Developer’s Interface Guide for Green Datacenter Architecture-based Computing Environment

Release 2.0.2

System Architecture Study Committee

Oct 11, 2024

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Current Issues in Data Centers

The amount of data worldwide is increasing at a rapid pace of about 30% annually, and the market size of data center servers continues to expand.

So far, the replacement of traditional data centers with more efficient large-scale cloud and hyperscale data centers has relatively restrained the increase in global data center power consumption. However, it is expected that the rapid increase in large-scale data centers will lead to a significant rise in the overall power consumption of data centers, and the current pace of technological advancement will not keep up with the increasing power demand.

In this context, there is a need to achieve further power savings not only by optimizing the power of individual hardware components but also by advanced management that includes power control of each individual hardware component.

1.2 Purpose of This Guide

Traditional systems have followed a form where various hardware components are aggregated into a server chassis, requiring the determination of the hardware configuration based on pre-planned system designs. Such an approach struggles to cope with resource usage situations or load fluctuations that deviate from pre-planned scenarios, leading to decreased power efficiency.

Recently, technologies represented by PCIe/CXL, among others, have advanced disaggregation technologies that allow some or most hardware components to be separated from the server chassis and be flexibly reassembled as needed. This technology makes it possible to allocate resources flexibly at the required scale and timing, allowing unnecessary components to be detached and powered down, thus further reducing power consumption.

However, compared to traditional machine-level control, this method increases complexity and necessitates the automation of operations to enhance power efficiency. Therefore, the purpose of this guide is to present the necessary requirements and specifications for autonomous system operation, aiming for standardization in the operation of systems based on disaggregation technologies.

This guide describes the elements to be considered in hardware components, firmware, fabric, and the operating system to achieve control based on disaggregation technologies.

1.3 Scope of This Guide

This guide defines the requirements and interfaces for various hardware components based on disaggregation technologies.

The aim is to share design standards necessary for interoperability between platforms that realize disaggregation system, and to enable power-saving operations through disaggregation control across various platforms.

This guide describes common design guidelines necessary for disaggregation control and does not cover proprietary designs.

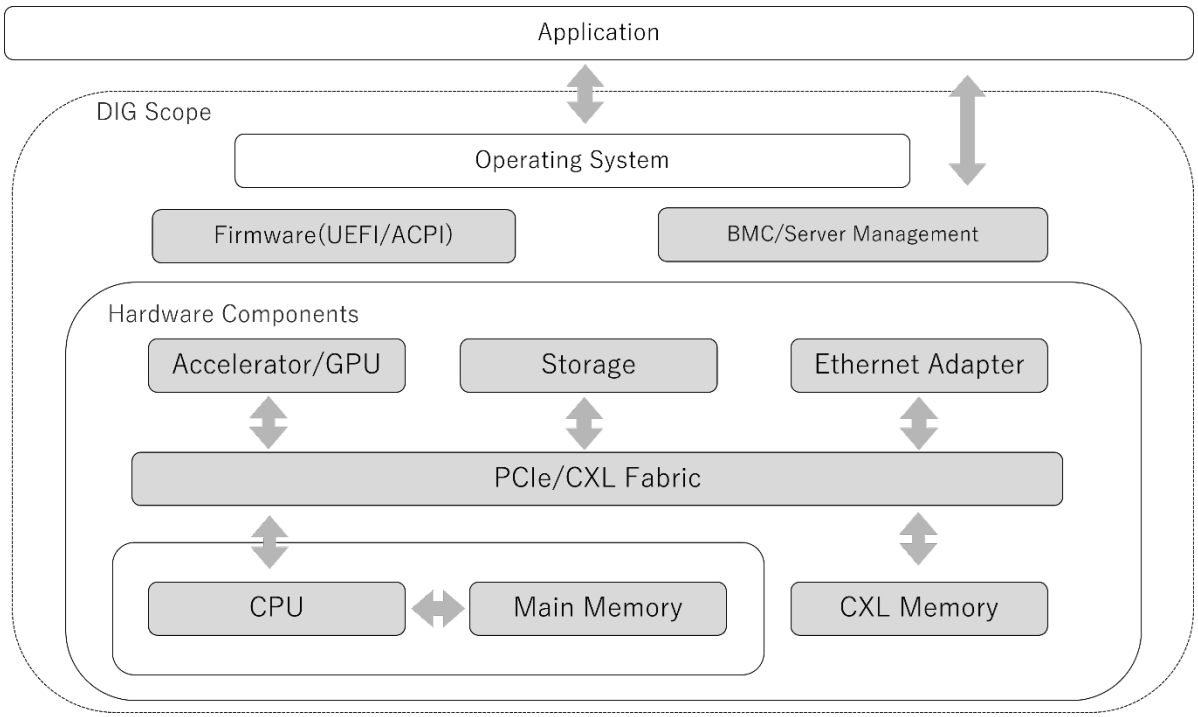

Fig. 1: Overall Configuration

1.4 Terminology

Advanced Configuration and Power Interface (ACPI)
A standard for managing the power and configuration of machines.

Baseboard Management Controller (BMC)
A dedicated component for managing, monitoring, and remotely controlling hardware.

Compute Express Link (CXL)
An interconnect standard for high-speed connections between CPUs and hardware components.

Digital Signal Processor (DSP)
A computation component specialized in digital processing.

Fabric Manager (FM)
A component that controls PCIe/CXL switches.

Fabric Manager API
The interface standard used by the FM to control PCIe/CXL switches.

Field Programmable Gate Array (FPGA)
A computation component realized with reprogrammable hardware.

General-Purpose computing on GPU (GPGPU)
A function that performs general-purpose computations using a GPU.

Global Fabric Attached Memory (GFAM)
A CXL feature that shares memory regions between hosts.

Graphics Processing Unit (GPU)
In this guide, mainly referring to components intended for GPGPU use.

Multi Logical Device (MLD)
A CXL feature that allows partitioning a single CXL memory into dedicated allocations for multiple hosts.

Non Uniform Memory Access (NUMA)
A configuration where the access cost to shared memory varies between processors.

NVM Express (NVMe)
An interface standard for non-volatile storage.

Out Of Band (OOB)
A dedicated communication means used for device management.

Remote Keyboard Video Mouse (KVM)
A function that allows remote use of keyboard, video, and mouse via BMC.

Secure Erase
A function that erases data on disks compliant with ATA or NVMe standards.

Serial over LAN (SoL)
A function that connects to the serial port of a device using LAN.

Unified Extensible Firmware Interface (UEFI)
A standard interface between the operating system and firmware.

Accelerator
A general term for computation components other than the CPU, such as GPU, DSP, FPGA.

Node
Computational resources that can run an operating system, configured using PCIe/CXL switches.

CHAPTER 2

System Architecture

2.1 Overview

Disaggregation technology connects each hardware component configured based on PCIe/CXL technology to a PCIe/CXL fabric configuration. This allows for the re-evaluation of device configurations at the hardware component level, enabling the realization of device configurations optimal for the tasks being executed.

This disaggregation technology autonomously performs these device configuration changes to achieve configurations that match the execution processes and load conditions. By doing so, it ensures that the necessary resources are secured for processing, while also enabling the shutdown of unnecessary devices, thereby improving power efficiency.

2.2 System Overview

2.2.1 Target System for Disaggregation Technology

The target system for disaggregation technology is configured as follows:

- Individual components are interconnected via fabric (such as switches).
  - It is assumed that each component exists within the area connected to the fabric network, typically within the rack or nearby.
  - The fabric network configuration needs to be pre-determined, although expansion and modification after the initial configuration are anticipated.
- It must be possible to obtain equipment information and performance information for each component.
- Connection management of each device can be performed by control software.
  - The fabric must have an external interface, and it must be possible to control and change the connections between components via external software.
  - The entire system, including the operating system, must be able to recognize and operate under the new configuration.
- It must be possible to control the power of unused components.

For such a system, control software performs as follows:

- Analyzes the characteristics of the executing workload to derive the necessary amount of resources for operation. The estimation of resource amounts involves analyzing the equipment and performance information of each component.

- Analyzes the device characteristics of each component to derive the characteristics required for executing the workload. The derivation of characteristics involves analyzing the equipment and performance information of each component.
- Based on the estimated workload and device characteristics, determines the optimal combination of components using machine learning or other methods. This calculates the most power-efficient hardware configuration corresponding to the execution workload characteristics.
- The control software implements configuration changes to the fabric via the external interface, altering the connection state of each component between online and offline. This rebuilds the node configuration to suit the execution workload in stages. Furthermore, the control software also deploys and configures the operating system and software in accordance with each node configuration change, controlling workload execution and deployment.
- During instances of resource shortage or surplus, reanalyzes and includes the migration cost in decision-making. This continuously optimizes the overall system for power efficiency.

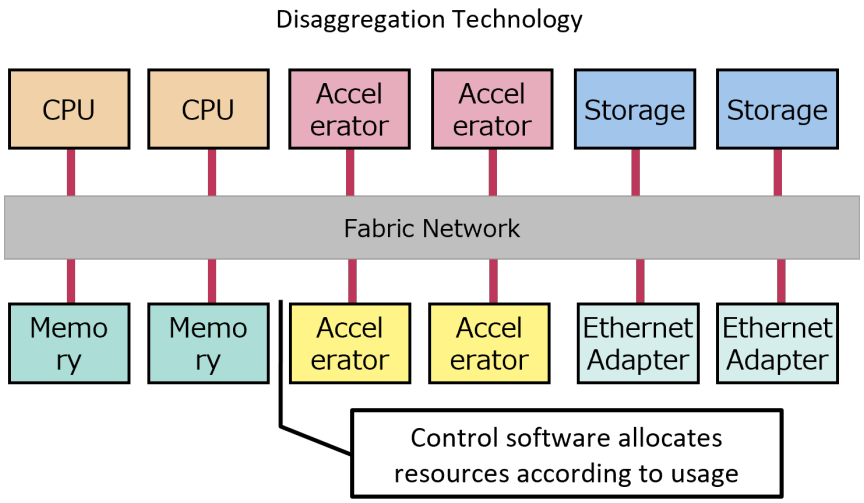

Fig. 2: Overview of Disaggregation Technology

2.2.2 Control Paths of Control Software

Control software obtains necessary information and performs control through three main paths.

1. Information acquisition and control through the external interface via OOB (Out-Of-Band) operations on hardware components connected to the fabric (PCIe/CXL switches).
    - Via the OOB of each hardware component
    - Via the OOB of nodes configured using fabric functions
2. Information acquisition and control via FM API for the fabric (PCIe/CXL switches).
    - Recognizing the configurations identified by PCIe/CXL switches
    - Controlling the connection and disconnection of each hardware component
3. Information acquisition and control using standard protocols via Ethernet adapters recognized by the operating system on the node or via BMC.

The HW Components in Fig. 3 represent the hardware components (excluding CPU and main memory) described in *Requirements for Hardware Components and Fabric*.

For the information obtained through each path, these are combined to prioritize using information that can accurately represent the actual state for analysis.

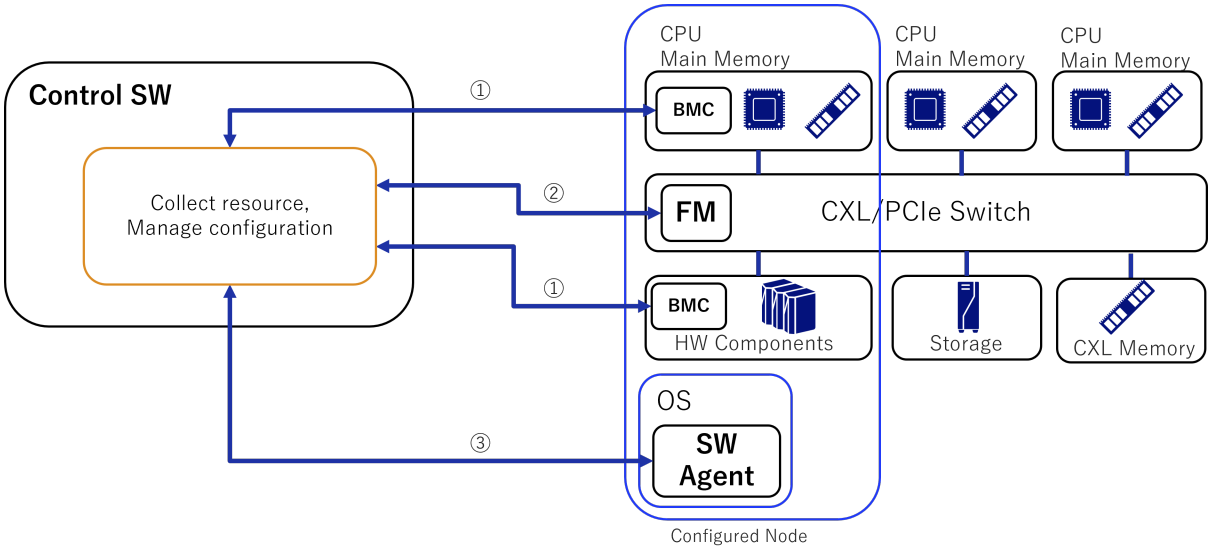

Fig. 3: Control Paths

Configuration changes are implemented via FM. While PCIe/CXL switches are assumed, any configuration with fabric control functionality is acceptable.

2.3 Control Software Stack

Based on the overall overview of a disaggregation data center, which includes multiple fabric networks and inter-node and inter-rack networks, this section discusses how system subject to the disaggregation technology is determined, i.e., how the control software is organized in a hierarchical manner to control the configuration of the entire system dynamically. This section describes how the control of the configuration of the fabric network is organized hierarchically before the control software of the system performs dynamic configuration changes. Although this section includes content that is outside the scope of this document, as described in *Scope of This Guide*, it is necessary to define the overall picture and the concept of the control software stack when considering use cases that also work with wide area optical path network.

Fig. 4 shows an image of the control hierarchy. At the bottom are examples of hardware components: memory (Mem), accelerators (ACC), CPUs, SSDs, and PCIe/CXL switch devices that make up the fabric network. In addition, as a future network interface technology, we describe Optical Smart NIC that can seamlessly support wide area networks, intra-datacenter networks(inter-rack networks), and fabric networks. Optical path networks are also described as wide area networks and intra-data center networks (inter-rack networks). The details of Optical Smart NICs are described in *Optical Smart NIC*, The Optical Smart NICs can be used as PCIe extenders or Ethernet adapters, and their functions can be switched via software using OOB control. Determining the functions of the Optical Smart NICs is performed by the Computing resource manager via intermediate management control before the Disaggregation OS is started. Therefore, the Disaggregation OS, which is the control software for the disaggregation system, recognizes and can use the Optical Smart NIC as a conventional device such as an Ethernet adapter or a PCIe Extender.

The top half of Fig. 4 illustrates the software modules responsible for control role.

- Datacenter orchestrator/Datacenter operating system: Receives management control and requests from operators and users for management control of the entire data center.
- Unified resource manager: Integrated management of compute and network resources, in particular, determining the fabric network configuration and network/compute resource assignments.
- Computing resource manager: Manages administrative control rights for computing resources.
- Network resource manager: Manages network resource slicing and administrative control rights.

- Disaggregation OS: Performs management and control of computing resources with in a fabric networks units. This corresponds to the control software described in the *System Overview*.
- Network OS: Provides management control of network resources.
- Server OS: An operating system Power oned on a compute node built by Disaggregation OS, assuming an operating system running on a common node such as Red Hat or Ubuntu.
- Network Element Controller: Control software installed in network devices such as optical switches.
- Storage SW: Software that runs on storage nodes and provides storage services.
- FM Controller: Software that controls the PCIe/CXL switch fabric.
- Intermediate Management Control: Software that provides an interface for OOB control of Optical Smart NICs, calculates optimal settings for Optical Smart NICs based on high-level requirements, and performs command conversion, etc.

