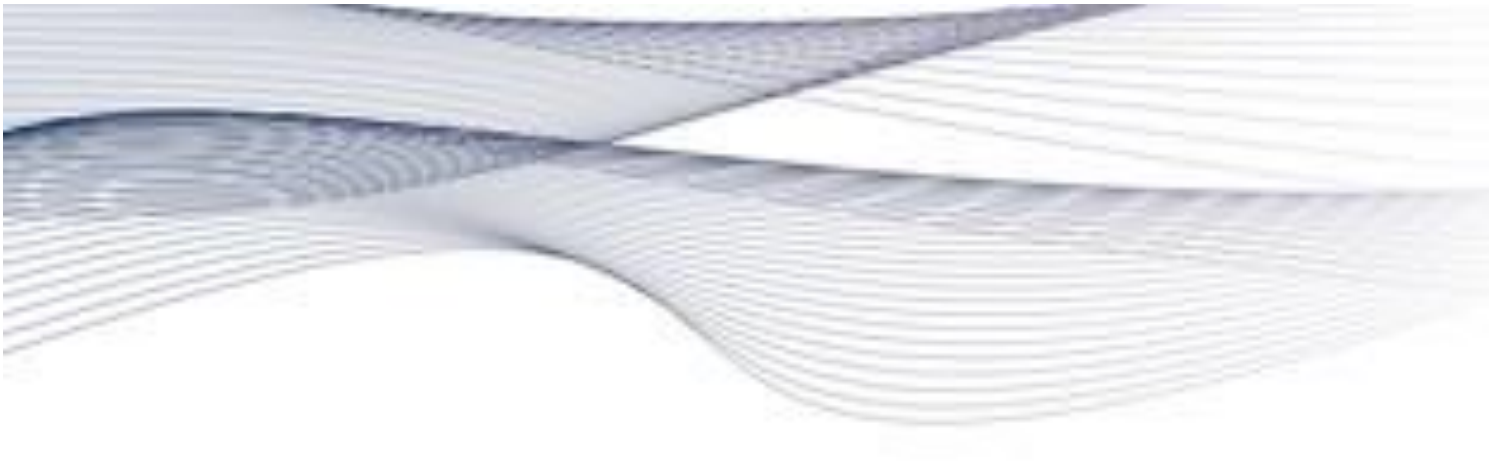

Aristaホワイトペーパー

EOS：次世代の拡張可能なオペレーティング・システム

今では、ネットワーク全体のパフォーマンス、耐障害性、プログラマビリティが次世代のクラウドおよびエンタープライズ・データセンター・ネットワークの基本的なビジネス要件になっています。プロビジョニングやネットワーク運用に関する俊敏性や展開のニーズでは、新しいレベルの自動化と、現在のデータセンター・インフラストラクチャとの統合が要求されます。このネットワーク・オペレーティング・システムの根底にある設計アーキテクチャは、こうした要件を満たすための基盤を提供します。

アリスタネットワークスは、こうした要件に対処するために、業界の先端を行くLinuxベースのネットワーク・オペレーティング・システムであるEOS（Extensible Operating System）を設計し、提供しています。EOSは、堅牢でプログラム可能かつ革新的なオペレーティング・システムであり、受賞歴があるAristaネットワーク・スイッチのポートフォリオ全体および仮想マシンのインスタンス（vEOS）で単一のソフトウェア・イメージが実行されることを特長としています。これにより、一貫性のある運用、ワークフロー自動化、および高可用性が保証されるとともに、データセンターの運用コストが大幅に削減されます。

このホワイトペーパーでは、最新のネットワーク・オペレーティング・システムを解説し、アリスタネットワークスのEOSアーキテクチャがどのようにして次世代のデータセンターを独特な形でサポートしているかを説明します。

ARISTA

旧来のネットワーク・オペレーティング・システムの制限事項

これまでのエンタープライズ・データセンター・ネットワークは長い間、ソフトウェアのクラッシュや計画外の停止の影響を受けやすい存在でした。複数のソフトウェア・リリースがあってスイッチ・プラットフォーム間でその種類が異なる場合、新しい機能やサービスの展開は長く時間を浪費する処理になっていました。

また、誤りの起こりやすい手作業での設定とも相まって、ネットワークのアップタイムに悪影響が及ぶことは避けられないことでした。

こうした問題の中心にあるのは、依然として現在のネットワーク機器のほとんどで実行されている古いモノリシック・ソフトウェアです。これにより、ネットワーク信頼性が制限されることになります。現実的には、こうした共有メモリ・システムのどこかに1つでもバグや欠陥があると、ソフトウェア不具合を封じ込める仕組みがないので、ネットワーク全体が中断の危険にさらされます。

