

GPU 高级调试与优化

GPU 的历史很短，只有十几年。但它发展迅猛，凭借强大的并行计算能力和高效率的固定硬件单元，在人工智能、区块链、虚拟和增强现实（VR/AR）、3D 游戏和建模、视频编解码等领域大显身手。而且这种趋势还在延续，基于 GPU 的应用和创新势头正猛。

但是从系统架构来看，针对 GPU 的架构转型还在进行过程中，目前 GPU 依然还处于外设的地位，还没有摆脱从属身份。因为这个根本特征，对 GPU 编程并不像对 CPU 编程那样直接，而调试和优化 GPU 程序的难度就更大了，要比 CPU 程序复杂很多。

本研习班从 GPU 的发展简史入手，沿着 GPU 的发展历程，从内部结构、硬件接口和软件接口三个维度螺旋推进，深入解析 GPU 编程的知识要点和难点。硬件方面，以 NVIDIA、AMD、INTEL 和 ARM 四大阵营的 GPU 产品为例，揭秘 GPU 的内部架构和执行逻辑，解读其强大并行能力的硬件基础。软件方面，覆盖编程模型（CUDA、OpenCL、OpenVX、DirectX 和 OpenGL）、驱动模型（DRM 和 WDDM）和工具三个层面。具体由以下 12 个主题讲解和 8 个动手试验组成。

时间：2018 年 5 月 18 日 - 5 月 20 日（周五-周日）

地点：上海

形式：实战演练、讲解和讨论点评

培训对象：使用 GPU 进行通用计算和图形加速的软件工程师（开发和测试）、技术经理和科研人员

主办单位：上海曜印网络科技有限公司、高端调试网站（advdbg.org）

序言：三轮演进（0.5 小时）

要点：VGA, TVGA, 硬件加速, 2D 加速和 3D 加速, Voodoo, 从可配置到可编程, G80, GPGPU, John Nickolls, Brook 和 CUDA, GPU 的四大功能模块

-----N 篇-----

第一部分：在 CUDA-GDB 中理解 CUDA（1.5 小时）

要点：CUDA 简要历史，CUDA 的 C 扩展，kernel 函数，WARP，CUDA 的线程组织，准备调试环境，单 GPU 调试和多 GPU 调试，远程调试，nvcc，-G 和 -g 选项，附加到已经运行的进程，启用 kernel 初始断点（break on launch），观察源代码和汇编指令，设置断点，单步跟踪，条件断点，观察 CUDA 的内建变量，PTX 指令集，理解 WARP、grid、block 和 thread，Grid-Stride Loops，观察 GPU 线程，观察 GPU 的调用栈，观察 GPU 的寄存器，观察错误信息，从 GPU 上打印信息

试验 1：编译和观察简单的 CUDA 程序

编译一个简单的 CUDA 程序，使用 CUDA SDK 中的二进制工具观察其内容，理解 CUDA 的编译过程和程序文件格式

试验 2: 改进和调试向量乘法程序

使用 *CUDA* 技术编写一个做向量乘法的小程序, 理解如何向 *kernel* 函数传递参数和传回计算结果

试验 3: 学习 *CUDA-GDB* 的基本用法

调试一个简单 *CUDA* 程序, 练习常用的 *CUDA-GDB* 扩展命令, 理解 *CUDA* 编程的关键概念

第二部分: 使用 *NSight* 调试 *CUDA* 程序 (1.5 小时)

要点: *NSight* 简介, 安装和设置环境信息, *CUDA 9.0.props*, 窗口布局, 在 VS 中编译 *CUDA* 程序, 产生调试信息 (-G), 本地调试模型, *NSight Monitor*, 设置断点, 观察变量, 在 *Cuda Info* 窗口中观察计算状态, *WarpWatch*, 调用栈, 源代码跟踪, *PTX/SASS* 汇编调试, 数据断点, *API Trace*, *OpenCL kernel* 追踪, 产生 GPU 转储 (core dump), 调优功能

试验 4: 使用 *Visual Studio* 和 *NSight* 调试 *CUDA* 程序

在 VS 中编译和调试一个典型的并行计算程序, 熟悉 *NSight* 提供的常用调试功能, 包括产生调试信息, 建立调试会话, 设置各种断点, 观察源代码和变量, 单步跟踪等

第三部分: 显存锥鉴 (1.5 小时)

