INTRODUCTION

Data center applications are transitioning from a monolithic architecture (typically found as a 3-tier application) to a more distributed architecture powered by lightweight micro-services implemented via containers. This trend towards cloud-native applications is driven by the need to deliver agility to business by innovating faster and by leveraging analytical insights that rely on various data sources.

Containers are a new atomic unit of computing that is ideal for these emerging cloud-native, distributed applications. Containers offer a layer of abstraction, not unlike to what virtual machines offered when virtualization first arrived on the scene. However, Containers are lightweight in using the host footprint because they share an underlying operating system unlike a VM which hosts its own guest OS.
Containers enable micro-services based distributed applications where an application is decomposed into multiple network-connected micro-services, with each micro-service packaged in its own container. This highly-distributed application architecture leads to tremendous increase in east-west traffic compared to monolithic applications.

Containers enable applications portability by allowing applications to be developed on a laptop and run on a server or in a cloud environment. “Develop once, run anywhere” is a powerful software development paradigm shift, which also benefits IT organizations to achieve continuous integration and continuous deployment (CI/CD) processes for application lifecycle management.

Containers run on a host as a group of addressable processes within an operating system. They have lower resource overhead compared to virtual machines, which results in higher server utilization. They also differ from VMs from a packaging and portability perspective.
A lot of interest in containers in the past couple of years has resulted in emerging container ecosystems. Data center operators are evaluating various container technologies, including Docker, Kubernetes, Mesosphere and Red Hat OpenShift.

<table>
<thead>
<tr>
<th>Deploy and Manage</th>
<th>VMs</th>
<th>Containers</th>
</tr>
</thead>
<tbody>
<tr>
<td>~10 seconds</td>
<td>One File System per VM</td>
<td>Shared and Layered File Systems Across Containers</td>
</tr>
<tr>
<td>Density</td>
<td>1k/rack</td>
<td>10k+/rack</td>
</tr>
<tr>
<td>Topologies</td>
<td>Single vSwitch</td>
<td>Single or Nested vSwitches</td>
</tr>
</tbody>
</table>

Container orchestration includes workflow capabilities such as (a) scheduling container placement, replication, upgrades and downgrades, and (b) resource management of memory, CPU, IPs, ports and images. Container orchestration, based on definition, could also include service management of multiple containers using higher level constructs such as labels, groups, namespaces etc.

**THE CONTAINER CHALLENGE FOR NETWORKING**

The container networking challenge thus becomes to automate network configuration for rapid container deployment, visualize the health of container instances on the network, and provide a consistent network operational experience with container ecosystems.

Tracing container-to-container paths, both logical and physical (across vSwitches and leaf and spine switches), are much needed for network troubleshooting efforts. Containers are expected to be much more ephemeral than VMs and hence their attachment and detachment history from the network becomes a critical part of historical analysis.

Legacy networks with their box-by-box operational model becomes a bottleneck for containerized environments. If each switch needs to be configured manually for attaching new servers as well as for container addition and deletion, the agility and elasticity of the distributed application is severely impacted. Legacy network is proprietary hardware based with clunky CLI and limited GUI capabilities, making orchestration and troubleshooting a difficult, if not impossible, proposition.
CONTAINER-READY NETWORKING SOLUTION FROM BIG CLOUD FABRIC™

Big Cloud Fabric, from Big Switch Networks®, is the next generation data center networking fabric that leverages SDN to deliver intelligent, agile and flexible container networking.

Big Cloud Fabric (BCF) leverages software-defined networking (SDN) to provide a one big “logical switch” governed by a centralized controller. This solution delivers simplified network operations, visibility and telemetry of containers and their hosts, and network automation for rapid application and micro-services deployment. The scale-out architecture of BCF accommodates future growth in east-west traffic, caused by an increase in micro-services deployment, without breaking the bank. Its SDN architecture works on open industry-standard switch hardware that allows vendor choice and also reduces costs.

Big Cloud Fabric’s container networking solution involves having a BCF agent in the container orchestration layer and a BCF container plugin in the host.

The BCF agent in the container orchestration layer allows multiple container environments to be securely instantiated on a single BCF fabric, allowing for a consistent network operational experience and dramatically reduced cost and complexity. Multi-container environments can even be co-located along with OpenStack and VMware environments on the same BCF, with each environment fully isolated for secure operation. There is no need to have container technology specific silo networks.

