Avi Vantage Platform Architecture

Industry's first intent-based application services platform based on a software-defined scale-out architecture for multi-cloud environments

EXECUTIVE SUMMARY

Application development, deployment, delivery and consumption are undergoing a radical transformation. Unfortunately, appliance-based application delivery controller (ADC) solutions provide load balancing, content switching, rewrite, optimization, and application security have not evolved to keep up with this transformation.

The Avi Vantage Platform, from Avi Networks, is purpose-built for the cloud and mobile era using a unique analytics-driven and software-defined architecture that separates data plane from control plane to mirror the needs of application usage.

Avi is intent-based and allows customers to focus on outcomes instead of manual inputs/configurations that drive outputs. Avi offers software load balancing, security (iWAF), and elastic service mesh for container applications on the Avi Vantage platform. This white paper describes the Avi Vantage architecture in detail.

TRENDS

Several recent industry trends have transformed the way applications are delivered and consumed.

1. **App-centricity and Self-Provisioning with Agile DevOps Practices**
   Applications, with lines of business (LoBs) adopting continuous integration and delivery (CI/CD) practices, have become the lifeblood of organizations. Agile DevOps teams focus on achieving quick application delivery. With an increased mobile access to applications, organizations are even considering a per-application or a per-tenant load balancer.
2. **Monolithic to Microservices and Container Architectures**
   The microservices architecture decomposes large monolithic applications to a set of smaller services called microservices that are developed, deployed, and scaled independently. Each self-contained microservice has the same network services requirements as the (former) larger application such as load balancing, security, content optimization, transformation, etc. The overall application may be stitched together on different clients using microservices, e.g., a Javascript program in a desktop Web browser, a native mobile app, and an HTML5 tablet app – all issuing REST calls to the same set of microservices in the backend.
3. **Multi-Cloud Environments Requiring "Public-cloud-like" Flexibility**
   Enterprise applications are being deployed in diverse locations -- on-premises, private clouds, and public clouds. The same application may be developed and tested in a private cloud and deployed in the public cloud. Regardless of the location, all applications require application delivery and security services from a single point of control for consistent, secure and agile application deployment.

ABOUT THIS DOCUMENT

This paper details how app-centric, microservices architectures, multi-cloud environments, and commodity x86 hardware have affected how applications are delivered. Unlike legacy hardware ADCs, the Avi Vantage Platform is a software-defined solution that delivers application services in the data center and/or the cloud with centralized management and control. This paper also delves into details of the architecture.

4. **Leveraging Commodity x86 Hardware**
   Applications are tested and deployed across diverse infrastructure, including bare metal, VMs, and containers on bare metal or in a VM. They can have unique networking deployment models that range from physical to virtual overlays to NAT with private subnets per host, etc. When legacy physical or virtual appliance-based ADCs are plugged into a physical fabric, traffic has to be stitched together and the devices have to be integrated into every networking/SDN control plane, making deployments unwieldy, clumsy, and complex.

SHORTCOMINGS OF LEGACY APPLICATION DELIVERY SOLUTIONS

Traditional ADCs have at least three major shortcomings in how they are deployed, consumed, and managed.

1. **From custom ASIC to standard x86**
   Commodity x86 CPUs are more than adequate to perform all advanced ADC functions, including crypto using specialized instructions (AES-NI from Intel) even for the most demanding applications. Switching all of Google applications to use TLS barely increased the overall load on CPU by 1-2% according to this blog. Observations at Amazon Web Services (Reference: AWS Re: Invent Conference 2013 talk) show average server CPU utilization in data centers at 12-15%. Hence, it makes sound economic and environmental sense to implement network services as software functions on commodity servers instead of specialized hardware appliances. This is also the rationale behind the Network Functions Virtualization (NFV) architecture.

2. **No separation of data, control and management planes**
   See Figure 1 for comparison between various approaches.

