

# 灵衢<sup>®</sup>系统高阶服务软件架构参考设计

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<sup>®</sup>和 UnifiedBus<sup>™</sup>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

<b>1 简介</b>	<b>6</b>
1.1 目的	6
1.2 范围	6
1.3 规范性引用文件	6
1.4 术语定义	6
<b>2 灵衢系统高阶服务软件架构参考</b>	<b>7</b>
2.1 灵衢系统高阶服务软件架构	7
2.2 灵衢系统高阶服务介绍	8
<b>3 灵衢系统高阶集群引擎服务 UBS Engine 组件</b>	<b>10</b>
3.1 简介	10
3.2 架构	11
3.3 使用场景	12
<b>4 灵衢系统高阶内存服务 UBS Mem 组件</b>	<b>13</b>
4.1 简介	13
4.2 架构	14
4.3 使用场景	14
<b>5 灵衢系统高阶通信服务 UBS Comm 组件</b>	<b>16</b>
5.1 简介	16
5.2 架构	17
5.3 使用场景	17
<b>6 灵衢系统高阶 IO 服务 UBS IO 组件</b>	<b>19</b>
6.1 简介	19
6.2 架构	20
6.3 使用场景	21
<b>7 灵衢系统高阶虚拟化服务 UBS Virt 组件</b>	<b>22</b>
7.1 简介	22
7.2 架构	22

## 目录

7.3 使用场景.....	25
<b>附录 A 缩略语.....</b>	<b>26</b>

## 图目录

图 2-1 灵衢超节点计算系统架构示意 .....	7
图 2-2 灵衢计算系统示意图 .....	7
图 2-3 灵衢系统高阶服务参考架构 .....	9
图 3-1 灵衢系统高阶集群引擎服务 UBS Engine 软件架构 .....	11
图 4-1 灵衢系统高阶内存服务 UBS Mem 系统位置示意 .....	13
图 4-2 灵衢系统高阶内存服务 UBS Mem 软件架构 .....	14
图 5-1 灵衢系统高阶通信服务 UBS Comm 系统位置示意 .....	16
图 5-2 灵衢系统高阶通信服务 UBS Comm 软件架构 .....	17
图 6-1 灵衢系统高阶 IO 服务 UBS IO 系统位置示意 .....	19
图 6-2 灵衢系统高阶 IO 服务 UBS IO 软件架构 .....	20
图 7-1 灵衢系统高阶虚拟化服务 UBS Virt 软件架构 .....	22
图 7-2 灵衢系统高阶虚拟化服务 Virt OVS 软件架构 .....	23
图 7-3 灵衢系统高阶虚拟化服务 Virt Optimizer 软件架构 .....	24
图 7-4 灵衢系统高阶虚拟化服务 Virt VM 软件架构 .....	24

# 1 简介

## 1.1 目的

本文档概述了支持面向智算和通算场景灵衢产品的系统高阶服务架构、所需的主要软件功能模块及接口功能。目标受众包括软件开发工程师、技术支持、企业技术负责人等。其他任何有兴趣了解灵衢系统高阶服务软件架构参考设计的人员也可阅读此文档，了解灵衢产品系统高阶服务软件，便于使用、开发、管理和维护灵衢产品。

## 1.2 范围

本文档介绍了面向智算和通算场景的灵衢产品系统高阶服务软件栈、各层功能组件和对接指导，包括：

1. 软件架构：说明系统高阶服务软件栈分层功能。
2. 功能组件：介绍系统高阶服务软件栈各层功能组件的定位、架构、功能、使用场景和使用指导。

## 1.3 规范性引用文件

《灵衢基础规范 2.0》

## 1.4 术语定义

中文名称	英文名称
灵衢系统高阶服务	UB Service Core ( UBS Core )
灵衢系统高阶通信服务	UB Service Core Communication ( UBS Comm )
灵衢系统高阶集群引擎服务	UB Service Core Engine ( UBS Engine )
灵衢系统高阶内存服务	UB Service Core Memory ( UBS Mem )
灵衢系统高阶 IO 服务	UB Service Core IO ( UBS IO )
灵衢系统高阶虚拟化服务	UB Service Core Virt ( UBS Virt )

# 2 灵衢系统高阶服务软件架构参考

## 2.1 灵衢系统高阶服务软件架构

超节点是灵衢计算系统的核心，通过灵衢重构计算系统，打破计算主机边界，实现智算和通算超节点扩展并获得性能规格收益，是面向未来 AI 时代的目标计算系统架构。

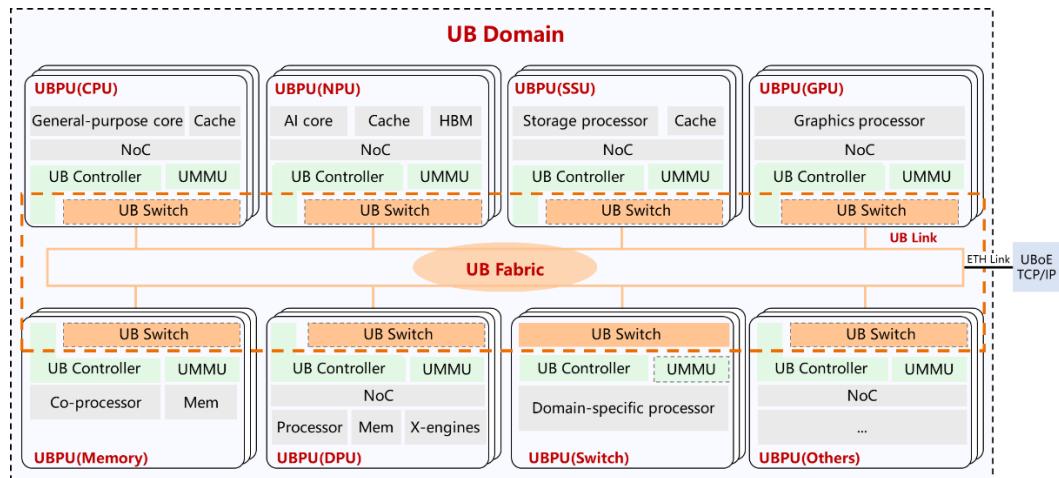


图 2-1 灵衢超节点计算系统架构示意

灵衢超节点打破传统主机边界，在超低时延、多协议归一的灵衢总线级互联的超节点域内，通过计算资源全量池化、平等互联，实现计算资源的灵活组合，超大规模组网和系统高可用性。

