

Fundamentals基础概念解释

Host-cpu & Device-gpu 必要环境

- host:
 - 8,16,32,64bit int, 以byte为基本内存单位
 - 32,64bit IEEE float
- host & device:
 - 两个硬件endian、representation相同
 - *基元类型在两个硬件上都应该是易操作的

Architecture

- Vulkan Layered API
 - core Vulkan Layer
 - Debug Layer
 - Validation Layer: assert API调用

Execution Model (硬件多线程)

GPU多线程 -- devices进程 -- queue families线程池队列 (相似work) -- queues线程

- queue families: 相似性质、互相兼容的线程, 承接某一种work
- work类型: video encode/decode, graphics, compute, transfer, sparse memory management (gc)
- queue与queue family相互多映射

Memory heap -- typed memory area

所有内存类型均为device visible, device从属于application

- device local
- device local, host visible
- host local, host visible

CommandBuffer

可反复提交, 多线程创建、执行, 在一个gpu线程上同步执行 (可能被gpu排序), 与cpu异步执行

- queue submission commands (in batches)
 - vkQueueSubmit/vkQueueBindSparse, 提交完成立即join
 - 提交、阻塞、自旋均为queue operation
 - 不同queue的提交顺序不受semaphore/fence外的任何约束影响, 同一个queue的开始提交按照提交顺序和其他implicit sync order

- fence / semaphore: 开始时前序CB已完成, 前序内存已可读
 - 开始时等待一个semaphore, 结束时trigger一群semaphore/fence =》写转读
 - 所有顺序依赖仅依靠semaphore/fence, CB boundary没有任何作用
 - 提交后, CB state 在 CB boundary上重置
- 不同queue(承载CB)之间乱序执行
 - 只要Action CB的执行状态一致, 一起运行, 可交换顺序
 - 更换framebuffer
 - 读写image
 - 写入queue pool
 - State Setting CB更新current state, 不可一起运行
 - synchronization CB引入前后内存的explicit依赖, trigger action CB

Vulkan Object

所有API的“first parameter”必须是可分派对象

所有Object可访问性和作用域为分配/构造它的VkDevice, 不同VkDevice的vkObject互为external object handle, 需要export/import跨界传输

Object -- Handle

- dispatchable: UniquePtr, 指向实例对象
- non-dispatchable: 不可分派对象
 - 不可分派对象 由int64成员伪装而成时: 带info的假handle, 未标记privateData⇔非UniquePtr
⇔“指针数值”可能被多个对象复用, 不影响每个不可分派对象构造析构
- external: 必须被import&export

Object LifeCycle

构造后obj结构不变

- vkCreate* & vkDestroy*
 - 无pool => 低频, 更慢
- vkAllocate* & vkFree*
 - 从pool/memory heap中获取 => 高频, 更快
- ownership: app负责gc, 析构先子后父, 在CB执行完成后析构
 - VkShaderModule & VkPipelineCache 在传入CB后不再被调用, 但必须等待CB返回再析构
 - (由VkRenderPass/VkPipelineLayout传入的) VkDescriptorSetLayout 被操作Desc的
VkUpdateDescriptorSets访问, 必须在最后一次Update后析构
 - Desc/Buffer/Event/Pool被析构后, 所有未执行完成的CB进入 invalid state
 - CB/Sync CB完成前不可被析构
- 特殊vkObjects的父子关系

- vkDevice仅在所有queue idle时可析构
 - vkQueue在vkDevice卸载时析构
 - vkFence,vkSemaphore,vkCB,vkCBPool在线程返回时析构
 - vkXxxPool析构时自动析构vkXxx
- vkInstance在所有vkDevice析构时可析构
- vkPhysicalDevice不可析构， 仅在所有vkInstance析构时隐式析构

