ubuCon ASIA 2021

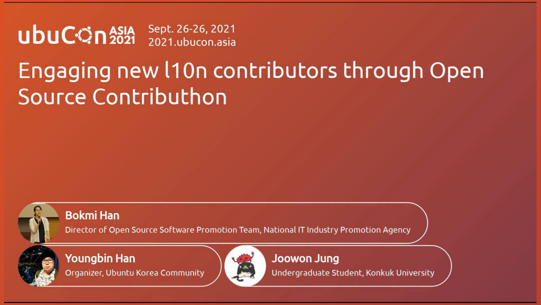
Automating boring and repetitive UbuCon Asia video and subtitle stuffs

Youngbin Han
youngbin@ubuntu-kr.org
launchpad.net/~ybhan

Hi, There!

- UbuCon Asia founder & organizer
  - It's my first experience to organize global event actually!
- Leader, Ubuntu Korea Community (Korean LoCo)
- Still a college student studying Software Engineering at Sungkonghoe University (on last semester! yay!)
- A DevOps engineer at Cloudmate Co. Ltd. who just started career last fall
- Some links
  - wiki.ubuntu.com/YoungbinHan
  - launchpad.net/~ybhan
  - github.com/sukso96100
  - youngbin.xyz

What we tried to automate

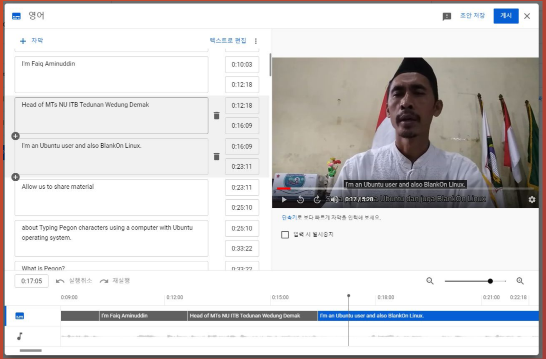

ubuCon ASIA 2021 Sept. 26-26, 2021 2021.ubucon.asia

Engaging new l10n contributors through Open Source Contributhon

Bokmi Han
Director of Open Source Software Promotion Team, National IT Industry Promotion Agency

Youngbin Han
Organizer, Ubuntu Korea Community

Joowon Jung
Undergraduate Student, Konkuk University



I'm Faiq Aminuddin	0:10:03 – 0:12:18
Head of MTs NU ITB Tedunan Wedung Demak	0:12:18 – 0:16:09
I'm an Ubuntu user and also BlankOn Linux.	0:16:09 – 0:23:11
Allow us to share material	0:23:11 – 0:25:10
about Typing Pegon characters using a computer with Ubuntu operating system.	0:25:10 – 0:33:22
What is Pegon?	0:33:22

I'm an Ubuntu user and also BlankOn Linux.
Saya pengguna Ubuntu dan juga BlankOn Linux
0:17 / 5:28

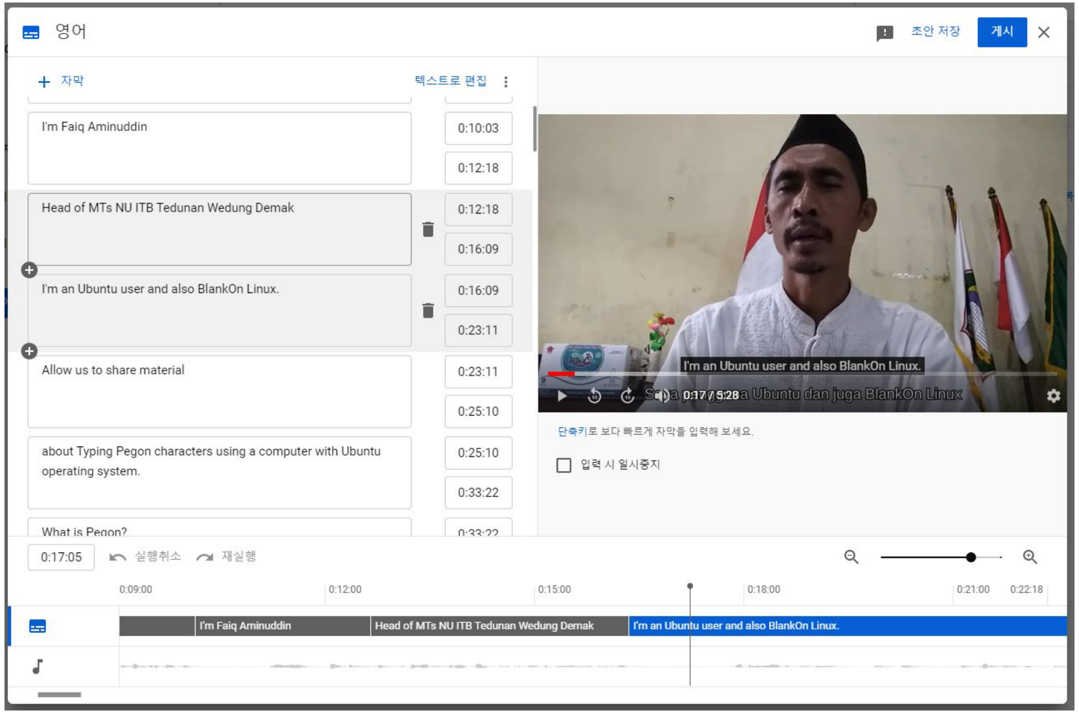
0:17:05

- Generating & Translating Subtitle for Non-English sessions.
  - Using some Speech-to-text, Machine translation APIs
- Generating and putting cover video(Session title, Speaker bio, sponsor logos) for each sessions recordings and merging with recordings
- Use GitHub Actions to trigger Video & SUbtitle batch jobs.

ubuCon ASIA 2021

Let’s talk about subtitles first.

Subtitle on Non-English sessions of UbuCon Asia 2021



I'm Faiq Aminuddin	0:10:03	0:12:18
Head of MTs NU ITB Tedunan Wedung Demak	0:12:18	0:16:09
I'm an Ubuntu user and also BlankOn Linux.	0:16:09	0:23:11
Allow us to share material	0:23:11	0:25:10
about Typing Pegon characters using a computer with Ubuntu operating system.	0:25:10	0:33:22
What is Pegon?	0:33:22	

I'm an Ubuntu user and also BlankOn Linux.

Sa[illegible]a Ubuntu dan juga BlankOn Linux

0:17 / 5:28

0:17:05

0:09:00 0:12:00 0:15:00 0:18:00 0:21:00 0:22:18

I'm Faiq Aminuddin | Head of MTs NU ITB Tedunan Wedung Demak | I'm an Ubuntu user and also BlankOn Linux.

We’ve been discussing about Non-English sessions and Subtitles for that from the beginning.

Hi, I’m looking for people who would like to join me to plan Ubuntu Asia virtual conference.
Would like to join to shape out some event details?

Yes, I like the idea of the event. But if we limit sessions to English only,
It will be difficult for me to gather enough speakers from my Local community.

Some LoCo organizers from country where English is not used as first language worried about language barrier even before forming the event committee.

Many reasons to worry about language barriers…
(For people from where English isn't first language)

- Really, Just don't understand or not good enough at English
- Can read and write English very well, But not good at and not confident with speaking.
    - Mostly because don't have much experience on having face to face conversation in English.
    - Or couldn't get enough chance to experience English conversation.
- Can read/write well, Fluent with English. But still not confident with giving presentation in English.
    - Worry about making mistakes during presentation.
    - Sometimes, Worry that other people from same country pointing out small mistakes.

In my cases, But not just limited to me.

Hey, I saw you've been working on interesting project,
Why don't you introduce that on UbuCon Asia?

Sure, It's my pleasure! But… **Can I give presentation in Korean?**

So, How we should accept and handle Non-English sessions?

