のケースで使用される。

1. 単一サービスの結合 : ユーザ (またはサービスプロバイダ) は、複数の単一のサービス (実行時間が比較的短い) を結合することによって新しいグリッドサービスを構築する。たとえば株価情報サービスと為替相場情報サービスを結合することで、海外の株価情報サービスを構築することができる。

2. ジョブワークフロー : ユーザ (またはサービスプロバイダ) は、複数のジョブを組み

合わせ、入力や出力といった実行順序を指定する。ここで述べるジョブとは、科学的ジョブおよび商用ジョブの両方を指す。科学的ジョブの例としては、ワークフローによるシミュレーションサービスとビジュアライゼーションサービスの結合がある。（他にも化合シミュレーションやデータグリッドなど多くの例がある。）科学的ジョブワークフローでは、サービス間で大量のデータ転送を必要とする場合がある。商用ジョブの例としては、各支店の売上結果を並行的に集計し、これらを本社に集約する場合などが考えられる。

3. ビジネスプロセスの記述：サービスプロバイダは複数のサービスを結合してビジネスプロセスを記述する。たとえば、旅行代理店は航空券の予約サービスやホテルの予約サービス、乗り物の予約サービスを結合し、新しい旅行予約サービスを構築する。この種のワークフローは Web サービスの分野で入念に調査されている。ビジネスプロセスでは、完了に長期（1 ヶ月など）を要し、例外処理の仕組み（予約の取り消しなど）が必要な場合もある。

4. システム管理：サービスプロバイダは、ワークフローを使ってシステム管理のためのサービスを記述する。たとえば、システム管理ワークフローは、ファイル転送サービスを使ってアプリケーションリポジトリからアプリケーションプログラムを取得し、これをグリッドサービスコンテナに導入する。

上に述べた例を組み合わせることも可能である。たとえば、国内各所から気象情報を取得（上記例：1）し、この情報を使って気象シミュレーションジョブを実行した後、結果をビジュアライズ（上記例：2）することなどが考えられる。

また、すべてがグリッドサービスとして OGSA に抽出されます。したがって、グリッドサービスとして抽出されたものはすべて、ワークフローで扱うことが可能である。

ワークフローの定義自体はグリッドサービスとして見なす必要がある。このため、ワークフローはグリッドサービスの仕様が要求する各種のルールに準拠していなければならない。たとえばワークフローの定義では、グリッドサービスポートタイプに FindServiceData オペレーションを持っていること、通知をサポートすること、ファクトリでワークフロー

インスタンスを作成することなどが必要な場合もある。

7.3 シナリオ

上に述べたように、ワークフローはさまざまなケースで使用される。ここでは「アプリケーション導入シナリオ」について説明する。ここには他のサービス/機能間の典型的な関係が示されている。

7.3.1 アプリケーション導入シナリオ

このシナリオでは、グリッドシステムのシステム管理者またはユーザがアプリケーションをグリッドコンテナに導入（インストール）するものと仮定する。

プロセスはサービス統合エンジンによって実行される。サービス統合では、最初に共有ストレージとして実装されていると見られるアプリケーションリポジトリからアプリケーションプログラムを取得する。ストレージは発見サービスを通じて探すことも可能である。ストレージにデータのキャッシュ/レプリケーション機能が装備される場合には、プログラムコードを効率よく取得できる。

ストレージに接続する際は、プログラムへのアクセスを制限するために認証/許可を実行することが重要である。認証と許可では、プログラムの提供が許可されているかどうかを判定するためのセキュリティポリシーを適用するために、ポリシー管理サービスが必要となる場合がある。

プログラムは、取得した後に管理サービスの一部である導入サービスを通じてインストールされる。ここでも認証/許可の実行が必要である。事前に予約サービスを使って資源（グリッドコンテナ）を予約しなければならない場合もある。

これらのすべてのプロセスについては、ロギングサービスを使ってログを記録し、このログ情報をアカウントिंगのためにアカウントングサービスに渡すことも要件に入れるとよい。ロギングとアカウントングでは、これらのポリシーを適用するため、ここでもポリシー管理サービスが必要となることがある。

7.4 関連資源

ワークフローの記述を解釈および実行するには、計算資源が必要である。

長期持続型のワークフローを管理する場合、ファイルのような不揮発性メモリやデータベースを必要とする。

7.5 OGSA プラットフォームの機能要件

上に述べたシナリオでは、以下の機能を必要とする。

1. ワークフロー

この機能を用い、複数のサービスを結合してアプリケーションの導入を実現する。この機能は [参考資料 : 1] では、「フロー」と表現されている。

2. 発見

上のシナリオでは、導入するアプリケーションプログラムを含むストレージサービスを発見するために、サービス発見機能が必要である。この機能は [参考資料 : 1] では、「発見とブローカリング」と表現されている。

3. 共有ストレージ

上のシナリオでは、共有ストレージはアプリケーションリポジトリとして使用されている。この機能は [参考資料 : 1] では、「データ共有」と表現されている。

4. 認証と許可

アプリケーションプログラムの取得とこれらのグリッドシステムへの導入には、認証 / 許可を必要とする場合がある。この機能は [参考資料 : 1] では、「複数のセキュリティインフラ」および「境界セキュリティソリューション」と表現されている。

5. アプリケーションの導入

この機能は、アプリケーションをグリッドコンテナに導入するために必要である。

この機能は [参考資料 : 1] では、「管理」機能に含まれている。

6. 事前予約

この機能は、予約した資源上でアプリケーションを実行するために必要な場合がある。この機能は [参考資料 : 1] では、「プロビジョニング」機能と表現されている。

7. ログインとアカウントिंग

アプリケーションプログラムの取得 / 導入などのプロセスでは、ログを記録し、この情報をアカウントिंगに役立てることもできる。この機能は [参考資料 : 1] では、「メータリングとアカウントिंग」と表現されている。

8. ポリシー

認証、許可、メータリング、アカウントिंगにはポリシーを必要とする。

7.6 OGSA プラットフォームサービスの利用

1. サービス統合サービス

このサービスは「ワークフロー」機能に対応し、「ワークフローエンジン」として使用される。

2. 名前解決および発見サービス

このサービスは「発見」機能に対応する。

3. セキュリティサービス

このサービスは「認証と許可」機能に対応する。いくつかのケースでは、セキュリティはサービスとして実装されていないが、機能は各サービスに付随している。しかし、許可の判定のようなセキュリティ機能の一部はサービスとして実装されている場合もある。

4. データ管理サービス

このサービスは「共有ストレージ」機能に対応する。

5. 管理サービス

このサービスは「アプリケーションの導入」機能を含む。

6. プロビジョニングと資源管理サービス

このサービスは「事前予約」機能を含む。

7. メータリングとアカウンティング

このサービスは「ロギングとアカウンティング」機能に対応する。

8. ポリシーサービス

このサービスは「ポリシー」機能に対応する。

7.7 セキュリティ要件

未登録ユーザによるワークフロー定義へのアクセスを拒否しなければならない場合もある。この機能を実装するには、ファクトリを使用したワークフローインスタンスの作成時、およびワークフローインスタンスへのアクセス時に認証と許可を実行する。

また、ワークフローから呼び出されるサービスにも認証と許可が必要な場合がある。これをサポートするには、GSIのような代理方式が必要であるだろう。

7.8 パフォーマンス要件

ワークフローから呼び出されるサービスの実行時間が長くても、ワークフローエンジンのパフォーマンスはあまり問題にはならない。しかし、それが短時間である場合には、ワークフローエンジンのパフォーマンスが重要になることもある。

また、ワークフロー定義から呼び出されるサービス間で大量のデータ転送が必要な場合、ワークフローエンジンでデータを送受信することは、効率的ではない。

したがって、サービス間で直接データを転送する記述が必要となる場合もある [参考資料：7]。

7.9 活用事例の状況分析

Web サービスの分野では、数々の重要な研究が行われている。たとえば、IBM による WSFL [参考資料 : 2]、Microsoft による XLANG [参考資料 : 3]、両社共同による BPEL4WS [参考資料 : 4]、SUN による WSCI [参考資料 : 5]、HP による WSCL [参考資料 : 6] などがある。グリッドコンピューティングの分野では、ANL が GSFL [参考資料 : 7] を提案した。また、WFMC (ワークフロー管理連合) が長期にわたってこの分野に取り組んでいる。これらの重要な研究は、OGSA のワークフロー仕様の基盤となり得るものである。

7.10 参考資料

1. Foster, I and Gannon, D. 「オーブングリッドサービスアーキテクチャプラットフォーム」 (2003)

http://www-unix.gridforum.org/mail_archive/ogsa-wg/doc00016.doc

2. 「Web サービスフロー言語」 (WSFL 1.0)、2001 年 5 月、

<http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf>

3. 「ビジネスプロセス設計のための XLANG Web サービス」、(2001)

http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/default.htm

4. 「Web サービス向けビジネスプロセス実行言語、バージョン 1.0」、2002 年 7 月、

<http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>

5. 「Web Service Choreography Interface (WSCI) 1.0 仕様」 (2002)、

<http://www.sun.com/software/xml/developers/wsci/>

6. 「Web サービス対話言語 (WSCL) 1.0」、2002 年 3 月、

<http://www.w3.org/TR/wscl10/>

7. 「GSFL : グリッドサービス向けワークフローフレームワーク」、2002 年 7 月、

<http://wwwunix.globus.org/cog/projects/workflow/>

8 グリッド資源リセラー

8.1 概要

グリッド資源オーナーにとって、直接エンドユーザと接することは、必ずしも望ましいことではない。

資源オーナーとエンドユーザ間にサプライチェーンを介在させることで、資源オーナーは

自身の中核機能（大量のスーパーコンピュータのメンテナンスなど）に傾注し、コストのかかる多くの顧客への対応やサポートを回避することができる。またその一方で、資源の再販を手がける少数の大口顧客（おそらく一社のみ）との取引が可能となる。

エンドユーザは、リセラーによって魅力的なパッケージにバンドルされた資源を購入することができる（アグリゲーション）。これらの資源は、実際には複数の資源オーナーから供給される。

リセラーは、自らは資源を保有することなく、バンドルした計算資源を再販することによって収益を得るため、リスクを最小に抑えることができる。一般にリセラーは、上流のプロバイダとの関係を維持することにより、資源の供給を維持している。しかし、リセラーは、エンドユーザとの契約によるサービス水準を保証するには、一時的または永久にプロバイダを変更しなければならないと考えることもある。しかしリセラーにとっては、資源維持のことを心配するのではなく、対象市場へのバンドル資源の販売とともに、良質なカスタマーケアの提供に専念することも大切である。

この活用事例は、Jon MacLaren、William Lee 共著、GESA 活用事例ドキュメント [1] に掲載されている「Computational Reseller」を原案としたものである。

8.2 顧客

グリッド資源リセラーのシナリオでは、主要な 3 種類の参加者が登場する。いずれも何らかの形でグリッドサービスの顧客となっている。最初は「資源オーナー」で、このシナリオには複数存在する可能性がある（リセラーの立場から見て）。資源オーナーは、たとえばスーパーコンピュータのような高価で希少な資源を所有するものと想定されるが、ケースとしての必然性はない。これらのオーナーは、少数の大口顧客、すなわちリセラーを相手に資源をバルクで販売したいと考える。オーナーは手持ちのすべての資源を確実に販売することに関心があるが、顧客であるリセラーが懸念するような資源の実際の利用には関心がない。しかし、資源オーナーとリセラーの間にはサービス品質保証契約が交わされる。

