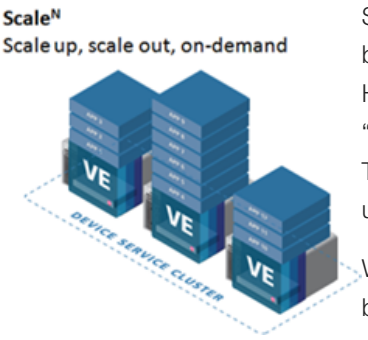
# F5 Friday: Sync, Share, and Scale

**Lori MacVittie, 2011-23-09**

#v11 *Scale$^N$ breaks out of the traditional infrastructure scalability mold*

We previously introduced Scale$^N$ but we didn't really dig into how it's enabled, other than to mention it's been made possible in part by leveraging F5's vCMP (virtual Clustered Multi-Processing) technology, which puts the "virtual" in "virtual networking."


The basic premise of infrastructure scalability is that if the component providing the scalability fails, well, the service for which it provides HA fails. That's not good. So it was that HA architectures employing a variety of models came about to ensure that such a scenario did not happen. Active-standby was the original model, in which one component (secondary) was always on standby, ready to assume duties should the active (primary) component fail. This was later deemed to be a waste of resources, and an active-active model became the HA architecture of choice for some organizations. This model was imperfect in its ability to assure HA because if the total load on the combined system exceeded the capacity of a single component and the primary failed, it was assured that there would be a disruption of service. Depending on what percentage of your revenue relies on your web presence, such disruptions can be disastrous.




So other models began to become popular. For some time now an N+1 model has been the most prevalent choice of HA architectures for those organizations desiring HA assurance and maximum efficiency. The N+1 model assumes any number of "active" components N, each with an independent dedicated secondary (standby). This is, understandably, inefficient as there are always components sitting idle – unused resources, in cloud-speak.

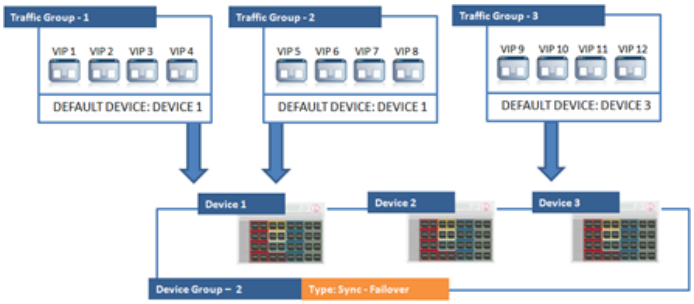
What F5 Scale$^N$ does to break out of that model is to eliminate the tight-coupling between the primary and standby components, allowing the primary to "fail over" to any available component configured to be a part of the HA group. What's more exciting, is the ability to eliminate the requirement to failover at the ***device*** or ***component*** level and move that upward to the application layer. This provides a level of fault isolation in the infrastructure architecture not previously offered by traditional HA architecture, as the old models assume an "all or nothing" approach to failover. If one application triggers a failover event, all applications are going to be affected. Not so with Scale$^N$, which allows individual applications to failover or purposefully move between components in a configured Device Service Cluster. For even more flexibility, components can be physical or virtual and need not be identical hardware or configuration.

## Scale$^N$ – Device Service Clusters

At the heart of Scale$^N$ are Device Service Clusters: a group of two or more BIG-IP devices in a trust relationship that can share resources and ensure high availability for application delivery. A trust relationship is established between two BIG-IP devices based on mutual authentication through the exchange of device certificates. Device Service Clusters come in two flavors: Sync-Only and Sync-Failover. The former, Sync-Only, is used to synchronize full device configuration, enabling consistency across devices. Sync-Failover groups are used to synchronize configuration objects for the purposes of managing failover events, which allows operators to group objects that should be synchronized in an easy to manage, folder-style paradigm. Objects include certificates, CRLs, data groups, external monitors, iApps, iRules, policies and profiles. Device Service Clusters utilize an encrypted, dedicated synchronization channel for secure communications and can contain BIG-IP devices with different "personalities"; it's an asymmetric module deployment model.

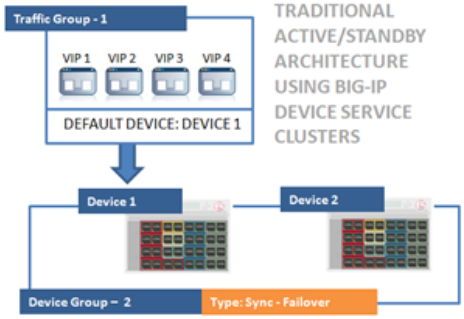
It's also important to note that a "device" is not necessarily a physical or virtual representation of a device, but can also represent a vCMP instance.

Additionally, we've added the notion of a "traffic group" – a group of self IP addresses, virtual addresses, and SNATs that can float between devices in a BIG-IP device group for the purpose of maintaining high availability. It's a lot like the shared IP address model used for device-level failure in an active-active or active-standby model in which the IP address "floats" between two devices and upon failure of the primary the secondary assumes control immediately. Grouping the devices that support a set of traffic groups allows distribution of traffic across all available devices in the event of a failure.




But it's not just failure with which we need to concern ourselves, it's *impending failure* due to capacity constraints. Imagine a scenario in which traffic is steadily increasing and at the rate it's going the servers – and infrastructure – will eventually be overwhelmed. Perhaps it's an unplanned event-based increase, perhaps it's an attack. Whatever the cause, we can project that if we don't add capacity now, we're going to suffer unacceptable downtime. F5 BIG-IP has always been capable of dynamically increasing service capacity by aggregating newly launched service instances into the appropriate resource pool, but it has never been able to scale ***itself***, and that – especially in the world of cloud computing – is an important capability.

