

AWS環境における仮想マシンと仮想ネットワークの 統合管理の実現と評価

井上 裕太郎¹ 手塚 詩央里¹ 横山 和俊¹

概要：IaaSのサービスであるAmazon Web Servicesは、システム開発において頻繁に利用されている。一般的にシステム開発を行う際、システム全体のバックアップを取っておく必要があるが、AWSにはシステム全体のキャプチャ及びリストアの機能が存在しないため、リソース別にそれらを行う必要がある。また、実施者はその方法などを学習しなければならないため、多くの時間や負担を強いられる場合がある。本稿ではAmazon Web Servicesで構築された仮想マシンと仮想ネットワークの統合的なキャプチャ及びリストアを行う統合管理方式を提案し、その有効性について述べる。

1. はじめに

Amazon Web Services(以降はAWSと略す)は、2006年の開始以降、クラウドコンピューティングの先駆けとなるサービスとなり、現在IaaSの中でトップのシェアを誇るサービスである[1]。AWSを利用することで、Amazonのデータセンタ上に自由かつ迅速に仮想マシンや仮想ネットワークなどのリソースを起動できることから、システム開発にAWSを利用する例は多い。一般的にシステム開発を行う際、仮想マシンと仮想ネットワークを含めたシステム構成全体のバックアップを取っておくことで、予期せぬ動作などでシステムに問題が発生した際に復旧を行うことができる。AWSの場合、EC2やVPCを利用して作成した仮想マシンや仮想ネットワークは一つの操作によって簡単に消去ができるため、そのような操作によりシステム全体が消去されてしまう可能性がある。しかし、AWSではシステム構成全体のキャプチャ及びリストアを一括で行える機能は存在しない。

現状のAWSでシステム全体のキャプチャ及びリストアを実施するには、各リソースそれぞれ個別に操作を行う必要がある。AWSが提供しているAWSマネジメントコンソールを使用してキャプチャを行う場合、各リソースについて別々のページで設定内容を記録する必要がある。リストアをする場合は記録した内容を元に、各リソースのページでリストアの操作を行う。

これらの操作には多くの操作時間と操作回数を要し、システムの構成規模が大規模なものになる程実施者の負担も

増加していく。また、AWSで構成されたシステム構成全体のキャプチャ及びリストアの操作手順などはAmazon側から明確にされていないため、実施者はキャプチャ及びリストアの方法を学習する必要がある。

これらのことから、現状のAWSにおいてシステム構成全体をキャプチャ及びリストアする際には多くの学習時間や操作時間、操作回数を必要とし、操作時間や操作回数においては対象のシステムが大きい程その負担が増大していく。

本研究ではAWSで構築された仮想マシンと仮想ネットワークを対象にした統合管理方式を提案し、その有効性を評価する。統合管理方式では、これらの問題点の解決方法として、インターフェースの改良と仮想マシンと仮想ネットワークの自動統合保存を実現している。インターフェースの改良では、キャプチャ及びリストアを行う際にAWSで各リソース別画面で行っていた操作を一画面で行えるようにする。自動統合保存ではシステム内のリソース同士の相互関係や設定を自動で分析し保存し、保存した情報を元に各リソースのリストアを行うことで、相互関係を保ったまま各リソースをリストアできるようにする。

2. 統合管理方式

2.1 統合管理方式の概要

統合管理方式ではAWSで構築されたシステムを対象に全構成情報取得機能がAWS内部の全構成情報のスナップショットを取得し、キャプチャ機能がスナップショットを元にVPC内に存在するリソースのバックアップをとる。そしてリストア機能がシステム全体を展開し復旧する。AWSでシステム開発を行う場合、1つのVPC内にサ

¹ 高知工科大学 情報学群
Kochi University of Technology

ブネットや EC2 などの各リソースを立ち上げているシステム構成が多いため [2]，統合管理方式にてキャプチャ及びリストアを行う対象は 1 つの VPC 内に存在する全てのリソースとする．図 1 に統合管理方式の概略図を示す．

