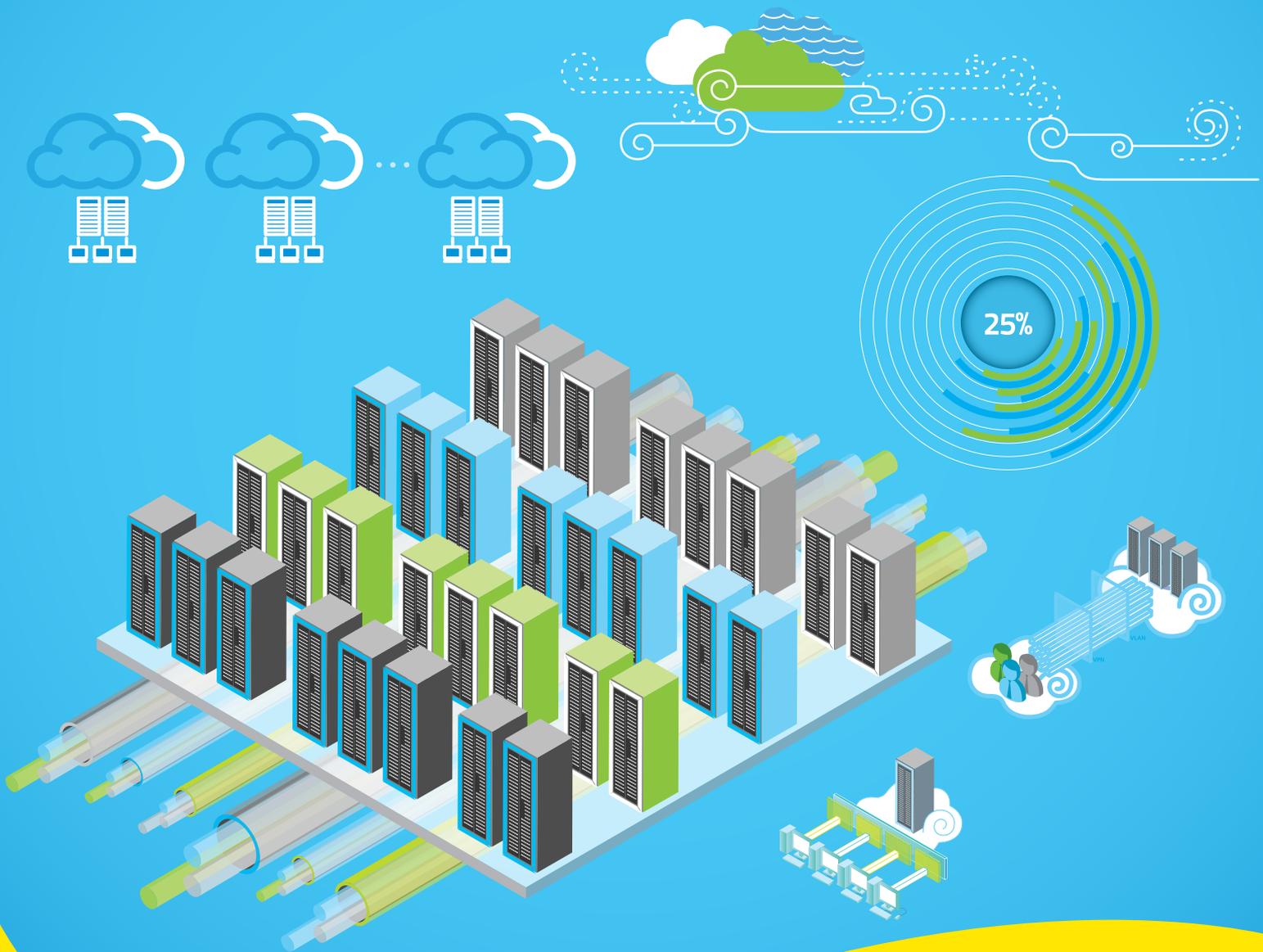


Software Defined Data Center – An Implementation view



Contents

SDDC - Introduction.....	3
SDDC – Architectural Overview.....	3
Software Defined Storage.....	4
Sample Implementation.....	5
Software Defined Networking.....	6
Some Other Thoughts.....	7
Testing SDDC – Automation Framework.....	10
Conclusion.....	11



