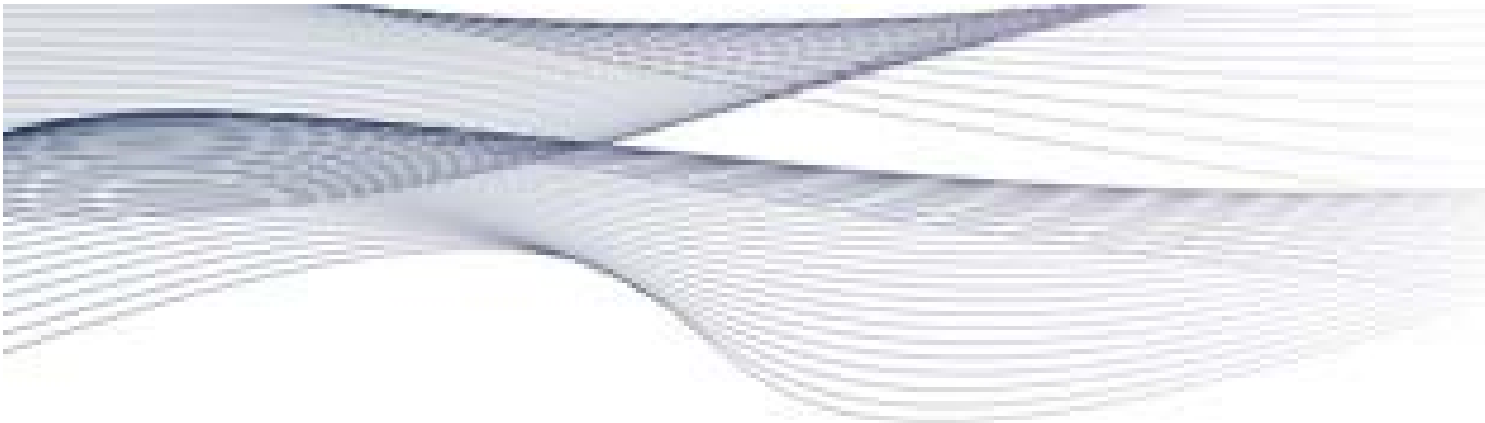




ARISTA 白皮书

EOS：下一代可扩展操作系统

现在，跨整个网络的性能、弹性和可编程性是下一代云和企业数据中心网络的基本业务需求。服务开通及网络运营的敏捷性和部署可扩展性需求，要求达到新的自动化级别以及实现与当前数据中心基础设施的集成。网络操作系统的底层设计提供满足这些要求的架构基础。

Arista Networks 已设计和交付业界领先的、基于 Linux 的网络操作系统 EOS（可扩展操作系统）来解决这些需求。EOS 是一个功能强大、可编程的、创新操作系统，具有单个软件映像，可跨 Arista 屡获殊荣的网络交换机的整个产品组合运行，也可在虚拟机实例 (vEOS) 中运行。这保证了一致的运作、工作流程自动化和高可用性，同时显著降低了数据中心运营费用。

本白皮书阐述了当前网络操作系统，并解释了 Arista 的 EOS 架构如何能够独一无二地支持下一代数据中心。

ARISTA

传统网络操作系统的局限性

传统的企业数据中心网络已经长久以来容易受到软件崩溃和计划外中断的影响。跨交换机平台的多个不同软件版本使新功能和服务的部署过程冗长且耗时。再加上容易出错的手动配置方式，网络正常运行时间不可避免地受到影响。

这个问题的核心是，今天大多数网络设备上仍然在运行老化的单一整体软件。这对网络的高度可靠性造成了根本性的限制。实际情况是，这些共享内存系统中任何位置的一个漏洞或缺陷会暴露整个网络导致中断，因为没有软件故障封闭机制。同时，由于多个任务之间没有隔离，因此很难从性能角度上扩展这些操作系统或可靠地添加新功能。这种传统软件的脆弱性本质上阻碍了网络操作系统扩展以实施新功能的能力，如与客户特定管理流程或现代数据中心环境中部署的其他系统和服务的集成。

