

aggregating information on resources of the Grid in a real-time, and present a prototype system.

P2P 技術を応用したグリッド資源発見機構の提案

古久保 慶^{†1,†3} 市川 昊 平^{†2,†3}
伊達 進^{†3,†1} 下條 真 司^{†4,†3}

グリッドは大規模かつ複雑な計算を必要とする分野で広く利用されている。グリッドを構成する計算機資源は、複数組織の多種多様な計算機資源からなり、その利用状況は常に変化するため、グリッドではユーザの要求に応じて適した計算機資源を発見・選択するサービスが必要不可欠である。しかし、既存の資源発見手法では、事前の静的な設定により階層的なネットワーク構造を持ったサーバ群に資源情報を集約・管理させるものが多く、動的な資源情報をリアルタイムに取得・提供できない、あるいはグリッドの構成自体の動的な変化に対応できないという問題が存在する。本研究では、物理的なネットワーク構造に依存せず論理的なネットワークを動的に構築可能な P2P 技術を応用するグリッド資源発見機構を模索している。本稿では、P2P 技術の 1 つである PIAX を用いてプロトタイプ構築した資源発見機構について報告する。

A proposal of P2P-based Resource Discovery Mechanism for the Grid

KOKUBO KEI,^{†1,†3} KOHEI ICHIKAWA,^{†2,†3}
SUSUMU DATE^{†3,†1} and SHINJI SHIMOJO^{†4,†3}

Recently, Grid computing is widely used in the fields where need large and complicated processing. A Grid environment is composed of heterogeneous resources and the availability of these resources changes over time. In such environment, users require a resource discovery mechanism that allows the users to find resources suitable for executing their jobs. However, most of existing resource discovery systems gather information on the resources of the Grid through a hierarchical structure which is composed of a couple of servers deployed based on a static configuration in advance. This mechanism does not support real-time and dynamic information acquisition and may be not useful to the dynamic changes in the Grid. We have therefore been investigating and exploring the feasibility of a resource discovery mechanism employing P2P technologies. In this paper, we propose a P2P-based strategy of discovering and

1. はじめに

近年、地理的に分散した複数の組織からプロセッサやストレージ、データといった資源を集約する技術としてグリッドコンピューティング (以下、グリッド)¹⁾ が広く利用されている。グリッドは、ユーザに仮想的に大規模な計算環境を提供する。また、グリッドにより接続される各組織は、同じ目的を持った仮想組織 (VO: Virtual Organization)¹⁾ を形成する。これにより、単一組織内の利用可能な計算機資源だけでは計算することが困難であった大規模計算が可能となりつつある。現在、グリッドは様々な分野で利用されており、特に大規模かつ複雑な計算を必要とする物理学²⁾ や創薬³⁾⁴⁾、医学⁵⁾ などの科学分野で顕著な利用実績がある。

グリッドが複数組織間で異なる計算機資源 (PC, クラスタシステム, スーパーコンピュータなど) を共有するため、各計算機資源に導入されている OS, ライブラリ, CPU 性能などが異なり、計算環境は均質ではない。また、利用可能な計算機資源の構成や計算機資源の CPU やメモリなどの資源の利用状況は常に化する。そのため、グリッドではユーザのジョブに適した計算機資源を収集・発見するサービスが必要不可欠である。ユーザは、計算機資源の OS やノード数などの静的な情報、あるいは CPU 利用率や利用可能メモリ容量などの動的な資源情報をクエリ条件として資源発見サービスを提供するサーバに送信し、ユーザが求める計算機資源群を発見・選択可能とする技術を必要としている。

Globus MDS⁶⁾ をはじめとする既存のグリッド資源発見サービスは、各計算機資源から情報を収集するインデックスサーバを事前の設定により階層的に接続させる。階層の下から上へと計算機資源の情報を定期的に収集、集約することによって、資源情報を管理する。

