

クラウド・コンピューティング —動向と今後の展望—

浦本 直彦

次世代のITサービス基盤技術およびサービスとして注目を集めているクラウド・コンピューティングは、初期の過熱気味の議論を経て、現在普及にむけての研究開発が進められている。本稿では、クラウド・コンピューティングの定義や技術要素、今後の展望などについて概観する。

キーワード：クラウド・コンピューティング，分散並列処理，解析クラウド

1. はじめに

クラウド・コンピューティング (Cloud Computing) は、さまざまな計算機資源やソフトウェアサービスを、ネットワーク経由でその実体を関知せずに利用することができるITサービス基盤技術およびビジネスモデルである。登場初期には、過度の期待も混じりつつ大きな注目を浴びたが、現在はその過熱ぶりも一段落し、普及に向けてさまざまな動きが展開されている。

クラウドの誕生は、産業システムの工業化の一例だと捉えることができる。例えば、動力は、かつて水力によってもたらされ、人々は水車を利用することができる川のほとりで作業していた。その後産業革命においては蒸気機関が工業化の原動力となった。そして、現在では、動力源となる電力は、動力を消費する場所から離れた所で生産され（消費者は発電所がどこにあるかを知る必要がない）、送電ネットワークを介して提供されている。安価で大量の資源をその実体を意識させずに提供するクラウド・コンピューティングは、コンピュータの世界に工業化をもたらすものであるといえよう。

本稿では、クラウド・コンピューティングの定義や形態をいくつかの視点から紹介する。また、技術要素として、スケールアウト、クラウドデータベース、解析クラウドを取り上げ、最後に、今後の展望を述べる。

2. クラウド・コンピューティングとは何か？

クラウド・コンピューティングという言葉をはじめて使ったのは、GoogleのCEOであるEric Schmidtであるとされている。2006年のことである。ただし、クラウドの概念がこの年に誕生したわけではない。Customer Relationship Management (CRM) ツールのクラウドによる提供で有名なSalesforce社が設立されたのは1999年であるし、Amazonが仮想サーバーやストレージを提供するAmazon Web Services (AWS) を開始したのは2002年である。また、グリッド・コンピューティングやユーティリティ・コンピューティングなど、クラウド・コンピューティングの特徴の一部をなす技術はそれ以前から知られていた¹。例えば、ITリソースを、水道や電気のように、必要なときに必要なだけ提供するユーティリティ・コンピューティングの概念は、人工知能分野のパイオニアであるJohn McCarthyによって、今を去ること50年前、1961年に提唱されている。

クラウド・コンピューティングは、複数のレイヤーにまたがる技術的要素とビジネス形態から構成されるため、その定義も複数存在する。ここでは、アメリカ国立標準技術研究所 (National Institute of Standards and Technology, NIST) が公開しているものを挙げる[1]。

「クラウド・コンピューティングは、設定可能な計算機資源 (例、ネットワーク、サーバー、ストレージ、

うらもと なおひこ
日本アイ・ビー・エム(株) 東京基礎研究所
〒242-8502 大和市下鶴間 1623-14

¹ WikipediaにCloud Computingというエントリができたのは、2007年3月のことである。ちなみに内容は、“Utility Computingを参照せよ”であった。

アプリケーション、サービス)の共有プールに対し、使いやすいオンデマンドのネットワークアクセスを可能にするためのモデルである。最小限の管理労力やサービス提供者とのやりとりによって、サービスの迅速な準備(プロビジョニング, provisioning)と公開を行うことができる」

NISTでは、さらに、クラウドに特有の5つの特徴、3つのサービスモデル、4つのデプロイメントモデルを定義している。これらを含むクラウド・コンピューティングの概念については多数の資料が公開されているため(文献[2][3]など)、本稿では必要最小限の説明にとどめる。

2.1 クラウド・コンピューティングの特徴

NISTの定義[1]では、以下の5つがクラウド・コンピューティングの特徴として挙げられている。これをすべて満たすものだけがクラウドであるとは限らないが、クラウドの基本的な性質を捉えたものである。

