Over the last decade, architects in the world’s largest data centers have built out hyperscale networks that offer orders of magnitude improvement in operating, performance and cost metrics. With the introduction of the Big Cloud Fabric, we bring hyperscale networking to a broader community.

DATA CENTER NETWORKING CHALLENGES

Applications are the engine for modern businesses—driving innovation, operational efficiency and revenue generation. They demand an infrastructure that is highly agile and is easy to manage, while reducing overall costs. Typically residing in the data centers, these applications, which include mission critical ERP systems, multi-tier Web applications, Big Data, etc. have placed new constraints on the networking infrastructure; support for high east-west traffic bandwidth, virtual machine (VM) mobility, multi-tenancy, app elasticity to list a few. Additionally, many such applications often consist of both physical and virtualized workloads, adding to the challenges in managing already complex data center networks.

As infrastructure teams have struggled to respond to those requirements, they have increasingly found that “networks are in their way”. Unlike the rest of the portfolio they manage, legacy networks remain highly static and require extensive manual intervention and operational overhead. While the speed, scale and density of equipment offered by traditional networking vendors has increased over the last two decades, the underlying architectures and business operating models have stayed fundamentally unchanged.

The early SDN solutions (overlay-underlay approaches) that tried to address these challenges have, in turn, resulted in additional costs and complexity. With limited choice or a migration path to next generation solutions, customers are stuck with fragile, inflexible and expensive networks that do not address their application-centric requirements.
HYPERSCALE DATA CENTER APPROACH TO NETWORK DESIGN

Over the last decade, hyperscale data center architects have consistently built out app-centric network designs that offer orders of magnitude improvement in operating and cost metrics compared to traditional approaches. Despite their scale, they are able to take advantage of new networking technologies and cost/performance improvements faster. They have been quietly outpacing networking vendors in innovation while the rest of the industry has been stuck in a legacy design paradigm.

Hyperscale networking is a design philosophy, not a cookie-cutter formula. There are attributes of this design philosophy that are very attractive to any data center architect:

- Significant reduction in CapEx (with bare metal switches)
- Quantum leap in automation and operational simplification (with software defined networking)
- Rigorous approach to resiliency and high availability within the network infrastructure, and
- Focus on replicable building blocks known as “pods”

Many architects consider these attributes in isolation rather than embracing them collectively to implement hyperscale network design in totality. When implemented singularly, we see more failures than successes. It is the holistic approach of embracing all the design principles in parallel that delivers the desired results.

**At Big Switch Networks, our mission is to bring hyperscale data center networking to all users.** The rest of this document outlines key hyperscale data center design principles, followed by a discussion of Big Switch’s [Big Cloud™ Fabric (BCF)](http://www.google.com/about/datacenters/gallery/#/all) solution that is architected based on these design principles for a broader user community.

*Figure 1: Google Data Centers*[^1]

[^1]: [http://www.google.com/about/datacenters/gallery/#/all](http://www.google.com/about/datacenters/gallery/#/all)
HYPERSCALE DESIGN PRINCIPLE #1:  
BARE METAL HARDWARE TO REDUCE COST

The term “bare metal” simply refers to hardware that is procured independent of the operating system and application software that runs on it. Unlike vertically integrated and proprietary mainframe computers, modern-day compute market has been fully disaggregated for decades. Customers have had the choice to buy bare metal server hardware from branded vendors (Dell, HP, Lenovo etc.) or white-box vendors (Quanta Computers, Super Micro etc.). Operating system and software applications are procured separately from software vendors, and installed on bare metal hardware. This disaggregation has led to tremendous innovation across all layers of the stack—CPU, HW, OS and Applications.

In contrast, the networking industry has been stuck in the architectural design of the 1990s. Networking software, hardware and ASICs have traditionally been provided by vertically integrated vendors resulting in high prices and an extremely slow pace of innovation. Over the last decade, things have finally started to change, with “merchant silicon” companies like Broadcom, Marvell and Intel (Fulcrum) leading the effort towards a disaggregation of networking hardware and ASICs. Merchant-silicon based Ethernet switches have been steadily gaining market share over fully proprietary solutions.