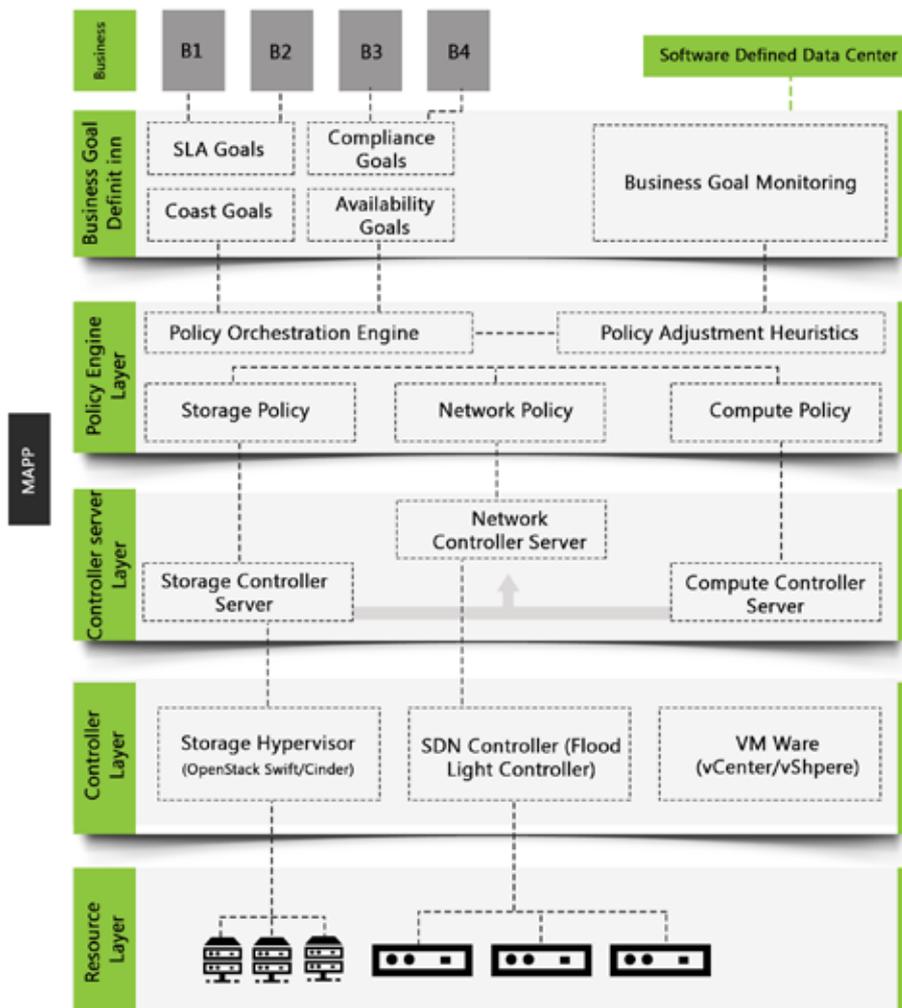
SDDC - Introduction

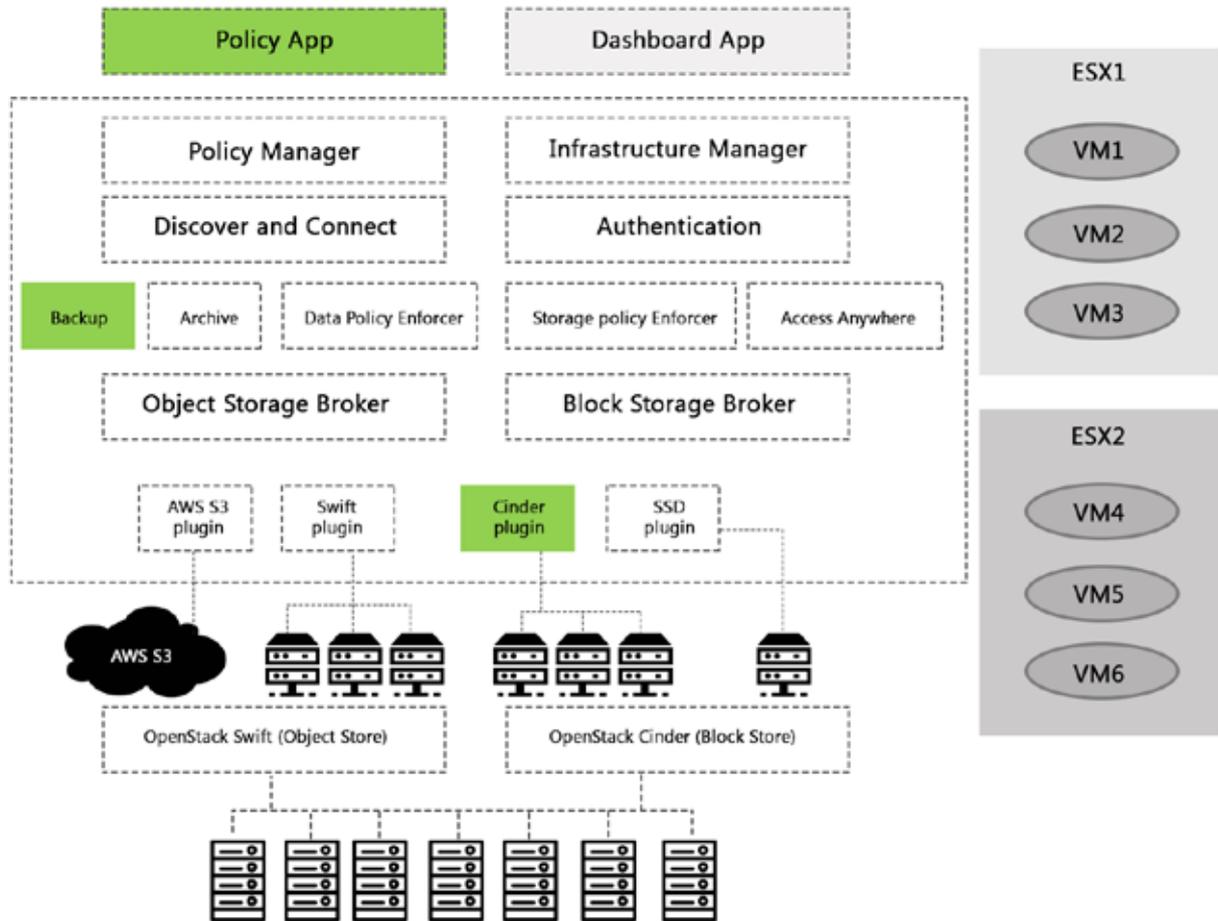
The pace of IT change is increasing remarkably. Mobility, cloud computing, social media, consumerization, big data and analytics have all become deeply interwoven within today's enterprise. The result is a paradigm shift in the way organizations use and consume IT resources. Today, new and dynamic service delivery models promise to revolutionize business and transform IT.

