2020
北京

多媒体开启新视界
MULTIMEDIA BRIDGE
TO A WORLD OF VISION

视频推理服务
使AI应用更高效

爱奇艺视频推理服务实践

目录

CONTENTS

简　介

PART 1

简介

- 周海维
- 爱奇艺深度学习云架构师 （2017-2020）
- 爱奇艺国际部广告平台架构师

- 分享爱奇艺视频推理的工程经验和优化方法
- 重点介绍视频变化服务的工程优化方法
- 不涉及具体算法，任务治理

视频质量增强

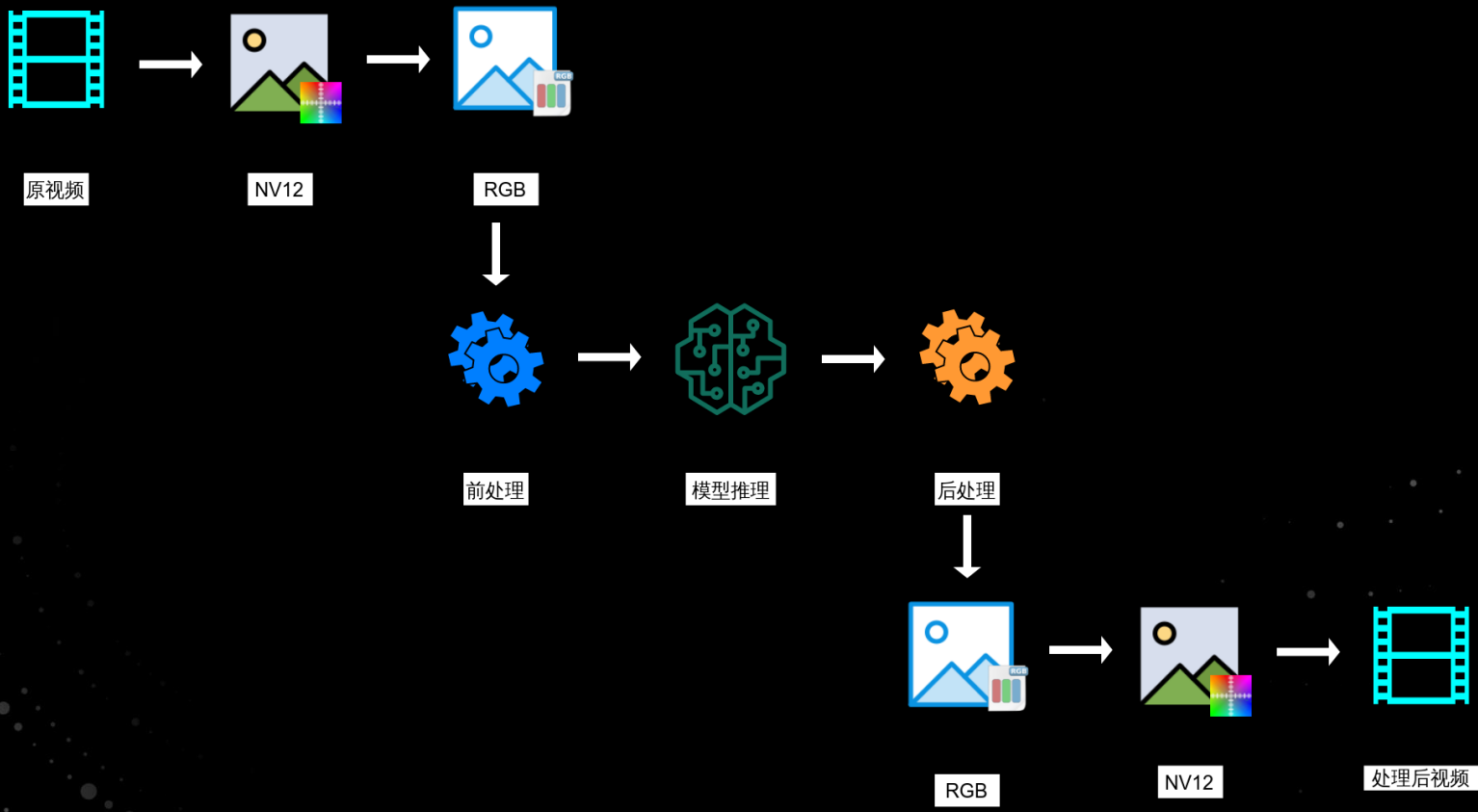
SDR2HDR

视频推理服务类型

2020
北京

	视频特征提取	视频变化
输入	视频	视频
输出	数值（概率，坐标）	视频
尺寸	缩小	保持或变大
帧率	只留关键帧/采样	保持或增大
计算量	大 可运行在CPU/GPU	巨大 一般运行在GPU
算法	CNN，RCNN	CNN，传统CV 主观性较强
用途	分类，物体识别	图像增强，高帧率，画面修复