早期的网络操作系统被逐步构建在定制内核的顶部，状态信息任意分布在整个系统之中，或者更糟，嵌入到了内核本身。在这些环境中，所有进程通常在共同的地址空间中运行，这会导致风险命运共享 - 一个缺陷可能导致整个系统崩溃并造成严重的网络中断。软件状态使用同步或异步轮询方式维护，这是一种效率低下的机制，通常每隔几毫秒或几秒就检查每个接口和内部数据结构的状态，造成了浪费的周期、可能的死锁和多个并发事件之间的竞争状况。

状态的任何变化都由脆弱的、费心设计的代码路径来处理，以使软件进程按照严格有序的序列作出反应，具体取决于基础事件。对这种架构中所有可能的代码路径进行测试会导致错综复杂的状况、长期的开发和扩大的客户测试以及延长的认证周期。

此外，软件功能按照进程之间的矩阵式消息传递来分层，使得进程间的通信犹如迷宫并造成系统错综复杂的状况，如下面的简化图 [1] 所示：

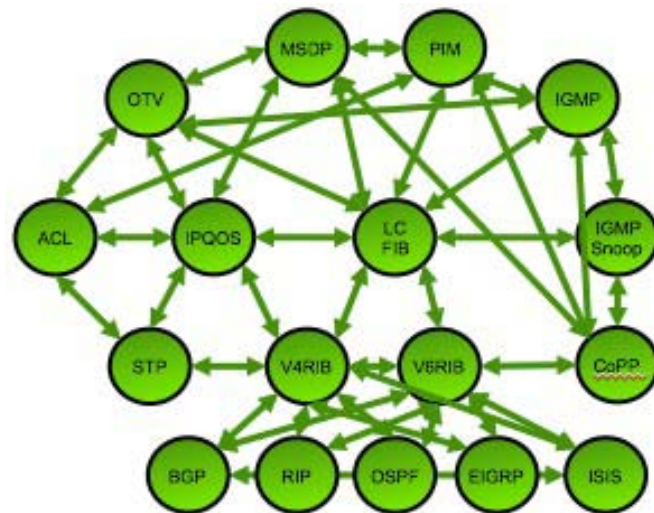


图 1：传统的网络操作系统通信（大约 2000-2010）

最后，内部状态不能暴露给用户和应用程序，并且这些堆栈中唯一可能的可编程性是经由高级别 API 包装器来进行，这可以提供与 SNMP 或文本中的 CLI 或 XML 格式相同的信息用于监视目的。

对于这种次优软件架构，构建云网络和软件定义的数据中心是不可能的。

ARISTA EOS: 现代解决方案

为了解决公共/私有/混合云网络的需求，同时能够使用新的应用程序和迎来软件定义的网络 (SDN) 时代，Arista 的可扩展操作系统 (EOS) 已从底层开始设计并针对这些要求苛刻的新数据中心环境进行了优化。

Arista 的可扩展操作系统是业界最先进的网络操作系统。它集现代软件和操作系统 (O/S) 概念、透明可重启的流程、开放平台开发、未经修改的 Linux 内核和有状态的发布/订阅数据库模型于一体。

为充分了解一个真实、开放的操作系统所提供的优势并探索全面自动化解决方案的潜力，对 EOS 的底层架构进行仔细审视是很有必要的。在以下章节中，我们将考察 EOS 架构的主要组件和特性，以及在 EOS 上构建的丰富网络服务阵列。

