

Oracleホワイト・ペーパー
2011年5月

Oracle Tuxedo : 動的言語のためのエンター プライズ・プラットフォーム

はじめに

スクリプト言語としても知られている動的言語は、以前から使用されていますが、最近になって、さまざまな理由から重要性を増しています。動的言語は、インタープリタ型言語であり、C/C++およびJavaとは異なり、コンパイルおよびバイト・コードを含みません。これにより、代替ソリューションよりも開発、テストおよび配置のサイクルが大幅に短縮されます。アプリケーション・コードに変更が加えられた時点で、テストおよび導入が行える状態になっています。プログラミング言語機能を侵害することはありません。動的言語は機能が豊富です。多くの動的言語はオブジェクト指向プログラミングをサポートし、コンパイル型プログラミング言語に相当する機能を含んでいます。また、アプリケーション、特にWebアプリケーションの開発に役立つ多数のフレームワークやライブラリがあります。たとえば、Ruby用のRails、Python用のDjango、PHP用のSymfonyなどの開発フレームワークがあります。これらのフレームワークにより、開発者に豊富な構成メンバーが提供され、アプリケーション開発は容易になります。これらのフレームワークとは別に、多数のサンプル・コードをWeb上で使用でき、アプリケーション開発の迅速化に役立ちます。そして、ほとんどの動的言語は直感的で比較的習得しやすいため、多数のプログラマーがこれらの言語に精通しています。スキルを持つ人材の増加により、新しいアプリケーション開発のコストを軽減し、市場提供開始までの時間を短縮できます。

このような動的言語は、オープンソースの領域に由来します。このオープンソースの性質により、動的言語は迅速に進化し、広範なインストール・ベースを持っています。標準化団体がないことを否定的にとらえる場合もありますが、今や充実した動的言語をエンタープライズ・アプリケーションに使用するために必要な制御メカニズムが十分に存在します。

多数の動的言語が、アプリケーション開発に使用されています。それらの動的言語の中でも、PHP、PythonおよびRubyは以前から使用されており、進歩してきました。ここでは、特にPHP、PythonおよびRubyに焦点を当てます。これらの動的言語は、広く受け入れられています。これらの言語を使用して新しいアプリケーションを開発する企業は、ますます増加しています。これらのアプリケーションは、重要なビジネス機能を提供し、一般的に、部門レベルで重要なニーズを実現します。これらのアプリケーションは、状況依存型アプリケーションと呼ばれることもあります。ビジネスでは、コンセプトから本番稼働までの処理を非常に迅速に行うことを必要としており、これら3つの言語はこの領域で十分な役目を果たしています。

一般的に、ほとんどの状況依存型アプリケーションのアーキテクチャには、配置用のHTTPサーバーや開発用の動的言語フレームワークなど、多数のコンポーネントが含まれています。ただし、アプリケーション・ロジックのコンテナとして機能するアプリケーション・サーバーはありません。これは、ライフ・サイクル管理、監視、またはアプリケーション・サーバーに要求される他の機能がないことを意味します。有効なアプリケーション・サーバー・プラットフォームが存在しない理由の1つに、ほとんどの状況依存型アプリケーションでは、最初の時点での設計の際に、将来的なRASPの要件（信頼性、可用性、スケーラビリティおよびパフォーマンス）が考慮されないことが挙げられます。本質的に、これらのアプリケーションには、特定の問題を解決するための迅速なソリューションを提供することが求められており、市場提供までの時間が重要です。しかし、これは状況依存型アプリケーションにスケーラビリティや高可用性が必要ないことを意味しているわけではありません。状況依存型アプリケーションの多くは、ビジネスとともに成長しています。状況依存型アプリケーションは、ユーザーの現在の問題を解決することができ、迅速に拡張できるという理由から、普及が進み、重要なものになっています。そこで、スケーラビリティおよび可用性が、資産を維持するために重要になります。

