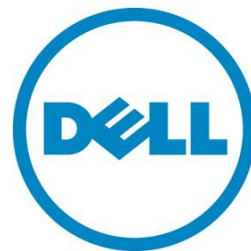


---

# Dell Force10 Hadoop Network

*The Scale out solution*

Dell Force10 Product Management



Hadoop Overview .....	3
Hadoop's Network Needs .....	4
The Dell Solution .....	5
The Network components .....	7
Management network Infrastructure .....	7
Data network and server access of LOM ports .....	7
Access Switch or Top of Rack(ToR): .....	7
Aggregation switches: .....	8
Network Architectures for a 4PB deployment .....	8
ECMP .....	10
10GbE Hadoop.....	11
Appendix .....	12
Stacking S60s.....	12
Uplinking the S60s .....	13
Server Gateway .....	14
Management network.....	14
VRRP on S4810.....	14
VLT on S4810 .....	14

## Hadoop Overview

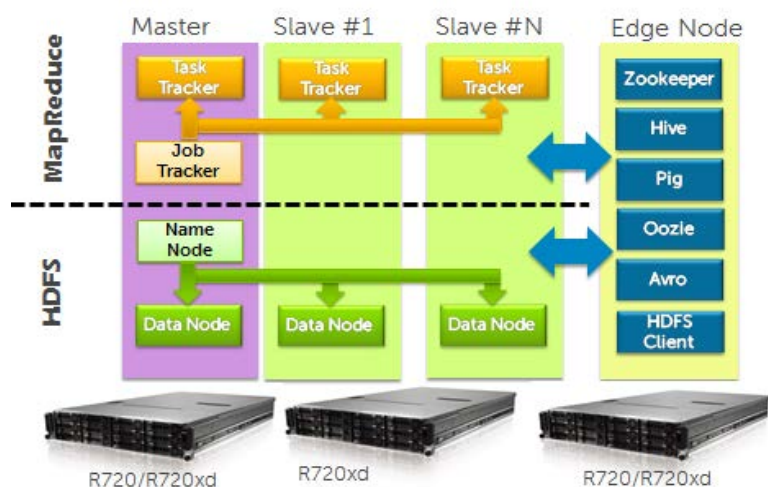
Hadoop is an emerging big data analytics technology that is used to mine unstructured data that is hard to put in a relational database for traditional data warehousing analysis and yet that's too valuable to be thrown away. Enormous amounts of data is collected every day from consumers and business operations in the form of anywhere between social and business interaction and machine logs that facilitate the operations that work silently behind the scenes.

Big Data has become popular in certain market segments like social media websites, Federal agencies, retail business, banking and securities. It is part of the new and emerging Big Data effort where the new mantra is to not throw away any data for business analysis and in many cases for regulatory and corporate policies. The need arises for a technology that can store data on cost affective hardware and scales with the growth of data.

The average size of an Hadoop cluster varies by customer segment depending on how much raw data it generates and how long they archive it. Most enterprise needs would be met with Hadoop clusters of 20-30 servers depending on storage and processing per server. But there are Federal and social media data centers that have hundreds of servers in a cluster.

The three types of machine roles in a Hadoop deployment are Edge, Master, and Slave.

