

灵衢[®]系统管控运维软件架构 与接口参考设计

文档版本：2.0

发布日期：2025年9月

版权所有 © 2025 华为技术有限公司。保留一切权利。

您对“本文档”的使用受知识共享（Creative Commons）署名 4.0 国际公共许可协议（以下简称“CC BY 4.0 协议”）的约束。CC BY 4.0 协议的完整内容可以访问如下网址获取：<https://creativecommons.org/licenses/by/4.0/legalcode.txt>。

使用、复制、修改、分发、或展示本文档的任何部分，即表示您同意受 CC BY 4.0 协议的约束。

灵衢[®]和 UnifiedBus[™]均为华为商标。本文档中提及或展示的其它商标、产品名称、服务名称以及公司名称，由各自的所有人拥有。

目录

1 简介	6
1.1 目的	6
1.2 范围	6
1.3 规范性引用文件	6
2 概述	7
3 应用场景	9
3.1 智算场景	9
3.2 通算场景	16
4 灵衡总线管理软件架构与接口	19
4.1 简介	19
4.2 软件架构	19
4.3 使用场景	20
4.4 接口说明	21
5 BMC 软件架构与接口	22
5.1 简介	22
5.2 软件架构	22
5.3 使用场景	23
5.4 接口说明	23
6 PoD 管理软件架构与接口	25
6.1 简介	25
6.2 软件架构	25
6.3 使用场景	26
6.4 接口说明	26
附录 A 缩略语	27
参考文献	28

图目录

图 2-1 基于灵衢的超节点整体架构	7
图 2-2 基于灵衢的超节点管控运维参考架构.....	8
图 3-1 PoD 场景管控运维参考软件架构	9
图 3-2 PoD 场景管控运维参考部署架构	10
图 3-3 Server 场景管控运维参考软件架构	13
图 3-4 Server 场景管控运维参考部署架构	14
图 3-5 通算场景管控运维参考软件架构	16
图 3-6 通算场景管控运维参考部署架构	17
图 4-1 基于灵衢的超节点总线管理参考软件架构	19
图 5-1 BMC 参考软件架构	22
图 6-1 PoDManager 参考软件架构	25

表目录

表 3-1 PoD 场景接口协议和格式要求	11
表 3-2 Server 场景接口协议和格式要求	15
表 3-3 通算场景接口协议和格式要求	18
表 5-1 支持 SNMP 协议版本说明	24
表 5-2 支持的 IPMI 规范版本与 ipmitool 工具版本	24

1 简介

1.1 目的

本文档目标受众包括运维系统开发者、管控系统开发者、运维工程师和系统管理员等，便于其使用、开发、管理和维护基于灵衢的超节点。华为技术有限公司已基于灵衢设计和实现相关产品，本文档内容的构建基于华为智算和通算基于灵衢的超节点，如需基于灵衢设计开发相关产品，可参考本文档内容。其他有兴趣了解基于灵衢的超节点管控运维软件架构的人员也可阅读此文档。

1.2 范围

本文档介绍了基于灵衢的超节点智算和通算场景管控运维所使用的软件架构和接口功能，包括：

1. 软件架构：描述管控运维软件栈架构。
2. 功能组件：描述管控运维软件栈各层功能组件的定位、功能、使用场景。
3. 接口定义：描述不同管控运维组件对各类用户提供功能的软件接口。

1.3 规范性引用文件

1. DSP0266 Redfish Specification Version 1.20.1
2. DSP8010 Redfish Schema Bundle Version 2024.1
3. IETF RFC 3411 An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks
4. IETF RFC 6241 Network Configuration Protocol (NETCONF)
5. Intelligent Platform Management Interface Specification Second Generation v1.5
6. Intelligent Platform Management Interface Specification Second Generation v2.0
7. OpenTelemetry Specification 1.47.0
8. 《灵衢基础规范 2.0》

2 概述

当使用灵衡构建面向智算和通算场景的超节点时，需要构建全栈的计算能力，以满足业务和管控运维需求，本章节概述了基于灵衡的超节点构建的全栈计算能力，重点介绍了管控运维相关能力。