Additionally, thanks to the vision and execution by several hyperscale data center operators, this disaggregation is also being extended across networking hardware and software. Hyperscale network infrastructure teams pioneered this trend by purchasing bare metal switching hardware directly from Original Device Manufacturers (ODMs). Combining this with their own increasingly sophisticated software stacks (developed by in-house R&D teams), they have cut CapEx costs dramatically.

More recently, with leadership and participation from the hyperscale pioneers, initiatives such as the Open Compute Networking Project\(^2\) and Open Network Linux\(^3\) (ONL) are expanding the bare metal switching reach to broader audiences. A mature bare metal networking supply chain is emerging to support this industry trend. Starting with the top-of-rack (ToR) and Spine switches, any data center operator can now procure high-end switch hardware from a range of sources. This networking HW/SW disaggregation enables tremendous flexibility and vendor choice, increases competition, dramatically reduces CapEx and most importantly drives rapid innovation—as each layer of the stack is able to innovate independently of other layers.

\(^3\) [http://opennetlinux.org/](http://opennetlinux.org/)
HYPERSCALE DESIGN PRINCIPLE #2: SOFTWARE DEFINED NETWORKING TO REDUCE COMPLEXITY

Embracing bare metal hardware begs the question “what about software?” The second key principle incorporated by hyperscale networking architects centers around the concept of Software Defined Networking (SDN). SDN refers to a separation of the network’s data and control plane, followed by a logical centralization of the control plane functionality. In practice, it implies that network’s policy plane, management plane and much of control plane is externalized from the hardware device itself, using an SDN controller, with few on-device off-load functions for scale and resiliency. The network state is logically centralized but hierarchically implemented, instead of being fully distributed on a box-by-box basis across access and aggregation switches.

Controller-based designs not only bring agility via centralized programmability and automation, but they also streamline fabric designs (e.g. leaf-spine L2/L3 Clos) that are otherwise cumbersome to implement and fragile to operate in a box-by-box design mind-set.

**Figure 3: Hierarchical SDN Architecture**

SDN Controller
- Proactive Connectivity and Policy Enforcement
- Single Pane of Glass Management
- Fabric Programmability and Analytics

Thin SDN OS
- Reduced Complexity with Lightweight Codebase
- Rapid Upgrades
- Hierarchical Intelligence for Scale-out & Resilient Operations

HYPERSCALE DESIGN PRINCIPLE #3: CORE-AND-POD DESIGNS TO INNOVATE FASTER

**TRADITIONAL DATA CENTER DESIGN**

Traditional datacenters have been optimized for North-South traffic and built using a tree-shaped network topology of core, aggregation and access switching/routing layers. Spanning tree protocols running inside the L2 switching domains create a loop-free tree topology, with multi-chassis link aggregation or other proprietary solutions (e.g. TRILL, Fabric Path) to improve upon spanning tree protocols’ various shortcomings. In addition to challenges with L2 protocols, this traditional approach to data center design presents other problems:

1. **Choke points for east-west traffic:** By some estimates over 70% of the data center traffic is between server nodes within the data center. Tree-shaped network topologies are well designed for north-south (client-server) traffic patterns, but impose choke points on east-west (server-server) traffic that increasingly dominates the data center.
2. **Not optimized for operational efficiency.** These designs are not optimized for operational requirements of scale-out data centers such as adding and removing infrastructure, manage failures, zero-impact upgrades etc.

3. **Risky to automate:** A failure in a single zone using traditional network design can have dramatic implications across adjacent zones and, potentially across the whole data center.

4. **Choke points at L2/L3 boundary:** In many of the traditional designs, L2/L3 boundary is at the aggregation layer and it becomes a choke point in terms of bandwidth, CPU utilization and ASIC table sizes—typically demanding forklift upgrade with bigger, more expensive boxes.

