ASSURING APPLICATION PERFOMANCE AT SCALE

ARISTA

VMTurbo and Arista Networks offer a unique approach to assuring application quality of service at scale

Executive Summary

Network-heavy, scale-out workloads place considerable stress on today's rapidly growing virtualized and cloud data centers. Network switches with 10, 40 or 100 GbE ports, an open and programmable network operating system and emerging software-defined networking (SDN) approaches aim to mitigate the potential bottlenecks that can precipitate at the network layer in such environments, as well as simplify network management.

Despite advances in both hardware and software, there remain significant challenges, as well as performance improvement opportunities. Specifically, incorporating network traffic and topology to VM placement decisions holds tremendous promise for increasing application performance and assuring quality of service. That is, with perfect knowledge and an optimal algorithm for placing VMs, how much faster might one or more scale out workloads run?

Seminal work conducted at Stanford University has demonstrated that by leveraging iterative placement algorithms with increasing degrees of network awareness to VM placement decisions can improve application performance – as measured by throughput or completion cycle – by as much as 70% compared to random placement.

While this work directionally supports the benefit of network consideration, it is limited to a confined set of system resources – CPU speeds, network topology, link capacities, and routing tables – and workload data – CPU utilization, traffic matrix. In practice, an ideal algorithm would account not only for these variables, but also those existing in memory and storage, eliminating potential bottlenecks at these layers that may negate the performance increases achieved above.

Arista and VMTurbo offer a joint solution that leverages Arista's Extensible Operating System (EOS) – including Latency Analyzer (LANZ™) and VM Tracer – and VMTurbo's Network Control Module and decision algorithms, to simultaneously consider compute, storage, and network utilization – including network traffic and topology information – in relation to real time demands of interdependent application workloads. Using microsecond output metrics from Arista and VMTurbo's Economic Scheduling Engine algorithm, the solution dynamically localizes frequently communicating workloads to maximize performance by minimizing latency at each possible contention point: compute, storage, and network. In addition, the solution enables data center operators to run as efficiently as possible.

This paper explores the benefits of incorporating network traffic and topology information in VM placement decisions, as well as the integration between Arista Networks and VMTurbo.

Network Challenges in the Software-Defined Data Center

Each virtualization wave has upended traditional best practices surrounding physical infrastructure. The mobility of workloads, brought forth by server and storage virtualization, taxes local and storage networks in ways that bare metal non-distributed application delivery simply did not. This new wave has presented an inevitable threat to performance.

Arista delivers a differentiated solution through offering a portfolio of 1/10/40 and 100GbE products that redefine network architectures, bring extensibility to networking, and dramatically change the price/performance of data center networks.

At the core of Arista's platform is the Extensible Operating System (EOS™), an open and programmable network operating system with single-image consistency across hardware platforms. EOS enables in-service upgrades and application extensibility and the ability to scale to tens of thousands of compute and storage nodes.

In addition, Arista provides a foundation for Software-Defined Networking (SDN) technologies such as VMware® NSX™ and Nuage Networks™ from which virtual networks can be provisioned, configured, and secured with the same ease as provisioning and configuring a virtual machine.

Despite these advances network-heavy, scale-out workloads that typify today's distributed, virtualized and cloud environments still present significant challenges for both virtualization and network architects. When designing topologies, architects must consider how to best achieve each of the following:

- Minimizing latency between workload endpoints
- Avoiding top-of-hierarchy network congestion
- Supporting "Tenant vMotion" from Private to Public Cloud resources
- Accelerating network access to distributed big data workloads
- Shaping traffic at ingress ports to minimize buffer congestion and packet loss

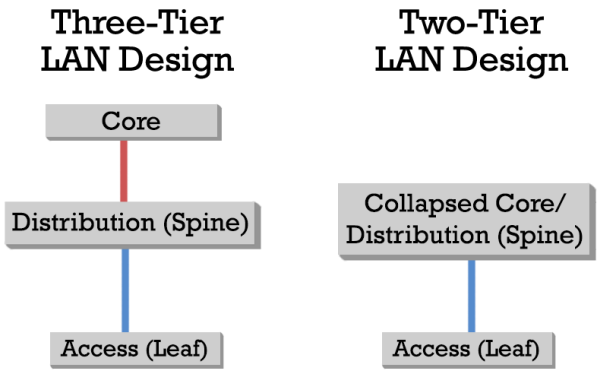
In truth, these problems are complex and related less to a priori network design and more to ad hoc network intelligence and how workloads utilize it. Specifically, how can network intelligence inform VM placement decisions such that latency is minimized and performance is amplified?