灵衢计算系统整体参考架构如下图所示：

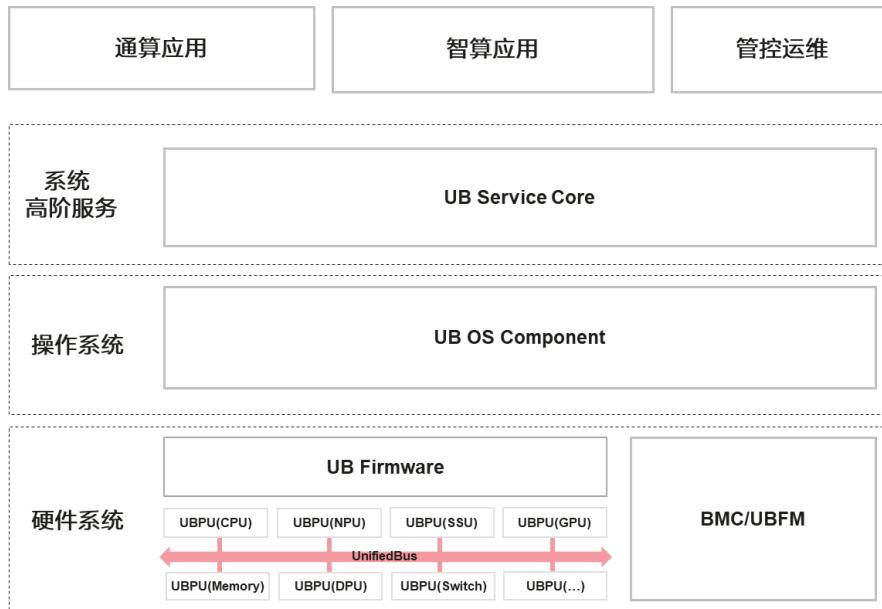


图 2-2 灵衢计算系统示意图

灵衢计算系统参考架构分成四层，共同发挥灵衢计算系统超节点架构优势：

1. **灵衢硬件系统，实现超节点可定义计算**

- UnifiedBus ( UB/UB Firmware/BMC ): 灵衢超节点计算系统基础，基于灵衢实现计算集群总线级互联、协议归一、平等协同、全量池化、大规模组网、高可用性。
- 灵衢互连结构管理器 ( UBFM ): 完成灵衢的互联配置，实现计算资源的灵活、高效聚合，优化计算互联 SLA ( 时延、带宽、可靠性 )，实现灵衢超节点灵活算力使能。

2. **操作系统灵衢组件 ( UB OS Component ) 完成灵衢和设备的使能，进行统一抽象与管理**

- UB OS Component 通过扩展 Linux 内核，原生支持超节点计算系统，实现对灵衢设备的统一抽象与管理。
- 同时保持 POSIX 接口兼容，实现现有应用的快速迁移。通过 openEuler 社区，使能操作系统生态，实现灵衢计算系统的开放创新。

3. **灵衢系统高阶服务 ( UB Service Core )，使能超节点性能最优**

充分发挥灵衢计算系统在内存池化、通信、分布式操作等方面的优势，同时简化灵衢资源管理与调度的复杂性，华为提供了系列的灵衢系统高阶服务 UB Service Core，使能计算业务应用，快速发挥灵衢超节点系统优势，获得性能收益。

4. **灵衢应用适配，实现主流开源开箱即用**

围绕智算、通算等典型场景主流开源软件，华为实现了相应的灵衢应用适配，如大数据灵衢适配、智算训推灵衢适配、云化 K8s 适配等，发挥灵衢计算系统超节点架构性能优势，实现开箱即用，获得 25% 以上的性能收益。

## 2.2 灵衢系统高阶服务介绍

UB 将多个节点紧密的连接在一起，形成了超节点。而传统的 Linux 操作系统是为单机准备的，其调度的资源是单节点内的资源（如内存，PCI-e 设备等）。面向灵衢超节点，为了应用快速使能超节点能力，灵衢系统构建了 UB Service Core，封装 UB 底层能力、集群拓扑等，简化并兼容现有生态，让应用用本地资源一样使用超节点资源，简化应用开发，使能灵衢超节点。

**UB Service Core** 构筑 5 大集群级系统服务，支持不同 OS 并全面开源至 openEuler 社区，释放超节点平等互联架构优势，全面使能应用加速 30~50%，促进灵衢系统软件生态构筑：

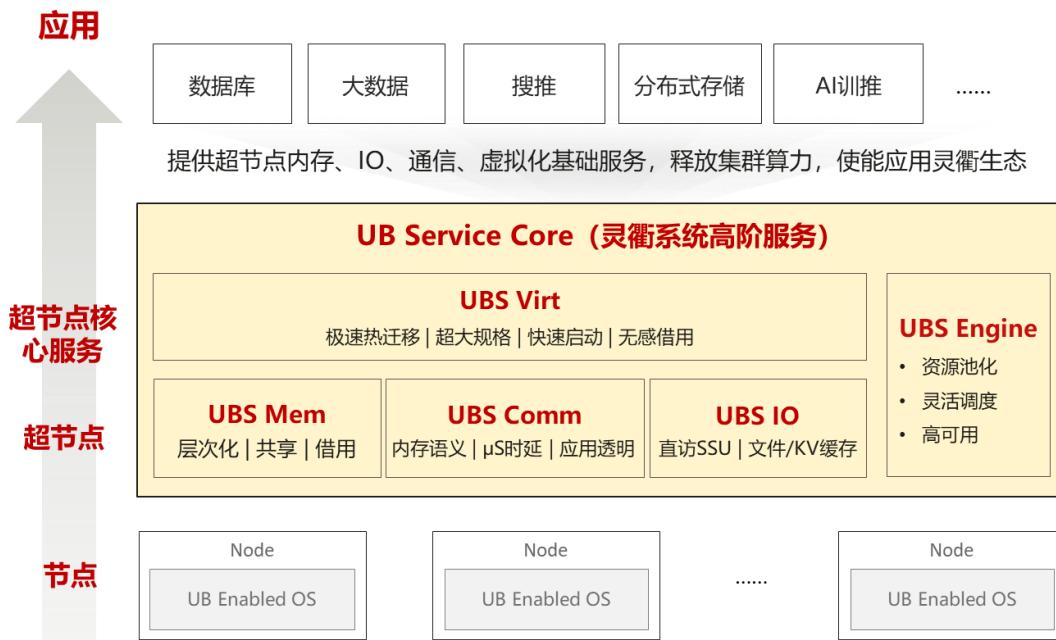


图 2-3 灵衢系统高阶服务参考架构

**UB Service Core** (简写 UBS Core)，包含如下 5 大部分：

1. **UB Service Core Engine** (简写 **UBS Engine**)：支持内存、DPU 资源池化管理与动态调度，支持分布式自选主，支持 N 节点场景下最多 N-1 节点失效的高可用，是灵衢计算系统的控制面核心参考实现。
2. **UB Service Core Memory** (简写 **UBS Mem**)：支持统一内存编程，实现灵衢超节点的共享内存、池化内存。
3. **UB Service Core Communication** (简写 **UBS Comm**)：基于超节点提供高性能、高可靠以及生态兼容（用户态 Socket/Verbs over UB）的通信协议。
4. **UB Service Core IO** (简写 **UBS IO**)：基于超节点，提供应用亲和的全局数据读写缓存系统高阶 IO 服务。
5. **UB Service Core Virt** (简写 **UBS Virt**)：支持虚拟化池化，热迁移策略决策，极速快恢与容灾，虚机/容器间极速通信等能力，使能虚拟化性能提升。

# 3 灵衢系统高阶集群引擎服务 UBS Engine 组件

## 3.1 简介

UBS Engine，作为超节点控制引擎，负责基于 UB 平等架构的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 UB 可定义计算的广泛使能。