基于灵衢的超节点整体架构如下图所示：

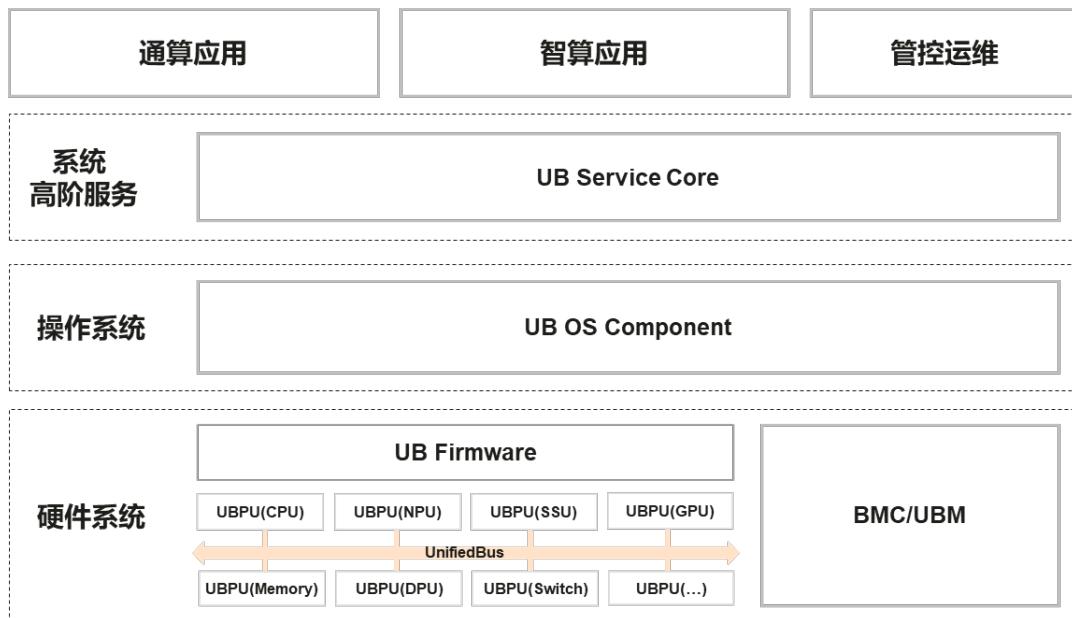


图 2-1 基于灵衢的超节点整体架构

基于灵衢的超节点需要支持运维系统、云管系统及应用对接，并规划对应的能力。本文档聚焦灵衢运维管理、控制、总线管理等管控运维部分。

基于灵衢的超节点管控运维参考架构如下图：

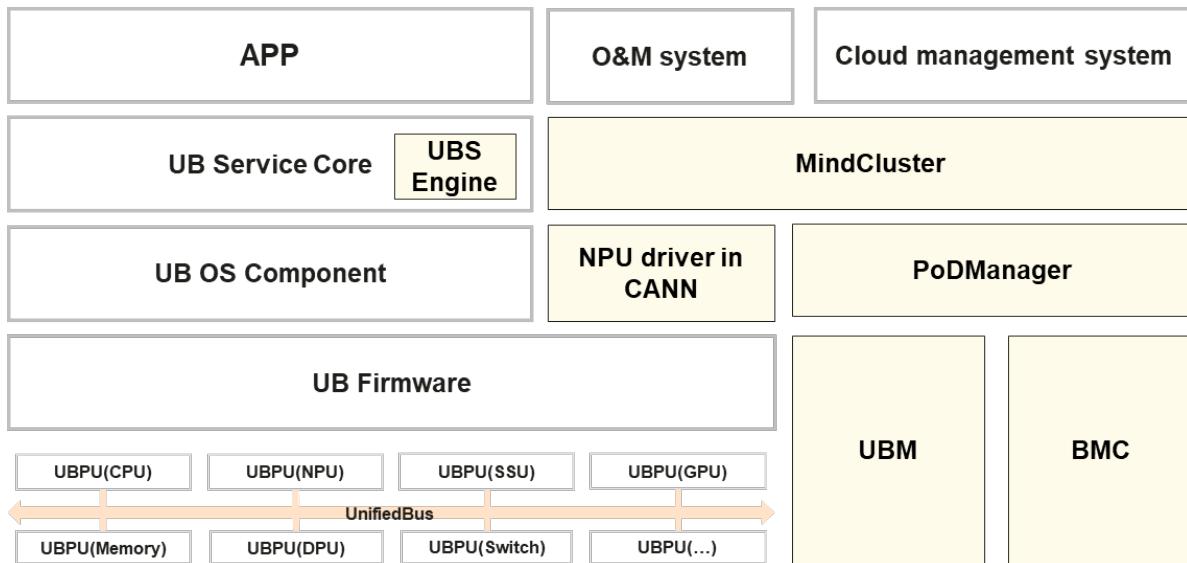


图 2-2 基于灵衢的超节点管控运维参考架构

面向智算场景，可基于灵衢构建 PoD 粒度或者 Server 粒度的超节点；面向通算场景，可基于灵衢构建 Server 粒度的超节点。

智算和通算应用场景，基于灵衢的超节点面向运维系统和云管系统应分别提供对应的管控运维组件。当前已规划和提供的管控运维组件如下：

- UB Service Core (UBS Engine): 基于灵衢的超节点的控制引擎，负责池化资源管理与灵活调度；提供简化易用、生态兼容的北向资源管理与控制 API，实现灵衢可定义计算的广泛使能。UBS Engine 详细介绍请参见《灵衢系统高阶服务软件架构参考设计》。
- MindCluster (智算集群系统软件): 围绕 AI 任务资源感知、任务调度、稳定运行等特性构建的一套集群级解决方案。MindCluster 为上层应用提供基于灵衢的超节点下 AI 任务的管理运维能力，详细介绍请参见昇腾社区相关资料。
- PoDManager (PoD 管理软件): PoD 形态基于灵衢的超节点的框级统一运维管理软件，为客户提供框级配置、部署、升级和监控的能力与接口。
- UnifiedBus Management (灵衢总线管理软件): 灵衢总线的管控面软件，简称 UBM，为总线互联系统提供 Entity 资源、总线网络层管控服务及管理运维功能。
- BMC: 基于灵衢的超节点硬件管理的核心部件，BMC 通过 Redfish、IPMI (智能平台管理接口) 等协议实现硬件级远程监控与管理。其核心职能为灵衢硬件的健康监测、远程控制和故障预警等。
- NPU driver in CANN (NPU 驱动): 为模组、标卡、服务器、集群等多形态的硬件产品提供归一化的管理能力，支撑基于灵衢的超节点的 UBPU (NPU) 管理，充分释放 UBPU (NPU) 硬件能力。

