WIND™

NFV: The Myth of Application-Level High Availability

WHEN IT MATTERS, IT RUNS ON WIND RIVER

EXECUTIVE SUMMARY

Telecom operators want to offer their customers services that perform as expected when the service is requested. In fact, detailed service level agreements (SLAs) make it a business imperative to do so. But achieving a high degree of service availability is difficult and challenging, requiring significant investment and a rigorous, methodical attention to detail.

This paper serves as a primer for a best-practices solution for high availability (HA). Using relevant industry references as a guideline, this paper details why cutting corners is a bad idea, and offers suggestions on a more comprehensive way to achieve the high degree of service ability with Network Functions Virtualization (NFV).

TABLE OF CONTENTS

INTRODUCTION

There is a growing misconception in the software industry that high reliability can be achieved entirely through application-level redundancy schemes (load balancing, check pointing, journaling, etc.). But while such methods provide a degree of protection from certain failure scenarios, application-level solutions alone are insufficient to meet the demanding high availability expectations of the competitive, SLA-driven telecommunications market. Simple, stateless applications (e.g., many web servers) may benefit from simple approaches to availability, but modern, stateful services require more comprehensive frameworks.

There are numerous problems and failures that network service providers can expect to face, and no single "one size fits all" approach can easily address all of them. A holistic, multilayered solution is required.

TERMS AND DEFINITIONS

To avoid potential confusion or ambiguities in definition, we will begin by covering the meaning of key terms and phrases typically used in the study of mission-critical systems. For the purposes of this paper, the following industry-standard definitions have been adopted:

- **Availability**: The availability of an item to be in a state to perform a required function at a given instant of time or at any instant of time within a given time interval, assuming that the external resources, if required, are provided[1]
- **Reliability:** The probability that an item can perform a required function under stated conditions for a given time interval[1]
- **Fault coverage:** The proportion of the system's failure rate that is successfully detected and recovered[2]
- **Resiliency:** The ability of the NFV framework to limit disruption and return to normal or, at a minimum, acceptable service delivery level in the face of a fault, failure, or an event that disrupts the normal operation[3]

AVAILABILITY CHALLENGES

Customer-facing, end user applications in modern networks are not monolithic, single-layered entities. Rather, they are composed of a series of carefully architected and managed components, working in close harmony with one another. This is especially true with regard to applications deployed in an NFV environment. Whenever the operation of one component within that environment (a virtual network function [VNF]; the NFV infrastructure [NFVI]; a virtualization infrastructure manager [VIM], etc.) is disturbed in some way, there is a potential for a customer-visible service impact. The number of customers impacted and the duration of that impact are key variables that both solution providers and operators seek to minimize.

The following table provides a few examples of the types of problems that can occur during system operation.

Table 1: Disruptive Events

Class of Event	Possible Impact	Impact Severity
Management system failure	Loss of application oversight	Low
Single server or network node failure	Loss of specific applications or services	Low-medium (assuming some degree of redundancy)
Network link failure	Service disruption	Low-medium (assuming some degree of redundancy)
Server maintenance—software patch or upgrade	Service disruption	Low-medium (assuming in-service software maintenance supported)
Application software error or crash	Service disruption or outage	Medium-high
Malicious network attack	Service disruption or outage	Medium-high
Network congestion or overload	Service disruption or outage	Medium-high
Catastrophic loss of site or point of presence	Loss of all underlying service provided by site	Highest

This list may be overly simplified, but it demonstrates a need for some form of failure mitigation to limit—or, preferably, eliminate—service impacts when failures occur. Unsurprisingly, industry evidence reveals that service impacts are highly costly, both in terms of lost productivity and revenue, and in terms of operator reputation.

According to a 2013 Heavy Reading study,[5] service providers are spending $15 billion per year dealing with network outages. Further, such events are the third largest cause of subscriber churn. Clearly, all possible efforts should be made to prevent outages.

Failure mitigation through application action is one technique commonly used to limit the impact of failures, and it does provide some measure of resiliency when a select failure occurs. A simple example of such mitigation is the traditional application-level "active/standby" model.

With this model, there are two instances of the application (or VNF, in the case of NFV): one that is actively providing service, and one that is not providing service, but is able to do so rapidly should the active, serving instance fail.

Such a model makes one critical assumption, one which may not always be accurate—that the active and standby instances are each hosted on different physical infrastructures, i.e. they are not both hosted on the same server.