UBS Engine 提供如下核心能力：

### 1. 简单易用，简化云管和应用的适配开发：

屏蔽复杂的 EID、CNA 等 UB ID 体系，统一对接集群 UBFM 和 OS ( UB OS Component )，提供集群单一的、简化北向 API。

### 2. 资源池化，结合负载感知的灵活调度，实现高性能：

构造内存池化、DPU 池化能力，资源灵活调度，支持云化性能提升、成本节省等。

支持业务感知的节点间内存动态调度，高 IO 场景应用性能提升，提升资源利用率。

### 3. 去中心化平等架构，实现 3 个 9→4 个 9 高可用：

池化内存跨节点故障不扩散，超节点业务可靠性持平。

UBS Engine 去中心化，支持 N-1 节点故障高可用。

### 4. 开放灵衢系统管控信息及接口对接生态应用，开箱即用：

UBS Engine 支持与 Prometheus 生态兼容的监控度量，部署、巡检、日志等运维北向开放。

## 3.2 架构

### 3.2.1 软件架构

UBS Engine 参考架构如下：

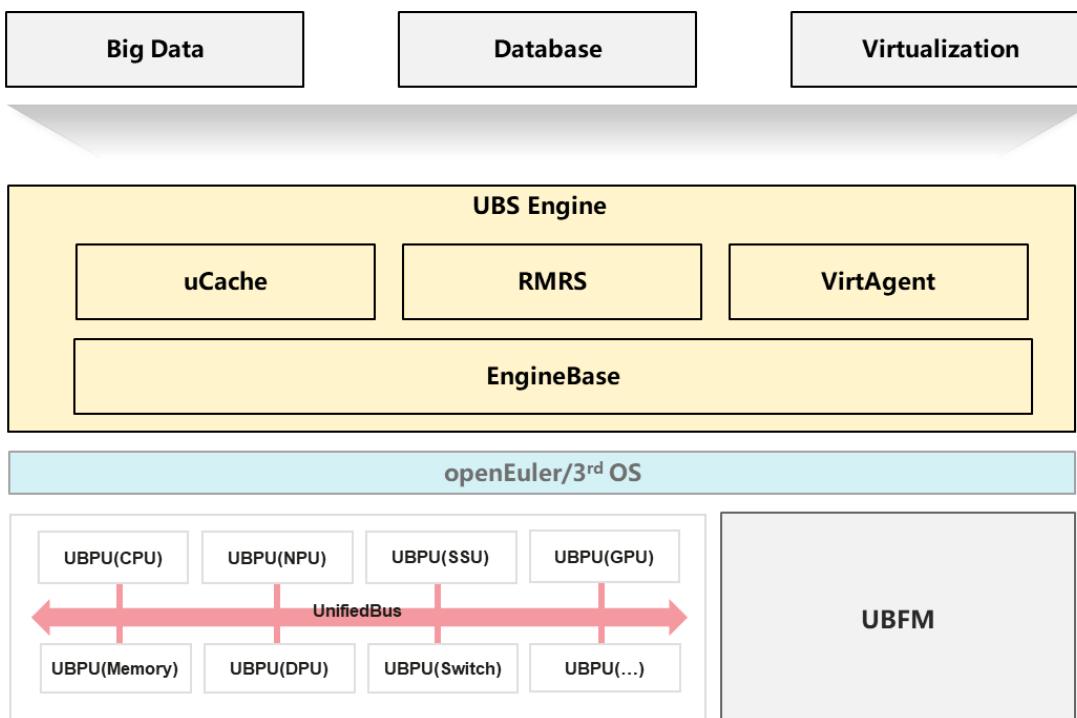


图 3-1 灵衢系统高阶集群引擎服务 UBS Engine 软件架构

### 3.2.2 软件架构说明

- EngineBase**：提供 UB 逻辑资源创建删除（如 bus instance、bonding eid 等），提供内存池化基础管理能力（调度、借用、归还、故障处理）；为北向云管系统/监控系统等提供节点级接入能力；整体去中心化，支持 N-1 节点故障高可用。
- uCache**：依托 UB 平等架构内存池化基础能力，将集群节点剩余内存组成全局 IO 缓存池，加速 IO 瓶颈应用，构建基于 UB 的全局 IO 缓存池。
- RMRS**：基于集群节点的内存碎片分布，依托内存池化基础能力，将集群内的内存碎片通过内存借用进行再利用，开启更多虚机，提升资源利用率。
- VirtAgent**：基于集群节点内存使用信息，依托内存池化基础能力，将集群内的内存通过内存借用进行削峰填谷，释放虚拟化竞争力，提升虚拟化资源利用率。

### 3.3 使用场景

**UBS Engine** 当前提供内存借用及基于业务的内存调优能力，其典型使用场景如下：

1. **共享内存场景：**数据库大数据场景使用共享内存进行主备数据同步，通过 UBS Mem 调用 UBS Engine 接口进行内存借用归还。
2. **虚拟化内存超分场景：**云管系统通过内存超分，提升内存使用率，由 UBS Engine 自动根据内存水线进行内存借用，避免虚机因节点内存不足而退出，提升系统可靠性。
3. **虚拟化内存碎片场景：**UBS Engine 识别各虚机的冷热数据信息和空闲内存信息，将节点虚机的冷数据分布到远端节点的碎片内存中，腾空节点本地内存，支持创建更多虚机，提升整体虚机密度。
4. **虚拟化超大规模虚机：**组合集群节点的内存资源，如创建 3T 规格以上的超大规模虚机。
5. **IO 加速场景：**大数据、数据库等场景中，借用富余内存到紧缺内存节点，扩展 PageCache 资源，加速 IO，提升 IO 瓶颈型应用性能。
6. **节点内存不足场景：**通过 OOM 异常处理流程，调用 UBS Engine 快速内存借用，避免因节点 OOM 导致应用异常，提升系统可靠性。

# 4 灵衢系统高阶内存服务 UBS Mem 组件

## 4.1 简介

UBS Mem 基于超节点为应用提供内存共享、内存数据缓存和内存数据存储服务，其核心职能包括：

1. **MemFabric**: 在超节点上提供分布式系统的内存统一编址与数据访问基础能力。
2. **SHMEM**: 在超节点上提供共享内存服务。
3. **MemCache**: 在超节点上提供内存缓存服务，支持数据高带宽访问、和弱一致性保证。
4. **MemStore**: 在超节点上提供内存存储服务，支持数据低时延访问、和强一致及可靠性保证。

