ICS 33.060.01
CCS M30

# 团　体　标　准

T/SCSTXXH 3—2025

# 四川省智算中心计算资源调度实施规范

**Implementation Specification for Computing Resource Scheduling in Intelligent Computing Center of Sichuan Province**

2025-10-31 发布　　2025-10-31 实施

四川省通信学会　发　布

# 目　　次

# 前　　言

本文件按照 GB/T 1.1—2020《标准化工作导则 第 1 部分：标准化文件的结构和起草规则》、四川省通信学会 2023 年发布的《四川省通信学会团体标准管理办法（修订稿）》进行起草。

智算中心作为未来计算力发展的重要基础设施，为上层应用及平台提供强大的计算能力、存储能力和网络能力。政府与企业均关注智算中心资源池的优化调度，以提升计算资源的利用率，降本增效。

随着人工智能技术的快速发展，智算中心作为支撑 AI 应用的基础设施，其计算资源的高效调度和优化配置成为关键。本实施规范旨在从总体层面指导智算中心如何实施计算资源的调度，涵盖资源调度目标与思路、调度方法与策略、关键技术与应用、实施方案及步骤等内容，以推动智算中心资源的高效利用和 AI 应用的快速发展。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由四川省通信学会负责归口管理。

本文件起草单位：中国移动通信集团四川有限公司、中通服咨询设计研究院有限公司、成都理工大学、四川中移通信技术工程有限公司。

本文件主要起草人员：张高毅、苟浩淞、代泽均、姚光乐、刘勇、梅洲、王洪辉、陈才华、贾勇、赵仕波、彭鹏、庞璐、李瑞佳、周馨、王宇。

# 四川省智算中心计算资源调度实施规范

## 1 范围

本文件规定了智算中心计算资源调度的思路、方法、关键技术及实施方案等内容。

本文件适用于指导智算中心及相关项目的计算资源调度规划、实施及优化工作。

## 2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 32399 《信息技术云计算参考架构》

GB/T 37737 《信息技术云计算分布式块存储系统总体技术要求》

GB/T 32400 《信息技术云计算概览与词汇》

GB/T 35301 《信息技术云计算平台即服务（PaaS）参考架构》GB/T 36327《信息技术云计算平台即服务（PaaS）应用程序管理要求》

GB/T 37739 《信息技术云计算平台即服务部署要求》

GB/T 36623 《信息技术云计算文件服务应用接口》

GB/T 35293 《信息技术云计算虚拟机管理通用要求》

GB/T 37732 《信息技术云计算云存储系统服务接口功能》

GB/T 37734 《信息技术云计算云服务采购指南》

GB/T 36325 《信息技术云计算云服务级别协议基本要求》

GB/T 37735 《信息技术云计算云服务计量指标》

GB/T 37741 《信息技术云计算云服务交付要求》

GB/T 36326 《信息技术云计算云服务运营通用要求》

GB/T 37738 《信息技术云计算云服务质量评价指标》

GB/T 40690 《信息技术云计算云际计算参考架构》

GB/T 37740 《信息技术云计算云平台间应用和数据迁移指南》

GB/T 37736 《信息技术云 云资源监控通用要求》

GB/T 34982 《云计算数据中心基本要求》

GB/T 34080 《基于云计算的电子政务公共平台安全规范》

GB/T 34079 《基于云计算的电子政务公共平台服务规范》

GB/T 34077 《基于云计算的电子政务公共平台管理规范》

GB/T 33780 《基于云计算的电子政务公共平台技术规范》
GB/T 34078 《基于云计算的电子政务公共平台总体规范》
GB/T 35279 《信息安全技术云计算安全参考架构》
GB/T 34942 《信息安全技术云计算服务安全能力评估方法》
GB/T 31168 《信息安全技术云计算服务安全能力要求》
GB/T 31167 《信息安全技术云计算服务安全指南》
GB/T 37972 《信息安全技术云计算服务运行监管框架》
GB/T 38249 《信息安全技术政府网站云计算服务安全指南》

