2007－HPC－111 (49)
2007／8／3

WSRFに基づく情報サービスのXACMLによるアクセス制御

竹房 あつ子[†] 中田秀基[†] 柳田誠也[†,††]
工藤知宏[†] 田中良夫[†] 関口智嗣[†]

グリッドでは資源やジョブに関する情報の提供は重要な機能のひとつであり，クラスタ監視ツールのGangliaやGlobus Toolkit 4のMDS4などの情報サービスが実現されている．しかしながら，複数の仮想組織に属するユーザが資源を共有する場合，全情報をすべてのユーザに開示することは好ましくない．本研究では，認可モデルとポリシ記述言語の標準として提案されているXACMLを用い，サイト毎に情報開示ポリシを定義して，各ユーザからの細粒度の情報アクセス制御を可能にする，WSRFに基づく情報サービスシステムを提案する．本研究で開発したプロトタイプでの予備実験から，(1) 情報収集に関するオーバヘッドはWSRF/GSIによるもので占められ，認可決定の時間は無視できる，(2) 複数サイトから情報収集する場合も，手続きを並行して行うとその所要時間は許容できる，(3)XACMLのポリシ数が多い場合も，判定に要する時間は全体の処理時間に対して十分小さいことが確認できた．

Access Contorol using XACML for WSRF-based Information Service

Atsuko Takefusa,[†] Hidemoto Nakada,[†] Seiya Yanagita,,[†,††]
Tomohiro Kudoh,[†] Yoshio Tanaka[†] and Satoshi Sekiguchi [†]

It is an important issue to provide information on Grid resources and job status. The Ganglia cluster monitoring tool and MDS in Globus Toolkit 4 can provide such information service. However, all the information should not be disclosed for all users, who might belong to different virtual organizations. We propose a WSRF-based information service system, which enables various policy definitions by XACML, a standard of authorization model and policy description language, and fine-grain access control of information. The experiments using our prototype system show: (1) The overhead of XACML authorization decision process is negligible, while that of WSRF/GSI is dominant, (2) Parallel information aggregation from multiple sites makes the overhead acceptable, and (3) Even for a large number of policies, the XACML overhead is still negligible over the total processing time.

1. はじめに

グリッドでは，複数の組織から提供される広域に分散する資源を用いて，高性能計算を行うことができる．この際，利用している計算機やネットワーク等の資源の状態や，ユーザのジョブの実行状況の監視要求が非常に大きい．従来，SSH等により割り当てられた計算ノードに直接ログインし，このような情報を直接取得することができたが，大規模計算を行う場合には計算ノード数が非常に多くなってしまったり，そもそも全ノードに個人のローカルアカウントが作られていない場合もあり，手作業での監視には限界がある．

これに対して，グリッドに関するツールキットを提供しているGlobus Toolkit 4[1]では，WSRF(Web Services Resource Framework)[2]に基づく汎用情報サービスをMDS4として提供している．また，Ganglia[3]では管理下の計算ノード群の負荷やメモリ使用量等の資源情報を，Webベースまたは MDS4を介してWSRFベースで公開することができる．これらの情報サービスシステムでは，基本的に全てのユーザに対して全ての情報を公開している．しかしながら，複数の仮想組織に属するユーザが資源を共有する場合，特に商用サービスとして多様なユーザに資源を共有させている場合，実行プロセス情報等を全てのユーザに開示することは好ましくない．個々のユーザやユーザの所属する仮想組織等により，開示する情報を細粒度で制御する必要がある．

本研究では，認可モデルとポリシ記述言語の標準として提案されているXACML (eXtensible Access Control Markup Language)[4]を用い，各サイト毎に情報開示ポリシを定義して，各ユーザからの細粒度のアクセス制御を可能にする，情報サービスシステムを

† 産業技術総合研究所 National Institute of Advanced Industrial Science and Technology (AIST)
†† 原所属：数理技研 SURIGIKEN Co., Ltd.

