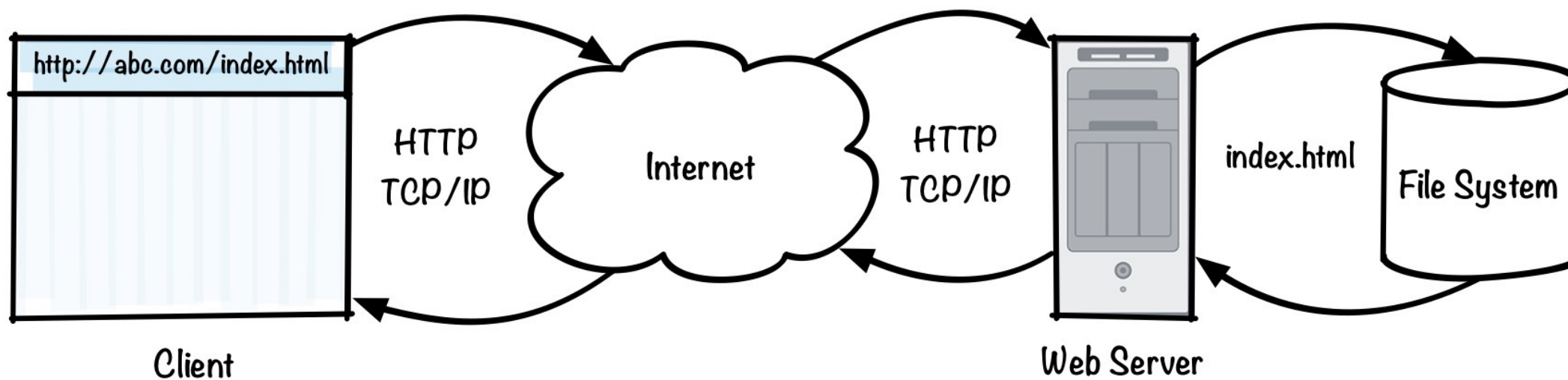


# Web服务器多线程模型

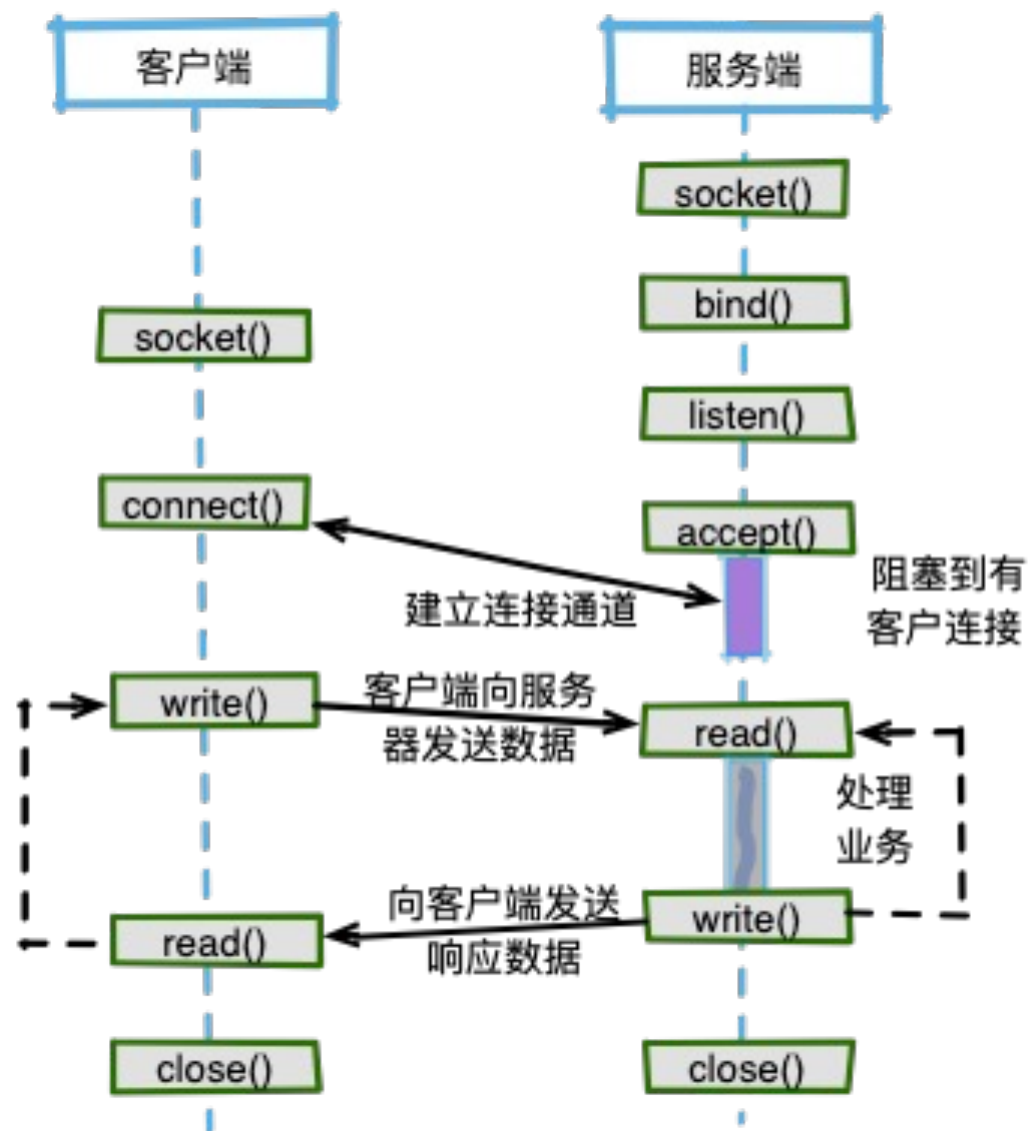
操作系统课程设计2023

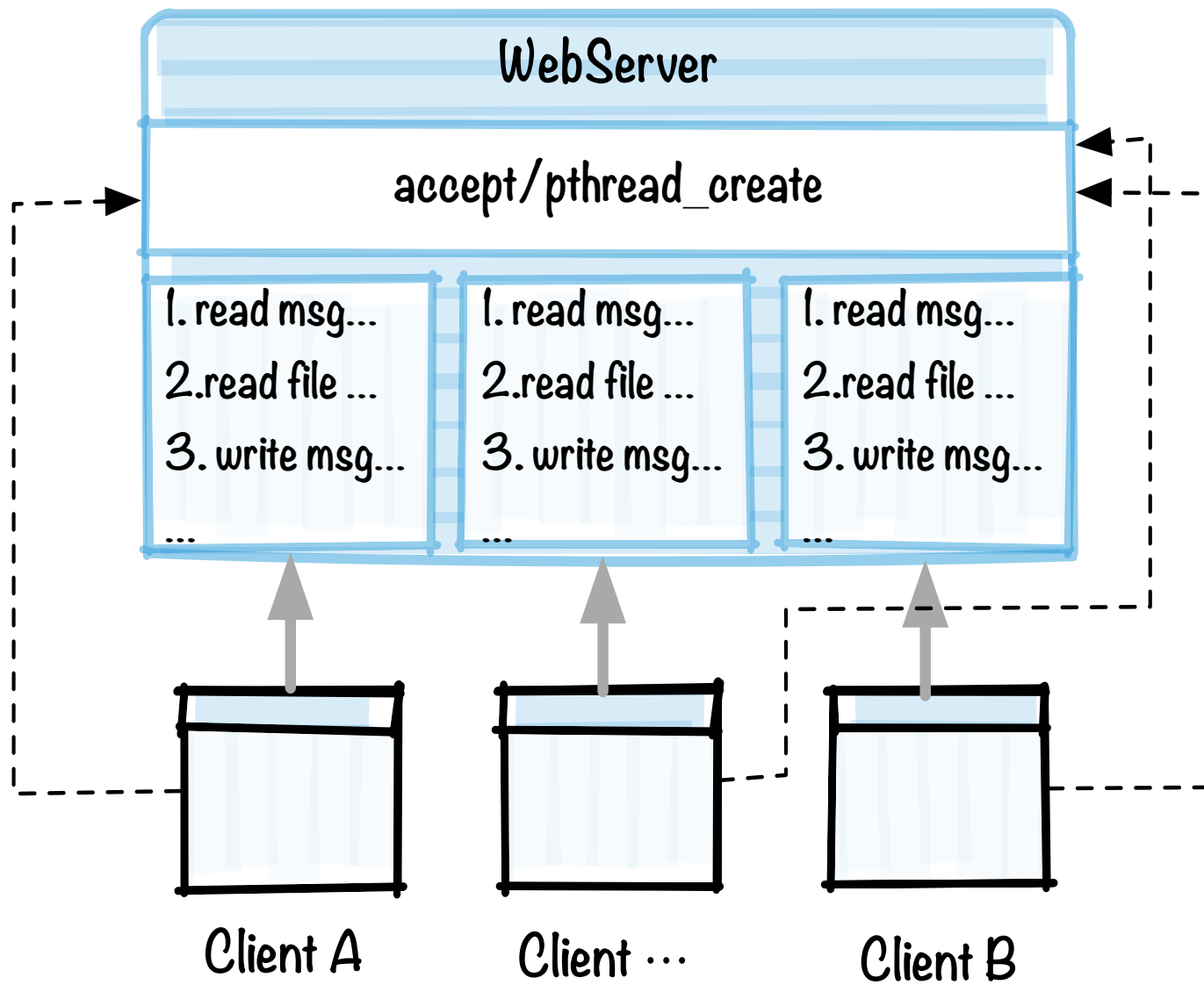
鲁强

# 一、Web服务器架构



## 二、Socket编程和多线程





### 实验3. Web服务器的多线程模型

**题目1.