

複合クラウドへの動的なアプリケーション自動最適配置の 自動交渉技術による実現

A Preliminary Proposal of Dynamic Application Allocation on Hybrid Federated Clouds by Automated Negotiations

石川貴文*¹ 福田直樹*²
Takafumi Ishikawa Naoki Fukuta

*¹静岡大学大学院情報学研究科情報学専攻
Graduate School of Informatics, Shizuoka University

*²静岡大学情報学部
Faculty of Informatics, Shizuoka University

We propose an approach of using federated clouds that is suitable for migrating an application automatically based on its needs and environmental requirements by agent negotiation technologies. Federated cloud is a cloud computing environment which is built by two or more clouds provided from different providers. By using agent negotiation, the application provider can contract a specific cloud service provider on more flexible conditions to allocate and run the application efficiently on federated cloud. In this paper, We describes negotiation model and a preliminary prototype system for building federated cloud -based automated application by automated negotiations among agents.

1. はじめに

複合クラウドとは、2つ以上のクラウド環境を利用して構築されるクラウド環境のことである [Rochwerger 11]。クラウド環境には、その基本処理性能や課金体系以外に、可用性 (availability) 品質、セキュリティに対するポリシー、サーバの設置地域 (国や県など) の公開や保証の有無などの違いがある。これらの違いを適切に組み合わせて、目的のサービス構築のために必要な各種品質を安価なコストで実現できるようにすることが、複合クラウドを利用する際の1つの目的となっている [Rochwerger 11]。

複合クラウドの関連研究としては、文献 [Rochwerger 11] 等がある。この論文で扱う複合クラウドモデル (federated-cloud-model) は、複数のクラウド サービスプロバイダが協力し、1つの巨大なクラウド環境を構築するというものである。複合クラウドに参加したクラウドサービスプロバイダのコンピュータリソースが不足した場合、同じ複合クラウドに参加した他のクラウドサービスプロバイダから、不足分を事前に同意した値段で借り受け、ユーザの要求に対応する。このモデルの概念を図1に示す。このモデルでは、クラウドサービスプロバイダは、不足したコンピュータリソースを自身で追加せず、ユーザの要求に対応できることがメリットである。一時的なユーザからの要求の増加に対応するため、プロバイダ自身がコンピュータリソースを追加し、平常時にコンピュータリソースの過配給となることを防ぐことができる。

本研究で構築する複合クラウドは、文献 [Rochwerger 11] のような、複数のクラウドサービスプロバイダが協力して、1つの巨大なクラウド環境を構築するというモデルではなく、クラウドサービスユーザが、運用するサービスに応じて適したクラウドサービスプロバイダを選択することによって複合クラウド環境を構築するというモデルである。文献 [Rochwerger 11] では、複合クラウドへの参加者の間でコンピュータリソースの貸し借りをを行うが、Google や Amazon といった誰もが知っている業界の大手の企業が、コンピュータリソースが足りなくなる可能性が低いと考えられる。契約しているクラウドサービスプロバイダのリソースがコンピュータリソースが足りなくなっ

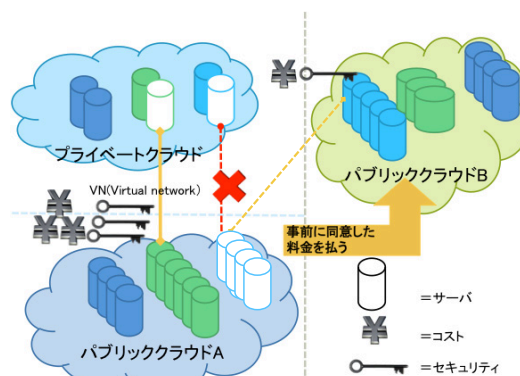


図1: 複合クラウド例

たならば、ユーザは Google や Amazon といった大手に乗り換えれば済む。よって、本研究で構築する複合クラウドでは違うモデルを採用した。本研究は、複合クラウド環境をどれだけ安いコストで構築することができるかを目的としているわけではない。コストも重要な要素ではあるが、それよりも、フレキシブル (融通のきく) な条件での契約を行うことを目的としている。