5. **Hard to innovate:** These designs make it difficult for customers to innovate as the only path available to them is buying newer, higher priced boxes from the same networking vendor. Customers typically evaluate a range of architectures available from various vendors and are forced into making a data center design choice at inception. Once that architectural choice is made they are completely locked-in with the vendor, and any change requires a complete rip and replace of the infrastructure.

While this classic data center design may work in highly oversubscribed deployments, modern cloud network architects demand high-performance, lower-cost alternatives in their data center with evolvable networking designs.

---

**Figure 4: Hyper-scale Data Center Scale-out Approach**

**Optimized for:**
- Multi-vendor data center networks
- Atomic units of automation
- Rapid adoption of faster/better/cheaper technologies

---

**CORE-AND-POD DATA CENTER DESIGN**

In contrast to a traditional core/aggregation/access layered design, a Core-and-Pod architecture consists of a routed Core connecting a number of independently designed Pods. A Pod is defined as the atomic unit within a larger data center that is typically designed and deployed all at once (network/compute/storage), and has a clear demark point (L3 interface to core). A pod may be as small as a few racks or as large as a few hundred, though in most non-hyperscale deployments a pod often corresponds to a practical 8-16 racks (approximately 16-32 physical switches and 5K-15k VMs).

Automation at the pod level is typically far simpler than at the data center level where capacity and design changes are more frequent. Adoption of new technologies and price/performance improvements in the form of new pod designs are independent of capacity planning, where new instances of a pod design are simply attached to the data center core with basic routing protocols or static routes.

The result, a desired separation of the innovation cycle from capacity planning decisions, is key to hyperscale data center designs but is also very accessible to a larger audience of network architects looking to attach a first modern pod design to a legacy core.
BIG SWITCH NETWORKS APPROACH—A BARE METAL SDN FABRIC

Embracing the hyperscale data center design principles, Big Switch’s Big Cloud Fabric (BCF) solution enables rapid innovation, easy provisioning and management, while reducing overall costs. By featuring SDN software and bare metal Ethernet switch hardware in pod designs, the Big Cloud Fabric delivers multi-hypervisor based hyperscale networking to a broad class of users. The BCF solution is available in two editions:

- **P-Clos**—consisting of leaf-spine physical Clos fabric controlled via SDN
- **Unified P+V Clos**—consisting of leaf-spine plus virtual switches (vSwitches) controlled by SDN.

The BCF solution components include:

- **The Big Cloud Fabric Controller**—a logically centralized and hierarchically implemented SDN controller implemented as a cluster of virtual machines or hardware appliances for high availability (HA)
- **Bare Metal Leaf and Spine Switch Hardware**—a variety of switch HW configurations (10G/40G) and vendors are available on the Big Switch hardware compatibility list
- **Switch Light™ Operating System**—a bare metal switch OS purpose built for SDN
- **Switch Light™ vSwitch (optional)**—a high performance SDN vSwitch for Unified P+V Clos designs

Network provisioning as well as most troubleshooting and automation is done via CLI, GUI, REST APIs and software integrations on the BCF Controller. An application/tenant-centric provisioning streamlines L2/L3 configuration, while the single-pane-of-glass BCF controller reduces the number of management consoles by a ratio of over 30:1. APIs and orchestrations system integrations (with OpenStack and CloudStack) integrate network provisioning, security and audit workflows, and the use of bare metal Ethernet switch hardware reduces 3-year costs by over 50%.

