



烽火通信

服务器集群管理平台

技术白皮书

版本：01

烽火通信科技股份有限公司

二零二四年八月

目录

技术白皮书	1
版本：01	1
烽火通信科技股份有限公司	1
服务器集群管理平台	4
1. 前言	4
2. 服务器集群管理平台背景及应用分析	5
2.1.1. 技术层面发展现状	5
2.1.2. 服务器管理平台应用技术分析	6
2.1.3. 发展挑战与解决方案	14
3. 服务器集群管理平台方案设计	16
3.1.1. 服务器集群管理平台架构设计	16
3.1.2. 服务器集群管理平台模块介绍	18
3.1.3. 服务器集群管理平台核心竞争指标	21
3.1.4. 服务器集群管理平台创新点	21
4. 服务器管理平台评测	22
4.1.1. 评测框架	22
4.1.2. 评测概述	23
4.1.3. 任务调度逻辑	25
4.1.4. 评测指标	25
4.1.5. 数据收集	25
4.1.6. 结果分析	25

4.1.7. 评测方法	26
4.1.8. 评测报告	27
5. 评测结果	27
5.1.1. BMC 模块结果	27
5.1.2. BIOS 模块结果	64
5.1.3. 统一身份认证模块结果	81
5.1.4. 资源监控中心模块结果	81
5.1.5. 算力监控中心模块结果	82
5.1.6. 调度中心结果	82
5.1.7. 调度任务管理结果	83
5.1.8. 调度任务详情结果	84
6. 故障定位	85
6.1.1. 故障定位步骤	85
6.1.2. 故障定位案例	86

服务器集群管理平台白皮书

1. 前言

随着人工智能（AI）技术的迅速发展，越来越多的企业和机构在各个行业中采用 AI 算法进行智能化转型。然而，AI 的强大性能依赖于高效的算力支持，特别是在大规模模型训练和推理过程中，需要强大的计算资源来支撑这些复杂的任务。当前，AI 模型训练常常涉及大量的数据处理、模型调优和迭代运算，这对算力提出了前所未有的高要求。在这种背景下，传统的计算资源调度方式已无法满足异构计算环境下对资源高效分配与管理的需求。随着异构硬件（如 NPU、GPU）的广泛应用，如何有效地整合、调度和优化这些计算资源，成为 AI 发展过程中亟待解决的关键问题之一。企业在实践中面临着以下挑战：

- (1) 算力资源利用率低下：大量算力资源在任务调度不当时容易出现闲置或过载的情况，导致整体效率低下。
- (2) 调度复杂性：在异构硬件环境下，手动调度计算任务复杂且易出错，导致人力成本上升且难以快速响应动态需求。
- (3) 任务优先级与资源匹配：在高并发任务环境中，如何确保资源按需分配并保证高优先级任务的及时完成，是一个复杂的问题。

为了解决上述问题，AI 集群服务器管理平台应运而生。该平台通过智能化的调度算法和自动化的资源管理，能够高效地利用企业现有的异构计算资源，提升算力的利用率和任务的执行效率。无论是在 AI 模型训练、推理，还是其他高性能计算场景中，AI 服务器管理平台都可以帮助企业优化资源分配、降低运营成本、加快任务执行速度，从而助力企业在竞争激烈的 AI 时代中占据优势。

本白皮书将详细介绍 AI 服务器管理平台的设计架构、核心功能、技术实现以及典型应用场景，旨在帮助企业和开发者全面了解和应用这一平台，从而最大化地发挥 AI 算力资源的潜力。

2. 服务器集群管理平台背景及应用分析

随着人工智能（AI）技术的迅速发展，越来越多的企业和机构在各个行业中采用 AI 算法进行智能化转型。然而，AI 的强大性能依赖于高效的算力支持，特别是在大规模模型训练和推理过程中，需要强大的计算资源来支撑这些复杂的任务。当前，AI 模型训练常常涉及大量的数据处理、模型调优和迭代运算，这对算力提出了前所未有的高要求。在这种背景下，传统的计算资源调度方式已无法满足异构计算环境下对资源高效分配与管理的需求。随着异构硬件（如 NPU、GPU）的广泛应用，如何有效地整合、调度和优化这些计算资源，成为 AI 发展过程中亟待解决的关键问题之一。企业在实践中面临着以下挑战：

算力资源利用率低下：大量算力资源在任务调度不当时容易出现闲置或过载的情况，导致整体效率低下。

调度复杂性：在异构硬件环境下，手动调度计算任务复杂且易出错，导致人力成本上升且难以快速响应动态需求。

任务优先级与资源匹配：在高并发任务环境中，如何确保资源按需分配并保证高优先级任务的及时完成，是一个复杂的问题。

为了解决上述问题，服务器集群管理平台应运而生。该平台通过智能化的调度算法和自动化的资源管理，能够快速定位故障，高效地利用企业现有的异构计算资源，提升算力的利用率和任务的执行效率。无论是在 AI 模型训练、推理，还是在其他高性能计算场景中，服务器集群管理平台都可以帮助企业优化资源分配、降低运营成本、加快任务执行速度，从而助力企业在竞争激烈的 AI 时代中占据优势。

2.1.1. 技术层面发展现状

2.1.1.1. 异构计算资源调度

当前，服务器集群管理平台的技术正在不断向多样化和智能化发展。针对 GPU、NPU、FPGA 等异构计算资源的调度成为技术热点，通过动态分配资源，提升任务执行效率。平台通常使用如 Kubernetes 等云原生架构，并结合 AI 优化算法，实现对不同计算资源的灵活分配。

2.1.1.2. 自动化运维与智能化管理

服务器集群管理平台普遍开始采用自动化运维技术，依托机器学习和数据分析，平台能够对任务执行进行预测，自动调整资源调度策略，提升算力资源利用率和调度的智能化水平。

2.1.1.3. 云边协同与分布式架构

为了应对分布式 AI 应用的需求，云边协同架构逐渐成为主流。通过服务器集群管理平台，云端和边缘计算能够协同工作，满足从核心数据中心到边缘节点的算力需求。这一架构设计极大地提升了任务执行的效率，特别是对于延迟敏感型应用，如自动驾驶和实时监控等。

2.1.2. 服务器管理平台应用技术分析

服务器集群管理平台的应用技术分析聚焦于如何通过技术创新解决 AI 计算过程中的资源管理、调度优化以及任务执行的复杂问题。以下内容将从调度算法、资源管理、模型训练与推理、边缘计算、数据安全等多个维度进行深入探讨。

2.1.2.1. 调度算法分析

算力管理调度的核心在于如何高效分配硬件资源，使得任务能够在满足性能要求的情况下，尽可能降低资源消耗。调度算法的发展可以分为几个主要方向：

2.1.2.1.1. 静态调度与动态分析

静态调度：传统的静态调度根据预设规则为任务分配固定的硬件资源，但这种方式无法适应实时变化的任务需求和硬件负载。

动态调度：更先进的动态调度根据实时的资源监控数据和任务需求进行资源分配，能够有效应对任务的突发性需求。通过引入动态负载均衡算法，系统可以根据任务优先级、节点利用率和功耗等参数动态调整任务的执行位置。

2.1.2.1.2. 优化调度算法

基于 AI 的调度优化：利用机器学习算法，对历史任务数据进行分析，学习任务执行的模式，从而预测未来任务的资源需求，进行智能调度。

能效优化调度：通过功耗监控，结合能效模型，平台可以在任务性能和能耗之间找到平衡点，优先调度那些在执行任务时功耗较低的节点。

2.1.2.2. 资源管理分析

在服务器集群管理平台中，资源管理是一个至关重要的环节，涉及对多种异构计算资源（如 GPU、NPU、CPU）的统一管理和分配。

2.1.2.2.1. 异构资源管理

GPU 与 NPU 资源管理：GPU 与 NPU 作为深度学习训练与推理的主要硬件加速器，平台需要根据任务的特点（如训练模型大小、数据规模、实时推理需求）合理调配这些资源。通过使用 Kubernetes 的设备插件（Device Plugin），平台能够动态挂载和管理这些加速器资源。

资源隔离与共享：在多租户场景下，不同任务可能会同时竞争同一组资源，平台通过容器化技术（如 Docker、Kubernetes）对资源进行隔离，同时通过资源共享机制（如弹性分配 GPU/CPU）最大化资源利用率。

2.1.2.2.2. 资源调度策略

按需分配：平台根据任务的实际资源需求，按需分配计算资源，避免资源浪费。例如，对于小型模型推理任务，仅分配部分 GPU 算力，而对于大型训练任务，可能需要分配多卡甚至多节点资源。

资源优先级管理：基于任务的优先级，平台能够进行不同的资源调度策略，例如对于紧急任务优先分配资源，低优先级任务则延后调度或分配较少资源。

2.1.2.2.3. 模型训练与推理技术分析

模型训练和推理是服务器集群管理平台的核心任务。为了在不同的 AI 场景下实现高效的计算，平台需要灵活的训练和推理机制。

2.1.2.2.3.1. 模型训练

分布式训练：面对大规模数据和复杂模型，平台支持分布式训练，利用 Horovod、TensorFlow 等深度学习框架的分布式能力，任务可以被拆分为多个子任务，并分配到多个 GPU/NPU 节点上并行执行。这种方式可以极大缩短训练时间，提升资源利用率。

训练任务优化：平台支持动态调整训练参数（如批次大小、学习率等），并结合 AI 算法自动优化训练过程。例如，针对不同类型的模型和数据集，平台可以智能调整超参数以获得更好的训练效果。

2.1.2.2.3.2. 模型推理

高效推理引擎：平台支持通过高效推理引擎（如 TensorRT）加速模型推理过程，优化推理延迟和吞吐量，尤其适用于边缘计算和实时性要求较高的场景。

异构推理：在异构环境下，平台能够根据任务的推理要求动态选择不同的硬件加速器进行推理。例如，使用 NPU 处理神经网络推理任务，同时将部分辅助计算任务交由 CPU 处理，最大化计算效率。

2.1.2.3. 边缘计算与 AI 融合

随着边缘智能的兴起，服务器集群管理平台正逐渐将计算能力向边缘扩展，以满足低延迟、本地化的 AI 推理需求。

2.1.2.3.1. 边缘计算架构

边缘节点部署：边缘计算的核心在于将算力下沉至靠近数据源的边缘节点，以减少数据传输的延迟。平台通过边缘设备和中心服务器的协同调度，将部分推理和数据处理任务分配到边缘节点上，极大提高了响应速度。

边缘 AI 推理优化：针对边缘设备的计算能力有限性，平台对 AI 模型进行轻量化处理（如模型剪枝、量化），在保证推理精度的前提下，降低模型的计算复杂度。

2.1.2.3.2. 边缘计算与中心协同

混合调度模式：平台通过中心与边缘节点协同工作，部分任务在中心节点进行训练，推理任务则分布式下发到边缘节点。通过这种方式，平台可以根据任务的特点选择最合适的计算位置，实现全局最优的资源利用。

2.1.2.4. 多租户数据隔离

容器隔离：通过容器化技术（如 Docker、Kubernetes），平台能够为不同的任务提供独立的执行环境，确保数据不被不同租户之间共享或泄露。

虚拟化技术：通过虚拟化技术，平台可以进一步增强不同任务和用户之间的隔离，避免物理资源共享带来的安全隐患。

数据加密：平台支持数据在存储和传输过程中的全程加密，确保数据即使在传输过程中被截取，也无法被解读。

访问权限控制：通过严格的权限管理，平台对不同用户、任务和数据集进行细粒度的权限控制，确保只有授权的用户才能访问特定数据和模型。

2.1.2.5. 运维与自动化管理

为了确保服务器集群管理平台能够在大规模任务执行下持续稳定运行，运维和自动化管理至关重要。

2.1.2.5.1. 智能运维

自动故障检测与修复：平台通过实时监控任务和硬件运行状态，能够快速检测并识别故障。当某节点发生故障时，系统可以自动调度其他节点接管任务，确保任务不中断。

资源预测与规划：基于历史数据和任务执行记录，平台通过 AI 算法预测未来的资源需求，提前进行资源规划与调度，避免任务执行中的资源瓶颈。

2.1.2.5.2. 自动扩展与缩容

弹性扩展：根据当前任务负载和资源使用情况，平台能够自动扩展或缩减计算节点的数量，确保资源在高峰期得到有效分配，低峰期节约成本。

2.1.2.6. Kubernetes 集群中的算力管理

Kubernetes (K8s) 是目前最流行的容器编排系统，用于管理和调度容器化的应用程序。在服务器集群管理平台中，K8s 起到了关键的基础设施角色，负责调度和管理多种异构计算资源（如 CPU、GPU、NPU 等）。

2.1.2.6.1. GPU 与 NPU 的管理与调度

K8s 集群通过设备插件 (Device Plugin) 机制支持对 GPU 和 NPU 资源的管理。以下是 K8s 集群在服务器集群管理平台中的资源管理技术细节：

GPU/NPU 设备插件：K8s 集群通过安装 NVIDIA GPU Operator 或 Huawei NPU Operator 等设备插件，实现对这些硬件加速器的自动发现、资源分配和管理。设备插件将 GPU/NPU 作为节点资源提供给 K8s 调度器，使其能够识别并分配这些计算资源。

Node Feature Discovery (NFD)：K8s 支持通过 NFD 发现节点的硬件特性，识别出支持加速计算的节点。在提交训练或推理任务时，NFD 自动标识出具有 GPU/NPU 加速能力的节点，并向调度器报告这些特性，供后续资源分配决策使用。

GPU Sharing (GPU 共享)：为了提高 GPU 资源利用率，K8s 支持 GPU 共享机制。通过 NVIDIA 的 Multi-Instance GPU (MIG) 或类似的技术，可以将一张 GPU 划分为多个逻辑实例，供多个容器或任务共享使用。这样，小规模推理任务不必独占整张 GPU，能够大幅度提升 GPU 利用率。

2.1.2.6.2. 容器化和计算资源的隔离

Namespace 和 ResourceQuota: 在多租户场景下, 为了避免资源冲突和过度使用, K8s 提供了 Namespace 和 ResourceQuota 机制。通过为每个租户或任务分配独立的命名空间, 并为其设置 CPU、内存、GPU 的配额, 确保不同租户的计算任务在物理资源层面隔离, 防止任务之间相互干扰。

调度策略与亲和性调度: 为了优化资源利用和减少任务调度延迟, K8s 支持多种调度策略, 包括基于亲和性 (Affinity) 和反亲和性 (Anti-Affinity) 的调度。在服务器集群管理平台中, 可以根据任务的特点, 指定 GPU/NPU 资源的亲和性, 确保计算任务优先分配到满足特定硬件需求的节点上。

2.1.2.7. 调度算法的优化与创新

调度算法是服务器集群管理平台的核心之一, 决定了平台在面对多任务并行时如何智能分配计算资源, 优化任务执行效率和资源利用率。

2.1.2.7.1. 动态调度算法

服务器集群管理平台通常采用动态调度算法来根据任务的实时需求调整资源分配。动态调度算法能够根据以下因素进行决策:

任务优先级: 调度算法会依据任务的优先级对任务进行排序, 确保高优先级任务能及时获得资源。通过基于优先级的调度, 紧急任务能够快速被执行, 而低优先级任务则可能被延迟或平缓调度。

负载均衡: 调度器在分配任务时, 会监控集群中各节点的负载情况。通过负载均衡算法 (如 Round Robin 或 Least Loaded), 平台能够避免单一节点过载, 确保任务分布合理, 优化集群整体的计算效率。

任务能耗优化: 动态调度算法不仅关注性能, 还结合节点的功耗状态进行调度。对于某些能耗敏感型任务, 调度算法会优先分配功耗较低、热耗散良好的节点, 以减少整体能源消耗。

2.1.2.7.2. 基于 AI 的智能调度

随着 AI 技术的发展，服务器集群管理平台开始引入基于 AI 的智能调度算法。这种调度算法通过对历史任务数据进行机器学习，预测未来任务的资源需求和执行时间，从而做出更加精准的调度决策。

任务执行时间预测：通过分析历史任务执行的时间、模型大小、数据集特性等，AI 调度器可以预测新任务的执行时间，从而提前分配合适的节点和资源，避免调度瓶颈。

资源预测和预分配：AI 模型能够预测接下来一段时间内的计算资源需求峰值，提前预留足够的资源，避免高峰时段资源不足。此外，AI 调度器还可以根据长期的资源使用趋势，动态扩展或缩减集群的计算节点。

2.1.2.7.3. 分布式调度与分片任务管理

对于大型 AI 模型训练任务，分布式训练是常见的技术方案。服务器集群管理平台通过调度分片任务在多个节点并行执行，提升计算效率。

分布式训练：K8s 通过 MPI Operator 或 Horovod 结合 Kubeflow 等框架实现分布式训练，将模型训练任务分片分配至不同的 GPU/NPU 节点进行计算。调度算法根据模型的大小、训练数据的规模和节点的计算能力，自动将训练任务合理拆分并分发，最终将结果汇总。

全局与局部调度协同：为了实现分布式任务的最优调度，平台调度器可以基于全局集群的资源情况进行全局调度，而对于边缘节点或本地集群，可以进行局部调度。这种全局-局部调度协同的方式，可以在多集群场景下提升调度效率。

2.1.2.8. 调度任务的管理与自动化流程

在服务器集群管理平台中，调度任务管理流程的自动化是关键，这包括任务的创建、参数配置、部署执行、日志跟踪和结果分析。

2.1.2.8.1. 任务创建与配置

任务定义与配置模板：平台提供了一套标准的任务定义和参数配置模板，用户可以根据任务类型（训练、推理、数据处理等）自定义任务的参数（如数据路径、模型超参数、计算节点等），

简化任务创建流程。

自动化任务下发：通过 Kubernetes 的 Job 和 CronJob 控制器，平台可以实现任务的自动化下发。对于训练任务，用户只需配置好任务模板，平台即可按时调度任务执行。

2.1.2.8.2. 模型训练与推理过程的监控

实时资源监控：在任务执行期间，平台实时监控节点的计算资源使用情况，包括 GPU/NPU 利用率、功耗、显存占用、温度等参数，确保任务在健康状态下执行。

日志追踪与调试：平台通过 K8s 的 kubectl logs 命令和内置的日志追踪模块，用户可以实时查看训练任务的日志。日志内容包括训练进度、模型精度、损失函数值等，用户可以根据这些信息调整任务参数，甚至重新调度任务。

2.1.2.8.3. 模型推理与调度优化

推理任务自动分配：对于在线推理任务，平台能够根据推理的实时性要求，将任务分配到延迟最低的节点执行。使用 TensorRT 等推理引擎时，平台会自动选择适合的硬件加速器并调优推理过程。

推理引擎加速：通过 NVIDIA TensorRT、华为昇腾 CANN 推理引擎等，平台能够极大提升推理速度，尤其是在图像分类、目标检测等深度学习任务中，推理引擎的优化让整体推理效率提升显著。

2.1.2.9. 数据安全与运维自动化技术

2.1.2.9.1. 数据安全技术

加密与审计：平台为任务数据的传输和存储提供全程加密（TLS、AES 等），并通过严格的日志审计记录每次任务执行和数据访问的历史，确保数据安全。

多租户隔离：平台使用 Kubernetes 的 NetworkPolicy 和 PodSecurityPolicy 实现多租户的网络和权限隔离，确保不同用户的任务和数据在网络上不会互相干扰。

2.1.2.9.2. 运维自动化与容灾恢复

自动化容灾恢复：平台通过 K8s 的 ReplicaSet 和 StatefulSet，实现任务的高可用和容灾恢复能力。对于训练任务，平台会定期保存检查点，任务中断时能够从最近的检查点恢复执行。

集群健康检查与自动修复：通过 Prometheus 和 Grafana 监控集群运行状态，平台能够及时发现节点故障，并触发自动修复机制，如重新调度任务或启动备份节点。

2.1.3. 发展挑战与解决方案

尽管服务器集群管理平台取得了显著进展，但在实际发展过程中仍然面临诸多挑战。主要的挑战包括平台需求、能耗、复杂度、生态合作、数据安全以及运维等方面。

2.1.3.1. 平台需求的快速变化

随着 AI 应用场景的日益复杂，不同行业对算力的需求差异巨大，平台需要灵活应对各种计算需求。AI 任务对 GPU、NPU 等资源的需求存在波动，调度平台必须适应任务的多样性和变化性。

解决方案：烽火通信采用智能调度算法，通过机器学习分析历史任务数据，预测算力需求，提前分配和调整资源。结合容器化技术（如 Kubernetes）与 AI 模型优化，使平台能够快速响应任务需求的变化。

2.1.3.2. 能耗问题

大规模 AI 计算任务往往消耗大量能源，尤其在数据中心或大型算力集群中，能耗成为重要问题。GPU、NPU 等硬件的高性能意味着高能耗，如何在保证性能的同时降低能耗是平台亟需解决的问题。

解决方案：烽火通信引入能耗感知的调度机制，基于任务的性能需求和资源特性选择能效最优的硬件。通过 AI 算法优化算力任务的调度路径，降低不必要的能耗损失。此外，发展绿色计算技术，如低功耗芯片、优化冷却系统，也将进一步降低算力平台的能源消耗。

