Pearson

异步图书
www.epubit.com

Vulkan 应用开发指南

Vulkan Programming Guide

[美]格拉汉姆·塞勒斯（Graham Sellers） 约翰·克赛尼希（John Kessenich）著
李晓波 等 译

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

Vulkan 应用开发指南

Vulkan Programming Guide

[美]格拉汉姆·塞勒斯（Graham Sellers） 约翰·克赛尼希（John Kessenich）著

李晓波 等译

人民邮电出版社

北京

图书在版编目（CIP）数据

Vulkan应用开发指南 / （美）格拉汉姆·塞勒斯（Graham Sellers），（美）约翰·克赛尼希（John Kessenich）著；李晓波等译. -- 北京 : 人民邮电出版社，2019.6
ISBN 978-7-115-50680-1

Ⅰ. ①V… Ⅱ. ①格… ②约… ③李… Ⅲ. ①图形软件—程序设计指南 Ⅳ. ①TP391.41-62

中国版本图书馆CIP数据核字(2019)第019537号

◆ 著　[美] 格拉汉姆·塞勒斯（Graham Sellers）
[美] 约翰·克赛尼希（John Kessenich）
译　李晓波　等
责任编辑　谢晓芳
责任印制　焦志炜

◆ 人民邮电出版社出版发行　北京市丰台区成寿寺路 11 号
邮编　100164　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
三河市君旺印务有限公司印刷

◆ 开本：800×1000　1/16
印张：21.75
字数：488 千字　2019 年 6 月第 1 版
印数：1－2 400 册　2019 年 6 月河北第 1 次印刷

著作权合同登记号　图字：01-2017-7892 号

定价：89.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315
广告经营许可证：京东工商广登字 20170147 号

内 容 提 要

本书系统地介绍下一代 OpenGL 规范 Vulkan，揭示了 Vulkan 的独特性和卓越的功能。本书主要内容包括：内存和资源、队列和命令、数据的移动、图像的展示、着色器和管线、图形管线对象、绘制命令、几何体的处理、片段的处理、同步、数据的回读以及多渲染通道等。

本书适合图形程序开发人员、熟悉图形和计算 API 的程序员阅读，也可供对 Vulkan 感兴趣的专业人士阅读。

译 者 序

不知不觉中，我已经在 3D 游戏引擎研发领域摸爬滚打了十多年。其间曾带领团队自主研发过多款大型 3D 引擎，也深入研究过几乎所有常用的商用和开源引擎，如 CryEngine、Unreal Engine 4、Unity、Ogre3D、Cocos2d-x、Urho3D 等。这些引擎都使用 OpenGL 和 Direct3D 这两种传统的图形 API 进行实时渲染。这些年来，这两个图形 API 不停地更新换代，支持的功能也越来越多，但是它们的基本设计理念并没有改变。

从 2013 年开始，AMD 公司和 EA DICE 合作开发了新一代的图形 API——Mantle，用于取代 OpenGL 和 Direct3D，Mantle 有非常巨大的性能提升，尤其是在 CPU 方面。之所以有如此突出的优点，是因为这种 API 完全不同于传统的 API，主要特点有：“薄”驱动降低了 API 验证和处理的性能开销，充分利用了多核 CPU 的性能，以及显式的 API 调用。

先讲“薄”驱动带来的性能提升。由于图形 API 的应用场景差别非常大，支持传统图形 API 的驱动程序需要做非常多的验证工作，因此导致了 API 调用需要消耗很多时钟周期。在新一代图形 API 里，一个新的理念是应用程序的开发者更清楚实际的需求，因此将很多原本由驱动程序完成的工作转移到了程序员身上，于是驱动程序里的 API 验证和操作大幅减少，从而使得驱动程序更“薄”。根据官方实测，单单这一项就能将性能提升 9 倍。

