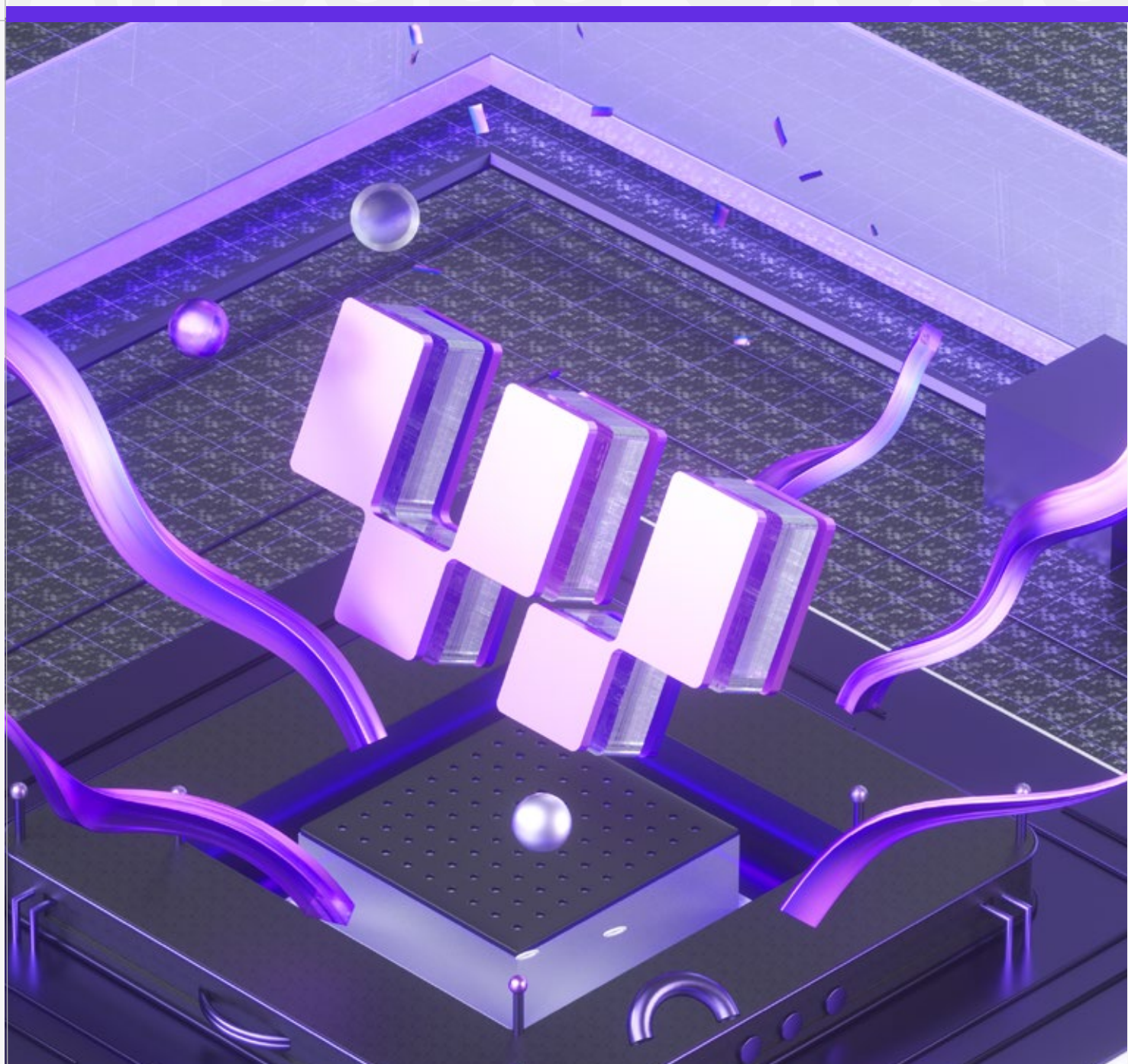
阿里云 CAICT 中国信通院

生成式AI卓越架构设计指导原则



前言 PREFACE

AI 时代的新挑战

智能化转型进入关键阶段

全球数字化、智能化转型正处于关键跃升阶段。人工智能正加速与各行各业深度融合，推动新兴产业形态和传统产业升级。各方对人工智能应用的稳定性、安全性和可信赖性提出了更高要求，智能化发展已成为产业演进和社会进步的重要驱动力。

全球智能革命不断深化，人工智能市场持续扩大

人工智能正以前所未有的速度重构全球产业格局。Gartner 预测，到2026年，全球多数企业将在生产系统中部署生成式AI能力，大模型驱动的认知计算正在深刻改变制造业、服务业等领域的价值链。

中国在人工智能技术研究和产业应用方面持续保持增长，相关企业数量和产业规模快速扩大，呈现出强劲的发展势头。随着行业智能化转型的深入，不同行业的差异化需求不断涌现，对 AI 算力、平台、算法模型和行业解决方案提出了更高要求。

人工智能技术快速演进，AI 应用面临全新挑战

人工智能技术进入体系化突破新阶段，推动软件工程向智能化演进。大语言模型（LLM）正在重塑软件开发模式，生成式AI推动人机协同开发逐渐成为主流；与此同时，对AI信任、风险与安全管理（TRiSM）的需求愈加迫切，模型运维（ModelOps）、智能体运维（AgentOps）、AI 安全与模型监控正成为企业关注的重点。

然而，AI 应用在大规模落地时仍面临诸多挑战：

- 数据依赖度高：高质量数据供给难度大，数据漂移可能导致模型性能退化。
- 模型迭代复杂：生命周期涵盖训练、验证、部署、监控与回滚，迭代过程对系统稳定性要求高。
- 资源需求波动大：训练阶段计算资源消耗巨大，推理阶段需低延迟与稳定性，增加了成本与扩展难度。
- 技术与标准不完善：AI场景下缺乏成熟的监控、可观测性与运维机制，行业内最佳实践尚未形成统一标准。
- 安全与合规挑战：数据隐私保护、算法偏见、模型攻击与可解释性要求日益突出。
- 成本与收益难平衡：持续监控、多模型管理和跨团队协作带来高昂成本，创新速度与风险控制需要兼顾。

指导原则目的与目标读者

本指导原则的编写目的，是为正在探索或已经部署生成式AI的企业与团队，提供一套系统化的架构方法论与最佳实践指引。它不仅适用于超大规模企业，也同样适用于中小企业（SMB）。随着大模型与 AI 应用逐步普及，中小企业在产品创新、业务流程优化、客户体验升级等方面，同样面临高可用架构设计、成本优化、安全合规等挑战，因此也亟需参考一套成熟的方法论。

