

高品質分散実行環境のための 計算・ネットワーク資源のグローバルスケジューリング手法

竹房 あつ子^{†1} 中 田 秀 基^{†1}
工 藤 知 宏^{†1} 田 中 良 夫^{†1}

高品質分散実行環境の構築に際し、ユーザの要求を満たす高品質資源群を適切に確保しつつ、グリッド上の資源群を有効活用するためには、グローバルスケジューリングが重要な課題となる。本稿では、高品質分散実行環境の構築を目的とした計算・ネットワーク資源のオンライングローバルスケジューリング手法を提案する。提案手法では、ユーザの計算機及びネットワークに関する資源要求に対して、複数資源マネージャからある時間帯における動的資源情報を入手し、その情報から複数の予約プランを組合せ最適化問題を解くことにより作成する。シミュレーションによる評価により、提案手法の有効性を示す。

A Scheme of Global Scheduling of Computing and Network Resources for High-quality Distributed Execution Environments

ATSUKO TAKEFUSA,^{†1} HIDEMOTO NAKADA,^{†1} TOMOHIRO KUDOH^{†1}
and YOSHIO TANAKA ^{†1}

A global scheduling scheme to satisfy requirements from users and resource administrators is one of key issues for constructing a high-quality distributed execution environment based on advance reservations. We propose an online global scheduling scheme for both computing and network resources for that purpose. For each user resource requirement, the proposed scheme solve a combinational optimization problem to determine multiple reservation plans by using dynamic resource information from resource managers. The simulation results show effectiveness of our global scheduling scheme.

1. はじめに

グリッドとネットワーク資源管理技術により、複数管理組織の提供する多様な資源で構成される高品質分散実行環境を動的に構築することが可能になった。特に、帯域が保証されたパスネットワークを資源のひとつとして、計算機やストレージ等の資源と同時に確保し、利用する試みが国内外で複数行われている¹⁾⁻³⁾。

高品質分散実行環境を提供するには、各資源を管理する資源マネージャ(RM)が、性能保証機能と同時確保のための事前予約機能を有することが前提となる。計算資源のみを対象としたマルチクラスタの同時確保では、KOALA⁴⁾ グリッドスケジューラやQBETS⁵⁾ バッチキュー予測サービスのように各ローカルスケジューラのキューの利用状況の取得または予測により、事前予約機能がない状況での同時確保を実現している。しかしながら、これらの手法は複数資源が同時に

利用可能になることを保証するものではないため、資源利用に対する課金が発生する場合はユーザが不利益を被るおそれがある。よって、複数資源の同時確保を確実に実現するためには事前予約が重要な機能であり、我々は、性能保証および事前予約機能を有する資源マネージャと連携し、高品質分散実行環境を構築・利用するためのGridARS資源管理フレームワーク⁶⁾を開発している。GridARSでは、グローバル資源コーディネータ(GRC)が複数RMと連携してユーザの要求する資源群を確保する。

ユーザの要求を満たす高品質資源群を適切に確保しつつ、グリッド上の資源群を有効活用するためには、GRCにおけるスケジューリング(グローバルスケジューリングと呼ぶ)手法が重要な課題となる。ユーザの観点では、指定した性能を保証する資源群を(a)早い時刻に確保する、(b)価格の安い資源群を確保する、(c)価格が高くて品質(可用性等)優先で確保する、等の選択肢がありうる。一方、資源管理の観点では、上記を考慮することに加え、(A)複数資源提供者に対して平等に負荷を割り当てる、(B)省エネや連

^{†1} 産業技術総合研究所 National Institute of Advanced Industrial Science and Technology (AIST)

