# C++后端/HPC/算子开发实习求职全指南（详细版）

## 一、岗位全景分析（需求、难度、适配性）

### （一）核心维度对比表

| 维度 | C++后端开发工程师 | 高性能计算工程师（HPC） | 算子开发（AI/HPC细分） |
| --- | --- | --- | --- |
| **社会需求** | ★★★★★（海量） | ★★★☆☆（中等） | ★★★☆☆（中等） |
| **需求分布领域** | 互联网（电商/社交/游戏）、云计算、金融科技、工业软件、嵌入式系统 | 超算中心、科研院所（气象/生物/石油）、AI训练平台、自动驾驶、芯片厂商 | 深度学习框架团队（TensorFlow/PyTorch）、AI芯片厂商（NVIDIA/寒武纪）、大模型企业、云计算厂商AI部门 |
| **学习难度** | 中等偏上 | 高 | 高 |
| **难度核心点** | 1. C++语法复杂度高，需掌握内存管理/模板编程；<br>2. 技术栈广（网络/数据库/分布式）；<br>3. 高并发场景需实战积累 | 1. 跨学科知识（编程+硬件+数值算法）；<br>2. 并行/异构编程门槛高；<br>3. 集群调度与性能调优需深度实践 | 1. 复合知识（C+++深度学习+硬件）；<br>2. 算子优化无通用方法论，需针对性调优；<br>3. 不同硬件架构适配难度大 |
| **学历要求** | 本科+（大厂核心岗/金融科技偏好硕士） | 硕士优先（科研机构/超算中心基本要求硕士） | 硕士优先（头部企业/芯片厂商偏好硕士） |
| **校招薪资（实习）** | 200-400元/天（大厂）、150-250元/天（中小企业） | 250-500元/天（头部企业/科研机构）、200-300元/天（普通企业） | 300-600元/天（头部AI/芯片企业）、250-400元/天（中厂） |
| **适配人群** | 1. 追求岗位数量、稳定性；<br>2. 擅长工程落地、业务逻辑拆解；<br>3. 能接受多技术栈并行学习 | 1. 有数学/物理基础，偏科研/算力方向；<br>2. 对硬件架构/并行计算感兴趣；<br>3. 能接受长周期项目打磨 | 1. 懂深度学习+异构计算；<br>2. 对性能优化有执念；<br>3. 关注AI技术前沿（大模型/推理加速） |
| **需求趋势** | 长期稳定，云原生/微服务方向需求增长 | 增速快，大模型训练/国产超算驱动需求 | 高增速，大模型推理优化/国产芯片适配缺口扩大 |

## （二）岗位分层需求说明

### 1. C++后端开发

- **初级岗（实习/校招）**：侧重C++基础、Linux网络编程（Socket/epoll）、MySQL基础操作，能完成简单业务模块开发（如接口编写、数据读写）；
- **中高级岗**：要求分布式系统设计（一致性哈希/RPC）、高并发优化（线程池/无锁编程）、数据库调优（索引/事务），能解决生产环境性能瓶颈。

### 2. HPC

- **初级岗（实习/校招）**：掌握OpenMP/MPI基础、CUDA入门，能完成简单并行程序开发（如矩阵乘法并行）；
- **中高级岗**：要求异构计算优化（GPU/ROCm）、集群调度（Slurm）、数值算法设计，能适配超算集群完成大规模计算任务。

### 3. 算子开发

- **初级岗（实习/校招）**：掌握CUDA基础、深度学习框架算子注册，能完成简单算子（如池化/激活）开发；
- **中高级岗**：要求算子融合/向量化优化、多硬件适配（GPU/NPU）、TensorRT推理加速，能针对大模型核心算子（如注意力算子）做性能调优。

