サーバレスアーキテクチャーによる作業量削減効果の測定

－ビジネスアジリティは向上するのか？－

IA−001：サーバレスアーキテクチャーのビジネス適用：石野 大輔

How to write Paper

－　Format Template　－

Team’s name：Leader’s name

新たなデジタル技術が日々開発されている状況の中，企業は競争力強化のため，そのような新技術を活用してビジネスモデルを素早く創出することが求められている．近年，ビジネスアジリティを向上させるアーキテクチャとしてサーバレスアーキテクチャーが注目されている．本論文では，サーバレスアーキテクチャーの適用により，ビジネスアジリティが向上するのか検証した．検証で得られた結果から，ビジネスアジリティ向上の是非と今後の課題・展望について述べる．

This manual guides you in writing an “JGS Paper” in the proper format and structure. The guideline is prepared as an input format template to help you write a properly formatted paper by following the instructions explained in the body. The abstract of your paper should be written in this style, with five key words and phrases as shown below.

Key Words & Phrases ：執筆要領，論文テンプレート，発明提出，フォーマット，参考文献

　　　　　　　　　　 paper guideline, paper template, invention submission, format, references

１．はじめに

日々新たなデジタル技術が開発されている昨今，IT業界のみならず様々な業種でデジタル技術を駆使したビジネスが創出され，企業価値を高めている．このような状況の中，我が国では更なる競争力強化のため，各企業にデジタルトランスフォーメーション(DX)を推進していくことを強く求めているが，現状では一部の先進的な企業を除く多くの企業がDXを推進できていない。その主な原因として，既存の基幹システムの老朽化が挙げられている．年を重ねることでシステムが複雑化することで，維持管理費が高騰，また技術継承もままならなくなり，新たなデジタル技術をビジネスに組み込めない状態となっている． 経済産業省のレポートによれば，レガシーシステムが継続して使用された場合，2025年～2030年の間に最大12兆円の経済損失が生じると推定されている[1]．

そのため，DXを推進していくためには，老朽化し複雑化したレガシーシステムの改修が必要となる．しかし既存システムの改修は大規模かつ長期のプロジェクトとなるためリスクも大きく，スピーディに効果を得ることができない．そこで，頻繁に改修が求められるビジネスロジック又は新規に作成するビジネスロジックについて，新たなデジタル技術を使用して構築することで，ビジネスモデルの変化に迅速に追従できるようにする必要がある．

近年，維持管理費を削減し，ビジネスアジリティを向上させるアーキテクチャとしてサーバレスアーキテクチャーが注目されている。サーバレスアーキテクチャーは，各クラウド事業者の責任範囲が広く，ユーザは業務ロジック構築とアプリケーション開発に注力できる点が大きな特徴である．本論文では，サーバレスアーキテクチャーの適用によりビジネスアジリティが向上するのか，検証用のモデルシステムを実装することで検証を行う．以下，2章でサーバレスアーキテクチャーの特徴を述べ，3章，4章にて成果物の作成量及び構築作業時間の2つの観点での検証結果について述べる．また，5章，6章にて検証作業にて確認された課題と課題を踏まえた今後の展望を述べる．

２．サーバレスアーキテクチャーとは

　本論文では「サーバレス」，「サーバレスアーキテクチャー」の２つの用語は、2019年度JGS論文「サーバレスアーキテクチャーの適用検討プロセスの提案」[2]の定義と同意とする．

サーバレスとは、「サーバ管理を必要としないアプリケーションの構築と実行の概念」[2]のことであり、代表的な実現方法としてFaaS（Function as a Service）とBaaS（Backend as a Service）がある．これらFaaSとBaaSの個別機能を組み合わせたシステム構成をサーバレスアーキテクチャーとする．[2]

2.1 サーバレスの特徴

　サーバレスには, 各クラウド事業者ごとに様々な特徴が存在するが，ここでは代表的な特徴について述べる．

1. サーバーの構築・運用が不要

サーバーやランタイム等のアプリケーションの実行に必要な基盤環境については，図１[xx]の通り、クラウド事業者の責任の下，運用・管理されている．そのため、環境構築を行う必要がなく、運用・管理に掛かる作業負荷も大幅に削減される．



図１．サーバレスの責任モデル

1. 柔軟なスケーラビリティ

処理量に応じて，自動的なスケーリングが実施可能．

３．サーバレスアーキテクチャー適用によるビジネスアジリティ向上の検証

本章では，1章で述べたサーバレスアーキテクチャーの適用効果測定する．サーバレスアーキテクチャーを使用してグッズ購買サイトを構築し，オンプレミス環境における作業と比較することで，作業負荷が軽減されることを検証した結果を述べる．検証観点は以下のとおり．

(1)ドキュメント作成量

(2)構築作業時間

3.1仮説と前提

3.1.1.仮設

サーバレスアーキテクチャー適用により，各開発工程のドキュメント作成量減少，構築作業時間の短縮により，開発期間が短縮することを仮設とする

3.1.2.前提

(1)利用するクラウドベンダー

本検証では，代表的なクラウド事業者[4-1]であるAmazonWebServiceの各サービスを用いて，グッズ構築サイトのシステム構築を行う．

1. モデルシステム構築

検証用のモデルシステムとして，東京オリンピック2020オリジナルグッズ購買システムを構築する．同システムに必要な要求事項を要求仕様書（別紙）として纏め，要求を充足するシステムを構築する．適用するサーバレスアーキテクチャーは，検証要件に類似した要件のServerless Airline Booking（航空券購買システム例）[4-2]を参考に検討を行った．システム構成は図xに示す．

