ETSI GS NFV-IFA 004 V2.3.1 (2017-08)

Network Functions Virtualisation (NFV) Release 2;
Acceleration Technologies;
Management Aspects Specification

Disclaimer

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
RGS/NFV-IFA004ed231

Keywords
acceleration, management, NFV, NFVI, virtualisation

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies functional requirements for both the Virtualised Infrastructure Manager (VIM) and the NFV Infrastructure (NFVI), for NFV acceleration from an infrastructure management perspective. This includes the controlling and management of acceleration resources, e.g. allocation, release and discovery of acceleration resources. The present document also identifies the corresponding impacts on VIM related specifications regarding functional requirements ETSI GS NFV-IFA 010 [4] and reference points (ETSI GS NFV-IFA 005 [2] and ETSI GS NFV-IFA 006 [3]).

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for main concepts in NFV".

[2] ETSI GS NFV-IFA 005: "Network Functions Virtualisation (NFV); Management and Orchestration; Or-Vi reference point - Interface and Information Model Specification".

[3] ETSI GS NFV-IFA 006: "Network Functions Virtualisation (NFV); Management and Orchestration; Vi-Vnfm reference point - Interface and Information Model Specification".

[4] ETSI GS NFV-IFA 010: "Network Functions Virtualisation (NFV); Management and Orchestration; Functional Requirements Specification".

[5] ETSI GS NFV-IFA 002: "Network Functions Virtualisation (NFV); Acceleration Technologies; VNF Interfaces Specification".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] Recommendation ITU-T Y.3300 (2014-06): "Framework of software-defined networking".

[i.2] ETSI GS NFV-INF 005: "Network Functions Virtualisation (NFV); Infrastructure; Network Domain".

[i.3] ETSI GS NFV-MAN 001: "Network Functions Virtualisation (NFV); Management and Orchestration".

[i.4] OpenStack Project.

NOTE: Available at http://www.openstack.org/.

[i.5] ETSI GS NFV-EVE 005: "Network Functions Virtualisation (NFV); Ecosystem; Report on SDN Usage in NFV Architectural Framework".

[i.6] ETSI GS NFV-IFA 001: "Network Functions Virtualisation (NFV); Acceleration Technologies; Report on Acceleration Technologies & Use Cases".

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETSI GS NFV 003 [1] and the following apply:

acceleration capability: ability provided by NFV-Resources to improve processing speed or capacity

NOTE: An acceleration capability can be exposed in an implementation independent manner.

EXAMPLE: TCP Checksum calculation, packet dispatching amongst queues, TCP Offload, IPSec Offload and RDMA are examples of acceleration capabilities provided by a NIC. Encryption and compression are examples of acceleration capabilities provided by a disk controller.

acceleration feature discovery: process of either reactively reporting or proactively registering certain features of given sets of acceleration resources to an acceleration management function

acceleration function chaining: grouping of a set of acceleration resources to provide high performance for mission-critical NFV services

NOTE: The operation of acceleration function chaining should be done within single NFVI-Node.

acceleration lifecycle management: management functionality provided by the acceleration management function to perform allocation, release or modification of given sets of acceleration resources

acceleration management function: function that provides management capability for acceleration resources

NOTE: The acceleration management function is a subset of acceleration related functionalities provided by VIM.

acceleration resource: NFV-Resource (hardware and/or software) that provides one or multiple acceleration capabilities

EXAMPLE: GPUs, video transcoding cards, NICs with TCP offload engine, crypto accelerator cards.

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI GS NFV 003 [1] and the following apply:

SDN Software Defined Networking
VNFCI VNF Component Instance

4 Overview

4.0 Introduction

NFV Acceleration deals with both hardware and software accelerations. Hardware acceleration is the use of specialized hardware (referred in the present document as hardware accelerator) to perform some function faster than is possible by executing the same function on a general-purpose CPU or on a traditional networking (or other I/O) device (e.g. NIC, switch, storage controller, etc.). On the other hand, software acceleration provides a set of one or more optional software layers (referred in the following as software accelerators) that are selectively added within elements of an NFV deployment (e.g. compute, hypervisor, VNF, etc.) to augment or bypass native software within a solution.

With the acceleration abstraction layer defined, hardware accelerators as well as software accelerators can be abstracted as a set of acceleration functions (or acceleration capabilities) which exposes a common API to either the VNF or the host.

NFV acceleration can be deployed according to different models as shown in the following figures 4.0-1 and 4.0-2. Figure 4.0-1 shows a disaggregated deployment model, i.e. a deployment where the NFV Accelerator is either physically decoupled from a specific host, and shared between two or more NFVI compute or networking resources, or pluggable software which could be shared between VNFs, figure 4-2 illustrates an integrated deployment model, i.e. a deployment where the NFV accelerators are packaged within a specific VNF, or physically integrated within a specific host.

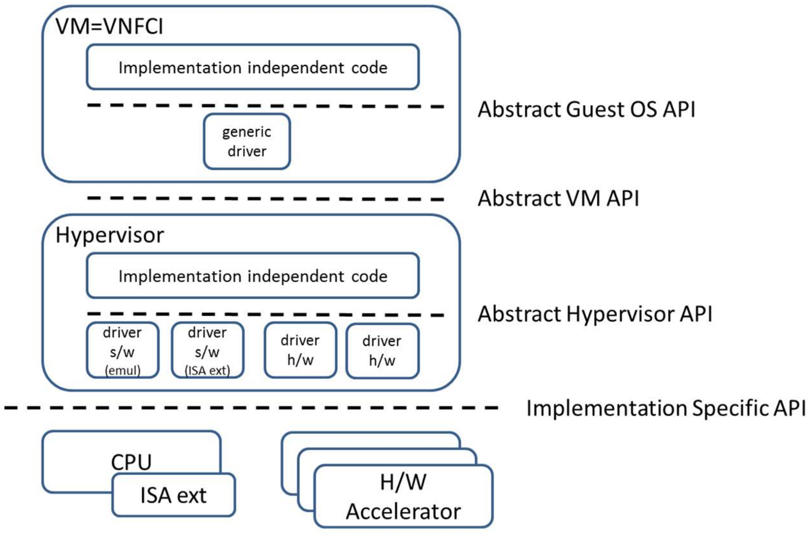

Figure 4.0-1: NFV acceleration abstraction layer implemented within hypervisor as a disaggregated model

