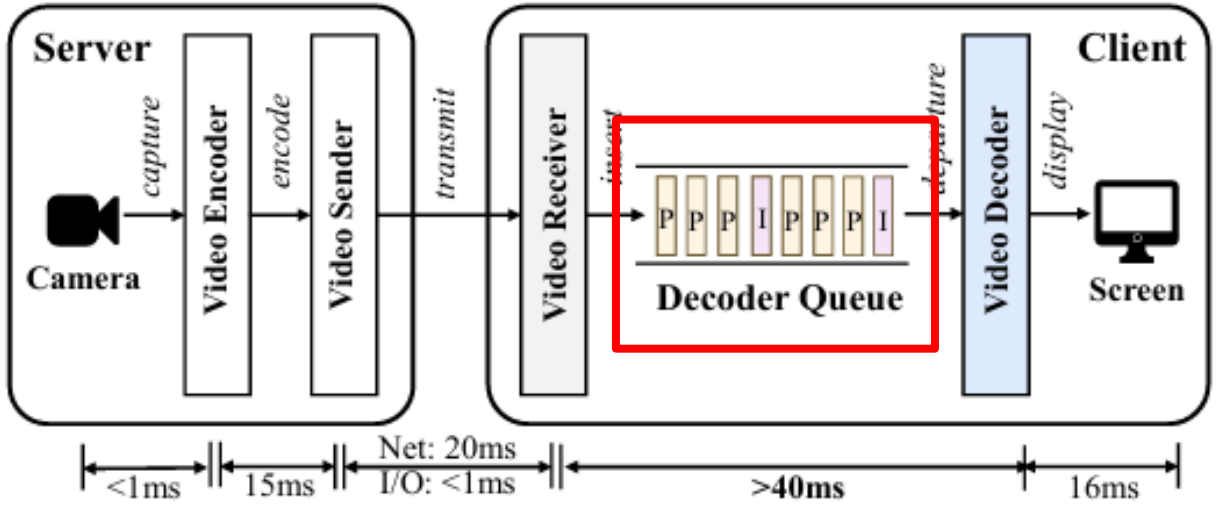
2024/07/15 Research Report

RTCL Lab Meeting
Presenter: Chaehoon Park

About Paper

Problem Definition



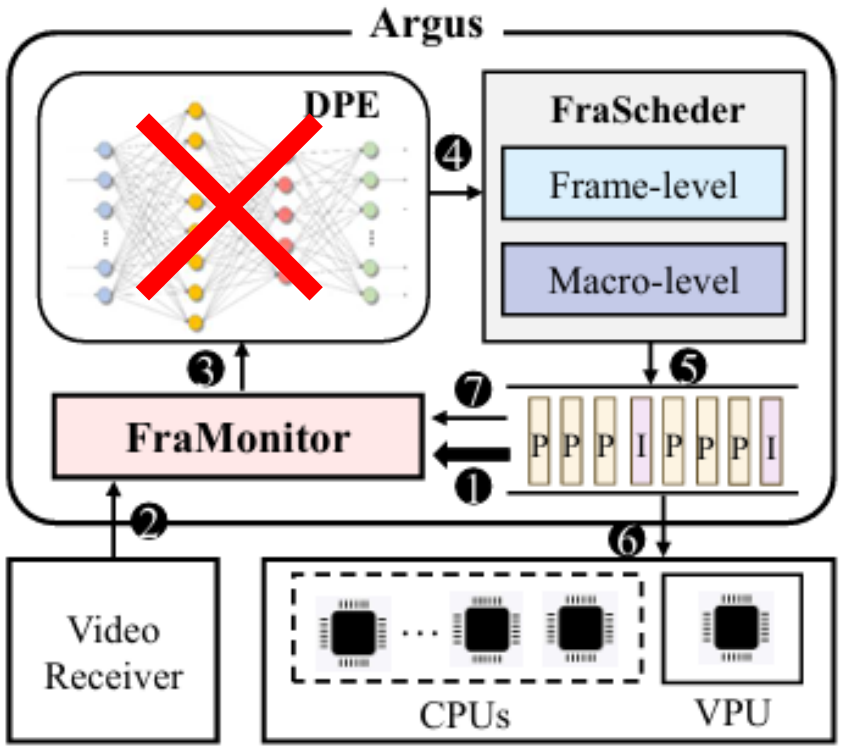
The encoding delay is light since videos are processed on commercial servers.