Scale$^N$ provides the means by which an additional instance of BIG-IP can be launched, added to an HA group, and synchronized to provide scale on-demand. Because of the ability to mix physical and virtual form-factors, this allows operations to meet sudden, potentially overwhelming traffic at the infrastructure layer by adding virtual instances that can be subsequently decommissioned. This same technique is also valuable in scaling out applications because it allows not only scalability, but resiliency (failover) to be maintained as load increases.




While BIG-IP has always been able to dynamically increase and decrease the size of application "pools" or "clusters" used to scale an application, the underlying failover models were still based on a traditional model. Scale$^N$ continues to support the same agile scaling model but adds a more resilient HA model under the covers to ensure that application failure is met with immediate mitigation through redirection of requests to alternative, available devices.

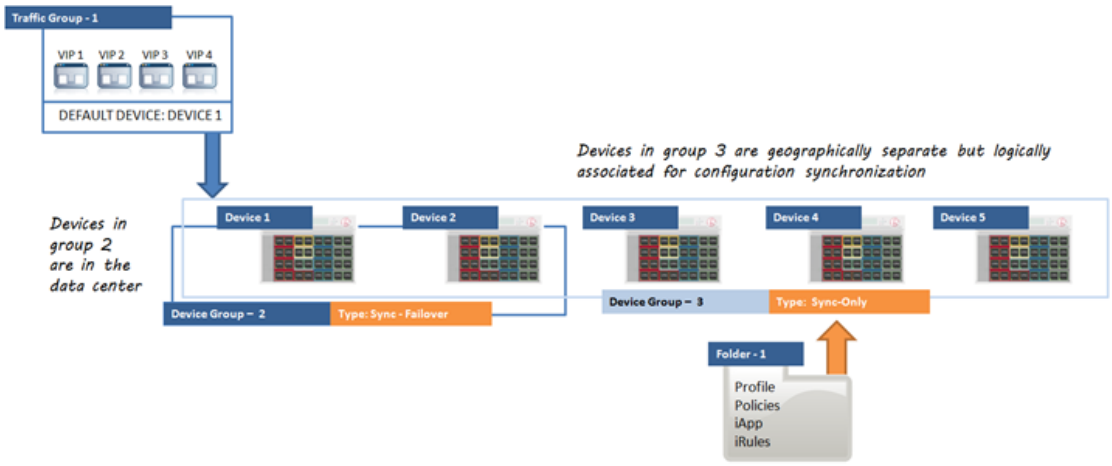
The flexibility of such a model allows for myriad architectures – those pre-positioned and those implemented on-demand through scripting and automation –to be designed and put into place to handle the increasing load coming from mobile devices, the adoption of HTML5, and growing video usage by customers and users alike. Combining iApp with ScaleN makes it possible to dynamically scale and migrate applications without losing the application delivery meta-data essential to deliver that application securely and with optimal performance.

It is important to note that Scale$^N$ does not preclude the implementation of long trusted and implemented HA architectures. Active/Standby and Active/Active are just as easily implemented as a more robust, modern architecture.

## CONSISTENCY

Just as important as the failover and HA aspects of Scale$^N$ are the enablement of capabilities coming from its internal implementation. The ability to configure synchronization groups, for example, can also be used to ensure consistency across devices deployed in disparate locations. This is particularly useful for those application delivery services, such as web acceleration and caching architectures, where remote devices participate in the application delivery process and thus need a consistent, up-to-date configuration to ensure compliance with corporate configurations.

For example, a common deployment of WebAccelerator comprises a redundant pair of devices at the corporate data center and single units in branch offices. With folders and device service clusters, operators can configure the systems to ensure consistent policies. Devices in the corporate data center are members of a device service cluster configured for synchronization and failover (sync-failover) to provide high availability. All the devices, including those at the various branch offices, are a member of a device service cluster configured for synchronization (sync-only). A folder is created and is then populated with the appropriate configuration objects – the policies, profiles, etc… that determine the behavior of the WebAccelerator device based on corporate and operational policies and requirements. Folders provide a familiar navigation and organizational paradigm in which objects can be grouped together and managed as a single entity. Synchronization is then configured to target the device service cluster for that folder, enabling synchronization of the associated objects across all of the instances.



This is a marked difference from how high availability devices have been configured in the past, and it is a leap forward for those engaged in the process of enabling a dynamic and often distributed data center. Scale$^N$ with its new scalability and and more flexible synchronization paradigm changes the way in which architects are able to design and deploy HA architectures. The decoupling of applications and services from the device, enabling a per-application or service failover paradigm as opposed to the old “all or nothing” architecture enables greater fault isolation, a key capability for supporting multi-tenant and shared environments to ensure tenants do not adversely impact one another.

With the cost of downtime continuing to grow in dollars and impact to reputation, the ability to design more modern, agile HA architecture is paramount to achieving more resilient, available and ultimately secure applications and services.

- **F5 Friday: How Can I Manage Thee? Let Me Count the Ways…**
- **F5 Monday? The Evolution To IT as a Service Continues … in the Network**
- **F5 Friday: The Gap That become a Chasm**
- **Cloud is an Exercise in Infrastructure Integration**
- **Beware the Cloud Programmer**
- **This is Why We Can’t Have Nice Things**
- **All F5 Friday Posts on DevCentral**
- **All BIG-IP v11 Posts on DevCentral**

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | f5.com

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com