

數字媒體軟件與系統開發

干皓丞，2101212850, 信息工程學院

2022 年 3 月 13 日

1 作業目標與章節摘要

下載 GPAC，理解並描述 random access 過程。

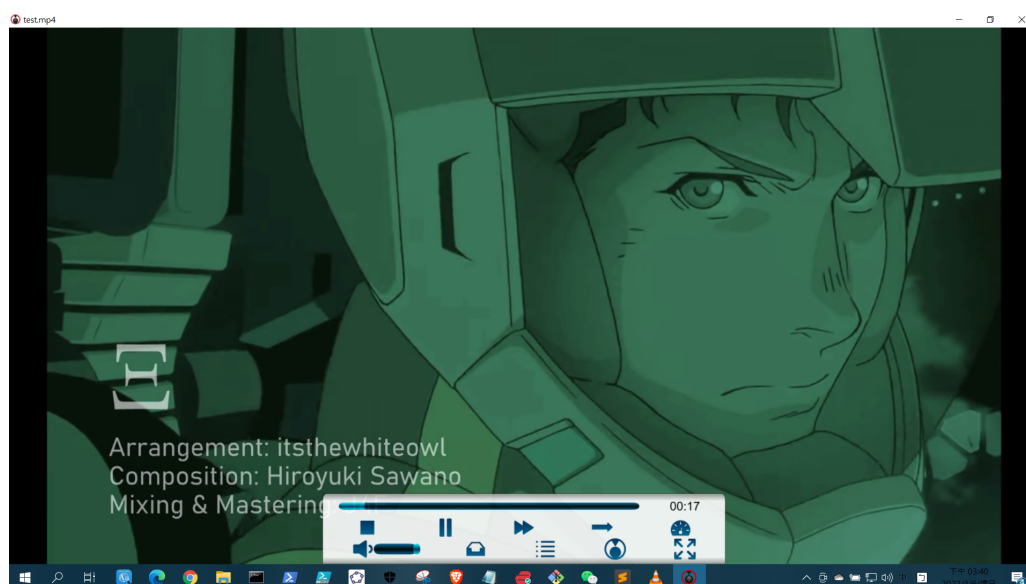


Fig. 1. 使用安裝好的 GPAC 的 Osmo4 播放

2 文章與作業狀況

作業可以從 GitHub 下的 [kancheng/kan-cs-report-in-2022](#) 專案找到，作業程式碼與文件目錄為 [kan-cs-report-in-2022/DMSASD/gpac-random-access](#)。實際執行的環境與實驗設備為 Google 的 Colab、MacBook Pro (Retina, 15-inch, Mid 2014)、Acer Aspire R7 與 HP Victus (Nvidia GeForce RTX 3060)。

3 作業內容概述

此作業分二大部分，第一部分說明 GPAC 使用與理解，第二部分則描述描述 Random Access 過程。

1. GPAC 使用與理解
2. 描述 Random Access 過程

4 GPAC 使用與理解

GPAC 是一個 LGPL v2.1 且在大多數情況下也可以在商業許可下使用的開源多媒體框架，其專案提供了使用者在處理、檢查、打包、流式傳輸、播放和與媒體內容交互的工具。此類內容可以為音頻、影像、字幕、元數

據、可縮放圖形、加密媒體、2D/3D 圖形和 ECMAScript 等任意組合。GPAC 以其廣泛的 MP4/ISOBMFF 功能而聞名，深受影像愛好者、學術研究人員、標準化機構和專業廣播公司的歡迎。