†1 大阪大学大学院情報科学研究科

Graduate School of Information Science and Technology, Osaka University

†2 関西大学ソシオネットワーク戦略研究機構

Research Institute for Socionetwork Strategies, Kansai University

†3 大阪大学サイバーメディアセンター

Cybermedia Center, Osaka University

†4 独立行政法人情報通信研究機構

National Institute of Information and Communications Technology

この場合、サーバ間の接続の設定は予め決定され、静的である。しかし、このような既存のグリッド資源発見手法には、以下の2つの問題点がある。第一に、グリッド環境が大規模化すると、グリッドの計算機資源の構成は組織の参加・脱退によって動的に変化する。そのため、事前の設定による静的な構造では、グリッドの構成自体の動的な変化に対応し難い。第二に、計算機資源のCPU利用率やメモリ利用量などの情報は常に動的に変化し、計算機資源は突発的な障害やメンテナンスによって常に利用可能であるか分からない。そのため、階層的なネットワーク構造を持つインデックスサーバに資源情報を集約させていく手法では、リアルタイムに計算機資源の現在の状態を表す情報を取得できない。計算機資源の現在の情報を取得できない場合、古い情報に基づいてジョブを実行してしまい、他のユーザのジョブが既に投入されている場合などが考えられる。

本研究では、グリッドを構成する計算機構成の動的な変化、資源情報の動的な変化に対応するためには、静的なネットワーク構造ではなく、計算機資源の構成によって動的かつ自由にネットワーク構造を変化可能なアーキテクチャが必要であると考えられる。そこで、階層的なサーバによって情報を集約・管理する手法ではなく、P2P技術を応用する手法を模索している。具体的には、P2P技術を適応することで計算機資源間で論理的なネットワークであるオーバーレイネットワークを複数形成し、その複数のオーバーレイネットワークを利用して計算機資源に関する情報を取得する手法について検討する。オーバーレイネットワークを用いることにより、ネットワークを構成する計算機の動的な参加・脱退に対応する。しかし、P2Pネットワークを応用する場合、ネットワーク上において交換される情報は増大しがちである。本研究では、この問題を解決するため、複数のオーバーレイネットワークの検索機能を利用することでユーザの要求を満たす計算機資源を効率的に絞り込み、資源情報検索を高速化することによって、リアルタイムな計算機資源の情報を提供・取得可能な手法の確立を試みる。

以下、2節では、本研究で応用するP2P技術であるPIAXについて述べる。3節では提案する資源発見機構について解説し、プロトタイプの実装について説明する。4節では本資源発見機構の性能と評価し、情報を収集するために要する時間を計測する。最後に5節では、本稿のまとめと今後の課題について整理する。

2. 要素技術

本節では、本研究で応用する要素技術であるP2P技術⁷⁾に関して概説する。特に、P2Pにおいてネットワーク形成に利用される技術であるオーバーレイネットワーク(以下、単にオーバーレイ)に関する説明を行う。その後で、本研究で利用するP2Pプラットフォームで

あるPIAXについて概説する。

2.1 P2P技術

P2Pネットワークにおいては、ネットワーク上にあるPCなどの端末は、ノードまたはピアと呼ばれ、クライアント・サーバ型のモデルのようにサーバが中央集権的にクライアントからの要求を処理するのではなく、ネットワークに参加するすべてのノードが対等の機能を保有する。各ノードは各自で情報を管理し、互いに直接通信可能なネットワークを構成する。このネットワーク上では、各ノードは物理的なネットワーク構成やIPアドレスに依存しない一意のID(ノードID)によって管理され、ノードIDを指定することによって互いに通信可能なネットワークを論理的に形成する。物理ネットワーク上に形成されたこの論理ネットワークをオーバーレイと呼ぶ。

オーバーレイでは、各ノード間には階層構造等の物理的な関係はなく、対等な関係であるため、一部のノードの動的な参加や脱退がネットワーク全体に影響を及ぼすことはない。そのため、ノードが動的に参加・脱退した場合にも自律的にネットワークを再構築可能である。