- Non-English talks with translation or interpretation? (e.g. Language specific tracks on other conferences)
    - Easiest way, low cost, but only specific group of people can understand the stories. (e.g. Korean session: Only who do Korean can understand)
- Many offline global events: Hired professional interpreters.
    - This option is too much expensive for us!
- Hiring subtitle translators?
    - Still requires bunch of cost for hiring person.
- Volunteers?
    - We don't know if anyone will apply for this time consuming jobs of the new and not widely known event.

ubuCon ASIA 2021

Infinite discussion loop about dealing with language barriers of speakers that brought headaches and the loop seems to never ends continue forever.

If we limit session to English only, We can't bring enough speakers! →

We need to accept Non-English sessions! →

"Language-specific track without translation" Only limited participants can understand! →

Hire professional translators, interpreters or subtitle writers? →

That's too expensive! We don't know how much income we may earn from sponsorship or donations! →

Let's try to gather volunteers who will work on subtitles. →

We don't even know how many volunteers we'll be possible to gather for this new event! (+ I'm too busy with work/school to gather volunteers.) →

Could be too much work for volunteers even if we gather enough number of people. →

Why don't we ask speakers to provide script of their presentation? →

What kind of conference ask speakers to do that?! →

We don't have enough resource for this! It might be better to accept only English sessions! →

Trying to reduce efforts and spending

- YouTube’s Auto Captioning feature
  - Generates subtitles automatically. Not perfect, But mostly ok for understanding the context.
  - Needs some touch from human to improve machine translated result.
  - But language support is limited.
- Google Cloud: Speech to text, Translation API
  - Supports almost any languages.
  - Need some scripts to convert recognition output to *.srt file.
  - Paid, But expected to be cheaper than hiring professionals.
  - Like YouTube’s Auto Caption, Still need some touch from human.
- AWS Transcribe, Azure Speech to text
  - Didn’t considered, Because of their limited language support.

Before using it

- Still need volunteers, But not all local teams need to gather volunteer.
    - People from some regions are just fine with English presentation.
    - Check which local team will gather volunteers and which local team won’t.
- Quick PoCs with pyTranscriber that uses Google Speech to text.
    - Testing with tech conference video in Korean, Japanese, Chinese and Indonesian.
    - https://github.com/raryelcostasouza/pyTranscriber
    - Result was similar with what YouTube’s Auto Caption provided
    - Of course, the result really depends on quality of the audio in videos

Setting up scripts & pipelines

- Python Scripts: For generating subtitles, translating into English
  - Used script from Google Cloud tutorial repo https://github.com/GoogleCloudPlatform/community/tree/master/tutorials/speech2srt
  - The code was written for old SDK version, needed to modify it for recent version.
- GitHub Actions
  - For running scripts in the GitHub repo when it's needed to be run.
  - Also includes json file with video file and translation locale source and target information.
  - Use Matrix build to run tasks per each videos in parallel
- Google Cloud Python Libraries & CLIs (gsutil)
  - To interact with GCP programmatically on CI environment
  - gsutil CLI: to interact with GCP storage bucket on CI workflow

Useful function I newly learned: “fromJSON()”

- Parameters for matrix build (List of session informations) should not be hard-coded into workflow yaml file.
- Can read JSON data from *.json file or from environment variable.

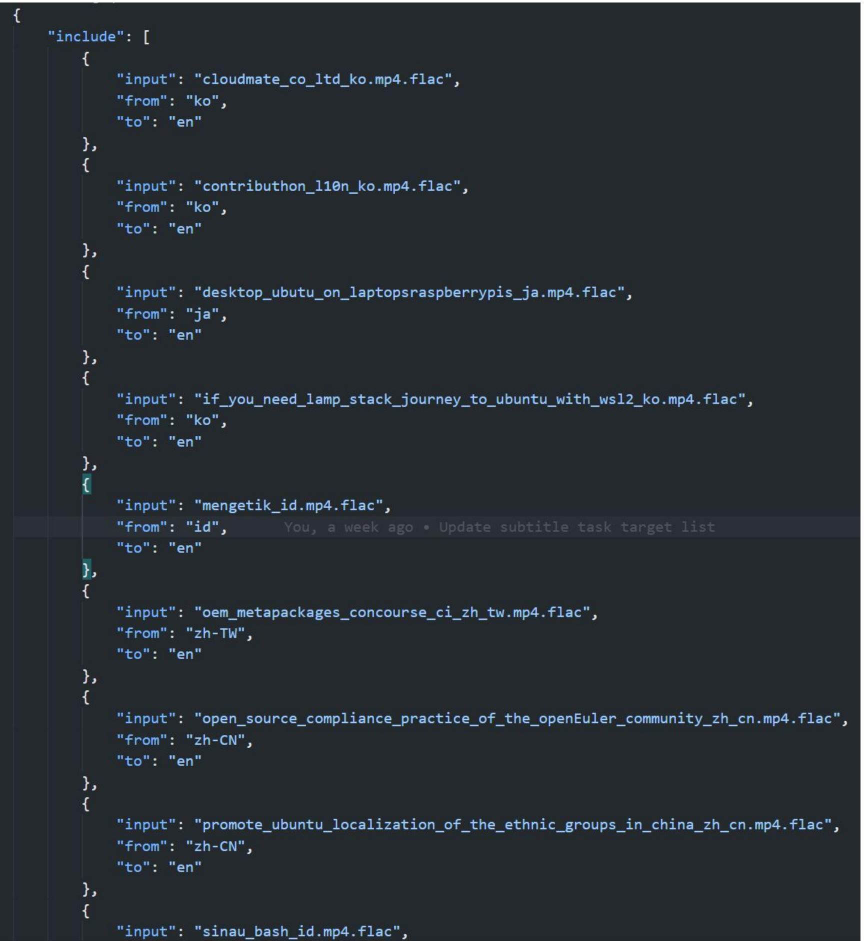
The JSON file for matrix build

```
{
    "include": [
        {
            "input": "contributhon_l10n_ko.mp4.flac",
            "from": "ko",
            "to": "en"
        },
        {
            "input": "desktop_ubutu_on_laptopsraspberrypis_ja.mp4.flac",
            "from": "ja",
            "to": "en"
        },
      ...
    ]
}]
```