UBS Mem 在灵衢系统架构的位置如下图所示：

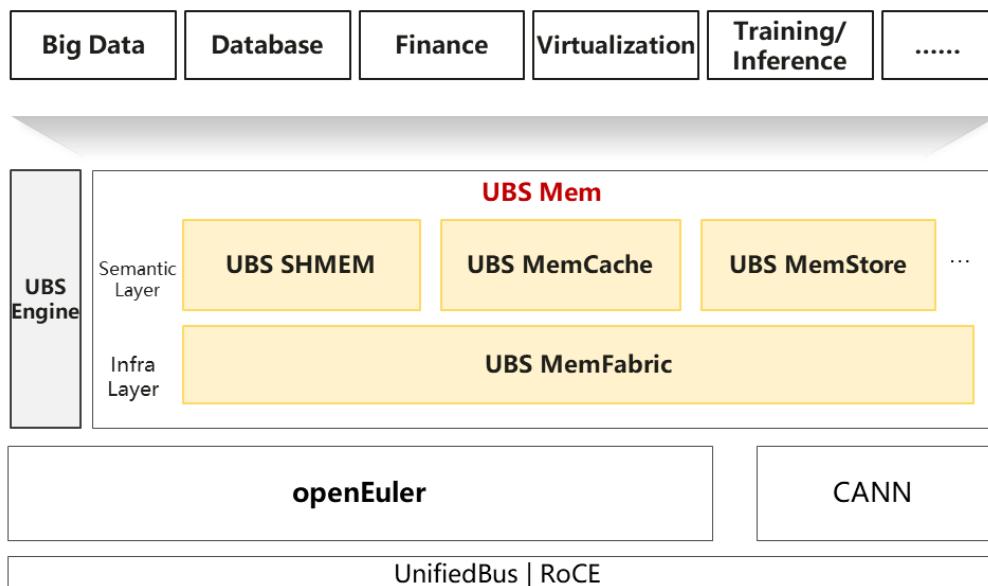


图 4-1 灵衢系统高阶内存服务 UBS Mem 系统位置示意

## 4.2 架构

### 4.2.1 软件架构

UBS Mem 软件通常采用分层解耦的模块化设计，参考架构如下：

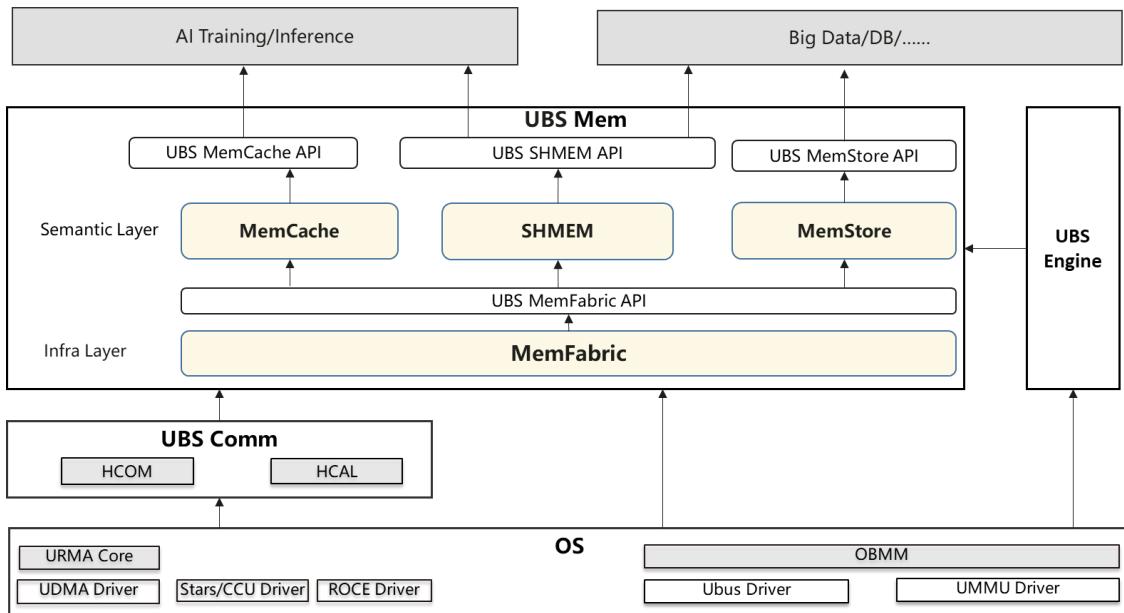


图 4-2 灵衢系统高阶内存服务 UBS Mem 软件架构

### 4.2.2 软件架构说明

UBS Mem 软件架构整体按照服务和层级解耦，支持功能组件灵活扩展，各层主要功能如下：

- 语义层：**通过解耦的服务，提供 SHMEM API、MemCache API、MemStore API 等接口和服务为 AI 推理、大数据、数据库、金融等场景提供共享内存资源、数据缓存、数据存储和内存语义通信等能力。
- 基础设施层：**屏蔽语义层和硬件的差异，提供硬件、OS 等的自适配能力，对语义层提供统一编址、数据访问基础能力。

## 4.3 使用场景

MemFabric、SHMEM、MemCache、MemStore 四个关键特性，其典型使用场景如下：

- MemFabric：**提供分布式系统的内存统一编址与数据访问，基于 UB、RoCE 网络实现 Memory 跨机统一虚拟编址，基于 xDMA 实现统一的数据访问，在此之上构建不同的语义库支撑不同场景的应用。
- SHMEM：**用于超节点内共享内存资源服务。通用计算场景，CPU 直驱的跨机 C2C 共享内存，软件实现跨机器 Cache Coherence 维护；智算场景，NPU 直驱的跨机 D2D 共享内存。

3. **MemCache**: 利用 UB 的多通路并发实现高带宽，支持数据的弱一致性内存数据缓存，支撑大模型推理的 KVCache 等场景高带宽诉求。
4. **MemStore**: 利用 UB Memory C2C 实现极低延迟的内存数据存储，满足数据的多副本强一致、极致低延迟与可靠性诉求，用于金融等场景的多副本数据内存存储等。

