AGENT SYSTEM ARCHITECTURE · 2026

Agent 全链路技术架构

从 GPU 集群到 Token 输出：模型、推理、Harness 与数据飞轮的完整拼图

适用：Coding Agent · Computer Use · Deep Research · 企业自动化

9 LAYERS
评测 → 观测

100 GITHUB REPOS
各阶段代表项目

1 FLYWHEEL
trace → retrain

00 评测 · 01 算力 · 02 训练 · 03 模型 · 04 推理 · 05 Harness · 06 平台 · 07 安全 · 08 观测

9 LAYERS · 100 REPOS · 1 FLYWHEEL

00 评测与任务定义

先定义“什么算完成”，再谈模型

- **任务域**
  Coding · 终端 · 浏览器 · 客服
  Deep Research · 数据分析
- **公开基准**
  SWE-bench Verified · Terminal-Bench
  OSWorld · WebArena · GAIA · τ-bench
- **私有 Eval**
  黄金集 · 回归集 · 对抗集
  线上失败回流 → 新用例
- **核心指标**
  任务成功率 · 步数 · $/task
  P95 时延 · 人工接管率
- **发布门禁**
  离线回归 + 在线 canary
  A/B · 自动回滚
- **评测驱动开发**
  先写 eval，再写 prompt / 训模型
  每个失败都变成测试用例

GITHUB REPOS · 8 REPOS

- openai/evals
- EleutherAI/lm-evaluation-harness
- UKGovernmentBEIS/inspect_ai
- SWE-bench/SWE-bench
- web-arena-x/webarena
- sierra-research/tau-bench
- THUDM/AgentBench
- huggingface/lighteval

01 算力与集群

稳定、可调度、可恢复的算力底座

- **GPU 节点**
  H100 / H200 / B200 / GB200
  NVLink / NVSwitch · 液冷
- **集群网络**
  InfiniBand NDR/XDR · RoCEv2
  rail-optimized · 无损 · 拥塞控制
- **存储**
  Lustre / WEKA / GPFS 并行文件系统
  对象存储 · checkpoint TB/s 写入
- **调度**
  Slurm / K8s + Volcano / Kueue
  gang scheduling · 拓扑感知 · 配额
- **可靠性**
  NCCL watchdog · XID/ECC 监控
  静默错误检测 · 异步 ckpt · 弹性训练
- **三池隔离**
  预训练 / RL rollout / 在线推理
  电力 · 散热 · 容量规划

GITHUB REPOS · 8 REPOS

- NVIDIA/nccl
- NVIDIA/gpu-operator
- volcano-sh/volcano
- kubernetes-sigs/kueue
- ray-project/ray
- NVIDIA/DCGM
- ofiwg/libfabric
- kubeflow/training-operator

万卡集群的瓶颈通常在网络与存储

02 数据与训练

Agent 能力的真正来源：
数据 × 环境 × RL

- **预训练语料**
  网页 · 代码 · 数学 · 多语
  去重 · 质量分类 · benchmark 去污染
- **中期训练**
  32k → 256k / 1M 长上下文
  工具调用格式 · 合成 Agent 轨迹
- **SFT**
  专家 plan-act-observe 轨迹
  失败恢复 · 拒绝采样
- **偏好优化**
  DPO / RLHF
  有用性 · 安全性 · 风格
- **RLVR**
  单测通过 · 数学对答案
  规则奖励 · 防 reward hacking
- **Agentic RL**
  环境 rollout · 异步采样
  off-policy 修正 · 成功率 + 效率
- **环境工场**
  浏览器 / 终端 / IDE / API mock
  可重置 · 可判定 · 万级并发
- **训练栈**
  Megatron / FSDP2 / DeepSpeed
  FP8 · 序列并行 · 激活重算
- **训练中评测**
  held-out 私有集 · 失败分类
  checkpoint 选择 · 回归门禁

GITHUB REPOS · 16 REPOS

- huggingface/datatrove
- allenai/dolma
- NVIDIA/NeMo-Curator
- modelscope/data-juicer
- NVIDIA/Megatron-LM
- pytorch/torchtitan
- deepspeedai/DeepSpeed
- volcengine/verl
- OpenRLHF/OpenRLHF
- huggingface/trl
- NovaSky-AI/SkyRL
- inclusionAI/AReaL
- unslothai/unsloth
- axolotl-ai-cloud/axolotl
- firecracker-microvm/firecracker
- e2b-dev/E2B

