ThoughtWorks®

METAL

薛陶

AGENDA

- Metal 介绍
- Metal 基本结构
- Metal Process Image
- MPS (Metal Performance Shaders)

ThoughtWorks®

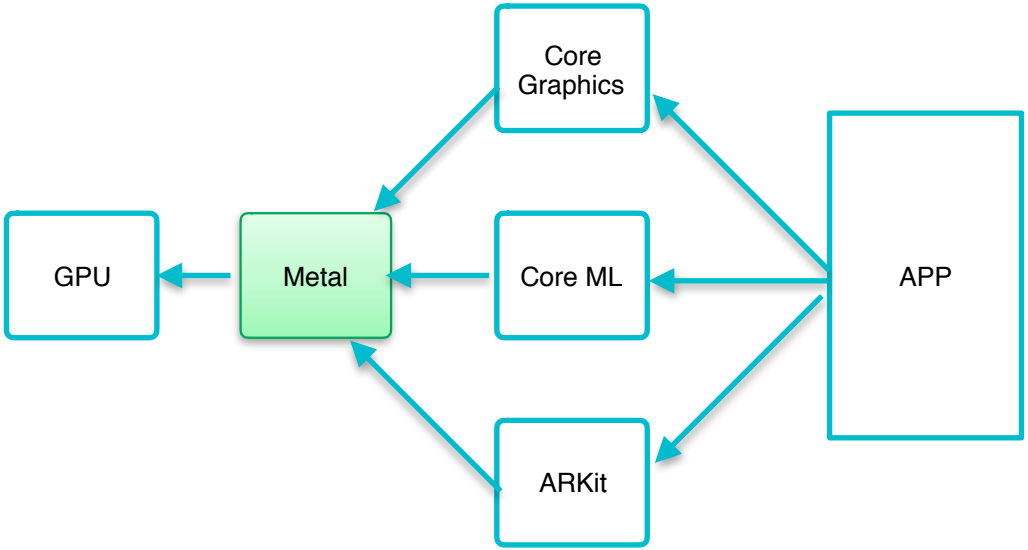
ThoughtWorks® IT大咖说 知识共享平台

METAL介绍

WHAT'S THE METAL

Metal 是一个和 OpenGL ES 类似的面向底层的图形编程接口，通过使用相关的 api 可以直接操作 GPU。

METAL层级关系



METAL特点

- 苹果平台独有
- 提供对GPU访问更底层权限
- 渲染性能快精度高
- 自定义的Shading Language
- 构造了一套IOS平台对内存和资源管理方式
- 有类似OpenCL的并行计算功能