## 3 术语、定义和缩略语

### 3.1 术语和定义

下列术语和定义适用于本文件。

3.1.1

**智算中心 Intelligent Computing Center**

智算中心是指集成了高性能计算、人工智能计算等多元化计算能力的数据中心，旨在提供高效、灵活、可扩展的计算资源，以支持复杂的数据处理、模拟仿真、深度学习等应用需求。

3.1.2

**计算资源调度 Computing Resource Scheduling**

计算资源调度是指在智算中心内部，根据业务需求、资源状态、调度策略等因素，动态地将计算资源（如 CPU、GPU、NPU、内存、存储等）分配给不同的任务或应用的过程。调度算法负责确定资源的分配方式和时机，以实现资源的高效利用和业务性能的优化。

3.1.3

**调度算法 Scheduling Algorithm**

调度算法是指用于计算资源调度的数学或启发式方法，通过评估任务的优先级、资源的需求和可用性等因素，确定任务的执行顺序和资源分配方案。常见的调度算法包括先来先服务（FCFS）、最短作业优先（SJF）、优先级调度、时间片轮转（Round Robin）等，以及针对特定应用场景的自定义算法。

3.1.4

**虚拟化技术 Virtualization Technology**

虚拟化技术是一种将物理硬件资源（如计算、存储、网络等）抽象为虚拟资源的技术，使得多个虚拟机或容器可以在同一物理硬件上并行运行，且彼此之间相互隔离。虚拟化技术提高了资源的利用率和灵活性，使得计算资源可以根据业务需求进行动态调整和优化。

3.1.5

**资源池 Resource Pool**

资源池是指将计算、存储、网络等资源按照一定规则进行汇聚和抽象，形成可管理、可调度的逻辑资源集合。资源池支持资源的动态分配和回收，是实现计算资源调度的基础。

3.1.6

**云管理平台 Cloud Management Platform**

云管理平台是指用于管理云环境中的计算、存储、网络等资源，提供资源监控、调度、配置、安全等功能的管理系统。云管理平台支持多用户、多资源池的管理，以及资源的自动化部署和运维。

