

OSS活用最前線、メインフレームで 新たなビジネスを作る

今回は「IT を活用しビジネスを成長させるためにはどのようなツールが必要なのか」をテーマに最新の OSS について紹介します。

ITをビジネス獲得の武器に イノベーションを起こした企業がやったこと

従来、企業における IT 活用の目的といえば、売上や人事情報の管理などの定型業務処理の効率化がメインでした。IT の定型業務処理への活用規模はさまざまですが、表やグラフを作成するのに Microsoft Excel を使うのもこれに当たります。

近年ではバックヤードの定型業務処理の効率化に IT を活用するだけでなく、ビジネス獲得の武器として活用されています。バックヤードからフロントヤードへと IT を活用する幅が広がっています。

フロントヤードにおける IT 活用の背景には、スマートフォンや SNS の普及が挙げられます。個人が気軽に情報を発信／受信できるようになったことで、社内の定型業務処理で IT システムが扱ってきたデータとは異なる種類のデータが増え続けています。例えば、Excel ファイルのように行と列で定義されたデータや、規則性のあるデータなどの構造化データに対し、Twitter の「つぶやき」や位置情報などの構造化されていないデータ（非構造化データ）はその代表的なものといえるでしょう。

個人が気軽に発する情報が増加した結果、企業は顧客の多様なニーズを明らかにできるようになりました。そして、ユーザーや消費者一人一人に向き合ったサービス——「One-to-One マーケティング」が求められるようになってきているのです。

イノベーションを起こし、人々の生活に影響を与えている企業は、IT を新たなビジネスモデル創出のために活用しています。データを直接ビジネスにつなげている企業は、例えばタクシーを 1 台も持たずに世界中にビジネス展開をする、タクシー配車サービスの「Uber」があります。Uber はスマートフォンを活用し、利用者の GPS 情報とタクシーの GPS 情報をマッチングさせ、利用者に最も近いタクシーを配送す

るサービスです。

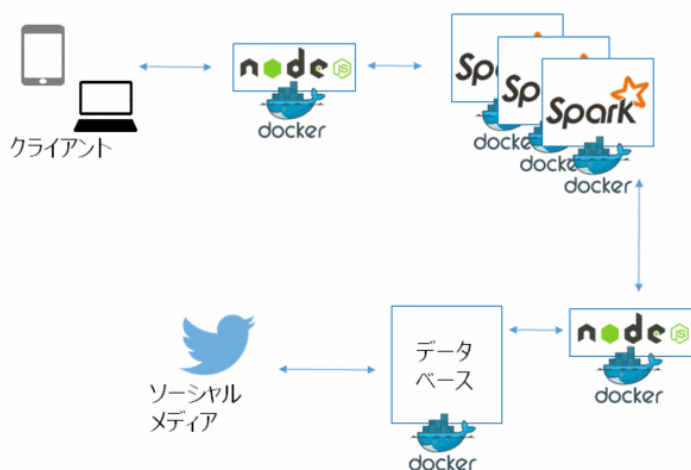
このように多様化するニーズに対応していくためには、データをいかに早く分析し、エンドユーザーに対して正確な情報を提供するかが成功の要因となります。市場ニーズの多様化や予測しきれないビジネス環境の変化に高いレベルでタイムリーに対応するためには常に最先端のテクノロジーが実装されるオープンソースソフトウェア（OSS）の活用が欠かせません。そのため、イノベーションを起こすには OSS の活用が鍵を握っています。

今、最も注目されているデータ活用のための OSS 3 大テクノロジーが「Node.js」「Apache Spark」そして「Docker」です。

Node.js はサーバサイドの JavaScript 実行環境です。Apache Spark（Spark）はデータ処理を分散環境で並列して実行するための処理フレームワークです。Spark はインメモリで処理ができるため、従来の並列分散処理ツールに比べ、処理速度が高速です。Docker はアジャイル開発に適したアプリケーションコンテナ管理ソフトウェアであり、システム運用の自動化などを実現しやすい IT インフラ基盤といえます。