另外，目前 CPU 的发展现状是，主频的提升基本上裹足不前，支持更多的内核是未来的发展方向。因此，多线程能够使得应用程序大幅提升性能。尽管传统图形 API 里也有多线程的概念，但都仅限于一些特殊的应用场景，如资源异步加载。虽然通过切换上下文也能同时在多个线程里调用图形 API，但是由于切换成本过大，反而会使性能大幅下降。为了提升每帧的绘制调用次数，商用引擎都会实现一种称为“渲染线程”的技术，原理就是，将图形 API 的调用从主线程转移到另外一个线程，这个线程只负责提交渲染命令，由此得名“渲染线程”。这种技术可以大幅提升每帧的绘制调用次数，但是理论上峰值也只能提升一倍（其他条件不变，并且性能瓶颈位于 CPU 上）。新一代图形 API 支持 “命令缓冲区”的概念，应用程序可以创建多个“命令缓冲区”。每

一帧中，每个线程都向对应的“命令缓冲区”中提交渲染命令，最后将这几个“命令缓冲区”一起提交。理论上，绘制调用的次数可以随着 CPU 的内核数量线性增长。对于个人电脑（Personal Computer，PC）和主机来说，很多应用程序的设计目的就是要“榨干硬件的每一个时钟周期”，这样就能够渲染更多的物体，提升场景的真实感。而对于移动端来说，能耗和发热问题使得在这种平台上不能像 PC 和主机那样长时间内使所有的 CPU 内核都满负荷运行，因为时间一长手机就会发热，从而导致硬件降频，帧率下降。经过实测发现，如果将一个线程里执行的任务分配给多个线程，就能够有效地缓解移动端由于发热而导致的降频。

最后，新一代图形 API 里，程序员拥有了更多的控制权，从而赋予了程序员根据具体应用场景对算法进行深度优化的能力。另外，很多过程也变得更加透明。虽然这增加了使用难度，但是非常值得。例如，把内存管理权限转交给了程序员，这样就不用再根据标志位猜测驱动内部是怎么实现的，也能根据实际的应用场景中是否申请了大量的小块内存、内存申请的频率等边界条件进行深度优化。

2015 年，Mantle 官方停止了更新。随后，基于这种设计理念（Vulkan 直接继承自 Mantle）诞生了 3 种新一代的图形 API——Vulkan、Metal 和 Direct3D 12。遗憾的是，后两者分别只支持 iOS/Mac OS 和 Windows 平台。作为一名“老”引擎程序员，我深知这意味着什么。引擎支持多个图形 API 会导致开发中大量的资源花在横向移植上，而不是花在优化算法这样的技术深度上，这对提升用户的体验是无益的。另外，目前支持多个图形 API 的引擎都会有一个针对各种图形 API 实现的抽象层，这样势必会增加性能开销。Vulkan 是个跨平台的图形 API，支持所有常用的平台，包括 Windows 系统、Linux 系统、Android 系统和 iOS/Mac OS。这无疑是图形引擎程序员的福音，真正做到了“一次编写，到处运行”。目前主流的游戏引擎都已经支持 Vulkan。

尽管 Vulkan 有诸多优点，但是学习曲线相对于传统图形 API 陡了很多，本书的出版恰逢其时。它是由 Vulkan 规范的制定者编写的。我也是首先通过本书英文版开始了解 Vulkan 的，并且业余时间也在尝试着翻译部分章节，希望能对国内的技术发展有所贡献。之后，我有幸认识了人民邮电出版社的张涛老师，便开始了本书的翻译工作。

翻译本书既让我激动，又让我倍感压力。首先，Vulkan 是一门全新的 API，出现了很多新的概念。原书作者尽管参与了 Vulkan 规范的制定，但仍然承认自己也还在学习中。另外，很多新的概念都是首次提出的，没有可以参考的相关中文资料，需要自己精确地翻译成中文，并且还要遵循中国人的思维习惯，因此在翻译过程中很多专业术语都经过了反复推敲。