API Syntax

Application Binary Interface 跨平台能力

vk_platform.h as Shared Library

implementation: Application Binary Interface via c

- 数据结构: size, align, layout
- 函数calling convention:
 - VKAPI_ATTR/VKAPI_CALL 返参
 - VKAPI_PTR * 指针函数返参
- Symbol&Naming
 - 由vk开头的符号是ABI保留关键字, app不可使用
- 获取更高版本feature/Extension方式
 - vkGetInstanceProcAddr/vkGetDeviceProcAddr

Command Syntax & Duration

基本数据结构: c99,stdint.h

uint32_t VkBool32 -- VK_TRUE,VK_FALSE

uint64_t VkDeviceSize -- 内存大小&offset

uint64_t VkDeviceAddress -- 指针大小[buffer address value?]

构造函数:

vkCreate(VkCreateInfo, pAllocator)/vkDestroy

vkAllocate(vkAllocateInfo)/vkFree,池化obj沿用Pool的allocator

RetrieveResult: vkGet/vkEnumerate, 入参不变时将返回完全相同的返参

Queues

Threading Behavior

默认多线程, 传统mutex需要通过pthread library等第三方库构建

被定义为externally synchronized的vkObject/NON-const app-owned mem: 需要线程安全的SimpleStoreUpdate-> app为所有相关CB设置内部memory barrier, 确保CB不同时执行deferred host operations中的external sync需要维持到所有deferred operation执行完成

需要sync的所有params一定被显式标记为externally sync
immutable的所有params一定internally sync & CB完成后析构

常见external sync params:

- 对线程的操作: Queue, Fence, Semaphore
- 事件触发: Event
- 特殊资源: CB本身, swapchain
- 可更新资源: Desc, Conversion
- 将被Bind/Map的资源: Memory/Buffer
- 将被CBCreate/Destroy的资源: Sampler, ShaderModule, Pipeline, CBPool, DescriptorPool, Device
 - 构造时标有EXTERNALLY_SYNC的VkPipelineCache

常见external sync params 数组: 每个element分别保证线程安全

- 重置/释放资源: Fence, Desc, CB
- 更新/创建资源: Desc, SwapChain
- VkQueuePresentKHR最终显示: Semaphore, SwapChain

隐式external sync params: 不直接作为parameter传入, 但是会受到对特殊parameter执行相关CB操作的影响

- 对vkDevice析构时, 影响vkQueue
- 分配CB时, 影响CBPool

Pipeline Configuration & Numeric Representation

CoreLayer Error

- 可能程序终止, 不会影响OS, 仅遵循OS约定 (进程内存不越界, 初始值, use-after-free不引起其他进程内存泄漏)
- ValidUsage会根据RuntimeLimit&Feature变化, 仅作为Vulkan支持功能的最小集合, 比UB更严格, 是ValidationLayer的唯一依据
 - Action CB 入参合法
 - State CB 设置后, 在恰当的位置检查PSO

