QCon
全球软件开发大会

InfoQ 极客传媒

重塑可观测边界：小红书在大模型时代的稳定性工程实践

王亚普

小红书可观测团队负责人

目录

01

小红书可观测现状以及 AI 时代面临的挑战

小红书可观测体系在稳定性工程中的定位

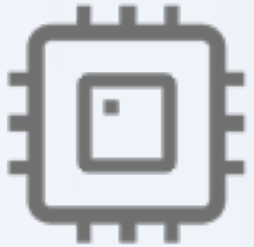
降发生

降影响

小红书可观测在 AI 时代的变化

- 观测对象变化
- 系统关系变化
- 运行行为变化
- 表达方式变化

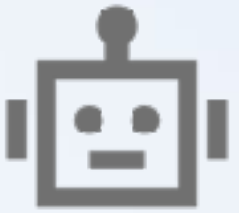
可观测边界变化



GPU 集群

大模型

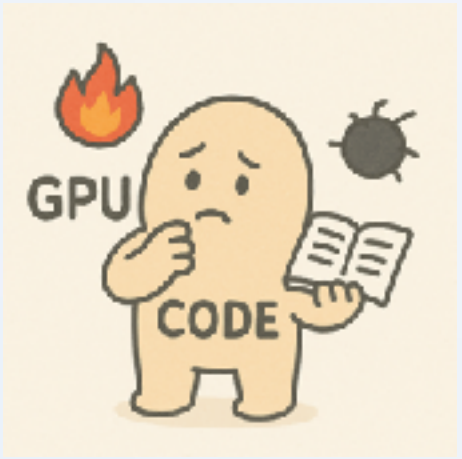
训练/推理



Agent

小红书可观测在 AI 时代面临的挑战

黑盒困局加剧



软硬件的问题多样性

观测对象变化带来的复杂性

故障放大效应

为什么需要 AI 可观测

- AI 训练稳定性管理已经成为智能时代的精密工程，尤其是在千卡甚至万卡以上规模，整个任务会发生各种故障，导致资源利用率不高或任务中断。

- 基础设施交付和运行中的质量保障是稳定性基础，GPU 高昂的成本使得业务对 GPU 交付质量、节点故障感知、处置时效性都要非常高的要求。

- AI 应用需兼顾可用性、性能与效果体验的多重质量目标，使得业务连续性保障面临更大挑战。

02

AI Infra 可观测：面向训推服务的稳定性建设

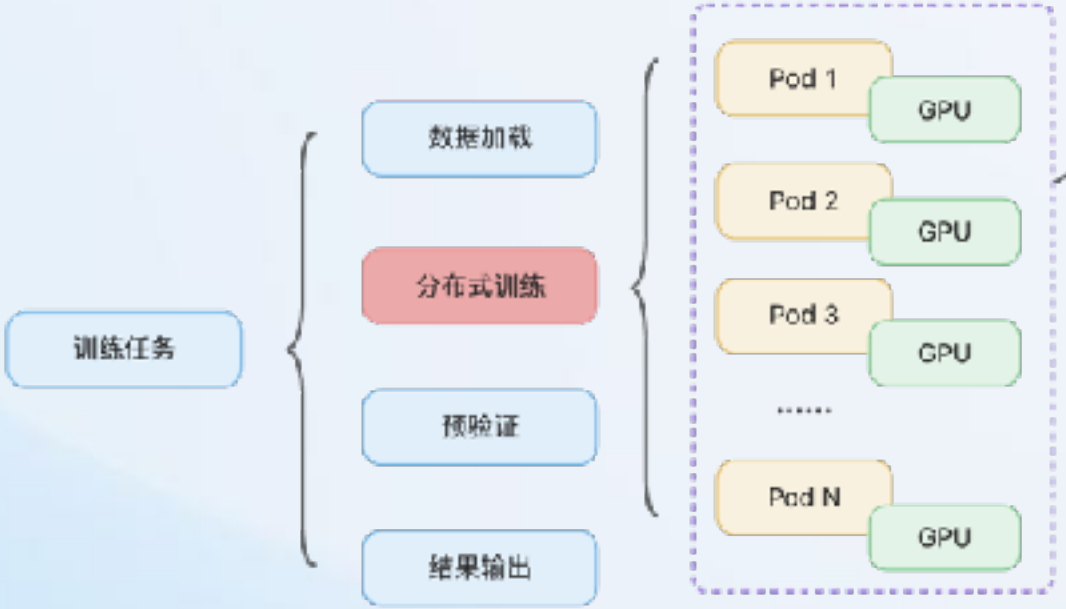
业务痛点

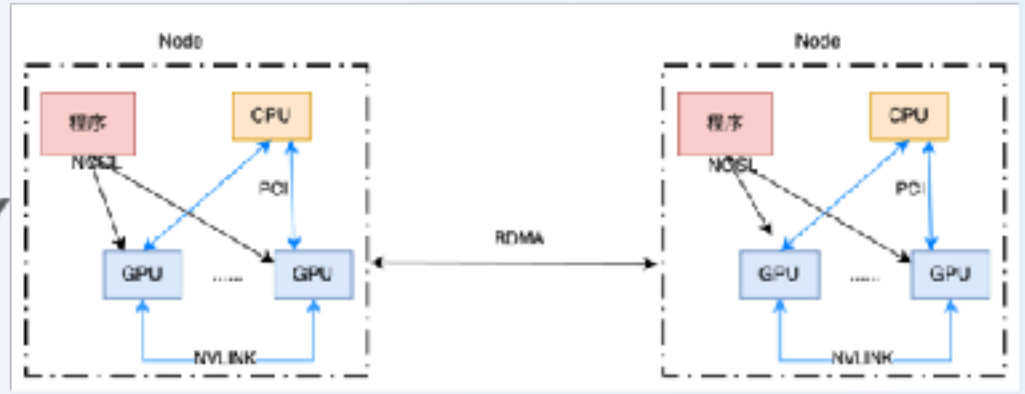
故障率高

环境异构挑战

故障发现与定位难

训练任务痛点问题分析



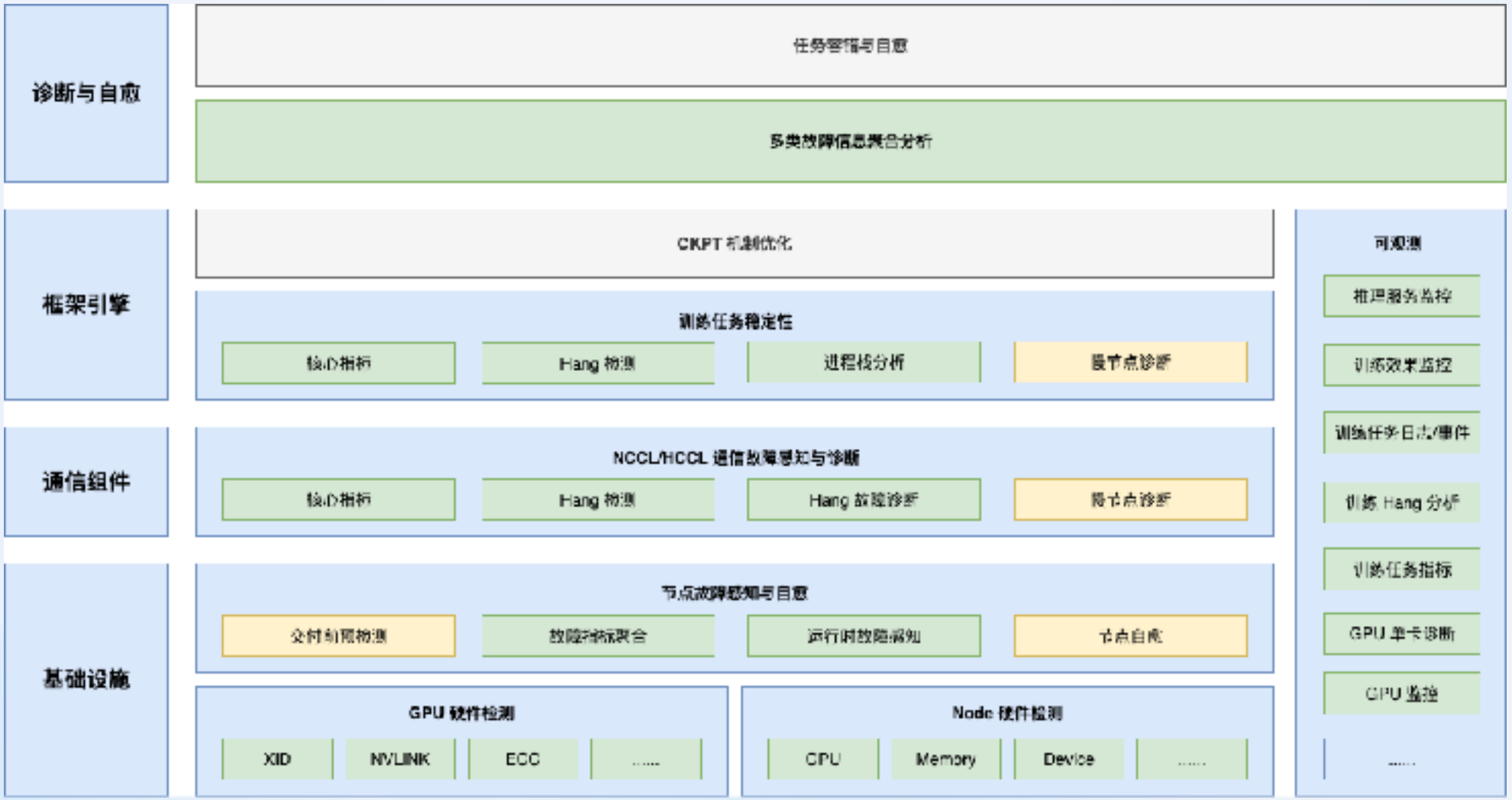


MFU = Model FLOPs Per Iteration / （GPU 单卡算力 * 卡数 * 一次迭代时间）

减少训练受影响的时间

训练的过程复杂，整个训练过程如果存在一个 GPU 卡异常，整个训练任务就会失败

整体建设思路


诊断与自愈
任务管理与自愈
多类故障信息聚合分析
框架引擎
CKPT 机制优化
训练任务稳定性
核心指标
Hang 检测
进程栈分析
慢节点诊断
通信组件
NCCL/HCCL 通信故障感知与诊断
核心指标
Hang 检测
Hang 故障诊断
慢节点诊断
基础设施
节点故障感知与自愈
交付前预检测
故障指标聚合
运行时故障感知
节点自愈
GPU 软件检测
XID
NVLINK
ECC
.....
Node 硬件检测
CPU
Memory
Device
.....
可观测
推理服务监控
训练效果监控
训练任务日志/事件
训练 Hang 分析
训练任务指标
GPU 单卡诊断
CPU 监控
.....