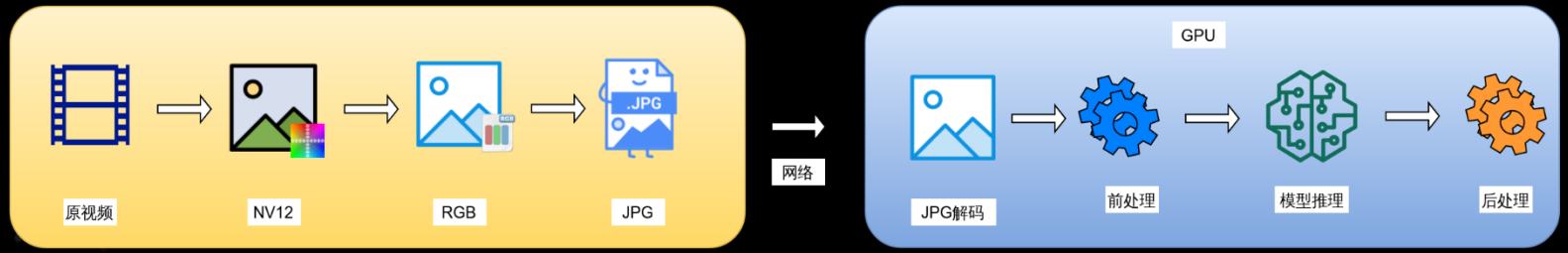
视频变化服务流程



现有方案

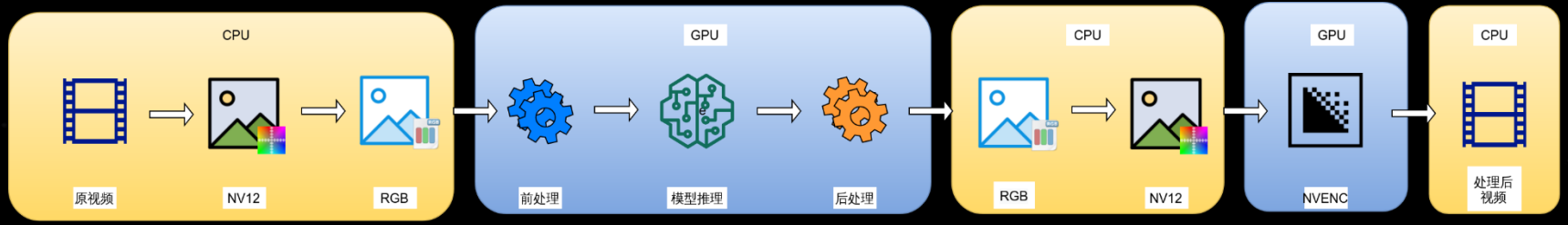
PART 2

从 图片推理 演进到 视频推理



- 单机CPU/GPU比例限制