# 5 灵衢系统高阶通信服务 UBS Comm 组件

## 5.1 简介

UBS Comm 面向超节点，提供高性能、高可靠以及生态兼容的通信协议栈。

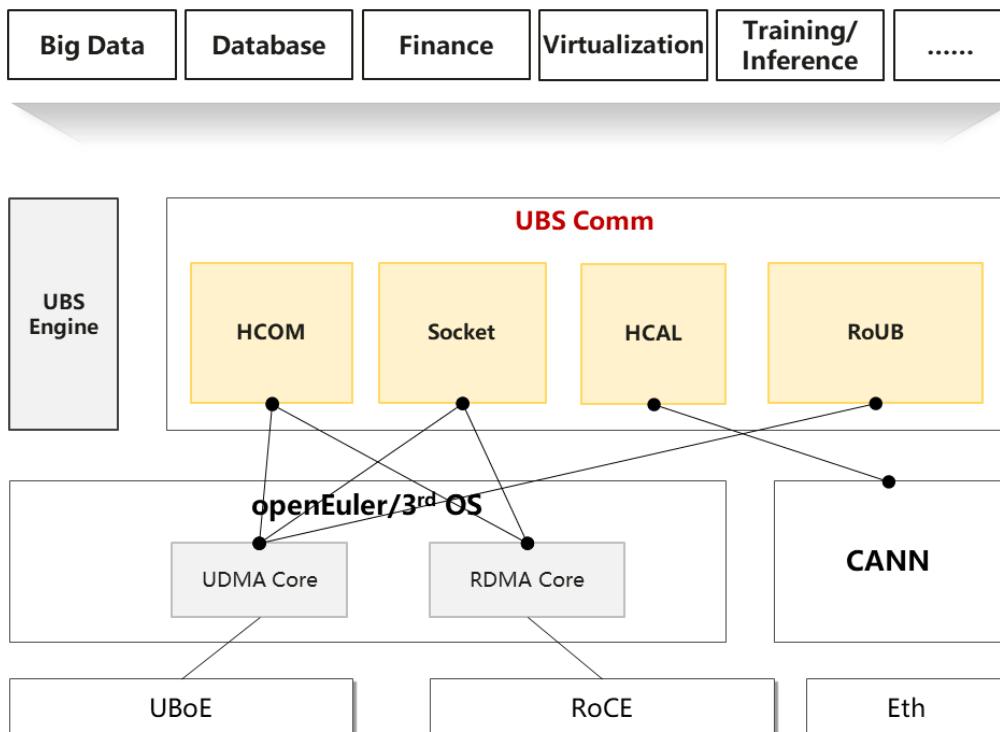


图 5-1 灵衢系统高阶通信服务 UBS Comm 系统位置示意

如上图所示，UBS Comm 包含以下 4 个关键特性：

1. **HCOM**: 北向提供统一 UB Native API，南向支持不同协议的基础通信库，简化编程难度，实现高可靠、高性能通信。
2. **Socket**: 应用零修改，数据面 Bypass TCP/IP 协议栈，Socket over UB 应用透明实现性能提升。
3. **HCAL**: 提供跨节点异构通信能力 (D2H 等)，支持 CPU-xPU 数据直接传输，提高异构访问效率。
4. **RoUB**: 支持 RDMA Verbs 语义接口，让原有的 RDMA 应用无缝迁移到 UB 网络。

## 5.2 架构

### 5.2.1 软件架构

UBS Comm 软件通常采用分层解耦的模块化设计，参考架构如下：

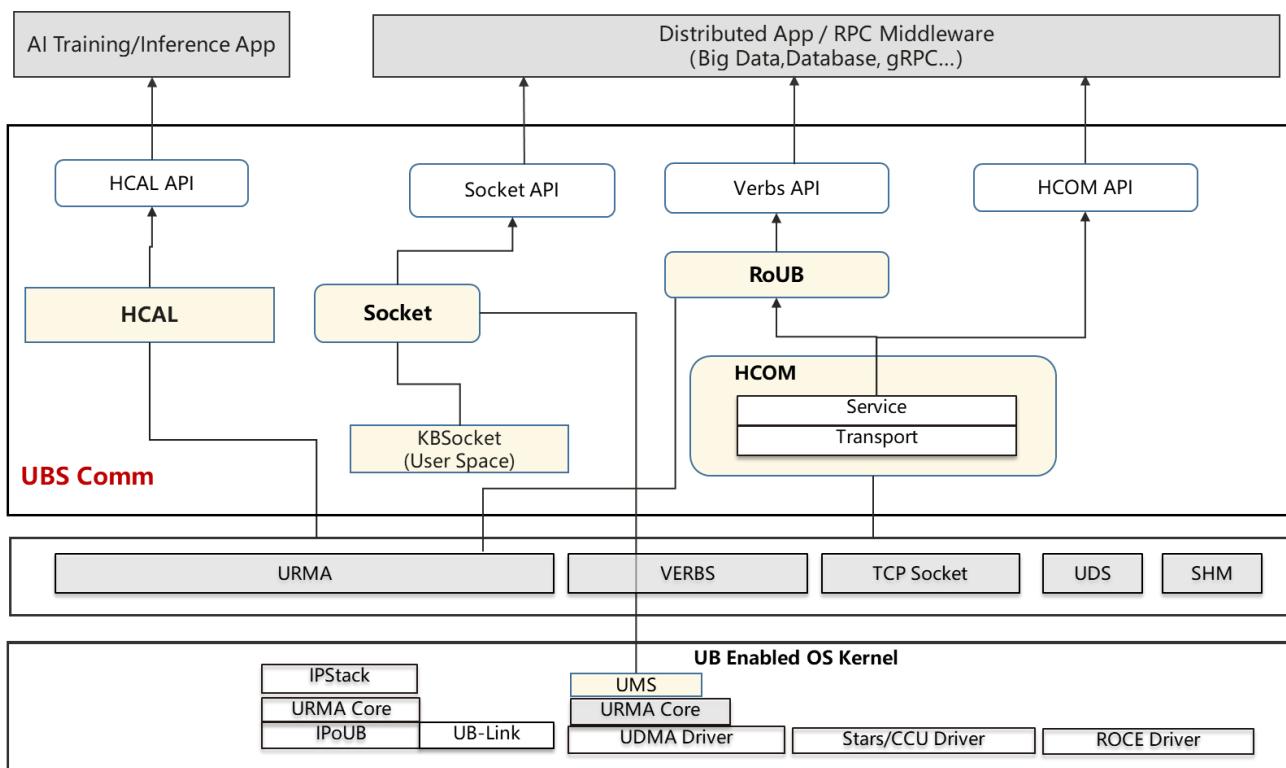


图 5-2 灵衢系统高阶通信服务 UBS Comm 软件架构

### 5.2.2 软件架构说明

UBS Comm 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. **北向接口层**：提供业界主流协议接口，包含兼容的 Socket 接口，以及 UB Native 接口 (HCOM/URPC 等)。
2. **南向适配层**：屏蔽应用层和硬件的差异，提供自适应的适配能力，支持跨代际演进。

## 5.3 使用场景

HCOM、RoUB、Socket、HCAL 四个关键特性，其典型使用场景如下：

1. **HCOM**: HCOM 是一个适用于高带宽和低延迟网络 C/S( Client/Server )架构应用程序的多协议、高性能通信框架。HCOM 提供一组支持各种协议的高级 API，并屏蔽了包括 RDMA、TCP、URMA、SHM 等低级 API 的复杂性与差异性，同时尽可能发挥硬件能力，以保证其拥有高性能。

2. **Socket**: 通过兼容 Socket 使能客户应用无感接入超节点，应用端到端性能提升。
3. **HCAL**: HCAL 通过 UBPU ( NPU/SSU/DPU/GPU/... ) / 跨机 CPU 之间直接通信，借道通信提升 AI 推理场景下的 KV Cache 通信性能，同时还可以通过插件的形态接入到生态中间件 NIXL 和 Transfer Engine 中。
4. **RoUB**: 将 UB 网络接口转换成 verbs 接口，从而使能灵衢网卡对传统 RDMA 应用的无感迁移至超节点。