目的

- 帮助企业在生成式 AI 的建设过程中，识别和解决设计的关键挑战。
- 帮助不同规模企业在安全、稳定、性能、成本、效率五个维度提供建议。
- 通过方法论与工具，降低企业在构建 AI 应用时的试错成本，加速 AI 落地。
- 协助企业从“能用AI”逐步走向“用好AI”，实现从云卓越到AI卓越的演进。

目标读者

本指导原则面向的读者群体十分广泛，涵盖了企业在使用生成式AI过程中涉及的多种角色：

架构团队	包括 CTO、架构师、研发、MLOps/DevOps 工程师，帮助他们理解如何构建高可用、可扩展的 AI 基础架构。
安全合规团队	包括安全专家、审计、数据治理人员，帮助他们建立 AI 数据全生命周期的安全与合规体系。
运维团队	包括运维、监控、IT 管理人员，帮助他们利用自动化与可观测性提升 AI 系统的稳定性与运维效率。
业务团队	包括业务负责人、产品经理、财务人员，帮助他们在 AI 项目中平衡业务价值、成本投入与长期可持续发展。

本文件起草单位及主要起草人

阿里云计算有限公司	何登成、张瑞、程超、施磊、张舫、朱彩辉、张瑄、周金龙、郑立异、王觯程、李鹏飞、李冬萌、李艳林、张玉峰、曹治政、杨继、孙磊、陈铖、赵星星、李春雷、潘碧玲
中国信息通信研究院	陈屹力、郑立、王海清、季可航、刘坤

CONTENT
目录

01.

Overview

概述

- 为什么需要“生成式 AI 卓越架构设计指导原则”
- 五大支柱在生成式 AI 中的延展

为什么需要“生成式AI卓越架构设计指导原则”

1.1

过去几年，阿里云通过卓越架构（Well-Architected Framework）服务了众多不同行业的大型客户，帮助他们在云上构建安全、稳定、高效的架构实践。通过卓越架构框架及评估工具，客户能够发现现有架构与最佳实践的差距，并在专家与合作伙伴的支持下不断迭代优化。这一方法论已成为企业用好云、管好云的重要基石。

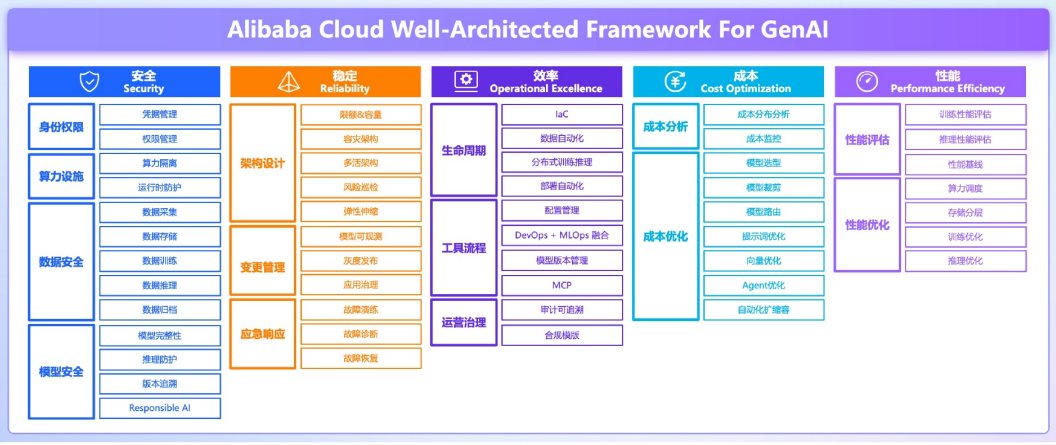
然而，随着人工智能尤其是生成式AI的迅速崛起，客户在云上的诉求正发生显著变化：

企业对 AI 技术的接受度和使用率不断提升，从探索逐渐走向生产级应用。

AI 正在渗透各行各业，从内容生成、辅助研发、产业决策，企业对大模型的训练、微调、推理提出了前所未有的需求。

相比传统业务，AI 工作负载在 GPU算力、数据合规、模型生命周期管理、成本等方面带来新的挑战，原有的架构框架需要延展与升级。

因此，阿里云基于多年服务大规模客户的经验，提出《生成式 AI 卓越架构设计指导原则》，在原有卓越架构的基础上，扩展出面向 AI 时代的设计原则与最佳实践。该指导原则既传承了卓越架构框架的核心价值，也针对生成式AI的特殊性，提供新的方法论与解决方案，帮助客户在 AI 战略落地过程中实现“卓越”。
值得强调的是，卓越架构框架提出的五大支柱（安全、稳定、效率、成本、性能）在生成式AI时代同样适用。



五大支柱在生成式AI中的延展

1.2

无论是保障 AI 数据的全生命周期安全，确保大模型训练推理的高可用与性能，还是优化 GPU 算力成本与资源效率，这些支柱依旧是企业评估与优化架构的核心维度，但其内涵和实践重点都需要结合AI的特性进行延展：

安全

生成式AI涉及的数据来源更为复杂，涵盖个人隐私、企业敏感信息和跨境数据流动。确保数据全生命周期的安全合规，以及模型输出的可信与可解释，成为构建AI应用的首要前提。

稳定

大模型训练与推理任务往往持续时间长、规模庞大，任何节点的故障都可能导致重大损失。AI架构需要具备面向失败的设计能力、全链路的容灾方案以及多层次的可观测性，以保证业务连续性。

效率

生成式AI应用的迭代速度远超传统软件，企业需要新的运维模式，支持多模型协同、快速灰度发布与持续监控，形成从开发到上线的闭环运维能力。

成本

AI场景对GPU等高性能算力的需求极为突出，若缺乏有效管理，极易造成资源浪费与成本失控。通过弹性调度、算力池化、Spot实例与混合精度计算等手段，企业可以在性能与成本间找到平衡。

性能

AI模型规模不断扩展，对存储 I/O、网络带宽和推理延迟的要求更高。通过分布式训练框架、推理加速引擎和边缘侧优化等技术，可以有效提升端到端性能，保障用户体验与业务价值。

02.