2.1.3.3. 平台复杂度的提升

异构计算资源的管理、复杂任务调度和大规模并发任务都大大增加了服务器集群管理平台的复杂度。不同硬件架构、任务类型和应用场景的差异性加剧了调度平台的运维难度。

解决方案：烽火通信通过模块化设计简化平台架构，分层管理计算资源，降低平台复杂度。借助 AI 自动化运维技术，可以对系统运行状态进行实时监控，自动检测和修复系统故障，减少人工干预。同时，提供开放 API 和标准化接口，以降低平台扩展和集成的复杂性。

2.1.3.4. 生态合作的必要性

解决方案：推动硬件和软件平台标准化，建立与主要硬件供应商的合作伙伴关系，形成资源共享与互通的生态。同时，通过 API 开放平台，吸引更多开发者加入，丰富平台功能和应用场景，形成可持续发展的生态系统。

解决方案：推动硬件和软件平台标准化，建立与主要硬件供应商的合作伙伴关系，形成资源共享与互通的生态。同时，通过 API 开放平台，吸引更多开发者加入，丰富平台功能和应用场景，形成可持续发展的生态系统。

2.1.3.5. 数据安全问题

AI 任务往往涉及海量敏感数据，尤其是在金融、医疗等行业中，数据安全成为服务器集群管理平台必须解决的关键问题。平台在调度过程中需要确保数据在传输和处理时的安全性，防止数据泄露和篡改。

解决方案：加强平台的数据安全防护机制，采用加密传输、访问控制、多重认证等手段，确保任务调度中的数据安全。引入隐私保护技术，如同态加密、联邦学习等，保障数据使用过程中隐私信息不被泄露。此外，定期进行安全审计，及时发现和修复安全漏洞。

2.1.3.6. 运维的复杂性与高效性

大规模服务器集群管理平台的运维挑战日益突出，如何保证系统的高可用性、实时响应性和

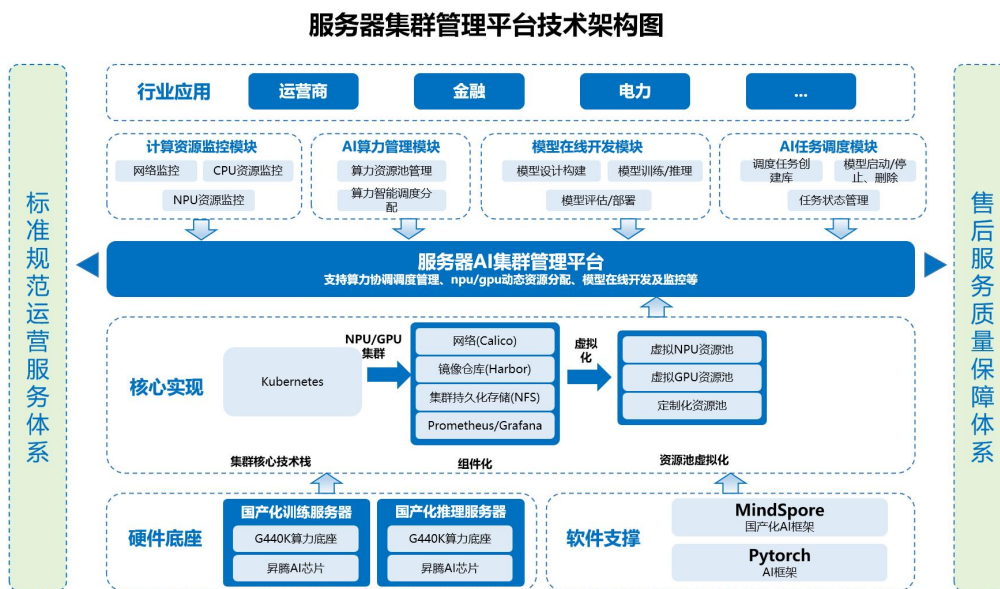
稳定性是运营中的核心问题。运维人员面对成千上万个计算节点和任务，需要高效管理并快速排除故障。

解决方案：烽火通信采用自动化运维工具，通过实时监控、日志分析和智能故障诊断技术，精准定位故障，降低人工运维的复杂性。结合 AIOps（AI for IT Operations）技术，平台能够自动检测异常情况并采取修复措施，减少故障带来的影响。同时，利用可视化管理工具，简化运维人员的操作流程，提升工作效率。

3. 服务器集群管理平台方案设计

3.1.1. 服务器集群管理平台架构设计

烽火通信服务器集群管理平台采用领先的技术架构，旨在为高性能计算需求提供全方位的支持。平台的核心架构包括功能模块、API 接口、资源虚拟化、Kubernetes 组件以及强大的基础设施，确保在不同应用场景中能够实现高效的计算资源管理与调度。以下是平台的具体技术架构图与功能详细描述。



平台总体架构分为以下几个部分：

(1)用户实例层：应用服务层是面向用户的接口，提供了各种功能性服务，帮助用户更好地管理和使用算力资源。该层包含了一系列的服务组件和工具，帮助开发者或运维人员与平台交互。

主要包括：

①AI 平台应用：平台在多种 AI 场景中提供算力调度与优化，例如深度学习模型训练、推理和大数据分析。

②算力资源纳管：通过全面的算力资源管理，整合 NPU 和 GPU 资源，构建高效、灵活的计算资源池，提升资源利用率。

③实时大数据分析：支持金融、医疗等行业中的大数据处理与分析，帮助企业优化决策。

④企业运维：平台具备强大的企业级运维功能，包括任务调度、资源监控和自动化管理。

(2)功能模块层：核心层，包含系统各功能分布：

①算力资源调度模块：该模块负责动态分配和管理计算资源，确保不同任务之间的高效协调。它根据任务优先级和资源使用情况，优化资源分配，提升运行效率。

②算力资源监控模块：实时监控平台的算力资源使用情况，帮助用户掌握资源的健康状况，提供数据支持以优化资源管理决策。

③模型运行状态监控：对深度学习模型的运行状态进行跟踪，确保模型训练与推理任务的正常执行，及时发现并解决潜在问题。

(3)API 接口层：平台提供灵活的 API 接口，方便用户和系统集成：

①Fabric8 API：平台通过 Fabric8 API 实现对 Kubernetes 的管理，简化开发人员的操作，增强平台与容器管理系统的集成。

②基于 Kubernetes-client 封装 API：根据前端需求，对 Kubernetes-client 进行了进一步封装，提供了易于调用的 API 接口，使得用户能够更高效地管理算力资源与任务。

(4)资源虚拟化层：负责 k8s 集群管理、算力资源调度、负载均衡等：

①虚拟 NPU 资源池：通过虚拟化技术，平台可以创建虚拟的 NPU 资源池，灵活分配和调度 NPU 资源，以提升计算效率和灵活性。

②虚拟 GPU 资源池：同样，平台支持虚拟化 GPU 资源池，为需要大量图形处理和深度学习任务的应用提供强大的计算支持。

③定制化资源池：根据行业需求，提供灵活的定制化资源池配置，满足特定应用场景的计算资源需求。

(5)Kubernetes 组件层：包括服务器、GPU、NPU 等计算资源：

①网络（Calico）：平台通过 Calico 提供高效、安全的网络方案，确保数据的高效传输与网络隔离。

②镜像仓库（Harbor）：镜像仓库支持容器镜像的集中管理与存储，确保部署流程的安全性与可控性。

③集群持久化存储（NFS）：支持集群内持久化存储，确保数据安全与一致性，提升系统的可扩展性与可靠性。

(6)基础设施层：用于不同计算节点之间的高效数据传输。这一层的异构资源通过虚拟化技术被抽象为可供调度的资源池，以便上层软件进行动态分配和调度。

①昇腾 NPU 资源：平台集成了华为昇腾 NPU，专为 AI 计算优化，提供强大的算力支持，适应复杂 AI 任务需求。

②英伟达 GPU 资源：支持 NVIDIA GPU 计算，帮助用户高效处理深度学习任务，提供强大的并行计算能力。

③存储设备：提供可靠的存储解决方案，确保高效、安全地存储和管理大量数据。

④网络设备：提供高效的网络基础设施，支持大规模数据的传输与处理，确保系统的高效运行。

3.1.2. 服务器集群管理平台模块介绍

3.1.2.1. BMC 模块

BMC 智能管理系统（以下简称 BMC 系统）是一款针对服务器的系统监测和管理软件。BMC 系统的主要特点如下：

(1) 为您提供优异的用户体验：BMC 系统提供可视化易操作的图形界面，便于您对服务器进行交互式操作。

(2) 为您提供高效的管理维护：BMC 系统提供远程管理和硬件监测功能，便于您随时接入、监测并管理服务器的运行状态。

(3) 为您提供高安全性的系统接入：BMC 系统提供丰富的管理接口，并对所有接口采用高度安全的加密算法。

本文档为您提供在 BMC 系统中进行服务器告警监测、故障定位、系统管理和数据配置的方法以及参数说明。对于数据单位是 TB、GB、MB、KB 或 B 的数值，统一采用 1024 进制进行单位换算。

功能介绍

通过使用“用户登录”界面的功能，您可以登录 BMC WebUI。通过 WebUI 进行界面操作，最多支持 4 个用户同时登录。默认情况下，系统超时时间为 5 分钟，即在 5 分钟内，如果您未在 WebUI

执行任何操作，系统将自动登出，此时需输入用户名和密码重新登录 WebUI。连续输入错误密码的次数达到设定的失败次数后，系统将对此用户进行锁定。锁定时间达到用户设置的锁定时长后，该用户方可正常登录。为保证系统的安全性，初次登录时，请及时修改初始密码，并定期更新。由于网络波动导致资源获取失败，可能会导致 BMC WebUI 显示异常，请刷新浏览器后，重新登录 BMC WebUI。

3.1.2.2. BIOS 模块

基本输入输出系统 BIOS (Basic Input Output System)，是加载在计算机硬件系统上的最基本的软件代码。BIOS 是在操作系统 OS (Operating System) 之下的底层运行程序，BIOS 是计算机硬件和 OS 之间的抽象层，用来设置硬件，为 OS 运行做准备，BIOS 在系统中的位置如

功能介绍：BIOS 主要功能是上电、自检、检测输入输出设备和可启动设备，包括 CPU/内存初始化，硬件扫描和寻找启动设备，启动系统，目前使用鲲鹏 920 处理器的服务器均使用 SPI (Serial Peripheral Interface) Flash 存储 BIOS 代码。



使用鲲鹏 920 处理器的服务器的 BIOS 是具有自主知识产权和专利的 BIOS 产品，具有可定制化，丰富的带外、带内配置功能和丰富的可扩展性等特点。

3.1.2.3. 统一身份认证模块

功能介绍： 多层用户体验，提供身份识别 与访问管理服务：

三级架构用户管理，组织/用户组/用户不同权限管理；

用户数据安全隔离；

用户行为审计；

运维监控管理，资源使用率可视化查询；

日志管理，可视化查询；

3.1.2.4. 资源监控中心模块

- (1) 整体资源显示：包括服务器资源如 CPU、GPU 使用、内存使用等；
- (2) NPU 资源监控可实时监控显示 AI 加速卡使用详情、AI 卡显存使用、功率、温度等；
- (3) 运行资源列表显示集群下 npu/gpu 详细情况，提供搜索功能。

3.1.2.5. 算力监控中心模块

基础信息：集群下服务器 AI 卡 npu 和 gpu 算力详情；

算力资源信息：集群下各节点服务器资源组下各 npu 及 gpu 算力使用情况，提供搜索功能。

3.1.2.6. 调度中心

调度中心：用户资源调度任务部署配置页面，核心参数包括：集群名，AI 框架类型，镜像构建方式、镜像选择、命名空间、数据路径等。

3.1.2.7. 调度任务管理

调度任务列表：列表展示调度部署核心参数信息，用户可对任务进行启动、停止、删除、以及详情操作。

启动：调度任务启动成功后，会立即执行配置的模型训练任务。

停止：强制停止正在运行的模型训练任务。

删除：对初始化或已停止的调度任务进行删除。

3.1.2.8. 调度任务详情模块

任务详情：该功能主要实时监控 模型执行结果信息，包含任务状态（运行中、已结束等状态），模型运行会自动分配节点服务器上（所属节点 ip）以及运行时间。

训练日志：实时获取模型训练日志进行页面显示。

下载日志：可一键下载服务器训练日志到本地。

3.1.3. 服务器集群管理平台核心竞争指标

计算性能	可支持高达 3000TFLOPS 的计算能力，满足大规模模型训练需求。
灵活扩展	支持横向扩展，每增加一个计算节点，整体性能提升至少 20%；支持纵向扩展，单节点性能提升可达 30%。
生态兼容性	支持多种 NPU 和 GPU 硬件，兼容主流深度学习框架。 支持超过 100 种不同的金融工具和软件包；API 调用成功率不低于 99.9%，平均 API 响应时间小于 200 毫秒。
智能调度	资源利用率平均提高 30%以上；任务响应时间缩短至少 50%。
用户友好	新用户首次登录后，可在 15 分钟内完成基本设置；用户界面评分（UI Score）达到 4.8/5（满分 5 分）。

3.1.4. 服务器集群管理平台创新点

服务器管理平台技术亮点包括了 AI 算力资源调度优化、NPU/GPU 虚拟池化、模型推理训练实时分析定位，主要包括如下：

智能化资源调度与优化 平台基于 AI 算法智能调度 NPU 和 GPU 等异构算力资源，实现计算资源的最优分配。通过深度学习和智能算法，平台可以根据任务需求实时动态调整资源使用，减少计算瓶颈并提升资源利用率。同时支持多任务并行调度，保证高效的资源分配和任务执行。

虚拟化资源池与弹性扩展 平台采用虚拟化技术，构建虚拟 NPU 和 GPU 资源池，支持计算资源的灵活调度和弹性扩展。企业可以根据业务需求，动态调整资源池大小，支持按需使用，降低硬件成本。同时通过虚拟化技术，实现不同类型算力资源的统一管理和调度，增强平台的可扩展性和兼容性。

实时监控与大数据分析 平台具备强大的实时监控与大数据分析能力，能够对算力资源、硬件设备、模型运行状态等进行全方位监控。结合大数据技术，平台能够实时采集、分析资源使用数据，生成可视化报表，为运维和决策提供数据支撑，确保任务执行过程中的稳定性和高效性。

4. 服务器管理平台评测

4.1.1. 评测框架

评测框架是对 AI 服务器集群管理平台进行系统评估的结构化方案，主要包括以下几个维度：

4.1.1.1. 性能评测

吞吐量：衡量单位时间内处理的任务数量，通常以每秒完成的任务数（TPS）为单位。

延迟：任务从提交到完成的时间，包括队列等待时间和实际执行时间，通常以毫秒（ms）为单位。

资源利用率：计算资源（CPU、GPU、NPU 等）的利用程度，通过监控工具实时收集资源占用情况。

4.1.1.2. 稳定性评测

容错能力：评估在出现节点故障时，平台是否能够自动转移任务到其他可用节点，减少任务中断的时间。

恢复时间：在故障发生后，系统恢复到正常状态所需的时间，理想情况下应在分钟级别。

4.1.1.3. 可用性评测

用户体验：通过用户调查收集反馈，评估任务创建、管理、监控等功能的易用性。

接口友好性：评估 API 文档的清晰度和易用性，包括接口的完整性和一致性。

4.1.1.4. 安全性评测

数据保护：评估数据在传输和存储过程中的安全性，包括加密措施和访问控制。

权限管理：检查用户访问权限设置，确保敏感数据只能由授权用户访问。

4.1.1.5. 资源管理评测

任务调度效率：分析任务调度日志，评估调度算法在资源分配中的表现。

负载均衡能力：监测在多任务执行时，各节点负载的均衡程度，确保资源不被单点过载。

4.1.2. 评测概述

4.1.2.1. 环境搭建

硬件环境：至少 10 台支持 GPU 和 NPU 的服务器，具有相似的配置。

软件环境：部署 Kubernetes 集群（v1.21.0 及以上），安装 AI 服务器集群管理平台，并配置 Prometheus 和 Grafana 进行监控。

4.1.2.2. 任务设计

设计包含深度学习、图像处理、自然语言处理等多种类型的任务，确保涵盖不同的计算需求。

设定不同的数据集规模（如 1GB、10GB 和 100GB），模拟真实的工作负载。

调度任务的设计通常包括多个关键操作，例如任务的启动、删除、启动和停止。以下是具体描述：

4.1.2.2.1. 任务启动

触发条件：任务可以通过用户操作（如点击“开始任务”按钮）或系统事件（如定时器到达）启动。

参数配置：用户在启动任务时，可以输入任务的相关参数，例如任务名称、执行时间、重复频率等。

状态更新：任务状态从“待启动”更改为“进行中”，并在任务调度系统中记录该任务的开始时间和其他相关信息。

4.1.2.2.2. 任务停止

触发条件：用户可以通过操作（如点击“停止任务”按钮）或任务自动达到某个条件（如任务完成或出错）停止任务。

清理操作：在停止任务时，系统应进行必要的清理操作，如释放占用的资源、记录停止时间及状态。

状态更新：任务状态从“进行中”更改为“已停止”，并记录停止原因（用户主动停止或系统自动停止）。

4.1.2.2.3. 任务删除

确认机制：用户在删除任务时，系统应弹出确认对话框，以防误操作。

任务清理：在任务被删除时，系统需要清理所有与该任务相关的数据，包括任务日志、资源分配等。

状态更新：将任务从系统的任务列表中移除，并更新数据库或任务管理系统。

4.1.2.2.4. 任务重启

条件限制：任务重启通常在任务已停止的状态下进行，防止重复启动正在进行的任务。

重新配置：在重启时，可以选择重新配置任务的参数（例如修改执行时间或频率）或保持原有配置。

状态更新：任务状态从“已停止”变为“进行中”，并更新相关的开始时间和执行记录。

4.1.3. 任务调度逻辑

调度策略：设计任务调度时，可以根据优先级、资源可用性等因素设定调度策略，决定任务的执行顺序。

并发控制：对同时运行的任务数量进行控制，避免资源竞争导致的性能下降。

异常处理：设计异常处理机制，确保在任务执行中出现错误时能够记录错误信息，并决定是否重试、停止或删除任务。

4.1.4. 评测指标

性能指标：吞吐量、延迟、资源利用率等。

稳定性指标：恢复时间、故障率等。

可用性指标：用户满意度评分、接口响应时间等。

4.1.5. 数据收集

采用 Prometheus 监控系统收集资源使用情况，使用 Grafana 可视化数据。

定期生成任务执行报告，包括执行时间、资源占用情况和成功率。

4.1.6. 结果分析

对收集到的数据进行统计分析，比较不同任务和负载下的性能表现。

根据评测结果，识别性能瓶颈和改进方向。

4.1.7. 评测方法

4.1.7.1. 定期评测方法

4.1.7.1.1. 性能测试

使用 TensorFlow Benchmarks 或 MLPerf 等标准基准测试工具，针对深度学习模型训练和推理任务进行性能测试。

记录每个任务的启动时间、执行时间、GPU/CPU 的实际占用情况和资源利用率（例如，GPU 利用率应达到 85% 以上）。

4.1.7.1.2. 稳定性测试

采用故障注入工具（如 Chaos Monkey），模拟节点故障和网络延迟，观察任务在故障情况下的恢复情况。

执行压力测试，通过 JMeter 生成高并发请求，记录在高负载下的系统响应时间和失败率。

4.1.7.1.3. 可用性测试

设计用户测试用例，评估任务创建、配置、提交和监控的流程耗时，确保操作的高效性。

收集用户反馈问卷，评估操作的直观性和易用性，目标是用户满意度不低于 80%。

4.1.7.2. 定性评测方法

4.1.7.2.1. 安全性评测

进行渗透测试，通过 OWASP ZAP 工具评估平台的安全性，检查常见的安全漏洞（如 SQL 注入、XSS 等）。

进行安全审计，检查权限管理设置，确保数据访问权限与预设角色一致，避免越权访问。

4.1.7.2.2. 资源管理评测

通过分析任务调度日志，评估平台的调度算法（如 FIFO、Round Robin、Kubernetes 的优先级调度）在任务执行过程中的表现。

检查调度器在负载增加时的反应时间，确保在任务高峰期，系统仍能保持较低的调度延迟（如小于 2 秒）。

4.1.8. 评测报告

评测完成后，将编写详细的评测报告，包括：

评测背景与目的：明确评测的必要性和目标。

环境配置与测试条件：详细列出硬件、软件环境和配置。

各项评测指标的结果与分析：提供图表和数据，说明性能、稳定性、可用性等方面的表现。

评测过程中的关键发现和改进建议：列出发现的问题及潜在的优化方向。

对平台的整体评估和未来改进方向的建议：总结平台的优势与不足，提供建设性的改进方案。

5. 评测结果

5.1.1. BMC 模块结果

BMC 智能管理系统（以下简称 BMC 系统）是一款针对服务器的系统监测和管理软件。BMC 系统的主要特点如下：

为您提供优异的用户体验：BMC 系统提供可视化易操作的图形界面，便于您对服务器进行交互式操作。






为您提供高效的管理维护：BMC 系统提供远程管理和硬件监测功能，便于您随时接入、监测并管理服务器的运行状态。

为您提供高安全性的系统接入：BMC 系统提供丰富的管理接口，并对所有接口采用高度安全的加密算法。

本文档为您提供在 BMC 系统中进行服务器告警监测、故障定位、系统管理和数据配置的方法以及参数说明。对于数据单位是 TB、GB、MB、KB 或 B 的数值，统一采用 1024 进制进行单位换算。