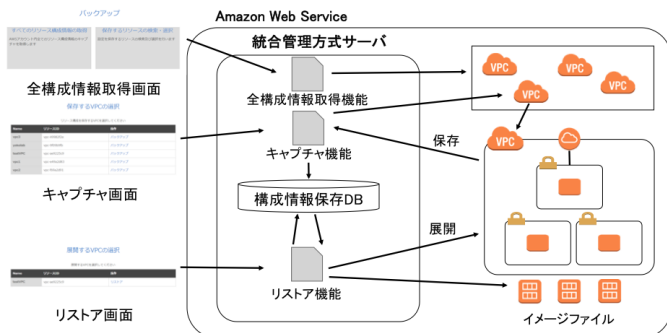


図 1 リソース同士の依存関係

2.2 統合管理方式の要件

統合管理方式を検討するにあたり，統合管理方式に求められる要件を表 1 示す．

表 1 統合管理方式に求められる要件

要件	理由
操作インターフェースは独自の GUI	AWS マネジメントコンソールはカスタマイズができないため
扱うシステムの構成は 1 から 5 セグメント	AWS でシステム開発を行う事例のほとんどが 1 から 5 セグメントで構築されているため
キャプチャ及びリストアの対象は VPC 内に含まれる全てのリソース	1 つの VPC 内に各リソースを立ち上げているようなシステム構成が多いため
同一設定情報を持つシステム構成の複数構築が可能	1 つのシステムから複数のパターンを作成して比較できるようにするため
同一設定情報を持つシステム構成同士の通信は不可	Elastic IP アドレスの重複が発生するため

2.3 リソースのキャプチャ方法

統合管理方式でリソースのキャプチャを行う場合，最初に AWS 内に存在する全構成情報を AWS の API を使用して取得し，存在する VPC の全容を把握する．その後保存する VPC を選択し，取得した全構成情報を解析し選択した VPC 内の各リソース情報を抽出してデータベースへ保存する．EC2 インスタンスについては AWS の API を使用してイメージファイルも作成する．

2.4 リソースのリストア方法

統合管理方式では，データベースに保存されたリソースの構成情報を保存して，システム構成を展開する．保存された構成情報を元に AWS の API を使用して各リソースを起動させ，構成情報の設定を行う．EC2 インスタンスは AWS の API を使用してイメージファイルから起動する．

リストアの際，各リソース同士の依存関係に注意しながら起動を行う必要がある．例えばサブネットを起動する場合 VPC ID を指定することで特定の VPC にサブネットを起動することができるが，VPC ID を指定しない場合サブネットは起動することができない．このようなリソース同士の依存関係を図 2 に示す．

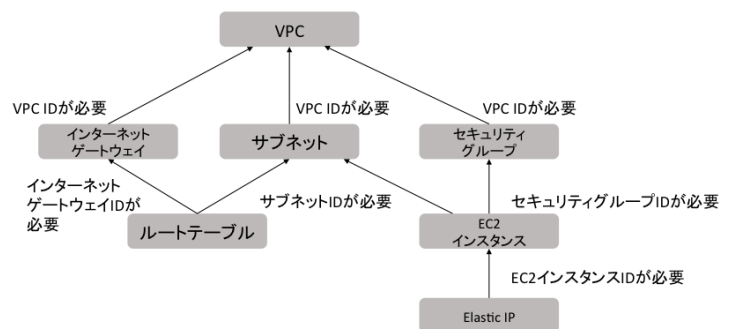


図 2 リソース同士の依存関係

3. 統合管理方式の実装方法

3.1 統合管理方式の構築環境

統合管理方式サーバは EC2 で仮想マシンとして構築する．統合管理方式サーバの詳細情報を表 2 に示す．

表 2 統合管理方式サーバの詳細情報

OS	Ubuntu
CPU	Intel Xeon CPU E5-2676
メモリ	1GB

Web インターフェースは HTML と CSS を用いて独自の GUI を作成し，プログラミング言語は PHP を使用した．AWS の操作については AWS SDK for PHP を使用した．