SECURITY

安全

在生成式 AI 架构中，安全是最核心也是最复杂的挑战。与传统应用相比，AI 系统的数据规模更大、模型更复杂、调用链更长，潜在风险也更加多样化：既包括数据在采集、存储、训练、推理、归档等环节的合规与保护问题，也包括算力和容器运行时的安全隔离，模型供应链中的第三方依赖与参数篡改风险，以及生成式 AI 独有的公平性、可解释性与滥用防护问题。阿里云在多年服务金融、政企、互联网等对安全要求高的客户的过程中，积累了全栈安全能力，并在生成式 AI 领域进一步扩展：提供覆盖数据全生命周期、算力与容器、模型供应链、Responsible AI 的整体安全防护框架，形成从基础设施到应用层的全链路安全能力。

- **数据全生命周期安全**
- **算力与容器安全**
- **模型供应链安全**
- **RESPONSIBLE AI**

数据全生命周期安全 2.1

数据是生成式 AI 的核心资产，其安全设计必须贯穿采集、存储、训练、推理和归档全流程。相比传统系统，生成式 AI 对数据安全提出了更高要求：不仅要保护机密性与完整性，还要防止滥用、篡改与投毒，并确保合规与可追溯性。

在金融、医疗等行业场景中，数据全生命周期安全尤为关键。例如在银行的智能客服中，客户交易数据若在训练阶段泄露，将导致严重合规风险；在医疗影像诊断中，若训练集被投毒，模型可能做出错误诊断，直接威胁人身安全。因此，企业必须在每个环节设计严格的安全防控措施。

数据采集

在接入阶段，应验证数据来源的合规性，采用 TLS/HTTPS 加密传输、API 鉴权和访问控制。敏感数据需提前脱敏或匿名化，避免早期暴露风险。对于跨境数据流动，应符合 GDPR、数据出境安全评估等要求。

数据存储

在存储层面，应实施细粒度访问控制和最小权限原则，结合加密存储、密钥托管与定期轮转，防止数据泄露或非法访问。多租户环境下建议引入零信任架构，结合 VPC 隔离和加密隧道，确保数据只在可信边界内流动。

数据训练

训练环节需重点防范数据投毒和偏见样本，可通过数据清洗、异常检测、分布漂移监测来降低风险。涉及跨机构合作的，可采用联邦学习和隐私计算，在不暴露原始数据的情况下实现模型协作训练。

数据推理

推理过程中常见的攻击包括 Prompt 注入、对抗样本攻击和越权调用。需要结合输入验证、内容过滤和输出审计提升可信度；在 RAG（检索增强生成）场景中，还需确保外部知识源的权威性和安全性，避免引入虚假或恶意内容。

数据归档与销毁

归档阶段应采用冷热分层存储与自动化生命周期管理，防止长期数据滞留带来合规风险。销毁环节应采用加密删除、覆盖写入等机制，并保留日志记录，满足等保、ISO/IEC 27001 等审计要求。

设计原则总结

DESIGN PRINCIPLES

采集与接入
数据来源合规、传输加密、数据脱敏。

存储与访问
最小权限控制、密钥托管、网络隔离。

训练防护
数据质量检测、投毒防御、协作隐私保护。

推理安全
输入验证、内容过滤、知识源可信。

归档与销毁
自动化生命周期管理、合规留存与安全删除。

算力与容器安全

2.2

算力与容器平台是大模型训练和推理的核心基础设施，安全风险一旦出现可能直接影响训练结果和业务连续性。在 AI 时代，GPU/TPU 集群成为攻击者新的重点目标，其安全性决定了整个 AI 系统的可信度。

异构算力隔离

在多租户环境下，GPU/TPU 等资源需通过虚拟化或沙箱技术实现隔离，避免横向攻击或越权访问。对于涉及金融、国防等高敏感任务的企业，可采用物理隔离或专用实例，减少攻击面。

容器运行时防护

容器作为大模型训练和推理的主要承载方式，应防御容器逃逸与恶意镜像风险。可采用沙箱化运行时，结合可信镜像签名、镜像仓库安全扫描，确保镜像来源可验证。

平台漏洞管理

训练平台需具备漏洞扫描与配置审查能力，及时修补依赖与组件缺陷。例如，某客户在使用开源 ML 框架时因未及时更新补丁而导致 GPU 任务中断，这类事件强调了漏洞管理的重要性。

密钥与凭据管理

算力与容器调用应通过集中化密钥托管（如阿里云 KMS）、临时凭据（STS Token）和最小权限策略管理，避免明文凭据暴露。

隐私计算与可信执行

在涉及敏感数据或跨组织建模时，可采用可信执行环境、安全多方计算与同态加密，保障算力层的全程安全。随着 Confidential Computing 的普及，AI 算力在可信硬件上的运行将成为行业趋势。

可观测性与运行监控

算力与容器运行时应进行全程监控与追溯。通过 GPU/CPU 利用率、内存带宽、I/O 吞吐等指标，结合异常检测模型，可以快速识别资源耗尽、拒绝服务或挖矿等异常行为。

DESIGN PRINCIPLES

设计原则总结

隔离优先
算力与容器在多租户环境下必须严格隔离。

可信镜像
运行时采用可信镜像和沙箱化运行环境。

漏洞管理
持续进行漏洞扫描与资源健康检查。

凭据最小化
凭据托管、临时授权与最小权限。

隐私计算
在敏感任务中启用硬件级别可信执行。

可观测性
算力与容器运行时全程监控与追溯。

模型供应链安全

2.3

生成式 AI 的供应链涵盖预训练模型、开源框架、第三方数据集、工具链与推理服务，任何环节的不安全都会传导至最终应用。例如，若引入的开源模型包含后门，企业可能在不知情的情况下将风险暴露给用户。