With the development of communication techniques and I/O protocols (e.g., 5G/6G, WiFi-6), the frames can be delivered within tens of microseconds

→ Bottleneck is located at the decoding phase.

About Paper

Architecture of Argus



"Argus: Real-Time HQ Video Decoding with CPU Coordinating on Consumer Devices"

CPU resources are not fully used when the VPU is busy

Argus: new real-time HQ video decoding solution migrates some frames to the CPU for decoding when the decoder queue is congested.

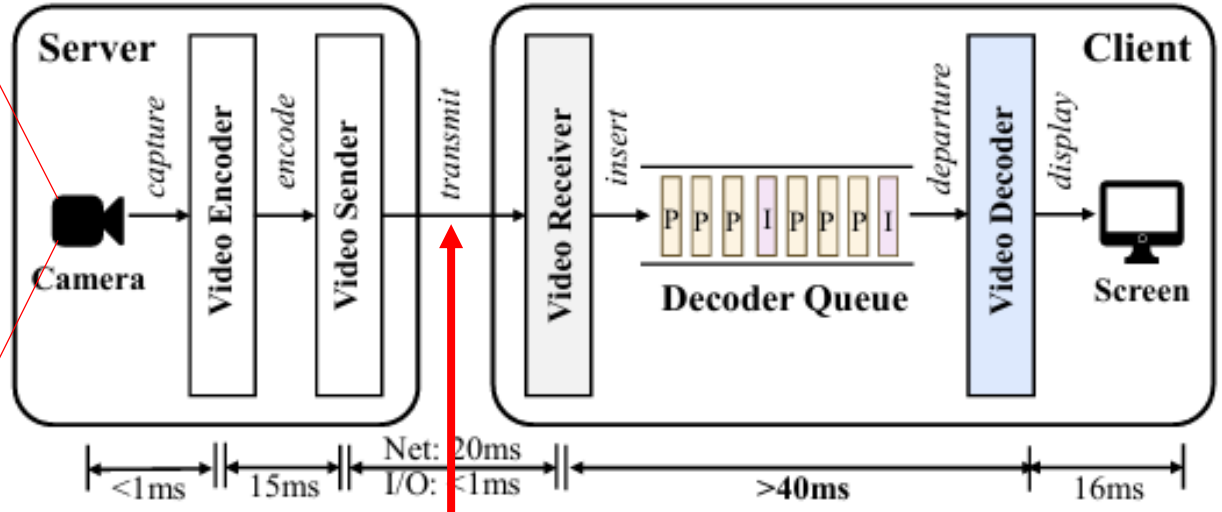
~~DPE (Intelligent decoder pressure estimation model)~~

: Identify the decoder queue busy state as early as possible
→ Take "Queue length > Threshold" as busy condition

FraScheder (Frame-characteristics aware scheduler)