次に中心人物の「資源リセラー」がいる。リセラーは（資源オーナーまたは川上のプロバイダの）顧客であり、（エンドユーザまたは川下のプロバイダの）プロバイダでもある。リセラーは資源利用のみに関心を持つ必要はなく、主要な関心事は収益の獲得、つまり、プリペイドの資源利用パッケージをすべての顧客に販売できるかどうかにある（携帯電話の「通話時間無制限」など）。また、資源が常に利用されているかどうかには関心がない。実際問題として供給過剰販売が考えられる。たとえば、全ユーザがプリペイド資源を同時に使用すれば、リセラーはトラブルに陥る。しかしこれは、極めて非現実的である。リセラーは資源の供給側と消費者の両者とサービス品質保証契約を結ぶ。リセラーはプロバイダよりも多く（かなりの規模）の消費者を獲得し、サプライチェーンが資源オーナーからエンドユーザに移行する自然なファンアウトを提供する。

最後に、資源リセラーの顧客である「エンドユーザ」がいる。彼らは資源の実際の消費者である。また、自分が使用している資源を誰が所有しているかは知らない。資源や関連するサービス、サポートはすべてリセラーから提供されるためである。エンドユーザは自分に適したリセラーを自由に選択できる。これはリセラーが提供するパッケージやその価格に基づいて判断する。

当然のことながら、資源オーナー、リセラー、エンドユーザの3者は異なる組織に属し、地理的にも散在している可能性がある。

以下に述べるシナリオでは、複数の資源オーナーと取引する単一のリセラー、そして多数のエンドユーザという仮定で説明する。しかし、このシナリオの要件を考える場合、（今日のインターネットプロバイダのような）リセラーチェーンの可能性を想定することも重要である。

8.3 シナリオ

この活用事例は極めて一般的であり、多くのシナリオが考えられる。また、事例は詳細部分を除いてすべて類似している。したがって一例のみ取り上げる。

8.3.1 化学計算向けリセラー

化学分野との強力なつながりと、スーパーコンピュータ上で動作する広範な化学向けアプリケーションをサポートする知識を持つリセラーの例を考える。リセラーは事業を成立させるため、スーパーコンピュータオーナーに、リセラーが資源を再販することを条件にバルクで資源を販売する提案を持ちかける。リセラーは再販する際の資源オーナーのポリシーを尊重して契約を交わす。ある資源オーナーはアカデミックユーザが使用することに限定したサイクルを提供し、別のオーナーは、非アカデミックユーザに販売するサイクルがより高額に設定されている 2 層の価格構造を提案する。両方の資源オーナーは、提供するサイクルを別のリセラーに販売しないことを条件として指定する。このため、リセラーはこのケースにおけるエンドユーザのみに販売することを決める。

リセラーは、スーパーコンピュータセンターの調達だけでなく、人気のあるすべての化学パッケージへのアクセスを提供したいと考える。場合によっては、リセラーは対象ソフトウェアのサブセットをインストールしている資源オーナーからライセンスをリースすることもできる。一方彼らは、これらのパッケージを直接メーカーからも調達するが、ターゲットマシン上のソフトウェアのステージング（またはインストール）のために調整しなければならない。

最後に、リセラーは宣伝キャンペーンを行ってユーザにそのサービスをアピールする。販売するのは、CPU サイクルや安全かつバックアップ装備のディスク記憶装置、ソフトウェアライセンスなどのプリペイド（「使用無制限」）アイテムを含む月額資源パッケージである。できる限り魅力的な商品にするために、リセラーは営業開始後 3 ヶ月間、「今回限りの」提案（目玉商品）として故意に赤字を出して資源を販売する。

ユーザの作業の実行を促進するために、リセラーは資源ブローカを提供する。ユーザが作業をブローカに送信すると、ブローカはユーザの選好と資源オーナーのポリシーをマッチングさせる。このマッチングと資源そのものの状態に関する情報に基づいて、ユーザのジョブがディスパッチされる。

リセラーがエンドユーザとのサービス品質保証契約に「違反」した場合、ユーザは補償資格を得ることも考えられる。この内容はサービス品質保証契約自体に明記されることもあ

る。

以下のように、参加者それぞれの立場によるこのシナリオの潜在的な利点をまとめておくと便利である。

1. 資源オーナー。スーパーコンピューティングセンターがそのサイクルをリセラーに販売したいと考える理由はたくさんある。

a) すべてのサイクルがこのように販売された場合、資源オーナーは多くの顧客と直接取引する必要がまったくない。高品質のカスタマーケアの維持にはコストがかかるため、これは有益な方法である。このポリシーにより、オーナーは少数の大口取引で資源を管理できる。

b) ローカルの利用率が低い期間中は、センターでは大量の別の冗長サイクルを一回限り販売したいと考える。

c) ローカルユーザの利用における季節的なピークと閑散期があるセンターでは、一定量のサイクル（月によって変化する）を販売して期待値に一致させることにより、利用の安定化をはかる。

2. 資源リセラー。リセラーは、ソフトウェアベンダから割引価格で取得できる（主に学研究に関する大口契約による）いくつかのライセンスを含め、さまざまな川上のプロバイダから調達した利用可能な資源をバンドルする。サンプル提案は次のとおりである。合理的な月額料金に関し、化学者は Cray T3E に関して 200「フリー」CPU 時間と、30 コースのガウス 98 スローインを取得する（これを超えると、当然ながら多額の料金請求が発生）。これらには、サービス停止によりジョブが配信されなかった場合の補償も含まれる（保険の一種）。市場のトレンドを熟知するリセラーは、将来の需要を予測し、価格の安い時期に事前に川上のベンダから資源を調達する。

3. エンドユーザ。化学者はリセラーから資源を調達したいと考える。バンドルされた資源を利用することで、すべてのパーティとの手動での取引コストを低減させるためである。また、より良好なカスタマーケアを期待し、川上のベンダが契約不履行となった場合でもリスクをリセラーと共有できると考える。最後にアカデミックユーザは、より安くバンド

ルを入手できる場合もある。バンドルを電気/携帯電話を取得した時と同じリセラーから取得するためである。これは、既存のマイクロトランザクション技術（テレコムやユーティリティなど）を有する企業によるリセラーへの参加を促進する。

8.4 関連資源

資源オーナーは資源を 1 つまたは複数の資源リセラーに販売する（GESAWG Computational Provider Scenario [1] も参照のこと）。

サプライチェーンの各資源リセラーは、1 つまたは複数の資源オーナーや川上の資源リセラーから資源を購入する。リセラーは、エンドユーザや川下の資源リセラーに資源を販売する前に、これらをバンドルする場合もある。

エンドユーザは資源リセラーからバンドル（の可能性のある）資源を購入する。

最終的にこれは、エンドユーザによって消費されるプロバイダから購入した資源なのである。

これはすべてグリッド資源である可能性がある。これらの資源は地理的に分散し、複数の資源オーナーに属していることも考えられる。

8.5 OGSA プラットフォームの機能要件

提示されているシナリオには多くの要件があるが、ここでは再販活動に固有の機能について説明し、作業のスケジューリングや他の活用事例で発生する実行のための包括的な要件は無視する。以下は OGSA プラットフォームドキュメント [参考資料 : 2] の 3.2 節に記載している再販に必要な見出しと機能である。

発見とブローカリング

シナリオでは、それぞれのリセラーがブローカを操作してユーザの作業を利用可能な資源にディスパッチする。ここでの最も重要な要件は、ブローカがユーザの選好と資源オーナーのポリシー間のある種のマッチングを達成できること（Condor ClassAd スキーム [参考資料 : 3] のようなものと考えられる）。可能性の評価リストを使用するブローカは、次に許容可能な所要時間やコストなどの情報をもとに、作業に特有の資源を選択する。

リセラーは資源オーナー（または川下のリセラー）を発見し、エンドユーザはリセラーを特定できなければならない。こうした実体のペアは、サービス品質保証契約を交わす必要がある。しかしこのシナリオの場合、これらは稀（一回限りの場合もある）にしか発生しない活動であり、ネットワーキングや広告といった既存の方式で達成されるものである。

メータリングとアカウントティング

シナリオにおけるアカウントティングや請求のモデルは、極めて複雑である。資源オーナーは大量のサイクルを1つまたは複数のリセラーに販売する。こうしたサイクルの価格については、2者間で交渉する。複数のリセラーで均一になる可能性はまずない。また資源オーナーは、サイクルが使用されるかどうかには、あまり関心がない。未使用のサイクルの部分的な返品については、2者間で取り決める。過剰な使用が発生した場合、資源オーナーはリセラーが使用できるサイクル量に制限しようとする。資源オーナーが超過使用を拒否するか、超過使用分により高額の料金を課すかは、ポリシーによって異なる。

リセラーは、経費やランニングコストの回収、収益の獲得に十分な資源パッケージの販売に最善を尽くさなくてはならない。ユーザにとっては、月間プランやオンラインなどの契約を、人手を介することなく行えるようにしなければならない。リセラーは利用ベースでエンドユーザに請求する必要があるが、これは OGSA プラットフォームの既存のプランでカバーされている。繰り返しになりますが、リセラーはプリペイドの資源利用に関する契約を通じて収益の大半を獲得する場合、収益最大化を図るために資源を過剰に販売することも可能である点に注意する必要がある（ホテルや航空券のオーバーブッキングのように）。資源オーナーと同様、リセラーの収益は資源の実際の利用度に基づく必要はない。

請求においては、異なる粒度の取引をサポートする必要がある。これは、購入注文/請求書やクレジットカードなどの各種支払いオプションを使用できるような仕組みを指す。

これまでは複数の異なる請求スキームを述べた。しかし OGSA プラットフォーム内では、上に述べたすべてのモデルを実現しなければならない。また、アカウントティングシステムでは（過少利用または過剰利用を含め）大多数の状況に対して自律的に動作できるように

する必要がある。GESA ワーキンググループ [参考資料 : 4] で設計されるシステムにはこれらを念頭に置いたケースが見られるが、この機能を、OGSA プラットフォームドキュメント [参考資料 : 2] (とくに 5.9 節を参照) で提案されている請求システムでどのようにカバーするかを理解するのは困難である。これらは主に、「アカウントスキーマ」をもとにした料金ベースの請求に焦点を当てており、再販の概念は考慮していない。

モニタリング

資源オーナーはさまざまなリセラーのクライアントによる利用状況を追跡し、資源の過剰利用を確認できなければならない。

ポリシー

エンドユーザと資源オーナーはリセラーと同様、複雑なポリシーに直面する可能性がある。リセラーは資源オーナーのポリシーに反するような方法で資源を販売することはできない。たとえば、産業ユーザにアカデミックユーザ価格でサイクルを販売することは許可されない。また、リセラーはユーザのポリシーに反する資源上でユーザの作業を実行することはできない。たとえば、「環境配慮型」ポリシーを有するユーザのジョブを、公害の責任を問われやすい企業が所有するコンピュータ上で実行することは許されない。

川上のすべてのプロバイダのポリシーを統合する何らかの方法があるに違いがない。

拡張型のサービス品質保証契約

これは OGSA プラットフォームの見出しではないが、このシナリオと他の GESA-WG 活用事例 [参考資料 : 1] にとっては必要なものである。コスト情報はパーティ間の SLA に組み込みたいと考える。ある状況では、SLA の補償レートについても定義できるようにしたいものである。たとえば、ユーザがある月に 24 時間またはそれ以上の間プリペイド資源にアクセスできない場合、2 ポンド返却される、などのケースが考えられる。これは GESA-WG グループ [参考資料 : 4] で現在進行している研究テーマである。