なぜ、この 3 つが注目されているのかというと、ニーズの多様化や変化に対応していくために、サービスを次々に作り変えることができるスピード感が重視されているからです。

図 1 「OSS3 大テクノロジー（docker、Node.js、Apache Spark）」の連携



Docker で変化に柔軟に対応できるシステム運用基盤を整え、Node.js でフロントヤードのアプリケーションのための実行環境を構築し、Spark で価値のある情報を見つけ出す、というアプローチで、変化に対応しやすい IT 環境を整備できると期待されています (図 1)。

図 1 ではこれら 3 つのテクノロジーの連係を表しています。ソーシャルメディアから取り込んだデータを Spark で分析し、クライアントに提供しています。アプリケーションのプラットフォームとしては Node.js が稼働しています。そしてこれらのテクノロジーは、全てアジャイルな開発を可能にする Docker 上で稼働しています。

OSS 3大テクノロジー(1)「Node.js」

今回は 3 大テクノロジーのうち、サーバサイドでの処理を担う JavaScript 実行環境「Node.js」について紹介します。

○リクエスト数の増加に対応するサーバサイド技術

Node.js が注目されている背景には、スマートフォンの普及によりモバイルアクセスが増加したことが挙げられます。PC のブラウザからしか Web サイトにアクセスできなかった時代と比べると、Web アプリケーションに対するアクセス数は格段に増加しています。

Apache のような従来の Web アプリケーションサーバは、個々のリクエストに対してそれぞれスレッドを用意して処理する「マルチスレッドモデル」で実装されています。複数のスレッドを立ち上げることで並列してリクエストを処理でき

ますが、スレッドはメモリを消費してしまうため、リクエストが増えれば増えるほどメモリが逼迫 (ひっばく) します。

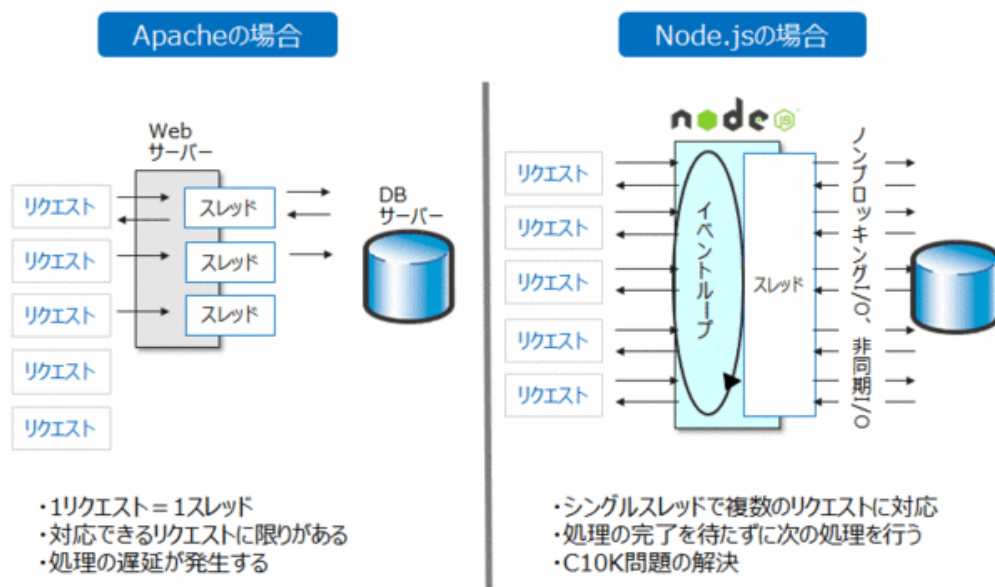
これは「C10K 問題」といわれ、サーバへの同時リクエストが 1 万件を超えると、スレッドの立ち上げ過ぎによるリソース不足で、リクエストを処理し切れなくなってしまいます。また、1 つの処理を完了しなければ次のリクエストの処理を行えないため、遅延が発生してしまいます。この場合は、サーバ自体を増やしてリクエストに対応するしか方策がありません。

