

Webアプリケーション

...

基本構造・動作環境

Webアプリケーション



Webアプリケーション(以後Webアプリ)とは、インターネットを経由して利用するアプリケーションのことです。多くの場合、Webブラウザで利用します。例えば、YoutubeやFacebookなどはモバイル端末向けに専用アプリを提供していますが、同じ機能をWebブラウザで利用することができます。

Webアプリの利点として、通常Webブラウザはデバイス(OS)ごとにすでに提供されているので、専用アプリを提供しなくても利用できます。

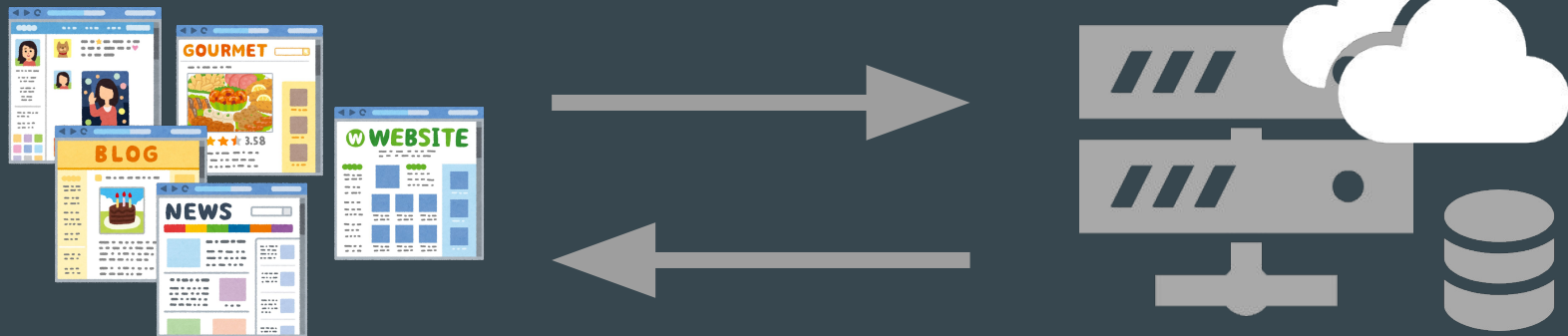
WebアプリはWebサーバー上で動作するプログラムです。PHPはWebアプリ専用のプログラミング言語ですが、PythonやJavaでもWebアプリを開発することができます。それらはWebフレームワークが外部から提供されていたり、標準の拡張機能を使って実現しています。



Webアプリケーション

つまりWebアプリは、Webサーバー側で動的にWebページを作成するプログラムが動作するソフトウェアです。

なので、SNSやショッピングサイト、CMSで制作されたブログなどはWebアプリと言えます。



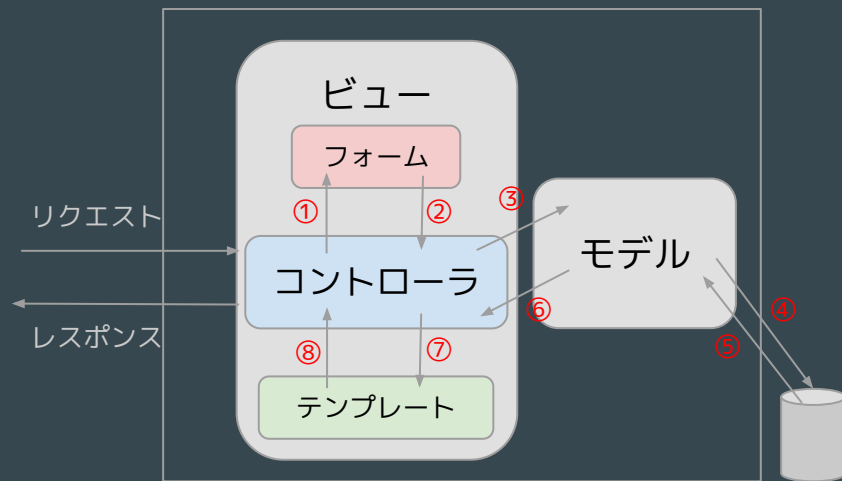
近年では高速で動作し各種メディアの再生向上や、オフライン実行などの技術革新が進み、多くのサービスがWebアプリで提供されている

基本構造

Webアプリもデスクトップアプリやモバイルアプリと同様に「MVCモデル」の構造となります。Webアプリの場合、ブラウザ画面に表示されるフォームなどを操作して動作するアプリの内部構造となります。

要素	役割
モデル(Model)	アプリのメイン処理やデータ格納など
ビュー(View)	ユーザーが操作・閲覧する画面を表示
コントローラ(Controller)	ユーザーからの要求を受け取り、処理をモデルに依頼したり必要な表示をビューに依頼