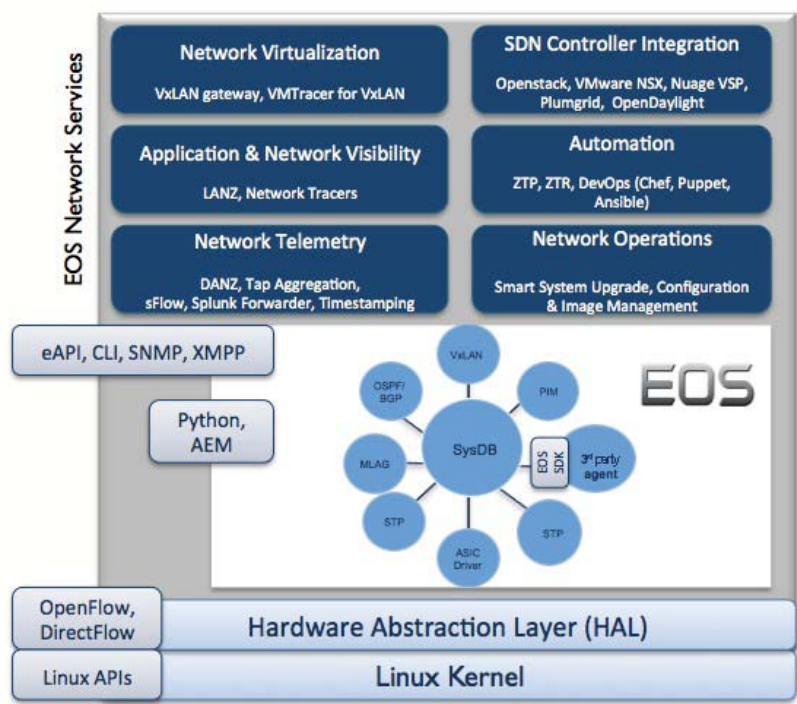


图 2: Arista EOS 架构

ARISTA EOS: 使用 Linux 内核

Arista EOS 的基础部分是一个未经修改的 Linux 内核。使用 Linux 内核的主要优点是：

- 它驻留于公共领域
- 数以百计的贡献者以及数百万的用户都对其稳定性、能力和功能集有所贡献
- 它在本质上是极其稳定和模块化的
- 数据包、应用程序存储库的可用性以及源代码的获取

- Linux 的使用非常普遍，因而在许多不同系统和环境中被广泛接受。

ARISTA EOS: SYSDB

将系统的功能控制分隔为多个进程大大增强了弹性和故障隔离，但需要一个机制来协调系统内的操作。这就是 Sysdb 的作用。

如图 2 所示，EOS 的核心部分是系统数据库。Sysdb 是一个内存内数据库（由机器在运行时生成），它运行于用户空间，包含系统的完整状态。如传统数据库一样，Sysdb 不包含任何应用程序逻辑，只负责保持状态。然而，Sysdb 并非被优化用于事务处理，而是被设计用于同步进程间的状态，由于其会在发生变化时通知关注的进程或代理，因而也被称为“代理”。

系统中的所有代理从 Sysdb 装入其配置和状态。这非常像一个文件系统装入，其中会为每个装入点指定只读或读写权限。当代理从 Sysdb 装入时，它将在该装入点中接收自己的所有状态的本地副本。由于 Sysdb 保持在 RAM 中，一旦关闭或重启交换机，信息将会丢失。

Sysdb 就像一个事件驱动的发布/订阅模型。如果代理的状态改变，Sysdb 将发送一个事件通知给该代理，这将更新其本地副本。同样地，当代理写入装入点时，它只改变其本地副本，并且写入会立即返回。

每个 EOS 代理会订阅到 Sysdb 以在其他代理的状态在 Sysdb 中发生变化时获得通知。当某个代理的状态已改变时，改变通知会缓冲并异步发送到 Sysdb，然后通知已订阅到此改变的代理的所有其他代理。

这种将状态传送到整个系统的集中式数据库方法和自动生成 Sysdb 代码的方式，可以减少风险和错误，改善软件功能速度，并让客户能够享有使用相同的 API 来接收 Sysdb 通知以定制并扩展交换机功能的灵活性。

ARISTA EOS: 主要优势

EOS 在保留继承 Linux 的安全性、稳定性、开放度、模块化和可扩展性的同时，提供极其稳定和可靠的数据中心通信服务。这一业界独特的组合有望显著改进下一代数据中心网络的功能和演变。