: Offloads several frames to CPUs to alleviate the pressure

Architecture



Virtual World

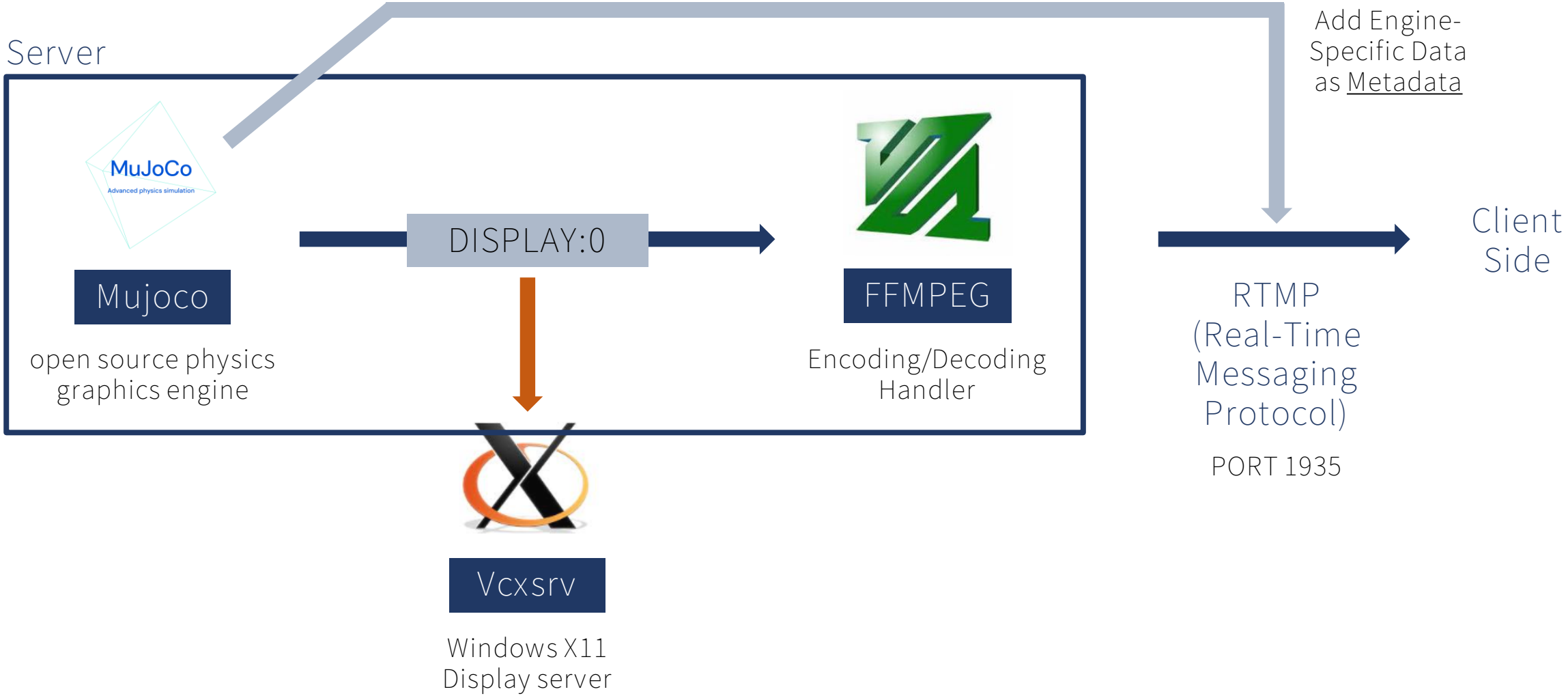
3D Graphics Engine

Offer Additional Information about frames (or macroblocks) that can help

(1) Identifying decoder pressure
(2) Judging which frames (or macroblocks) to schedule in parallel

Architecture

Server Side



Architecture

Additional Information that 3D graphics engine can offer

1) Decoding Delay Estimation

~~Add DPE neural network input features~~
Serve as scheduling timing in FraScheder

(a) Frames with more complicated contents require more macroblock subdivisions in these details, leading to increased decoding computational complexity.

- Number of distinct objects rendered on the screen → Not yet..
- Frame Complexity Value: ex) Σ((Object Texture Complexity) x (Rendered Area Size))
- Object Texture Complexity: pre-calculated number of macroblocks consisting texture

(b) Frames with significant motion or changes require more motion vectors, thus increasing decoding time.

- Camera velocity & angular velocity (+ acceleration, etc.)
- Peripheral Object Transition Value: ex) Σ((Object Velocity) x (Rendered Area Size))