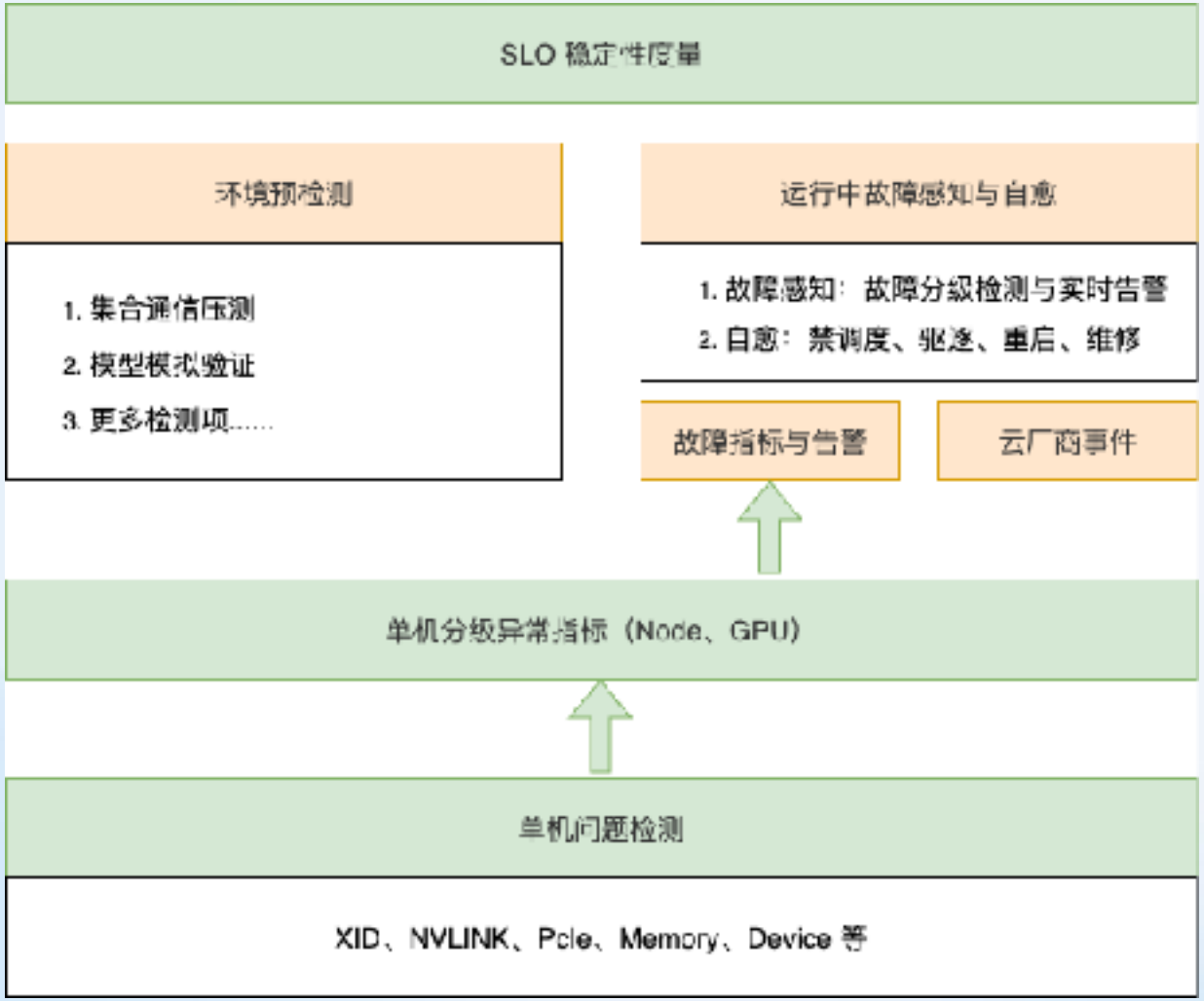
2-1

GPU 资源稳定性保障

GPU 故障等级定义

故障等级	对 GPU 的影响	示例
1 Notice	不影响正常运行	如 Corrected ECC Error、Pending Recovery
2 Warning	可以使用，存在一定风险	如 Uncorrected ECC Error 累计值高
3 Error	无法使用，需要重启恢复	如 SRAM UCE、掉卡、驱动卡死
4 Fatal	无法使用，需要替换/维修	如 Row Remapping Failure