Loading JSON file with “fromJSON()” on runtime

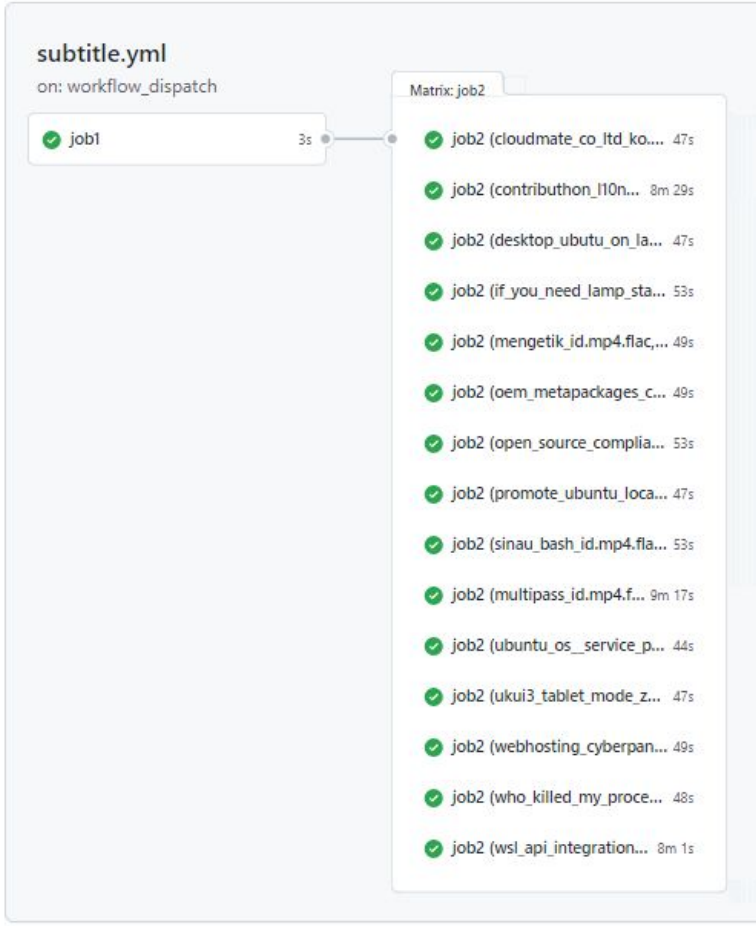
```yaml
jobs:
  job1:
    runs-on: ubuntu-latest
    outputs:
      matrix: ${{ steps.set-matrix.outputs.matrix }} # Declare Job output variable
    steps:
    - uses: actions/checkout@v2
    - id: set-matrix
      run: |
        JSON=$(cat ./videos.json) # Load JSON file into variable
        JSON="${JSON//'%'/%25}" # Handle escape sequence (%)
        JSON="${JSON//$'\n'/%0A}" # Handle escape sequence (\n)
        JSON="${JSON//$'\r'/%0D}" # Handle escape sequence (\r)
        echo "::set-output name=matrix::${JSON}" # Set Job output as one-line JSON string data

  job2:
    needs: job1
    runs-on: ubuntu-latest
    strategy:
      matrix: ${{fromJSON(needs.job1.outputs.matrix)}} # Get JSON string data from previous job and load as JSON object
```

fromJSON() in action



```
{
    "include": [
        {
            "input": "cloudmate_co_ltd_ko.mp4.flac",
            "from": "ko",
            "to": "en"
        },
        {
            "input": "contributhon_l10n_ko.mp4.flac",
            "from": "ko",
            "to": "en"
        },
        {
            "input": "desktop_ubutu_on_laptopsraspberrypis_ja.mp4.flac",
            "from": "ja",
            "to": "en"
        },
        {
            "input": "if_you_need_lamp_stack_journey_to_ubuntu_with_wsl2_ko.mp4.flac",
            "from": "ko",
            "to": "en"
        },
        {
            "input": "mengetik_id.mp4.flac",
            "from": "id",      You, a week ago • Update subtitle task target list
            "to": "en"
        },
        {
            "input": "oem_metapackages_concourse_ci_zh_tw.mp4.flac",
            "from": "zh-TW",
            "to": "en"
        },
        {
            "input": "open_source_compliance_practice_of_the_openEuler_community_zh_cn.mp4.flac",
            "from": "zh-CN",
            "to": "en"
        },
        {
            "input": "promote_ubuntu_localization_of_the_ethnic_groups_in_china_zh_cn.mp4.flac",
            "from": "zh-CN",
            "to": "en"
        },
        {
            "input": "sinau_bash_id.mp4.flac",
```

fromJSON()



subtitle.yml
on: workflow_dispatch

job1 3s

Matrix: job2

- job2 (cloudmate_co_ltd_ko.... 47s
- job2 (contributhon_l10n... 8m 29s
- job2 (desktop_ubutu_on_la... 47s
- job2 (if_you_need_lamp_sta... 53s
- job2 (mengetik_id.mp4.flac,... 49s
- job2 (oem_metapackages_c... 49s
- job2 (open_source_complia... 53s
- job2 (promote_ubuntu_loca... 47s
- job2 (sinau_bash_id.mp4.fla... 53s
- job2 (multipass_id.mp4.f... 9m 17s
- job2 (ubuntu_os_service_p... 44s
- job2 (ukui3_tablet_mode_z... 47s
- job2 (webhosting_cyberpan... 49s
- job2 (who_killed_my_proce... 48s
- job2 (wsl_api_integration... 8m 1s

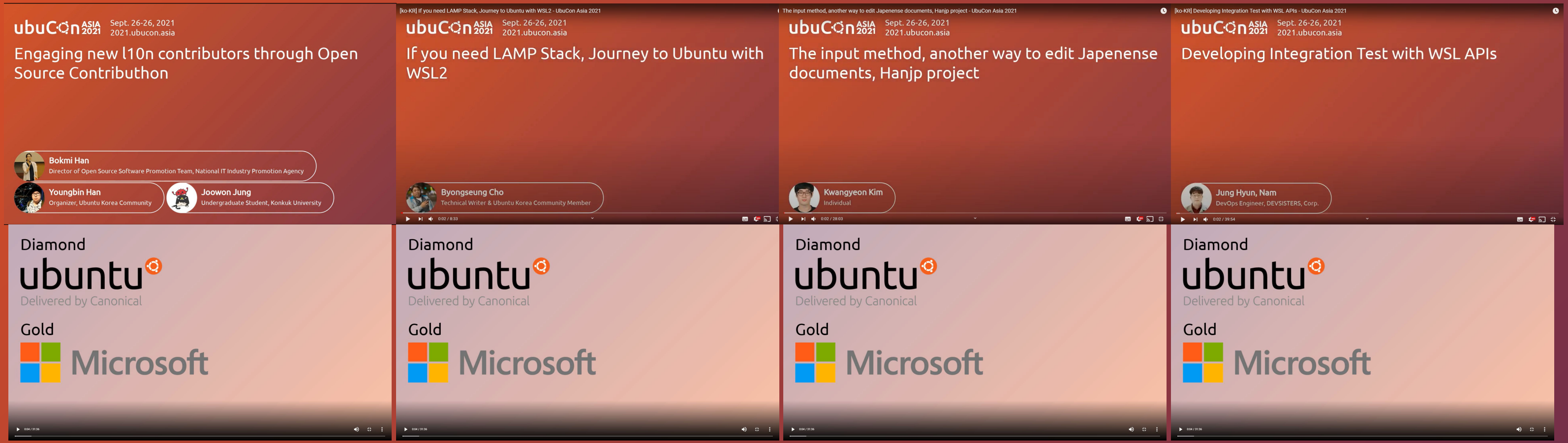
So, Was it worked?

- In most cases, yes. (Korean, Indonesian, Chinese(China))
- Did not worked well with some video:
  - If the breathing between words or sentences too long (Japanese)
    - Google Cloud speech-to-text placed “|” between words. weird.
  - If the audio includes noises and sounds blurry (Chinese(Taiwan)
    - Drops some whole sentences.
- Other Speech-to-text are better in some cases
  - The recording in Japanese: Did not work well with Google Cloud Speech-to-text but worked ok with YouTube Auto Caption. (Another weird thing…)
  - The recording in Chinese(Taiwan): YouTube Auto Caption has no support for Chinese, Tried with Azure Speech-to-text instead. The result wasn’t still great enough. But slightly better then GCP’s. (Didn’t drop some whole sentences.)

Putting short introductory video on each recordings

We automated generating and translating subtitle with CI. Why not also automate rendering videos?