These classifications are intended to organize and facilitate understanding of a series of control flows, but are not mandatory for software implementation.

The following is an example of a sequence of control flow in the configuration and modification of a fabric network before the control software performs dynamic configuration changes, using the above classification. First, the Datacenter orchestrator/Datacenter operating system receives requests from operators, users, etc. To configure or change the fabric network and queries the Unified resource manager. The Unified resource manager identifies the compute and network resources required to configure the fabric network and requests their respective settings from the Computing resource manager and Network resource manager. The Computing Resource Manager determines the role of the Optical Smart NICs as either Ethernet adapters or PCIe extenders and then sends a setting request to the intermediate management control to set the function of the Optical Smart NICs. The network resource manager configures the network routes via the Network OS and Network element controller. Then, the computing resource manager delegates the management rights of the configured Optical Smart NICs and the computing resources in the configured fabric network to the Disaggregation OS. With these preliminaries, the Disaggregation OS can arbitrarily perform dynamic configuration changes within the pre-determined fabric networks and compute resources, if the Disaggregation OS determines that additional network and compute resources are needed during operation, it can make a request to a higher level Manager. Conversely, it can also return the management rights of resources it deems unnecessary to the upper-level manager and release them.

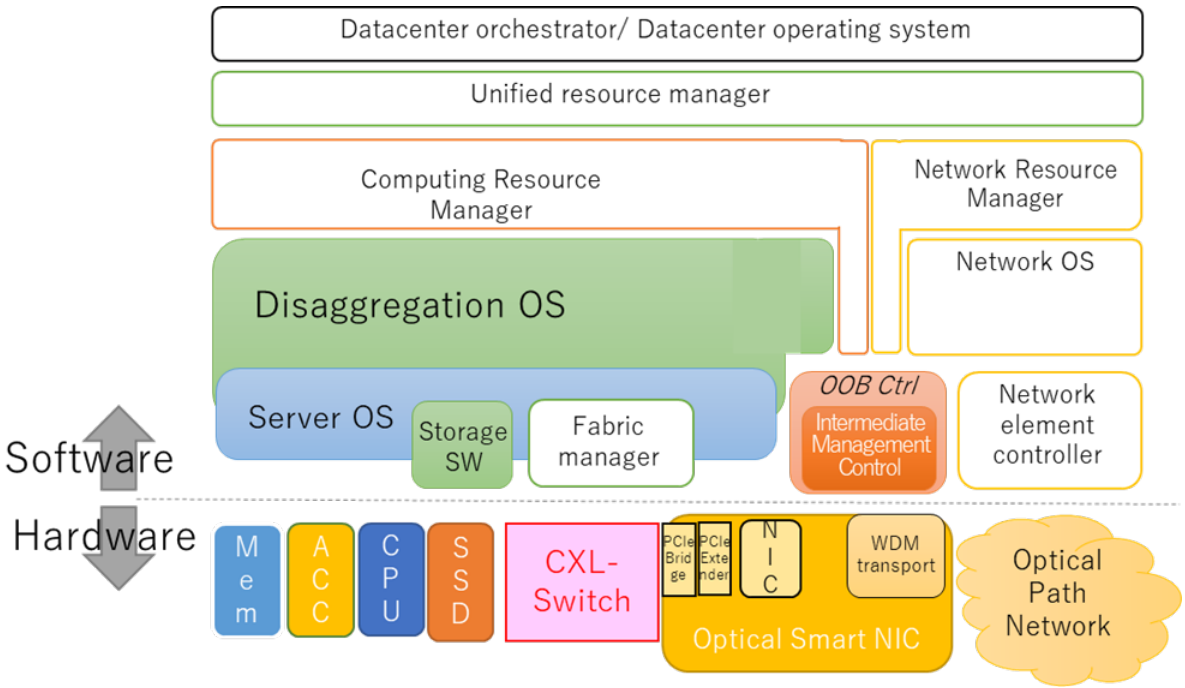

Fig. 4: Control Software Stack Image

CHAPTER 3

Requirements for Hardware Components and Fabric

Describes the requirements necessary for managing and controlling each hardware component and fabric from control software.

3.1 CPU

Control software achieves dynamic configuration by managing and controlling CPU devices. Therefore, CPU devices need to have an external interface for control software.

In dynamic configuration control, it is necessary to build a configuration that optimizes power based on the characteristics of the workload while meeting the hardware requirements required by the application. To do this, it is necessary to obtain hardware specs and metrics for the CPU device.

The requirements for CPU devices are described below.

3.1.1 External Interface

Describes the requirements needed for the external interface of the CPU device to be managed and controlled by the control software.

Required Items

- An IP-based connection to the external interface must be possible.
- Must be able to connect to PCIe/CXL fabric.
- Hardware components physically connected to the CPU device without passing through PCIe/CXL fabric must be manageable and controllable using the same external interface as the CPU device.

Optional Items

- Should be able to authenticate the source of the connection to the external interface.

3.1.2 Management

Describes the requirements necessary for managing the CPU device to achieve dynamic configuration control.

Required Items

- The ability to obtain information that uniquely identifies hardware components or relatively unique information for uniquely identifiable hardware components.
- The ability to obtain spec information to determine whether the hardware requirements needed by the application can be met.
- The ability to obtain metrics information necessary to measure the characteristics of the workload.
- The ability to obtain the power state.
- The ability to obtain the presence of power control functionality.
- The ability to obtain power consumption (quantity).
- If the CPU device is physically connected to other hardware components without passing through PCIe/CXL fabric, the ability to obtain information about those hardware components. Also, the ability to obtain information that determines the physical connection between the CPU device and each hardware component.

Optional Items

- Spec information should be obtainable regardless of whether the CPU device is in use or not.
- Metrics information should be obtainable regardless of whether the operating system is running or not.
- The ability to estimate the power consumption (quantity) at maximum performance.
- The ability to obtain information for each hardware component physically connected to the CPU device.
- The ability to obtain information about hardware components logically connected via PCIe/CXL fabric. Also, the ability to distinguish other hardware components physically connected without passing through PCIe/CXL fabric.

3.1.3 Control

Describes the requirements necessary for controlling the CPU device to achieve dynamic configuration control.

Required Items

- Power control must be possible.

Optional Items

- If metric values are cumulative, it should be possible to reset the cumulative values.
- The ability to control each hardware component physically connected to the CPU device.
- Must have power-saving functions according to the load.
- If the CPU device is configured as a node, power control as a node must be possible.

3.2 Main Memory/Dedicated Memory

Control software achieves dynamic configuration by managing and controlling main memory/dedicated memory devices. Main memory/dedicated memory devices are physically connected to CPU devices or accelerator devices and are managed and controlled using the external interface of those devices.

In dynamic configuration control, it is necessary to build a configuration that optimizes power based on the characteristics of the workload while meeting the hardware requirements required by the application. To do this, it is necessary to obtain hardware specs and metrics for main memory/dedicated memory devices.

The requirements for main memory/dedicated memory devices are described below.

3.2.1 Management

Describes the requirements necessary for managing main memory/dedicated memory devices to achieve dynamic configuration control.

Required Items

- The ability to obtain information that uniquely identifies hardware components or relatively unique information for uniquely identifiable hardware components.
- The ability to obtain spec information to determine whether the hardware requirements needed by the application can be met.
- The ability to obtain metrics information necessary to measure the characteristics of the workload.
- The ability to obtain information that identifies devices physically connected.
- The ability to obtain the power state. If it cannot be obtained as a main memory/dedicated memory device, it should match the power state of the connected device.
- The ability to obtain the presence of power control functions when power control is possible.
- The ability to obtain power consumption (quantity). If it cannot be obtained as a main memory/dedicated memory device, it should be included in the power consumption (quantity) of the connected device.

Optional Items

- Spec information should be obtainable regardless of whether the main memory/dedicated memory device is in use or not.
- Metrics information should be obtainable regardless of whether the operating system is running or not.
- The ability to obtain the absence of power control functionality when power control is not possible.
- The ability to estimate the power consumption (quantity) at maximum performance. If it cannot be obtained as a main memory/dedicated memory device, it should be included in the power consumption (quantity) of the connected device at maximum performance.

3.2.2 Control

Describes the requirements necessary for controlling main memory/dedicated memory devices to achieve dynamic configuration control.

Required Items

None

Optional Items

- Power control should be possible.
- If metric values are cumulative, it should be possible to reset the cumulative values.
- Memory content should be securely erasable.

3.3 CXL Memory

Control software achieves dynamic configuration by managing and controlling CXL memory devices. Therefore, CXL memory devices need to have an external interface for control software.

In dynamic configuration control, it is necessary to build a configuration that optimizes power based on the characteristics of the workload while meeting the hardware requirements required by the application. To do this, it is necessary to obtain hardware specs and metrics for CXL memory devices.

The requirements for CXL memory devices are described below.

3.3.1 External Interface

Describes the requirements needed for the external interface of the CXL memory device to be managed and controlled by the control software.

Required Items

- An IP-based connection to the external interface must be possible.
- Must be able to connect to PCIe/CXL fabric.

Optional Items

- Should be able to authenticate the source of the connection to the external interface.

3.3.2 Management

Describes the requirements necessary for managing the CXL memory device to achieve dynamic configuration control.

Required Items

- The ability to obtain information that uniquely identifies hardware components or relatively unique information for uniquely identifiable hardware components.
- The ability to obtain spec information to determine whether the hardware requirements needed by the application can be met.
- The ability to obtain metrics information necessary to measure the characteristics of the workload.
- The ability to obtain the power state.
- The ability to obtain the presence of power control functionality.
- The ability to obtain power consumption (quantity).

Optional Items

- Spec information should be obtainable regardless of whether the CXL memory device is in use or not.
- Metrics information should be obtainable regardless of whether the operating system is running or not.
- The ability to estimate the power consumption (quantity) at maximum performance.

3.3.3 Control

Describes the requirements necessary for controlling the CXL memory device to achieve dynamic configuration control.

Required Items

- Power control must be possible.

Optional Items

- If metric values are cumulative, it should be possible to reset the cumulative values.
- Memory content should be securely erasable.

3.4 Accelerators

Control software achieves dynamic configuration by managing and controlling accelerator devices. Therefore, accelerator devices need to have an external interface for control software.

In dynamic configuration control, it is necessary to build a configuration that optimizes power based on the characteristics of the workload while meeting the hardware requirements required by the application. To do this, it is necessary to obtain hardware specs and metrics for accelerator devices.

The requirements for accelerator devices are described below.

3.4.1 External Interface

Describes the requirements needed for the external interface of the accelerator device to be managed and controlled by the control software.

Required Items

- An IP-based connection to the external interface must be possible.
- Must be able to connect to PCIe/CXL fabric.

Optional Items

- Should be able to authenticate the source of the connection to the external interface.

3.4.2 Management

Describes the requirements necessary for managing the accelerator device to achieve dynamic configuration control.

Required Items

- The ability to obtain information that uniquely identifies hardware components or relatively unique information for uniquely identifiable hardware components.
- The ability to obtain spec information to determine whether the hardware requirements needed by the application can be met.
- The ability to obtain metrics information necessary to measure the characteristics of the workload.
- The ability to obtain the power state.
- The ability to obtain the presence of power control functionality.
- The ability to obtain power consumption (quantity).

Optional Items

- Spec information should be obtainable regardless of whether the accelerator device is in use or not.
- Metrics information should be obtainable regardless of whether the operating system is running or not.
- The ability to estimate the power consumption (quantity) at maximum performance.

3.4.3 Control

Describes the requirements necessary for controlling the accelerator device to achieve dynamic configuration control.

Required Items

- Power control must be possible.

Optional Items

- If metric values are cumulative, it should be possible to reset the cumulative values.
- Must have power-saving functions according to the load.

3.5 Storage

Control software achieves dynamic configuration by managing and controlling storage devices. Therefore, storage devices need to have an external interface for control software.

In dynamic configuration control, it is necessary to build a configuration that optimizes power based on the characteristics of the workload while meeting the hardware requirements required by the application. To do this, it is necessary to obtain hardware specs and metrics for storage devices.

The target storage devices are those physically connected to the CPU device or connected via PCIe/CXL fabric. Storage devices managed by a separate storage management system and providing storage functionality to nodes are excluded from the scope.

The requirements for storage devices are described below.

3.5.1 External Interface

Describes the requirements needed for the external interface of the storage device to be managed and controlled by the control software.

Required Items

- An IP-based connection to the external interface must be possible.
- Must be able to connect to PCIe/CXL fabric.

Optional Items

- Should be able to authenticate the source of the connection to the external interface.

3.5.2 Management

Describes the requirements necessary for managing the storage device to achieve dynamic configuration control.

Required Items

- The ability to obtain information that uniquely identifies hardware components or relatively unique information for uniquely identifiable hardware components.
- The ability to obtain spec information to determine whether the hardware requirements needed by the application can be met.
- The ability to obtain metrics information necessary to measure the characteristics of the workload.
- The ability to obtain the power state.
- The ability to obtain the presence of power control functions when power control is possible.
- The ability to obtain power consumption (quantity).

Optional Items

- Spec information should be obtainable regardless of whether the storage device is in use or not.
- Metrics information should be obtainable regardless of whether the operating system is running or not.
- The ability to obtain the absence of power control functionality when power control is not possible.
- The ability to estimate the power consumption (quantity) at maximum performance.

3.5.3 Control

Describes the requirements necessary for controlling the storage device to achieve dynamic configuration control.

Required Items

None

Optional Items

- If metric values are cumulative, it should be possible to reset the cumulative values.
- Power control should be possible.
- Storage content should be securely erasable.

3.6 Ethernet Adapter

Control software achieves dynamic configuration by managing and controlling Ethernet adapter devices. Therefore, Ethernet adapter devices need to have an external interface for control software.

In dynamic configuration control, it is necessary to build a configuration that optimizes power based on the characteristics of the workload while meeting the hardware requirements required by the application. To do this, it is necessary to obtain hardware specs and metrics for Ethernet adapter devices.

Note that the target is the hardware components recognized by the operating system as Ethernet adapters. In cases where the operating system recognizes the hardware as a different type of hardware due to functionalities such as those provided by Smart NICs, or in cases where the hardware is not recognized by the operating system at all, those components are considered out of scope.

The requirements for Ethernet adapter devices are described below.

3.6.1 External Interface

Describes the requirements needed for the external interface of the Ethernet adapter device to be managed and controlled by the control software.

Required Items

- An IP-based connection to the external interface must be possible.
- Must be able to connect to PCIe/CXL fabric.

Optional Items

- Should be able to authenticate the source of the connection to the external interface.

3.6.2 Management

Describes the requirements necessary for managing the Ethernet adapter device to achieve dynamic configuration control.

Required Items

- The ability to obtain information that uniquely identifies hardware components or relatively unique information for uniquely identifiable hardware components.
- The ability to obtain spec information to determine whether the hardware requirements needed by the application can be met.
- The ability to obtain metrics information necessary to measure the characteristics of the workload.
- The ability to obtain the power state.
- The ability to obtain the presence of power control functions when power control is possible.
- The ability to obtain power consumption (quantity).

Optional Items

- Spec information should be obtainable regardless of whether the Ethernet adapter device is in use or not.
- Metrics information should be obtainable regardless of whether the operating system is running or not.
- The ability to obtain the absence of power control functionality when power control is not possible.
- The ability to estimate the power consumption (quantity) at maximum performance.

3.6.3 Control

Describes the requirements necessary for controlling the Ethernet adapter device to achieve dynamic configuration control.

Required Items

None

Optional Items

- Power control should be possible.
- If metric values are cumulative, it should be possible to reset the cumulative values.

3.7 Optical Smart NIC

An Optical Smart NIC is treated as a device that constitutes a fabric, either as an Ethernet adapter or as a PCIe/CXL extender.