同時に、複数のタスク間の分離が行われていない場合、これらのオペレーティング・システムをパフォーマンスの観点から、または新しい機能を高い信頼性で、スケーリングすることは困難です。こうしたレガシ・ソフトウェアの危うい特性により、ネットワーク・オペレーティング・システムを拡張して、お客様に特有の管理プロセスや、最近のデータセンター環境に展開されているその他のシステムおよびサービスとの統合など、新しい機能を実装することはできません。

以前のネットワーク・オペレーティング・システムは、カスタマイズされたカーネルの上に積み上げ方式で構築されていました。また、状態情報は、システム全体に恣意的に分散していたり、さらに悪い場合はカーネルそのものの内部に埋め込まれたりしていました。これらの環境では通常、すべてのプロセスが共通のアドレス空間で実行され、その結果、危険な連鎖障害（1つの欠陥がシステム全体をクラッシュさせ、深刻なネットワーク障害を引き起こす可能性がある状況）が発生します。ソフトウェアの状態は、同期または非同期のポーリングを使用して管理されます。通常、ポーリングは数ミリ秒または数秒おきにすべてのインターフェースおよび内部データ構造の状態をチェックする非効率的な仕組みであり、結果的にサイクルを浪費し、デッドロックや複数の並列イベントとの競合状態を引き起こす可能性があります。

何らかの状態の変化は、注意深く作成されたコードパスによって処理されます。その結果、ソフトウェア・プロセスは、基本となるイベントに応じて、厳密な順序付けに従って反応します。考えられるすべてのコードパスをそのようなアーキテクチャでテストすると、展開は複雑で時間のかかるものになり、お客様によるテストと認定サイクルの期間が延びます。

また、ソフトウェアの機能は、プロセス間で発生するマトリクス化されたメッセージングによって階層化されています。そのため、プロセス間通信は錯綜し、システムが複雑になります（*簡略化された*以下の図[1]を参照）。

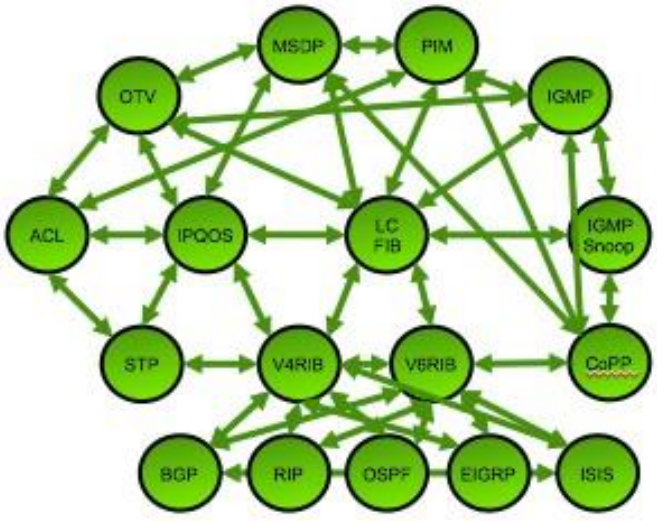

図1：旧来のネットワーク・オペレーティング・システムの通信（2000～2010年頃）

さらに、内部の状態はユーザーやアプリケーションには公開できず、これらのスタックでは高水準のAPIラッパーによってのみプログラミングを行うことができます。こうしたAPIは、SNMPまたはCLIと同じ情報をモニタリングのためにテキストまたはXML形式で提供できます。

クラウド・ネットワークやソフトウェア定義のデータセンターの構築は、このような最適とはいえないソフトウェア・アーキテクチャでは不可能です。

Arista EOS：現代のネットワークOS

パブリック/プライベート/ハイブリッド・クラウド・ネットワークのニーズに対処するとともに、新しいアプリケーションを実現してSoftware Defined Network（SDN）時代の先駆けとなるために、アリスタネットワークスのExtensible Operating System（EOS）は基礎から設計され、こうした要求の厳しい新たなデータセンター環境向けに最適化されています。

アリスタネットワークスのExtensible Operating Systemは、業界で最先端のネットワーク・オペレーティング・システムです。現代的なソフトウェアとオペレーティング・システム（OS）アーキテクチャ、透過的に再起動可能なプロセス、オープンなプラットフォーム開発、改変なしのLinuxカーネル、ステートフルなパブリッシュ/サブスクライブ・データベース・モデルを組み合わせたものになっています。

真のオープン・オペレーティング・システムの利点を十分に理解し、総合的な自動化ソリューションの可能性を探るためには、EOSの基盤となるアーキテクチャを詳しく見直す必要があります。以降のセクションでは、ESOアーキテクチャの主な構成要素および属性、またEOS上に構築されている充実したネットワーク・サービス群について説明します。