# 6 灵衢系统高阶 IO 服务 UBS IO 组件

## 6.1 简介

UBS IO 基于 UB 实现数据的全局读写缓存，缩短 IO 的处理路径，支持业务端到端性能提升。UBS IO 在 UB 系统架构的位置如下图所示：

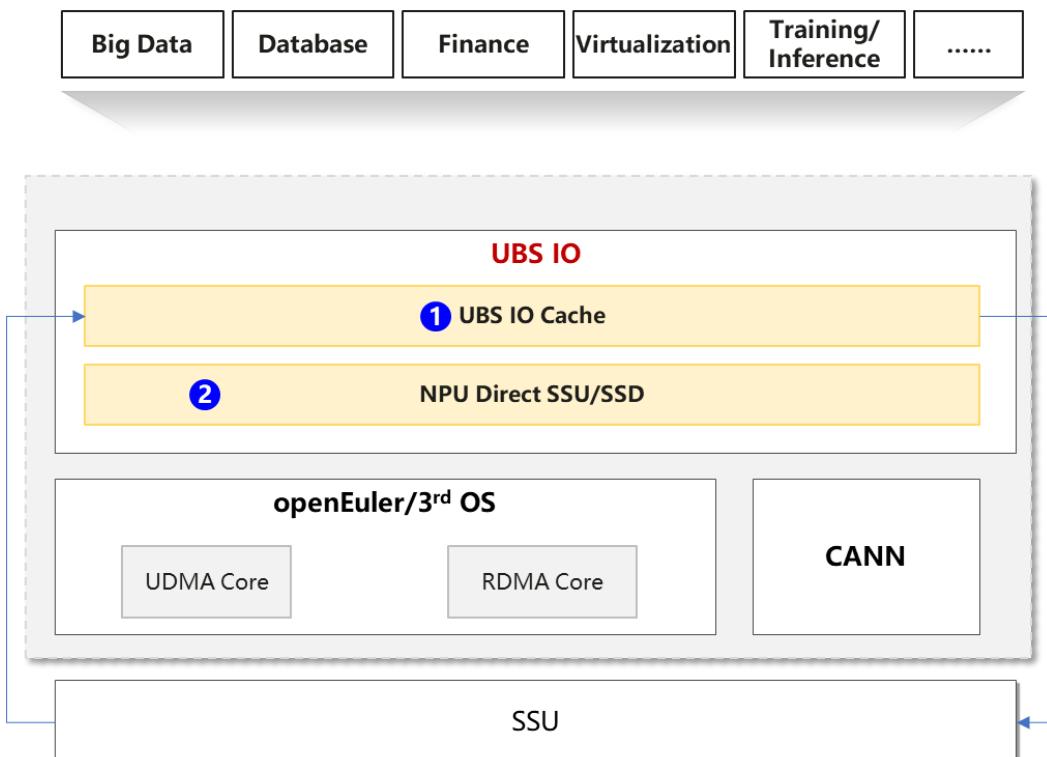


图 6-1 灵衢系统高阶 IO 服务 UBS IO 系统位置示意

UBS IO 的主要功能包括：

- ① **UBS IO 缓存**：基于超节点，实现对应用亲和的全局数据读写缓存，缩短 IO 的访问路径，支持各种不同应用性能提升；
- ② **NPU Direct SSD/SSU**：bypass Memory 上的数据直通到本地 SSD/SSU，提高本地 DRAM 的带宽利用率，加速 HBM 到 SSD/SSU 之间的数据传输效率，提升推理场景性能。

## 6.2 架构

### 6.2.1 软件架构

UBS IO 的软件架构如下图所示：

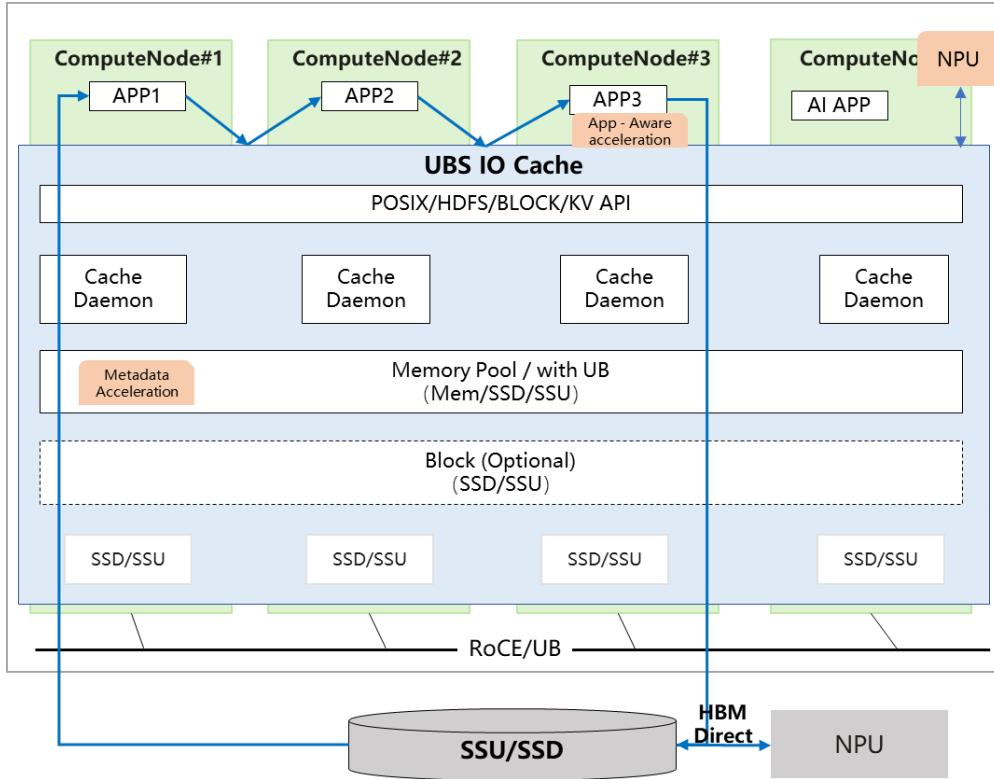


图 6-2 灵衢系统高阶 IO 服务 UBS IO 软件架构

### 6.2.2 软件架构说明

如上图所示，UBS IO 主要功能模块如下：

1. **APP-Aware acceleration(应用感知加速)**: 识别应用的数据访问模式，根据访问模式提前预取或者淘汰数据。
2. **多协议支持**: 支持 POSIX/HDFS/block 等协议，以提升本模块的应用范围。
3. **Cache Daemon**: 实现对缓存数据的管理与访问，包括对缓存介质的空间管理、数据访问、跨节点的数据读取等。
4. **Metadata Acceleration(元数据加速)**: 基于硬件，如网卡/DPU 等加速 IO 元数据的访问，以提升数据在本模块的处理效率。
5. **HBM Direct( HBM 直通加速 )**: 实现 HBM 到存储介质( SSD/SSU )或存储系统( 如 OceanDisk 等 )的直通访问，加速数据在他们之间的访问效率。

## 6.3 使用场景

UBS IO 的典型使用场景如下：

1. **Spark/Hive 数据分析 IO 加速**: 针对存算分离的场景下，数据读取成为瓶颈的现状，加速 Spark/Hive 中状态数据保存逻辑，提升应用端到端性能。
2. **大模型训练 CKPT 加速**: 大模型训练中 CKPT 的保存与读取性能对模型训练及故障恢复至关重要，通过将 CKPT 的数据缓存到本模块中，可以加速训练中 CKPT 的保存性能，通过将 CKPT 预取到本模块中，可以缩短大模型训练的恢复时间，提升训练效率。
3. **大模型推理 KV Cache 加速**: 对 KV Cache 的访问性能是推理性能的关键因素，通过将 HBM 中的数据直通到本地或者远端存储系统，可以提高 KV Cache 的访问带宽，从而提升推理的带宽。
4. **HPC 数据缓存加速**: HPC 计算过程中会产生较多的小文件，这些小文件会被频繁的读取和写入，将这些小文件保存在本模块中，并利用计算节点间的高速网络，实现对这些小文件的快速访问，提升计算效率。