オーバーレイには大きく分けて構造化オーバーレイと非構造化オーバーレイがある。構造化オーバーレイは、ノード間接続や情報の格納といったオーバーレイ構成時のルールが一定の数学的ルールに従うオーバーレイである。例えば、構造化オーバーレイの代表例である分散ハッシュテーブル(DHT: Distributed Hash Table)は、情報のあるハッシュ値に対応付けるハッシュ関数に基づいて情報の分散やノード間の接続のルールを定めている。各ノードは、ノードIDと近いハッシュ値に関する情報を管理する。また、各ノードが管理するハッシュ値の範囲は、DHTを構成するノードの参加・離脱により自律的に変化する。一方、非構造化オーバーレイは、オーバーレイ構成時のルールを持たず、オーバーレイへの参加順などがランダムな規則で構成される。そのため、オーバーレイ上に情報はランダムに配置されることになり、情報検索は非効率になりがちである。構造化オーバーレイは、オーバーレイの構成に一定の規則を持たせることで非構造化オーバーレイよりも効率的に情報検索が可能である。本研究では、リアルタイムな資源情報の収集を実現するために、構造化オーバーレイを用いる。

2.2 P2Pプラットフォーム: PIAX

本研究では、P2Pを実現する基盤技術としてPIAX(P2P Interactive Agent eXtensions)⁸⁾を用いる。PIAXは、大阪大学を中心として開発されている、エージェントベースのP2Pプラットフォームである。PIAXは、オーバーレイ上に自律的に動作するプログラムであるエージェントを配備することで、オーバーレイ上でのアプリケーションの実装・機能追加を容易に実現可能としている。PIAXでは、ノードは動的にオーバーレイに参加・脱離可能である。

また、オーバーレイ上のエージェントに対してクエリを送信する際、IP アドレスやノード ID などの要求先の資源の位置情報を知る必要がなく、クエリ条件のみによってクエリの送信先を指定することが可能である。また、PIAX は、複数の構造化オーバーレイを用いて P2P ネットワークを形成可能であり、ユーザの用途によってオーバーレイを切り替える機能を実現する。複数のオーバーレイを用いることで、様々なユーザのクエリ条件に対応し、ユーザクエリを満たす資源を絞り込むことで効率的な検索サービスが実現可能である。特に、PIAX が保有するノードを動的に参加・脱退可能な機能や複数のオーバーレイの効率的な検索機能は、1 節で述べた 2 点の問題を解決し、資源検索グリッド環境下での資源発見機構を実装する上で有効であると考えられる。そのため、本研究では基盤技術として PIAX を用いる。

3. 資源発見機構

本節では、提案する資源情報の取得・提供手法について解説する。また、本手法を実装する資源発見機構のプロトタイプに関して詳説する。

3.1 提案手法

本研究では、資源発見・収集機能を保有する PIAX エージェントを設計・実装し、PIAX が構成する構造化オーバーレイネットワークを用いることで効率的な資源情報の取得・提供が可能とする資源発見機構を試みる。図 1 に本資源発見機構の概要を示す。

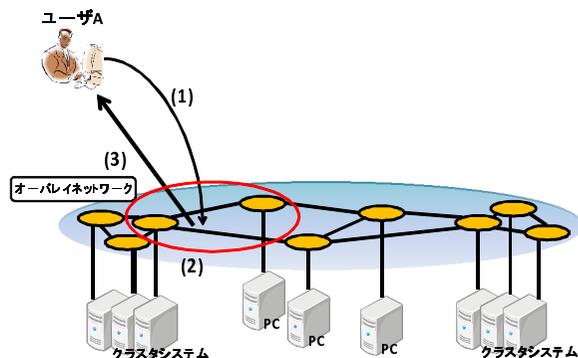


図 1 提案する資源発見機構の概要

本研究では、計算機資源として単一の PC 資源とクラスタシステムが混在する環境を想

定している。各計算機資源は PIAX によって作られる構造化オーバーレイにより接続し、資源発見機構の機能をj提供するエージェントを配備する。クラスタシステムでは、クラスタのマスタのみならずワーカもオーバーレイを構成する。