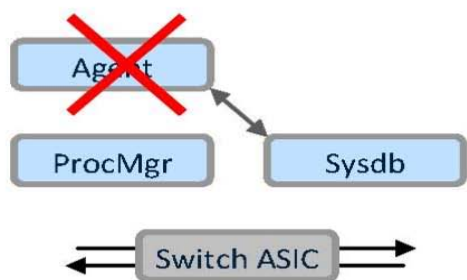
- Arista EOS 为独特的多进程状态共享架构，将状态信息与进程本身分开。这反映了 Arista 的核心软件设计理念，使故障恢复和实时软件更新建立在进程级基础之上，不影响系统的运行状态。
- 此外，协议处理、安全功能、管理服务，甚至设备驱动程序都是在用户地址空间而非内核本身中运行。这大大提高了整体稳定性，而且通过坚持保有 Linux 环境纯净性的设计原则，Arista 能够轻松扩展操作系统的附加功能。

- EOS 的相同二进制映像可以跨任何产品系列进行部署。这可以提高每个平台的测试深度，并使功能和缺陷修正方案兼容所有平台。这也使得用户在其数据中心环境中部署、认证和验证新版本变得更为简单。
- 最重要的是，通过为用户提供对 Linux 内核不受限制的访问，用户现在可以开始利用真正的数据中心自动化特性。用户能够获得写入 Shell 或 Python 脚本的权限，开始在 IT 企业的其他部分中以类似方式利用网络。
- 安全性，EOS 支持身份验证机制（如 TACACS+ 和 RADIUS AAA 特性），允许任何企业将任意交换机完全锁定为仅允许授权的和安全的访问。
- EOS 提供的开发框架支持“可扩展性”这一核心概念。它采用开放式基础架构，提供同类最佳的软件开发模型，可改善功能速度和正常运行时间，简化维护工作，并允许选择不同的工具和选项。

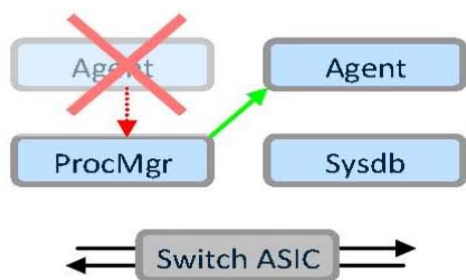
ARISTA EOS：弹性

本部分将详细说明 EOS 多进程状态共享架构如何能够提供更高可用性、减少维护窗口、改进可管理性，和实现更严密的安全性。

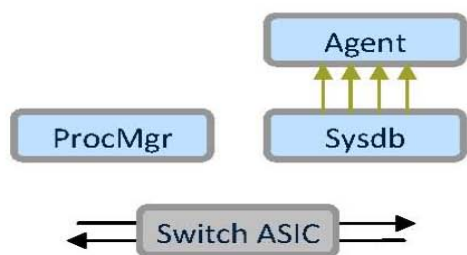
系统高可用性的关键是故障封闭和软件自修复能力。在大多数嵌入式平台上，任何软件故障都将导致重新加载，造成几秒甚至几分钟的停机时间。在 EOS 中，任何故障都将封闭在故障产生的代理或驱动程序内。如果故障导致代理崩溃，则 EOS 进程管理器 (ProcMgr) 会立即将其重启。如果故障导致代理挂起或循环，ProcMgr 会检测状况并重启代理。因此，EOS 内的故障可以自修复。这一过程如下图 [4] 所示。



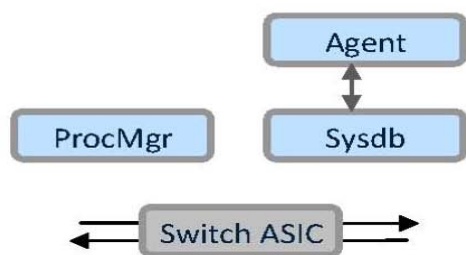
1. The agent experiences a fault. The agent exits without affecting packet forwarding or other processes.