前往 GPAC (<https://gpac.wp.imt.fr/>) 下載，安裝後即可以使用。開啟終端機查看版本。

1. GPAC 的官方開發文件: <https://doxygen.gpac.io/>
2. GPAC 的 Wiki: <https://github.com/gpac/gpac/wiki/Howtos>

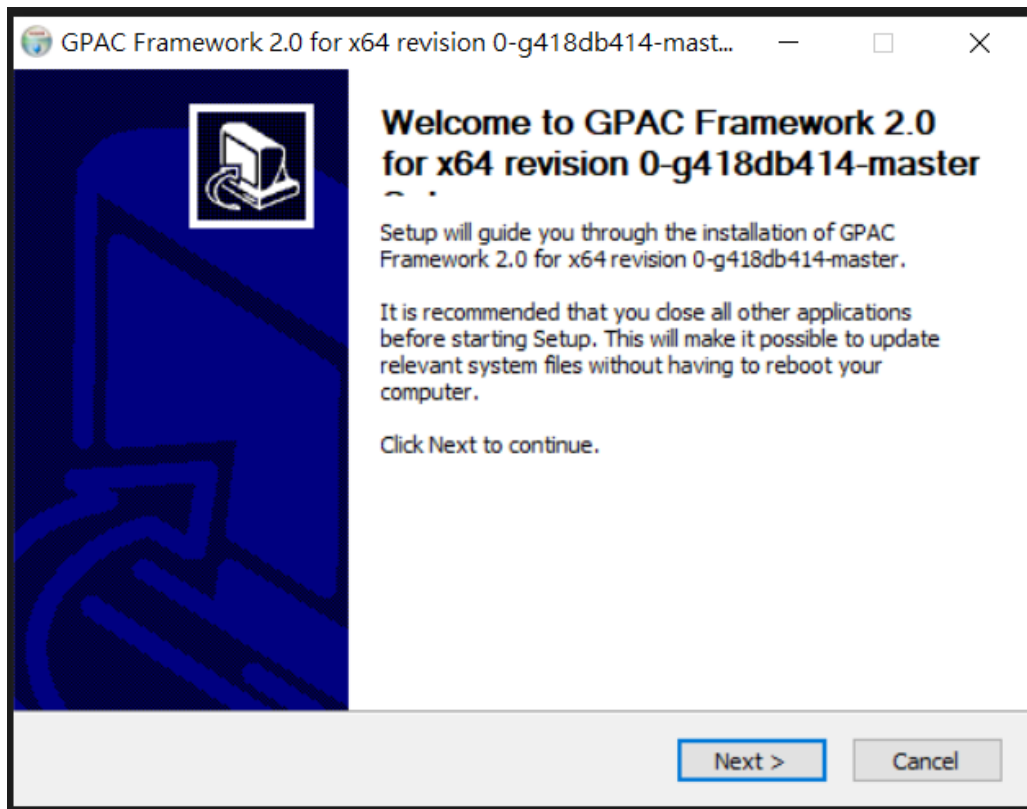


Fig. 2. 在此使用 GPAC 的 Windows 版本安裝

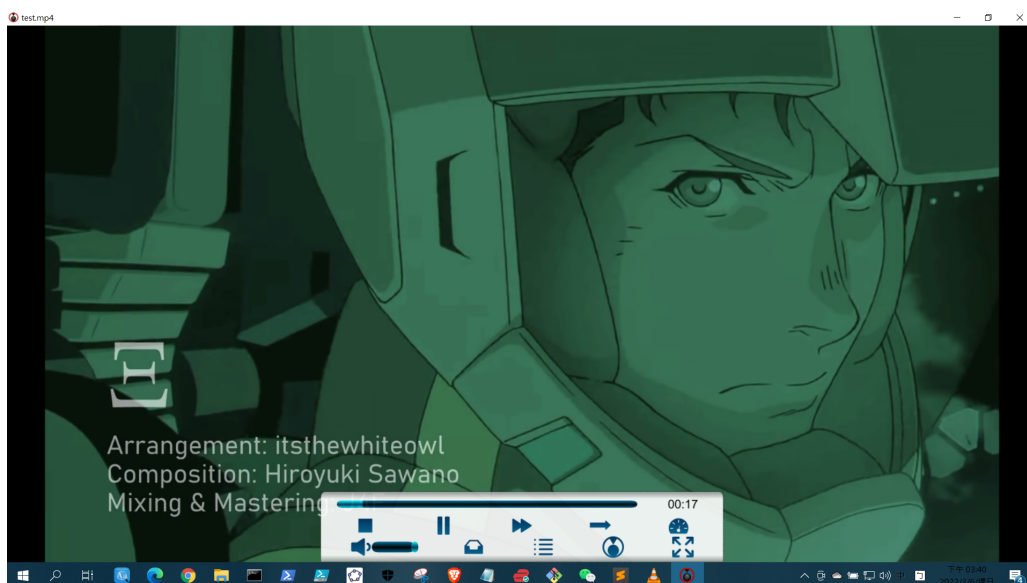
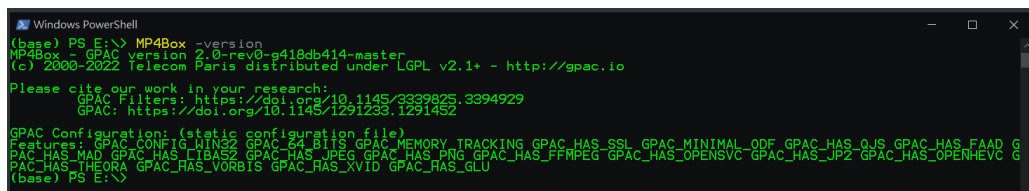


Fig. 3. 完成

1. 查看 mp4box 版本

```
1 mp4box -version
```



```
Windows PowerShell
(base) PS E:\> MP4Box -version
MP4Box - GPAC version 2.0-rev0-g418db414-master
(c) 2000-2022 Telecom Paris distributed under LGPL v2.1+ - http://gpac.io

Please cite our work in your research:
GPAC Filters: https://doi.org/10.1145/3339825.3394929
GPAC: https://doi.org/10.1145/1291233.1291452

GPAC Configuration: (static configuration file)
Features: GPAC_CONFIG_WIN32 GPAC_24_BITS GPAC_MEMORY_TRACKING GPAC_HAS_SSL GPAC_MINIMAL_ODF GPAC_HAS_QJS GPAC_HAS_FAAD G
PAC_HAS_MAD GPAC_HAS_LIBAS2 GPAC_HAS_JPEG GPAC_HAS_PNG GPAC_HAS_FFMPEG GPAC_HAS_OPENSVC GPAC_HAS_JP2 GPAC_HAS_OPENHEVC G
PAC_HAS_THEORA GPAC_HAS_VORBIS GPAC_HAS_XVID GPAC_HAS_GLU
(base) PS E:\>
```

Fig. 4. 查看 GPAC 的 mp4box 版本

2. 查看 mp4box 操作指令

```
1 # 1
2 mp4box -h
3 # 查看 mp4box 中的所有幫助信息
4
5 # 2
6 mp4box -h general
7 # 查看 mp4box 中的通用幫助信息
8
9 # 3
10 mp4box -info test.mp4
11 # 查看 test.mp4 文件是否有問題
12
13 # 4
14 mp4box -add test.mp4 test-new.mp4
15 # 修復 test.mp4 文件格式不標準的問題，並把新文件保存在 test-new.mp4 中
16
17 # 5
18 mp4box -inter 10000 test-new.mp4
19 # 解決開始播放 test-new.mp4 卡一下的問題，為 HTTP 下載快速播放有效，10000ms
20
21 # 6
22 mp4box -add file.avi new_file.mp4
23 # 把 avi 文件轉換為 mp4 文件
24
25 # 7
26 mp4box -hint file.mp4
27 # 為 RTP 準備，此指令將為文件創建 RTP 提示跟蹤信息。
28 # 這使得經典的流媒體服務器像 darwinstreamingserver 或 QuickTime 的流媒體服務器通
    過 RTSP/RTP 傳輸文件
29
30 # 8
31 mp4box -cat test1.mp4 -cat test2.mp4 -new test.mp4
32 # 把 test1.mp4 和 test2.mp4 合併到一個新的文件 test.mp4 中，要求編碼參數一致
33
34 # 9
35 mp4box -force-cat test1.mp4 -force-cat test2.mp4 -new test.mp4
```

```

36 # 把 test1.mp4 和 test2.mp4 強制合併到一個新的文件 test.mp4 中，有可能不能播放
37
38 # 10
39 mp4box -add video1.264 -cat video2.264 -cat video3.264 -add audio1.aac -cat
    audio2.aac -cat audio3.aac -new muxed.mp4 -fps 24
40 # 合併多段音視頻並保持同步
41
42 # 11
43 mp4box -split *time_sec* test.mp4
44 # 切取 test.mp4 中的前面 time_sec 秒的視頻文件
45
46 # 12
47 mp4box -split-size *size* test.mp4
48 # 切取前面大小為 size KB的視頻文件
49
50 # 13
51 mp4box -split-chunk *S:E* test.mp4
52 # 切取起始為 S 秒，結束為 E 秒的視頻文件
53
54 # 14
55 mp4box -add 1.mp4#video -add 2.mp4#audio -new test.mp4
56 # test.mp4 由 1.mp4 中的視頻與 2.mp4 中的音頻合併生成