What if the underlying platform on which the application is running (e.g., OpenStack) has no knowledge of the fact that there is an active and a standby instance of the application, and proceeds to deploy both the active and the standby on the same physical server? Should that physical server suffer a sudden outage (e.g., power failure), then both instances of the application are instantly lost, and a customer-visible outage will almost certainly follow.

While this set of circumstances may seem unlikely, a well-known network operator related just such an incident to the author of this paper during a conversation at an industry event in Europe in 2014. In that case, a critical network function was deployed on an enterprise cloud implementation. Both instances (active and standby) were on a shared physical server. A single fault on that server resulted in a complete outage of the application and the service it provided, with the predictable unhappy consequences for the operator.

To avoid such scenarios, system-level protection mechanisms are required. To entrench this thinking, the ETSI NFV Expert Group on Availability and Resiliency stipulated the following requirement in its comprehensive NFV Specification:

> [Req.9.5.2] The VNFs with the same functionality should be deployed in independent NFVI fault domains to prevent a single point of failure for the VNF. In order to support disaster recovery for a certain critical functionality, the NFVI resources needed by the VNF should be located in different geographic locations; therefore, the implementation of NFV should allow a geographically redundant deployment.[6]

Underscoring the importance of this requirement, the same underlying need is repeated elsewhere in the same specification:

> It shall be assured that mechanisms which contribute to reliability and availability have the same effect when virtualised. For instance, the availability gained by running two server instances in a load sharing cluster can only be preserved if the virtualisation layer runs the two virtualised instances on two unique underlying host server instances (i.e. anti-affinity).

Despite good intentions, this example clearly demonstrates that application-level HA solutions alone are insufficient to protect the system in all circumstances; additional protection schemes are necessary.

PREVENTION

Another important method of maximizing availability is to proactively seek out and identify faults that can occur within a system. Once identified, recovery and self-healing mechanisms can be initiated, thereby preventing small problems from growing unchecked into large problems. For this reason, it is important to deploy a system with a broad, comprehensive fault coverage framework. To state the obvious, every component within a system contributes to faults, not just the application layer. Network interface cards, hard drives, cooling units, power supplies, data base management systems, critical operating system processes—all experience faults that may ultimately result in service outages if left unchecked.

What guidance does the ETSI NFV Resiliency Specification provide on this topic in relation to NFV?

> The basis for a resilient system is a set of mechanisms that reduce the probability of a fault leading to a failure (fault tolerance) and reduce the impact of an adverse event on service delivery.[4]

In other words, if the end goal is a resilient system, multiple mechanisms are required to meet that expectation. While some faults may be detected at an application level, applications running in a virtualized, NFV environment are simply unable to detect all problems themselves. If there are no other protection mechanisms in place, availability will suffer.

A more comprehensive, multidimensional solution is needed; one in which application-layer protections are reinforced by protections in other layers.

LAYERED APPROACH

Figure 1 depicts one of the standard NFV architectural references published by ETSI.[7] Among the main functional blocks listed are the NFVI, the NFV Management and Orchestration block (commonly referred to as MANO), and the block containing the VNFs themselves, with their respective element managers (EMs). Each of these blocks has distinct responsibilities and interfaces.

Among the responsibilities of each of these blocks are various forms of fault detection, reporting, and remediation. In fact, the ETSI NFV Expert Group on Availability and Resiliency devoted an entire chapter to this important topic, "Failure Detection and Remediation."[8] Areas covered include the role of the hypervisor in detecting hardware faults; the role of the VIM in detecting NFVI faults; the role of MANO in detecting VIM faults; and of course the role of the VNF in detecting its own faults (application-level HA).

Further, there is even a section detailing how "liveness checking"[9] can be implemented to "detect and react to failures immediately" vs. waiting for an application to detect a problem. In total, this chapter contains an impressive 25 requirements on the subject. The message from ETSI NFV is clear: a comprehensive, layer-by-layer structured approach is required to achieve the HA and resiliency targets expected by operators.

The Wind River® Titanium Cloud NFVI software platform was purpose-built to achieve these goals. Titanium Cloud implements a multilayered, proactive fault detection and recovery system.