** 将上述多线程Web服务器与实验2中多进程模型的Web服务器性能进行对比，说明它俩各具有什么优缺点。具体对比的指标包括如下：

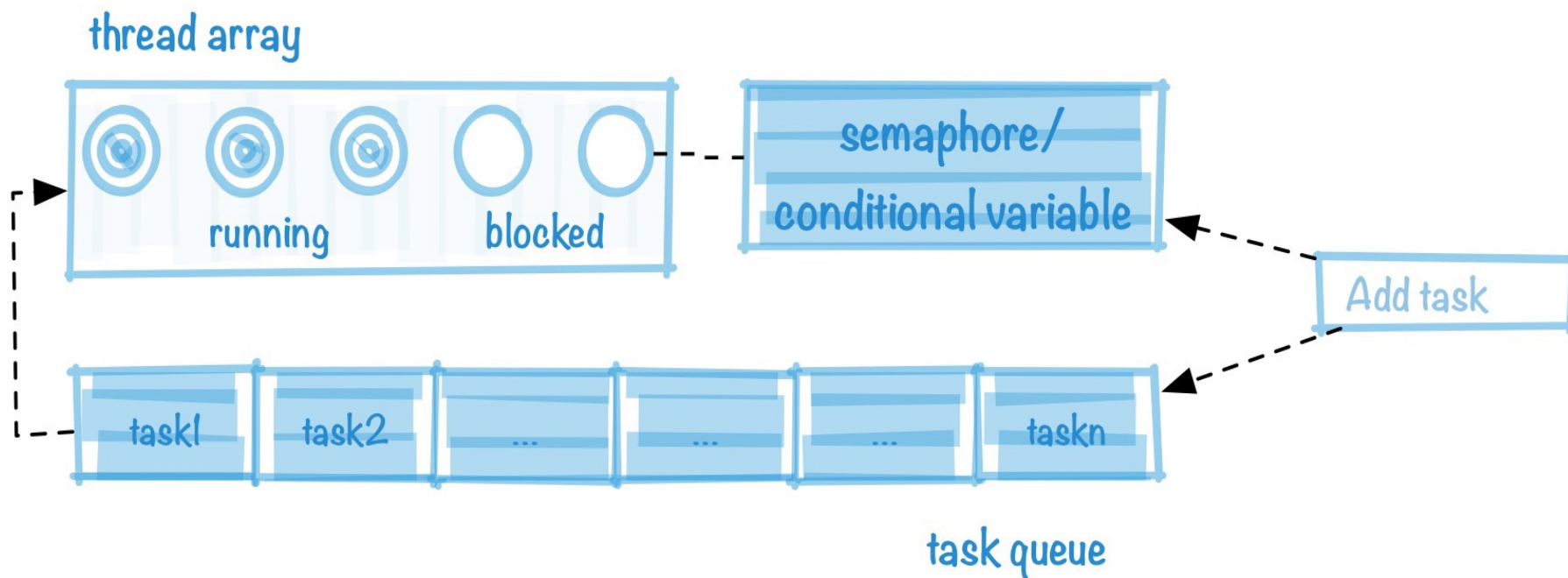
- 使用http\_load命令来测试这两个模型下Web服务器的性能指标。并根据这些测试指标对比，分析为什么这两种模型会产生不同的性能结果？
- 对这两个模型中的socket数据读取、发送、网页文件读取和日志文件写入四个I/O操作分别**计时**，并打印出每个进程或线程处理各项I/O计时的平均时间。例如，编写的程序应该打印出如下结果。

```
共用 10000ms 成功处理 100 个客户端请求，其中
平均每个客户端完成请求处理时间为 5100ms。
平均每个客户端完成读 socket 时间为 500ms。
平均每个客户端完成写 socket 时间为 1000ms。
平均每个客户端完成读网页数据时间为 110ms。
平均每个客户端完成写日志数据时间为 50ms。
```