METAL的应用场景

图形图像

meitu美图

视频

快手

游戏引擎

unity

AR/VR

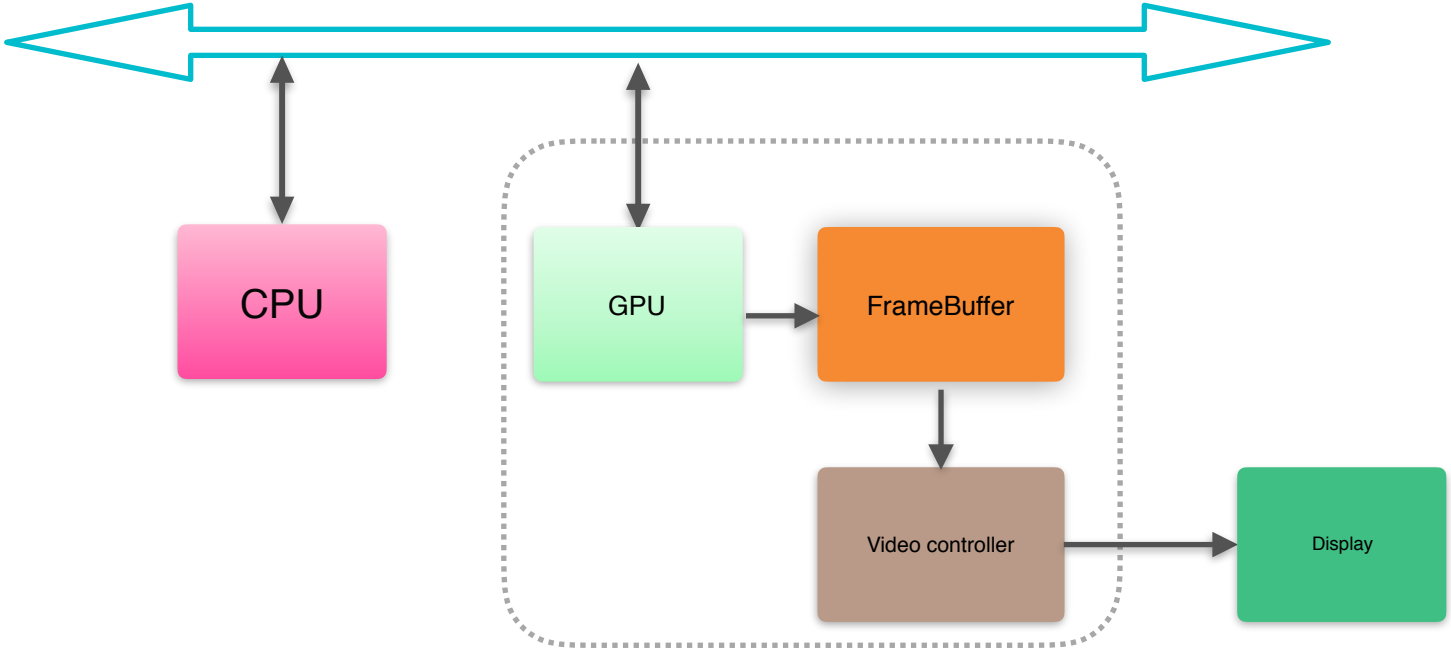
机器学习/深度学习

ThoughtWorks®

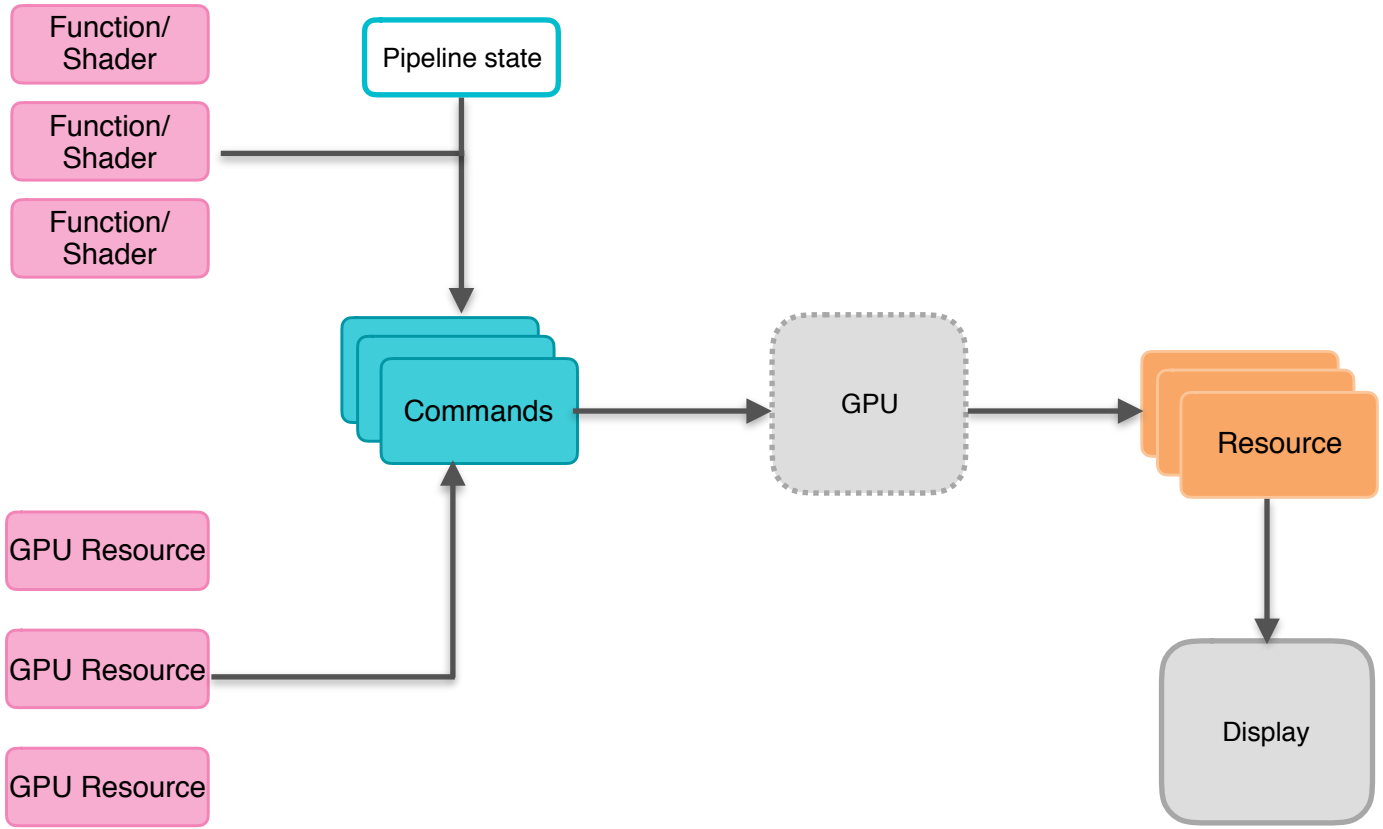
ThoughtWorks® IT大咖说 知识共享平台

METAL基本结构

GPU相关



METAL数据输入输出



FUNCTION / SHADER(着色器)