本機構は以下のように動作する。(1) ユーザは、CPU の計算性能 (flops) やクラスタシステムのノード数といった各計算機資源において静的な情報と、CPU 利用率や使用可能メモリ容量といった動的な情報を組み合わせてユーザクエリをj発行する。(2) 本機構はこのクエリを解釈し、クエリ条件に合致する計算機資源上のエージェントにのみクエリを送信する。(3) 計算機資源上のエージェントは、送信されたクエリを受け取り、計算機資源の資源情報を取得しユーザに返信する。このようにクエリの送信をユーザクエリ条件に合致する計算機資源に限定することで、高速に資源情報を取得することが可能である。本研究では、このような絞り込み機能を実現するため LL-net と DOLR の 2 つの構造化オーバーレイをj応用する。

図 2 に LL-net と DOLR の概略を示す。LL-net⁹⁾ は二次元座標軸上に、P2P ネットワーク上の各ノードを仮想的に割り付け、ある特定の座標範囲内のノードに限定して、クエリを送信する機能を持つ。通常、LL-net はノードの地理的な位置情報のマッピングに使用され、緯度・経度を二次元座標軸として採用する。しかし、必ずしも緯度・経度情報を座標軸に採用しなければならないわけではなく、任意の情報を座標軸に採用可能である。本稿では、試験的にユーザクエリとして一般的な CPU の計算性能とクラスタシステムのノード数を座標軸にとり、各計算機資源を LL-net オーレイネットワーク上にマッピングし、クエリ送信の絞り込みに利用する。

DOLR¹⁰⁾ は属性名・属性値という二つの情報のペアを各ノードに割り当てることができ、指定した属性名・属性値に合致するノードに限定して、クエリを送信する機能を持つ。本稿では各計算機資源に対し、クラスタ名や VO 名を属性として割り当て、これらの情報をユーザクエリに含める。本稿では、これら 2 つのオーバーレイネットワークを用いて、検索対象となる計算機資源を絞り込み、高速な検索を実現することを試みる。

3.2 プロトタイプの実装

本研究では、プロトタイプ実装として、クラスタシステムのノード数や CPU 性能、クラスタ名や VO 名などの静的な情報を基づくユーザクエリ条件により、CPU 利用率や利用可能メモリ容量などの動的な情報を収集する機構を設計する。また、クエリの送信先を絞り込むために、LL-net と DOLR の二つのオーバーレイを用いてユーザクエリ条件に合致する計算機資源の情報のみをユーザに返信させるように設計する。

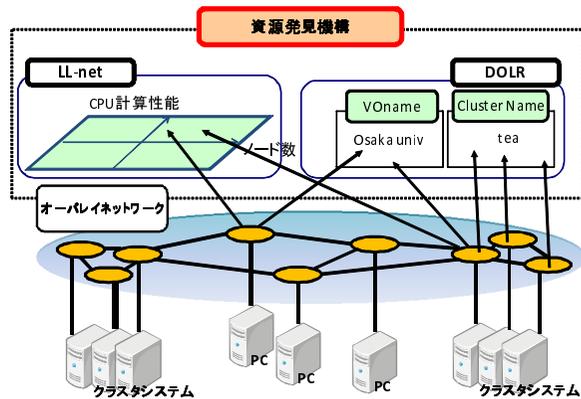


図 2 LL-net と DOLR の概略図

本研究で構築したプロトタイプの詳細を図 3 に示す。本プロトタイプシステムは、ユーザインターフェース部 (ClientPeer : CP)、資源情報取得部 (ResourceInfoGetter エージェント : RIGA)、クラスタ内部情報集約部 (Master エージェント : MA) からなる。ユーザは、ClientPeer を通してクラスタ名、VO 名やノード数、CPU 計算性能を指定してクエリを送信し、ResourceInfoGetter エージェントや Master エージェントから得られた結果から利用する計算機資源を選択する。