7.6 OGSA プラットフォームサービスの利用

以下のサービス (または適切ならばインタフェース) は、前節の機能を提供するために必

要なものである。

1. ポリシー [10]

ここに述べたシナリオは、ポリシー定義のための高度な要件を持ち、OGSA プラットフォーム内で扱うものである。とくに必要と考えられるのは、サプライチェーン内の複数のポリシーの統合である。

[10] [参考資料 : 2] のポリシーインタフェースの説明は、非常に漠然としていて明瞭ではない。

2. メータリングとアカウントिंग

このインタフェースで、本ドキュメントで述べているシナリオの要件に対処するのであれば、インタフェースをより柔軟にする必要がある。

3. プロビジョニングと資源管理

SLA 契約とモニタリングに必要です。この機能は前節で述べた拡張 SLA を処理できるようにする必要があります。

4. ブローカリング [11]

ブローカリング機能が必要である。このポリシーマッチングの側面はポリシーインタフェースで扱うことになると考えられる。

5. モニタリングサービス [12]

このサービスはモニター機能に使用する。

8.7 セキュリティ要件

資源オーナーとリセラーチェーンは、ユーザがプライバシーの保証を得られるようにしなければならず、この要件が必要とされる。

[11] この機能は OGSA プラットフォームインタフェースに追加すべきものである。

[12] この機能は OGSA プラットフォームサービスに追加すべきものである。

8.8 パフォーマンス要件

リセラーチェーンの各ステップに時間がかかる場合は、ユーザが資源にアクセスした際に良好なパフォーマンスを得られるようにする必要がある。

8.9 活用事例の状況分析

グリッドにはこの種の活用事例は他に見当たらないものと考えられる（アプリケーションサービスプロバイダは存在するが、これらもまた作業の処理に使用する計算資源を所有するため、リセラーとは見なされない）。他の領域に数百もの事例が存在することは確かだが、その大半はインターネットや携帯電話の供給に関するものである。対応可能な技術が実現すれば、リセラービジネスが確立されることは間違いありません。

8.10 参考資料

1. Keahey, K., MacLaren, J., Newhouse, S. (Eds.) 「GESA 活用事例」、2003 年 2 月

<http://www.doc.ic.ac.uk/~sjn5/GGF/draft-ggf-gesa-use-cases-01-7.pdf>

2. Foster, I. および Gannon, D. (Eds.) 「オーブングリッドサービスアーキテクチャプラットフォーム」(2003)

http://www-unix.gridforum.org/mail_archive/ogsa-wg/doc00016.doc

3. Raman, R., Livny, M. and Solomon, M. 「マッチメーカー：高スループットコンピューティング向け分散資源管理」、『高性能分散コンピューティングに関する第 7 回 IEEE 国際シンポジウム講演録』、1998 年 7 月 28～31 日、Chicago, IL.

4. 「グリッド経済サービスアーキテクチャワーキンググループ (GESAWG)」、

ホームページ：http://www.ggf.org/3_SRM/gesa.htm

9 インターグリッド

9.1 概要

この活用事例は、多くの点で商用データセンターと類似している（商用データセンターのすべての機能を必要とするため）。しかしこの活用事例ではさらに、地理的に分散した会

社が、（特定のデータセンターだけでなく）外部世界にアクセスする上で、能力をフルに発揮しながら稼動するグリッドに焦点を当てる。このため、このシナリオでは航空宇宙産業と金融業界の2つの垂直的市場を選択している。ただし、活用事例はテレコムや製造その他の垂直的産業にも適用は可能である。ここでは以下の機能に焦点を当てる。実際にはグリッド対応ではなく変更が困難な多くのアプリケーション、複合グリッドおよび非グリッドデータセンター、Web サービス標準によるコラボレーションや相互運用で合意した複数の企業間のグリッド、全体が一体化して機能する計算/サービス/データの組み合わせなど。ユーティリティコンピューティングの包括的な概念をグリッドの視野に入れることで、産業環境での運用を成功させる。また、移行や複合オペレーティング環境、メンテナンス要件といった産業の問題を考慮する。

9.2 顧客

垂直的産業（製造、テレコムなど）におけるグリッド。たとえば航空宇宙や自動車産業。ユーザベースでの主要な役割は、グリッド管理者、IT システムインテグレータ、事業活動マネージャなどが該当する（日常的なグリッドユーザもいるが、その要件は事業活動マネージャの要件に含まれる）。これらの役割は商用センターの活用事例で定義されている。

9.3 シナリオ

ここでは航空宇宙および金融業界を例に、活用事例を紹介する。

金融業界のオペレーティング環境は、データセンター（Linux や UNIX 系、メインフレームシステム、地理的に分散するストレージ管理システム）とユーザのマシン（PC やワークステーション）で構成される。グリッドは、Linux や UNIX 系と同様に、Windows XP ベースのサーバやメインフレームとすべての機能セットにおいてインタフェースを取る必要がある。

金融業界の場合、移行は極めて重要な要素である。多くの財務アプリケーションやシステムはレガシであるため、グリッドに対応せず、いずれの固有のモデルにも適応しない。業界では、アプリケーションやシステムの停止を最小限にとどめることのできる安全な移行パスが定義されない限り、グリッドをデータセンターのオプションとは見なさないかも知れない。また、非グリッドアプリケーションはグリッドアプリケーションによって標準

的なグリッドとしてアクセスできること、またそのパフォーマンスは低下させることはできないことが条件となる（パフォーマンスの向上は可能）。

一般に大規模な航空宇宙関連の企業は、多くのデータセンターを各所に所有している。またこの業界は、科学向けの高性能な産業用コンピューティングアプリケーションを、ビジネスアプリケーションとともに大量に常駐配備している。ビジネスアプリケーションや場合によってはコンピューティングアプリケーションの中にはレガシ系もあるため、変更が極めて困難である。これらはグリッド対応のシステムにインストールすることはできるが、いかなるグリッド固有のメリットも引き出すことはできない。このため、グリッドの複合環境と非グリッドネットワークおよびシステムが採用されることになる。しかしこの方法は、システムの地理的および仮想的な分離やユーザのアクセス管理等の問題をいっそう複雑にする。ここでは、アプリケーションはグリッドインフラを認識していないものの、依然として必要に応じて柔軟な資源を取得できるデータセンターの包括的なユーティリティコンピューティング要件を提起する。一方グリッド管理者にとっても、自身のシステムと非グリッド環境を共存させる完全な自由が得られる。

Web サービスを利用し、アクセスするアプリケーション（とくにビジネスサイドの）は多数存在する。また、グリッド環境に共存する完全な Web サービス環境もあるかもしれない。したがって、グリッドと Web サービスの共存に対するニーズがあり、競合する要件や環境は一種の妨げとなる。

一般に大規模な航空宇宙関連企業は、企業のファイアウォールの外側にあると考えられる設計センターを持っているが、共同設計には契約上の合意がある。このため、正確な実体ごとにシステムとアプリケーションの完全な管理の問題がある。アイデンティティ管理、さまざまなレベルのアクセスで IP や資源を保護したいと考える 2 社による完全な共同環境で、あらゆるレベルのセキュリティに関連する一般的な問題がある。グリッドのセキュリティとコラボレーション方式はこれを可能にしなければならない。あらゆる種類の状況はセキュリティの側面から発生する可能性があるため、ここからまたユーティリティコンピューティングの概念が明らかになる。したがって、IT ネットワークにおけるレイヤ 2 デバイスやファイアウォールその他の非コンピューティングデバイスを管理し、柔軟性を

持たせ、厳密なセキュリティ要件に一致させることが肝要になる。

航空宇宙関連企業では、あらゆる種類のグリッド環境が可能である。これには計算グリッドやデータグリッド、サービスグリッドその他名称ベースのグリッド系のスキームなどがある。

これらは共存しなければならない。以下にこれらの例のいくつかを取り上げ、異なる目的を持つグリッドがあることを紹介する。計算グリッド：

極めて集約的なコンピューティングが必要である。例：計算流体力学（CFD）：空気力学、「構造」：ストレステスト、衝突シミュレーション、バードストライクシミュレーション。

これらのいくつかは、並行化され、分散メモリメッセージパッシングアーキテクチャ上で十分に実行できる束縛問題である。これらは「ユーティリティコンピューティング向け」、すなわちオンデマンド容量機能は、仮想化や（データセンターにはない既存の PC やワークステーションを用いた）資源共有といったグリッドが提供する他の機能と同じように重要であるということである。

データグリッド：

例：翼構造シミュレーション分析は、1 テラバイトのメモリシングルシステム画像を必要とする。これは 64xcpu Superdomes（たとえばハイエンドの Hewlett-Packard システム）で稼動し、翼構造全体の圧力と曲げに関して 20 テラバイトのデータ（高圧縮）を生成するものと考えられる。したがってこのレベルのデータ処理機能は、データグリッドの立場から見れば極めて重要である。こうした主要なシミュレーションでは、1 テラバイトのメモリ要件 [13] を持つようなオペレーションをサポートできる OS やミドルウェアに変更し、大量のデータ冗長性と同期性を持たせる必要がある。さらにこれは、個々の顧客が実行する必要があるカスタマイズ向けのグリッドミドルウェアおよび API 要件にも当てはまる。

サービスグリッド：

十年にもわたる大規模プロジェクトを実施する代わりに、航空機システム全体をシミュレ

ーションできるようにすることは大きな挑戦である。多くの外注および設計コンサルティング会社（既存のプロジェクトのリスク共有パートナー数は 12 社など）との共同作業による物理的試験をできる限り廃止することは、財務的にも意義がある。このため、異なるコンピューティング活動（コンサルティング 1 社との試験や対話など）は、サービス契約によるサービスとして計画される。Web サービスやグリッドによる相互作用を通じて、コンピューティングサービスは複数のサイト/企業で実行される。また、サードパーティユーティリティプロバイダとの間で余剰計算サイクルを売買することも検討される（グリッド資源リセラーの活用事例）。解決すべき困難な問題は以下のとおりである。

セキュリティは協力する異なる企業間で保証

上に述べた異なるすべてのサービス要件を満たす資源の管理（構成、必要な資源の量の算定など）

信頼性を充足させ、すべての異なるレベル（資源レベルでサービスレベルに対して）の SLM 要件を満たす仕組みの保証

グリッド間の相互運用性は重要な要件である。したがってある種の相互運用基準を設ける必要がある。1 社が有する個々のデータセンターでは異なるグリッドミドルウェアを使用している可能性がある。

メンテナンスはもう一つの厳密なビジネス要件である。ユーザビリティやトラブルシューティング方式のようなものはこの点で関連性がある。すべてのシステムを停止させることなくグリッドソフトウェアをアップグレードできることも必要である。グリッドにアクセスする人は誰でも利用できるように、グリッドユーザインタフェースに関するいくつかのガイドラインが設けられるかも知れない。

9.4 関連資源

商用データセンターの活用事例を想定した資源ニーズでは、資源ニーズを基盤に据える必要がある。

また、グリッドの非計算資源を処理するニーズも重要視される。たとえば、ソフトウェアインタフェースを装備するデバイス（航空機システム-グリッドの概念が適用され、この資源を共有。とくに異なるチームによる試験やモデリング向け）やファイアウォール、ル