3.1.7

**业务平台 Service Platform**

业务平台是指运行于智算中心之上，用于提供特定业务功能或服务的应用系统或平台。业务平台可以根据业务需求，动态申请和释放计算资源，以实现业务的快速部署和弹性扩展。

3.1.8

**资源利用率 Resource Utilization**

资源利用率是指计算资源在实际使用中被有效利用的程度，通常以资源的使用率或负载率来衡量。提高资源利用率可以降低运营成本，提高智算中心的整体性能。

3.1.9

**负载均衡 Load Balancing**

负载均衡是指将任务或流量分散到多个计算资源上，以实现资源的均衡利用和性能的优化。负载均衡技术可以提高系统的可用性和容错性，减少单点故障的风险。

3.1.10

**安全策略 Security Policy**

安全策略是指用于保护智算中心计算资源免受未经授权访问、恶意攻击、数据泄露等威胁的一组规则和措施。安全策略包括访问控制、身份认证、数据加密、安全审计等方面。

## 3.2 缩略语

下列缩略语适用于本文件。

AI：人工智能（Artificial Intelligence）

CPU：中央处理器（Central Processing Unit）

GPU：图形处理器（Graphics Processing Unit）

NPU：神经网络处理器（Neural Network Processing Unit）

FCFS：先来先服务（First Come First Served）

FCoE：以太网光纤通道（Fibre Channel over Ethernet）

HA：高可用（High Availability）

IaaS：基础设施即服务（Infrastructure as a Service）

IP：互联网协议（Internet Protocol）

JVM：虚拟机（Java Virtual Machine，Java）

LB：负载均衡（Load Balance）

NFS：网络文件系统（Network File System）

NTP：网络时间协议（Network Time Protocol）

OS：操作系统（Operating System）

PaaS：平台即服务（Platform as a Service）

QoS：服务质量（Quality of Service）

RAM：随机存取存储器（Random Access Memory）

SaaS：软件即服务（Software as a Service）

SDN：软件定义网络（Software Defined Networking）

SJF：最短作业优先（Shortest Job First）

SLA：服务等级协议（Service Level Agreement）

SSD：固态硬盘（Solid State Drive）

TLS：传输层安全性协议（Transport Layer Security）

VPC：虚拟私有云（Virtual Private Cloud）

VM：虚拟机（Virtual Machine）

vCPU：虚拟中央处理器（Virtual CPU）

## 4 调度目标与思路

### 4.1 调度目标

智算中心计算资源调度的总体目标是实现计算资源的优化配置和高效利用，提升智算中心的整体性能和计算能力，满足多样化的业务需求，促进业务的快速发展和创新。

#### 4.1.1 提升资源利用率

a） 提高计算资源使用率

通过合理的资源调度，确保 CPU、GPU、NPU 等计算资源在不同业务场景下的高效使用，重点提升 GPU/NPU 在 AI 计算任务中的计算单元活跃度，避免资源浪费和闲置。

b） 优化负载分配

根据业务需求和资源使用情况，动态调整负载分配，确保资源的均衡利用，提高整体系统的性能。

#### 4.1.2 确保业务连续性

a） 故障恢复与容灾能力

建立故障恢复机制和容灾备份策略，确保在资源故障或灾难情况下，业务能够迅速恢复并继续运行。

b） 高可用性与弹性伸缩

实现资源的高可用性，确保业务在面临高峰或突发事件时，能够弹性地扩展或收缩资源，满足业务需求。

#### 4.1.3 降低运营成本

a） 资源池化与共享

通过资源池化和共享，降低硬件设备的重复购置和运营成本，提高资源的复用率。

b） 节能减排与绿色计算

优化资源调度，降低能源消耗和排放，推动智算中心向绿色、低碳方向发展。

**4.1.4** 提升服务质量

a） 服务响应速度

通过快速、灵活的资源调度，提高业务响应速度，满足用户对高效服务的需求。

b） 服务质量保障

建立服务质量监控和保障机制，确保业务在资源调度过程中的稳定性和可靠性。

**4.1.5** 促进业务创新与发展

a） 资源灵活配置

支持业务场景的多样化需求，提供灵活的资源配置选项，促进业务的创新和发展。

b） 智能化调度与预测

引入智能化调度算法和预测模型，提高资源调度的精准度和效率，为业务的快速发展提供有力支持。

**4.1.6** 增强安全性与合规性

a） 资源访问控制

建立严格的资源访问控制机制，确保只有合法用户或系统能够访问和使用资源。

b） 数据加密与保护

对敏感数据进行加密处理，保护数据的机密性和完整性，防止数据泄露和非法访问。

c） 合规性要求

遵守相关法律法规和行业标准，确保资源调度过程的合规性和安全性。

## 4.2 调度思路

**4.2.1** 资源池化

将智算中心的计算资源抽象为虚拟资源池，实现资源的统一管理和灵活调度。通过资源池化，可以简化资源管理复杂度，提高资源调度的灵活性和效率。

**4.2.2** 动态监测

建立实时、全面的资源监测体系，对智算中心的计算资源使用情况进行持续监测和分析。通过动态监测，可以及时发现资源瓶颈和潜在风险，为调度策略的制定提供数据支持。

**4.2.3** 智能调度

采用先进的调度算法和模型，根据业务需求、资源状况、服务质量等因素，动态调整计算资源的分配。智能调度算法应能够综合考虑多个维度，实现资源的优化配置和高效利用。

**4.2.4** 优先级管理

根据业务的优先级和重要性，对计算资源进行分级管理。在资源紧张时，优先保障关键业务和关键服务的资源需求，确保业务的连续性和稳定性。

**4.2.5** 弹性扩展

根据业务发展需求，动态调整计算资源的规模和配置。在业务高峰期，通过增加资源投入、优化资源分配等方式，提升系统的处理能力；在业务低谷期，通过释放冗余资源、降低运营成本等方式，提高资源利用效率。