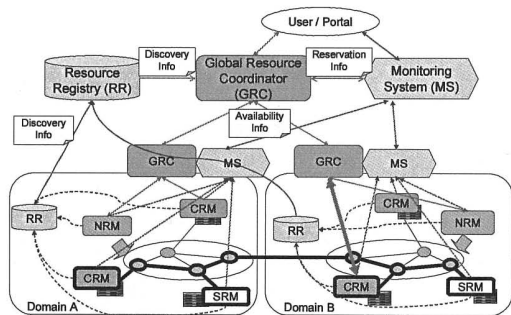


図 1 資源管理フレームワークの概要

携関係から、特定の資源に優先的に割り当てる、(C) ユーザのサービスレベルにより、高レベルのユーザが確実に資源確保できるように配慮する、等の方針が考えられる。これらの多様な指標を考慮しつつ、分散する計算資源および拠点間ネットワーク資源を同時確保するための適切なグローバルスケジューリング手法に関する議論は不十分である。

本研究では、高品質分散実行環境の構築を目的とした計算・ネットワーク資源のオンライングローバルスケジューリング手法を提案し、シミュレーションでの評価により、その有効性を示す。提案するグローバルスケジューリング手法は、ユーザの計算機及びネットワークに関する資源要求に対して、複数 RM から利用可能な資源情報を入手し、複数の予約プランを組合せ最適化問題を解くことにより作成する。評価では、ユーザの要求と資源管理要件に対し、提案手法が有効であり、その予約成功率が高いことを示す。

2. グローバルスケジューリングモデル

2.1 資源管理フレームワークの概要

高品質分散実行環境構築・利用のため、我々は図 1 に示す資源管理フレームワーク GridARS の開発を進めている。本フレームワークは資源管理システム (RMS)⁶⁾、資源レジストリ (RR)、資源モニタリングシステム (MS)⁷⁾ で構成される。図中の Domain A, B は一組織、あるいは連携関係の複数組織を表し、ユーザに対して管理組織を跨った高品質分散実行環境を事前予約に基づいて提供する。

RMS はグローバル資源コーディネータ (GRC) と各資源を管理する資源マネージャ (RM) で構成され、GRC と RM が連携してユーザに高品質分散実行環境を提供する。図 1 における NRM, CRM, SRM は、それぞれネットワーク、計算、ストレージの RM を表す。GRC は複数存在し、複数 GRC が階層的に連携して、あるいは独立した GRC 間で競合して各ユーザの要求する資源を確保する。GRC のうち、グローバルスケジューリング機能 (プランナ) をもつものが、

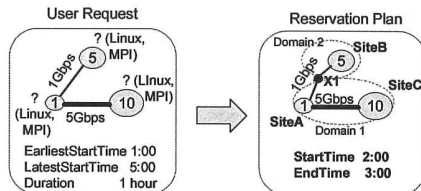


図 2 ユーザの資源要求とプランナによる予約プラン

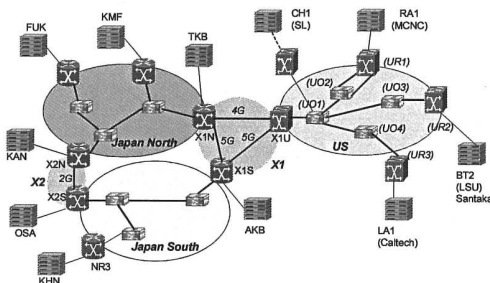


図 3 G-lambda と EnLIGHTened の実験における資源構成

資源確保の予約プランを決定し、下位の GRC または RM に対して資源確保手続きを行う。

RR は、各 RM の所在や各 RM の管理する資源の静的な情報 (総 CPU/コア数、2 拠点間のパスとその容量、など) を、RM または下位の RR から収集し、GRC に提供する。ある時刻に利用可能な動的資源の情報は、GRC が各 RM から直接取得する。我々は RM が商用サービスとして資源提供することも想定しており、その場合、各 RM は動的資源情報をすべて公開することはできないためである。