- オンデマンド・セルフサービス (On-demand self-service) : ユーザ側から自動的に、必要な機能のプロビジョニングが可能である。
- 多様なネットワークアクセス (Broad network access) : さまざまなクライアントから、標準的なネットワークプロトコルで利用できる。
- リソースの確保と割り当て (Resource pooling) : コンピュータ資源が、マルチテナントモデルでプールされ、複数のユーザに提供可能である。ユーザは資源の物理的な位置に関知しない。
- 迅速な弾力性 (Rapid elasticity) : 提供される機能が、迅速に、弾力的にプロビジョニングできる。
- サービス利用状況の計測と報告 (Measured Service) : メータリング機能を使って、計算機資源の利用状態をコントロールしたり最適化したりすることができる。

2.2 サービスモデル

さらに、クラウドがサービスとして提供する資源のレベルの違いから、以下のようなサービスモデルが考えられる。

- Infrastructure as a Service (IaaS) : 仮想サーバー、ストレージ、ネットワークなどの計算機資源を提供するサービス。Amazon S3/EC2などがIaaSの代表例である。
- Platform as a Service (PaaS) : クラウド利用者が

が作成したアプリケーションを実行するための環境を提供するサービス。Google App EngineなどがPaaSの代表例である。

- Software as a Service (SaaS) : クラウド環境が提供するソフトウェアサービス。クラウド利用者はブラウザ経由で用いることが多い。Salesforce.comやGoogle gmailなどがSaaSの代表例である。

様々なレベルの機能を(抽象化された)サービスとして提供する“X as Service (XaaS)”という言葉もよく使われる。例えば、マルウェアやウイルスの検知やモニタリングなどのSecurity機能をサービスとして提供するSecurity as a Service、複数のクラウドサービスを連携するIntegration as a Service、データセンターそのものを仮想的に提供するDatacenter as a Service[4]などがこれにあたる。

2.3 デプロイメントモデル

最後に、クラウドが配置される場所によって、以下のような分類がなされる。

- プライベート・クラウド : 単一の企業・組織内で利用されるクラウド
- コミュニティ・クラウド : 複数の組織やコミュニティ内で使われるクラウド。業界クラウド、インダストリクラウドと呼ばれることもある。
- パブリック・クラウド : 公開され多数のユーザからアクセスされるクラウド
- ハイブリッド・クラウド : 上記クラウドを組み合わせたもの。NISTの定義には明記されていないが、社内のITシステムとの統合を含む場合もある。

この他、企業が用いるクラウド(プライベート・クラウドやハイブリッド・クラウド)を総称して、エンタープライズ・クラウドという言い方をすることがある。また、企業内のITシステムを、オンプレミス(on-premise)、企業の外にあるパブリック・クラウドをオフプレミス(off-premise)と呼ぶことも多い。

さらに、別の観点からクラウド・コンピューティングを見てみよう。

2.4 サービス提供基盤としてのクラウド

クラウドを、ITサービスを提供するための基盤技術として考えると、クラウド・コンピューティングは、インターネットの登場、そしてWebの誕生に始まり、Webサービス、SOA、Web 2.0と進んできたサービス提供基盤の進化形態と考えることができる。Web

サービスやSOAによって抽象化・自動化・標準化されたサービスの考え方が、ハードウェア、ソフトウェア、そして通信技術の向上と相まって、より広範囲の計算機資源やサービス形態へと拡張されたものがクラウドであるという流れである。

2.5 仮想化環境としてのクラウド

クラウド・コンピューティングが普及した技術的な要因の一つは、仮想化技術の発展である。企業や組織における経営課題の一つは、IT資産に関するコストの増加、特に、新規投資ではなく、既存のシステムの運用や改修、電気代にかかるコストの増大である。仮想化技術によるサーバー統合やネットワーク・ストレージの仮想化技術がこの問題を解決するために適用され、クラウド・コンピューティングの技術を下支えしている。仮想化対象は、サーバーやネットワーク、ストレージだけでなく、データベース、メッセージキュー、キャッシュなどの資源も仮想化されてサービスされている。