Because, creating these videos short videos and putting on the beginning of each recordings by hand is really… boring and repetitive job.


ubuCon ASIA 2021 Sept. 26-26, 2021 2021.ubucon.asia

Engaging new l10n contributors through Open Source Contributhon

Bokmi Han, Director of Open Source Software Promotion Team, National IT Industry Promotion Agency

Youngbin Han, Organizer, Ubuntu Korea Community

Joowon Jung, Undergraduate Student, Konkuk University

[ko-KR] If you need LAMP Stack, Journey to Ubuntu with WSL2 - UbuCon Asia 2021

ubuCon ASIA 2021 Sept. 26-26, 2021 2021.ubucon.asia

If you need LAMP Stack, Journey to Ubuntu with WSL2

Byongseung Cho, Technical Writer & Ubuntu Korea Community Member

The input method, another way to edit Japenense documents, Hanjp project - UbuCon Asia 2021

ubuCon ASIA 2021 Sept. 26-26, 2021 2021.ubucon.asia

The input method, another way to edit Japenense documents, Hanjp project

Kwangyeon Kim, Individual

[ko-KR] Developing Integration Test with WSL APIs - UbuCon Asia 2021

ubuCon ASIA 2021 Sept. 26-26, 2021 2021.ubucon.asia

Developing Integration Test with WSL APIs

Jung Hyun, Nam, DevOps Engineer, DEVSISTERS, Corp.

Diamond ubuntu Delivered by Canonical

Gold Microsoft

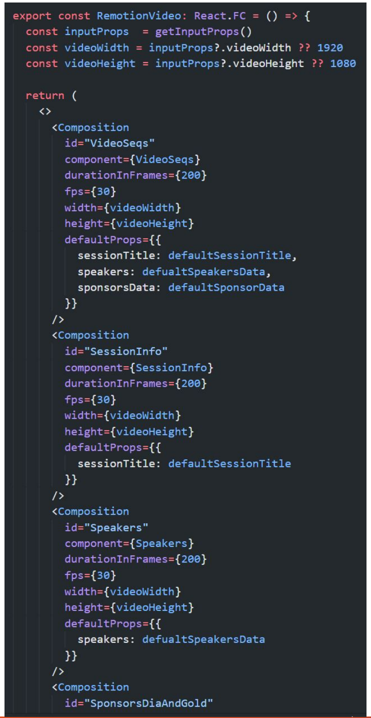
•••

•••

20 Pre-recordings + 10 Post-event recordings...

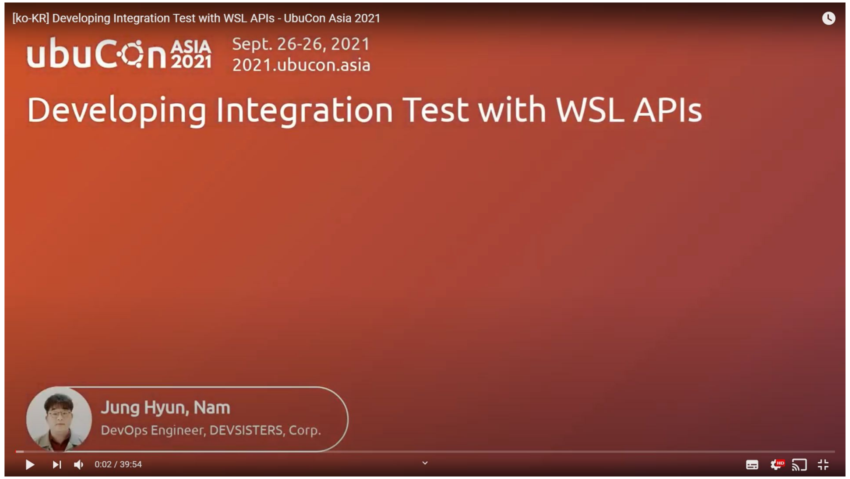
Trying out Remotion.dev: Write(!) video with React.js

github.com/remotion-dev/remotion

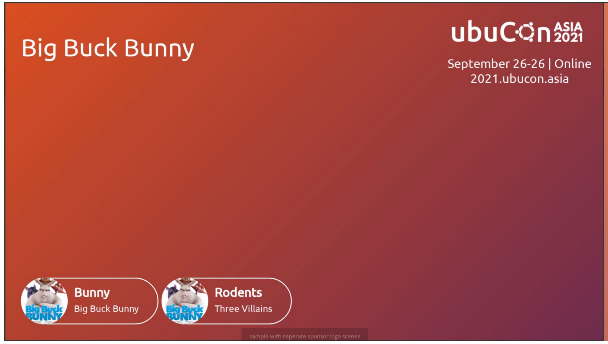


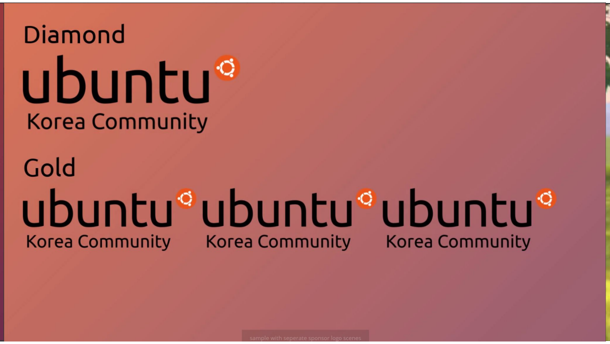
```
export const RemotionVideo: React.FC = () => {
  const inputProps  = getInputProps()
  const videoWidth = inputProps?.videoWidth ?? 1920
  const videoHeight = inputProps?.videoHeight ?? 1080

  return (
    <>
      <Composition
        id="VideoSeqs"
        component={VideoSeqs}
        durationInFrames={200}
        fps={30}
        width={videoWidth}
        height={videoHeight}
        defaultProps={{
          sessionTitle: defaultSessionTitle,
          speakers: defualtSpeakersData,
          sponsorsData: defaultSponsorData
        }}
      />
      <Composition
        id="SessionInfo"
        component={SessionInfo}
        durationInFrames={200}
        fps={30}
        width={videoWidth}
        height={videoHeight}
        defaultProps={{
          sessionTitle: defaultSessionTitle
        }}
      />
      <Composition
        id="Speakers"
        component={Speakers}
        durationInFrames={200}
        fps={30}
        width={videoWidth}
        height={videoHeight}
        defaultProps={{
          speakers: defualtSpeakersData
        }}
      />
      <Composition
        id="SponsorsDiaAndGold"
```

npx remotion render ...


[ko-KR] Developing Integration Test with WSL APIs - UbuCon Asia 2021
ubuCon ASIA 2021
Sept. 26-26, 2021
2021.ubucon.asia
Developing Integration Test with WSL APIs
Jung Hyun, Nam
DevOps Engineer, DEVSISTERS, Corp.
0:02 / 39:54

Writing video template &
testing with 30s Big Buck Bunny video





Testing with long videos (30mins, 1hours) on local machine

Seemed to be ok

- 30min ~ 45min videos: OK
- 1hour videos: Also OK
- Much longer videos
  - Fumihito(JP LoCo) tested with 2~3 hours videos: Mostly OK
- But It took too much time to render
  - Took 2~4 hours to render 30~45min videos on laptop with Intel 10th Gen i7
  - This means It will take much longer time on GitHub Actions

Modify remotion video to accept parameters

- Update video code to accept JSON data as parameters
  - e.g. Title, Speaker info, Sponsor logos…
  - https://www.remotion.dev/docs/parametrized-rendering

```
{
  "videoPath": "having_fun_flutter.mp4",
  "sessionTitle": "Having fun with Flutter Desktop development",
  "speakers": [
    {
      "name": "Rafal Wachol",
      "bio": "Software Engineer, Flutter Community",
      "photoPath": "https://2021.ubucon.asia/.../profile.jpg"
    }
  ],
  "sponsorsData":[...]
}
```