翻译过程中，我一直战战兢兢，如履薄冰，生怕自己的翻译错误误导了读者。所以经过深思熟虑，打算借助技术社区的力量。幸运的是，翻译过程中遇到了很多志同道合的朋友，他们很乐意一起为中国的技术发展添砖加瓦。在此，首先要感谢的是参与了初稿翻译的朋友们，他们分别是邱龙云、马百川、梁跃、卢云庚、王冠群（排名按照贡献度）。然而，由于每个人的翻译水平、对 Vulkan 的理解程度、用语习惯都不一样，我又对初稿进行了 3 次大规模的审校，甚至部分章节又重新进行了翻译。此外，还要感谢王伟亮参与了终稿的审校。

最后，感谢妻子的默默奉献，为了帮助我尽快完成本书，她揽下了所有的家务，使我可以心无旁骛地将所有业余时间都花在这上面。也要感谢儿子尧尧，他的调皮捣蛋竟成了这个过程中最好的调味剂。

本书尽管经过了多遍审校，但是由于水平有限，难免有纰漏，诚恳地希望读者能够指出。

李晓波

译者简介

李晓波，2007 年毕业于北京理工大学，获得硕士学位，研究方向是“虚拟现实在化工场景动态搭建中的应用”。毕业后一直从事大型 3D 引擎的自主研发工作，带领团队研发过多款 3D 游戏引擎，并获得软件著作权。所研发的引擎已经应用于多款大型 3D MMO 客户端游戏中。也深入研究过几乎所有常用的商用和开源引擎，包括 CryEngine、Unreal Engine 4、Unity、Ogre3D、Cocos2d-x、Urho3D 等。两年前有感于游戏行业开发低效的现状，曾创立了北京疯狂引擎科技有限公司，专注于引擎技术服务，个人网站是 CrazyEngine 网站。业余时间密切关注 VR、AR、MR 的发展趋势，同时对技术培训方向也有兴趣，正在制作一些技术专题的教学视频。目前就职于北京一家中型手机游戏公司，负责研究 Unity 引擎源码，为公司的几个在研项目提供技术支持。

致　谢

首先，我要感谢 Vulkan 工作组的成员。经过不知疲倦和极长时间的工作，我们创作了本书，相信它会成为未来几年中计算机图形和计算加速坚实的基础。我尤其想要肯定 AMD 的同行在开发最初的 Mantle 规范中的贡献，而 Vulkan 源于此。

我要感谢本书的审校者 Dan Ginsburg 和 Chris“Xenon”Hanson，感谢他们提供的宝贵意见，没有这些反馈，本书肯定会包含更多错误和遗漏。我还要感谢我的同事 Mais Alnasser，他提供了很好的反馈，并进一步提升了本书的质量。还要感谢 AMD 的 Vulkan 团队的其他人，他们的工作使我能够在公众可以访问 Vulkan 之前测试大部分示例代码。

英文原书封面图片由 Agoro Design 的 Dominic Agoro-Ombaka 在短时间内制作。感谢他在这么紧的时间里制作了封面。

非常感谢编辑 Laura Lewin 以及 Addison-Wesley 团队的其他成员。他们允许我反复调整时间表，延期交稿，以随性的方式工作，对他们来说，这个过程通常是痛苦的。感谢他们对我如此信任。

最后，我要感谢我的家人——我的妻子 Chris 和我的孩子 Jeremy 和 Emily。“爸爸，你还在写你的书吗？”已成为我们家中最常听到的“咏叹调”。我感谢他们的耐心、爱和支持，有了这些我才能在过去的几个月里整理出了一本新书。

格拉汉姆·塞勒斯（Graham Sellers）

关于本书

这是一本关于 Vulkan 的书。Vulkan 是一种应用程序编程接口（Application Programming Interface，API），用于控制图形处理单元等设备。虽然逻辑上 Vulkan 继承自 OpenGL，但它与 OpenGL 在形式上完全不同。经验丰富的从业者会注意到，Vulkan 使用起来非常麻烦。你需要编写很多应用程序代码才能让 Vulkan 做一些有用的事情，更不用说炫酷的事情了。OpenGL 驱动程序所做的许多事情现在都是 Vulkan 应用程序编写者的责任了。这些事情包括同步、调度、内存管理等。因此，你会发现本书中有很多专门讨论这些主题的内容。当然，它们不仅适用于 Vulkan，还适用于一般主题。