要点: 系统架构, 内存映射, *PCI Aperture*, *GART*, *GTT*, 访问主内存, *UVA/UMA*, *Batch Buffer*, *CUDA* 中的内存类型, 内存共享, 内存复制, 使用本地共享内存 (shared memory), 使用 *CUDA memory checker* 检查内存问题 (越界访问), 使用 *Nsight* 的内存调优功能

第四部分: 使用 *NSight* 调试图形程序 (1.5 小时)

要点: 感受 3D 图形魅力, 从 *DX9* 到 *DX12*, *WDDM*, *UMD*、*KMD* 和 *DX* 运行时, *DRM*, *GEM* 和 *KMS*, 启用 *DRM* 的调试功能, *HLSL*, *DirectCompute* 的遗憾, *PIX*, *Intel GPA*, *HUD*, *HUD* 图表, 热键, *HUD* 控制界面, VS 的图形分析器 (*VSGA*), *Nsight* 的图形调试

试验 5: 使用 *NSight* 调试 3D 图形程序

在 VS 中编译和调试一个典型的 3D 程序, 使用 *HUD* 功能观察 GPU 工作细节, 使用 *VSGA* 深入分析某一帧画面的产生细节, 认识 *DirectX* 软件栈中的关键组件和执行过程

第五部分: *NVidia GPU* 微架构 (1.5 小时)

要点: *G80*, 从 *SIMD* 到 *SIMT*, *warp*, *SM* (Streaming Multiprocessors), *Fermi* 微架构, *PTX* 指令集, *GigaThreads* 调度器, *ECC* 支持, 第三代 *SM*, *Kepler* 微架构, *Hyper-Q*, *Grid Management Unit* (*GMU*), *SMX*, 动态并行, *Maxwell* 微架构, *SMM*, 指令缓存, *WARP* 调度器, 指令分发单元, *Pascal* 微架构, 伏特微架构, *Tensor Core*, 软件仿真 (*GPUSim*)

----I 篇---

第六部分: 在 *Code-Builder* 中理解 *OpenCL* (1 小时)

要点: *OpenCL* 版本, 标准导读, 执行模型, 运行时, *OpenCL* 的执行硬件, *CPU* 模拟, *SIMD*, *kernel* 函数, *Code-Builder* 简介, 离线编译和在线编译, *NDRange*, 启动算核函数, 使用 *Code-Builder* 的调试功能, 观察 device 信息, context, 对象树, 命令队列, 内建函数

第七部分：计算机视觉加速接口(OpenVX)和英特尔实现（1.5 小时）

要点：OpenVX 简介，框架对象，数据对象，图（graph），节点，节点参数，执行模型，回掉，用户 kernel，常用功能的接口函数，Intel CV SDK，Vision Algorithm Designer（VAD），自动产生 C++代码，人脸检测实力分析和演示

试验 6：使用 Code-Builder 调试人脸检测程序

调试 Intel CV SDK 中的人脸检测示例程序，练习常用的调试功能，理解 OpenVX 的工作原理

第八部分：英特尔 GPU 综述（50 分钟）

要点：Intel 显卡简史，GEN 架构，固定功能单元和通用计算单元（EU），EU 结构，Slice 和 SubSlice，公开的编程手册，寄存器，GRF 和 ARF，VLIW，GEN 指令集，LINUX 驱动，i915，SRB 驱动，开源项目（Beignet，NEO，CM，IGC）

-----优化篇-----

第九部分：使用 CUDA profiler 优化 CUDA 程序（1.5 小时）

要点：测量 GPU 的时间，nvprof，命令行选项，指定收集范围，定义输出目标，Visual Profiler 基础，配置远程目标，导入数据，观察时序图，识别重要事件：CPU 缺页，GPU 缺页，数据迁移，内存复制；采样视图，分析热点，源代码和汇编结合分析，Profile API，定义别名，定制标记，创建调优会话

试验 7：使用 nvprof 和 Visual Profiler 优化 CUDA 程序

使用 nvprof 观察 CUDA 程序的执行细节，收集性能数据，在 Visual Profiler 中进行深入分析

第十部分：使用 GPUView 分析 CPU 与 GPU 交互（1 小时）

要点：ETW 基础，log.cmd，收集事件，识别典型问题：GPU/CPU Starvation，CPU/GPU Idle，线程切换，分析线程切换原因，实例演示

试验 8：使用 GPUView 观察 GPU 的工作状态

安装 GPUView，使用 GPUView 分析一个典型 3D 图形程序，理解 GPUView 的常用功能

-----A 篇-----

第十一部分：AMD GPU 和 HSA（1.5 小时）

要点：从 ATI 到 AMD，HSA 联盟，公开的技术手册，APU 和 GCN，GCN3 解析，硬件架构，支持 HSA 的 LINUX 驱动，调试支持，软件工具链：GpuOpen.com，Radeon GPU Analyzer (RGA)，使用 GPU PerfStudio 调试图形程序，CodeXL 简介，使用 CodeXL 调试和优化 OpenCL 程序，CodeXL 功能演示