3 应用场景

3.1 智算场景

3.1.1 PoD 形态产品

智算场景下用户可构建以 PoD 为基础的基于灵衢的超节点，其管控运维参考软件架构如下图所示：

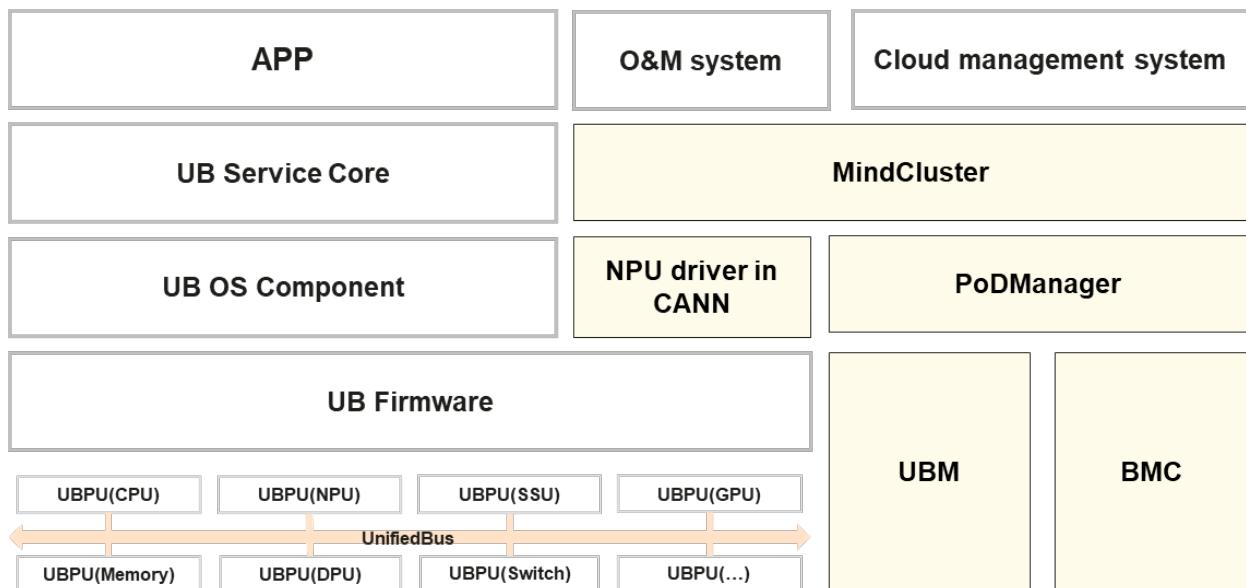


图 3-1 PoD 场景管控运维参考软件架构

3 应用场景

管控运维参考部署架构如下图所示：

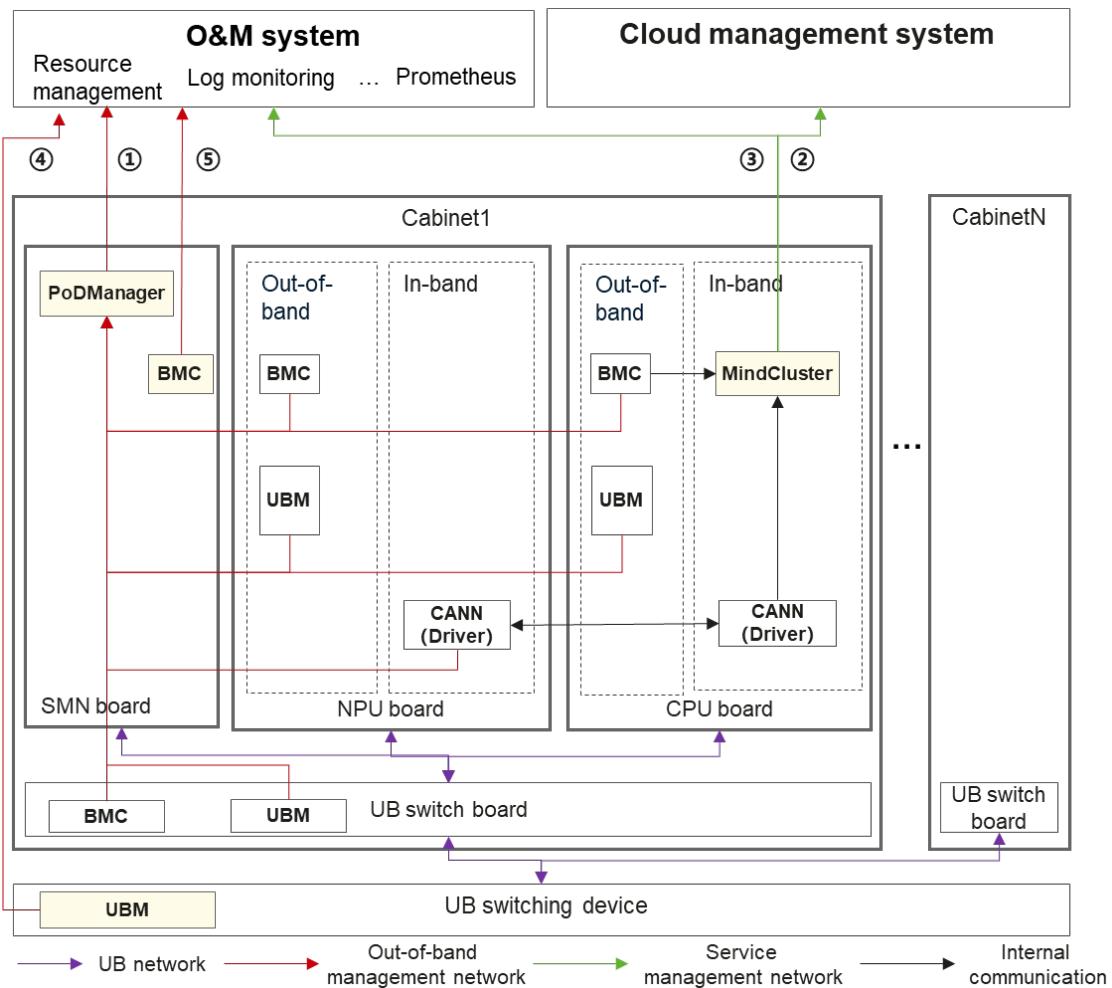


图 3-2 PoD 场景管控运维参考部署架构

该场景下，基于灵衢的超节点机柜内通过灵衢交换板将柜内计算板等和 UB 网络互联，柜外通过灵衢交换设备将所有机柜进行 UB 网络互联。

该场景超节点需针对不同硬件设备规划相关管控运维组件，已规划组件有 PoDManager、BMC、UBM、MindCluster、CANN (Driver)。其中 PoDManager 对柜内进行统一管理，其南向通过对接 BMC 进行硬件监控、告警上报、故障管理等，通过对接 UBM 和 CANN (Driver) 进行总线管理和 NPU 管理。

注：该场景下柜内 BMC（SMN 板 BMC 除外）和柜内 UBM 不对外提供接口，由柜内的 PoDManager 统一对外提供接口。

管控运维系统对接开发说明：

1. 运维系统对接场景：

(1) 场景 1：柜内硬件带外管理：

a 柜内硬件（SMN 板除外）带外管理：

应对接 PoDManager 的北向接口。具体方法见表 3-1 路径①相关内容；

3 应用场景

b 柜内 SMN 板带外管理:

应对接 SMN 板上的 BMC 北向接口，具体方法见表 3-1 路径⑤相关内容；

(2) 场景 2：柜外交换设备管理:

应对接 UBM 北向接口，具体方法见表 3-1 路径④相关内容；

(3) 场景 3：AI 处理器带内监控运维:

推荐对接 MindCluster 北向接口。具体方法见表 3-1 路径③相关内容。

2. 云管系统对接场景:

场景 1：提升基于灵衢的超节点智算任务调度性能:

推荐对接 MindCluster 北向接口进行集群智算任务管理，具体方法见表 3-1 路径②相关内容。

各对接路径详细接口协议和格式要求如下表所示：

表 3-1 PoD 场景接口协议和格式要求

模块	接口序号	接口路径	协议	格式	数据模型
PoDManager	①	PoDManager -> O&M system	HTTP/1.1、 HTTP/2	JSON	遵从 redfish 标准的物理资源 模型、任务资源模型、事件 资源模型、证书服务资源模 型等。 注：见第 6 章节相关内容。
			RESTCONF	XML 或 JSON	基于 Yang 模型的总线物理资 源模型。 注：见第 6 章节相关内容。
			NETCONF	XML	基于 Yang 模型的总线物理资 源模型。 注：见第 6 章节相关内容。
			SNMP V1、 SNMP V2C、 SNMP V3	SNMPTrap 报文格式	灵衢总线设备告警、系统状 态、系统信息、事件、资 产、用户、功率、上下电管 理信息模型。 注：见第 6 章节相关内容。
			Telemetry	Protobuf 或 JSON	总线物理资源监控信息模 型。 注：见第 6 章节相关内容。
MindCluster	②	MindCluster -> Cloud management system	gRPC	Protobuf	资源节点、AI 处理器的设备 发现、状态和健康状态上报 功能，向 Kubernetes 提供任 务调度 AI 处理器资源操作的 数据模型。

3 应用场景

模块	接口序号	接口路径	协议	格式	数据模型
					注：参见昇腾社区相关资料。
			ConfigMap	JSON	集群中 AI 任务状态，任务使用 AI 处理器资源的信息模型。 注：参见昇腾社区相关资料。
	③	MindCluster -> O&M system	HTTP/1.1 、 HTTP/2	JSON	对接 Prometheus 和 Telegraf (InfluxData 的数据采集器) 标准接口上报 AI 处理器的利用率、温度、电压等的信息模型。 注：参见昇腾社区相关资料。
UBM	④	UBM -> O&M system	RESTCONF	XML 或 JSON	基于 Yang 模型的总线物理资源模型、总线路由管理、总线端口管理等数据模型。 注：见第 4 章节相关内容。
			NETCONF	XML	基于 Yang 模型的总线物理资源模型、总线路由管理、总线端口管理等数据模型。 注：见第 4 章节相关内容。
			SNMP V1、 SNMP V2C、 SNMP V3	SNMPTrap 报文格式	灵衢总线设备告警信息模型。 注：见第 4 章节相关内容。
			Telemetry	Protobuf 或 JSON	基于 Yang 模型的 Protobuf 或者 JSON 格式的端口速率统计等高性能采集数据模型。 注：见第 4 章节相关内容。
BMC	⑤	BMC -> O&M system	HTTP/1.1、 HTTP/2	JSON	符合 redfish 标准的 BMC 物理资源模型、任务资源模型、事件资源模型、证书服务资源模型等。 注：见第 5 章节相关内容。
			SNMP V1、 SNMP V2C、 SNMP V3	SNMP 协议 报文格式	系统状态、系统信息、事件、资产、用户、功率、上下电管理、告警信息模型。 注：见第 5 章节相关内容。

模块	接口序号	接口路径	协议	格式	数据模型
			IPMI1.5、IPMI 2.0	IPMI 协议报文格式	系统信息、性能监控、存储管理、功率、环境监控、维护诊断、告警&事件、用户、网络、安全模型。 注：见第 5 章节相关内容。