Figure 1. Hadoop functional schema

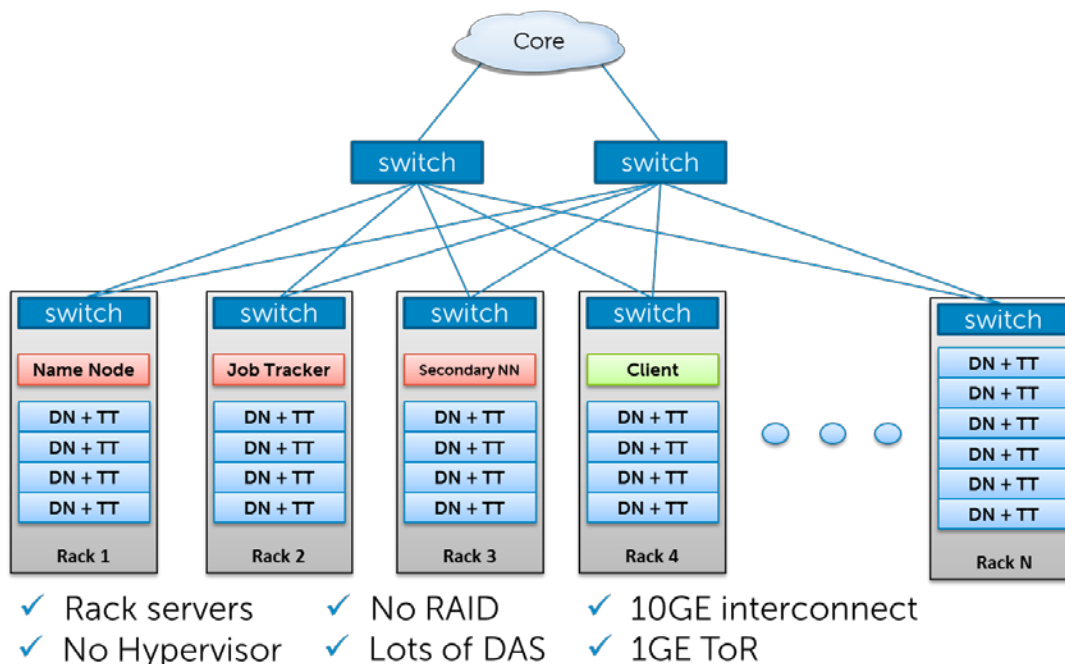


The Master nodes oversee two key functional pieces that make up Hadoop: storing lots of data (HDFS), and running parallel computations on all that data (Map Reduce). The Master also called the Name Node, oversees and coordinates the data storage function (HDFS), while the Job Tracker oversees and coordinates the parallel processing of data using Map Reduce. Slave nodes make up the vast majority of machines and perform the storage of data and running the computations against it. Each slave runs both a Data Node and Task Tracker daemon that communicate with and receive instructions from their master nodes. The Task Tracker daemon is a slave to the Job Tracker, the Data Node daemon a slave to the Name Node. Edge machines act as an interface between the client submitting the jobs and the processing capacity in the cluster.

The role of the Client machine is to load data into the cluster, submit Map Reduce jobs describing how that data should be processed, and then retrieve the results of the job when once the request is processed.

The physical layout of the cluster is shown in this figure below.

Figure 2. Hadoop Cluster

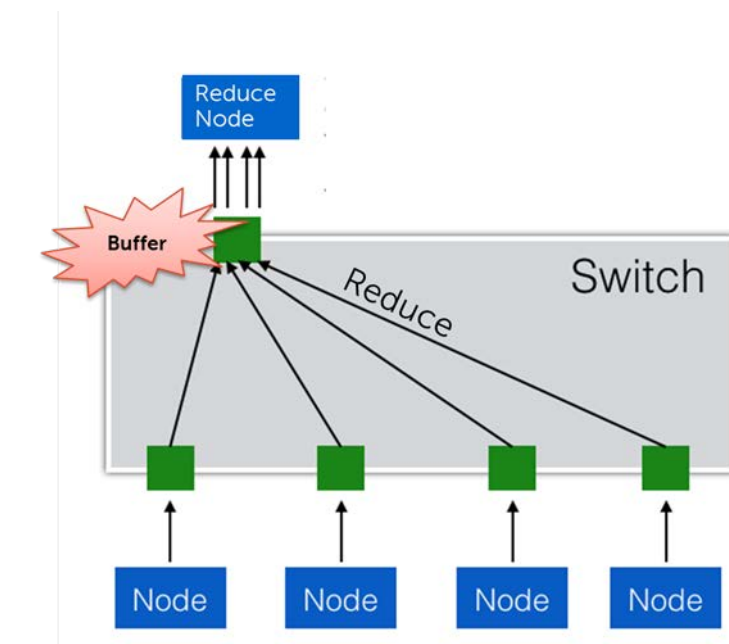


Rack servers with respective roles are populated in racks and are connected to a top of rack (ToR) switch using 1 or 2 GbE links. 10GbE nodes are gaining interest as machines continue to get denser with CPU cores and disk drives. The ToR has uplinks connected to another tier of switches that connect all the other racks with uniform bandwidth, and form a cluster. A majority of servers are slave nodes with lots of local disk storage and moderate amounts of CPU and DRAM. Some of the machines will be master nodes with a different configuration favoring more DRAM and CPU, less local storage.

### Hadoop's Network Needs

As much as we may think that any general purpose Ethernet switch would work for Hadoop, Map Reduce presents a challenge that only a switch with deep buffers can meet. Incast is a many-to-one communication pattern commonly found in MapReduce computing framework. Incast begins when a singular Parent server places a request for data to a cluster of Nodes which all receive the request simultaneously. The cluster of Nodes may in turn all synchronously respond to the singular Parent. The result is a micro burst of many machines simultaneously sending TCP data streams to one machine. This simultaneous many-to-one burst can cause egress congestion at the network port attached to the Parent server, overwhelming the port egress buffer resulting in packet loss which induces latency. A switch with sufficient egress buffer resources will be able to better absorb the Incast micro burst, preventing the performance detrimental packet loss and timeouts.

Figure 3. TCP InCast in Hadoop



For this reason Dell Force10 introduced a deep buffer switch, the S60 that has an industry high of 1.25GB of optimized dynamic buffer. At line rate the switch can hold over 2.5 secs worth of 64Byte packet size traffic and this capacity grows with larger packet sizes. Dell Force10 S60 switches significantly reduce the time it takes to complete the Map Reduce or HDFS tasks and drives higher performance eliminating network bottlenecks.