The BCF Controller enables full visibility of traffic from container-to-container across both virtual and physical networks, and automatic network configuration of the network segment to which the containers belong. Rapid application and micro service deployment is now possible without having to wait for manual box-by-box operational steps required in a legacy network. Container end-points connected on the network now become visible. This helps application troubleshooting on the network. Container health insights via network traffic is provided via fabric analytics features. Given containers’ ephemeral nature with quick addition and removal based on usage, historical replay capability is also provided to understand the underlying dynamics for better troubleshooting and planning.

In short, Big Cloud Fabric makes networking container-ready. Networking finally operates at the speed of containers.
SOLUTION IMPLEMENTATION DETAILS

Before we describe the specifics of BCF container solution, let us review the container network stack as shown below.

The container orchestrator, via an orchestrator agent, communicates with containers running on the host, using a host plugin.

Leveraging this model, Big Cloud Fabric’s container networking solution involves having a BCF agent in the container orchestration layer and a BCF container plugin in the host.

BCF container plugin is architected to support both CNI plugin model (required by Kubernetes and Mesosphere) and libnetwork plugin model (required by Docker/Docker swarm).

A CNI plugin model implementation requires the orchestrator to spawn containers on the host at run time, and directly invoke the CNI plugin. So, the BCF orchestrator agent invokes the host CNI plugin that plumbs the newly spawned container to the network fabric, while satisfying the IPAM requirements of the container.

The libnetwork plugin model, on the other hand, allows for the container engine to directly invoke the plugin at the time of container orchestration.

Figure 6: BCF’s container solution leverages container plug-ins on the host and agent on container orchestrator.
SOLUTION DEPLOYMENT AND BENEFITS

A typical deployment of BCF’s Container solution involves 3 simple steps:

1. Deploy Big Cloud Fabric
2. Deploy container cluster of your choice, with master nodes and agent nodes
3. Install BCF Switch Light VX and BCF Container plugin/agent, using Ansible playbook.

The master node may be used to deploy applications and orchestrate container pods on the cluster nodes. BCF plugin takes care of container networking tasks including, but not limited to, managing IP address, attaching container to virtual Switch, and configuring BCF controller.

The solution benefits include no user error during container node auto detection and LAG formation. Also, BCF provides end to end container visibility and analytics, which are very useful for troubleshooting and historical analysis needs.
With virtual pod (vPod) technology, BCF can support overlapping IPs and VLANs across isolated vPods, in a single SDN fabric. BCF’s orchestration integration allows multiple hypervisor and container orchestration instances to work in parallel. With tenant-native constructs as part of its architecture, BCF is ideal for managed private clouds, engineering (dev & test) environments.

The multi container orchestration solution is beneficial for continuous integration (CI), continuous deployment (CD) environments. Another use case to leverage the vPod deployment would be for an independent software vendor’s (ISV) quality assurance (QA) teams, as they can share a common physical network across multiple test teams.

Figure 9:
BCF vPod Deployment:
CONCLUSION

Emerging cloud-native applications are driven by the need to innovate. They are increasingly being architected using micro-services, implemented via containers. Containers represent a new atomic unit of computing and are lightweight, more agile and ephemeral compared to VMs. However, agility of any application deployment process is constrained by the weakest link in the chain. That turns out to be the legacy network infrastructure, which shackles network operators to box-by-box configuration, provisioning and troubleshooting.

BCF delivers a container-ready next generation data center networking solution. It allows plumbing of containers in the network fabric, allowing for rapid operational velocity while reducing, and in some cases eliminating, user errors. The BCF solution supports multiple container orchestrators at the same time. This allows IT teams to deploy consistent operational practices across various container ecosystems and support multiple innovation cycles.

BCF’s container-ready next generation networking solution is based on an SDN-based leaf-spine fabric architecture that is deployed on open network hardware to work within flat IT budgets without sacrificing IT innovation.

RESOURCES

• The links provide more information on why Big Cloud Fabric may be needed, overview of next-generation DC switching architecture, and its drivers and benefits.

• Demo video of BCF container solution may be found in here at the 37:30 mark.

• Big Cloud Fabric Community Edition is free and can be downloaded here.

• To try BCF on-line, sign up for our free Big Switch Labs and test drive variety of use cases.