ータの構成などである。

これらの資源は企業内で地理的に分散している。これらの資源には複数のユーザが（さまざまなレベルのアクセス権で）アクセスする。

9.5 OGSA プラットフォームの機能要件

商用データセンターの活用事例がベースである。追加的要件は以下のとおりである。

発見とブローカリング。さまざまなレベルのセキュリティを考慮して組織間のサービスを発見およびブローカリングできること。

メータリングとアカウントティング：異種のストレージシステムへのアクセス、こうしたシステムに対するインタフェースフォーマットの選択、複数のアカウントやアカウントティングシステムの処理に必要な情報を含めたアカウントティング要件。

モニタリング。ライフサイクルマネジメントと障害対策を強調する資源および資産を対象とした、グローバルかつ組織横断的な表示。アクションの自動化とそれを満たす十分なデータが必要。複合グリッド/非グリッド環境に常駐するため、この情報をあらゆる場所で利用するには API が必要。

プロビジョニング。複合環境により、非グリッドプロビジョニングシステムとの互換性。解決手段はグリッドプロビジョニング API と対話するための API を指定すること。

資源衝突の解決。完全な混在環境では、資源利用と管理の同じ資源または概念（VO など）の衝突が考えられる。これは解決しなければならない（最低でもガイドラインを設ける必要がある）。

資源へのバッチおよび対話的アクセスを提供する利用モデル。

資源利用の管理と監視のサポート、および関連するすべてのパーティによる SLA 違反の検出。

負荷分散：負荷分散の他にセキュリティ要件を動的に検討することが追加要件となる。

レガシアプリケーション管理。レガシアプリケーションは変更不可能ではあるが、貴重なために廃止できないか複雑過ぎて書き換えできないものを指す。グリッドのインフラはこれらを中心に構築し、非グリッドユーザも利用できるようにする必要がある。

管理。環境の管理に用いた標準的な慣行を「体系化」および「自動化」することができること。この目的は、管理者によって指定された上位の構成や管理ポリシーに基づいて、システムが下位構成の詳細を管理するために自己編成および自己記述できるようにするこ

と。他に、非グリッド対応のサービスのグリッド対応サービスへの移行に関する重要な問題がある。グリッドの容易な可用性を確保するには、何らかのユーザビリティガイドラインも必要と考えらる。グリッドソフトウェアのアップグレードは、グリッド環境全体を停止させることは許されないが、システムごとに停止して、あるいは全く停止させることなくアップグレードできることが重要である。

プログラミングモデル：グリッド対応アプリケーションに対するガイドラインと手法（レガシアプリケーションを含め、結局最初の改訂で困難となる。たとえば、マルチスレッドまたはマルチプロセスのプログラムに適用した場合など）。

プログラムの実行：アプリケーションやネットワーク遅延の指定許容値。アプリケーションリアルタイム要件を取得するための優先付けスケジューリング（資源、メッセージなど）。標準的なジョブの記述（環境、ジョブなど）

ロギング：詳細なトラブルシューティングには十分なログ情報が必要です。管理者が必要とするログ情報のレベルはさまざまであると言える。

ポリシー（セキュリティやアイデンティティの観点から）

コラボレーション要件（資源やサービスへのアクセスを管理するさまざまなレベルのセキュリティ）

いくつかのユーザインタフェースが必要です。これらは、ユーザが送信、監視、プログラムの操作を行うために必要なインタフェースである。また、管理やパフォーマンス調整のための情報を提供するインタフェースを設けることで、計算処理の監査や資源活用的側面からの異なる実行プログラムの比較を可能にし、計算資源にインストールされたバージョンや個々の資源の実行状況、アカウント情報など、各プログラム実行についての情報を提供する。

9.6 OGSA プラットフォームサービスの利用

この活用事例はグリッドアプリケーションの広範な領域をカバーするため、ここではOGSA アーキテクチャ仕様のすべてのサービスを必要とする。

9.7 セキュリティ要件

最初に商用データセンターの活用事例要件をベースとする。これに加え、ソフトウェアとインタフェースを取るシステム、ルータ、ファイアウォールなどの非計算資源の利用、ア

クセス、フレキシングが必要です。完全なインターグリッド環境でのオペレーションを成功させるには、望ましい形で操作および配分する必要がある。また、セキュリティは、ここで管理者が検討するための主要な要素となる。

9.8 パフォーマンス要件

ここでも商用データセンターの要件をベースとする。科学向けアプリケーションが稼動するインターグリッドの一部では、実行速度の注意書きも検討すべき要素である。

総合的な性能要件から、CIO は目下の重要な問題を解決するために、企業の資源を素早く導入したいと考える。たとえば今日の金融業界の場合、中には手作業で 13 週以上にもわたるプロセスが必要な例もある。企業がデータセンターを整理統合すれば、データセンターの潜在的損失は膨らむ。重要な顧客対応サービスを別の施設に再配備できることも重要である。したがって移行と導入は、業界の品質のミッションクリティカル性と関連している。

システムのパフォーマンスを十分に管理するには、特定されたシステムについて、アプリケーションの実行前、および実行中のモニタリングと予測を行えることが条件となる。これらには、ネットワーク帯域幅やレイテンシ、ジッタ、CPU の負荷、情報サービス照会時間、ディスク容量、速度、マルチキャスト性能、リモートメモリおよびデータサイズ、アクセス時間、アプリケーションのタイミング、CPU の処理速度（仕様とベンチマーク）などがある。

これらの中には、既存のパフォーマンス管理ツールを使って収集できるものや、グリッドシステムや API ベースのツールで供給しなければならないもの（動的、自動調整を必要とするもの）もあります。

インターグリッドシナリオにおける現実のパフォーマンスおよび障害管理要件を実感するため、金融会社を例にとる。ここではサーバ 40,000 台とエージェントオフィスマシン 120,000 台を 19 のデータセンターに分散して使用し、今後数年間で 4 箇所に削減しようとしているものとする。人員を増やすことなく、継続的に導入される一連のサービスの複

雑性を管理できることがその動機となっている（移行およびパフォーマンス要件）。災害復旧では、一つのデータセンターを失っても重要なサービスを 24 時間以内に復旧できることが条件である（現在は 7～10 日）。信頼性要件は、年間に数分の停止にとどめなければならない。

9.9 活用事例の状況分析

これは明らかに調査段階にある。グリッド環境で稼動する産業用およびビジネスアプリケーションは未知の領域である。商用データセンターが成功すれば、全社規模で適切な産業仕様を持つグリッド（インターグリッド）が試行できる。

9.10 参考資料

1. Forrester 「Organic IT」レポート

10 インタラクティブグリッド

10.1 概要

グリッド技術に有用なアプリケーションとしては、バッチプロセッシングだけでなく、インタラクティブプロセッシングも考えることができる。今日一つのプラットフォーム上での動作に制限されているアプリケーションは、インタラクティブグリッドの概念を用いることで、ネットワーク/グリッドで動作させ、個人ユーザの対話を実現できる可能性がある。この場合、アプリケーションと生成プロセスはジョブパラメータの送り先と実行結果を認識していること（またはミドルウェアエージェントがそれらをルーティングする）、ユーザインタフェースはファームアウトまたは再開されたプロセスとシームレスに対話および同期できることが条件となる。オンラインメディアの活用事例と比べ、この活用事例では分散実行（顧客のシナリオによってはスレッドベースまたは手続きベースの場合も）の顕著な高粒度性が強調される。

インタラクティブグリッドのもう一つの側面は、自動スケジュールに基づく作業のスケジュールおよび実行機能、そして一連のアクションの自動化である。これは、ジョブが、人間かどうかは関係なく外部のエージェントによって動的に管理されることを意味する。

コンピューショナルステアリング [14] は、この結果から生じるインタラクティブグリッドのもう一つの側面である。これは、グリッドユーザが実行時にコンピュータや必要な資源を対話的、動的に操作できることを意味する。また、ユーザは標準的な対話的手続きを通じて、新しいグリッドロケーションにアクセスし、基盤となる物理資源の変更を動的に要求できるという意味でもある。

[14] コンピューショナルステアリングは、プログラムの状態の変更やデータ出力の管理、プログラム実行の開始と停止、資源配分の変更、基本的資源の変更などを定義に取り入れることが可能である。

動的ステアリングの場合、ユーザはプログラムや環境要件、システムの状態を監視し、変更を行えるようにしなければならない。

10.2 顧客

この活用事例は以下のタイプの顧客に適用できる。

アプリケーションがグリッドまたは非グリッド環境で動作しているかどうかを認識していない個人グリッドユーザ。アプリケーションベンダはアプリケーションを仮想化してグリッド環境の一部となるように特化した設計を行う必要がないため、将来のアプリケーションや利用シナリオの仮想化にとってとくに重要。

限定的な計算資源と、大規模ではないにしても一台の PC よりは多くの資源を必要とするアプリケーションを持つ中小企業顧客。

グラフィカルインタフェースを使用するグリッドユーザ（より複雑になるにつれて、すべてのユーザインタフェースはグラフィカルになる傾向があり、同期を必要とし、対話的使用が可能となる）。

現在特定のプラットフォームで動作しているレガシアプリケーションユーザ。ただし将来的には、グリッド対応のコードなしでネットワーク（グリッド）で動作できるようになる。

10.3 シナリオ

1. グリッドユーザが管理する UI ベースのオペレーション

2. 純粋な並行性（どの粒度に対しても）と普及コンピューティング [15]（バッチジョブだけではない）

これまで、科学およびアカデミックユーザは、バッチジョブで作業をグリッドコンピューティングシステムに送信していた。このため、バッチジョブからの結果を待つ間の遅延時間は許容できるものであった。しかし、グリッドコンピューティングの商業市場での役割が大きくなるにつれて、ユーザはリアルタイムで結果を監視および操作できることを期待するようになる。

現行のバッチモデルでは、それぞれの計算ジョブが計算資源のタスクと要件の命令を持つグリッド管理システムに送信される。グリッド管理システムは資源を割り当て、ジョブを完了すると、結果を返信する。ユーザは次のバッチジョブが送信されるまでは、中間結果を確認することも、変更を送信することもできない。新しいインタラクティブグリッドモデルでは、ユーザはグラフィカルな表示を通じてリアルタイムで結果が得られ、ジョブが処理中であってもその調整や操作、データ変更を行うことができる。

グリッドシステムのインタラクティブ技術は、実行時間の低減や広範な計算集約型アプリケーションの結果の改善に有用である。こうしたアプリケーションには、グラフィックスのビジュアライゼーションやレンダリング、CAD/MCAD などのエンジニアリングアプリケーション、デジタルコンテンツの作成、ストリーミングメディア、ビデオゲーム、テキスト編集および電子メールアプリケーションなどがある。また、インタラクティブグリッド技術によって可能となるリモートアクセスでは、高価な専用ソフトウェアに必要なライセンス数を制限することで、省コスト化を図ることができる。

並行化、プロセスレベル、スレッドレベル、さらには命令レベルの並行化は、インタラクティブグリッドの概念を導入することにより、グリッドでビジュアライズすることができる。

10.4 関連資源

この活用事例で必要なものは汎用の計算資源の利用のみである（CPU 資源およびこれらで動作する各種オペレーティングシステム、記憶装置資源）。

10.5 OGSA プラットフォームの機能要件

商用データセンターの活用事例は、ここでは、すべての機能の書き換えを回避するためのベースとなることを想定している。しかしインタラクティブグリッドの動的特性は、対話のきめ細かな移動性と仮想化の観点から、従来のサービスに新たな要件を提起する。これらの機能要件は以下のとおりである。