Increased East-West Network Traffic

Traditional 3-Tier networks – Core, Distribution, & Access – are suboptimal for today's distributed application design running on virtualized environments. Network traffic has shifted from predominantly North-South traffic (prevalent in the days of client/server applications) to mainly East-West traffic. This traffic is comprised of both data transfer

between interdependent VMs comprising multi-tier workloads, and virtual machine migrations (vMotions).

2-Tier network designs collapse the Core and Distribution Tiers into a single Spine Tier, yielding the 2-Tier Spine-Leaf configuration illustrated below. Furthermore, 2-Tier networks feature ultrafast Ethernet and switching at the Spine accommodating rates of 10 Gbps, 40Gbps or up to 100Gbps. Leaf devices are typically cheaper commodity devices with lower speeds.



On a 3-Tier Network, workload data must often travel all the way up the network topology, only to travel back down to its Leaf destination, introducing latency. Every additional switch "hop" also introduces potential packet loss. These performance threats explain the observed trend away from legacy 3-Tier Networks and toward more appropriate 2-Tier Networks.

Organizations reluctant to transition from legacy networks to 2-Tier designs are often justified by the high cost of re-architecting. Though the tradeoff between investing in re-architecting and maintaining a legacy network is one that must be determined by your organization, the reality of increased East-West traffic presents a real challenge for all.

Network Localization & Traffic Engineering

Even if the Spine in a 2-Tier Network design is built of fast non-blocking switches with enough capacity, delivering the "last mile" of bandwidth through Leaf switching may reveal unexpected bottlenecks. Furthermore, application performance issues often lie outside of the network domain.

Erickson et al. (2014) demonstrated that by introducing increasing degrees of network awareness to VM placement decisions, application performance – as measured by throughput or completion cycle – could improve by as much as 70% compared to random

placement.[1] Though seemingly intuitive, this work proves that incorporating network considerations into VM placement decisions *is* deserving of attention.

Consider a typical 3-tier application, the cornerstone of many Web services. This workload consists of three distinct components, each residing on its own VM:

1. Load-balanced Web server
2. Application server
3. Database

The Web server receives requests from end users, invoking some logic within the application server, which in order to fulfill the incoming requests performs a query on the database. The output of the query is pushed back through the application server where additional logic may be added before sending information back to the Web server. The Web server then, presents the information back to the end-user. The amount of data exchanged between tiers depends on the nature of the application and the end-user request, which can be very taxing at peak times.

Our 3-tier Web application example, while common, is very simple. In practice, distributed workloads can consist of dozens of tiers, and scaled horizontally, hundreds or even thousands of VMs working in unison.

Commonly used workload distribution tools such as VMware® DRS and Citrix® XenServer® Workload Balancer aim to evenly balance workloads across compute and storage resources. However, they lack awareness of network traffic and often push workloads apart, across the network, which introduces latency. To mitigate this latency, architects employ common network localization tactics: dedicated clustering and affinity rules.

Dedicated Clustering

Dedicated clustering is the practice of confining application tiers – most often databases – to a dedicated, low-density, high-performance cluster. Consider the following situation. In order to guarantee database performance, administrators create separate database clusters where a small number of high performing database VMs run on powerful servers and fast storage devices. Since the database VMs are isolated, performance is stellar and protected from interference. The same could be done with application servers, and then a special cluster could run load-balanced web servers whose numbers can scale appropriately with demand.

This option bears a significant flaw. When demand peaks, and a large number of clustered Web server VMs begin sending a lot of traffic to an app server, their immediately-connected network device can become saturated. Additionally, the app server queries pass through

[1] Erickson, D., Heller, B., McKeown, N., and Rosenblum, M. Using Network Knowledge to Improve Workload Performance in Virtualized Data Centers. In *IC2E*, Boston, March 2014.

the network device on the database cluster, which also becomes saturated. Although the Spine has plenty of capacity, the application will experience high latencies as slow Leaf switches struggle to accommodate the traffic.

Affinity Rules

Affinity rules are settings that establish a relationship between two or more VMs and hosts. In the 3-tier application example, we could define a simple affinity rule which requires each application tier VM – web, application, database – to always reside on the same host. This strategy would leverage virtual switches (vSwitches) to constrain communication between each application tier to the host itself.

If frequently-talking VMs run on the same host, their packets will never cross the host boundary and will avoid going to LAN switches at all. While this solution seems convenient, it also bears inevitable pitfalls – especially at scale.