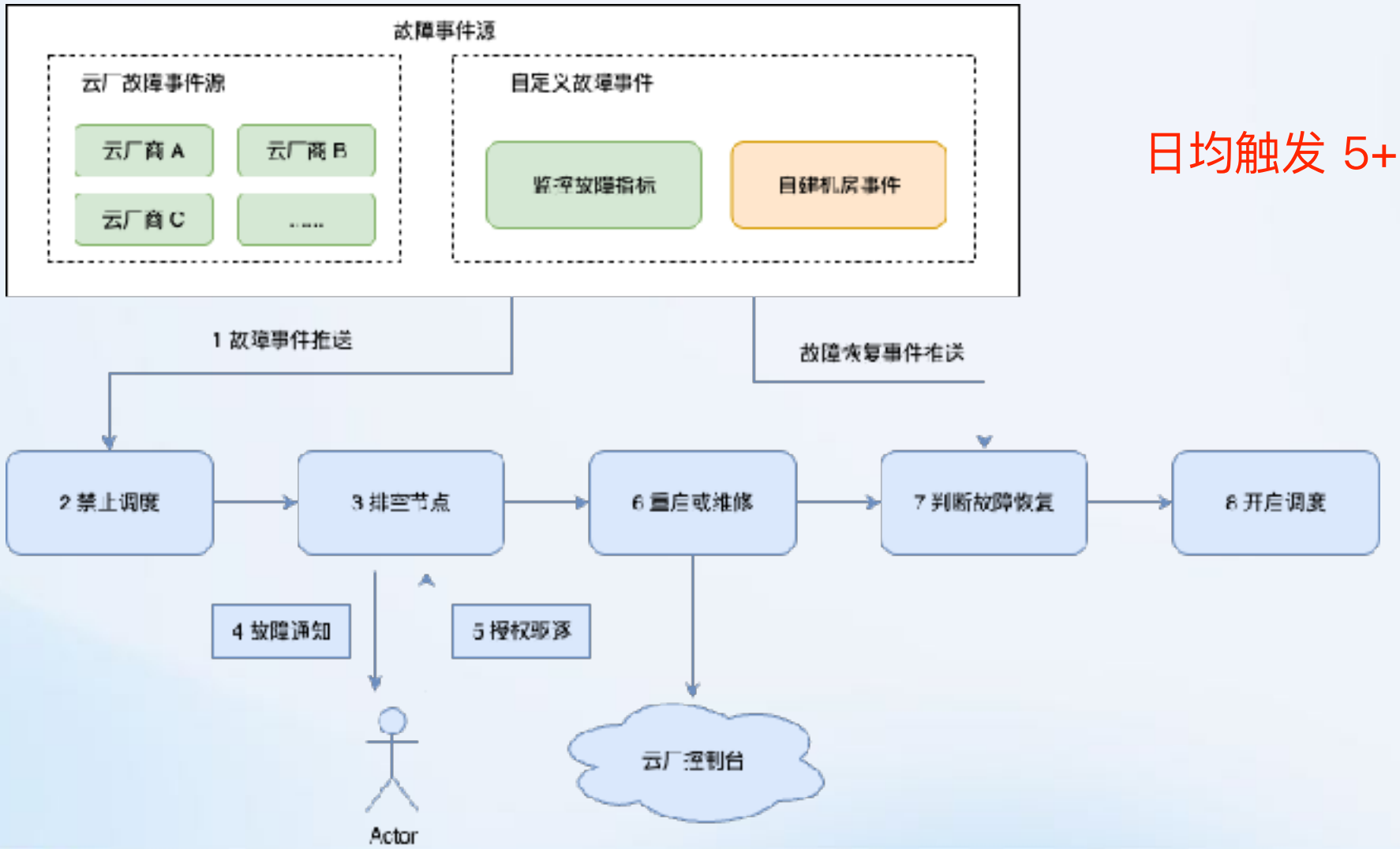
GPU 稳定运营与故障感知



场景支撑

整体异常分析	单节点问题排障
GPU 可用性 SLO	节点/GPU 可用性状态
多维异常度量	单卡诊断/监控下钻

GPU 节点自愈



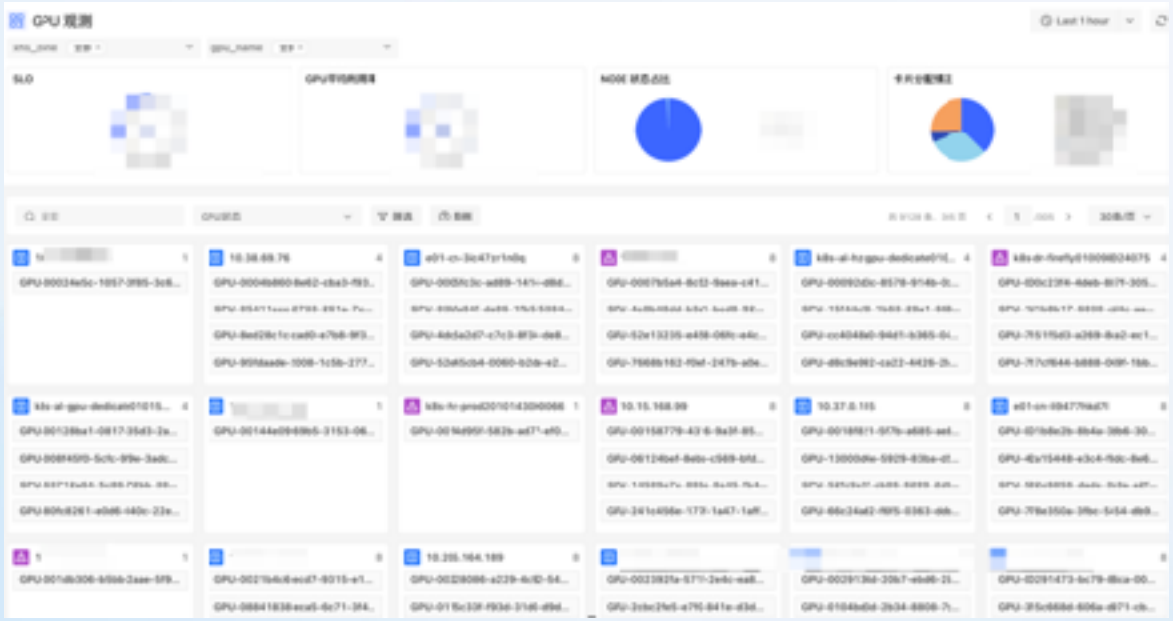
日均触发 5+ 自愈流程

GPU 观测落地成果与最佳实践

提供 GPU 全局监控能力以及单机故障诊断能力，支持 20+ 故障指标识别，

可解决 80%+ GPU 硬件故障诊断定位。

整体异常分析



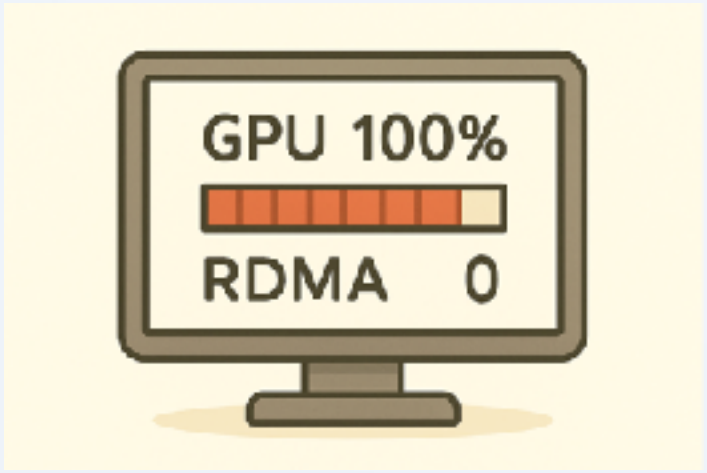
单节点问题排障

2-2

训练任务 Hang 的发现与故障定位

训练任务发生 Hang 可能的表现

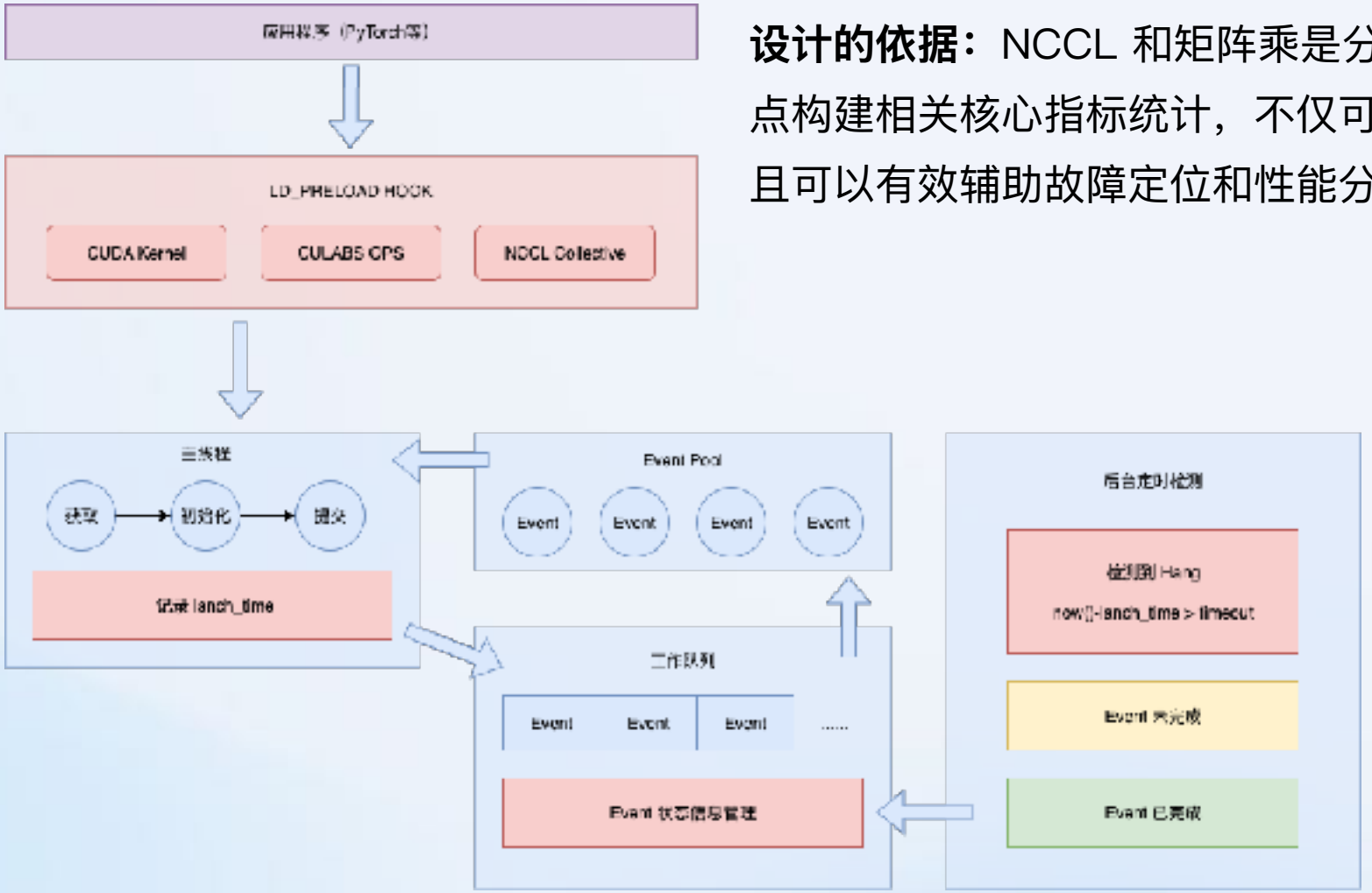
所有任务程序日志不输出



指标突变并持续维持某个状态

进程调用栈不再变化

问题分析与设计思路



设计的依据：NCCL 和矩阵乘是分布式训练的核心操作，以此为锚点构建相关核心指标统计，不仅可以实现较精确的 Hang 检测，而且可以有效辅助故障定位和性能分析。

判定 Hang 的核心思路：对矩阵计算、NCCL 调用等核心函数进行 Hook 拦截，监控对应的操作记录；当事件超过指定的超时时间未拿到返回结果，判断为 Hang。

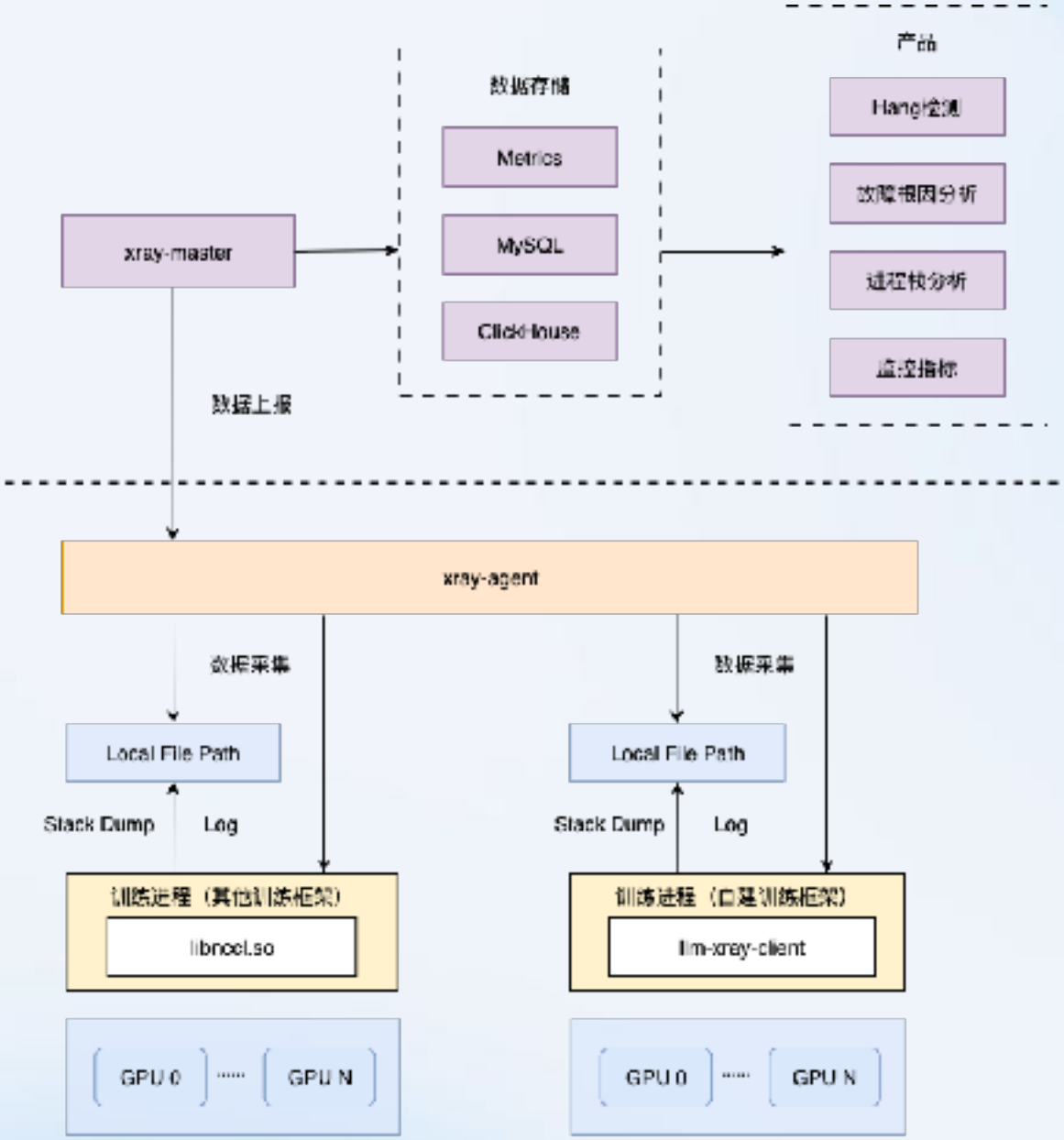
整体技术方案

技术方案差异化选型：

- 自建训练框架：xray-llm-client 低侵入方式，提供更全面的功能
- 其他训练框架：仅提供 NCCL 层面的监控和检测能力，与框架无关

轻依赖的数据采集：

- 保持 xray-llm-client 足够轻量，复杂功能尽可能上移
- Unix Domin Socket
- DaemonSet/HostPID/进程PID映射关系