2.6 エコシステムとしてのクラウド

クラウド・コンピューティングの成長は、GoogleやAmazonのようなパブリック・クラウドベンダーの成功、IBM、Microsoft、Oracleのような大手ITベンダーのクラウドへの積極的な投資により知名度とマーケットが拡大していることに起因しているが、注目すべきは、様々なベンダーやコミュニティが既存のサービスを組み合わせながら、有用な技術やサービスを提供し、巨大なエコシステムが構築されつつあることにある。このようなエコシステムがうまく機能することで、さらなる市場の拡大が期待できる。

2.7 研究としてのクラウド

クラウド・コンピューティングは、幅広い技術要素を含むため、研究エリアも、ネットワーク、システム管理、信頼性・可用性、仮想化技術、分散処理、データベース、パフォーマンス、セキュリティなど多岐にわたっている。そのためそれぞれの学会でクラウド・コンピューティングに関するセッションが組まれていることもあり、関連分野を一覧するのは難しい。クラウドに特化した著名な国際会議としては、IEEE Cloud Computingなどがあり、最近のトレンドを知ることができる (<http://www.thecloudcomputing.org>)。

また、クラウドは研究の道具としても重要なインフラとなってきた。大量の計算機資源を安価に利用することで、これまで実現できなかった大規模な実験

環境を構築することができる。例えば、Amazonが提供するIaaSであるElastic Compute Cloud (EC2)では、標準的なLinuxサーバーの利用価格は、わずか1時間8.5セントである(他のベンダーも同程度の価格帯でサービスを行っている)。サーバーは1時間単位で利用できるため、必要な分だけを利用することができる。

3. クラウド・コンピューティングを支える技術

前述したように、クラウド・コンピューティングは様々な技術要素を含んでおり、すべてのトピックを限られた紙面で説明することは不可能である。本稿では、読者の興味を想定して、いくつかのトピックを紹介する。まず、クラウドを広域分散システムであると考えたときに、スケーラビリティは本質的な問題である。3.1節では、スケールアウトとスケールアップという2つのアプローチについて説明する。また、このような広域分散システム上で、どのようにデータを管理し検索するかを説明するために、3.2節ではクラウドデータベースをとりあげる。3.3節では、それらの仕組みの上に構築される解析クラウドについて説明する。

3.1 スケールアウトとスケールアップ

クラウド・コンピューティングの特徴の一つは、計算機資源利用におけるスケーラビリティである。利用者の要求に合わせて必要な資源を必要なだけ使用する(使わなければ返すことができる)ために、多くのクラウドサービスではスケーラビリティの確保が技術的な課題となる。スケーラビリティを確保するためには、2つのアプローチがある。

スケールアップ: 処理を行うサーバー単体の能力(CPU、メモリなど)を増強することで、スケーラビリティを確保する。

スケールアウト: サーバーの台数を増やすことで、スケーラビリティを確保する。

スケールアップでは、管理がしやすいこと、従来のミドルウェアやアプリが利用しやすいといった利点がある反面、サーバーのコストが増大するという課題がある。一方、スケールアウトは、コモディティ化した安価なサーバーを多数並べることで性能を担保するため、コスト的に競争力が高く²、並列性が高い処理を行うときには威力を発揮するが、多数のサーバーの管

² ライセンス料が必要なソフトウェアが必要なときにはスケールアウトでもコストが増大する。

理（例、故障時）や専用のミドルウェアなどが必要となる。クラウド・コンピューティングのコンテキストでは、Google に代表されるクラウドベンダーが提供するサービス（検索やデータの加工）がスケールアウトに適したワークロード（処理の形態）であることもあり、スケールアウトがよく言及されるが、スケールアウトでなければクラウドではないというわけではなく、トランザクションや複雑なデータベースクエリを必要とするエンタープライズアプリケーションでは、スケールアップアプローチが適している場合もある。