First, this approach requires a priori knowledge of VM communication pathways. With perfect information, the administrator may define and build an entire inventory of affinity rules across the environment. However, large organizations run hundreds of applications.

Furthermore, VM communication behavior is transient. VM-A may communicate with VM-B heavily for some duration, but then switch from VM-B to VM-C or VM-B to VM-E and VM-F for an extended period after that. Even an affinity rule that groups these machines together creates unnecessary constraints on the data center, segmenting it and detracting from the dynamic capabilities virtualization offers in the first place.

Is manual affinity definition truly scalable? The second pitfall of affinity rules yields a simple answer: Not really. Isolating VMs in this fashion can work, permitting that demand on the workload does not overwhelm the assigned compute, storage, and internal network. As soon as demand on the workload peaks, the compute and storage infrastructure forces you to separate each tier of the load as far as possible from one other onto separate devices.

Once the tiers are separated, the potential for network latency is reintroduced. As is evident, static affinity rules seem to be a solution, but cannot truly scale for performance. Conversely, dedicated clustering accepts that under peak times, latency is inevitable.

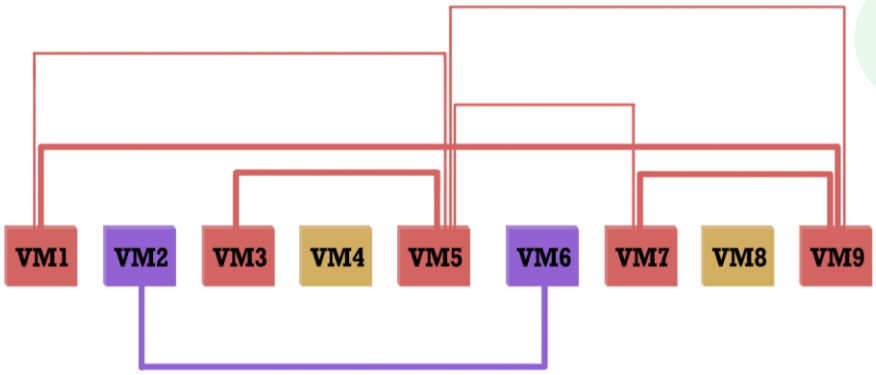
A Tradeoff

These two tactics represent opposing strategies, neither of which is resilient under peak demand conditions. Dedicated clustering avers that it is best to provide application tiers with lush compute and storage resources, at the risk of overloading the network. Affinity rules avoid traversing the network at the risk of overwhelming local compute and storage.

Which is better – latency due to CPU ready queues and storage I/O, or that due to congested Leaf switches? Neither. The goal is to minimize all of it. The challenge is that doing so in an

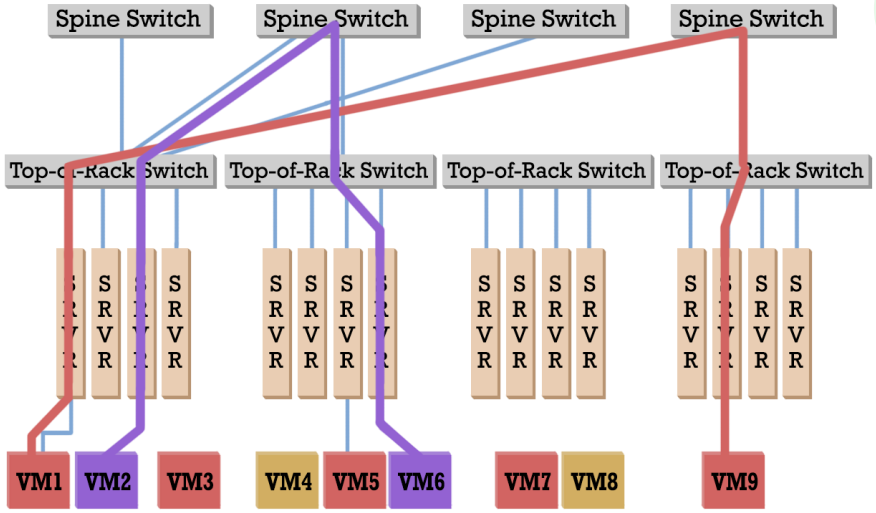
environment of hundreds of applications – each with dozens of tiers, distributed across thousands of VMs sharing hundreds of hosts, datastores, and network devices – is a very difficult problem to solve. When one considers the unpredictability of demand, the difficulty is amplified. Is there a solution?