Function / Shader（着色器）是一段能够针对3D对象进行操作、并被GPU所执行的程序。

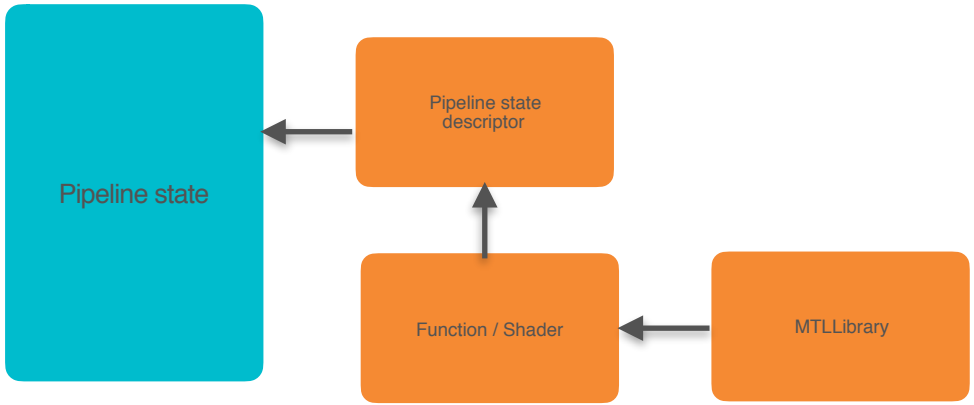
着色函数并不是一个统一的标准，不同的图形接口的着色器标准并不相同。

Metal的着色函数：

- Vertex Function
- Fragment Function
- Kernel Function

管线PIPLINESTATE

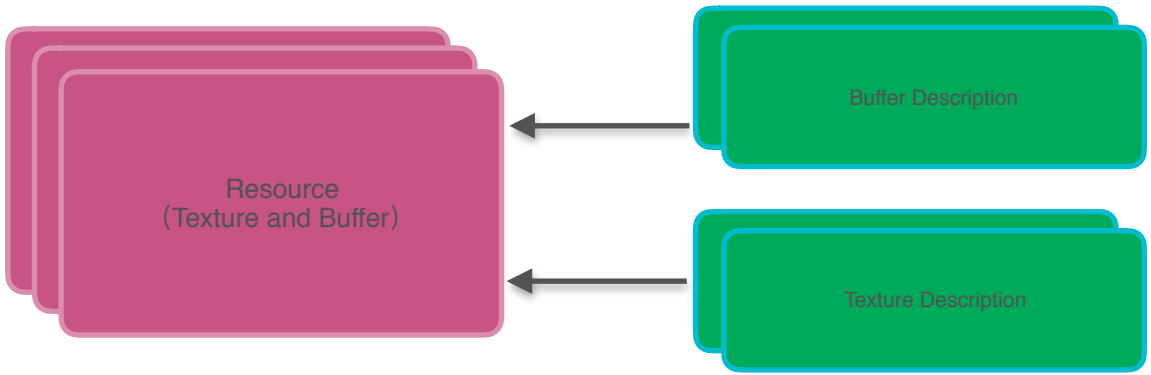
管线就好比是 CPU 和 GPU 直接的管道，通过它来配置运行在 GPU中的着色器



