.Deb

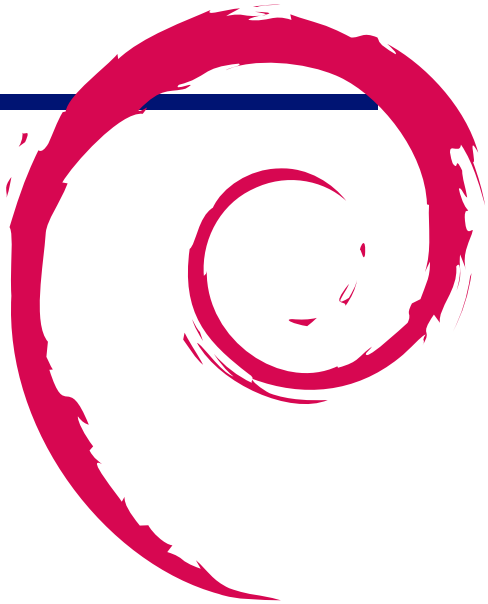
銀河系唯一のDebian専門誌

2020 年 4 月 18 日

Wireguard・Jitsi特集

デビアン勉強会

目次



1 最近の Debian 関連のミーティング報告

杉本　典充

1.1 2020 年 3 月度 東京エリア・関西合同 Debian 勉強会

2020 年 3 月 21 日 (土) に東京エリア Debian 勉強会と関西 Debian 勉強会の合同でオンラインによる Debian 勉強会を開催しました。参加者は 15 名でした。

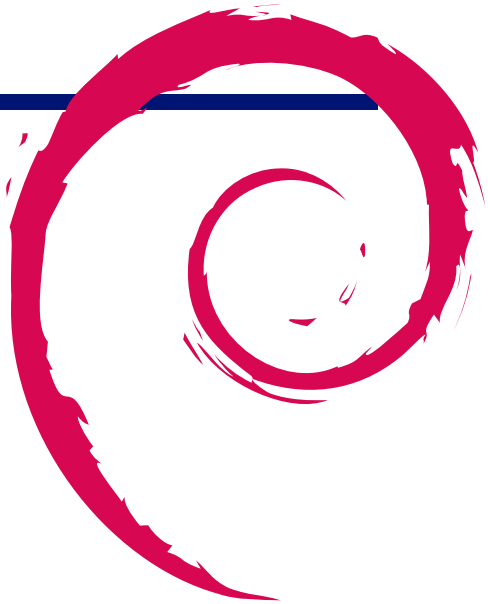
オンラインで行うにあたりいくつかのビデオ会議システムを試験利用しながら進行しました。利用したビデオ会議システムは Google ハングアウト、上川さんが独自に構築したストリーミング配信サーバ、Jitsi サーバでした。

前半はセミナー発表を行い、タイトル「動画配信環境をしらべてみた」として上川さんが発表しました。RTMP というストリーミング配信を行うプロトコルの説明、RTMP に対応したストリーミング配信サーバを動作させて勉強会の参加者は VLC や ffplay コマンドでストリーミング配信を視聴することができました。また OBS Studio という RTMP を扱う便利なアプリケーションの紹介もあり、ストリーミング配信の仕組みを学ぶことができました。

後半は BoF を行い、議題「今回のビデオ会議システムの品質について、流行りのビデオ会議システムについて」として情報を持ち寄りつつ、勉強会当日に利用したビデオ会議システムや代替のビデオ会議システムについて情報共有を行いました。

BoF の議事録は以下 URL で公開しています。

- `https://tokyodebian-team.pages.debian.net/2020-03_tokyodebian_bof.txt`



2 事前課題

杉本 典充

今回の事前課題は以下です。

1. Wireguard はご存じですか
2. Jitsi はご存じですか

2.1 dictoss

1. 知らない
2. 使ったことがある

2.2 yy_y_ja_jp

1. 知らない
2. 使ったことがある

2.3 uwabami

1. 知っているが、使ったことはない
2. 知っているが、使ったことはない

2.4 yosuke_san

1. 知っているが、使ったことはない
2. 知っているが、使ったことはない

2.5 t3rkwd

1. 知っているが、使ったことはない
2. 使ったことがある

2.6 koedoyoshida

1. 知っているが、使ったことはない
2. 知っているが、使ったことはない

2.7 kare-zeri-123

1. 知らない
2. 知らない

2.8 hatsanhat

1. 知らない
2. 知っているが、使ったことはない

2.9 Kazuhiro NISHIYAMA (znz)

1. 使ったことがある
2. 使ったことがある

2.10 TANIGUCHI Takaki (takaki_t)

1. 知っているが、使ったことはない
2. 知らない

2.11 SHIMOYAMA Yoshihiro (kurokouji)

1. 知らない
2. 知らない

2.12 sato_makoto

1. 知らない
2. 知っているが、使ったことはない

2.13 fpond

1. 知らない
2. 知らない

2.14 su_do

1. 知らない
2. 知らない

2.15 **榎真治** (enoki)

1. 知っているが、使ったことはない
2. 使ったことがある

2.16 **さとうとものり** (totosan365)