**4.2.6** 安全防护

在资源调度过程中，加强安全防护措施，确保数据的安全性和完整性。通过加密、隔离、审计等手段，防止数据泄露、篡改和非法访问等安全事件的发生。

## 5 调度方法与策略

### 5.1 实时监控与数据采集

**5.1.1** 实时监控的方法与技术

a） 分布式监控架构

智算中心应采用分布式监控架构，通过部署多个监控节点，实现对计算资源的全面覆盖。每个监控节点负责收集其所在区域的资源使用情况，并上报给中央监控中心进行统一处理。

b） 实时数据采集技术

利用高效的数据采集技术，如基于消息队列的异步通信、流式数据处理等，确保监控数据的实时性和准确性。这些技术能够实时捕获计算资源的状态变化，为调度决策提供及时的信息支持。

c） 可视化监控工具

采用可视化监控工具，如仪表盘、热力图等，直观展示计算资源的实时状态。这些工具能够帮助管理员快速识别资源瓶颈和潜在风险，提高调度决策的效率和准确性。

**5.1.2** 监控指标的确定

a） CPU 使用率

反映计算节点的 CPU 资源占用情况，是评估计算性能的重要指标。

b） 内存使用率

反映计算节点的内存资源占用情况，对于内存密集型应用尤为重要。

c） GPU 使用率

反应 GPU 计算核心资源占用情况，是评估通用并行计算任务负载与效率的关键指标。

d） NPU 使用率

反应神经网络处理单元资源占用情况，是评估 AI 计算任务执行效率与性能的核心指标。

e） 网络带宽

反映计算节点之间的数据传输速率，对于分布式计算和网络通信密集型应用具有重要意义。

f） 磁盘 I/O

反映计算节点的磁盘读写速度，对于存储密集型应用至关重要。

g） 任务队列长度

反映等待执行的任务数量，是评估系统负载和调度效率的重要指标。

**5.1.3** 数据采集方式

a） 主动采集

监控节点定期向计算节点发送采集请求，获取其资源使用情况。这种方式适用于对实时性要求不高的场景。

b） 被动采集

计算节点主动向监控节点上报其资源使用情况。这种方式适用于对实时性要求较高的场景，能够确保数据的及时性和准确性。

c） 混合采集

结合主动采集和被动采集的方式，根据实际需求灵活调整采集策略。

**5.1.4** 数据处理和分析

a） 数据清洗

对采集到的原始数据进行清洗，去除无效、重复或异常数据，确保数据的准确性和可靠性。

b） 数据聚合

将多个监控节点的数据进行聚合，形成全局资源使用情况视图。这有助于管理员从整体上把握计算资源的分布和使用情况。

c） 数据分析

利用数据分析技术，如时间序列分析、聚类分析等，对监控数据进行深入挖掘和分析。这有助于发现资源使用的规律和趋势，为调度决策提供科学依据。

d） 异常检测

通过设定阈值、建立预警模型等方式，实时监测资源使用情况的异常变化。一旦发现异常，立即触发预警机制，确保系统能够及时响应和处理。

## 5.2 动态调度算法

**5.2.1** 核心原则

a） 基于需求的调度

根据任务的资源需求（如 CPU、内存、GPU、NPU 等）进行资源分配，确保任务能够获得所需的计算资源。

b） 基于优先级的调度

根据任务的优先级进行资源分配，确保高优先级任务优先获得资源，以满足关键业务或紧急需求。

c） 基于资源利用率的调度

根据计算节点的资源利用率进行资源分配，确保资源负载均衡，避免资源过载或闲置，提升整体资源利用效率。

**5.2.2** 核心组件与流程

a） 需求分析与预测系统

1） 实时分析任务的资源需求，包括 CPU、内存、GPU、NPU 等。

2） 利用机器学习算法预测未来任务需求趋势，为资源分配提供决策依据。

b） 优先级评估与排序模块

1） 根据任务类型、紧急程度、业务价值等因素评估任务优先级。

2） 对任务进行排序，确保高优先级任务优先获得资源。

c） 资源监控与利用率评估系统

1） 实时监控计算节点的资源使用情况，包括 CPU、内存、GPU、NPU、存储及网络等。

2） 评估资源利用率，识别资源瓶颈，预测资源需求变化。

d） 动态资源分配引擎

1） 根据任务需求和资源状态，结合优先级和资源利用率，动态调整计算节点的资源分配。