本书的目标读者是熟悉其他图形和计算 API 的有经验的程序员。因此，书中对许多与图形相关的主题在没有深入介绍的情况下进行了讨论，有一些前向引用、代码示例是不完整的，仅进行局部说明，而不是你可以输入的完整程序。然而，本书网站上提供的示例代码是完整的，并经过了测试，可以作为一个很好的参考。

Vulkan 旨在用作大型复杂图形和计算应用程序与图形硬件之间的接口。以前由驱动程序实现的 API（如 OpenGL）所承担的许多功能和职责现在由应用程序承担。复杂的游戏引擎、大型渲染组件和商业中间件非常适合实现这些 API 的功能；它们比驱动有更多关于其特定行为的信息。Vulkan 不适合简单的测试应用程序，它还不是讲授图形概念的辅助手段。

本书前几章介绍了 Vulkan 和构建 API 的一些基本概念。随着对 Vulkan 系统的深入探讨，本书将讨论更多高级主题，最终产生一个更复杂的渲染系统，展示 Vulkan 的一些独特方面并讨论其功能。

第 1 章简要介绍了 Vulkan 及其基础概念。该章讲述了创建 Vulkan 对象的基础知识，并展示 Vulkan 系统入门的基础知识。

第 2 章介绍了 Vulkan 的内存系统，这也许是该接口最基础的部分。该章展示了如何分配内存，这些内存由 Vulkan 设备以及在应用程序内运行的 Vulkan 驱动程序和系统组件使用。

第 3 章介绍了命令缓冲区并讨论了向其中提交命令缓冲区的队列。该章展示了 Vulkan 进程如

何工作，以及如何为应用程序构建要发送到设备执行的命令包。

第 4 章介绍了几个 Vulkan 命令，这些命令都专注于移动数据。该章使用第 3 章讨论的概念来构建命令缓冲区，这些缓冲区可以复制和格式化存储在资源与内存（第 2 章介绍过）里的数据。

第 5 章讲述了如何将应用程序生成的图像显示到屏幕上。展示（presentation）是用于与窗口系统交互的术语，它是特定于平台的，因此该章深入研究了一些特定于平台的主题。

第 6 章介绍了 Vulkan 使用的二进制着色语言 SPIR-V。该章还介绍了管线对象，展示了如何使用 SPIR-V 着色器构建一个管线，并介绍了计算管线（在 Vulkan 中可用于完成计算工作）。

第 7 章介绍了图形管道，其中包括使用 Vulkan 渲染图元所需的所有配置。

第 8 章讨论了 Vulkan 中可用的各种绘图命令，包括索引和非索引绘制、实例化与间接命令。该章展示了如何将数据导入图形管线以及如何绘制更复杂的几何图形。

第 9 章深入讲解了 Vulkan 图形管线的前半部分，以及曲面细分和几何着色器阶段。该章展示了这些阶段可以完成的一些更高级的操作，并涵盖了直到光栅化阶段的管线。

第 10 章讲述了光栅化期间和之后发生的所有事情，这些工作用于将几何图形转换为可以向用户显示的像素流。

第 11 章介绍了 Vulkan 应用程序可用的各种同步原语，包括栅栏、事件和信号量。这些共同构成了有效利用 Vulkan 并行性的应用程序的基础。

第 12 章讨论了将 Vulkan 中的数据读入应用程序所涉及的问题。该章展示了如何按照时序安排 Vulkan 设备执行的操作、如何收集有关 Vulkan 设备操作的统计信息，以及如何将 Vulkan 生成的数据回读到应用程序中。