BMC WebUI 可执行的基本操作如表 7-1 所示。

表 7-1 基本操作

操作	说明
切换界面语言	在登录界面或其他界面中，从下拉列表中切换语言。
查看系统信息	选择“首页 > 更多详情 > 系统信息”。 “系统信息”界面显示服务器的基本信息，包括产品信息、处理器、内存、网络适配器、传感器和其他部件的信息。
查看联机帮助	在 BMC WebUI 页面中，单击  。
查看用户信息	在登录 BMC 界面后，鼠标移至界面右上角  后的用户名，例如“test”。 弹出当前用户信息窗口，显示用户所属的用户名、角色、IP 和时间。
退出系统	鼠标移动至界面顶部的用户名，单击下拉菜单的“退出登录”。
对操作系统上下电	单击  ，可以对操作系统进行上下电操作。 绿色表示操作系统已经上电，黄色表示操作系统已经下电。
设置服务器面板的 UID 灯状态	与服务器自身 UID 灯状态一致，通过本界面即可查看服务器的 UID 灯，不需要去机房查看。 鼠标移至 BMC 界面右上角的  可以从列表中选“点亮”、“关闭”或“闪烁”。 “闪烁”时长为 255 秒。
查看服务器当前告警个数和级别	单击告警个数或告警级别，可以跳转到“维护诊断 > 告警&事件 > 当前告警”页面。 <ul style="list-style-type: none"> ：表示紧急告警，可能会使设备下电、系统中断。因此需要您马上采取相应的措施进行处理。

操作	说明
	<ul style="list-style-type: none"> • : 表示严重告警，会对系统产生较大的影响，有可能中断系统的正常运行，导致业务中断。 • : 表示轻微告警，不会对系统产生大的影响，但需要您尽快采取相应的措施，防止故障升级。


功能介绍

通过使用“用户登录”界面的功能，您可以登录 BMC WebUI。

- ◆ 通过 WebUI 进行界面操作，最多支持 4 个用户同时登录。
- ◆ 默认情况下，系统超时时间为 5 分钟，即在 5 分钟内，如果您未在 WebUI 执行任何操作，系统将自动登出，此时需输入用户名和密码重新登录 WebUI。
- ◆ 连续输入错误密码的次数达到设定的失败次数后，系统将对此用户进行锁定。锁定时间达到用户设置的锁定时长后，该用户方可正常登录。
- ◆ 为保证系统的安全性，初次登录时，请及时修改初始密码，并定期更新。

- ◆ 由于网络波动导致资源获取失败，可能会导致 BMC WebUI 显示异常，请刷新浏览器后，重新登录 BMC WebUI。说明：

如果使用 Internet Explorer 登录 BMC WebUI，需要先开启兼容视图和勾选“使用 TLS 1.2”，操作步骤如下：

- 开启兼容视图：
 1. 单击浏览器右上角的。
 2. 在弹出的快捷菜单中，单击“兼容性视图设置”。
 3. 在弹出的“兼容性视图设置”窗口中的“添加此网站”中输入 BMC 的 IP 地址，并单击“添加”。
 4. 去掉“使用 Microsoft 兼容性列表”的勾选。

开启兼容视图可以解决使用 Internet Explorer 登录 BMC WebUI 后显示不正常的问题。

- 勾选“使用 TLS 1.2”：

1. 选择“Internet 选项 > 高级”。
2. 确保“安全”区域中已勾选“使用 TLS 1.2”。

参数说明

表 1-2 用户登录

参数	描述
用户名	<p>登录 BMC 系统的用户。</p> <ul style="list-style-type: none"> • “域名”选择“这台 BMC”时，支持输入的用户名的最大长度为 20 个字符。 • “域名”选择“这台 BMC”之外的其他选项时，支持输入的用户名的最大长度为 255 个字符。 <p>登录时请注意以下事项：</p> <ul style="list-style-type: none"> • 使用本地用户登录 BMC 时，“域名”可选择“这台 BMC”或“自动匹配”。 • 使用 LDAP 方式登录 BMC 时，支持如下两种格式的用户名： <ul style="list-style-type: none"> - LDAP 用户名（此时“域名”可选择“自动匹配”或指定的域名）。 - LDAP 用户名@域名（此时“域名”可选择“自动匹配”或指定的域名）。 • 使用 Kerberos 方式登录 BMC 时，支持如下两种格式的用户名： <ul style="list-style-type: none"> - Kerberos 用户名（此时“域名”可选择“自动匹配”或指定的域名）。 - Kerberos 用户名@域名（此时“域名”可选择“自动匹配”或指定的域名，且域名中的字母必须为大写）。 • Kerberos 用户名或 Kerberos 用户名@域名支持单点登录。
密码	<p>登录用户的密码，为了保证安全，用户应定期修改自己的登录密码。</p> <p>说明</p>

参数	描述
	以 LDAP 方式或 Kerberos 方式登录 BMC WebUI 时，密码最大长度为 255 个字符。

操作步骤

本指南以 Internet Explorer 11 为例介绍登录 BMC WebUI 的操作步骤。

步骤 1 确认使用 BMC 系统的客户端需具备可用版本的操作系统、浏览器，如果需要使用远程控制功能，则需同时具备可用版本的 Java 运行环境，具体版本要求请参考表 3-68。

步骤 2 配置客户端（例如 PC）IP 地址，使其与 BMC 管理网口网络互通。

步骤 3 通过网线将 PC 连接到 BMC 管理网口。

步骤 4 打开 Internet Explorer，在地址栏中输入 BMC 管理网口地址：“https://ipaddress/”，并按“Enter”。

弹出如图 3-1 所示的安全告警窗口。

图 1-2 安全告警



说明：

登录时可能会弹出“安全告警”界面，您可以选择忽略此告警信息或根据需要执行以下操作屏蔽该界面：

- 如果您有可信任的证书，可以为 BMC 导入信任证书和根证书。
- 如果您没有可信任的证书，且可以保证网络安全的情况下，可以在 Java 的安全列表中将 BMC 添加为例外站点或降低 Java 安全级别。由于该操作可能降低用户的安全性，请谨慎使用。



步骤 6 选择其中一种方式登录 BMC WebUI。

- ◆ [使用本地用户登录 WebUI](#)
- ◆ [使用 LDAP 用户登录 WebUI](#)
- ◆ [使用 Kerberos 用户登录 WebUI](#)

使用本地用户登录 WebUI

步骤 1（可选）在登录界面中，将界面切换至目标语言。

步骤 2 按照[参数说明](#)，输入登录 BMC WebUI 的用户名和密码。

- 如果使用 Internet Explorer 且升级后第一次登录 BMC WebUI，界面可能会提示用户名或密码错误且无法登录，同时按下“Ctrl”+“Shift”+“DEL”，在弹出的窗口中单击删除，这样可以清除浏览器缓存中的内容。再次尝试登录，可以进入 BMC WebUI。
- 如果使用 Internet Explorer 无法登录 BMC WebUI，在 Internet Explorer 中打开“工具 > Internet 选项 > 高级”页面，单击“重置”后，可以正常登录。

----结束

使用 Kerberos 用户登录 WebUI

Kerberos 运行环境：

- ◆ 客户端支持操作系统版本为 Windows 10 64 位，浏览器版本为 Internet Explorer 11。
- ◆ Kerberos 服务器支持操作系统版本为 Windows Server 2012 R2 64 位和 Windows Server 2016 64 位。

Kerberos 用户支持两种方式登录：

- ◆ 通过 kerberos 域用户登录。
- ◆ 通过 SSO 一键登录。

在登录前，请确保以下设置满足要求：

- ◆ 在 BMC WebUI 的“用户&安全 > Kerberos”中，已启用 Kerberos 功能，完成 Kerberos 功能及用户组配置。
- ◆ 在 Kerberos 服务器端已创建 Kerberos 用户组及用户名，并将用户加入 Kerberos 用户组。此用户为登录客户端 OS 的用户。

通过 Kerberos 域用户登录。

步骤 1（可选）在 BMC 登录界面中，将界面切换至目标语言。

步骤 2 按照[参数说明](#)，输入登录 BMC WebUI 的 Kerberos 用户名和密码。

步骤 3 在域名下拉列表中，选择 Kerberos 用户域（例如“ADMIN.COM(KRB)”）或“自动匹配”。

步骤 4 单击“登录”。

成功登录后，显示“首页”界面。

----结束

通过 SSO 一键登录。

步骤 1 使用已在 Kerberos 服务器配置过的 Kerberos 用户名与密码登录客户端 OS。

步骤 2 在浏览器中输入 BMC 的 FQDN 地址，如“https://主机名.域名”。

打开 BMC 登录界面。

步骤 3 单击“单点登录”。

成功登录后，显示“首页”界面。

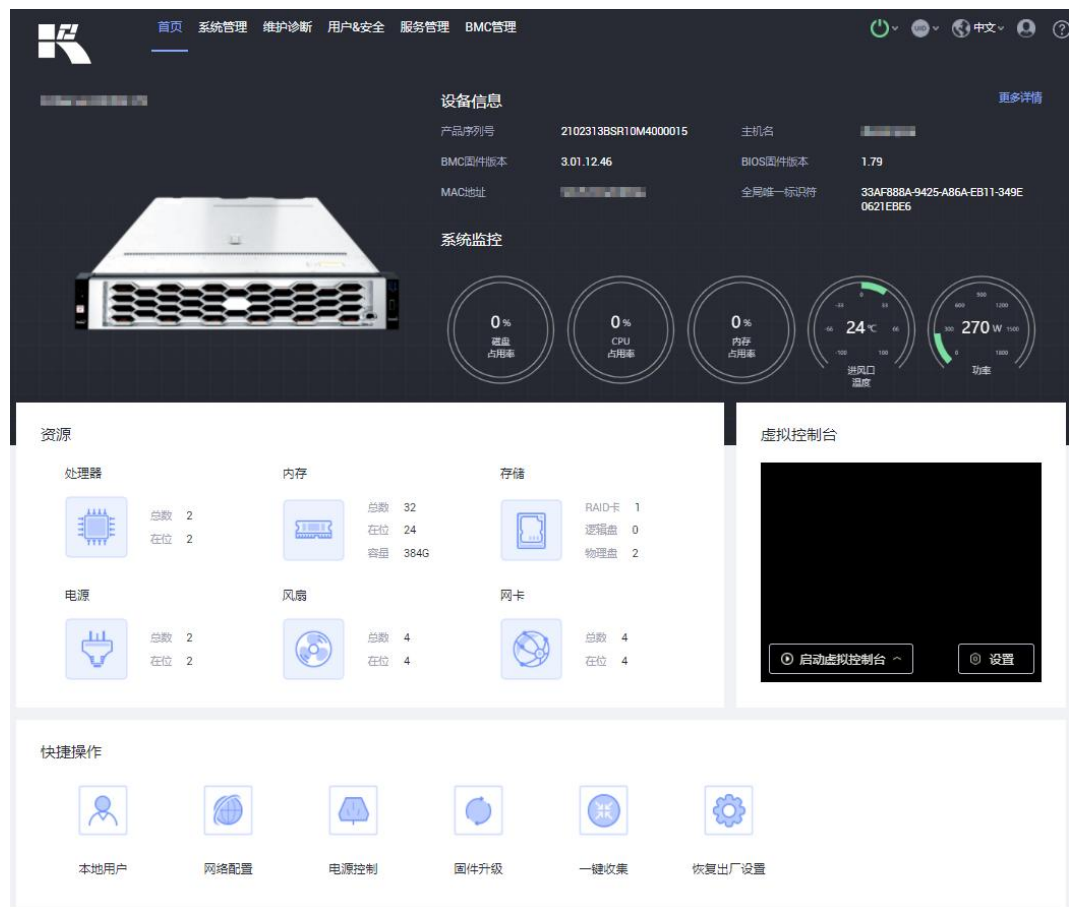
功能介绍

“首页”界面提供了：

- ◆ 服务器的基本信息和节能统计信息。
- ◆ 虚拟控制台。
- ◆ 服务器关键部件的信息及其快捷入口。
- ◆ 系统监控项信息及其快捷入口。
- ◆ 其他常用操作的快捷入口。

在导航栏中选择“首页”，打开如图 3-3 所示界面。

图 1-4 首页 (R220K V2 型号为例)



参数说明

表 1-3 基本信息

区域	展示的信息
设备信息	<p>提供服务器的基本信息，包括：</p> <ul style="list-style-type: none"> • 产品序列号：服务器的序列号。 • 主机名：BMC 的主机名称。 • BMC 固件版本：BMC 系统的固件版本。 • BIOS 固件版本：BIOS 的固件版本。 • MAC 地址：BMC 管理网口物理地址。 • 全局唯一标识符：全球唯一标识。 <p>单击“更多详情”可以跳转到“系统管理 > 系统信息 > 产品信息”界面。</p>
系统监控	<p>提供系统监控快捷入口，包括：</p>

区域	展示的信息
	<ul style="list-style-type: none"> • 磁盘/CPU/内存占用率：单击本入口可以直接跳转到“系统管理 > 性能监控”界面。 • 进风口温度：单击本入口可以直接跳转到“系统管理 > 风扇&散热”界面。 • 功率：单击本入口可以直接跳转到“系统管理 > 电源&功率 > 功率”界面。 <p>说明</p> <ul style="list-style-type: none"> • 当磁盘占用率、CPU 占用率、内存占用率显示的当前值为 0%时，表示未检测到该检测项的当前值。请在 OS 侧安装并运行 iBMA 2.0。 • 当磁盘占用率、CPU 占用率、内存占用率显示为 0%<当前值<门限值时，表示资源使用情况正常。 • 当磁盘占用率、CPU 占用率、内存占用率显示为门限值≤当前值≤100%时，表示资源使用情况已超出紧急预警区间，需要立即处理。 • 功率的检测情况，因服务器不同而采用不同的检测区域。 • 当进风口温度显示为当前值<一级门限值时，表示服务器温度正常。 • 当进风口温度显示为一级门限值≤当前值<二级门限值时，表示温度已超出正常范围，需要处理。 • 当进风口温度显示为当前值≥二级门限值时，表示温度已超出紧急预警区间，需要立即处理。
资源	<p>提供资源信息快捷入口，包括：</p> <ul style="list-style-type: none"> • 处理器：单击本入口可以直接跳转到“系统管理 > 系统信息 > 处理器”界面。 • 内存：单击本入口可以直接跳转到“系统管理 > 系统信息 > 内存”界面。 • 存储：单击本入口可以直接跳转到“系统管理 > 存储管

区域	展示的信息
	<p>理”界面。</p> <ul style="list-style-type: none"> • 电源：单击本入口可以直接跳转到“系统管理 > 电源&功率 > 服务器上下电”界面。 • 风扇：单击本入口可以直接跳转到“系统管理 > 风扇&散热”界面。 • 网络适配器：单击本入口可以直接跳转到“系统管理 > 系统信息 > 网络适配器”界面。
虚拟控制台	<p>从本入口可以进入 HTML5 集成远程控制台或 Java 集成远程控制台。</p> <p>单击“启动虚拟控制台”，在弹出的列表选择独占或共享模式的 HTML5 集成远程控制台或 Java 集成远程控制台。</p> <p>单击“设置”，可以直接跳转到“虚拟控制台”界面。</p> <p>关于虚拟控制台的详细介绍和常见异常帮助请参见：</p> <ul style="list-style-type: none"> • 3.9 虚拟控制台 • 3.9.1 HTML5 集成远程控制台 • 3.9.2 Java 集成远程控制台 • 3.10 远程虚拟控制台异常帮助
快捷操作	<p>提供常用操作的快捷入口，通过以下入口可以快速跳转到相关界面，包括：</p> <ul style="list-style-type: none"> • 本地用户：单击本入口可以直接跳转到“用户&安全 > 本地用户”界面。 • 网络配置：单击本入口可以直接跳转到“BMC 管理 > 网络配置”界面。 • 电源控制：单击本入口可以直接跳转到“系统管理 > 电源&功率 > 服务器上下电”界面。 • 固件升级：单击本入口可以直接跳转到“BMC 管理 > 固件升级”界面。 • 一键收集：单击本入口可以直接下载收集到的维护相关信息，收集到的具体内容请参见本文档 3.11 一键收集信息说明。

区域	展示的信息
	<ul style="list-style-type: none"> • 恢复出厂配置：单击本入口可以弹出“恢复默认”窗口，根据需要确定是否恢复出厂设置。 <p>恢复配置操作会恢复所有用户配置的信息，例如以下配置项，但不限于这些：</p> <ul style="list-style-type: none"> - 当前串口互联状态 - 功率封顶配置 - 删除用户上传的 LDAP 和 SSL 证书 - 用户名、密码、有效期、组信息、登录锁定信息 - IP 获取模式、IP 地址、掩码、默认网关 - SNMP 配置 - 告警上报的 SNMP TRAP 配置、SMTP 配置
用户上次登录信息	<p>登录 BMC 后的前十秒会显示本用户上一次登录的信息，包括：</p> <ul style="list-style-type: none"> • 用户名 • 登录 IP 地址 • 登录时间

通过“系统信息”界面的功能，您可以获取服务器的基本信息，包括产品信息、处理器、内存、网络适配器、传感器和其他部件的信息。

界面描述

在导航栏中选择“系统管理 > 系统信息”，单击“产品信息”，打开如图 3-4 所示界面。

图 1-5 产品信息

产品信息	系统信息
产品名称 生产厂商 资产标签 产品序列号 产品位置	计算机名称 计算机描述 操作系统内核版本 域名/工作组 iBMA服务 iBMA运行状态 iBMA驱动 操作系统版本
主板信息	
BMC固件版本 BIOS版本 CPLD版本 BMC主UBoot版本 BMC备UBoot版本 PCB型号 PCB 版本 单板ID 主板厂商 主板型号 主板序列号 部件编码	

参数说明

表 1-4 产品信息

参数	描述
产品信息	
产品名称	产品名称。
生产厂商	产品的生产厂商。
资产标签	产品的资产标签。 取值范围：长度为 0~48 个字符的字符串，允许输入数字、英文字母和特殊字符。 说明 BMC 的普通用户没有权限设置产品资产标签，仅管理员、操作员或具有“常规配置”权限的自定义用户可以设置产品资产标签。
产品序列号	服务器的产品序列号。
产品位置	服务器的产品位置。 取值范围：长度为 0~64 个字符的字符串，允许输入数字、英文字母和特殊字符。
主板信息	
BMC 固件版本	服务器的 BMC 固件的版本号。

参数	描述
BIOS 版本	BIOS 的版本号。
CPLD 版本	复杂可编程逻辑器件 (CPLD, Complex Programmable Logical Device) 的版本号。
BMC 主 Uboot 版本	用于嵌入式系统的开机引导程序的主用镜像版本号。全称为 Universal Boot Loader。
BMC 备 Uboot 版本	用于嵌入式系统的开机引导程序的备用镜像版本号。全称为 Universal Boot Loader。
PCB 版本	印刷电路板 (PCB, Printed Circuit Board) 的版本号。
单板 ID	单板的 ID。
主板厂商	主板的生产厂家。
主板型号	主板的型号。
主板序列号	主板的序列号。
部件编码	部件的编码。
系统软件	
说明	<ul style="list-style-type: none"> 您必须先先在服务器 OS 侧安装 iBMA 2.0 并完全启动后, 方可在“系统信息”区域框中查询到完整的系统软件信息。 若服务器 OS 侧未安装 iBMA 2.0, 请获取最新的 iBMA 用户文档及软件包, 并参考 iBMA 用户文档安装 iBMA 2.0。
计算机名称	显示服务器操作系统中定义的计算机名称。
计算机描述	显示服务器操作系统的计算机描述信息。
操作系统内核版本	当操作系统改为 Linux 系统时, 显示其内核版本信息。
域名/工作组	显示服务器操作系统侧的域名或所属工作组。
iBMA 服务	显示服务器操作系统中安装 iBMA 版本信息。
iBMA 运行状态	显示 iBMA 软件运行状态。
iBMA 驱动	显示 iBMA 的驱动版本信息。

参数	描述
操作系统版本	显示服务器操作系统的版本信息。

在导航栏中选择“系统管理 > 系统信息”，单击“处理器”，打开如图 3-5 所示界面。

图 1-6 处理器

CPU1		CPU2	
名称	CPU1	名称	CPU2
厂商		厂商	
型号		型号	
处理器ID	10-D0-1F-48-00-00-00-00	处理器ID	10-D0-1F-48-00-00-00-00
主频	2600 MHz	主频	2600 MHz
核数/线程数	64 cores/64 threads	核数/线程数	64 cores/64 threads
一级/二级/三级缓存	8192/32768/65536 KB	一级/二级/三级缓存	8192/32768/65536 KB
状态	启用	状态	启用
序列号	75C6FAD302404424	序列号	71F57AD30120EF24
其他参数	64-bit Capable Multi-Core Execute Protection Enhanced Virtualization Power/ Performance Control	其他参数	64-bit Capable Multi-Core Execute Protection Enhanced Virtualization Power/ Performance Control