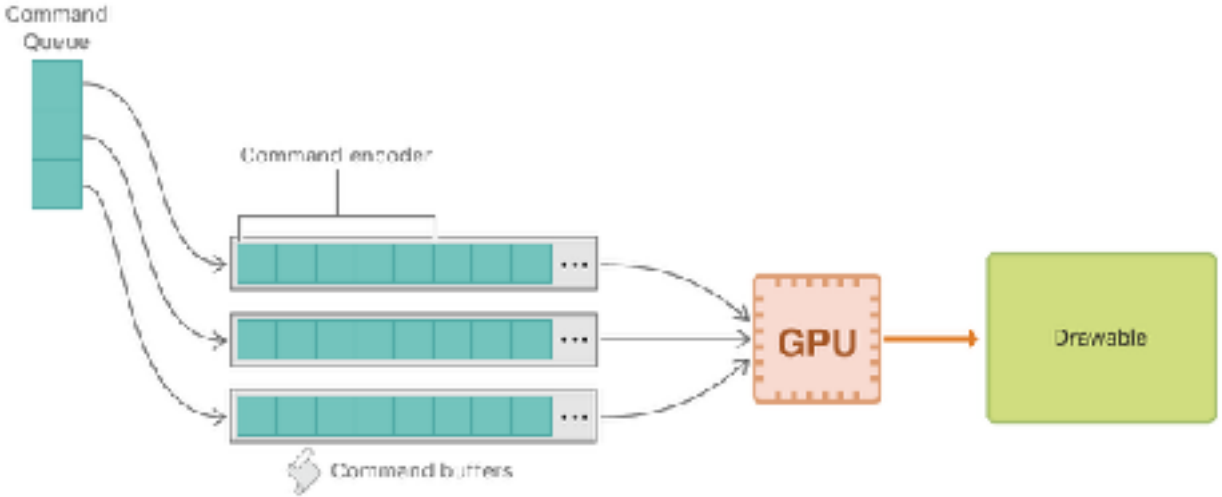
GPU RESOUCES

- MTLBuffer 代表着未格式化的内存，可以是任何类型的数据。 Buffer 用来做顶点着色和计算状态。

- MTLTexture 代表着有着特殊纹理类型和像素格式的格式化的图像数据。用来做顶点，面和计算的源



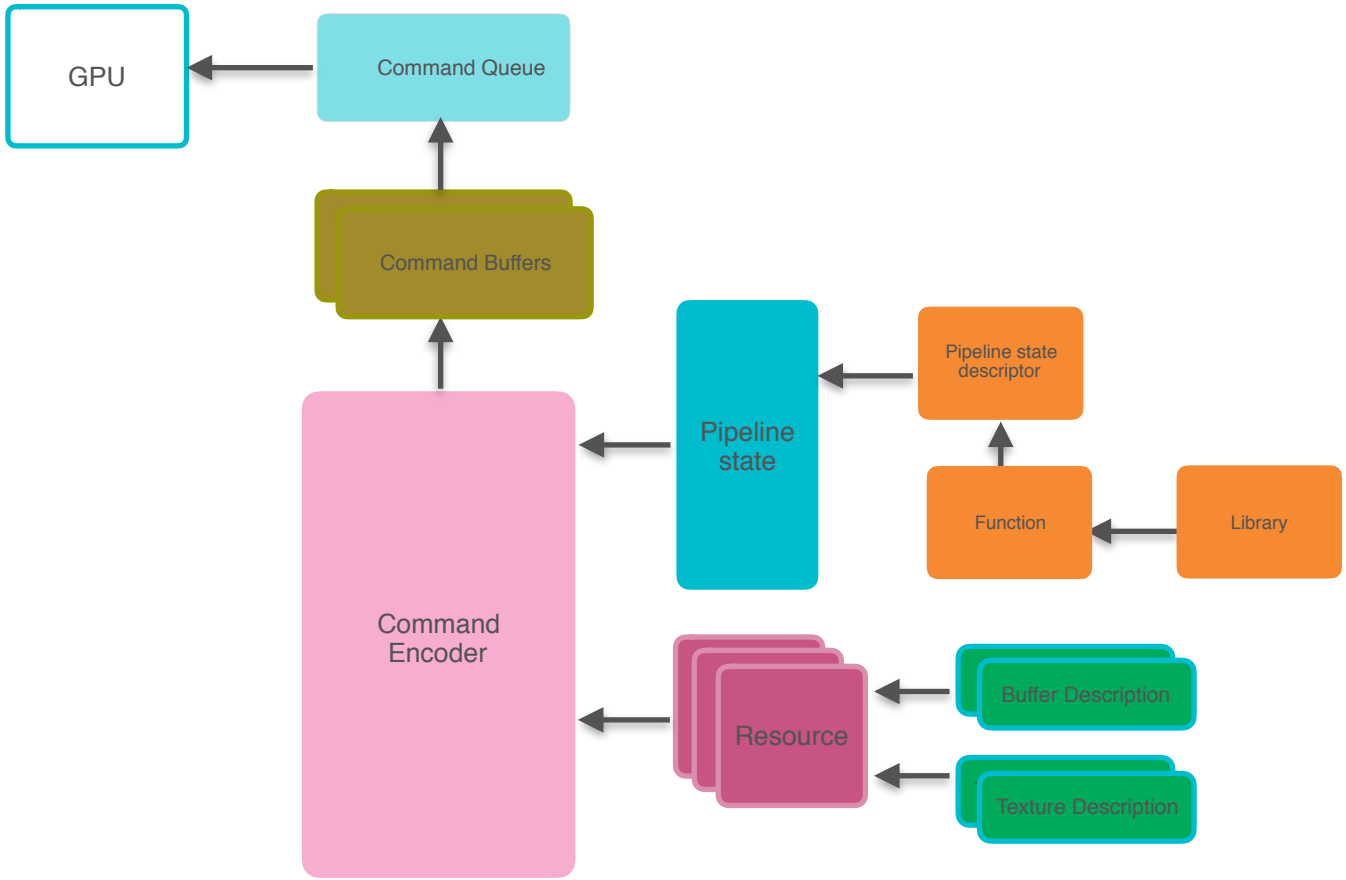
METAL COMMAND



Metal中Command中有三种 Encoder：

- MTLRenderCommandEncoder 渲染 3D 编码器