第三方模型与依赖

引入的预训练模型和开源组件需经过完整性校验与漏洞检测。企业应优先选择可信来源（如官方库、阿里云模型服务）的模型与依赖仓库，减少后门与恶意篡改风险。

模型完整性保护

对模型权重文件应实施加密存储、访问控制和哈希校验，必要时结合签名验证与参数比对机制，确保模型未被篡改。对访问行为需进行日志记录与审计。

推理防护

推理阶段应防范 Prompt 注入、对抗样本和越权调用，可通过输入过滤、输出内容审计与速率限制机制保障安全。

版本与依赖追溯

在模型迭代中，应记录训练数据、超参数、依赖组件和服务版本，确保在安全事件发生时能快速定位与追溯。这不仅是安全问题，也关系到合规与模型可解释性。

设计原则总结

DESIGN PRINCIPLES

来源可信

优先使用经过认证的模型与依赖。

模型完整性

对模型文件进行加密与完整性校验。

推理防护

输入过滤、输出审计、防御对抗攻击。

版本追溯

全链路的版本与依赖管理。

RESPONSIBLE AI

2.4

生成式 AI 不仅要技术安全，还要符合**公平性、可解释性、合规性和滥用防护**要求。
Responsible AI 是从通向“可信赖”AI的关键桥梁。

公平性与偏差治理

通过数据质量检测、偏差分析与去偏方法，降低训练和推理过程中的不公平性风险。例如在招聘系统中，若训练数据存在性别或地域偏见，AI 将可能放大这种歧视。

可解释性与透明性

在金融、医疗等敏感行业，应提供特征重要性分析、结果可视化和版本追溯能力，确保模型决策过程可解释。建立算法备案与模型登记机制，满足监管要求。

合规与监管

不同国家和地区对 AI 有严格的法律规范。欧盟要求高风险 AI 系统满足透明性和可追溯性；中国《生成式人工智能服务管理办法》要求平台落实内容安全和可追溯责任。企业需建立合规治理流程，实现责任认定。

滥用防护与内容安全

生成式 AI 系统需内置内容检测与过滤机制，对不当或违法内容进行实时拦截，并结合人工审核与反馈机制持续优化。例如在社交平台，AI 生成的有害内容必须被快速发现与处理。

插件与用户行为审计

随着 AI Agent 和插件生态的兴起，越权调用和恶意插件成为新风险。需对第三方插件采用最小权限与沙箱隔离，对用户调用进行全链路日志与异常行为分析，防止滥用。

DESIGN PRINCIPLES

设计原则总结

公平性
主动检测并缓解偏差，确保结果公正。

可解释性
提供透明性、版本追溯与算法备案。

合规溯源
通过水印与审计机制满足监管要求。

输入输出防护
防范越权请求、提示注入与有害生成。

插件与用户治理
严格准入、沙箱隔离与行为审计。

分级防护
根据内容风险等级实施分层防护。

03.

RELIABILITY

稳定

生成式 AI 模型对算力和架构稳定性的要求极为严苛。无论是数千卡规模的大模型训练，还是支撑亿级请求量的推理服务，系统都必须在网络抖动、硬件故障、流量突增等不可避免的异常情况下，仍能保持稳定运行。业界在长期服务大规模互联网和企业级应用的过程中，逐步沉淀出稳定性设计的最佳实践，确保 AI 系统能够实现高可用与高可靠。

- 弹性调度
- 模型推理的 SLA 与冗余架构
- 分布式训练的容错与检查点恢复
- 监控与可观测性
- 灾备设计

弹性调度

3.1

在生成式 AI 的训练和推理过程中，GPU/TPU 等异构算力是最核心的资源。一方面，这些算力昂贵且有限，如果缺乏弹性调度和故障切换能力，就可能因单点故障或流量突增而导致任务中断；另一方面，算力利用率直接决定总体成本与投资回报率（ROI）。

通过**大规模算力调度平台**实现 GPU/TPU 的统一编排与动态调度，结合多可用区、多地域的资源池，为 AI 任务提供高可用和跨集群的算力保障。当算力节点发生硬件故障或网络异常时，系统会自动识别并迁移任务，避免训练和推理中断。这种**“面向失败”**的设计理念，能够确保在设计之初就具备冗余、隔离和弹性扩展的能力。

同时，通过**混合实例与异构混部**，企业可以在不同业务场景下灵活选择 GPU、CPU 或低阶加速卡的组合；对于非关键任务，可以通过**抢占式实例** 获取弹性算力资源，以降低成本。在多租户或多业务单元场景下，还可通过**算力配额（quota）管控机制，**为关键业务提供弹性保障，对低优先级业务设定上限，避免关键业务的资源被抢占从而导致服务雪崩，并提升算力投资回报率。

此外，平台应具备**算力健康检查与容量管理**的能力，实时监控 GPU/CPU 的利用率、显存占用和网络带宽，并在接近阈值时自动扩容或预警，确保集群在高负载下依旧平稳运行。

设计原则总结

DESIGN PRINCIPLES

面向失败

假设节点必然失败，必须规划冗余与自动切换。

异构混部

灵活利用 GPU/TPU/CPU 混合资源，提升利用率与稳定性。

弹性伸缩

根据训练/推理流量与 SLA 动态调整 GPU/TPU/CPU 资源。

配额管控

按业务单元划分算力配额，关键任务弹性保障，低优先级任务设上限。

健康监控

实时监控算力状态，自动扩容与预警。

模型推理的 SLA 与冗余架构

3.2

生成式 AI 推理服务通常面向海量用户请求，任何一次中断或延迟都会直接影响业务连续性与用户体验。因此，推理架构必须围绕**高可用与可恢复**设计。

业界常见做法包括：

计算层

多实例部署与自动伸缩，确保部分节点失效不影响整体。

网络与流量层

通过负载均衡与全局流量调度实现跨可用区和跨地域冗余。

应用层

基于AI网关做模型代理，超时重试，Failover，灰度，Token级 限流 和 额度管理，连接级别并发控制。

此外，还需关注 算力吞吐能力与异常恢复机制。可通过指标如首 **Token 延迟（TTFT）**、**每秒生成 Token 数（TPS）**、**端到端延迟（E2E Latency）**、**吞吐量（Throughput）**来评估负载上限，并结合压力测试发现瓶颈。推理异常时，可通过**重试、回退至小模型或缓存结果、服务降级路径**等方式保障连续性。