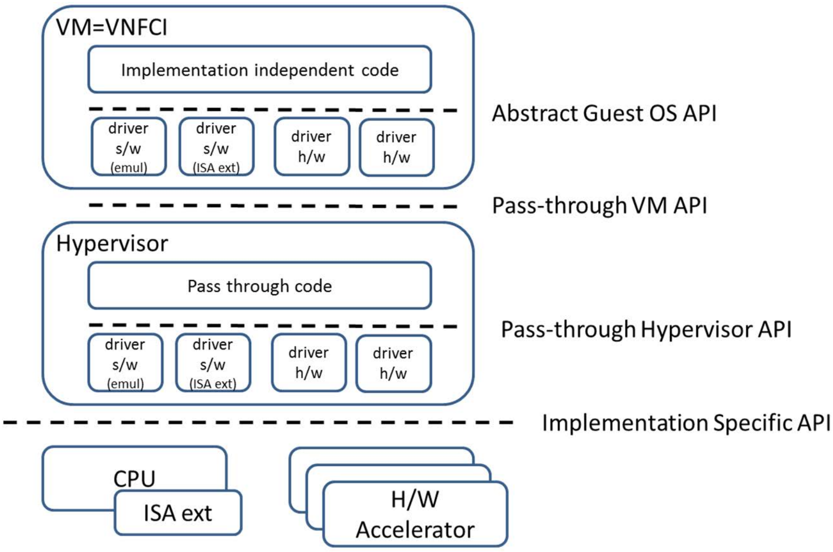

Figure 4.0-2: NFV acceleration abstraction layer implemented within VNF as an integrated model

4.1 Problem statement

As shown in the figures 4.0-1 and 4.0-2, the type of NFV acceleration deployment varies from an integrated deployment to a disaggregated deployment), the management aspect of NFV Acceleration focuses on the latter one in the present document.

In order to accommodate the possibility of a disaggregated deployment, the management of NFV accelerators is needed in order to ensure higher-level orchestration capabilities, utilization monitoring, and customer SLAs are supported.

This management support should take into consideration that NFV accelerators may be deployed as a pool in which resources are dynamically allocated to other specific purposes within the NFVI or to specific VNFs based on needs and policies, and that such assignments will need to be recorded, modified, suspended, or possibly even support live migration.

4.2 Management of acceleration resources

4.2.1 Overview

Figure 4.2.1-1 provides an illustration of the architecture for the management of acceleration resources.

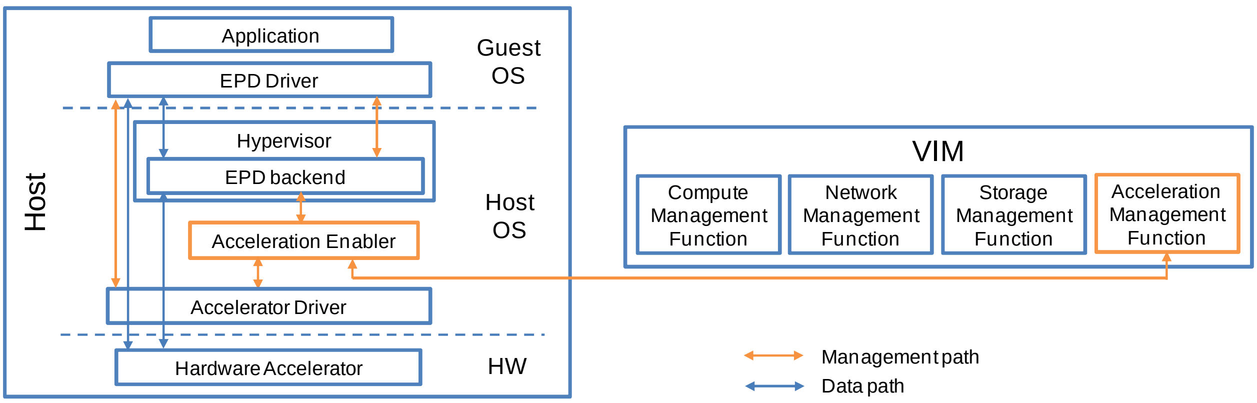

Figure 4.2.1-1: Acceleration resource management architecture

As shown in figure 4.2.1-1 illustrates the acceleration resource management architecture, which comprises several components: acceleration management function, acceleration enabler, extensible para-virtualised device (EPD) Driver, EPD backend and accelerator driver. The acceleration management function is located in the VIM and is in charge of the global management of the acceleration resources. The acceleration enabler is located in the NFVI, typically as software in a host server, or API library provided by the accelerator driver, etc. The acceleration enabler provides the local management of acceleration resources inside the NFVI- Node. The EPD driver and EPD backend are as defined in ETSI GS NFV-IFA 002 [5].

As illustrated above, the acceleration enabler exposes functions provided by the accelerator driver and can be utilized by the EPD backend and/or the acceleration management function. The enabler's main role is to help the acceleration management function to discover, collect information and manage the local accelerators residing in different NFVI-Nodes.

The acceleration management function can utilize the acceleration enablers within the VIM's administration domain to form a global view of the whole acceleration resources. Cooperating with compute, network, and storage management functions inside the VIM, the acceleration management function can help the VIM to find and maintain the most appropriate resources to host a VNF.

4.2.2 Feature discovery

4.2.2.0 Introduction

Features of NFV acceleration resources need to be discovered, understood and managed such as it is possible to efficiently determine what acceleration resources are suitable for what requests and for how long while taking into accounts SLA and performance metrics without compromising flexibility and portability.

4.2.2.1 Acceleration resources discovery

4.2.2.1.1 Discovery mechanism

Structural metadata with fine granularity to allow for deeper structured information and enable greater levels of technical manipulation is required for resources discovery to be effective and scalable.

These metadata can be introduced to the resource management system when acceleration resources are added to the NFVI or could be requested by the VIM. Upon a request from the VIM the NFVI is required to respond with the appropriate metadata information.

4.2.2.1.2 Acceleration resources features

For acceleration resources to be used efficiently within the NFV architectural framework, their features need to be understood such as when an acceleration resource is requested, the most appropriate resource is given. Table 4.2.2.1.2-1 provides the functional requirements for acceleration resources features.

Table 4.2.2.1.2-1: Functional requirements for acceleration resources features

Numbering	Feature	Functional requirements description
ACC_RES_FEATURE.001	UID	Each acceleration resource shall have a unique identifier.
ACC_RES_FEATURE.002	version	Each acceleration resource shall specify the version of its accelerator.
ACC_RES_FEATURE.003	Type	Each acceleration resource shall have a clear type (e.g. Crypto, FFT, IPSec, etc.).
ACC_RES_FEATURE.004	Capabilities	Each acceleration resource shall indicate its acceleration specific capabilities.
ACC_RES_FEATURE.005	Number of Channels	Each acceleration resource shall indicate how many channels it supports.
ACC_RES_FEATURE.006	Number of Contexts	Each acceleration resource shall indicate how many contexts it supports.
ACC_RES_FEATURE.007	Allows Migration	Each acceleration resource shall indicate if it supports live migration capabilities.
ACC_RES_FEATURE.008	QoS	Each acceleration resource shall indicate the quality of service level it supports.
ACC_RES_FEATURE.009	Data Format	Each acceleration resource shall indicate the data format they operate on.
ACC_RES_FEATURE.010	Re-Programmability	Each acceleration resource shall indicate whether it requires a hardware image to be programmed with before it can operate.
ACC_RES_FEATURE.011	Resource Availability	Each acceleration resource shall indicate the level or amount of availability that are currently unused and can be allocated.