本研究では、効果的な複合クラウドを動的かつ自動的に構築可能とするために、自動交渉技術を用いることを考える。自動交渉技術を用いることで、固定料金制 (各クラウドサービスプロバイダの料金表) では実現の難しいような柔軟な条件で、利用者側はクラウドを利用できるようになると同時に、プロバイダ側も自身の計算資源等の運用の都合にあわせたサービスの提供条件の変更を利用者側に促しやすくなると考えられる。

本研究で扱う自動交渉技術とは、交渉主体それぞれの効用空間をソフトウェアエージェントに持たせて、お互いが合意できる条件で、交渉プロトコルに基づいて、お互いの効用空間をできるだけ公開しないでソフトウェアにより自動的に見つけ出す技術である [伊藤 10]。エージェント間交渉に関して様々な関連研究 [伊藤 10, 服部 06, 藤田 08, 石川 12, 鶴橋 12] が行われている。クラウド環境を構築する場面で適用すれば、ユー

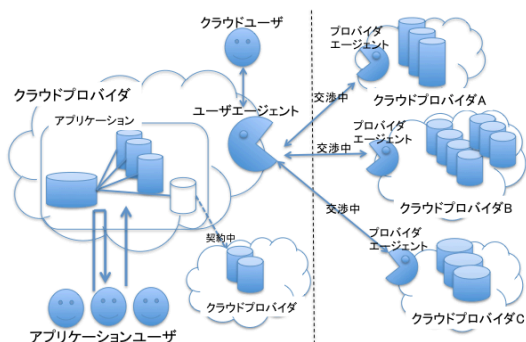


図 2: 本研究での複合クラウドモデル概念図

が必要とする CPU, ストレージ容量, 1GB あたりの通信料金, サービスの可用性を示す尺度である SLA(Service Level Agreement)[Rochwerger 11] などについて交渉し, ユーザとプロバイダ双方にとって同意できる条件で契約できるように合意点を自動的に見つけ出せる可能性がある.

本研究では, 自動交渉技術を用いて, 個々のアプリケーションサービスに最適な複合クラウド環境を構築するためのフレームワークの実現を目指す.

2. 提案手法

2.1 前提条件

本研究では, 検討の対象をより本質的な部分に集中するために, 次の i) ~ iv) の 4 つを仮定する.

i) アプリケーションプロバイダ (以降は単にユーザと呼ぶ) は, アプリケーションをプラットフォーム間での移動が容易なプラットフォーム上で構築しているものとする. 具体的には, 本論文中での検討内容については, MiLog[Fukuta 00] をアプリケーションのプラットフォームとして利用する. ii) クラウド側で提供される資源は MiLog のプラットフォームの形で提供され, そのプラットフォームを稼働するための機材には MiLog が事前にインストールされ, 起動している. iii) ユーザ側, プロバイダ側の双方は, 自身の効用空間を明確に定義でき入力できる. iv) 本研究では, ユーザー一人に対し, 複数のプロバイダが存在する場面を仮定する.

ここで iii) で示したユーザとプロバイダの効用空間の入力は, 交渉条件の各要素 (ストレージ容量, メモリ容量や CPU 性能など) の 1 単位あたりで得られる利得 (ユーザの場合), あるいはそれを提供することで発生する利益 (クラウドサービスプロバイダの場合) を入力する. 例えばユーザの入力した条件が, ストレージ使用 1GB あたりの利益が 100 円, メモリ 1GB あたりの利益が 1000 円なのであれば, メモリ 1GB 減らすかわりにストレージ 10GB 増やすようにできれば同等の効用値となる. この場合, ユーザのエージェントは, 支払う金額が同じであれば, ストレージ容量よりもメモリ容量が高い Offer を提示する. また, クラウドサービスプロバイダ側が入力した条件が, ストレージ使用 1GB あたり 100 円, メモリ 1GB あたり 1000 円なのであれば, クラウドサービスプロバイダのエージェントは, ストレージ容量よりもメモリ容量が高い Offer を提示する. この場合, ユーザ, クラウドサービスプロバイダ双方にとって高い効用値が得られる合意点を発見しやすい.

