ARISTA

Technical Bulletin

Arista Advanced Event Management

"Event Driven Networking for the Cloud"

Why is AEM needed?

- Minimize downtime
- Reaction to a changing environment
- Reduce 'operational malaise'
- Predictive fault management

Critical components:

- Event Manager
- Event Monitor
- Linux tools
- CLI Scheduler

Introduction

The Data Center has gone through many different stages from statically deployed services to today's highly dynamic environments. As the Data Center has evolved there have been many challenges and corresponding solutions (e,g, virtualization, performance, power, etc...). One challenge that has yet to be completely addressed is the operation and monitoring of these dynamic environments. Reacting to events, maximizing uptime, collecting performance statistics and implementing predictive fault management have all lagged behind. Traditionally these functions are based on proprietary implementations of open standards that do not fit today's dynamic environments.

While networks protocols handle dynamic topology changes this is not always done in an optimal fashion or coordinated with the application layer leading to optimal use of the existing network resources. Once an issue is known troubleshooting and the associated reaction times can take hours for services to be restored resulting in lost revenue, substantial damage to a company's reputation, staff morale downturn, and customer loyalty damages. These losses can be substantial and are completely unnecessary.

In an attempt to react to network events and piece together the operational puzzle the amount and frequency of data collection has increased dramatically. Unfortunately increasing the amount of data collected does not directly translate into solving the challenges. There are many cases where the increase in data collection actually has the opposite effect, overloading the network operations staff producing 'an operational malaise' where critical pieces of information get lost in the fog of data.

Traditional methods have served the static data center well in the past, but with the movement toward the virtualized cloud, an event driven service model is needed. Arista Networks solves these challenges by leveraging the power of Arista's Extensible Operating System (EOS), providing a unique set of tools to help directly address the challenge of advanced event management.

What is Advanced Event Management?

Advanced Event Management (AEM) is a powerful and flexible set of tools to automate tasks, customize the behavior of the system and associated operation of the switching infrastructure. Leveraging an open operating system, AEM allows operators to fully utilize the intelligence within EOS to respond to real-time events, automate routine tasks, and take local automated action based on changing network conditions. Simplifying the overall operations, AEM provides the tools to customize alerts and actions. It is composed of three major components:

- Event Manager
- Event Monitor
- Linux tools

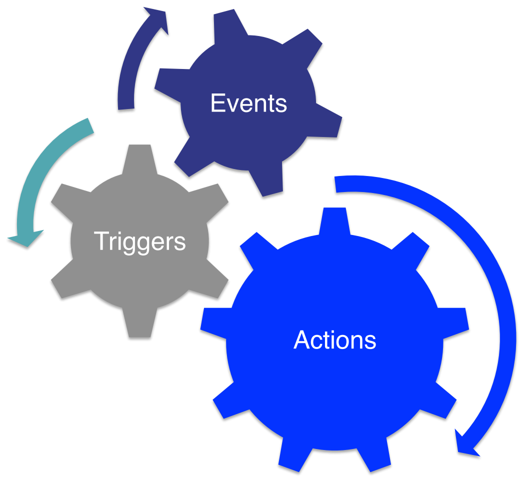
These services singly or combined allow engineers, operators, and partners to take advantage of the functions inherent in EOS. Arista hardware can be customizable in ways that work intimately with each customers unique operating environment. While other vendors are busy imposing limitations on accessing the internal workings of the operating system, only Arista opens this functionality. AEM allows user customization of the system, now the system can programmatically react to events ushering in the age of the event driven service model.

Event Manager

Event Manager reacts to changing system events and leverages the underlying Linux OS allowing the creation of useful tools and even new features that can tailor a system to act as required by the operational model not dictated by a vendor. Event Manager is comprised of three components:

- Events
- Triggers
- Actions.

Events are a binary value and triggers are the transitions of those states. Events and triggers are associated with system objects to watch for changes in their state (e.g. link up/down).