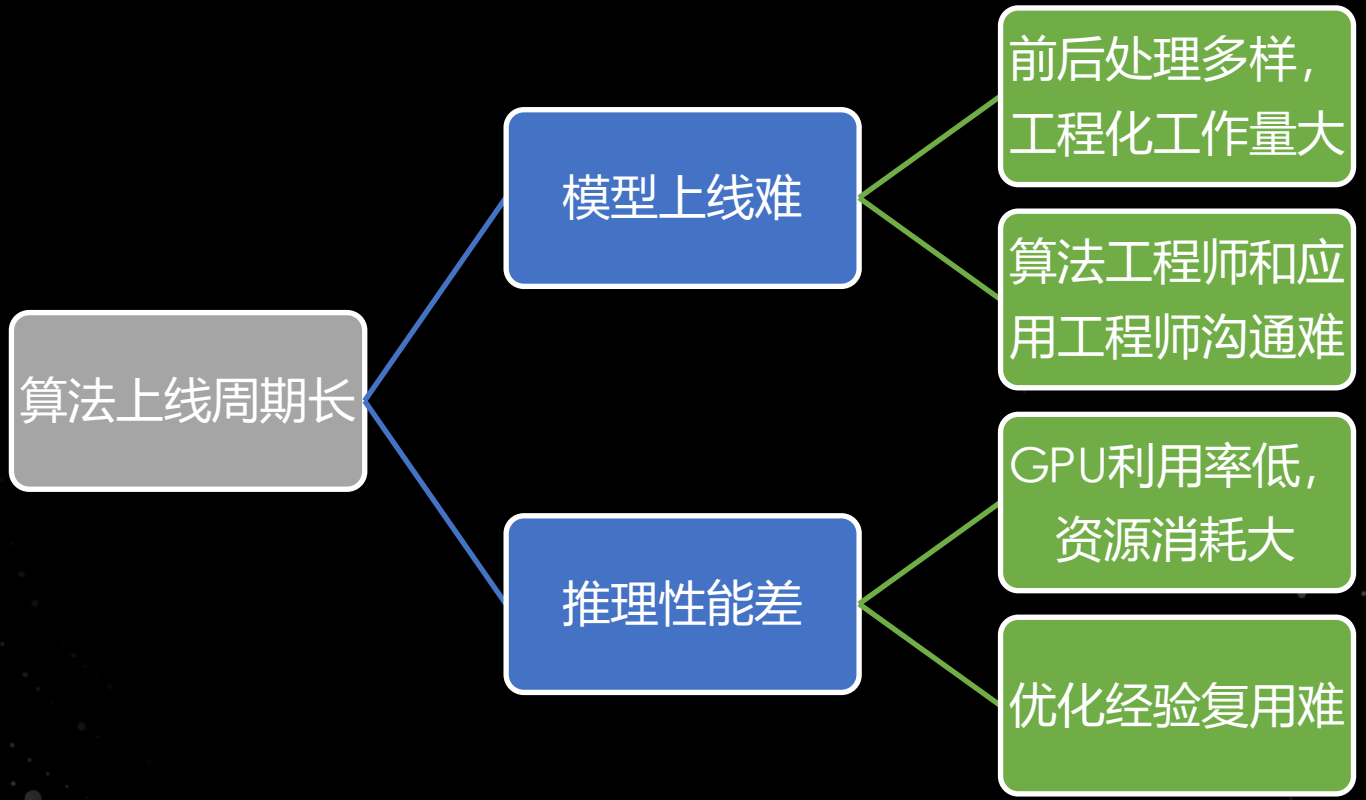
ffmpeg



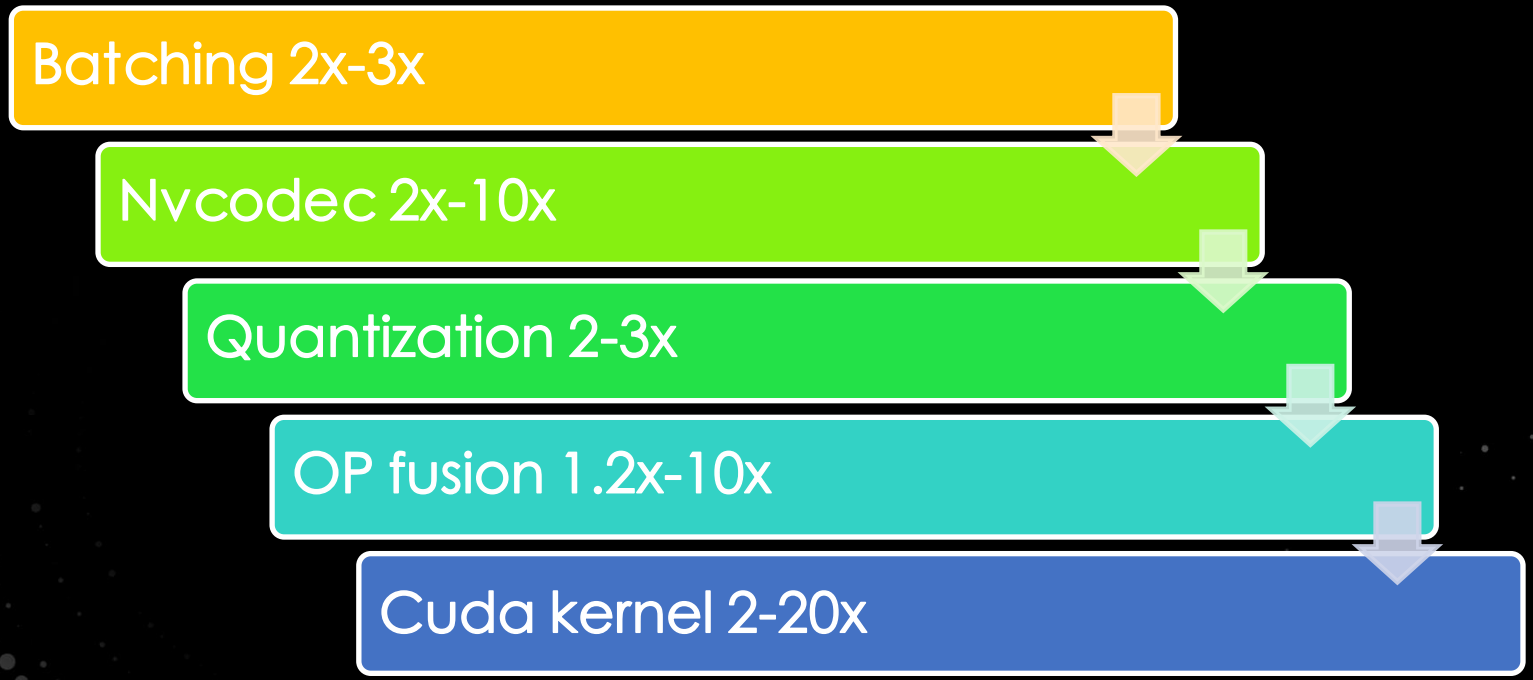
- 1080p RGB 5.93MB
- Pageable Mem <-> GPU 3GB/s
- Pinned Mem <-> GPU 6GB/s