4.2.2.2 Feature discovery in NFV acceleration management

Table 4.2.2.2-1 provides the functional requirements for acceleration feature discovery.

Table 4.2.2.2-1: Acceleration feature discovery

Numbering	Functional requirements description
NFV_ACC_MGMT_FD.1	The VIM may support the capability of feature discovery for NFV acceleration resources, which may involve requesting NFVI to report necessary features.

4.2.3 Lifecycle management of acceleration resources

4.2.3.1 Introduction

Lifecycle management of acceleration resources refers to a class of management functions that take care of the accelerators after they are discovered or allocated till power down or release. Typical functions are acceleration resource allocation, release, configuration, information management, and performance management.

4.2.3.2 Functionalities

Lifecycle management of acceleration resources consists of two parts: local management by the NFVI and global management of VIM. NFVI is responsible for management of acceleration resources residing in a compute node, network nodes, storage nodes or shared resource pools, etc. The VIM is responsible for the management and control of the whole acceleration resources within its management domain.

NOTE 1: The local management functions are usually implemented by accelerator providers and may vary among different accelerators. The functions may be accessible to the VNF utilizing the accelerator and/or the VIM, or solely designed for manual/static control. It is recommended that these functions are accessible to the VIM through standard interfaces to reduce management cost.

More specifically, the management functionality provided by the NFVI can include:

- Local accelerator management, which is responsible for virtualisation/slicing of accelerators, attachment/detachment of accelerators to virtualisation containers (e.g. VMs), configuration, power control, etc.

NOTE 2: Virtualisation or slicing of accelerators refers to the act of allocating portions of the accelerator resources to create a virtual accelerator. For example, some FPGAs provide partial reconfiguration feature which allows the user to modify the logic of a portion of the FPGA blocks. This feature can be utilized to dynamically create an accelerator.

- Local resource information management, which is responsible for accelerator discovery, registration, status/performance collection, capacity management, etc.
- Fault management, which handles fault detection and diagnosis, alarm/event notification, fault recovery (e.g. hot-swap), etc.
- Driver management, which deals with firmware/driver upgrade, accelerator re-programming, etc.

Functionality provided by the VIM includes:

- Accelerator selection: Using this functionality, the VIM selects an appropriate accelerator based on the acceleration capability requirement indicated in the request and acceleration resource information to meet both the acceleration capability requirement (e.g. choosing an accelerator which is of the required type, has enough capacity to perform the acceleration request and meets other constraints like NUMA support) and acceleration resource management policies (e.g. choosing an accelerator which has lower workload to achieve load balancing).
- Accelerator management: The VIM interacts with the NFVI local accelerator management functionality to manage or control certain accelerator, which can include allocation, modification, release, configuration, etc.
- Global resource information management: The VIM aggregates the information (e.g. accelerator type, capability, capacity, status, performance, etc.) collected by the local resource information management functions. The VIM makes use of this information to make management decisions and forward the information to the NFVO/VNFM, NFVO and/or VNFM may forward this information to the OSS.
- Fault management: the VIM can receive alarm/event notifications sent by the local fault management functions. The VIM conducts fault recovery operations (e.g. re-allocate accelerator) and/or transmits the notifications to NFVO/VNFM, the NFVO and/or the VNFM may forward this information to the OSS.

4.2.3.3 Functional requirements

Table 4.2.3.3-1 provides the functional requirements for the lifecycle management of acceleration resources.

Table 4.2.3.3-1: Lifecycle management functional requirements

Numbering	Functional requirement description
NfvAccMgmt.LCM.001	The NFVI shall support the capability of attaching/detaching an accelerator to a virtualisation container (e.g. VM) (see note 1).
NfvAccMgmt.LCM.002	The NFVI may support the capability of virtualising/slicing hardware accelerator.
NfvAccMgmt.LCM.003	The NFVI shall support the capability to collect accelerator performance metrics.
NfvAccMgmt.LCM.004	The NFVI shall support the capability to report acceleration resource's information (see note 2).
NfvAccMgmt.LCM.005	The NFVI may support the capability to configure an accelerator.
NfvAccMgmt.LCM.006	The NFVI may support the capability of accelerator driver management.
NfvAccMgmt.LCM.007	The NFVI should support the capability to reserve acceleration resources.
NfvAccMgmt.LCM.008	The NFVI should support the capability of accelerator fault management.
NfvAccMgmt.LCM.009	The VIM shall support the capability to choose an appropriate accelerator based on the acceleration capability requirement in the virtualisation resource allocation request and acceleration resource information. Stored in the VIM (see notes 2 and 3).
NfvAccMgmt.LCM.010	The VIM shall support the capability to interact with the management functions provided by the NFVI to conduct lifecycle management operations on selected accelerators (see note 4).
NfvAccMgmt.LCM.011	The VIM shall support the capability to aggregate accelerators' information collected by the NFVI (see note 2).
NfvAccMgmt.LCM.012	The VIM should support the capability to collect fault information related to accelerators from the NFVI.
NOTE 1: An accelerator can be attached to multiple virtualisation containers. NOTE 2: Acceleration resource information include type, capability, capacity, status, performance, etc. NOTE 3: The acceleration capability requirement include type, capacity, NUMA support, KPI, etc. NOTE 4: Lifecycle management operations include allocation, release, update, etc.	

5 NFV acceleration management impact on VIM

5.1 General

5.1.1 Impact documented in ETSI GS NFV-IFA 010

The general functional requirements for NFV acceleration management on VIM are documented in clause 8.6 "Function Requirements For NFV Acceleration Management" in ETSI GS NFV-IFA 010 [4].

As defined in these requirements, when NFV acceleration is applied, VIM shall support the management of the NFV acceleration resources. VIM shall also support the capability to provide acceleration capability information to NFVO, retrieve feature related information provided by the NFV acceleration resources and translate the acceleration capability requirement (e.g. bandwidth value) into acceleration resource context (e.g. number of FPGA blocks).

5.2 Role of SDN controllers

Software-defined networking (SDN) is "a set of techniques that enables to directly program, orchestrate, control and manage network resources, which facilitates the design, delivery and operation of network services in a dynamic and scalable manner" Recommendation ITU-T Y.3300 [i.1]. SDN can be used to provide connectivity services for VNFs and acceleration resources in scenarios such as presented later in this clause:

- VNF and accelerator interconnection;
- Accelerator chaining;
- Traffic offload.