Software Defined Data Center (SDDC) is the next evolution of the underlying technology, where software delivers greater levels of intelligence and value, on top of standardized hardware. Over the last few years, more and more corporate executives have been evaluating the possible benefits of migrating to a Software Defined Data Center (hereafter referred to as SDDC). Some early adopters have moved beyond the evaluation stage to begin planning and even get into implementation.

The way we have created our understanding of Software Defined Data Center is by taking into account the network, storage and compute aspects of a data center and viewing the whole system together. We have, then gone into two aspects of SDDC – Software Defined Networking (SDN) and Software Defined Storage (SDS). The rest of the document describes our view of these aspects.

SDDC – Architectural Overview





- **Policy Manager:** It stores the policies and manages the changes to the policies. There are two levels of policies:
 - Data Policy
 - Storage Policy

The Data Policy can be maintained at 3 levels: User Level, Application Level and VM Level.

- **User Level Policies:** The policies are stored per user. When a policy is created, it can be associated with any user in the active directory. This association will be done by the admin. A representational set of parameters will be:

- User Directories to be backed up
- Destination for the backup
- Frequency of backup

Retention period post deletion of the original

Number of versions to be maintained

- **Application Level Policy:** These policies are stored per application like exchange, database, CRM etc. When policy is created, the associated VMs running the application is associated with the policy. All the VMs associated with the application will be protected by the policy. This association is done by the admin. A representational set of parameters will be:

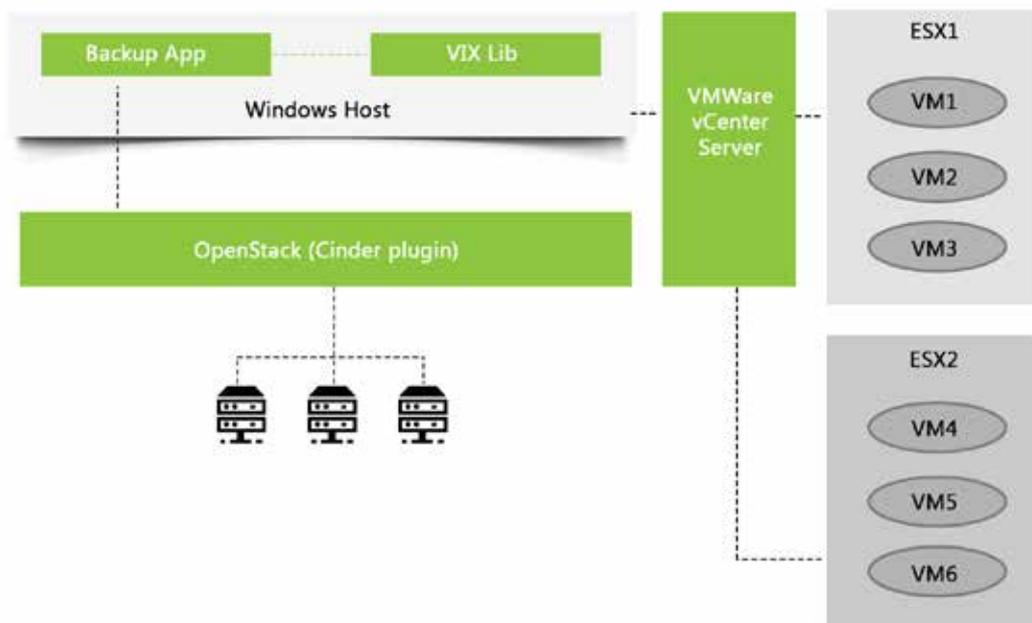
- Destination of VM Backup
- Disaster recovery flag
- Frequency of backup
- Versions of backups to be retained

- **VM Level Policy:** These policies are stored per VM. These are created when the VM is created and is associated per VM. There can be multiple parameters for this policy but there will be some protection specific parameters. Some representational parameters would be:
 - Frequency of backup
 - Data protection vs VM protection
 - DR flag (Last backup creates a new VM if the original goes down)
 - Versions of backup to be retained

The Storage Policy is represented at 2 levels: Hierarchical Storage Policy and Archiving Policy.