ResourceInfoGetter エージェントは各計算機資源上に配備され、ユーザのクエリに応じて、資源情報を返答する。クラスタのマスタノード上には ResourceInfoGetter エージェントに加えて、Master エージェントが配備され、クラスタのワーカーノードからの資源情報を集約し、ユーザに返答する。全てのワーカーノードがユーザに直接返答を返すとオーバーレイ上に流れる情報が増大するため、ワーカーノードの情報は Master エージェントが集約して、返答する。例えば、Master エージェントは CPU の計算性能として、ワーカーの計算性能の平均を返答する。

Master エージェントとクラスタシステムに属さない単一の PC 資源上の ResourceInfo-Getter エージェントは、自身のノード数と CPU 計算性能に基づいて、自らを LL-net 上にマッピングし、ClientPeer より直接クエリを受ける。各エージェントには、DOLR により ClusterName、VOname の二つの属性が与えられる。ClusterName はクラスタ名であり、VOname は所属する VO 名である。ClusterName を持つ計算機資源はクラスタシステムで

あり、それ以外は単一の PC 資源である。クラスタを構成する計算機資源はすべて同一の ClusterName の属性値を持つ。

具体的な動作例を以下に示す。図 3 のオーバーレイは単一の PC 資源 PC a、PC b とクラスタシステム Cluster A、Cluster B により構成されている。そこにユーザ端末 C がオーバーレイに参加している。Cluster A および Cluster B は、VO 名が Osaka Univ、CPU の計算性能がそれぞれ 4.6GFlops、6.0GFlops、かつノード数がそれぞれ 40、20 のクラスタシステムであると仮定する。以下、図 3 にユーザが資源情報を取得するまでの流れを示す。(1) ユーザ端末 C の ClientPeer により例えば「ノード数が 20~40 で、各ノードの CPU の計算性能が 3.0GFlops~6.0GFlops、VO 名が Osaka Univ」というクエリを発行する。(2) ユーザクエリは LL-net や DOLR のオーバーレイ上の計算機資源の中から、クエリ条件を満たす計算機資源 Cluster A および Cluster B にのみクエリを送信する。(3) クエリ条件を満たす計算機資源 Cluster A および Cluster B のマスタの Master エージェントはクエリを受けると DOLR の ClusterName に基づき、ワーカーの ResourceInfoGetter エージェントに対してさらにクエリを送り、クラスタのワーカーの ResourceInfoGetter エージェントは、クエリを受けると自身の資源情報を ResourceInfo オブジェクトとして計算機資源 Cluster A および Cluster B の Master エージェントに返す。(4) Master エージェントは、ワーカーの情報を集約して ClusterInfo オブジェクトとして ClientPeer に返す。(5) ユーザは ClientPeer が受け取った情報を取得する。

以上のフローによってユーザは資源情報を取得することが可能である。ユーザが ClientPeer から受け取る情報は、(3) の際に取得される情報である。また、本資源発見機構では Globus MDS のように計算機資源の資源情報をインデックスサーバに集約させる手法ではなく、クエリに返答する瞬間にエージェントが計算機資源から資源情報を取得する手法を採用。よって、リアルタイムに資源情報を取得することが可能であると考えられる。次節でその評価を行う。

4. 評価実験

本節では、提案した資源発見手法を評価するため、本資源発見機構のプロトタイプの性能・有効性をそれぞれ評価する。本論文で提案した資源発見手法はグリッドの構成自体の動的変化への対応、CPU 利用率などの動的な情報のリアルタイムな取得を実現するものである。前者に関しては本研究で用いた PIAX によって担保される機能であるため、本稿ではその評価に関する議論は省略する。本節では後者のリアルタイムな情報取得に関する評価に関して報告する。評価として、実際にプロトタイプを実装し、計算機資源上にオーバーレイを