```

5 描述 Random Access 過程

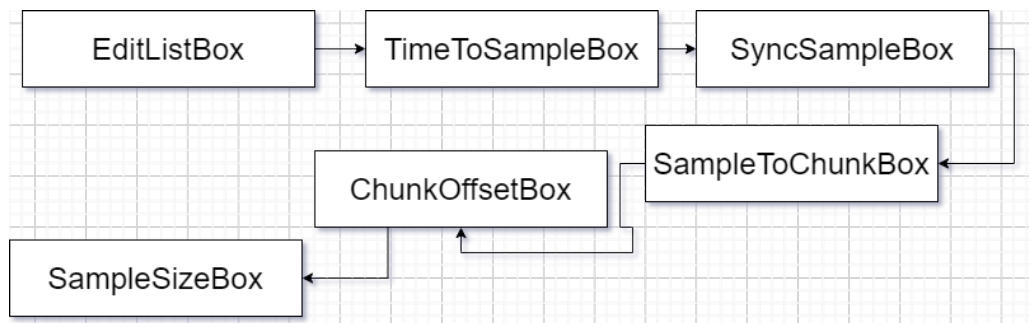


Fig. 5. 描述 Random Access 過程圖

If you want to seek a given track to a time T ,

假如想要最一個文件進行隨機訪問，而該訪問進行為 T 時刻 EX: 第三分五十秒。

1. If the track contains an edit list, determine which edit contains the time T by iterating over the edits. $T_{\text{movie}} = T_{\text{start}} + T'$

如果軌道包含編輯列表，則通過迭代編輯確定哪個編輯包含時間 T 。 $T_{\text{movie}} = T_{\text{start}} + T'$

先找當中有沒有 edit list，當中 edit list 實際上存著一段段的樣本信息，這裡面會有一個關鍵字段的有開始時刻 (T_{start}) 的連續樣本。這個過程就是將 T 換算成 $T_{\text{start}} + T'$ ，而這個 T' 就是換算出來的新的時間。

2. Convert to media time scale $T_{\text{media}} = T_{\text{start}}' + T''$

然後再將原先的過程轉換為媒體時間尺度 $T_{\text{media}} = T_{\text{start}}' + T''$

3. Use time-to-sample box to find the first sample prior to the given time

使用 time-to-sample box 找到給定時間之前的第一個樣本，也就是對應的樣本編號。

4. Consult the sync sample table to seek to which sample is closest to, but prior to, the sample found above 因為找到的樣本不一定是可以解的，為了保險會先查閱同步樣本表以尋找最接近但先於上面所找到的樣本的樣本編號。

5. Use the sample-to-chunk table to determine in which chunk this sample is located.

使用 sample-to-chunk 來確定該樣本位於哪個 chunk 中。

6. use the chunk offset box to figure out where that chunk begins

找到後根據使用 the chunk offset box 來確定該 chunk 開始的物理存儲位置。

7. Starting from this offset, you can use the information contained in the sample-to-chunk box and the sample size box to figure out where within this chunk the sample in question is located.

從這個偏移量開始，您可以使用包含在 sample-to-chunk box 和 the sample size box 中的信息來確定有問題的樣本在這個 chunk 中的位置。

尋找專案中有關 Random Access 的部分，利用 find 和 grep 指令，尋找 GPAC 與 Random Access 過程有關的檔案。

```
1 find . -name "*.c" | xargs grep "random access" *.*
```

```
USER@Aspire-R7 MINGW64 /d/USERDATA/Downloads/gpac-master/gpac-master
$ find . -name "*.c" | xargs grep "random access" *.*
grep: ./: Is a directory
grep: ./github: Is a directory
grep: ./applications/deprecated/old_arch/osmo4_wce/smartphone: No such file or directory
grep: 2003: No such file or directory
grep: (armv4)/release/install/archive.bat: No such file or directory
grep: ./applications/deprecated/old_arch/osmo4_wce/smartphone: No such file or directory
grep: 2003: No such file or directory
grep: (armv4)/release/install/build_installer.bat: No such file or directory
grep: ./applications/deprecated/old_arch/osmo4_wce/smartphone: No such file or directory
grep: 2003: No such file or directory
grep: (armv4)/release/install/gpac.inf: No such file or directory
grep: ./applications/deprecated/old_arch/osmo4_wce/smartphone: No such file or directory
grep: 2003: No such file or directory
grep: (armv4)/release/install/readme.txt: No such file or directory
grep: ./applications/generators/SVG/Tiny-1.2-NG: Is a directory
./applications/mp4box/fileimport.c: /*make random access for storage*/
./applications/mp4box/main.c: "The input file must have enough random access points in order to
./applications/mp4box/main.c: MP4BOX_ARG("rap", "ensure that segments begin with random access points,
FUN | ARG_EMPTY),
./applications/mp4box/main.c: MP4BOX_ARG("frag-rap", "ensure that all fragments begin with random access
./applications/mp4box/main.c: "MP4Box can encode BIFS or LAsER streams and insert random access
urn in the scene at the same time. In MPEG-4 terminology, this is called the __scene carousel__."
./applications/mp4box/main.c: MP4BOX_ARG("rap", "signal random access points in RTP packets (MPEG-4 Sys
grep: ./build/xcode/gpac.xcodeproj: Is a directory
grep: ./build/xcode/gpac4ios.xcodeproj: Is a directory
./extra_lib/include/zlib/zlib.h: random access is desired. Using Z_FULL_FLUSH too often can seriously de
./include/gpac/constants.h: /*!M: is-RAP: DIMS unit is a random access point*/
./include/gpac/filters.h: - Packets have default attributes such as timestamps, size, random access
./include/gpac/html5_media.h: Bool needs_rap; /* MSE need random access point flag
./include/gpac/ietf.h: /*random access point flag present*/
./include/gpac/internal/swf_dev.h: 1- all BIFS AUs in sprites are random access
./include/gpac/isomedia.h:/*! enables mfra (movie fragment random access computing) when writing movie fr
./include/gpac/isomedia.h:\param IsRandomAccessPoint set to GF_TRUE if you want to indicate that this is
./include/gpac/isomedia.h:The resulting sample will always be marked as random access
./include/gpac/media_tools.h:\param is_rap set to GF_TRUE to signal a random access point
./include/gpac/mpegts.h: /*! random access indicator*/
./include/gpac/rtp_streamer.h:\param is_rap indicates if the AU is a random access
./include/gpac/rtp_streamer.h:\param is_rap indicates if the AU is a random access
./include/gpac/rtp_streamer.h:\param is_rap indicates if the AU is a random access
./include/gpac/scene_manager.h: /*AU is RAP - random access indication - may be overridden by encoder*/
./include/gpac/scene_manager.h:stream, this AU being a random access for the stream
grep: ./packagers/osx/GPAC_app: Is a directory
```