MS では、予約された資源で構成される高品質分散実行環境の稼働時の資源利用状況をユーザに提供する。この情報をもとに、ユーザは確保した資源の安定稼働の確認や、余剰資源や高負荷資源のための再構成を行う。GRC 及び各 RM は再構成を行うための手段を提供し、具体的な再構成方針はユーザが決定する。

2.2 ユーザの資源要求

我々はこれまでマルチサイトの資源とその間のパスネットワークを同時に確保して高品質分散実行環境を動的に構築し、その上でアプリケーションを実行する試みを行ってきた^{1),2),8)}。事例として、GridRPC および MPI で実装された力学シミュレーション、HD ビデオデータのライブストリーム通信が挙げられる。本研究のグローバルスケジューリング手法では、これらのアプリケーションで必要とされる多様な資源の同時確保のための予約プランの作成を目的とし、予約時刻の異なるワークフロータイプの資源確保は今後の課題とする。

図 2 にユーザの資源要求 (左) と、その要求から GRC のプランナが生成する予約プラン (右) の例を

示す。計算資源の要求では、計算サイト数、各サイトでの CPU (コア) 数、属性情報 (OS, 実行環境等) を、ネットワーク資源要求では要求する拠点間の帯域、レイテンシ、その他の属性情報 (メディアタイプ、可用性等) を指定する。また、各資源に対して時刻の要求を指定することができる。図 2 は、本研究で対象とする同時確保要求であり、全資源の確保時間帯を直接、または EarliestStartTime (*EST*), LatestStartTime (*LST*), Duration (*D*) パラメタにより指定する。*EST* は最も早い開始時刻、*LST* は最も遅い開始時刻、*D* は予約時間幅を示しており、*LST* + *D* が最終的なデッドライン (締切時刻) に相当する。

GRC のプランナは、ユーザの資源要求を受け取ると、図 3 のような資源群から適切な資源群を選定し、予約プランを作成する。図 3 は、G-lambda⁹⁾ および米国 EnLIGHTened Computing プロジェクトによる実証実験 (ELGL 実験) で用いられた資源群を示している²⁾。図 2 右の予約プラン例では、割り当てられた計算資源 SiteA, SiteB, SiteC と計算資源間のネットワーク経路、および資源を確保する開始時刻と終了時刻が決定される。予約プランにおける資源間のネットワーク経路は、実際のルータやスイッチで構成されるネットワークのトポロジを表すものではなく、抽象化されたトポロジを表す。これは、NRM 管理下の具体的なネットワークの構成は一般に公開されないためである。すなわち、1 つのドメイン内のネットワーク資源を利用する場合は 1 つのパスで、複数ドメインを跨る場合は複数のパスで経路が構成される。図 2 では、SiteA-SiteC 間の経路は 1 つのネットワークドメイン Domain1 内のパスで表されるが、SiteA-SiteB 間の経路は X1 を経由して Domain1 と Domain2 の 2 つのパスで構成される。

3. グローバルスケジューリング手法

3.1 スケジューリングの手順と方針

GRC における資源確保手順は以下の通りである。

1. 静的資源情報をあらかじめ RR から取得する。
2. ユーザの資源要求を受け取る。
3. GRC のプランナで資源要求に対する予約プランを複数作成する。この際、プランナは動的資源情報を資源提供候補となる各 RM に問い合わせる。GRC から複数 RM への問い合わせは同時に行うため、この遅延は $O(1)$ となる⁶⁾。
4. プランナの作成した予約プランから、GRC と複数 RM が連携して資源を確保する。
5. 確保が成功した場合は終了、失敗した場合はその情報をユーザに提供し、ユーザは新たな条件で再度資源要求を行う。