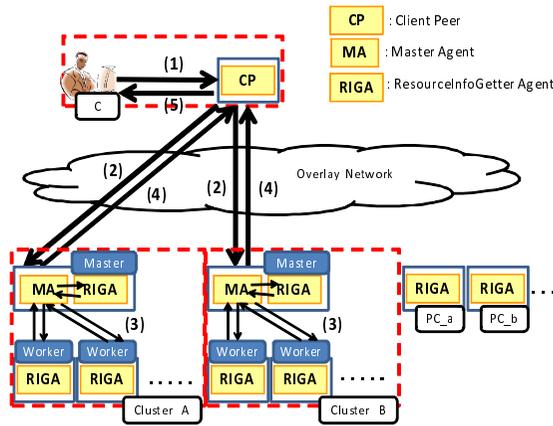


図 3 プロトタイプとその動作例

構成し、エージェントを用いて資源情報を取得するのに要する時間を処理時間として測定する。評価環境として 100Mbps の同一 LAN 内における 40 ノードの計算機資源を用いた。計算機資源の環境構成は以下の表 1 の通りである。ここで処理時間とは、ユーザがクエリを送信してからユーザが返答を取得するまでの時間である。また、絞り込み検索機能の評価として LL-net による絞り込み機能により検索範囲を絞り込んだ場合に短縮される時間を測定した。測定は 100 回試行を繰り返し、その平均を取った。取得する情報は、リアルタイム性が求められる CPU 利用率や利用可能メモリ容量などの動的な情報とした。

表 1 計算機資源の環境構成

	OS	CPU	メモリ	ネットワーク
計算機資源	CentOS4.4	PentiumIII 1.4GHz	1GB	100BASE-T

4.1 情報取得に要する時間の評価

資源発見機構のオーバーレイを構成する計算機資源の数を 10, 20, 30, 40 と変化させた場合の資源発見に要する時間を計測した。簡単のため計算機資源はすべて単一の PC 資源であり CPU の計算性能、所属する VO などすべて等しいものとした。図 4 にその結果を示す。

計算機資源の数が 40 である場合、処理時間は 152 (msec) である。これはユーザがクエ

リを送信してから、情報を取得するまでの時間である。したがって、40 台の計算資源に対してクエリを送信する際、取得される資源情報には最大 152 msec の遅延が生じる可能性があることを意味している。この処理時間は現状の MDS と比較すると十分に高速であると言える。文献¹¹⁾によると、現状の MDS はデフォルトの設定状態であると資源情報の取得に数十秒の時間が掛かり、キャッシュを有効にするなど最適なチューニングを行った上でようやく情報取得時間は 500msec に短縮されるとある。ただし、ここで MDS から得られる資源情報はキャッシュされた古い情報に基づくものであり、実際の情報取得に生じる遅延はキャッシュ更新間隔に依存してより大きくなる。したがって、152 msec という遅延は現状の MDS に比較すれば十分に小さいと言える。しかし、10, 20, 30, 40 と計算機資源の数を増加させた場合の処理時間の増加量は指数的ではないものの増加するため、計算機資源の数が膨大になると高速に情報を取得することは困難であると考えられる。