故障定位的核心思路（基础场景）

故障表现：

- **受影响节点：**表现基本一致
- **故障节点：**与其他节点有明显表现差异

通过对比差异快速定位故障源头

指标诊断

正常Rank	故障Rank
100%	0%
GPU利用率	GPU利用率

xray-llm-client： Hook Cublas & NCCL，生成Hang状态指标

Cublas　NCCL　指标缺失检测

训练进程调用栈诊断流程

栈采集&存储
xray-agent采集所有rank栈信息

→

栈分类&聚合
识别barrier等待通信等待状态

→

异常识别
个别rank卡在其他函数上

→

火焰图展示
直观展示栈类别差异和占比

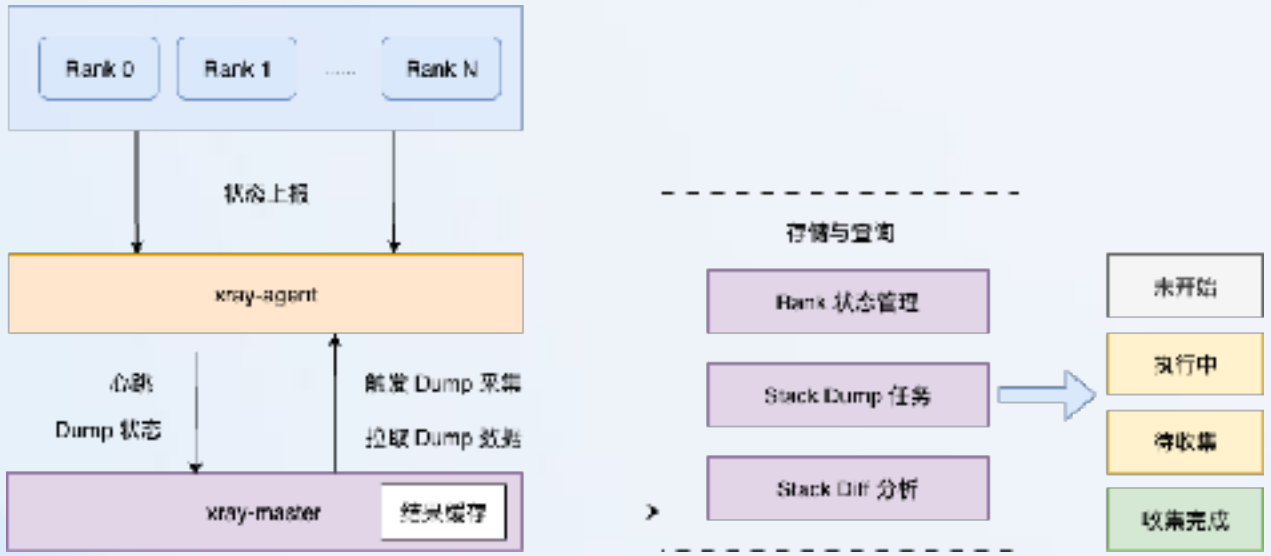
正常Rank：
barrier等待 | 通信等待

故障Rank：
卡在特定函数（Hang源头）

分析方法：
栈数据对比 → 确定根因

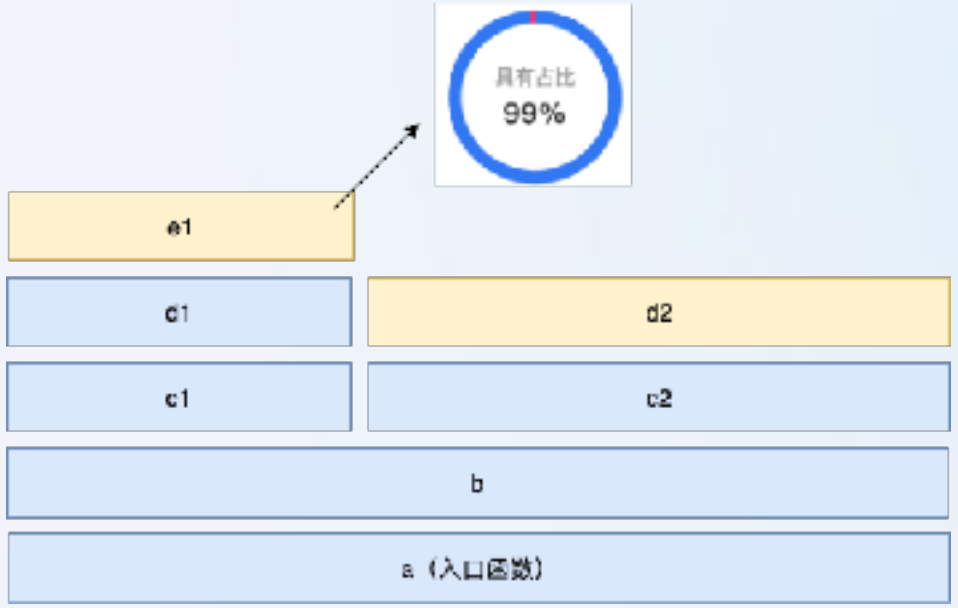
进程栈采集与聚合

栈采集



只要有一个 rank hang，认为这个训练任务 hang 住，
触发一次 stack dump

栈聚合



- 相同前缀、线程名聚合
- 寻找分叉点

故障定位的核心思路（复杂场景）

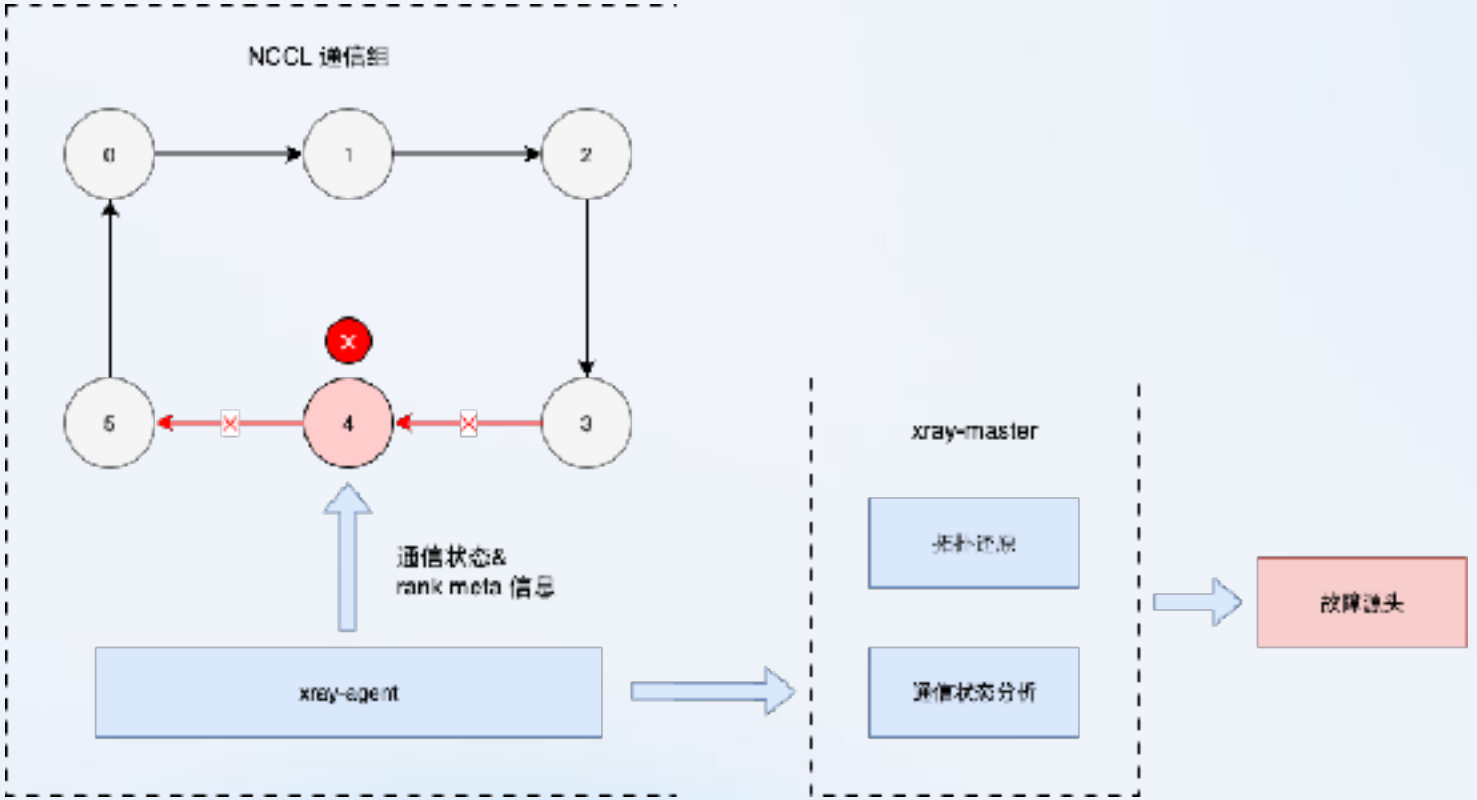
故障源头 rank 和其他 rank，在 NCCL 的通信状态上有差异

故障表现

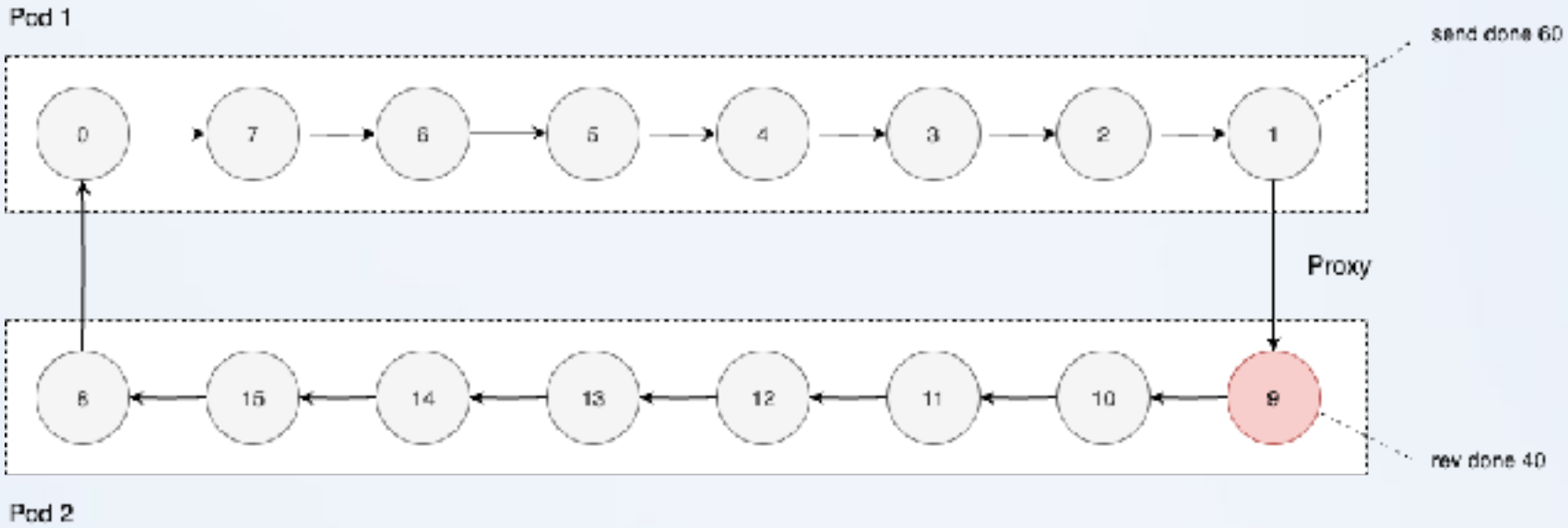
❌ 所有节点表现一致

- GPU全部打满 100%
- 无法从指标区分
- 应用进程栈无差异

⚡ 常规手段失效，需要深入NCCL层面分析



NCCL 网络通信故障的定位与拓扑还原