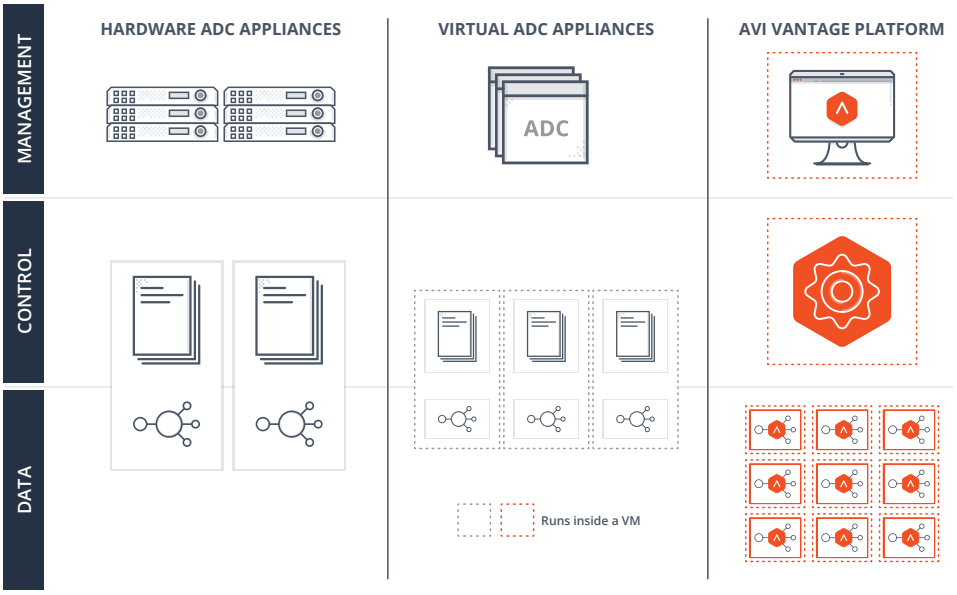

Figure 1: Separation of data, control and management planes

A secure cloud service portal should provide:

- **Enterprise class security:** Central administration for policy management of tenants/users/roles, security profiles, and the ability to access audits/logs/events/alerts from a single point for all services
- **Automation and self-service:** A self-service portal where users can create/modify/delete a service using a UI or APIs that also serve as a central policy repository for all service instances
- **Central control:** Lifecycle management for the service data plane, including automatic placement, capacity management (elasticity) and high availability for all services

Legacy ADCs are managed as individual appliances, not as service instances. Some appliance vendors provide a manager of managers for device management, but the pitfalls of element/device management versus SDN are well known. A true SDN architecture is needed to provide services for the cloud. See Table 1.

Legacy ADC	Shortcomings
Use proprietary hardware	Spares, installation, scaling issues
No separation of control of data planes	Each ADC appliance needs to be individually configured, managed, and troubleshot
Don't use real-time user-to-app analytics	ADCs are overprovisioned and underutilized

Table 1: Application traffic is directed to the optimal VIP

3. **Analytics-driven application delivery**
   ADCs reside at a strategic location in the application topology, view every transaction that traverses the system and have direct access to a wealth of user, client, application, and infrastructure information. However, traditional ADCs provide little information beyond time series graphs or SNMP counters for a few basic metrics such as packets/ bytes or requests/ connections, and they neither provide nor consume any of the rich analytics information they have access to.

AVI VANTAGE PLATFORM

Avi Networks has built a next-generation software-defined solution providing secure, reliable, and scalable application delivery services for applications in the data center and/or the cloud from a single point of management and control. At the heart of the Avi Vantage that is the industry/s first solution that separates the data plane from the control plane. See Figure 2.