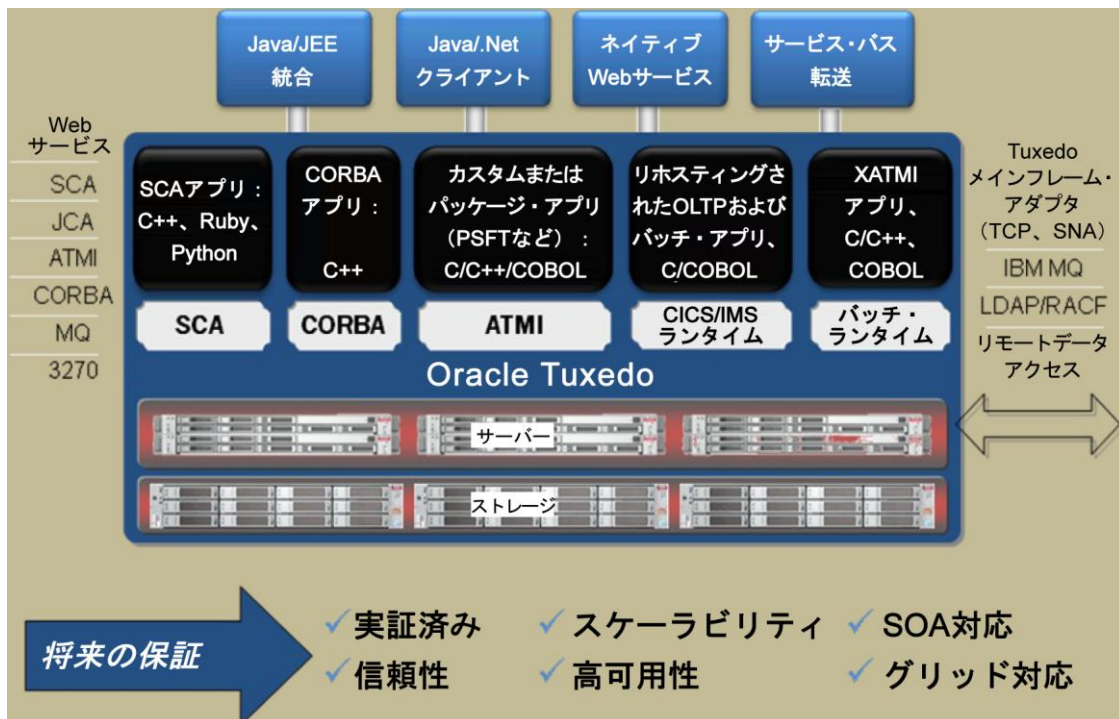
また、状況依存型アプリケーションの使用が組織とともに拡大するにつれ、ゆくゆくは、他のエンタープライズ・アプリケーションと通信する必要があります。これらの他のエンタープライズ・アプリケーションは、異機種環境においてJava、C/C++、またはCOBOLで記述されている可能性があります。状況依存型アプリケーションであれエンタープライズ・アプリケーションであれ、さまざまなアプリケーション間で通信が可能になるよう、状況依存型アプリケーションがエンタープライズ・レベルのアプリケーションになる手段が必要です。では、この問題をどのようにして解決できるでしょうか。この点を考えながら、Oracle Tuxedoを紹介していきます。

Oracle Tuxedoの概要

Tuxedoは、C/C++、COBOL、PHP、PythonおよびRubyで記述されたアプリケーションを開発および配置するためのアプリケーション・サーバーです。Tuxedoは、さまざまなプログラミング言語で記述されたビジネス・ロジックのコンテナをはじめ、アプリケーション・サーバーに要求された機能を提供します。Tuxedoは、起動/停止、一時停止/再開、より多くのインスタンスの動的な開始、再起動など、これらのサービスのライフ・サイクル管理も提供します。さらに、Tuxedoは、高可用性およびスケーラビリティを実現するためのクラスタリング機能を提供します。

アプリケーションは、XATMI、SCA、CORBAなどの多数のプログラミング・モデルの1つと、サポート対象のプログラミング言語の1つで開発できます。C/C++、COBOL、PHP、PythonおよびRubyで記述されたサービスおよびアプリケーションは、同一のアプリケーション・サーバー内で共存でき、

Tuxedoランタイムが提供するすべてのインフラストラクチャ・サービスに同等にアクセスできます。Tuxedoランタイム内でホストされると、当然、異機種アプリケーション間の相互操作が簡単になります。Tuxedoのサービスは、メタデータ主導のアプリケーション開発の手法を使用して開発でき、ミドルウェア特有のAPIやインフラストラクチャの詳細について心配する必要はありません。



Tuxedoは、そのリソース多重化機能により、標準のデータベース・ドライバおよびAPIを使用しながら、データベース中心のアプリケーションのスケラビリティを向上させることができます。たとえば、Oracle Databaseにアクセスする際に、データベースへの接続がTuxedoによって多重化される一方で、アプリケーションは依然としてPython向けのcx_oracleを使用します。言い換えると、すべてのクライアントがそれぞれデータベースへの接続を開く必要がないということです。アプリケーションのインスタンスそれぞれにデータベースへの接続が1つ確立され、受信リクエストすべてにその同じ接続が再利用されます。