The control software does not manage or control the Optical Smart NIC device.

If the operating system recognizes the Optical Smart NIC as a different type of hardware component or is unable to recognize it at all, it must meet the requirements for that specific type of hardware component.

3.8 Chassis/Rack

Control software achieves dynamic configuration by managing and controlling chassis/racks. Therefore, chassis/racks need to have an external interface for control software.

Dynamic configuration control requires obtaining information on the hardware components contained within the chassis/rack as well as the hardware specs and metrics of the chassis/rack.

The requirements for chassis/rack are described below.

3.8.1 External Interface

Describes the requirements needed for the external interface of the chassis/rack to be managed and controlled by the control software.

Required Items

- An IP-based connection to the external interface must be possible.

Optional Items

- Should be able to authenticate the source of the connection to the external interface.
- Should have an external interface for each contained hardware component.

3.8.2 Management

Describes the requirements necessary for managing the chassis/rack to achieve dynamic configuration control.

Required Items

- The ability to obtain information that uniquely identifies hardware components or relatively unique information for uniquely identifiable hardware components.
- The ability to obtain spec information and metrics information for each contained hardware component.
- The ability to obtain metrics information for resources consumed by the chassis/rack, excluding the contained hardware components.
- The ability to obtain the power state.
- The ability to obtain the presence of power control functionality.

Optional Items

- If metric values are cumulative, it should be possible to reset the cumulative values.
- The ability to estimate the power consumption (quantity) at maximum performance.

3.8.3 Control

Describes the requirements necessary for controlling the chassis/rack to achieve dynamic configuration control.

Required Items

- Must be able to control each contained hardware component.

Optional Items

- Should be able to control the power of the chassis/rack.

3.9 Node

Control software achieves dynamic configuration by managing and controlling nodes. Therefore, nodes need to have an external interface for control software.

The external interface is assumed to be either exclusively occupied by the node or shared with other hardware components.

Additionally, it is necessary to deploy the operating system to the node using existing methods. Existing methods might include:

- Network installation via PXE boot
- Disk cloning
- Installation from BMC virtual media (ISO mount/USB redirect)

The requirements for nodes are described below.

3.9.1 External Interface

Describes the requirements needed for the external interface of the node to be managed and controlled by the control software.

Required Items

- An IP-based connection to the external interface must be possible.

Optional Items

- Should be able to authenticate the source of the connection to the external interface.

3.9.2 Management

Describes the requirements necessary for managing the node to achieve dynamic configuration control.

Required Items

- The ability to obtain information that uniquely identifies the node.

Optional Items

- The ability to obtain spec and metrics information for the hardware components that make up the node.
- The ability to obtain the power state.
- The ability to obtain the boot status and operating state of the operating system.

3.9.3 Control

Describes the requirements necessary for controlling the node to achieve dynamic configuration control.

Required Items

- Power control must be possible.

Optional Items

- The ability to shut down the operating system.
- The ability to specify or control the order of boot devices.
- The ability to control virtual media (ISO mount/USB redirect) for the node.
- Must have functionalities to remotely control the operating system, such as SoL or KVM.

3.10 Operating System

Control software achieves dynamic configuration through the management and control of the operating system. The management and control of the operating system utilize both standard interfaces provided by the operating system and device-specific interfaces.

The assumed operating system is Linux, and the Linux kernel is not modified to implement the system described in this guide.

Control software agents perform the management and control of the operating system, achieving dynamic configuration control by communicating with control software over an IP-based connection.

Dynamic configuration control dynamically changes the configuration based on the resources required by the workloads running on the node. Therefore, the operating system needs to support hot-plugging and unplugging.

Additionally, to measure workload load, it is necessary to obtain metrics on a per-workload basis.

The requirements for the operating system are described below.

3.10.1 External Interface

Describes the requirements needed for the external interface of the operating system to be managed and controlled by the control software agent.

Required Items

- Management and control must be possible using standard interfaces provided by the operating system or device-specific interfaces.

Optional Items

None

3.10.2 Management

Describes the requirements necessary for managing the operating system to achieve dynamic configuration control.

Required Items

- The ability to recognize hardware components logically connected via PCIe/CXL fabric and obtain spec and metrics information. The information should satisfy the requirements of dynamic configuration control when combined with information obtained from hardware components.
- The ability to obtain metrics information necessary to measure the characteristics of the workload.
- The ability to obtain identifying information from both the hardware components and the operating system, ensuring that matching information can be correlated.

Optional Items

None

3.10.3 Control

Describes the requirements necessary for controlling the operating system to achieve dynamic configuration control.

Required Items

- The ability to logically connect and disconnect hardware components and recognize them.
- The ability to shut down the operating system, ensuring that the power state of hardware components is turned off.

Optional Items

- The ability to control the power-saving functions of hardware components.

3.11 PCIe/CXL Fabric

Control software implements dynamic reconfiguration of nodes through the management and control of a fabric composed of PCIe/CXL switches. The control software achieves dynamic reconfiguration of nodes by controlling the FM (Fabric Manager), which manages the PCIe/CXL switches. Therefore, the FM must have external interfaces accessible to the control software.

Dynamic configuration control is intended to be executed regardless of the operating state of the node. Therefore, the PCIe/CXL fabric needs to support hot-plugging. Control software performs hot-plugging on the operating system running on the node, then performs configuration changes (logical connection and disconnection of hardware components) on the PCIe/CXL fabric, and handles the configuration change requests by executing the necessary processes on the operating system afterward.

The requirements for PCIe/CXL fabric and FM are described below.

3.11.1 External Interface

Describes the requirements needed for the external interface of the PCIe/CXL fabric to be managed and controlled by the control software.

Required Items

- An IP-based connection to the external interface must be possible.

Optional Items

- Should be able to authenticate the source of the connection to the external interface.

3.11.2 Management

Describes the requirements necessary for managing the PCIe/CXL fabric to achieve dynamic configuration control.

Required Items

- The ability to obtain a list of PCIe/CXL switches existing in the fabric.
- The ability to obtain unique identifiers for the PCIe/CXL switches.
- The ability to obtain the ports (upstream ports, downstream ports) existing in the PCIe/CXL switches.
- The ability to obtain the logical connection state of the ports in the PCIe/CXL switches.
- The ability to obtain information about the CPU physically connected to the upstream port of the PCIe/CXL switch.
- The ability to obtain information about the hardware components physically connected to the downstream ports of the PCIe/CXL switch.
- The ability to obtain identifying information from both the information obtainable from the PCIe/CXL switches and the information obtainable from the OOB of each hardware component, ensuring that matching information can be correlated.

Optional Items

- The ability to obtain the power state of each hardware component.
- The ability to obtain the presence or absence of power control functionality for each hardware component.

3.11.3 Control

Describes the requirements necessary for controlling the PCIe/CXL fabric to achieve dynamic configuration control.

Required Items

- The ability to perform logical connection and disconnection between upstream ports (CPU) and downstream ports (each hardware component).
- Updates to the logical connection state of upstream ports (CPU) and downstream ports (each hardware component) must be possible regardless of the power state of each hardware component.
- Updates to the logical connection state of upstream ports (CPU) and downstream ports (each hardware component) must not affect other CPUs.

Optional Items

- The ability to control the power of each hardware component physically connected to the PCIe/CXL switch.

CHAPTER 4

System Implementation

The system targeted by this disaggregation technology assumes technologies that will be realized after 2030.

4.1 Reference Implementation

In the reference implementation, the configuration assumes a more general disaggregation system based on technologies expected to be realized after 2030. The reference implementation aims to achieve disaggregation control assuming more general configuration changes, beyond the current state of the art.

- The fabric is based on a network of switches consisting of CXL 3.0 or later, but assumes broader device sharing.
- It assumes large-scale fabric extension and configuration changes, and specifies the information necessary for device selection in control.
- To achieve autonomous configuration management, it defines the functions required for each external interface.
- In addition to static configuration changes, it assumes dynamic configuration control while the power is ON.

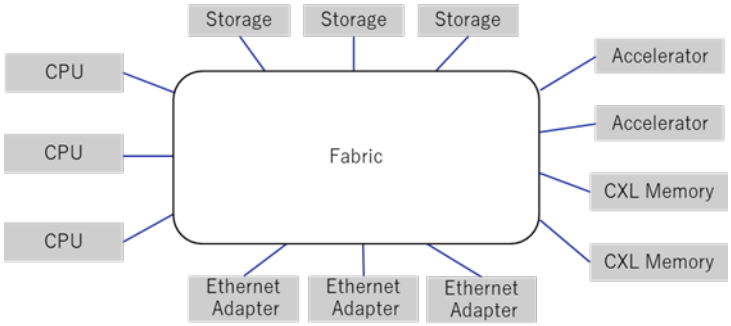

Fig. 5: Reference Implementation Configuration Image

4.1.1 CPU

- Assumes a configuration of CXL 3.0.

4.1.2 Memory (Main Memory, Dedicated Memory, CXL Memory)

- Assumes volatile and non-volatile memory.
- CXL memory assumes partitioning and sharing.

4.1.3 Storage

- Assumes a configuration of PCIe Gen6.

4.1.4 Ethernet Adapter

- Assumes the existence of Ethernet adapters in devices connected to a chassis with a CPU or PCIe/CXL switches.

4.1.5 Operating System

- CXL 3.0
  - Global Fabric Attached Memory Device
  - Dynamic Capacity Device
  - Device P2P
- PCIe 6.0
- PCIe/CXL Hot-plug, ACPI Hot-plug
- NVMe 1.4
- UEFI 2.10
- ACPI 6.5

4.1.6 PCIe/CXL Fabric

- Assumes CXL 3.0 and a fabric configuration consisting of single or multiple PCIe/CXL switches (including Spine-Leaf configurations).

CHAPTER 5

Reference Implementation

This section outlines the management and control items required for hardware components in a reference implementation.

It is assumed that the necessary management and control items can be obtained from external interfaces on the hardware components, the FM API for fabrics, or the operating system. However, it is desirable to obtain these through OOB (Out-Of-Band) or FM API that can manage and control the hardware components regardless of their power state. Additionally, it is assumed that an identifier for handling hardware components can be obtained through each interface.

5.1 CPU

5.1.1 Management

This section describes the management items for CPU devices. Management is implemented using external interfaces.

Static (Specification) Information

Table 1: Static (Specification) Information Retrieval Items for CPU

Item	Description	Required	Remarks
Type	Type of processor	Yes	
Identifier	Information that uniquely identifies the hardware component such as the processor's serial number	Yes	
Device Information	Vendor name, manufacturer, etc.	Yes	
Model	Model name or part number	Yes	
Operating Frequency	Processor's clock speed	Yes	
Number of Cores	Total number of cores in the processor	Yes	
Number of Threads	Total number of threads in the processor	Yes	
Cache Memory	Cache connected or integrated with the CPU	Yes	
Architecture	Processor architecture	Yes	Examples include x86, ARM, etc.
Instruction Set	Processor's instruction set	Yes	Examples include x86, ARM-A64, etc.
Features	Support for control operations	Yes	Information to determine whether control operations via OOB are supported
Family	Processor's CPUID		
Power Consumption	Nominal Thermal Design Power (TDP) in watts		

Dynamic (Metrics) Information

Table 2: Dynamic (Metrics) Information Retrieval Items for CPU

Item	Description	Required	Remarks
Power Consumption	Energy consumption	Yes	
Health	Status of the hardware component	Yes	
Power State	Power state	Yes	
Number of Cores	Total number of active cores in the processor		
Operating Frequency	Active clock speed		
Utilization Rate	CPU usage percentage (%)		
Core Status	Usage rate, cache hit rate, IPC		
Cache Status	Cache hit rate and miss rate		
CPU Memory Bandwidth Usage Rate	Memory/buffer bandwidth usage rate (%)		
Measurement Time	Time of metrics measurement		
Measurement Interval	Interval of metrics measurement		

5.1.2 Control

This section describes the control items for CPU devices. Control is implemented using external interfaces.

Table 3: Control Function Items for CPU

Item	Description	Required	Remarks
Power On	Power ON control	Yes	
Power Off	Power OFF control	Yes	
Reset Sensors	Reset the values of sensors that measure cumulative values		

5.2 Main/Dedicated Memory

5.2.1 Management

This section describes the management items for main/dedicated memory devices. Management is implemented using external interfaces.

Static (Specification) Information

Table 4: Static (Specification) Information Retrieval Items for Main/Dedicated Memory

Item	Description	Required	Remarks
Identifier	Information that uniquely identifies the hardware component such as the memory's serial number	Yes	
Device Information	Vendor name, manufacturer, etc.	Yes	
Model	Model name or part number	Yes	
Total Memory Capacity	Total memory capacity	Yes	
Media Type	Media type of the memory device	Yes	Definitions according to section 6.71.5.6 of [redfish_spec]
Category	Category of the memory device	Yes	Definitions according to section 6.71.5.7 of [redfish_spec]
Device Type	Detailed type of the memory device	Yes	Definitions according to section 6.71.5.10 of [redfish_spec]
Memory Clock	Operating frequency of the memory in MHz	Yes	
Features	Support for control operations	Yes	Information to determine whether control operations via OOB are supported
Volatile Memory Capacity	Capacity of the volatile memory		
Non-Volatile Memory Capacity	Capacity of the non-volatile memory		
Bus Width	Memory bus width in bits		
Data Width	Memory data width in bits		
Supported Memory Operating Frequencies			
Alignment	Alignment size		
Minimum Alignment Unit	Minimum unit of alignment		
Power Consumption	Estimated Power Consumption or Nominal Thermal Design Power (TDP) (W)	Yes	It is assumed to have values that can estimate power consumption, like the PCIe Power Budgeting Data Register in Chapter 9.8.3 of [cxl_spec].

Dynamic (Metrics) Information

Table 5: Dynamic (Metrics) Information Retrieval Items for Main/Dedicated Memory

Item	Description	Required	Remarks
Power Consumption	Energy consumption	Yes	
Hardware Health	Status of the hardware component	Yes	
Power State	Power state	Yes	
Enabled/Disabled	Indicates whether the memory device is enabled		
Usage Rate	Memory usage rate (%)		
Bandwidth Usage Rate	Bandwidth usage rate of the memory (%)		
Memory Clock	Operating frequency of the memory in MHz		
Memory Health	Memory status (DataLossDetected, LastShutdownSuccess, PerformanceDegraded)		For NVDIMM. Definitions according to section 6.74.3 of [redfish_spec]
Measurement Time	Time of metrics measurement		
Measurement Interval	Interval of metrics measurement		

5.2.2 Control

This section describes the control items for main/dedicated memory devices. Control is implemented using external interfaces.

Table 6: Control Function Items for Main/Dedicated Memory

Item	Description	Required	Remarks
Power On	Power on the hardware component		Required only if the main/dedicated memory can be power controlled individually
Power Off	Power off the hardware component		Required only if the main/dedicated memory can be power controlled individually
Data Erasure	Erase the data in the memory		Required for non-volatile memory
Reset Sensors	Reset the values of sensors that measure cumulative values		

5.3 CXL Memory

5.3.1 Management

This section describes the management items for CXL memory devices. Management is implemented using external interfaces.