- MTLComputeCommandEncoder 计算编码器
- MTLBlitCommandEncoder 位图复制编码器，拷贝buffer texture同时生成mipmap

METAL结构图

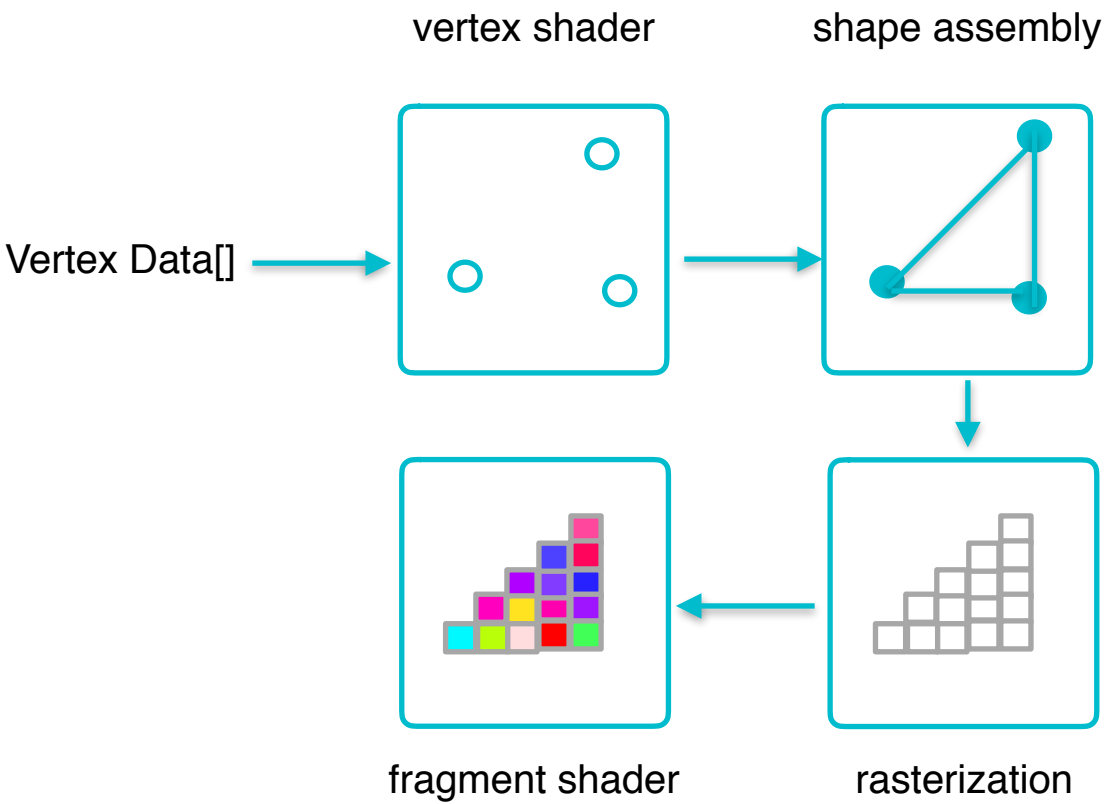

GPU
Command Queue
Command Buffers
Command Encoder
Pipeline state
Pipeline state descriptor
Function
Library
Resource
Buffer Description
Texture Description