スケールアウト型の並列処理プログラミングモデルとして代表的なものが、Google が提案した MapReduce [5] である。MapReduce では、Map と Reduce という 2 種類の関数を定義することで多数の計算機上での並列処理を可能にする。

Map 関数：データ列を入力として受け取り、何らかの処理を行い、key-value ペアの列を返す。

Reduce 関数：key とそれに対する value の集合を受け取り、何らかの処理を行い、key とより小さな value の集合を返す。

図 1 に、ワードカウントを行う MapReduce 処理を示す。入力は、（重複を含む）単語の列であり、Map 関数は、単語（key）とそのカウント（value、ここではすべて値 1 が付与される）をペアにして返す。Map 関数と Reduce 関数の適用の間では、Shuffle 関数が呼ばれ、key ごとに key-value ペアをソートする処理が行われる。最後に、Reduce 関数が、同じ key を持つペアを受け取り、カウントをマージして、単語とそのカウントを出力する。並列処理を行う際には、いかに計算ノード同士のやりとりを最小化することで処理の局所化を行うかが鍵となるが、MapReduce では、同じ key を持つデータを同じ計算ノード上に配

置することで局所性を実現する。

MapReduce アルゴリズムは、非常に単純なものであり、汎用的なプログラム言語（例えば、MPI (Message Passing Interface)）に比べるとその記述力は制限的である。しかし、MPI では、開発者が、プログラムの並列実行に関する様々な制約を意識しながら開発を行うのに対し、MapReduce は、開発者は Map および Reduce 関数の開発を行うだけで、並列化を意識する必要がない。Churra は、MapReduce を用いて機械学習分野で利用されるアルゴリズムを実装し、並列処理による性能向上が見込めることを報告しており [6]、MapReduce が様々な領域で使えることを示している。実際、現在様々な分野への利用が始まりつつある。MapReduce アルゴリズムを Java で実装し、オープンソースとして提供しているのが、Apache Hadoop (<http://hadoop.apache.org/>) である。Hadoop と、それを元にしたデータ解析プラットフォームについては、本特集の古関氏の解説などを参考にしてほしい。

3.2 クラウドデータベース

多くの SaaS サービスにおいてデータベースミドルウェアとそれに対するクエリ言語は必要不可欠であるが、時には数千台にも及ぶ多数のサーバー上の並列分散処理を前提とした場合、従来から使われている関係データベース (RDB) ではスケールしないという問題がある。そこでスケールアウト向けのデータベースとして、Key Value Store (KVS) が注目を浴びている。RDB と KVS の特徴を表 1 にまとめる。

典型的な KVS では、データ構造の基本はスキーマを持たない key-value ペアである。同じ key を持つデータを同じ計算ノードに配置することで局所性を確保する考え方は MapReduce と共通である。データをできるだけ均等に配置し検索を行うためのアルゴリズム（例、分散ハッシュテーブル [7] などが用いられる）、ノード故障の際にリカバリするためのデータ複製技術

表 1 RDB と KVS の特徴

	関係データベース (RDB)	Key Value ストレージ (KVS)
データ構造	関係テーブル	Key value ペア
データスキーマ	テーブルスキーマ	スキーマなし
クエリ言語	SQL	簡略化されたクエリ
大規模並列化	困難	容易
基本となる原理	ACID 特性 <ul style="list-style-type: none"> 原子性 (Atomicity) 一貫性 (Consistency) 独立性 (Isolation) 永続性 (Durability) 	BASE 特性 <ul style="list-style-type: none"> 基本的に利用可能である (Basically Available) 一時的に不整合な状態になり得る (Soft State) 最終的には一貫性が保たれる (Eventually Consistent)