- 不支持batching

已有方案的缺点


算法上线周期长
模型上线难
推理性能差
前后处理多样，工程化工作量大
算法工程师和应用工程师沟通难
GPU利用率低，资源消耗大
优化经验复用难

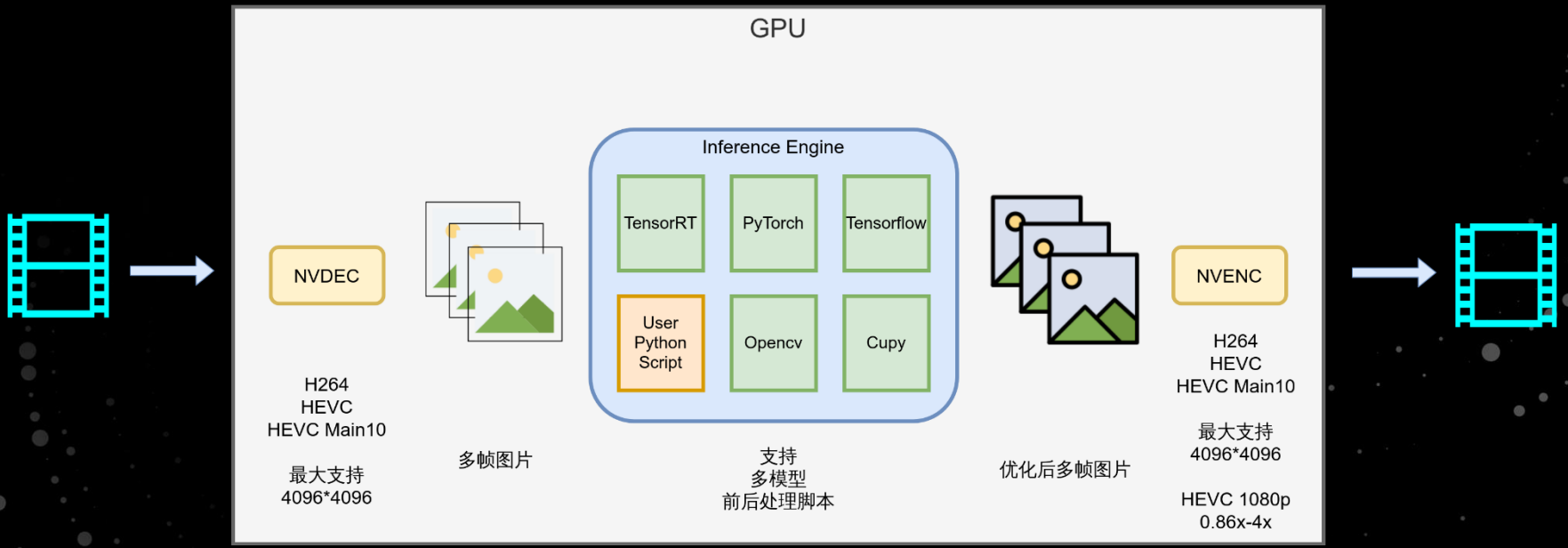
GPU常用优化手段


Batching 2x-3x
Nvcodec 2x-10x
Quantization 2-3x
OP fusion 1.2x-10x
Cuda kernel 2-20x