Static (Specification) Information

Table 7: Static (Specification) Information Retrieval Items for CXL Memory

Item	Description	Required	Remarks
Identifier	Information that uniquely identifies the hardware component such as the memory's serial number	Yes	
Device Information	Vendor name, manufacturer, etc.	Yes	
Model	Model name or part number	Yes	
Total Memory Capacity	Total memory capacity	Yes	
Media Type	Media type of the memory device	Yes	Definitions according to section 6.71.5.6 of [redfish_spec]
Category	Category of the memory device	Yes	Definitions according to section 6.71.5.7 of [redfish_spec]
Device Type	Detailed type of the memory device	Yes	Definitions according to section 6.71.5.10 of [redfish_spec]
Memory Clock	Operating frequency of the memory in MHz	Yes	
Features	Support for control operations	Yes	Information to determine whether control operations via OOB are supported
Volatile Memory Capacity	Capacity of the volatile memory		
Non-Volatile Memory Capacity	Capacity of the non-volatile memory		
Bus Width	Memory bus width in bits		
Data Width	Memory data width in bits		
Supported Memory Operating Frequencies			
Alignment	Alignment size		
Minimum Alignment Unit	Minimum unit of alignment		
Power Consumption	Estimated Power Consumption or Nominal Thermal Design Power (TDP) (W)	Yes	It is assumed to have values that can estimate power consumption, like the PCIe Power Budgeting Data Register in Chapter 9.8.3 of [cxl_spec].

Dynamic (Metrics) Information

Table 8: Dynamic (Metrics) Information Retrieval Items for CXL Memory

Item	Description	Required	Remarks
Enabled/Disabled	Indicates whether the memory device is enabled	Yes	
Power Consumption	Energy consumption	Yes	
Measurement Time	Time of metrics measurement		
Measurement Interval	Interval of metrics measurement		
Usage Rate	Memory usage rate (%)	Yes	
Bandwidth Usage Rate	Bandwidth usage rate of the memory (%)	Yes	
Memory Clock	Operating frequency of the memory in MHz	Yes	
Health	Status of the hardware component	Yes	
Memory status	DataLossDetected, LastShutdownSuccess, PerformanceDegraded		For NVDIMM. Definitions according to section 6.74.3 of [redfish_spec]
Power State	Power state	Yes	
Link State	Link state of the PCIe port. LTSSM (Link Training Status State Machine) state	Yes	Definitions according to section 7.6.7.1.2 of [cxl_spec]

5.3.2 Control

This section describes the control items for CXL memory devices. Control is implemented using external interfaces.

Table 9: Control Function Items for CXL Memory

Item	Description	Required	Remarks
Power On	Power on the hardware component	Yes	
Power Off	Power off the hardware component	Yes	
Data Erasure	Erase the data in the memory		Required for non-volatile memory. Refer to section 8.2.9 of [cxl_spec], Events command
Reset Sensors	Reset the values of sensors that measure cumulative values		

5.4 Accelerator

5.4.1 Management

This section describes the management items for accelerator devices. Management is implemented using external interfaces.

Static (Specification) Information

Table 10: Static (Specification) Information Retrieval Items for Accelerator

Item	Description	Required	Remarks
Device Information	Vendor name, manufacturer, etc.	Yes	
Model	Model name or part number	Yes	
Operating Frequency	Processor's clock speed	Yes	
Number of Cores	Total number of cores in the processor	Yes	
Number of Threads	Total number of threads in the processor	Yes	
Identifier	Information that uniquely identifies the hardware component such as the processor's serial number	Yes	
Architecture	Processor architecture	Yes	Definitions according to section 6.106.5.15 of [redfish_spec]
Instruction Set	Processor's instruction set	Yes	Definitions according to section 6.106.5.7 of [redfish_spec]
Type	Type of processor	Yes	Definitions according to section 6.106.5.17 of [redfish_spec]
Family	Processor identification information	Yes	
Cache Memory	Memory and cache connected or integrated with the processor	Yes	
Power Consumption	Nominal Thermal Design Power (TDP) in watts		
Features	Support for control operations	Yes	Information to determine whether control operations via OOB are supported

Dynamic (Metrics) Information

Table 11: Dynamic (Metrics) Information Retrieval Items for Accelerator

Item	Description	Required	Remarks
Operating Frequency	Active clock speed		Required if dynamically changed
Number of Cores	Total number of active cores in the processor	Yes	
Utilization Rate	Processor usage percentage (%)	Yes	
Cache status	Cache misses and Hit rate		
Core status	Instructions per clock cycle and Cache occupancy rate		
Processor Memory Bandwidth	Dedicated memory bandwidth usage rate (%)		
Power Consumption	Energy consumption	Yes	
Measurement Time	Time of metrics measurement		
Measurement Interval	Interval of metrics measurement		
Health	Status of the hardware component	Yes	
Power State	Power state	Yes	

5.4.2 Control Functions

This section describes the control items for accelerator devices. Control is implemented using external interfaces.

Table 12: Control Function Items for Accelerator

Item	Description	Required	Remarks
Power On	Power ON control	Yes	
Power Off	Power OFF control	Yes	
Reset Sensors	Reset the values of sensors that measure cumulative values		

5.5 Storage

5.5.1 Management

This section describes the management items for storage devices. Management is implemented using external interfaces.

Static (Specification) Information

Table 13: Static (Specification) Information Retrieval Items for Storage

Item	Description	Required	Remarks
Device Information	Vendor name, manufacturer, etc.	Yes	
Model	Model name or part number	Yes	
Capacity	Capacity of the storage (Bytes)	Yes	
Block Size	Block size (Bytes)	Yes	
Transfer Speed	Transfer speed between the storage controller and storage (Gb/s)	Yes	
Identifier	Information that uniquely identifies the hardware component such as the storage's serial number	Yes	
Redundancy	Presence and mode of redundancy configuration		
I/O Features	Supported I/O access features		Definitions according to section 6.129.5.1 of [redfish_spec]
Power Consumption	Estimated power consumption		Assumed to be values that can estimate power consumption, such as the PCIe Power Budgeting Data Register in section 9.8.3 of [cxl_spec]
Features	Support for control operations	Yes	Information to determine whether control operations via OOB are supported
I/O Size	Optimal I/O size for the storage		
I/O Ratio	Predicted Read/Write ratio available for storage use		

Dynamic (Metrics) Information

Table 14: Dynamic (Metrics) Information Retrieval Items for Storage

Item	Description	Required	Remarks
Usage	Storage usage or available capacity	Yes	
Allocation Status (Bytes)	The capacity allocated by the storage system for data		When using thin provisioning
Used capacity (Bytes)	The amount of capacity consumed for data		When using thin provisioning
Guaranteed capacity in storage (Bytes)	The capacity guaranteed by the storage system for data		When using thin provisioning
Maximum allocatable capacity (Bytes)	The maximum capacity that can be allocated for data		When using thin provisioning
Transfer Speed	Transfer speed between the storage controller and storage (Gb/s)	Yes	
Power Consumption	Energy consumption	Yes	
Measurement Time	Time of metrics measurement		
Measurement Interval	Interval of metrics measurement		
Redundancy	Status of redundancy configuration		
Health	Status of the hardware component	Yes	
Power State	Power state	Yes	

5.5.2 Control

This section describes the control items for storage devices. Control is implemented using external interfaces.

Table 15: Control Function Items for Storage

Item	Description	Required	Remarks
Power On	Power on the hardware component		Assumes control via OOB. If power can be turned off, perform as a power-saving operation
Power Off	Power off the hardware component		Assumes control via OOB. If power can be turned off, perform as a power-saving operation
Data Erasure	Erase the data in the storage	Yes	
Reset Sensors	Reset the values of sensors that measure cumulative values		

5.6 Ethernet Adapter

5.6.1 Management

This section describes the management items for Ethernet adapter devices. Management is implemented using external interfaces.

Static (Specification) Information

Table 16: Static (Specification) Information Retrieval Items for Ethernet Adapter

Item	Description	Required	Remarks
Device Information	Vendor name, manufacturer, etc.	Yes	
Model	Model name or part number	Yes	
Transfer Speed	Transfer speed of the Ethernet adapter (bit/s)	Yes	
Identifier	Information that uniquely identifies the hardware component such as the Ethernet adapter's serial number	Yes	
Power Consumption	Estimated power consumption		Assumed to be values that can estimate power consumption, such as the PCIe Power Budgeting Data Register in section 9.8.3 of [cxl_spec]
Features	Support for control operations	Yes	Information to determine whether control operations via OOB are supported
Ethernet Information	Information of Ethernet functions belonging to the Ethernet adapter	Yes	Assumes MAC is obtained via OOB. Settings information such as IP is assumed to be obtained from the operating system
Ethernet Adapter Features	Features of the Ethernet adapter	Yes	
Bandwidth Limitation	Bandwidth limitation of the Ethernet adapter		

Dynamic (Metrics) Information

Table 17: Dynamic (Metrics) Information Retrieval Items for Ethernet Adapter

Item	Description	Required	Remarks
Transfer Speed		Yes	
Enabled/Disabled	Whether the hardware component is enabled or disabled	Yes	
MTU	Maximum Transmission Unit		
Usage Rate	Device CPU usage rate		
Host Bus RX Utilization	Host bus (e.g., PCIe) RX usage rate (percentage)		
Host Bus TX Utilization	Host bus (e.g., PCIe) TX usage rate (percentage)		
RX bytes	The number of RX bytes for the network adapter	Yes	
TX bytes	The number of TX bytes for the network adapter	Yes	
RX average queue depth	The average RX queue depth for the network adapter	Yes	
TX average queue depth	The average TX queue depth for the network adapter	Yes	
Power Consumption	Energy consumption	Yes	
Measurement Time	Time of metrics measurement		
Measurement Interval	Interval of metrics measurement		
Health	Status of the hardware component	Yes	
Power State	Power state	Yes	

5.6.2 Control

This section describes the control items for Ethernet adapter devices. Control is implemented using external interfaces.

Table 18: Control Function Items for Ethernet Adapter

Item	Description	Required	Remarks
Power On	Power on the hardware component		Required if the Ethernet adapter can be power controlled individually
Power Off	Power off the hardware component		Required if the Ethernet adapter can be power controlled individually
Reset Sensors	Reset the values of sensors that measure cumulative values		

5.7 Optical Smart NIC

This hardware component is treated as either an Ethernet adapter or a device configuring the fabric as a PCIe/CXL extender. If treated as an Ethernet adapter, it should be handled as a component based on the requirements specified in *Ethernet Adapter*.

5.8 Chassis/Rack

5.8.1 Management

This section describes the management items for chassis/rack devices. Management is implemented using external interfaces.

Static (Specification) Information

Table 19: Static (Specification) Information Retrieval Items for Chassis/Rack

Item	Description	Required	Remarks
Device Information	Vendor name, manufacturer, etc.	Yes	
Model	Model name or part number	Yes	
Identifier	Information that uniquely identifies the hardware component such as the chassis's serial number	Yes	
Power Consumption	Maximum power consumption of the chassis		
Features	Support for control operations	Yes	Information to determine whether control operations via OOB are supported
Type	Type of chassis	Yes	Definitions according to section 6.20.5.1 of [redfish_spec]
Chassis Configuration	Hardware components included in the chassis, such as CPU and memory	Yes	
Power Configuration	Number of power supplies and power capacity		

Dynamic (Metrics) Information

Table 20: Dynamic (Metrics) Information Retrieval Items for Chassis/Rack

Item	Description	Required	Remarks
Power State	Power state of the chassis		Definitions according to section 6.20.5.7 of [redfish_spec]
Power	Output power of the power supply		
Power Consumption	Energy consumption		
Measurement Time	Time of metrics measurement		
Measurement Interval	Interval of metrics measurement		

5.8.2 Control

This section describes the control items for chassis/rack devices. Control is implemented using external interfaces.

Table 21: Control Function Items for Chassis/Rack

Item	Description	Required	Remarks
Power On Chassis	Power on the chassis		Perform as a power-saving operation if power can be turned off
Power Off Chassis	Power off the chassis		Perform as a power-saving operation if power can be turned off
Power On Power Supply	Power on the power supply		Perform as a power-saving operation if power can be turned off
Power Off Power Supply	Power off the power supply		Perform as a power-saving operation if power can be turned off

5.9 PCIe/CXL Switch

5.9.1 Management

This section describes the management items for PCIe/CXL switch devices. Control is implemented using FM.

Static (Specification) Information

Table 22: Static (Specification) Information Retrieval Items for PCIe/CXL Switch

Item	Description	Required	Remarks
Device Information	Vendor name, manufacturer, etc.	Yes	
Model	Model name or part number	Yes	
Identifier	Information that uniquely identifies the hardware component, such as the chassis's serial number	Yes	
Features	Support for control operations	Yes	Used to determine if control operations via FM are supported

Dynamic (Metrics) Information

None

5.9.2 Control

This section describes the control items for PCIe/CXL switch devices. Control is implemented using FM.

Table 23: Control Function Items for PCIe/CXL Switch

Item	Description	Required	Remarks
Connect	Connect the CPU to other hardware components	Yes	Section 7.6.7.2 of [cxl_spec]
Disconnect	Disconnect the connected hardware components	Yes	Section 7.6.7.2 of [cxl_spec]
Event Notification	Receive completion notifications in case of asynchronous control		Section 8.2.9 of [cxl_spec]

CHAPTER 6

BMC/Server Management

This section describes the required items for the BMC in the reference implementation.

Hardware components have external interfaces for management and control. These external interfaces can exist at the hardware component level or at the chassis/rack level that bundles multiple hardware components. Management and control are performed according to the Redfish protocol specifications on these external interfaces.

6.1 Redfish

Each component can be controlled using the Redfish protocol via OOB. The Redfish protocol for each component must meet the following requirements:

- The Redfish protocol must communicate over HTTPS.
- The Redfish protocol version should be based on version 1.20.1 [redfish_spec].

6.1.1 Data Model List

The following models defined in the "Redfish Data Model Specification 2024.2" and "Swordfish Scalable Storage Management API Specification Version: 1.2.6" are used.

Table 24: Redfish Data Model List

Model Name	Version	Chapter Number	Notes
AccountService	1.15.1	6.2	
BootOption	1.0.6	6.15	
Chassis	1.25.1	6.20	
ComputerSystem	1.22.2	6.25	
CXLLogicalDevice	1.2.0	6.34	
Drive	1.20.0	6.35	
DriveMetrics	1.2.1	6.36	
EnvironmentMetrics	1.3.2	6.39	
EthernetInterface	1.12.2	6.40	
FabricAdapter	1.5.3	6.46	
Location	-	4.10	
Manager	1.19.1	6.66	
ManagerAccount	1.12.1	6.67	
Memory	1.20.0	6.71	
MemoryMetrics	1.7.3	6.74	
NetworkAdapter	1.11.0	6.81	

continues on next page

Table 24 – continued from previous page

Model Name	Version	Chapter Number	Notes
NetworkAdapterMetrics	1.1.0	6.82	
NetworkDeviceFunction	1.9.2	6.83	
NetworkDeviceFunctionMetrics	1.2.0	6.84	
PCIeDevice	1.15.0	6.92	
PCIeFunction	1.6.0	6.93	
Port	1.13.0	6.95	
PowerSupply	1.6.0	6.103	
Processor	1.20.1	6.106	
ProcessorMetrics	1.6.4	6.107	
Redundancy	-	4.12	
Role	1.3.2	6.112	
Sensor	1.10.0	6.118	
SensorExcerpt	-	6.137.5.3	
ServiceRoot	1.17.0	6.121	
Session	1.7.2	6.122	
SessionService	1.1.9	6.123	
Status	-	4.17	
Storage	1.17.0	6.127	
StorageController	1.7.3	6.128	
StorageControllerMetrics	1.0.3	6.129	
VirtualMedia	1.6.4	6.144	
Volume	1.10.0	6.146	
VolumeMetrics	1.1.1	9.6.42	swordfish

6.1.2 Data Model Relationships

The relationships of Redfish data are described here.

The relationships of models for the management of BMC, accounts, and sessions, regardless of the type of component, are shown in Fig. 6.