3.2 全構成情報の取得

図 3 のすべてのリソース構成情報の取得ボタンをクリックすることで，AWS 内に存在する全てのリソースの構成情報を AWS の API を使用して取得する．この時取得した全構成情報のスナップショットは Amazon S3 に JSON 形式で保存される．この時使用した API を表 3 に示す．その後，保存されたスナップショットを統合管理方式内に AWS の API を使用して保存する．この時使用した API を表 4 に示す．

バックアップ



図 3 全構成情報取得画面

表 3 全構成情報取得処理に使用した AWS の API

API	操作
putConfigurationRecorder	リソース構成情報を取得するためのレコーダを設定
putDeliveryChannel	S3 バケットに保存するためのチャンネルを設定する
startConfigurationRecorder	ConfigurationRecorder の設定に従ってリソース構成情報のレコードを始める
deliverConfigSnapshot	ConfigurationRecorder によって取得したリソース構成情報を DeliveryChannel の設定に従って S3 バケットへ JSON 形式で保存する

表 4 JSON ファイル取得処理に使用した AWS の API

API	操作
listObjects	指定したバケット内のディレクトリに存在するファイル名を取得する
getObject	指定したファイルを指定したローカルディレクトリへ保存する

3.3 VPC の統合キャプチャ

図 4 のバックアップボタンをクリックすることで、選択した VPC の統合キャプチャを行うことができる。選択した VPC の ID を元に全構成情報から VPC 内のリソース ID を取得し、データベースへ格納する。データベースへ保存したリソースの ID を元に全構成情報からリソースの設定情報を取得し、データベースへ保存する。

EC2 インスタンスを保存する際はリソースの設定情報をデータベースへ保存した後、保存したインスタンス ID を使用して EC2 インスタンスのマシンイメージを作成する。この時使用した AWS の API を表 5 へ示す。この API によって返却されたイメージ情報からマシンイメージ ID を取得し、データベースへ保存する。

保存するVPCの選択

リソース構成を保存するVPCを選択してください

Name	リソースID	操作
vpc3	vpc-49982f2e	バックアップ
yokolab	vpc-9f09b9fb	バックアップ
testVPC	vpc-ae9225c9	バックアップ
vpc1	vpc-e49a2d83	バックアップ
vpc2	vpc-f69a2d91	バックアップ

図 4 キャプチャ画面

表 5 マシンイメージ作成処理に使用した AWS の API

API	操作
createImage	指定したインスタンスのマシンイメージを作成し、作成したイメージの情報を連想配列で返却する

3.4 VPC の統合リストア

AWS のリソース同士には図 2 のような依存関係があるので次の順番でリソースのリストアを行う。

- (1) VPC
- (2) インターネットゲートウェイ
- (3) サブネット
- (4) ルートテーブル
- (5) セキュリティグループ
- (6) EC2 インスタンス

リストアに使用した AWS の API を表 6 に示す。

4. 評価

4.1 評価の目的

本稿では AWS で構築されたシステムをキャプチャ及びリストアする際に、多くの操作時間と操作回数及び学習時間を要する問題を解決する方法として統合管理方式を実装した。そこで、この方式の有効性を評価するため、通信インフラの基礎知識かつ AWS の基礎知識を有した 5 名の被験者を対象に、操作時間と操作回数の評価及び学習時間の評価を行う。