1. 知っているが、使ったことはない
2. 知っているが、使ったことはない

2.17 zinrai

1. 知っているが、使ったことはない
2. 知らない

2.18 ysaito

1. 知らない
2. 使ったことがある

2.19 matoken

1. 使ったことがある
2. 使ったことがある

2.20 dancerj

1. 知らない
2. 使ったことがある

2.21 ipv6waterstar

1. 知らない
2. 知らない

2.22 kozo2

1. 知らない
2. 知っているが、使ったことはない

2.23 lurdan

1. 知っているが、使ったことはない
2. 使ったことがある

2.24 NOKUBI Takatsugu (knok)

1. 使ったことがある
2. 使ったことがある

2.25 yuyhiraka

1. 知っているが、使ったことはない
2. 知らない

2.26 **あ** (cpa119)

1. 知らない
2. 知らない

2.27 hichon

1. 知っているが、使ったことはない
2. 知っているが、使ったことはない

2.28 akitaka shiogai (sallop)

1. 知らない
2. 知らない

2.29 kazken3

1. 知らない
2. 使ったことがある

2.30 nom4476

1. 知っているが、使ったことはない
2. 知らない

2.31 jsynth21

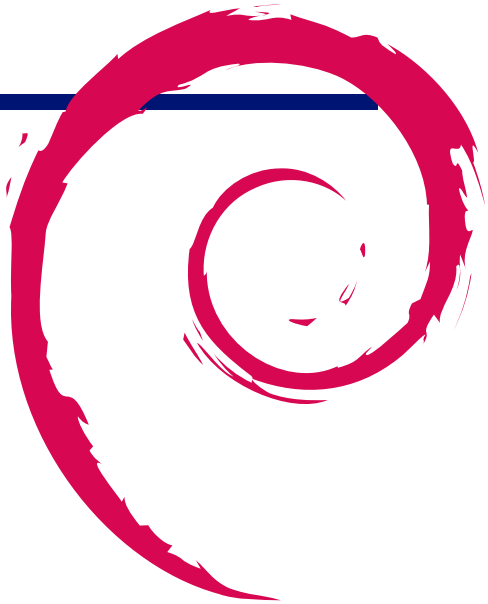
1. 知らない
2. 知らない

2.32 nogajun

1. 知っているが、使ったことはない
2. 使ったことがある

2.33 iwamatsu

1. 知っているが、使ったことはない
2. 知っているが、使ったことはない



3 Jitsi を使ったビデオ会議サーバの作り方

杉本　典充

3.1 はじめに

Debian Project は開発や開発者同士のコミュニケーションをオンライン上で行うことを前提として作業を進めています。ただ、開発者同士のオフラインによる繋がりを定期的に持ってもいいのではないかという考えから東京エリア Debian 勉強会が 2005 年に始まりました。

しかし、2019 年の年末あたりから流行し始めた新型コロナウイルス（COVID-19）の感染拡大を抑えるには人との接触をいかに回避できるかが重要になっています。そのため、直接人と会わずにコミュニケーションをとることができるビデオ会議のニーズが急激に高まってきています。

Debian 勉強会を含めたコンピュータの勉強会は以前から活発に行われていますが集会スタイルで行うことが多かったため、従来のやり方では開催が困難になっている実態があります。

そこで Debian 勉強会をオンラインで開催する手法としてビデオ会議システムをオープンソースで構築できる Jitsi（ジッチー）というサーバアプリケーションを試しているためご紹介します。

3.2 Jitsi について

Jitsi[*1]とは「More secure, more flexible, and completely free video conferencing」と謳うオープンソースなビデオ会議サーバの機能をもつアプリケーションです。Jitsi は ビデオ会議の機能を実現するために WebRTC の技術を利用しており、多くの PC ではグラフィカルな Web ブラウザを使ってビデオ会議を行うことができます。また、モバイル向けに Android 版および iOS 版のアプリケーションも存在します。

Jitsi のソースコードは github[*2]で公開しており、Java・Lua・html・javascript を利用して開発しています。

ビデオ会議システムのセキュリティは大丈夫であるか、という疑問に対して Jitsi はセキュリティとプライバシーについて以下のように説明しています[*3]。

- 会議室は最初の一人がログインしたタイミングで生成し、全員が退出すると会議室は破棄する設計である
  - 全員が退出した後は名前、電子メールアドレス、プロフィール写真は保持されない
  - 全員が退出した後に同名の会議室に入った場合は以前の同名の会議室とは別扱いになり、データを引き継がない
- ビデオ会議のビデオおよびオーディオのストリーミングデータはは DTLS-SRTP[*4]を用いて暗号化した上で通信を行う

[*1] https://jitsi.org/
[*2] https://github.com/jitsi
[*3] https://jitsi.org/blog/security/
[*4] https://tools.ietf.org/html/rfc5763