本研究では、手順 3 で予約プランを作成するオンライングローバルスケジューリング手法を提案する。グ

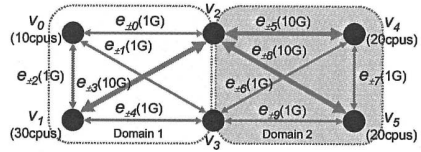


図 4 資源グラフ

ローバルスケジューリングでは、1 節で述べたように、ユーザは資源要求に加え、(a) 時刻優先、(b) 価格優先、または (c) 品質優先の方針で予約プランの作成を求める。一方、GRC の資源管理者は、(A) 複数 RM に対して平等に負荷を割り当てる、(B) 省エネや組織間連携のため、特定の資源を優先し、(C) ユーザのサービスレベルを考慮する、という方針が考えられる。これらの多様な指標を考慮しつつ、計算・ネットワーク資源の予約プランを作成するため、グローバルスケジューリングを組合せ最適化問題として解く。また、それを事前予約に適用する。

3.2 組合せ最適化問題に基づくスケジューリング

図 3 のような実際の資源構成を抽象化し、図 4 に示すような有向グラフ $G = (V, E)$ として資源群を表す。 V は G の点の集合、 E は G の枝の集合であり、グラフの各点 v_q は計算資源サイトまたはネットワークドメイン間の交換点を、各枝 e_{+r} ($= e_{o,p}$) または e_{-r} ($= e_{p,o}$) は各 NRM が提供する資源間のパスを表す。図 4 では、パスは 2 つの NRM ドメイン Domain1 および Domain2 から提供されており、 v_0 を始点、 v_1 を終点とする枝を e_{+0} 、 v_1 を始点、 v_0 を終点とする枝を e_{-0} で表す。グラフの点および枝の括弧内は、提供可能な CPU (コア) 数および帯域であり、それぞれ wc_i ($i \in V$)、 wb_k ($k \in E$) と表す。図 4 では、 v_2 、 v_3 はドメイン間交換点であり、利用可能な CPU は存在しない。同様に、遅延、可用性など、その他の属性情報パラメータを追加することができる。また、提供する CPU および帯域の単位あたりの価値を、それぞれ vc_i ($i \in V$)、 vb_k ($k \in E$) と表す。 wb_k および vb_k は、 e_{+r} 、 e_{-r} で共有する。

次に、資源要求を完全グラフ $G_r = (V_r, E_r)$ で表す。 V_r は必要とする計算資源サイト (点) の集合、 E_r は V_r 間を結ぶ枝の集合を示す。また、 G_r に必要な CPU 数および帯域をそれぞれ rc_j ($j \in V_r$)、 rb_l ($l \in E_r$) とする。

以上のパラメータから、以下の変数を整数計画法で求めることで予約プランを決定する。

$$x_{i,j} \in \{0, 1\} \quad (i \in V, j \in V_r) \quad (1)$$

$$y_{k,l} \in \{0, 1\} \quad (k = (m, n) \in E, m, n \in V, l = (o, p) \in E_r, o, p \in V_r) \quad (2)$$

ここで、資源要求されるサイトに対して選択される計算資源サイトの要素 $x_{i,j}$ は 1、その他のサイトは 0 とする。また、資源要求されるネットワークに対して選

択されるパスの要素 $y_{k,l}$ は 1, 使用されないパスの要素は 0 となる。

3.1 節のスケジューリング方針を (b) 価格優先とした場合, 目的関数と制約条件は次のように表せる。

Minimize

$$\sum_{i \in V, j \in V_r} v c_i \cdot r c_j \cdot x_{i,j} + \sum_{k \in E, l \in E_r} v e_k \cdot r b_l \cdot y_{k,l} \quad (3)$$