参数说明

表 1-5 处理器

参数	描述
基本信息	<p>显示服务器所有在位的处理器的信息。</p> <ul style="list-style-type: none"> 处理器的名称、厂商、型号、处理器 ID、主频、序列号。 该型号 CPU 支持的核数/线程数。 缓存：包括 CPU 的一级、二级、三级缓存的容量。 状态：CPU 的状态信息。 其他参数：该 CPU 支持的其他技术参数。

内存

界面描述

在导航栏中选择“系统管理 > 系统信息”，单击“内存”，打开如图 3-6 所示界面。

图 1-7 内存

名称	厂商	容量	主频	类型	位置
^ DIMM000		16384 MB	2933 MHz	DDR4	mainboard
详细信息 名称 DIMM000 厂商 容量 16384 MB 最小电压 1200 mV 主频 2933 MHz 类型详细信息 Synchronous Registered (Buffered)					
		部件编码 18ASF2G72PDZ-2G9E1 序列号 24D38F61 位宽 72 bit Rank数 2 rank 类型 DDR4 位置 mainboard			
v DIMM100		16384 MB	2933 MHz	DDR4	mainboard

参数说明

表 1-6 内存

参数	描述
基本信息	显示服务器内存信息。 <ul style="list-style-type: none"> 内存满配个数和当前在位个数。 内存的名称、厂商、容量、主频、类型以及位置。
详细信息	单击内存名称左侧的 v ，显示内存的详细信息，包括内存的部件编码、序列号、位宽、Rank 数量、最小电压、类型详细信息。

网络适配器

界面描述

在导航栏中选择“系统管理 > 系统信息”，单击“网络适配器”，打开如图 3-7 所示界面。






图 1-8 网络适配器





参数说明

表 1-7 网络适配器

参数	描述
	<ul style="list-style-type: none"> 您必须先安装在服务器 OS 侧安装 iBMA 2.0 并完全启动后，方可在“网络适配器”页签中查询到完整的网络信息。 若服务器 OS 侧未安装 iBMA 2.0，请获取最新的 iBMA 用户文档及软件包，并参考 iBMA 用户文档安装 iBMA 2.0。
以太网卡	<p>显示服务器安装的板载网卡或 PCIe 网卡的名称、型号、芯片厂商、单板 ID、厂商、芯片型号、PCB 版本、资源归属（归属 CPU、PCH 或 PCIe Switch）、总线信息、PCIe 槽位号（PCIe 网卡独有）等信息。</p> <ul style="list-style-type: none"> 单击以太网卡子菜单的网卡名称，可以查看成员端口的详细信息，包括端口、状态、网口类型、介质类型、速率、自动协商和全双工状态。 <p>说明</p> <p>以太网卡“端口属性”的状态含义包括以下几种：</p> <ul style="list-style-type: none"> --：表示服务器未安装 iBMA，并且无法获取物理连线状态。 连接：表示服务器未安装 iBMA，物理连线

参数	描述
	<p>状态处于连接状态。</p> <ul style="list-style-type: none"> • 断开：表示服务器未安装 iBMA，物理连线状态处于断开状态。 • NoLink：表示服务器已安装 iBMA，端口未连线，但端口状态为 Up。 • LinkUp：表示服务器已安装 iBMA，端口已连线，且端口状态为 Up。 • LinkDown：表示服务器已安装 iBMA，端口状态为 Down。 <p>• 单击“端口属性”下方的 ，可查看指定网卡的网络属性，包括端口名称、固件版本、驱动名称、驱动版本、总线信息、MAC 地址、永久物理地址、IPv4 信息（地址/子网掩码/网关）、IPv6 信息（地址/前缀长度/网关）、VLAN 信息（VLAN ID、VLAN 使能状态、VLAN 优先级使能状态）。</p> <p>• 单击“端口属性”下方的 ，可查看指定网卡的连接视图，包括交换机名称、交换机连接 ID、交换机连接端口 ID 以及交换机端口 VLAN ID。</p> <p>• 单击“端口属性”下方的 ，可查看指定网卡的 DCB 信息和报文统计信息。</p> <p>• 如果网口安装了光模块，单击“端口属性”下方的 ，可查看指定网卡的光模块信息，包括厂商、序列号、部件名称、设备类型、设备连接类型、接收丢失状态、发送错误状态、波长、设备识别信息、当前温度、当前发送偏置电流、当前发送功率和当前接收功率。</p> <p>• 如果网口插上了电缆，单击“端口属性”下方的 ，可查看指定网卡的电缆信息，包括厂商、序列号、部件名称、设备类型以及设备连接类型。</p> <p>说明</p>

参数	描述
	<p>如果网卡的固件版本不支持使用某个网口，该网口的网络属性显示为空。例如某网卡有 Port1、Port2 两个网口，如果该网卡的固件版本不支持使用 Port2，则 Port2 的网络属性显示为空。</p>
FC 卡	<p>显示服务器安装的 FC 卡的名称、厂商、型号、芯片型号、芯片厂商。</p> <ul style="list-style-type: none"> 单击 FC 卡子菜单的 FC 名称，可以查看成员端口的详细信息，包括端口、FC ID、端口类型、状态。 单击端口属性下方的 ，可以查看指定 FC 卡的网络属性，包括速率、WWPN（World Wide Port Name）、WWNN（World Wide Node Name）、固件版本、驱动名称、驱动版本。
Team	<p>显示汇聚网口的名称、状态、工作模式、IPv4 信息（地址/子网掩码/网关）、IPv6 信息（地址/前缀长度/网关）、MAC 地址、VLAN 信息（VLAN ID、VLAN 使能状态、VLAN 优先级使能状态）。</p> <p>单击汇聚网口子菜单的网口名称，可以查看成员端口的详细信息，包括网卡名称、网口名称、端口号、MAC 地址和状态。</p>
Bridge	<p>显示桥接网口的名称、状态、IPv4 信息（地址/子网掩码/网关）、IPv6 信息（地址/前缀长度/网关）、MAC 地址、VLAN 信息（VLAN ID、VLAN 使能状态、VLAN 优先级使能状态）。</p> <ul style="list-style-type: none"> 单击桥接网口子菜单的网口名称，可以查看成员端口的详细信息，包括网口名称、端口、状态、网口类型和介质类型。 单击端口属性下方的 ，可以查看成员端口的网络属性。

传感器

界面描述

在导航栏中选择“系统管理 > 系统信息”，单击“传感器”，打开如图 3-8 所示界面。

图 1-9 传感器

门限传感器									
序号	传感器	当前值	状态	紧急下门限	严重下门限	轻微下门限	轻微上门限	严重上门限	紧急上门限
1	CPU1 Core Rem (°C)	--	--	--	--	--	--	--	--
2	CPU1 DDR VDDQ (V)	--	--	--	1.14	--	--	1.26	--
3	CPU1 DDR VDDQ2 (V)	--	--	--	1.14	--	--	1.26	--
4	CPU1 DDR VPP1 (V)	--	--	--	2.24	--	--	2.74	--
5	CPU1 DDR VPP2 (V)	--	--	--	2.24	--	--	2.74	--
6	CPU1 DTS	--	--	--	--	--	-1	--	--
7	Cpu1 Margin	--	--	--	--	--	--	--	--
8	CPU1 MEM Temp (°C)	--	--	--	--	--	95	--	--
9	CPU1 VCCIO (V)	--	--	--	0.84	--	--	1.16	--
10	CPU1 VCore (V)	--	--	--	1.23	--	--	2.04	--
11	CPU1 VDDQ Temp (°C)	--	--	--	--	--	110	--	--
12	CPU1 VRD Temp (°C)	--	--	--	--	--	110	--	--
13	CPU1 VSA (V)	--	--	--	0.45	--	--	1.21	--
14	CPU2 Core Rem (°C)	--	--	--	--	--	--	--	--
15	CPU2 DDR VDDQ (V)	--	--	--	1.14	--	--	1.26	--

参数说明

表 1-8 传感器

参数	描述
传感器	<p>传感器是指监控服务器各类指标的模块，可以是逻辑模块或物理实体。</p> <p>说明</p> <p>在搜索框中设置搜索条件后，系统将自动显示符合条件的传感器信息。</p>
当前值	<p>传感器当前监控到的指标信息。</p> <p>说明</p> <p>如果显示为--，表示传感器无法监控到指标。</p>
状态	<p>门限传感器扫描状态：</p> <ul style="list-style-type: none"> OK：表示传感器正常。 --：传感器无法监控到指标。

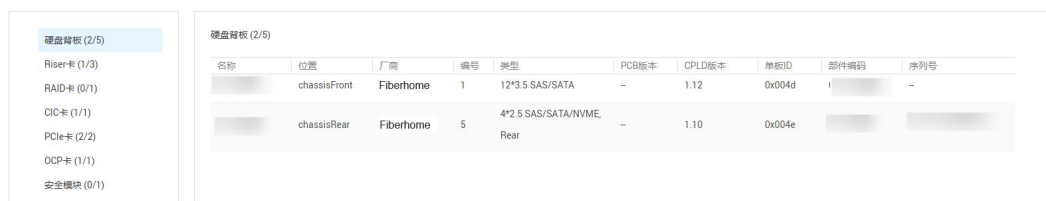
参数	描述
	<ul style="list-style-type: none"> • NC: 表示传感器检测到轻微告警。 • CR: 表示传感器检测到严重告警。 • NR: 表示传感器检测到紧急告警。
紧急下门限	使传感器产生紧急告警的下门限值。
严重下门限	使传感器产生严重告警的下门限值。
轻微下门限	使传感器产生轻微告警的下门限值。
轻微上门限	使传感器产生轻微告警的上门限值。
严重上门限	使传感器产生严重告警的上门限值。
紧急上门限	使传感器产生紧急告警的上门限值。
搜索	在搜索框中设置搜索条件后，系统自动显示符合条件的传感器信息。

其他

界面描述

在导航栏中选择“系统管理 > 系统信息”，单击“其他”，打开如图 3-9 所示界面。

图 1-10 其他



The screenshot shows a sidebar menu on the left with options: 硬盘背板 (2/5), Riser卡 (1/3), RAID卡 (0/1), CIC卡 (1/1), PCIe卡 (2/2), OCP卡 (1/1), and 安全模块 (0/1). The main content area displays '硬盘背板 (2/5)' with a table of hardware details.

名称	位置	厂商	编号	类型	PCB版本	CPLD版本	单板ID	部件编码	序列号
	chassisFront	Fiberhome	1	12*3.5 SAS/SATA	-	1.12	0x004d		
	chassisRear	Fiberhome	5	4*2.5 SAS/SATA/NVME Rear	-	1.10	0x004e		

参数说明

表 1-9 其他

参数	描述
硬盘背板	显示服务器硬盘背板信息。 <ul style="list-style-type: none"> • 硬盘背板满配个数和当前在位个数。 • 硬盘背板的名称、位置、厂商、编号、类型、PCB 版本、CPLD 版本、单板 ID、部件编码以及序列号。
Riser 卡	显示服务器 Riser 卡信息。

参数	描述
	<ul style="list-style-type: none"> • Riser 卡满配个数和当前在位个数。 • Riser 卡的名称、厂商、槽位、类型、PCB 版本、单板 ID、部件编码以及序列号。
RAID 卡	显示服务器 RAID 卡信息。 <ul style="list-style-type: none"> • RAID 卡满配个数和当前在位个数。 • RAID 卡的名称、位置、厂商、编号、类型、PCB 版本、CPLD 版本、单板 ID、资源归属、部件编码以及序列号。
CIC 卡	显示服务器 CIC 卡信息。 <ul style="list-style-type: none"> • CIC 卡满配个数和当前在位个数。 • CIC 卡的名称、厂商、PCB 版本、单板 ID、描述、序列号以及部件编码。
PCIe 卡	显示服务器 PCIe 卡信息。 <ul style="list-style-type: none"> • PCIe 卡满配个数和当前在位个数。 • PCIe 卡的描述、位置、厂商、槽位、制造商 ID、设备 ID、子厂商 ID、子设备 ID 以及资源归属。
OCP 卡	显示服务器 OCP 卡信息（仅针对支持 OCP 卡的服务器）。 <ul style="list-style-type: none"> • OCP 卡满配个数和当前在位个数。 • OCP 卡的描述、位置、厂商、槽位、制造商 ID、设备 ID、子厂商 ID、子设备 ID 以及资源归属。
安全模块	显示服务器安全模块信息。 <ul style="list-style-type: none"> • 安全模块满配个数和当前在位个数。 • 安全模块的协议类型、协议版本、厂商、厂商版本以及自检状态。

功能介绍

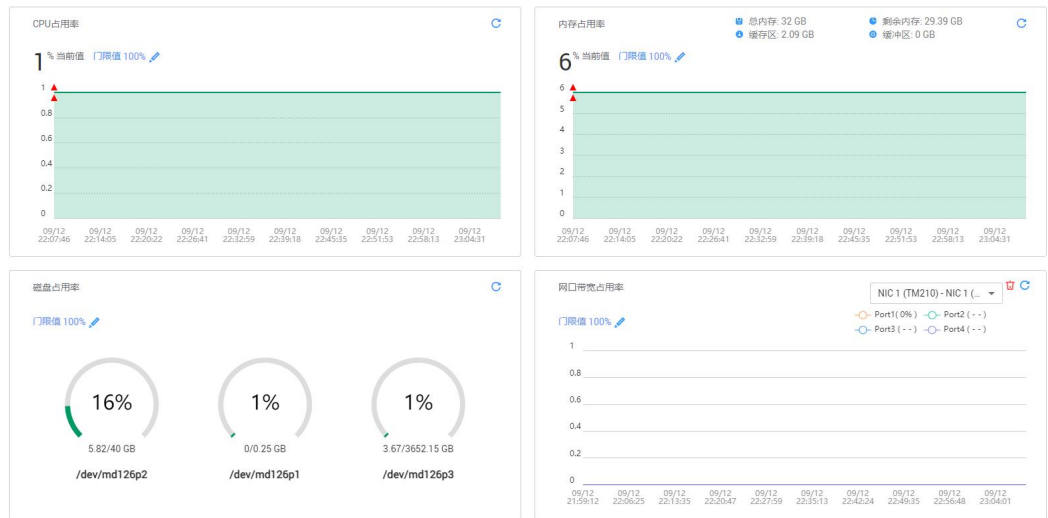
通过“性能监控”界面，您可以：

- ◆ 查看 CPU 最近一小时的占用率。
- ◆ 查看内存最近一小时的占用率。
- ◆ 查看所有磁盘的占用率及磁盘容量信息。
- ◆ 查看所有网口的带宽占用率。

界面描述

在导航栏中选择“系统管理 > 性能监控”，打开如图 3-10 所示界面。




图 1-11 性能监控



参数说明

表 1-10 性能监控

参数	描述
CPU 占用率	<p>运行的程序占用 CPU 资源的比例。</p> <p>说明</p> <ul style="list-style-type: none"> 您必须先服务器 OS 侧安装 iBMA 2.0，并完全启动后，方可查看 CPU 占用率信息。 若服务器 OS 侧未安装 iBMA 2.0，请获取最新的 iBMA 用户文档及软件包，并参考文档安装 iBMA 2.0。
内存占用率	<p>运行的程序占用内存的比例。</p> <p>说明</p> <ul style="list-style-type: none"> 您必须先服务器 OS 侧安装 iBMA 2.0，并完全启动后，方可查看内存占用率信息。 若服务器 OS 侧未安装 iBMA 2.0，请获取最新的 iBMA 用户文档及软件包，并参考文档安装 iBMA 2.0。
磁盘占用率	<p>磁盘分区中已使用的空间占整个分区空间的比例、磁盘分区路径、已使用容量及磁盘分区总容量。</p>

参数	描述
	说明 <ul style="list-style-type: none"> 您必须先 在服务器 OS 侧安装 iBMA 2.0，并完全启动后，方可查看磁盘占用率信息。 若服务器 OS 侧未安装 iBMA 2.0，请获取最新的 iBMA 用户文档及软件包，并参考文档安装 iBMA 2.0。
网口带宽占用率	服务器网卡提供的所有网口的带宽占用比例。 说明 <ul style="list-style-type: none"> 您必须先 在服务器 OS 侧安装 iBMA 2.0，并完全启动后，方可查看网口带宽占用率信息。 若服务器 OS 侧未安装 iBMA 2.0，请获取最新的 iBMA 用户文档及软件包，并参考文档安装 iBMA 2.0。
当前值	服务器当前 CPU、内存、磁盘或网口带宽的占用率。
门限值	服务器当前 CPU、内存、磁盘或网口带宽占用率的门限值，占用率超出设置的门限值后，BMC 会上报一个正常事件。 取值范围：0~100 的整数值。
	打开编辑门限值的输入框。
	刷新相关性能监控项的统计信息。
	清空网口带宽占用率统计信息。


设置门限值

步骤 1 单击待设置目标区域框的 。

弹出门限值输入框。

步骤 2 根据界面提示的取值范围，在输入框中输入门限数值。

步骤 3 单击  保存设置。

设置门限值后，您可以单击  刷新占用率曲线。

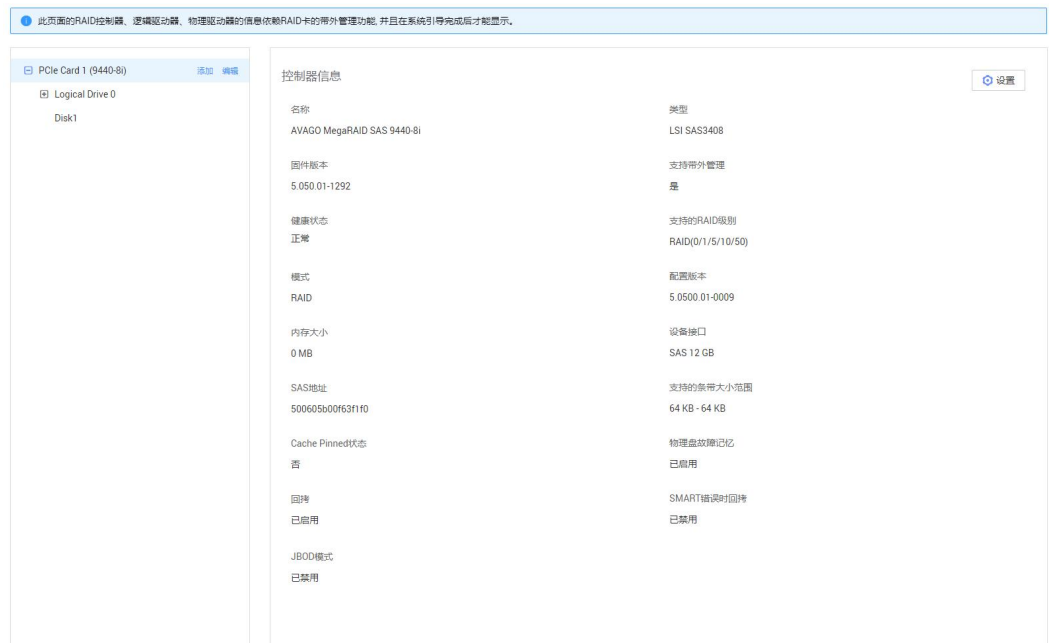
功能介绍

通过使用“存储管理”界面的功能，您可以查看和配置服务器当前存储设备的信息。

界面描述

在导航栏中选择“系统管理 > 存储管理”，打开如图 3-11 所示界面。

图 1-12 存储管理



参数说明

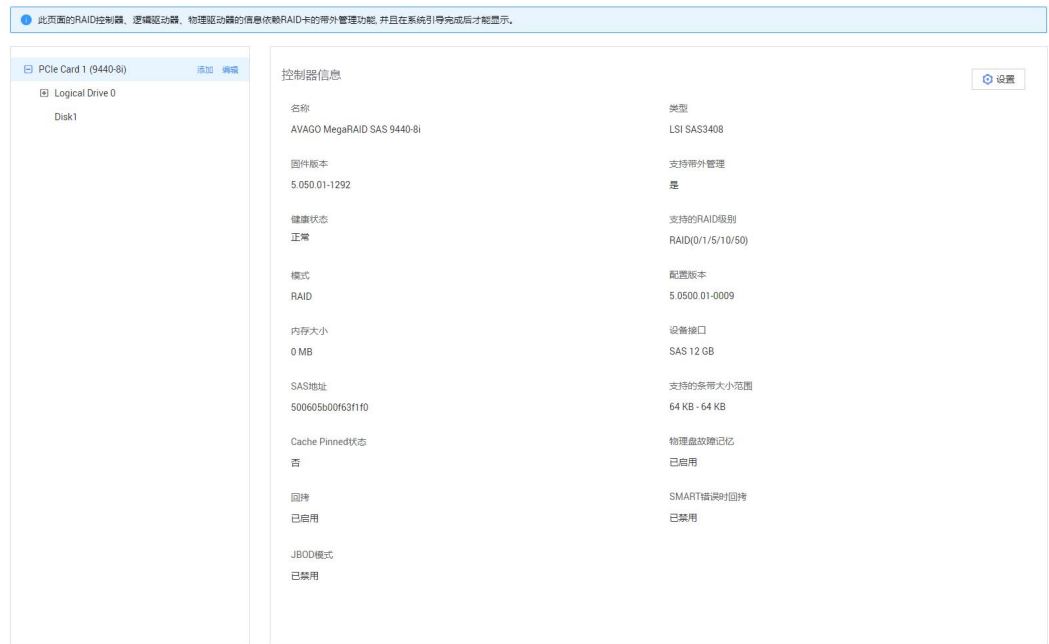
表 1-11 存储管理

参数	描述
RAID 控制器	<p>RAID 控制器信息：</p> <ul style="list-style-type: none"> 控制器名称、类型、固件版本、是否支持带外管理、健康状态、支持的 RAID 级别、模式、配置版本、内存大小、设备接口、SAS 地址、支持的条带大小范围、Cache Pinned 状态、物理盘故障记忆启用状态、回拷启用状态、SMART 错误时回拷启用状态、JBOD 模式启用状态。 BBU 名称、状态、健康状态。 <p>说明</p> <ul style="list-style-type: none"> RAID 控制器不支持带外管理且未安装运行 iBMA 2.0 的情况下，仅显示控制器名称、类型、固件版本以及是否支持带外管理。

