Tutorial 7：执行优化与异构路径

题目

1. [单选题] 为什么 kernel 更快后，端到端指标仍可能没有明显改善？A. 因为 kernel 从不影响系统性能 B. 因为排队、通信和状态成本也可能主导整体表现 C. 因为所有 benchmark 都是错误的 D. 因为 GPU 不再执行模型

---

2. [判断题] 只要减少计算量，端到端性能就一定会提升。

---

3. [多选题] 哪类 workload 更适合图执行或编译类优化？A. 形状稳定 B. 路径重复度高 C. 请求长度极度波动 D. 运行路径较固定

---

4. [简答题] 为什么 launch 开销不能被忽略？

5. [计算题] 某系统原始总时延由三部分构成：排队 40ms、计算 40ms、数据搬运 20ms，总计 100ms。若仅将计算部分优化 25%，新的总时延是多少？端到端加速比约为多少？

---

6. [判断题] 如果 benchmark 中 kernel 时间下降，就足以证明系统整体收益成立。

---

7. [简答题] 为什么带宽与放置方式会直接左右真实收益？

8. [伪代码题] 请写出一个简短伪代码：若 `compute_gain > transfer_cost + maintenance_cost`，则接受该优化；否则拒绝。

9. [多选题] 下列哪些现象说明优化可能只快在局部？A. kernel 时间下降 B. TTFT 基本不变 C. goodput 基本不变 D. 端到端延迟大幅下降

---

10. [简答题] 在什么情况下，应拒绝把某项优化接入真实系统？

参考答案

1. B。排队、通信和状态管理成本也可能主导整体表现。
2. 错。减少计算量并不自动等于端到端更快。
3. A、B、D。这类 workload 更适合此类优化。
4. 因为频繁启动本身就会消耗显著时间。
5. 新计算时延 = 40 x 0.75 = 30ms；新总时延 = 40 + 30 + 20 = 90ms；端到端加速比约为 100 / 90 = 1.11。
6. 错。局部 kernel 改善并不足以自动证明整体收益。
7. 因为数据放置不当会将理论收益转化为额外延迟和搬运成本。
8. 示例伪代码：

```
function decide(compute_gain, transfer_cost, maintenance_cost):
     if compute_gain > transfer_cost + maintenance_cost:
          return "accept"
     return "reject"
```

9. A、B、C。这类现象说明收益可能停留在局部层面。
10. 当它只改善局部而维护成本明显过高时，应谨慎拒绝接入。