リニア・スケラビリティ

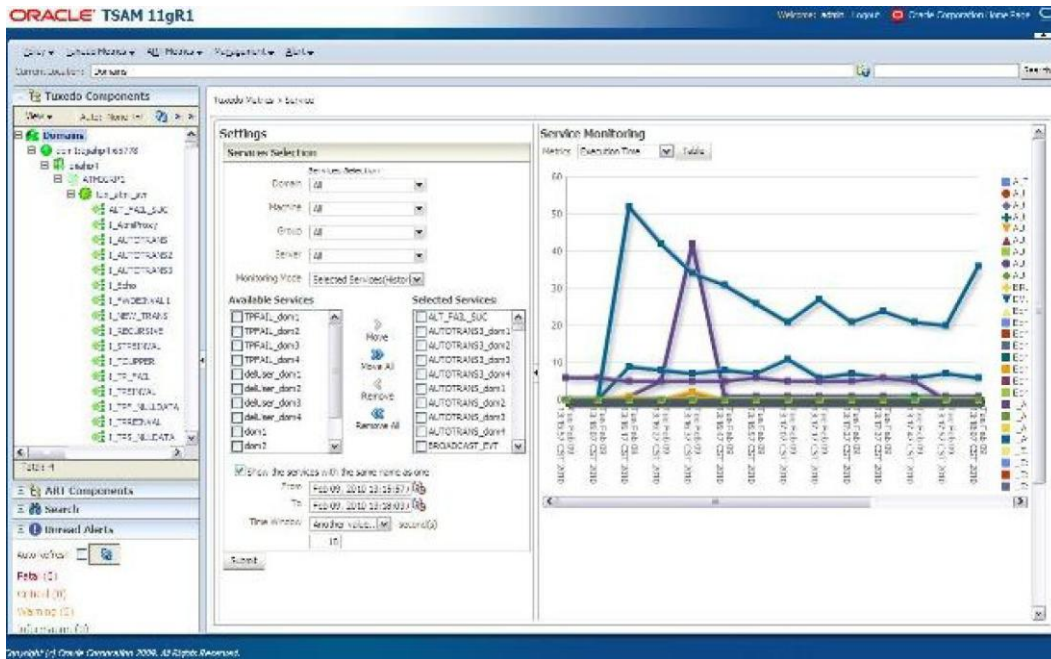
Tuxedoでは、追加のハードウェア・ノードが構成に加えられることで、ほぼリニアなスケラビリティが提供されます。Tuxedoドメインを使用することで、アプリケーションをさまざまなトポロジで配置して、ほぼリニアなスケラビリティを実現できます。Tuxedoドメインは、ノードのクラスタに似ており、アプリケーション・サービスの位置の透過性を提供します。サービスはドメイン内のどこにでも置くことができ、受信リクエストは自動的に使用可能なサーバーおよびサービスヘルディングされます。サービスの消費者は、リクエストがどこで処理されているかを意識する必要がありません。Tuxedoでは、このような決定を構成やワークロードに基づいて動的に行うことができます。

Tuxedoのドメイン配置トポロジにより、ビジネスの成長に応じてアプリケーションを拡張できるだけでなく、ドメインのノード間やドメイン間で同じサービスの複数のインスタンスが構成されている場合に、あらゆるシングル・ポイント障害を排除することもできます。ノードの1つが使用できなくなると、特定のサービスへのリクエストは、サービスの提供が可能なノードへ自動的にルーティングされます。多くのTuxedoアプリケーションはこれらの構成により99.999%以上の高可用性が提供されます。互いに通信し合う何千ものドメインを使用して配置されるTuxedoアプリケーションのインストールが多数あります。

このような構成では、ソフトウェアやハードウェアのアップグレードにおいても、計画停止時間を必要としません。ノードをドメイン内で一度に1つずつアップグレードしながら、残りのノードを通常どおり運用し続けることができるため、ローリング・アップグレードが可能になります。アプリケーションのユーザーは、中断のないサービスを受け続け、停止を体験することはありません。

監視ツール

ほとんどすべてのエンタープライズ・アプリケーションは、リアルタイムでアプリケーションを監視し、統計を収集するためのツールを必要とします。Oracle Tuxedo System and Application Monitor (Oracle TSAM) は、ある特定のサービス呼出しにかかった時間や、メッセージが発信元からビジネス・サービスに到達するまでにかかった時間などのデータとともに、包括的な監視情報をリアルタイムで提供します。Oracle TSAMは、特定のサービスのCPU使用率やサービスの応答時間などの統計情報も収集できます。このデータのすべてが、パフォーマンスのボトルネックの特定や、キャパシティ計画に加えて本番環境の問題の根本原因を判断する際に非常に役立つ可能性があります。Oracle TSAMに含まれる他の機能は次のとおりです。キューのメッセージにかかる時間、XAトランザクション監視、品質保証契約の実施。品質保証契約 (SLA) は、サービスの一部で設定でき、SLAに違反があるたびにアラートが生成されたり、自動的にアクションが実行されたりするように構成できます。SLAの一例に、ミッション・クリティカルなサービスの応答時間があります。サービスの応答時間が数ミリ秒を超えると、Oracle TSAMは発生時に勤務中のマネージャーに警告を出します。