环境与轨迹数据 = Agent 时代的护城河

03 模型设计

围绕 Agent 场景做架构与能力取舍

- **架构选择**
  Dense vs MoE · GQA / MLA
  128k–1M context · KV 成本
- **推理能力**
  长 CoT · 可控 thinking budget
  自检 · 回溯 · 计划分解
- **工具原生**
  tool-call 格式进训练分布
  JSON Schema · 并行调用
- **多模态**
  截图理解 · GUI grounding
  坐标输出 · OCR
- **模型梯队**
  旗舰 + 蒸馏小模型
  路由器按任务难度分流
- **开源基座**
  Qwen3 · Llama · DeepSeek-V3
  Gemma · Mistral · MiniCPM

GITHUB REPOS · 8 REPOS

- QwenLM/Qwen3
- meta-llama/llama-models
- deepseek-ai/DeepSeek-V3
- huggingface/transformers
- Dao-AILab/flash-attention
- deepseek-ai/DeepEP
- NVIDIA/TransformerEngine
- huggingface/tokenizers

架构选择决定推理成本的上限

04 推理服务

把权重变成低延迟、低成本的 Token

- **推理引擎**
  vLLM · SGLang
  TensorRT-LLM · LMDeploy
- **批处理**
  continuous batching
  chunked prefill · prefix caching
- **KV Cache**
  PagedAttention · PD 分离
  HBM → DRAM → SSD 分层
- **解码加速**
  投机解码 · FP8 / INT4
  MoE EP · CUDA Graph
- **网关路由**
  模型级联 · 会话粘性
  限流 · 多副本 · 多区域
- **SLO**
  TTFT · TPOT · cache hit
  $/task · P99 · 可用性

成本关键： Agent 负载 = 长 system prompt + 多轮工具调用 → prefix cache 命中率直接决定单位成本

GITHUB REPOS · 12 REPOS

- vllm-project/vllm
- sgl-project/sglang
- NVIDIA/TensorRT-LLM
- InternLM/lmdeploy
- LMCache/LMCache
- ai-dynamo/dynamo
- NVIDIA/Triton-Inference-Server
- BerriAI/litellm
- ggml-org/llama.cpp
- ollama/ollama
- huggingface/text-generation-inference
- llm-d/llm-d

prefix cache 是 Agent 推理的命门

PRODUCTION TRACE → RETRAIN

05 Agent Harness

模型之外的体验分水岭

- **Agent Loop**
  observe → plan → act → observe
  步数 / token / 时间预算 · 取消与恢复
- **上下文工程**
  系统提示 · 工具描述 · 压缩/摘要
  长期记忆 · 子 Agent · RAG
- **工具层**
  JSON Schema · 幂等 · 超时/重试
  给模型看的可读错误 · 权限
- **沙箱**
  Firecracker / gVisor · 快照恢复
  网络出口白名单 · 资源限额
- **编排**
  durable execution · 队列
  human-in-the-loop · 审批
- **输出控制**
  grammar / JSON 约束解码
  流式 SSE · 引用 · 审计

体验分水岭： 同一模型，工具设计与上下文管理可带来数倍任务成功率差异
评估重点：长任务完成率 · 工具调用正确率 · 失败恢复 · $/task

GITHUB REPOS · 16 REPOS

- langchain-ai/langgraph
- openai/openai-agents-python
- huggingface/smolagents
- microsoft/autogen
- SWE-agent/SWE-agent
- All-Hands-AI/OpenHands
- browser-use/browser-use
- modelcontextprotocol/servers
- microsoft/playwright-mcp
- mem0ai/mem0
- letta-ai/letta
- temporalio/temporal
- google/gvisor
- sst/opencode
- cline/cline
- block/goose

Harness 迭代速度 > 模型换代速度

06 Agent 平台与框架

把原型变成可运营的产品
低代码与代码优先两条路线

- **代码优先框架**
  LangGraph · OpenAI Agents SDK
  Pydantic AI · Mastra · Agno
- **多 Agent 协作**
  AutoGen · CrewAI
  Semantic Kernel · Google ADK
- **低代码编排**
  Dify · Coze · n8n
  Langflow · Flowise
- **RAG 与知识库**
  RAGFlow · FastGPT · LlamaIndex
  Haystack · 文档解析 · 混合检索
- **发布与运营**
  Web / API / SDK · 多渠道
  多租户 · RBAC · SSO
- **协议与集成**
  MCP · A2A · OpenAPI · OAuth
  插件市场 · 模板 DSL

