

NMOP
Internet-Draft
Intended status: Standards Track
Expires: 21 September 2025

Y. Yue, Ed.
W. Zhu, Ed.
T. Sui, Ed.
China Unicom
20 March 2025

DPU-Based Bare Metal Management and Control Solution
draft-yue-nmop-dpu-management-control-00

Abstract

This document proposes a DPU-based bare metal management solution to address inefficiencies in management and resource utilization associated with traditional bare metal deployments. The core idea of this solution is to leverage the DPU's high-performance processing and network acceleration capabilities, transforming traditional network/resource management into a DPU-centric control model. This not only simplifies bare-metal operations but achieves unified management across virtualized and physical environments through a consolidated framework.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 21 September 2025.

Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights

and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
2. Conventions and Definitions	2
3. Use cases	3
4. Requirements	3
4.1. Component Offloading Strategy	4
4.2. Management Integration Plan	5
4.2.1. Compute Mounting	5
4.2.2. Network Mounting	5
4.2.3. Storage Mounting	6
4.3. Components and Relationships	6
4.3.1. Management Platform	6
4.3.2. Cloud Resource Scheduling and Management Components	6
5. Additional Considerations	7
6. Security Considerations	7
7. IANA Considerations	7
8. Normative References	7
Authors' Addresses	7

1. Introduction

DPU (Data Processing Unit) is an emerging hardware acceleration technology designed to offload data-plane tasks (e.g., high-performance networking and packet processing) while integrating CPU clusters for control-plane offloading. Widely adopted in data centers, DPUs enable unified resource management, network virtualization, AI acceleration, and security hardening through their high-throughput processing, ultra-low latency, and purpose-built acceleration engines.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 RFC2119 [RFC8174] when, and only when, they appear in all capitals, as shown here. Abbreviations and definitions used in this document: *DPU: Data Processing Unit. *HPC: High-Performance Computing. *VM: Virtual Machine. *NICs: Network Interface Cards. *SPDK: Storage Performance Development Kit.

3. Use cases

Due to exclusive access to bare metal resources with no performance loss, they are ideal for high-IO applications and high-performance computing (HPC) such as massive data collection/mining, high-performance databases, and large-scale online games. Compared to virtual machines, DPU-based bare metal resources provide physical server characteristics, while compared to physical servers, they offer the elasticity of virtual machines.

1. In large-scale data center telecom cloud scenarios, 5G new calling services (including media applications) require physical servers with GPUs and DPU-based bare metal shared pool management technology to enable rapid resource provisioning.

2. In edge computing scenarios, the acceleration capabilities of DPUs are leveraged to achieve fast bare metal startup, addressing high-concurrency CDN demands.

4. Requirements

Bare metal deployment is a method of running operating systems and applications directly on physical hardware, which provides an effective solution for high-performance computing (HPC) and large-scale data processing tasks. However, traditional bare metal deployment faces challenges such as inflexible resource management and poor resource utilization, where each bare metal server typically runs only a single application.

OpenStack can effectively manage and allocate computing, storage, and network resources in data centers. Through its Ironic subproject, OpenStack manages bare metal servers. Ironic offers an API framework and task orchestration services for operations like bare metal server provisioning, power management, and rebooting. It can be deployed independently or integrated with other OpenStack services (e.g., Nova, Neutron, Glance), enabling seamless integration of bare metal servers into existing cloud platforms while delivering a unified user experience equivalent to virtual machine (VM) services.

Leveraging its data processing capabilities, the DPU provides optimized support for OpenStack bare metal services and network management. The DPU enables diskless boot, hot-swappable cloud storage, and storage management for bare metal servers. Simultaneously, it offloads and integrates network configuration and protocol processing tasks within OpenStack networks. This simplifies network topology, enhances flexibility, reduces costs, and strengthens security and isolation, thereby delivering robust guarantees for OpenStack's efficient operation and resource management.

The DPU addresses the following critical requirements for efficient resource management:

1. In bare metal scenarios, traditional NICs cannot support remote cloud disk mounting before the host operating system boots, thus failing to enable diskless boot and requiring each bare metal server to have a local system disk; when deploying bare metal servers via PXE installation, critical information such as BMC IP addresses, usernames, and passwords must be pre-registered; furthermore, the ports and network configurations connecting bare metal servers to external switches demand prior planning, resulting in inflexibility where bare metal servers cannot be interchanged with virtual machine (VM) servers during elastic deployments, preventing unified pool management of bare metal, VM, and container resources while significantly prolonging deployment cycles—all of which can be resolved by a DPU-based solution that leverages external storage as boot disks with preconfigured image parameters to enable on-demand provisioning.
2. Essentially, the DPU is a dedicated processor designed to provide data-centric infrastructure virtualization services, including network, storage, security, and management functions for data centers. The DPU's specialized processing units resolve performance bottlenecks caused by generalized infrastructure virtualization. # Architecture Design

4.1. Component Offloading Strategy

The Nova-compute, Cinder-volume, Neutron, and other components of OpenStack were originally deployed on the host machines of compute nodes. With the introduction of DPUs, OpenStack now requires additional operations to manage DPUs, leading to multiple system management strategies. All components installed on compute nodes, such as Nova, Cinder, and Neutron, need to be offloaded and managed directly on the DPUs.