Subject to

$$\forall j \in V_r, \sum_{i \in V} x_{i,j} = 1 \quad (4)$$

$$\forall i \in V, \sum_{j \in V_r} x_{i,j} \leq 1 \quad (5)$$

$$\forall i \in V, \sum_{j \in V_r} r c_j \cdot x_{i,j} \leq w c_i \quad (6)$$

$$\forall l \in E_r, \sum_{k \in E} y_{k,l} \begin{cases} \geq 1 & (r b_l \neq 0) \\ = 0 & (r b_l = 0) \end{cases} \quad (7)$$

$$\forall k \in E, \sum_{l \in E_r} r b_l \cdot y_{k,l} \leq w b_k \quad (8)$$

$$\begin{aligned} \forall l = (o, p) \in E_r, \forall m \in V \\ \sum_{n \in V, m \neq n} y_{(n,m),(o,p)} - \sum_{n \in V, m \neq n} y_{(m,n),(o,p)} \\ = \begin{cases} x_{m,o} - x_{m,p} & (r b_l > 0) \\ 0 & (r b_l = 0) \end{cases} \quad (9) \end{aligned}$$

式 (3) の目的関数は, 選択する計算・ネットワーク資源の価格の合計を最小化することを表している。また, (a) 時刻優先の場合も同様に計算する。ただし, 3.3 節でユーザー方針の反映方法を述べる。(c) 品質優先の場合は, 目的関数をユーザーの満たす品質が高くなるように設定する。一方, 資源管理者の方針 (A) および (B) を満たすには, それぞれの方針に従って資源に重みづけし, 別の目的関数を加える。(C) の方針については, 動的資源情報取得時に各ユーザーのサービスレベルに即した情報を取得することで反映することができるため, 別の目的関数の設定は不要である。

式 (4)~(6) は計算資源に関する制約条件, 式 (7)~(9) はネットワーク資源に関する制約条件を示す。式 (4) は, 要求サイト j に割り当てられる実際のサイトは 1 つのみであること, 式 (5) は, 各資源サイト i は資源要求の中で 1 つ以上割り当てられないことを表す。式 (6) は, 各資源サイト i が要求される CPU 数以上の CPU を提供可能であることを示している。

式 (7) は, 要求サイト間のパス l の帯域の確保要求がある場合は, $y_{k,l}$ の総和が 1 以上に, ない場合は 0 になることを表す。すなわち, ドメイン内のパスが選択される場合には 1, 2 ドメインに跨るパスが選択される場合は 2 となる。式 (8) は, 各パス k が要求される帯域以上の帯域を提供可能であることを示す。

式 (9) は, 流量保存則¹⁰⁾ を応用して導く。流量保存則とは, グラフ上の 2 拠点間の経路上の各点において, 流入する流量と流出するが流量の差が, 供給量と等しくなることを示す。すなわち, 流量を f とすると, 始点の供給量は f , 終点の供給量は $-f$, 通過点の供給量は 0 となる。式 (9) では, パス l の帯域確保要求がある場合, 各要求パス $l = (o, p)$ (o はパス l の始点, p はパス l の終点) と計算資源サイトおよび交換点を含む資源上の各点 m に対して, 流量 $f = 1$ として流量保存則を適用する。すなわち, m が l の始点であれば, 式 (9) の値は 1, 終点であれば -1 , 通過点であれば 0 となる。この際, m が始点ならば $x_{m,o} = 1$, 終点ならば $x_{m,p} = 1$ となるので, 式 (9) の値を $x_{m,o} - x_{m,p}$ として表すことができる。これにより, $x_{i,j}$ と $y_{k,l}$ を関連付ける。

提案する組合せ最適化問題に基づくスケジューリング手法は, 事前予約がない場合のグローバルスケジューリングにも適用可能である。また, 特定のサイトや条件をユーザーがあらかじめ指定している場合にも, 容易に適用することができる。

3.3 事前予約への適用

次に, 事前予約への適用について述べる。3.2 節の手法は, ある時間帯における予約プランを作成するものであるが, ユーザーの資源要求では, 確保する時間帯は 2.2 節で述べたように直接時間帯が指定される場合と, EST, LST, D で範囲指定される場合がある。前者は, 動的資源情報を取得する際にユーザーの指定した時刻を指定して利用可能な資源を問い合わせ, 取得した情報をもとに 3.2 節の手法を用いて予約プランを作成すればよい。一方, 後者は $[EST, LST + D]$ の時間帯の中から, 確保可能な資源を探索しなければならない。