# 二、全维度知识体系（通用+专属，附学习深度）

## （一）通用核心知识（必学，附掌握程度）

| 知识模块 | 核心内容 | 掌握程度要求 | 学习资源推荐 |
| --- | --- | --- | --- |
| **C++核心语法** | 1. 基础：C++11/14/17（智能指针、Lambda、STL容器/算法、nullptr、范围for）；<br>2. 进阶：右值引用/移动语义、模板元编程、RAII、函数对象/绑定器；<br>3. 底层：内存管理（堆/栈/内存泄漏）、编译器优化（内联/ constexpr） | 1. 基础：能手写无BUG代码；<br>2. 进阶：理解原理并能落地项目；<br>3. 底层：能排查内存问题、分析代码性能 | 1. 《C++ Primer》（第5版）；<br>2. 《Effective Modern C++》；<br>3. 侯捷C++视频教程 |
| **Linux系统编程** | 1. 进程/线程：fork/exec、pthread库（互斥锁/条件变量/原子操作）；<br>2. IPC：管道、消息队列、共享内存、信号量；<br>3. IO：open/read/write/lseek、mmap内存映射；<br>4. 网络：Socket（TCP/UDP）、epoll（LT/ET模式）、HTTP/TCP协议 | 1. 能独立编写进程/线程通信程序；<br>2. 能基于epoll实现高并发网络程序；<br>3. 理解协议底层原理（如TCP三次握手） | 1. 《UNIX环境高级编程》；<br>2. 《Linux高性能服务器编程》；<br>3. 牛客网Linux系统编程实战课 |
| **工具与工程能力** | 1. 调试：gdb（断点/堆栈/变量监控）、valgrind（内存泄漏检测）、perf（性能分析）；<br>2. 版本控制：Git（分支/PR/冲突解决）；<br>3. 编译：CMake（项目配置、静态/动态库）、Makefile；<br>4. 编码规范：Google C++ Style、代码评审流程 | 1. 能独立调试复杂程序（内存泄漏/段错误）；<br>2. 能搭建规范的C++项目工程；<br>3. 能使用perf分析性能瓶颈 | 1. Git官方文档；<br>2. CMake实战教程（B站）；<br>3. 《Linux性能优化大师课》 |

| 知识模块 | 核心内容 | 掌握程度要求 | 学习资源推荐 |
| --- | --- | --- | --- |
| **计算机基础** | 1. 体系结构：CPU缓存（缓存行/预取）、NUMA架构、SIMD指令集；<br>2. 操作系统：进程调度、内存分页、文件系统；<br>3. 数据结构与算法：链表/树/哈希表、排序/查找/动态规划、时间/空间复杂度分析 | 1. 能手撕高频算法题（LeetCode中等难度）；<br>2. 理解硬件对代码性能的影响；<br>3. 能设计高效数据结构 | 1.《算法导论》；<br>2.《计算机组成原理》（唐朔飞）；<br>3. LeetCode Hot 100 |
| **数学基础** | 1. 通用：线性代数（矩阵运算、向量）、概率论；<br>2. 进阶（HPC/算子）：数值分析（数值逼近/线性方程组求解）、微积分 | 1. 通用部分：能理解并实现矩阵运算；<br>2. 进阶部分：能看懂数值算法伪代码并落地 | 1.《线性代数及其应用》；<br>2.《数值分析》（李庆扬） |

## （二）方向专属知识（优先级从高到低，附学习重点）

| 岗位方向 | 专属核心知识 | 学习重点 |
| --- | --- | --- |
| **C++后端开发** | 1. 网络编程进阶：<br>- epoll反应堆模型、Reactor/Proactor模式；<br>- TCP粘包/拆包、心跳机制、超时重连；<br>2. 数据库：<br>- MySQL索引（B+树）、事务（ACID）、连接池设计；<br>- SQL优化、分库分表、Redis缓存（过期策略/持久化）；<br>3. 分布式基础：<br>- 一致性哈希、RPC框架（brpc/muduo/gRPC）；<br>- 负载均衡、分布式锁、CAP理论；<br>4. 高并发：<br>- 线程池设计、无锁编程（CAS）、协程；<br>- 限流/熔断/降级（Sentinel原理） | 1. 能独立实现高并发TCP服务器；<br>2. 能设计并优化数据库访问层；<br>3. 理解RPC核心原理并能实现简易版本；<br>4. 能解决高并发场景下的竞态/性能问题 |
| **HPC** | 1. 并行计算：<br>- OpenMP（多线程并行、数据/任务并行）；<br>- MPI（点对点通信、集体通信、分布式矩阵运算）；<br>2. 异构计算：<br>- CUDA编程（线程层次、内存模型、核函数优化）；<br>- ROCm/HIP（可选，适配AMD GPU）；<br>3. 硬件架构：<br>- CPU多核调度、GPU流多处理器（SM）、共享内存优化；<br>4. 数值计算：<br>- Eigen/BLAS/LAPACK矩阵运算；<br>- 有限元/有限差分法（可选，按领域）；<br>5. 集群管理：<br>- Slurm/PBS集群调度、Lustre高性能存储 | 1. 能实现OpenMP/MPI并行程序并调优；<br>2. 能基于CUDA实现GPU加速程序；<br>3. 能分析并行程序性能瓶颈（负载不均衡/通信开销） |