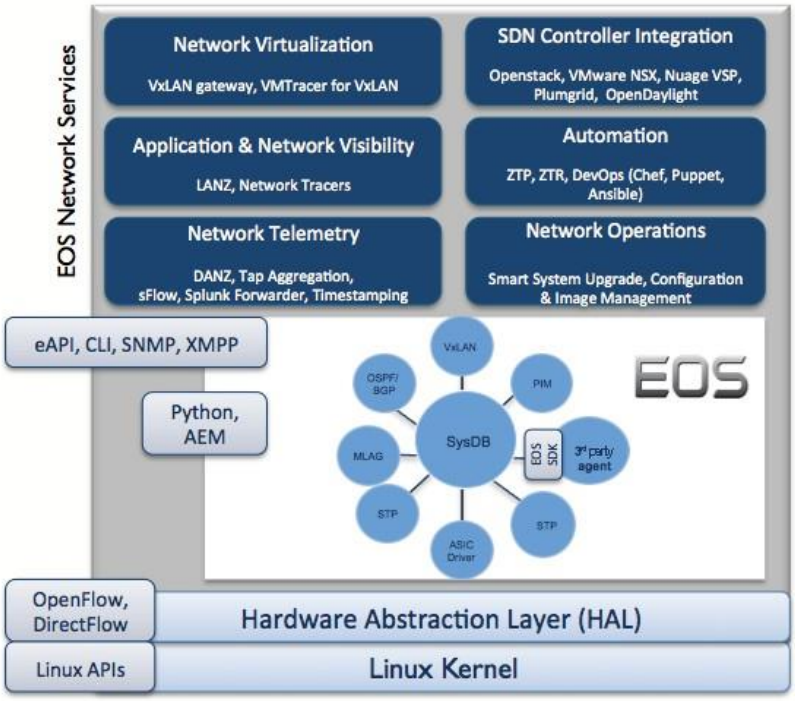

図2：Arista EOSのアーキテクチャ

Arista EOS：Linuxカーネルの使用

Arista EOSの基盤になっているのは、改変されていないLinuxカーネルです。Linuxカーネルを使用することの主な利点は、次のとおりです。

- 広くパブリックに公開されている
- 何百もの貢献者と何百万というユーザーが安定性、処理能力、機能セットを支えている
- きわめて安定していて本質的にモジュール形式になっている
- パッケージ、アプリケーション・リポジトリを利用でき、ソース・コードを入手可能

- Linuxは広い範囲で使用されているため、多くの種類のシステムや環境の間できわめて広範に受け入れられている

Arista EOS：SysDB

システムの機能的な制御を複数のプロセスに分割すると、耐障害性と障害分離性が大幅に強化されますが、そのためにはシステム内の動作を調整するための仕組みが必要になります。その役割を果たすのがSysDBです。

図2に示すように、EOSの中心部にはシステム・データベースが存在します。SysDBは（実行時に機器によって生成される）インメモリ・データベースです。インメモリ・データベースは、ユーザー空間で実行され、システムの全体的な状態を格納します。従来のデータベースと同様、SysDBはアプリケーション・ロジックを一切含んでおらず、状態の保存のみに関与します。トランザクション用には最適化されておらず、変化があった場合に関係のあるプロセス（「エージェント」とも呼ばれます）に通知することにより、プロセス（エージェント）間で状態の同期をとることを目的としています。

システム内のすべてのエージェントは、SysDBにある各自の設定および状態をマウントします。これは、マウント・ポイントごとに読み取り専用または読み取り–書き込みのアクセス権が指定されているファイルシステムのマウントにとてもよく似ています。SysDBからのマウントを行う際、エージェントはそのマウント・ポイントにおけるすべての状態のローカル・コピーを自分用に受け取ります。SysDBはRAM内に保持されているため、スイッチが電源オフになったり再起動されたりすると、情報が失われます。

SysDBは、イベント駆動型のパブリッシュ/サブスクライブ・モデルです。　エージェントの状態が変化した場合、SysDBはそのエージェントにイベント通知を送信し、そのエージェントは自らのローカル・コピーを更新します。同様に、マウント・ポイントへの書き込みを行う際には、自らのローカル・コピーのみを変更すれば、書き込み内容がすぐに返されます。

それぞれのEOSエージェントは、SysDB内でほかのエージェントの状態が変化したときに通知を受け取るために、SysDBをサブスクライブします。エージェントの状態が変化した場合、その変更通知はバッファに格納され、非同期でSysDBに送信されます。その後、SysDBは、変化があったエージェントをサブスクライブしているほかのすべてのエージェントに通知を行います。

こうして、一元的なデータベースを利用してシステム全域で状態情報の受け渡しを行う方法や、SysDBのコードの生成を自動化する方法により、リスクや誤りが減少し、ソフトウェア機能の更新サイクルが短縮されます。また、お客様はAPIを使用して、SysDBからの通知の受信と、スイッチ機能のカスタマイズや拡張を柔軟に行うことができます。