- **Hierarchical Storage Policy:** It is maintained to ensure automatic data movement to comply to cost and response time SLAs. Depending on these, the data will move from local store of the VM to OpenStack storage. The applications should access the storage as if nothing has moved. The policy will define how the data will move from SSD to Open Stack Storage for live access and how data will move from OpenStack to S3 for Object access. This is more like hierarchical storage management.
- **Archival Policy:** It defines what data needs to be archived at user level or application level. An archive destination is created in the OpenStack storage per user and per application. For example, archive aware applications can have the link to all archived data. Else the data is stored on a separate storage and accessed separately. The user can search the archive for specific data when required.

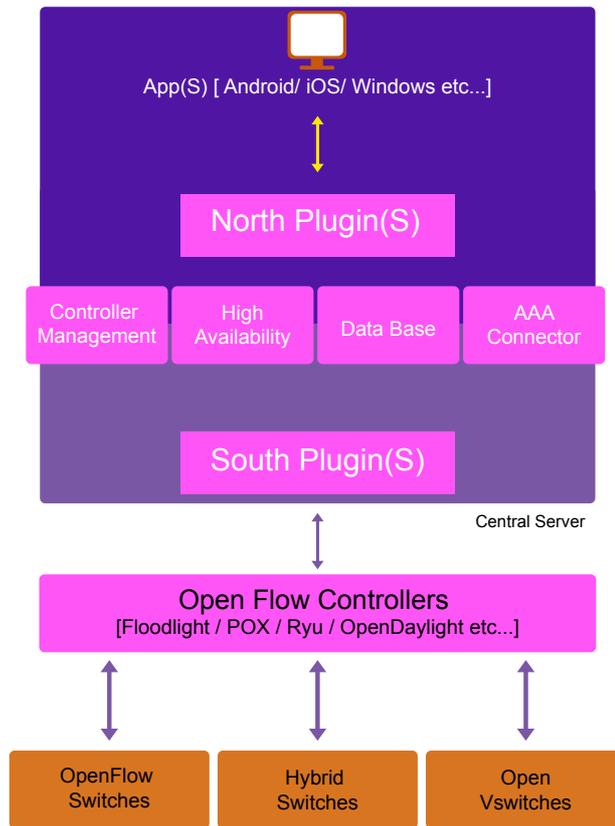
Sample Implementation



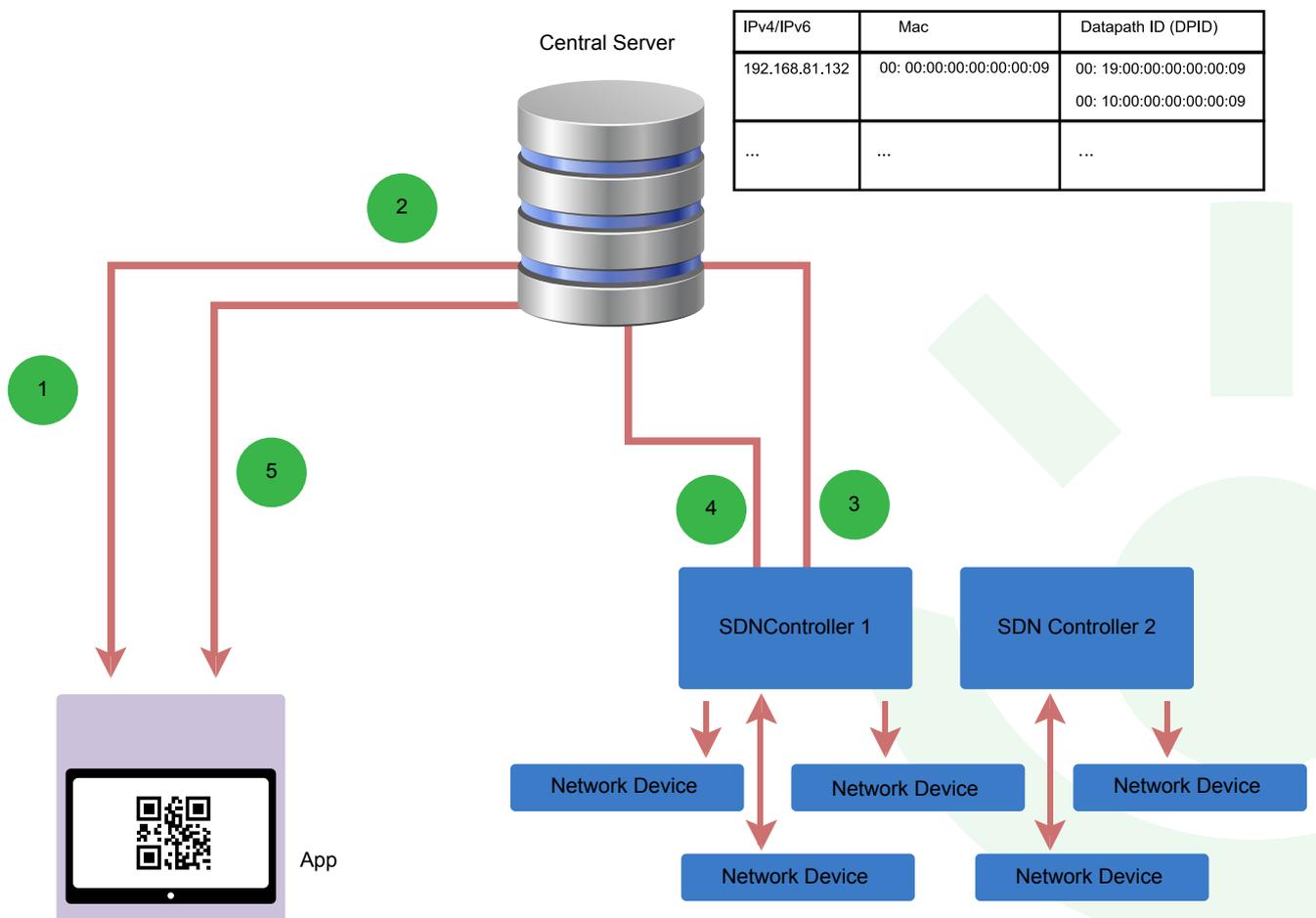
An application has been developed to allow backup of data in a datacenter. The data specific to the VMs could be backed up without the need to install an agent on these VMs. Features implemented in Backup application:

- List of all the VMs along with their properties
- Browsing of all the files and directories in a selected VM
- Backup of selected files and directories to a local storage or to the storage exposed by OpenStack

Software Defined Networking

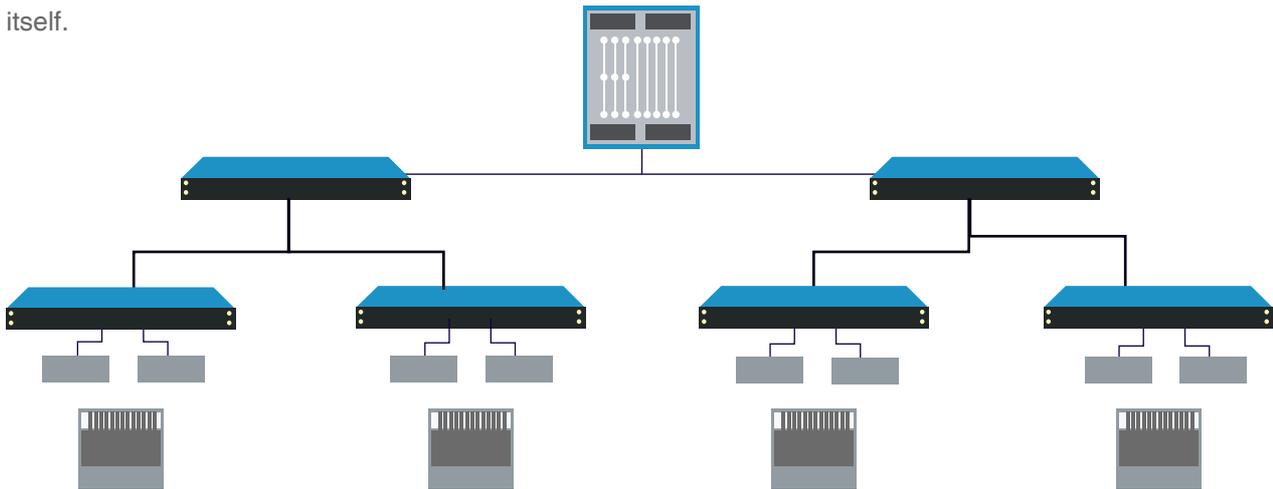


Sample Implementation



- Field engineer scans the QR Code and the MAC address is sent to the Central Server
- The Hash Table maintained by the Central Server is scanned for the IP address of the SDN Controller, with received MAC address as the key
- Central server establishes connection with the retrieved IP address of the controller and sends the query received from the app to the SDN controller
- The SDN Controller searches for the network element information maintained by it and send back the queried result to the controller
- The Central Server sends back the result to the app and the app displays all the queried information on the screen.

Using the retrieved information, field engineer can either decide to debug (or) configure the network element from the app itself.



The above topology of the network is displayed on the app on the field engineer's tablet with the scanned network element highlighted. The field Engineer can further click on the network element for further process he intends to perform such as debugging/Configuring the network element. This is just an illustration of app features. The requirements can be designed according to one's requirements.

Some other thoughts

Bandwidth on Demand Services

In a typical network of a service provider, the traffic consists of flows from multiple applications and utilities from different classes of customer. Many of these applications are unique and have their own requirements with respect to different network parameters such as bandwidth requirement, delay, jitter, etc. which will affect the quality and usability of this application, which in turn affects the user experience.

Issue with static bandwidth allocation:

- Available bandwidth is limited and insufficient to meet the traffic demand.
- Traffic requirement differs based on different users.
- Traffic requirement differs based on different time zones.
- Bandwidth wastage due to static allocation.
- Bandwidth is used by unwanted applications.

Thus, traffic management is the key to properly prioritize different applications across the limited bandwidth and ensure that these requirements are met as per the agreed user plan.

Dynamic Bandwidth Allocation as a Solution

To effectively use dynamic bandwidth allocation using DPI and SDN to achieve improved network, performance should result in optimal user experience and improved network resource utilization.

What is Dynamic Bandwidth Allocation?