DESIGN PRINCIPLES

设计原则总结

冗余与多活

在计算、网络和应用层实现多层冗余。

压测验证

标准化压力测试与基准评测工具验证鲁棒性。

异常恢复

通过重试、回退、降级等机制保障业务不中断。

容量评估

通过 TTFT、TPS、E2E Latency、Throughput 等指标评估瓶颈。

灰度与回滚

变更过程做到灰度发布，出现问题及时回退，变更过程可观测。

分布式训练的容错与检查点恢复 3.3

大规模分布式训练需要数百甚至上千张 GPU/TPU卡，持续周期可能长达数周。任何硬件或网络异常都可能导致巨额浪费，因此必须具备容错与恢复能力。

业界常见机制包括：

节点级故障隔离与任务迁移

确保个别节点异常不会影响整体训练。

分布式训练框架优化（如 ELASTICDL、HOROVOD 等）

支持断点续训与任务重调度。

分布式检查点机制

定期保存模型参数、优化器状态与中间结果至持久化存储，故障时可快速恢复。

分层存储策略

高频权重存储在高性能介质，历史版本归档至低成本存储。

扩展性评测

验证通信延迟、梯度同步效率、数据加载瓶颈与 GPU 利用率。

DESIGN PRINCIPLES

设计原则总结

断点续训
周期性检查点保存，故障后快速恢复。

任务重调度
节点级容错与任务迁移机制。

分层存储
平衡性能与成本的检查点保存策略。

扩展性验证
评估梯度同步、通信延迟等指标。

容错评测
设计阶段进行标准化容错与恢复演练。

监控与可观测性

3.4

生成式 AI 系统链路复杂，必须通过可观测性体系实现**指标监控、全链路追踪、日志分析与审计。**

关键指标包括：

首 TOKEN 延迟（TTFT）

衡量用户体验。

端到端延迟（E2E LATENCY）

SLA 核心指标。

每秒生成 TOKEN 数（TPS）

反映模型吞吐能力。

吞吐量（THROUGHPUT）

并发能力评估。

GPU/CPU 利用率、显存占用、网络带宽

算力层健康度。

全链路追踪可记录请求在前端、API 网关、Agent、向量数据库和大模型之间的路径，快速定位异常。统一日志体系可标准化记录请求上下文、模型信息与延迟，支持审计与追溯。

在大模型场景下，还可对**推理链路日志**进行存储与分析，提升可解释性与合规性，并帮助发现潜在风险。

DESIGN PRINCIPLES

设计原则总结

全栈指标

覆盖业务、应用、模型和基础设施层。

全链路追踪

快速定位瓶颈与异常。

统一日志与审计

标准化记录请求与结果。

推理日志审计

对推理链路日志化，提升解释性与合规性。

灾备设计

3.5

即便具备容错与监控能力，AI 系统仍需应对极端场景，如大规模网络中断、硬件故障、自然灾害或云节点异常。因此，必须在设计之初就引入**跨可用区与跨地域的灾备机制。**

常见做法包括：

分层容灾

基础设施层采用多可用区部署与网络冗余；数据层使用多地域副本与异地备份；模型层通过模型仓库与热备机制实现快速切换。

跨地域多活架构

关键推理服务部署在多个地域，通过全局流量调度与多活数据库保持就近访问与冗余。

FALLBACK 机制

当主模型服务不可用时，自动Fallback到备用模型，从而提升模型可用性，避免某个模型异常带来的请求不可用。

自动化灾备演练

通过编排工具构建定期演练流程，确保灾备机制在真实场景下可用。

DESIGN PRINCIPLES

设计原则总结

分层容灾
基础设施、数据与模型层分别制定策略。

多活架构
跨地域部署保证高 SLA。

模型热备与回滚
保留模型版本仓库与热备实例。

Fallback
模型支持自动Fallback机制。

自动化演练
定期演练，确保灾备方案有效。

04.

OPERATIONAL EXCELLENCE

效率

在生成式 AI 的落地过程中，系统架构不仅要关注安全与稳定，还需要在运维和治理层面实现高效率。与传统应用相比，AI 系统的复杂性更高，涉及数据处理、模型训练、微调、部署和迭代等多个环节。如果缺乏高效的工具与平台支持，企业往往会面临 成本高昂、交付缓慢、风险难控 的问题。

业界普遍采用 DevOps 与 MLOps 融合、一体化管控平台、自动化治理与合规审计 等体系化能力，来构建覆盖全生命周期的高效运维框架，从而实现快速迭代、稳定交付与合规运营。这一支柱的核心目标，是帮助企业 在保障质量和合规的前提下提升研发与交付效率，让 AI 能力真正转化为业务价值。

- **AI 全生命周期运维**
- **DEVOPS + MLOPS 一体化**
- **统一接口与治理能力**
- **自动化治理与合规审计**

AI 全生命周期运维

4.1

AI 系统的运维范畴远超传统应用。从数据采集与清洗，到模型训练、推理部署，再到上线后的迭代优化，都需要形成闭环的全生命周期管理。

数据采集环节

通过自动化工具完成数据采集、标注、清洗和治理，减少人工干预带来的错误和低效。大规模场景下可以借助数据湖、数据安全管控平台以及特征存储，确保数据复用与一致性。

模型训练环节

利用分布式训练框架、断点恢复、实验追踪与注册中心，提升模型研发的效率和可控性。这使得企业能够在版本演进中实现实验的快速回溯，有效避免重复性工作。

部署环节

通过多环境 CI/CD 流水线、灰度发布与滚动升级，实现模型快速上线。针对推理服务，可使用弹性伸缩与无服务器（Serverless）架构，提高资源利用率。

迭代环节

上线后引入持续评估与反馈回路，支持再训练和模型替换。通过数据漂移检测、模型性能监控，可以在性能下降时自动触发再训练任务，保障服务质量。

这种覆盖全链路的运维方式，使企业能够快速响应业务需求与数据变化，并持续优化模型性能。结合日志审计与可观测性平台，企业可以在每个阶段获得实时反馈，确保过程透明、合规与可控。

设计原则总结

完整闭环

覆盖数据、模型、部署、迭代的全生命周期。

自动化

在各环节引入自动化流程，减少人工干预。

反馈驱动

基于监控与日志分析持续优化模型与运维流程。

全链路可追溯

每个阶段保留审计记录，实现可追溯性与合规性。

DEVOPS + MLOPS 一体化

4.2

随着生成式 AI 应用的快速迭代，单纯依赖传统 DevOps 或独立 MLOps 都存在局限：

- **前者难以覆盖模型生命周期的特殊需求（如实验追踪、数据版本化、模型发布）；**
- **后者则容易造成工具孤岛，算法团队与运维团队之间缺乏统一协作平台。**

一体化模式通过将代码、数据、模型与运维工作流整合在统一流水线中，实现跨团队的协作与治理。这种模式既继承了 DevOps 的快速迭代优势，又结合了 MLOps 对数据与模型的精细化管理需求，使研发、数据、算法与运维团队能够在同一平台高效协作。结合版本控制、灰度发布与回滚机制，企业能够在保障可控性的同时快速实现模型迭代与上线。