METAL PROCESS IMAGE

内容提示

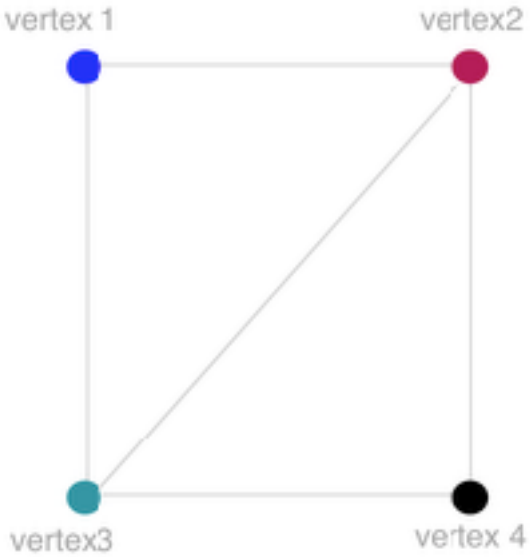
- 渲染图像
- Metal Compute

METAL渲染管线


vertex shader
shape assembly
Vertex Data[]
fragment shader
rasterization

顶点数据配置

顶点是展示位笛卡尔坐标系的集合，定义图形则是通过笛卡尔坐标系的集合来定义。



（vertex 1， vertex 2， vertex 3，
vertex 2， vertex 4， vertex 3）

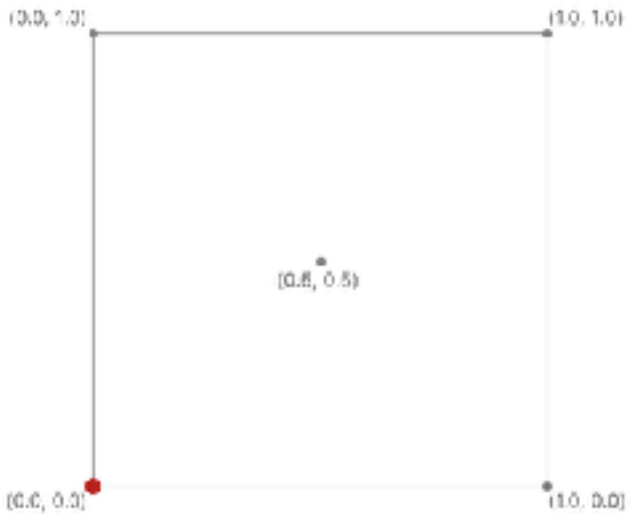
顶点着色器

顶点着色器被使用在传统的基于顶点的操作，例如位移矩阵、计算光照方程、产生贴图坐标。

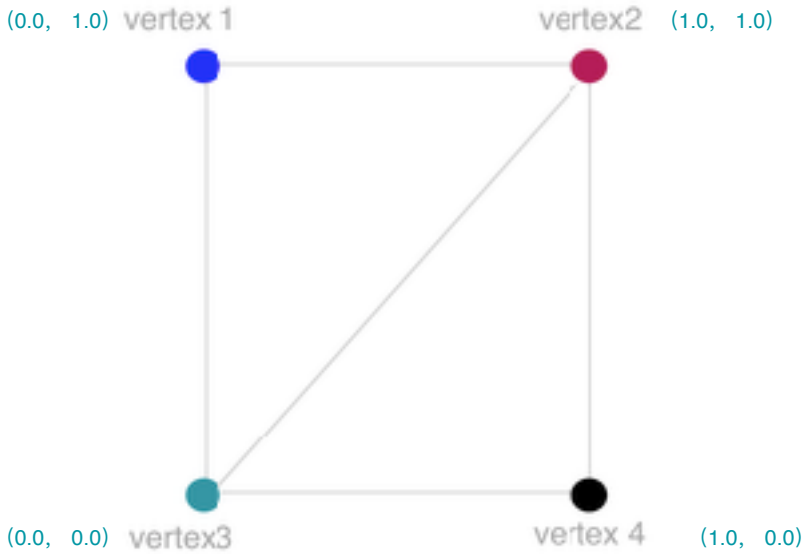
```
// Vertex Function
vertex RasterizerData
vertexShader(uint vertexID [[ vertex_id ]],
             constant AAPLVertex *vertexArray [[ buffer(AAPLVertexInputIndexVertices) ]])

{

    RasterizerData out;

    float2 pixelSpacePosition = vertexArray[vertexID].position.xy;


    out.clipSpacePosition.xy = pixelSpacePosition;

    out.clipSpacePosition.z = 0.0;

    out.clipSpacePosition.w = 1.0;

    out.textureCoordinate = vertexArray[vertexID].textureCoordinate;

    return out;
}
```