4.2 操作時間と操作回数の評価

4.2.1 評価項目

AWS マネジメントコンソールを用いた従来の方法と統合管理方式で、システム構成のキャプチャ及びリストアを行う場合の計測する項目を表 7 に示す。

評価は 1, 3, 5 セグメント構成のシステムをキャプチャ及びリストアの対象とする。これら 3 つのシステムに含まれるリソースを表 8 に示す。

4.2.2 評価結果と考察

各被験者の操作時間を図 5 に示し、各被験者の左クリッ

ク数を図 6 に示す。また、各被験者の画面遷移数を図 7 に示す。

表 6 リストア処理に使用した AWS の API

API	操作
createVpc	VPC を作成する
describeRouteTable	指定条件にあるルートテーブルの概要を取得する
describeSecurityGroup	指定条件にあるセキュリティグループの概要を取得する
createInternetGateway	インターネットゲートウェイを作成する
attachInternetGateway	指定された VPC にインターネットゲートウェイを関連付ける
createSubnet	サブネットを作成する
createRouteTable	ルートテーブルを作成する
associateRouteTable	指定されたサブネットと関連付ける
createRoute	指定されたルートテーブルにルートを作る
createSecurityGroup	セキュリティグループを作成する
authorizeSecurityGroupIngress	指定されたセキュリティグループのインバウンド設定を行う
runInstances	EC2 インスタンスを起動する
associateAddress	指定された EC2 インスタンスに指定された IP アドレスを関連付ける

表 7 評価項目

項目	定義
操作時間	キャプチャ及びリストアの実行時間から終了時間までに要した操作時間
クリック数	キャプチャ及びリストアの実行に要した左クリックの回数
画面遷移回数	キャプチャ及びリストアの実行時に別画面へ移動した回数

表 8 評価対象のシステムに含まれるリソース

	1 セグメント構成	3 セグメント構成	5 セグメント構成
VPC	1	1	1
インターネットゲートウェイ	1	1	1
サブネット	1	3	5
ルートテーブル	2	2	2
セキュリティグループ	1	2	3
EC2 インスタンス	1	3	5
Elastic IP	1	1	3

