

インタークラウド環境における仮想システム構築の最適化サービスに関する検討

玉家武博[†], 斎藤篤志[†], 三浦克宜[‡], 棟朝雅晴^{*}

北海道大学工学部[†], 北見工業大学情報処理センター[‡], 北海道大学情報基盤センター^{*}

1. 背景と目的

クラウドコンピューティングのサービスを提供するプロバイダの数とサービスの多様性の増加により、ユーザーが自身の持つ要件に適したクラウドサービスを利用可能な候補の中から選択することが困難になりつつある。そこで、本研究では様々なクラウドのリソースが相互に連携するインタークラウド環境下で、ユーザーのリクエストに基づき、ユーザーの代わりに仮想システムを構成するクラウドサービス（仮想マシンなど）を自動的に最適化し、構築するサービスの実現に向けた検討を行う。

2. システムアーキテクチャの検討

現在、様々なプロバイダがクラウド IaaS (Infrastructure as a Service) サービスとして仮想マシンを提供しており、ユーザーは Web サービス API を通じてそれら进行操作することができる。仮想マシンの料金体系や性能に関する情報は Web 上のドキュメントで公開されている。現在、各プロバイダは独自の方法により自身のクラウドサービスを記述しており、プロバイダ間で簡単にサービスの比較を行うことが出来ない。また、様々なプロバイダのクラウドサービスに関する情報をプログラムから一元的に取得することもできない。よって、プロバイダに依らない統一された方法でクラウドサービスの情報を記述し、それらをプログラムからアクセスできる形で提供する、システム内で情報源の役割を果たすコンポーネントが必要である。

次に、候補となるクラウドサービスは膨大な数となり、それらの性能をすべて調べながら比較することは非常に時間のかかるタスクである。

また、ユーザーの持つ抽象的な要件（スループットなど）を実際のクラウドサービスの性能と結びつけることも容易ではない。よって、ユーザーの持つ要求内容に基づいて適切なクラウドサービスを膨大な数の候補の中から選択するコンポーネントも必要となる。

最後に、複数のプロバイダのクラウドサービスに跨る仮想システムを構築するために、プロバイダに依らず、一元的かつ自動的に仮想マシンなどをデプロイし、仮想システムを使用するための準備を行うコンポーネントも必要である。

以上 3 つのコンポーネントにより、本来はユーザーが自身で行わなければならなかった、(1) クラウドサービスの情報の収集、(2) 要求に基づく最適なサービスの選択、(3) 仮想システムのデプロイ、というタスクをすべて自動化し、システム上で行うことが出来る。

本研究では提案するシステムの機能を上記の 3 つのコンポーネントに分割し、クラウドサービスの情報源となるサービスを Cloud Metadata Service (CMS)、仮想システムの最適化を行うサービスを Optimization Engine (OE)、仮想システムの自動デプロイを行うサービスを Deployment Agent (DA) とそれぞれ名づけた。これらをすべて Web サービス API で連携させ、ユーザーがこれらの Web サービスを利用するための GUI も実装する。

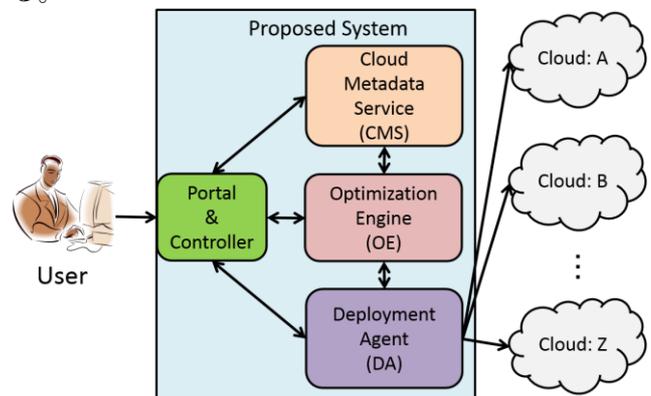


図 1 提案するシステム構成

3. システムの構成

本研究で提案するシステムの全体的なアーキ

A study on virtualized systems deployment optimization services in the inter-cloud environment

[†] Tamaya Takehiro, [†] Saito Atsushi

[‡] Miura Katsunori

^{*} Munetomo Masaharu

Graduate School of Engineering, Hokkaido University[†]