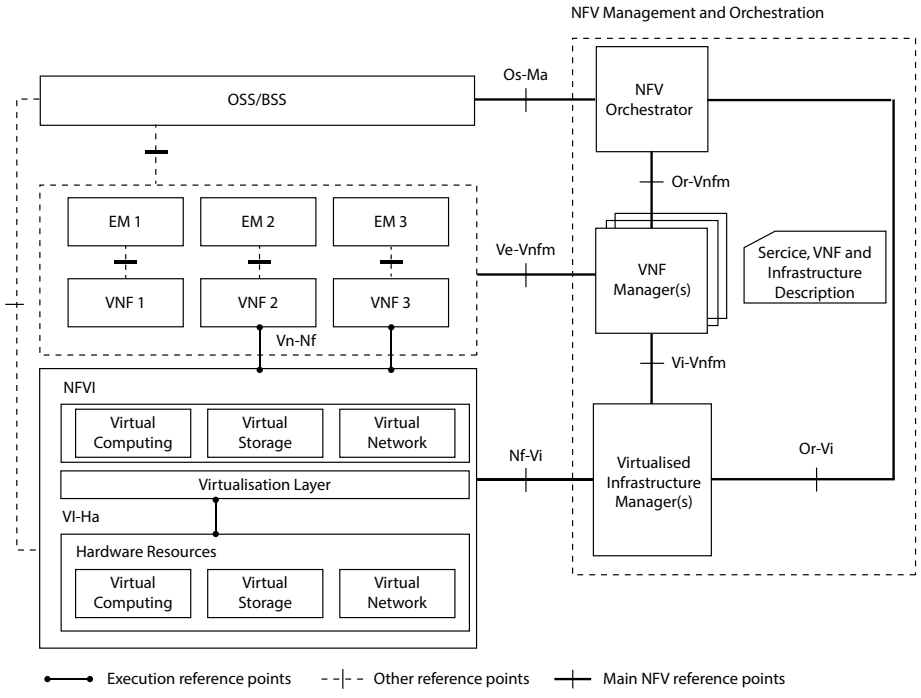

Figure 1: NFV architectural framework

Through policies and metadata, it understands and enforces application deployment models; it constantly monitors system operations at the NFVI, VIM, VNF, and VNFM levels. Upon failure detection, Titanium Cloud autonomously reacts, taking self-healing and service preservation actions. Standard telco alarms are generated, informing operators of system status and remediation in progress or completed. Performance metrics are pegged and available for off-board retrieval and analysis, if so desired.

CONCLUSION

By implementing a layered approach to HA and following recommendations from ETSI, Titanium Cloud has set a high bar with a platform availability target of six nines (99.9999%), or no more than 30 seconds planned plus unplanned downtime per year. This enables services deploying on Titanium Cloud to achieve five nines (99.999%) availability, or no more than five minutes downtime per year (planned plus unplanned). Achieving such targets without a comprehensive, multilayered approach to HA is not possible—it's simply a myth.

REFERENCES

1. *"Network Functions Virtualisation (NFV); Resiliency Requirements," ETSI GS NFV-REL 001 V1.1.1, page 9.*
2. *"Wind River Titanium Server Reliability, Availability, Maintainability (RAM) Modeling Analysis," Version 4.0, KerNett Consulting Inc., page 7.*
3. *"Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV," ETSI GS NFV 003 V1.2.1, page 8.*
4. *"Network Functions Virtualisation (NFV); Resiliency Requirements," ETSI GS NFV-REL 001 V1.1.1, page 33.*
5. *Heavy Reading, Mobile Network Outages and Degradations, October 2013.*
6. *"Network Functions Virtualisation (NFV); Resiliency Requirements," ETSI GS NFV-REL 001 V1.1.1, page 42.*
7. *"Network Functions Virtualisation (NFV); Architectural Framework," ETSI GS NFV 002 V1.2.1, page 14.*
8. *"Network Functions Virtualisation (NFV); Resiliency Requirements," ETSI GS NFV-REL 001 V1.1.1, page 42.*
9. *"Network Functions Virtualisation (NFV); Resiliency Requirements," ETSI GS NFV-REL 001 V1.1.1, page 46.*

WIND

Wind River is a global leader in delivering software for the Internet of Things. The company's technology is found in more than 2 billion devices, backed by world-class professional services and customer support. Wind River delivers the software and expertise that enable the innovation and deployment of safe, secure, and reliable intelligent systems.