参数	描述
	<ul style="list-style-type: none"> 您可以从 RAID 控制卡用户指南的“技术规格”章节中查询该 RAID 卡是否支持 BMC 带外管理。 请不要在 RAID 卡侧将其工作模式设置为 JBOD，BMC 无法识别该模式下的 RAID 卡。详细信息请参考各服务器的 RAID 控制卡用户指南。
逻辑盘	<p>逻辑盘信息：</p> <p>名称、状态、RAID 级别、容量、条带大小、SSCD 功能启用状态、默认读策略、当前读策略、默认写策略、当前写策略、默认 IO 策略、当前 IO 策略、物理盘缓存状态、访问策略、初始化类型、后台初始化启用状态、二级缓存启用状态、一致性校验运行状态、系统盘符、是否为启动盘。</p> <p>说明</p> <ul style="list-style-type: none"> RAID 控制器不支持带外管理且未安装运行 iBMA 2.0 的情况下，无法显示 RAID 控制器下的逻辑盘信息。 您可以从 RAID 控制卡用户指南的“技术规格”章节中查询该 RAID 卡是否支持 BMC 带外管理。
物理盘	<p>物理盘信息：</p> <p>厂商、容量、型号、序列号、固件版本、固件状态、介质类型、接口类型、支持的速率、协商速率、SAS 地址(0)、SAS 地址(1)、电源状态、温度、热备状态、重构状态、巡检状态、健康状态、剩余磨损率、定位状态和累计通电时间。</p> <p>说明</p> <ul style="list-style-type: none"> RAID 控制器不支持带外管理且未安装运行 iBMA 2.0 的情况下，RAID 控制器下挂载的物理盘仅显示接口类型。 直通硬盘仅支持显示健康状态、定位状态和接口类型，且接口类型显示为“SAS/SATA”。 您可以从 RAID 控制卡用户指南的“技术规格”章节中查询该 RAID 卡是否支持 BMC 带外管理。

参数	描述
	<ul style="list-style-type: none"> • 仅 SATA 硬盘及希捷 SAS 硬盘支持累计通电时间的查询。 • 对于 NVMe 硬盘，如果服务器 OS 为 Windows 或 VMware，由于其不支持 NVMe 硬盘接口的速率协商特性，此处“协商速率”显示为“NA”。 • 剩余磨损率表示 SSD 硬盘的使用寿命。剩余磨损率越大，表示硬盘的损耗越小，使用寿命越长；剩余磨损率越小，表示硬盘的损耗越大，使用寿命越短。例如剩余磨损率为 100%，表示硬盘没有损耗。 • M.2 硬盘不支持显示定位状态信息。
控制器配置项	<ul style="list-style-type: none"> • 回拷 • SMART 错误时回拷 • JBOD 模式
逻辑盘配置项	<ul style="list-style-type: none"> • 创建逻辑盘 • 删除逻辑盘 • 修改逻辑盘属性 <p>说明 RAID 卡模式为 JBOD 时，不支持查询和配置逻辑盘信息。</p>
物理盘配置项	<ul style="list-style-type: none"> • 定位状态 • 热备状态 • 固件状态 <p>说明 M.2 硬盘无定位状态配置项。</p>

图 1-13 查看控制器属性

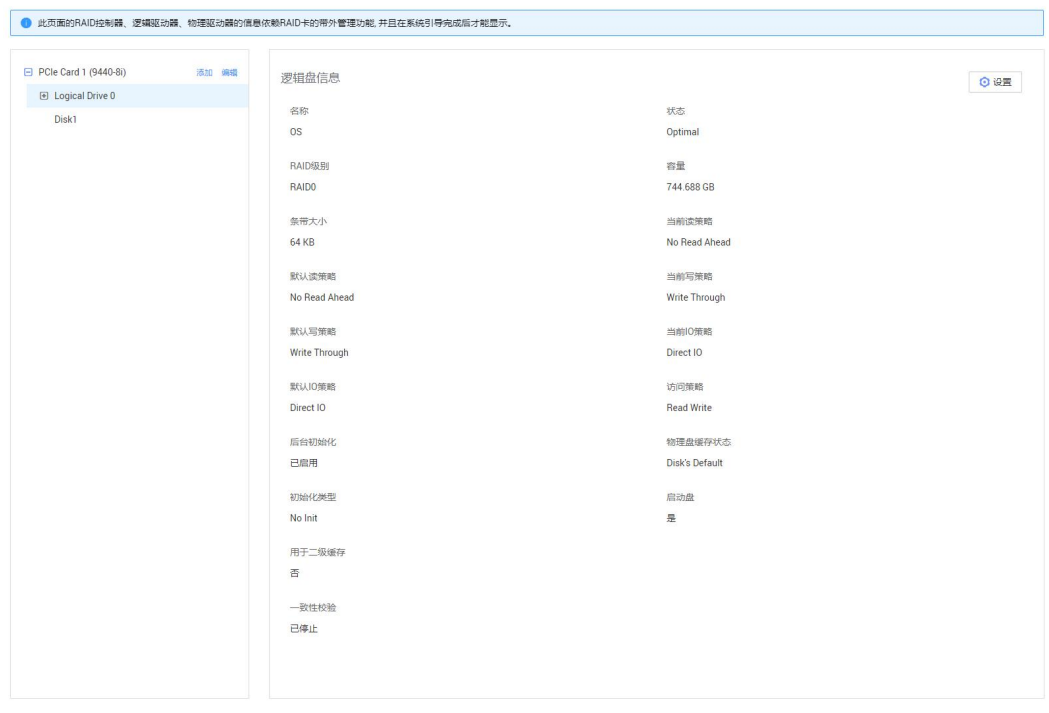


步骤 1 在导航栏中选择“系统管理 > 存储管理”。

步骤 2 选中待查看的 RAID 组。

右侧区域显示 RAID 组的基本属性，如图 3-13 所示。

图 1-14 查看 RAID 组属性



———结束

步骤 1 在导航栏中选择“系统管理 > 存储管理”。

步骤 2 选中待查看的物理磁盘，可以是 RAID 组中的成员盘，也可以是独立的磁盘。

图 1-15 查看物理磁盘属性（成员盘）

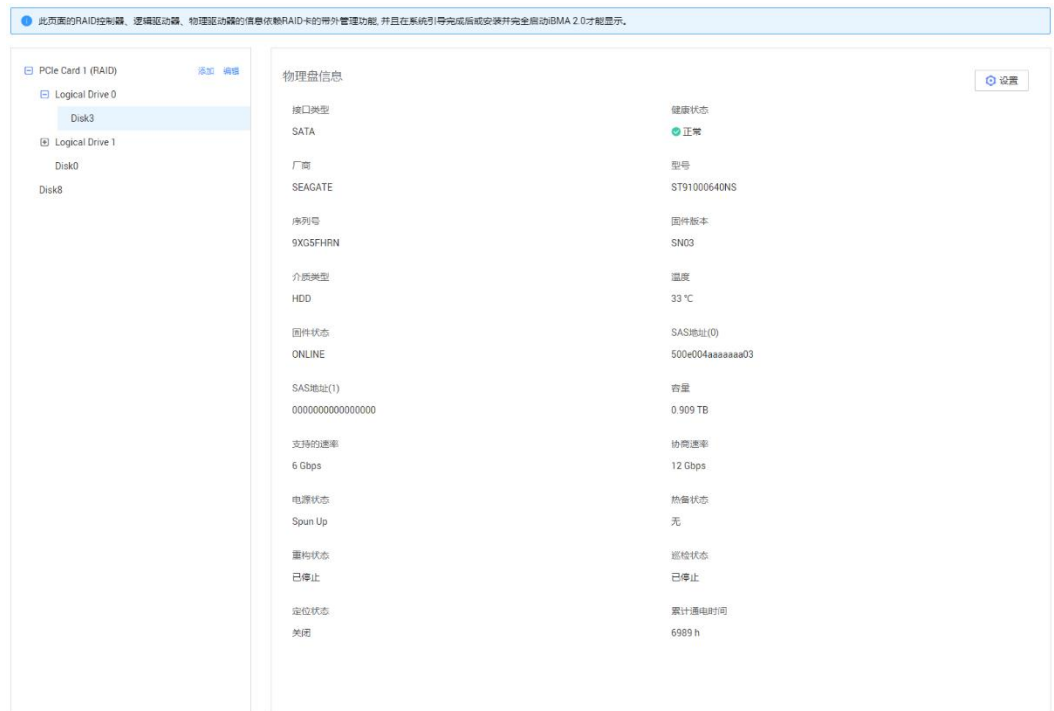
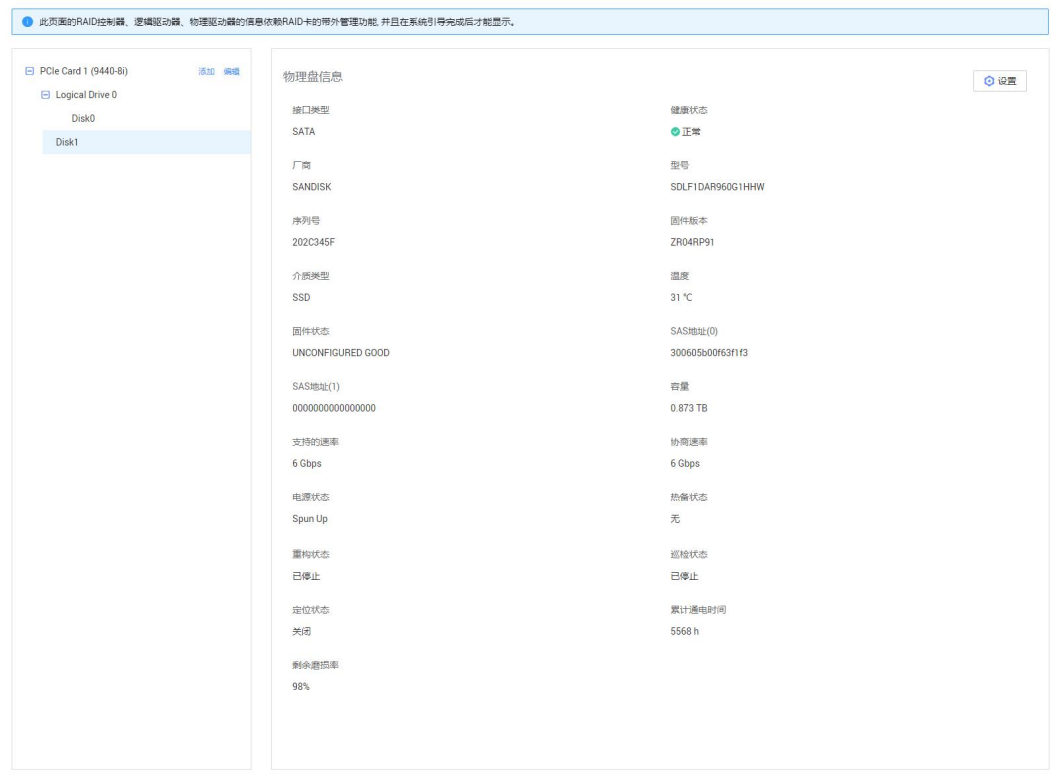


图 1-16 查看物理磁盘属性（单独磁盘）



修改 RAID 控制器属性

表 1-12 控制器配置项说明

配置项	说明
回拷	具备冗余功能的 RAID 的一块成员盘故障之后，热备盘自动替换故障数据盘并开始同步。当更换新的数据盘之后，热备盘中的数据会回拷至新数据盘，回拷完毕后，原热备盘会恢复其热备状态。
SMART 错误时回拷	当控制器检测到 SMART 错误时，执行回拷操作。
JBOD 模式	控制器可对所连接的物理盘进行指令透传，在不配置逻辑盘的情况下，用户指令可以直接透传到物理盘，方便上层业务软件或管理软件访问控制物理盘。
恢复默认设置	单击“恢复默认配置”，可将 RAID 控制器的属性恢复为默认值。
导入 Foreign 配置	单击“导入 Foreign 配置”，可以导入 Foreign 磁盘包含的 RAID 配置信息，无需输入配置文件。

表 1-13 创建逻辑盘配置项说明


配置项	说明
名称	逻辑盘的名称。
二级缓存	是否使能 CacheCade。
条带大小	每个物理盘上的数据条带的大小。
读策略	逻辑盘的数据读策略，包括： <ul style="list-style-type: none"> • Read Ahead: 使能预读取功能。控制器可以预读取顺序数据或预测需要即将使用到的数据并存储在 Cache 中。 • No Read Ahead: 关闭预读取功能。
写策略	逻辑盘的数据写策略，包括： <ul style="list-style-type: none"> • Write Through: 当磁盘子系统接受到所有传输数据后，控制器将给主机返回数据传输完成信号。 • Write Back with BBU: 在控制器无 BBU 或 BBU 损坏的情况下，控制器将自动切换到 Write Through 模式。 • Write Back: 当控制器 Cache 收到所有的传输数据后，将给主机返回数据传输完成信号。
IO 策略	应用于特殊的逻辑盘读取，不影响预读取 Cache。包括： <ul style="list-style-type: none"> • Cached IO: 所有读和写均经过 RAID 控制器 Cache 处理。仅在配置 CacheCade 1.1 时需要设置为此参数值，其他场景不推荐。 • Direct IO: 在读、写场景中的定义不同： <ul style="list-style-type: none"> - 在读场景中，直接从物理盘读取数据。（如果“读策略”被设置为“Read Ahead”，此时读数据经过 RAID 控制器的 Cache 处理。） - 在写场景中，写数据经过 RAID 控制器的 Cache 处理。（如果“写策略”被设置为“Write Through”，此时写数据不经过 RAID 控制器的 Cache 处理，直接写入物理盘。）

配置项	说明
物理盘缓存策略	物理盘 Cache 策略，包括： <ul style="list-style-type: none"> • Enable: 读写过程中数据经过物理盘写 Cache，使写性能提升，但当系统意外掉电时，如果没有保护机制，数据会丢失。 • Disable: 读写过程中数据不经过物理盘写 Cache，当系统意外掉电时，数据不会丢失。 • Disk's default: 保持默认的缓存策略。
访问策略	逻辑盘的访问策略，包括： <ul style="list-style-type: none"> • Read Write: 可读可写。 • Read Only: 只读访问。 • Blocked: 禁止访问。
初始化类型	创建逻辑盘后，对其采用的初始化方式，包括： <ul style="list-style-type: none"> • No Init: 不进行初始化。 • Quick Init: 只把逻辑盘的前 100MByte 空间进行全写 0 操作，随后此逻辑盘的状态就变为“Optimal”。 • Full Init: 需要把整个逻辑盘都初始化为 0，才会结束初始化过程，在此之前逻辑盘状态为“initialization”。
RAID 级别	逻辑盘的 RAID 级别。 说明 RAID 级别为 1 时，仅支持选择 2 个物理盘配置为 RAID1。
每个 Span 的成员盘数	当 RAID 级别配置为 10、50、60 时，需要设置子组中物理盘个数。
物理盘	要加入逻辑盘的物理盘。
可用容量	逻辑盘的可用容量。
容量	逻辑盘的容量。

步骤 4 参考表 3-13 的说明进行配置，并单击“保存”。

步骤 1 在导航栏中选择“系统管理 > 存储管理”。

步骤 2 删除逻辑盘。

- ◆ 单击待删除的逻辑盘右侧的 ，可单独删除指定逻辑盘。
- ◆ 单击 RAID 控制器右侧的“编辑”后，勾选要删除的逻辑盘并单击“删除”，可批量删除多个逻辑盘。
弹出操作确认对话框。

步骤 3 单击“确定”。

———结束

步骤 1 在导航栏中选择“系统管理 > 存储管理”。

步骤 2 选中待操作的逻辑盘。

步骤 3 单击“设置”。

打开逻辑盘编辑菜单如图 3-18 所示，界面中各配置项的含义如表 3-14 所示。

图 1-19 修改逻辑盘



表 1-14 修改逻辑盘配置项说明

配置项	说明
名称	逻辑盘的名称。
默认读策略	逻辑盘的数据读策略，包括： <ul style="list-style-type: none"> • Read Ahead：使能预读取功能。控制器可以预读取顺序数据或预测需要即将使用到的数据并存储在 Cache 中。 • No Read Ahead：关闭预读取功能。
默认写策略	逻辑盘的数据写策略，包括：

配置项	说明
	<ul style="list-style-type: none"> • Write Through: 当磁盘子系统接受到所有传输数据后, 控制器将给主机返回数据传输完成信号。 • Write Back with BBU: 在控制器无 BBU 或 BBU 损坏的情况下, 控制器将自动切换到 Write Through 模式。 • Write Back: 当控制器 Cache 收到所有的传输数据后, 将给主机返回数据传输完成信号。
默认 IO 策略	<p>应用于特殊的逻辑盘读取, 不影响预读取 Cache。包括:</p> <ul style="list-style-type: none"> • Cached IO: 所有读和写均经过 RAID 控制器 Cache 处理。仅在配置 CacheCade 1.1 时需要设置为此参数值, 其他场景不推荐。 • Direct IO: 在读、写场景中的定义不同: <ul style="list-style-type: none"> - 在读场景中, 直接从物理盘读取数据。(“读策略”设置为“Read Ahead”时除外, 此时读数据经过 RAID 控制器的 Cache 处理。) - 在写场景中, 写数据经过 RAID 控制器的 Cache 处理。(“写策略”设置为“Write Through”时除外, 此时写数据不经过 RAID 控制器的 Cache 处理, 直接写入物理盘。)
BGI 状态	是否启用后台初始化。
访问策略	<p>逻辑盘的访问策略, 包括:</p> <ul style="list-style-type: none"> • Read Write: 可读可写 • Read Only: 只读访问 • Blocked: 禁止访问
物理磁盘缓存状态	<p>物理盘 Cache 策略, 包括:</p> <ul style="list-style-type: none"> • Enabled: 读写过程中数据经过物理盘写 Cache, 使写性能提升, 但当系统意外掉电时, 如果没有保护机制, 数据会丢失。 • Disabled: 读写过程中数据不经过物理盘写 Cache, 当系统意外掉电时, 数据不会丢失。

配置项	说明
	<ul style="list-style-type: none"> Disk's default: 保持默认的缓存策略。
是否为启动盘	是否设置该逻辑盘为系统启动盘。
SSCD 缓存功能	是否使用 CacheCade 逻辑盘做缓存。

步骤 1 在导航栏中选择“系统管理 > 存储管理”。

步骤 2 选中待操作的逻辑盘。

步骤 3 单击  展开成员盘。

步骤 4 选中要操作的成员盘。

步骤 5 单击成员盘后的“设置”。

弹出成员盘编辑窗口，如图 3-19 所示，界面中各配置项的含义如表 3-15 所示。

图 1-20 编辑成员盘属性

编辑

定位状态 启用 禁用

热备状态 无 全局 局部

固件状态

配置项	说明
定位状态	物理盘是否已开启定位指示灯。 说明 M.2 硬盘无定位状态配置项。
热备状态	物理盘的热备状态，包括：

	<ul style="list-style-type: none"> • 无：不设置 • 全局：设置为全局热备盘 • 局部：设置为局部热备盘
固件状态	<p>物理盘的状态，包括：</p> <ul style="list-style-type: none"> • UNCONFIGURED BAD：不可用 • ONLINE：在线 • OFFLINE：离线 • UNCONFIGURED GOOD：空闲 • JBOD：直通（OS 直接管理） <p>说明</p> <p>RAID 控制器的 JBOD 模式为“禁用”时，物理盘的固件状态不允许设置为“JBOD”。</p>


表 1-15 成员盘配置项说明

步骤 6 参考表 3-15 的说明进行配置，并单击“确认”。

——结束

步骤 1 在导航栏中选择“系统管理 > 存储管理”。

步骤 2 鼠标移至待操作的物理盘名称。

步骤 3 单击 。

步骤 4 根据实际需要在弹出的提示框中单击“是”。



说明

- 若使用的是支持 first login 密码功能（即 BIOS 默认无密码，第一次进 Setup 界面时，会提示设置新密码）的 BIOS 版本，必须设置完新密码后才能登录进入 Setup 界面。具体请参见 [3.2 进入 BIOS 界面](#)。