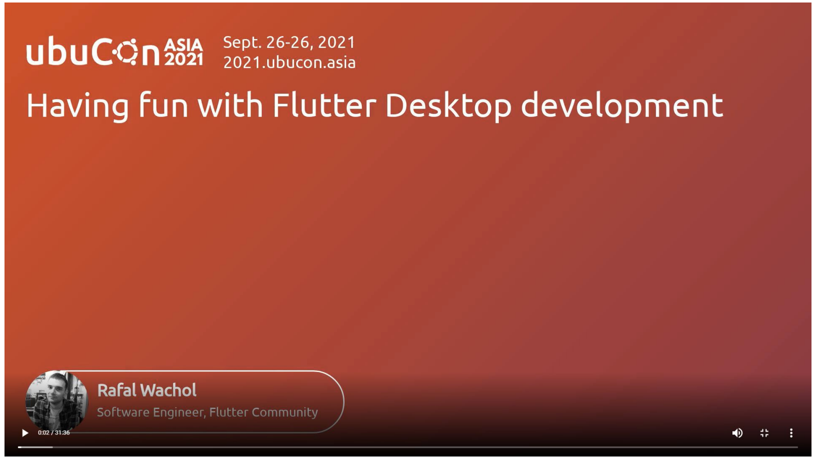
having_fun_flutter.json

```
{
  "videoPath": "oem_metapackages_concourse_ci.mp4",
  "sessionTitle": "OEM metapackages & Concourse CI",
  "speakers": [
    {
      "name": "Shih-Yuan Lee (FourDollars)",
      "bio": "Software Engineer, Canonical",
      "photoPath": "https://2021.ubucon.asia/.../profile.jpg"
    }
  ],
  "sponsorsData":[...]
}
```

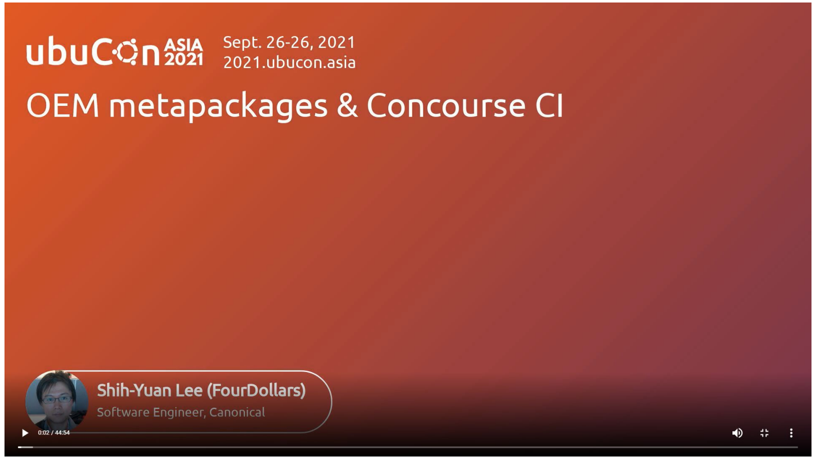
oem_metapackages_concourse_ci_zh_tw.json

Modify remotion video to accept parameters

- Update video code to accept JSON data as parameters
  - e.g. Title, Speaker info, Sponsor logos…
  - https://www.remotion.dev/docs/parametrized-rendering


ubuCon ASIA 2021
Sept. 26-26, 2021
2021.ubucon.asia
Having fun with Flutter Desktop development
Rafal Wachol
Software Engineer, Flutter Community
0:02 / 31:36

having_fun_flutter.mp4


ubuCon ASIA 2021
Sept. 26-26, 2021
2021.ubucon.asia
OEM metapackages & Concourse CI
Shih-Yuan Lee (FourDollars)
Software Engineer, Canonical
0:02 / 44:54

oem_metapackages_concourse_ci_zh_tw.mp4

Also use Matrix build with fromJSON() of course.

A difference: Run glue script to generate JSON file for matrix build on runtime.

JSON parameter files

cloudmate_co_ltd_ko.json
contributhon_l10n_ko.json
fast_reboot_kexec.json
hanjp_im_project.json
having_fun_flutter.json
ukui3_tablet_mode_zh_cn.json
webhosting_cyberpanel_id.json
who_killed_my_processes_ko.json
wsl_api_integration_test_ko.json
...

build_matrix.py

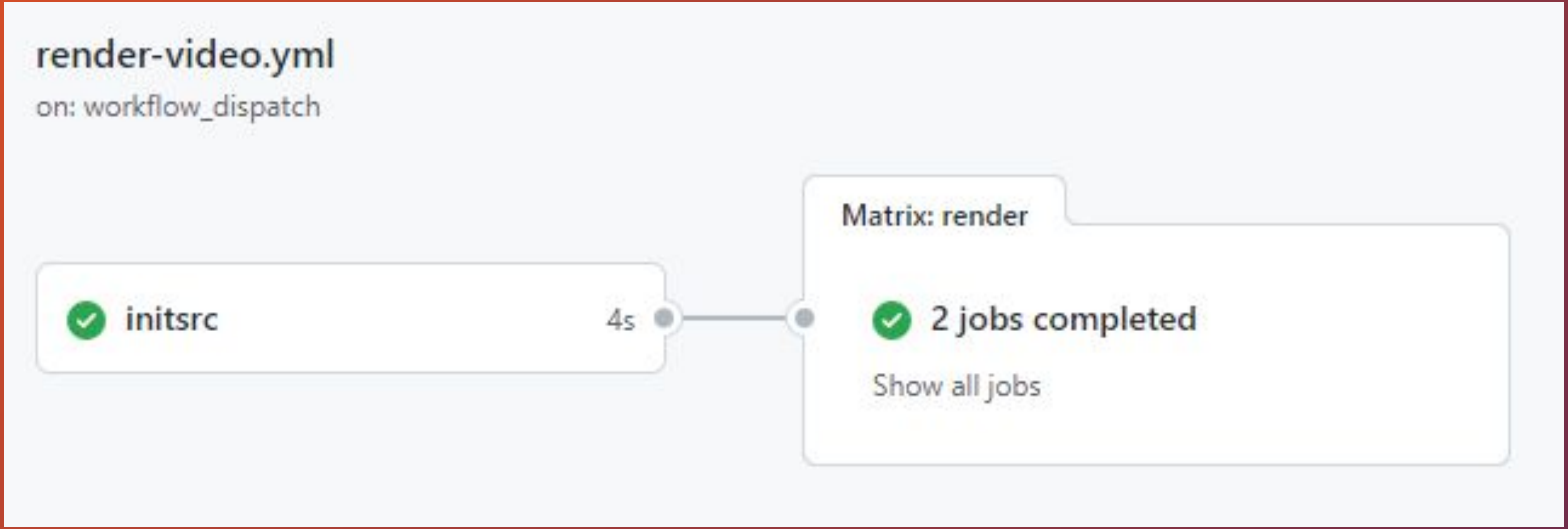
JSON for Matrix build parameter

```
[
        {
        “path”:”cloudmate_co_ltd_ko.json”,
        “output”:”cloudmate_co_ltd_ko.mp4”,
        ”videoPath”:”cloudmate_co_ltd_ko.mp4”
        },
        {
        “path”:”contributhon_l10n_ko.json”,
        “output”:”contributhon_l10n_ko.mp4”,
        ”videoPath”:”contributhon_l10n_ko.mp4”
        }
        ...
]
```

Load into workflow

```
jobs:
  initsrc:
    runs-on: ubuntu-latest
    outputs:
      matrix: ${{ steps.set-matrix.outputs.matrix }}
    steps:
    - uses: actions/checkout@v2
    - name: Setup build matrix
      id: set-matrix
      run: |
        python build_matrix.py
        JSON=$(cat ./matrix.json)
        JSON="${JSON//'%'/%25}"
        JSON="${JSON//$'\n'/%0A}"
        JSON="${JSON//$'\r'/%0D}"
        echo "::set-output name=matrix::${JSON}"
  render:
    needs: initsrc
    runs-on: ubuntu-latest
    strategy:
      matrix: ${{fromJSON(needs.initsrc.outputs.matrix)}}
```

Seemed to be worked well with few short videos...