2） 引入弹性伸缩机制，根据任务负载自动调整计算资源。

3） 确保资源负载均衡，避免资源过载或闲置。

e） 智能任务调度器

1） 采用多种调度策略，如优先级调度、时间片轮转、抢占式调度等。

2） 根据任务优先级、资源需求及资源利用率，动态调整任务执行顺序。

3） 实时监测任务执行状态，确保任务按时完成。

f） 冲突检测与解决机制

1） 实时监测任务之间的资源竞争和冲突。

2） 采用任务迁移、资源预留、任务重排等策略解决冲突。

3） 确保资源分配的公平性和高效性。

g） 反馈与优化系统

1） 收集任务执行结果和资源利用情况的反馈数据。

2） 根据反馈数据，动态调整调度策略和资源分配权重。

3） 引入自适应学习机制，持续优化调度算法，提升系统性能。

**5.2.3** 实施步骤

a） 系统初始化

部署需求分析与预测系统、优先级评估与排序模块、资源监控与利用率评估系统，配置动态资源分配引擎和智能任务调度器。

b） 实时监控与评估

实时采集计算节点的资源使用情况，评估资源利用率，预测资源需求趋势；同时分析任务需求，评估任务优先级。

c） 动态资源分配

根据任务需求、优先级及资源利用率，动态调整计算节点的资源分配，确保资源负载均衡，高优先级任务优先获得资源。

d） 智能任务调度

根据任务优先级、资源需求及资源利用率，动态调整任务执行顺序，确保任务高效执行。

e） 冲突检测与解决

实时监测任务之间的资源竞争和冲突，采用相应策略解决冲突，确保资源分配的公平性和高效性。

f） 反馈与优化

收集任务执行结果和资源利用情况的反馈数据，根据反馈数据动态调整调度策略和资源分配权重，持续优化调度算法。

### 5.3 优先级管理策略

#### 5.3.1 需求洞察与资源策略

a） 精准需求解析

1）充分了解客户的业务需求，包括算力资源的目标、规模、性能要求、应用场景等。

2）对不同任务进行细致分析，明确其对 CPU、GPU、NPU、内存、网络带宽等资源的需求。

b） 前瞻资源规划

1）根据任务需求，评估所需的算力资源规模和类型，制定相应的资源规划策略。

2）预测未来的算力需求，进行适当的资源预留，避免资源不足或浪费。

#### 5.3.2 任务分类与优先级排序

a） 明确任务分级

根据任务的性质、优先级和紧急程度，将任务分为不同的类别，如高优先级任务、中优先级任务和低优先级任务。

b） 优先级动态调整

1）设定任务的优先级顺序，确保高优先级任务优先获得资源，并最大化资源利用率。

2）在资源紧张时，优先满足高优先级任务的需求，同时考虑低优先级任务的资源分配。

#### 5.3.3 智能调度与负载均衡

a） 智能算法驱动

采用先进调度算法，如遗传算法、粒子群优化，根据任务的实际需求和当前资源状态，动态地调整任务的执行顺序和资源分配。

b） 均衡负载分配

1）使用负载均衡算法，将任务合理地分布到不同的计算节点上，避免资源过度集中导致性能瓶颈。

2）根据各节点的负载情况，动态地调整资源分配策略，确保所有任务都能够及时完成。

#### 5.3.4 性能提升与资源复用

a） 性能优化策略

1）通过合理的资源分配和任务调度，提高系统的整体性能。

2）使用并行计算、内存共享等技术，提升算力任务的并发处理能力，减少资源浪费。

b） 资源高效复用

1）在可能的情况下，复用已分配的资源，以减少资源开销。

2）通过任务并行和资源共享，提高资源利用率，降低成本。

#### 5.3.5 弹性扩展与故障应对

a） 弹性资源管理

1）根据需求的动态变化，实施弹性伸缩策略，实现资源的自动扩容和缩减。

2）使用自动化的资源伸缩工具或云计算平台，根据负载情况和业务需求，自动调整算力资源的数量和规模。

b） 故障快速响应

1）建立故障监测和容灾机制，对算力资源进行故障预防、故障检测和快速恢复。

2）通过实施合适的备份和冗余策略，确保故障时的数据完整性和业务连续性。

#### 5.3.6 安全与隐私保障

a） 数据安全加固

1）在算力管理和调度过程中，确保数据的安全性和隐私保护。

2）采用身份认证、访问控制、加密传输等安全措施，防止未经授权的访问和数据泄露。