### The Dell Solution

Let's familiarize ourselves with [Dell's offering](#) in the form of a 3 rack size cluster. A cluster is a sizing reference that would refer to the complete infrastructure for Hadoop deployment in a data center. A cluster is what's referred to as unit that can serve the business needs for an enterprise. A cluster could be fitted in as little as a single rack or tens of racks. Dell's architecture shows how a medium size 3 rack cluster that can grow to more than 12 racks in a simple setup. This cluster could share the same Name Node and management tools for operating the Hadoop environment.

We then move on to share our ideas on how a 10GbE server based Hadoop would look like and what Dell has to offer as a solution. We believe a Hadoop deployment can be scaled to thousands of nodes using a cost effective CLOS fabric based network topology.

It's seen that Hadoop deployments are normally sized in the number of nodes deployed in a cluster. A node could have different storage, memory, CPU specifications. So a accurate sizing is not possible as it would be an uneven comparison of different types of nodes. We introduce the Data Node, also know as Slave Node, storage as our sizing tool for an enterprise that knows how much data they have to analyze and then figure out how much hardware they would need. This would translate into the number of servers and that in turn would determine the networking portion of the cluster.

First the rationale behind the number 3 used below.

Hadoop has the concept of "Rack Awareness". As the Hadoop administrator, you can manually define the rack number for each slave Data Node in the cluster. There are two primary reasons for this: data loss prevention, and network performance. It should be noted that each block of data is replicated to multiple machines to prevent a scenario where a failure of one machine results in a loss of data. In order to avoid this, somebody needs to know where Data Nodes are located in the network topology and use that information to make an intelligent decision on where data replicas should exist in the cluster. That role is performed by the Master role server called the Name Node. The replication of the data needed to prevent loss is usually 3 times the actual data.

The server hardware configuration suggested by Dell is detailed below according to the role. It plays in the cluster.

Figure 4. Server hardware configuration

Machine Function	Master Node	Secondary Master Node (serves as Admin Node)	Edge Node	Slave Node
Platform	PowerEdge R720xd			
CPU	2x E5-2640 (6-core)			
RAM (Minimum)	96GB			48GB
LOM	4x1GbE			
DISK	6x 600GB 10K SAS 2.5"			24 x 1TB SATA 7.2K 2.5"

Based on the above table using a certain type of server let's try to standardize a cluster size used in the Dell Reference Architecture we can define a Hadoop deployment size in terms of usable storage needed.

Dell Hadoop architecture size can be defined as:

24TB per node x 60 nodes per cluster of 3 racks / 3(data replication) = **480TB** usable data size.

Customers could use this formula for Sizing:

Raw business data → Compress → add rate of data growth → x3 = total storage needed.

Hadoop can very well work with compressed data so to save on storage space data compression is often used. The other fact to consider would be the rate at which the data would grow so the sizing should include growth. This way just enough hardware is purchased that fits the need

Next, we suggest network configurations that can support a **2 Peta Byte** Hadoop deployment. But first let's discuss the building blocks of the network.

## The Network components

The Dell Hadoop network consists of 2 major network infrastructure layouts.

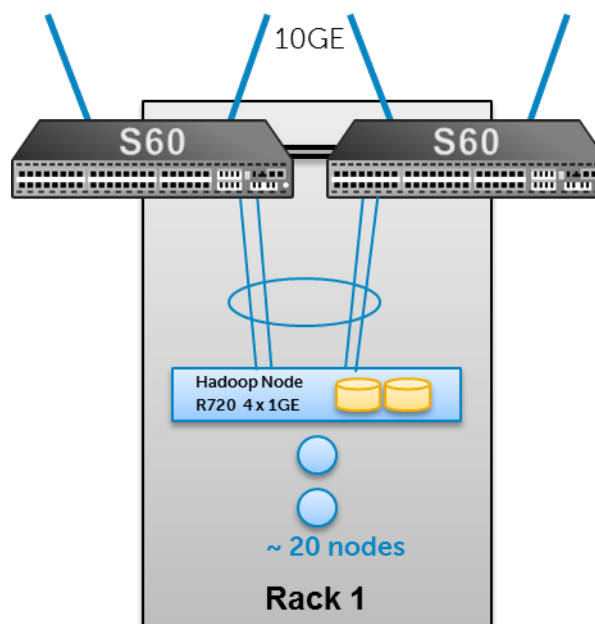
**Management network Infrastructure-** The BMC management network consists of the aggregation of all BMC ports on the nodes, shares the same top-of-rack switch but a separate VLAN. It could potentially be a separate out-of-band network but in-band connections would save extra costs.

**Data network and server access of LOM ports** - The most commonly used 2RU server offered by Dell for Hadoop are C6100 and R720xd, each may have 4 or more 1GbE interfaces to uplink. These ports are terminated on 2 separate 1GbE Ethernet switches that are stacked together in the same rack. Server connections to the network switches could be one of the four options: Active-Active LAG in load-balance formation, Active-Backup in failover/failback formation, Active-Active round robin based on gratuitous ARP or just a single port connection. The architecture sizing and performance does not depend which method of teaming is used.