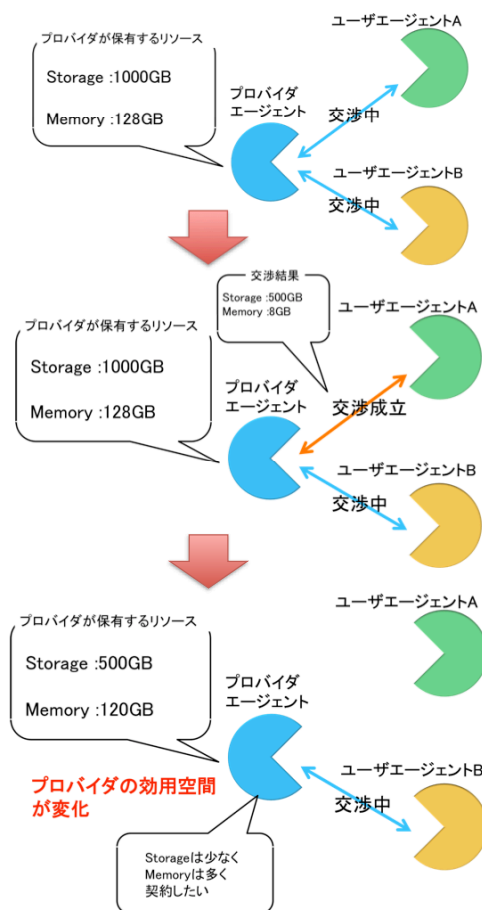


図 3: 交渉中の効用空間の変化例

また, プロバイダは交渉前に, 予め交渉に利用するリソースを自身のリソースから確保し交渉を行うこととする. 予めリソースを確保しておくため, 他者の交渉結果が影響し, プロバイダの効用空間が交渉中に変化しないこととする. 交渉中に効用空間の変化する例を図 3 に示す. 複数者で交渉を行う場合, 先に成立した交渉によって効用空間が変化する. プロバイダエージェントとユーザーエージェント A との交渉結果は, ストレージが多くメモリが少ない内容である. そのため, プロバイダが保有するコンピュータリソースはストレージが不足気味となる. そのため, クラウドサービスプロバイダは, ユーザーエージェント B との交渉ではストレージよりもメモリが多い内容で契約したいと考える. そのため, クラウドサービスプロバイダの効用空間はユーザーエージェント B との交渉中に変化する. そのため, ユーザーエージェント B との交渉で高い効用を得られるようにするには, 交渉中に戦略を切り替える必要があるが, 本研究では, この問題を扱わないこととする.

2.2 提案手法

図 4 に本提案手法によるアプリケーション自動配置機構の概観を示す. ユーザは, ユーザ入力ページから交渉条件 (ストレージやメモリなど各要素の 1 単位あたりで得られる利得) を入力する. 入力した条件 (効用空間) を, MiLog 上で起動しているユーザーエージェントに与える. クラウドサービスプロバイダは, プロバイダ入力ページから, 交渉条件を入力し, プロバイダエージェントに渡される. ユーザと交渉して合意できる可能性のあるプロバイダが見つかったら, 交渉を開始する.