In the description of these scenarios, the term Network Controller ETSI GS NFV 003 [1] is used as per ETSI GS NFV-INF 005 [i.2] and encompasses SDN controllers.

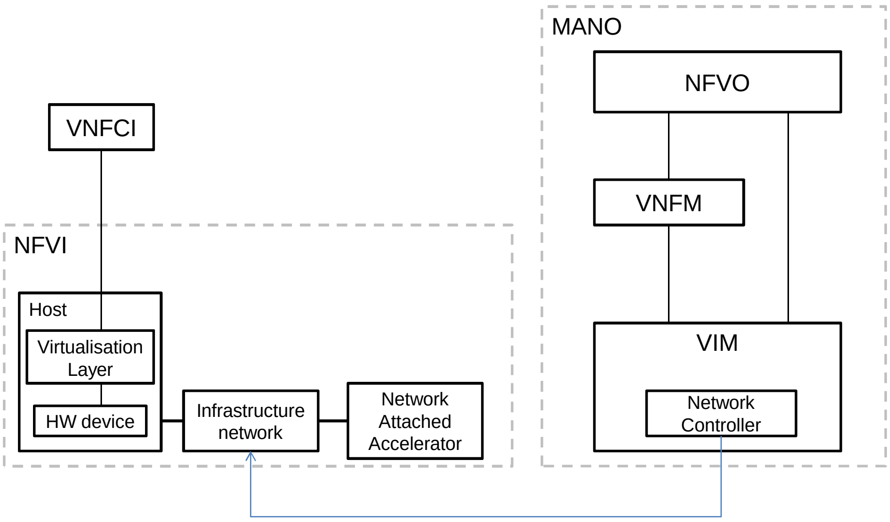

Figure 5.2-1: VNF and accelerator interconnection

VNF and accelerator interconnection scenario is illustrated in figure 5.2-1. For accelerators that are accessible through the network, (i.e. network-attached accelerator), SDN is used to provide connectivity between VNF and accelerators. Upon receiving a request for accelerator resource allocation, the VIM shall be able to establish connectivity in the NFVI infrastructure network through a network controller to enable the communication between the VNFCI and the network attached accelerator. Note that the SDN controller being located in the VIM is one of the options described in ETSI GS NFV-EVE 005 [i.5].

NOTE 1: Generally, the virtual network ETSI GS NFV 003 [1] for communication between the VNFCI and the accelerator is not different from the virtual network for communication between VNFCIs. The network controller simply configures the forwarding rules to attach the accelerator to the internal network of the VNF. If certain constraints are needed (e.g. separate data and control channel) and provided (e.g. via request parameters, VNFD, VIM settings, etc.), the network controller should apply those constraints and configure the VNF internal network's forwarding rules accordingly.

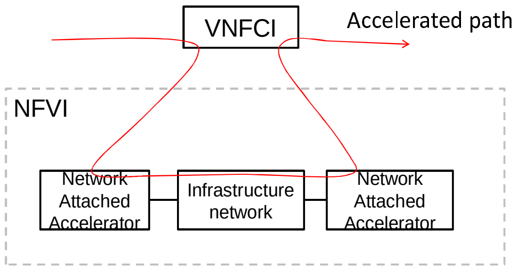

Figure 5.2-2: Accelerator chaining

Figure 5.2-2 illustrates the accelerator chaining scenario. The network controller can chain different accelerators to provide an acceleration service for a certain VNF. The function provided by the network controller is similar to the one provided in the VNF and accelerator interconnection scenario.

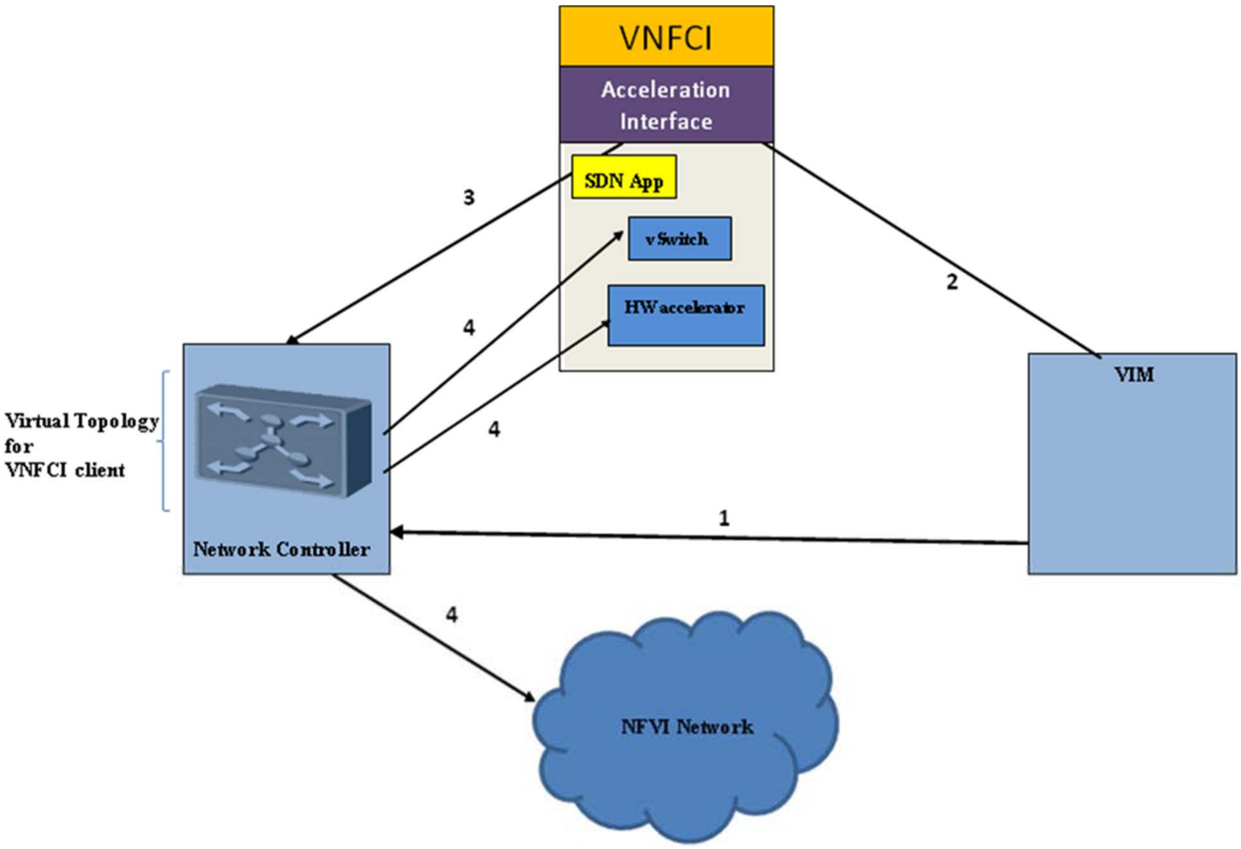

Figure 5.2-3: Traffic offload

Figure 5.2-3 is an illustration of the traffic offload scenario. For fast path accelerator ETSI GS NFV-IFA 001 [i.6], SDN is used to offload the traffic to the NFVI Network.