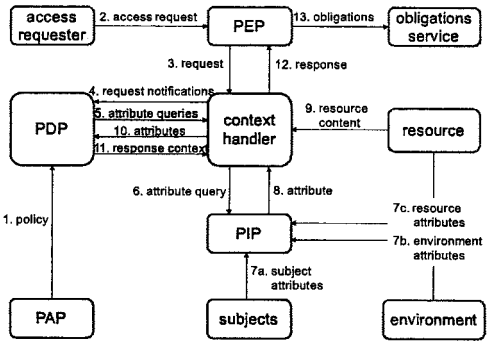

図 1 XACML 認可モデル

提案する．本システムは，各サイトで情報を収集して XACML で定義したポリシに基づく認可決定により情報提供する InformationCollector(IC) と，複数サイトの認可された情報を統合してユーザに提供する InformationAggregator からなり，いずれも WSRF に基づくインタフェースを提供している．

GT4 の WS Core とサンマイクロシステムズの提供する XACML の参照実装を用いて，提案する情報サービスシステムのプロトタイプを開発した．開発したシステム用いて予備的に評価した結果，XACML による認可決定の時間はポリシ数が多い場合も十分小さく，複数 IC から情報収集する場合もその所要時間は許容できることを示す．

2. XACML

2.1 XACML 認可モデル

ウェブサービスのセキュリティに関する標準として，OASIS[5] で規定された SAML (Security Assertion Markup Language)[6] と XACML (eXtensible Access Control Markup Language) がある．SAML は異なるセキュリティドメイン間でのシングルサインオンを目的として，主に認証・ユーザ属性・認可決定を記述するための XML と，各リクエスト／レスポンスプロトコルと SOAP バインディングに関する仕様を規定している．一方，XACML では認可決定を下すのに用いるルールを表現するための XML を規定している．これにより，コンテンツのアクセス制御ポリシを表現することができる．

XACML の認可モデルを**図 1** に示す．主なモジュールを以下に示す．

PIP Policy Information Point. 属性情報を提供する．

PAP Policy Administration Point. ポリシ（ルールのセット），またはポリシセットを生成する．

PDP Policy Decision Point. 適用可能なポリシを評価し，認可決定を下す．

PEP Policy Enforcement Point. 認可決定をもとにアクセス制御を実行する．

context handler (CH) コンテキストを処理するシステムであり，決定要求を XACML 形式へ，認可決定を応答形式へ変換する．

obligation service PEP のポリシ実行に際して，指定された仕事を同時に行う．

認可の手順は以下の通りである．

1. PAP が PDP で利用可能なポリシまたはポリシセットを生成する．
2. リクエスタが PEP にアクセス要求を送る．
3. PEP はその要求を CH に送る．
4. CH は要求を XACML 形式に変換して PDP へ送信する．
5. PDP は subject, resource, action, environment に関する属性情報を CH に要求する．
6. CH は各属性情報を PIP に要求する．
7. PIP は要求された属性情報を取得する．
8. PIP はその属性情報を CH に送信する．
9. CH は必要に応じてコンテキストに resource を含める．
10. CH は要求された属性情報を PDP に送り，PDP はそれを用いて該当するポリシを評価する．
11. PDP は CH に認可決定を送る．
12. CH はその認可決定を PEP の応答形式に変換して，PEP に送信する．
13. PEP は責務を果たす．すなわち，アクセスが許可されれば PEP はそのリソースへのアクセスを許可し，そうでなければアクセスを否認する．

2.2 XACML のコンテキスト

XACML のコンテキストには，認可要求，ポリシまたはポリシセット，認可決定応答がある．上述した認可モデルでは，認可要求を受け取ると，ポリシおよびポリシセットに従って PDP で認可決定を行っている．以下に各コンテキストの構成を説明する．

2.2.1 認可要求

認可要求は**図 2** のように構成されている．`<Request>` タグの下に，誰が，何に対して，何を行いたいかを，それぞれ`<Subject>`, `<Resource>`, `<Action>`タグで記述する．