在金融风控、医疗诊断等高敏感行业，一体化框架还能够提升可审计性和合规性。例如，所有模型的输入输出均纳入统一日志体系，方便满足不同区域的法规对可追溯性的要求。

DESIGN PRINCIPLES

设计原则总结

一体化

统一 DevOps 与 MLOps 流程，避免工具割裂。

协作

支持跨团队在同一平台高效协同。

版本可控

代码、数据与模型统一纳入版本管理，支持灰度与回滚。

快速交付

流水线自动化，缩短迭代与上线周期。

统一接口与治理能力 4.3

在生成式 AI 的全生命周期中，企业需要调用和管理大量服务与外部接口，例如数据处理、模型训练、推理调用、监控告警与合规审计。如果缺乏统一的治理机制，接口容易分散在不同系统和团队中，导致效率低下、风险增加。
为解决这一问题，业界逐渐形成共识：需要通过**标准化协议与统一接口层 来承载多样化的服务调用**。企业可以将计算、存储、网络、数据库、日志等不同服务的 API 调用纳入统一治理框架，避免重复开发和运维割裂，提升整体效率与可控性。

典型能力包括：

统一鉴权与访问控制

确保调用过程安全透明，避免凭据分散。

调用链监控与日志分析

实现跨系统可观测性和问题快速定位。

第三方插件与服务扩展

降低集成成本，加快创新速度。

合规审计与访问追踪

保障外部服务集成的可控与合规，满足监管要求。

在这一背景下，Model Context Protocol (MCP) 作为新兴标准化协议逐渐受到关注。它通过规范模型与外部服务、插件、数据源的上下文交互，提供统一的调用语义和接口规范，避免了各类服务之间的割裂。MCP 的引入，使生成式 AI 系统能够在复杂生态中实现跨平台、一致化的接口治理，并天然具备可追溯与可观测能力。

在大型企业实践中，统一治理不仅依赖 API 网关进行集中管理，还可以结合 MCP 协议的能力，使内部团队和外部合作伙伴能够基于同一接口规范进行协作，显著降低沟通与集成成本。阿里云推出了 OpenAPI MCP 服务，将阿里云的 API 封装成 MCP 服务，帮助企业更高效地完成系统集成，同时确保接口层的安全与合规。

设计原则总结

统一治理
减少碎片化接口带来的复杂性。

安全合规
在 API 层引入鉴权、访问控制与审计。

效率提升
通过 MCP 和标准化 SDK 降低对接成本。

可扩展性
支持第三方插件与外部服务快速接入。

可观测性
调用链与日志透明化，便于追踪与优化。

自动化治理与合规审计 4.4

随着 AI 系统规模扩大，依靠人工治理和审计已无法满足需求，必须通过自动化手段实现高效与合规。

治理层面

自动化策略引擎可对不合规配置和异常行为进行实时告警与修正。例如，当发现 GPU 长时间空转或存储桶被意外公开时，系统可自动触发修复。

合规层面

预置的合规模板与策略可帮助企业满足不同地域和行业的监管要求，如《生成式人工智能服务管理暂行办法》、《通用数据保护条例》(GDPR)、《健康保险可携性和责任法案》(HIPAA) 等，避免人工逐条对照带来的延迟与错误。

审计层面

全链路操作留痕确保过程可追溯。结合可视化审计平台，安全与合规团队能够快速生成合规报告，减少审计成本。

未来，自动化合规还会与 AgentOps 和 AIOps 结合：系统可以在检测到模型性能下降或违规内容生成时，自动触发治理与审计流程，进一步提升效率与可控性。

DESIGN PRINCIPLES

设计原则总结

原生合规
合规要求融入AI全生命周期。

自动化
通过规则与编排自动治理资源与行为。

实时性
不合规行为需即时发现与修复。

审计可追溯
所有调用与操作可追踪，符合监管要求。

05.

COST OPTIMIZATION

成本

生成式 AI 的算力与存储消耗远超传统业务，如果缺乏合理的成本治理，企业容易面临巨额账单和资源浪费。在很多案例中，企业在上线生成式 AI 服务的首月，就发现云资源账单翻倍甚至数倍增长。成本不仅是财务指标，它与架构设计、资源调度、模型选择密切相关：

- 算力层面，GPU/TPU 是成本核心，占总开销大头；
- 存储层面，长期保存的海量训练数据、版本化模型文件带来显著负担；
- 推理层面，高并发调用若缺乏优化，容易造成 Token 重复消耗和 GPU 空转；
- 治理层面，若缺乏可观测性与透明化，企业往往无法定位“成本黑洞”。

业界在服务大规模 AI 训练与推理任务的过程中，总结出一套系统化的成本优化实践，涵盖算力选型、存储分层、资源可观测性、模型复用等方法，帮助企业在满足业务 SLA 的同时实现性能与成本的平衡。

成本类别	涉及项
算力成本	训练 GPU、推理 GPU、CPU 辅助资源
存储成本	模型权重、训练数据集、缓存、日志
网络成本	跨区域同步、模型下载、API 调用流量
运维成本	监控平台、调度系统、CI/CD 流水线

- **GPU 算力成本优化**
- **分层资源管理**
- **可观测性与优化工具**
- **模型复用与迁移学习**
- **构建 AI 成本治理平台**

GPU 算力成本优化

5.1

GPU 是生成式 AI 成本的核心部分，单块高性能 GPU 的投入远高于 CPU，因此**算力利用率**直接决定总体开销。不同阶段的 GPU 选型偏好会有差异

维度	训练阶段	推理阶段
算力需求	高并发、长时间运行、强一致性	实时性高、短延迟、高吞吐
成本敏感度	可接受较高单次开销	极度敏感，需单位推理成本最小化
资源利用率	容易出现空转（调试、失败重试）	易受突发流量冲击导致扩容浪费
优化重点	分布式训练效率、检查点管理	批处理、缓存、弹性扩缩容