Send: posted->transmitted->done

Rev: done<-transmitted<-Received<-posted

故障1: done < transmitted

状态：Sending on network

含义：GPU已生产数据，网络已提交WR，但发送尚未完成

Network (RDMA) 故障

故障2: transmitted < posted

状态：Waiting on GPU

含义：CPU准备好了buffer，但GPU还没提供数据

GPU Device 故障

故障1: received < posted

状态：Receiving from network

含义：CPU准备好buffer，但数据还在网络传输中

Network 故障

故障2: done < transmitted

状态：Waiting on GPU

含义：已flush完成，确保GPU收到数据，但GPU处理延迟

GPU Device 故障

业务实战案例分享：监控指标

训练总览

NCCL 通信指标

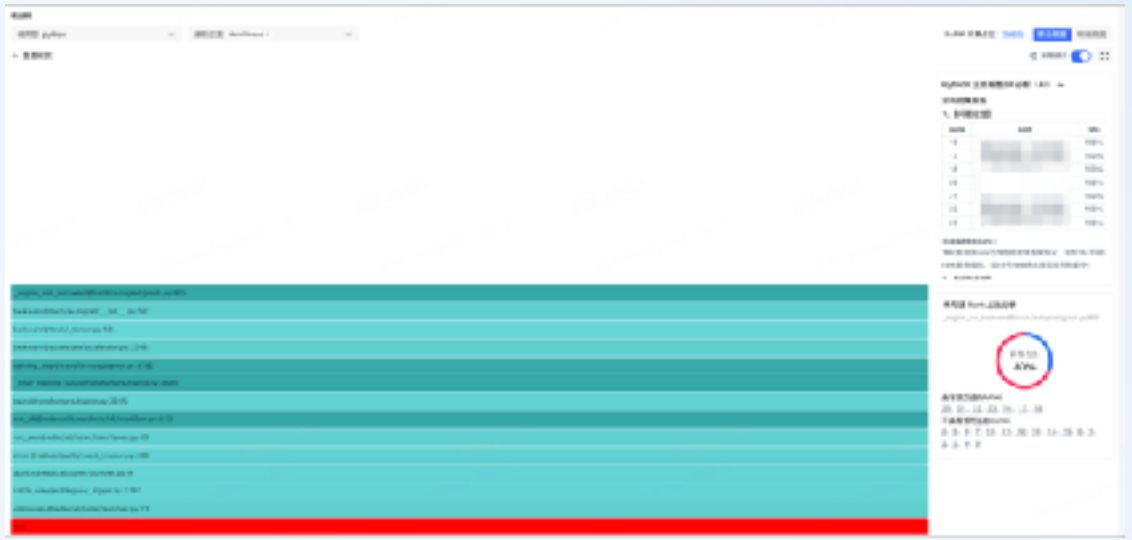
基础指标

矩阵乘指标

业务实战案例分享：智能诊断报告

诊断报告

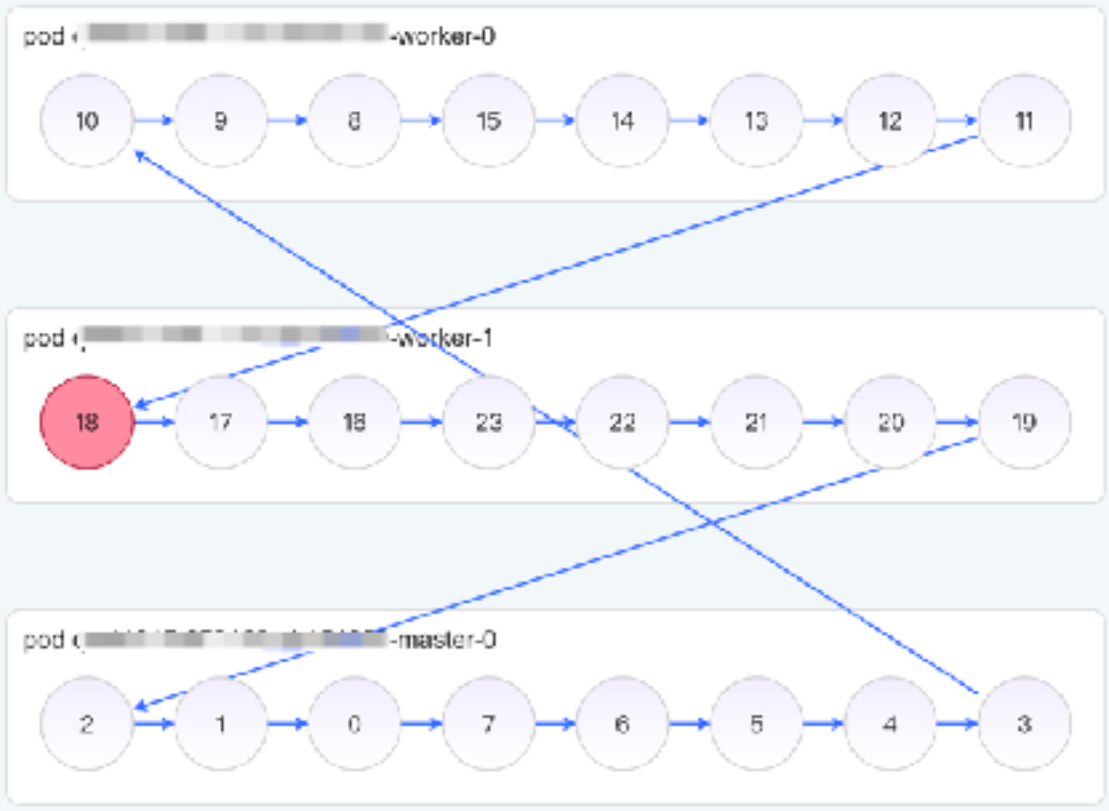
栈分析



业务实战案例分享：NCCL 网络通信故障定位

tree

ring



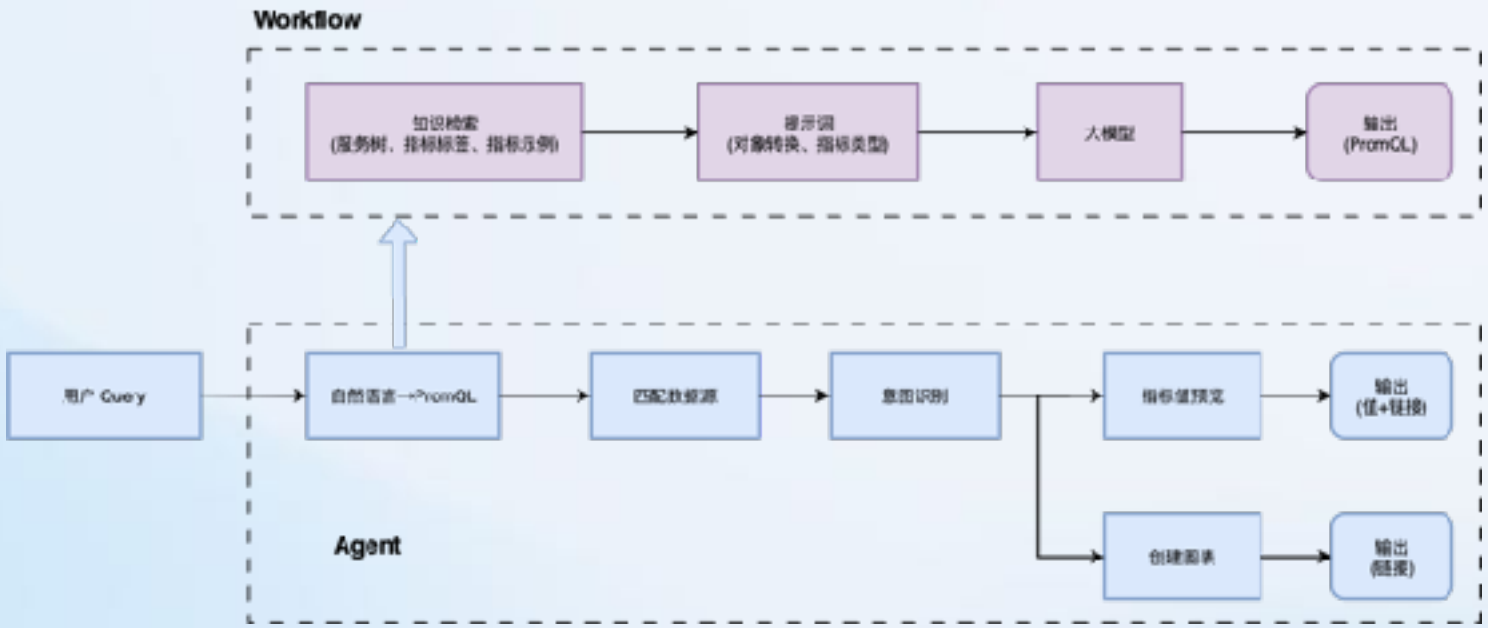
03

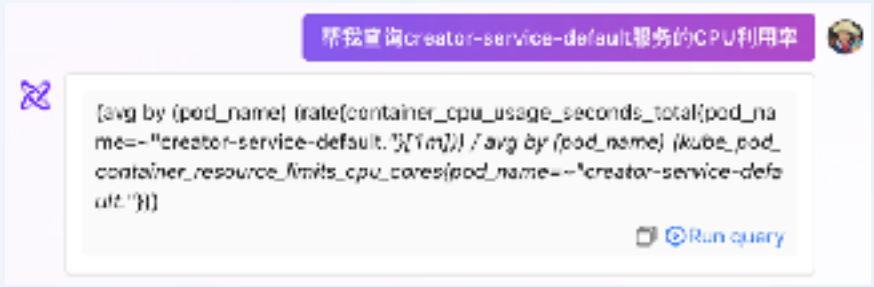
面向稳定性提效的 AI Agent 场景建设与探索

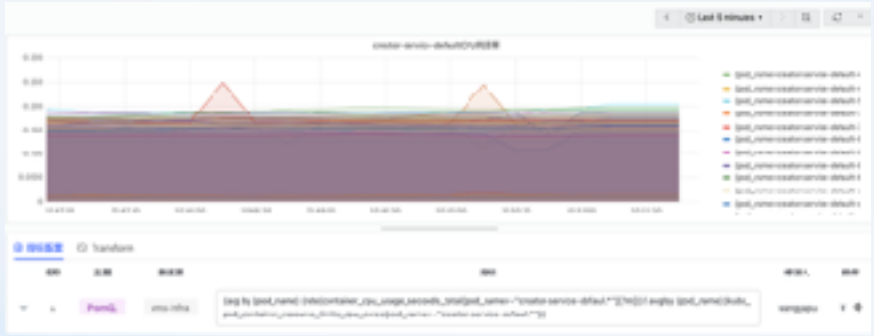
从可观测 AI 助手开始