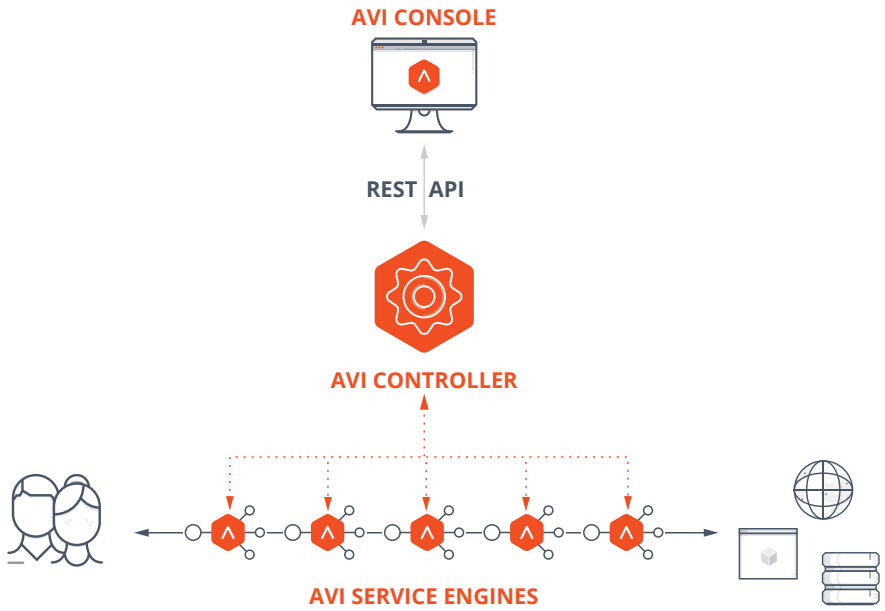

Figure 2: Avi Vantage Platform

AVI SERVICE ENGINES

Here are a few use cases:

Avi Service Engines provide a single-pass data plane for load balancing, health monitoring, content optimization, content transformation, and application security. They also serve as probes that analyze every application transaction. Avi Service Engines are co-located as VMs or containers with applications on standard x86-based servers, forming a distributed microservices infrastructure and providing comprehensive application delivery services. See Figure 3.

Single-Pass Pipeline Architecture

Processing on Avi Service Engines occurs in modular, event-driven, non-blocking, asynchronous pipelines performing integrated layer processing for all OSI layers (L2-L7) in a single pass from packet reception to transmission. The pipelines are free of context switches, system calls and interrupt processing, and they avoid significant socket, kernel, and device driver interrupt overheads.

Efficient Memory and Cache Management

Avi Service Engines use several advanced memory and cache management techniques to optimize data transfers and scale performance across cores. Techniques include zero-copy data transfers, memory mapped buffers, packed, and cache aligned data structures, lockless queues and rings, fixed sized pools, large TLB pages, per-core exclusive state, and more.

Harnessing the Power of x86

Avi Service Engines fully leverage the x86 AES-NI instruction set to provide scalable cryptographic performance on commodity x86 CPUs. Elliptic curve cryptography provides even further speedup: 4x the performance for the same cipher strength at the same CPU utilization.