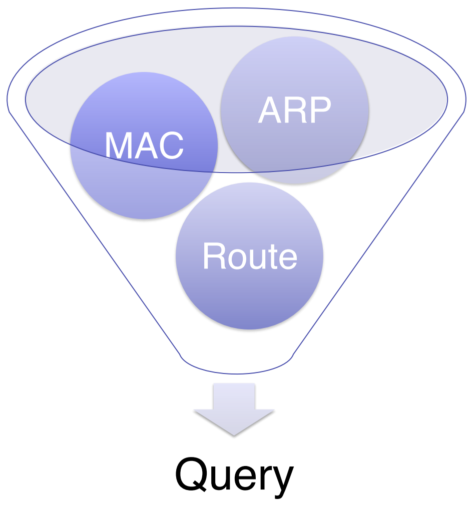
Triggers initiate an action this can be a BASH / CLI script, an executable or something written in a more advanced scripting language (e.g. Python, Perl, Ruby, etc..). Utilize well known tools customization of the system, based off changing system conditions, is possible.

Event Monitor

How many times is relevant historical information lost? Not information collected and then misplaced or deleted, the data that was never collected as the volume is too great and the practicality of not collecting it outweighs the usefulness of the data. The fact is, especially in times of trouble, once a data center is experiencing an error condition all data is relevant and can be the potential needle in a haystack that helps solve the issue at hand.

When a problem happens, restoration of service is the immediate issue that needs to be addressed. Once service is restored, a complete root cause analysis is a general requirement to ensure the issue does not re-occur. Unfortunately these two goals many times are orthogonal, as much of the data is never logged during normal systems operation and/or lost when measures are taken to restore service.

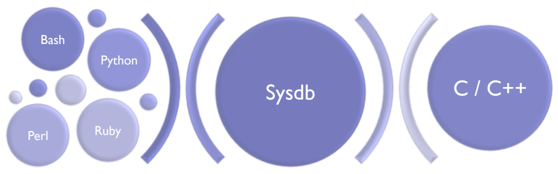
"…Arista's focus on a modern software-defined network operating system, that is customizable with built-in automation capabilities is a welcome change for the cloud-era…"
- Zeus Kerravala, ZK Research



Event Monitor solves this problem by collecting and storing critical information that is historically not collected, except by frequent and intrusive polling techniques (until now). Now critical information (e.g. ARP, MAC and route tables) about the state of the network is collected and kept in a local database that can be viewed on demand. Using prebuilt or user customizable SQL queries, this information can be used to determine the state of the network at the time of the problem, instead of waiting for the problem to occur again and again…

Linux tools

Offering a full compliment of Linux based tools, allows the end user complete control of the operation of the system. Building a truly automated data center is now possible.



All system information and state information are available via the Arista EOS System Database (Sysdb). Ways to interact with Sysdb is only limited by the imagination, a native Python client is included providing the ability to build custom alerts, reactions, functions and features.

Interaction with the system can be based on:

- CLI / Bash scripting
- Advanced scripting (Python, PERL, Ruby, etc...)
- Compiled executables

With thousands of open source resources available, Arista EOS provides the only open system architecture in the industry--and is setting the trend for future leading-edge solution engineering in the Data Center.

Summary

Traditionally, the internal workings of the system are vendor proprietary functions, which force the end user to wait for a vendor to implement a specific function / feature. In many cases this can takes years, if it happens at all. The proprietary implementation approach has handicapped network automation. Archaic command line interpreters, lack of vendor commitment to open standards, flexible programming languages, and modern development frameworks have limited innovation. Whether it is monitoring link integrity, programmatically reacting to events, or dynamically configuring the system, AEM provides the flexibility to accomplish any task required.

The era of the cloud is upon us, dynamic environments and increased automation is the norm not the exception. Arista EOS was designed to be an open and programmable network operating system, one designed for the cloud era.

In the sprit of open systems Arista EOS Central provides a repository of tutorials and examples visit:

http://eos.aristanetworks.com

To learn more about, or contribute to the extensibility of Arista EOS, please contact us:

info@aristanetworks.com