The overall architecture involves inserting DPUs into bare-metal physical servers of cloud-based products via PCIe. In OpenStack, the management components for bare-metal servers are directly offloaded and deployed from the servers to the DPUs, forming the cloud resource scheduling and management components (as illustrated in the diagram). Leveraging the data plane acceleration capabilities of FPGA or ASIC-based hardware on the DPUs, data forwarding is accelerated, and these components are unified under DPU management.

Under this architecture, all computing resources and network devices are managed directly by the DPUs. OpenStack perceives the scheduled bare-metal devices as native DPU-managed resources. Control operations, such as creating or deleting bare-metal instances, are issued by OpenStack to the DPUs. The DPUs then directly manage these operations on the compute node's host machines and external storage system images for device creation or deletion.

4.2. Management Integration Plan

Bare-Metal Adaptation in OpenStack: To enable bare-metal registration in OpenStack and provide an upper-layer management system with a resource view of bare-metal types, the mounting of compute, network, and storage resources must be implemented for bare-metal scenarios.

4.2.1. Compute Mounting

Mount CPU resources from bare-metal servers to OpenStack bare-metal instances. OpenStack requests CPU resources from bare-metal nodes. Based on the request, the bare-metal nodes select an appropriate node. OpenStack then loads the selected bare-metal node's information into the instance, completing CPU allocation to the instance.

4.2.2. Network Mounting

Create virtual NICs (Network Interface Cards) on servers and mount them to OpenStack bare-metal instances. Step 1: Create virtual NICs and Open vSwitch ports. Servers create virtual NICs based on physical or virtual network devices. Open vSwitch ports are then configured on the DPUs to enable communication with the external LAN. The virtual NICs are connected to these ports to ensure network connectivity for the bare-metal instance. Step 2: Mount virtual NICs to bare-metal instances. OpenStack retrieves virtual NIC information via the Neutron component, injects this information into the bare-metal instance, and notifies the DPU. The instance can then detect and utilize the virtual NIC, completing the mounting process.

4.2.3. Storage Mounting

DPUs create virtual disks and mount them to OpenStack bare-metal instances. Step 1: Create virtual disks. The DPU sends a request via the storage client to the storage server to create a physical disk, retrieves basic hardware configuration details, and establishes a virtual disk using SPDK (Storage Performance Development Kit) technology. By maintaining metadata linking the virtual disk to the physical disk, the DPU ensures seamless integration. Step 2: Mount virtual disks to bare-metal instances. Substep 1: Use the Cinder component to create virtual disk metadata in OpenStack. Substep 2: OpenStack retrieves virtual disk information from the DPU via the Nova component, binding the metadata to the DPU's virtual disk. Substep 3: Inject the virtual disk information into the bare-metal instance, finalizing the mounting process

4.3. Components and Relationships

The management platform includes nova-api, cinder-api, and neutron-api, while the cloud resource scheduling and management components include Nova-compute, Cinder, Neutron agent, and BM Agent.

4.3.1. Management Platform

nova-api: Receives requests to create bare-metal instances and invokes nova-scheduler to allocate an ironic node. cinder-api: Receives requests to create/delete storage devices for bare-metal instances and schedules the creation/deletion of virtio-blk devices on the ironic node. neutron-api: Receives requests to create/delete network devices for bare-metal instances and schedules the creation/deletion of virtio-net devices on the ironic node.

4.3.2. Cloud Resource Scheduling and Management Components

nova-compute: Uses a specified disk image to instruct Cinder to create a system disk volume. Cinder clones the image volume via snapshots (not full copies) on the backend storage, which is completed rapidly, typically within seconds. Cinder: Manages storage devices based on commands from the control node. Neutron agent: Manages network devices and handles flow table distribution for virtual switches according to control node instructions. BM Agent: Runs on the DPU SoC and communicates with ironic-conductor via RPC. It manages storage and network resources for bare-metal instances on the local node. During deployment, it receives deployment commands from ironic-conductor, connects cloud disks within the DPU, and mounts them to the host. During runtime, it dynamically adds or removes disks/NICs for the host based on commands from ironic-conductor.

5. Additional Considerations

For implementation and evaluation: Deploying a DPU-based unified bare-metal management architecture requires balanced hardware and software configurations. Proper hardware (e.g., DPUs) and driver installations are critical for deployment success. Effectiveness is evaluated through: Performance testing: Measuring network acceleration improvements (e.g., throughput, latency). Resource utilization analysis: Quantifying efficiency gains in CPU, storage, and network usage. Security validation: Ensuring secure resource isolation and compliance with policies.

6. Security Considerations

TBD

7. IANA Considerations

TBD

8. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Yi Yue (editor)
China Unicom
Beijing
China
Email: yuey80@chinaunicom.cn

Wei Zhu (editor)
China Unicom
Beijing
China
Email: zhuw93@chinaunicom.cn

Tengfei Sui (editor)
China Unicom
Beijing
China
Email: suitf@chinaunicom.cn