选型分叉： Dify / Coze / n8n 适合快速交付；LangGraph / OpenAI Agents SDK 适合深水区
企业常见组合 = 低代码编排 + 自建 Harness + 私有模型

GITHUB REPOS · 16 REPOS

- langgenius/dify
- coze-dev/coze-studio
- n8n-io/n8n
- langflow-ai/langflow
- FlowiseAI/Flowise
- infiniflow/ragflow
- labring/FastGPT
- open-webui/open-webui
- danny-avila/LibreChat
- run-llama/llama_index
- deepset-ai/haystack
- microsoft/semantic-kernel
- pydantic/pydantic-ai
- google/adk-python
- mastra-ai/mastra
- agno-agi/agno

07 安全与权限

能执行代码和转账时，
安全不是附加项

CROSS-CUTTING

- **最小权限**
  工具白名单 · 短期凭证
  按租户 / 会话隔离
- **策略引擎**
  OPA 策略 · 出站管控
  配额 · 审批流
- **注入防御**
  不可信内容隔离 · 双 LLM
  URL / 文件 / 网页沙箱
- **动作确认**
  高风险操作二次确认
  转账 / 删除 / 发布
- **数据安全**
  PII 脱敏 · 留存策略
  数据出境 · 密钥托管
- **审计**
  全量动作日志
  可追溯 · 可回放
- **持续红队**
  garak · promptfoo
  注入 / 越狱 / 工具滥用
- **供应链**
  依赖扫描 · 模型来源
  镜像签名 · SBOM

GITHUB REPOS · 8 REPOS

- guardrails-ai/guardrails
- NVIDIA/NeMo-Guardrails
- meta-llama/PurpleLlama
- microsoft/presidio
- promptfoo/promptfoo
- NVIDIA/garak
- protectai/rebuff
- open-policy-agent/opa

08 观测与数据飞轮

让每一次生产调用
变成下一轮训练数据

DATA FLYWHEEL

- **全链路 Trace**
  每步 token · 时延
  工具结果 · 成本
- **任务指标**
  成功率 · 步数 · 接管率
  $/task · 失败分类
- **成本核算**
  token 明细 · 缓存命中
  GPU 利用率 · 分摊
- **失败归因**
  工具 / 推理 / 环境
  自动聚类 · 回归用例
- **离线回归**
  黄金集 · 私有 eval
  版本对比 · 门禁
- **在线实验**
  A/B · canary
  自动回滚 · 灰度
- **数据精选**
  脱敏 · 去重 · 难度分层
  轨迹质量打分
- **再训练**
  SFT / RL → 新 ckpt
  再评测 → 再发布

GITHUB REPOS · 8 REPOS

- langfuse/langfuse
- Arize-ai/phoenix
- comet-ml/opik
- Helicone/helicone
- open-telemetry/opentelemetry-python
- HumanSignal/label-studio
- wandb/wandb
- confident-ai/deepeval

没有 trace 就没有飞轮

一次请求的 Token 旅程

从 HTTP 请求到流式 Token —— 中间是多次模型调用与工具执行的循环

01 **接入**
Auth · rate limit
session / tenant
EDGE / API

02 **上下文组装**
system + tools + memory
history 按 token 预算裁剪
HARNESS

03 **路由**
模型级联
区域 / 配额 / 成本
ROUTER

04 **Prefill**
tokenize · KV cache
prefix cache 命中
GPU · PREFILL

05 **Decode**
流式 token
grammar / tool-call
GPU · DECODE

06 **工具执行**
schema 校验
沙箱 → observation
SANDBOX

07 **循环**
回到 02，直到
预算 / 完成 / 失败
HARNESS

08 **输出**
SSE 流式返回
trace · cost · feedback
STREAM + TRACE

observation 回填 → 在 step / token / 时间预算内循环

落地路线 · 务实版

先验证闭环，再降本，最后才规模化

注意 · 预训练是最后才考虑的事

PHASE 1 · 0–3 月
验证闭环
前沿 API + Harness + 沙箱 + Eval
先证明任务成功率与单位经济模型
EXIT · 端到端成功率 > 基线，$/task 可算

PHASE 2 · 3–9 月
降本可控
开源基座 + 领域 SFT + vLLM 自托管
私有 eval 回归 + prefix cache 优化
EXIT · 成本降 5–10×，质量不降

PHASE 3 · 9 月+
规模化
Agentic RL + 自建训练集群 + 数据飞轮
只有规模证明后才考虑预训练
EXIT · 成功率持续爬坡，边际成本下降