TEXTURE COORDINATES

对于2D的纹理，纹理坐标是从0.0到1.0的范围



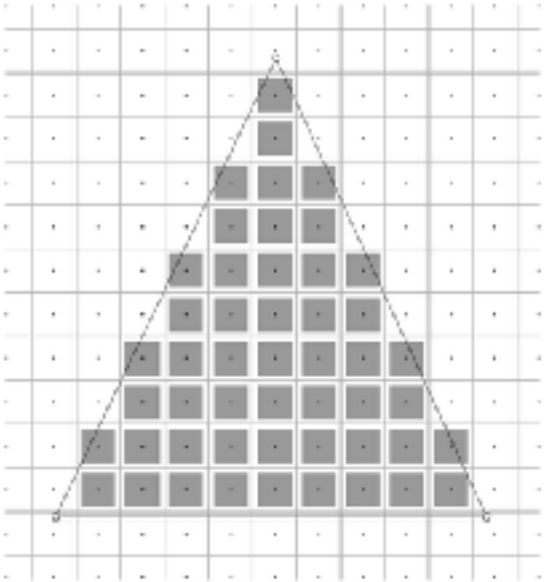
纹理坐标



顶点坐标

光栅化

在顶点函数执行三次以后，下一个状态管线光栅化开始光栅化状态，管线的光栅化单元生成片段。



片段着色器

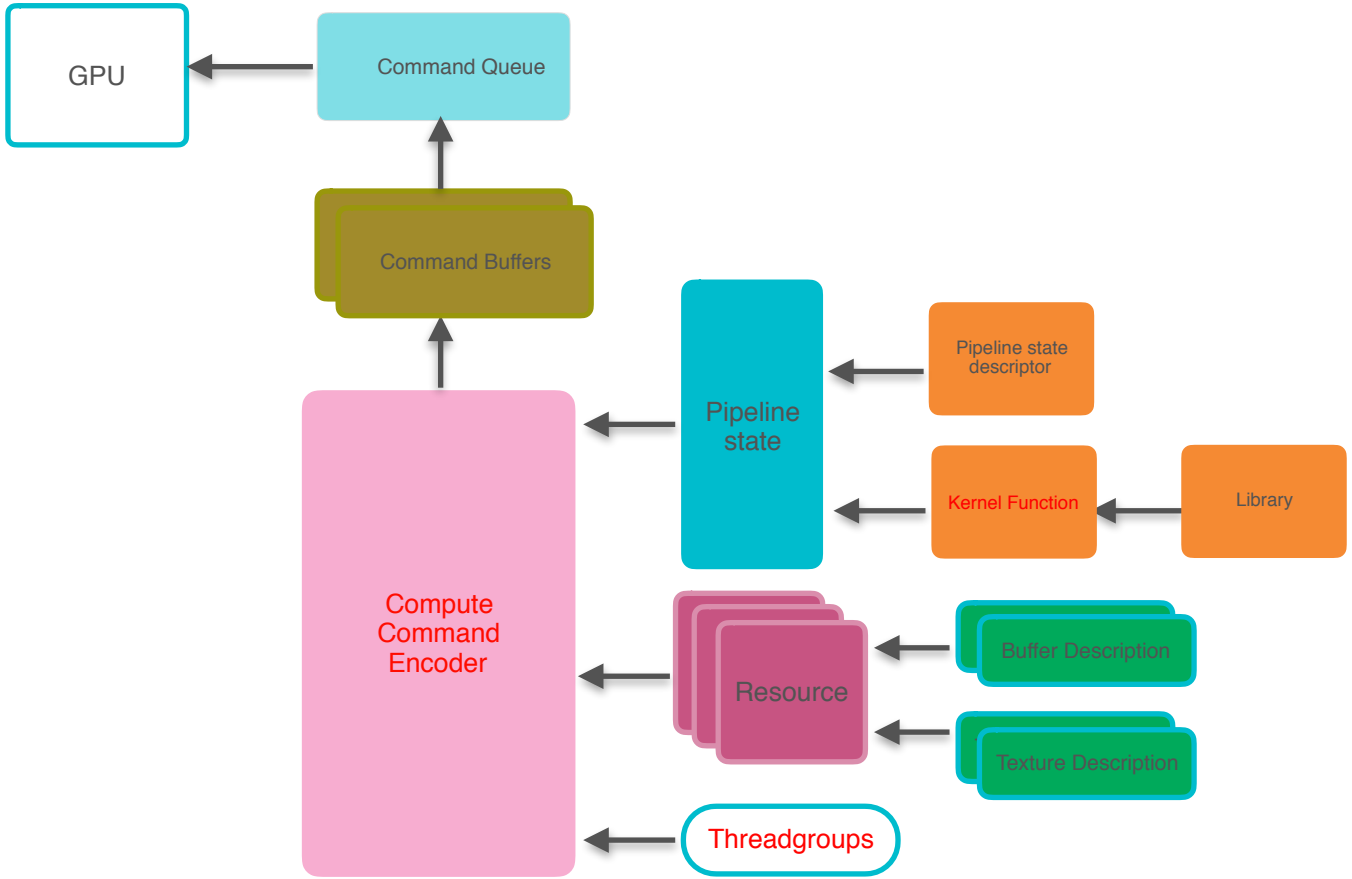
片段函数的主要任务是处理传进来的片段数据, 然后计算可绘画对象的像素的颜色值.

```
// Fragment function
fragment float4
samplingShader(RasterizerData in [[stage_in]],
               texture2d<half> colorTexture [[ texture(AAPLTextureIndexBaseColor) ]])
{
    constexpr sampler textureSampler (mag_filter::linear,
                                      min_filter::linear);

    const half4 colorSample = colorTexture.sample (textureSampler, in.textureCoordinate);

    return float4(colorSample);
}
```