| 岗位方向 | 专属核心知识 | 学习重点 |
| --- | --- | --- |
| **算子开发** | 1. 深度学习框架：<br>- 计算图原理、算子注册机制、自动微分；<br>- PyTorch/TensorFlow C++扩展开发；<br>2. 异构编程：<br>- CUDA/HIP高级优化（共享内存/常量内存）；<br>- 不同硬件（GPU/NPU/TPU）架构适配；<br>3. 算子优化：<br>- 算子融合、向量化计算、内存访问优化；<br>- 精度校准（FP16/INT8）；<br>4. 推理加速：<br>- TensorRT/ONNX Runtime使用；<br>- 大模型算子（注意力/卷积）优化 | 1. 能开发自定义深度学习算子并支持自动微分；<br>2. 能优化算子性能（比原生提升20%+）；<br>3. 能基于TensorRT做推理加速 |

# 三、实习项目全攻略（从设计到简历呈现）

## （一）项目推荐（分方向，附详细技术方案）

| 岗位方向 | 项目名称 | 技术栈明细 | 核心功能模块 | 落地步骤 | 简历量化亮点 |
| --- | --- | --- | --- | --- | --- |
| **C++后端开发** | 高性能TCP网络服务器 | C++11、Linux Socket、epoll（ET模式）、线程池、异步日志、连接池 | 1. 网络层：epoll反应堆模型处理连接；<br>2. 线程层：主线程接收连接，工作线程处理请求；<br>3. 功能层：Echo/HTTP服务、TCP粘包处理；<br>4. 辅助层：异步日志（落盘优化）、连接超时清理 | 1. 学习epoll原理，实现基础Reactor模型；<br>2. 编写线程池，实现任务异步调度；<br>3. 增加日志模块，优化IO效率；<br>4. 压测并优化（如关闭Nagle算法） | 1. 单机支持10k+并发连接，请求响应延迟<10ms；<br>2. 异步日志落盘性能提升50%，无内存泄漏；<br>3. 处理TCP粘包准确率100%，支持百万级请求无崩溃 |

| 岗位方向 | 项目名称 | 技术栈明细 | 核心功能模块 | 落地步骤 | 简历量化亮点 |
| --- | --- | --- | --- | --- | --- |
| | 轻量级RPC框架 | C++11、Protobuf、TCP、服务注册发现、负载均衡、超时重试 | 1. 协议层：Protobuf序列化/反序列化；<br>2. 网络层：封装TCP通信，实现请求/响应协议；<br>3. 服务层：服务注册/发现、同步/异步调用；<br>4. 优化层：轮询负载均衡、超时重试（3次） | 1. 定义Protobuf服务接口；<br>2. 实现客户端/服务端通信模块；<br>3. 开发服务注册中心（简易版）；<br>4. 增加负载均衡和容错机制 | 1. 支持跨语言调用（C++/Python），接口调用成功率99.9%；<br>2. 负载均衡策略降低单节点负载30%；<br>3. 超时重试机制减少请求失败率80% |
| **HPC** | OpenMP/MPI矩阵乘法并行优化 | C++、OpenMP、MPI、Eigen、perf性能分析 | 1. 串行版本：基于Eigen实现矩阵乘法；<br>2. OpenMP版本：多线程并行（数据分片）；<br>3. MPI版本：4节点分布式并行（数据通信优化）；<br>4. 对比：串行/OpenMP/MPI性能测试 | 1. 实现串行矩阵乘法；<br>2. 基于OpenMP做线程级并行；<br>3. 基于MPI做分布式并行；<br>4. 使用perf分析瓶颈并优化（如缓存优化） | 1. OpenMP版本（8核CPU）比串行提速8-10倍；<br>2. MPI分布式版本（4节点）提速3倍，通信开销降低20%；<br>3. 矩阵规模10000×10000计算耗时从120s降至12s |
| | CUDA图像滤波并行加速 | C++、CUDA C、OpenCV、NVIDIA Nsight、共享内存优化 | 1. CPU版本：Sobel边缘检测/高斯模糊；<br>2. CUDA版本：核函数实现、共享内存优化；<br>3. 对比：CPU/GPU性能、精度对比；<br>4. 优化：线程块划分、内存访问模式优化 | 1. 实现CPU版本图像滤波；<br>2. 编写CUDA核函数，实现GPU版本；<br>3. 优化共享内存访问，减少全局内存读写；<br>4. 使用Nsight分析并调优 | 1. GPU版本（NVIDIA RTX 3090）处理1920×1080图像提速50+倍；<br>2. 共享内存优化后，GPU利用率从60%提升至90%；<br>3. 边缘检测精度与CPU版本一致，误差<0.1% |