Node.js は「ノンブロッキング I/O ・非同期 I/O モデル」「イベントループモデル」によってこのような問題を解決します (図 2)。Node.js 自体はマルチスレッドではなく、単一のスレッドで全てのリクエストを処理するシングルスレッドです。シングルスレッドによってメモリの消費を抑えることができますが、通常であれば 1 つのリクエストの処理が完了するまで次のリクエストを受け取ることができません。

そこで Node.js では、ノンブロッキング I/O ・非同期 I/O モデルによって、シングルスレッドでも前のリクエストの処理の完了を待つことなく、次のリクエストの処理を行うことができるようになっていきます。前のリクエストはコールバック関数によって他のリクエストの処理に影響を与えることなく結果を受け取ることができます。また、イベントループモデルによってリクエストを次々に処理に回すことができます。

Node.js であれば、Apache のように増加するリクエストに対応するためにサーバ自体を増やしていく必要がなくなります。同時に大量のリクエストに対しリアルタイムに対応できるため、例えば不特定多数が参加するソーシャルゲームや

図 2 Web サーバ「Apache」と「Node.js」の挙動の違い



オンラインゲームの統計データに活用されています。前段で紹介したタクシー配車サービスの Uber では、リアルタイムに利用者とタクシーをマッチングさせる配車システムに Node.js を用いています。

○ NoSQL データベースとの親和性

Web アプリケーションの開発環境の変化は、このような Web アプリケーションサーバだけでなく、データベースなどの周辺環境にも及んでいます。MySQL のような従来型のリレーショナルデータベースだけでなく、さまざまな形式のデータが扱えてスケーラブルな NoSQL データベースが使われるようになってきました。

久しく Web アプリケーションの開発は「LAMP スタック (Linux、Apache、MySQL、PHP)」といわれてきましたが、アクセス数の増加やアジャイルな開発に対応していくために、近年では MEAN スタック (MongoDB、Express、AngularJS、Node.js) が注目されています。

○ 公開モジュールのエコシステム

Node.js には「npm」というモジュールを管理する機能があります。豊富なモジュールやフレームワーク、リポジトリが備わっており、その数は 70 万を超えています。npm を使うことで簡単にモジュールを追加できるだけでなく、自分が作成したモジュールを公開することも可能です。npm は