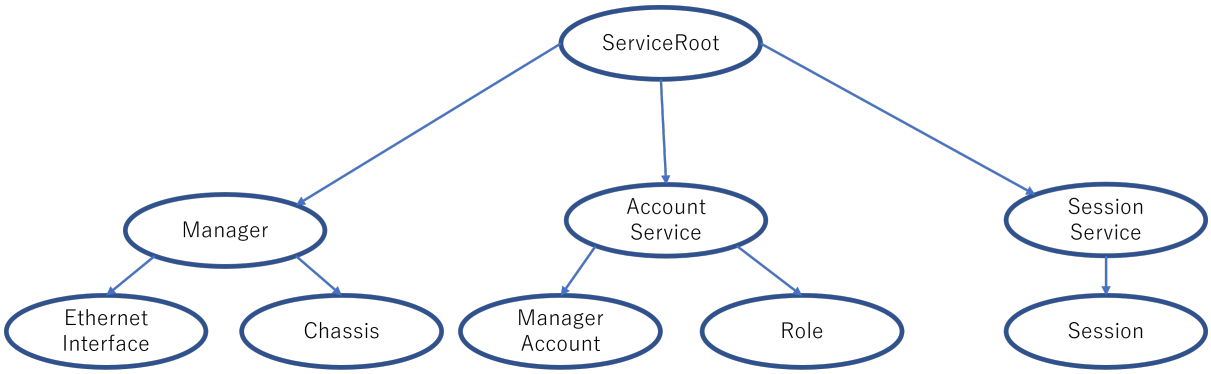

Fig. 6: Redfish Model Relationships (Common)

The relationships of models related to components are shown in Fig. 7. For each chassis equipped with devices, there is one ServiceRoot, and the models for the devices included in the chassis are related to the ServiceRoot. The diagram below describes the case where there is one each of Processor (CPU, Accelerator), Memory (main memory, CXL memory), NetworkAdapter (Ethernet adapter), and Drive (Storage).

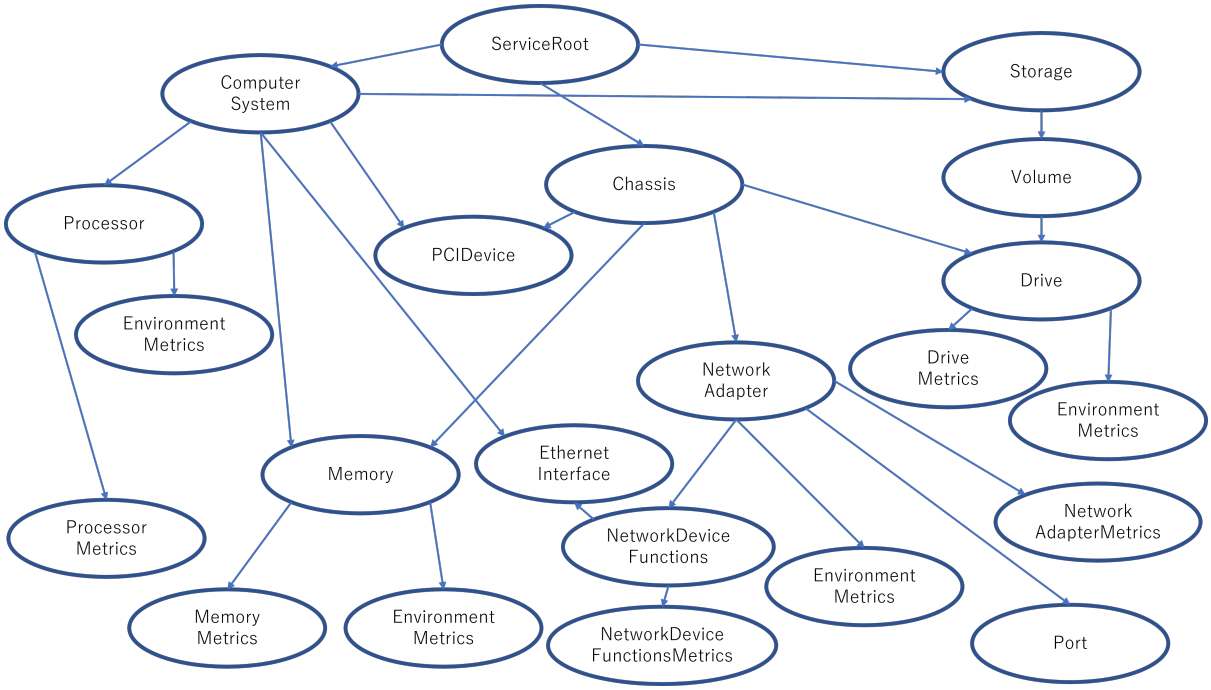

Fig. 7: Redfish Model Relationships (Components)

6.1.3 Data Model Details

The properties of each model necessary for implementing disaggregation technology are described. Properties required by Redfish specifications are not included.

AccountService

This section describes the model for the service that manages Redfish service accounts.

Table 25: AccountService Properties

Property	Re-quired	Description	Remarks
Accounts	Yes	List of BMC accounts	Refer to Section 6.1.3
Roles	Yes	List of roles	Refer to Section 6.1.3

Chassis

The model for the chassis that houses each component is described. To uniquely identify a chassis, at least one of the following properties is required:

- Location (must be unique relative to the parent model in the model relationships)
- Manufacturer/PartNumber/SerialNumber
- UUID

Table 26: Chassis Properties

Property	Required	Description	Remarks
Actions.#Chassis.Reset		Power operation at the chassis level	ResetType: On (Required), ForceOff (Required)
ChassisType	Yes	Type of chassis	
Drives		Physical disk drive	Only if there is a disk in the chassis.
EnvironmentMetrics		Metrics at the chassis level	Refer to Section 6.1.3
FabricAdapters		Fabric adapter device	Only if there is a fabric adapter in the chassis.
Links.ContainedBy		Parent chassis housing the chassis	
Links.Contains		Child chassis housed within the chassis	
Links.Drives		Physical disk drive	Only if there is a disk in the chassis.
Links.ManagedBy	Yes	Manager controlling the chassis	Refer to Section 6.1.3
Links.ManagersInChassis		BMCs located within the chassis, etc.	Refer to Section 6.1.3
Links.PowerSupplies		Power supply providing power to the chassis	Refer to Section 6.1.3
Links.PoweredBy		Power supply providing power to the chassis	Refer to Section 6.1.3
Links.Processors		CPU/GPU device	Only if there is a CPU/GPU in the chassis.
Location		Mounting position within the rack	Refer to Section 6.1.3
Manufacturer		Manufacturer	
MaxPowerWatts		Maximum power consumption	
Memory		Memory device	Only if there is memory in the chassis.
Model		Model number	
MinPowerWatts		Minimum power consumption	
NetworkAdapters		Network adapter device	Only if there is a network adapter in the chassis.
PCIeDevices	Yes	PCIe device	Only if there is a PCIe device in the chassis.
PartNumber		Part number	
PowerState		Power state	Required if power operations are supported.
Processors		CPU/GPU device	Only if there is a CPU/GPU in the chassis.
Sensors		Sensors	Refer to Section 6.1.3
SerialNumber		Serial number	
Status	Yes	HW status	Refer to Section 6.1.3
UUID		Unique identifier	

ComputerSystem

This section describes the model of a node constructed by connecting various components using fabric switches.

Table 27: ComputerSystem Properties

Property	Required	Description	Remarks
Actions.#ComputerSystem.Reset		Power control of the node	ResetType: On (Required), ForceOff (Required)
Boot.BootOptions		Boot settings	
Boot.BootOrder		Boot order	
BootProgress.LastBootTimeSeconds		Time taken for the last operating system boot	
BootProgress.LastState		Boot state	
BootProgress.LastStateTime		Time of the last state update	
EthernetInterfaces		Ethernet adapter device	Only if there is an Ethernet adapter in the chassis.
FabricAdapters		Fabric adapter device	Only if there is a fabric adapter in the chassis.
Links.ManagedBy	Yes	BMC or other device controlling the node	Refer to Section 6.1.3
Manufacturer		Manufacturer	
Memory		Memory device	Only if there is memory in the node.
MemorySummary		Memory summary	
Model		Model number	
PCIeDevices		PCIe device	Only if there is a PCIe device in the node.
PartNumber		Part number	
PowerState		Power state	Required if power operations are supported.
ProcessorSummary	Yes	CPU/GPU summary	
Processors	Yes	CPU/GPU device	
SerialNumber		Serial number	
Storage		Physical/Logical disk device	Only if there is a disk in the node.
SystemType	Yes	Type of node	
UUID		SMBIOS UUID	
VirtualMedia		Virtual media	Refer to Section 6.1.3

CXLLogicalDevice

This section describes the model of each logical device logically partitioned by the CXL's MLD function.

Table 28: CXLLogicalDevice Properties

Property	Required	Description	Remarks
Identifiers.DurableName		Identifier	
Identifiers.DurableNameFormat		Type of identifier	
MemorySizeMiB		Memory size	
QoS.AllocatedBandwidth		Bandwidth allocated to the logical device (multiple of 256)	
QoS.LimitPercent		Bandwidth limitation percentage of the logical device	
Status	Yes	Status	Refer to Section 6.1.3

Drive

The model for physical disks such as HDDs and SSDs is described. To uniquely identify a physical disk, at least one of the following properties is required:

- Location (must be unique relative to the parent model in the model relationships)
- Identifiers
- Manufacturer/PartNumber/SerialNumber

Table 29: Drive Properties

Property	Required	Description	Remarks
Actions.#Drive.Reset		Power control of the physical disk	ResetType: On (Required), ForceOff (Required)
Actions.#Drive.SecureErase		Secure erase	
BlockSizeBytes		Block size	
CapableSpeedGbs		Data transfer speed	
CapacityBytes		Capacity	
EnvironmentMetrics		Power consumption metrics	Refer to Section 6.1.3
HotspareType		Type of spare disk	Only if it is a spare disk
Identifiers.DurableName		Identifier	
Identifiers.DurableNameFormat		Type of identifier	
Links.Chassis	Yes	Chassis housing the physical disk	Refer to Section 6.1.3
Links.NetworkDeviceFunctions		Functions providing network connectivity to the physical disk	Refer to Section 6.1.3
Links.Storage		Storage associated with the physical disk	Refer to Section 6.1.3
Links.Volume		Logical disk associated with the physical disk	Refer to Section 6.1.3
Location		Mounting position within the chassis	Refer to Section 6.1.3
Manufacturer		Manufacturer	
MediaType	Yes	Type of media	
Metrics		Metrics of the physical disk	Refer to Section 6.1.3
Model		Model number	
NegotiatedSpeedGbs		Transfer speed with the storage controller	
PartNumber		Part number	
PredictedMediaLifeLeftPercent		Predicted remaining life	
SerialNumber		Serial number	
Status	Yes	HW status	Refer to Section 6.1.3

DriveMetrics

This section describes the model for the metrics of physical disks.

Table 30: DriveMetrics Properties

Property	Required	Description	Remarks
BadBlockCount		Number of bad blocks	
CorrectableIOReadErrorCount		Number of correctable read errors	
CorrectableIOWriteErrorCount		Number of correctable write errors	
NVMeSMART		SMART information	SMART attributes based on NVMe specifications
UncorrectableIOReadErrorCount		Number of uncorrectable read errors	
UncorrectableIOWriteErrorCount		Number of uncorrectable write errors	

EnvironmentMetrics

This section describes the model for the power consumption metrics of various components.

Table 31: EnvironmentMetrics Properties

Property	Required	Description	Remarks
Actions.#EnvironmentMetrics.ResetMetrics		Reset metrics	
EnergyJoules.DataSourceUri		Source of information	Refer to Section 6.1.3
EnergyJoules.Reading	Yes	Energy consumption (Joules)	

EthernetInterface

This section describes the model for an Ethernet adapter.

Table 32: EthernetInterface Properties

Property	Required	Description	Remarks
EthernetInterfaceType		Type of adapter	
IPv4Addresses		IPv4 allocation status	
IPv6Addresses		IPv6 allocation status	
InterfaceEnabled		Enabled/Disabled	
LinkStatus		Link status	
Links.Chassis		Chassis housing the ethernet interface	Refer to Section 6.1.3
Links.NetworkDeviceFunctions		Related network functions	Refer to Section 6.1.3
Links.Ports		Related Ports	Refer to Section 6.1.3
MACAddress		MAC address	
MTUSize		MTU	
PermanentMACAddress		Permanent MAC address	
SpeedMbps		Transfer speed	
Status		HW status	Refer to Section 6.1.3

FabricAdapter

The model for the fabric adapter is described. To uniquely identify a fabric adapter, at least one of the following properties is required:

- Location (must be unique relative to the parent model in the model relationships)
- Manufacturer/PartNumber/SerialNumber
- UUID

Table 33: FabricAdapter Properties

Property	Required	Description	Remarks
FabricType	Yes	Type of fabric	
Location		Mounting position within the chassis	Refer to Section 6.1.3
Manufacturer		Manufacturer	
Model		Model number	
PartNumber		Part number	
Ports		Related Ports	Refer to Section 6.1.3
SerialNumber		Serial number	
Status	Yes	HW status	Refer to Section 6.1.3
UUID		Unique identifier	

Location

This section describes the model for the location of hardware components.

Table 34: Location Properties

Property	Required	Description	Remarks
PartLocation.LocationOrdinalValue	Yes	Value indicating the location	
PartLocation.LocationType	Yes	Type of location	
PartLocationContext		Description of the location	

Manager

The model for the BMC (Baseboard Management Controller) is described. To uniquely identify a BMC, at least one of the following properties is required:

- Location (must be unique relative to the parent model in the model relationships)
- Manufacturer/PartNumber/SerialNumber
- UUID

Table 35: Manager Properties

Property	Required	Description	Remarks
EthernetInterfaces		Ethernet configuration	Refer to Section 6.1.3
Links.ManagerForChassis		Chassis managed by the manager	Refer to Section 6.1.3
Links.ManagerForServers		Nodes managed by the manager	Refer to Section 6.1.3
Links.ManagerInChassis		Chassis housing the manager	Refer to Section 6.1.3
Location		Mounting position within the chassis	Refer to Section 6.1.3
ManagerType	Yes	Type of manager	
Manufacturer		Manufacturer	
Model		Model number	
PartNumber		Part number	
PowerState		Power state	
SerialNumber		Serial number	
UUID		Unique identifier	

ManagerAccount

This section describes the model for a BMC (Baseboard Management Controller) account. There must be at least one account of the Redfish type.

Table 36: ManagerAccount Properties

Property	Required	Description	Remarks
AccountTypes	Yes	Type of account	
UserName	Yes	Name of the account	

Memory

The model for memory, including main memory and CXL Type 3 devices, is described. For memory devices other than those integrated into CPU or GPU devices, at least one of the following properties is required to uniquely identify them:

- Location (must be unique relative to the parent model in the model relationships)
- Manufacturer/PartNumber/SerialNumber

Table 37: Memory Properties

Property	Required	Description	Remarks
Actions.#Memory.Reset		Power control of the memory device	ResetType: On (Required), ForceOff (Required)
Actions.#Memory.SecureEraseUnit		Erase the contents of memory	
AllocationAlignmentMiB		Alignment size	
AllocationIncrementMiB		Minimum allocation size of memory	

continues on next page

Table 37 – continued from previous page

Property	Required	Description	Remarks
AllowedSpeedsMHz		Supported clock speeds	
BusWidthBits		Bus width	
CXL.LabelStorageSizeBytes		LSS capacity	
CXL.StagedNonVolatileSizeMiB		Total non-volatile capacity	
CXL.StagedVolatileSizeMiB		Total volatile capacity	
CapacityMiB		Capacity	
DataWidthBits		Data width	
Enabled		Enabled/Disabled status	
EnvironmentMetrics		Power consumption metrics	Refer to Section 6.1.3
Links.Chassis	Yes	Chassis housing the memory	Refer to Section 6.1.3
Links.Processors		Processors associated with the memory	Refer to Section 6.1.3
Location		Mounting position within the chassis	Refer to Section 6.1.3
Manufacturer		Manufacturer	
MaxTDPMilliWatts		TDP (Thermal Design Power)	
MemoryDeviceType	Yes	Type of memory device	Type as defined by SMBIOS
MemoryMedia	Yes	Type of media	
MemoryType	Yes	Type of memory	
Metrics		Metrics of the memory	Refer to Section 6.1.3
Model		Model number	
NonVolatileSizeMiB	Yes	Capacity of non-volatile region	Zero if no non-volatile region
OperatingSpeedMhz		Operating clock speed	
PartNumber		Part number	
PersistentRegionNumberLimit		Maximum number of persistent regions	
PersistentRegionSizeLimitMiB		Maximum capacity of persistent regions	
PersistentRegionSizeMaxMiB		Maximum capacity of a single persistent region	
SerialNumber		Serial number	
Status	yes	HW status	Refer to Section 6.1.3
VolatileRegionNumberLimit		Maximum number of volatile regions	
VolatileRegionSizeLimitMiB		Maximum capacity of volatile regions	
VolatileRegionSizeMaxMiB		Maximum capacity of a volatile region	
VolatileSizeMiB	yes	Capacity of volatile region	0 if there is no volatile region

MemoryMetrics

This section describes the model for metrics associated with memory.