図 1 MapReduce アルゴリズム (ワードカウント)

などが技術的な課題である。クエリについては、joinを許さないなど、RDBのクエリ言語であるSQLよりも制限されている。従来のRDBとSQL環境と対比して、KVS環境は、NoSQL[8]と呼ばれることもある。現在、複数のKVS実装が公開されており、実際に使ってみることができる (<http://ja.wikipedia.org/wiki/NoSQL>, <http://nosql-database.org/>などを参照のこと)。

RDBとKVSの根底に流れる思想を理解するために、従来のトランザクションシステム（とそれを支えてきたRDB）と、クラウドを前提とした広域分散型システムの特性の違いを考えてみよう。

ACID特性は、トランザクションシステムを特徴付ける特性として広く知られており、原子性 (Atomicity)、一貫性 (Consistency)、独立性 (Isolation)、永続性 (Durability) からなる。例えば、銀行における口座振替のようなシステムでは、口座間での状態が不整合にならないようにトランザクションの実行、失敗した場合のロールバックなどが行われる。

一方、クラウド環境では、複数の計算機同士がネットワークを介して分散しているため、一貫した状態を保つことが技術的にもコスト的にも困難になる。そこで、厳密な一貫性を緩和する代わりに可用性を重視するBASE特性[9]が注目されている。BASE特性は、基本的に利用可能である (Basically Available)、一時的に不整合な状態になり得る (Soft State)、最終的には一貫性が保たれる (Eventually Consistent) という性質から構成され、現在のクラウド環境の特徴をうまく示している³。

3.3 解析クラウド

クラウド・コンピューティング適用のエリアとしては、メールやSNSなどのソーシャルサービス、特定期間に多数のサーバーを必要とする開発・テスト環境、複数の企業や業界を連携するためのインダストリクラウドなどが考えられるが、様々な情報源からの巨大なデータを解析し、その結果を元に高価値サービスを行う解析クラウド (Analytics Cloud) が現在注目を集

³ クラウド環境でどのようなシステムを動かすべきかは、技術的およびビジネス的な側面から、活発な議論が行われている。例えば、現在ACID特性を満たしたシステムのうちの一部は、厳密な一貫性を必要としないのではないかと (BASE特性を満たす環境で構築することでコストを下げることができる)、逆に、クラウド環境でトランザクションシステムを構築するにはどのような課題を克服しなければならないか、などは興味深いテーマである。

めている。

解析サービスをクラウドで行う利点を幾つか挙げてみる。

- データ収集の容易さ：インターネット経由で、各種センサーやデバイスを含む様々な情報源から発生するデータを容易に収集することができる。また、無償で公開されている各種データ (報告書、文献抄録、特許情報など) を定期的に収集、加工することが容易である。
- 共有の容易さ：加工したデータを元に複数のユーザが解析処理を行う際に、データインデックスやアクセスのためのポータルを異なる組織で共有することで、効率のよいデータ収集・管理が可能となる。
- スケーラビリティと弾力性：扱うデータ量が巨大になっても、計算資源をスケラブルに提供することができる。

解析クラウドの例として、IBMが公開した自動車業界向け市場情報分析サービスがある[10]。米国運輸省道路交通安全局 (NHTSA: National Highway Traffic Safety Administration) が英語で公開している自動車に関する利用者からの提供情報をクラウドに取り込み、テキストマイニング技術を適用することで、複数の利用者が同時に最新データの分析を行うことができる。NHTSAのデータは公開され定期的に更新されており、個々のユーザがそれぞれ更新データを取り込み分析用に加工するにはコストがかかるが、パブリック・クラウドサービスがデータ更新を行うことで、利用者の手間をはぶくことができる。

このように解析クラウドは、今後の展開が期待されるが、解くべき課題もある。例えば、NHTSAのような公開された情報であれば問題ないが、医療情報や、車の走行データ (GPS情報など)、個人情報を含むデータの場合は、プライバシーの問題が生じる。また、大量のセンサーデータを扱う際には、データの信頼性が問題になってくる場合もあるだろう。このような課題を克服するための研究開発が今後進むと思われる。