| 岗位方向 | 项目名称 | 技术栈明细 | 核心功能模块 | 落地步骤 | 简历量化亮点 |
| --- | --- | --- | --- | --- | --- |
| **算子开发** | 自定义高性能卷积算子 | C++、CUDA、PyTorch C++扩展、自动微分、性能对比 | 1. 算子实现：基于CUDA实现卷积核（支持不同核大小/步长）；<br>2. 自动微分：实现反向传播算子；<br>3. 集成：接入PyTorch框架；<br>4. 对比：与PyTorch原生算子性能/精度对比 | 1. 推导卷积算子数学公式；<br>2. 编写CUDA核函数实现前向计算；<br>3. 实现反向传播算子；<br>4. 封装为PyTorch扩展并测试 | 1. 性能较PyTorch原生卷积算子提升30%，内存占用降低15%；<br>2. 支持3×3/5×5卷积核，步长1-3，精度损失<0.5%；<br>3. 兼容PyTorch自动微分，可直接用于模型训练 |
| | TensorRT多头注意力算子优化 | C++、TensorRT、ONNX、精度校准、推理加速 | 1. 算子转换：ONNX导出多头注意力算子；<br>2. TensorRT优化：算子融合、FP16精度校准；<br>3. 推理引擎：构建优化后的推理引擎；<br>4. 对比：优化前后推理速度/精度 | 1. 导出PyTorch多头注意力算子为ONNX；<br>2. 使用TensorRT进行算子融合和精度校准；<br>3. 构建推理引擎并测试；<br>4. 调整算子并行粒度优化性能 | 1. 推理速度提升40%，批量处理（batch=32）耗时从8ms降至4.8ms；<br>2. FP16精度校准后，精度损失<1%；<br>3. 适配TensorRT 8.x，支持动态batch_size |

## （二）项目加分项（落地方法）

| 加分项类型 | 具体内容 | 落地步骤 | 简历呈现方式 |
| --- | --- | --- | --- |
| 开源贡献 | 向muduo/brpc/PyTorch提交PR | 1. 阅读开源项目源码，定位小BUG/优化点；<br>2. 提交Issue并沟通；<br>3. 编写修复代码，提交PR；<br>4. 跟进审核并修改 | 1. 开源贡献：为muduo框架修复epoll ET模式下的连接泄漏问题（PR #xxxx已合并）；<br>2. 优化：为PyTorch自定义算子添加性能注释，提升可读性 |
| 竞赛经历 | 数学建模/蓝桥杯/C++竞赛 | 1. 组队参加HPC相关竞赛（如全国大学生数学建模竞赛）；<br>2. 聚焦并行计算/性能优化方向；<br>3. 整理竞赛成果（论文/代码） | 1. 全国大学生数学建模竞赛省级二等奖（基于MPI实现气象数据并行处理）；<br>2. 蓝桥杯C/C++组省赛一等奖（高性能服务器开发） |

| 加分项类型 | 具体内容 | 落地步骤 | 简历呈现方式 |
| --- | --- | --- | --- |
| 技术博客 | 解析项目难点/技术原理 | 1. 梳理项目中的核心技术（如epoll原理/CUDA优化）；<br>2. 在CSDN/掘金撰写图文教程；<br>3. 附代码链接和效果对比 | 1. 技术博客：《CUDA图像滤波优化实战：从串行到GPU加速50倍》（阅读量1w+）；<br>2. GitHub开源项目：xxx（star 50+） |