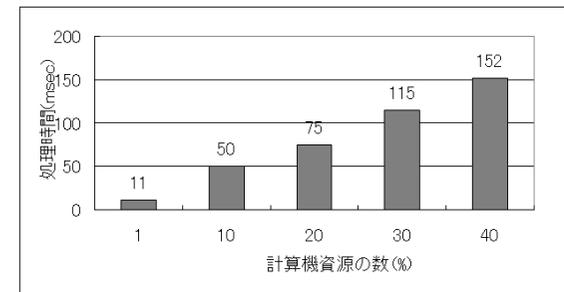


図 4 ユーザクエリ条件を満たす計算機資源の割合と処理時間の関係

4.2 LL-net による絞り込み機能の有効性の評価

LL-net による絞り込み機能の有効性を評価するため、20 台の計算機資源を用いて評価実験を行った。計測方法としては、初めに図 5 のように 20 台の計算機資源をそれぞれクラスシステムと仮定し、ノード数と計算性能をランダムに選び LL-net の二次元座標上に配置する。その上でユーザクエリ条件を満たす計算機資源の数の割合を 100% から 20% まで 20% ずつ減少させた。具体的には図 5 の A の領域 (CPU 性能 36 以下、ノード数 36 以下) が対象となるクエリを送信すると、全体の 40% が対象となり、B の領域 (CPU 性能 18 以下、ノード数 18 以下) を対象とすると、全体の 20% が対象となるように、LL-net 上の資源配置を予め工夫しておき、クエリの対象計算資源を 20% ずつ減少できるようにしておいた。

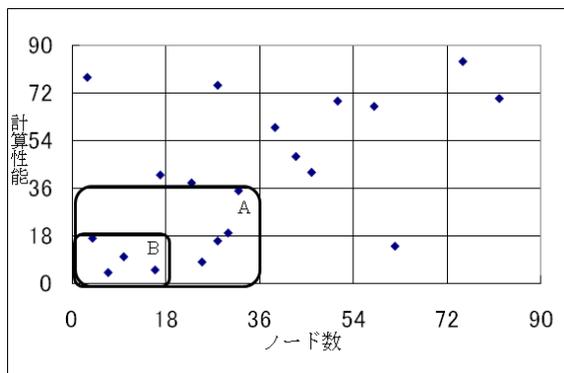


図 5 各計算機資源の LL-net の二次元座標上での分布図

図 6 は計測結果をまとめたものである。この結果から、ユーザクエリ条件を満たす計算機資源のみから資源情報を取得することで、資源情報を取得するために要する処理時間が短縮されていることがわかる。また、20 ノード全ての計算機資源が応答するときと比較し、20% の計算機資源が応答する際には約 61% の処理時間の短縮効率が得られることが判明した。これより、絞り込み機能を利用することにより、全ての計算機資源に対してクエリを送信した場合に比較してより高速に資源情報を取得可能であることがわかる。

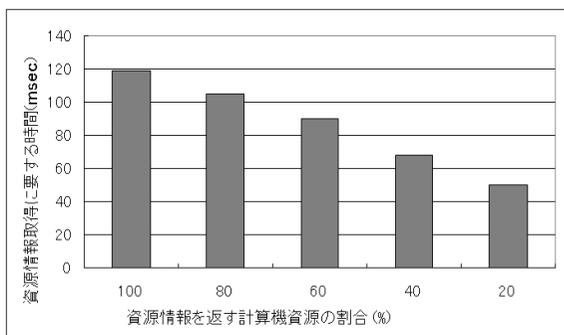


図 6 ユーザクエリ条件を満たす計算機資源の割合と処理時間の関係

5. まとめと今後の課題

本研究では、既存の資源収集サービスにおける資源発見手法がグリッドの動的な変化に対応できないという問題とリアルタイムに資源情報を取得し難いという問題に着目し、問題を解決するために P2P 技術を応用したグリッド資源発見手法を試みた。具体的には、P2P 基盤技術として PIAX を採用し、計算機資源間で複数のオーバーレイを構成し、オーバーレイ上に配備されたエージェントにより情報を取得する手法を提案、報告した。PIAX が提供するオーバーレイにより自律的にネットワークを再構築しグリッドの動的な変化に対応する。また、複数のオーバーレイを応用し、ユーザクエリ条件を満たす計算機資源のみから資源情報を取得する手法を模索した。本稿では、計算機資源のノード数やノードの計算性能などの静的な情報を基に、CPU 利用率や利用可能メモリ量などの動的な情報を高速に取得することを可能とした。

本稿では本研究の資源発見手法を実現するプロトタイプを実装した。構造化オーバーレイとして LL-net と DOLR を用い、資源発見機構を実現するエージェントを実装した。評価実験から、既存の資源発見サービスである Globus MDS に比べ、数十の計算機資源に対しては十分高速に資源情報を取得可能であり、LL-net の絞り込み検索機能により検索する計算機資源を限定することで、すべての計算機資源を取得する場合に比較して高速に資源情報を取得できることを確認した。