Valid Usage Conditions @ Validation Layer

仅编译时可知, 函数调用时对传参类型检测, 所有State提交完成时检测DrawState

Implicit Valid Usage: 无特别说明则Valid

- Object Handle: 成功创建, 未释放, 成员与PassBy合法

- VK_NULL_HANDLE/NULL可以被传入vkDestroy/vkFree
- pointer: 类型安全、align在cpu合法的内存
- string: char[], 以NULL结尾
- Enum: enumerant有定义, 非_MAX_ENUM(c 32b-Enum标志), 非为VulkanLayer内部预留的 reserved value
 - 为extension使用switch的default
- Flags位标志:
 - uint32_t == vkFlags
 - "enumType": Vk*Flags
 - "enumerant": Vk*FlagBits
 - 会存在未定义的bit flag位
 - 因c enumerant 最高 0x80000000, 仅30~0bit合法
 - uint64_t == VkFlags64
 - "enumType": Vk*Flags2
 - "enumerant": Vk*FlagBits2
 - 注意特殊的互斥flagbits也会导致flag不合法
- vk内置Structure:
 - VkStructureType sType成员, 表示Structure为XxxInfo, 与结构名对应
- Structure Pointer Chain:
 - void *pNext成员, 指向下一个ExtendingStructure/NULL
 - 不出现两个相同Structure
 - 遍历时跳过不支持的structure
 - vk提供的基本结构体:
 - VkBaseInStructure迭代cpu发送到gpu信息, readonly
 - VkBaseOutStructure迭代gpu返回cpu信息
 - 有nested结构体
- Extension:
 - Instance Level feature
 - 必须在vkEnumerateInstanceExtensionProperties中
 - 必须被vkInstanceCreateInfo开启
 - Physical-device-level feature
 - 必须被[ExtendingPhysicalDeviceCoreFunctionality](#)支持
 - Device Level feature
 - 必须在vkEnumerateDeviceExtensionProperties中
 - 必需被VkDeviceCreateInfo开启
- Newer Core Version:
 - Instance Level
 - vkEnumerateInstanceVersion

- VkApplicationInfo::apiVersion
- (Physical) Device Level
 - VkPhysicalDeviceProperties::apiVersion
 - VkApplicationInfo::apiVersion

State and State Query: Return Code 返回码

- VkResult-Success Code
 - VK_NOT_READY: fence/query未完成
 - VK_THREAD_IDLE_KHR: deferred operation未完成, 但当前线程已idle
 - VK_THREAD_DONE_KHR: deferred operation全部分派, 尚未完成
 - VK_SUBOPTIMAL_KHR: swapchain不再严格符合surface描述, 但依然可以工作
 - VK_INCOMPLETE: 返参数组还不够
 - VK_PIPELINE_COMPILE_REQUIRED: pipeline已经触发编译, 尚未完成
- VkResult-Error Code
 - VK_ERROR_*
 - VK_ERROR_INCOMPATIBLE_DISPLAY_KHR backbuffer不兼容
 - VK_ERROR_INVALID_OPAQUE_CAPTURE_ADDRESS 因内存不可达导致分配Buffer失败; ShaderGroupHandle因Info失效而分配失败;
 - VK_ERROR_UNKNOWN: 错误输入; application/implementation出错 -> 使用最新Validation Layer验证app;
- Error性质:
 - RuntimeError时, 所有output parameter UB
 - VK_ERROR_OUT_OF_*_MEMORY不会导致此前的Object UB
 - 一些性能关键的CB可能延迟error report

Numeric Representation & Computation

shader的range & precision 见[SPIR-V](#)

非shader的range & precision 见[data format](#)

<https://www.khronos.org/registry/DataFormat/specs/1.3/dataformat.1.3.html>

- 数据格式
 - 16b = 1 sign + 5 exponent + 10 mantissa
 - u11b = 5 exponent + 6 mantissa
 - u10b = 5 exponent + 5 mantissa
- 浮点数range & precision不合法不导致崩溃
 - 浮点数计算精度约为 $1e-5$
 - 浮点数magnitude $\geq 2^{32}$
 - unsign负数 $\Rightarrow 0$

- 超过magnitude \Rightarrow Inf或最大数
- rounding 方式未规定

Normalized Fixed-Point Int \Leftrightarrow normalized Floating Point Conversion

signed fixed-point integer: signed two's complement $\in [-2^{b-1} + 1, 2^{b-1} - 1]$ (非normalized)

floating point : $f = \max(\frac{c}{2^{b-1}-1}, -1.0) \in [-1, 1]$

注意 -2^{b-1} 在int中合法, 在fp中会被clamp

从fp转向int时, 必须round to nearest

Common Object Types

Offsets

framebuffer中的int32 pixelPos.xy

Extents

framebuffer中的int32 pixelPos.xyz(width,height,depth)

Rectangles = Offset + Extent

API Name Alias

仅为命名不规范的API向前兼容