# 四、求职全流程指南（从岗位搜索到面试）

## （一）岗位精准搜索（分方向，附关键词/适配企业）

| 岗位方向 | 核心搜索关键词 | 常见岗位变体 | 适配企业列表 |
| --- | --- | --- | --- |
| **C++后端开发** | C++后端开发实习、服务端开发实习、分布式系统实习、中间件开发实习 | 1. 后端研发实习生(C++方向)；<br>2. 高性能网络开发实习；<br>3. 金融科技C++开发实习；<br>4. 分布式存储开发实习 | 互联网大厂：腾讯/阿里/字节跳动/百度；<br>金融科技：蚂蚁金服/京东数科/陆金所；<br>云计算：阿里云/腾讯云/华为云；<br>中小企业：各类创业公司（ToB/ToC） |
| **HPC** | 高性能计算实习、CUDA开发实习、并行计算实习、超算研发实习 | 1. HPC开发实习生；<br>2. GPU计算实习生；<br>3. 超算应用开发实习；<br>4. 数值模拟算法实习 | 超算中心：国家超算天津中心/深圳中心；<br>科研院所：中科院计算所/自动化所；<br>企业：NVIDIA/华为/字节跳动（AI Lab）/商汤科技；<br>自动驾驶：特斯拉/小鹏/理想 |
| **算子开发** | 算子开发实习、AI高性能计算实习、深度学习算子优化实习 | 1. AI框架底层开发实习；<br>2. 异构计算算子适配实习；<br>3. 大模型推理优化实习；<br>4. 芯片算法适配实习 | AI框架：百度（PaddlePaddle）/字节跳动（ByteDance AI Lab）；<br>芯片厂商：NVIDIA/寒武纪/地平线/壁仞科技；<br>大模型企业：OpenAI（国内分支）/智谱AI/商汤科技 |

## （二）求职渠道（按优先级，附使用技巧）

| 渠道类型 | 推荐平台/方式 | 优势 | 详细使用技巧 |
| --- | --- | --- | --- |
| 核心招聘平台 | 1. 实习僧<br>2. BOSS直聘<br>3. 牛客网<br>4. 智联校园/前程无忧校园 | 1. 实习真实度高；<br>2. 沟通反馈快；<br>3. 技术岗集中+内推多；<br>4. 覆盖广 | 1. 实习僧：筛选“可转正”+“大厂”标签，按发布时间排序（优先7天内），每天刷10个目标岗位；<br>2. BOSS直聘：完善技术栈标签（C++/CUDA/epoll），主动向HR打招呼（附简短自我介绍：211硕士+XX项目+掌握XX技能）；<br>3. 牛客网：关注“实习内推”板块，参与大厂专场招聘，投递后在评论区@HR；<br>4. 智联校园：筛选“国企/央企”“科研机构”，适合HPC方向 |
| 特色补充渠道 | 1. 企业官网/公众号<br>2. 高校就业网<br>3. 内推（师兄/牛客/脉脉）<br>4. 技术社区（GitHub/Stack Overflow）<br>5. 科研机构官网 | 1. 官方直招无中介；<br>2. 竞争小匹配度高；<br>3. 跳过简历筛选；<br>4. 硬核技术岗集中；<br>5. HPC专属 | 1. 企业官网：关注目标企业招聘公众号（如“字节跳动招聘”“华为招聘”），设置实习岗位提醒；<br>2. 高校就业网：关注985/211高校就业网（如清华/北大/北航），HPC/算子开发岗位多；<br>3. 内推：找师兄师姐内推（附简历+项目链接），牛客网内推贴留言（注明“211硕士+XX方向+可实习6个月”）；<br>4. 技术社区：GitHub招聘板块筛选“C++”“HPC”“CUDA”关键词，Stack Overflow Jobs关注海外大厂远程实习；<br>5. 科研机构：国家超算中心/中科院官网，关注“实习招聘”栏目，投递时附科研相关经历（如有） |