2. ProcMgr detects process exit and starts a new agent instance. Packet forwarding continues.



3. The new agent loads its state from Sysdb without data path disruption.



4. The system resumes normal operation.

图 3：EOS 故障封闭和自修复

EOS 多进程状态共享架构允许在交换机正常操作期间执行更多维护任务，它也是减少维护窗口的关键。任何 EOS 代理都可以实时修补和重启，不中断交换机操作。数据流不受模块升级过程影响，所以不存在用户可察觉的停机时间。

改进操作的关键内容是支持第三方管理和业务流程整合以及任务自动化软件的能力。由于 EOS 为子系统和代理提供了一个强大的、受保护的环境，

它也可以安全地运行经过验证的第三方代理，在特定客户环境内调整交换机行为以优化可管理性或自动化常规任务。因此，Arista EOS 客户可以放心地进行部署，深知其网络操作系统将防范任何第三方软件问题。

改进安全性的关键内容是将安全漏洞的影响封闭在有漏洞的代理内。例如，如果 SNMP 子系统存在一个漏洞，则漏洞利用可能会读取所有 SNMP 可访问状态；但是，漏洞利用将不能创建附加用户帐户、重新配置接口，或运行外部软件。换句话说，正如 EOS 架构会将故障封闭到单个模块中，它也会封闭安全漏洞的影响。最后，依靠与提高管理和业务流程支持相同的可扩展性机制，第三方软件将可以实施自定义安全策略或入侵检测以进一步提高安全性。

ARISTA EOS: 可编程性和可扩展性

EOS 在所有层上都可以编程 – Linux 内核、硬件转发表、交换机配置和 CLI、交换机控制平面以及管理层。EOS 的这种可编程性能够实现与众多第三方应用程序的快速集成，以进行虚拟化、管理、自动化、编排和网络服务。Arista EOS 提供一套丰富的可编程接口，包括：

- Linux shell 访问和 API
- OpenFlow、DirectFlow
- Sysdb API
- Python、Perl 脚本编制、高级事件管理 (AEM)
- EOS SDK
- 基于 JSON 的 eAPI、CLI、SNMP、XMPP

Arista EOS 具有根级管理员的完全 Linux shell 访问权限，并有广泛的基于 Linux 的工具可用。OpenFlow 和 DirectFlow 允许客户对交换机的转发状态进行编程，以便根据应用需求微调数据包转发。Sysdb API 提供对所有内部状态的访问，包括低级别的计数器、温度测量、电源状态以及监视和管理系统本身的所有其他必要参数。基于 JSON 的 EOS API (eAPI) 提供与工具方便的、基于 Web 的集成，这些工具通常用于管理计算和存储资源以及编排系统。甚至以 Python 编写的 CLI 也是可定制的。基于 Python、Perl 等的脚本也可以被开发作为第三方或其本身与虚拟和物理应用、SDN 类型的控制器平台和 4-7 层服务进行集成。

使用 EOS 软件开发工具包 (SDK)，客户能够以 C++ 或 Python 开发其自己的自定义 EOS 应用程序。这种 EOS 开发模式允许这些第三方应用程序成为 EOS 以及其他 EOS 代理的一级成员应用程序。SDK 为 Arista EOS 中可用的软件抽象提供编程语言绑定，因此第三方代理可以访问交换机状态并对网络事件作出反应。例如，这些应用程序可以管理接口、IP 和 MPLS 路由，访问控制列表 (ACL)，以及使用一组 API 在交换机和监视或网络控制器之间进行通信。SDK 既面向需要事件驱动通知的长期运行进程，也面向需要与其他 EOS 代理进行高性能相互作用的脚本。通过模块化架构造就的 Sysdb 和内在故障隔离机制实现状态分离，可允许客户开发并安装自己的应用程序，而不必担心中断整个系统。

客户可利用这种丰富的可编程性以快速的周转时间来构建定制应用程序、与生态系统工具集成，甚至解决功能空白，这真正使 EOS 成为现代敏捷性云数据中心正确的网络操作系统选择。

ARISTA EOS: 网络服务

在 EOS 业界领先的基础之上，Arista 添加了一套全面的网络服务，包括丰富的 2 层/3 层功能、网络虚拟化支持、网络可见性和遥测、自动化、SDN 控制器集成。所有这些服务均利用 EOS 提供的丰富可扩展性来与第三方工具集成，并以务实的、全网络方法来解决一些客户的难点。例如，ZTP 有助于自动化和模板化大量交换机的服务开通，智能系统升级 (SSU) 有助于执行不中断的软件升级，CloudVision 能够使用 OVSDB、eAPI 或 OpenFlow 实现与 SDN 控制器的全网络集成。