XPilot Ai

自然语言&配置自动化 → 数据分析&根因诊断 → 最佳实践 Workflow → 业务自定义场景







产品定位与目标

以 AI 驱动协助研发与 SRE 完成稳定性全生命周期工作，提升用户效率和体验，降低使用门槛

XPilot AI 稳定性工作台

XPilot AI 助手

风险前（预防与加固）	风险中（应急响应）	风险后（复盘与改进）
强弱依赖关系	故障定位	复盘报告
容量规划	影响面分析	知识沉淀与答疑
性能劣化	应急协作	预案验证
服务画像	故障自愈	业务场景数据度量

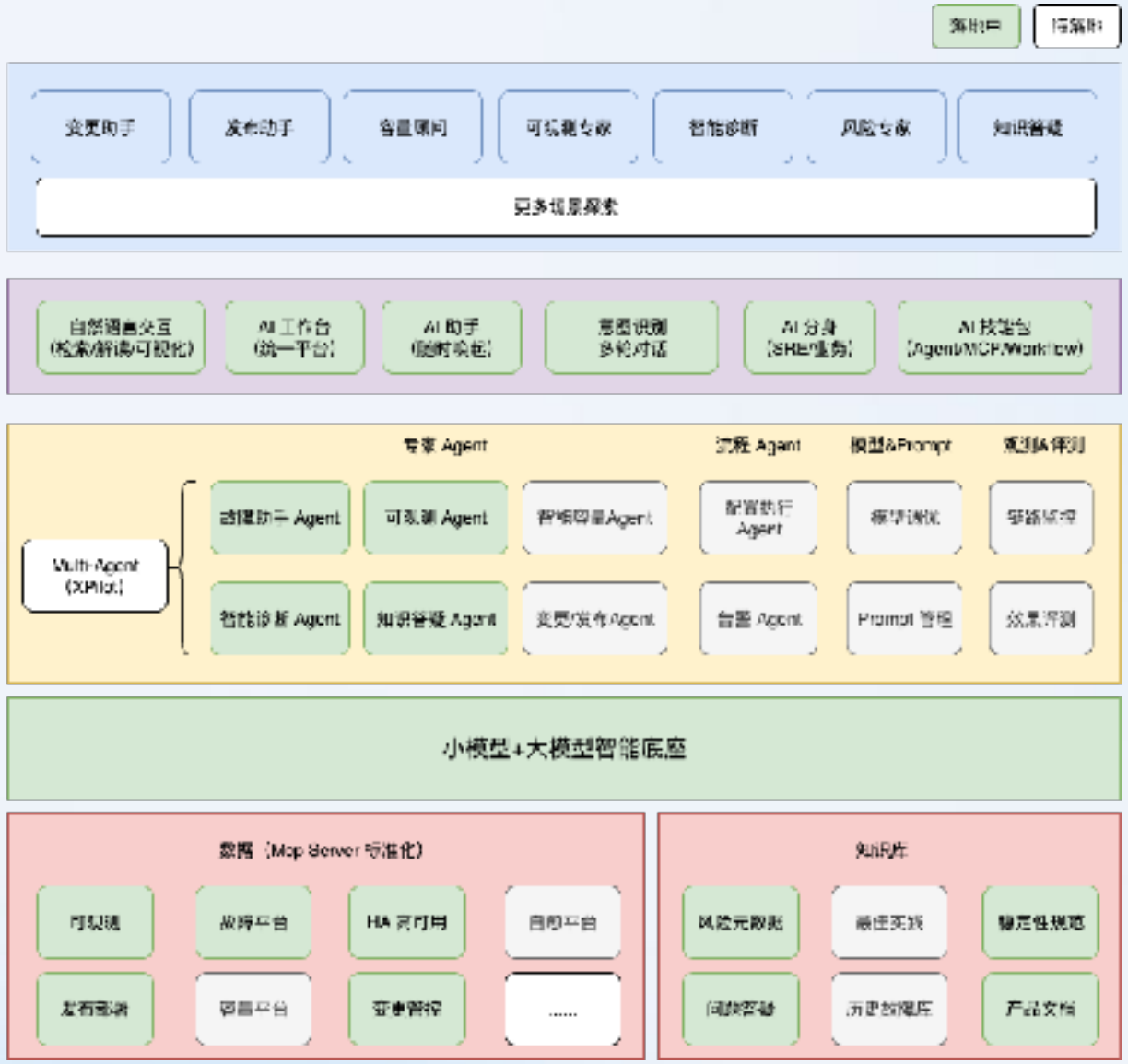
稳定性 AI 工作台产品大图

角色驱动/高频通用

统一交互/技能封装

团队协同/自主规划与协作

数据标准化与开放



大模型的优势与劣势（现阶段）

大模型适合做什么	大模型不适合做什么
语言理解与生成	精确计算任务
知识整合与逻辑推理	大数据量处理
语义推理与解释	多维聚合与下钻
启发式或探索式任务	用户友好的可视化表达

基建角度：需要解决的核心问题

- 原始数据量大，大模型缺乏高效的数据处理能力，如可观测领域
- 跨团队之间的数据交互问题，各团队 Agent 建设处于百花齐放的阶段
- 缺乏对数据上下文的记忆管理能力，无法提供很好的自由对话效果
- 缺少工具动手能力以及用户友好的可视化解释方式

传统 AIOps 的故障定位能力

局限性

扩展性有限

专家经验无法复用

可解释性和用户交互不友好

大模型和小模型协同发展

小模型	大模型	AI 时代在 OPS 方向带来的改变
特征提取	理解和推理	构建智能运维核心
数据压缩	可编程	生成式 AI 改善产品交互
根因诊断	可解释	提升产品易用性和多样性
关联分析	专家经验学习	小模型+LLM+Agent 增强产品竞争力