If network acceleration needs to be performed, the VIM (see steps 1 and 2 in figure 5.2-3) instantiates an SDN application and configures the SDN application and Network Controller with the necessary attributes to establish an adjacency, linking it to a specific VNFCI and [Vn-Nf]/N/Lx service. In step 3, the VNFCI requests changes through the API to the SDN application. In turn, the SDN application communicates with the Network Controller to offload the traffic to the Infrastructure Network. The request should be validated by the Network Controller, only endpoints included in the same [Vn-Nf]/N/Lx service are accepted in the request. Requests to reroute to/from non-adjacent entities should be rejected. In step 4, the Network Controller converts the requests into forwarding rules on the underlying network elements providing the [Vn-Nf]/N/Lx service. Step 4 also consists in programming the Infrastructure Network.

NOTE 2: [Vn-Nf]/N/Lx stands for [Vn-Nf]/N/L2 and [Vn-Nf]/N/L3 as used per ETSI GS NFV-INF 005 [i.2].

6 NFV acceleration management impact on reference points

6.0 Introduction

The impact of NFV acceleration management on VIM reference points are defined in ETSI GS NFV-IFA 005 [2], ETSI GS NFV-IFA 006 [3] and ETSI GS NFV-IFA 010 [4]. In this clause all requirements are therefore referenced from these documents to maintain consistency and clarity.

6.1 General Impact on Or-Vi and Vi-Vnfm

The general functional requirements for NFV acceleration management on Or-Vi and Vi-Vnfm are documented in clause 6.13 "Functional requirements for NFV acceleration management" and clause 7.9 "Functional requirements for NFV acceleration management" in ETSI GS NFV-IFA 010 [4].

As defined in these requirements, when NFV acceleration is applied, the NFVO shall support the capability to receive and query acceleration capability information from the VIM, as well as select a VIM that has enough available acceleration capabilities to support acceleration capability requirement(s) of the VNF. The NFVO and the VNFM shall support the capability to request to the VIM the allocation and release of necessary acceleration resources to meet acceleration capability requirement(s) of the VNFs when VNF-related Resource Management in indirect mode is applicable, as well as retrieve acceleration capabilities requirement(s) of the VNF from the VNF deployment template.

6.2 Impact on Or-Vi

The specific information element requirements on Or-Vi for NFV acceleration management are documented in clauses 8.3.3.2, 8.4.3.2.2, 8.4.3.7.2, 8.4.4.2.2, 8.4.4.6.2, 8.15.3.4.2, 8.15.3.5.2 and 8.15.5.3.2 of ETSI GS NFV-IFA 005 [2] with regard to the attribute "accelerationCapability".

The requirements mainly specify that when NFV acceleration applies for Or-Vi, if the virtualised compute resource requires resources with certain acceleration capabilities (e.g. packet acceleration library). It should be noted that the cardinality can be 0, if no particular accelerationCapability is requested/provided.

6.3 Impact on Vi-Vnfm

The specific information element requirements on Vi-Vnfm for NFV acceleration management are documented in clauses 8.3.2.2, 8.4.2.2, 8.4.2.7, 8.4.3.2, 8.4.3.6, 8.4.13.3 and 8.4.15.2 of ETSI GS NFV-IFA 006 [3] with regard to the attribute "accelerationCapability".

The requirements mainly specify that when NFV acceleration applies for Vi-Vnfm, if the virtualised compute resource requires resources with certain acceleration capabilities (e.g. packet acceleration library). It should be noted that the cardinality can be 0, if no particular accelerationCapability is requested/provided.

6.4 Impact on Nf-Vi

The impact of acceleration on the Nf-Vi reference point is not addressed in the present document.

Annex A (informative): Acceleration management information flows

A.1 Introduction

This annex represents a non-exhaustive collection of information flows related to the management of acceleration resources. All information flows in this annex are informative and are created for the purpose of describing acceleration management procedures.

Some information flows are based on VNF lifecycle management flows in ETSI GS NFV-MAN 001 [i.3]. For that reason, the VNF lifecycle management flows may need to be updated in the future.

For the sake of simplicity failure cases are not considered in these information flows. Similarly, some optional operations in VNF lifecycle management flows in ETSI GS NFV-MAN 001 [i.3] are omitted.

A.2 Acceleration resource allocation

The acceleration resource allocation procedure is embedded in the VNF instantiation procedure. Depending on the entity on which the VNFM invokes the resource allocation operation (i.e. direct/indirect mode of VNF-related resource management), the information flows are slightly different as illustrated in figures A.2-1 and A.2-2.