Arista EOS：主なメリット

EOSは、セキュリティ、安定性、オープン性、モジュール性、拡張性に関するLinuxの資産を継承しつつ、きわめて堅牢で信頼性の高いデータセンター通信サービスを実現します。こうした組み合わせは、次世代データセンター・ネットワークの機能性と進化を大いに促進する機会を提供します。

- Arista EOSは、状態情報をプロセスそのものから分離する独自のマルチプロセス状態共有アーキテクチャです。これは、アリスタネットワークスの中核となるソフトウェア設計思想を反映したものであり、システムの実行状態に影響を与えることなく、障害回復やリアルタイムのソフトウェア・アップデートをプロセス・レベルで可能にします。

- また、プロトコル処理、セキュリティ機能、管理サービス、さらにはデバイス・ドライバーも、カーネル内ではなく、ユーザー・アドレス空間で実行されます。これにより、全体としての安定性が大きく向上します。また、元のLinux環境を改変しないという設計方針を維持することで、アリスタネットワークスは自社のオペレーティング・システムを追加機能によって容易に拡張することができます。

- EOSの同一バイナリ・イメージを製品のどのファミリーにも展開できます。これにより、各プラットフォームでのテスト効率が高まり、機能やバグ解決の互換性をすべてのプラットフォーム間で維持できます。また、ユーザーが各自のデータセンター環境で新しいリリースの展開、認定、および検証を行うのもずっと容易になります。

- 特に重要な点は、Linuxシェルへの制限のないアクセスを許可することで、ユーザーが真のデータセンター自動化機能を利用できるようになったことです。ユーザーは、IT組織のほかの部署と同じようなやり方で、シェル・スクリプトまたはPythonスクリプトを書いてネットワークを効率的に運用することができます。

- セキュリティに関して、EOSはTACACS+およびRADIUS AAA機能のような認証メカニズムをサポートしています。これにより、どんな企業でも、任意のスイッチを許可されたアクセスや安全なアクセスのみに制限することができます。

- EOSは、拡張性というコアとなる考え方を実現する開発フレームワークを提供します。オープンな基盤と、ソフトウェア開発モデルにより、機能を俊敏に実現し、稼働率を高め、メンテナンスを容易にし、幅広いツールやオプションの選択肢を提供します。

Arista EOS：耐障害性

ここでは、EOSのマルチプロセス状態共有アーキテクチャによって可用性の向上、メンテナンス期間の短縮、管理容易性やセキュリティの向上がどのように実現されるかを詳しく説明します。

システムの可用性を高める鍵は、障害隔離とソフトウェアの自己回復性にあります。ほとんどの組み込みプラットフォームでは、ソフトウェアにフォールト（欠陥）があるとリロードが生じ、何秒間か、ときには何分間ものダウンタイムが発生します。EOSでは、フォールトがあっても、その発生源であるエージェントまたはドライバー内に封じ込められます。フォールトによってエージェントがクラッシュした場合は、EOSプロセス・マネージャー（ProcMgr）がそのエージェントをただちに再起動します。フォールトによってエージェントが終了またはループ状態に陥った場合は、ProcMgrがその状況を検出し、エージェントを再起動します。そのため、EOS内のフォールトについては、自己回復が可能です。このプロセスを以下の図[4]に示します。