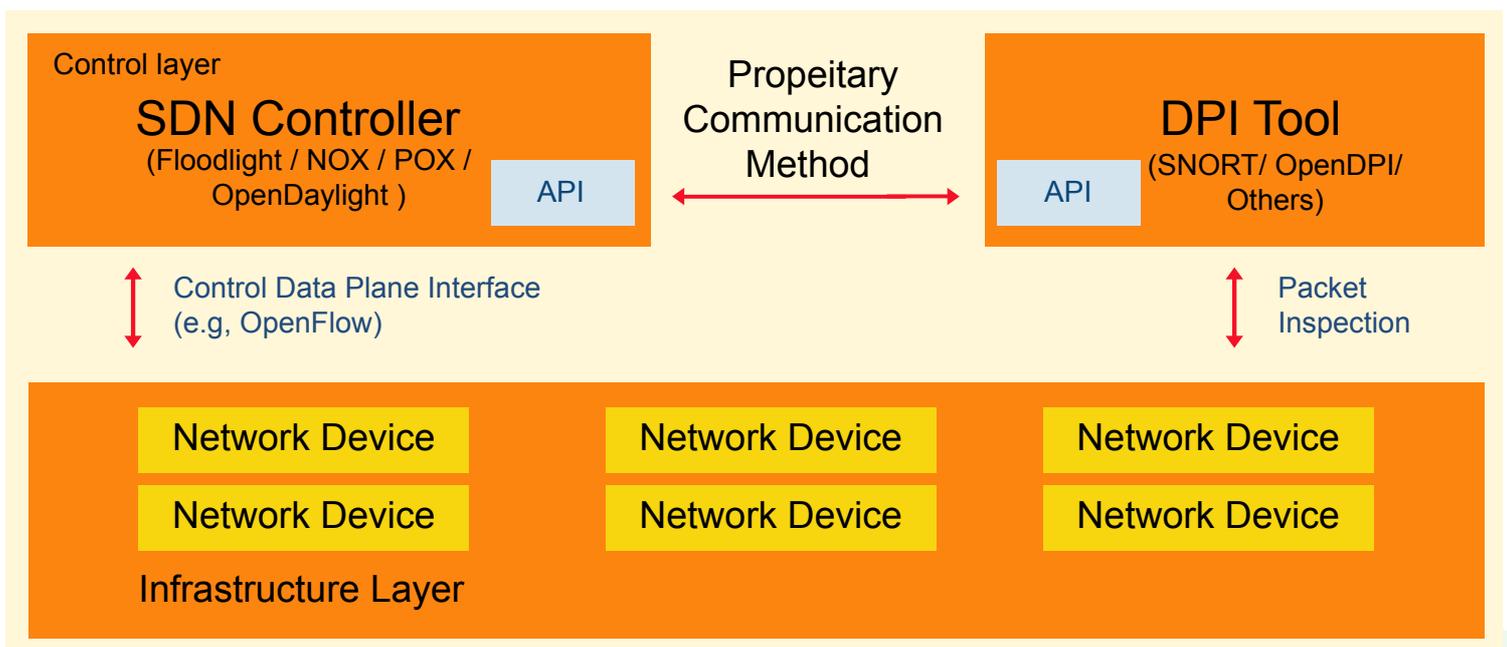
DBA is to allocate available bandwidth dynamically in an optimized way as per the user application as and when the need arises, based on the user subscription plan using different traffic management techniques.

Traffic management combines various actions such as monitoring, discovery, control, classification and optimization of the traffic based on the user behavior and available bandwidth to improve the network performance, which intern will enhance the user experience.

Traffic can be broadly classified as follows:

- Payload based classification: based on the payload type.
- Traffic behavior based: based on a statistical method that uses statistical analysis of the traffic behavior like inter-packet arrival, session time and so on.

We can achieve traffic management of the network, based on the above said classifications using DPI and SDN respectively, So that we can allocate bandwidth dynamically to meet the customer demand