- 根据上面的计时数据结果，分析并说明哪些多进程模型和多线程模型中哪些I/O操作是最消耗时间的？。
- 思考一下，怎么修改线程模型，才能提高线程的并发性能？

**题目2.** 调整http\_load命令参数，增加其并发访问线程数量，会发现随着并发访问数量在达到一定数量后，再增多会导致多线程Web服务进程的性能出现下降的现象。试分析产生上述现象的原因是什么？

### 三、线程池模型



## 实验4 Web服务器的线程池模型

题目1. 添补相应的程序代码到上面函数中“.....”位置处。

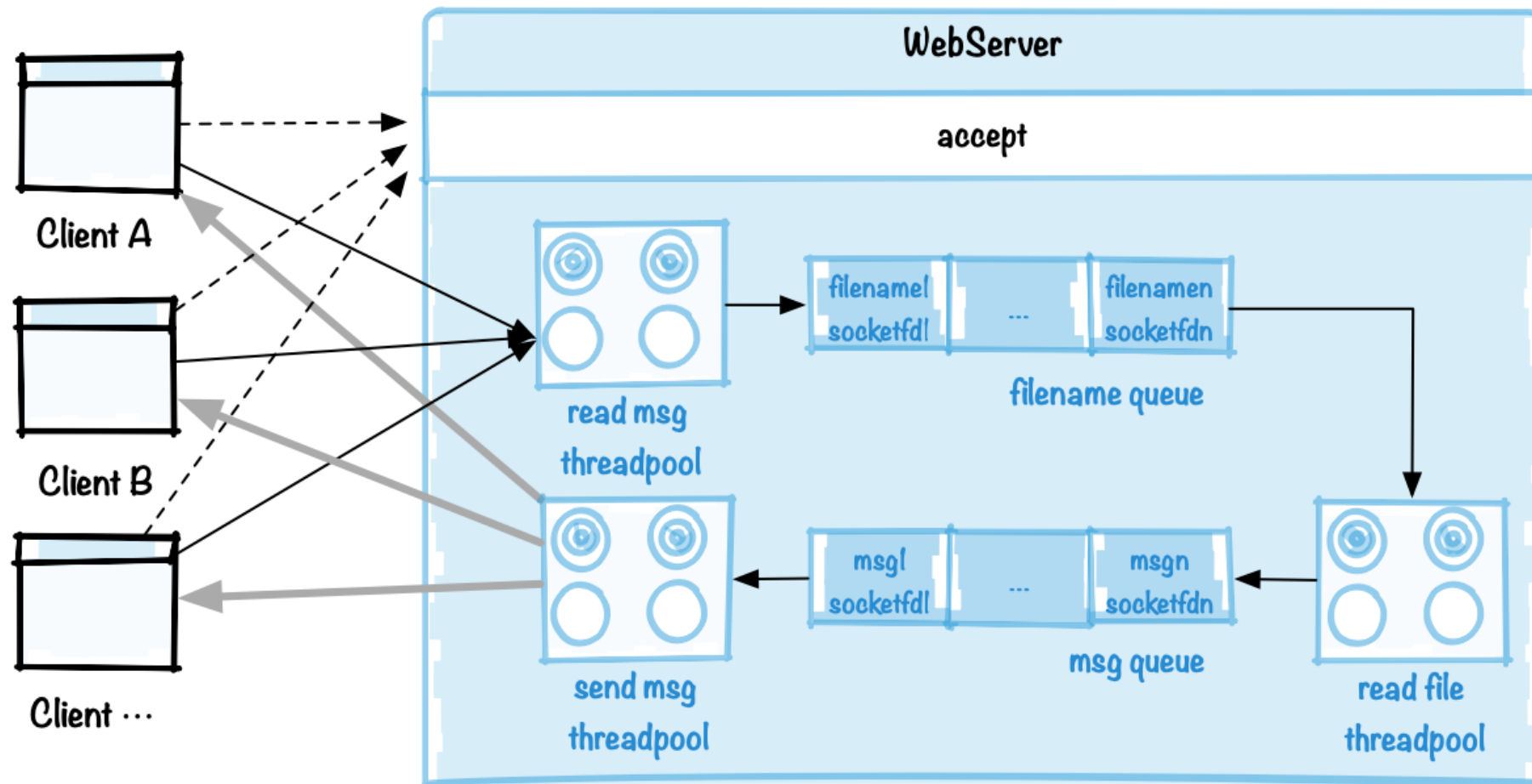
题目2. 完成函数push\_taskqueue, take\_taskqueue, init\_taskqueue和destory\_taskqueue。

题目3. 添加必要的程序代码，以最终完成线程池。

题目4. 利用实现的线程池，替换实验3中Web服务的多线程模型。

题目5. 调整线程池中线程个数参数，以达到Web服务并发性能最优。利用http\_load及其它性能参数，分析和对比多线程模型与线程池模型在Web服务进程中的优点和缺点。

## 四、业务分割模型





## 实验5 Web服务器的业务分割模型

**题目1.** 实现上述的业务分割模型的Web服务程序。

**题目2.** 在程序里面设置性能监测代码，通过定时打印这些性能参数，能够分析此Web服务程序的运行状态。例如，线程池中线程平均活跃时间及其阻塞时间，线程最高活跃数量、最低活跃数量、平均活跃数量；消息队列中消息的长度等。除此以外还可以利用相关系统命令来监测系统的I/O、内存、CPU等设备性能。

**题目3.** 通过上述的性能参数和系统命令，对Web服务程序进行逻辑分析，发现当前程序存在性能瓶颈的原因。进而通过控制各个线程池中的线程数量和消息队列长度，来改善此程序的性能。

## 五、考核及实验报告撰写要求

- 1. 每人独立完成每个题目
- 2. 内容包含完成每个题目的思考、设计方案、源代码（带注释）、实验实现过程（附相关抓图）说明、实验运行结果展示及相应的分析
  - 只提交代码，没有分数！！！！
  - 提交实验报告雷同、成绩为零
- 4. 最终考核包括答辩成绩和实验报告成绩
  - 答辩 40% + 报告60%