我々は商用サービスを考慮して, G-lambda プロジェクトで規定している資源予約インタフェース GNS-WSI を用いている。GNS-WSI では, 各 RM における詳細な予約テーブルを公開しない代わりに, 指定した時刻における利用可能な資源情報を提供している。このインタフェースに基づき, 以下の手順で予約プランを決定する。

1. $[EST, LST + D]$ から予約時間帯候補を等間隔に N 個選ぶ。
 2. 資源候補となりうる各 RM に対し, N 個の時間帯の動的情報を一度に問い合わせる。
 3. 2 の情報から 3.2 節の手法を用い, それぞれの候補の予約プランを n 個 ($n \leq N$) 決定する。
 4. 決定した n 個の予約プランから, ユーザーの方針に従って最終的な予約プランを決定する。
- 1 では, 予約時間帯候補数 N を大きく設定するほど最適な資源群の選択ができるが, オンラインスケジューリングでは応答時間とのトレードオフがあるため, 予約時間帯の候補数 N はあらかじめ規定する。
- 2 では, 我々の用いる GNS-WSI バージョン 3 では

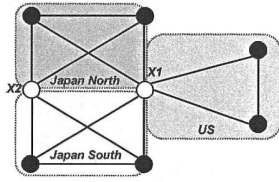


図 5 シミュレーション環境

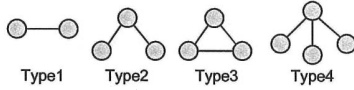


図 6 資源要求の種類

表 1 シミュレーション設定

環境設定	
RMS 構成	GRC=1, NRM=3, CRM=10
サイト数/ドメイン	4/JN, 3/JS, 3/US
ドメイン交換点	X1 (JN, JS, US), X2 (JN, JS)
CPU 数	JN{8, 16, 32, 64}, JS{8, 16, 32}, US{8, 16, 32}
CPU 単価	1
帯域 [Gbps]	ドメイン内 = 10, 交換点接続 = 20
帯域単価	ドメイン内 = 5, 交換点接続 = 3
資源要求設定	
ユーザ	UserA, UserB
資源要求の種類	Type1,2,3,4 (一様分布)
平均要求到着間隔 [min]	5, 10 (ポアソン到着)
LST - EST	平均要求到着間隔 × 3
予約時間幅 [min]	30, 60, 120 (一様分布)
予約 CPU 数	1, 2, 4, 8 (一様分布)
予約帯域 [Gbps]	1

1 回の問い合わせオペレーションで複数条件の問い合わせを同時に行うことができるため、これによるオーバーヘッドは小さい。

3 では N 個の動的資源情報に対して独立に予約プランを決定することができるため、3 に要する遅延も N によらず $O(1)$ とすることができる。

4 の n 個の予約プランの優先順位は、ユーザの方針に従って決定する。(a) 時刻優先であれば、EST に近い順に n 個の予約プランをソートする。(b) 価格優先または (c) 品質優先の場合は、それぞれのパラメータで優位な順にソートする。各予約プランに対して GRC は順次確保を試みて、ユーザの方針を反映させる。

4. 提案手法の評価

4.1 シミュレーションモデル

本手法の有効性を示すため、シミュレーションによる評価を行う。シミュレーション環境は、ELGL 実験で用いた図 3 を想定し、3 つのネットワークドメインと 2 つのドメイン交換点で構成される図 5 のような環境を用いた。簡略化のため図 5 では各ドメインに対