発見とブローカリング。これらの機能は、インタラクティブグリッド機能のサポートが可能なグリッド資源を認識できなければならない。ジョブ要件とユーザインプットに基づき、資源をリアルタイムで調整できることが条件である。

メータリングとアカウントティング。資源の利用状況を測定し、明らかにしなければならない。これは、実行の計算操作モデルでは、バッチ環境の予定モデルよりも複雑である。

モニタリング。モニタリングエージェント インタラクティブグリッドミドルウェアを通じて開始されるスタンドアロンのソフトウェアエージェントにより、SLA の実行状況などのセキュリティとパフォーマンスを監視。

データ共有。ユーザごと、ジョブの対話ごとの整合性を可能にし、管理するデータアーカイブとキャッシング。

ポリシー。従来は組織的プロセスとして実装、または手動で管理しなければならないポリシーの実行を自動化する上で、階層システムの複数のステージでポリシーを表現できることは重要である。

移送管理。データ転送や、他のデータ共有アプリケーションのサポートにおいて、帯域を動的にスケジューリングまたはプロビジョニングできることは重要なことである。

セッション管理。SLA や QoS 要件の実行を含むジョブのパフォーマンスを維持するものです。階層的セッションをサポートする。このセッションには、グローバルスコープや個別のレイヤスコープを持つもの、ポリシーで記述されるユニットコンピューテーションを追跡する内部のサブセッションなどがある。

10.6 OGSA プラットフォームサービスの利用

この活用事例は今日実装されていますが、必要となるすべてのサービスを完全には理解していない。しかし、これまでの経験に基づく初期のリストは以下のとおりである。

名前解決と発見サービス：インタラクティブグリッドに参加可能なデバイスを区別する

機能。

データ管理サービス：データキャッシングを考慮し、ユーザごと、ジョブごとの整合性を管理。また、データ転送のための帯域を動的にスケジューリングできる必要がある。

障害対応サービス：ユーザごと、ジョブごとの障害対応。

ポリシーサービス：階層システムの複数のステージにおけるポリシーとポリシー処理

10.7 セキュリティ要件

セキュリティ方式には、一連のセッションベースのアクションが必要です。階層的なセッションモデルがあるが、これらのセキュリティは不正なアクセスや悪用の防止のために調節できなければならない。したがってグローバルおよび個別のセッションについて許可管理を実行できるようにする。

10.8 パフォーマンス要件

バッチグリッドよりも極めて動的なスケールでユーザの要件に資源をマッチングさせることは、パフォーマンスの問題につながる。

10.9 活用事例の状況分析

この活用事例は現在調査段階にあります。Hewlett-Packard はこの初期のバージョン/デモを製作した。

10.10 参考資料

11 Grid Lite

11.1 概要

この活用事例は、グリッドの用途を PDA、携帯電話、ファイアウォールなどの小型機器に拡張するものである。主要な要件は、たとえば PDA がグリッド環境の一部となるために、この機器に不可欠な一連のグリッドサービスを特定することである。「Grid Lite」環境で動作するグリッドソフトウェアコンポーネントは、「標準的」な計算グリッドノードで要求されるものよりも小面積で、一般に、より効率的にする必要がある。Grid Lite インフラを導入すれば、他のグリッドアプリケーションやユーザは、これらのグリッド対

応の PDA その他の小型機器で作業を行ったり、その逆の運用も可能となる。

ファームウェアインタフェース（組み込みのオペレーティングシステム）をより多く搭載しているレイヤ 2、レイヤ 3 機器は、仮想化およびグリッドデバイス化することも必要である。ここでの主な要件は、仮想化と真のソフトウェアベースのリモート構成 / プロビジョニングである。

移動性は、コネクティビティ / 仮想化および同期化 / 対話性の点で多くの問題を抱えている。

11.2 顧客

この活用事例は以下の顧客に適用可能である。

PDA を使用する個人グリッドユーザ

小型機器またはネットワーク機器、その他プリンターのような非計算機器を製造するメーカー

予算に制約があり、自社環境を完全に仮想化したいが、ネットワークにファイアウォールやルータなどのレイヤ 2 機器が設置されているために仮想化を実現できない中小企業顧客

11.3 シナリオ

準備中です。

11.4 関連資源

この活用事例では、PDA や携帯電話、器具、ファイアウォールなどのレイヤ 2 機器、その他のネットワーク機器を必要とする。

11.5 OGSA プラットフォームの機能要件

この活用事例にとくに当てはまる要件で、他の活用事例では取り上げていないものは以下のとおりである。

発見とブローカリング。レイヤ 2 機器および移動性の機器に必要な発見およびレジストリ技術。大量の資源を処理する能力。

モニタリング。同期および非同期モデルを組み込むモニタリングモデル（オフライン処理とモバイル処理用）。

データ共有。オフラインおよびオンラインアクションの整合性に必要なデータアーカイブとキャッシングデータ機能および管理

プロキシグリッドクライアント方式：多くの機器では、グリッドクライアントの動作を受け付けられない可能性があります。これらのケースではプロキシ技術が必要である。

省スペースの必須サービスグループ：グリッドサービスのグルーピングでは省スペースの機器を考慮する必要がある。

11.6 OGSA プラットフォームサービスの利用

Grid Lite の予備的サービス要件は、以下のとおりです。詳細は Grid Lite インフラの実装にともない、明らかになる。

名前解決と発見サービス：レイヤ 2 機器や PDA、携帯電話のような移動性の機器の効率的なネーミングと発見。モバイル機器の場合、ローカル環境から遠ざかることによる発見の問題も重要である。

セキュリティサービス：発見と同じように、移動性は確保できるがセキュリティ機能を組み込めない機器や、VPN ベースのコネクションに関するセキュリティの問題も解決しなければなりません。ここではファイアウォールの再構成も問題となる。

スケジューリングサービス：移動性への対処

ブローカリングサービス：移動性機器の取り扱い

データ管理サービス：このサービスでは、通信で極低帯域幅を利用するケースも多いことを考慮する必要があります（コーディングスキーマなど）

プロビジョニングと資源管理サービス：レイヤ 2 や PDA のような機器のプロビジョニング

障害対応サービス：Grid Lite 機器の障害対応

ポリシーサービス：Grid Lite 機器のポリシーとポリシーの取り扱い。とくに移動性とオフラインの同期アクションのポリシー

モニタリングサービス：Grid Lite 機器のモニタリング（モニタリング方針は移動性機器やレイヤ 2 機器によって異なる場合がある）

11.7 セキュリティ要件

VPN ベースのセキュリティモデルを適用できることが条件である。無効なセキュリティ方式を備えた機器は、要求に応じてグリッドの一部として動作させることが必要である。

11.8 パフォーマンス要件

主要なグリッドアクティビティに大量のデータを転送しなければならない場合、帯域幅の問題が発生する可能性がある。

11.9 活用事例の状況分析

この活用事例は現在、調査段階である。

11.10 参考資料

準備中です。

12 仮想組織のグリッドポータル

12.1 概要

グリッドを通じて人々が多くの VO のメンバーになり、それぞれの VO が各種の計算や道具ベースのデータその他の資源へのアクセスを提供する場合、収集した資源を VO メンバーに利用してもらうための一元的なエンドユーザ向け画面として、VO がグリッドポータルを構築するのはごく当然のことである。VO に参加するユーザが利用する「ワンストップショップ型」のポータルを構築することにより、VO はユーザの資源利用とアクセス性を活発にすることができる。

これらのグリッドポータルには、以下のような共通する要素がある。

- 各種の広範かつ情報価値のある材料を提供する VO の公共性

- 一連のコラボレーティブツール（ディスカッション、ファイルストレージ、カレンダー、告知その他）の提供

- VO メンバー向けに用意された大容量のデータストアへのアクセス

- VO メンバー向けに用意された計算資源の利用機能

これらのポータルは、通常 Web ベースその他のツールを組み合わせたものです。一般に、

必須機能はグリッド対応の Web サーバを通じて提供し、より高度なツールはユーザのデスクトップに導入される。

複数のグリッドポータルで再利用可能ないくつかの共通な要素があり、ユーザが一つのポータルから別のポータルに移動する際のアクションを簡素化する場合、仮想組織グリッドポータルの開発と導入のためのベストプラクティスと技術を構築しておくことが重要である。

12.2 顧客

この機能の顧客は、ユーザが資源にアクセスするためのコンポーネントを提供する有能な仮想組織です。仮想組織グリッドポータルは、多くの点でこの文書で述べている他の多くのシナリオでも利用可能な機能である。このシナリオは、他のシナリオのための特定のポータルを説明したものではなく、すべての仮想組織グリッドポータルで利用する可能性のある共通のツールや機能に焦点を当てている。

12.3 シナリオ

希少な物理資源に焦点を当てた共同研究中心のグリッドが増えている。こうした資源は大規模、あるいは高コストである場合が多く、世界でもごく限られた場所にしか導入されていない。この種の共同研究の例としては、天文学、高エネルギーおよび原子物理学、核融合研究、地震エンジニアリングなどがある。

これらの広範な共同研究には、一般に次のような特徴がある。

地理的に分散した場所からの計算やデータ、インストルメントへのアクセス
参加者が遠距離からのミーティングと共同作業をする環境が必要

これらの共同研究の大半は、世界的規模と組織横断的な性格を有す。共同研究では、さまざまな規模の多くのグループが、広範な研究課題に応じて動的に形成されます。こうした研究課題には、実験デザイン、実験スケジューリング、装置オペレーション、管理、研究結果の発表など他にも数多くある。これらのグループはすべて、あらゆるタイムゾーンに属する世界各地に分散したメンバーで運営しなければならない。

これらの共同研究では、データやアイデア、各グループの対話の安全性を確保することが極めて重要である。設備の利用という点で全般的には共同研究であっても、その中のサブグループ内には研究成果を追究するための競争が見られることも少なくない。また、あらゆるタイプの実験機器の管理と操作、あるいは実験機器からリアルタイムで得られるデータのモニタリングを行う場合は、適切なセキュリティとアクセスコントロールが不可欠である。

12.4 関連資源

グリッドポータルは、仮想組織の範囲内にある、あらゆるタイプのグリッド対応資源へのインタフェースとアクセスを提供することができる。これらはコンピュータ資源から物理センサー、データ資源までさまざまである。これらは集中的に配置することも分散配置することも可能である。

12.5 OGSA プラットフォームの機能要件

このシナリオはクロスカッティング的な性格を持つため、OGSA プラットフォームの機能要件は、OGSA プラットフォーム文書で述べたすべてのサービスにわたる。

12.6 OGSA プラットフォームサービスの利用

このシナリオはクロスカッティング的な性格を持つため、グリッドポータルは、OGSA プラットフォーム文書で述べたすべてのサービスを利用する可能性がある。仮想組織グリッドポータルは、次節で述べるように、OGSA プラットフォームのセキュリティ機能にある種の重圧をかける。

12.7 セキュリティ要件

複数の仮想組織のメンバーが適正な構成員数に達し、より多くのユーザがグリッドと対話するようになると、いくつかの新たな問題が最前線に発生し、解決が必要になる。これらの問題は次の2つのカテゴリに大別される。

セキュリティプロキシ機能

信用管理の問題

セキュリティプロキシ機能は、重要かつやや短期的な問題である。セキュリティプロキシ機能の必要性を理解するには、ユーザが携帯電話から WAP を使用し、グリッドポータルでバッチジョブのモニタリングや何らかの方法でこれを操作できるようにしたいとする。近い将来、携帯電話が OGSA プロトコルやサービスを完全にサポートするようになるとは考えられない。携帯電話ユーザがグリッド内で操作を実行するためには、WAP プロトコルを携帯電話へ、グリッドプロトコルを他のグリッドへ伝送するプロキシが必要となります。