- 如果设置的 BIOS 密码被遗忘，可参考 [A.1 如何重置 BIOS 密码](#)来重置密码。

该任务指导用户通过 BIOS 程序对 BIOS 密码进行设置或修改。

5.1.2. BIOS 模块结果

基本输入输出系统 BIOS (Basic Input Output System)，是加载在计算机硬件系统上的最基本的软件代码。BIOS 是在操作系统 OS (Operating System) 之下的底层运行程序，BIOS 是计算机硬件和 OS 之间的抽象层，用来设置硬件，为 OS 运行做准备，BIOS 在系统中的位置如图 2-1 所示。

BIOS 主要功能是上电、自检、检测输入输出设备和可启动设备，包括 CPU/内存初始化，硬件扫描和寻找启动设备，启动系统，目前使用鲲鹏 920 处理器的服务器均使用 SPI (Serial Peripheral Interface) Flash 存储 BIOS 代码。

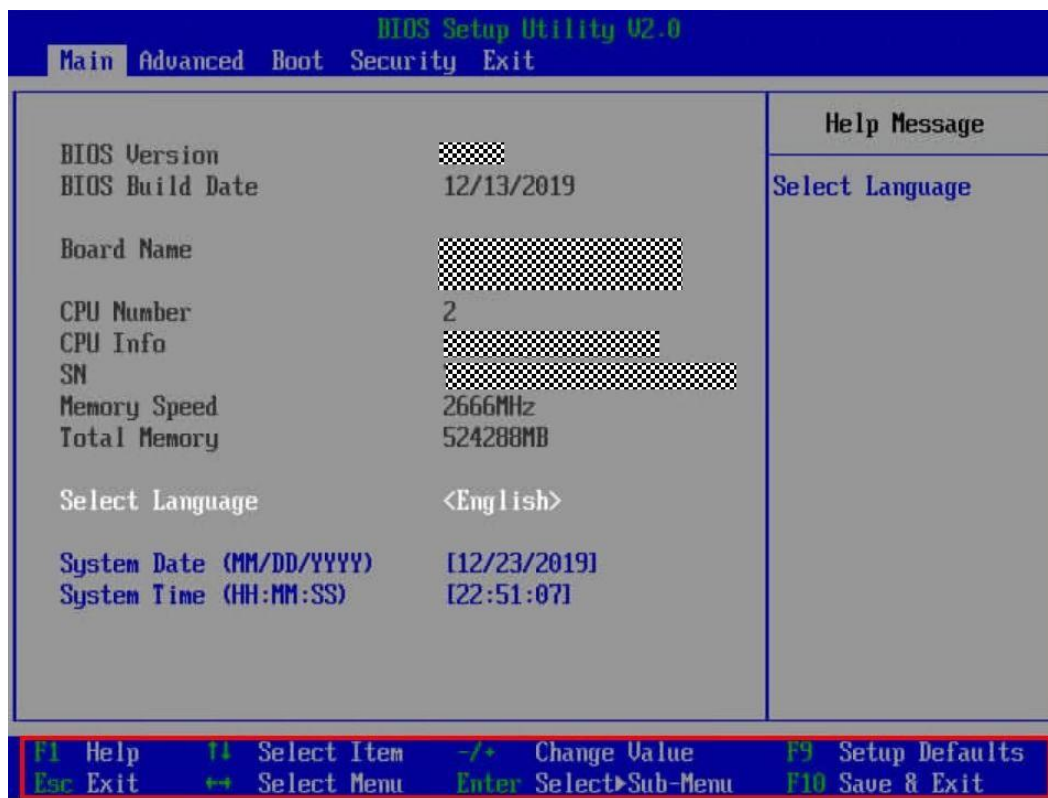
使用鲲鹏 920 处理器的服务器的 BIOS 是具有自主知识产权和专利的 BIOS 产品，具有可定制化，丰富的带外、带内配置功能和丰富的可扩展性等特点。

图 2-1 BIOS 在系统中的位置



BIOS 界面中的操作均需要通过键盘完成，各功能键说明如图 2-2 所示。

图 2-2 BIOS 键盘操作

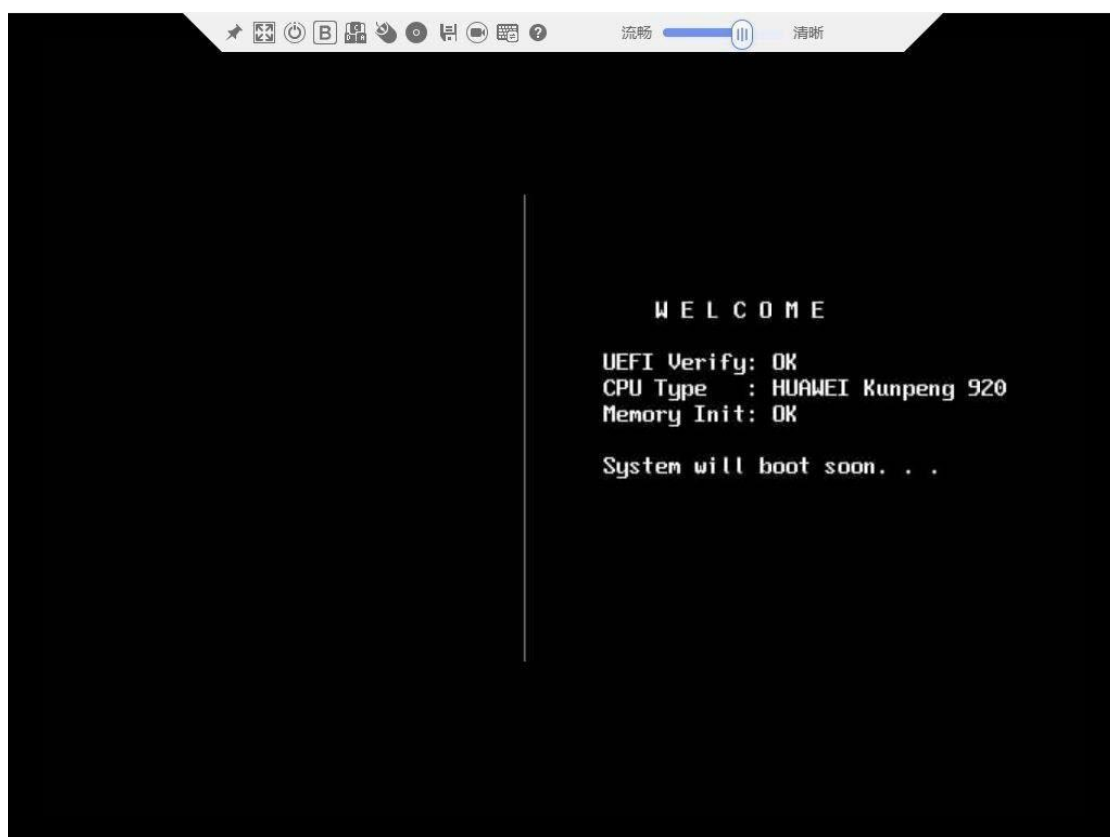


- “F1”：显示参数的帮助信息。
- “Esc”：退出或返回至上一个界面。
- “↑”或“↓”：上下选择参数。
- “←”或“→”：
 - 左右选择参数。
 - 返回上一个界面或进入当前参数的子菜单。
- “-”或“+”：改变参数值大小。
- “Enter” / “Ctrl+M”：选择当前参数或进入当前参数的子菜单。
- “F9”：恢复 BIOS 默认设置。
- “ ”：保存设置并退出。

进入 BIOS 界面前，环境安全启动信息。如图 3-1 所示。

- BSBC 校验 IMU 成功后，界面右上角红色信息为“IMU Verify Success!”。
- IMU 校验 UEFI 成功后，界面右上角红色信息为“UEFI Verify Success!”。

图 3-1 环境安全启动信息



指导用户在需要进行系统启动设置或系统信息查询的情况下，进入 BIOS 界面。

步骤 1 连接好本地线缆（电源线、网线等）并外接键盘、鼠标、显示器或进入 BMC WebUI 的远程管理界面。

📖 说明

进入 BMC WebUI 远程管理界面的具体步骤请参见相应的 BMC 用户指南。

步骤 2 将服务器上电。

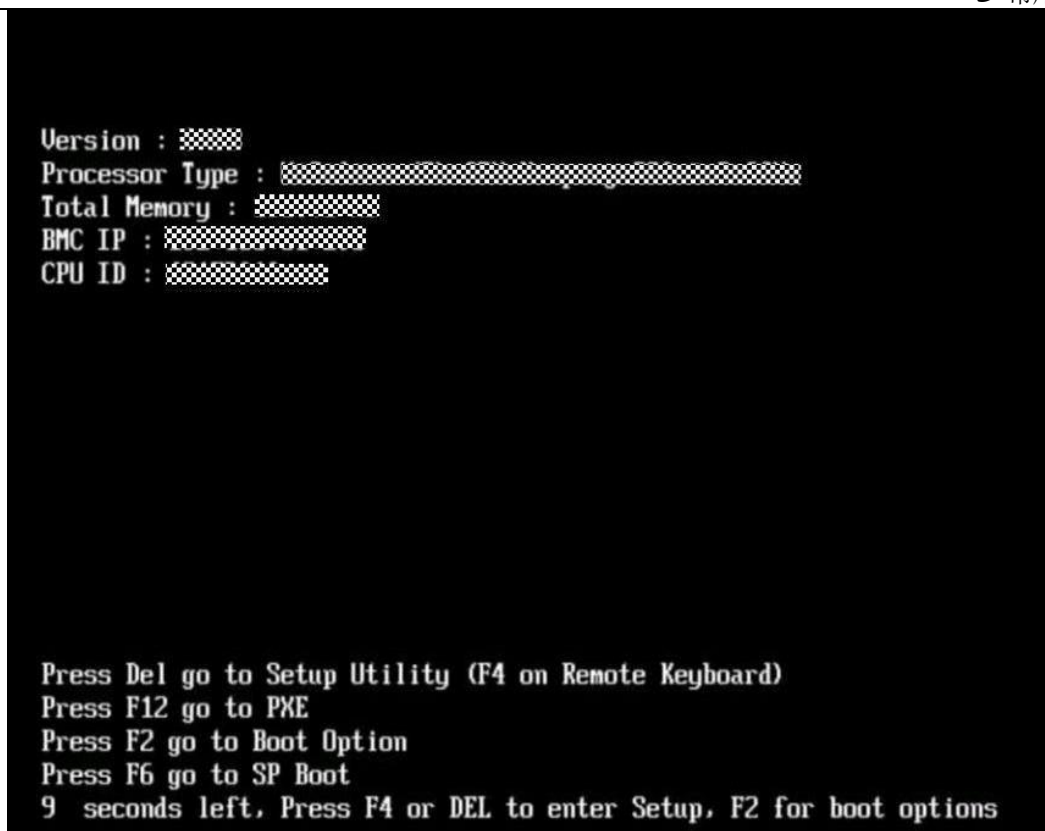
步骤 3 当出现如 [图 3-2](#) 界面时，按键盘的“Delete”或“F4”键。

- 若弹出输入当前密码对话框时，如 [图 3-3](#) 所示，继续执行 [步骤 4](#)，跳过 [步骤 5](#)。
- 若弹出设置新密码提示框时，如 [图 3-4](#) 所示，则跳过 [步骤 4](#)，继续执行 [步骤 5](#)。

📖 说明


- 按键盘的“F12”键从网络启动快捷方式。
- 按键盘的“F2”键进入选择启动项界面。
- 按键盘的“F6”键进入 Smart Provisioning 起始界面。

图 3-2 BIOS 启动界面



步骤 4 输入密码。

在弹出的“Input current password”对话框中输入密码，

如图 3-3 所示。  说明

- 第一次登录后，请立即设置管理员的密码，具体操作步骤请参见

3.6 设置 BIOS


密码。如不修改密码，在弹出提示修改密码信息时，直接按

“Enter”进入 Setup 界面。

- 从安全性考虑，建议定期修改管理员的密码。
- 在输入密码的过程中，默认连续三次输入错误时，机器将会被锁定，按 Security 界面中 Set Lock Time 选项设定的数值等待一段时间后，机器自动解锁。密码错误锁定次数以及锁定时间可分别通过 Security 界面中的“Set Lock Count”和“Set Lock Time”选项设定，具体请参见 4.4 Security。
- 如果设置的 BIOS 密码被遗忘，可参考 A.1 如何重置 BIOS 密码来

重置密码。图 3-3 输入当前密码对话框

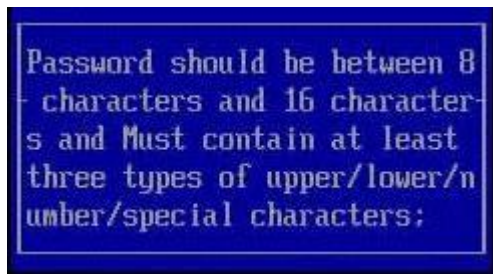


步骤 5 设置并输入
新密码。  说明

若使用的是支持 first login 密码功能（即 BIOS 默认无密码，第一次进 Setup 界面时，会提示设置新密码）的 BIOS 版本，必须设置完新密码后才能登录进入 Setup 界面。

1. 弹出设置新密码提示框时，如 [图 3-4](#) 所示，按“Enter”。

图 3-4 设置新密码提示框



2. 在弹出的“Input new password”对话框中输入新密码，如 [图 3-5](#) 所示。

 说明

密码长度必须在 8~16 位之间，至少包含特殊字符、大写字母、小写字母及数字这四种字符中的三种，其中必须包含特殊字符。

图 3-5 输入新密码对话框



3. 输入新密码后，按“Enter”。弹出密码确认对话框，如图 3-6 所示。

图 3-6 密码确认对话框



4. 再次输入设置的密码后，按“Enter”。弹出成功设置新密码提示框，如图 3-7 所示。

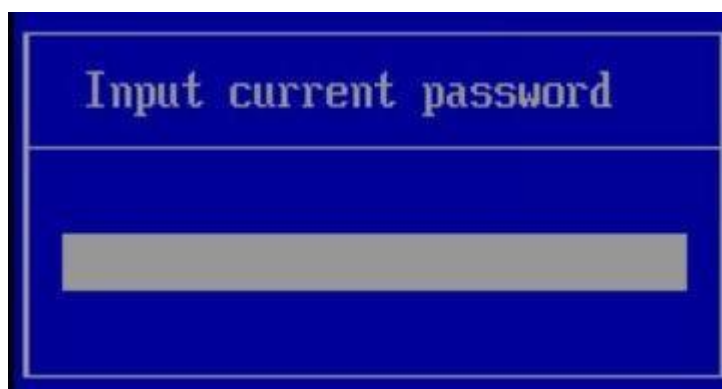
图 3-7 成功设置新密码提示框



5. 按“Enter”。

弹出“Input current password”对话框，如图 3-8 所示。

图 3-8 输入当前密码对话框



6. 输入设置的新密码。

步骤 6 按“Enter”，进入 Setup 界面。

-----结束

该任务指导用户通过 BIOS 程序查询服务器 BMC 的 IP 地址。

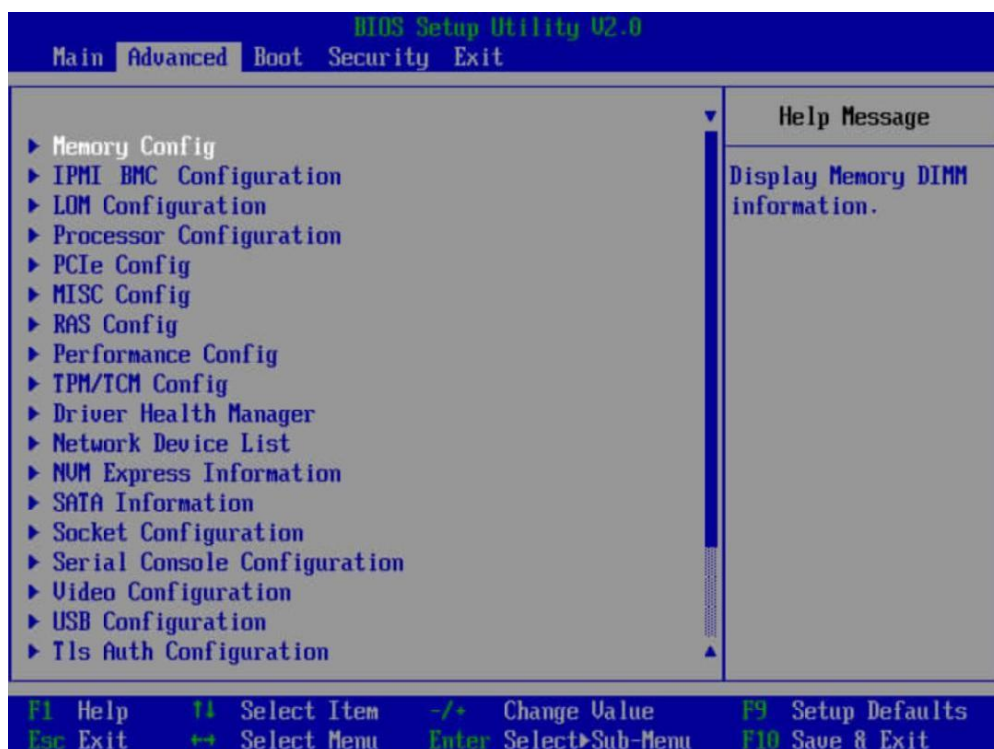
说明

BMC IP 地址范围限制为：1. x. x. x ~ 223. x. x. x，不包含 127. x. x. x。

步骤 1 进入 BIOS 界面，具体操作步骤请参见 3.2 进入 BIOS 界面。

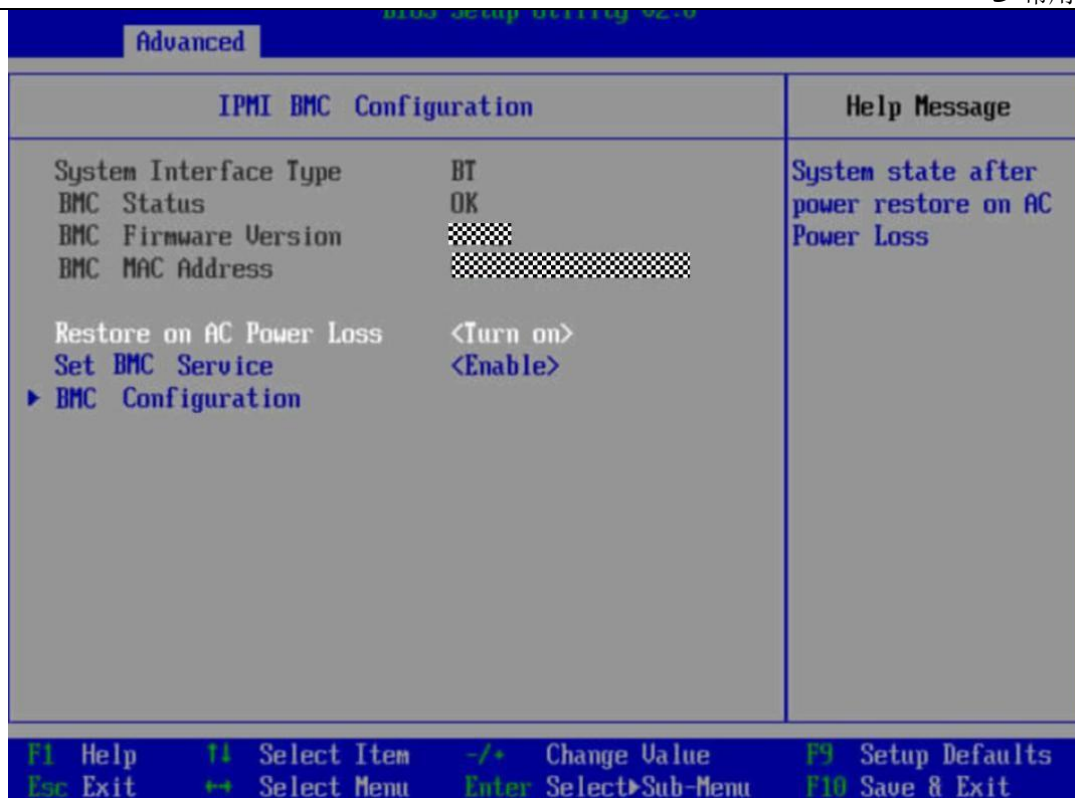
步骤 2 按“←”、“→”方向键切换至“Advanced”界面，如图 3-9 所示。

图 3-9 Advanced 界面



步骤 3 按键盘的上下方向键选择“IPMI BMC Configuration”，按“Enter”。进入“IPMI BMC Configuration”界面，如图 3-10 所示。

图 3-10 IPMI BMC Configuration 界面



步骤 4 按键盘的上下方向键选择 “ BMC Configuration”，按“Enter”。进入 “BMC Config” 界面，如图 3-11 和图 3-12 所示。