2.2.2 ポリシおよびポリシセット

ポリシの構成を**図 3** に示す．このポリシを適用する要求は，`<Target>`タグで記述することができる．`<Target>`タグの下には，`<Subject>`, `<Resource>`, `<Action>`を記述するすることができ，これらは認可要求における各タグと対応している．`<Rule>` タグでは，判定規則を記述することができる．ひとつのポリシファイルの中で`<Rule>`を複数記述することもできる．その場合，`<Policy>`タグ内の `RuleCombiningAlgID` 属性において，複数の規則を用いた場合の判定方法を指定することがで

```
<?xml version="1.0" encoding="UTF-8"?>
<Request>
  <Subject>
    <Attribute
      AttributeId="属性識別子"
      DataType="属性の型">
    <AttributeValue>属性値</AttributeValue>
  </Subject>
  <Resource>
    <Attribute
      AttributeId="属性識別子"
      DataType="属性の型">
    <AttributeValue>属性値</AttributeValue>
  </Resource>
  <Action>
    <Attribute
      AttributeId="属性識別子"
      DataType="属性の型">
    <AttributeValue>属性値</AttributeValue>
  </Action>
</Request>
```

図 2　認可要求の構成

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
  PolicyId="ポリシ識別子"
  RuleCombiningAlgID="アクセス可否の判定方法">
  <Description>ポリシの説明</Description>
  <Target>ポリシを適用する要求</Target>
  <Rule>判定規則</Rule>
</Policy>
```

図 3　ポリシの構成

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet
  PolicySetId="ポリシセット識別子"
  PolicyCombiningAlgID="アクセス可否の判定方法">
  <Description>ポリシセットの説明</Description>
  <Target>ポリシセットを適用する要求</Target>
  <Policy>ポリシ</Policy>
  <PolicyIdReference>既存ポリシの ID
  </PolicyIdReference>
</PolicySet>
```

図 4　ポリシセットの構成

きる．その方法として，deny-overrides, permit-overrides, ordered-permit-overrides, ordered-deny-overrides, first-applicable がある．

ポリシセットの構成を**図 4** に示す．ポリシセットは複数のポリシをまとめたものであり，ある要求に対して複数のポリシを同時に適用する場合に利用する．**<PolicySet>**内の **PolicyCombiningAlgId** 属性を用いることで，複数のポリシを用いた場合の判定方法をポリシの場合と同様に指定することができる．ポリシセットにおけるポリシの記述方法として，**<PolicySet>**タグ下の**<Policy>**および**<PolicySet>**タグで直接ポリシを記述する，**<PolicyIdReference>**または**<PolicySetIdReference>**で既存ポリシまたはポリシセットの ID を指定することができる．

```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
  <Result>
    <Decision>認可決定結果</Decision>
  </Result>
</Response>
```

図 5　認可決定応答の構成

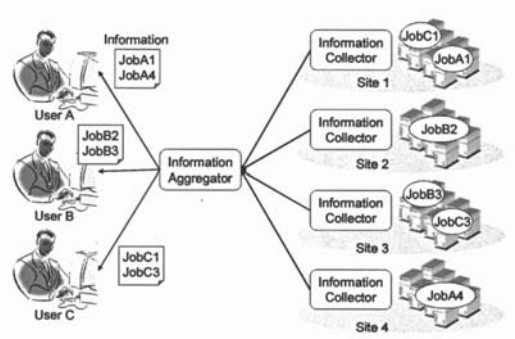

図 6　情報サービスシステムモデル

2.2.3　認可決定応答

図 5 に認可決定応答の構成を示す．**<Result>**タグ下の**<Decision>**で認可決定結果を記述する．

3. WSRF に基づく情報サービスシステム

図 6 に提案する情報サービスシステムモデルを示す．本システムは複数サイトに分散する各ユーザの情報を統合して提供する InformationAggregator(IA) と，各サイトにおいて認可された情報を提供する InformationCollector(IC) からなり，いずれも WSRF インタフェースを提供している．IA では，ユーザから情報取得要求を受け取ると，関連するサイトの IC にそのユーザの権限を移譲して問い合わせる．各 IC は，問い合わせを受けたユーザの属性情報とユーザが望む情報のアクセスポリシとを照らし合わせ，認可されると要求された情報を提供する．IA は各 IC から取得した情報を統合し，要求したユーザに送信する．