准确性提升

学习+效率提升

智能诊断底座

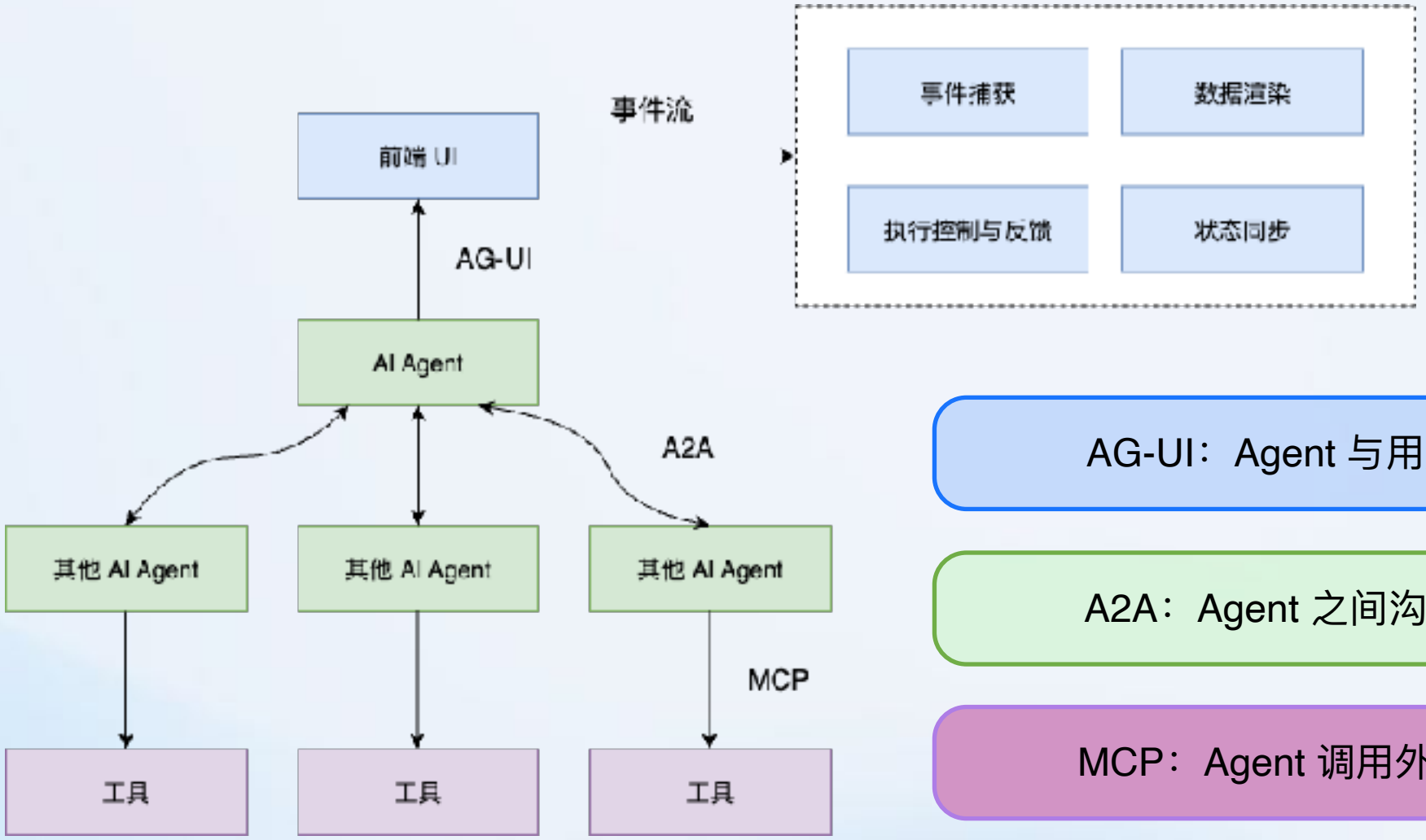
传统 AI 分析与 LLM 深度结合

灵活可插拔的工具调用能力

可复用的智能诊断 Agent

赋能多个技术平台产品

Agent 协议“三剑客”