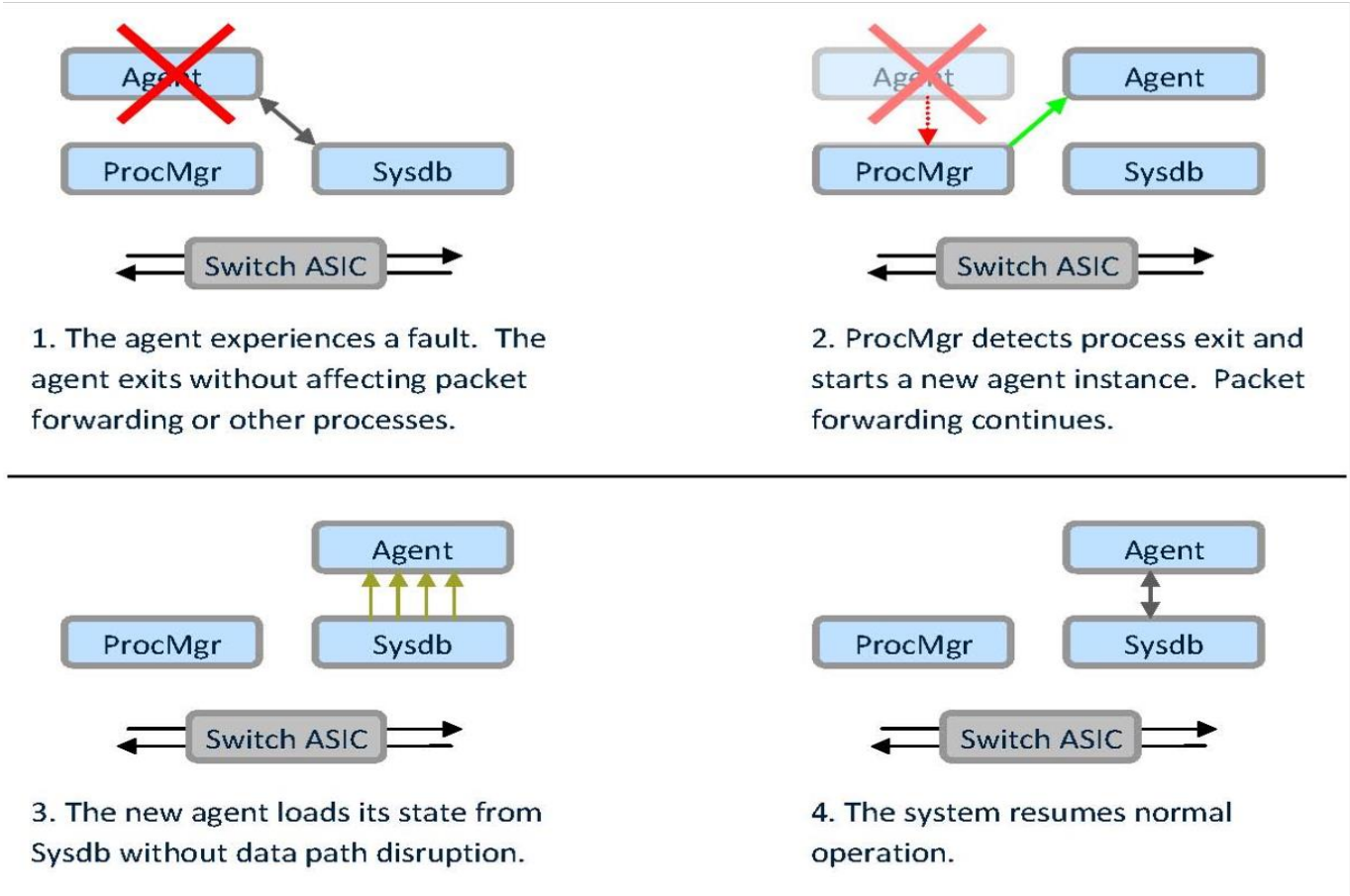

図3:EOSによる障害隔離と自己回復

また、EOSのマルチプロセス状態共有アーキテクチャは、メンテナンス期間の短縮にとっても重要です。これは、スイッチの通常稼働中により多くのメンテナンス作業を実行できるようにすることで実現されます。どのEOSエージェントも、実行中に修正したり、スイッチの動作を中断することなく再起動したりすることができます。データ・フローは、モジュールのアップグレード処理の影響を受けません。そのため、ユーザーが認識できるダウンタイムはありません。

運用を改善する鍵は、サードパーティによる管理およびオーケストレーション統合やタスク自動化のソフトウェアをサポートする機能にあります。EOSは、サブシステムおよびエージェント向けの堅牢で保護された環境を提供しているため、検証済みのサードパーティ・エージェントも安全に実行できます。また、スイッチの動作を調整して管理を最適化したり、特定のお客様の環境内で一般的なタスクを自動化したりできます。そのため、Arista EOSをご利用のお客様は、サードパーティ・ソフトウェアの問題がこのネットワーク・オペレーティング・システムによって極小化されることを知ったうえで安心して展開を行うことができます。

セキュリティ向上のために重要なのは、セキュリティ脆弱性の影響を脆弱なエージェント内に封じ込めることです。たとえば、SNMPサブシステムに脆弱性がある場合は、SNMPがアクセス可能なすべての状態がエクスプロイトによって読み取られるおそれがあります。しかし、そのエクスプロイトは追加のユーザー・アカウントを作成したり、インターフェイスの再設定や外部ソフトウェアの実行を行ったりすることはできません。つまり、EOSアーキテクチャではフォールトが1つのモジュールに隔離されるため、セキュリティ脆弱性の影響も封じ込められます。さらに、管理およびオーケストレーションのサポートを強化する同じ拡張性のメカニズムにより、サードパーティ・ソフトウェアでさらなるセキュリティ強化のためにカスタム・セキュリティ・ポリシーや侵入検知を実装できます。

Arista EOS：プログラマビリティと拡張性

EOSは、Linuxカーネル、ハードウェア・フォワーディング・テーブル、スイッチの設定およびCLI、スイッチのコントロール・プレーン、管理レイヤといったすべてのレイヤにわたってプログラム可能です。EOSのこうしたプログラマビリティにより、仮想化、管理、自動化、オーケストレーション、ネットワーク・サービス用の幅広いサードパーティ・アプリケーションとの迅速な統合を行うことができます。Arista EOSは、以下のように豊富な一連のプログラム可能なインターフェイスを備えています。