*Figure 5: Sample 16-Rack Big Cloud Fabric (BCF) Topology*

---

6 1G servers can be connected via appropriate choice of cables and/or optics
5 Relative to a 16 rack row of traditional L2/L3 switches (32 ToRs, 2 Aggregation Chassis)
6 Relative to CapEx + Maintenance of hardware, software, cables and optics of a 16 rack row using popular incumbent L2/L3 networking devices. Model available on request.
THE BIG CLOUD FABRIC ARCHITECTURE

The BCF architecture consists of a physical switching fabric, which is based on a leaf-spine Clos architecture. Bare metal Leaf and Spine switches running Switch Light™ Operating System form the individual nodes of this physical fabric. With the (optional) Switch Light vSwitch, the fabric configuration and control can be extended to the vSwitches resident in server hypervisors. The control intelligence is hierarchically placed: most of it in the Big Cloud Fabric Controller (where configuration, automation and troubleshooting occurs), and some of it off-loaded to Switch Light for resiliency and scale-out.

Some of the key benefits from this architectural approach include:

APPLICATION / TENANT CENTRIC CONFIGURATION
STREAMLINES L2/L3 CONFIGURATION

Traditional network configuration requires considerable time from engineers and architects translating a high level logical design in to a box-by-box configuration. The result is that network segments are generally slow to provision, error prone and difficult to audit. In the Big Cloud Fabric design, configuration in the CLI, GUI or REST API is based on the concept of tenants, logical segments and logical routers. Each tenant has administrative control over a logical L2/L3/policy design that connects the edge ports under the tenant’s control. The Big Cloud Fabric controller has the intelligence to optimally translate the logical network design into forwarding tables of the leaf/spine switch ASICs and vSwitch memory, freeing network engineers and architects from the tedious and error-prone process of translating a high level logical design to low level box-by-box configuration.

LOGICALLY CENTRALIZED CONTROLLER REDUCES MANAGEMENT CONSOLES BY OVER 30:1

With all configuration, all automation and most troubleshooting done via the Big Cloud Fabric controller as opposed to box-by-box, the number of management consoles involved in provisioning new physical capacity or new logical applications goes down dramatically. In a traditional network design example of a 16 rack cloud pod with dual access switches and two aggregation switches, a total of 34 management consoles are required to manage the pod. The Big Cloud Fabric design has only one—the controller console—that performs the same functions. The result is massive time savings, reduced error rates and dramatically simpler automation designs. As a powerful management tool, the controller console exposes a web-based GUI, a traditional networking-style CLI and REST APIs.
APIS FOR INTEGRATED NETWORK PROVISIONING, SECURITY AND AUDIT WORKFLOWS

Various REST APIs on the controller are available to integrate workflows around provisioning new applications’ logical L2/L3 designs, ensuring that they comply with security policies and then auditing connectivity after the fact. Initial integrations with OpenStack (commercially supported and available as open source) serve as a case study in how these APIs are intended to be used. By integrating network policy configuration with the OpenStack Heat templating system, a network security team needs to only review and approve new network templates rather than every new application. This integrated workflow can increase consistency of security policy while shaving off a significant number of calendar days and weeks in application deployment. A second set of Big Switch APIs allows tenants and security teams to audit multi-node connectivity after the fact, setting up automated network self-tests that check the connectivity (or lack thereof) between various VMs, bare metal servers and services in an application deployment. For example, when a 3-tier application is deployed, this multi-node connectivity check across the fabric can be used to audit network policy and physical connectivity to ensure that web tier VMs can connect to app tier VMs but not database VMs on a regular basis and alert if any changes occur.

BARE METAL SWITCH HARDWARE REDUCES CAPEX BY OVER 50%7

By using bare metal switch hardware as opposed to traditional networking vendor custom hardware, hard costs can be dramatically reduced. In order to look at “all in” hard costs on a three year basis, The chart on the right shows a comparison of three architectures for a 16 rack, heavily virtualized IaaS pod. By adding up hardware, software licenses, maintenance agreements and optics/cables, a complete picture of the “all in” hard costs on a three year basis shows that the cost savings of the bare metal SDN approach are dramatic.