METAL COMPUTE

通用计算是一个很广泛的概念，一般说的“通用计算”指的是“GPU通用计算”，意思就是用GPU来处理一些原本CPU可以处理的计算。

METAL COMPUTE



METAL COMPUTE STEPS

- 创建一个计算状态（MTLComputePipelineState）
- 指定资源和相关对象
- 指定compute command encoder
- 调度计算函数(compute function)执行指定的次数

METAL COMPUTE

compute shader:

```
constant half3 kRec709Luma = half3(0.2126, 0.7152, 0.0722);

kernel void
grayscaleKernel(texture2d<half, access::read>  inTexture  [[texture(AAPLTextureIndexInput)]],
                texture2d<half, access::write> outTexture [[texture(AAPLTextureIndexOutput)]],
                uint2                          gid         [[thread_position_in_grid]])
{
    if((gid.x >= outTexture.get_width()) || (gid.y >= outTexture.get_height()))
    {
        return;
    }

    half4 inColor  = inTexture.read(gid);
    half  gray     = dot(inColor.rgb, kRec709Luma);
    outTexture.write(half4(gray, gray, gray, 1.0), gid);
}
```

RESULT

before

after

ThoughtWorks®

ThoughtWorks® IT大咖说 知识共享平台

METAL PERFORMANCE SHADERS

METAL PERFORMANCE SHADERS

Metal Performance Shaders框架包含一系列高度优化的计算和图形着色器，旨在轻松高效地集成到Metal应用程序中。 这些数据并行原语被特别调整，以利用每个GPU系列的独特硬件特性来确保最佳性能。

METAL PERFORMANCE SHADER

在iOS 9和tvOS 9中，Metal Performance Shaders框架引入了一系列常用的图像处理算法，用于对Metal纹理进行图像效果。

- 卷积滤波器（Sobel，Gaussian）
- 形态运算符（扩张，侵蚀）
- 直方图运算符（均衡，规范）

图像处理DEMO

METAL PERFORMANCE SHADER

在iOS 10和tvOS 10中，Metal Performance Shaders框架为以下内核提供了额外的支持：

- 卷积神经网络（CNN）使用先前获得的训练数据实现和运行深入学习
- 图像处理进行颜色转换
- 矩阵乘法。

ThoughtWorks®

ThoughtWorks® IT大咖说 知识共享平台

THANKS