3.1.2 Server 形态产品

智算场景下用户可构建以 Server 为基础的基于灵衢的超节点，其管控运维参考软件架构如下图所示：

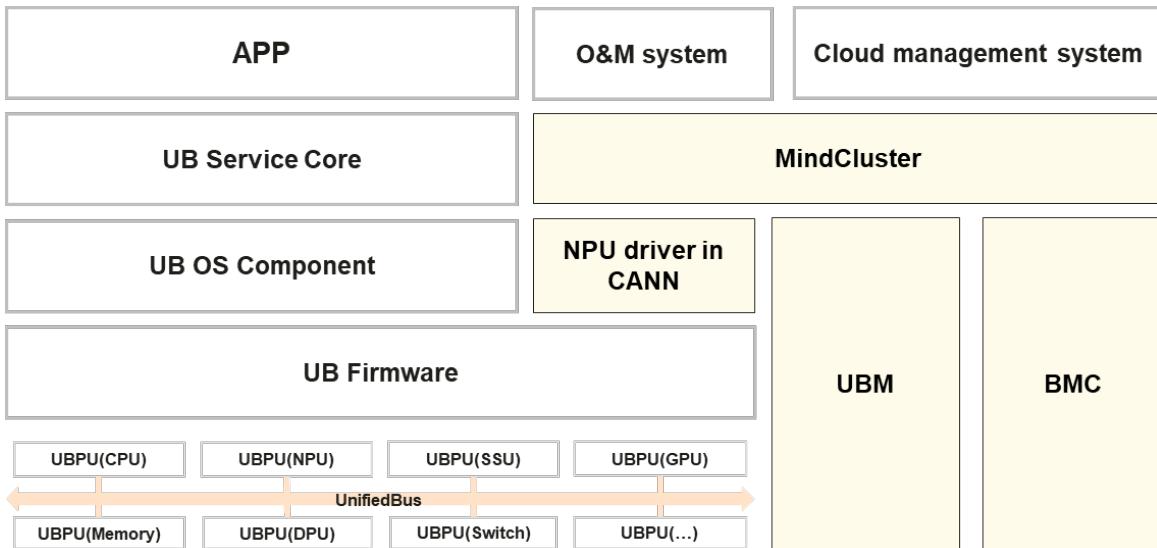


图 3-3 Server 场景管控运维参考软件架构

管控运维参考部署架构如下图所示：

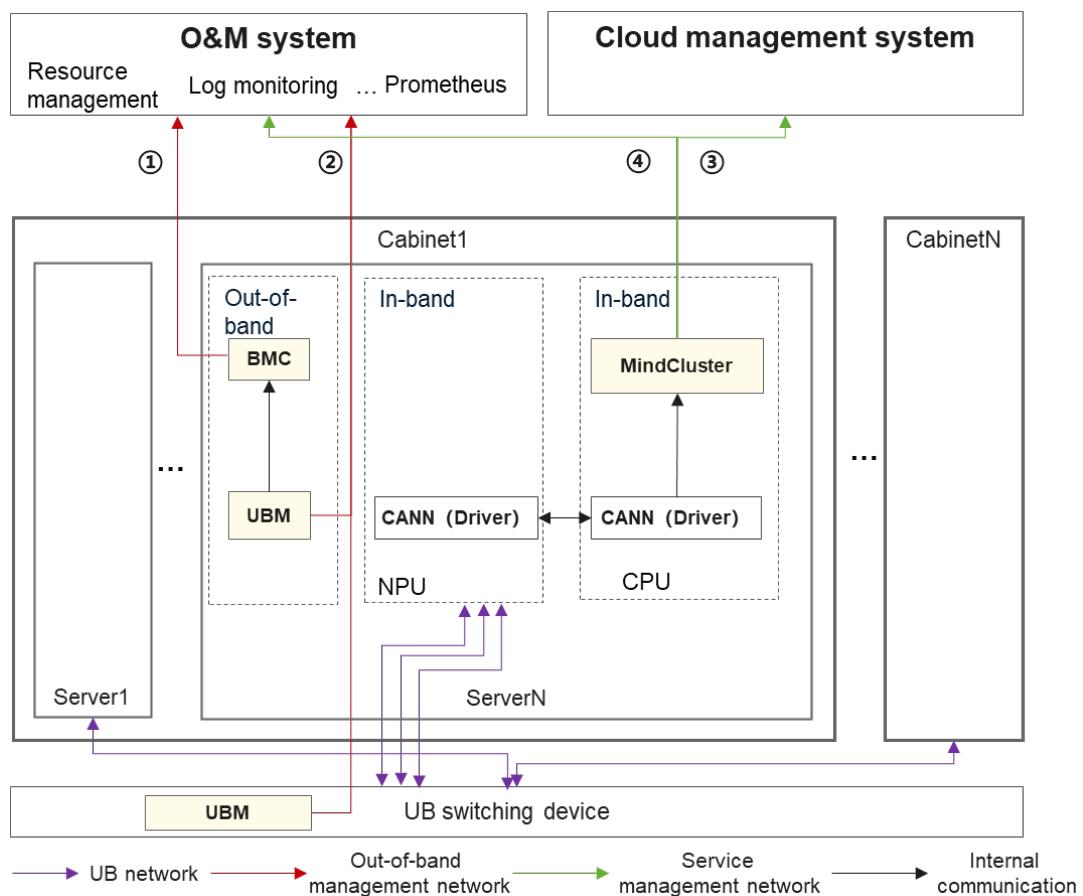


图 3-4 Server 场景管控运维参考部署架构

该场景超节点机柜内无交换板，Server 内的 NPU 直连 UB 网络。该场景超节点需针对不同硬件设备规划相关管控运维组件，已规划组件有 BMC、UBM、MindCluster。

管控运维系统对接开发说明：

1. 运维系统对接场景：

(1) 场景 1：Server 硬件带外管理：

应对接 BMC 的北向接口。具体方法见表 3-2 路径①相关内容；

(2) 场景 2：Server 灵衢总线管理&交换设备管理：

应对接 UBM 北向接口，具体方法见表 3-2 路径②相关内容；

(3) 场景 3：AI 处理器带内监控运维：

推荐对接 MindCluster 北向接口。具体方法见表 3-2 路径④相关内容。

2. 云管系统对接场景：

场景 1：提升基于灵衢的超节点智算任务调度性能：

推荐对接 MindCluster 北向接口进行集群智算任务管理，具体方法见表 3-2 路径③相关内容。

3 应用场景

各对接路径详细接口协议和格式要求如下表所示：

表 3-2 Server 场景接口协议和格式要求

模块	接口序号	接口路径	协议	格式	数据模型
BMC	①	BMC -> O&M system	HTTP/1.1、 HTTP/2	JSON	符合 redfish 标准的 BMC 物理资源模型、任务资源模型、事件资源模型、证书服务资源模型等。 注：见第 5 章节相关内容。
			SNMP V1、 SNMP V2C、 SNMP V3	SNMP 协议报文格式	系统状态、系统信息、事件、资产、用户、功率、上下电管理、告警信息模型 注：见第 5 章节相关内容。
			IPMI1.5、IPMI 2.0	IPMI 协议报文格式	系统信息、性能监控、存储管理、功率、环境监控、维护诊断、告警&事件、用户、网络、安全模型。 注：见第 5 章节相关内容。
UBM	②	UBM -> O&M system	RESTCONF	XML 或 JSON	基于 Yang 模型的总线物理资源模型、总线路由管理、总线端口管理等数据模型 注：见第 4 章节相关内容。
			NETCONF	XML	基于 Yang 模型的总线物理资源模型、总线路由管理、总线端口管理等数据模型 注：见第 4 章节相关内容。
			SNMP V1、 SNMP V2C、 SNMP V3	SNMPTrap 报文格式	灵衡总线设备告警信息模型 注：见第 4 章节相关内容。
			Telemetry	Protobuf 或 JSON	基于 Yang 模型的 Protobuf 或者 JSON 格式的端口速率统计等高性能采集数据模型。 注：见第 4 章节相关内容。
MindCluster	③	MindCluster -> Cloud management	gRPC	Protobuf	资源节点、AI 处理器的设备发现、状态和健康状态上报功能，向 Kubernetes 提供任