- Linuxシェルのアクセスおよび API
- OpenFlow、DirectFlow
- SysDB API
- Python、Perlによるスクリプト記述、高度イベント管理（AEM）
- EOS SDK
- JSONベースのeAPI、CLI、SNMP、XMPP

Arista EOSでは、Linuxシェルへの完全なアクセス権がrootレベルの管理者向けに用意されているほか、Linuxベースの多種多様なツールを利用できます。OpenFlowおよびDirectFlowにより、お客様は、アプリケーションのニーズに基づいてパケット・フォワーディングを微調整するために、スイッチのフォワーディング状態をプログラムすることが可能です。SysDB APIは、低レベルのカウンタ、温度測定、電力供給状態をはじめ、システムのネイティブ・レベルでのモニタリングおよび管理に必要なすべてのパラメーターなど、すべての内部状態へのアクセスを可能にします。JSONベースのEOS API（eAPI）は、コンピューティングおよびストレージ・リソースやオーケストレーション・システムの管理によく使用されるツールとの容易な統合を実現します。Pythonで記述されたCLIでさえ、カスタマイズできます。Python、Perlなどに基づくスクリプトも、仮想および物理アプリケーション、SDNタイプのコントローラ・プラットフォームおよびレイヤ4～7のサービスとのサードパーティまたはネイティブに開発できます。

お客様は、EOSのソフトウェア開発キット（SDK）を使用して、カスタマイズされた独自のEOSアプリケーションをC++やPythonで開発できます。こうしたEOSの開発モデルにより、これらのサードパーティ・アプリケーションをほかのEOSエージェントとともに、EOSの重要な要素として利用できるようになります。SDKを使用すると、サードパーティ・エージェントがスイッチの状態にアクセスしてネットワーク・イベントに反応できるように、Arista EOSで使用可能なソフトウェア抽象化へのプログラミング言語のバインディングを実現できます。これらのアプリケーションは、たとえば、インターフェイス、IPおよびMPLSルート、アクセス制御リスト（ACL）を管理したり、スイッチとモニタリングおよびネットワーク・コントローラとの間での通信を行うために一連のAPIを使用したりできます。このSDKでは、イベント駆動型の通知を必要とする長期的に実行されるプロセスと、ほかのEOSエージェントとのパフォーマンスの高いやりとりを必要とするスクリプトの両方を対象としています。SysDBによる状態分離と、モジュール型アーキテクチャによって実現される固有の障害分離のおかげで、お客様は、システム全体の中断についての不安を感じることなく、独自のアプリケーションを開発してインストールすることができます。

お客様は、こうした奥深いプログラマビリティを利用して、カスタマイズされたアプリケーションの開発、エコシステム・ツールとの統合、さらには、迅速なターンアラウンド・タイムによる機能間ギャップへの対処を行うことができます。なお、EOSはこのターンアラウンドタイムの短さにより、現代の俊敏なクラウド・データセンターに適したネットワークOSになっています。

Arista EOS：ネットワーク・サービス

アリスタネットワークスは、業界をリードするEOS基盤の上に、ネットワーク・サービスの総合スイートを追加しています。これには、充実したレイヤ2/レイヤ3機能、ネットワーク仮想化のサポート、ネットワーク可視性およびテレメトリ、自動化、SDNコントローラ統合が含まれます。これらすべてのサービスは、EOSが提供する優れた拡張性を利用しています。こうした拡張性は、お客様の問題への対処時にサードパーティ・ツールと統合したりネットワーク全体にわたる実用的なアプローチを採用したりするためのものです。たとえば、ZTPは多数のスイッチのプロビジョニングの自動化やテンプレート化に役立ち、スマート・システム・アップグレード（SSU）は中断を伴わないソフトウェア・アップグレードの実行に役立ちます。また、CloudVisionはOVSDB、eAPI、またはOpenFlowを使用した、SDNコントローラとのネットワーク全体にわたる統合を可能にします。

Arista EOS：スマート・システム・アップグレード

ネットワーク・サービスの1つの例として、スマート・システム・アップグレード（SSU）が挙げられます。SSUは、ネットワークでの円滑なソフトウェア・アップグレードを実現します。新しいネットワーク・サービスや修正プログラムの導入のために中断を発生させることなくシステム・ソフトウェア・アップグレードを実行できることは、データセンター環境で要求される可用性レベルを維持するうえで非常に重要です。