Table 38: MemoryMetrics Properties

Property	Required	Description	Remarks
BandwidthPercent		Bandwidth utilization	
BlockSizeBytes		Block size	
CapacityUtilizationPercent		Utilization rate	
HealthData.DataLossDetected		Indicates if data loss was detected	
HealthData.LastShutdownSuccess		Indicates if the last shutdown was successful	
HealthData.PerformanceDegraded		Indicates if performance is degraded	
HealthData.PredictedMediaLifeLeftPercent		Predicted life remaining	
OperatingSpeedMHz		Operating clock speed	

NetworkAdapter

The model for the network adapter is described. For network adapter devices other than those integrated into CPU or GPU devices, at least one of the following properties is required to uniquely identify them:

- Location (must be unique relative to the parent model in the model relationships)
- Manufacturer/PartNumber/SerialNumber

Table 39: NetworkAdapter Properties

Property	Required	Description	Remarks
Actions.#NetworkAdapter.Reset		Power control for the network adapter device	ResetType: On (required), ForceOff (required)
Controllers.ControllerCapabilities		Capabilities of the network adapter	
EnvironmentMetrics		Metrics for power consumption	Refer to Section 6.1.3
Location		Installation position within the chassis	Refer to Section 6.1.3
Manufacturer		Manufacturer	
Metrics		Metrics for the network adapter	Refer to Section 6.1.3
Model		Model number	
NetworkDeviceFunctions		Network functions	Refer to Section 6.1.3
PartNumber		Part number	
Ports	Yes	Ports on the network adapter	Refer to Section 6.1.3
SerialNumber		Serial number	
Status	Yes	Hardware status	Refer to Section 6.1.3

NetworkAdapterMetrics

The model for metrics associated with the network adapter is described.

Table 40: NetworkAdapterMetrics Properties

Property	Required	Description	Remarks
Actions.#NetworkAdapterMetrics.ResetMetrics		Reset metrics	
CPUCorePercent		DPU usage rate	
HostBusRXPercent		Host bus receive usage rate	
HostBusTXPercent		Host bus transmit usage rate	
RXBytes		Number of received bytes	
TXBytes		Number of transmitted bytes	

NetworkDeviceFunction

This section describes the model for network functions associated with a network adapter.

Table 41: NetworkDeviceFunction Properties

Property	Required	Description	Remarks
AssignablePhysicalNetwork-Ports	Yes	Associated Port	Refer to Section 6.1.3
DeviceEnabled		Enable/Disable	
Ethernet.EthernetInterfaces		Associated EthernetInterface	Refer to Section 6.1.3
Ethernet.MACAddress	Yes	MAC address	
Ethernet.MTUSize		MTU	
Ethernet.MTUSizeMaximum		Maximum MTU	
Ether-net.PermanentMACAddress	Yes	Permanent MAC address	
Ethernet.VLAN		VLAN	
Metrics		Metrics of the network function	Refer to Section 6.1.3
Limits.BurstBytesPerSecond		Burst size (bps)	
Limits.BurstPacketsPerSecond		Burst size (packets)	
Limits.Direction		Bandwidth control direction	
Lim-its.SustainedBytesPerSecond		Sustained bandwidth (bps)	
Lim-its.SustainedPacketsPerSecond		Sustained bandwidth (packets)	
Status	Yes	HW status	Refer to Section 6.1.3

NetworkDeviceFunctionMetrics

The model for metrics associated with network functions is described.

Table 42: NetworkDeviceFunctionMetrics Properties

Property	Required	Description	Remarks
Actions.#NetworkDeviceFunctionMetrics.ResetMetrics		Reset metrics	
RXAvgQueueDepthPercent		Receive queue depth usage rate	
RXBytes		Number of received bytes	
RXFrames		Number of received frames	
TXAvgQueueDepthPercent		Transmit queue depth usage rate	
TXBytes		Number of transmitted bytes	
TXFrames		Number of transmitted frames	

PCIeDevice

The model for PCIe/CXL devices such as memory and physical disks is described. PCIe/CXL devices connected to a fabric switch are associated with both their respective component models (e.g., Disk, Memory) and PCIeDevice. To uniquely identify a PCIe device and its corresponding component model, at least one of the following properties is required and must match the value in the component model:

- Slot.Location (must be unique relative to the parent model in the model relationships)
- Manufacturer/PartNumber/SerialNumber

Table 43: PCIeDevice Properties

Property	Required	Description	Remarks
CXLDevice.DeviceType		Type of CXL device	
CXLDevice.MaxNumberLogicalDevices		Maximum number of MLDs (Multiple Logical Devices)	
CXLLogicalDevices		MLDs (Multiple Logical Devices)	Refer to Section 6.1.3
DeviceType		Type of device	
Links.Chassis	Yes	Chassis housing the PCIe device	
Manufacturer		Manufacturer	
Model		Model number	
PartNumber		Part number	
PCIeFunctions		PCI functions	Refer to Section 6.1.3
PCIeInterface.LanesInUse		Number of lanes in use by the device	
PCIeInterface.MaxLanes		Maximum number of supported lanes	
PCIeInterface.MaxPCIeType		Supported PCIe generation	
PCIeInterface.PCIeType		PCIe generation	
SerialNumber		Serial number	
Slot.Lanes		Number of lanes supported by the slot	
Slot.Location		Mounting position of the slot	Refer to Section 6.1.3
Slot.PCIeType		PCIe generation supported by the slot	
Slot.SlotType		Type of slot	

PCIeFunctions

The model for PCIe functions related to PCIe/CXL devices is described.

Table 44: PCIeFunctions Properties

Property	Required	Description	Notes
BusNumber		PCI bus number (in hexadecimal)	
ClassCode	Yes	PCI class code (in hexadecimal)	
DeviceClass		Corresponding value for PCI class code	
DeviceId	Yes	PCI Device ID	
DeviceNumber	Yes	PCI device number (in hexadecimal)	
FunctionId		PCI function	
FunctionNumber	Yes	PCI function number (in hexadecimal)	
FunctionProtocol	Yes	PCIe or CXL	
FunctionType	Yes	Physical or virtual	
SegmentNumber	Yes	PCI segment/domain number (in hexadecimal)	
VendorId	Yes	PCI vendor code	

Port

This section describes the model for ports associated with network adapters and fabric adapters.

Table 45: Port Properties

Property	Required	Description	Remarks
CXL.ConnectedDeviceMode		Connected Device Mode as defined by the CXL specification	
CXL.ConnectedDeviceType		Connected Device Type as defined by the CXL specification	
CXL.CurrentPortConfigurationState		Current Port Configuration State as defined by the CXL specification	
CXL.MaxLogicalDeviceCount		Supported Logical Device Count as defined by the CXL specification	
CurrentSpeedGbps		Current transfer speed	
EnvironmentMetrics		Power consumption metrics	Refer to Section 6.1.3
FunctionMaxBandwidth.AllocationPercent		Maximum percentage of bandwidth allocated to the network function	
FunctionMaxBandwidth.NetworkDeviceFunction		Network function	Refer to Section 6.1.3
FunctionMinBandwidth AllocationPercent		Minimum percentage of bandwidth allocated to the network function	
FunctionMinBandwidth.NetworkDeviceFunction		Network function	Refer to Section 6.1.3
LinkConfiguration.CapableLinkSpeedGbps		Capable transfer speed	
LinkStatus		Link status	
MaxSpeedGbps		Maximum transfer speed	
PortType	Yes	Type of port	
Status	Yes	HW status	Refer to Section 6.1.3

PowerSupply

This section describes the model for power supplies associated with a chassis.

Table 46: PowerSupply Properties

Property	Required	Description	Remarks
Action.#PowerSupply.Reset		Power control of the power supply	ResetType: On (Required), ForceOff (Required)
Links.PoweringChassis	Yes	Chassis being powered	Refer to Section 6.1.3

Processor

This section describes the model for processors such as CPUs and GPUs. To uniquely identify a processor, at least one of the following properties must be required:

- Manufacturer/PartNumber/SerialNumber
- Socket (must be unique relative to the higher-level model in the tree structure)

Table 47: Processor Properties

Property	Required	Description	Remarks
Action.#Processor.Reset	Yes	Power control of the processor	ResetType: On (Required), ForceOff (Required), ForceRestart (Optional): CPU Reset
BaseSpeedMHz		Operating frequency (specification)	
Enabled		Enabled/Disabled	
EnvironmentMetrics		Power consumption metrics	Refer to Section 6.1.3
Family		Processor family	
InstructionSet		Instruction set	
Links.Chassis	Yes	Chassis housing the processor	Refer to Section 6.1.3
Links.ConnectedProcessors		Processors connected to this processor	Refer to Section 6.1.3
Links.FabricAdapters		Adapters for connecting the processor to a fabric	Refer to Section 6.1.3
Links.Memory		Memory associated with the processor	Refer to Section 6.1.3
Links.NetworkDeviceFunctions		NIC functions associated with a DPU (Data Processing Unit) such as a SmartNIC	Refer to Section 6.1.3
Links.PCIeDevice		PCIe devices associated with the processor	Refer to Section 6.1.3
Manufacturer		Manufacturer	
MemorySummary.Metrics		Memory metrics	Refer to Section 6.1.3
MemorySummary.TotalCacheSizeMiB		Total cache size	
MemorySummary.TotalMemorySizeMiB		Total memory size	
Metrics		Processor metrics	Refer to Section 6.1.3
Model		Model number	
OperatingSpeedMHz		Operating speed (MHz)	
PartNumber		Part number	
PowerState	yes	Power state	
ProcessorArchitecture		Processor architecture	
ProcessorId.EffectiveFamily		Processor family	ProcessorId is information obtained via the CPUID instruction
ProcessorId.EffectiveModel		Processor model	
ProcessorId.IdentificationRegisters		Registers	
ProcessorId.MicrocodeInfo		Microcode	

continues on next page

Table 47 – continued from previous page

Property	Required	Description	Remarks
ProcessorId.ProtectedIdentificationNumber		PPIN	
ProcessorId.Step		Processor stepping or revision	
ProcessorId.VendorId		Processor vendor ID	
ProcessorMemory.CapacityMiB		Capacity of memory or cache	ProcessorMemory refers to memory or cache that is directly attached or integrated with the processor
ProcessorMemory.IntegratedMemory		Whether it is directly attached or integrated	
ProcessorMemory.MemoryType	yes	Type of memory	
ProcessorMemory.SpeedMHz		Memory clock speed (MHz)	
ProcessorType	yes	Type of processor	
SerialNumber		Serial number	
Socket		Socket position	
Status	yes	HW status	Refer to Section 6.1.3
TDPWatts		TDP (Watts)	
TotalCores		Number of cores	
TotalEnabledCores		Number of enabled cores	
TotalThreads		Number of threads	

ProcessorMetrics

This section describes the model for metrics associated with processors.

Table 48: ProcessorMetrics Properties

Property	Required	Description	Remarks
BandwidthPercent		CPU utilization	
Cache.CacheMiss		Number of cache misses (in millions)	
Cache.CacheMissesPerInstruction		Cache misses per instruction	
Cache.HitRatio		Cache hit ratio	
Cache.Level		Cache level	
Cache.OccupancyBytes		Cache occupancy (in bytes)	
Cache.OccupancyPercent		Cache occupancy percentage	
CoreMetrics.CoreCache		CPU core-level cache metrics	Includes the same properties as Cache.*
CoreMetrics.CoreId	Yes	Identifier for the CPU core	
CoreMetrics.InstructionsPerCycle		Instructions per cycle	
KernelPercent		Kernel utilization	
LocalMemoryBandwidthBytes		Local memory bandwidth (in bytes)	
OperatingSpeedMHz		Operating frequency	
RemoteMemoryBandwidthBytes		Remote memory bandwidth (in bytes)	
UserPercent		User utilization	

Redundancy

This section describes the model for redundancy settings.

Table 49: Redundancy Properties

Property	Required	Description	Remarks
Mode	Yes	Redundancy mode	
RedundancySet		Redundancy configuration	IDs of members in the redundancy set
Status	Yes	Status	Refer to Status

Role

This section describes the model for a role. There must be a role that signifies an administrator.

Table 50: Role Properties

Property	Required	Description	Remarks
Name	Yes	Name of the role	

Sensor

This section describes the model for a sensor.

Table 51: Sensor Properties

Property	Required	Description	Remarks
Reading	Yes	Sensor value	
ReadingTime	Yes	Time the value was read	
ReadingType	Yes	Type of value	

ServiceRoot

This section describes the model for the root object of a Redfish service.

Table 52: ServiceRoot Properties

Property	Required	Description	Remarks
AccountService	Yes	Account management	Refer to Section 6.1.3
Chassis	Yes	List of chassis	Refer to Section 6.1.3
Managers	Yes	List of managers	Refer to Section 6.1.3
SessionService	Yes	Session management	Refer to Section 6.1.3
Storage		List of storage devices	Refer to Section 6.1.3
Systems		List of nodes	Refer to Section 6.1.3

Session

This section describes the model for a session.

Table 53: Session Properties

Property	Required	Description	Remarks
SessionType	Yes	Type of session	
UserName	Yes	Name of the account	

SessionService

This section describes the model for the service that manages Redfish sessions.

Table 54: SessionService Properties

Property	Required	Description	Remarks
Sessions	Yes	List of sessions	

Status

This section describes the model for the status of hardware and software.

Table 55: Status Properties

Property	Required	Description	Remarks
Health	Yes	Health status when there are no dependent resources	
State	Yes	Resource state	

Storage

This section describes the model for a storage system.

Table 56: Storage Properties

Property	Required	Description	Remarks
Controllers		List of storage controllers	Refer to Section 6.1.3
Drives	Yes	List of physical disks	Refer to Section 6.1.3
Redundancy		Redundancy	Refer to Section 6.1.3
Status	Yes	Status	Refer to Section 6.1.3
Volumes	Yes	List of logical disks	Refer to Section 6.1.3

StorageController

This section describes the model for a storage controller.

Table 57: StorageController Properties

Property	Required	Description	Remarks
Metrics		Metrics for the storage controller	Refer to Section 6.1.3
SpeedGbps		Transfer speed (in Gbps)	

StorageControllerMetrics

This section describes the model for metrics for a storage controller.

Table 58: StorageControllerMetrics Properties

Property	Required	Description	Remarks
NVMeSMART		SMART information	SMART attributes based on NVMe specifications

VirtualMedia

This section describes the model for virtual media located on a node's BMC (Baseboard Management Controller).