将来的に、すべての機器がネイティブに OGSA サービスやプロトコルをサポートするようになれば、これは不要となるだろう。

信用管理はセキュリティプロキシと関連しているが、いくつか重要な点で異なる。携帯電話が直接グリッドプロトコルを実行できないように、グリッドの信用データを伝送して識別情報を正しく確立することはできない。このため、ユーザの信用情報を処理できる中間的な仕組みが必要となる。

ユーザが異なる信用方式と信用機関を持つ数千もの仮想組織に参加することで、問題はあっという間に複雑になる。いくつかの点で、こうした信用の管理は完全に制御不能となる。モバイルユーザが、一日の間に異なるワークステーション間を移動するような場合にはとくにこの傾向が強まる。数百の信用情報を、使用に先立って、ユーザが一日中使用する可能性のあるすべての機器にインストールすることは、実用的ではない。キー素材をインストールする不便さ以上に、人があちこち移動することにより、安全でない場所にキー素材を設置するという重大な危険性をはらんでいる。

このための理想的な解決策は、ユーザが使用しようとしているコンピュータには実際に設置せずにキー素材を組み込むことのできる「スマートカード」を使用することである。ただし残念なことに、スマートカードは市販のすべてのハードウェア / オペレーティングシステムの組み合わせで効果的にサポートされているわけではないという点である。

最終的な結論は、ユーザの信用情報を管理するための仕組みが必要であるということである。現在基本的な仕組みとして MyProxy [参考資料] 方式を利用できるが、その信用情報を管理および使用するにはある程度の高度な知識を必要とする。また、ユーザはそれぞれの信用情報の場所と目的を追跡し続けなければならない。

これらのセキュリティの問題は仮想組織グリッドポータルに固有のものではないが、迅速かつ容易にポータルを導入できる企業が増加するにつれて、こうした問題はすぐに重要になってくるであろう。

12.8 パフォーマンス要件

一般に、グリッドポータルにはパフォーマンスに関して重要な問題はない。しかし、グリッドポータルが、グリッドプロトコルを通じて利用できる非グリッド対応ツールや資源間で、プロキシとして動作しなければならない状況も多々ある。この種のプロキシアクティビティのいくつかは、存続期間が短く、トランザクショナルな性格を持つ。購読する OGSA サービスに対してプロキシが必要な携帯電話で動作する Brew [参考資料] アプリケーションのような長期間については、他のプロキシアクティビティを維持管理しなければならない場合がある。

12.9 活用事例の状況分析

仮想組織グリッドポータルで未だ検討されていないニーズは主に、次の 2 つの基本カテゴリに分類される。

上に述べたようなセキュリティと信用機能の強化に対するニーズ
基底のサービスの「ユーザの意見」を反映するハイレベルなサービスのニーズ

「ユーザ中心型」サービスのニーズを理解するには、Globus ツールキットの GridFTP 機能を理解し、コマンドラインプログラム、globus-url-copy でプログラマーが使用する GridFTP API を比較してみるとよいだろう。GridFTP API は極めて強力かつ柔軟で、GridFTP のすべての機能を高度なプログラマーに公開する。globus-url-copy コマンドは、

(最も単純な場合で) URL の形式で 2 つのパラメータを使用し、データをコピーする。

仮想組織グリッドポータルに移行するにつれて、GridFTP API よりも globus-url-copy コマンドにより近い、シンプルでハイレベルな機能を提供する OGSA サービスへのアクセスがいつそう必要になるものと考えられる。仮想組織のプログラマーは、新しい作業を処理するためにこうしたいくつかの上位サービスを容易に構成できる小型のアプリケーションを記述することが要求される。これらのアプリケーションは、JAVA や C ではなく、JSP、Perl、TCL/TK などの言語で記述されるかも知れない。また、下位(強力/柔軟)サービスの書き手がこのような上位のサービスをも提供することは完全に可能であり、また望ましいことである。同じグループが実装する両サービスの利点は、上位のサービスほど下位のサービスのテストにとって自然な仕組みとなることである。

当初より、この概念と同種のものと思われる研究がいくつか行われている。

Globus ツールキット COG [参考資料] は、グリッドの機能を「より使いやすい」形態にカプセル化する例の一つです。COG は簡素なグリッドツールの構築を、各種の簡素な言語で実現した。この種の研究の初期の例としてもう一つ、グリッドポータル開発キット (GSDK) があります。これは上位のグリッド機能を、JSP 言語で開発できる一連の JavaBeans にカプセル化したものである。

ここで重要なことは、主要なニーズは、これらの言語/環境による特定の実装ではなく、上位のユーザ指向の機能性を提供するサービスを通じて、広範なポータルツールキットを開発できるということである。これらのサービスはより基本的な OGSA サービスに基づいて構築されるレイヤと見なすことができます。

12.10 参考資料

- [1] A Toroidal LHC Apparatus (ATLAS) <http://atlas.web.cern.ch/Atlas/>
- [2] Brew、<http://www.qualcomm.com/brew/>
- [3] Commodity on the Grid (COG)、<http://www.globus.org/cog/>
- [4] Compact Muon Solenoid (CMS) <http://cmsinfo.cern.ch/Welcome.html>
- [5] DZero <http://www-d0.fnal.gov/>

- [6] European Virtual Observatory (EVO)
- [7] The George E. Brown Jr. Network for Earthquake Engineering Simulation (NEES) www.neesgrid.org
- [8] Japanese Virtual Observatory (JVO) <http://jvo.nao.ac.jp/>
- [9] Laser Interferometer Gravitational Wave Observatory LIGO <http://www.ligo.caltech.edu/>
- [10] MyProxy、<http://grid.ncsa.uiuc.edu/myproxy/>
- [11] National Fusion Collaboratory (FusionGrid) <http://www.fusiongrid.org/>
- [12] Sloan Digital Sky Survey (SDSS) <http://www.sdss.org/>
- [13] US National Virtual Observatory (NVO) <http://www.us-vo.org/>

13 Persistent Archive

13.1 概要

我々は、仮想データグリッド技術で提供された機能を使い、多くの大量データ科学保存環境を構築した (California Digital Library、NARA persistent archive、NFS National Science Digital Library など)。保存環境は、新旧プロトコルや新旧ソフトウェアシステム、新旧ハードウェアシステム間のマッピングを管理し、真正のレコードを維持する適切な抽出レイヤを提供することによって、技術の進化に対処する。保存環境は一般にデジタルエンティティをコレクションに組織化する。

真正性は、適切なメタデータ属性をコレクションに追加することによって追跡し、出所の記述、データに関して実行された操作の追跡、監査証跡の管理、アクセスコントロールの管理を行う。データが変更されていないことをチェックするには検証機能が提供されます。仮想データグリッドは次の 2 つの必須機能を提供します。

- 仕様からの「誘導データ製品」構築のサポート。誘導製品は、新しいエンコーディングフォーマットに対するデジタルエンティティの「転換移行」や、または「アーカイブ形式」コレクションの構築に使用されるアーカイブプロセスのアプリケーションにもなり得る。
- サービスの実行に関連する完了状態の管理。なお、「アーカイブプロセス」のアプリケ

ーションの結果を記述する「完了状態」は、真正性をチェックするために保存する必要がある。

Persistent Archive は、一過性の「実行状態」だけでなく、「完了状態」も保存される点で仮想データグリッドと異なる。Persistent Archive は標準のリモートデータアクセストランスペアレンシーをベースとしている。

- 場所に依存しないネーミング規則を提供する論理名前空間
- リモートストレージシステム（ファイルシステムやアーカイブ、データベース、Web サイトなど）上で実行される一連のオペレーションを特徴付けるストレージリポジトリ抽出
- データベース内のコレクションの管理に使用する一連のオペレーションを特徴付ける情報リポジトリ抽出
- Persistent Archive でサポートする一連のサービスを特徴付けるアクセス抽出

保存環境は、コレクションのアーカイブ形式の作成に使用されるアーカイブプロセスをサポートします。アーカイブプロセスの構成は以下のとおりである。

- 評価 保存するデジタルエンティティの分析
- アクセション デジタルエンティティのデータグリッドへの受け入れ管理。これは通常、登録ステップ、それに次ぐデータ伝送ステップに対応
- 配置 デジタルエンティティを保持する階層コレクションの作成
- 記述 出所および真正メタデータの各デジタルエンティティへの割り当て
- 保存 転換移行によるアーカイブ形式の作成、およびデータの格納
- アクセス 登録されたデジタルエンティティの発見と取得のサポート

13.2 顧客

大量のデータコレクションを収集するグループ、または 3 年以上（技術が旧式化するまでの時間）にわたってコレクションを管理しようとするグループはすべて、同等の技術を必要とします。ユーザには、NARA、Library of Congress、NHPRC state persistent archives、NSF NSDL、NVO、NIH BIRN、NASA ADG、NASA IDG、DOE PPDG などが含まれる。

科学向けのデータを扱う際は、とくに次の3種類の機能が重要です。

- 並行 I/O のサポートにより、ウィンドウサイズとシステムバッファサイズを最適化することなく効果的にデータを送信
- 登録、ロード、アンロード、削除などのバルクオペレーションのサポート。
- リモートストレージリポジトリにおける直接的なデータのサブセット化、メタデータの抽出、バルクオペレーションのためのリモートプロキシのサポート。

我々が共同研究を行っているすべてのコミュニティで、小規模なデータセットを処理しています（ネットワークレイテンシよりも小さなサイズ* 帯域幅遅延製品）。全体として、これらのデータは数十テラバイトからペタバイトで測定される。

一例として 2 Micron All Sky Survey がある。これは 500 万画像のコレクションでデータ容量は総計 10TB になる。画像はコレクションに登録され、コンテナに集約、さらに HPSS アーカイブに格納される。コンテナはアーカイブで見られたファイル数を最小化するために使用される。SDSC では、アーカイブは 700 TB データ以上を格納しますが、1,700 万ファイルに過ぎません。わずか 10 テラバイトのデータのために HPSS 名前空間に追加された 500 万の名前は、無効として表示されます。画像をコンテナに集約することにより、10 テラバイトを 147,000 「ファイル」に格納することができました。コンテナに画像を書き込む際にこれらを格納したことから、同じ空域のすべての画像は同じコンテナに格納され、モザイク構築を極めて容易にサポートできるようになりました。

リモートプロキシ利用のもう一つの例は、Digital Palomar Observatory Sky Survey です。このケースでは、各画像は 2GB の容量です。関心対象の星の範囲を抽出するには、処理プラットフォームで画像全体の移動を必要とし、これには 4 分かかりました。リモートプロキシは、リモートストレージシステムで直接画像の切り抜き操作をサポートするように記述され、完了時間は数秒に短縮された。

我々がサポートするすべてのコレクションは、マルチサイトです。サイト間のレプリケーションは以下の目的に不可欠である。

- 災害復旧。火災や地震でコレクションを消失することはできない。
- フォールトトレランス。サイトが停止しても、代替サイトを通じて引き続きデータにア

クセスすることができる。

- パフォーマンス。サイト間で負荷分散方式のアクセスが可能である。
- Curation。データは異なる機関に常駐する専門家によって維持管理される。主要なコピーは専門家のいるサイトで保持するのが一般的である。