Fig. 6. 尋找 GPAC 與 Random Access 過程有關的檔案


```
1 ./ applications /mp4box/ fileimport .c
2 ./ applications /mp4box/ main .c
3 ./ extra_lib /include /zlib /zlib .h
4 ./ include /gpac /constants .h
5 ./ include /gpac /filters .h:
6 ./ include /gpac /html5_media .h:
7 ./ include /gpac /ietf .h:
```

```

8 ./include/gpac/internal/swf_dev.h:
9 ./include/gpac/isomedia.h:
10 ./include/gpac/isomedia.h:
11 ./include/gpac/media_tools.h:
12 ./include/gpac/mpegts.h:
13 ./include/gpac/rtp_streamer.h:
14 ./include/gpac/scene_manager.h:
15 ./share/doc/man/gpac-filters.1:mfra (bool, default: false):
16 ./share/doc/man/mp4box.1:
17 ./src/filters/mux_isom.c:
18 ./src/media_tools/html5_mse.c:
19 ./src/media_tools/m2ts_mux.c:
20 ./src/scene_manager/text_to_bifs.c:

```

同時可以從 GPAC 的官方文件看到 Random Access 的部分。



libgpac
Documentation of the core library of GPAC

Main Page Modules Data Structures Files

GF_M2TS_TimelineCodebook
GF_M2TS_Program
GF_M2TS_ES
GF_M2TS_PESHeader
GF_M2TS_SECTION_ES
tag_m2ts_dvb_sub
tag_m2ts_dvb_teletext
tag_m2ts_metadata_descrip
GF_M2TS_Header
GF_M2TS_AdaptationField
discontinuity_indicator
random_access_indicator
priority_indicator
PCR_flag
PCR_base

GF_M2TS_AdaptationField

struct GF_M2TS_AdaptationField

MPEG-2 TS packet adaptation field

Data Fields		
u32	discontinuity_indicator	discontinuity indicator (for timeline splicing)
u32	random_access_indicator	random access indicator
u32	priority_indicator	priority indicator
u32	PCR_flag	PCR present flag
u64	PCR_base	PCR base value

Fig. 7. 文件 GPAC 與 Random Access

0. MP4 架構分析 - MP4Box.js

<https://github.com/gpac/mp4box.js>

這個 JavaScript 庫可以在瀏覽器 (和 node.js 中) 處理 mp4 文件，並支持解析。靈感來自 GPAC 項目中的 MP4Box 工具。它可以用來：獲取有關 mp4 文件的信息，將 mp4 文件分段，以便與媒體源擴展 API 一起使用，從 mp4 中提取樣本來創建 TextTracks。其提供了一個線上分析 MP4 的网站，可以對 MP4 進一步的解析，並且更直觀的幫助我們了解此次的作業內容。首先可以看見對這個檔案的 overview，也是兩個 Track。