今後の課題としては、本稿のプロトタイプでは、LL-net と DOLR という二つのオーバーレイのみの利用になるが、PIAX は、オーバーレイを追加することで、その他のオーバーレイを実現可能である。そこで今後は、ユーザの資源情報検索に対する要求をより調査し、それに沿った新たなオーバーレイを選択・構築することを考えている。評価実験に関しては、より詳細な評価が必要であると考え。具体的には、まずより多くの計算資源を用いた大規模な環境における評価が必要であると考え。本稿では同一 LAN 内の計算機資源に対して評価実験を行ったが、広域環境での評価実験が必要であると考え。また、本稿では LL-net の評価のみを扱ったが、DOLR の絞り込み検索機能の評価も必要であると考え。さらに、オーバーレイより取得する情報量と処理時間との関係を評価する必要があると考え。

謝辞 本研究は、文部科学省特定領域研究 (18049050) の研究助成および財団法人国際コミュニケーション基金の研究助成を受け実施した。ここに記して謝意を表す。また、PIAX について技術的なご助言をいただいた大阪大学大学院情報科学研究科 石 芳正 特任研究員に謝意を表す。

参 考 文 献

- 1) Ian, F., Carl, K. and Steven, T. : The Anatomy of the Grid: Enabling Scalable Virtual Organizations , *International Journal of High Performance Computing Applications*, Vol.15, No.3, pp.200–222 (2001).
- 2) Campana, S., Barberis, D., Brochu, F., De S., Donno, A. and et al: Analysis of the ATLAS Rome production experience on the LHC computing grid, e-Science and Grid Computing, 2005. First International Conference on., pp.8–89 (2005).
- 3) Andrew, C., Ian, F. and Dean, G. : Grid technologies empowering drug discovery, *Drug Discovery Today* , Vol.7, No.20, pp.176–180 (2002).
- 4) Marshall, J. Levesque, Kohei, I., Susumu, D. and Jason, H. Haga : Bringing Flexibility to Virtual Screening for Enzymatic Inhibitors on the Grid, in Proceedings of The 9th IEEE/ACM International Conference on Grid Computing (Grid 2008), pp.201–208, (2008).
- 5) BIRN : Biomedical Informatics Research Network, <http://www.nbirn.net/>.
- 6) Karl, C., Carl, K., Steven, F. and Ian, F. : Grid Information Services for Distributed Resource Sharing, 10th IEEE International Symposium on High Performance Distributed Computing, pp.181–194 (2001).
- 7) 江崎浩 : P2P 教科書, インプレス R&D (2001).
- 8) 吉田幹, 寺西裕一, 春本要, 下條真司 : マルチオーパレイと分散エージェントの機構を統合化した P2P プラットフォーム PIAX : A P2P Platform for Integration of Multi-Overlay and Distributed Agent Mechanisms , *情報処理学会論文誌* , Vol.49, No.1, pp.402–413 (2008).
- 9) 金子雄, 春本要, 福村真哉, 下條真司, 西尾章治郎 : ユビキタス環境における端末の位置情報に基づく P2P ネットワーク , *情報処理学会論文誌*, Vol.46, No.18, pp.1–15 (2005).
- 10) Dabek, F., Zhao, B., Druschel, P., Kubiatowicz, J. and Stoica, I. : Towards a common API for structured peer-to-peer overlays , *Peer-to-Peer Systems II : Springer Lecture Notes in Computer Science*, Vol.2735, pp.33–44 (2003).
- 11) 村木 健介, 川崎 康博, 水谷 泰治, 伊野 文彦 and 萩原 兼一 : 短時間処理向け資源管理を実現するための Globus Toolkit の性能評価, 情報処理学会研究報告. ハイパフォーマンスコンピューティング, Vol.2735, pp.79–84 (2005).