b） 隐私严格保护

遵守相关法律法规，保护用户隐私信息，避免数据泄露和滥用。

#### 5.3.7 持续监控与优化

a） 实时监控体系

1）通过实时监控和采集算力资源的状态和性能指标，建立统一的监控平台。

2）全面了解资源利用率、负载情况和故障预警，为管理和调度提供基础数据支持。

b） 策略持续优化

1）根据监控数据，持续优化资源调度策略，提高资源利用率和服务质量。

2）不断学习新技术和行业趋势，优化算力资源管理和调度的各个环节，以满足不断演进的需求。

## 6 关键技术与应用

### 6.1 关键技术

#### 6.1.1 算力资源池化

a） 定义

算力资源池化是指将物理计算资源（如 CPU、GPU、NPU、FPGA、ASIC 等）封装成独立的虚拟计算环境，实现算力资源的按需分配和弹性扩展。

b） 技术实现

通过虚拟化技术和容器技术，将物理计算资源转化为虚拟资源池，供上层应用按需调用。

c） 优势

提高算力资源的利用率，降低总体成本，提供弹性计算能力，支持大规模计算任务。

#### 6.1.2 智能调度算法

a） 定义

智能调度算法是指根据任务的计算需求和优先级，动态调整算力资源的分配，确保高优先级任务优先执行，最大化资源利用率。

b） 技术实现

基于人工智能算法（如强化学习、深度学习等）对算力资源进行智能调度。

c） 优势

提高计算任务的执行效率，减少计算时间，更好地应对突发性任务的需求。

#### 6.1.3 负载均衡技术

a） 定义

负载均衡技术是指根据各节点的负载情况，将任务合理地分布到不同的节点上，避免资源过度集中导致性能瓶颈。

b） 技术实现

通过负载均衡算法（如轮询、最少连接数等）实现资源的均衡分配。

c） 优势

提高整体的计算效率，确保系统的稳定运行。

#### 6.1.4 容错处理机制

a） 定义

容错处理机制是指通过备份机制和容错算法，确保在硬件故障或网络延迟等异常情况下，任务能够持续执行和系统的稳定运行。

b） 技术实现

采用数据备份、恢复和容灾等策略，以及冗余计算资源部署。

c） 优势

提高系统的可靠性和可用性，确保任务的连续执行。

#### 6.1.5 高性能存储技术

a） 定义

高性能存储技术是指采用高性能、高扩展性的存储系统，满足海量数据的存储需求。

b） 技术实现

采用分布式存储系统、对象存储系统以及数据仓库等。

c） 优势

提高数据存储和访问的效率，确保数据的安全性和可靠性。

### 6.2 应用

#### 6.2.1 AI 训练与推理

a） 应用场景

AI 训练和推理是智算中心的主要应用场景之一。

b） 实施方法

通过算力资源池化和智能调度算法，为 AI 训练和推理任务提供强大的计算能力。

c） 效果

提高 AI 模型的训练速度和推理精度，加速 AI 应用的落地。

#### 6.2.2 大数据处理

a） 应用场景

大数据处理是智算中心的另一个重要应用场景。

b） 实施方法

利用负载均衡技术和高性能存储技术，实现大数据的快速处理和存储。

c） 效果

提高大数据处理的效率，支持实时数据分析和决策。

#### 6.2.3 高性能计算

a） 应用场景

高性能计算是智算中心在科研、工业设计等领域的重要应用。

b） 实施方法

通过算力资源池化和智能调度算法，为高性能计算任务提供充足的计算资源。

c） 效果

提高计算任务的执行效率，缩短科研和工业设计的周期。

#### 6.2.4 云计算与边缘计算

a） 应用场景

云计算和边缘计算是智算中心在分布式计算领域的重要应用。

b） 实施方法

通过算力资源池化和智能调度算法，实现云计算和边缘计算的高效协同。

c） 效果

提高分布式计算的效率，支持边缘计算和云计算的无缝对接。

## 7 实施方案与步骤

### 7.1 实施方案

#### 7.1.1 资源管理架构设计

a） 总体架构

智算中心的计算资源管理应构建统一的资源调度与管理体系，形成“资源池化—调度引擎—策略引擎—监控系统”四层架构，实现算力、存储、网络等资源的统一调度与高效利用。系统架构应具备高可扩展性、容错性及安全性。