Table 59: VirtualMedia Properties

Property	Required	Description	Remarks
Actions.#VirtualMedia.EjectMedia	Yes	Eject media	
Actions.#VirtualMedia.InsertMedia	Yes	Insert media	
ConnectedVia	Yes	Location of the media	
Inserted	Yes	Whether media is inserted	
MediaTypes	Yes	Types of media	
Status	Yes	Status	Refer to Section 6.1.3
TransferMethod	Yes	Transfer method	
TransferProtocolType	Yes	Transfer protocol	

Volume

This section describes the model for a logical disk. To identify a logical disk, at least one of the following properties is required:

- DisplayName
- Identifiers

Table 60: Volume Properties

Property	Required	Description	Remarks
AccessCapabilities		Accessibility such as Read, Write, etc.	
BlockSizeBytes		Block size	
Capacity.Data.AllocatedBytes		Capacity allocated by the storage system for data	
Capacity.Data.ConsumedBytes		Data consumption	
Capacity.Data.GuaranteedBytes		Capacity guaranteed by the storage system for data	
Capacity.Data.ProvisionedBytes		Maximum capacity that can be allocated for data	
Capacity.Metadata.AllocatedBytes		Capacity allocated by the storage system for metadata	
Capacity.Metadata.ConsumedBytes		Metadata consumption	
Capacity.Metadata.GuaranteedBytes		Capacity guaranteed by the storage system for metadata	
Capacity.Metadata.ProvisionedBytes		Maximum capacity that can be allocated for metadata	
CapacityBytes		Capacity	
DisplayName		Name	
Identifiers.DurableName		Identifier	
Identifiers.DurableNameFormat		Identifier type	
MaxBlockSizeBytes		Maximum block size	
Metrics		Metrics for the logical disk	Refer to Section 6.1.3
OptimumIOSizeBytes		Optimal I/O size in bytes	
RAIDType		RAID type	
RecoverableCapacitySourceCount		Number of resources available as substitutes	
RemainingCapacityPercent		Remaining capacity	
Status	Yes	Status	Refer to Section 6.1.3
VolumeType	Yes	Volume type	
VolumeUsage		Usage method	

VolumeMetrics

This section describes the model for metrics for a logical disk.

Table 61: VolumeMetrics Properties

Property	Required	Description	Remarks
CorrectableIOReadErrorCount		Number of correctable read errors	
CorrectableIOWriteErrorCount		Number of correctable write errors	
UncorrectableIOReadErrorCount		Number of uncorrectable read errors	
UncorrectableIOWriteErrorCount		Number of uncorrectable write errors	

Additionally, to control the interaction between the Redfish interface and PCIe/CXL fabric, a method for identifying components across each external interface is required.

6.1.4 URI

The following APIs are used for information retrieval and control. The prefix /redfish/v1 is omitted.

Table 62: Redfish URI

HTTP Method	URI	Description
Get	/	Retrieve ServiceRoot.
Get	/AccountService	Retrieve AccountService.
Get	/AccountService/Accounts	Retrieve ManagerAccount for the specified account.
Get	/AccountService/Accounts/{AccountId}	Retrieve a list of roles.
Get	/AccountService/Roles	Retrieve the specified role.
Get	/AccountService/Roles/{RoleId}	Retrieve the specified role.
Get	/Chassis	Retrieve a list of chassis.
Get	/Chassis/{ChassisId}	Retrieve the specified chassis.
Post	/Chassis/{ChassisId}/Actions/Chassis.Reset	Perform a power operation on the chassis.
Get	/Chassis/{ChassisId}/Drives	Retrieve a list of physical disks in the chassis.
Get	/Chassis/{ChassisId}/Drives/{DeviceId}	Retrieve the specified drive.
Post	/Chassis/{ChassisId}/Drives/{DeviceId} /Actions/Drive.Reset	Perform a power operation on the physical disk.
Post	/Chassis/{ChassisId}/Drives/{DeviceId}/Actions /Drive.SecureErase	Erase the contents of the physical disk.
Get	/Chassis/{ChassisId}/Drives/{DeviceId}/EnvironmentMetrics	Retrieve the specified EnvironmentMetrics.
Post	/Chassis/{ChassisId}/Drives/{DeviceId}/EnvironmentMetrics /EnvironmentMetrics.ResetMetrics	Reset the cumulative values stored as metric values.
Get	/Chassis/{ChassisId}/Drives/{DeviceId}/Metrics	Retrieve the specified DriveMetrics.
Get	/Chassis/{ChassisId}/EnvironmentMetrics	Retrieve the specified EnvironmentMetrics.
Post	/Chassis/{ChassisId}/EnvironmentMetrics /EnvironmentMetrics.ResetMetrics	Reset the cumulative values stored as metric values.
Get	/Chassis/{ChassisId}/FabricAdapters	Retrieve a list of fabric adapters in the chassis.
Get	/Chassis/{ChassisId}/FabricAdapters/{FabricAdapterId}	Retrieve the specified fabric adapter.
Get	/Chassis/{ChassisId}/Memory	Retrieve a list of memory modules in the chassis.
Get	/Chassis/{ChassisId}/Memory/{MemoryId}	Retrieve the specified memory.

continues on next page

Table 62 – continued from previous page

HTTP Method	URI	Description
Post	/Chassis/{ChassisId}/Memory/{MemoryId}/Actions/Memory.Reset	Perform a power operation on the memory.
Post	/Chassis/{ChassisId}/Memory/{MemoryId}/Actions/Memory.SecureEraseUnit	Erase the contents of the memory.
Get	/Chassis/{ChassisId}/Memory/{MemoryId}/EnvironmentMetrics	Retrieve the specified EnvironmentMetrics.
Post	/Chassis/{ChassisId}/Memory/{MemoryId}/EnvironmentMetrics/EnvironmentMetrics.ResetMetrics	Reset the cumulative values stored as metric values.
Get	/Chassis/{ChassisId}/Memory/{MemoryId}/MemoryMetrics	Retrieve the specified MemoryMetrics.
Get	/Chassis/{ChassisId}/NetworkAdapters	Retrieve a list of network adapters in the chassis.
Get	/Chassis/{ChassisId}/NetworkAdapters/{NetworkAdapterId}	Retrieve the specified NetworkAdapter.
Get	/Chassis/{ChassisId}/NetworkAdapters/{NetworkAdapterId}/EnvironmentMetrics	Retrieve the specified EnvironmentMetrics.
Post	/Chassis/{ChassisId}/NetworkAdapters/{NetworkAdapterId}/EnvironmentMetrics/EnvironmentMetrics.ResetMetrics	Reset the cumulative values stored as metric values.
Get	/Chassis/{ChassisId}/NetworkAdapters/{NetworkAdapterId}/Metrics	Retrieve the specified NetworkAdapterMetrics.
Get	/Chassis/{ChassisId}/NetworkAdapters/{NetworkAdapterId}/NetworkDeviceFunctions	Retrieve a list of network device functions in the network adapter.
Get	/Chassis/{ChassisId}/NetworkAdapters/{NetworkAdapterId}/NetworkDeviceFunctions/{NetworkDeviceFunctionId}	Retrieve the specified network device function.
Get	/Chassis/{ChassisId}/NetworkAdapters/{NetworkAdapterId}/NetworkDeviceFunctions/{NetworkDeviceFunctionId}/Metrics	Retrieve the specified NetworkDeviceFunctionMetrics.
Get	/Chassis/{ChassisId}/NetworkAdapters/{NetworkAdapterId}/Ports	Retrieve a list of ports in the network adapter.

continues on next page

Table 62 – continued from previous page

HTTP Method	URI	Description
Get	/Chassis/{ChassisId}/NetworkAdapters/{NetworkAdapterId} /Ports/{PortId}	Retrieve the specified port.
Get	/Chassis/{ChassisId}/PCIeDevices	Retrieve a list of PCIe devices in the chassis.
Get	/Chassis/{ChassisId}/PCIeDevices/{PCIeDeviceId}	Retrieve the specified PCIeDevice.
Get	/Chassis/{ChassisId}/PCIeDevices/{PCIeDeviceId} /CXLLogicalDevices	Retrieve a list of logical devices in the PCIe device.
Get	/Chassis/{ChassisId}/PCIeDevices/{PCIeDeviceId} /CXLLogicalDevices/{CXLLogicalDeviceId}	Retrieve the specified CXLLogicalDevice.
Get	/Chassis/{ChassisId}/PCIeDevices/{PCIeDeviceId} /PCIeFunctions	Retrieve a list of functions in the PCIe device.
Get	/Chassis/{ChassisId}/PCIeDevices/{PCIeDeviceId} /PCIeFunctions/{PCIeFunctionId}	Retrieve the specified PCIeFunction.
Get	/Chassis/{ChassisId}/PowerSubsystem/PowerSupplies /{PowerSupplyId}	Retrieve the specified PowerSupply.
Post	/Chassis/{ChassisId}/PowerSubsystem/PowerSupplies /{PowerSupplyId}/Actions/PowerSupply.Reset	Perform a power operation on the power supply.
Get	/Chassis/{ChassisId}/Sensors/	Retrieve a list of power circuits in the chassis.
Get	/Chassis/{ChassisId}/Sensors/{SensorId}	Retrieve the specified sensor.
Get	/Managers	Retrieve a list of BMCs.
Get	/Managers/{ManagerId}	Retrieve the specified Manager.
Get	/Managers/{ManagerId}/EthernetInterfaces	Retrieve a list of Ethernet adapters in the Manager.
Get	/Managers/{ManagerId}/EthernetInterfaces /{EthernetInterfaceId}	Retrieve the specified EthernetInterface.
Get	/Managers/{ManagerId}/VirtualMedia	Retrieve a list of virtual media in the BMC.
Get	/Managers/{ManagerId}/VirtualMedia/{VirtualMediaId}	Retrieve the specified VirtualMedia in the BMC.
Post	/Managers/{ManagerId}/VirtualMedia/{VirtualMediaId} /Actions/VirtualMedia.EjectMedia	Eject the virtual media.

continues on next page

Table 62 – continued from previous page

HTTP Method	URI	Description
Post	/Managers/{ManagerId}/VirtualMedia/{VirtualMediaId} /Actions/VirtualMedia.InsertMedia	Insert the virtual media.
Get	/SessionService	Retrieve the SessionService.
Get	/SessionService/Sessions	Retrieve a list of sessions.
Get	/SessionService/Sessions/{SessionId}	Retrieve the specified session.
Get	/Storage	Retrieve a list of storage systems.
Get	/Storage/{StorageId}	Retrieve the specified storage.
Get	/Storage/{StorageId}/Controllers	Retrieve a list of storage controllers in the storage system.
Get	/Storage/{StorageId}/Controllers/{ControllerId}	Retrieve the specified storage controller.
Get	/Storage/{StorageId}/Volumes	Retrieve a list of volumes in the storage system.
Get	/Storage/{StorageId}/Volumes/{VolumeId}	Retrieve the specified volume.
Get	/Storage/{StorageId}/Volumes/{VolumeId}/Metrics	Retrieve the specified volume metrics.
Get	/Systems	Retrieve a list of nodes.
Get	/Systems/{SystemId}	Retrieve the specified computer system.
Put	/Systems/{SystemId}	Update the value of Boot.BootOrder.
Post	/Systems/{SystemId}/Actions/ComputerSystem.Reset	Perform a power operation on the node.
Get	/Systems/{SystemId}/BootOptions	Retrieve a list of bootable devices.
Get	/Systems/{SystemId}/BootOptions/{BootOptionId}	Retrieve the specified boot option.
Get	/Systems/{SystemId}/EthernetInterfaces	Retrieve a list of Ethernet adapters in the node.
Get	/Systems/{SystemId}/EthernetInterfaces /{EthernetInterfaceId}	Retrieve the specified Ethernet interface.
Get	/Systems/{SystemId}/FabricAdapters	Retrieve a list of fabric adapters in the node.
Get	/Systems/{SystemId}/FabricAdapters/{FabricAdapterId}	Retrieve the specified fabric adapter.
Get	/Systems/{SystemId}/Memory	Retrieve a list of memory modules in the node.
Get	/Systems/{SystemId}/Memory/{MemoryId}	Retrieve the specified memory.
Post	/Chassis/{ChassisId}/Memory/{MemoryId}/Actions /Memory.Reset	Perform a power operation on the memory.
Post	/Chassis/{ChassisId}/Memory/{MemoryId}/Actions /Memory.SecureEraseUnit	Erase the contents of the memory.
Get	/Systems/{SystemId}/Memory/{MemoryId}/EnvironmentMetrics	Retrieve the specified environment metrics.

continues on next page

Table 62 – continued from previous page

HTTP Method	URI	Description
Post	/Systems/{SystemId}/Memory/{MemoryId} /EnvironmentMetrics/EnvironmentMetrics.ResetMetrics	Reset the cumulative values stored as metric values.
Get	/Systems/{SystemId}/Memory/{MemoryId}/MemoryMetrics	Retrieve the specified memory metrics.
Get	/Systems/{SystemId}/PCIeDevices	Retrieve a list of PCIe devices in the node.
Get	/Systems/{SystemId}/PCIeDevices/{PCIeDeviceId}	Retrieve the specified PCIe device.
Get	/Systems/{SystemId}/PCIeDevices/{PCIeDeviceId} /PCIeFunctions	Retrieve a list of functions in the PCIe device.
Get	/Systems/{SystemId}/PCIeDevices/{PCIeDeviceId} /PCIeFunctions/{PCIeFunctionId}	Retrieve the specified PCIe function.
Get	/Systems/{SystemId}/Processors	Retrieve a list of processors in the node.
Get	/Systems/{SystemId}/Processors/{ProcessorId}	Retrieve the specified processor.
Post	/Systems/{SystemId}/Processors/{ProcessorId} /Actions/Processor.Reset	Perform a power operation on the CPU.
Get	/Systems/{SystemId}/Processors/{ProcessorId} /EnvironmentMetrics	Retrieve the specified environment metrics.
Post	/Systems/{SystemId}/Processors/{ProcessorId} /EnvironmentMetrics/EnvironmentMetrics.ResetMetrics	Reset the cumulative values stored as metric values.
Get	/Systems/{SystemId}/Processors/{ProcessorId} /ProcessorMetrics	Retrieve the specified processor metrics.
Get	/Systems/{SystemId}/Storage	Retrieve a list of storage systems in the node.
Get	/Systems/{SystemId}/Storage/{StorageId}	Retrieve the specified storage.
Get	/Systems/{SystemId}/Storage/{StorageId}/Controllers	Retrieve a list of storage controllers in the node.
Get	/Systems/{SystemId}/Storage/{StorageId} /Controllers/{ControllerId}	Retrieve the specified storage controller.
Get	/Systems/{SystemId}/Storage/{StorageId}/Drives	Retrieve a list of physical drives in the node.

continues on next page

Table 62 – continued from previous page

HTTP Method	URI	Description
Get	/Systems/{SystemId}/Storage/{StorageId} /Drives/{DriveId}	Retrieve the specified drive.
Post	/Chassis/{ChassisId}/Drives/{DeviceId} /Actions/Drive.Reset	Perform a power operation on the physical disk.
Post	/Chassis/{ChassisId}/Drives/{DeviceId} /Actions/Drive.SecureErase	Erase the contents of the physical disk.
Get	/Systems/{SystemId}/Storage/{StorageId}/Volumes	Retrieve a list of logical drives in the node.
Get	/Systems/{SystemId}/Storage/{StorageId}/Volumes/{VolumeId}	Retrieve the specified volume.
Get	/Systems/{SystemId}/Storage/{StorageId}/Volumes/{VolumeId} /Metrics	Retrieve the specified volume metrics.
Get	/Systems/{SystemId}/VirtualMedia	Retrieve a list of virtual media in the node.
Get	/Systems/{SystemId}/VirtualMedia/{VirtualMediaId}	Retrieve the specified virtual media.
Post	/Systems/{SystemId}/VirtualMedia/{VirtualMediaId} /Actions/VirtualMedia.EjectMedia	Eject the virtual media.
Post	/Systems/{SystemId}/VirtualMedia/{VirtualMediaId} /Actions/VirtualMedia.InsertMedia	Insert the virtual media.