render-video.yml

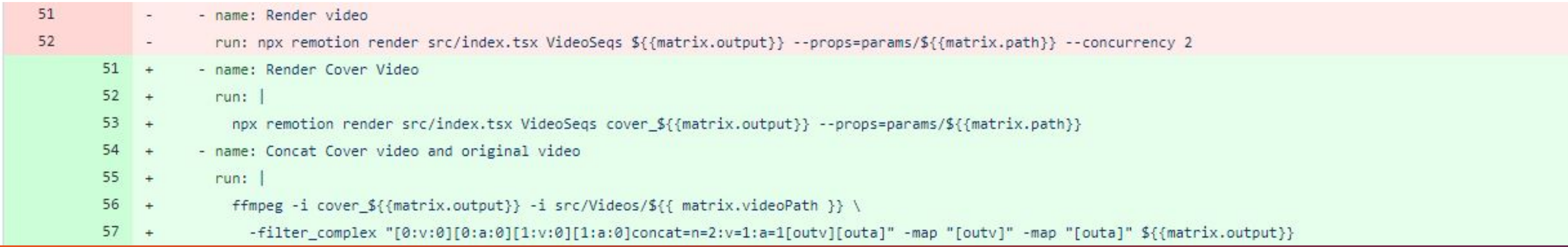
on: workflow_dispatch

Then… Met another bunch of exceptions

- Couldn’t handle videos bigger than 2GB
  - Node.js’s V8 limits maximum size of file to 2GB that can be loaded into buffer.
- Long videos took forever to render with cover
  - GitHub Actions limits single workflow run to maximum 6 hours.
  - Remotion uses headless chromium to render video
- Some speakers submitted videos in different resolution or aspect ratio.
  - We told speakers to submit 1920*1080 video.
  - But some people gave us 1080*720 or even different aspect ratio, 1140*1080

Dealing with V8’s 2GB limit, Remotion’s render time issue

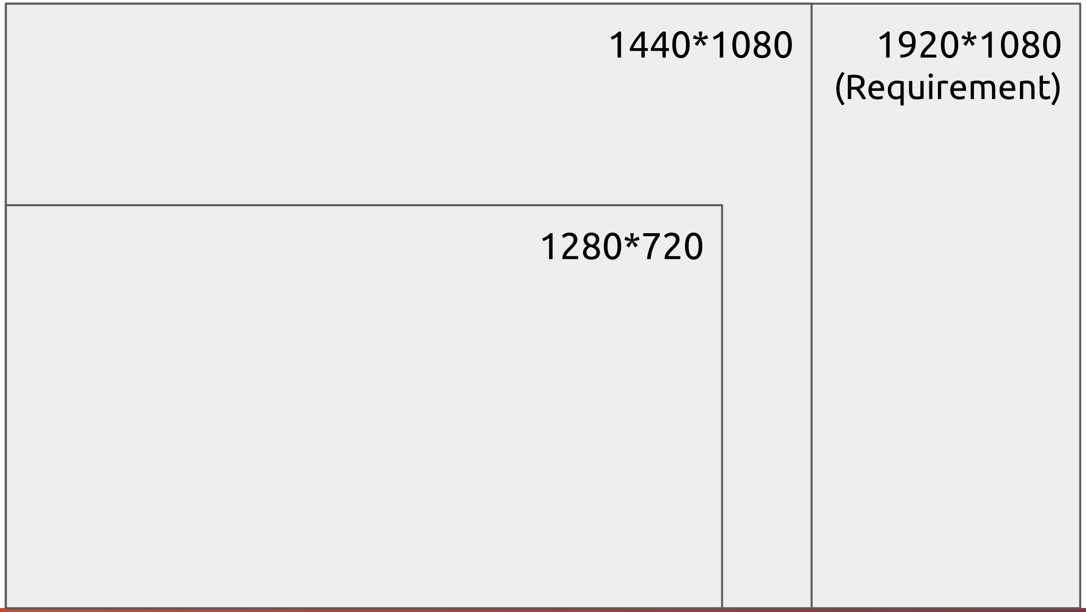
- Previously, Used both cover video and recordings on Remotion’s timeline
- New way: Use Remotion for just rendering short cover video and concatenate with session recording using ffmpeg.
    - Remove session recording sequence and parameters from Remotion code.
- Used ffmpeg’s concatenate filter to concatenate videos.
    - https://trac.ffmpeg.org/wiki/Concatenate



```diff
-     - name: Render video
-       run: npx remotion render src/index.tsx VideoSeqs ${{matrix.output}} --props=params/${{matrix.path}} --concurrency 2
+     - name: Render Cover Video
+       run: |
+         npx remotion render src/index.tsx VideoSeqs cover_${{matrix.output}} --props=params/${{matrix.path}}
+     - name: Concat Cover video and original video
+       run: |
+         ffmpeg -i cover_${{matrix.output}} -i src/Videos/${{ matrix.videoPath }} \
+           -filter_complex "[0:v:0][0:a:0][1:v:0][1:a:0]concat=n=2:v=1:a=1[outv][outa]" -map "[outv]" -map "[outa]" ${{matrix.output}}
```

Dealing videos with different resolution and aspect ratio

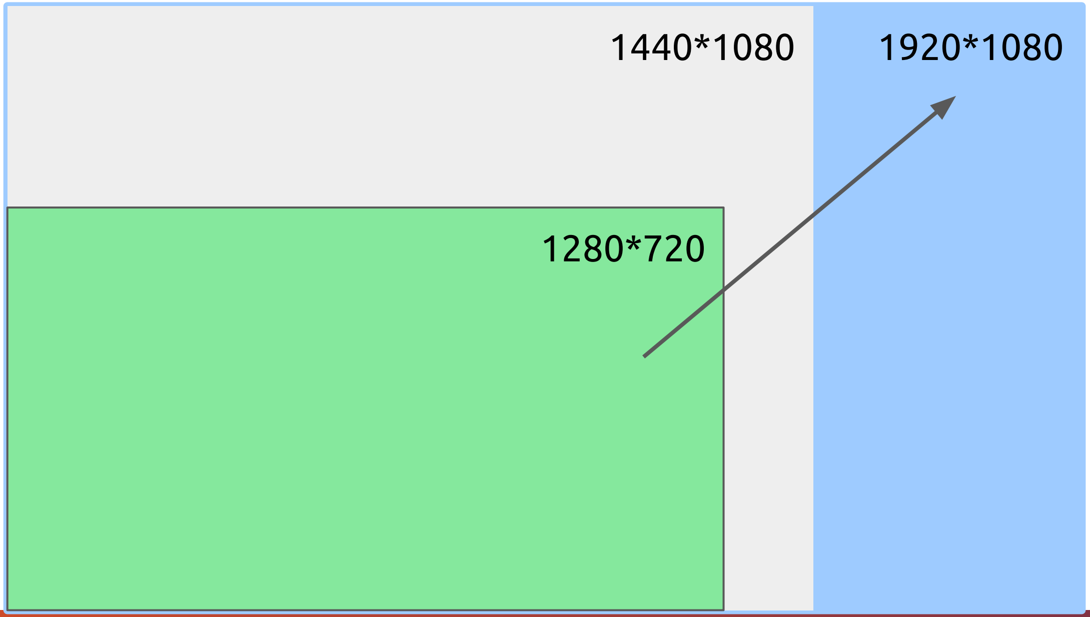
- ffmpeg Can’t concatenate videos with different resolution.
- Also It can’t concatenate videos with different SAR(Sample aspect ratio)



Same aspect ratio (16:9) with different resolution

- Upscaled using ffmpeg's scale filter and lanczos algorithm
  - http://trac.ffmpeg.org/wiki/Scaling