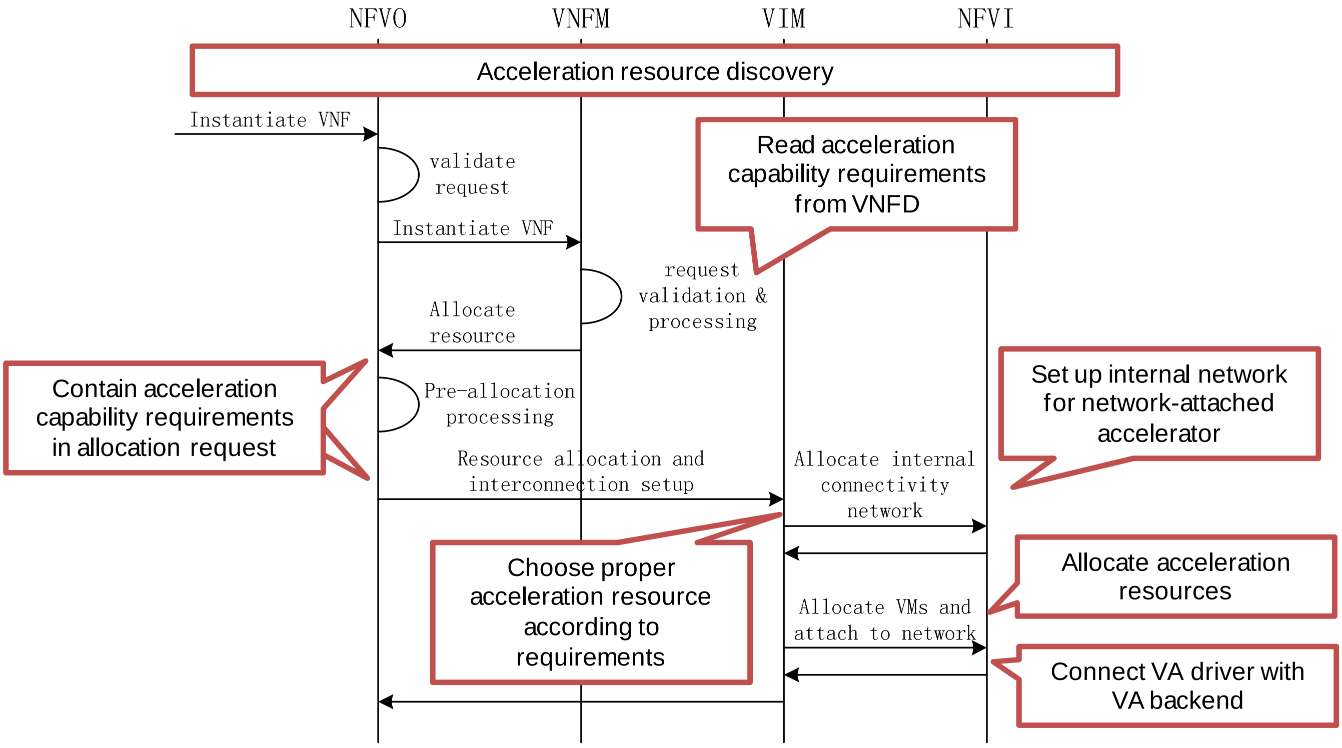

Figure A.2-1: Acceleration resource allocation in indirect mode

In the case where the VNFM invokes the resource allocation operation to the NFVO (i.e. indirect mode), the main steps for acceleration resource allocation are:

0) (pre-condition) Acceleration resources are discovered by VIM; related information can be shared with NFVO/VNFM.

1) The NFVO receives the VNF instantiation request, validates it, and sends it to the VNFM.

2) When processing the VNF instantiation request, the VNFM reads acceleration capability requirements of the VNF from VNFD and can then modify/implement the input instantiation data with that information.

3) The VNFM sends resource allocation request to the NFVO. The NFVO can select a VIM based on the request parameters (including acceleration capability requirements) and send the request to the selected VIM.

4) After receiving the request, the VIM selects proper resources (e.g. host) to according to the request parameters (including acceleration capability requirements).

5) The VIM allocates the internal connectivity network. It can also set up internal network (e.g. through SDN controller) for network-attached accelerator(s) if needed.

6) The VIM allocates the needed compute resources (e.g. VMs), storage resources and acceleration resources. During the process, the VIM can translate the acceleration capability requirements into acceleration resource context. It can also configure the VA backend to connect with VA driver during booting process of compute resources (e.g. VMs) if needed.

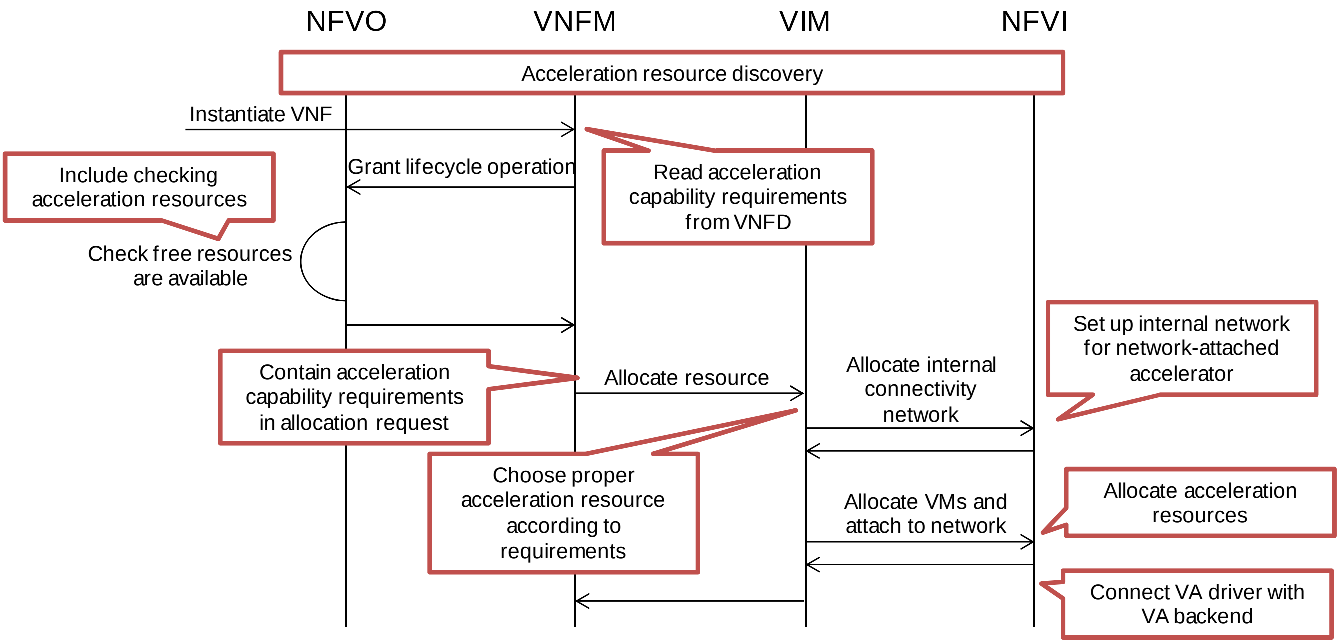

Figure A.2-2: Acceleration resource allocation in direct mode

In the case where the VNFM invokes the resource allocation operation to the VIM (i.e. direct mode), the main steps for acceleration resource allocation are:

0) (pre-condition) Acceleration resources are discovered by VIM; related information can be shared with NFVO/VNFM.

1) The VNFM receives the VNF instantiation request. When processing it, the VNFM reads acceleration capability requirements of the VNF from VNFD and can modify/implement the input instantiation data with that information.

2) The VNFM asks the NFVO for granting of the instantiation operation. The NFVO checks the feasibility of the resources (including acceleration resources), selects an appropriate VIM and sends a response (containing the VIM identifier) to the VNFM.

3) The VNFM sends a resource allocation request (including acceleration resources) to the selected VIM.

4) After receiving the request, the VIM selects proper resources (e.g. host) according to the request's parameters (including acceleration capability requirements).

5) The VIM allocates the internal connectivity network. It can also set up internal network (e.g. through a SDN controller) for network-attached accelerator(s) if needed.