模块	接口序号	接口路径	协议	格式	数据模型
		system			务调度 AI 处理器资源操作的数据模型 注：参见昇腾社区相关资料。
			ConfigMap	JSON	集群中 AI 任务状态，任务使用 AI 处理器资源的信息模型 注：参见昇腾社区相关资料。
④		MindCluster -> O&M system	HTTP/1.1 、 HTTP/2	JSON	对接 Prometheus 和 Telegraf 标准接口上报 AI 处理器的利用率、温度、电压等的信息模型。 注：参见昇腾社区相关资料。

3.2 通算场景

通算场景下用户可构建以 Server 为基础的基于灵衢的超节点,其管控运维参考软件架构如下图所示:

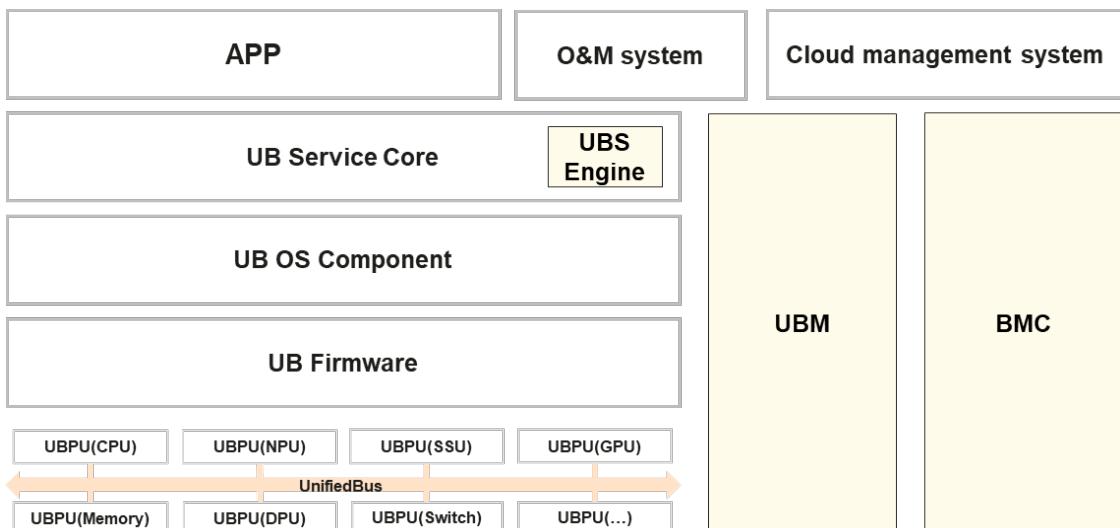


图 3-5 通算场景管控运维参考软件架构

管控运维参考部署架构如下图所示：

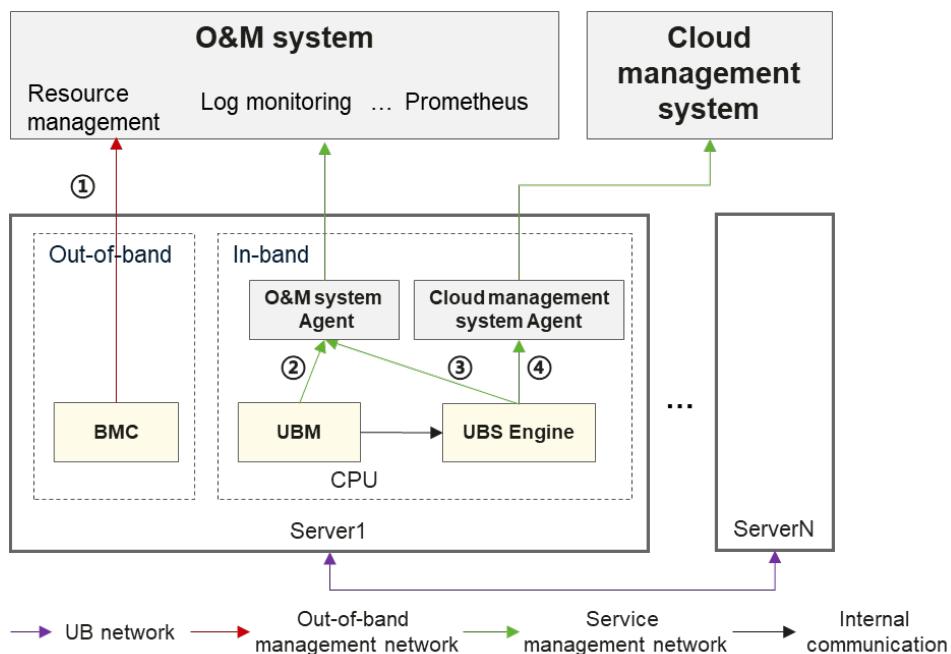


图 3-6 通算场景管控运维参考部署架构

该场景超节点需针对不同硬件设备规划相关管控运维组件，已规划组件有 BMC、UBM、UBS Engine。UBM、UBS Engine 部署在带内，通过 agent 与运维系统和云管系统对接。