<https://gpac.github.io/mp4box.js/test/reader.html>

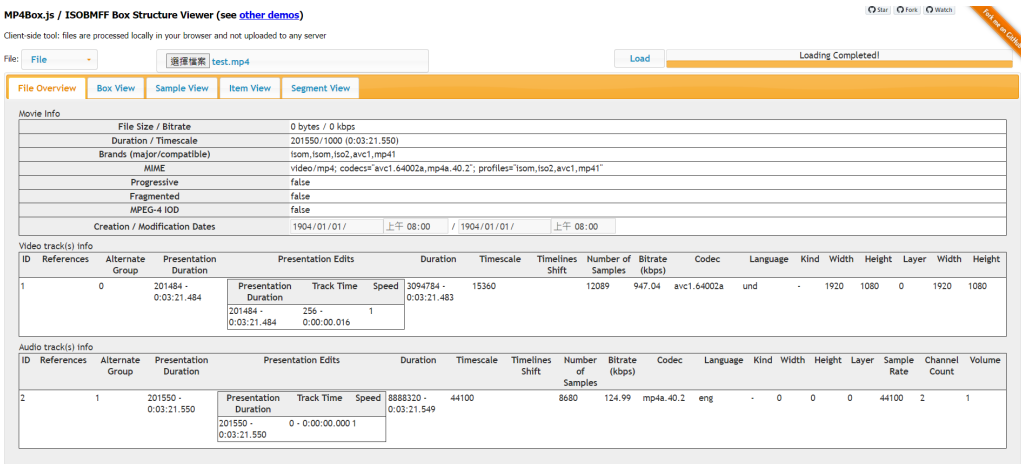


Fig. 8. MP4 架構分析

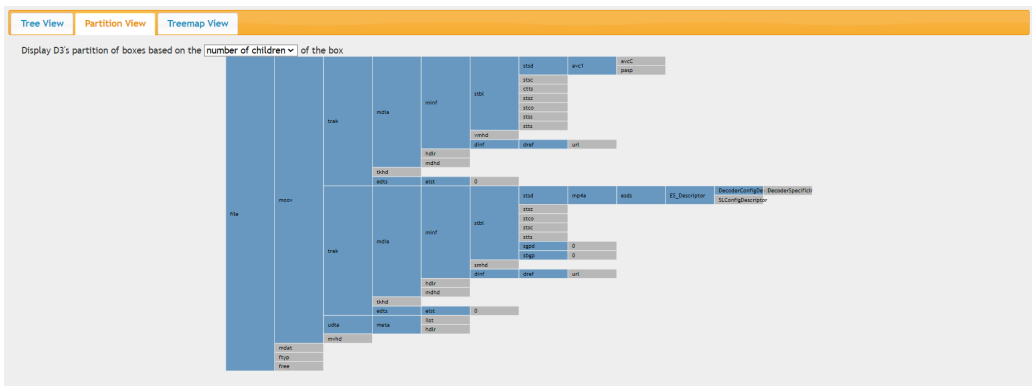


Fig. 9. MP4 架構分析



Fig. 10. MP4 架構分析

1. 技術文件

本節描述如何尋找。查找主要通過使用 sample table box 中的子 box 來完成。如果 edit list 存在，也必須查閱它。如果你想尋找一個給定的軌道到一個時間 T，其中 T 是 movie header box 的 time scale，執行以下操作：

1. If the track contains an edit **list**, determine which edit contains the time T by iterating over the edits. The start time of the edit **in** the movie time scale must then be subtracted **from** the time T to generate T', the duration

into the edit in the movie time scale. **T' is next** converted to the time scale of the track's media to generate T''. Finally, the time in the media scale to use is calculated by adding the media start time of the edit to T''.

(如果 track 有 edit list，遍歷所有 edit，找到 T 在哪一個 track 里。將 edit 的開始時間轉換為 movie 的 time scale 為單位得到 edit_T，T 減去 edit_T，得到 T'，也就是在 edit 里面的持續時間。將 T' 轉換成 track 媒體的 time scale，得到 T''。最後將 T'' 加上 edit_T，可以得到以 track 媒體的 time scale 為單位的 T'''，而這個 T''' 就是後續用來求 sample 的時間)

2. The time-to-sample box for a track indicates what times are associated with which sample for that track. Use this box to find the first sample prior to the given time. EditListBox TimeToSampleBox SyncSampleBox SampleToChunkBox ChunkOffsetBox SampleSizeBox

3. The sample that was located in step 1 may not be a random access point. Locating the nearest random access point requires consulting two boxes. The sync sample table indicates which samples are in fact random access points. Using this table, you can locate which is the first sync sample prior to the specified time. The absence of the sync sample table indicates that all samples are synchronization points, and makes this problem easy. The shadow sync box gives the opportunity for a content author to provide samples that are not delivered in the normal course of delivery, but which can be inserted to provide additional random access points. This improves random access without impacting bitrate during normal delivery. This box maps samples that are not random access points to alternate samples

that are. You should also consult this table if present to find the first shadow sync sample prior to the sample in question. Having consulted the sync sample table and the shadow sync table, you probably wish to seek to whichever resultant sample is closest to, but prior to, the sample found in step 1.

4. At this point you know the sample that will be used for random access. Use the sample-to-chunk table to determine in which chunk this sample is located.

5. Knowing which chunk contained the sample in question, use the chunk offset box to figure out where that chunk begins.

6. Starting from this offset, you can use the information contained in the sample-to-chunk box and the sample size box to figure out where within this chunk the sample in question is located. This is the desired information.

2. 程式碼分析 MP4Box

Random Access 主要看以下兩個檔案 isom_read.c、stbl_read.c

```
1 /...../gpac-master/src/isomedia/isom\_read.c
```

```
2
```

```
3 /...../gpac-master/src/isomedia/stbl\_read.c
```

其中，對於時間 T 的轉換，並且獲得相應的 time scale 位於第一個 isom_read.c 里面。而在各個 box 中尋找 sample、chunk、offset……等，位於第二個 stbl_read.c 中。Step1、將時間已經轉化為 media time scale 單位，且獲得 sample。

isom_read.c 具體如下：


```

1 //return the timescale of the movie, 0 if error
2 GF_EXPORT
3 u32 gf_isom_get_timescale(GF_ISOFile *movie)
4 {
5     if (!movie || !movie->moov || !movie->moov->mvhd) return 0;
6     return movie->moov->mvhd->timeScale;
7 }
8
9
10 //return the duration of the movie, 0 if error
11 GF_EXPORT
12 u64 gf_isom_get_duration(GF_ISOFile *movie)
13 {
14     if (!movie || !movie->moov || !movie->moov->mvhd) return 0;
15
16     //if file was open in Write or Edit mode, recompute the duration
17     //the duration of a movie is the MaxDuration of all the tracks...
18
19 #ifndef GPAC_DISABLE_ISOM_WRITE
20     gf_isom_update_duration(movie);
21 #endif /*GPAC_DISABLE_ISOM_WRITE*/
22
23     return movie->moov->mvhd->duration;
24 }
25 //return the duration of the movie, 0 if error
26 GF_EXPORT
27 u64 gf_isom_get_original_duration(GF_ISOFile *movie)
28 {
29     if (!movie || !movie->moov || !movie->moov->mvhd) return 0;
30     return movie->moov->mvhd->original_duration;
31 }

```

```

1 //Return the media time given the absolute time in the Movie
2 GF_EXPORT
3 GF_Err gf_isom_get_media_time(GF_ISOFile *the_file, u32 trackNumber, u32
    movieTime, u64 *MediaTime)
4 {
5     GF_TrackBar *trak;
6     u8 useEdit;
7     s64 SegmentStartTime, mediaOffset;
8     trak = gf_isom_get_track_from_file(the_file, trackNumber);
9     if (!trak || !MediaTime) return GF_BAD_PARAM;
10
11     SegmentStartTime = 0;
12     return GetMediaTime(trak, GF_FALSE, movieTime, MediaTime, &
        SegmentStartTime, &mediaOffset, &useEdit, NULL);