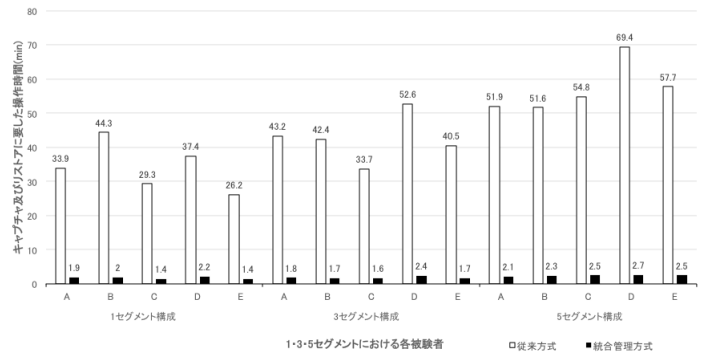


図 5 各セグメント構成における被験者の操作時間比較

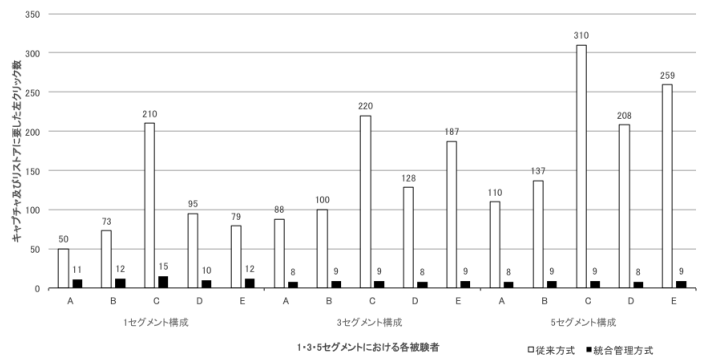


図 6 各セグメント構成における被験者の左クリック数比較

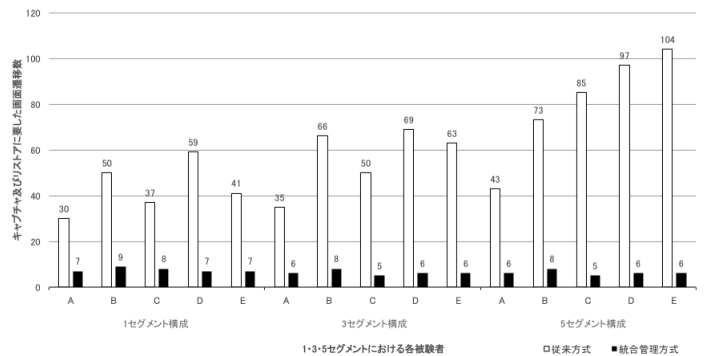


図 7 各セグメント構成における被験者の画面遷移数比較

図 5, 6, 7 より統合管理方式を用いることで 3 つの評価要素すべてを削減できたことが分かる。具体的には操作時間では約 95%, 左クリック数では 91%, 画面遷移数では約 87%削減できている。また、従来の方式ではセグメント数が増えるにつれて被験者の評価要素の値が増加しているのに対して、統合管理方式では全ての評価要素においてほぼ一定の値を保っていることが分かる。

操作時間の大幅な削減の理由は、従来方式ではキャプチャの際に各リソースの構成情報を記録する必要があったが、統合管理方式ではその操作をする必要がなくなったからであると考えられる。操作回数の削減の理由は、キャプ

チャ及びリストアを行う際、従来方式では各リソース別の画面での操作であったが、統合管理方式ではキャプチャとリストアを一つの画面で行えることが挙げられる。全ての要素においてほぼ一定の値を保つことができた理由としても、これが要因していると考えられる。

これらの結果より、統合管理方式では被験者の能力に依存せず、短時間でシステム構成をキャプチャ及びリストアできることがわかった。

4.3 学習時間の評価

AWS マネジメントコンソールを用いた従来の方法または統合管理方式で、システム構成をキャプチャ及びリストアするにあたり、各方式の操作方法を記述した資料を用意した。この資料を試行前及び試行中に読んだ時間の合計を計測した。各被験者の学習時間を図8に示す。

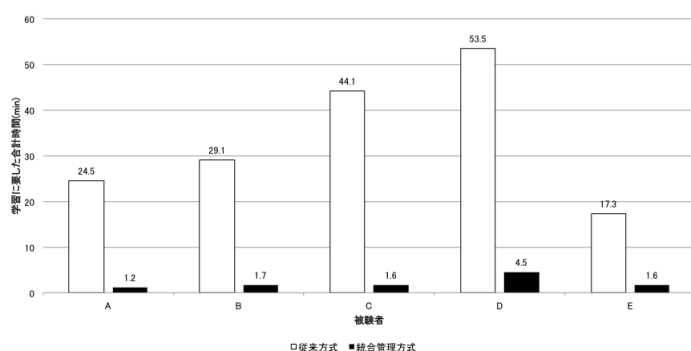


図8 各被験者の学習時間比較

図8より、従来の方法に比べ、統合管理方式では被験者の学習時間を約94%削減できたことが分かる。従来の方法では各リソースごとに別の画面での操作及び内容の記録を被験者自身が行う必要があったため、画面の操作方法や記録すべき内容の学習に時間がかかる。一方で統合管理方式では、設定内容の記録は自動的に行われ、画面の操作方法も比較的簡単なため学習にかかる時間が削減できたと考えられる。

この結果から、従来方式に比べ統合管理方式では操作手順及び操作方法の簡略化を実現できたことがわかった。

5. まとめ

本稿ではAWSで構築されたシステム構成全体のキャプチャ及びリストアに、多くの操作時間を要し、さらに実施者が操作方法や手順を学習する必要があるため、多くの負担がかかる問題を解決する方法として統合管理方式を提案し、これを実装した。4章の結果より統合管理方式を用いることで操作時間などの大幅な削減が可能であり、またユーザの能力に依存することなく短時間かつ容易にシステム構成のバックアップ及びリストアを行えることがわかった。

以上のことからこの方法の有効性を示すことができた。

今後の課題としてはAWSの他のリソースへの対応やGUIの改善が挙げられる。これらを行うことで統合管理方式で扱うことのできるシステムの幅が広がったり、ユーザの操作性を上げることができるため、より良いシステムになると考えられる。

参考文献

- [1] 加藤英雄, "決定版 クラウドコンピューティング ユーザーは雲のかなた", 共立出版株式会社, 2011.
- [2] Amazon Web Service, <http://aws.amazon.com/jp>.