Node.js のエコシステムとして機能し、これらを支えるコミュニティが存在します。

メインフレームでスタートアップ風にデータインフラを整備すると、シンプルになる

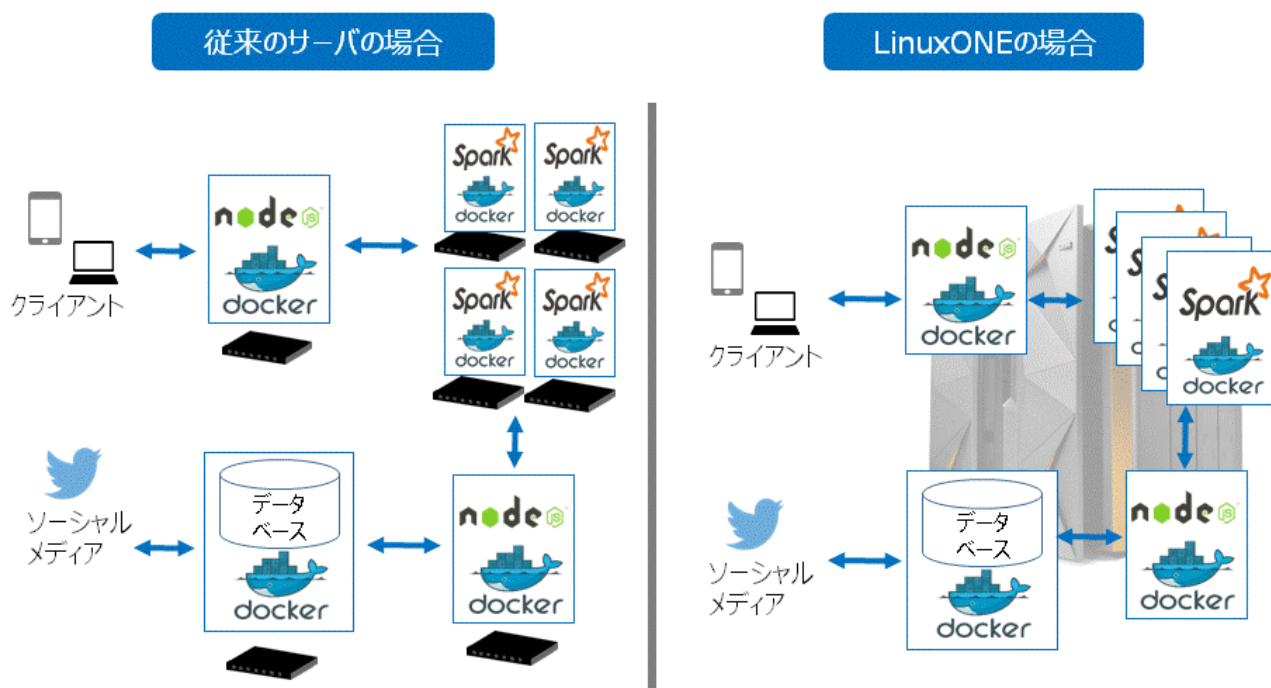
データを活用しビジネスに貢献させていくためには、Node.js や Spark などのツールを活用するだけでなく、それらを動かすインフラも重要になります。それは、ビジネスニーズが変化していく中で、システムにかかる負荷の予想が困難になっていくからです。

急激に負荷が増加した場合においても安定したパフォーマンスを維持できる高い処理能力がインフラに必要になります。また、セキュリティの確保も重要になります。データを安全に保管し、サイバー攻撃に対応したインフラである必要があります。

このようなシステム基盤の要件を満たすことができる IT インフラの 1 つが LinuxONE (<http://www-03.ibm.com/systems/jp/linuxone/>) です。

LinuxONE はもちろん Node.js、Spark、Docker といった OSS に対応しています。LinuxONE は高負荷な状態でも安定して稼働し、暗号化においても最高水準の技術を実装しています。各コンポーネントを冗長化することで無停止を実現します。また I/O 処理についても、ノード間のメッセージ通信が多くなったとしても遅延なくパフォーマンスを発揮

図 3 LinuxONE によるサーバ統合



できます。

また、物理/仮想を問わず、従来のサーバ環境で図 1 のようなシステムを構築する場合、Node.js や Spark、データベースなど、複数のコンポーネントごとにサーバを分散させる必要があるため、インフラが複雑になりがちですが、LinuxONE であれば、複数のコンポーネント全てを 1 台に

統合させることが可能であり、複雑なインフラから開放されるという利点があります（図 3）。

LinuxONE は定型業務処理を行うための従来のような IT インフラではなく、「イノベーションを起こし、ビジネスをドライブし続けられる次世代 IT インフラ」として位置付けられます。

著者：栗村 彰吾

さまざまなオープンソースソフトウェア（OSS）に携わり、現在は「LinuxONE」を中心に、提案および構築を担当。お客さまのミッションクリティカルな業務にふさわしいシステムの提供に尽力する傍ら、LinuxONE の活用に関するセミナー講師として「メインフレームの価値」や「なぜ今メインフレームか」を啓蒙する活動にも従事する。

お問い合わせ先▶ **日本アイ・ビー・エム株式会社** 〒103-0015 東京都中央区日本橋箱崎町19-21 http://ibm.biz/linuxone_jp
IBMアクセスセンター **0120-550-210**

※この冊子は、キーマンズネット (<http://www.keyman.or.jp/>) に 2016 年 12 月に掲載されたコンテンツを一部改訂・再構成したものです。 <http://www.keyman.or.jp/kc/30009341/>