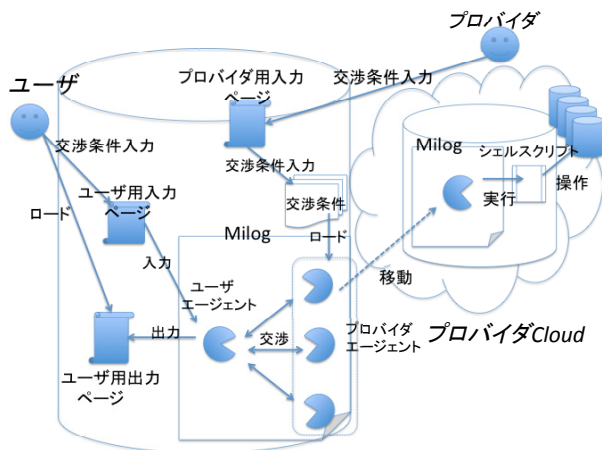


図 4: システムの構成

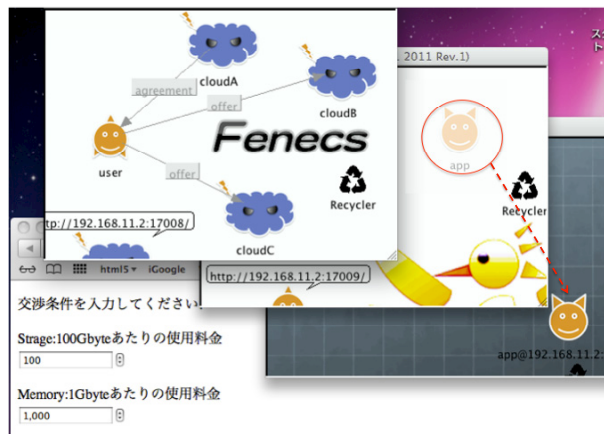


図 6: 交渉に基づくアプリケーションの移動

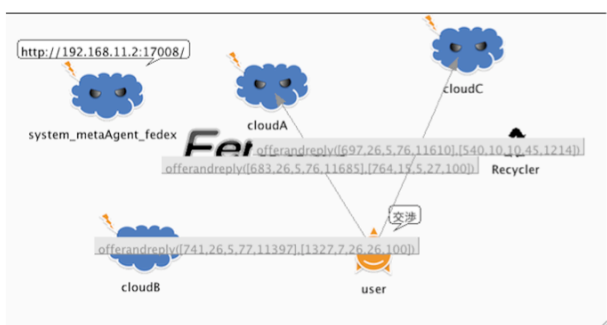


図 5: エージェント間交渉の実装例

交渉が終了したら、ユーザエージェントとプロバイダエージェントは、各サーバに移動し結果をそれぞれ表示する。プロバイダエージェントは、交渉結果を基に、シェルスクリプト等を実行し、アプリケーション実行プラットフォームのインスタンスを起動させる。ユーザエージェントは、契約したプロバイダのサーバへユーザのアプリケーション実行プラットフォームのアプリケーションを移動させる。移動させたアプリケーションは、新しく契約したプロバイダ上で動作を継続し、アプリケーションサービスを提供できるようにする。

3. 実装

図 5, 図 6 は、本提案手法に基づくシステムの試作時の動作例である。図 5 はアプリケーションプロバイダであるユーザのエージェントとクラウドプロバイダのエージェント間での交渉の様子を示している。MiLog 上で自動交渉を実装する、user(ユーザエージェント)から提案された Offer に対し、cloudA(プロバイダエージェント)から counter offer をその返答として受け取ることができないこととした。ここで、提案に合意の場合には単に counter offer として提案と同じものを返す。すなわち、本システムでは、Offer の提案を offerandreply([X],[Y])という形式で実装した。X には相手へ提案する Offer を格納し、Y には自らが提案した Offer に対する相手からの counter offer が格納される。相手への Offer の提案とその Offer への counter offer を MiLog 上のエージェント間通信述語 query/2 の結果として受けとるまでの動作を 1 つのクエリとすることで、相手からの counter offer を受け取れるよう実装した。cloudA

は user からの Offer に同意できる場合、user からの Offer をそのまま counter offer として B に格納して返す。

図 6 は交渉成立後のアプリケーションの移動の様子を示している。user からの Offer に対して、cloudA から、同意 (agreement) が返されている。cloudA は、交渉結果を基に、クラウド上に利用可能な計算機リソースを準備するとともに、そのリソース上で利用する仮想マシン (VM) を設定し起動させ、起動した VM の IP アドレスを user に伝える。図 6 中の右ウィンドウが、起動した VM 上での動作状況を表している。その後、もともと動作していたアプリケーション (図 6 中の中央ウィンドウにいた App という名のアプリケーションを実行するためのエージェント) が、新しく用意されたクラウド環境上の MiLog 実行環境 (右のウィンドウで表示される領域) に移動する。MiLog の持つ機能により、移動後の App には、移動前のアドレスを用いてもアクセス可能することができる。