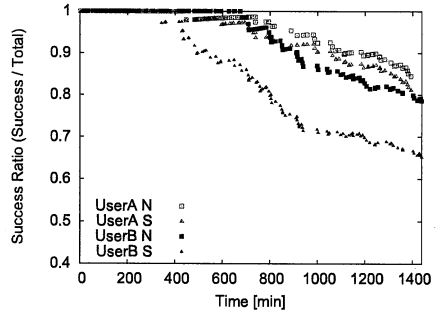


図 7 資源確保成功率の比較 (平均到着間隔=10[min])

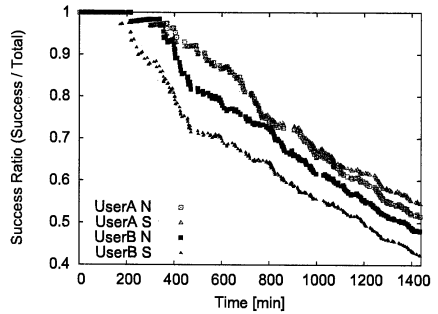


図 8 資源確保成功率の比較 (平均到着間隔=5[min])

して 2 つの計算サイトを示したが、ドメイン内のサイト数は表 1 のようになっている。ドメイン内のサイト (黒丸) 間は完全グラフで接続され、ドメイン内の各交換点 (白丸) とドメイン内の全サイトもそれぞれ接続されている。

シミュレーション設定の概要を表 1 に示す。シミュレーションでは、UserA と UserB が 1 つの GRC に対して資源予約要求をする。資源要求の種類は、ELGL 実験のアプリケーションを想定し、図 6 のいずれかを生成するようにした。各ユーザにおける資源要求設定は両方とも表 1 に従う。最初の 24 時間までに各ユーザの資源要求が到着し、それぞれ次の 24 時間の時間帯の中から資源の予約を要求をする。

GRC での予約時間帯候補数 N は 10 とし、各候補に対して確保コストが最小となる組合せを探索する。ソルバにはフリーの GLPK (GNU Linear Programming Kit) を用いた¹¹⁾。また、候補間での優先順位は時間優先とした。実験では、UserB に対してサービレベル (SL) が低い場合も想定した。具体的には、SL が低い場合は各 RM の利用可能な資源のうち、半分までしか要求できないようにした。

4.2 評価結果

図 7, 図 8 に平均到着間隔が 10 [min] および 5 [min] における、UserA と UserB の資源要求に対するグローバルスケジューリングの成功率を示す。図 8 の方がよ

り負荷が高い場合を表している。横軸は資源要求受付時刻、縦軸は各資源要求の割り当てが終了した時点での割り当て成功率である。各プロットの白抜きは UserA の結果、黒塗りは UserB の結果であり、三角 (S) / 四角 (N) は SL の設定の有 / 無を示す。

図 7 で SL なしの場合の UserA と UserB を比較すると、成功率はほぼ同程度であることが分かる。一方、SL ありの場合では、UserA の成功率は最終的に 0.84 から 0.81 とやや下がっているのに対し、UserB の成功率は 0.79 から 0.65 に著しく低下していた。図 8 でより負荷の高い場合で比較すると、UserA では 0.51 から 0.54 と上がっているのに対し、UserB では 0.48 から 0.42 と低下していた。これにより、SL の設定により低 SL のユーザの確保率を下げるのが可能であり、負荷が高い状況では、他のユーザの成功率を上げることが示唆された。

全資源利用可能時間に対する要求する資源確保の割合 (資源要求率) は、平均到着間隔が 10[*min*] のときで CPU は 1.36 は帯域は 0.81 (確定するパスがドメイン間とドメイン内で 1 対 1 であるとした場合)、5[*min*] では 2.71 と 1.63 であり、非常に高負荷な環境設定で実験を行っている。すなわち、10[*min*] の場合は計算資源では 1059[*min*] のところで資源要求率が 100[%] となるが、その際の予約成功率は SL なしの場合で UserA, B とも、0.85 以上であり、提案手法では高い予約成功率を示すことが分かった。