管控运维系统对接开发说明：

1. 运维系统对接场景：

(1) 场景 1：Server 的硬件带外管理：

应对接 BMC 的北向接口。具体方法见表 3-3 路径①相关内容；

(2) 场景 2：Server 的灵衢总线管理：

应通过带内的运维系统 Agent 对接 UBM 北向接口，具体方法见表 3-3 路径②相关内容；

(3) 场景 3：基于灵衢的超节点池化节点和内存资源的监控度量：

推荐通过带内的运维系统 Agent 对接 UBS Engine 北向接口。具体方法见表 3-3 路径③相关内容。

2. 云管系统对接场景：

场景 1：基于灵衢的超节点池化资源管理与灵活调度编排，实现高性能、高可用的可定义计算：

推荐通过带内的云管系统 Agent 对接 UBS Engine 北向接口，具体方法见表 3-3 路径④相关内容。

3 应用场景

各对接路径详细接口协议和格式要求如下表所示：

表 3-3 通算场景接口协议和格式要求

模块	接口序号	接口路径	协议	格式	数据模型
BMC	①	BMC -> O&M system	HTTP/1.1、 HTTP/2	JSON	遵从 redfish 标准的 BMC 物理资源模型、任务资源模型、事件资源模型、证书服务资源模型等。 注：见第 5 章节相关内容。
			SNMP V1、 SNMP V2C、 SNMP V3	SNMP 协议 报文格式	系统状态、系统信息、事件、资产、用户、功率、上下电管理、告警信息模型 注：见第 5 章节相关内容。
			IPMI1.5、 IPMI 2.0	IPMI 协议报 文格式	系统信息、性能监控、存储管 理、功率、环境监控、维护诊 断、告警&事件、用户、网络、 安全模型。 注：见第 5 章节相关内容。
UBM	②	UBM -> O&M system	RESTCONF	XML 或 JSON	基于 Yang 模型的总线物理资源 模型、总线路由管理、总线端口 管理等数据模型 注：见第 4 章节相关内容。
			SNMP V1、 SNMP V2C、 SNMP V3	SNMPTrap 报文格式	灵衢总线设备告警信息模型 注：见第 4 章节相关内容。
			Telemetry	Protobuf 或 JSON	基于 Yang 模型的 Protobuf 或者 JSON 格式的端口速率统计等高 性能采集数据模型。 注：见第 4 章节相关内容。
UBS Engine	③	UBS Engine -> O&M system	CLI	OS 命令行 格式	基于灵衢的超节点高阶集群引擎 服务操作节点和内存资源的监控 度量等数据模型 注：详情参见《灵衢系统高阶服务软 件架构参考设计》相关内容。
	④	UBS Engine -> Cloud management system	SDK	本地函数调 用格式	基于灵衢的超节点高阶集群引擎 服务操作节点和内存资源的调度 管理等数据模型 注：详情参见《灵衢系统高阶服务软 件架构参考设计》相关内容。

4 灵衢总线管理软件架构与接口

4.1 简介

灵衢总线管理软件 (UnifiedBus Management) 是灵衢总线的管控面软件，简称 UBM，为总线互联系统提供 Entity 资源、总线网络层管控服务和管理运维功能。智算产品的 UBM 运行在服务器主板、PoD 计算板、总线交换板、总线交换设备的专用处理器中；通算产品的 UBM 运行在 HostOS 中。运维系统通过 Restconf、Netconf 等北向接口协议实现对总线设备监控与管理。

本章节定义了 UBM 的架构和接口功能，用于支撑灵衢总线的运维管理，方便用户的运维系统或云管系统对接。

4.2 软件架构

本章节定义了 UBM 参考架构，UBM 面向智算和通算场景提供了相同的软件架构。

UBM 为基于灵衢的超节点总线管理的核心，其参考软件架构如下图所示：

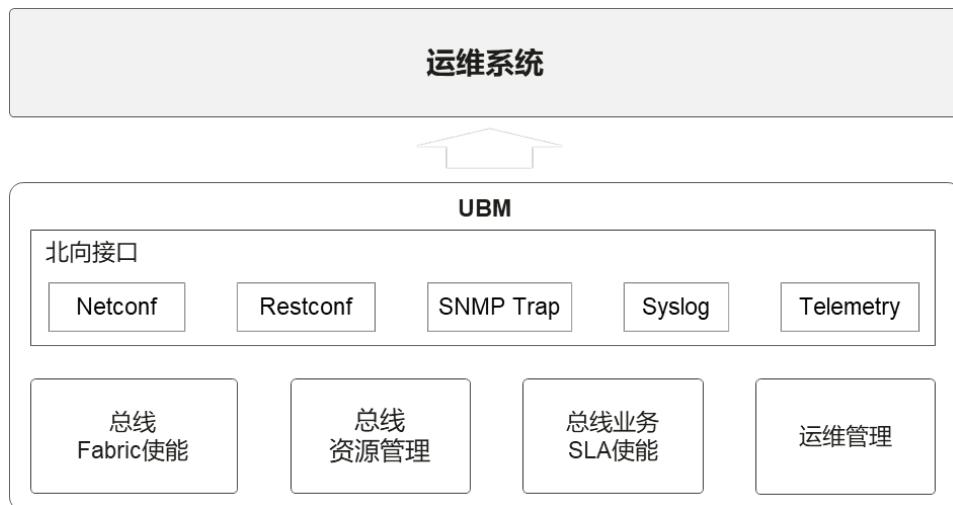


图 4-1 基于灵衢的超节点总线管理参考软件架构

UBM 负责总线计算板、总线交换板与总线交换设备的互联资源连通性管理、计算资源和通信管理，具体功能如下：

1. 总线 Fabric 使能：UBM 通过对总线 Fabric 拓扑、互联地址及 CNA 路由的管理，自动使能总线互联资源连通能力。
 - (1) 拓扑管理：支持 UB MESH、CLOS 总线拓扑，提供 UB 拓扑信息，动态感知拓扑设备变化/链路变化，提供拓扑查询和运维接口。

(2) 互联地址管理：对 UB 域内的 UBPUs 的 CNA&EID 地址进行分配及回收，提供 CNA&EID 地址的查询和运维接口。

(3) CNA 路由管理：负责 UB 域内 CNA 地址路由生成、路由策略控制及 CNA 路由生成，提供 CNA 路由的查询和运维接口。

2. 总线资源管理：

实现对 UBPUs、Entity 资源及 UB Memory 总线消息转发能力的规划与配置，提供 URMA 建链服务和虚拟机 UB 总线资源服务，为 UB 业务提供计算资源和通信能力。

3. 总线业务 SLA 管理

负责总线 Entity 设备与总线网络层的拥塞避免、VL 管理及信用证管理等，为不同 SL 类型业务消息提供区分服务和业务质量保证机制。

4. 运维管理

具备对总线计算板、总线交换板及总线交换设备的故障定界与性能监测能力；支持通过 Telemetry 主动推送监控指标；提供对 UBM 运行中系统/设备故障进行实时感知、记录系统日志和管理告警通告的能力。

5. 提供丰富的北向接口

支持通过 Netconf/Restconf/SNMP Trap/Telemetry 北向接口对接运维系统。

4.3 使用场景

UBM 为智算/通算的基于灵衢的超节点计算硬件提供总线管控功能，其典型使用场景如下：

1. 灵衢总线互联初始化：负责灵衢总线的启动初始化，通过识别 UB 域内的 UBPUs、Switches，在端侧完成 Entity 资源规划和配置。支持 UB MESH、CLOS 总线组网，端到端完成 CNA 地址的分配和设置。根据分配的 CNA 地址生成网络层转发表，打通 UB 域内总线互联路由，提供高效转发能力。
2. 灵衢总线运行态管理：提供总线设备动态配比、URMA 建链、虚拟机 UB 总线创建、互联故障管理、租户分区配置、动态感知总线网络故障实时调整转发路径、总线业务质量、端口状态管理、光模块管理以及 Telemetry 性能采集等功能。
3. 灵衢总线交换设备健康监测：实时监测总线交换板、总线交换设备运行状态，包括控制 CPU 的占用率、内存负载、温度，以及风扇转速、电压等传感器数据。
4. 灵衢总线管理软件安装部署：支持总线交换板、计算板与总线交换设备的免配置安装部署，支持总线管理软件升级、固件更新、系统重启等操作；
5. 灵衢总线故障管理：通过日志记录与告警机制识别硬件异常。
6. 安全能力：支持的管控面接入认证、数字签名和可信启动等安全能力。