Tuxedoには、Oracle TSAMに加えて、アプリケーションの動的な構成および管理のためのツールも含まれています。構成の変更を行うためにアプリケーションを停止する必要はありません。構成の変更は、MIB (Management Information Base) APIを使用して動的に行えます。Tuxedoは、これらのMIB APIにSNMPインタフェースも提供し、これによってHP Open ViewやBMC Patrolなどの一般的なツールを使用したTuxedoによる管理が可能になります。

標準の統合

今日のSOAや異種環境の時代においては、まったく異なるプラットフォーム上でホストされるサービスやアプリケーションにアクセスできることが、ほとんどのアプリケーションに必要とされています。JEEアプリケーションは、C/C++で記述されたバックエンド・サービスにアクセスできることが必要であり、その逆も同じです。同様に、Webアプリケーションは、メインフレーム・サービスなどにアクセスする必要があります。Tuxedoでは、他のプラットフォーム上で稼働するアプリケーションと統合するためのいくつかのオプションを提供します。Tuxedoには、J2EEとTuxedoアプリケーションとの間の双方向接続のためのJCA Adapterが含まれます。同様に、Tuxedo Mainframe Adapterは、IBMメインフレームとTuxedoアプリケーションとの間の双方向接続を提供します。Tuxedoには、ネイティブのWebサービス・スタックが含まれ、これによってHTTP経由のSOAPの使用によるあらゆるTuxedoサービスへのアクセスが可能になります。Tuxedoが提供する他の接続オプションには、.Net、MQSeriesおよびJava ClientからTuxedoサービスへのアクセスなどがあります。

たとえば、PHPサービスが、Javaベースのアプリケーション・サーバーでホストされるWebサービスを呼び出す必要がある場合は、起動する必要があるWebサービスのWSDLをインポートし、既存の構成ファイルの一部を編集するだけで済みます。コードを記述する要はありません。同様に、あらゆる標準Webサービス・クライアントがSOAP/HTTPを使用してPHPサービスにアクセスできるように、PHPサービスにWebサービスとしてアクセスする場合は、TuxedoのネイティブWebサービス・スタックを使用すればよく、コードを記述する必要はありません。

同様に、PHPサービスがメインフレームCICS環境で実行中のトランザクションを起動する必要がある場合は、Tuxedoのメインフレーム・アダプタを使用することで実行できます。実際、CICSトランザクションは、Tuxedo内で稼働するPHPサービスも起動できます。