ARISTA EOS: 智能系统升级

智能系统升级 (SSU) 是网络服务的一个实例。SSU 为网络提供优美的软件升级。执行不中断的系统软件升级以引入新的网络服务或缺陷修复程序的能力，对于保持数据中心环境所需的可用性级别至关重要。

早期的网络操作系统通常遵循电信类模型的不中断服务软件升级 (ISSU)，生产中的单个平台最多能够接入到逐步升级的代码版本，其中也许包含一些新的功能或修复程序。传统的 ISSU 方法一直以来背负着必需的巨大量复杂软件开发。特定 ISSU 代码必须写入到

所有可能的检查和权衡的说明之中，当系统在两个不同版本的软件之间跳转时，会尝试保持和转换所有硬件和软件状态。随着功能集的增多、功能彼此之间的相互作用，以及相关状态信息的增长，这一挑战不断加剧。这种错综复杂的状况的副作用是，极端案例屡见不鲜，而且并非能够解决所有情况下的问题。即使额外增加测试周期，对发布时间表造成了进一步的‘ISSU 负担’，也不能保证把握到所有案例

利用开放式、可编程 EOS 的卓越架构以及与其他应用程序和基础设施组件的直接集成，对于软件维护 SSU 站在了更加全面的网络角度上，允许透明化删除或添加网络元件，而通信将被转移或者完全不受影响。Arista 的 SSU 设计作为数据中心基础设施维护的完整解决方案，具有以下主要优势：

- 智能插入和移除网络元件，可定制为骨干角色或叶子角色
- 通过编程升级到新软件版本，不会导致系统中断
- 与所有应用程序和基础架构元件开放式集成
- 简化的解决方案：有意避免了其他方法所需的大量状态维护和状态转换过程的复杂性

因此，Arista 的服务开通模型允许经由零接触服务开通 (ZTP) 或零接触更换 (ZTR) 实现交换机的初始部署或更换，在此之后，SSU 框架可通过支持无缝升级（根据情况，在叶子层或骨干层）来确保连续运行。

ARISTA vEOS: EOS 作为虚拟机

Arista vEOS 将物理交换机上运行的 EOS 软件平台扩展为基于虚拟机的产品。vEOS 继承了本文所述的所有 EOS 架构特性和优势。vEOS 是实际的 EOS 映像，而不是模拟器。

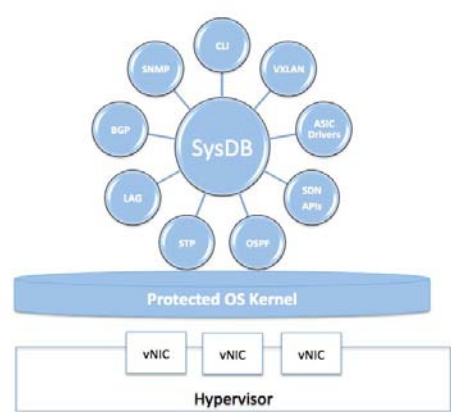


图 4: vEOS

vEOS 可用于构建虚拟网络拓扑和验证新的 EOS 特性与功能。已利用 EOS 丰富的可编程接口的客户，可以使用 vEOS 开发和测试他们的脚本或

应用程序，而无需使用物理交换机。vEOS 也使 Arista 的各类合作伙伴能够轻松访问 EOS 来设计、构建和测试为我们的客户实现同类最佳集成的扩展。vEOS 也是一种有效的培训工具 - 基于虚拟机 (VM) 的映像的简便性和灵活性提供了一种简单方法来熟悉 EOS 和了解各种 EOS 功能、工具以及故障排除技术。

EOS 架构总结

事件驱动的架构：所有状态变化将触发通知，通过 Sysdb 传递给已针对该事件注册的所有进程。这可使系统在认真的负载下运行，从而效率更高和更有弹性。

粒度模块化和自修复弹性：EOS 提供独立模块的软件故障封闭和状态故障修复，从而实现超强的系统稳定性。EOS 还允许独立模块的不中断服务软件升级，对应用程序流量不造成任何影响。