First we take a look at the components within a single rack.

**Access Switch or Top of Rack(ToR):** The servers connect to two ToR switches in each rack. Dell's reference architecture recommends the S60 for 1GbE connectivity and the S4810 for 10GbE server connectivity. Within the rack, two S60s are stacked together over a 24G stacking link. This functionality enables switches to be managed as a single unit and allowing the servers to connect to different switches for resiliency. The S60 ToR switches have two expansion slots for a 2 port 10G module or a 2 port stacking module. This architecture recommends one module of each type in the two slots of each ToR. The stacking links connect the 2 ToR switches back to back creating a stack of two switches. The 2 x 10GbE interfaces from each ToR connects into the Aggregation switches, one port from each ToR to each Aggregation switch, forming a LAG of 4 ports from the stack for uplink.

Figure 5. Single Rack View

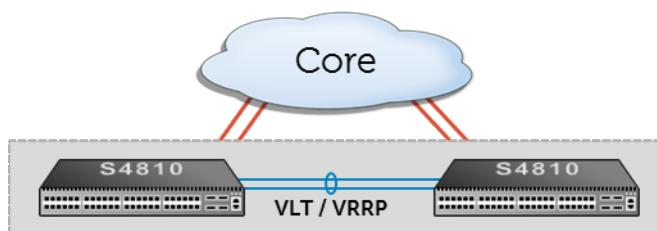


This is what's needed in a single rack.

Total Racks	<b>1 (6-20 nodes)</b>
Top of Rack switch	2xS60 (2 per rack)
Aggregation switch	Not needed
Server	2RU R720/R720xd or C6100
Over-subscription at ToR	1:1
Modules in each ToR	1x12-2port Stacking, 1x10G -2 port uplink

**Aggregation switches:** The hadoop racks interconnect through the aggregation switch which is a pair of S4810 switches. The uplinks are 10GbE interfaces from the top-of-rack S60 to the S4810. The recommended architecture uses the Virtual Link Trunking (VLT) between the two S4810s. This feature enables a multi-chassis LAG from the stacked ToR switches in each rack. The stacks in each rack would divide their links between this pair to achieve the powerful active-active forwarding capability at full bandwidth, without any requirement for spanning tree. For scaling to larger deployments, layer-3 routing from ToR to the aggregation is a good option. S4810, the recommended aggregation switch is 10GbE and 40GbE capable. The 40GbE interfaces on the S4810 could be converted into 4x10GbE, thereby converting this switch into 64 ports of 10GbE, which can scale the Hadoop deployments to hundreds of nodes.

Figure 5. The Aggregation view



## Network Architectures for a 4PB deployment

Once the single rack is deployed from the server and network perspective, we can take a look at the multi-rack view and then move on to configure the aggregation switches that connect the racks together. This section shows the S4810 aggregating the clusters together to enable inter-rack traffic as well as the management network. There are two separate VLANs for data and management; all port-channels on S4810 and ToR are tagged with these two VLANs.

The Hadoop network could easily scale to 240 1GbE nodes using this network scheme. The following table shows the network inventory details in a POD of 3 racks.