第十二部分：ARM GPU（1.5 小时）

要点：Mali，典型应用，在深度学习中的应用（inference），Midgard 微架构解析，内部结构，通用计算执行单元，DS-5 简介，DS-5 编译工具，DS-5 调试器，Streamline 性能分析工具，DS-5 IDE 介绍，系统时域范围的自下而上和自上而下分析方法，自定义标注，Log 的图形化显示，OpenCL Kernel 跟踪，找到代码热点，调用链分析，矩阵乘法的优化，ARM GPU 软件工具链，OpenVX 实现，开源代码解读，实例演示：在 Midgard GPU 上调试 OpenCL 程序（使用 OpenCL 加速的人脸检测应用）

讲师介绍



张银奎 (Raymond Zhang)，绰号“格蠹老雷”，1996 年毕业于上海交通大学信息与控制工程系，在软件产业工作 20 余年，一多半时间在 INTEL 公司的上海研发中心工作，其中五年多工作在 INTEL 的 GPU 部门，以调试显卡驱动代码和各类 GPU 有关的问题为乐。业余时间喜欢写作和参与各类技术会议，发文数百万字，探讨各类软件问题。2015 年起获微软全球最有价值技术专家 (MVP) 奖励。著有《软件调试》和《格蠹汇编》二书，曾经主笔《程序员》杂志调试之剑专栏。在多家跨国公司历任开发工程师、软件架构师、开发经理、项目经理等职务，对 IA-32 架构、操作系统内核、驱动程序、虚拟化技术、云计算、软件调优、尤其是软件调试有较深入研究。从 2005 年开始公开讲授“Windows 内核及高级

调试”课程，曾在微软的 Webcast 和各种技术会议上做过《Windows Vista 内核演进》、《调试之剑》（全球软件战役研究峰会）、《感受和思考调试器的威力》（CSDN SD2.0 大会）、《Windows 启动过程》、《如何诊断和调试蓝屏错误》、《Windows 体系结构——从操作系统的角度》（以上三个讲座都是微软“深入研究 Windows 内部原理系列”的一部分）等。翻译（合译）作品有《现代 x86 汇编语言编程》、《21 世纪机器人》、《观止——微软创建 NT 和未来的夺命狂奔》、《数据挖掘原理》、《机器学习》、《人工智能：复杂问题求解的结构和策略》等。

特邀讲师：Joey Ye



从显卡的 TVGA 时代开始便在当时处于领导地位的 Trident 公司工作，曾参与制定多项 GPU 设计规范，带领团队实现 cmodel，为 GPU 集成电路设计提供软件验证。后来加入英特尔公司亚太研发中心，主要负责 GPU 芯片的 Power On、调试、功能的研发以及 GPU 技术在游戏、AR/VR 和机器视觉上的应用。目前工作于 ARM 开放智能实验室，主要负责和领导深度学习在 ARM 平台上 GPU 和 CPU 的实现和加速。

在显卡和 GPU 领域耕耘二十年，对 GPU 领域的主流技术都有涉猎，精通 OpenCL 编译技术、3D 图形加速技术等。长期从

事 GPU 底层软件的开发和调试，对嵌入式系统的 CPU 和 GPU 协同工作有深刻理解。擅长调试复杂的交互问题和对优化各类 GPU 应用。

附录 1：往届研习班部分照片







附录 2：报名与收费

标准收费：7000 元每人

优惠条款：

- 距离上课日 15 天前报名可以享受 8 折优惠
- 6 人及以上可以享受团购价 4550 元每人

报名或垂询

1) 联系课程顾问：

- a) Lisa Zhang ， 电话：13801874134， 电子邮件：lisa.zhang@leshanting.cn
- b) Cindy Long， 电话：13621638537， 电子邮件：cindy.long@leshanting.cn

2) 识别下图二维码，关注格友公众号提交报名信息

公司付款

收款单位：上海曜印网络科技有限公司

银行账号：**1001122409003035262**

开户行：中国工商银行上海分行静安新城支行