これまでのネットワーク・オペレーティング・システムは通常、旧来のインサービス・ソフトウェア・アップグレード（ISSU）モデルに従っていました。このモデルでは、稼働中の単一プラットフォームが、おそらくいくつかの新機能や修正が含まれる、よりバージョン番号の大きなコードへのカットオーバーを行える程度でした。従来のISSUによる方法は、大がかりで複雑なソフトウェア開発が必要になるため、長い間大きな負担になっていました。特殊なISSUコードを可能性のある抑制と均衡をすべて考慮して記述する必要があり、システムでは異なる2つのバージョンのソフトウェアの切り替え時にハードウェアとソフトウェアのすべての状態の保持しようとしなければなりませんでした。この問題は、機能セットが増え、機能間の相互作用が生じて、関連する状態情報が増えるにつれ、大きくなっています。こうした複雑さの副作用は、おさまりのよいケースがありふれていて、すべてのシナリオに対処できないことです。テスト・サイクルを追加しても、「ISSU対応」の日数が追加されてリリースのスケジュールが延びるだけで、すべてのケースを押さえられる保証はありません。

オープンでプログラム可能なEOSの優れたアーキテクチャと、ほかのアプリケーションおよびインフラストラクチャ・コンポーネントとの直接統合を利用しているSSUでは、より全体的なネットワークの観点でソフトウェアのメンテナンスをとらえており、ネットワークの透過的な除去または追加が可能になっています。その際、トラフィックへの影響を完全になくすこともできます。アリスタネットワークスのSSUは、データセンター・インフラストラクチャ向けの完成されたソリューションとして設計されており、主に次の利点を提供します。

- スパインまたはリーフの役割にカスタマイズされた、ネットワーク要素のインテリジェントな挿入および除去
- システムの機能停止を伴わない、新しいソフトウェア・リリースへのプログラム可能なアップグレード
- すべてのアプリケーションおよびインフラストラクチャ要素とのオープンな統合
- 簡素化されたソリューション：ほかのアプローチでは必要となる骨の折れる状態管理や状態変換の処理の複雑さを意図的に回避できます。

そのため、アリスタネットワークスのプロビジョニング・モデルでは、ゼロ・タッチ・プロビジョニング（ZTP）またはゼロ・タッチ・リプレイスメント（ZTR）によるスイッチの初期展開や入れ換えが可能であり、その後もSSUフレームワークにより、リーフまたはスパイン・レイヤのどちらでもシームレスなアップグレードが適宜サポートされることで持続的な運用が保証されます。

Arista vEOS：仮想マシンとしてのEOS

Arista vEOSは、物理スイッチ上で動作するEOSソフトウェア・プラットフォームを仮想マシンベースのオファリングへと拡張したものです。vEOSには、本書で説明した、EOSアーキテクチャのすべての属性と利点が引き継がれています。vEOSは実際のEOSイメージであり、シミュレータではありません。

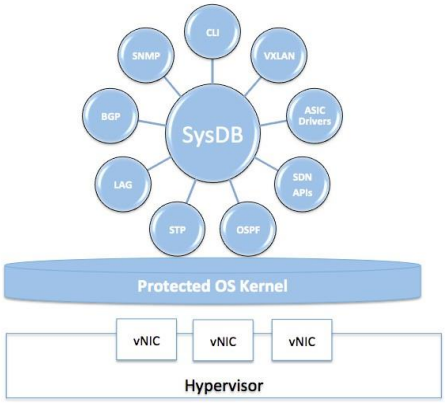

図4：vEOS

vEOSを使用すると、仮想ネットワーク・トポロジを構築してEOSの特長や機能を検証できます。EOSの豊富に用意されたプログラム可能インターフェイスを利用しているお客様は、物理的なスイッチを必要とすることなく、自社のスクリプトやアプリケー

ションの展開およびテストのためにvEOSを使用できます。また、vEOSにより、さまざまなAristaパートナーがEOSを利用して、当社のお客様にとって最適な組み合わせの統合を可能にする拡張機能を設計、構築、テストすることができます。vEOSは、有効なトレーニング・ツールでもあり、VMベースのイメージの単純さと柔軟性により、EOSに慣れ親しんで多様なEOSの機能、ツール、トラブルシューティング技法を学ぶための手軽な方法を実現します。

EOSアーキテクチャのまとめ

イベント駆動型のアーキテクチャ:すべての状態変化は、該当するイベントに登録されているすべてのプロセスにSysDBが通知を送信するきっかけとなります。これにより、システムは負荷の高い状況でも優れた効率と従来より高い耐障害性で動作することができます。

きめ細かいモジュール性と自己回復による耐障害性:EOSでは、システムの安定性向上のために、ソフトウェア障害分離とステートフル障害修復を個々のモジュールで実現しています。また、EOSでは、アプリケーションのトラフィックに一切影響を与えることのない、個々のモジュールのインサービス・ソフトウェア・アップグレードも可能になっています。

豊富なネットワーク機能:EOSは、レイヤ2、レイヤ3、サービス品質(QoS)、アクセス制御リスト(ACL)、管理容易性、セキュリティ、仮想化に関する豊富な業界トップクラスの機能で構成されています。