Figure 6. Multi-rack view

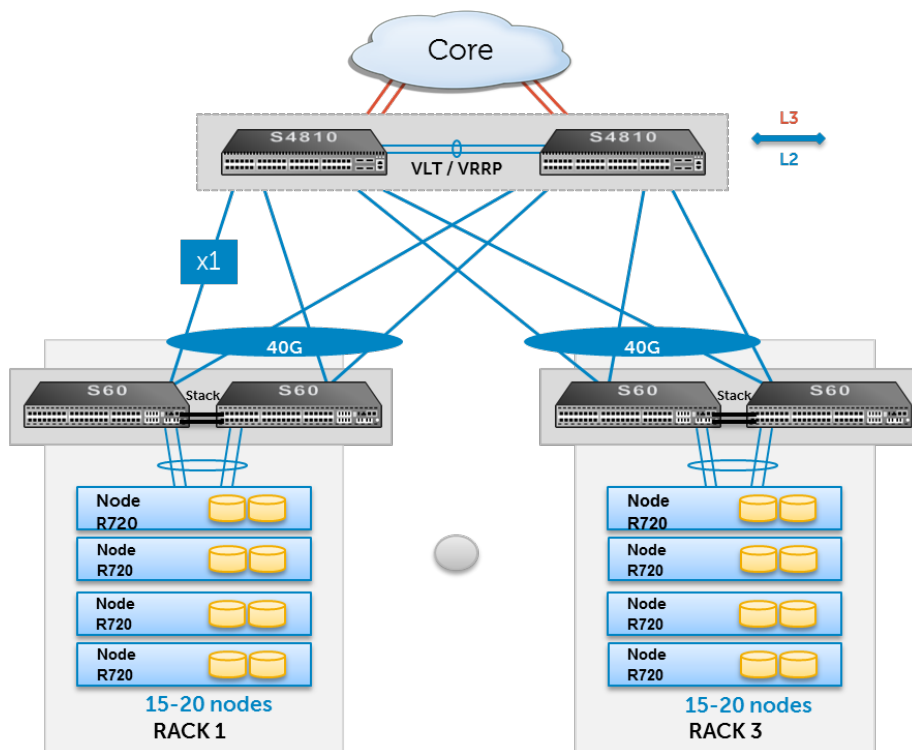


Table 2: Multi-rack view of Network Equipment

Total Racks	<b>Up to 12 racks or 240 nodes</b> (15-20 nodes per rack)
Top-of-Rack switch	6-24 S60 (2 per rack)
Aggregation switch	2xS4810
Server	2RU R720/R720xd
Over-subscription at ToR	1:1
Modules in each ToR	1x12-2port Stacking, 1x10G -2 port uplink

This POD architecture of 3 racks could have 480 TB of storage due to the fact that each Data Node can be filled with 24 x 1 TB SATA disk drives. When you scale this upto more than 4 PODs, or 12 to 13 racks, you can get to a total storage capacity of 2 PetaBytes.

As illustrated in the previous figure, the network can easily support a large deployment of more than 240 nodes that can suit well for medium to large data analytics requirement.

Next we have a reference architecture that scales even more by using a Layer 3 routing based fan out for aggregation that consists of 4xS4810 switches instead of 2. This network can scale up to 600 nodes, delivering capacity that is enough to run a large scale Hadoop network. This network scales to 30racks that can handle higher than 4 PB of data. The differentiating factor in this one is the absence

of stacking within the rack which allows us to get 2 additional 10G interfaces per switch. Each switch is a fault domain in its own with 2 such domains in a single rack. The 4 x 10GbE links can now uplink into 4 switch spine forming a CLOS fabric that has a high degree of stability and scalability built into it.

Figure 7. 600 Node Hadoop Cluster

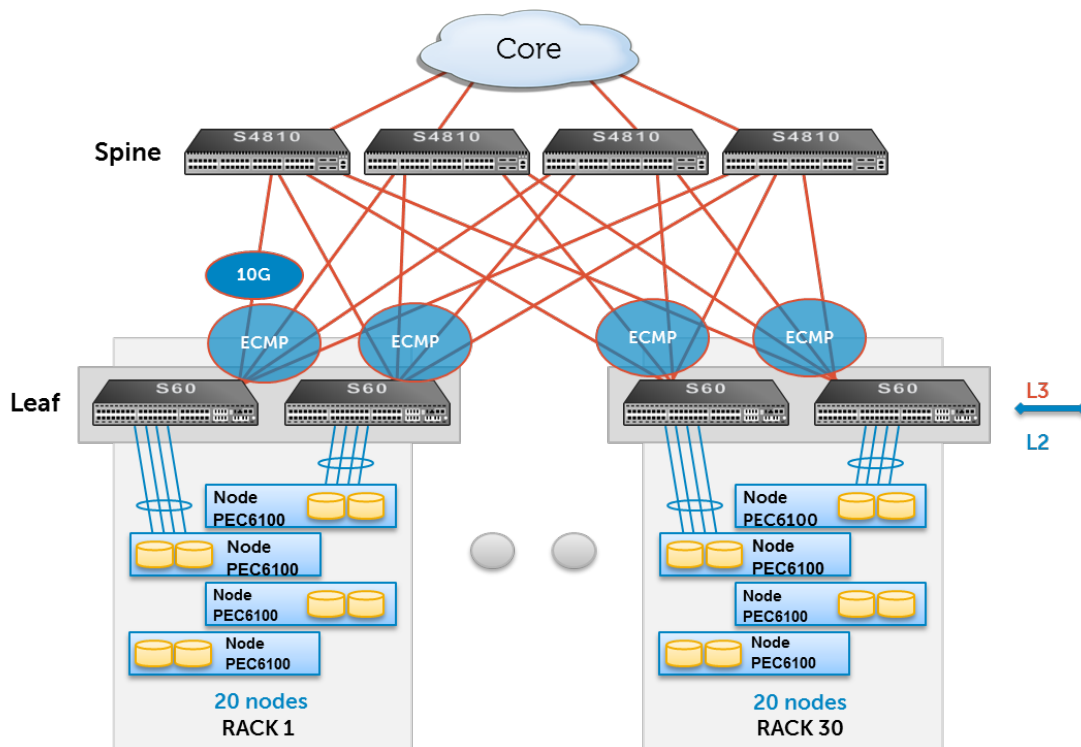


Table 3: Multi-rack view of Network Equipment

Total Racks	<b>Up to 30 racks or 600 nodes</b> ( max 20 nodes per rack)
Top of Rack switch	Up to 60 S60 (2 per rack)
Pod-interconnect switch	4xS4810
Server	2RU R720
Over-subscription at ToR	1:1
Modules in each ToR	1x12-2port Stacking, 1x10G -2 port uplink