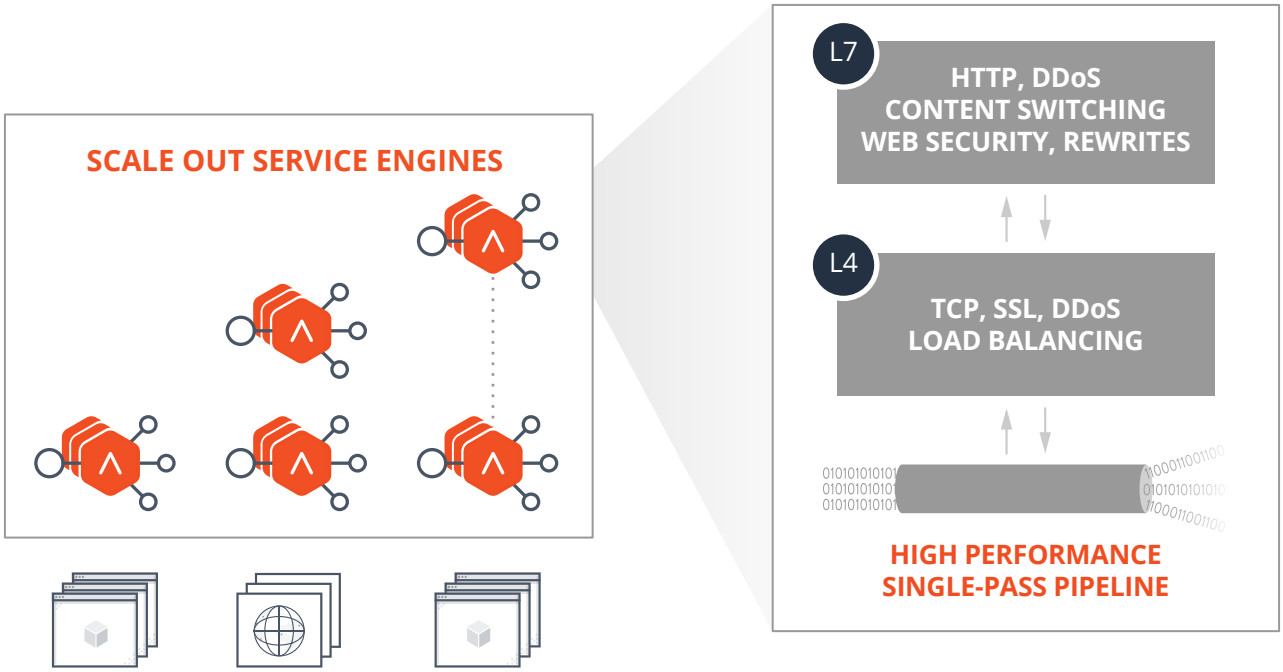

Figure 3: Avi Service Engines

Data Collection and Analysis Without Performance Impacts

Avi Service Engines examine terabits of traffic flowing through the data plane pipelines. High efficiency filters discard all but the most significant analytics data points and logs, significantly reducing the data by up to a factor of 1,000. Key metrics, analytics data points, and logs are then aggregated and streamed to the analytics engine on the Avi Controller using compression, variable length encoding, and a wire protocol in an efficient yet extensible format.

High Availability

Avi Service Engines can be configured in a variety of high availability options: Dedicated Active-Standby Pair, Shared N-Way Active-Active Service Engines, Shared N-Way Active-Standby and Shared Best-Effort. All existing connections are typically dropped in active-hot tandby failover with legacy appliances. On the other hand, existing connections on the running Service Engines in an active-active cluster are preserved on an Avi Service Engine failover. For example, a single Service Engine failure in an active-active cluster of four Service Engines results in disruption for just 25% of existing connections with the Avi data plane. See Figure 4.

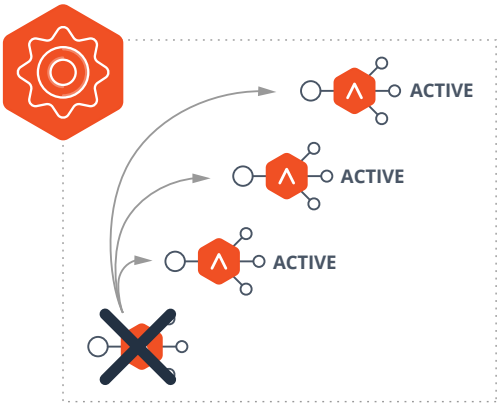

Figure 4: Avi Service Engines in an active-active cluster

Elastic Scale

All applications have periods of low, medium, and high load and performance requirements based on real-time user demand. The Avi Networks solution uniquely scales out the data plane performance in response to the application needs using several techniques. See Figure 5.

- **Automatic creation of Service Engines:** As more virtual services are created, the Avi Controller dynamically creates new Service Engines and places new virtual services on those newly created Service Engines.
- **Pooling resources to meet a spike in scale:** As the performance and load requirements of a single application varies, Avi Service Engines work together to gracefully add or remove Service Engines handling the load for a single virtual service.
- **Redistribution of load across Service Engines:** Virtual services can be gracefully migrated across Service Engines to redistribute load.