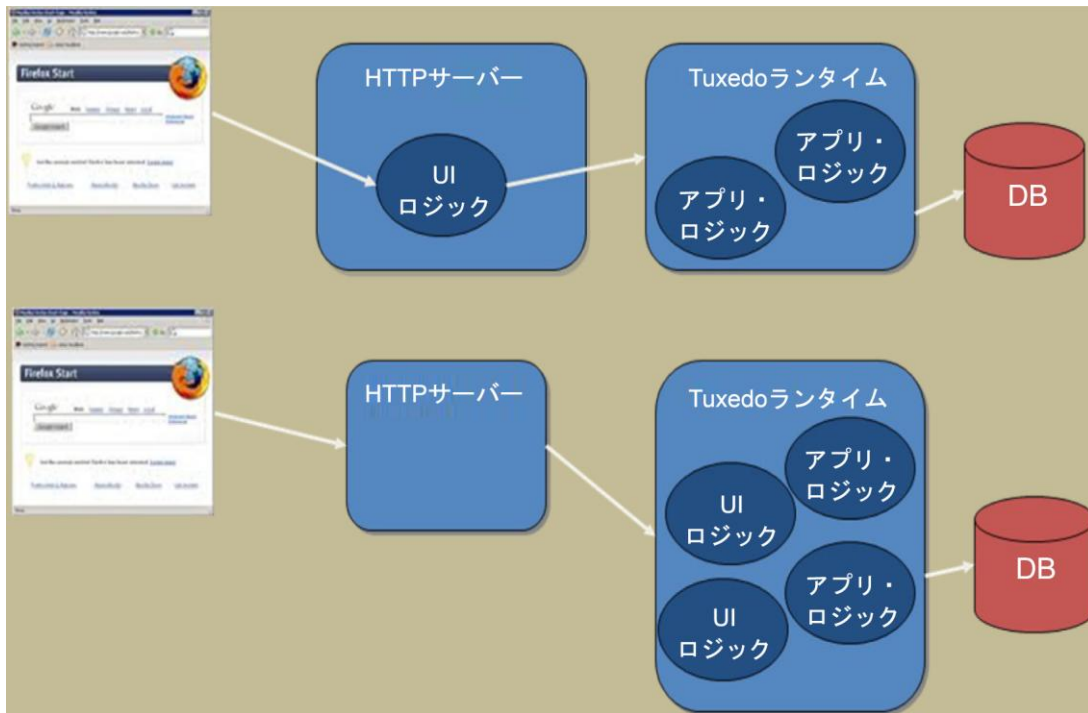
Tuxedoの使用場所

Oracle Tuxedoは、通信、金融サービス、物流、輸送、小売、公的機関などの多くの垂直業界で非常に広範囲で使用されています。これらの垂直業界のほとんどの組織では、Tuxedoインフラストラクチャを使用して、ミッション・クリティカルなアプリケーションを確実に実行しています。たとえば、米国最大の小売チェーンの1つは、7,000を超える店舗にTuxedoドメインをインストールしています。同様に、ある特定の時点の財務トランザクションの1つで、Tuxedoベースのアプリケーションが高い確率で使用されています。さらに、多数のISVアプリケーションにTuxedoが組み込まれています。たとえば、すべてのOracle PeopleSoftインストールはTuxedo上で稼働します。

PHP、Python、Rubyを使用したエンタープライズ・アプリケーションの開発

最初に説明したように、多数の状況依存型アプリケーションは動的言語で記述されています。これらの状況依存型アプリケーションのほとんどには、最終的に、エンタープライズ・アプリケーションのサービスの品質が求められます。動的言語は、迅速なアプリケーション開発環境を提供します。Tuxedoは、最適なランタイム・インフラストラクチャを使用可能にします。オラクルは、両者の最良の部分を組み合わせて、PHP、PythonおよびRubyアプリケーション向けのエンタープライズ・プラットフォームを作成しました。これらの動的言語によるアプリケーションは、Java、C/C++、COBOLで記述されたこれまでのアプリケーションと同レベルのサービス品質を提供します。基本的に、Tuxedoは、PHP、PythonおよびRubyで記述されたビジネス・ロジックをホストするコンテナを提供し、C/C++またはJavaアプリケーションと同様に拡張できます。アプリケーション・プラットフォームは、開発環境に中立です。開発者は、IDEまたはフレームワークを自分で選択して使用できます。このプラットフォームを使用すれば、エンタープライズ・アプリケーションに必要なとされるサービス品質を伴う新しいWebアプリケーションやバックエンド・サービスを開発できます。

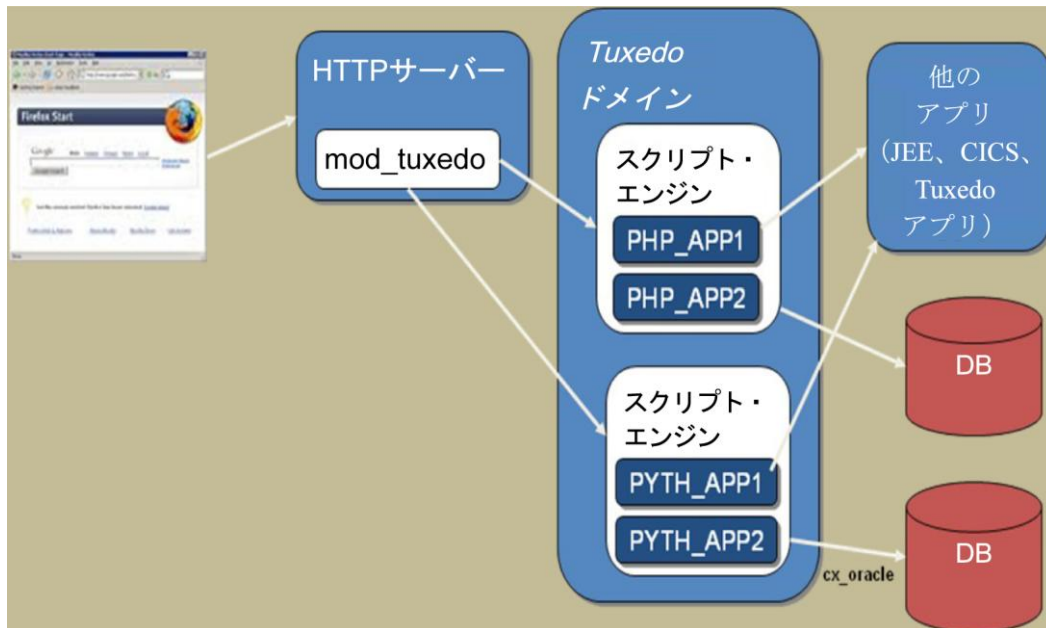
Tuxedoを使用すると、ビジネス・ロジックおよびUIレイヤーを一緒にまたは別々に配置できます。下の図に示すように、UIレイヤーはHTTPサーバーにとどまるか、またはTuxedo内のアプリケーション・ロジックと一緒に配置できます。Tuxedoを使用すると、配置にはほとんど無限の柔軟性があります。アプリケーション・サービスのインスタンスを数個だけ、または何百と配置できます。または、ビジネス要件に応じて、インスタンスの数をリアルタイムで増加させることができます。PHP、PythonおよびRubyで記述されたサービスのインスタンスはすべて、クラスタ環境でシングル・ポイント障害を排除しながら機能します。同じことが、HTMLコードを生成するサービスまたはUIロジックに当てはまります。UIロジックは、アプリケーション・サービスになることができ、Tuxedoランタイム・インフラストラクチャの利点をすべて得られます。