6.2 Power Management

Each component is assumed to be able to have its power controlled individually to reduce power consumption. Moreover, a mechanism to notify the ability or inability of power control is assumed to assist the control software in making decisions about power management.

When the control software configures the nodes and results in unused components, these components should be powered off, if possible, and remain off until needed.

6.3 Monitoring

Regarding the performance and status information acquisition by the BMC, it is expected that performance values assumed to be obtained from OOB (Out-Of-Band) for each component can be retrieved regardless of the component's usage status. Control software should optimize power consumption based on information including unused or stopped state data. Therefore, there might be cases where information in the unused state is necessary, such as when the power state alteration does not prove advantageous compared to keeping the current power state.

It is expected that performance and information retrieval can be performed at the component level. If information like power consumption cannot be obtained individually per component, power optimization would need to be carried out based on estimated values, which may lead to reduced efficiency.

CHAPTER 7

Firmware

This section outlines the requirements for firmware in the reference implementation.

While there are no specific requirements for management and control by the control software, there are necessary requirements for performing dynamic reconfiguration such as hot-plugging devices connected via the PCIe/CXL fabric.

7.1 UEFI

7.1.1 CXL Support

The necessary functionalities to support CXL are listed in Table 63.

Table 63: UEFI Requirements for CXL

Chapter in Specification	Item	Description
[uefi_spec] 11.12.6	Coherent Device Attribute Table Type	A structure exposed by CXL components that has performance characteristics. Based on CDAT, ACPI's SRAT and HMAT tables are constructed.

7.2 ACPI

7.2.1 Hot-Plug Support

The necessary functionalities for hot-plugging PCIe/CXL devices using ACPI are listed in Table 64.

Table 64: ACPI Requirements for Hot-Plug

Chapter in Specification	Item	Description
[acpi_spec] 6.1.1	_ADR (Address)	The address of the bus where the device is located. Used by Linux to determine if a device can be ejected.
[acpi_spec] 6.1.11	_SUN (Slot User Number)	A unique ID number of the slot. Used by Linux when enumerating devices controlled by ACPI.
[acpi_spec] 6.3.2	_EJD (Ejection Dependent Device)	A device that is dependent on another device. Used by Linux to also remove this device when the dependent docking station is removed.
[acpi_spec] 6.3.3	_EJx (Eject)	Eject the device. Used by Linux to eject a device using _EJ0.
[acpi_spec] 6.3.6	_RMV (Remove)	Determines if a device that does not implement _EJx can be ejected. Used by Linux to determine if a device can be ejected. Not needed if _EJ0 is implemented.
[acpi_spec] 6.3.7	_STA (Status)	Retrieve the status of the device. Used by Linux to check the presence of a device.

For NUMA configurations, the necessary functionalities are listed in Table 65.

Table 65: ACPI Requirements for NUMA

Chapter in Specification	Item	Description
[acpi_spec] 6.2.14 ([acpi_spec] 17.3.1/[acpi_spec] 17.4.1)	_PXM (Proximity)	Proximity Domain of the device.
[acpi_spec] 6.1.15 ([acpi_spec] 17.3.1)	_SLI (System Locality Information)	Distance between Proximity Domains.
[acpi_spec] 6.2.18 ([acpi_spec] 17.4.1)	_HMA (Heterogeneous Memory Attributes)	ACPI's HMAT table.

The necessary functionalities for hot-plugging a PCI Host Bridge are listed in Table 66.

Table 66: ACPI Requirements for PCI Host Bridge Hot-Plug

Chapter in Specification	Item	Description
-	_CBA (Memory mapped Configuration Base Address)	From Arm's Boot Requirement 2.0, Appendix F.

7.2.2 CXL Support

The necessary ACPI specifications to support CXL are listed below.

- System Resource Affinity Table

  A table that allows the operating system to recognize the NUMA configuration of CPU/memory during boot.

- Heterogeneous Memory Attribute Table

  A table indicating the latency from the CPU to memory.

- CXL Early Discovery Table

  A table used to discover CXL Host Bridges.

CHAPTER 8

Operating System

Describes the control and monitoring functions required by the operating system to achieve dynamic configuration control.

8.1 Driver Configuration

Regarding the operating system drivers, no modifications are made to achieve the system outlined in this guide. Instead, the system is realized using the standard drivers provided by the operating system and hardware component vendors.

8.2 Control Interface (I/F)

Describes how the operating system controls each hardware component.

8.2.1 CPU

Describes how to control the CPU device on the operating system.

Power Saving

The CPU power-saving function is controlled via the *cpufreq* subsystem in *sysfs*.

The control software agent uses commands such as *cpupower frequency-set* to control it.

8.2.2 Main Memory/Dedicated Memory

Describes how to control the main memory/dedicated memory device on the operating system.

Secure Erase

The secure erase function of non-volatile memory is controlled via the ACPI driver in *sysfs*.

The control software agent uses commands such as *ndctl* to control it.

Volatile memory is excluded.

8.2.3 CXL Memory

Describes how to control CXL memory devices on the operating system.

Secure Erase

Non-volatile memory employs the *Secure Erase* command of CXL.io.

Volatile memory is excluded.

8.2.4 Storage

Describes how to control storage devices on the operating system.

Logical Connection and Disconnection

The logical connection (online) and disconnection (offline) functions of storage are controlled via the disk driver in *sysfs*.

If logical volumes are configured on the operating system, such as mount points or Logical Volume Manager (LVM), appropriate pre-processing is conducted before disconnection.

Secure Erase

Uses functions to sanitize storage content according to ATA, SCSI, and NVMe specifications.

The control software agent uses commands such as *hdparm* and *nvme-cli* to control it.

8.2.5 Ethernet Adapter

Describes how to control Ethernet adapter devices on the operating system.

Logical Connection and Disconnection

The operating system does not explicitly control the hot-plug function of PCIe or ACPI.

If there is profile information configured on the operating system, such as *NetworkManager*, appropriate pre-processing is conducted before disconnection.

8.2.6 Optical Smart NIC

The Optical Smart NIC can function either as an Ethernet adapter or as an extender for PCIe/CXL fabric. Since it cannot be recognized as a device by the operating system when functioning as an extender, from the operating system's perspective, it is treated the same as an Ethernet adapter.

8.3 Monitoring Interface (I/F)

Describes how the operating system monitors each hardware component.

Static (spec) information prioritizes information obtained from OOB or FM of each hardware component.

Dynamic (metrics) information of each hardware component prioritizes information obtained from OOB or FM of each hardware component. Information at the workload level prioritizes the operating system's information.

For Linux, it is assumed that performance information of each hardware component described in this chapter will be retrieved.

8.3.1 CPU

Describes the information obtained by the operating system for the CPU device.

Static (Spec) Information

Describes the static (spec) information obtained from the operating system for the CPU device.

Table 67: Monitoring Interface for CPU Devices (Static Information)

Item	Required	Source(TBD)	Remarks
Vendor Name/Manufacturer	yes	Manufacturer in SMBIOS Type 4	
Model Name/Part Number		Part Number in SMBIOS Type 4	
Operating Frequency		Max Speed in SMBIOS Type 4	
Number of Processors		physical.id in /proc/cpuinfo	
Number of Cores		cpu.cores in /proc/cpuinfo	
Number of Threads		Number of entries in /proc/cpuinfo	
Serial Number	yes	Serial Number in SMBIOS Type 4	
Architecture		/proc/sys/kernel/osrelease	
Instruction Set			
Family		Family in SMBIOS Type 4	
Cache Memory		L1/L2/L3 Cache Handle in SMBIOS Type 4	
TDP			
Control Support			

Dynamic (Metrics) Information

Describes the dynamic (metrics) information obtained from the operating system for the CPU device.

Table 68: Monitoring Interface for CPU Devices (Dynamic Information)

Item	Required	Source(TBD)	Remarks
Operating Frequency	yes	cpu MHz in /proc/cpuinfo	
Number of Cores	yes	Core Enabled in SMBIOS Type 4	
Utilization	yes	/proc/stat	
Cache Misses			
Cache Hit Rate			
CPU Core Utilization		/proc/stat	
CPU Core (Instructions per Cycle)			
CPU Core (Cache Usage Rate)			
Memory Bandwidth			
Power Consumption			

8.3.2 Main Memory/Dedicated Memory

Describes the information obtained by the operating system for the main memory/dedicated memory device.

Static (Spec) Information

Describes the static (spec) information obtained from the operating system for the main memory/dedicated memory device.

Table 69: Monitoring Interface for Main Memory/Dedicated Memory Devices (Static Information)

Item	Required	Source(TBD)	Remarks
Vendor Name/Manufacturer	yes	Manufacturer in SMBIOS Type 17	
Model Name/Part Number		Part Number in SMBIOS Type 17	
Total Memory Capacity		Size in SMBIOS Type 17	
Volatile Memory Capacity		Volatile Size in SMBIOS Type 17	
Non-Volatile Memory Capacity		Non-volatile Size in SMBIOS Type 17	
Serial Number	yes	Serial Number in SMBIOS Type 17	
Media Type			
Category		Memory Technology in SMBIOS Type 17	
Device Type		Memory Type in SMBIOS Type 17	
Bus Width			
Data Width		Data Width in SMBIOS Type 17	
Memory Clock		Speed in SMBIOS Type 17	
TDP			
Control Support			

Dynamic (Metrics) Information

Describes the dynamic (metrics) information obtained from the operating system for the main memory/dedicated memory device.

Table 70: Monitoring Interface for Main Memory/Dedicated Memory Devices (Dynamic Information)

Item	Required	Source(TBD)	Remarks
Enabled/Disabled	yes		
Usage	yes	MemTotal-MemFree in /proc/meminfo	Used by the *free* command
Bandwidth Usage			
Memory Clock		Configured Memory Speed in SMBIOS Type 17	
Power Consumption			Optional if main memory is integrated into another component and included in that component.

8.3.3 CXL Memory

Describes the information obtained by the operating system for CXL memory devices.

Static (Spec) Information

Describes the static (spec) information obtained from the operating system for CXL memory devices.

Table 71: Monitoring Interface for CXL Memory Devices (Static Information)

Item	Required	Source(TBD)	Remarks
Vendor Name/Manufacturer	yes		
Model Name/Part Number			
Total Memory Capacity			Required if not retrievable via OOB/FM.
Volatile Memory Capacity			
Non-Volatile Memory Capacity			
Serial Number	yes		
Media Type			
Category			
Device Type			
Bus Width			
Data Width			
Memory Clock			
TDP			
Control Support			

Dynamic (Metrics) Information

Describes the dynamic (metrics) information obtained from the operating system for CXL memory devices.

Table 72: Monitoring Interface for CXL Memory Devices (Dynamic Information)

Item	Required	Source(TBD)	Remarks
Enabled/Disabled	yes		
Usage	yes		
Bandwidth Usage			
Memory Clock			
Power Consumption			

8.3.4 Accelerators

Describes the information obtained by the operating system for accelerator devices.

Static (Spec) Information

Describes the static (spec) information obtained from the operating system for accelerator devices.

Table 73: Monitoring Interface for Accelerator Devices (Static Information)

Item	Required	Source(TBD)	Remarks
Vendor Name/Manufacturer	yes		
Model Name/Part Number			
Operating Frequency			
Number of Processors			
Number of Cores			
Number of Threads			
Serial Number	yes		
Architecture			
Instruction Set			
Family			
Cache Memory			
TDP			
Control Support			

Dynamic (Metrics) Information

Describes the dynamic (metrics) information obtained from the operating system for accelerator devices.

Table 74: Monitoring Interface for Accelerator Devices (Dynamic Information)

Item	Required	Source(TBD)	Remarks
Operating Frequency	yes		
Number of Cores	yes		
Utilization	yes		
Cache Misses			
Cache Hit Rate			
CPU Core Utilization			
CPU Core (Instructions per Clock)			
CPU Core (Cache Usage Rate)			
Memory Bandwidth			
Power Consumption			

8.3.5 Storage

Describes the information obtained by the operating system for storage devices.

Static (Spec) Information

Describes the static (spec) information obtained from the operating system for storage devices.

Table 75: Monitoring Interface for Storage Devices (Static Information)

Item	Required	Source(TBD)	Remarks
Vendor Name/Manufacturer	yes	/sys/class/block/<dev>/device/vendor	
Model Name/Part Number		/sys/class/block/<dev>/device/model	
Capacity		/sys/class/block/<dev>/size	
Block Size		bsize in statfs()	
Transfer Speed			
Serial Number	yes	hdparm -I /dev/<dev>, smartctl -a /dev/<dev>	
Redundancy			
I/O Functionality			
TDP			
Control Support			
I/O Size		/sys/block/<dev>/queue/minimum_io_size, /sys/block/<dev>/queue/optimal_io_size	
I/O Ratio			

Dynamic (Metrics) Information

Describes the dynamic (metrics) information obtained from the operating system for storage devices.

Table 76: Monitoring Interface for Storage Devices (Dynamic Information)

Item	Required	Source(TBD)	Remarks
Usage	yes		
Allocated Capacity			
Consumed Capacity			
Guaranteed Capacity			
Maximum Allocatable Capacity			
Transfer Speed			
Redundancy			
Power Consumption			

8.3.6 Ethernet Adapter

Describes the information obtained by the operating system for Ethernet adapter devices.

Static (Spec) Information

Describes the static (spec) information obtained from the operating system for Ethernet adapter devices.

Table 77: Monitoring Interface for Ethernet Adapter Devices (Static Information)

Item	Required	Source(TBD)	Remarks
Vendor Name/Manufacturer	yes	/sys/class/net/<eth>/device/vendor	
Model Name/Part Number		/sys/class/net/<eth>/device/device	
Transfer Speed		/sys/class/net/<eth>/speed	
Serial Number	yes		
TDP			
Ethernet Info (IP Address)			
MAC Address		/sys/class/net/<eth>/address	
MTU		/sys/class/net/<eth>/mtu	
Ethernet Adapter Features (NPIV, NPAR)			
Bandwidth Limitation			

Dynamic (Metrics) Information

Describes the dynamic (metrics) information obtained from the operating system for Ethernet adapter devices.

Table 78: Monitoring Interface for Ethernet Adapter Devices (Dynamic Information)

Item	Re-quired	Source(TBD)	Remarks
Transfer Speed		/sys/class/net/<eth>/speed	
Enabled/Disabled			
MTU		/sys/class/net/<eth>/mtu	
Device CPU Utilization			
Host Bus RX Utilization			
Host Bus TX Utilization			
RX Bytes		Receive bytes in /proc/net/dev	
TX Bytes		Transmit bytes in /proc/net/dev	
RX Average Queue Depth			
TX Average Queue Depth			
Power Consumption			

8.3.7 Optical Smart NIC

The Optical Smart NIC can function either as an Ethernet adapter or as an extender for PCIe/CXL fabric. Since it cannot be recognized as a device by the operating system when functioning as an extender, from the operating system's perspective, it is treated the same as an Ethernet adapter.

Bibliography

[redfish_spec] Redfish Data Model Specification (https://www.dmtf.org/sites/default/files/standards/documents/DSP0268_2024.2.pdf)

[pcie_spec] PCI Express® Base Specification Revision 6.0 12 December 2021

[cxl_spec] Compute Express Link Specification (https://computeexpresslink.org/cxl-specification/)

[uefi_spec] UEFI Specification (https://uefi.org/specs/UEFI/2.10/)

[acpi_spec] ACPI Specification (https://uefi.org/specs/ACPI/6.5/)

Developer's Interface Guide for Green Datacenter Architecture-based Computing Environment

Publication Date 10/11/2024
Publisher System Architecture Study Committee