图 3-11 BMC Config 界面 1

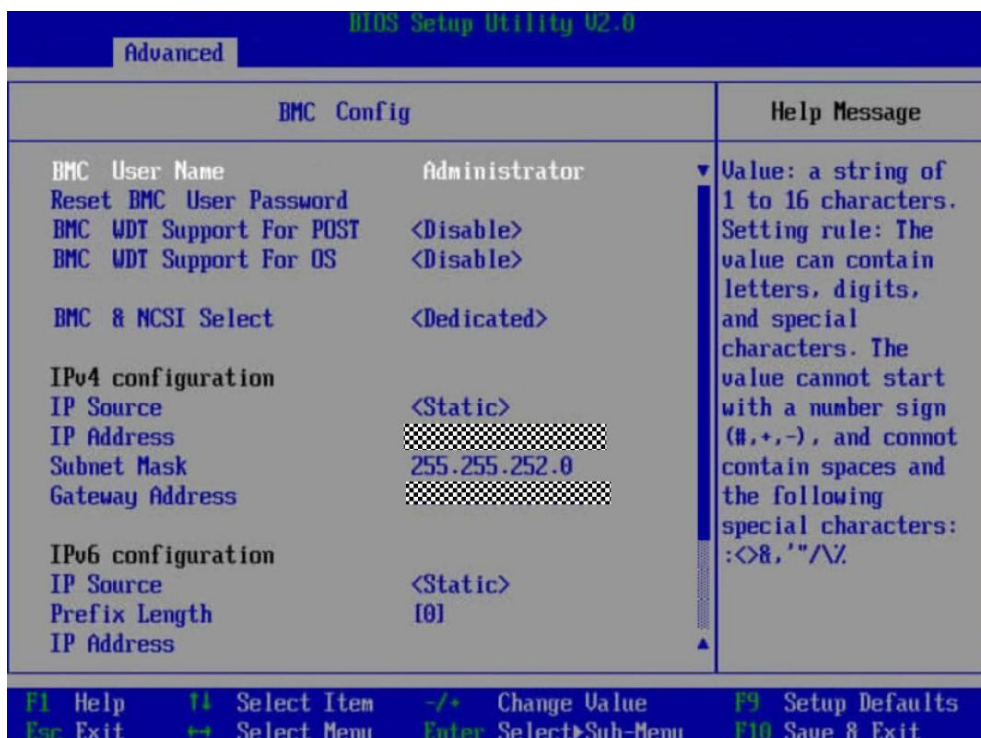
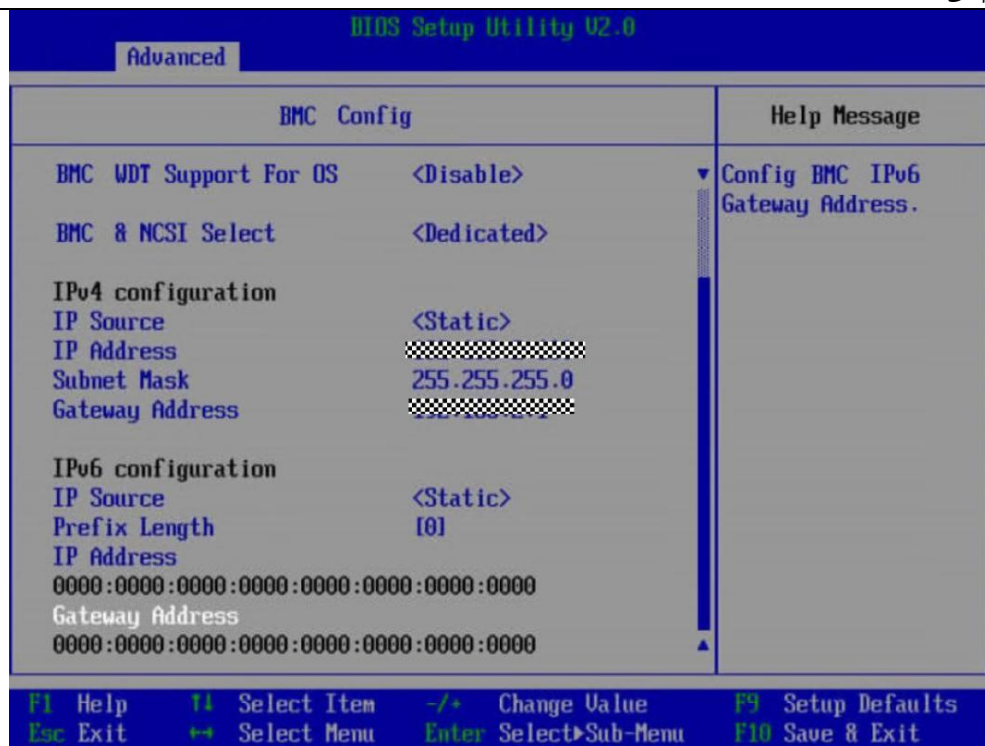


图 3-12 BMC Config 界面 2

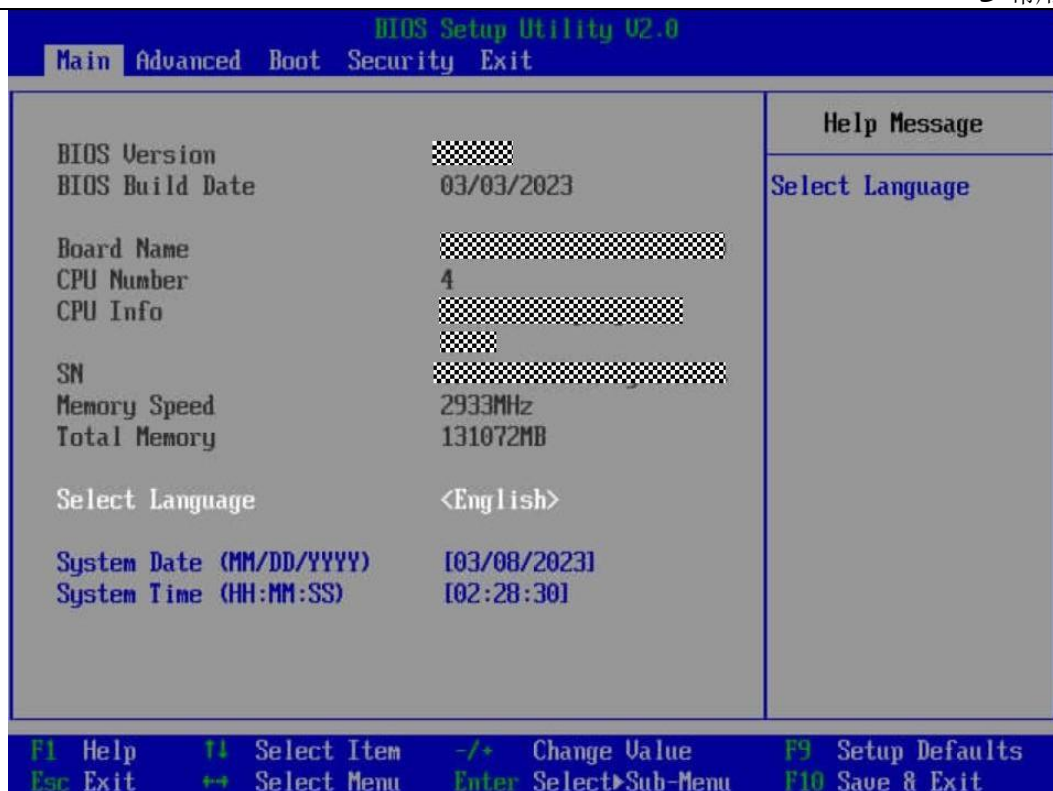


步骤 5 查看所需的 IP 地址信息。

----结束

步骤 1 进入 BIOS 界面，具体操作步骤请参见 3.2 进入 BIOS 界面。

步骤 2 按“←”、“→”方向键切换至“Main”界面，如图 3-13 所示。



步骤 3 选择“Select Language”，按“Enter”。弹出选择语言对话框。

步骤 4 根据需求选择“English”或者“中文”，按“Enter”。步骤 5 设置完成后，按“F10”。

弹出“Save configuration changes and exit?”对话框。步骤 6 选择“Yes”并按“Enter”保存设置。

服务器将自动重启使设置生效。

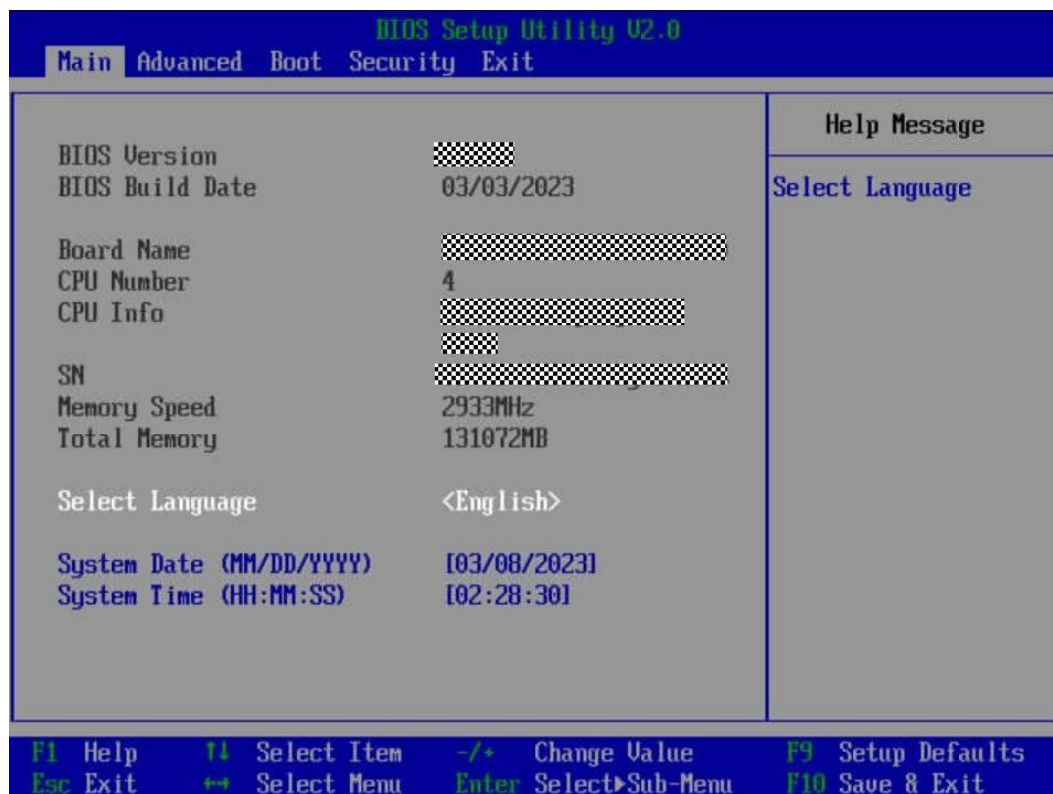
----结束

该任务指导用户通过 BIOS 程序设置 BIOS 系统的日期和时间。

步骤 1 进入 BIOS 界面，具体操作步骤请参见 [3.2 进入 BIOS 界面](#)。

步骤 2 按“←”、“→”方向键切换至“Main”界面，如图 3-14 所示。

图 3-14 Main 界面



步骤 3 设置系统日期和时间。

1. 选择“System Date”设置系统日期。

系统日期的格式为“月/日/年”。按“Tab”或两次“Enter”在月、日、年之间切换，设置完成后，立即生效。可以通过以下方式更改数值：

- 按“+”：数值增加 1。
- 按“-”：数值减小 1。
- 按数字键：选中要更改的数值后按“Enter”，修改完成后，按“Enter”退出修改。

2. 选择“System Time”设置系统时间。

系统时间是 24 小时制，格式是“时:分:秒”。按“Tab”或两次“Enter”在时、分、秒之间切换，设置完成后，立即生效。可以通过以下方式更改数值：

- 按“+”：数值增加 1。
- 按“-”：数值减小 1。
- 按数字键：选中要更改的数值后按“Enter”，修改完成

后，按“Enter”退出修改。

-----结束



说明

- 若使用的是支持 first login 密码功能（即 BIOS 默认无密码，第一次进 Setup 界面时，会提示设置新密码）的 BIOS 版本，必须设置完新密码后才能登录进入 Setup 界面。具体请参见 3.2 进入 BIOS 界面。

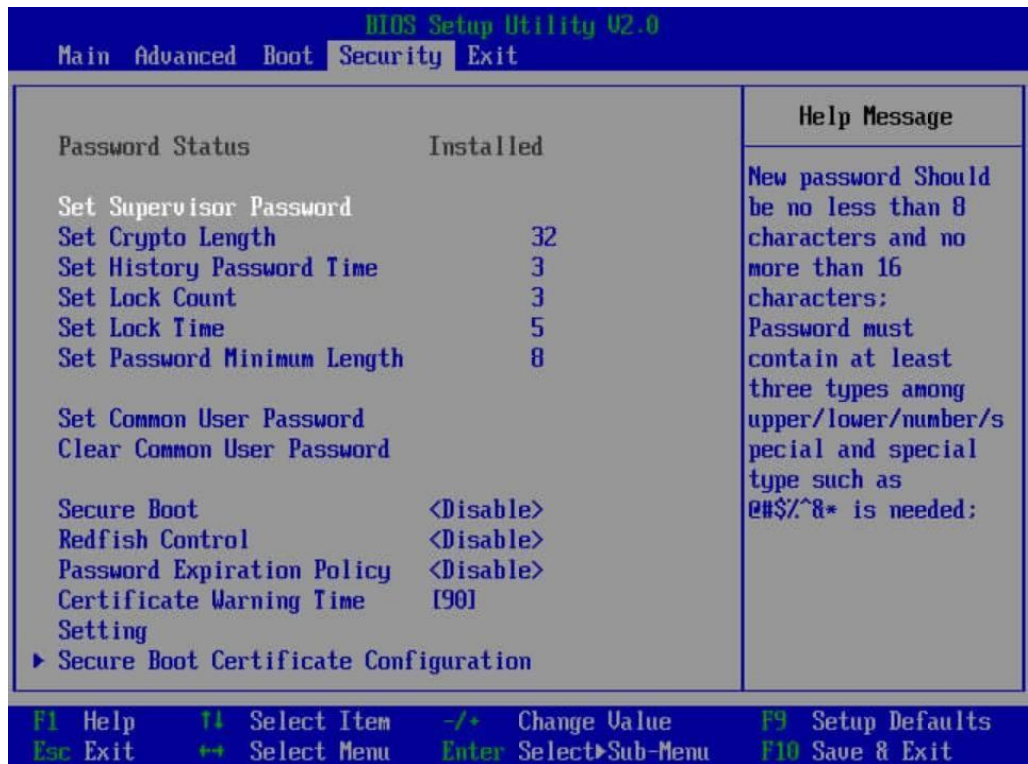
- 如果设置的 BIOS 密码被遗忘，可参考 A.1 如何重置 BIOS 密码来重置密码。

该任务指导用户通过 BIOS 程序对 BIOS 密码进行设置或修改。

步骤 1 进入 BIOS 界面，具体操作步骤请参见 3.2 进入 BIOS 界面。

步骤 2 按“←”、“→”方向键切换至“Security”

界面，如图 3-15 所示。图 3-15 Security 界面 1



说明

不同 BIOS 版本页面会有差异，请以实际为准。

步骤 3 选择 “Set Supervisor Password” 选项，按 “Enter”，可以设置/修改管理员登录密码，设置/修改前需要输入原密码。

说明

- 密码长度必须在 8~16 位之间，至少包含特殊字符（包括空格）、大写字母、小写字母及数字这四种字符中的三种，其中必须包含特殊字符。
- 不能设置最近 3~6 次的历史密码为新密码。
- 若使用的是支持 first login 密码功能（即 BIOS 默认无密码，第一次进 Setup 界面时，会提示设置新密码）的 BIOS 版本，不支持默认密码。

步骤 4 （可选）设置成功后，选择 “Clear Supervisor Password” 可清除已经设置的密码，清除前需要输入当前密码。

说明

若使用的是支持 first login 密码功能（即 BIOS 默认无密码，第一次进 Setup 界面时，会提示设置新密码）的 BIOS 版本，不支持“Clear Supervisor Password”参数。

步骤 5 设置完成后，按 “F10”。

弹出 “Save configuration changes and exit?” 对话框。步骤 6 选择 “Yes” 并按 “Enter” 保存设置。

服务器将自动重启使设置生效

5.1.3. 统一身份认证模块结果

功能介绍： 多层用户体验，提供身份识别 与访问管理服务：

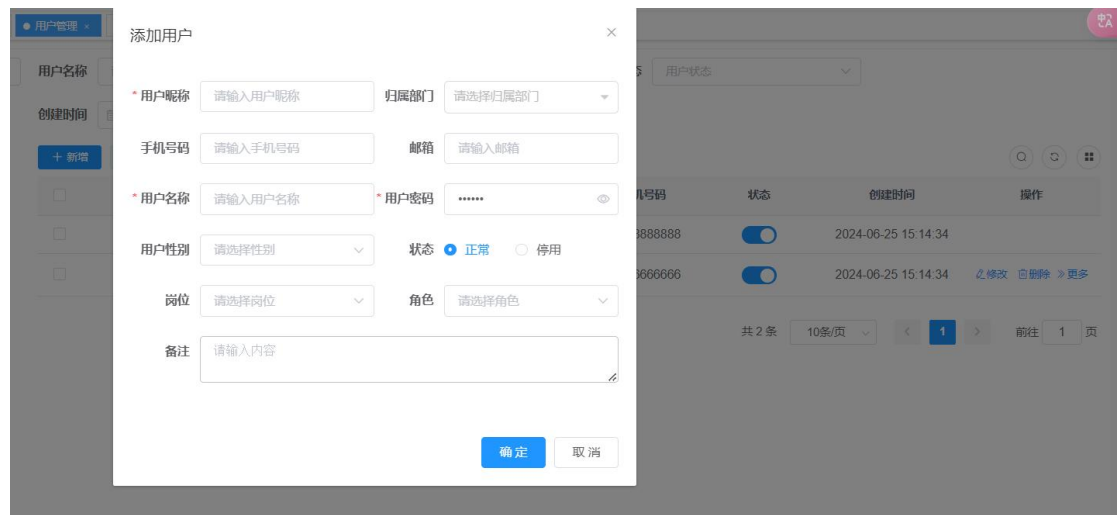
三级架构用户管理，组织/用户组/用户不同权限管理；

用户数据安全隔离；

用户行为审计；

运维监控管理，资源使用率可视化查询；

日志管理，可视化查询；



5.1.4. 资源监控中心模块结果

(4) 整体资源显示：包括服务器资源如 CPU、GPU 使用、内存使用等；

(5) NPU 资源监控可实时监控显示 AI 加速卡使用详情、AI 卡显存使用、功率、温度等；

(6) 运行资源列表显示集群下 npu/gpu 详细情况，提供搜索功能。

资源监控

类型: 节点IP:

序号	节点IP	芯片id	芯片名	类型	集群	AI卡使用率(%)	功率 (W)	温度(°C)	Memory-Usage(MB)	HBM-Usage(MB)
1	10.199.139.164	0	9108	npv	k8s_cluster_default	0	71.3	42	1270/13553	0/32768
2	10.199.139.164	1	9108	npv	k8s_cluster_default	0	67.4	34	2491/15665	0/32768
3	10.199.139.164	2	9108	npv	k8s_cluster_default	0	70.0	35	2165/15665	1/32768
4	10.199.139.164	3	9108	npv	k8s_cluster_default	0	70.1	42	2303/15567	0/32768
5	10.199.139.164	4	9108	npv	k8s_cluster_default	0	72.6	44	1157/13553	0/32768
6	10.199.139.164	5	9108	npv	k8s_cluster_default	0	66.8	36	2709/15665	0/32768
7	10.199.139.164	6	9108	npv	k8s_cluster_default	0	69.4	35	2137/15665	0/32768
8	10.199.139.164	7	9108	npv	k8s_cluster_default	0	67.8	40	2228/15567	0/32768
9	10.199.139.168	0	9108	npv	k8s_cluster_default	0	68.7	41	2377/15038	0/32768
10	10.199.139.168	1	9108	npv	k8s_cluster_default	0	64.3	36	2335/15038	0/32768
11	10.199.139.168	2	9108	npv	k8s_cluster_default	0	65.2	36	2360/15038	0/32768

5.1.5. 算力监控中心模块结果

基础信息：集群下服务器 AI 卡 npv 和 gpu 算力详情；

算力资源信息：集群下各节点服务器资源组下各 npv 及 gpu 算力使用情况，提供搜索功能。



5.1.6. 调度中心结果

调度中心：用户资源调度任务部署配置页面，核心参数包括：集群名，AI

框架类型，镜像构建方式、镜像选择、命名空间、数据路径等。

创建部署

* 任务名称	<input type="text"/>	* 集群	k8s_cluster_default
* AI框架	Mindspore	* 部署类型	单机
* Worker个数	<input type="text" value="1"/>	* 镜像类型	预置镜像
* 镜像选择	10.199.139.162:8080/x86/ascendhub.f		

资源配置

* 命名空间	<input type="text"/>	* 资源组	npu
CPU	<input type="text" value="1"/>	CPU限制	<input type="text" value="1"/>
内存	<input type="text" value="1024"/>	内存限制	<input type="text" value="1024"/>
* 使用AI卡数	8卡	GPU显存	<input type="range" value="0"/>
GPU限制	<input type="text" value="10"/>	数据挂载	文件管理
数据路径	/workspace/dataset		
		代码路径	/workspace/code