4.4 接口说明

本章节提供 UBM 的接口协议说明和接口定义，用于指导开发者，北向接口通过 Netconf/Restconf/SNMP Trap/Telemetry 提供多种北向管理方案。

4.4.1 RESTCONF 接口说明

RESTCONF 是一种基于 HTTP 的协议的编程接口，支持对网络设备的数据进行增、删、改、查操作。RESTCONF 编程接口符合 IT 业界流行的 RESTful 风格，为用户提供高效开发 Web 化运维工具的能力。

UBM 支持采用 RESTCONF 协议和客户运维系统对接。

注：支持的 RESTCONF 的接口详细说明会随对应产品同步提供。

4.4.2 NETCONF 接口说明

NETCONF 是基于 XML 的网络配置和管理协议，并使用基于 RPC 机制实现客户端与服务器之间的通信。NETCONF 提供了一种通过运行网络管理软件的中心计算机（即网络管理工作站，客户端）来远程管理和监控设备（服务器端）的方法。

通过 NETCONF 协议，可对远端设备的配置进行查询、安装、修改和删除等操作。该协议允许设备公开一个完整、正式的 API。网络管理的应用程序可以使用相关 API 发送和接收完全或者部分配置数据集。

UBM 支持采用 NETCONF 协议和客户运维系统对接。

注：支持的 NETCONF 的接口详细说明会随对应产品同步提供。

4.4.3 SNMP trap 接口说明

简单网络管理协议（以下简称 SNMP）是用于网络管理的协议，规定了在网络环境中对设备进行监视和管理的标准化管理框架、通信的公共语言、相应的安全和访问控制机制。

UBM 提供了 SNMP 的编程接口，支持 SNMP Trap 操作。通过第三方管理软件调用 SNMP 接口可以方便地对服务器集成管理。

UBM SNMP trap 支持 SNMP 协议 V1/V2C/V3 版本。

注：支持的 SNMP trap 的接口详细说明会随对应产品同步提供。

4.4.4 Telemetry 接口说明

Telemetry 是一种远程的从物理设备或虚拟设备上高速采集数据的技术。设备通过推模式（Push Mode）周期性的主动向采集器推送设备的接口流量统计、CPU 或内存数据等信息，相对传统拉模式（Pull Mode）的一问一答式交互，提供了更实时更高速的数据采集功能。

UBM 支持采用 Telemetry 方式和客户运维系统对接，支持配置设备侧的 Telemetry 静态订阅和动态订阅等。

注：支持的 Telemetry 的接口详细说明会随对应产品同步提供。

5 BMC 软件架构与接口

5.1 简介

BMC 是计算硬件带外管理的核心部件，灵衢 BMC 是基于灵衢的超节点配套的 BMC，支持基于灵衢的超节点的硬件带外管理。其主要功能包括：硬件信息管理、配置管理、本地和远程诊断、部署升级、运行能耗智能管理、故障智能预测、安全入侵智能感知等。

5.2 软件架构

灵衢 BMC 软件架构支持 BMC 间的低时延、高带宽通信，可组成级联汇聚、平等协同、孤立平铺等多种管理组网形式，为基于灵衢的超节点提供完整的管理视图。参考软件架构如下：



图 5-1 BMC 参考软件架构

BMC 软件架构分层解耦，支持功能组件灵活扩展，各层主要功能如下：

1. 北向接口：提供业界主流协议的接口，如 Redfish、SNMP、IPMI 等，并新增基于灵衢的超节点设备的硬件管理相关接口。
2. 资源模型：通过管理资源模型化表达，实现 BMC 业务与北向接口解耦，新增定义基于灵衢的超节点的物理资源模型，提供基于灵衢的超节点架构的视图化管理能力与模型同步能力。
3. 运行组件：提供包括远程管理、故障预测、健康监测、配置管理等灵衢硬件带外管理的核心功能。
4. 基础运行框架：提供解耦、灵活组合的组件化运行框架。
5. 设备模型：通过基于灵衢的超节点硬件设备的模型抽象，屏蔽应用层和硬件的差异，支持基于灵衢的超节点硬件的自适应适配和协同管理。

6. 工程工具：对资源模型与设备模型进行可视化管理，支持代码自动生成，提供基于灵衢的超节点管理仿真环境。

5.3 使用场景

BMC 典型使用场景如下：

1. 远程运维管理：支持灵衢节点的远程开关机、重启、操作系统安装、硬件诊断等操作。
2. 硬件状态监控：采集灵衢节点的 CPU 温度、风扇转速等数据，动态调整散热策略。
3. 固件与配置维护：
 - (1) 批量固件升级：支持灵衢架构下多节点多固件并行升级和升级任务编排；
 - (2) 配置备份与恢复：支持 BMC 设置快速克隆、备份和一键还原。
4. 安全与审计：支持灵衢节点的接入认证与可信校验。

5.4 接口说明

基于灵衢的超节点支持资源池化，BMC 管理接口由传统单机管理接口转向多节点可组合的管理视图模式。BMC 管理接口模型应支持节点易伸缩扩容，同时提供设备物理层完整的机框、系统与链路视图，支持运维平台和基于灵衢的超节点管理解耦；并针对基于灵衢的超节点，新增灵衢线缆、灵衢池化节点等的资产管理、故障管理、升级等相关管理接口。

本章节提供基于灵衢的超节点硬件管理 BMC 的接口协议说明和接口定义。基于灵衢的超节点硬件管理 BMC 提供了 Redfish、SNMP、IPMI（智能平台管理接口）等丰富的北向管理能力。

5.4.1 Redfish 接口说明

本章节扩展了 Redfish 可扩展平台管理的接口，并基于基于灵衢的超节点的硬件管理需求对 Redfish 接口的要求实施了细化，为基于灵衢的超节点的硬件管理定义了全面的 Restful API，可解决基于灵衢的超节点硬件管理常见的运维和业务问题。基于灵衢的超节点配套 BMC 支持 Redfish 1.20.1 规范。

注：Redfish 接口的详细说明会随对应产品同步提供。

5.4.2 SNMP 接口说明

灵衢 BMC 提供了 SNMP 的编程接口，支持 SNMP Get/Set/Trap 操作。通过第三方管理软件调用 SNMP 接口可以方便地对服务器集成管理。SNMP 代理支持 V1/V2C/V3 版本。

灵衢 BMC 支持的 SNMP 协议版本说明如下：

表 5-1 支持 SNMP 协议版本说明

SNMP 版本号	说明
SNMP V1	Get/Set 操作可以使用不同的团体名；
SNMP V2C	Get/Set 操作可以使用不同的团体名；
SNMP V3	支持安全用户名与登录用户名相同，安全用户与其他接口（Web、CLI、IPMI LAN）共用一套本地用户。鉴权算法支持选择 MD5 或 SHA、SHA256、SHA384、SHA512，加密算法支持选择 DES 或 AES、AES256

本章节介绍了灵衢 BMC 所支持的 SNMP 接口定义及相关属性；如需获取服务器支持的 SNMP 接口的详细信息，请参考服务器灵衢配套 BMC SNMP 接口说明。

注：该接口说明会随对应产品同步提供。

5.4.3 IPMI 接口说明

灵衢 BMC 兼容 IPMI 1.5/IPMI 2.0 规范，使用第三方工具（如：ipmitool），通过基于 BT 接口的 BT 协议或 LAN 通道的 UDP/IP 协议实现对服务器的有效管理。灵衢 BMC 支持 AES-CBC-128 加密算法，以及 HMAC-SHA1/HMAC-SHA256 鉴权和完整性校验算法。