例えば，図 6 において User A が実行中である JobA1 と JobA4 の実行状況に関する情報の取得要求を送ると，IA は Site 1 と Site 4 に情報取得要求を送信する．各サイトでジョブを実行中のユーザにのみ情報を提供するというポリシを設定している場合，Site 1 と Site 4 において User A のジョブが実行中であるため，そのジョブの実行状況に関する情報を IC が提供する．IA がこれらの情報をまとめて User A に提供する．

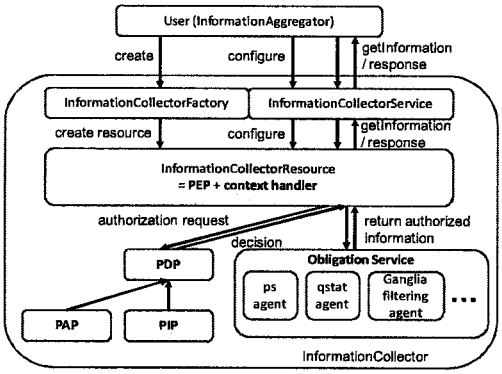

図 7 InformationCollector システムアーキテクチャ

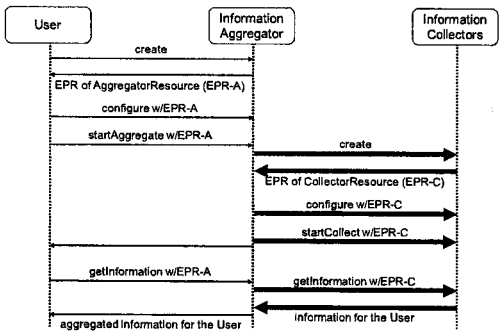

図 8 情報取得プロセス

3.1 InformationCollector アーキテクチャ

提案システムでは，IC でのみ認可決定手続きを行っており，他のユーザの実行状況等の認可されていない情報がサイト外に流出しないように設計されている．**図 7** に IC のシステムアーキテクチャを示す．IC は XACML の認可モデルに基づいており，WSRF インタフェースモジュール (InformationCollectorFactory, InformationCollectorService)，ステートフルサービスインスタンスモジュール (InformationCollectorResource(ICR))，認可決定関連モジュール (PDP, PAP, PIP)，Obligation Service モジュールで構成される．ICR は PEP と context handler の役割も担っている．

WSRF インタフェースモジュールでは，情報サービスの受付と，その後のサービスオペレーションの窓口となる．ICR は各情報サービス要求のサービスインスタンスであり，各要求に関する情報を管理すると共に，認可決定関連モジュールへの認可決定要求と，その認可決定に基づく Obligation Service モジュールからの認可情報の取得を行う．Obligation Service モジュールでは，認可された情報を提供する．本システムでは，様々な情報を取得するために各情報に対応したエージェントを用意し，各エージェントが動的または静的な情報を提供する．図 7 では，`ps` や `w` など UNIX コマンドから得られる結果を提供するもの，TORQUE[7] や GridEngine[8] 等のバッチキューイングシステムからキューの情報を提供するもの，Ganglia のモニタリング情報を提供するものを想定している．各サイトの管理者は，この Obligation Service モジュールに開示したい情報をあらかじめ登録することができる．ユーザは登録されている情報の中から取得したい情報を指定して，認可されればその情報を得ることができる．