*言語によって詳細は異なる



動作環境

Webアプリを動作させるためには、Webサーバー側でクライアントからのリクエストに対してプログラムを動作させる仕組みが必要となります。

その仕組みがCommon Gateway Interface(CGI)と言われるものです。

現在では、より高速化したFastCGIやWSGIなどの手法が一般的になっています。

そしてプログラムが動作時に扱うデータを永続的に効率よく操作できるようにリレーショナル・データベースが利用されています。

これらの環境を造語としてLAMP環境と呼びます。

※LAMP: Linux, Apache, MySQL, PHP/Python

*実際には上記WebサーバーやDB、プログラム以外にも多く利用されている



仮想環境（サーバー仮想化）

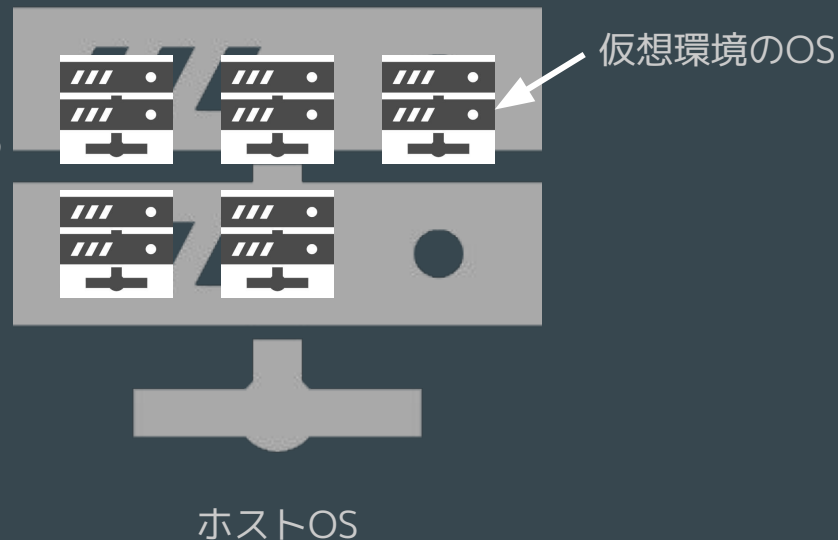
1つのOS上に複数のOSを稼働させ、各種サーバー環境を実現する技術です。

【主なメリット】

- 運用の手間やコストを削減できる
- 物理サーバーのリソースを効率よく利用できる
- 各種サーバーの追加が比較的容易にできる
- 新しいOSに対応していないシステムも利用できる
- 災害などでシステム異常が発生しても比較的早く復旧できる

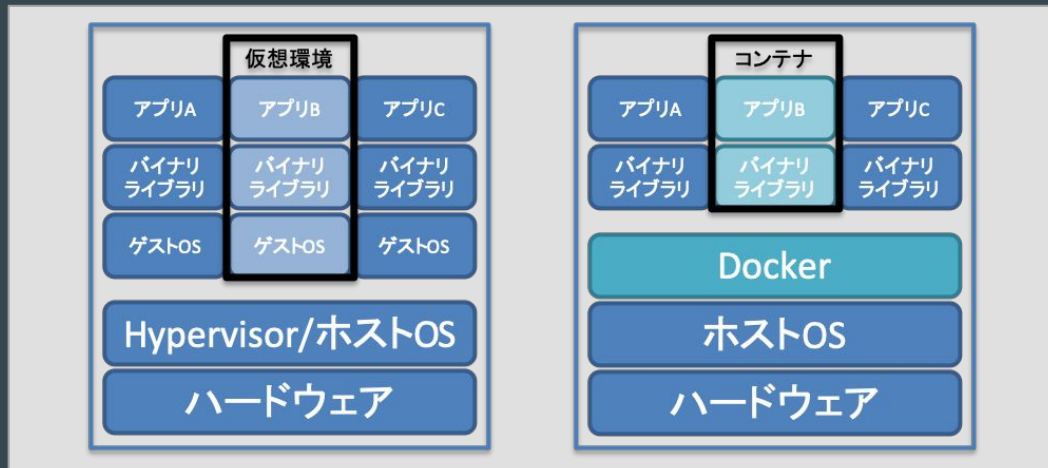
【主なデメリット】

- 構築と運用に専門的な知識が必要となる
- 小規模環境ではあまりメリットがない
- 物理サーバーより性能が低下する場合もある
- 障害の範囲が大きくなる恐れがある



Docker

ホストOS（Windows、Mac、Linux）上で、仮想環境のOSなしにWebサーバーなどのサーバーで動作するアプリを別のOS上にあるかのように動作させる技術です。仮想環境とは、一般的にはホストOS上に別のOSを動作させる技術です。Dockerは仮想環境のOSがなくてもアプリケーションを動作させる技術です。Dockerではコンテナでアプリケーションが動作します。



Dockerの技術は仮想環境よりもさらにパフォーマンスの向上が期待できるものとして注目されています。

サーバー構築

各言語で作成したWebアプリが動作するサーバー環境をそれぞれ構築します。

サーバー環境はDockerを用いて構築します。

複数のコンテナで構成するのでDocker Composeを利用します。

ComposeはYAML形式のファイルで設定を行い、1つのコマンドで設定内容に基づいたすべてのサービスを生成・起動します。