4. アプリケーションシナリオへの適用

具体的なアプリケーションへの適用シナリオとして、次に示すような状況に適用した場合における本提案手法の効果について考える。

ユーザはこのシナリオ開始時点では、ストレージに重みを多くする効用空間を持っているとする。時間の経過に伴って、ストレージよりもメモリ容量が必要となる処理要求が増加した場合、ユーザはアプリケーションに対する要請の変化に対応するために、ストレージ容量の重みを減らし、メモリの重みを増やした効用空間とする必要が出てくる。この場合のユーザの交渉条件は user[Storage = 500, Memory = 100, Cost = 10000] から user[Storage=100, Memory=500, Cost = 10000] となるとする。その際、3つのクラウドサービスプロバイダが A から C まで存在するとして、それぞれが持つ効用空間が、

providerA[Storage = 500, Memory = 300, Cost = 8000],
 providerB[Storage = 100, Memory = 1000, Cost = 15000],
 providerC[Storage = 200, Memory = 500, Cost = 9000]

とする。この場合、プロバイダ B は初期提示コストが大きく上回るため交渉相手からは除外されたとする。よってユーザの交渉相手はプロバイダ A とプロバイダ C となる。プロバイダ A は、メモリ容量よりもストレージの重みが大きいため、ストレージを多く、メモリ容量が少ない Offer を提案するように行動する。プロバイダ C はストレージよりもメモリ容量の重

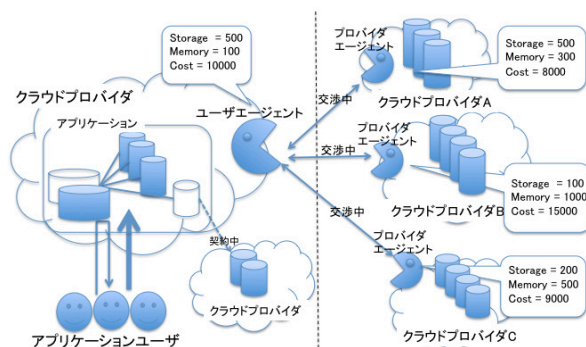


図 7: シナリオ例 1

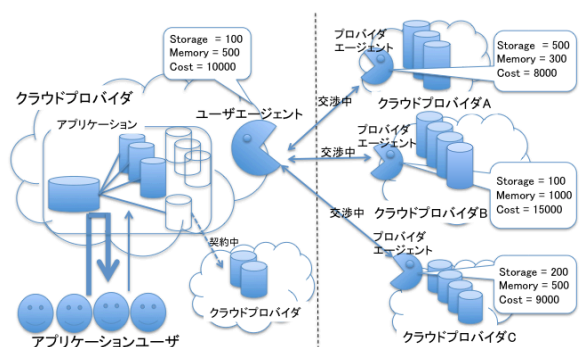


図 8: シナリオ例 2

みが大きいので、ストレージを少なくメモリ容量を多い Offer を提案する行動をとる。この場合、ユーザにとって適した（より高い効用値が見込める）交渉相手はプロバイダ C となる。

ユーザとプロバイダ C との契約が成立し、アプリケーションを移動した後、ユーザにとってより高い効用が見込める交渉相手プロバイダ D が現れたとする。プロバイダ D の効用空間は [Storage = 100, Memory = 500, Cost = 8000] とする。この場合、ユーザがすぐにプロバイダ D と交渉しアプリケーションを移動させてしまうことも可能ではあるが、プロバイダ C との契約にあたって一定の固定コスト (契約初期費用等) が加算されることも加味し、ユーザにとってより高い効用が見込める交渉相手が現れたとしても、そこでの交渉ではプロバイダ変更ですでにかかったコストも考慮して交渉の有無を判断する必要がある。