### ECMP

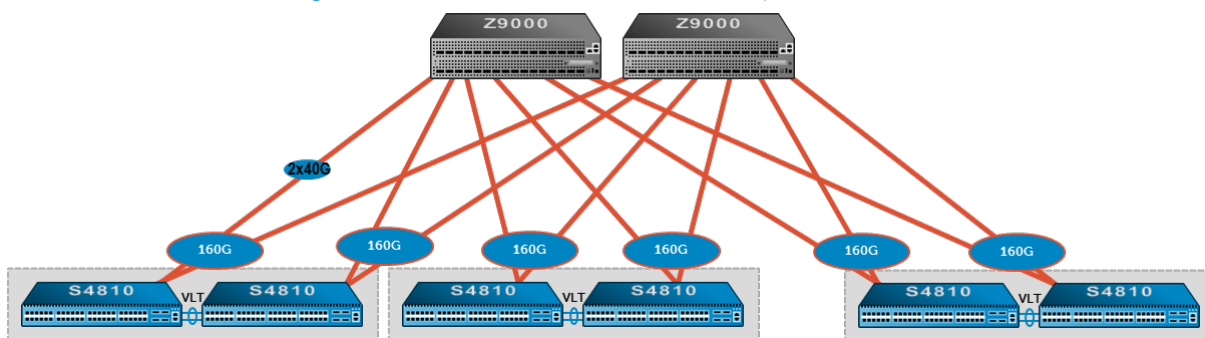
The CLOS fabric in the above figure is based on Equal Cost Multi- Path feature that enables flow based load-balance of traffic across multiple routing interfaces where the cost of reaching the next hop is equal. Scales out networks routinely deploy this feature available only in high end switches and routers, to balance traffic over multiple links ensuring high-availability and full bandwidth. The sub-second convergence times on the Dell Force10 switches ensures that a failure of any link will be resolved by balancing the traffic over the alternate paths.

OSPF point-to-point links are formed from each s60 to the 4xS4810s. This means that from any point in the network to any other rack the number of hops remain the same.

## 10GbE Hadoop

In future Hadoop applications would increasingly be deployed on 10GbE servers. That brings about an enormous economy of scale in the usage of hardware. That would need 10GbE switches in the racks. Dell Force10 has a ready solution to cover the 10GbE needs. This can be achieved using the S4810 as a Top-of-rack switch and the option of using S4810 or Z9000, the 10G/40G high density switch in the aggregation. The scale that is achieved by that configuration can grow into thousands of nodes using a CLOS architecture which was used in the 600 1GbE node solution above. Running 40GbE Ethernet switches like the Z9000 in aggregation can achieve a scale of more than 2000 nodes using a CLOS fabric.

Figure 8. CLOS Fabric of 768 10GbE ports



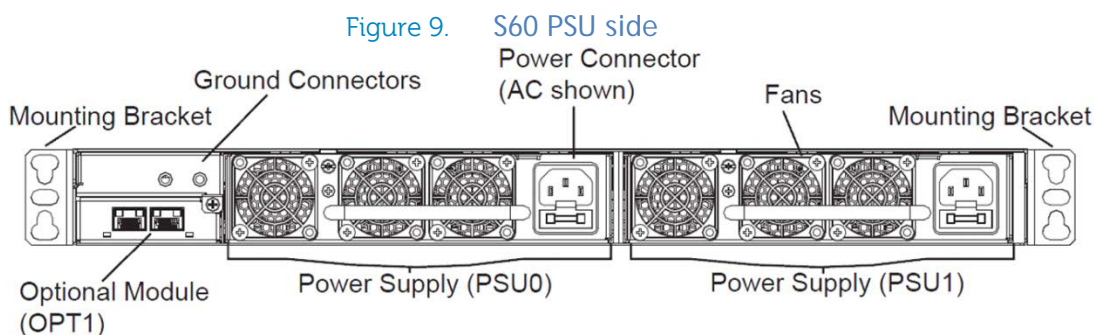
In the figure above we see an example of a CLOS fabric that grows horizontally. This technique of network fabric deployment has been used by some of the largest Web 2.0 companies whose business range from social media to public cloud in their data centers. Some of the largest Hadoop deployments also use this new approach to networking in the recent times. Dell [Hadoop solutions](#) has hands on experience in building Hadoop and Big Data analytics farms while Dell force10 is a trusted vendor in the field of networking. Dell can help an enterprise solve its Big Data needs with a scalable end-to-end solution.

## Appendix

### Stacking S60s

The following configuration helps stack the two S60s together within the rack. This configuration assumes the stacking module in both S60s is in slot 0 (IO facing side) and the 10GbE uplink module is in slot 1 (power supply and fan side).