优化方案

PART 3

All in GPU

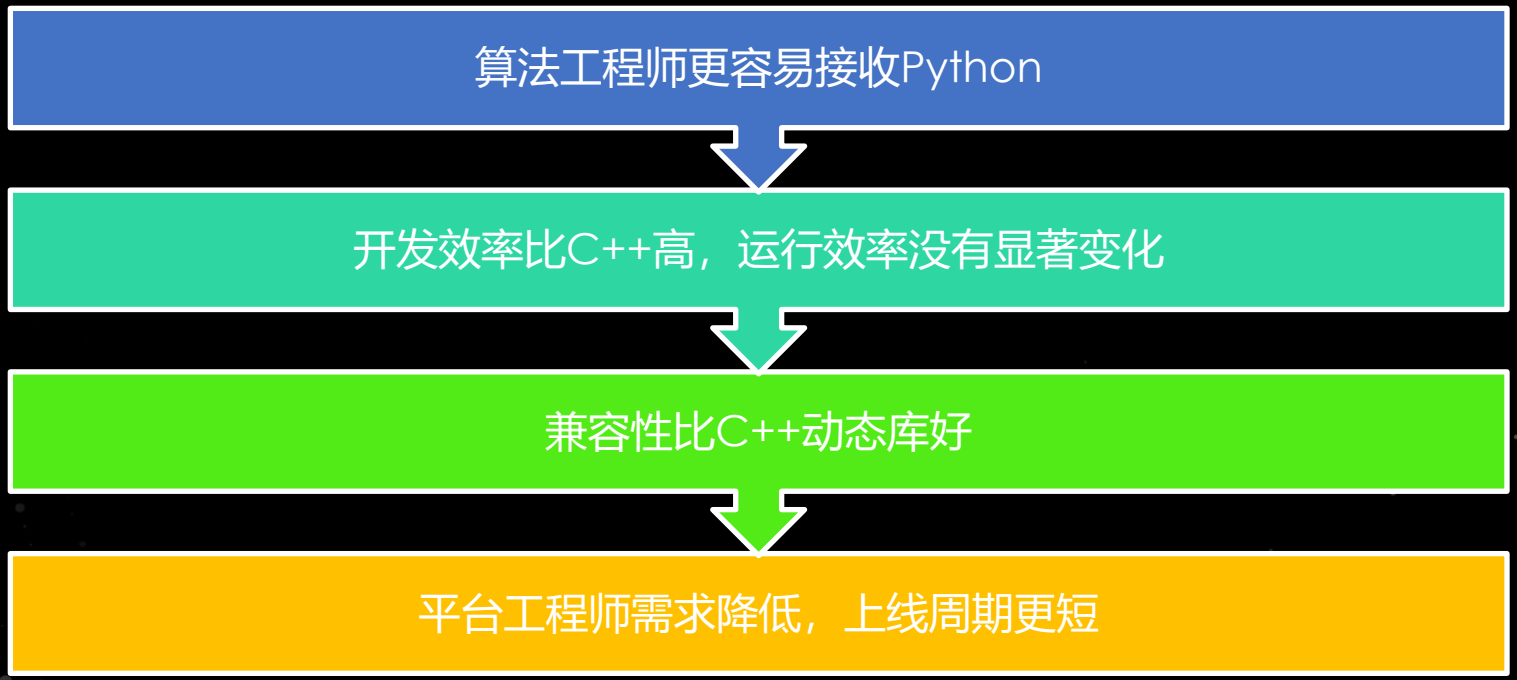

GPU
NVDEC
H264
HEVC
HEVC Main10
最大支持
4096*4096
多帧图片
Inference Engine
TensorRT
PyTorch
Tensorflow
User
Python
Script
Opencv
Cupy
支持
多模型
前后处理脚本
优化后多帧图片
NVENC
H264
HEVC
HEVC Main10
最大支持
4096*4096
HEVC 1080p
0.86x-4x