# 7 灵衢系统高阶虚拟化服务 UBS Virt 组件

## 7.1 简介

UBS Virt 作为灵衢虚拟化场景套件，提供虚机、裸机、容器运行实体，使能灵衢计算系统性能优势，同时支持通算和智算场景。

其核心职能包括：

- 灵衢虚拟化网络使能：**基于 UB 扩展基础设施网络的数据面和控制面，通过平等池化能力，支持网络资源的灵活配比，从系统级构筑竞争力，使能灵衢虚拟网络能力。
- 灵衢超级虚拟化使能：**提供超节点计算系统下超级虚机和超级容器的使能，提供跨节点级的池化使能、调度、决策等的能力。
- 虚拟化加速：**提供 xPU 虚拟化加速套件。

## 7.2 架构

### 7.2.1 软件架构

UBS Virt 参考逻辑架构如下：

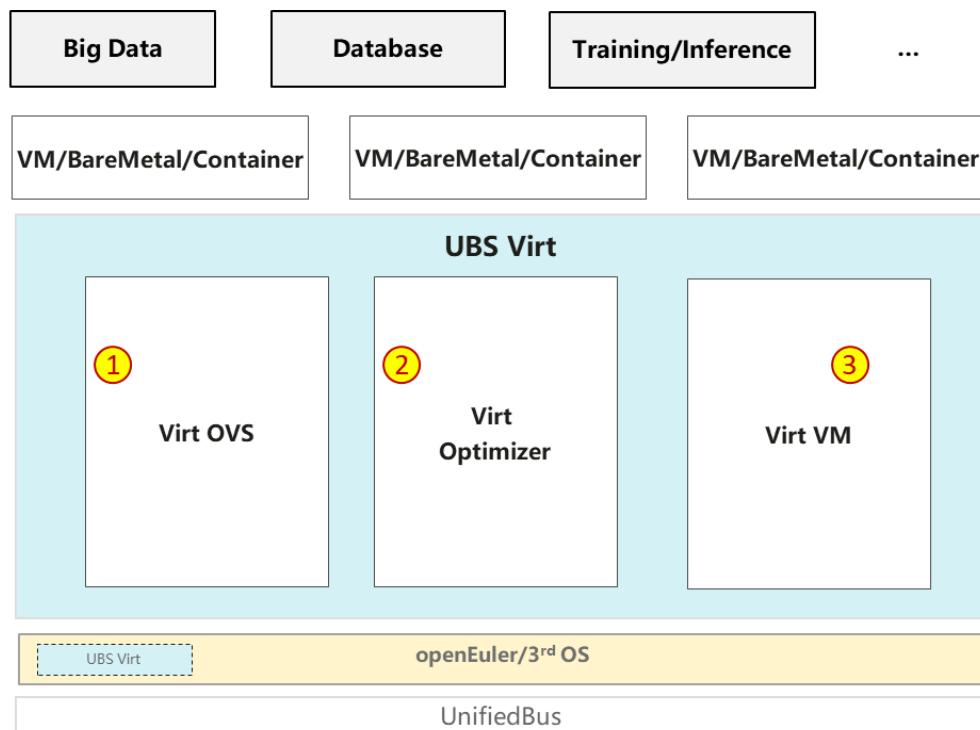


图 7-1 灵衢系统高阶虚拟化服务 UBS Virt 软件架构

如上图所示，UBS Virt 包含以下 3 个关键特性：

- ① **Virt OVS**: 基于超节点架构，使能灵衢网络虚拟化能力，支持高性能虚拟网络、多租隔离和 QoS 控制。
- ② **Virt Optimizer**: 提供智算场景下，虚拟化加速套件与性能调优工具，协同 Host、Guest、Hypervisor、CANN 驱动等多模块提供系统级性能优化。
- ③ **Virt VM**: 基于灵衢总线级池化能力，打破单服务器边界，提供超大规格虚拟机。

## 7.2.2 软件架构说明

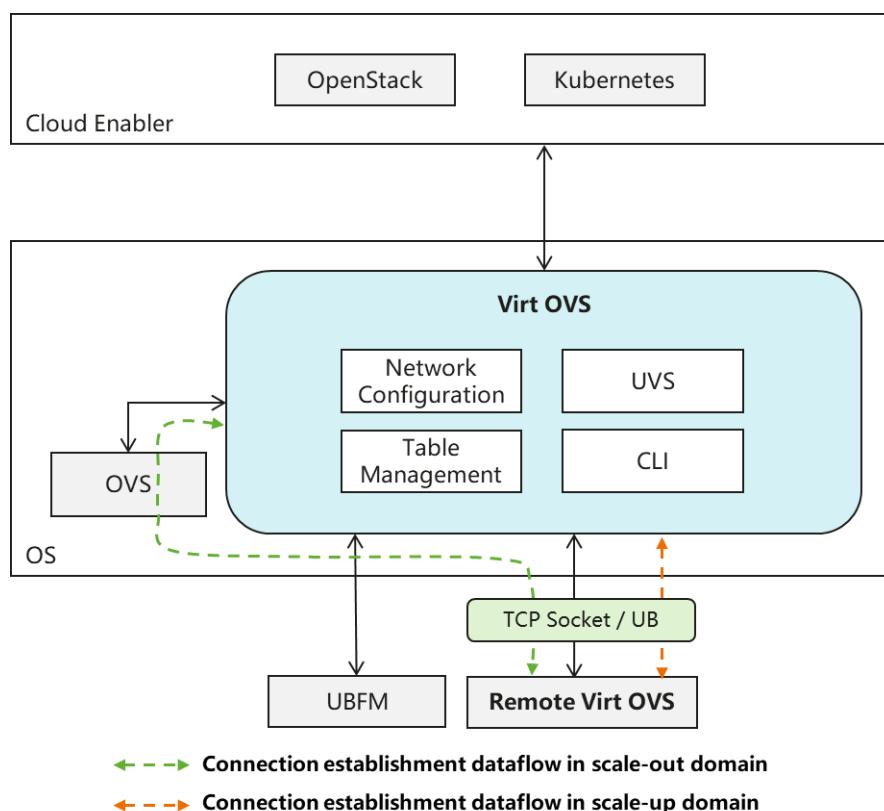


图 7-2 灵衢系统高阶虚拟化服务 Virt OVS 软件架构

Virt OVS ( Virt Open Virtual Switch ) 软件架构如上图所示，其组件功能如下：

1. **Network Configuration ( 网络配置管理模块 )**: 支持 UB 网络配置，包括 EID、UPI、TP、优先级等事务层和传输层网络属性，以实现多租隔离、QoS 控制等网络能力。
2. **UVS ( Unified Virtual Switch ) 模块**: 支持虚拟网络 TP 层协商建链，打通虚拟网络通道。
3. **Table Management ( 表项信息库 )**: 支持 SIP、DIP、vTP 表项信息持久化，保障组件重启/故障时信息恢复。
4. **CLI ( Command-line Interface ) 命令工具**: 提供命令行接口工具，支持云管、虚拟化管理员通过 CLI 实现相关配置。