Architecture

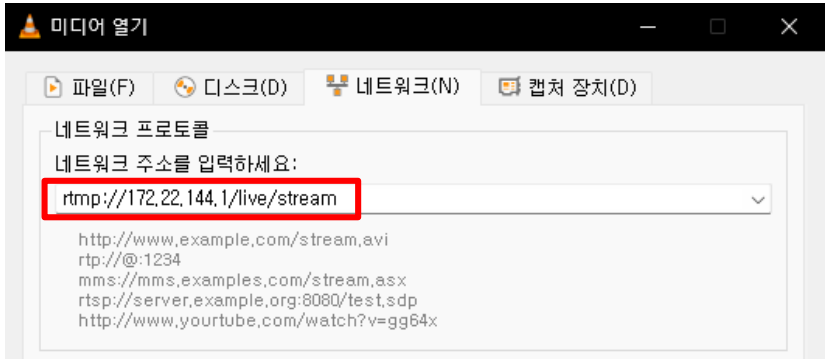
Client Side

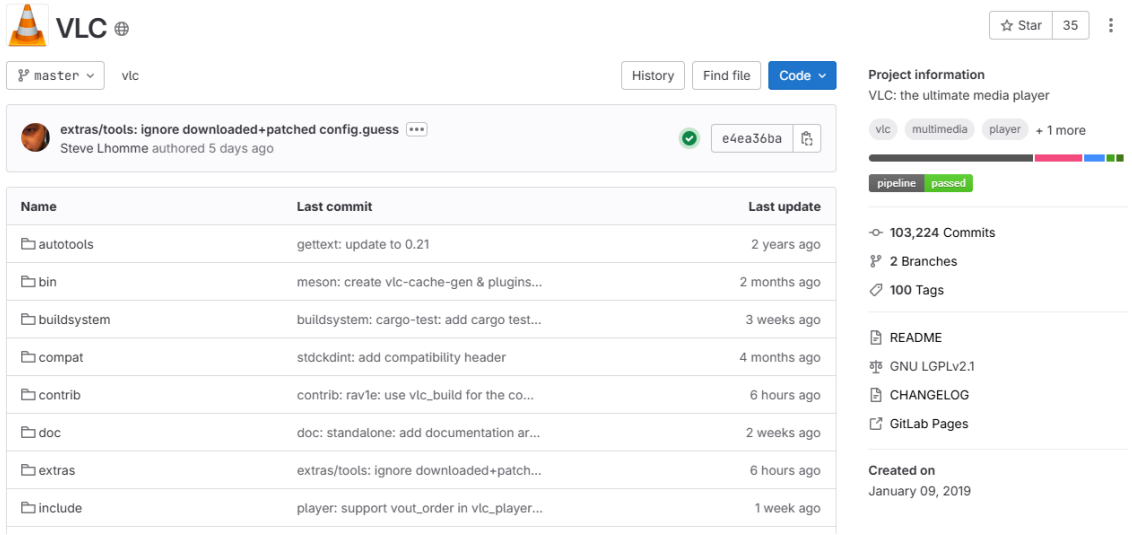
VLC media player

VLC는 DVD, 오디오 CD, VCD 와 다양한 스트리밍 프로토콜뿐 아니라 대부분의 멀티미디어 파일을 재생할 수 있는 무료 오픈 소스 크로스 플랫폼 멀티미디어 재생기이자 프레임워크입니다.

다운로드 VLC

Version 3.0.21 • Windows 64bit • 40 MB
42,984,319 다운로드 현황





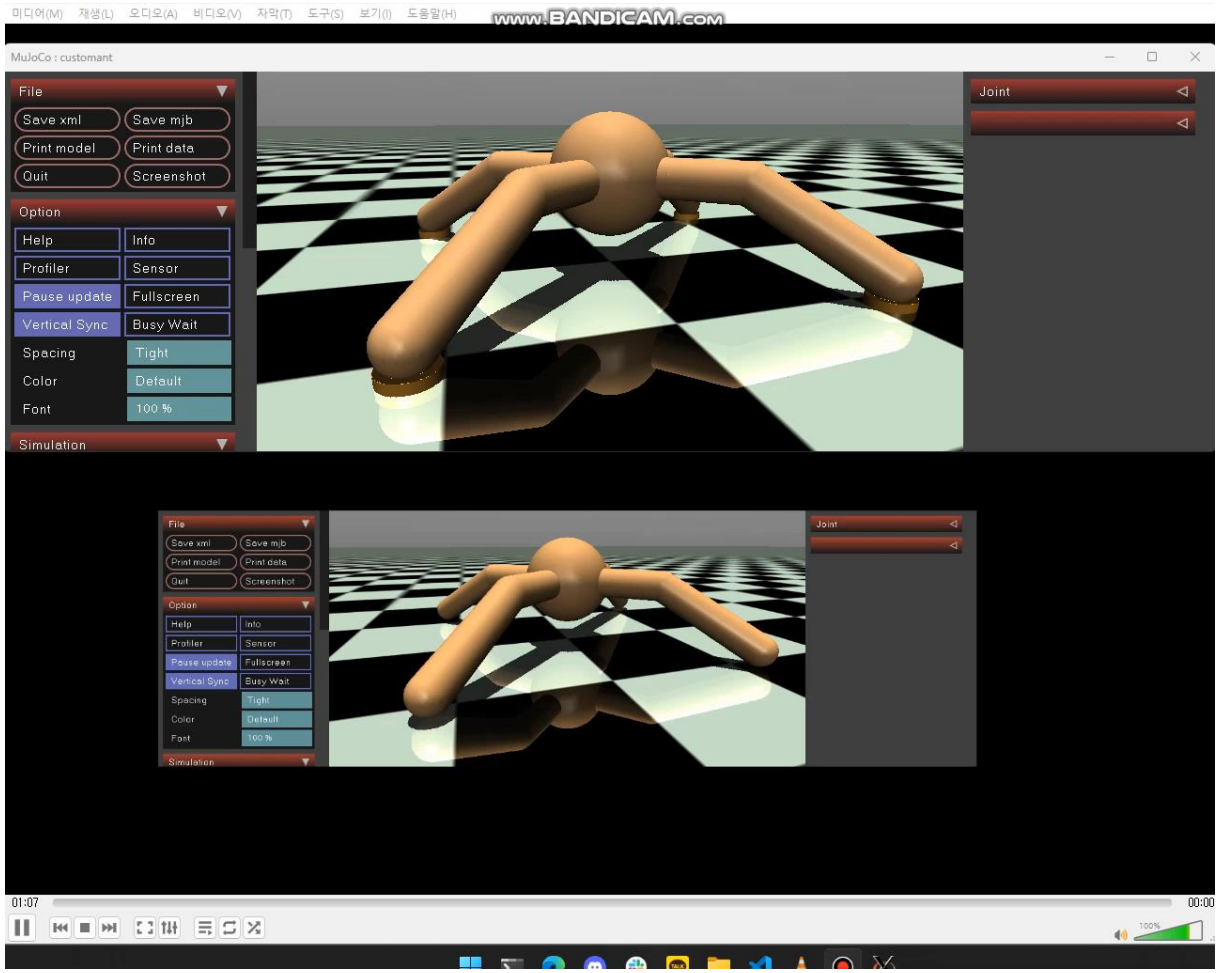
open source...