6) The VIM allocates the needed compute resources (e.g. VMs), storage resources and acceleration resources. During the process, the VIM can translate the acceleration capability requirements into acceleration resource context. It can also configure the VA backend to connect with VA driver during the booting process of compute resources (e.g. VM) if needed.

A.3 Acceleration resource release

As for the acceleration resource allocation information flows, the acceleration resource release procedure is embedded in the VNF termination procedure, as illustrated in figure A.3-1.

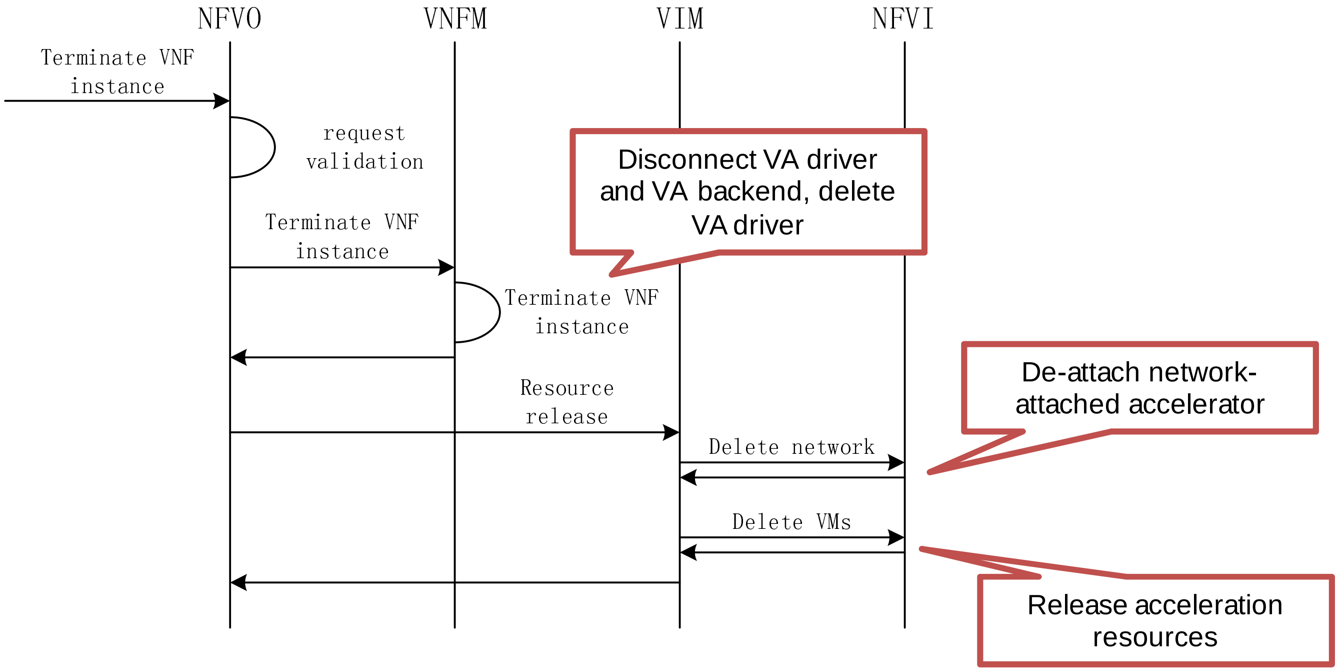

Figure A.3-1: Acceleration resource release in indirect mode

The main steps for acceleration resource release are:

1) The NFVO receives VNF instance termination request, validates it and sends the request to the VNFM.

2) The VNFM terminates the VNF instance. This step can include a (graceful) shutdown of the VNF, during which, the VA driver can be disconnected to the VA backend and be deleted.

3) The VNFM acknowledges the completion of VNF termination to the NFVO.

4) The NFVO sends resource release request to the VIM.

5) The VIM deletes the internal network. This step can include de-attaching network-attached accelerators from the network.

6) The VIM deletes the allocated compute resources (VM), storage resources and releasable acceleration resources (e.g. virtualised accelerator).

Annex B (informative): An implementation example for the acceleration resource management architecture

B.1 Introduction

The main body of the present document describes the acceleration management architecture from functional perspective, i.e. explaining which functions each function block can support for the management of acceleration resources. This annex describes an implementation example of the acceleration resource management architecture and depicts how these functions can be combined and utilized to form a complete management system.

NOTE: This annex is only an example of the implementation of the acceleration resource management architecture, not a recommendation nor a reference design.

B.2 Architecture Description

B.2.1 Overview

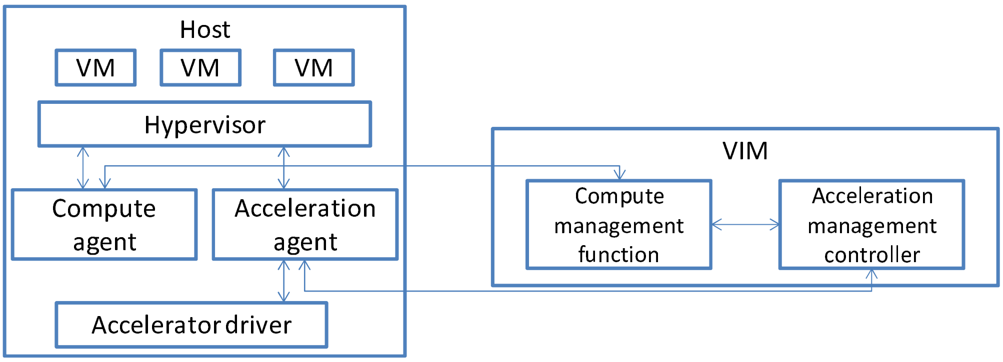

Figure B.2.1-1: Acceleration management architecture example

OpenStack [i.4] is a widely known cloud management system and a potential candidate for VIM. The OpenStack Compute (i.e. project Nova) is a component of OpenStack which manages the compute resources in the infrastructure. It adopts a horizontally scalable architecture which is composed of a centralized compute management function and multiple compute agents. The compute agents are resided in different hosts and communicate with the hypervisors to manage the VMs. This implementation example adopts a similar way to implement the acceleration management architecture, as shown in figure B.2.1-1. The acceleration management architecture is composed of an acceleration management controller (which corresponds to the acceleration management function in clause 4.2.1) and an acceleration agent (which corresponds to the acceleration enabler in clause 4.2.1). The acceleration management controller is the centralized orchestrator of acceleration resources. It receives requests from the compute/network/storage management functions, arranges those requests and communicates with the acceleration agents to fulfil the requests. The acceleration agent manages communications with accelerators. It collaborates with the accelerators' drivers and/or the hypervisor to collect information of and/or conduct operations on the accelerators.