OTHER BENEFITS

The Big Cloud Fabric design offers numerous other benefits beyond the ones described earlier, including:

• **Scale-out design flexibility** that comes from leaf/spine design using simple switch hardware as opposed to an edge-aggregation design that relies on fragile enhancements to the spanning tree protocols and chassis switch hardware.
• **High bisection bandwidth utilization** enabled by the multi-path design of the Clos fabric.
• **DC-grade resilience** that allows the fabric to operate in the face of link or node failures as well as in the rare situation when both controllers are unavailable (headless mode).
• **Multi-hypervisor support**, for both “P-Clos” design (leaf-spine SDN fabric) and a “Unified P+V Clos” design (vSwitch-leaf-spine SDN fabric)
• **Zero touch fabric management**, including auto-installation, auto-configuration and auto-upgrade of Switch Light OS.

7 Model available on request.
USING BIG CLOUD FABRIC: A 3-TIER APPLICATION EXAMPLE

While it is important for data center architects to understand the Fabric internals, most users of the data center expect to deal with a much simplified logical view of the networking infrastructure. The Big Cloud Fabric supports such a multi-tenant model, which is easily customizable for the specific requirements of different organizations and applications. This model increases the speed of application provisioning, simplifies configuration, and helps with analytics and troubleshooting.

Some of the important terminology used to describe the functionality includes:

- **Tenant**—A logical grouping of L2 and/or L3 networks and services.
- **Logical Segment**—An L2 network consisting of logical ports and end-points. This defines the default broadcast domain boundary.
- **Logical Router**—A tenant router provides routing and policy enforcement services for inter-segment, inter-tenant, and external networks.
- **External Core Router**—A physical router that provides connectivity between Pods within a data center and to the Internet.
- **Tenant Services**—Services available to tenants and deployed as dedicated or shared services (individually or as part of a service chain).

TENANT WORKFLOW

In the most common scenario, end consumers or tenants of the data center infrastructure deal with a logical network topology that defines the connectivity and policy requirements of applications. As an illustrative example, the canonical 3-tier application in Figure 9, shows various workload nodes of a tenant named “BLUE”. Typically, a tenant provisions these workloads using orchestration software such as OpenStack, CloudStack etc or BCF Controller GUI/CLI directly. As part of that provisioning workflow, the Big Cloud Fabric Controller seamlessly handles enabling the logical topology onto the physical switches and vSwitches.

Mapping Logical to Physical

As shown in Figure 9, the Blue Tenant has three logical network segments. Each of the three segments represent the broadcast domain for the 3-tiers - Web, App and Database. Note that in the above example, Web1,2 and App1,2 are virtualized workloads but DB1,2 are physical endpoints. Following the rules defined by the data center administrator, the orchestration system provisions requested workloads across different physical nodes within the data center. As an example, the logical topology shown in Figure 9 could be mapped on the pod network as shown in Figure 10.

The task of providing optimal connectivity between these loads dispersed across the pod, while ensuring tenant separation and security is handled by the Big Cloud Fabric Controller. In a Unified P+V deployment mode, Big Cloud Fabric makes no distinction between the two types of workloads and simply programs the boundary nodes of the fabric (leaf switches for physical workloads and hypervisor vSwitches for virtualized workloads).
In order to simplify the example, we only show racks that host virtualized and physical workloads in the figure below, but similar concepts apply for implementing tenant connectivity to external router and chaining shared services. An illustrative sample set of entries in various forwarding tables highlight some of the salient features of the Big Cloud Fabric described in earlier sections.

- L3 routing decision is made at the first hop virtual switch or Leaf switch (no hairpinning)
- L2 forwarding across the Pod without special fabric encapsulation (no tunneling)
- Full load-balancing across the various LAG links (vSwitch, Leaf and Spines)
- Full mesh connectivity within the physical fabric for resilience

**Figure 10: Mapping Logical to Physical Topology**
BIG CLOUD FABRIC INTERNAL SOFTWARE/HARDWARE ARCHITECTURE

The Big Cloud Fabric Controller implements the key principle of SDN by externalizing much of the management and control plane functionality from fabric elements into a cluster of SW nodes running on x86 servers. Working in conjunction with the thin Switch Light software running on individual physical and virtual switch nodes (including several control-plane off-load functions), the Big Cloud Controller ensures that the application intent (connectivity rules, policy enforcement, service insertion etc.) is quickly and optimally distributed to the relevant BCF nodes. In effect, this means that the:

- Global view of the fabric topology is maintained in the Pod level Logical Router within the controller and distributed in appropriate chunks to the Leaf, Spine or Hypervisor Logical Routers.
- Switch Light software running independently on each of the Fabric nodes, in turn ensures that any localized network state update (e.g. ARP, DHCP etc. requests) or exception conditions (e.g. link failure) are quickly addressed locally whenever possible and then reported back to the controller for remediation and broader propagation.

This hierarchical approach to handling management and control plane functionality is based on the experience from hyperscale designs where it was obvious that apart from a few very high-frequency functions such as adjacency management or fast reroute, most of the network requirements in a data center are driven as a part of the workflow being managed by the Orchestration system (e.g. OpenStack, CloudStack etc.) or custom OSS/BSS systems.
Such an approach provides the best of both worlds—eliminating a layer of complexity introduced by distributed state propagation protocols (e.g. STP, ISIS etc.) flooding the data center segments, while ensuring the network can react rapidly to certain local triggers, continue operations even in the rare situations when the dual-controllers become unreachable (headless resilient mode).

To get a better understanding of the Big Cloud Fabric architecture, described below is a high level sample control flow of the Big Cloud Fabric:

1. When powered-on, each fabric node (e.g. leaf/spine switch) boot-ups via the ONIE boot-loader which then initiates the Switch Light software installation process. Typically Switch Light software images are downloaded from the Controller using the Zero Touch Fabric (ZTF) capability. Each leaf/spine switch then boots up with valid Switch Light image. The BCF Controller also provisions the fabric node.

2. As individual fabric nodes come online, the Big Cloud Fabric Controller builds a global L2/L3/policy table with information collected from the orchestration system as well as that reported through the Switch Light software on the individual vSwitches, leaf and spine switches participating in the fabric.

3. The switch software syncs locally relevant state with the Controller, and the forwarding tables are downloaded and installed in to the switch ASICs or forwarding tables.

4. When a new end point (e.g. VM or physical server) connects to the fabric for the first time on one of its external ports (physical or virtual depending on the software edition), the first hop switch sends its L2/L3 information to the Controller. Upon learning a new L2/L3 address, the Controller compiles an optimal update that is then pushed to all the fabric nodes in the network.

5. When tenant logical configuration is changed, the Controller compiles those changes to edits required to the various L2/L3/policy tables in the vSwitch, leaf and spine switches in the fabric, and synchronizes changes to ensuring no impact to other tenants.

**BIG CLOUD FABRIC: ADDRESSING REQUIREMENTS FOR A RANGE OF DC WORKLOADS**

Big Cloud Fabric is designed from the ground up to satisfy the requirements of physical, virtual or combination of physical and virtual workloads. Some of the typical Pod deployment scenarios include:

- Private/Public clouds
- OpenStack Pods using Nova or Neutron Networking
- CloudStack / CloudPlatform Pods
- High Performance Computing / Big Data Pods
- Virtual Desktop Infrastructure (VDI) Pods
<table>
<thead>
<tr>
<th>FEATURE</th>
<th>DESCRIPTION / BENEFIT</th>
</tr>
</thead>
</table>
| Zero Touch Fabric (ZTF)               | ZTF enables complete control of physical switches within BCF without manually interacting with the switches. It tremendously simplifies day-to-day network operations:                                                                                                                                                                                                 |• auto-configuration and auto-upgrade of Switch Light OS,  
• automatic topology updates and event notifications based on fabric link state changes.  
• auto-scaling of the fabric—adding or removing nodes and/or links within the fabric requires no additional configuration changes on the controller. |
| Fabric LAG                            | Fabric LAG combines the underlying LAG functionality in switching ASICs with the centralized visibility of the SDN controller to create a highly resilient and efficiently balanced fabric. Compared to spanning tree protocols or even traditional MLAG/ECMP based approaches to multi-path fabric formation, Fabric LAG technology enables significantly reduced convergence time on topology changes and dramatically reduced configuration complexity.                       |
| Fabric Sync                           | Fabric Sync intelligently synchronizes Controller Information Base (CIB) with fabric node’s Forwarding Information Base (FIB) using the OpenFlow protocol. During a topology change, only delta updates are synced across impacted switches. Fabric Sync ensures strong CIB-FIB consistency as it is the single point of control for maintaining all forwarding and associated policy tables.                                      |
| Resilient Headless Mode               | In situations when both controllers are unreachable, fabric nodes are considered to be running in Headless mode. In this mode, all provisioned services continue to function as programmed prior to entering the Headless mode. Additionally, multiple levels of redundancy enable a highly resilient and self-healing fabric even during headless mode.                                                        |
| Centrally-managed Fabric              | Big Cloud Fabric Controller provides single pane of glass for entire fabric.                                                                                                                                                                                                                                                                                               |• Administrators can configure, manage, debug or troubleshoot, and upgrade the fabric nodes using CLI, GUI, or REST API.  
• REST APIs, CLI and GUI have application and tenant awareness.  
Single Pane of Glass fabric management enhances operational simplicity by providing a centralized dashboard for fabric management as well as quick and easy access to troubleshooting, analytics and telemetry information. Additionally, it provides simplified workflow for network operators and administrators. |
| Fabric View                           | Fabric View is the set of features that provides Advanced Multi-node Troubleshooting, Analytics & Telemetry in the Big Cloud Fabric solution.                                                                                                                                                                                                                             |
| API-first Fabric                      | Big Cloud Fabric Controller is highly programmable due to its “API-first” design principle and can be implemented as a closed loop feedback system. For example, security applications can dynamically detect threats and program the BCF controller for mitigation. The BCF GUI and CLI utilize the underlying REST APIs—hence are by definition consistent and hardened.                                               |
| Tenant-aware Fabric                   | Big Cloud Fabric provides built-in multi-tenancy via tenant-aware configurations, tenant separation and fine-grain inter-tenant access control. Configuration in the CLI, GUI or REST API is based on the concept of logical tenants.                                                                                                                                       |
| Service-aware Fabric                  | Big Cloud Fabric supports virtual and physical service insertion and service chaining. Services can be shared across tenants or dedicated to a specific tenant.                                                                                                                                                                                                             |
CONCLUSION

Big Switch is bringing the design principles pioneered by hyperscale data center operators to a broader audience. The Big Cloud Fabric’s flexible, scale-out design allows users to start at the size and scale that satisfies their immediate needs while future-proofing their growth needs. By providing a choice of hardware and software solutions across the layers of the networking stack and pay-as-you-grow economics, starting small scale and growing the fabric gradually instead of locking into a fully integrated proprietary solution, provides a path to a modern data center network.

Any service provider or enterprise IT organization can now realize the benefits of a bare metal SDN design:

- **Rapid App Agility**: An application/tenant-centric configuration to streamline L2/L3 configuration
- **Significant OpEx Reduction**: A logically central controller to reduce the number of management consoles\(^8\) by a ratio of over 30:1. [add zero touch and trouble-shooting as well?]
- **Simplified Network Operations**: APIs and integrations, starting with OpenStack, to integrate network provisioning, security and audit workflows,
- **Dramatic CapEx Reduction**: Bare metal Ethernet switch hardware to significantly reduce 3-year infrastructure costs

To learn more about how Big Cloud Fabric can enable hyperscale data center networking for your infrastructure, please contact info@bigswitch.com.

\(^8\) Relative to a 16 rack row of traditional L2/L3 switches (32 ToRs, 2 Aggregation Chassis)