13.3 シナリオ

主なシナリオは、上に挙げたアーカイブプロセスの実行です。ストレージ資源ブローカは上に述べたすべての機能を実装しており、複数の Persistent Archive のサポートを本稼動で利用しています。主な事例は以下のとおりである。

- California Digital Library。各連邦政府の Web サイトの閲覧により 1,690 万のデジタルエンティティ、1.5TB のデータを収集する。デジタルエンティティは SRB 論理名前空間に登録され、Web ブラウザの http インタフェースを通じてアクセスする。これにより、元のアクセスに使用したのと同じ Web 方式でアーカイブされた資料を表示することができます。各デジタルエンティティの URL は、デジタルエンティティの登録に使用した論理名前空間に、属性としてマッピングされる。

- NARA Persistent Archive。このプロジェクトでは、NARA デジタルホールディングスが SRB データグリッドに登録され、U Md、NARA、SDSC 間で複製されます。現在 1.5TB 以上のデータが登録されている。

- NSF National Science Digital Library。SDSC は Cornell の中央リポジトリに登録されている各デジタルエンティティのコピーを保持する Persistent Archive を運営している。デジタルエンティティの数は急速に増加している。システムは現在 1,500 万のデジタルエンティティを保有しており、平均容量は 50 キロバイトである。

13.4 関連資源

Persistent Archive は、最大 1 ペタバイトのデータと数千万ファイルを格納する。ストレージ資源ブローカは次のプラットフォームにインストールされる。

- Sun、AIX、Linux、64-bit Linux、HP True-64、Mac OS X、Windows NT

また、以下のアクセスに利用される。

- ファイルシステム (Unix、Mac OS X、Windows、Linux)、アーカイブ (HPSS、Unitree、ADSM、DMF)、データベース (DB2、Oracle、Sybase、Informix、SqlServer、Postgres)、オブジェクトリングバッファ、階層資源マネージャ、Web サイト、FTP サイトなど。

また、次のアプリケーションエリアで要求される API を通じて、システムにアクセスを提供する。

- C ライブラリコール、C++ライブラリコール、Unix Shell コマンド、Python ライブラリ、Windows DLL ライブラリ、Windows ブラウザ、Web ブラウザ、Open Archives Initiative、WSDL、Java

13.5 OGSA プラットフォームの機能要件

ここには、優先アクセス方式はユーザコミュニティによって指定されるという問題がある。コミュニティでは、すべてのケースにおいて、分散データへのアクセスにはレガシ API の使用を依然として優先する傾向にある。一例として、Caltech による CMS 高エネルギー物理プロジェクトがある。同社は Python をもとにした Clarens と呼ばれる分析プログラムを開発した。このため、SRB との対話には Python I/O ライブラリを要求しています。

デジタルライブラリコミュニティ (NSF NSDL プロジェクト) は、メタデータの交換にオープンアーカイブイニシアチブプロトコルの使用を必要とした。これはサイト間で交換されるメタデータの簡素なパッケージングである。

Web サービス記述言語環境は、Java をベースとしている。このため、我々は SRB に純粋な Java インタフェースを実装した。

現行の Persistent Archive と、OGSA ベースの Persistent Archive 向けに提供されるサービスの主な差異は、機能を複合セットに統合している点である。我々は、論理名前空間へのファイルのバルク登録、ストレージリポジトリへのデータのバルクローディング、データのバルク抽出、データのバルク削除の管理能力を最適化する必要に迫られてきている。

言い換えれば、一つの要求を発行したら、次に 10,000 のオペレーションを 10,000 のファイルに対して実行しなければならないということである。これを実現する方法は以下のとおりである。

- 許可、ファイルの格納場所の決定、ファイルのアクセス、ファイルの取得を一つのコマンドに統合すること。データグリッドはこれらの個々のオペレーションを、ユーザとの追加的な対話を必要とせずに処理しなければならない。
- バルク登録のサポート。これはリモートファイルに関する位置情報の一連のメタデータ連結ファイルへの集約、およびファイルのバルクロードのメタデータレジストリへの集約です。1 秒間に 600 ~ 1000 命令の速度が必要である。
- バルクローディングのサポート。これはファイルのコンテナへの集約と位置情報のメタデータカタログへの集約の組み合わせである。

第二の差異は、複数のサービスにわたって動作する整合性制約機能の実装です。複製されるコンテナに関するアクセスコントロールを考えてみる。SRB では、アクセスコントロールは、コンテナに登録されるそれぞれのデジタルエンティティで、コンテナのすべてのコピーについて適用される。アクセスコントロールは論理名前空間のプロパティです。論理名前空間のオペレーションは、属性として論理名前空間にマッピングされる「完了状態」情報となり、メタデータカタログに格納されます。問題をより具体的にするために、複製されたコンテナに集約されているファイルへの書き込みについて考える。データグリッドは以下の実装が必要である。

- 論理名前空間へのアクセスコントロールのマッピング
- コンテナに関する書き込みロックの管理
- レプリカコピーに関する同期フラグの管理
- レプリカを同期させるための方式

データが暗号化または圧縮される際は、同じような一連の制約が発生する。暗号化や圧縮状態は、論理名前空間のプロパティとする必要があり、データがどこに移動しても転送前に正確な暗号化アルゴリズムが使用できること、正確な暗号化アルゴリズムがクライアントによって呼び出されることが条件となる。

必要とされる一連のサービスは、アプリケーションエリアに強く依存する。このため、マルチテラバイトのデータセットによる 3D ビジュアライゼーションでは、部分的なファイルの読み込み、シーク、データの 3D レンダラーへのページングの機能が必要である。データのページングをサポートする OGSA サービスは、3D レンダリングシステムには重過ぎるかも知れない。サービスは、データやメタデータの操作にも必要である。メタデータの操作例としては、リモートストレージリポジトリにあるファイルからのメタデータの自動抽出、メタデータのメタデータリポジトリへのバルクロードがある。また、メタデータの発見例としては OAI ベースのメタデータ抽出や、抽出されたメタデータの HTML または XML ファイルへのフォーマット化がある。興味深いメタデータサービスとしては、アクセスコントロールリストのデジタルエンティティへのプロビジョンの他にメタデータ属性へのプロビジョンがある。

データグリッドにとって主要な課題は、「完了状態」の整合性ある管理である。どれほど大規模なコレクションでも、デジタルエンティティに関して整合性を保った状態でメタデータを維持しなければならない。我々はデータベースを利用して「ハード状態」リポジトリにある状態情報を管理する。メタデータのアップデートはサービス内で行われ、内部の状態情報は部分的な完了状態にあるオペレーション向けに維持される（レプリカへの書き込みなど、結果的にコピー全体で同期させる必要がある）。

明示的なデータオペレーションは以下のとおりである。

許可の変更 データグリッドコレクションまたはデータセットに関するアクセス許可の変更に変更可能

コピー データグリッドコレクションまたはデータセットの内容を、デフォルトのストレージ資源またはその他のストレージ資源内に格納されている新しいコレクションまたはデータセットにそれぞれコピー。

作成 新しいコンテナまたはコレクションを作成

データセットの受け入れ 添付ファイルとして提示されたデータセットをデータグリッド要求に挿入

データセットのダウンロード 添付ファイルとしてのデータセットをデータグリッド応

答にダウンロード

削除 データグリッドコレクションまたはデータセットを削除

リスト コレクションまたはコンテナの内容を表示

チケットの準備 新しいグリッドチケットを準備

名前の変更 コレクションまたはデータセットの名前を変更

レプリケート コレクションまたはデータセットの内容を複製

SeekN'Read データセットのポイントをシークし、指定されたバイトを添付ファイルとして読み込む (GET)

SeekN'Write データセットのポイントをシークし、提示されたバイトを添付ファイルとして書き出す (PUT)

13.6 OGSA プラットフォームサービスの利用

OGSA データサービス、Persistent Archive を利用する場合は、バルク登録、ロード、アンロード、削除機能を実装する。

13.7 セキュリティ要件

Persistent Archive は、格納されたデータへのアクセスコントロールを提供する必要がある。現行の SRB は、GSI 1.1 および GSI 2.4 と相互運用する。次のステップでは GSI 3 と相互運用する予定である。

13.8 パフォーマンス要件

最終目標は可能な限りすべての帯域幅を利用し、1 秒間に 1,000 ファイルを登録することである。

13.9 活用事例の状況分析

現在は OGSA を使用していない。その代わりにネイティブの API と WSDL/SOAP を実装していない。

13.10 参考資料

[1] R. Moore、A. Merzky 「Persistent Archive Concepts」、Global Grid Forum

Persistent Archive Research Group, Persistent Archive Recommendations についての
草案、2003年5月3日

[2] R. Moore, 「 Common Consistency Requirements for Data Grids, Digital
Libraries, and Persistent archives 」、Grid Protocol Architecture Research Group,
Global Grid Forum、日本、東京、2003年3月5日

[3] R. Moore、C. Baru、「 Virtualization Services for Data Grids 」, 『 Grid
Computing : Making the Global Infrastructure a Reality 』の章より、John Wiley &
Sons Ltd、2003年

[4] Arcot Rajasekar、Michael Wan、Reagan Moore、George Kremenek、Tom
Guptil、「 Data Grids, Collections, and Grid Bricks 」、Mass Storage Systems に関す
る第 20 回 IEEE Symposium および Mass Storage Systems and Technologies に関する
第 11 回 Goddard Conference の議事録、San Diego、2003年4月

[5] Michael Wan、Arcot Rajasekar、Reagan Moore、Phil Andrews、「 A Simple
Mass Storage System for the SRB Data Grid 」、Mass Storage Systems に関する第 20
回 IEEE Symposium および Mass Storage Systems and Technologies に関する第 11 回
Goddard Conference、San Diego、2003年4月

[6] Arcot Rajasekar、Michael Wan、Reagan Moore、Arun Jagatheesan、George
Kremenek、「 Real Experiences with Data Grids - Case studies in using the
SRB 」、High-Performance Computer Architecture に関する国際シンポジウム、日本、九
州、2002年12月

[7] R. Moore、「 The San Diego Project : Persistent Objects 」、the Workshop on
XML as a Preservation Language の議事録、Urbino、Italy、2002年10月

[8] Edward A. Fox、Virginia Tech : Reagan W. Moore、San Diego Supercomputer
Center; Ronald L. Larsen、University of Pittsburgh; Sung Hyon Myaeng、Chungnam
National University; および Sung-Hyuk Kim、Sookmyung Women's University、
「 Toward a Global Digital Library : Generalizing US-Korea Collaboration on Digital
Libraries 」、D-Lib Magazine、2002年10月、<http://www.dlib.org/>

[9] Arcot Rajasekar、Reagan Moore、Bertram Ludäscher、Ilya Zaslavsky、「 The
Grid Adventures : SDSC's Storage Resource Broker and Web Services in Digital
Library Applications : Digital Libraries に関する第 4 回 Russian Conference、

Dubna, Russia, 2002年10月

[10] R. Marciano, B. Ludaescher, I. Zaslavsky, R. Moore, K. Pezzoli, 「Multi-level Information Modeling and Preservation of eGOV Data」、第一回国際会議、EGOV 2002、Aix-en-Provence、France、2002年9月3日

[11] G. Bruce Berriman, David Curkendall, John Good, Joseph Jacob, Daniel S. Katz, Mihseh Kong, Serge Monkewitz, Reagan Moore, Thomas Prince, Roy Williams, 「An Architecture for Access to a Compute Intensive Image Mosaic Services in the NVO」 SPIE Conference 4686 「Virtual Observatories」、Hawaii、2002年8月

[12] R. Moore, A. Merzky, 「Persistent Archive Basic Components」、Persistent Archive Research Group, Global Grid Forum; 2002年7月27日

[13] A. Rajasekar, M. Wan, R. Moore, 「mySRB and SRB, Components of a Data Grid」、第11回 High Performance Distributed Computing conference、Edinburgh、Scotland、2002年7月

[14] R. Moore, 「Preservation of Data, Information, and Knowledge」 the World Library Summit 議事録、Singapore、2002年4月

[15] R. Boisvert, P. Tang, 「The Architecture of Scientific Software」 pp. 273-284、 『Data Management Systems for Scientific Applications』 Kluwer Academic Publishers、2001年

[16] C. Chen, 「Global Digital Library Development」 pp. 197-204、 『Knowledge-based Data Management for Digital Libraries』 Tsinghua University Press、2001.