配置におけるこの柔軟性により、PHP、PythonおよびRubyアプリケーションに可能性が広がり、バックエンド・サービスのビジネス・ロジックの記述、または新しいWebアプリケーションの開発に使用できるようになりました。また、これらのエンタープライズ・ランタイム環境のすべての利点は、個人の選択による開発環境でアプリケーションを開発する柔軟性を失うことなく、得ることができます。

ユースケース1：新しいWebアプリケーションの開発

組織は、前に説明したように、PHPまたはPythonによるアプリケーション、つまり状況依存型アプリケーションを使用していることがあります。状況依存型アプリケーションは急速に発展してきましたが、問題はありません。これらの既存のアプリケーションは、Tuxedoのスキプティング・エンジンで、コードを変更することなく配置できます。動的言語で記述された既存のアプリケーションをTuxedoインフラストラクチャ内に配置し実行するために、コードを変更する必要はありません。アプリケーションが、Rails、Django、Symfonyなどのフレームワークの1つを使用して記述されていた場合でも関係ありません。アプリケーションは、データベースを直接呼び出すか、一部の別のサービスを呼び出します。これも問題ありません。このようなアプリケーションはすべて、Tuxedoのスキプティング・エンジンでホストできます。実際に、いくつかの一般的なPHPオープンソース・アプリケーションは、アプリケーション・コードを変更することなく、Tuxedoに問題なく配置されています。その1つは、すべてPHPで記述されたmediawikiです。当然、このアプリケーションがTuxedoランタイムにあれば、すぐにTuxedoの利点がすべて得られます。これで、このアプリケーションは、ビジネスの成長に伴ってリニア的に拡張できるようになり、可用性が向上します。また、このアプリケーションは、Tuxedoまたは他のアプリケーション・サーバーのいずれかで稼働する他のアプリケーションと自動的に統合できるようになります。



HTTPリクエストは、Apache HTTPサーバーのmod_tuxedoに渡されます。それから、mod_tuxedoはリクエストをTuxedoのスクリプティング・エンジンへ渡し、それによってアプリケーションが起動されます。Tuxedoのスクリプティング・エンジンは、ランタイム環境における相違点の詳細があれば、それを隠す処理を行います。

Mod_tuxedoとスクリプト・エンジンは、ソリューションの2つのコンポーネントです。これらを1つ1つ見てみましょう。

Mod_tuxedo :

Mod_tuxedoは、Oracle HTTP Server、Apache HTTPサーバーおよびOracle iPlanet Web Serverのためのアドオン・モジュールです。Mod_tuxedoは、PHP、PythonおよびRubyで記述されたアプリケーションに送信するためのHTTPペイロードを準備し、変換します。この軽量ソリューションにより、HTTP呼出しを傍受してTuxedoサービス呼出しへ変換します。Mod_tuxedoは、Apacheの他のモジュールと同じ方法で構成されます。すなわち、httpd.confを使用することにより、Mod_tuxedoはprefork、workerなどのApache処理モジュールもサポートできます。