丰富的网络特性：EOS 包括丰富的、业界领先的 2 层、3 层、服务质量 (QoS)、访问控制列表 (ACL)、可管理性、安全性和虚拟化特性。

在所有层上都可以编程：EOS 在所有层上都可以编程 – Linux 内核、硬件转发表、交换机配置和 CLI、交换机控制平面，以及用于自动化或网络

监视或与第三方工具和业务流程系统集成管理层。与系统状态相互作用的众多可编程接口（从 eAPI 到 Linux API 或 OpenFlow，以及 DirectFlow 到 EOS SDK）可以用于开发自定义应用程序、与第三方工具或管理系统相互作用，甚至影响交换机转发行为。

可扩展性：通过 Sysdb 造就的状态分离，允许用户直接在 Arista 产品组合上运行自己的应用程序，实现与 Linux 和内部工具、脚本及应用程序的本地集成。基于 Python、Perl 等的脚本也可以被开发作为第三方或其本身与虚拟和物理应用、SDN 类型的控制器平台和 4-7 层服务进行集成。

智能系统升级：EOS 内的智能系统升级 (SSU) 应用程序，可实现从网络正常插入和移除交换机，通过利用网络冗余以及与先进的 ECMP 和 MLAG 拓扑的集成可实现网络的零停机时间升级。

网络可见性：Tracer 和监视功能用于大数据/Hadoop (MapReduce Tracer)、虚拟化 (VM Tracer)、网络路径 (Path Tracer)、延迟分析 (LANZ) 以及整个系统运行状况监视。

网络遥测：网络 TAP 聚合和数据分析 (DANZ) 可实现网络的流量镜像和监视，对用户流量没有任何影响。与 Splunk、基于 sFlow 的收集器，和应用程序监视工具（如 Corvil 或 ExtraHop）集成，提供流量可视性。

网络自动化：EOS 本身支持 Puppet、Chef、CFEngine 和 Ansible，使网络配置具有与数据中心环境中的服务器和存储相同的方式。此外，EOS

可支持工具，大大降低了网络运营成本。例如，零接触服务开通 (ZTP) 自动化网络基础设施的服务开通，缩短新服务产生的时间，同时消除人为错误的风险，零接触更换 (ZTR) 提供更换交换机的自动化服务开通，显著减少了发生故障的交换机的平均更换时间。

网络虚拟化：虚拟化环境，如 VMware 的 NSX、Microsoft 的 System Center、OpenStack 上的 KVM，以及裸机物理服务器，全部可以通过 EOS 互连。重叠技术，如 VXLAN，可在包含虚拟和物理网络基础设施的混合环境中实现更大规模的、多租户网络。

vEOS: Arista vEOS 将物理交换机上运行的 EOS 软件平台扩展为基于虚拟机的产品。它使网络设计和验证可提供一种简单方式在 EOS 上开发和测试扩展与应用程序，它也是增加对 EOS 所提供的功能和工具的熟悉程度的一种有效培训工具。

总结

Arista 的 EOS 软件是部署云数据中心的关键基础。它是最先进的、弹性、可编程操作系统，提供业界领先的网络服务、运营创新和集成功能。

有关更多信息，请访问：<http://www.aristanetworks.com/en/products/eos>

参考资料

Linux 作为交换机操作系统：五大经验教训

<https://eos.aristanetworks.com/2013/11/linux-as-a-switch-operating-system-five-lessons-learned/>

ARISTA

圣克拉拉市—公司总部
5453 Great America Parkway
Santa Clara, CA 95054
电话：408-547-
5500
www.arista.com

爱尔兰—国际总部
4130 Atlantic Avenue
Westpark Business
Campus Shannon
Co. Clare, Ireland

新加坡—亚太地区行政办事处
9 Temasek Boulevard
#29-01, Suntec Tower Two
Singapore 038989

版权所有 © 2015 Arista Networks, Inc. 保留所有权利。CloudVision 和 EOS 是注册商标，Arista Networks 是 Arista Networks, Inc. 的商标。其他所有公司名称是其各自持有者的商标。本文档中的信息如有更改，恕不另行通知。有些功能可能尚不可用。Arista Networks, Inc. 对于本文档中可能出现的任何错误不承担任何责任。

01/15