Slice buffer design

Table of Contents

1. Overview for new design 2

1.1 Origin design 2

1.1.1. Single thread 2

1.1.2. Multi thread 3

1.2 New design in review 4

1.2.1 Single thread 4

# Overview for new design

## Origin design

SSlice\* pSliceInLayer // slice buffer for all slices in layer

### 1.1.1. Single thread

pSliceInLayer

…

…

…

Slc3

Slc2

Slc1

Slc0

**reallocate** when current slice index larger than max slice num

…

pSliceInLayer

Slc2

Slc1

Slc0

Slc0

Slc1

Slc2

…

…

…

…

### 1.1.2. Multi thread

pSliceInLayer

Slc0

Slc1

Slc2

Slc3

…

…

Thrd0

Thrd1

**reallocate** when current slice index larger than max slice num ,

--step 0: thread[0] detect that current slice index larger than max slice num;

--step 1: thread[0] need to wait thread[1] completed current slice encoding task

--step 2: thread[1] stop slice encoding and thread[0] reallocate slice buffer,

--step 3: thread[0]/thread[1] start to encode new slice

pSliceInLayer

Slc0

Slc1

Slc2

...

**Step 0/1**

Thrd1

Thrd0

Thrd0

pSliceInLayer

Slc0

Slc1

Slc2

…

**Step 2**

Slc0

Slc1

Slc2

...

…

…

…

**Step 3**

Thrd2

Thrd0

## 

## 1.2 New design in review

SSlice\* pSliceInLayer; //will be removed and replaced by pSliceInThread[]

SSlice\*\* ppSliceInLayer; // point to actual slice buffer

//based on slice index

SSlice\* pSliceInThread[MaxThreadNum]; // actual slice buffer

### 1.2.1 Single thread

Slc0

Slc1

Slc2

…

ppSliceInLayer

pSliceInThread[0]

Slc0

Slc1

Slc2

…

**reallocate** when current slice index larger than max slice num

Slc0

Slc1

Slc2

…

ppSliceInLayer

pSliceInThread[0]

Slc0

Slc1

Slc2

…

Slc0

Slc1

Slc2

Slc3

…

…

**reallocate**

…

Slc0

Slc1

Slc2

Slc3

…

### 1.2.2. Multi-thread

Slc0

Slc1

Slc2

Slc3

Slc4

…

ppSliceInLayer

pSliceInThread[0]

Slc4

Slc2

Slc0

…

Slc1

Slc3

Slc5

…

pSliceInThread[1]

**for reallocate**, each thread will do it independently, and will update ppSliceInLayer by ***main thread*** when all slices in layer are encoded.

ppSliceInLayer

…

Slc4

Slc3

Slc2

Slc1

Slc0

**Main thread**

Slc0

Slc1

Slc2

Slc3

Slc4

…

…

Slc4

Slc2

Slc0

…

…

Slc0

Slc2

Slc4

…

…

Slc1

Slc3

Slc5

…

…

**Thread 1**

pSliceInThread[0]

**Reallocate**

**/update**

**Reallocate**

**Thread 0**

pSliceInThread[1]

# 2. Slice Buffer and thread

## 2.1 Before encoding one layer

the status of slice buffer and thread :

example:

thread: 3 threads

slices: 9 slices in layer

bThreadBufferUsage[iThrdIdx] = false

**Slice buffer**

pSliceInThread[1]

pSliceInThread[2]

pSliceInThread[0]

Thrd\_0

**Idle**

**thread**

Thrd\_2

Thrd\_1

**Get\_UnUsed\_Buffer()**

slice in layer

Get\_Idle\_Thrd()

Slc8

Slc1

Slc2

Slc0

….

**Slice tasks**

Get\_waiting\_task()

**Task manager**

Encode\_One\_Slice()

## 2.2. Normal case for thread index and slice buffer index

**Task manager**

Thrd\_2

Slice\_2

pSliceBuffer[2]

Thrd\_1

Slice\_1

pSliceBuffer[1]

Thrd\_0

Slice\_0

pSliceBuffer[0]

Thrd\_2

Slice\_8

pSliceBuffer[2]

…

…

…

Final map for thread index and slice buffer index, slice index

ppSliceInLayer

Slc0

Slc1

Slc2

Slc3

…

Slc8

pSliceBuffer[0]

Slc6

Slc3

Slc0

Slc1

Slc4

Slc7

pSliceBuffer[1]

Thrd\_0

Thrd\_1

Thrd\_2

pSliceBuffer[1]

Slc8

Slc5

Slc2