4. おわりに—今後の展望

本稿では、現在注目を集めているクラウド・コンピューティングの歴史や技術要素について説明した。サービス提供基盤としてのクラウド・コンピューティングは、現在、一般ユーザや企業ユーザへの普及が進みつつあり、今後の展開が期待される。本節では、2つ

の方向性について紹介したい。

一つは、クラウド・コンピューティングの企業アプリへの浸透である。コストの低減やIT資産の弾力的な運用、データや処理の分散配置のために、今後、企業でのクラウド使用は増大していくと思われる。本稿でも紹介したACID特性を持ったトランザクションシステムがクラウドに移行するにはまだまだチャレンジが多いが、周辺アプリの段階的な移行の過程の中で議論が進められていくだろう。

もう一つは、携帯やスマートフォンのモバイルデバイスや各種センサーやメーター、車や医療機器などのエンドポイントがクラウドに接続された世界におけるデータ解析やそれに基づくサービスの発展である。今後、インターネットは人と計算機そして物理的な「モノ」をつなぐ“Internet of Things” [11]へと発展していくことが予想されている。ここでもクラウドの果たす役割は大きい。

クラウド・コンピューティングの普及を促す上で大きな力となるのが、標準化である。クラウドそのもの、あるいはシステム管理、ストレージ、セキュリティなど、クラウドを支える周辺技術について、それぞれ標準化コミュニティが立ち上がっているが、その歩みはこれからであり、議論の加速が望まれる（標準化コミュニティのリストは、<http://cloud-standards.org>を参照されたい）。

2011年3月11日に日本を襲った東北地方太平洋沖地震は、社会インフラとしてのIT技術に対して、多くの課題を投げかけた。一方で、被災者やその家族をつなぐコミュニケーションツールや様々な情報のリアルタイムでの提供など、クラウドを含むIT技術が果たした貢献も大きい。そして、復興を支える技術とし

て、災害やテロに対して頑健で適用力を持つクラウド技術の発展が大いに求められていくだろう。

参考文献

- [1] P. Mell and T. Grance: The NIST Definition of Cloud Computing v15, National Institute of Standards and Technology (NIST) (2009).
- [2] M. Armbrust et al.: Above the clouds: A Berkeley view of cloud computing, *Science*, 53 (2009), 07-013.
- [3] 丸山不二夫他, 雲の世界の向こうをつかむクラウドの技術 (アスキー・メディアワークス 2009).
- [4] ルイス・アンドレ・バロツソ, ウルス・ヘルツル, Google クラウドの核心—巨大データセンターの変貌と経済学, 丸山不二夫, 首藤一幸, 浦本直彦 (監修), 高嶋優子, 徳弘太郎 (翻訳) (日経 BP 社 2010).
- [5] J. Dean and S. Ghemawat: MapReduce: Simplified Data Processing on Large Clusters, *Communications of the ACM*, 51 (2004), 1-13.
- [6] C.-T. Chu et al., B. Schölkopf, J.C. Platt, T. Hoffman, Eds.: Map-Reduce for Machine Learning on Multicore, Architecture 19, 281, The MIT Press (2007).
- [7] Distributed Hash Table, Wikipedia, http://en.wikipedia.org/wiki/Distributed_hash_table
- [8] M. Stonebraker: SQL databases v. NoSQL databases, *Communications of the ACM*, 53 (2010), 10.
- [9] D. Pritchett: BASE: An Acid Alternative. *ACM Queue*, 6, (2004), 48-55.
- [10] IBM プレスリリース: 自動車業界向け市場情報分析サービスを、クラウドで提供 (2010) <http://www-06.ibm.com/jp/press/2010/02/1601.html>
- [11] Internet of Things, Wikipedia, http://en.wikipedia.org/wiki/Internet_of_Things