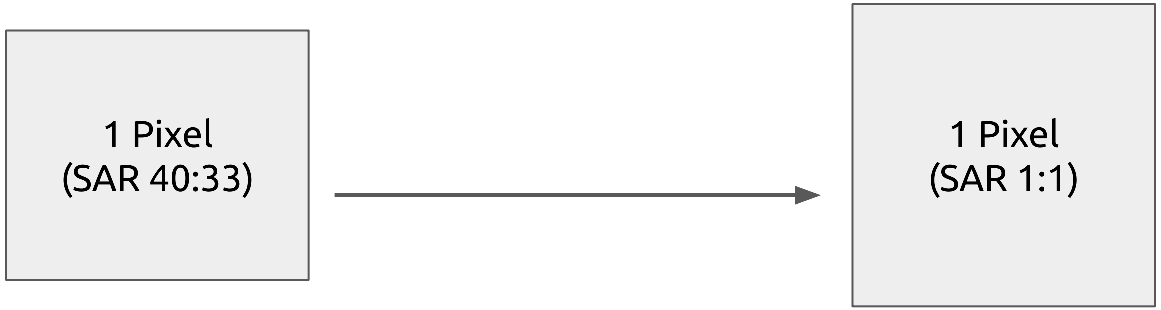
```
ffmpeg -i input.mp4 -vf scale=1920x1080:flags=lanczos output.mp4
```



Same resolution with different SAR(Sample Aspect Ratio)

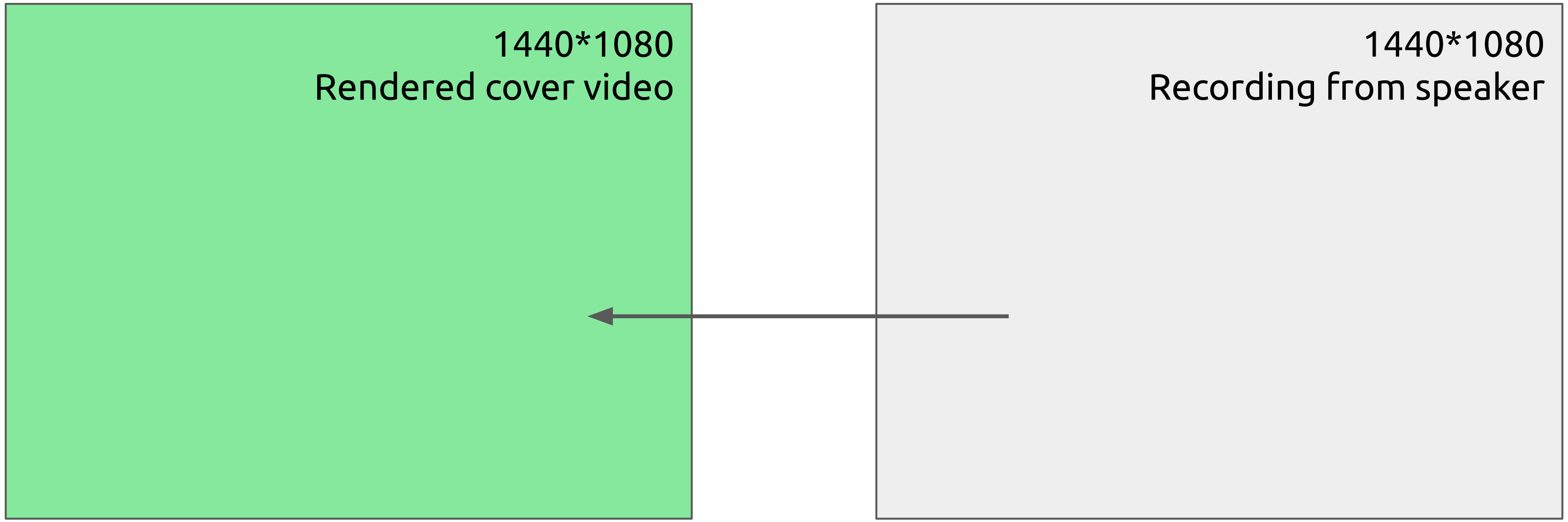
- Upscaled using ffmpeg's setsar filter
  - https://ffmpeg.org/ffmpeg-filters.html#setdar_002c-setsar

```
ffmpeg -i input.mp4 -vf "setsar=1" output.mp4
```



Recording with different aspect ratio (e.g. 1440*1080)

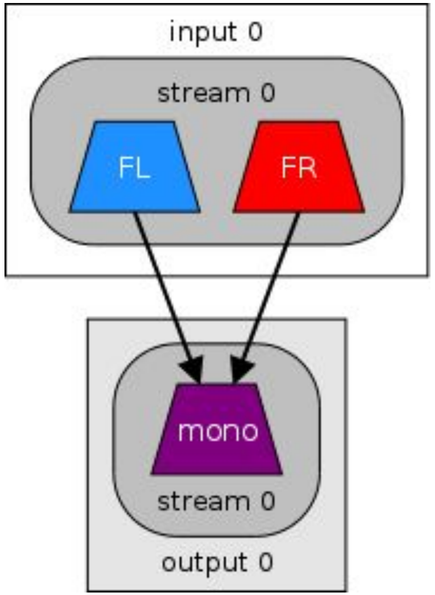
- Different aspect ratio, But width or height is same with required resolution.
    - e.g. -> 1440*1080's width is different but height is same with 1920*1080
- Modify Remotion code to accept video resolution information, and render cover video with that resolution.



Extract audio file from rendered video

- Audio extracted from this step is used for generating subtitles
  - https://trac.ffmpeg.org/wiki/AudioChannelManipulation

```
ffmpeg -y -i input.mp4 -ac 1 output.flac
```



Video input

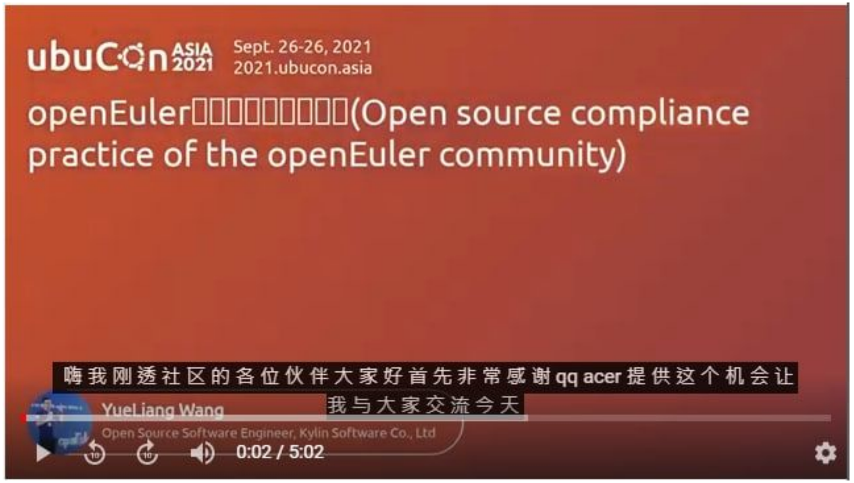
Downmix to mono stream

Extracted mono audio output

Don’t forget to install CJK fonts!
(or other required fonts if any)

- Just install font packages (e.g. Noto Fonts) on CI before running video rendering task

```
sudo apt install fonts-noto-cjk
```