すべてのレイヤにわたってプログラム可能:EOSは、Linuxカーネル、ハードウェア・フォワーディング・テーブル、スイッチの設定およびCLI、スイッチのコントロール・プレーン、管理レイヤといったすべてのレイヤにわたってプログラム可能であり、自動化、ネットワーク・モニタリング、またはサードパーティ・ツールやオーケストレーション・システムとの統合を実現することができます。eAPIからLinux APIまたはOpenFlowおよびDirectFlowや、EOS SDKにいたるまでのプログラム可能な幅広いインターフェイスを利用して、カスタムアプリケーションを開発したり、サードパーティ・ツールや管理システムと連動させたり、さらにはスイッチのフォワーディング動作をコントロールすることもできます。

拡張性:SysDBを介した状態分離により、ユーザーは、独自のアプリケーションをAristaプラットフォーム上で直接実行して、Linuxや、社内のツール、スクリプト、およびアプリケーションとのネイティブ統合を実現できます。Python、Perlなどのスクリプトは、仮想および物理アプリケーション、SDNタイプのコントローラ・プラットフォーム、レイヤ4～7サービスとのサードパーティまたはネイティブに開発できます。

スマート・システム・アップグレード:EOS内のスマート・システム・アップグレード(SSU)アプリケーションは、ネットワークに対するスイッチの挿入および除去や、ネットワーク冗長性と高度なECMPおよびMLAGトポロジとの統合を利用した、ダウンタイムが発生しないネットワークのアップグレードを可能にします。

ネットワークの可視性:トレーサーおよびモニタリング機能としては、ビッグ・データ/Hadoop(MapReduce Tracer)、仮想化(VM Tracer)、ネットワーク・パス(Path Tracer)、レイテンシー分析(LANZ)、さらにシステム全体の正常性モニタリング用のものが存在します。

ネットワーク・テレメトリ:ネットワーク・タップ・アグリゲーションとデータ分析(DANZ)は、トラフィック・ミラーリングや、ユーザーのトラフィックに影響を与えないネットワークのモニタリングを可能にします。Splunk、sFlowベースのコレクター、アプリケーション・モニタリング・ツール(Corvil、ExtraHopなど)との統合により、トラフィックの可視性を実現します。

ネットワーク自動化:EOSは、Puppet、Chef、CFEngine、Ansibleをネイティブでサポートしています。これらのツールでは、データセンター環境内のサーバーやストレージと同じ方法でネットワーク設定を行うことができます。また、EOSはネットワークの運用コストを大幅に削減するツールもサポートしています。たとえば、ゼロ・タッチ・プロビジョニング(ZTP)は、ネットワーク・インフラストラクチャのプロビジョニングを自動化し、新しいサービスを実稼働に移すまでの期間を短縮しつつ、ヒューマン・エラーのリスクをなくします。また、ゼロ・タッチ・リプレイスメント(ZTR)は、差し替えるスイッチの自動プロビジョニングを実現し、障害の発生したスイッチの平均交換時間を大幅に短縮します。

ネットワーク仮想化:VMwareのNSX、MicrosoftのSystem Center、OpenStackによるKVMなどの仮想化環境や物理的なベアメタル・サーバーはすべて、EOSを介して相互に接続することができます。オーバーレイ・テクノロジー(VXLANなど)により、

仮想および物理ネットワーク・インフラストラクチャが含まれている混在環境でより規模の大きいなマルチテナント・ネットワークを実現できます。

vEOS：Arista vEOSは、物理スイッチ上で動作するEOSソフトウェア・プラットフォームを仮想マシンベースへと拡張したものです。vEOSでは、ネットワークの設計と検証が可能であり、EOS上での拡張機能やアプリケーションの開発およびテストを簡単に行うことができます。また、EOSで提供する必要がある機能やツールに慣れ親しむための有効なトレーニング・ツールにもなります。

まとめ

アリスタネットワークスのEOSソフトウェアは、クラウド・データセンターを展開するための重要な基盤になっています。非常に高度で、耐障害性とプログラマビリティに優れたオペレーティング・システムであり、業界をリードするネットワーク・サービス、運用上のイノベーション、統合機能を実現しています。

詳細については、以下を参照してください。http://www.aristanetworks.com/en/products/eos

参考資料

スイッチのオペレーティング・システムとしてのLinux：5つの教訓
https://eos.aristanetworks.com/2013/11/linux-as-a-switch-operating-system-five-lessons-learned/

ARISTA

アリスタネットワークスジャパン合同会社

〒170-6045　東京都豊島区東池袋3-1-1　サンシャイン60　45F
Tel: 03-5979-2012（代表）

西日本営業本部

〒530-0001 大阪市北区梅田2-2-2
ヒルトンプラザウエストオフィスタワー19階
Tel: 06-6133-5681

お問い合わせ先

Japan-sales@arista.com

01/15