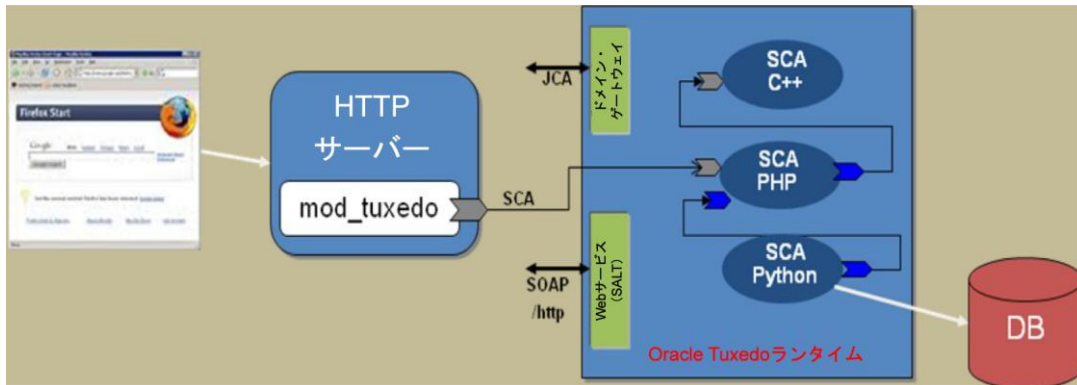
スクリプト・エンジン :

Tuxedoランタイムは、多数のシステム・サーバーで構成されます。システム・サーバーはそれぞれ、ライフ・サイクル管理、リクエスト・ルーティング、ロードバランシング、トランザクション管理など、さまざまなランタイム・サービスを提供します。スクリプト・エンジンは、FastCGIに基づいたアプリケーション、およびPHP、Python、Rubyで記述されたアプリケーションをホストするためのシステム・サーバーです。スクリプト・エンジンは、1つ以上のアプリケーションをホストでき、起動および停止、アプリケーションの動的リロード、アプリケーション・キャッシュなどのライフ・サイクル管理サービスを提供できます。スクリプト・エンジンは、Tuxedoのランタイム・インフラストラクチャのすべての利点を受け継いでいます。スクリプト・エンジンは、1つ以上のインスタンスを同じノード上に置くように構成するか、またはスケーラビリティおよび高可用性を実現するために複数のノード間に分散させて構成することができます。さらに、PHP、PythonおよびRubyアプリケーションは、Tuxedoツールを通じて監視し、さまざまな統計および包括的なパフォーマンス・デー

タを収集できます。

ユースケース2 : ビジネス・サービスの開発

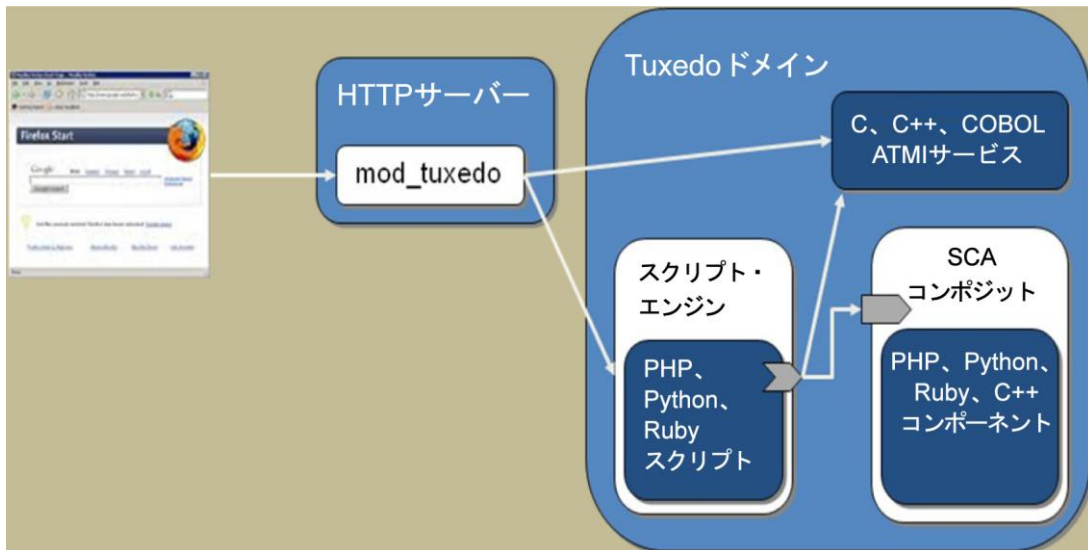
組織では、PHPを使用して新しいビジネス・サービスの開発を計画する場合があります。このようなサービスでは、たとえば、Webストアでのアイテムの販売記録、クレジット・カードの検証などの特定のビジネス問題を解決できるでしょう。また、ビジネス・サービスは、バックエンドでデータベースにアクセスし、引き換えにHTMLコードを生成する場合もあるでしょう。Tuxedoの配置のために、このようなビジネス・サービスをどのように開発するのでしょうか。Tuxedoは、Service Component Architecture (SCA) 標準に基づいた、簡単に使用できるプログラミング・モデルを提供します。PHPで新しいサービスを開発する手法は、C++向けのものと同じであり、メタデータ駆動型です。開発者は、ビジネス・ロジックに重点を置き、Tuxedoを意識することはありません。つまり、開発者が習得して使用する必要があるようなTuxedo特有のAPIはありません。ビジネス・ロジックは、ほとんどスタンドアロン・プログラムと同じように見えます。