灵衢 BMC 支持的 IPMI 规范版本与 ipmitool 工具版本列表如下：

表 5-2 支持的 IPMI 规范版本与 ipmitool 工具版本

规范版本	IPMI 1.5 IPMI 2.0
Ipmitool 工具版本	ipmitool 1.8.10 ipmitool 1.8.11 ipmitool 1.8.13 ipmitool 1.8.14 ipmitool 1.8.18 ipmitool 1.8.19

如需获取 IPMI 命令详细信息，请参考灵衢配套 BMC IPMI 接口说明。

注：该接口说明会随对应产品同步提供。

6 PoD 管理软件架构与接口

6.1 简介

PoDManager 是 PoD 形态基于灵衢的超节点框级统一运维的管理软件，为客户提供框级配置、部署、升级与监控的能力和接口。

PoDManager 的南向对接各业务节点的 BMC 和 UBM，进行框内汇聚管理，屏蔽框内各业务节点的管理接口，实现 PoD 内运维管理自闭环；北向通过标准的运维管理接口与运维系统/云管系统对接，通过提供计算资源相关的 Redfish 接口和网络资源相关的 NetConf 接口，实现对 PoD 内计算单元和灵衢总线单元的管理运维。

6.2 软件架构

PoDManager 参考软件架构如下：

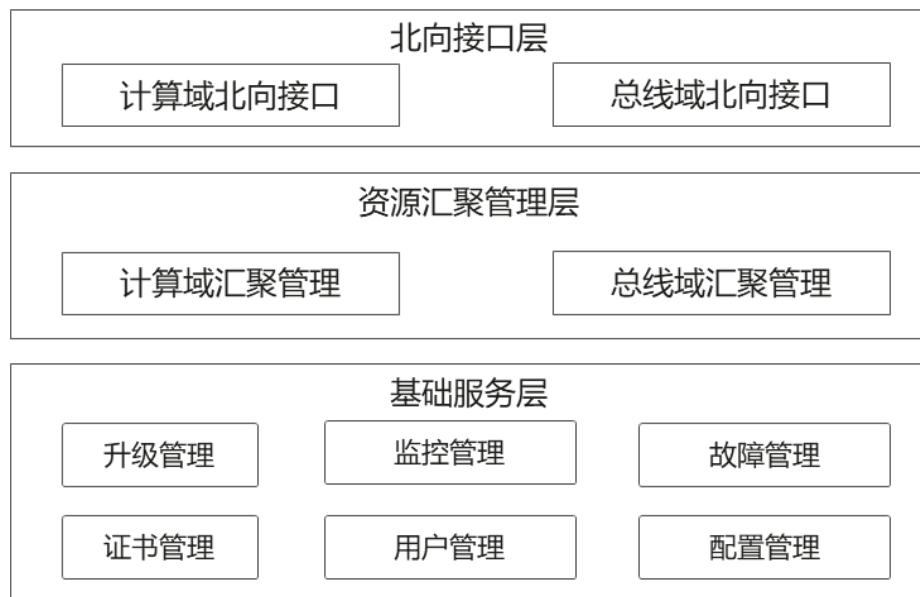


图 6-1 PoDManager 参考软件架构

1. 北向接口层：分别提供计算域和总线域的业界主流协议。计算域支持 Redfish、SNMP Trap、总线域支持 Netconf、Restconf 与 SNMP Trap。
2. 资源汇聚管理层：对接各业务节点的 BMC，进行计算域汇聚管理；对接各业务单板的灵衢总线网络管理软件，实现总线域汇聚管理。
3. 基础服务层：提供配置管理、日志管理、升级管理、监控管理、告警管理、证书管理等基础运维功能。

6.3 使用场景

PoDManager 负责框级带外管理，其典型使用场景如下：

1. 远程运维管理：通过网络接口实现 PoD 内节点或 OS 域的远程开关机、重启、系统安装等操作；
2. 硬件告警监控：支持 PoD 内各硬件单元的告警实时采集与上报；
3. 硬件性能监控：采集整柜节点的 CPU 温度、功耗、每个节点 CPU/NPU 的占用率、HBM 带宽等；
4. 固件与配置维护：
 - (1) 基础配置：包括 IP 配置、北向对接配置、基于灵衢的超节点配置与业务板 UBM 的 ZTP 配置；
 - (2) 整框固件批量升级：远程更新 BIOS/BMC/CPLD/UBM 的固件，支持多节点并行，支持 PoDManager 自升级；
 - (3) 整框节点配置备份与恢复：自动备份各节点配置数据，在更换节点硬件设备后，自动恢复新节点的配置数据；
5. 日志采集：整柜节点的 BMC/UBM/NPU 管理软件的日志采集，PoDManager 自身的日志采集；
6. 安全与审计：
 - (1) 访问控制：实现基于角色的用户权限管理；
 - (2) 日志审计：记录操作日志，支持合规性检查和故障溯源。

6.4 接口说明

北向接口通过 Redfish/Netconf/Restconf/SNMP Trap/Telemetry 提供多种北向管理方案。

注：PoDManager 的接口详细说明会随对应产品同步提供。

附录 A 缩略语

缩略语	缩略语全称
AES	高级加密标准 (Advanced Encryption Standard)
API	应用程序编程接口 (Application Programming Interface)
BIOS	基本输入输出系统 (Basic Input Output System)
BMC	基板管理控制器 (Baseboard Management Controller)
CLI	命令行接口 (Command-line Interface)
CPLD	复杂可编程逻辑器件 (Complex Programmable Logic Device)
DES	数据加密标准 (Data Encryption Standard)
HTTP	超文本传输协议 (Hypertext Transfer Protocol)
HTTPS	HTTPS 加密协定 (Hypertext Transfer Protocol Secure)
IPMI	智能平台管理接口 (Intelligent Platform Management Interface)
NETCONF	一种网络配置和管理协议 (Network Configuration Protocol)
OS	操作系统 (Operating System)
REST	表现层状态转换 (Representational State Transfer)
RESTCONF	RESTFUL 风格的网络数据操作的协议 (RESTful Configuration Protocol)
RPC	远程过程调用 (Remote Procedure Call)
SHA	安全散列算法 (Secure Hash Algorithm)
SL	服务优先级 (Service Level)
SMN	系统管理节点 (System manage node)
SNMP	简单网络管理协议 (Simple Network Management Protocol)
UBM	灵衢总线管理软件 (UnifiedBus Management)
URMA	统一远程内存访问 (Unified Remote Memory Access)
VL	虚拟通道 (Virtual Lane)
XML	可扩展标记语言 (Extensible Markup Language)

参考文献

- [1] https://www.dmtf.org/sites/default/files/standards/documents/DSP0266_1.20.1.pdf
- [2] <https://www.dmtf.org/dsp/DSP8010>
- [3] <https://datatracker.ietf.org/doc/rfc3411/>
- [4] <https://datatracker.ietf.org/doc/rfc6241/>
- [5] <https://opentelemetry.io/docs/specs/otel/>
- [6] <https://kubernetes.io/docs/home/>
- [7] <https://prometheus.io/docs/introduction/overview/>
- [8] <https://support.influxdata.com/s/>
- [9] <https://www.hiascend.com/>