3.2 情報取得プロセス

IA と IC の連携による情報取得プロセスを**図 8** に示す．ユーザは IA に対して create で情報取得受付要求を送ると，そのインスタンスである InformationAggregatorResource(IAR) へのポインタである EPR (EPR-A) が送られる．User は EPR-A を用いて configure コマンドで取得したい情報セットを設定する．その後 startAggregate を実行すると，IA から IC に対して情報収集処理が開始される．IA から関連する複数の IC に対して並行して同時に手続きが進められる．IA の場合と同様に，create で情報取得受付要求を送り，そのインスタンス ICR へのポインタ EPR-C を受け取ると，それを用いて configure 要求を送信する．その後，startCollect によりサイト内での情報収集処理を開始する．各サイトでの情報収集が開始されると，User は getInformation または subscribeAggregate コマンドにより，情報取得要求を IA に送信する．IA は IC に対して getInformation または subscribe により，各サイトで認可された情報を統合して提供する．ここで，subscribeAggregate/subscribe は，WS-Notification[9] に基づき通知ベースで情報を提供するものである．

3.3 情報サービスシステムの応用

情報サービスシステムの応用として，グリッドコアロケーションシステムとの連携が考えられる．我々は分散する多様な資源をコアロケーションするためのフレームワーク，GridARS[10] を提案している．GridARS では，WSRF に基づくインタフェースを提供しており，事前予約でユーザが要求した資源を確保し，予約時刻になると確保した資源が自動的に利用可能となる．GridARS の GRS (Global Resource Scheduler) では予約に関する情報を保持しており，GRS の予約サービスインスタンスへのポインタ (EPR) と共に情報開示要求を情報サービスシステムに送れば，情報サービスシステムがユーザの権限で GRS の予約サービスインスタンスにアクセスして予約情報を取得し，予約資源に関する情報を予約時刻に提供することもできる．

3.4 情報サービスシステムの実装

情報サービスシステムの実装では，WSRF の参照実装である Globus Toolkit 4 (GT4) と，サンマイクロシステムズが提供している XACML バージョン 1.x に基づく参照実装[11] を用いた．

GT4 の提供する GSI (Grid Security Infrastruc-

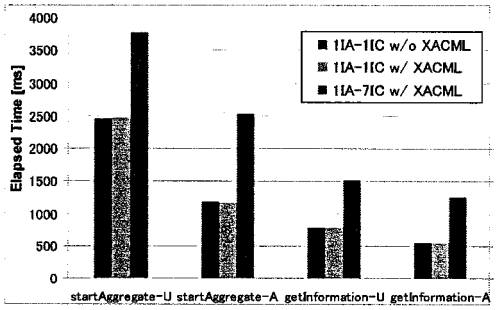

図 9 情報収集開始処理および情報取得の所要時間

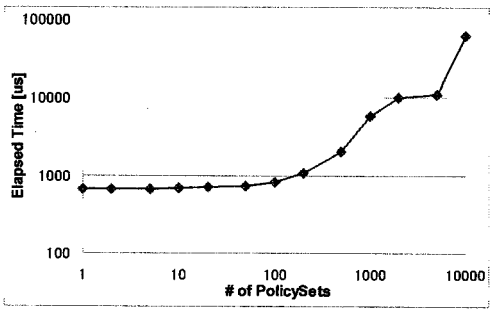

図 10 XACML 認可判定の所要時間

ture) を用いたユーザの認証，grid-mapfile を用いたグローバルユーザ名とローカルサイトのユーザ名のマッピング情報の取得，各手続きと通常の情報取得時および通知ベースでの情報取得時の SSL によるセキュアな通信を実現している．また権限移譲機能により，ユーザは IA とのやり取りのみで，そのユーザの証明書を IC に送ることができる．

また，XACML の参照実装では指定されたポリシファイルとポリシ評価関数から認可判定を行う．本情報サービスシステムでは，ポリシ検索モジュールを別途実装し，認可判定の度に指定したディレクトリ下のポリシファイルとポリシ評価関数を読み込んで PDP を構築する．これにより，GT4 のコンテナの再起動などをすることなく，容易にポリシおよびポリシ評価関数を追加・修正することができる．