常见优化策略包括：

混合实例部署

结合不同规格的 GPU/CPU 节点，在训练与推理任务间动态分配算力，提升资源匹配度。

按需或竞价实例

利用云上闲置算力资源运行非关键或具备容错能力的任务，能够显著降低总体投入。

共享集群

在团队内部或跨业务单元共享 GPU 集群，避免算力长期闲置，提高整体利用效率。

对于模型 API 调用场景，还可以通过以下方式降低推理成本：

模型版本选择

在保证业务效果的前提下，使用更轻量的模型版本，并结合提示工程优化与少样本学习技术弥补精度，通常能够减少大规模算力依赖。

批处理调用

将请求批量提交或在非高峰时段集中运行，提升吞吐率，降低单位推理成本。

上下文缓存

在多轮对话或长文本场景中缓存重复内容，避免重复推理消耗。

合并请求

将多条输入集中处理，减少重复 Token 消耗。

设计原则总结

弹性优先
使用可伸缩和可替代的算力组合，避免单一规格依赖。

API 优化
模型选择、批处理调用、上下文缓存与合并请求协同使用。

集群共享
通过共享算力池提升利用率。

分层使用
将关键任务与非关键任务分层运行，前者用稳定资源，后者用低成本资源。

调度优化
基于优先级和 SLA 动态分配算力。

分层资源管理 5.2

在 AI 生命周期中，数据和模型的存储方式存在差异：部分需要高频读写（如训练数据、推理缓存），部分仅用于归档和审计。为此，可以采用**冷热分层存储策略**：

- **高频数据放置在高性能存储系统，保障训练与推理的效率；**
- **低频数据归档到低成本存储中，以降低长期存储压力。**

在计费模式上，不同任务也需要区别对待：

- **稳定任务适合长期包周期模式，降低总体支出；**
- **实验性或突发任务更适合按需付费或竞价实例，减少资源浪费。**

此外，容器化和作业调度平台能够支持动态算力分配：根据任务负载自动扩缩容，空闲时释放资源，提升整体利用率。

设计原则总结

冷热分层
依据访问频率区分存储层次。

混合计费
长期任务用包周期，短期任务用按需或竞价。

弹性伸缩
按需伸缩，提高利用率。

SLA 优先
保障关键业务的性能与可靠性。

可观测性与优化工具

5.3

成本优化离不开**资源可观测性**。如果缺乏对 GPU 利用率、存储访问模式、带宽消耗等指标的实时监控，企业难以及时发现浪费。

通过统一的监控与日志分析平台，企业可以：

识别问题

发现 GPU 空转、数据冷热不均、网络瓶颈等情况；

策略优化

自动释放闲置算力、动态调整存储层次、优化推理服务并发配额；

成本透明化

让各业务团队清楚看到自身任务的消耗，推动内部责任归属和精细化治理；

能效管理

在绿色计算场景中，监控能耗与资源利用率，兼顾经济效益与可持续发展。

DESIGN PRINCIPLES

设计原则总结

全栈可观测

对算力、存储、网络进行全链路监控。

透明化

让成本与资源使用情况可视化。

优化闭环

结合监控数据实现自动调度与优化。

异常发现

快速定位空转或过度消耗资源的任务。

模型复用与迁移学习

5.4

全量训练一个大模型通常需要投入大量算力，而很多业务场景并不需要从零开始。通过迁移学习与模型复用，企业可以在已有预训练模型基础上进行微调，避免重复消耗资源。

进一步的优化手段包括模型小型化：

微调专属模型

根据特定场景定制小模型，减少推理时的 Token 消耗。

模型量化

将高精度权重转为低精度格式，显著压缩模型体积，提高推理速度。

模型蒸馏

通过小模型学习大模型的输出，实现高效替代，兼顾性能与成本。

模型剪枝

去除冗余参数，减少计算量，提升推理效率。

这些方法能够帮助企业在保持业务效果的同时，降低算力开销，加快模型迭代，形成良性循环。

DESIGN PRINCIPLES

设计原则总结

复用优先

优先基于已有模型进行微调，避免重复训练。

迁移学习

利用开源或商业模型库降低定制成本。

小型化

通过量化、蒸馏、剪枝提升推理效率。

资产共享

通过模型管理平台实现团队间的复用。

构建 AI 成本治理平台

5.5

当企业AI应用规模扩大，单纯依赖人工优化或零散工具已难以应对复杂的资源消耗问题。为实现可持续的成本治理，企业需要建设一个集资源调度、成本监控、自动化优化与组织协同于一体的AI成本治理平台，将成本管理从“被动响应”转向“主动治理”。

该平台不仅是技术基础设施的组成部分，更是连接财务、运维与AI研发团队的桥梁，推动形成“谁使用、谁负责”的精细化治理文化。一个完整的AI成本治理平台应包含以下核心模块：

资源调度

基于Kubernetes + Volcano、KubeFlow或自研调度器，支持GPU拓扑感知、混合精度任务调度、抢占式任务管理，实现训练与推理任务的统一调度与资源隔离。

成本计量

集成云厂商账单API（如阿里云Billing）、Prometheus监控数据与命名空间标签，按项目、团队、模型、任务维度进行资源消耗拆解，精确计算每项AI服务的单位成本（如“每千Token推理成本”）。

可观测性

提供全栈监控能力，涵盖：GPU利用率、显存占用与溢出情况、推理延迟与QPS波动、存储访问频率与冷热分布、支持异常检测与根因分析，快速定位“空转实例”或“低效训练任务”

自动化优化

基于策略引擎实现智能调度与资源回收，比如自动识别连续2小时GPU利用率低于10%的任务并告警或终止、推理服务夜间自动缩容至0、模型权重自动迁移至低频存储。

权限与配额控制

实现基于角色的资源访问控制，设置GPU资源申请审批流程，对高规格实例实行配额管理，防止资源滥用。

成本分摊与透明化

支持“Showback”（展示成本）与“Chargeback”（内部结算）机制，定期生成各团队资源消耗报告，并在内部团队中公开排名，推动责任归属与优化激励。

设计原则总结

DESIGN PRINCIPLES

可扩展架构

支持混合云环境接入，适配不同AI框架。

细粒度计量

支持按命名空间、标签等维度进行资源归属划分。

自动化优先

通过策略驱动实现“监控→分析→决策→执行”的闭环优化。

透明可追溯

所有资源使用行为可审计，成本数据可下钻至具体模型与负责人。

平台化统一

避免工具碎片化，整合调度、监控、计费于一体，降低运维复杂度。

06.