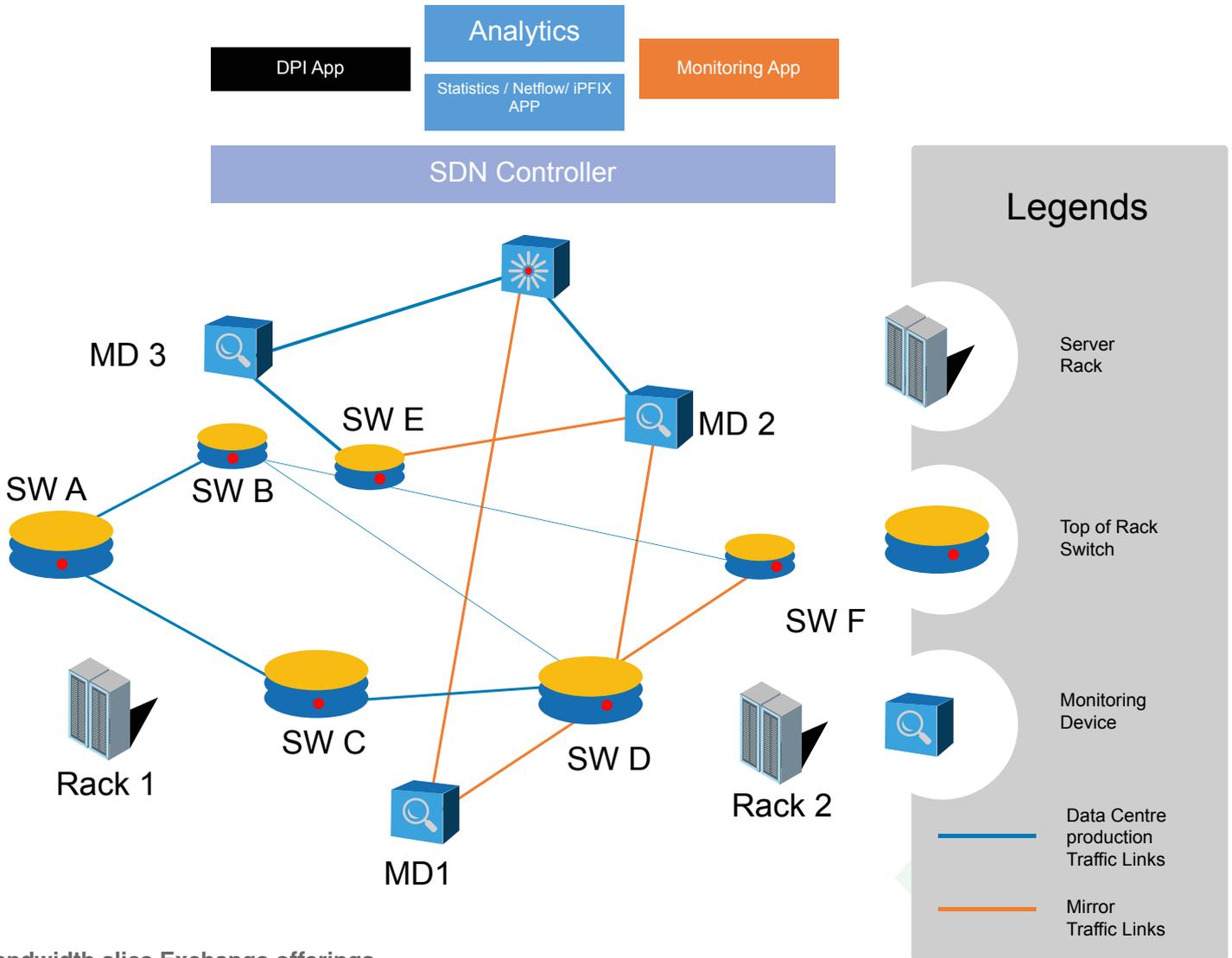


Network Monitoring is a critical function in a data center. From doing recording of VOIP calls, performance audits, and applying forensics – the use of Networking Monitoring are varied.

The modern day data center throws newer challenges for monitoring systems – like virtualized workflow, emergence of 10, 40 and 100G fabrics, and a very high East West traffic flow and thus forcing investment of expensive Network Packet Brokers in each segment in a distributed approach or compromising the available bandwidth in an aggregated approach.

With the advent of Software Defined Networking, there are possible solutions and techniques, which can be used to face and overcome those challenges with dynamic programming of network and leveraging the SDN controller intelligence and cost effective bare metal switches to offer a robust and cost effective solution for Monitoring. Not only that, the power of SDN offers additional features which enable reduction in the replicated traffic, distributed monitoring functionality and very granular access control.

Further, the availability of Big Data platforms and Analytics further help in applying predictive orchestration of the monitoring solutions, thus optimizing scaling needs and at the same time giving accurate visibility to the administrator



Bandwidth slice Exchange offerings

It is difficult to determine the traffic usage patterns and hence there would be scenarios where a client would need access bandwidth, over and above the committed bandwidth from the service provider.

With SDN, it is possible to create network slices and the same slices can be exchanged in real time. The network slices can be created with the help of the SDN controller, which allocates the network resources at a granular level of a flow, and thus groups certain flows into a "Slice".

The SDN controller also exposes REST API based northbound interface, which can provide information to Billing and exchange applications which can facilitate the bandwidth exchange in real time.

Testing SDDC – Automation Framework

Storage Management Applications(UI) automation Selenium++

Storage Management Applications(API) automation using SOAPUI, RestAssured, Junit, TestNG based Frameworks

Compute
Virt-test-api(KVM)
Test-api(VMware_server_hyper-v)

Switch
Virt-test-api (OpenSwitch)
test-api(Cisco, brocade)

Storage
test-api(OpenStack)
test-api(NetApp, EMC, Hitachi)

API's for combined Compute, Switch & Storage
[Flexipad(VMware, Cisco, NetApp)]
[vblock(VMware, Cisco, NetApp)]
[vspex(VMware/Citrix, VMware/Microsoft, Cisco/Dell/HP/IBM, Cisco/brocade)]