Demo

Baseline Demo

Server Side

Client Side



5 sec delay...

Demo

Baseline Demo

FFMPEG live streaming

Encoder: Lavf61.5.101 (planning to change into H.264)
Fps: 29
Q: 2.0 (Quality factor. Lower means better bitrate)
Average Bitrate: 1416.2kbits/s
Dup: 0
Drop: 225

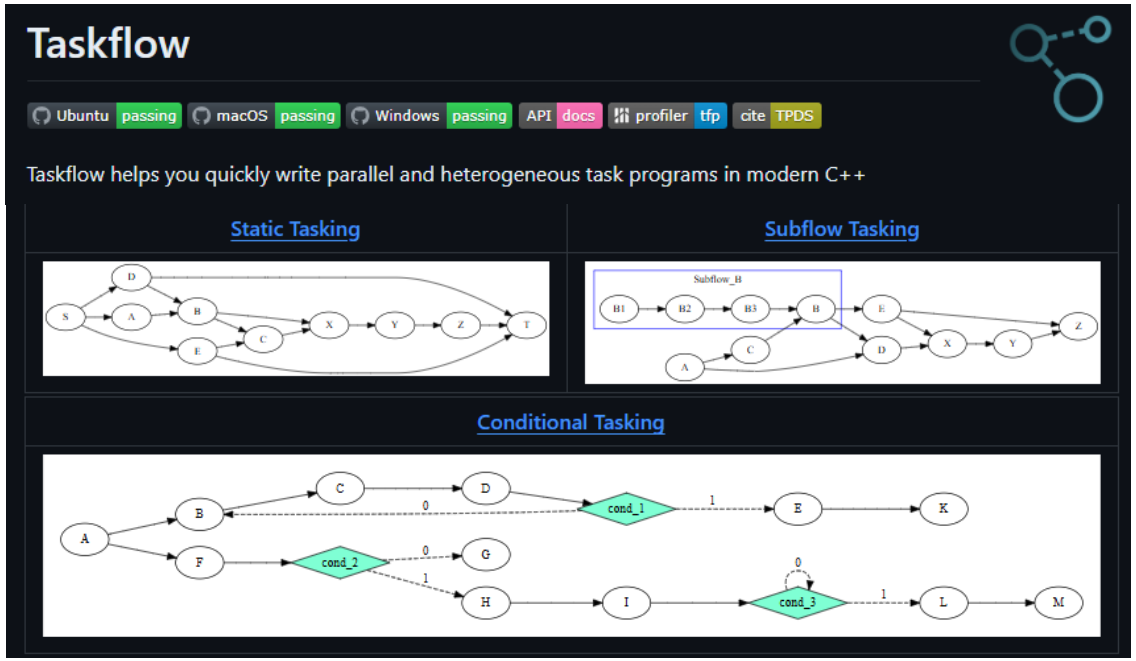
Assume no temporal, spatial locality used while transmission
Assume 1920x1080 Image size ≈ 2MB

→ Transmission Delay for Extreme case:
2MB / ((1416.2kbits/s) / (8bits/1B))
= 11.57sec