5.1.7. 调度任务管理结果

调度任务列表：列表展示调度部署核心参数信息，用户可对任务进行启动、停止、删除、以及详情操作。

启动：调度任务启动成功后，会立即执行配置的模型训练任务。

停止：强制停止正在运行的模型训练任务。

删除：对初始化或已停止的调度任务进行删除。

序号	任务名称	任务状态	集群名称	类型	节点IP	创建时间	操作
1	test8899	已停止	k8s_cluster_default	gpu	10.199.139.164	2024-09-04 09:32:50	启动 停止 删除 详情
2	test0904	已停止	k8s_cluster_default	gpu	10.199.139.168	2024-09-04 09:22:04	启动 停止 删除 详情
3	asdfas	已停止	k8s_cluster_default	gpu	10.199.139.164	2024-09-03 19:23:02	启动 停止 删除 详情
4	asfda	已停止	k8s_cluster_default	gpu		2024-09-03 19:18:19	启动 停止 删除 详情
5	kikjk8	已停止	k8s_cluster_default	gpu		2024-09-03 19:15:19	启动 停止 删除 详情

共 24 条 3 页/页 < 1 2 3 4 5 > 前往 1 页

5.1.8. 调度任务详情结果

任务详情：该功能主要实时监控 模型执行结果信息，包含任务状态（运行中、已结束等状态），模型运行会自动分配节点服务器上（所属节点 ip）以及运行时间。

训练日志：实时获取模型训练日志进行页面显示。

下载日志：可一键下载服务器训练日志到本地。

任务详情 ×

任务编号: 1831143348473466880
任务名称: test8899

任务状态: 已停止
运行时间: 43s

创建时间: 2024-09-04 09:32:50
所属节点IP: 10.199.139.164

训练日志
下载完整训练日志
下载YAML任务配置
下载k8s执行日志

```

2024-09-04 09:35:06,502:INFO:--> rank_id: 0
2024-09-04 09:35:06,502:INFO:--> save_ckpt_dir: /home/HwHiAiUser/samples/Resnet50_Cifar_for_MindSpore/scripts/outputs/resnet50/cifar10/ckpt
2024-09-04 09:35:06,502:INFO:--> log_dir: /home/HwHiAiUser/samples/Resnet50_Cifar_for_MindSpore/scripts/outputs/resnet50/cifar10/log
2024-09-04 09:35:06,502:INFO:--> logger: <LOGGER resnet (NOTSET)>
2024-09-04 09:35:06,502:INFO:
2024-09-04 09:37:33,240:INFO:epoch: [1/3] loss: 1.354008, epoch time: 146.737 s, per step time: 627.082 ms
2024-09-04 09:37:40,108:INFO:epoch: [2/3] loss: 1.603233, epoch time: 6.868 s, per step time: 29.348 ms
2024-09-04 09:37:47,008:INFO:epoch: [3/3] loss: 0.882593, epoch time: 6.899 s, per step time: 29.485 ms
2024-09-04 09:37:49,025:INFO:if run eval and enable_cache Remember to shut down the cache server via "cache_admin --sto

```

关闭

6. 故障定位

6.1.1. 故障定位步骤

(1) 查看任务状态和基础信息

- 在调度任务管理模块中，查看该任务的详细信息，包括任务当前状态（如“失败”）、分配的服务器节点、任务运行时长等。
- 任务状态监控模块可以显示是否是因为资源不足（如 GPU/NPU 内存不足、超时等）导致的任务中止。
- 核查调度中心中的节点健康状态，确认是否因为节点故障导致任务失败。

(2) 分析训练日志

- 通过调度任务详情模块，实时查看模型训练日志。
- 如果任务失败，可以一键下载完整的训练日志，并通过日志中错误栈信息确认是否有代码层面的问题（如模型代码出错、依赖包错误等）。
- 通过分析日志中的错误提示，可以快速确定问题发生的代码行数和上下文，了解模型训练在什么阶段出现了问题（如数据加载、模型构建、梯度更新等）。

(3) 查看资源监控信息

- 通过平台的资源监控中心模块，查看该任务运行期间的资源消耗情况，包括 GPU/NPU 的使用率、显存、功耗、温度等。
- 确认是否因为硬件资源过载（如显存不足）导致训练失败。

- 监控图表可以帮助确认问题是否与特定的硬件资源或时段相关，排除资源使用异常导致的问题。

(4) 检查调度配置和依赖环境

- 在调度中心配置页面，检查任务的调度参数是否正确，如是否选择了正确的 AI 框架、镜像版本、数据路径等。
- 确认该任务的镜像是否存在问题（如镜像缺少必要的库、依赖项版本不匹配等）。
- 检查集群中是否有其他任务与该任务冲突，导致资源竞争或调度错误。

(5) 多任务并发问题

- 如果集群中同时运行了多个任务，可以检查是否存在多任务并发导致的系统瓶颈或网络瓶颈。
- 在调度任务管理模块中，检查其他正在运行的任务状态，确认是否有其他任务过度占用资源，影响了当前任务的执行。

6.1.2. 故障定位案例

(1) 案例一：大模型训练报错：

根据 web 界面调度任务详情中对应的 k8s 输出日志可看到，训练过程中模型报错，服务器上查看那具体日志如下：

```

File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/nn/cell.py", line 701, in __call__
    output = self._run_construct(args, kwargs)
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/nn/cell.py", line 482, in _run_construct
    output = self.construct(*cast_inputs, **kwargs)
File "/root/mindformers-dev/scripts/mf_parallel0/mindformers/core/loss/loss.py", line 362, in construct
    log_softmax, one_hot_label = self._log_softmax(logits, label)
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/nn/cell.py", line 705, in __call__
    raise err
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/nn/cell.py", line 702, in __call__
    _pynative_executor.end_graph(self, output, *args, **kwargs)
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/common/api.py", line 1215, in end_graph
    self._executor.end_graph(obj, output, *args, *(kwargs.values()))
RuntimeError: Output_idx 0 of node ValueNode<None> None node:0xffffc0453b020 output addr is not exist.

```

```

[WARNING] PARALLEL(744385,ffff71b6e0,python):2024-04-02-07:34:13.387.903 [mindspore/ccsrc/frontend/parallel/pass/pass_utils.cc:119] ExtractBackendMatMul backward matmul dx dx map sizes:9
[ERROR] GE_ADP(744385,ffff71b6e0,python):2024-04-02-07:35:18.799.520 [mindspore/ccsrc/transform/graph_ir/graph_runner.cc:441] CompileGraph Call GE CompileGraph Failed, ret is: 1343225857
2024-04-02 07:35:21.066 - mindformers[mindformers/tools/cloud_adapter/cloud_monitor.py:43] - 30304 - Traceback (most recent call last):
File "/root/mindformers-dev/scripts/mf_parallel0/mindformers/tools/cloud_adapter/cloud_monitor.py", line 34, in wrapper
    result = run_func(*args, **kwargs)
File "/root/mindformers-dev/scripts/mf_parallel0/run_mindformer.py", line 39, in main
    trainer.train()
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/_checkparam.py", line 1313, in wrapper
    return func(*args, **kwargs)
File "/root/mindformers-dev/scripts/mf_parallel0/mindformers/trainer/trainer.py", line 411, in train
    self.trainer.train()
File "/root/mindformers-dev/scripts/mf_parallel0/mindformers/trainer/causal_language_modeling/causal_language_modeling.py", line 113, in train
    self.training_process()
File "/root/mindformers-dev/scripts/mf_parallel0/mindformers/trainer/base_trainer.py", line 784, in training_process
    model.train(config.runner_config.epochs, dataset,
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/train/model.py", line 1068, in train
    self._train(epoch,
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/train/model.py", line 114, in wrapper
    func(self, *args, **kwargs)
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/train/model.py", line 623, in _train
    self._train_dataset_sink_process(epoch, train_dataset, list_callback,
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/train/model.py", line 708, in _train_dataset_sink_process
    outputs = train_network(*inputs)
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/nn/cell.py", line 680, in __call__
    out = self.compile_and_run(*args, **kwargs)
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/nn/cell.py", line 1020, in compile_and_run
    self.compile(*args, **kwargs)
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/nn/cell.py", line 997, in compile
    _cell_graph_executor.compile(self, phase=self.phase,
File "/root/miniconda3/envs/mindspore2.2.11_py39/lib/python3.9/site-packages/mindspore/common/api.py", line 1547, in compile
    result = self._graph_executor.compile(infbh=args, kwargs=kwargs, phase=self._use_vm_mode())
RuntimeError: Compile graph kernel_graph_1 failed.

```

解决方案：报错显示 mindspore 算子不兼容导致报错，更换 mindspore 版本为 2.2.12 问题得到解决，重新在服务器管理平台配置调度任务再次启动训练，能正常训练，问题得到解决。

通过 Prometheus Metrics 接口以及 k8s 日志输出可查看模型运行过程中出现的故障得到快速定位。

```

...
# HELP machine_npu_nums Amount of npu installed on the machine.
# TYPE machine_npu_nums gauge
machine_npu_nums 8
# HELP npu_chip_info_error_code the npu error code
# TYPE npu_chip_info_error_code gauge
npu_chip_info_error_code{id="0"} 0 1613993498553
npu_chip_info_error_code{id="1"} 0 1613993498588
npu_chip_info_error_code{id="2"} 0 1613993498615
npu_chip_info_error_code{id="3"} 0 1613993498645

```

```

npu_chip_info_error_code{id="4"} 0 1613993498676
npu_chip_info_error_code{id="5"} 0 1613993498685
npu_chip_info_error_code{id="6"} 0 1613993498715
npu_chip_info_error_code{id="7"} 0 1613993498742
# HELP npu_chip_info_hbm_total_memory the npu hbm total memory
# TYPE npu_chip_info_hbm_total_memory gauge
npu_chip_info_hbm_total_memory{id="0"} 32255 1613993498553
npu_chip_info_hbm_total_memory{id="1"} 32255 1613993498588
npu_chip_info_hbm_total_memory{id="2"} 32255 1613993498615
...

```

(2) 案例 2 机器 npu 卡通信问题

主机器在拉起训练后无法找到 loss 节点, 在请求时间达到最大时报错 `getting socket timeout`。

```

return self._exec_pip(obj, *args, phase=phase_real)
File "/root/.miniconda3/envs/mindspore2.2_py39/lib/python3.9/site-packages/mindspore/common/api.py", line 121, in wrapper
    results = fn(*arg, **kwargs)
File "/root/.miniconda3/envs/mindspore2.2_py39/lib/python3.9/site-packages/mindspore/common/api.py", line 1608, in _exec_pip
    return self._graph_executor(args, phase)
RuntimeError: Exec graph failed

Ascend Error Message:
E10006: Getting socket times out. Reason: 1. The remote does not initiate a connect request. some NPUs in the cluster are abnormal. 2. The remote does not initiate a connect request because the collective communication operator is started too late or is not started by some NPU in the cluster. 3. The communication link is disconnected. (For example, the IP addresses are not on the same network segment or the TLS configurations are inconsistent.)
Solution: 1. Check the rank service processes with other errors or no errors in the cluster. 2. If this error is reported for all NPUs, check whether the time difference between the earliest and latest errors is greater than the connect timeout interval (120s by default). If so, adjust the timeout interval by using the HCCL_CONNECT_TIMEOUT environment variable. 3. Check the connectivity of the communication link between nodes. (For details, see the TLS command and HCCL connectivity check examples.). For details: https://www.hiascend.com/document
Traceback (most recent call last):
  Call ops_kernel_info_store LoadTask fail[FUNC:Distribute][FILE:hccl_task_info.cc][LINE:323]
  Call ops_kernel_info_store unloadTask fail[FUNC:Release][FILE:hccl_task_info.cc][LINE:605]
  GraphManager RunGraphWithStreamAsync failed, session id = 0, graph id = 2, stream = 0x50ff1e30. [FUNC:RunGraphWithStreamAsync][FILE:inner_session.cc][LINE:517]
  [Run][Graph]Run graph with stream asyn failed, error code = 1343225860, session id = 0, graph id = 2, stream = 0x50ff1e30. [FUNC:RunGraphWithStreamAsync][FILE:ge_api.cc][LINE:826]
(Please search "Ascend Error Message" at https://www.mindspore.cn for error code description)

C++ Call Stack: (For framework developers)
mindspore/ccsrc/plugin/device/ascend/hal/hardware/ge_graph_executor.cc:983 RunGraphRefMode

[WARNING] MD(34874, fffc96696920, python):2023-11-10-08:03:39.692.852 [mindspore/ccsrc/minddata/dataset/engine/datasetops/data_queue_op.cc:163] -DataQueueOp]
preprocess_batch: 100;
batch_queue: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0;
push_start_time -> push_end_time
2023-11-10-07:19:36.493.638 -> 2023-11-10-07:19:36.494.078
2023-11-10-07:19:36.495.354 -> 2023-11-10-07:19:36.495.725
2023-11-10-07:19:36.497.145 -> 2023-11-10-07:19:36.497.518
2023-11-10-07:19:36.498.927 -> 2023-11-10-07:19:36.499.395
2023-11-10-07:19:36.500.645 -> 2023-11-10-07:19:36.501.018
2023-11-10-07:19:36.502.359 -> 2023-11-10-07:19:36.502.735

```

两台机器间 npu 在 hccl 工具验证下通信正常。

主副节点间在训练拉起后均显示在创建通信组时无法找到主节点 rank

0 device。

```
92 [WARNING] DISTRIBUTED(252506, ffc73d46c90, python):2023-11-10-07:23:17.138.713 [mindspore/ccsrc/distributed/collective/collective_manager]
  * :222] CreateCommunicationGroup] Start to create communication group: 2-6222080562290339684 [const vector][2, 10]
  * :222] CreateCommunicationGroup] This rank: 0 is not in the group ranks: [const vector][2, 10]. This may cause some exception when
  * initializing the group.
93 [WARNING] DISTRIBUTED(252506, ffc73d46c90, python):2023-11-10-07:23:17.275.470 [mindspore/ccsrc/distributed/collective/collective_manager]
  * :220] CreateCommunicationGroup] Start to create communication group: 4-6301172352641561019 [const vector][0, 1, 2, 3]
94 [WARNING] DISTRIBUTED(252506, ffc73d46c90, python):2023-11-10-07:23:17.275.568 [mindspore/ccsrc/distributed/collective/collective_manager]
  * :222] CreateCommunicationGroup] This rank: 0 is not in the group ranks: [const vector][2, 10]. This may cause some exception when
  * initializing the group.
95 [WARNING] DISTRIBUTED(252506, ffc73d46c90, python):2023-11-10-07:23:17.363.746 [mindspore/ccsrc/distributed/collective/collective_manager]
  * :220] CreateCommunicationGroup] Start to create communication group: 4-6301172352641561019 [const vector][0, 1, 2, 3]
96 [WARNING] DISTRIBUTED(252506, ffc73d46c90, python):2023-11-10-07:23:19.674.379 [mindspore/ccsrc/distributed/collective/collective_manager]
  * :220] CreateCommunicationGroup] Start to create communication group: 2-511848487187618470 [const vector][2, 6]
97 [WARNING] DISTRIBUTED(252506, ffc73d46c90, python):2023-11-10-07:23:19.674.462 [mindspore/ccsrc/distributed/collective/collective_manager]
  * :222] CreateCommunicationGroup] This rank: 0 is not in the group ranks: [const vector][2, 6]. This may cause some exception when initiali
  * zing the group.
98 [WARNING] DISTRIBUTED(252506, ffc73d46c90, python):2023-11-10-07:23:19.738.108 [mindspore/ccsrc/distributed/collective/collective_manager]
  * :220] CreateCommunicationGroup] Start to create communication group: 8-12158923167718251343 [const vector][0, 1, 2, 3, 4, 5, 6, 7]
99 [WARNING] PARALLEL(252506, ffc73d46c90, python):2023-11-10-07:23:21.272.485 [mindspore/ccsrc/frontend/parallel/step_parallel.cc:2283]
  * GetSensLossPairs] Can not find the loss cnode
940 -
941 \
942 |
943 /
944 -
945 \
946 |
947 /
948 -
949 \
950 |
951 /
952 -
953 \
954 |
955 /
956 -
957 [ERROR] GE_ADPT(252506, ffc73d46c90, python):2023-11-10-08:04:49.110.642 [mindspore/ccsrc/transform/graph_ir/graph_runner.cc:352]
  * RunGraphWithStreamAsync] Call GE RunGraphWithStreamAsync Failed, ret is: 4294967295
  * RunGraphWithStreamAsync] GE RunGraphWithStreamAsync Failed, ret is: 4294967295 - ERROR - Traceback (most recent call last):
  * File "/usr/local/lib/python3.8/site-packages/mindformers/tools/cloud_adapter/cloud_monitor.py:42", in monitor
  * File "/usr/local/lib/python3.8/site-packages/mindformers/tools/cloud_adapter/cloud_monitor.py:42", in monitor
```

```
951 [WARNING] PARALLEL(35084, ffbcf9f6920, python):2023-11-10-07:24:19.427.677 [mindspore/ccsrc/frontend/parallel/step_parallel.cc:463]
  * Redistribution] The index is out of range, the index is 1, the vector size is 1next node is #157 mindspore_nn_wrap_cell_wrapper_VirtualData
  * setCell_construct_182:CNode_183{[0]: ValueNode<FuncGraph> 227_mindformers_core_loss_loss_Softmax_construct_184, [1]: CNode_183, [2]:
  * CNode_183}
952 [WARNING] PARALLEL(35084, ffbcf9f6920, python):2023-11-10-07:24:19.494.311 [mindspore/ccsrc/frontend/parallel/step_parallel.cc:463]
  * Redistribution] The index is out of range, the index is 1, the vector size is 1next node is #157 mindspore_nn_wrap_cell_wrapper_VirtualData
  * setCell_construct_182:CNode_183{[0]: ValueNode<FuncGraph> 227_mindformers_core_loss_loss_Softmax_construct_184, [1]: CNode_183, [2]:
  * CNode_183}
953 [WARNING] PARALLEL(35084, ffbcf9f6920, python):2023-11-10-07:24:19.560.636 [mindspore/ccsrc/frontend/parallel/step_parallel.cc:463]
  * Redistribution] The index is out of range, the index is 1, the vector size is 1next node is #157 mindspore_nn_wrap_cell_wrapper_VirtualData
  * setCell_construct_182:CNode_183{[0]: ValueNode<FuncGraph> 227_mindformers_core_loss_loss_Softmax_construct_184, [1]: CNode_183, [2]:
  * CNode_183}
954 [WARNING] PARALLEL(35084, ffbcf9f6920, python):2023-11-10-07:24:19.627.509 [mindspore/ccsrc/frontend/parallel/step_parallel.cc:463]
  * Redistribution] The index is out of range, the index is 1, the vector size is 1next node is #157 mindspore_nn_wrap_cell_wrapper_VirtualData
  * setCell_construct_182:CNode_183{[0]: ValueNode<FuncGraph> 227_mindformers_core_loss_loss_Softmax_construct_184, [1]: CNode_183, [2]:
  * CNode_183}
955 [WARNING] PARALLEL(35084, ffbcf9f6920, python):2023-11-10-07:24:19.693.761 [mindspore/ccsrc/frontend/parallel/step_parallel.cc:463]
  * Redistribution] The index is out of range, the index is 1, the vector size is 1next node is #157 mindspore_nn_wrap_cell_wrapper_VirtualData
  * setCell_construct_182:CNode_183{[0]: ValueNode<FuncGraph> 227_mindformers_core_loss_loss_Softmax_construct_184, [1]: CNode_183, [2]:
  * CNode_183}
956 [WARNING] PARALLEL(35084, ffbcf9f6920, python):2023-11-10-07:24:20.706.611 [mindspore/ccsrc/frontend/parallel/step_parallel.cc:463]
  * Redistribution] The index is out of range, the index is 1, the vector size is 1next node is #157 mindspore_nn_wrap_cell_wrapper_VirtualData
  * setCell_construct_182:CNode_183{[0]: ValueNode<FuncGraph> 227_mindformers_core_loss_loss_Softmax_construct_184, [1]: CNode_183, [2]:
  * CNode_183}
957 [WARNING] DISTRIBUTED(35084, ffbcf9f6920, python):2023-11-10-07:35:25.601.889 [mindspore/ccsrc/distributed/collective/collective_manager.cc]
  * :220] CreateCommunicationGroup] Start to create communication group: 2-16682069373226611640_parallel_optimizer [const vector][11, 15]
958 [WARNING] DISTRIBUTED(35084, ffbcf9f6920, python):2023-11-10-07:35:25.601.989 [mindspore/ccsrc/distributed/collective/collective_manager.cc]
  * :222] CreateCommunicationGroup] This rank: 0 is not in the group ranks: [const vector][11, 15]. This may cause some exception when
  * initializing the group.
959 -
960 \
961 |
962 /
963 -
964 \
965 |
966 /
967 -
968 \
969 |
970 /
971 -
972 \
973 |
974 /
```

解决方案：

使用 `hccn_tool -i 1 -ping -g adress xxx.xxx.xxx.xxx pkt 8164` 命令测试最大发包量，得出 mtu 设置错误，重新设置服务器集群的交换机 mtu 参数解决通信问题。