The Challenge Illustrated



1

Communication pathways between VMs are common in today's scale-out distributed workloads.



2

These interdependent VMs are often placed far across the network, creating latency. During peaks, dedicated clustering of like tiers can overwhelm Leaf switches. Affinity Rules designed to keep workloads close together can overwhelm local compute and storage resources.

Controlling the Tradeoffs among Network, Compute, & Storage

There is no static solution that can effectively control this tradeoff. Heeding the work of Erickson, as well as the challenges described herein, it is evident that application performance can be improved and latency minimized if and only if VM placement decisions are made with complete real-time knowledge of:

1. Workload demand
2. Network traffic-matrix of application endpoints (i.e. sFlow or NetFlow)

3. Resource capacity – compute, storage and network

Economic Scheduling Engine

VMTurbo's patented algorithm abstracts the data center and all its networked entities into a common data model of resources. This Economic Scheduling Engine maps the end-to-end relationships between discrete resources in the IT stack with a holistic understanding of the supply chain from physical resource supply to end user service delivery. Using the common data model and a pricing function based on utilization entities make trades in real time – make decision on where to consume resources and when to supply more or less of a resource – resulting in automated actions that continually assure performance while maximizing efficiency.

The decisions are made with the full understanding of the entire stack and everything in the data center, e.g., hosts, data stores, VMs, applications, containers, zones, etc., is a buyer and a seller. The commodities traded are compute resources, such as Memory, CPU, IO, Ready Queues, IOPS, Latency, Transactions per Second, etc.

For example, a host sells Memory, CPU, IO, Network, CPU Ready Queues, Ballooning, Swapping, etc. A Data Store sells IOPS, Latency, Storage Amount. A VM buys these resources and sells VMem, VCPU, VStorage, etc. An Application buys these resources and sells Transactions per Second.

Minimizing Latency with vPods and dPods

VMTurbo has introduced two new entities into its algorithm to control the complex tradeoff of workload chattiness vs. sufficient access to compute/storage resources. Leveraging Flow Collector output – sFlow or NetFlow, VMTurbo discovers the network traffic-matrix and dynamically defines each group of "chatty" communicating VMs as an entity called a *vPod*.

In the previously illustrated traffic matrices, each grouping of communicating VMs would be grouped as a vPod. VMs 1, 3, 5, 7, and 9 would be one vPod, and VMs 2 and 6 would be another vPod. VM 4 and VM 8 are not a vPod, as they are not communicating at the moment.

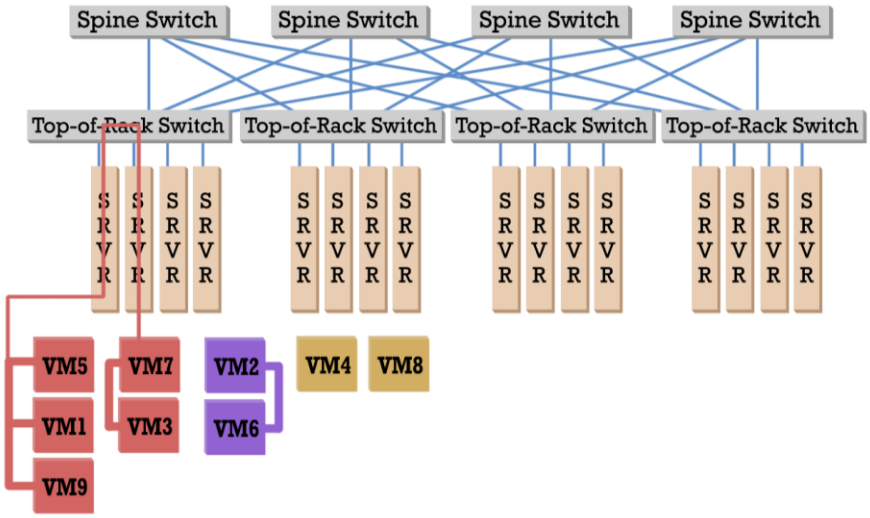
The construct of vPod eliminates the need to inventory static affinity rules, and provides maximum flexibility of VM migrations when communication between vPod members is low. When demand on a vPod is high, the vPod migrates as a unit, consuming resources from an entity called a *dPod*.

A dPod is a set of resource providers located close together (physically) on the network – i.e. a group of hosts and storage residing under the same Top of Rack switch.