## （三）投递策略（时间/频率/简历）

### 1. 投递时间规划

| 实习类型 | 投递时间窗口 | 备注 |
| --- | --- | --- |
| 日常实习 | 全年可投，3-5月、9-11月为旺季 | 旺季岗位数量是淡季的2-3倍，优先投递；<br>每周一/周四投递（HR集中看简历） |
| 暑期实习 | 1-3月申请，5-8月实习 | 转正率最高（大厂可达30%-50%），1月开始准备简历和项目，2月集中投递 |
| 寒假实习 | 10-12月申请，1-2月实习 | 竞争较小，适合补充项目经验 |

### 2. 投递频率与节奏

- 每日投递：3-5家（精投，而非海投），重点岗位（大厂/高匹配）单独定制简历；
- 重复投递：重点公司间隔1个月可重复投递（如字节跳动/华为），每次投递更新简历（补充新项目/技能）；
- 跟进节奏：投递后3天未回复，通过BOSS直聘/邮件跟进（简短询问：“您好，我投递了XX岗位，想确认简历是否已查看，附我的项目链接：XXX”）。

### 3. 简历优化（分方向模板）

| 模块 | C++后端开发简历重点 | HPC简历重点 | 算子开发简历重点 |
| --- | --- | --- | --- |
| 技能栏 | 1. 编程语言：C++11/14/17（精通）、Python（基础）；<br>2. 核心技术：Linux网络编程（epoll/Socket）、MySQL、Redis、RPC、分布式系统；<br>3. 工具：gdb/valgrind/CMake/Git | 1. 编程语言：C++（精通）、CUDA（精通）、Python（基础）；<br>2. 核心技术：OpenMP/MPI、CUDA异构计算、Eigen/BLAS、性能分析（perf/Nsight）；<br>3. 工具：CMake/Git/Slurm | 1. 编程语言：C++（精通）、CUDA（精通）、Python（熟练）；<br>2. 核心技术：PyTorch/TensorFlow C++扩展、TensorRT、算子优化、异构计算（GPU/NPU）；<br>3. 工具：NVIDIA Nsight/ONNX Runtime/CMake |
| 项目经历 | 1. 突出高并发/分布式/数据库优化；<br>2. 量化成果（并发数/延迟/性能提升）；<br>3. 描述技术难点（如epoll优化/TCP粘包） | 1. 突出并行/异构编程/性能调优；<br>2. 量化提速倍数/硬件利用率；<br>3. 描述并行策略（数据分片/内存优化） | 1. 突出算子开发/优化/硬件适配；<br>2. 量化性能提升/精度损失；<br>3. 描述优化方法（算子融合/内存访问优化） |
| 其他 | 可补充后端相关竞赛/开源贡献（如muduo） | 可补充数学建模/超算竞赛经历 | 可补充深度学习相关项目/开源贡献（如PyTorch） |

## （四）笔面试准备（分方向）

### 1. 笔试重点

| 岗位方向 | 核心考点 | 刷题资源 | 备考策略 |
| --- | --- | --- | --- |
| **C++后端开发** | 1. C++基础：智能指针、内存管理、模板、多线程；<br>2. 网络编程：epoll、TCP/UDP、HTTP；<br>3. 数据库：MySQL索引、事务、SQL优化；<br>4. 算法：链表/树/动态规划/高并发场景题 | 1. LeetCode Hot 100；<br>2. 牛客网C++后端笔试真题；<br>3. 《剑指Offer》 | 1. 每天刷2道算法题（中等难度）；<br>2. 整理C++高频考点（如智能指针实现）；<br>3. 刷大厂历年笔试真题（腾讯/阿里/字节） |
| **HPC** | 1. C++基础：内存管理/多线程；<br>2. 并行计算：OpenMP/MPI原理/编程题；<br>3. CUDA：核函数/内存模型/性能优化；<br>4. 数学：矩阵运算/数值分析；<br>5. 算法：并行算法设计 | 1. 牛客网HPC笔试真题；<br>2. CUDA编程实战题；<br>3. 《并行程序设计导论》 | 1. 掌握OpenMP/MPI编程题（如并行求和/矩阵乘法）；<br>2. 理解CUDA性能优化原则；<br>3. 复习数值分析基础（线性方程组求解） |