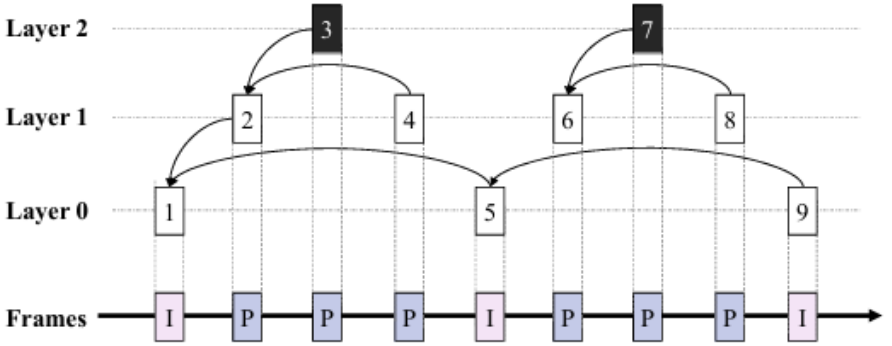
Maybe it reduces a lot when using locality and leads to smaller portion of transmission delay...
(larger portion of decoding delay)

What I am implementing...

- Scheduler



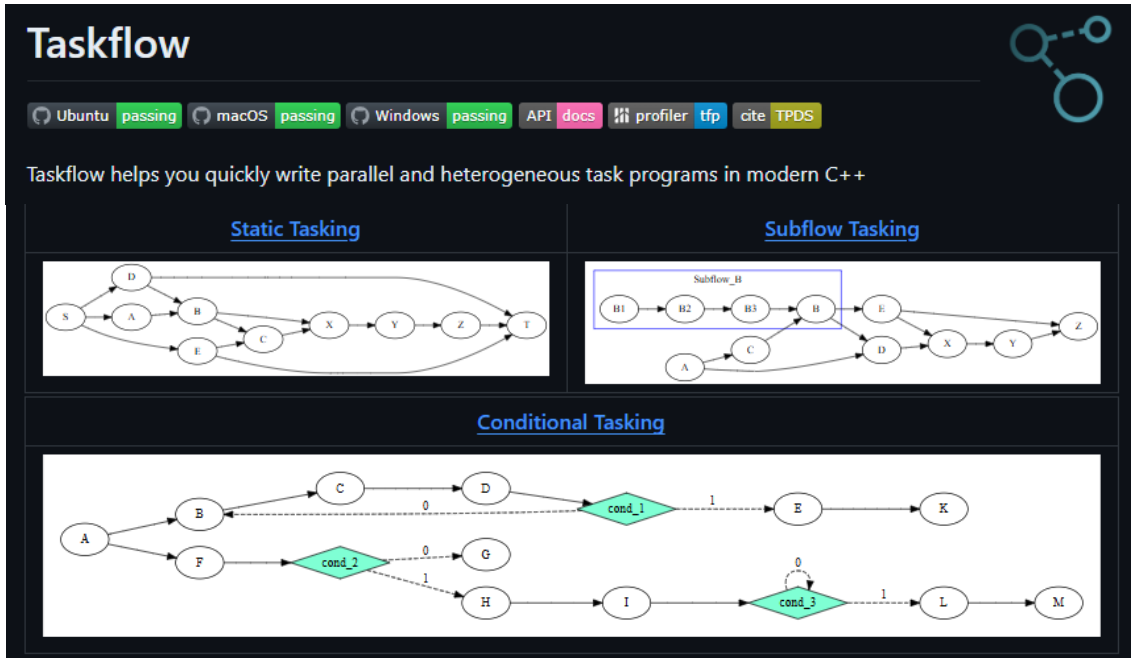
supports heterogeneous tasking by harnessing the power of CPU-GPU collaborative computing



Determine frame dependency based on SVC(Scalable Video Coding) layer

What I am implementing...

- Scheduler



supports heterogeneous tasking by harnessing the power of CPU-GPU collaborative computing

```cpp
#include <taskflow/taskflow.hpp>  // Taskflow is header-only

int main(){

  tf::Executor executor;
  tf::Taskflow taskflow;

  auto [A, B, C, D] = taskflow.emplace(  // create four tasks
    [] () { std::cout << "TaskA\n"; },
    [] () { std::cout << "TaskB\n"; },
    [] () { std::cout << "TaskC\n"; },
    [] () { std::cout << "TaskD\n"; }
  );

  A.precede(B, C);  // A runs before B and C
  D.succeed(B, C);  // D runs after  B and C

  executor.run(taskflow).wait();

  return 0;
}
```

End of Presentation