Information Processing Center, Kitami Institute of Technology,[‡]

Information Initiative Center, Hokkaido University^{*}

テクチャを図1に示す。

CMSはMichael Smit[1]らによって提案されたクラウドに関する情報を提供するREST Webサービスを基に設計した。URIパターンの設計や結果として送信するテキストの形式(JSON)は基にしたWebサービスと統一しているが、サポートするURIパターンの種類と取り扱う情報の項目に関しては本研究で必要なもののみ限定している。

CMSは各プロバイダのクラウドサービスごとに固有のURIを割り当て、それらに対するGETメソッドを用いたHTTPリクエストを受理する。例えば、情報を提供するIaaSのプロバイダの一覧を取得する場合、URIエンドポイントの後に、「/iaas」と続けてリクエストを送信する。また、あるプロバイダの仮想マシンの情報一覧を取得する場合、URIは「/iaas/(プロバイダ名)/instances」となる。

CMSはレスポンスとして、JSONのデータを返す。インスタンスに関する情報には、CPUのコア数、メモリといった仮想マシンとしての性能に関する情報だけでなく、そのインスタンスがホストされるデータセンターの地理的な位置といった情報も含む。CMSで提供する情報はすべてDB上に保管されており、CMSはDBから必要な情報を取得し、JSONで表現したものをレスポンスとして送信する。

OEはユーザーのリクエストと候補となるクラウドサービスの情報に基づき、仮想システムを構成する仮想マシンを選択し、最適化された仮想システムを提案するWebサービスである。

OEは単一のURIのみをサポートし、そのURIに対するPOSTメソッドを用いたHTTPリクエストを受理する。ユーザーのリクエストはJSONで記述されOEに送信される。リクエストを受け取るごとにOEはCMSと通信を行い、候補となるクラウドサービスの情報を取得する。

最適化の手法として、本研究では多目的遺伝的アルゴリズムNSGA-III[3]を用いる。遺伝的アルゴリズムの中では、候補となる仮想システムを構成するためのクラウドサービスの組み合わせが一つ一つの解として扱われる。遺伝的アルゴリズムの性質として、一度の実行で複数のパレート最適解を最終的な出力として導き出せるというものがある。本研究ではこの性質を利用して、最適化の結果、仮想システムを構築するための複数のパレート最適解を出力する。OEは最適化された仮想システムの情報をJSON形式のテキストにまとめ、レスポンスとして送信する。

DAは最適化された仮想システムを構成する仮

想マシンなどを、様々なクラウド上に一元的にデプロイするWebサービスである。DAに対するリクエストはデプロイする仮想マシンの情報とクラウドAPIを操作するためのキーなどの情報を含む。DAはそれらの情報を基に、Apache LibcloudのAPIを利用して仮想マシンのデプロイを行う。Apache Libcloudは様々なクラウドのAPIを抽象化し、統一されたインタフェースによって各クラウドへの操作を可能とするPythonのライブラリである。これを用いることで、DAがサポートするクラウドを追加していく作業を大幅に削減することが出来る。

DAはデプロイした仮想マシンに関する情報をJSONで表現し、レスポンスとして送信する。

4. 今後の課題

仮想マシンなどのクラウドサービスに関してドキュメントで公表されている性能は必ずしも実際のパフォーマンスを表していない場合がある。そのため、将来的には候補となるクラウドサービスの性能をドキュメントからだけではなく、ベンチマークなどによって取得し、CMSに反映させる予定である。

また、システムの実用性をさらに上げるため、ユーザーがリクエストをすべて記述するのではなく、リクエストされる仮想システムとして汎用的なものをテンプレートという形であらかじめシステム側で用意しておくことを検討している。そして、ユーザーがそれをもとに自身の要求に合うように細かい部分を修正していくというリクエストの作り方が可能な仕様を盛り込むことで、ユーザーの負担を減らしながらも要求を表現できる仕組みを作っていきたい。

参考文献

1. Smit, M., Pawluk, P., Simmons, B., & Litoiu, M. A web service for cloud metadata. 2012 IEEE Eighth World Congress on Service pp. 361-368. 2012.
2. Apache Libcloud, <https://libcloud.apache.org/>
3. Deb, K., & Jain, H. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 18, NO. 4, AUGUST 2014. 577-601