大模型推理基础设施

第 2 讲工作负载与评价指标

张书豪

华中科技大学计算机学院

研究生课程课件

教学目标

核心结论

本讲围绕工作负载与评价指标展开，建立刻画推理系统问题的基本分析坐标。

- 区分离线、在线、RAG 与智能体四类典型服务场景。
- 理解 TTFT、TPOT、P99 与 goodput 等核心指标的系统含义。
- 说明单点指标为何不足以代表端到端系统收益。

本讲的基本关系

核心结论

第二讲的核心关系是：工作负载先定义问题边界，评价指标再定义优化目标与解释方式。

- 相同指标在不同工作负载下可能对应不同的系统含义。
- 相同 workload 在不同优化目标下也可能导向不同的系统设计。
- 因而所有实验和比较都需要先交代 workload，再报告指标。

典型工作负载类型

核心结论

不同工作负载会重新定义系统瓶颈，因此系统设计不能脱离场景。

- 离线批推理关注总吞吐与资源利用率。
- 在线对话关注 TTFT、TPOT 与尾时延。
- RAG 服务关注检索到生成的全链路时延。
- 智能体工作流关注跨轮状态、工具调用与任务级完成时间。

四类工作负载的系统差异

核心结论

同样被称为“推理”的服务，在不同 workload 下会呈现显著不同的主导矛盾。

- 离线推理更偏向吞吐与成本优化。
- 在线对话更偏向 TTFT 与尾时延控制。
- RAG 更偏向检索与生成链路协同。
- 智能体更偏向多阶段流水与跨轮状态维护。

为什么工作负载变了，系统问题也会变

核心结论

长上下文、MoE、多模态与智能体工作流会从容量、调度、通信与状态维护多个层面放大系统复杂度。

- 长上下文放大 prefill 代价与 KV 压力。
- MoE 放大负载倾斜与通信波动。
- 智能体将单轮生成扩展为多阶段流水与持续写入问题。

长上下文如何改变瓶颈

核心结论

长上下文会同时推高 prefill 计算代价、KV 驻留压力与排队放大效应。

- prompt 变长会直接增加 prefill 开销。
- KV 驻留时间更长，状态占用更大。
- 长短请求混部时更容易出现尾时延恶化。

核心评价指标

核心结论

评价指标的选择实质上对应系统优化目标的定义。

- TTFT：首 token 等待时间。
- TPOT：相邻输出 token 的平均间隔。
- Throughput：单位时间完成的请求数或 token 数。
- P95/P99：尾时延表现。
- Goodput：满足服务目标的有效吞吐。

goodput 的形式化表达

核心结论

goodput 的关键不在于重新命名 throughput，而在于把服务目标是否满足纳入统计口径。

$$\text{Goodput} = \frac{\#\{\text{requests meeting SLO}\}}{\Delta t}$$

- 当大量请求违反 SLO 时，原始 throughput 的解释价值会迅速下降。
- goodput 更适合比较调度、缓存与资源治理策略的真实服务效果。

指标之间为什么会冲突

核心结论

指标冲突是系统优化中的常态，因此必须首先明确主目标与可接受代价。

- 更大的 batch 通常提高吞吐，但可能恶化 TTFT。
- 更激进的缓存策略可能提高命中率，但也提高管理代价。
- 更高设备利用率并不等于更低 P99。

为什么平均值常常不够

核心结论

用户体验和服务质量常由尾部与波动决定，而不是平均值。

- 更大的 batch 可能提高平均吞吐，但也可能恶化 TTFT。
- 更高命中率未必自动带来端到端时延下降。
- 单次最好结果并不等于稳定收益。

为什么要引入 goodput

核心结论

goodput 的意义在于把“服务目标是否满足”纳入吞吐定义，而不是只统计原始完成量。

- 若大量请求违反 SLO，原始 throughput 的意义会迅速下降。
- goodput 更适合比较不同调度或资源治理策略。
- 它将系统目标从“完成更多”转化为“有效完成更多”。