Connect the ports on module 1 (PSU side from the left) to similar ports on the second S60 using the stacking cables. The stack is automatically detected and formed without user configuration. Using the CLI command, 'show system brief', verify that the stacking module is detected by the S60.



When you are adding units to a stack, you can either:

- Allow FTOS to automatically assign the new unit a position in the stack, or
- Manually determine each unit's position in the stack by configuring each unit to correspond with the stack before connecting it. Three configurable system variables affect how a new unit joins a stack: priority, stack number, and provision.

After the new unit loads, its running and startup configurations is synchronized with the stack.

```
TOR-Rack1#stack-unit renumber
TOR-Rack1(conf)# stack-unit priority <higher priority determines primary role>
```

After connecting the switches together run the following command to check the status of the stack

```
TOR-Rack1#show system brief

Stack MAC : 00:01:e8:d5:ef:81

-- Stack Info --

Unit   UnitType  Status  ReqTyp  CurTyp  Version  Ports
-----
0      Standby   online  S60     S60     8.3.3.7  52
1      Management online  S60     S60     8.3.3.7  52
```

## Uplinking the S60s

The following configuration helps create configurations for the uplink for the stack. This configuration assumes the 10GbE uplink module is in slot 1 (power supply and fan side). The uplink ports are going to be numbered 0/51, 0/52 and 1/51, 1/52 respectively. All four 10GbE interfaces should be part of a single LAG or port-channel. The following configurations show that.

```
# Put the user ports in the switchport mode
TOR-Rack1(config)# interface range gigabitethernet 0/1 - 47
TOR-Rack1(config-if-range-gi-0/1-47)# no shutdown
TOR-Rack1(config-if-range-gi-0/1-47)#switchport
TOR-Rack1(config-if-range-gi-0/1-47)#end

# Repeat the same for ports on the second unit
TOR-Rack1(config)# interface range gigabitethernet 1/1 - 47
<snip>...

# Create port-channel of the 4 10G ports.The example below shows it for 1
port.
# Repeat the same configs for other 10G ports 0/52,1/51 and 1/52.

TOR-Rack1(conf)#interface TenGigabitethernet 0/51
TOR-Rack1(conf-if-Te-0/51)#no shutdown
TOR-Rack1(conf-if-Te-0/51)#port-channel-protocol lacp
TOR-Rack1(conf-if-Te-0/51-lACP)#port-channel 1 mode active

# Change the defaults on the port-channel that gets created automatically
# From the above commands.

TOR-Rack1(conf)#interface port-channel 1
TOR-Rack1(conf-if-po-1)#no shutdown
TOR-Rack1(conf-if-po-1)#switchport

# Add the Data ports 0 through 30 and the port-channel 1 to vlan 100

TOR-Rack1#config
TOR-Rack1 (conf)#int vlan 100
TOR-Rack1 (conf-if-vlan)#tagged po 1
TOR-Rack1 (conf-if-vlan)#untagged gi 0/0-30
TOR-Rack1 (conf-if-vlan)#untagged gi 1/0-30
TOR-Rack1 (conf-if-vlan)#show conf
!
interface Vlan 100
no ip address
tagged Port-channel 1
untagged gi 0/0-30
untagged gi 1/0-30
```

So far the configuration is sufficient to link the nodes to the ToR switches, stacking the ToR and uplinks from ToR.

The uplink port-channel links are all active and forward traffic to the aggregate switches. Each flow, with a unique combination of a source and destination, gets hashed internally and gets load-balanced across the port-channel.

## Server Gateway

The nodes in a rack have a single virtual IP address as their gateway for routing purpose. The VRRP protocol runs on the aggregation S4810s. It does not need any configuration on the ToR. The VRRP master owns the virtual IP and does the routing but the combination of VLT and VRRP ensures that backup also routes or switches the traffic if it has a path in its forwarding table. This is an active-active capability where routing is independent of whether the switch owns the virtual IP.

## Management network

The BMC ports from all the nodes connect to the same ToR switch as the data ports. However the management vlan is separate from the data vlan. Ports 0 to 30 on the ToR are reserved for data connections and 31 to 48 for network management. This is achieved by creating a separate VLAN on the ToR and adding all the management ports to that VLAN.

```
TOR-Rack1(conf)#int vlan 300
TOR-Rack1(conf-if-vlan)#tagged po 1
TOR-Rack1(conf-if-vlan)#untagged gi 0/31-47
TOR-Rack1(conf-if-vlan)#untagged gi 1/31-47
```

## VRRP on S4810

The following configuration shows sample VRRP configuration on the S4810s. This configuration is created on the VLAN interfaces of the S4810. Since there is only a single VLAN 100 in the cluster of three racks, only one instance of this configuration is needed.

```
Force10_VLTpeer1(conf)#int vlan 100
Force10_VLTpeer1(conf-if-gi-1/1)#vrrp-group 100
Force10_VLTpeer1(conf-if-gi-1/1-vrid-111)#virtual-address 10.10.10.1
#One or more these virtual IP addresses can be configured, which can be used
#as the unique gateway per rack or cluster.
Force10_VLTpeer1(conf-if-gi-1/1-vrid-111)# priority 125
# Priority from 1-255 can be used to determine which switch owns the VIP and
#becomes the VRRP master.

# Repeat the same configuration on the second VLT peer, except for a
#different priority.
```

## VLT on S4810

The second part of configuration is the pod-interconnect switches that run VLT with each other.