最后，第 13 章重新讨论了前面介绍的一些主题，将各个方面联系在一起以生成更高级的应用程序——使用复杂的多通道体系结构和多个队列进行处理的延迟渲染应用程序。

附录 A 包含了 Vulkan 应用程序可用的命令缓冲区构建函数表，提供了查看其属性的快速参考。

Vulkan 是一个庞大和复杂的新系统。在一本书中涵盖 API 的全部细节是非常困难的。除了本书之外，还鼓励读者彻底阅读 Vulkan 规范，以及阅读其他有关使用异构计算系统和计算机图形（使用其他 API）的图书。这些材料将为本书所涉及的数学和其他概念提供一个良好的基础。

关于示例源码

本书配套的示例代码可从 vulkanprogrammingguide 网站获取。有其他图形 API 使用经验的用户可能会觉得 Vulkan 非常复杂，这主要是因为本来由驱动程序承担的许多责任已委托给应用程序

了。但是，在许多情况下，简单的示例代码就可以很好地完成工作。因此，我们创建了一个简单的应用程序框架，该框架处理所有示例和应用程序中通用的大部分功能。这并不意味着本书是关于如何使用该应用程序框架的教程，只是为了保持示例代码简洁。

当然，当在整本书中讨论特定的 Vulkan 功能时，将包括代码片段，其中许多代码实际上可能来自本书的示例框架（而不是任何特定示例）。本书讨论的一些功能可能在代码包中没有示例。对于一些主要与大规模应用相关的高级功能尤其如此。这里没有简短的 Vulkan 示例。在许多情况下，单个示例程序演示了许多功能的用法。每个示例使用的功能都列在该示例的 readme 文件中。同样，本书中的示例和代码清单之间没有一对一的对应关系。

LunarG 的官方 Vulkan SDK 可从 LunarG 网站获取。在撰写本书时，SDK 版本是 1.0.22。较新版本的 SDK 可以兼容旧版本，因此建议用户在尝试编译和运行示例应用程序之前获取最新版本的 SDK。SDK 还附带了一些自己的示例，建议运行这些示例以验证 SDK 和驱动程序是否已正确安装。

除了 Vulkan SDK 之外，你还需要安装 CMake，以便为示例创建构建环境。你还需要一个最新的编译器。代码示例使用了 C++ 11 的几个特性，并且严重依赖于 C++标准库来处理线程和同步原语。众所周知，这些功能在各种编译器运行时的早期版本中都存在问题，因此请确保编译器是最新的。我们已经在 Windows 系统上使用 Microsoft Visual Studio 2015，并在 Linux 系统上使用 GCC 5.3 进行了测试。这些示例已在 64 位 Windows 7、Windows 10 和 Ubuntu 16.10 系统上进行了测试，最近的驱动程序来自 AMD、Intel 和 NVIDIA。

值得注意的是，Vulkan 是一个跨平台、跨供应商和跨设备的系统。其中许多示例应该适用于 Android 和其他移动平台。我们希望将来可以将示例移植到尽可能多的平台上，非常感谢读者的帮助和贡献。

勘误

Vulkan 是一项新技术。在撰写本书时，该规范刚在几周前开始使用。虽然作者和撰稿人参与了 Vulkan 规范的创建，但它庞大而复杂，并且有很多贡献者。书中的一些代码没有经过全面的测试，虽然我们相信它正确，但也可能包含错误。当我们将示例放在一起时，可用的 Vulkan 实现仍然存在 Bug，验证层没有捕获尽可能多的错误，并且规范本身存在漏洞和不清楚的部分。和读者一样，我们仍然在学习 Vulkan，所以尽管本书出于技术准确性进行过编辑，但是仍建议读者通过访问 vulkanprogrammingguide 网站来查看任何更新。

在 InformIT 网站上注册账号，以便在下载和更新可用时方便地访问。要开始注册，请转至 InformIT 网站并创建账户。输入英文原书 ISBN（9780134464541），然后单击 Submit 按钮。一旦完成该过程，你将在 Registered Products 下找到任何可用的资源。