AG-UI：Agent 与用户沟通的桥梁

A2A：Agent 之间沟通协作的标准

MCP：Agent 调用外部工具的标准

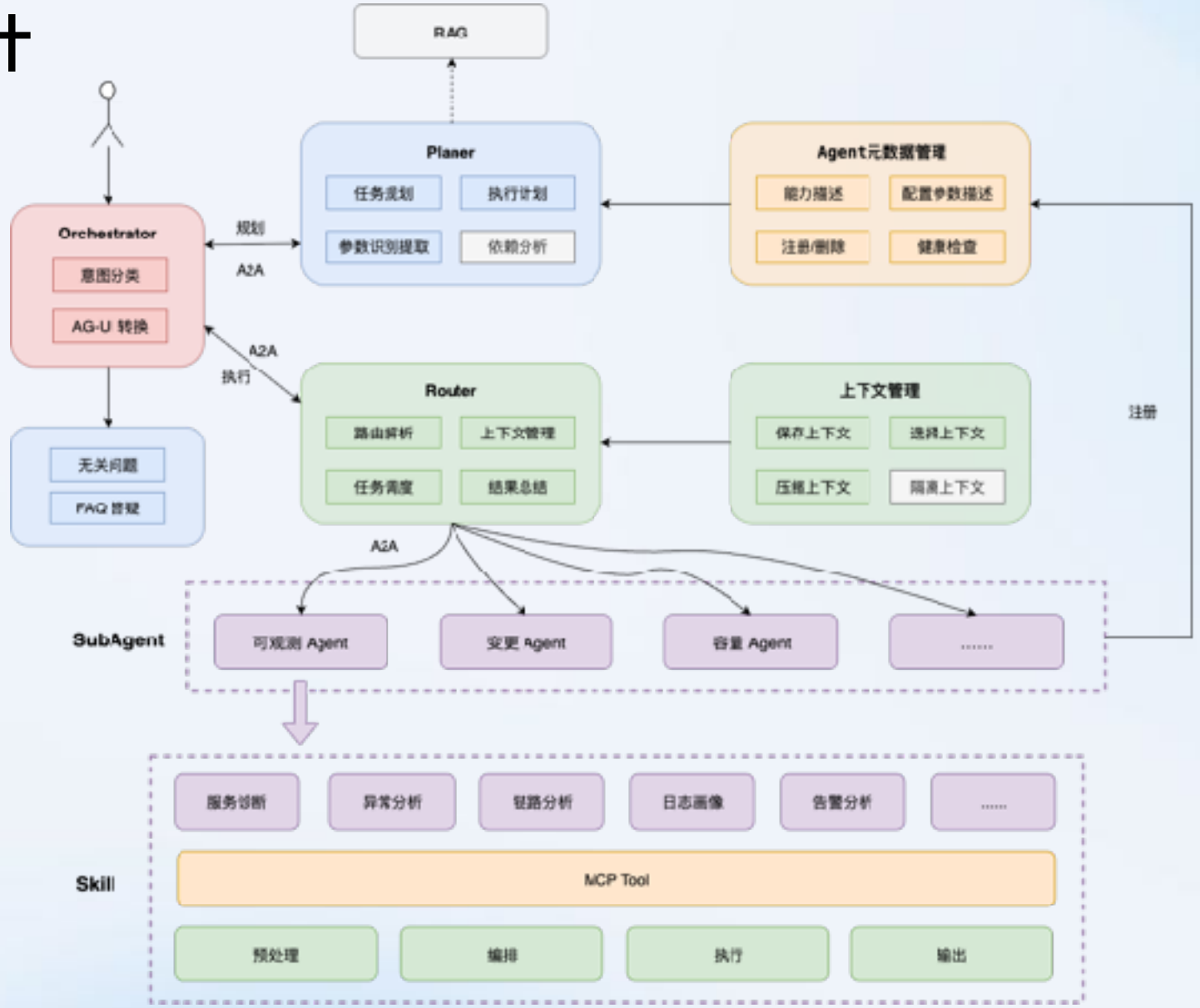
Multi Agent 技术架构设计

中央集权

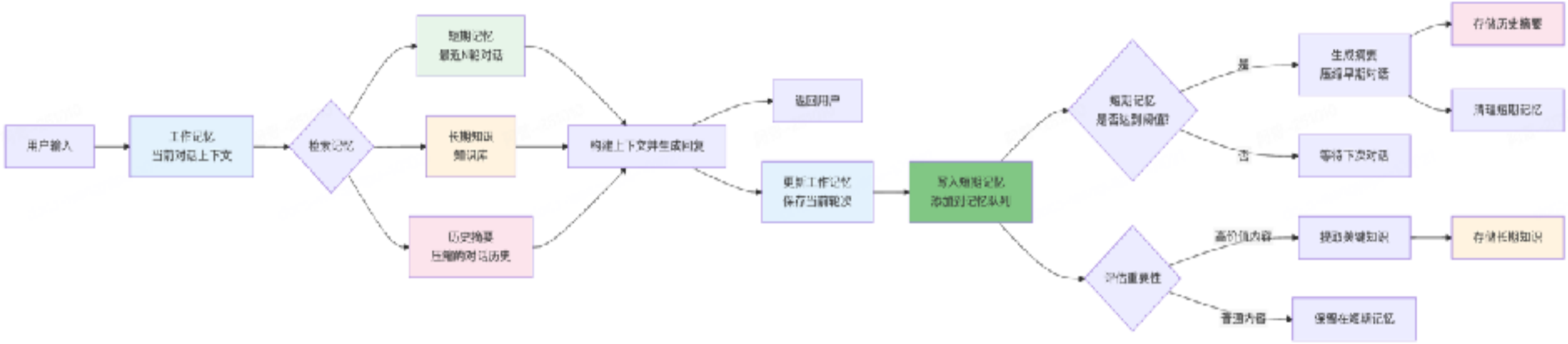
分层委派

分而治之

模块化管理



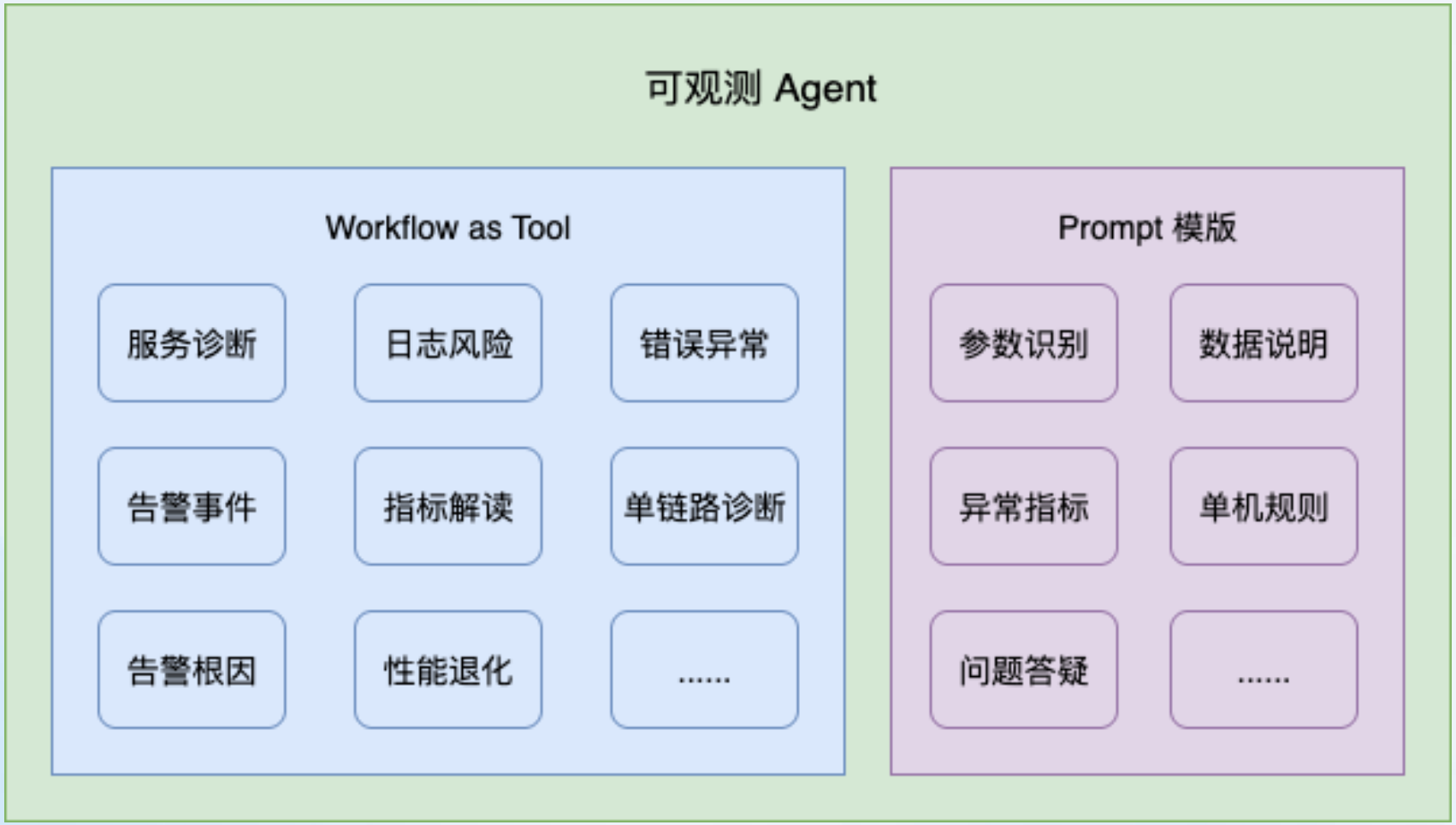
多层次记忆模块设计



记忆的价值不在于量，而在于信息价值的最大化

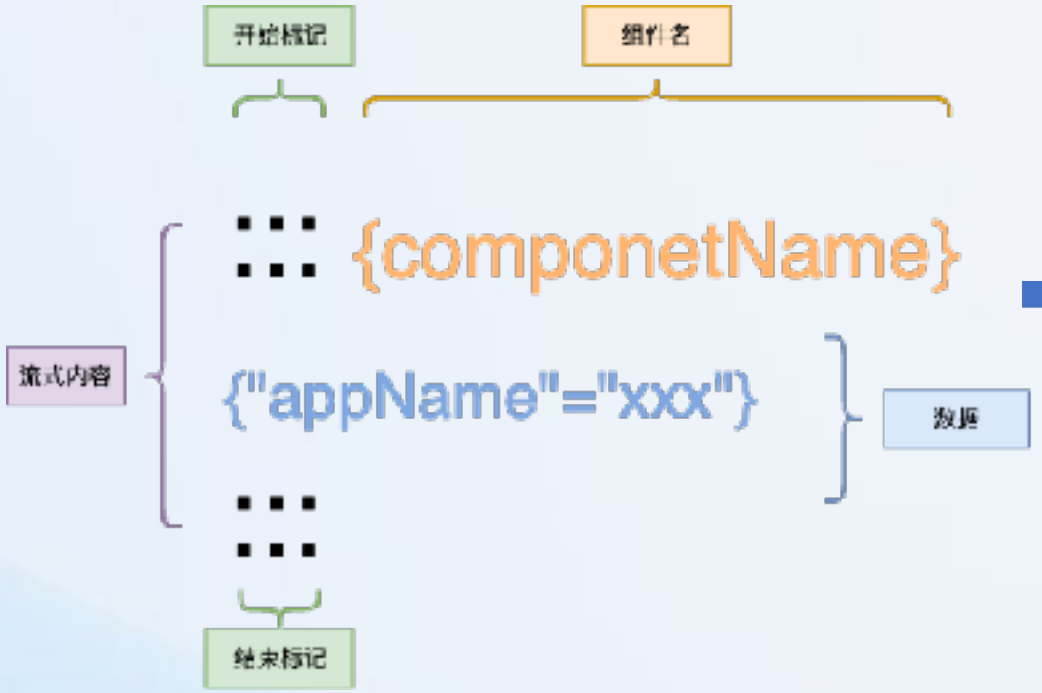
工作记忆 | 短期记忆 | 长期知识 | 历史摘要

Flow Engineering



- 复杂任务：拆解，LLM + function 组合，推荐 LangGraph
- 模型在一个受控的、多步骤的流程里工作，而不是一次性“黑箱”输出
- 简单任务：配置文件，write prompt，not code
- prompt 按场景抽象可复用模板

交互与输出范式



参数确认

指标诊断

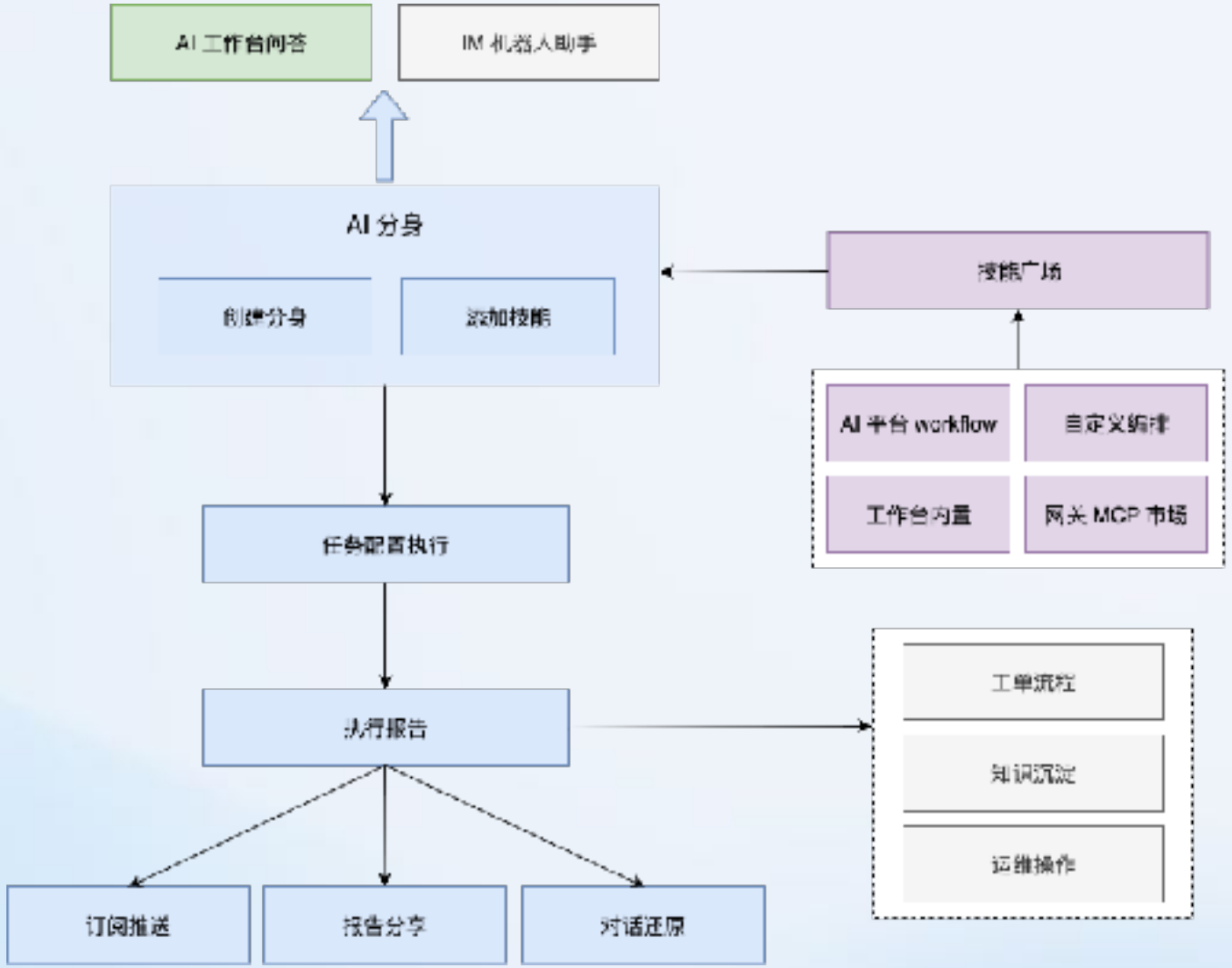
思考过程

工具调用展示

数据下载

……

业务提效赋能：AI 分身



专家经验转化为可复用 Agent

7*24 小时的专属分身

自然语言+工具构建个人技能

日常工作助手

最佳实践分享：风险治理

日志风险运营

风险工单分析

最佳实践分享：诊断分析

服务诊断分析

异常告警根因分析

最佳实践分享：AI 分身

技能与任务管理

工作日志

未来规划

RL + Observability

Observability for RL

- 行为与轨迹观测
- 长尾 Request 分析
- 分布式 Profiling

双向赋能

RL for Observability

- 智能告警
- 异常检测（专家规则）
- 时序理解与推理

未来规划

AI 时代
Ops 新的范式会出现

QCon
全球软件开发大会

InfoQ 极客传媒

THANKS

大模型正在重新定义软件

Large Language Model Is Redefining The Software