Figure 10. S4810 VLT interconnect



Following these steps we will configure VLT on the pair of S4810s that interconnect the racks. To configure virtual link trunking, you must create a VLT domain, configure a backup link and

interconnect trunk, and connect the peer switches in a VLT domain to an attached access device (switch or server). Before this, RSTP should be configured as a best practice on the s4810 as well as the S60s.

```
Force10_VLTpeer1(conf)#protocol spanning-tree rstp
Force10_VLTpeer1(conf-rstp)#no disable
Force10_VLTpeer1(conf-rstp)#bridge-priority 4096

#Repeat the same on VLTPeer2 with a different bridge priority to make it the
root.

Force10_VLTpeer2(conf-rstp)#bridge-priority 0
```

The next figures show a sample VLT configuration. VLT works over an ICL or primary link and a backup link. In absence of a direct path to the destination, the ICL link would carry the traffic to the peer. The backup link is only for heartbeat status from the peer and no data traffic flows over it.

Configure a VLT domain

```
vlt domain <domain id >
```

Fix the VLT system parameters to avoid negotiations (for faster convergence)

```
primary-priority <value> (suggestion: 1 for the primary, 8192 for sec)
system-mac mac-address <value> (same MAC address on all VLT peers)
unit-id <value> (suggestion: 0 for the primary, 1 for secondary)
```

Avoid picking random MAC addresses that could be reserved or multicast.

Prepare your port-channel for VLTi (interconnect) configuration .To become a VLTi the port-channel should be in default mode (no switchport).

Note: The system will automatically include needed VLANs to be tagged into the VLTi. You do not need to manually tag VLANs on the VLTi.

Configure the core VLT peering relationship across the port-channel that will become the VLT interconnect (VLTi)

Note: it is recommended to build the VLTi port-channel statically to minimize negotiations in the VLT domain core.

```
(conf-vlt-domain)# peer-link port-channel <LAG-ID>
```

Configure the VLT backup link (used for health checks)

```
(conf-vlt-domain)# back-up destination <ip-address>
```

The backup link should be a different link than the VLTi and if possible following a diverse path. This could be the management interface IP address.

Note: It is recommended that VLTs that are facing hosts/switches should be preferably built by LACP, to benefit from the protocol negotiations. However static port-channels are also supported

```
interface port-channel <id-number>
vlt-peer-lag port-channel <id-number>
```

Table 1. Sample VLT configuration on peer1

```
Force10_VLTpeer1(conf)#vlt domain 999
Force10_VLTpeer1(conf-vlt-domain)#peer-link port-channel 100
Force10_VLTpeer1(conf-vlt-domain)#back-up destination 10.11.206.35
Force10_VLTpeer1(conf-vlt-domain)#exit

Force10_VLTpeer1(conf)#interface ManagementEthernet 0/0
Force10_VLTpeer1(conf-if-ma-0/0)#ip address 10.11.206.23/16
Force10_VLTpeer1(conf-if-ma-0/0)#no shutdown
Force10_VLTpeer1(conf-if-ma-0/0)#exit

Force10_VLTpeer1(conf)#interface port-channel 100
Force10_VLTpeer1(conf-if-po-100)#no ip address
Force10_VLTpeer1(conf-if-po-100)#channel-member fortyGigE 0/56,60
Force10_VLTpeer1(conf-if-po-100)#no shutdown
Force10_VLTpeer1(conf-if-po-100)#exit

Force10_VLTpeer1(conf)#interface port-channel 110
Force10_VLTpeer1(conf-if-po-110)#no ip address
Force10_VLTpeer1(conf-if-po-110)#switchport
Force10_VLTpeer1(conf-if-po-110)#channel-member fortyGigE 0/52
Force10_VLTpeer1(conf-if-po-110)#no shutdown
Force10_VLTpeer1(conf-if-po-110)#vlt-peer-lag port-channel 110
Force10_VLTpeer1(conf-if-po-110)#end

Force10_VLTpeer1# show vlan id 10
Codes: * - Default VLAN, G - GVRP VLANs, P - Primary, C - Community, I - Isolated
       Q: U - Untagged, T - Tagged
          x - Dot1x untagged, X - Dot1x tagged
          G - GVRP tagged, M - Vlan-stack, H - Hyperpull tagged

      NUM      Status      Description                               Q Ports
      10      Active
                                     U Po110(Fo 0/52)
                                     T Po100(Fo 0/56,60)
```