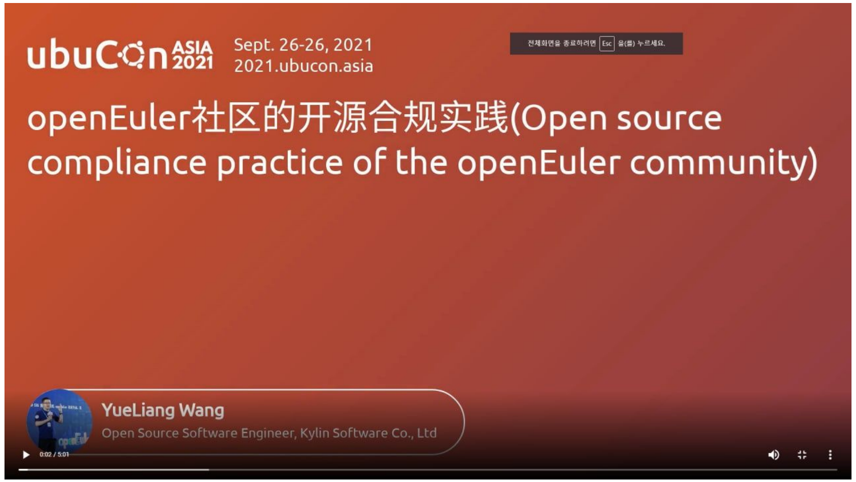
ubuCon ASIA 2021 Sept. 26-26, 2021 2021.ubucon.asia

openEuler□□□□□□□□□□(Open source compliance practice of the openEuler community)

嗨我刚透社区的各位伙伴大家好首先非常感谢qq acer提供这个机会让
我与大家交流今天

YueLiang Wang
Open Source Software Engineer, Kylin Software Co., Ltd

0:02 / 5:02



ubuCon ASIA 2021 Sept. 26-26, 2021 2021.ubucon.asia

전체화면을 종료하려면 Esc 을(를) 누르세요.

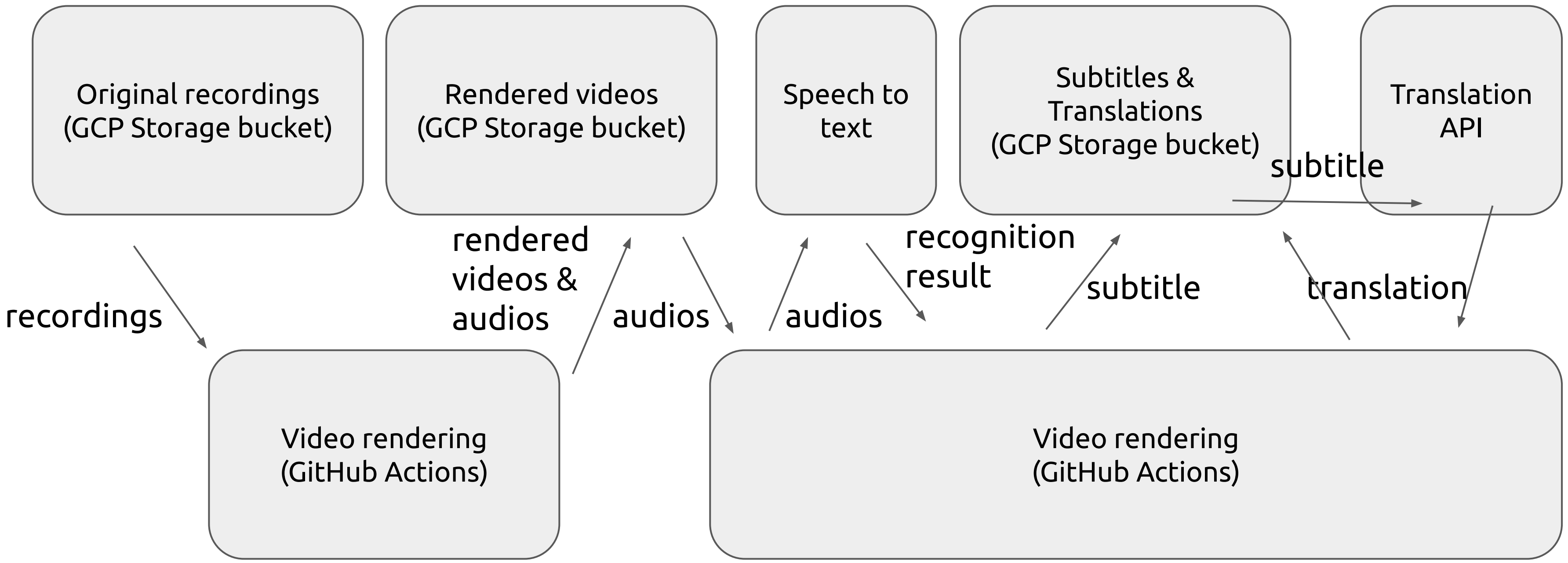
openEuler社区的开源合规实践(Open source compliance practice of the openEuler community)

YueLiang Wang
Open Source Software Engineer, Kylin Software Co., Ltd

0:02 / 5:01

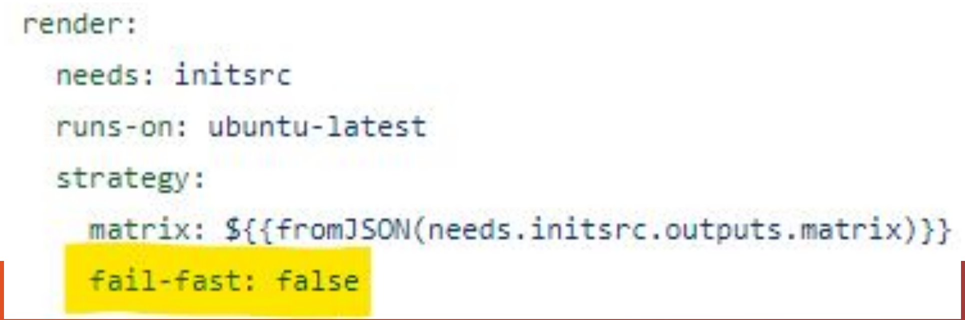
Putting things all together and some troubleshootings

Putting things all together


Original recordings (GCP Storage bucket)
Rendered videos (GCP Storage bucket)
Speech to text
Subtitles & Translations (GCP Storage bucket)
Translation API
subtitle
recordings
rendered videos & audios
audios
audios
recognition result
subtitle
translation
Video rendering (GitHub Actions)
Video rendering (GitHub Actions)

One more thing to fix: Disabling fail-fast strategy

- CI/CD in many software projects used to find any bugs or other problems quickly, earlier and then fix it.
- To do that, the “fail-fast” strategy is enabled by default in GitHub Actions.
  - On matrix build, If any parallel job fails, other all running parallel job stops automatically.
- This is good for software development, But what we’re doing here is making GitHub Actions to do boring, repetitive and time consuming job.
  - If all other video rendering tasks stops because of one failure. We need run a fresh job and render all things from the start again. Which is very inefficient.
  - We want other parallel jobs to continue even if there is failed parallel jobs.
- So, We disabled “fail-fast” strategy.



```
render:
  needs: initsrc
  runs-on: ubuntu-latest
  strategy:
    matrix: ${{fromJSON(needs.initsrc.outputs.matrix)}}
    fail-fast: false
```

Conclusion

- We wanted to accept Non-English session that most people can understand the context. Which requires subtitle or interpretation.
- We choose to use automation tools to generate and translate subtitle and ask volunteers to review and correct it.
  - Used Google's Speech to text and Translation API
  - We could reduce amount of work for volunteers but it highly depends on audio quality and speaker's pronunciation.
- We automated video rendering job using Remotion, ffmpeg and GitHub Actions
  - Writing video with react.js and configuring it to accept parameter is powerful to render introductory video with same format.
  - ffmpeg is useful to automate various video related jobs: concatenate, rescale, extract audio, adjusting pixel aspect ratios, etc.
  - Remotion uses Node.js and Headless Chromium: Can't load file bigger the 2GB, Takes bunch of time for rendering long video.
- Considerations when using GitHub Actions: 6h limit, fail-fast by default

Thank you!

Repos

https://github.com/ubucon-asia/2021-video-template

https://github.com/ubucon-asia/video-transcribe-tools

ubuCon ASIA 2021