5. 関連研究

VIOLA³⁾ では、我々の GridARS と同様に計算資源とネットワーク資源を事前予約に基づき同時に確保することを目的としたシステム開発を進めているが、そのグローバルスケジューリングのアルゴリズムに関する研究は十分に行われていない。

安藤ら¹²⁾ は、本研究同様、複数計算・ネットワーク資源を確保するための予約プランを生成する予約アルゴリズムを提案している。提案するアルゴリズムは資源群を無向グラフで表し、バックトラック法により資源群を探索する。我々の研究は高品質分散実行環境構築を目的としているため、コアロケーションのための手法を提案しているが、安藤らのアルゴリズムではワークフローにも対応する。ただし、バックトラック法を用いているため、適切な解を探索する精度は低い。

6. まとめ

本研究では、高品質分散実行環境の構築を目的とした計算・ネットワーク資源のオンライングローバルスケジューリング手法を提案し、シミュレーションでの評価により、その有効性を示した。提案するグローバルスケジューリング手法では、ユーザの計算機及びネットワークに関する資源要求に対して、複数資源マネー

ジャから利用可能な資源情報を入手し、組合せ最適化問題を解くことにより予約プランを作成する。評価では、提案手法はユーザの要求と資源管理要件に対して有効に機能し、予約成功率が高いことを示した。今後は様々な環境下での、より詳細な評価実験を行う。

謝辞 本研究の一部は、情報通信研究機構 (NICT) の委託研究「ダイナミックネットワーク技術の研究開発」により実施した。

参考文献

- 1) Takefusa, A. et al.: G-lambda: Coordination of a Grid Scheduler and Lambda Path Service over GMPLS, *Future Generation Computing Systems*, Vol.22(2006), pp.868-875 (2006).
- 2) Thorpe, S.R. et al.: G-lambda and EnLIGHTened: Wrapped In Middleware Co-allocating Compute and Network Resources Across Japan and the US, *Proc. GridNets2007* (2007).
- 3) Barz, C. et al.: Co-Allocation of Compute and Network Resources in the VIOLA Testbed, TR 0051, CoreGrid (2006).
- 4) Mohamed, H. and Epema, D.: Experiences with the KOALA Co-Allocating Scheduler in Multiclusters, *Proc. 5th IEEE/ACM Int'l Symp. on Cluster Computing and the GRID (CCGrid2005)* (2005).
- 5) Nurmi, D., Brevik, J. and Wolski, R.: QBETS: Queue Bounds Estimation from Time Series, *Proc. 13th Workshop on Job Scheduling Strategies for Parallel Processing* (2007).
- 6) 竹房, 中田, 工藤, 田中, 関口: 多様な資源を事前予約で同時確保するためのグリッドコアロケーションシステムフレームワーク GridARS, 情報処理学会論文誌コンピューティングシステム (ACS20), Vol.48, No.SIG18, pp.32-42 (2007).
- 7) Takefusa, A. et al.: Design of a Domain Authorization-based Hierarchical Distributed Resource Monitoring System in cooperation with Resource Reservation, *Proc. HPC Asia 2009 (in print)* (2009).
- 8) Kudoh, T.: GRID computing and a role of photonic networks, *Proc. SPIE Asia Pasific Optical Communications (APOC2008)*, Vol. 7137, 713713 (2008).
- 9) G-lambda: <http://www.g-lambda.net/>.
- 10) Ahuja, R.K., Magnanti, T.L. and Orlin, J.B.: *Network Flows: Theory, Algorithms, and Applications*, Prentice Hall (1993).
- 11) GLPK (GNU Linear Programming Kit): <http://www.gnu.org/software/glpk/glpk.html>.
- 12) 安藤誠士郎, 合田憲人: グリッド上の事前予約スケジューリング手法の性能評価, 情報処理学会研究報告 2007-HPC-113, pp.37-42 (2007).