スクリーンショットの画面

自動的に生成された説明

図xx xxxxxx

3.2検証方法

3.2.1構築作業時間

要求仕様書を基に，採用するAWSサービスで必要な設計や設定について，AWS開発者ガイドやガイド記載のチュートリアルをとおして確認する．確認結果を踏まえ、モデルシステムで実装する機能ごとに、AWS各サービスに設定する値をパラメータシート（別冊x）として纏める。同シートに沿って，AWSコンソール，AWS CLIを利用して構築作業を実施する．

3.2.2ドキュメント作成量

オンプレミス環境でウォーターフォールモデルによる開発を行う場合に必要となる成果物一覧を作成する．3.2.1の構築作業結果より，サーバレスアーキテクチャー適用による開発で必要となる成果物を記載し，比較・評価の結果を評価シート（別冊x）として纏める。

3.3検証結果

3.3.1 構築作業時間

グッズ購買システムの構築作業時間は10時間であった．うち，Lambda関数コーディング時間は7時間であった．

3.3.2ドキュメント作成量

オンプレミス環境でウォーターフォールモデルによる開発を行う場合に必要なドキュメントは68本，サーバレスアーキテクチャの適用による開発で必要なドキュメントは56本であった．サーバレスアーキテクチャーの開発ではオンプレミス開発との比較でドキュメント作成量が18%減少した．

オンプレミス開発におけるドキュメント作成時のボリュームを基準として、同じドキュメントをサーバレスアーキテクチャの適用による開発を行った場合の作業ボリュームについて，以下に記載する。

・オンプレミス開発時と作業ボリュームが変わらないドキュメント：38本

・オンプレミス開発時より作業ボリュームが減少するドキュメント：18本（サーバレスアーキテクチャの適用による開発全体の32%）

オンプレミス環境で開発時よりドキュメントの作業ボリュームが減少する理由としては，利用者は要求事項に合わせて，サーバレスの各サービスが提供する機能（バックエンド接続，拡張性，バックアップ等）を利用することを選択し，設定値を決めれば良いためである．構築作業は、利用者が設定値を入力して構成を確定（保存）すると，クラウド事業者にて自動構築される．

４．考察

3.3検証結果より，サーバレスアーキテクチャーを適用することで開発期間の短縮が可能であり，ビジネスアジリティの向上が確認できた．開発期間が短縮された主たる要因は， [参照]評価シートより，非機能関連のドキュメント作成量および構築作業時間が大きく削減されたことである．一方で，本検証では，業務ロジックおよびアプリケーション開発に対しては，開発期間の短縮効果は確認できなかった．

５．サーバレスアーキテクチャー適用おける課題

今回の検証結果より、次のような課題が認められた。

1. 開発言語の選択

Lambda関数構築時に、開発者の得意な言語をそれぞれの関数で選定したため、複数言語が混在したシステムとなった。

AWSのLambdaでは、複数の言語をサポートしているため、開発工程前にコーディング規約を定める必要がある。

1. 運用・監視の複雑化

検証作業にて、エラー原因を特定するため、ログの解析を行ったが、当初ログの設定が無効になっており、問題切り分けに難航した。

また、ログの出力先や分析方法は各サービス毎に異なる為、各々のサービスの連携に関してエラー分析が複雑化し、従来システムのような統一的な運用が難しい。

1. クラウド人材の育成

AWS未経験者にて本検証を行ったため、構築作業前の事前学習に延べ１ヶ月程度費やした（AWSの基本概念、各サービス仕様等）。AWSサービスは日進月歩で進化しており、エンタープライズレベルの開発を行うためには、クラウド人材の育成が必須となる。

６．今後の展望

サーバレスアーキテクチャーを用いることで、開発時に作成するドキュメント量と開発作業量が削減され、ビジネスアジリティが向上することがわかった。 また、今回検証できていない以下の２つの観点を追求することで運用コストの削減と更なる高速開発を実現できる可能性を秘めている.

１つ目は、監視運用である.クラウド事業者からは、様々な監視サービスが提供されており、これらサービスを組み合わせて使うことで、障害時の自動復旧や日々のモニタリングよる障害の未然防止などに役立つ。

２つ目は、各サービスの疎結合化である.各アプリケーションをマイクロサービスとして独立させることで、仕様変更時にも対応範囲を局所化し、 低コスト・短納期で開発が可能となる.さらにドメイン駆動設計やアジャイル開発と組み合わせることで、更なる効果を期待できる.

ただし、これらを実現させるためには、高度な技術力とクラウドに対する豊富な知識が求められるため、企業としてはクラウド人材の育成が今後の課題である.

・アプリ開発への注力

・疎結合アプリケーション

参考文献　　　　(記述例，[3]は架空のもの)

1. 大野侚郎，「論文作法」基礎の基礎，2002年度IBMプロフェッショナル論文発表会講演社内限定資料，2002年11月
2. 木下是雄，理科系の作文技術，中公新書，ISBN4-12-100624-0，1981
3. 技術太郎，ペリルの分類を利用したリスク事象特定法，XX学会誌，Vol.5,No.3，pp.25-30，2003
4. PMI Standards Committee, *Project Management Body of Knowledge-2000 Edition*, Project Management Institute, Newtown Square, Pennsylvania, ISBN1-880410-23-0, 2000
5. IBM Japan, http://www.ibm.com/jp/, 2003.2.28

添付１：メンバーリスト ( 会社名アイウエオ順 )

添付２：Member list

(alphabetical order of company name )添付１：メンバーリスト ( 会社名アイウエオ順 )

添付２：Member list

(alphabetical order of company name )