4. 予備実験

情報サービスシステムおよび XACML の認可決定に要するオーバヘッドを調査するため，2 つの予備実験を行った．

4.1 情報の収集・取得に関する所要時間

1 つめの実験として，情報サービスシステムを利用して情報を取得する際に要するオーバヘッドを，IA と IC が 1 対 1 の場合と 1 対 7 の場合で比較した．実験では，IA と IC は同じクラスタ内に設置した．各ホスト間の遅延は約 40[us] で，1 つの 1[Gbps] のスイッチに接続されている．全ホストの構成は Pentium 4 2.8[GHz]，1[GB] メモリ，CentOS4.3 となっている．

図 9 に情報収集開始処理 (startAggregate) および情報取得処理 (getInformation) の所要時間を [ms] で示す．横軸には startAggregate と getInformation をユーザ側で測定した時間 (-U)，IA 側で測定した時間 (-A) を示している (図 8 参照)．また、3 つの測定結果 1IA-1IC w/o XACML，1IA-1IC w/ XACML，1IA-7IC w/ XACML は，IA と IC が 1 対 1 で XACML での認可判定がないものとあるもの，1 対 7 で XACML での認可判定があるものの結果を示している．グラフでは 10 回測定したうちの最短値を採用した．

この測定では，以下の 2 つのポリシをポリシセットとして認可判定を行った．

- ユーザ証明書の DN が gridmap-file 内のローカルユーザアカウントへのマッピングがあるか
- 予め指定した時刻内の問い合わせか

図 9 より，1 対 1 のケースで認可判定の有無で比較すると，認可判定に要するオーバヘッドがほとんどないことが分かる．実験では，認可判定を含む IC 内での処理に要する時間を同時に測定したが，いずれも 0.8-3[ms] 程度であり，情報の収集，取得に関するオーバヘッドは SOAP 通信のためのデータ型の変換といった WSRF/GSI に関する手続きでほぼ占められていた．1 対 1 と 1 対 7 の場合の比較では，1 対 7 の方が WSRF での 7 つの IC への手続きのオーバヘッドのため所要時間が長くなってしまう．しかしながら，IA から IC への手続きは並行して行っているため，IA 側で測定した値で比較すると，IC の数に比例することなく startAggregate では 2.16 倍，getInformation では 2.31 倍程度にとどまっていた．以上から，XACML での認可決定に要する時間は無視できるものであり，複数 IC から情報を集める場合でもその所要時間は IC 数に比例することなく許容できることが示された．

4.2 XACML での認可決定所要時間

次に，XACML での認可決定に際して，ポリシ数に対する所要時間の変化を調査した．その結果を**図 10** に示す．縦軸には所要時間を [us] で，横軸にはポリシセット数を示す．ポリシセットは 4.1 節で用いた 2 つのポリシを 1 セットとして，10000 セットまで変化させた．グラフの値は 10000 回の測定の平均値である．

図 10 から，ポリシセット数が 100 までは 1[ms] 以下，5000 までは 10[ms] 前後となっており，いずれも 4.1 節での全体の処理時間と比較すると無視できる．一方，10000 のときは 62[ms] 程度と急激に長くなっていたが，依然許容範囲であると言える．従って，XACML による認可判定のオーバヘッドはポリシ数が多い場合も全体の処理時間に対して十分短いことが示された．

5. 関連研究

Globus Toolkit 4 (GT4) では，WSRF に基づくグリッドシステムにおいて，サービスインスタンスであるリソースへのアクセスの認可のため，XACML に基づく認可フレームワークを提案している[12]．このフレームワークでは，認可の粒度が WSRF におけるリソース単位であり，本情報サービスシステムより粒度が大きい点が異なる．本研究で扱う個々の情報をリソースにより提供することも可能であるが，その場合はリソース量が膨大になり，コンテナの負荷が高くなってしまう．また，この機能は GT4.2 以降での公開予定であり，現状では利用することはできない．