```

```

13 }
14
15
16 //Get the stream description index (eg, the ESD) for a given time IN MEDIA
    TIMESCALE
17 //return 0 if error or if empty
18 GF_EXPORT
19 u32 gf_isom_get_sample_description_index(GF_ISOFile *movie, u32 trackNumber, u64
    for_time)
20 {
21     u32 streamDescIndex;
22     GF_TrackBar *trak;
23     trak = gf_isom_get_track_from_file(movie, trackNumber);
24     if (!trak) return 0;
25
26     if ( (movie->LastError = Media_GetSampleDescIndex(trak->Media, for_time,
        &streamDescIndex)) ) {
27         return 0;
28     }
29     return streamDescIndex;
30 }

```

```

1 //return a sample given a desired display time IN MEDIA TIME SCALE
2 //and set the StreamDescIndex of this sample
3 //this index allows to retrieve the stream description if needed (2 media in 1
    track)
4 //return NULL if error
5 //WARNING: the sample may not be sync even though the sync was requested (
    depends on the media)
6 GF_EXPORT
7 GF_Err gf_isom_get_sample_for_media_time(GF_ISOFile *the_file, u32 trackNumber,
    u64 desiredTime, u32 *StreamDescriptionIndex, GF_ISOSearchMode SearchMode,
    GF_ISOSample **sample, u32 *SampleNum, u64 *data_offset)
8 {
9     GF_Err e;
10    u32 sampleNumber, prevSampleNumber, syncNum, shadowSync;
11    GF_TrackBar *trak;
12    GF_ISOSample *shadow;
13    GF_SampleTableBox *stbl;
14    Bool static_sample = GF_FALSE;
15    u8 useShadow, IsSync;
16
17    if (SampleNum) *SampleNum = 0;
18    trak = gf_isom_get_track_from_file(the_file, trackNumber);
19    if (!trak) return GF_BAD_PARAM;
20
21    stbl = trak->Media->information->sampleTable;

```

```

22
23 #ifndef GPAC_DISABLE_ISOM_FRAGMENTS
24     if (desiredTime < trak->dts_at_seg_start) {
25         desiredTime = 0;
26     } else {
27         desiredTime -= trak->dts_at_seg_start;
28     }
29 #endif
30
31     e = stbl_findEntryForTime(stbl, desiredTime, 0, &sampleNumber, &
        prevSampleNumber);
32     if (e) return e;

```

Step2、接著在 stbl_read.c 中調用 stbl_findEntryForTime 函數，得到 sample 編號

```

1 //Get the sample number
2 GF_Err stbl_findEntryForTime(GF_SampleTableBox *stbl, u64 DTS, u8 useCTS, u32 *
    sampleNumber, u32 *prevSampleNumber)
3 {
4     u32 i, j, curSampNum, count;
5     s32 CTSOffset;
6     u64 curDTS;
7     GF_SttsEntry *ent;
8     (*sampleNumber) = 0;
9     (*prevSampleNumber) = 0;
10
11     if (!stbl->TimeToSample) return GF_ISOM_INVALID_FILE;
12
13     /*CTS is ALWAYS disabled for now to make sure samples are fetched in
        decoding order. useCTS is therefore disabled*/
14 #if 0
15     if (!stbl->CompositionOffset) useCTS = 0;
16 #endif
17
18     //our cache
19     if (stbl->TimeToSample->r_FirstSampleInEntry &&
        (DTS >= stbl->TimeToSample->r_CurrentDTS) ) {
20         //if we're using CTS, we don't really know whether we're in the
            good entry or not
21         //(eg, the real DTS of the sample could be in a previous entry
22         i = stbl->TimeToSample->r_currentEntryIndex;
23         curDTS = stbl->TimeToSample->r_CurrentDTS;
24         curSampNum = stbl->TimeToSample->r_FirstSampleInEntry;
25     } else {
26         i = 0;
27         curDTS = stbl->TimeToSample->r_CurrentDTS = 0;
28         curSampNum = stbl->TimeToSample->r_FirstSampleInEntry = 1;
29         stbl->TimeToSample->r_currentEntryIndex = 0;
30

```

```
31 | }
```

Step3、在 step1 中的可能不是 random access point，所以要找到最近的 random access point

```
1 //if no syncTable, disable syncSearching, as all samples ARE sync
2 if (! trak->Media->information->sampleTable->SyncSample) {
3     if (SearchMode == GF_ISOM_SEARCH_SYNC_FORWARD) SearchMode =
4         GF_ISOM_SEARCH_FORWARD;
5     if (SearchMode == GF_ISOM_SEARCH_SYNC_BACKWARD) SearchMode =
6         GF_ISOM_SEARCH_BACKWARD;
7 }
```

```
1 //get the sync sample num
2 if (IsSync) {
3     //get the SyncNumber
4     e = Media_FindSyncSample(trak->Media->information->sampleTable,
5                             sampleNumber, &syncNum, SearchMode);
6     if (e) return e;
7     if (syncNum) sampleNumber = syncNum;
8     syncNum = 0;
9 }
10 //if we are in shadow mode, get the previous sync sample
11 //in case we can't find a good SyncShadow
12 else if (SearchMode == GF_ISOM_SEARCH_SYNC_SHADOW) {
13     //get the SyncNumber
14     e = Media_FindSyncSample(trak->Media->information->sampleTable,
15                             sampleNumber, &syncNum,
16                             GF_ISOM_SEARCH_SYNC_BACKWARD);
17     if (e) return e;
18 }
```