いったんビジネス・サービスが配置されると、Tuxedoが提供する多数の方法の1つを使用してアクセスできます。これには、HTTPサーバーからのアクセスやWebサービスとしてのアクセスが含まれます。PHPで記述されたこのサービスは、C/C++やCOBOLなどの他の言語で記述されたサービスからアクセスできるようになります。開発者は、統合の詳細を認識する必要はありません。インフラストラクチャの詳細はすべて、Tuxedoによって処理されます。

ユースケース3 : 既存のTuxedoサービスの表示

組織では、良好に機能しているTuxedoビジネス・サービスを大規模なアプリケーションの一部として使用している可能性があります。これらのサービスは、おそらくC/C++またはCOBOLで開発されています。ビジネス上の理由により、これらのサービスはWebブラウザからアクセスする必要が出てきました。では、どのようにしてブラウザベースのHTMLアプリケーションをTuxedoに接続できるのでしょうか。アプリケーションの多くは、Tuxedoの前にJSP/Servletエンジンを使用してきました。現在は、Webアプリケーションは、PHP、PythonまたはRubyで作成でき、バックエンドのTuxedoサービスに通信し、HTMLコードを生成します。既存のTuxedoアプリケーションを変更する必要はありません。前のユースケースで説明したように、HTTPリクエストはmod_tuxedoによって傍受されてからスクリプト・エンジンへ転送されます。スクリプト・エンジンは、Webアプリケーションを実行し、既存のTuxedoサービスへの呼出しを作成します。また、バックエンド・サービスと同じコンテナにあるUIをホストすることにより、ITオペレーターおよび管理者が管理および監視するアプリケーション・サーバーが1つ少なくなります。



結論

Oracle Tuxedoは、PHP、PythonおよびRubyエンタープライズ・アプリケーションのための業界屈指の唯一のプラットフォームです。Tuxedoを使用すると、PHP、PythonおよびRubyアプリケーションは、JavaやC/C++で記述されたアプリケーションと同じレベルのサービス品質を提供できます。Tuxedoの実証済みの信頼性、可用性、スケーラビリティおよびパフォーマンスにより、開発者はビジネス・ロジックに重点を置いてランタイムの詳細に煩わされることなく、アプリケーションの開発および配置が簡単になります。さらに、これらのアプリケーションはサイロで運用する必要がなくなり、俊敏性と、動的言語に関連した他のすべての利点を維持しながら、他のエンタープライズ・アプリケーションと同様にクリティカルなビジネス・ニーズを提供できます。

ORACLE®

Oracle Tuxedo : 動的言語のための
エンタープライズ・プラットフォーム
2011年5月
著者 : Deepak Goel

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

海外からのお問い合わせ窓口 :
電話 : +1.650.506.7000
ファクシミリ : +1.650.506.7200
oracle.com



Oracle is committed to developing practices and products that help protect the environment

Copyright © 2011, Oracle and/or its affiliates. All rights reserved. 本文書は情報提供のみを目的として提供されており、ここに記載される内容は予告なく変更されることがあります。本文書は一切間違いがないことを保証するものではなく、さらに、口述による明示または法律による黙示を問わず、特定の目的に対する商品性もしくは適合性についての黙示的な保証を含み、いかなる他の保証や条件も提供するものではありません。オラクル社は本文書に関するいかなる法的責任も明確に否認し、本文書によって直接的または間接的に確立される契約義務はないものとします。本文書はオラクル社の書面による許可を前もって得ることなく、いかなる目的のためにも、電子または印刷を含むいかなる形式や手段によっても再作成または送信することはできません。

Oracleは米国Oracle Corporationおよびその子会社、関連会社の登録商標です。その他の名称はそれぞれの会社の商標です。

Hardware and Software, Engineered to Work Together