Python脚本支持

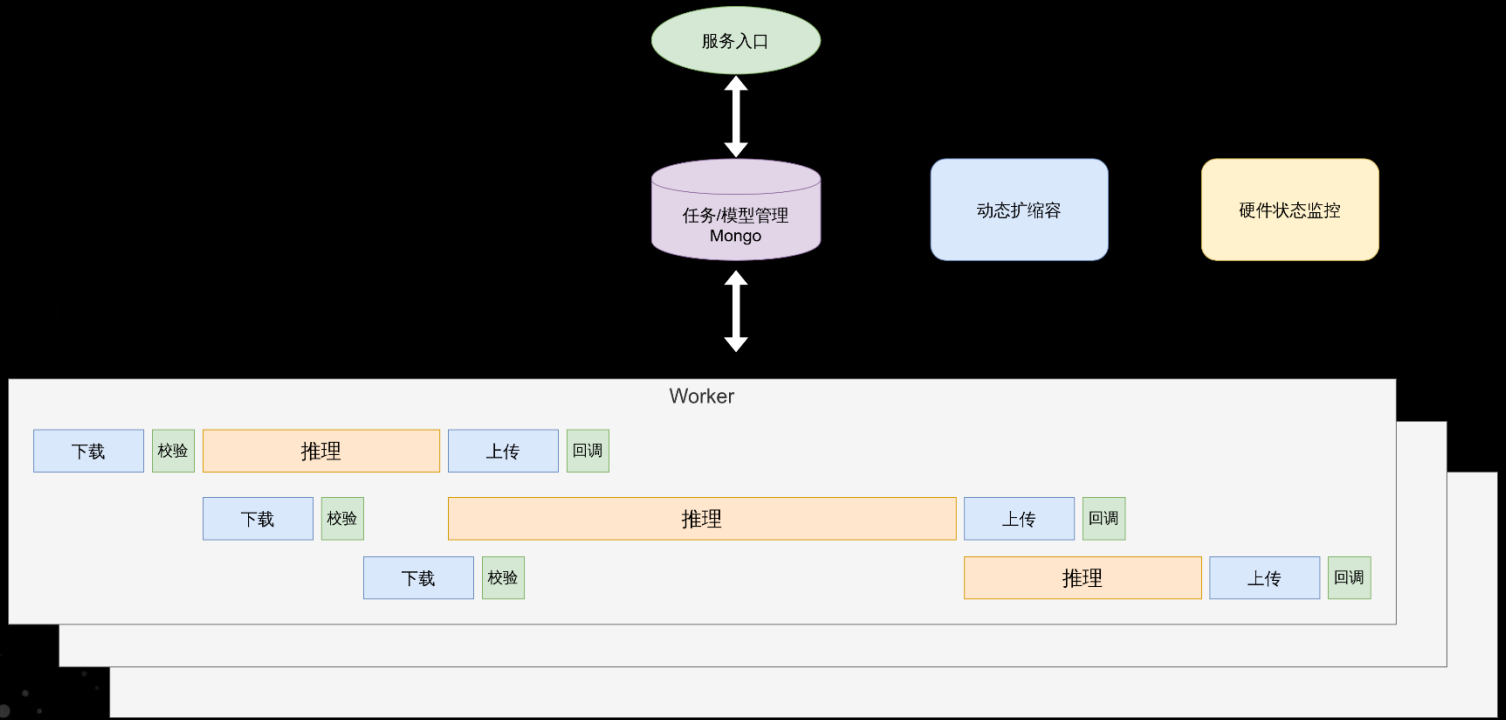

```python
def setup(config):
  # setup model here
  pass

def warmup(_):
  # warmup model
  pass

def configure(stream_config):
  # configure per stream
  pass

def process(data, profiling=False):
  # batching process frames
  pass
```

```python
import numpy as np
import cupy as cp
from PIL import Image

setup('{"model":"/data/haiwei/resnet_v2_fp32_savedmodel_NCHW/"}')

warmup()

configure('{"user-param" : "test"}')

im = Image.open('test.jpg')

data = {
    'height' : im.height,
    'width' : im.width,
    'format' : 'RGB',
    'data': [ cp.asarray(np.array(im)).data.ptr for i in range(16)]
    }

process(data)
```