IO Tools to simulate different traffic

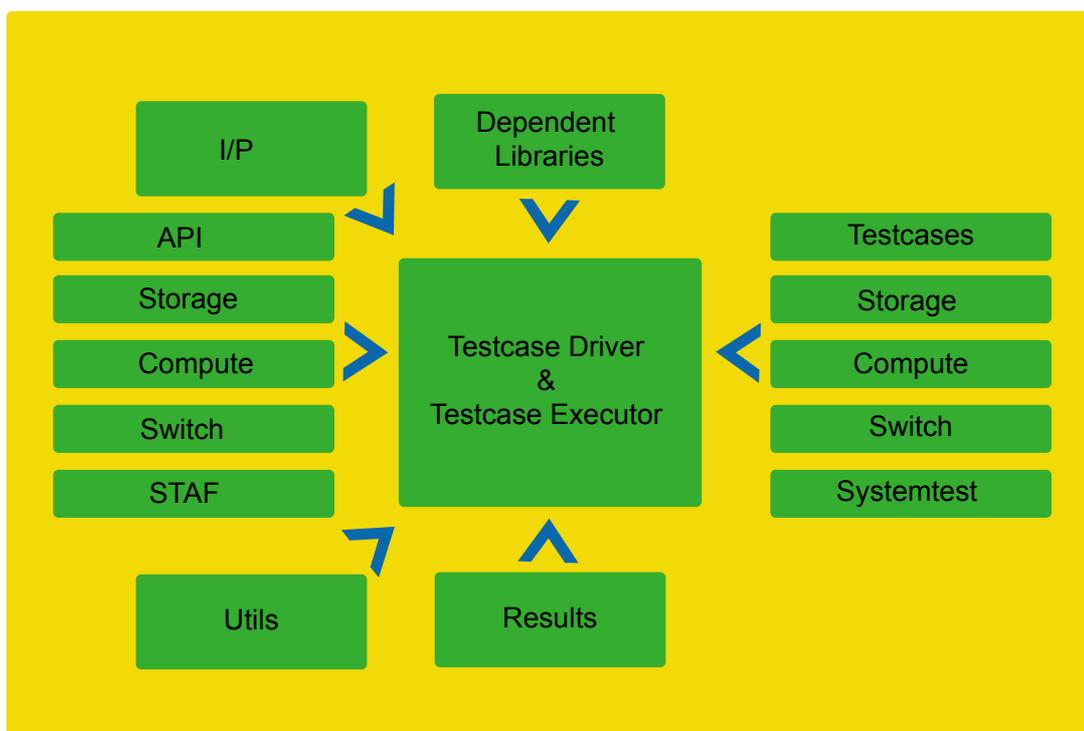
Autotest

Framework & Tools

- Open Source Framework AUTOTEST
- Selenium++ Automation framework with customized reporting
- Unit/API testing framework
- Test Data Generation, Logs Analyzers

Sample Implementation

UI to Trigger Automation and Display Results



Windows + iSCSI
IOZONE

Sun ZFS + iSCSI



Linux + NFS client
FSSTRESS

Sun ZFS + NFSv4



Windows + CIFS
IO Meter

Sun ZFS + SMB

The above three scenarios have been implemented and tested with the following steps:

- Create a Share(Lun-SAN/export-CiFS or NFS)
- Map a share to Windows/Linux Machine
- Run the test cases using the IO Tools
- Create a snapshot of the volume
- Create a mirror of the volume
- Create an archive of the Volume
- Restore a snapshot of the volume
- Break the mirror relation
- Restore the broken mirror relation
- Capture the results.

Conclusion

Happiest Minds has a focus on SDDC in terms of SDS, SDN and SDDC automation. We have been developing certain components to demonstrate our understanding and showcasing capabilities. Coupled with this, we have an ecosystem to help our enterprise customers to consult, implement and manage the data center specific solutions. This gives us a firsthand understanding of the challenges in the field, thus helping us to in providing a point of view on product features which are being developed. We look forward to working with product companies to help them succeed in their vision of SDDC.

About the Author



Jagadeesh CV

Strategic, focused, and business oriented enterprise, Solution architect, Pre-sales, Post-sales and Practitioner with 15 years of experience in information technologies. Master of establishing productive relationships with partners and customers. Data Center Enterprise Technology Roadmaps, Transformation Strategies, Private Cloud Implementation, Public, Private Cloud Integration, Cloud Computing adoption Roadmap for Enterprises, ROI models, Performance Engineering, Infrastructure Consolidation and Optimization, Enterprise Social Collaboration, Vendor Management, Evangelism.

Happiest Minds

About Happiest Minds Technologies:

Happiest Minds, the Mindful IT Company, applies agile methodologies to enable digital transformation for enterprises and technology providers by delivering seamless customer experience, business efficiency and actionable insights. We leverage a spectrum of disruptive technologies such as: Big Data Analytics, AI & Cognitive Computing, Internet of Things, Cloud, Security, **SDN-NFV**, RPA, Blockchain, etc. Positioned as “Born Digital . Born Agile”, our capabilities spans across **product engineering**, digital business solutions, **infrastructure management** and security services. We deliver these services across industry sectors such as retail, consumer packaged goods, edutech, e-commerce, banking, insurance, hi-tech, engineering R&D, manufacturing, automotive and travel/transportation/hospitality.

Headquartered in Bangalore, India; Happiest Minds has operations in USA, UK, The Netherlands, Australia and Middle East.

© Happiest Minds. All Rights Reserved.

Email: business@happiestminds.com

Visit Us: www.happiestminds.com

Follow us on