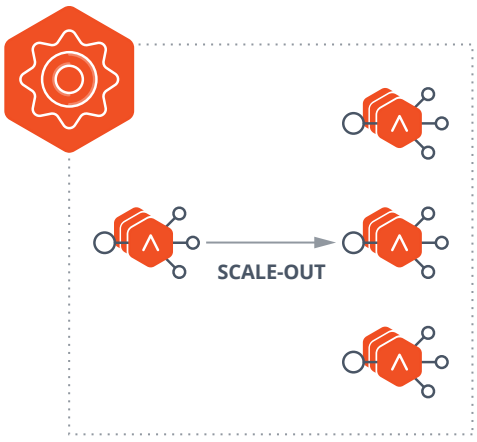

Figure 5: Avi Service Engines elastic scale

POLICY ENGINE (RESIDES WITHIN AVI CONTROLLER)

The Avi Controller cluster houses the policy engine and the API endpoint for configuring services. The Avi Controller cluster is architected as an active-active, scalable control plane cluster, unlike traditional active-standby appliance control planes. Avi Controller exposes a declarative policy model using first class REST objects to the administrator. Unlike traditional appliances providing REST APIs bolted on top of the CLI, all communication with the Avi Controller is over REST using native Avi policy objects. Avi UI and Avi CLI only use REST APIs to communicate with the Avi Controller. In an automated cloud environment, API agility and efficiency are critical for service availability and productivity.

INLINE ANALYTICS MODULE (INTEGRATED WITHIN AVI CONTROLLER)

The Avi Controller cluster also houses scalable, streaming, real-time analytics engines for Inline Analytics™. Avi analytics engines use various techniques to optimize the transport, storage, and analysis of streaming data points. Over 500 data points are collected and processed every second per service. See Figure 6.

- An append-only schema for low latency, high bandwidth data storage, and retrieval
- Sharded and replicated data storage backends for performance and high availability
- In memory and on disk reverse indexes for real-time log analytics
- Real-time quantile calculation to extract the most significant dimensions
- State efficient anomaly detection algorithms for deviation detection
- A weighted decision tree analysis for health score factor drilldowns
- Spatial and temporal correlation of anomalies for root cause analysis

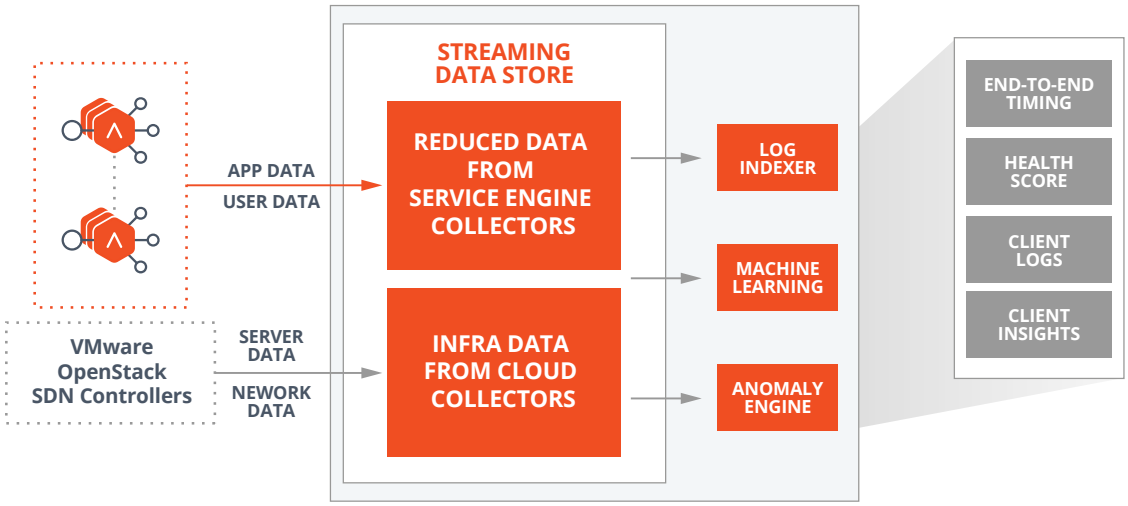

Figure 6: Avi Controller Inline Analytics

Inline Analytics processing subsystems provide comprehensive analytics dashboards about applications, end users, and the infrastructure:

- App Analytics with comprehensive end-to-end timing and customizable dashboard with key metrics time series for every virtual service
- A single App Health Score with drilldown, encompassing current and past application performance, infrastructure utilization, and anomalous behavior
- Log Analytics providing real-time search capability for application transactions with dimensional analytics for over 20 dimensions such as geo, OS, browser, etc.
- Client Insights providing dimensional analytics on Real User Metrics (W3C Navigation timing and Resource timing) such as page load time, DNS query, connection establishment, data transfer, etc.
- Anomaly detection for significant deviation from auto learned baselines and auto correlation of events, configuration changes, and anomalies across space and time

Using inline analytics and machine learning techniques, Avi's distributed data plane continuously and automatically adjusts the performance, placement, and capacity of application delivery services based on end-user and application insights derived by the Inline Analytics engine. Some examples of closed-loop application delivery are:

- Servers consume 40% power when idle and 85% at low CPU utilization according to this Facebook blog.
  - Avi's ServerSaver™ algorithm constantly monitors server response time and loads the fewest servers required at any given load while still providing superior levels of service.
  - Avi Vantage dynamically learns application load patterns based on time of day and day of the week, and it autoscales both Avi Service Engines and backend servers – without the need for guessing or preprovisioning capacity.

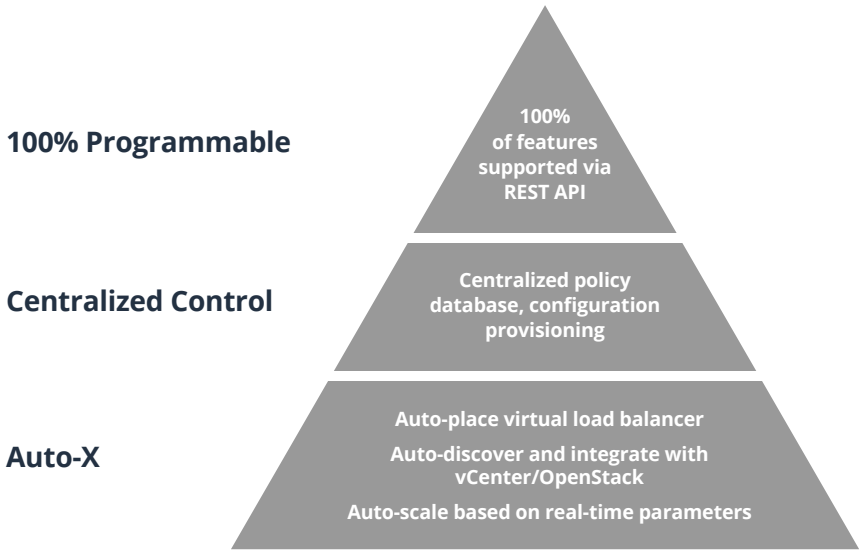

Figure 7: Simplified operations delivered by Avi Networks

CONCLUSION

Avi Networks aligns application delivery, security, and analytics with three technology megatrends impacting enterprise data centers: multi-cloud adoption, software-only and microservices-based application architectures.

Using a distributed architecture, Avi Vantage provides elastic and scalable application delivery services on commodity x86 servers across data centers and the cloud. It is purpose-built for the software-defined enterprise and uses a unique analytics-driven and software-defined architecture.

The analytics engine provides actionable insights to administrators for operational simplification; in addition, the performance, placement, and capacity of application delivery services are automatically adjusted in response to application load, user traffic, and infrastructure resources. See Figure 7 above.