b） 资源池化

1） 计算资源（包括 CPU、GPU、NPU 等）、存储资源及网络资源应实现资源池化管理，通过虚拟化或容器化技术进行统一抽象与分配。

2） 资源池应支持弹性扩缩容、跨域资源调度及资源隔离控制。

3） 资源池化平台应提供统一的资源服务接口（API），供上层调度与业务系统调用。

4） 资源池应具备资源使用率统计、配额分配与资源健康状态检测功能。

c） 调度引擎设计

1） 调度系统应采用分层调度架构，包括全局调度器和节点调度器两级结构。

2） 全局调度器应负责跨集群、跨区域的资源分配与调度策略制定，节点调度器应负责本地任务的资源绑定、任务执行与状态上报。

3） 调度引擎应支持多种调度算法插件（如优先级调度、公平调度、能耗调度等），并具备算法可扩展能力。

4） 调度引擎宜采用开源框架（如 Kubernetes、Apache Mesos）构建，结合自研模块满足智算中心特定需求。

5） 调度引擎应具备任务重调度、故障恢复及任务迁移能力，以保障系统连续性与任务可靠性。

d） 策略引擎设计

1） 智算中心应建立统一的策略引擎，实现资源分配、任务调度、能耗控制及容灾恢复等策略的集中管理与执行。

2） 策略引擎应支持动态策略加载与热更新，允许根据业务需求实时调整。

3） 策略引擎宜支持以下策略类型：

优先级策略：根据任务等级、业务类型或用户权限确定资源分配顺序；

公平分配策略：在多用户或多部门环境中平衡资源使用；

资源预留策略：为关键任务、核心业务或紧急任务预留资源配额；

能耗优化策略：通过节点休眠、负载合并等方式降低功耗；

容灾策略：在节点或系统故障时自动触发任务迁移与恢复。

e） 监控与日志系统

1） 智算中心应建立覆盖计算、存储、网络及任务运行状态的监控体系，实时监测资源利用率、任务执行状态与系统健康度。

2） 监控系统应支持告警与阈值配置功能，能够在资源异常或系统故障时自动触发告警。

3） 日志系统应统一采集调度行为、任务运行记录及异常事件，并支持日志审计与追溯分析。

4） 监控与日志系统宜采用 Prometheus、Zabbix 等主流监控框架，并结合 Grafana 或等效工具进行可视化展示。

5） 日志与监控数据应纳入安全策略体系，防止信息泄露与数据篡改。

### 7.1.2 关键技术与工具选择

a） 容器化技术

1） 智算中心应采用容器化技术（如 Docker、Podman 等）实现应用与运行环境的解耦。

2） 容器镜像应通过统一镜像仓库进行管理，并具备版本控制、安全扫描与签名校验机制。

3） 容器运行环境应支持快速部署、迁移与扩缩容。

b） 编排与调度平台

1） 智算中心应选用 Kubernetes 作为容器编排与调度核心平台。

2） Kubernetes 集群应采用高可用（HA）架构，保证调度系统的稳定性与连续性。

3） 对 AI 训练、推理等特殊场景，可结合 KubeFlow、Ray 等框架提升分布式任务管理能力。

c） 高性能存储解决方案

1） 智算中心应根据业务特征选用分布式文件系统（如 Ceph、Lustre）或对象存储系统。

2） 应采用 NVMe SSD 等高性能介质提升数据读写速率。

3） 应建立存储分层（冷热数据分层）与缓存机制，以提高资源利用率与访问效率。

d） 网络通信优化

1） 智算中心宜采用高速互连技术（如 InfiniBand、RDMA over Ethernet）以减少通信延迟。

2） 网络架构应支持 SDN 技术，实现带宽调度与流量隔离。

3） 网络应具备 QoS 控制能力，保障关键任务通信优先级。