B.2.2 Information flows

B.2.2.0 Introduction

This clause illustrates how the example architecture described in clause B.2.1 works by describing some acceleration resource management information flows.

B.2.2.1 Acceleration resource allocation example

Figure B.2.2.1-1 shows the acceleration resource allocation procedure.

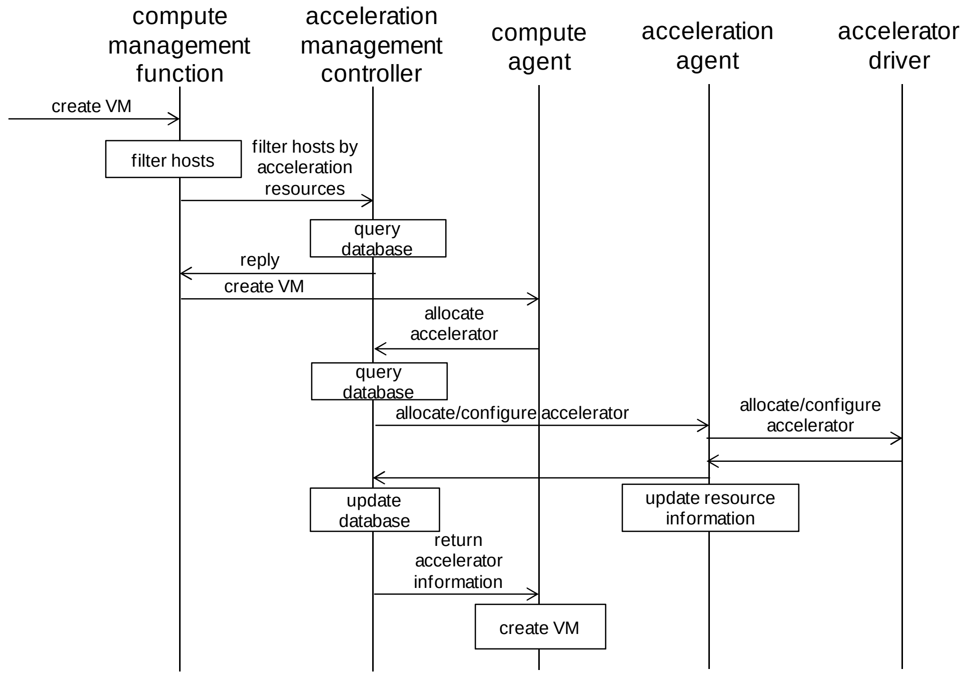

Figure B.2.2.1-1: Acceleration resource allocation

The main steps for the acceleration resource allocation procedure are:

1) The compute management function receives create VM request.

2) The compute management function filters appropriate hosts by factors like available resources, affinity/anti-affinity rules, etc.

3) The compute management function calls the acceleration management controller to further filter the hosts by acceleration resource constraints.

4) Acceleration management controller queries its database, selects a host and returns it to the compute management function.

5) The compute management function sends create VM request to the compute agent located in the selected host.

6) The compute agent sends an allocate accelerator request to the acceleration management controller. The request contains acceleration capability requirements like acceleration type and algorithm type, and may contain other constraints like bandwidth, NUMA, etc.

7) After receiving the request, the acceleration management controller queries the database, and select appropriate acceleration resource(s) based on the acceleration capability requirements.

8) The acceleration management controller sends allocation and/or configuration instructions to the acceleration agent. The instructions contain accelerator type, algorithm type and may contain other information like bandwidth, NUMA, etc.

9) The acceleration agent conducts the instructions in cooperation with the accelerator driver to allocate and/or configure accelerator(s) that fulfil the acceleration capability requirements (i.e. acceleration type and algorithm type, may include bandwidth, NUMA, etc.). If supporting programming (e.g. FPGA), the accelerator(s) could be instructed to load image or binary file that corresponds to the acceleration capability requirements during the allocation/configuration process.

10) After that, the acceleration agent sends a reply to the controller and updates its local resource information.

11) On receiving the reply, the acceleration management controller updates its database and returns the allocated accelerator's information to the compute agent.

12) The compute agent boots up the VM and attaches the accelerator(s) to it after receiving the reply.

B.2.2.2 Acceleration resource discovery

Figure B.2.2.2-1 illustrates the acceleration resource discovery procedure.

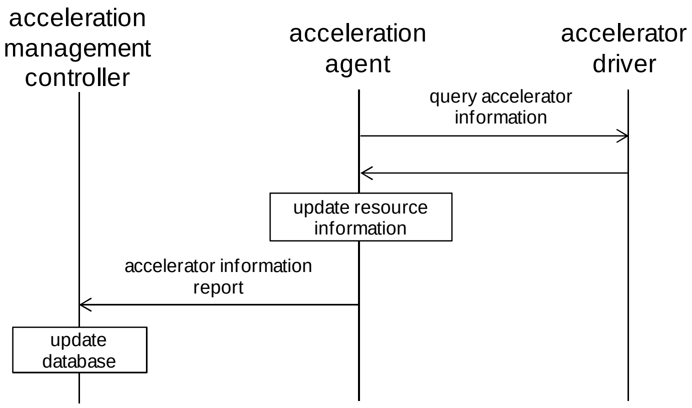

Figure B.2.2.2-1: Acceleration resource discovery

The main steps for the acceleration discovery procedure are:

1) The acceleration agent periodically queries information of the local accelerators through APIs provided by the accelerator drivers. The response basically contains acceleration type and algorithm type, and may provide other information like bandwidth (including total bandwidth and consumed bandwidth), NUMA, etc.

2) After receiving the reply, the acceleration agent updates its local resource information.

3) The acceleration agent reports the information of the local accelerators and of the host (e.g. name) to the acceleration management controller.

4) The acceleration management controller updates its database after receiving the reported information.

Annex C (informative): Authors & contributors

The following people have contributed to the present document:

Rapporteur:

- Zhipeng, Huang, Huawei

Other contributors:

- Francois-Fredric Ozog, 6wind
- Xu Yang, Huawei
- Jiangbang Wu, Huawei
- Peng Yuan, Huawei
- Abdel Rabi, Altera
- LE GRAND Olivier, Orange

Annex D (informative): Bibliography

- ETSI GS NFV-INF 003: "Network Functions Virtualisation (NFV); Infrastructure; Compute Domain".
- ETSI GS NFV-INF 004: "Network Functions Virtualisation (NFV); Infrastructure; Hypervisor Domain".
- ETSI GS NFV-SWA 001: "Network Functions Virtualisation (NFV); Virtual Network Functions Architecture".

History

Document history		
V2.1.1	April 2016	Publication
V2.3.1	August 2017	Publication