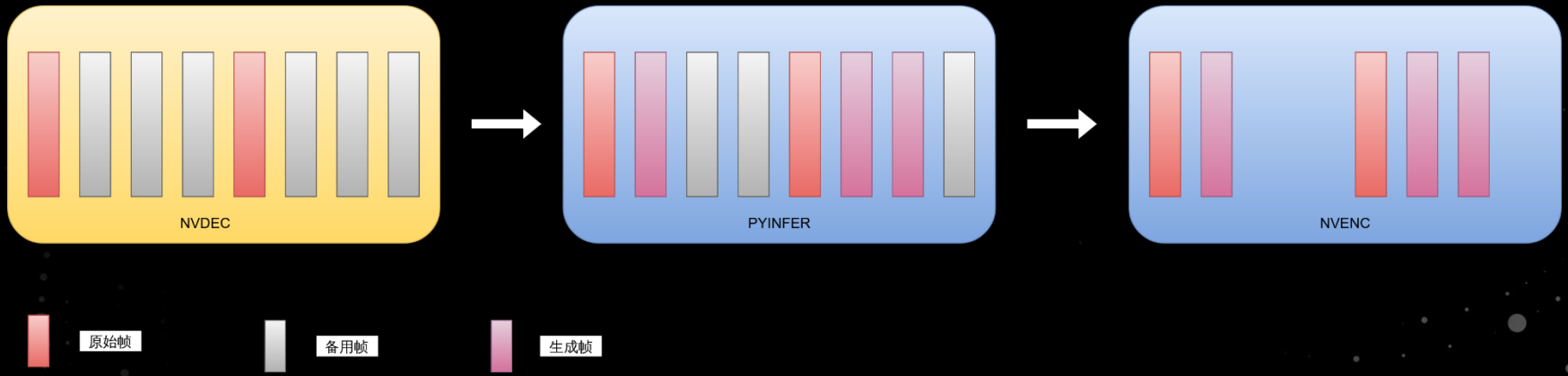
Python脚本支持


算法工程师更容易接收Python
开发效率比C++高，运行效率没有显著变化
兼容性比C++动态库好
平台工程师需求降低，上线周期更短

任务流水线

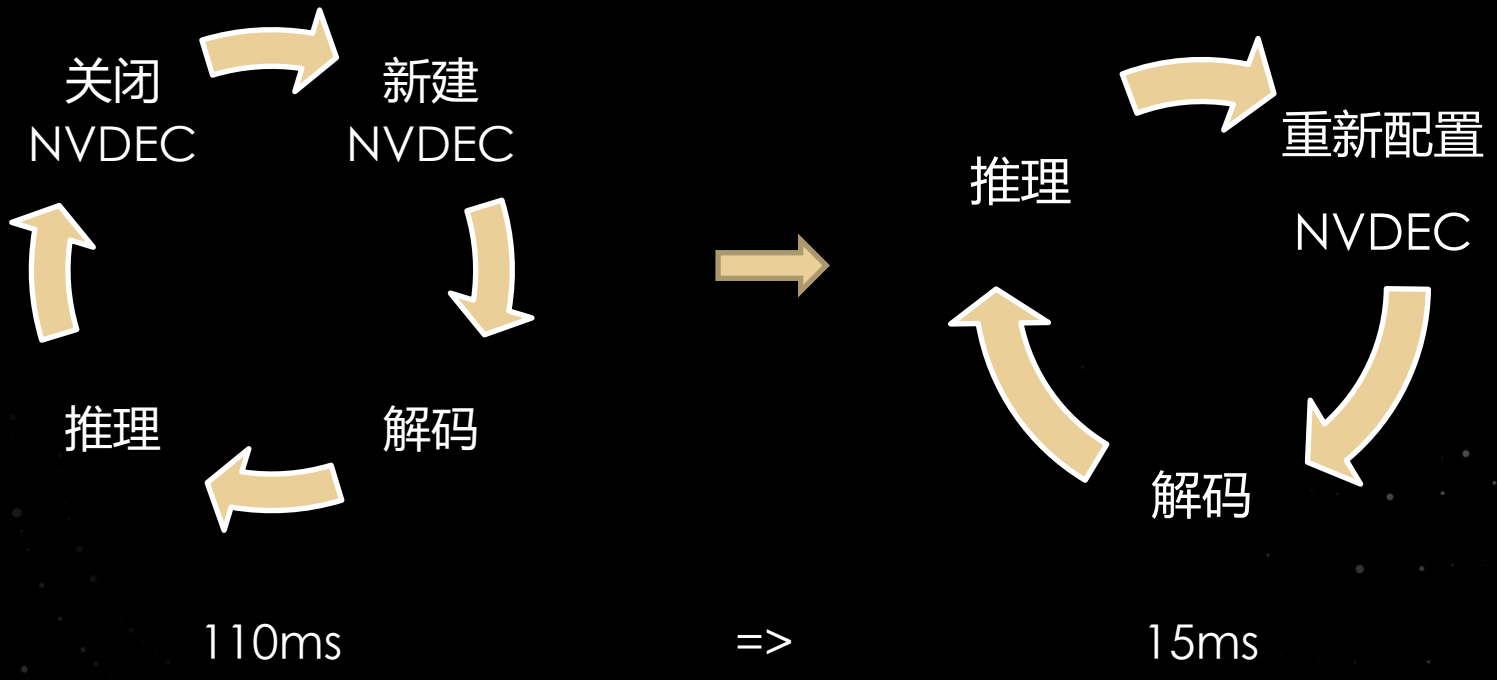

服务入口
任务/模型管理
Mongo
动态扩缩容
硬件状态监控
Worker
下载
校验
推理
上传
回调
下载
校验
推理
上传
回调
下载
校验
推理
上传
回调