調用 stbl_GetSampleRap 函數，找到與給定的 sample 編號最近的 RAP(random access point)，如果當前 sample 就是 RAP，則設置 RAP 這個 flag

```
1 //Retrieve closes RAP for a given sample - if sample is RAP, sets the RAP flag
2 GF_Err stbl_GetSampleRAP(GF_SyncSampleBox *stss, u32 SampleNumber, GF_ISOSAPType
3     *IsRAP, u32 *prevRAP, u32 *nextRAP)
4 {
5     u32 i;
6     if (prevRAP) *prevRAP = 0;
7     if (nextRAP) *nextRAP = 0;
8     (*IsRAP) = RAP_NO;
9     if (!stss || !SampleNumber) return GF_BAD_PARAM;
10
11     if (stss->r_LastSyncSample && (stss->r_LastSyncSample < SampleNumber))
12     {
13         i = stss->r_LastSampleIndex;
14     } else {
```

```

14         i = 0;
15     }
16     for (; i < stss->nb_entries; i++) {
17         //get the entry
18         if (stss->sampleNumbers[i] == SampleNumber) {
19             //update the cache
20             stss->r_LastSyncSample = SampleNumber;
21             stss->r_LastSampleIndex = i;
22             (*IsRAP) = RAP;
23         }
24         else if (stss->sampleNumbers[i] > SampleNumber) {
25             if (nextRAP) *nextRAP = stss->sampleNumbers[i];
26             return GF_OK;
27         }
28         if (prevRAP) *prevRAP = stss->sampleNumbers[i];
29     }
30     return GF_OK;
31 }

```

Step4、使用 sample_to_chunk box 找到 sample 的 chunk

```

1 //get the number of "ghost chunk" (implicit chunks described by an entry)
2 void GetGhostNum(GF_StscEntry *ent, u32 EntryIndex, u32 count, GF_SampleTableBox
   *stbl)
3 {
4     GF_StscEntry *nextEnt;
5     u32 ghostNum = 1;
6
7     if (!ent) {
8         stbl->SampleToChunk->ghostNumber = 0;
9         return;
10    }
11
12    if (!ent->nextChunk) {
13        if (EntryIndex+1 == count) {
14            //not specified in the spec, what if the last sample to
              chunk is no written?
15            if (stbl->ChunkOffset->type == GF_ISOM_BOX_TYPE_STCO) {
16                GF_ChunkOffsetBox *stco = (GF_ChunkOffsetBox *)
                  stbl->ChunkOffset;
17                ghostNum = (stco->nb_entries > ent->firstChunk)
                  ? (1 + stco->nb_entries - ent->firstChunk) :
                  1;
18            } else {
19                GF_ChunkLargeOffsetBox *co64 = (
                  GF_ChunkLargeOffsetBox *)stbl->ChunkOffset;
20                ghostNum = (co64->nb_entries > ent->firstChunk)
                  ? (1 + co64->nb_entries - ent->firstChunk) :

```

```

21         1;
22     }
23     } else {
24         //this is an unknown case due to edit mode...
25         nextEnt = &stbl->SampleToChunk->entries[EntryIndex+1];
26         ghostNum = nextEnt->firstChunk - ent->firstChunk;
27     }
28     } else {
29         ghostNum = (ent->nextChunk > ent->firstChunk) ? (ent->nextChunk
30         - ent->firstChunk) : 1;
31     }
32     stbl->SampleToChunk->ghostNumber = ghostNum;
33 }

```

Step5、根據 stco (ChunkOffsetBox) 獲取對應 Chunk 在文件中的偏移位置。

```

1 //Get the offset, descIndex and chunkNumber of a sample...
2 GF_Err stbl_GetSampleInfos(GF_SampleTableBox *stbl, u32 sampleNumber, u64 *
   offset, u32 *chunkNumber, u32 *descIndex, GF_StscEntry **out_ent)
3 {
4     GF_Err e;
5     u32 i, k, offsetInChunk, size, chunk_num;
6     GF_ChunkOffsetBox *stco;
7     GF_ChunkLargeOffsetBox *co64;
8     GF_StscEntry *ent;
9
10    (*offset) = 0;
11    (*chunkNumber) = (*descIndex) = 0;
12    if (!out_ent) (*out_ent) = NULL;
13    if (!stbl || !sampleNumber) return GF_BAD_PARAM;
14    if (!stbl->ChunkOffset || !stbl->SampleToChunk || !stbl->SampleSize)
        return GF_ISOM_INVALID_FILE;
15
16    if (stbl->SampleSize && stbl->SampleToChunk->nb_entries == stbl->
        SampleSize->sampleCount) {
17        ent = &stbl->SampleToChunk->entries[sampleNumber-1];
18        if (!ent) return GF_BAD_PARAM;
19        (*descIndex) = ent->sampleDescriptionIndex;
20        (*chunkNumber) = sampleNumber;
21        if (out_ent) *out_ent = ent;
22        if (stbl->ChunkOffset->type == GF_ISOM_BOX_TYPE_STCO) {
23            stco = (GF_ChunkOffsetBox *)stbl->ChunkOffset;
24            if (!stco->offsets) return GF_ISOM_INVALID_FILE;
25            if (stco->nb_entries < sampleNumber) return
                GF_ISOM_INVALID_FILE;
26
27            (*offset) = (u64) stco->offsets[sampleNumber - 1];
28        } else {

```

```

29         co64 = (GF_ChunkLargeOffsetBox *)stbl->ChunkOffset;
30         if (!co64->offsets) return GF_ISOM_INVALID_FILE;
31         if (co64->nb_entries < sampleNumber) return
            GF_ISOM_INVALID_FILE;
32
33         (*offset) = co64->offsets[sampleNumber - 1];
34     }
35     return GF_OK;
36 }