### 7.1.3 调度策略制定与实施

a） 基于优先级的调度

1） 应建立任务分级机制，将任务划分为关键任务、核心任务与普通任务。

2） 应为高优先级任务预留算力资源，以保障其调度优先执行。

3） 调度系统可支持手动和自动双模式优先级调整机制。

b） 公平调度

1） 应在多部门或多用户场景中采用公平调度策略，防止资源被单一用户长期独占。

2） 公平调度算法宜采用加权公平队列（WFQ）或基于时间片轮转的策略。

3） 应定期统计各用户或任务资源使用量，动态调整权重参数以保持公平性。

c） 弹性调度

1） 调度系统应根据任务负载动态扩缩计算节点，实现资源弹性伸缩。

2） 当资源不足时，系统应支持任务迁移、资源抢占或临时资源调用机制。

3） 弹性调度策略宜结合预测算法，对负载趋势进行提前调整。

d） 节能调度

1） 在系统低负载时，应自动关闭空闲节点或降低 CPU 频率以减少能耗。

2） 调度系统应具备能耗监测功能，定期评估节点能效比。

3） 可通过 AI 模型预测任务周期，实现基于时间的节能策略。

#### 7.1.4 实施与运维要求

a） 智算中心应建立统一的资源调度管理平台，支撑资源申请、任务调度、状态监控和日志审计等全流程管理功能。

b） 应建立资源使用审计与配额管理制度，确保资源分配合规、可追溯。

c） 应制定调度系统的容灾备份方案，确保系统在异常情况下可快速恢复。

d） 应建立持续优化机制，定期评估调度算法性能并进行调整改进。

e） 运维人员应定期开展调度系统安全检测与性能测试，保证系统长期稳定运行。

### 7.2 实施步骤

#### 7.2.1 需求分析与规划

a） 调研现状

评估现有计算资源、业务需求及未来增长预期。

b） 制定方案

基于分析结果，设计资源管理架构、选择技术与工具、制定调度策略。

#### 7.2.2 系统部署与集成

a） 环境准备

搭建物理或虚拟环境，部署必要的硬件与软件基础设施。

b） 资源池构建

配置资源池，包括 CPU、GPU、NPU、存储等资源的虚拟化与配置。

c） 调度引擎安装与配置

安装并配置 Kubernetes 等调度引擎，集成监控与日志系统。

#### 7.2.3 策略配置与测试

a） 策略实施

根据制定的调度策略，在调度引擎中配置相应的规则与参数。

b） 系统测试

进行功能测试、性能测试与压力测试，确保系统稳定且符合预期。

#### 7.2.4 用户培训与支持

a） 培训

对用户进行资源管理系统的操作培训，确保他们能够高效利用资源。

b） 文档编写

编写用户手册、操作指南与常见问题解答，方便用户自助解决问题。

#### 7.2.5 上线运行与持续优化

a） 正式上线

完成所有测试与准备工作后，正式将系统投入运行。

b） 监控与优化

持续监控系统运行状态，根据反馈调整调度策略，优化资源使用效率。

c） 定期评估

定期进行资源使用效率评估，及时调整资源分配策略，确保资源持续优化利用。

# 本标准用词说明

1 为便于执行本标准条文时区别对待，对要求严格程度不同的用词说明如下：

1） 表示很严格，非这样做不可的：

正面词采用“必须”，反面词采用“严禁”；

1） 表示严格，在正常情况下均应这样做的：

正面词采用“应”，反面词采用“不应”或“不得”；

2） 表示允许稍有选择，在条件许可时首先应这样做的：

正面词采用“宜”，反面词采用“不宜”；

3） 表示有选择，在一定条件下可以这样做的：

正面词采用“可”反面词采用“不可”；

2 条文中指定应按其他有关标准执行的，写法为“应符合……的规定”或“应按……执行”。