高帧率变换服务



- 解码时按比例添加备用帧
- 编码时保留原始帧和生成帧，跳过备用帧

短视频分类服务



模型效果偏差问题

现象

- 少数推理结果差异较大

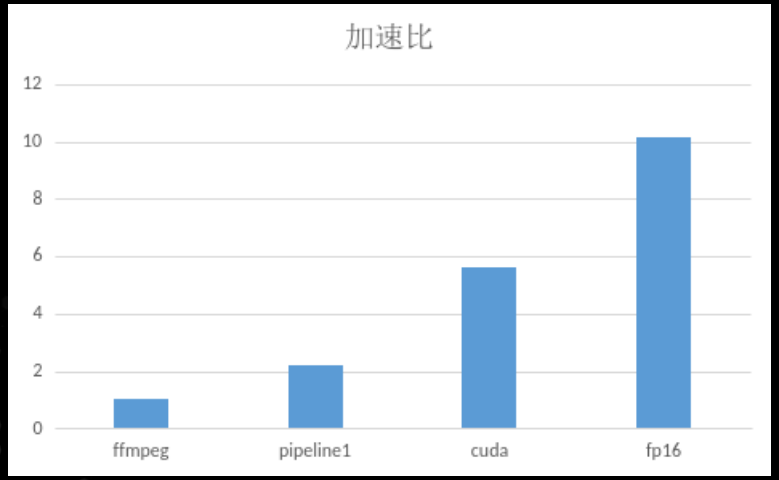
原因

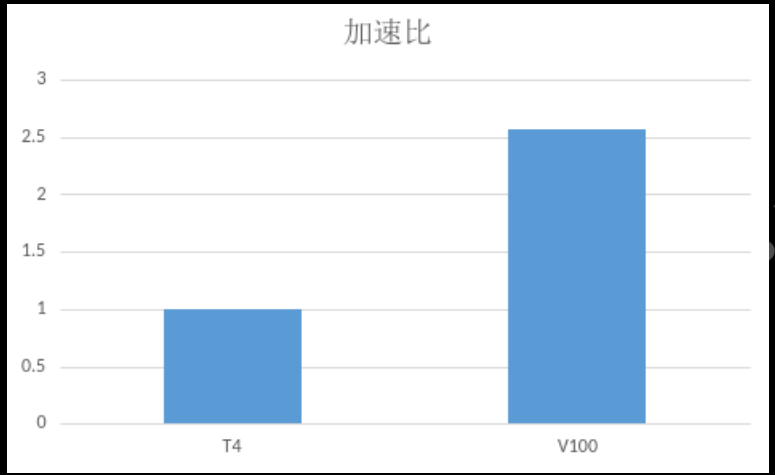
- 图像伸缩算法不一致
- 色彩空间转换取整误差

方案

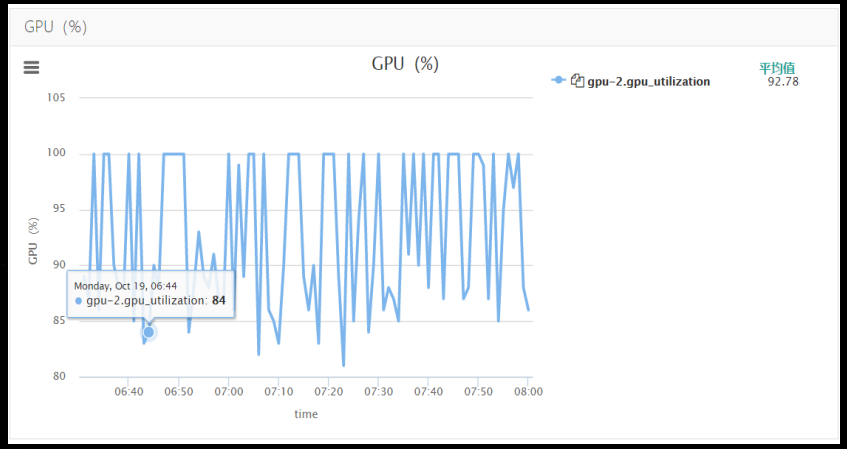
- 使用同一套预处理操作，重新训练模型

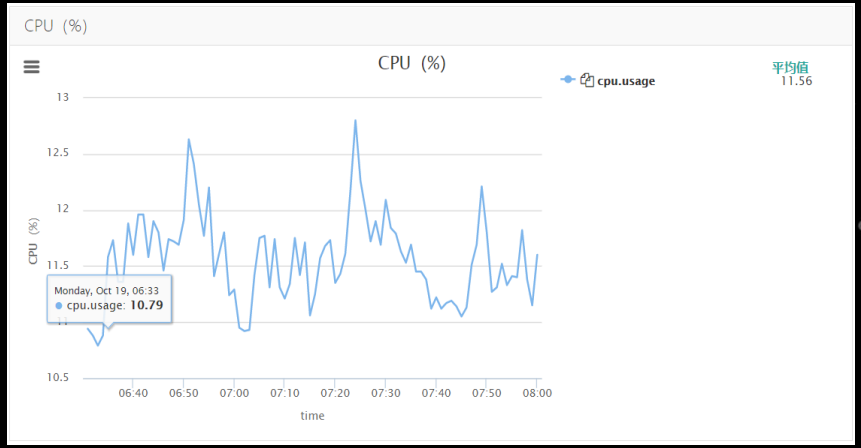
优化效果


加速比
12
10
8
6
4
2
0
ffmpeg
pipeline1
cuda
fp16


加速比
3
2.5
2
1.5
1
0.5
0
T4
V100

资源利用率





总结

PART 4

应用经验


• 使用Python脚本加速模型迭代部署过程
验证过程
扩大验证范围
• 效果稳定后，进行简单优化
• 对耗时操作进行专项操作优化
大范围生产

总结

集成的优化措施	•视频推理过程集中在GPU上完成，减少CPU/GPU数据搬运开销 •使用任务流水线方式，并行CPU/GPU操作 •减少重复优化
集成Python引擎	•满足灵活算法需求 •降低使用门槛
统一的开发测试环境	•算法工程师专注于算法处理逻辑 •平台工程师专注于性能优化

Thank you