```

```

1     //ok, get the size of all the previous samples in the chunk
2     offsetInChunk = 0;
3     //constant size
4     if (stbl->SampleSize && stbl->SampleSize->sampleSize) {
5         u32 diff = sampleNumber - stbl->SampleToChunk->
            firstSampleInCurrentChunk;
6         offsetInChunk += diff * stbl->SampleSize->sampleSize;
7     } else if ((stbl->r_last_chunk_num == chunk_num) && (stbl->
            r_last_sample_num == sampleNumber)) {
8         offsetInChunk = stbl->r_last_offset_in_chunk;
9     } else if ((stbl->r_last_chunk_num == chunk_num) && (stbl->
            r_last_sample_num + 1 == sampleNumber)) {
10         e = stbl_GetSampleSize(stbl->SampleSize, stbl->r_last_sample_num
            , &size);
11         if (e) return e;
12         stbl->r_last_offset_in_chunk += size;
13         stbl->r_last_sample_num = sampleNumber;
14         offsetInChunk = stbl->r_last_offset_in_chunk;
15     } else {
16         //warning, firstSampleInChunk is at least 1 - not 0
17         for (i = stbl->SampleToChunk->firstSampleInCurrentChunk; i <
            sampleNumber; i++) {
18             e = stbl_GetSampleSize(stbl->SampleSize, i, &size);
19             if (e) return e;
20             offsetInChunk += size;
21         }
22         stbl->r_last_chunk_num = chunk_num;
23         stbl->r_last_sample_num = sampleNumber;
24         stbl->r_last_offset_in_chunk = offsetInChunk;
25     }
26     //OK, that's the size of our offset in the chunk
27     //now get the chunk
28     if (stbl->ChunkOffset->type == GF_ISOM_BOX_TYPE_STCO) {
29         stco = (GF_ChunkOffsetBox *)stbl->ChunkOffset;
30         if (stco->nb_entries < (*chunkNumber)) return
GF_ISOM_INVALID_FILE;
31         (*offset) = (u64) stco->offsets[(*chunkNumber) - 1] + (u64)

```



```

    offsetInChunk;
32     } else {
33         co64 = (GF_ChunkLargeOffsetBox *)stbl->ChunkOffset;
34         if (co64->nb_entries < (*chunkNumber) ) return
GF_ISOM_INVALID_FILE;
35         (*offset) = co64->offsets[(*chunkNumber) - 1] + (u64)
offsetInChunk;
36     }
37     return GF_OK;
38 }

```

Step6、根據 stsz (SampleSizeBox) 獲取 Sample 在 Chunk 內的偏移位置並加上第 4 步獲取的偏移，計算出 Sample 在文件中的偏移。

```

1  GF_Err stbl_SearchSAPs(GF_SampleTableBox *stbl, u32 SampleNumber, GF_ISOSAPType
   *IsRAP, u32 *prevRAP, u32 *nextRAP)
2  {
3      u32 i, j, count, count2;
4      assert(prevRAP);
5      assert(nextRAP);
6      (*prevRAP) = 0;
7      (*nextRAP) = 0;
8      (*IsRAP) = RAP_NO;
9
10     if (!stbl->sampleGroups || !stbl->sampleGroupsDescription) return GF_OK;
11
12     count = gf_list_count(stbl->sampleGroups);
13     count2 = gf_list_count(stbl->sampleGroupsDescription);
14     for (i=0; i<count; i++) {
15         GF_SampleGroupDescriptionBox *sgdp = NULL;
16         Bool is_rap_group = 0;
17         s32 roll_distance = 0;
18         u32 first_sample_in_entry, last_sample_in_entry;
19         GF_SampleGroupBox *sg = gf_list_get(stbl->sampleGroups, i);
20         switch (sg->grouping_type) {
21             case GF_ISOM_SAMPLE_GROUP_RAP:
22             case GF_ISOM_SAMPLE_GROUP_SYNC:
23                 is_rap_group = 1;
24                 break;
25             case GF_ISOM_SAMPLE_GROUP_ROLL:
26                 break;
27             default:
28                 continue;
29         }
30         for (j=0; j<count2; j++) {
31             sgdp = gf_list_get(stbl->sampleGroupsDescription, j);
32             if (sgdp->grouping_type==sg->grouping_type) break;
33             sgdp = NULL;

```

```

34     }
35     if (! sgdp) continue;
36
37     first_sample_in_entry=1;
38     for (j=0; j<sg->entry_count; j++) {
39         u32 first_rap_in_entry , last_rap_in_entry;
40         last_sample_in_entry = first_sample_in_entry + sg->
            sample_entries[j].sample_count - 1;
41
42         /*samples in this entry are not RAPs, continue*/
43         if (! sg->sample_entries[j].group_description_index) {
44             first_sample_in_entry += sg->sample_entries[j].
                sample_count;
45             continue;
46         }
47         if (!is_rap_group) {
48             GF_RollRecoveryEntry *entry = gf_list_get(sgdp->
                group_descriptions , sg->sample_entries[j].
                group_description_index - 1);
49             roll_distance = entry ? entry->roll_distance :
                0;
50         }
51
52         /*we consider the first sample in a roll or rap group
            entry to be the RAP (eg, we have to decode from this
            sample anyway)
53         except if roll_distance is strictly negative in which
            case we have to rewind our sample numbers from
            roll_distance*/
54         if (roll_distance < 0) {
55             if ((s32) first_sample_in_entry + roll_distance
                >=0) first_rap_in_entry =
                first_sample_in_entry + roll_distance;
56             else first_rap_in_entry = 0;
57
58             if ((s32) last_sample_in_entry + roll_distance
                >=0) last_rap_in_entry = last_sample_in_entry
                + roll_distance;
59             else last_rap_in_entry = 0;
60         } else {
61             first_rap_in_entry = first_sample_in_entry;
62             last_rap_in_entry = last_sample_in_entry;
63         }
64
65         /*store previous & next sample RAP - note that we do not
            store the closest previous RAP, only the first of
            the previous RAP group

```

```

66         as RAPs are usually isolated this should not be an issue
67         */
68         if (first_rap_in_entry <= SampleNumber) {
69             *prevRAP = first_rap_in_entry;
70         }
71         *nextRAP = last_rap_in_entry;
72
73         /*sample lies in this (rap) group, it is rap*/
74         if (is_rap_group) {
75             if ((first_rap_in_entry <= SampleNumber) && (
76                 SampleNumber <= last_rap_in_entry)) {
77                 (*IsRAP) = RAP;
78                 return GF_OK;
79             }
80         } else {
81             /*prevRAP or nextRAP matches SampleNumber,
82              sample is RAP*/
83             if ((*prevRAP == SampleNumber) || (*nextRAP ==
84                 SampleNumber)) {
85                 (*IsRAP) = RAP;
86                 return GF_OK;
87             }
88         }
89
90         /*first sample in entry is after our target sample,
91          abort*/
92         if (first_rap_in_entry > SampleNumber) {
93             break;
94         }
95         first_sample_in_entry += sg->sample_entries[j].
96             sample_count;
97     }
98 }
99 return GF_OK;
100 }

```