| 岗位方向 | 核心考点 | 刷题资源 | 备考策略 |
| --- | --- | --- | --- |
| **算子开发** | 1. C++基础：模板/内存管理/多线程；<br>2. CUDA：核函数/优化；<br>3. 深度学习：计算图/自动微分/算子原理；<br>4. 算法：矩阵运算/性能优化题 | 1. PyTorch/TensorFlow官方文档；<br>2. 牛客网AI算法笔试真题；<br>3. CUDA编程题 | 1. 理解深度学习算子数学原理；<br>2. 练习CUDA算子编程（如卷积/池化）；<br>3. 掌握TensorRT优化流程 |

### 2. 面试重点（分类型）

| 面试类型 | 核心问题方向 | 回答技巧 |
| --- | --- | --- |
| 技术一面（基础） | 1. C++：智能指针实现、右值引用、内存泄漏排查；<br>2. Linux：epoll原理、进程/线程区别、IPC方式；<br>3. 项目：核心功能/技术选型/难点解决 | 1. 结合项目回答（如“我在TCP服务器中使用智能指针管理连接，避免内存泄漏”）；<br>2. 原理+实践结合（如“epoll ET模式需要循环读取数据，我在项目中通过XX方式实现”） |
| 技术二面（深度） | 1. 项目深挖：性能瓶颈/优化思路/替代方案；<br>2. 技术难点：高并发场景下的锁竞争/分布式一致性；<br>3. 技术视野：云原生/国产芯片/大模型算力 | 1. 提前复盘项目（列出3-5个难点+解决方案+优化效果）；<br>2. 表达技术思考（如“我尝试过XX优化方法，效果不好，后来改用XX，因为XX”）；<br>3. 关注行业热点（如“我了解到国产NPU在算子适配方面的挑战是XX”） |
| HR面 | 1. 实习时长/可到岗时间；<br>2. 职业规划；<br>3. 团队合作/抗压能力 | 1. 明确实习时长（如“可实习6个月，每周5天”）；<br>2. 职业规划贴合岗位（如“希望深耕C++后端，从高并发服务器开发向分布式架构方向发展”）；<br>3. 结合项目说团队经历（如“项目中我负责网络层，和同学配合完成线程池开发”） |

# 五、长期发展规划（实习到就业）

## （一）技能融合与拓展

| 基础方向 | 拓展方向 | 核心学习内容 | 适配岗位 |
| --- | --- | --- | --- |
| C++后端开发 | 高性能后端+CUDA | 1. CUDA编程；<br>2. GPU加速后端服务；<br>3. 异构计算框架 | 高性能服务器开发、AI后端开发 |
| HPC | HPC+算子开发 | 1. 深度学习算子原理；<br>2. 大模型算力优化；<br>3. 国产超算适配 | 大模型训练算力优化、超算中心AI算力 |

| 基础方向 | 拓展方向 | 核心学习内容 | 适配岗位 |
| --- | --- | --- | --- |
| 算子开发 | 算子开发+芯片适配 | 1. 国产NPU/CPU架构；<br>2. 跨硬件算子适配；<br>3. 编译器优化 | 芯片算法适配、编译器开发 |

## （二）职业发展路径

| 岗位方向 | 实习→校招路径 | 3-5年发展目标 |
| --- | --- | --- |
| C++后端开发 | 日常实习（积累项目）→暑期实习（转正）→校招（大厂后端开发工程师） | 1. 初级→中级：负责核心业务模块，解决高并发/分布式问题；<br>2. 中级→高级：参与架构设计，主导中间件/分布式系统开发 |
| HPC | 科研机构/大厂HPC实习→校招（HPC工程师）→资深HPC工程师 | 1. 初级→中级：负责并行程序开发/算力调优；<br>2. 中级→高级：主导超算集群适配/大模型训练算力优化 |
| 算子开发 | AI/芯片企业算子开发实习→校招（算子开发工程师）→资深算子优化工程师 | 1. 初级→中级：负责通用算子开发/优化；<br>2. 中级→高级：主导大模型核心算子/国产芯片算子适配 |