MonALISA[13] は Ganglia や MRTG[14] 等のモニタリングツールで収集された情報をレポジトリに格納し，クライアントインタフェースから提供する．認証は行うものの，開示する情報の認可は行っていないため，同一レポジトリにアクセスするユーザは全ての情報にアクセス可能である．レポジトリおよびレポジトリの情報を用いた各サービスへのインタフェースは，ウェブサービスに基づいている．

Inca[15] は TeraGrid[16] におけるユーザレベルグリッドモニタリングシステムであり，エージェントによりユーザの権限で情報を収集・提供する．Inca では，複数サイトの情報を集中管理しており，サイト外に収集した情報が流出する点，サイト毎に様々なポリシを定義して細粒度のアクセス制御を行わない点，WSRF インタフェースではなく Web アプリケーションとして実装されている点で本研究と異なる．

6. まとめと今後の課題

本研究では，認可モデルとポリシ記述言語の標準である XACML を用い，各サイト毎に情報開示ポリシを定義して，各ユーザからの細粒度のアクセス制御を可能にする，情報サービスシステムを提案した．本システムは WSRF に基づき，各サイトで情報を収集し，認可した情報を提供する InformationCollector(IC) と，複数 IC からの情報を統合してユーザに提供する InformationAggregator からなる．

また，GT4 の WS Core とサンマイクロシステムズの提供する XACML の参照実装を用いて，提案する情報サービスシステムのプロトタイプを開発し，予備実験を行った．実験から，(1) 情報収集では WSRF/GSI のオーバヘッドが大きく，XACML による認可決定の時間は無視できる，(2) 複数 IC から情報収集する場合も，手続きを並行して行うためその所要時間は許容できる，(3)XACML のポリシ数が多い場合も，判定に要する時間は全体の処理時間に対して十分小さいことが確認できた．

今後は，多様なアクセス制御ポリシの適用，情報サービスシステムと GridARS グリッドコアロケーションフレームワークとの連携と，ユーザインタフェースの構築をすすめる．

謝辞 本研究の一部は，文部科学省科学技術振興調整費「グリッド技術による光パス網提供方式の開発」による．

参考文献

1) Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems, *IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779*, pp.2–13 (2005).
2) OASIS Web Services Resource Framework (WSRF) TC: Web Services Resource 1.2 (WS-Resource) Committee Specification (2006).
3) Ganglia Monitoring System: `http://ganglia.info/`.
4) OASIS: eXtensible Access Control Markup Language (XACML) Version 2.0 (2005).
5) OASIS: `http://www.oasis-open.org/`.
6) OASIS: Assertions and Protrocols for the OASIS Security Assertion Markup Language (SAML) V2.0 (2005).
7) TORQUE Resource Manager: `http://www.clusterresources.com/resource-manager.php`.
8) Grid Engine: `http://gridengine.sunsource.net/`.
9) OASIS Web Services Notification (WSN) TC: Web Services Base Notification 1.3 (WS-BaseNotification) Public Review Draft 02 (2005).
10) Takefusa, A., et al.: GridARS: An Advance Reservation-based Grid Co-allocation Framework for Distributed Computing and Network Resources, *Proc. of 13th Workshop on Job Scheduling Strategies* (2007).
11) Sun's XACML Implementation: http://sunxacml.sourceforge.net/.
12) Lang, B., Foster, I., Siebenlist, F., Ananthakrishnan, R. and Freeman, T.: A Multipolicy Authorization Framework for Grid Security, *Proc. of NCA06* (2006).
13) Legrand, C., et al. : MonALISA: An Agent based, Dynamic Service System to Monitor, Control and Optimize Grid based Applications, *Proc. of CHEP2004* (2004).
14) MRTG Monitoring Tool: `http://www.mrtg.org/`.
15) Smallen, S., Olschanowsky, C., Ericson, K., Beckman, P. and Schopf, J.M.: The Inca Test Harness and Reporting Framework, *Proc. of the IEEE/ACM SC2004 Conference* (2004).
16) TeraGrid: `http://www.teragrid.org/`.