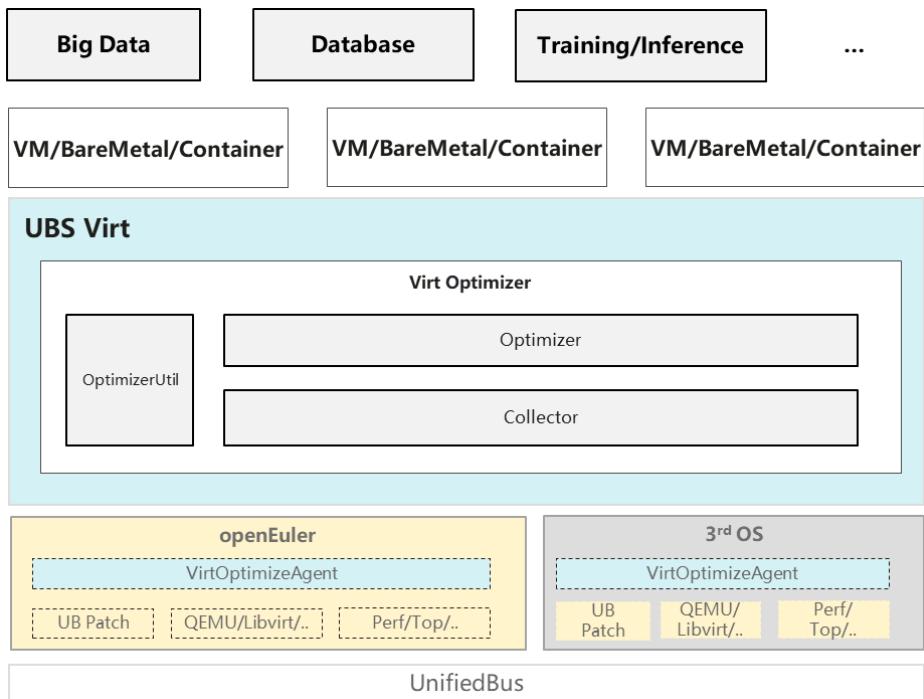


图 7-3 灵衢系统高阶虚拟化服务 Virt Optimizer 软件架构

Virt Optimizer 软件架构如上图所示，其组件功能如下：

1. **Virt Optimizer(加速套件)**: NPU 加速库，完成 NPU 虚拟化性能加速，包括计算、存储和网络。
2. **Virt Optimizer Agent**: 虚拟化调优工具，完成 Host/Guest 自动化探测调优工具，Guest 内 HDK 虚拟化驱动与 Hypervisor 协同加速。

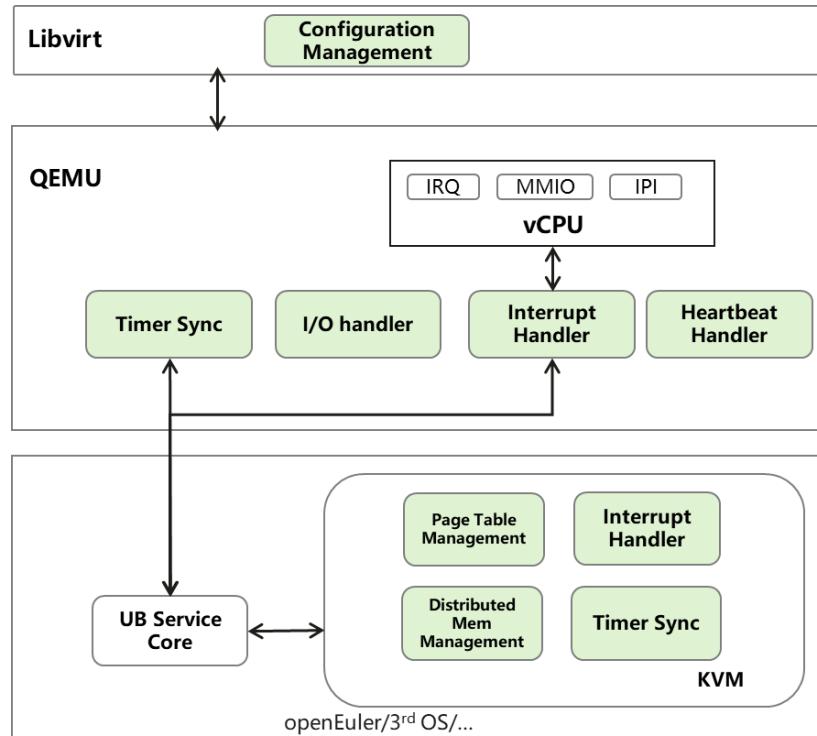


图 7-4 灵衢系统高阶虚拟化服务 Virt VM 软件架构

Virt VM 软件架构如上图所示，其组件功能如下：

1. **Configuration Management ( 配置管理模块 )**：负责解析云管系统下发的超节点拓扑信息。
2. **Timer Sync ( 时间同步模块 )**：确保超节点内 Guest OS 的时间同步。
3. **I/O Handler ( I/O 处理模块 )**：负责超节点内 QEMU 设备 I/O 中断处理与访问。
4. **Interrupt Handler ( 中断转发模块 )**：支持超节点内 QEMU 设备 vGIC 中断转发。
5. **Heartbeat Handler ( 心跳处理模块 )**：负责探测超节点内 UB 链路及 QEMU 状态。
6. **Page Table Management ( KVM-页表管理模块 )**：负责超节点内 GPA/HVA 映射管理。
7. **Interrupt Handler ( KVM-中断转发模块 )**：负责超节点内设备 MSIX 中断转发。
8. **Distributed Memory Management ( KVM-分布式内存管理模块 )**：负责超节点内的 Host OS 内存状态同步（如 share、modify、pin 等）。
9. **Timer Sync ( KVM-时钟同步模块 )**：负责超节点内 Host OS 上时钟同步。

## 7.3 使用场景

UBS Virt 典型使用场景如下：

1. **数据面场景**：基于超节点，扩展基础设施网络的数据面和控制面，通过 UB 平等池化能力，支持网络资源的灵活配比，从系统级构筑竞争力，使能灵衢虚拟网络能力。
2. **虚机场景**：支持超大规格虚机、内存借用、极速热迁移、自动探测并调优 xPU 虚拟化性能等。
3. **容器场景**：支持容器热迁移、快速启动等。

# 附录 A 缩略语

缩略语	全称
API	应用编程接口 ( Application Programming Interface )
CANN	昇腾异构计算架构 ( Compute Architecture for Neural Networks )
CNA	简短网络地址 ( Compact Network Address )
DIP	目地端 IP 地址 ( Destination IP )
EID	UB 实体身份标识 ( Entity Identifier )
HCAL	异构通信加速库 ( Heterogeneous Communication Acceleration Library )
HCOM	高性能通信组件 ( Hyper Communication Library )
RDMA	远程直接内存访问 ( Remote Direct Memory Access )
RoCE	基于融合以太的远程内存直接访问协议 ( Remote Direct Memory Access over Converged Ethernet )
SIP	源端 IP 地址 ( Source IP )
SHM	共享内存 ( Shared Memory )
TCP	传输控制协议 ( Transmission Control Protocol )
TP	传输 ( Transport )
UBFM	灵衢互连结构管理器 ( UB Fabric Manager )
URPC	统一远端功能调用 ( Unified Remote Procedure Call )
URMA	统一远程内存访问 ( Unified Remote Memory Access )