Using topological probabilities, VMTurbo defines four levels of *flow*, each of which is increasingly more expensive than that before it within the Market:

Flow Level	Flow Description	Relative Cost
Level 0	Intrahost Flow	$
Level 1	Intra-dPod Flow (Host to Host Migration)	$$
Level 2	Cross-dPod Flow (Cluster to Cluster Migration)	$$$
Level 3	Cross-Cloud Flow (Private to Public Migration)	$$$$

By pricing higher level flows more expensively, the Economic Scheduling Engine forces VMs or Containers to continuously make placement decisions that minimize the cost of flow – i.e. stay on the same host. However at the same time these entities need to satisfy their workload demand for other resources (e.g. CPU, Memory, IOPS, Storage).



Dynamically defined vPods self-manage the tradeoff between network, compute, and storage – migrating themselves to the most economic dPod which will simultaneously maximize workload performance and resource utilization.

When communication between vPod members subsides, the vPod disaggregates until demand drives it back together. By dynamically localizing workload flows, the desired tradeoff is always attained, where application performance is assured while physical resources are consumed as efficiently as possible. VMTurbo refers to this scenario as the Desired State: a continually shifting state in which application performance is assured while the physical infrastructure is utilized as efficiently as possible. The Desired State can only be achieved through real time analysis and corresponding action.

Arista LANZ™ Integration

LANZ provides real time, microsecond tracking of congestion and latency in the network and provides statistics and metrics on the state of buffer queues across the system. VMTurbo will leverage LANZ data to refine the Desired State of the network and drive the actions necessary to attain and control it.

Integrating LANZ metrics simply requires the addition of a new commodity, representing real time network buffer state, to the VMTurbo Market. All entities generating demand for buffer (VMs, containers, etc.) must buy that commodity in order to access the switch, influencing VMTurbo's Economic Scheduling Engine to consider the tradeoffs between the network state and the state of other components in the data center (such as compute and storage).

VMTurbo's Economic Scheduling Engine will recommend VM placement changes based on the above tradeoff. For example, if a VM is communicating with a different VM on a different host through a highly utilized buffer, the price of that buffer will increase and VMTurbo will seek an alternative communication pathway requiring less utilized buffers, while considering the price of all other necessary resources. Under such conditions, VMTurbo would enact some combination of VM migrations to drive the environment to its Desired State.

Arista VM Tracer Integration

VM Tracer provides visibility into the entire network topology through a single pane of glass. VMTurbo leverages VM Traer information to refine dPod discovery and accurately map the relationships between compute, storage and Arista switches. VM Tracer topological insight provides perfect visibility, yielding more accurate VM placement recommendations that account for both the network traffic matrix (Flow) and Arista network topology.

Summary

Network-heavy, scale-out workloads typical in today's virtual and cloud environments have driven a rapid increase in East-West network traffic. Static management tactics such as dedicated clustering and affinity rules force organizations to choose between network latency and compute/storage latency, a tradeoff which cannot be controlled statically.

VMTurbo and Arista offer a solution that continually controls the tradeoffs between network, compute, and storage, driving VM placement decisions that will assure application quality of service while maximizing utilization of the underlying infrastructure.

About VMTurbo

VMTurbo's Demand-Driven Control platform enables organizations to manage cloud and enterprise virtualization environments to assure application performance while maximizing resource utilization. VMTurbo's patented decision-engine technology dynamically analyzes demand from applications, containers, network and VDI and adjusts configuration, resource allocation and workload placement to meet service levels and business goals. With this unique understanding into the dynamic interaction of demand and supply, VMTurbo is the only technology capable of closing the loop in IT operation by automating the decision-making process to maintain an environment in a healthy state.

The VMTurbo platform first launched in August 2010 and since that time more than 40,000 users worldwide have deployed the platform, including JP Morgan Chase, Colgate-Palmolive and Salesforce.com. Using VMTurbo, our customers ensure that applications get the resources they need to operate reliably, while utilizing their most valuable infrastructure and human resources most efficiently.

About Arista Networks

Arista Networks was founded to deliver software driven cloud networking solutions for large data center and computing environments. Arista offers a broad portfolio of Gigabit Ethernet solutions including 1/10/40 and 100GbE switches that redefine network architectures, bring extensibility to networking and dramatically change the price/performance of data center networks. At the core of Arista's platform is EOS (Extensible Operating System), an advanced network operating system, designed to build software driven cloud networks. EOS provides a single image consistency across hardware platforms and a modern core architecture enabling in-service upgrades and application extensibility