工作负载与指标的先后关系

核心结论

指标不能脱离场景单独解释；只有先交代 workload，数字才具有明确含义。

- 离线批处理更接近资源利用率问题，在线对话更接近响应性问题。
- RAG 与 Agent 会把“生成”嵌入更长链路，因此单点吞吐不再足以代表体验。
- 同一套指标在不同 workload 下会具有不同含义。

例子：不同在线 workload 的差异

核心结论

即便都属于在线请求，不同 workload 的长度分布、链路结构与服务目标也可能完全不同。

场景	分布特征	更敏感的系统问题
对话服务	到达连续，prompt 长度波动大	TTFT、尾时延、短请求体验
RAG 服务	检索与生成交织，链路更长	全链路时延、阶段协同
Agent 服务	多轮工具调用与多次生成	任务级完成时间、跨轮状态维护与一致性

例子：均值为何经常产生误导

核心结论

系统实验中最常见的误区，是只看均值而忽视分布、尾部与服务目标。

表面现象	可能遗漏的信息	更合适的补充指标
平均 TTFT 正常	长请求与突发到达已推高 P99	P95/P99、队列等待
平均吞吐更高	短请求体验可能显著恶化	TTFT、短请求完成时间
完成量增加	满足 SLO 的请求比例可能下降	Goodput、SLO 达成率

代码例子：benchmark 如何固定分布并生成指标

核心结论

工作负载与指标分析需要学生看到：实验结果并非“随便跑一遍”即可解释。

输入分布控制

```
seed(0)
prompt_token_ids = [[randint(...)
...]]
sampling_params =
[SamplingParams(...)]
```

结果计算逻辑

```
t = time.time()
llm.generate(..., use_tqdm=False)
throughput = total_tokens /
(time.time() - t)
```

- 输入长度分布与输出长度分布需要分别说明。
- 仅有总体 throughput 仍不足以支撑 TTFT、P99 与队列等待层面的完整结论。

例子：throughput、TTFT 与 P99 的张力

核心结论

工作负载与评价指标这一讲的重点，是接受真实系统中的指标张力并据此陈述取舍。

- 更大的 batch 往往提高总体吞吐，但也可能抬高等待时间与 TTFT。
- 更保守的调度可能稳定尾时延，却压低设备利用率与平均吞吐。
- 因而任何优化都应首先说明优先保留的指标与可接受的代价。

课堂练习：为同一系统写出不同的指标解释

核心结论

同一组数字可以对应不同解释，关键在于问题定义、工作负载设定与目标指标选择。

1. 若一个系统吞吐提高 15%，但 TTFT 恶化 30%，如何判断它是否更适合在线服务。
2. 给出一个适用于 Agent 工作流的 SLO 组合，而不只写单个 TTFT 阈值。
3. 说明为什么指标解释必须回到 workload 与场景。

课堂讨论

核心结论

指标选择本身构成系统问题定义的重要组成部分。

- 为什么以 TTFT 为核心的服务不会只优化峰值吞吐。
- RAG 服务与对话服务的主指标为何不同。
- 什么情况下应报告 goodput 而非 raw throughput。

课堂练习

核心结论

本讲练习采用“场景 - 指标 - 瓶颈”三列表格式，以训练问题刻画能力。

- 每个场景至少写出两个主指标。
- 每个指标至少写出一个可能冲突指标。
- 每个冲突都要对应一个系统机制取舍。

本讲总结

核心结论

明确工作负载与评价指标，是后续讨论调度、KV 与执行优化问题的共同前提。

- 不同场景对应不同瓶颈。
- 指标定义决定优化方向与论文叙事边界。

对应 Tutorial

核心结论

本讲对应 Tutorial 1。建议先独立作答，再翻到下一页查看参考答案。

- 文件：build/tutorials/Tutorial_01_ 工作负载与评价指标.pdf
- 动手运行、日志记录与提交要求统一查看 build/experiments/ 和对应 experiment 源目录下的 assignment_spring-2026/。