また、アプリケーションの運用に適したクラウドサービスプロバイダが存在していたとしても、プロバイダ側を無視してユーザ側のみが最大限の効用を得られるような条件での契約を行うことは困難である。実際に交渉の結果として得られる合意結果は、ユーザとクラウドサービスプロバイダ双方の効用空間を公開して、お互いの効用値が最も高くなる条件（すなわち、ある種の最適解）とは、必ずしも一致しない。エージェント間交渉では、お互いの効用空間を非公開にして交渉を行うため、その合意結果の品質には双方のエージェントの交渉戦略の性能が大きく影響する。エージェントの持つ交渉戦略の実装方法および、グローバルな視点から見た割り当ての最適性に関する議論については、本論文ではこれ以上議論せず、今後の課題とする。

5. おわりに

本システムでのサービス（またはアプリケーション）は、プラットフォームとして MiLog を使用しており、MiLog が実装している機能により、移動後のアプリケーションのアクセスが容易に行える。MiLog 以外のプラットフォームを使用するアプリケーションを移動を実現するためには、移動に伴う環境（IP アドレスやサーバ等へのアクセス方法）の変更に柔軟に対応可能なアプリケーションの構築フレームワークの提案とその実装、および、アプリケーションの移動時にかかるコストを考慮した自動交渉の実装が必要となる。

本研究では、クラウドサービスプロバイダは交渉前に、一定のリソースを予め確保（予約）し、交渉結果に応じて、予め確保したリソースを開放しており、プロバイダの効用空間は交渉中一定であることを仮定している。実際のクラウド環境では、他のユーザとの交渉結果が影響し、プロバイダの保有するリソースが、ユーザとの交渉中に変化する可能性がある。これらの条件を素朴な線形の効用空間およびそれを仮定した交渉戦略で扱うことは難しい。これらの問題を解決するために、複雑な非線形な効用空間を扱えるようにし、効用空間の変化に対応する新しい戦略の導入を行えるようにすることが今後の課題である。

参考文献

- [Rochwerger 11] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P. Massonet, H. Munoz, and G. Toffetti: “Reservoir - When One Cloud Is Not Enough”, IEEE computer, vol. 44, no. 3, pp.45–p51, March 2011.
- [Fukuta 00] N. Fukuta, T. Ito, and T. Shintani: “MiLog: A Mobile Agent Framework for Implementing Intelligent Information Agents with Logic Programming”, In the Proceedings of the First Pacific Rim International Workshop on Intelligent Information Agents (PRIIA2000), pp.113–123, 2000.
- [伊藤 10] 伊藤孝行, “マルチエージェントの自動交渉モデルとその応用”, 知能と情報 (日本知能情報ファジィ学会誌), Vol.22, no.3, pp.295–302, 2010.
- [服部 06] 服部宏充, 伊藤孝行, Mark Klein, “非線形効用関数を持つエージェントのためのオークションに基づく交渉プロトコル”, 電子情報通信学会論文誌 D-I, 電子情報通信学会, Vol. J89-D, No. 12, pp.2648–2660, 2006.
- [藤田 08] 藤田桂英, 伊藤孝行, 服部宏充, “複数論点交渉問題におけるエージェントの効用空間の公開範囲に基づく交渉手段の実現”, コンピュータソフトウェア (日本ソフトウェア科学会論文誌)「ソフトウェアエージェントとその応用特集号」, 日本ソフトウェア科学会, Vol. 25, No.4, pp.167–180, 2008.
- [石川 12] 石川貴文, 福田直樹, “自動交渉技術を用いた複合クラウドへのアプリケーション自動最適配置フレームワークの試作”, 情報処理学会第 74 回全国大会, 3T-8.
- [鶴橋 12] 鶴橋吉矩, 福田直樹, “複数者間の同時交渉に適用可能なエージェント間多論点交渉システムの試作”, 情報処理学会第 74 回全国大会, 3T-7.