[17] A. Rajasekar, R. Moore, 「Data and Metadata Collections for Scientific Applications」、High Performance Computing and Networking (HPCN 2001)、Amsterdam、Holland、2001年6月

[18] H. Stockinger, O. Rana, R. Moore, A. Merzky, 「Data Management for Grid Environments」 European High Performance Computing and Networks Conference、Amsterdam、Holland、2001年6月

[19] R. Moore, 「Knowledge-based Grids」 Mass Storage Systems に関する第18回 IEEE Symposium の議事録および Mass Storage Systems and Technologies に関する

第 9 回 Goddard Conference の議事録、San Diego、2001 年 4 月.

[20] B. Ludäscher、R. Marciano、R. Moore、「Preservation of Digital Data with Self-Validating、Self-Instantiating Knowledge-Based Archives」ACM SIGMOD Record、30 (3)、54-63、2001 年

14 相互認可

14.1 概要

SA3-RG が作成した「グリッドの認証、許可、アカウントینگ要件」は情報提供型文書で、グリッドのセキュリティ要件について解説しています。この文書に述べる重要な要件の一つは、相互認可の必要性である。

ただしこの要件は、現在のグリッドツールキットの技術では提供することはできない。

ジョブがグリッド上の特定の資源に送信されると、ユーザはこの資源によるジョブの実行と、この結果生成されたデータを、ジョブ送信の資源をターゲットとした動作を通じて暗黙に処理することを許可する。しかし一方で、負荷分散やサービスの期待品質を満たすために指定した資源がこのジョブを別の資源に転送または再送信することもある。この二次的なリモート資源は、仮想組織によって信用される可能性はあるが、グリッドジョブのオーナーからは信用されない。

14.2 顧客

相互認可要件は一般に、グリッドジョブの知的財産 (IP) や結果データを保護するための特定のセキュリティニーズを持つ大規模サイトの顧客から提示される。これには活用事例の商用データセンターや National Fusion Collaboration が該当しますが、追加的または固有のセキュリティニーズが発生する。

14.3 シナリオ

このニーズは、重要なデータを使用または生成する、あるいはジョブ自体が IP 価値を有するグリッドジョブをユーザが送信するシナリオで発生する可能性がある。グリッド VO は各種のコンピュータを信用する可能性があるが、ユーザはこのジョブを、セキュリティが侵害されたと分かっている OS や、特定の OS セキュリティ機能やアップデートを備え

た OS 上のみで実行しようとは考えない。

14.4 関連資源

利用される資源は、ユーザに対してコールバックサービスを提供しなければならない。コールバックは、資源がグリッドジョブを別のリモート資源に転送または再送信する前に使用する。このコールバックは二次的リモート資源とユーザの関連グリッドジョブを識別する。ユーザはこの相互認可コールを使って二次的リモート資源を許可する。

14.5 OGSA プラットフォームの機能要件

以下のリストには、この活用事例に必要な OGSA 文書の機能要件が含まれている。

- ポリシー

- 複数のセキュリティインフラ

- 境界セキュリティソリューション。

14.6 OGSA プラットフォームサービスの利用

以下のリストには、相互認可の活用事例で使用されている OGSA 文書のサービスが含まれている。

- 名前解決と発見

- セキュリティ

- ポリシー

- イベント

- サービスの統合

14.7 セキュリティ要件

セキュリティ要件は上に述べたとおりである。

14.8 パフォーマンス要件

相互認証プロセスは、自動化および手段化する必要がある。

14.9 活用事例の状況分析

現在の活用事例は、今のところこの要件に対応できるようには見えない。OGSA はグリッドとグリッドを構成する資源やコンピュータを仮想化する。相互認可の要件は、ジョブ分散のエンドツーエンドの知識を必要とする。

14.10 参考資料

http://www.gridforum.org/2_SEC/SAAA.htm

15 資源利用サービス (RUS)

15.1 概要

資源利用サービス (RUS) は、分散した異種環境におけるアプリケーションやミドルウェア、オペレーティングシステム、物理 (計算およびネットワーク) 資源が生み出す資源利用メトリクスの調停を容易にする。これはオーブングリッドサービスアーキテクチャの中核サービスの 1 つである。

15.2 顧客

RUS は、通常はコスト配分やキャパシティプランニングに関連するシナリオなど、さまざまな動機で資源消費の測定に関心を持つ顧客が活用する。想定される顧客層は、商業分野および科学分野に属しているものと考えられる。

15.3 シナリオ

RUS は、コスト配分 (入金取り消しなど)、キャパシティおよびトレンド分析、不正行為および侵入検知、動的プロビジョニング、サービス品質保証契約の遵守、Web サービスの料金設定、作業負荷管理などをベースとする広範な利用シナリオのサポートを目的としている。

15.4 関連資源

関連資源には、利用状況を測定する必要のあるすべての資源が含まれる。

15.5 OGSA プラットフォームの機能要件

以下のリストは、オーブングリッドサービスアーキテクチャ文書で述べた機能と、資源利

用サービスで実施されるこれらの機能に対する関係を表したものである。

発見とブローカリング

RUS は、資源利用メトリクスを生成する資源を特定するため、発見機能を使用する場合がある。

メータリングとアカウンティング

RUS はこの機能の主要なコンポーネントである。

データ共有

既知の要件はない。

仮想組織

既知の要件はない。

モニタリング

RUS は、モニタリングファブリックが提供する機能を使用して利用メトリクスを収集する。

ポリシー

ポリシーサービスは、RUS インスタンスの構成と統合を推進する。

セキュリティ

セキュリティは、従来の認証、許可、アカウンティング (AAA) システムが提供するアカウンティング機能をサポートしなければならない。商用ベースのシナリオには、消費メトリクスに AAA システムから取得したアカウントコードでタグ付けする資源利用サービスを必要とするものもある。

15.6 OGSA プラットフォームサービスの利用

以下のリストは、RUS に対するオープングリッドサービスアーキテクチャ文書で述べたサービスの関係を表したものである。

中核サービス：名前解決と発見

RUS は、この中核サービスを利用して参照へのハンドルを解決する。

中核サービス：サービスドメイン

RUS は、このサービスは利用しないものと考えられる。

中核サービス：セキュリティ

RUS は、AAA 機能を利用してアカウントコードを取得する。また、資源利用メトリクスに対する不正なアクセスから保護するためのセキュリティサービスも必要である。許可管理は、オペレーションの呼び出しとサービスデータ要素へのアクセスの両方において必要である。

中核サービス：ポリシー

RUS インスタンスは、ポリシーサービスを通じて設定される。

データおよび情報サービス：データ管理

RUS は、このサービスは利用しないものと考えられる。

データおよび情報サービス：メッセージング、キューイング、ロギング

RUS はメッセージングとキューイングを使ってメトリクスを交換します。RUS は監査と回復のためにロギングサービスが必要である。

データおよび情報サービス：イベント

資源メトリクスはイベントであり、RUS はイベントサービスを活用および一致させなければならない。

データおよび情報サービス：メータリングとアカウントティング

RUS はこの一連のサービスのメンバーである。

データおよび情報サービス：トランザクション

RUS は、このサービスは利用しないものと考えられる。

コンピュータ計算と資源の管理：サービスの統合

資源の利用は分散環境で測定されるため、RUS インスタンスは他のインフラ（メッセージングなどの）サービスと回線で接続（統合）されていなければならない。

コンピュータ計算と資源の管理：管理

管理サービスは RUS の導入、変更、識別を管理する。

コンピュータ計算と資源の管理：プロビジョニングと資源管理

プロビジョニングシステムは、RUS から取得した資源利用メトリクスを使用し、これらのプロビジョニングを決定する。

コンピュータ計算と資源の管理：予約とスケジューリングサービス

RUS は、このサービスは利用しないものと考えられる。

コンピュータ計算と資源の管理：導入サービス

導入サービスは、RUS をサポートするソフトウェアの導入に使用される。

15.7 セキュリティ要件

RUS では、潜在的に重要な資源利用情報へのアクセスを保護するために、セキュリティサービスを必要とする。また、RUS は AAA システムから抽出したアカウント情報を利用する。

15.8 パフォーマンス要件

増大する資源利用データのコストを最小化するには、RUS を最も効率的に実装することが肝要である。一般に資源消費の測定コストは、総資源消費コストのごく一部にとどめるようにする必要がある。

15.9 活用事例の状況分析

RUS は基盤となる資源で生成されたメトリクスを消費するため、資源測定を管理する標準的な意味やポリシーが必要かも知れない。おそらくこの機能は、Web サービス分散管理 (WSDM) または共通管理モデル (CMM) で取り組むべき課題であると思われる。

15.10 参考資料

http://www.ggf.org/3_SRM/rus.htm

16 編者の連絡先

Ian Foster、Distributed Systems Laboratory、Mathematics and Computer Science
Division Argonne National Laboratory、Argonne、IL 60439

電話番号：630-252-4619 電子メール：foster@mcs.anl.gov

Dennis Gannon、Indiana University、Bloomington、IN 47405

電話番号：812-855-5184 電子メール：gannon@cs.indiana.edu

Hiro Kishimoto、Grid Computing & Bioinformatics Laboratory

Fujitsu Laboratories Limited、Kawasaki、Japan 211-8588

電話番号 : +81-44-754-2628 電子メール : hiro.kishimoto@jp.fujitsu.com

Jeffrin J. Von Reich, Software Engineering Business Unit,
Hewlett Packard, Fort Collins, CO, USA – 80528

電話番号 : +1-9708980700 電子メール : jeffrin_von-reich@hp.com

謝辞

この文書は、以下の組織の協力により完成することができました。IBM、W-31-109-Eng-38、DE-AC03-76SF0098 の契約に基づく米エネルギー省、Office of Advanced Scientific Computing Research の数学、情報、計算科学部門サブプログラム、全米科学財団、NASA Information Power Grid プロジェクト、Hewlett Packard。

この文書に貢献いただいた以下の方々およびグループに心より謝意を表します。Nedim Alpdemir、Takuya Araki、Boas Betzler、Kate Keahey、William Lee、Tan Lu、Norman Paton、Jon MacLaren、Andreas Savva、Charles Severance、David Snelling、Ravi Subramaniam、Fred Maciel、Takuya Mori、Andrew Grimshaw、Jem Treadwell、Latha Srinivasan 各氏、本資料の検討およびレビューに参加いただいた GGF ワーキンググループおよび OGSA ワーキンググループメンバーの他すべてのメンバー。

参考資料

参考資料は各章に記載しています。