PERFORMANCE EFFICIENCY

性能

在生成式 AI 系统中，性能是决定用户体验与成本效率的核心因素。无论是大规模训练还是在线推理服务，都需要在算力利用率、数据流处理效率及响应延迟等多个维度实现最优平衡。性能优化不仅依赖底层硬件，还需要数据流架构、分布式训练框架、推理服务化设计与调度平台的全链路协同。阿里云在云原生基础设施、分布式存储与计算平台上的长期实践，已经形成了一整套可复用的性能优化方法，能够帮助企业实现在高负载与复杂场景下的稳定运行，同时兼顾资源效率与业务连续性。

- 高效的数据流与存储架构
- 分布式训练框架优化
- 大模型推理优化
- 大模型推理优化

高效的数据流与存储架构 6.1

在大模型训练和推理中，数据访问效率往往成为瓶颈。训练过程需要对海量样本进行高并发访问，推理过程则依赖快速的数据调度。

常见优化方法包括：

弹性扩展的对象存储与高并行文件系统

使用对象存储（如 AWS S3、阿里云 OSS）支持海量数据弹性扩展，并结合高并行文件系统（如 Lustre、Ceph）实现训练数据的高速读写，确保训练过程不会因 I/O 瓶颈而受阻。

云原生数据库

在结构化数据场景下提供高并发查询与低延迟访问，适合对话式 AI、推荐系统等应用。

冷热数据分层管理

高频数据存放在性能优先的存储中，低频数据归档在成本更低的层次，兼顾性能与开销。

近数据计算

将计算下沉到靠近数据的节点，减少网络传输开销，提高整体吞吐。

数据一致性校验

在多副本和分布式环境下保证数据一致性，避免因数据漂移或版本不一致导致训练偏差。

在金融风控场景中，低延迟查询和批量数据吞吐直接影响模型实时性；在医疗影像训练中，数据预处理与高速读取是保障模型精度的前提；在互联网推荐中，近实时的日志流计算决定用户体验。

设计原则总结

高吞吐
存储架构需支持并发读写，满足分布式训练需求。

分层存储
冷热数据分层以平衡性能与成本。

近数据计算
尽量让计算靠近数据，降低延迟。

一致性保障
保证多副本、多节点环境下的数据一致性。

分布式训练框架优化

6.2

大模型训练几乎必然依赖分布式框架。但在实践中，通信效率、调度策略和资源利用率往往成为主要瓶颈。

常见优化路径包括：

同步与异步训练策略

根据模型规模与网络环境选择参数服务器模式、全局同步或混合模式，平衡收敛速度与资源利用率。

通信优化

梯度压缩、通信合并、分层同步等技术，减少通信带宽消耗。

高效互联

优化节点间的网络传输，降低分布式通信延迟。

弹性训练

支持节点的动态加入或退出，确保在资源波动时任务仍能继续。

Profiling 驱动的调优

通过分布式 Profiling 工具链量化通信延迟、同步效率与算子性能瓶颈，为迭代优化提供依据。

在大规模分布式场景中，良好的调度机制能够避免部分节点成为性能瓶颈，从而提升整体训练效率。

DESIGN PRINCIPLES

设计原则总结

通信优化
采用压缩与合并策略降低通信开销。

弹性训练
支持节点动态扩展，保证任务不中断。

框架适配
依据模型规模与任务需求选择合适的策略。

性能评估
通过基准测试与 Profiling 验证扩展性。

大模型推理优化

6.3

推理阶段直接影响用户体验和业务 SLA，目标是在降低延迟与成本的同时保持输出质量。优化需从**模型设计、硬件加速、服务架构**三个维度协同实现。

常见优化策略：

模型轻量化

通过蒸馏、量化、剪枝等方式减少模型规模，提高推理速度。

硬件加速

充分利用 GPU、NPU 等加速芯片，实现低延迟推理。

服务化管理

以平台化方式管理推理任务，支持多租户隔离、弹性伸缩和负载均衡。

架构优化

通过上下文缓存（KV Cache）、输入计算与生成解耦调度，减少重复计算，提高响应速度。

动态批处理与流批混合推理

在低延迟和高吞吐之间取得平衡，适应不同业务场景。

在交互式对话、搜索增强生成等场景中，上述方法能够有效降低响应延迟，并提升用户体验。

DESIGN PRINCIPLES

设计原则总结

轻量化
通过压缩技术减少模型规模。

硬件加速
利用专用硬件降低延迟。

服务化
大模型推理以服务平台方式统一管理和伸缩。

架构优化
采用缓存与分离策略提升效率。

SLA 驱动
以业务目标为核心优化资源分配。

智能调度与算力优化

6.4

随着 AI 任务规模与复杂度提升，算力调度成为关键环节。智能算力编排平台需要对 GPU、CPU、NPU 等异构资源进行统一调度，依据任务优先级、资源利用率和 SLA 动态分配，并支持自动扩缩容与任务迁移，保障整体性能。

核心能力包括：

统一编排

不同算力资源池的统一调度与隔离管理。

智能分配

基于优先级、任务特性和 SLA 动态分配算力。

弹性伸缩

根据实时负载波动自动扩展或收缩，避免资源浪费。

跨层 Profiling

从硬件层到应用层的全栈性能分析，识别瓶颈。

架构级优化

通过算子融合、注意力机制优化、稀疏激活和专家模型等方法提升效率。

在多任务、多租户环境下，调度平台还能保障关键业务的性能不受影响，并为非关键任务提供灵活调度，从而提升整体集群效率。

设计原则总结

统一编排
平台化调度异构算力资源。

动态资源分配
基于优先级与 SLA 动态分配算力。

弹性伸缩
按负载波动自动扩缩容。

Profiling 驱动
通过全栈分析识别性能瓶颈。

资源最大化
在集群层面实现算力高效利用。

07. 结束语

生成式 AI 正在深刻改变企业的业务模式与技术架构。从数据安全到算力保障，从模型供应链到 Responsible AI，再到稳定性、效率、成本与性能，阿里云卓越架构在生成式 AI 场景下进行了系统性延展与升级，为企业提供了覆盖全生命周期的参考框架与最佳实践。

可以看到，卓越架构的**五大支柱（安全、稳定、效率、成本、性能）**在 AI 时代依然适用，并在大模型、隐私合规、推理优化和智能调度等维度上得到进一步深化。这不仅帮助企业在复杂多变的技术环境中降低风险、提升效能，也为生成式 AI 的规模化落地奠定了坚实基础。

未来，随着模型规模的不断提升、AI 与业务深度融合以及全球合规环境的持续演进，企业对架构治理的要求将进一步提高。阿里云将持续在**AI原生基础设施、基础大模型能力、以及围绕AI的工具和产品**等方面继续投入，帮助企业更好的拥抱 AI。

我们相信，生成式 AI 的发展不仅是技术的跃迁，更是企业数字化能力的再次升级。通过阿里云卓越架构方法论，企业可以从“用好云”走向“用好 AI”，真正实现从**云卓越（Well-Architected）**向 **AI 卓越**的演进。

生成式AI卓越架构设计指导原则