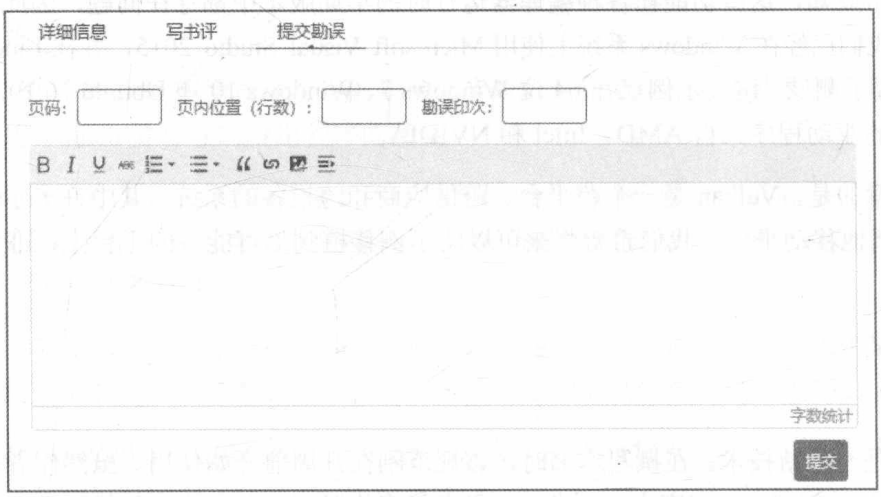
服务与支持

本书由异步社区出品，社区（https://www.epubit.com/）为您提供相关后续服务。

提交勘误

作者和编辑尽最大努力来确保书中内容的准确性，但难免会存在疏漏。欢迎您将发现的问题反馈给我们，帮助我们提升图书的质量。

当您发现错误时，请登录异步社区，按书名搜索，进入本书页面，单击“提交勘误”，输入勘误信息，单击“提交”按钮即可。本书的作者和编辑会对您提交的勘误进行审核，确认并接受后，您将获赠异步社区的 100 积分。积分可用于在异步社区兑换优惠券、样书或奖品。



扫码关注本书

扫描下方二维码，您将会在异步社区微信服务号中看到本书信息及相关的服务提示。

与我们联系

我们的联系邮箱是 contact@epubit.com.cn。

如果您对本书有任何疑问或建议，请您发邮件给我们，并请在邮件标题中注明本书书名，以便我们更高效地做出反馈。

如果您有兴趣出版图书、录制教学视频，或者参与图书翻译、技术审校等工作，可以发邮件给我们；有意出版图书的作者也可以到异步社区在线提交投稿（直接访问 www.epubit.com/selfpublish/submission 即可）。

如果您是学校、培训机构或企业，想批量购买本书或异步社区出版的其他图书，也可以发邮件给我们。

如果您在网上发现有针对异步社区出品图书的各种形式的盗版行为，包括对图书全部或部分内容的非授权传播，请您将怀疑有侵权行为的链接发邮件给我们。您的这一举动是对作者权益的保护，也是我们持续为您提供有价值的内容的动力之源。

关于异步社区和异步图书

“异步社区”是人民邮电出版社旗下 IT 专业图书社区，致力于出版精品 IT 技术图书和相关学习产品，为作译者提供优质出版服务。异步社区创办于 2015 年 8 月，提供大量精品 IT 技术图书和电子书，以及高品质技术文章和视频课程。更多详情请访问异步社区官网 https://www.epubit.com。

“异步图书”是由异步社区编辑团队策划出版的精品 IT 专业图书的品牌，依托于人民邮电出版社近 30 年的计算机图书出版积累和专业编辑团队，相关图书在封面上印有异步图书的 LOGO。异步图书的出版领域包括软件开发、大数据、AI、测试、前端、网络技术等。

异步社区

微信公众号

目　录