  - パケットの暗号化及び復号は会議の参加者が 2 人の場合は参加者のデバイス同士が直接処理し、3 人以上の場合は Jitsi Videobridge (JVB) が中継して復号と変換処理を行う方式をとっている
  - ネットワークのトラフィックはメモリのみ存在し、永続ストレージに保存されることはない
- モデレータの権限をもつユーザは、指定したユーザへ「キック（退出)」「ミュート（音量ゼロ)」を実行できる
- 会議室名はできるだけランダムな長い名前にすることが望ましい
  - よくある単語名や名詞を会議室名に使うと常に誰か会議室にいる状態になり、会議室が破棄されなくなるため

3.3 Jitsi サーバの作り方

Jitsi のインストール情報

Jitsi のインストール情報は以下のページで紹介しています。今回はこの手順を参考にサーバを構築しています。

- 公式情報
  - `https://github.com/jitsi/jitsi-meet/blob/master/doc/quick-install.md`
  - `https://github.com/jitsi/docker-jitsi-meet`
    NAT が必要なサーバで動作する設定、クライアント・サーバの通信で利用するポート番号の説明があります
- Debian 10 Buster に Jitsi をインストールした方の情報
  - `https://www.scaleway.com/en/docs/setting-up-jitsi-meet-videoconferencing-on-debian-buster/`

3.3.1 DNS 名の事前決定

Jitsi サーバをグローバル環境、またはイントラネットのどちらで環境を構築する場合でも DNS 名を先に決めておくことをお勧めします。

理由は Jitsi の debian パッケージをインストールする過程で DNS 名を入力するダイアログが表示され、実際に利用する DNS 名を入力しておくとインストール後のファイル名や通信設定の大部分が自動で設定されるためかなり楽ができます。

今回は、DNS 名を ”vc2.pcdennokan.wjg.jp” としてグローバル環境にサーバを構築する前提とします。なお、DNS サーバはダイナミック DNS の サービスである MyDNS.JP[*5]を利用しています。

3.3.2 サーバの準備

Jitsi サーバは以下の条件を満たす性能及び場所に作成するのがよいと考えます。

- RAM は 4GB 以上が望ましい（デフォルト状態での java プロセスの起動オプションに”-Xmx3072m”[*6]を指定しており、Java の Out of Memory エラーが発生しにくい）
- インターネット回線の容量が大きく、できればトラフィックの単価が安いサーバやサービスを使う（ストリーミングを行うためネットワークのトラフィック量がかなり激しい）
- ビデオ会議の同時利用人数が多い場合は CPU コア数が多い方がよい（同時利用人数が多いとロードアベレージが上がる傾向がある）
- ディスク I/O はほとんどないアプリケーションをインストールする分の容量があればよい

[*5] `https://www.mydns.jp/`

[*6] /usr/share/jitsi-videobridge/lib/videobridge.rc に ”# VIDEOBRIDGE_MAX_MEMORY=3072m” というパラメータがあります。

今回は作成する Jitsi サーバは以下の環境で用意しました。

- クラウドサービス
  - Google Compute Engine (GCE)
- サーバのスペック
  - マシンタイプ: N1 ハイ CPU マシンタイプ n1-highcpu-4
  - vCPU: 4 コア
  - RAM: 3.60 GB
  - ネットワーク下り帯域幅: 10 Gbps
  - Disk: 10 GB
  - OS: Debian 10.3 Stretch

GCE で仮想サーバを作成するときに http と https の通信を許可する設定のチェックボックスを有効にしておきます。

3.3.3 ファイアウォール

Jitsi サーバで通信を行うためにファイアウォールを設定して必要な通信を許可する必要があります。
PC やモバイルアプリでビデオ会議を行う場合は以下のポート番号の通信を許可してください。

- 80/tcp
- 443/tcp
- 10000/udp

クライアント PC と Jitsi サーバが通信する構成は以下の URL の図がわかりやすいと思います。
`https://github.com/jitsi/docker-jitsi-meet#architecture`

3.3.4 サーバへのログイン

手元の Debian 上から GCE の仮想サーバを ssh で操作するには、Google Cloud SDK のインストールが必要です。次の URL の手順に従い ”google-cloud-sdk” パッケージをインストールしておいてください。

- `https://cloud.google.com/sdk/docs/downloads-apt-get?hl=ja`

google-cloud-sdk パッケージをインストールして認証情報の設定が完了後、以下のコマンドを実行すると ssh で仮想サーバにログインできます。

```
$ gcloud compute ssh videochat2 --zone asia-northeast1-c --ssh-flag="-p 22"
```

3.3.5 Debian サーバの初期セットアップ

GCE で用意している Debian 10.3 buster のイメージはタイムゾーンは UTC、ロケールは C が初期値になっています。日本語環境で利用するため、設定を変更します。

タイムゾーンの変更

現在のタイムゾーンを確認します。

```
$ date
Sun Apr 12 02:22:51 UTC 2020
```

タイムゾーンを変更するには、dpkg-reconfigure コマンドで設定します。

```
$ sudo dpkg-reconfigure tzdata
  → Asia を選択
  → Tokyo を選択

Current default time zone: 'Asia/Tokyo'
Local time is now:      Sun Apr 12 11:23:20 JST 2020.
Universal Time is now:  Sun Apr 12 02:23:20 UTC 2020.
```

タイムゾーンが変更されたことを確認します。

```
$ date
Sat Apr 18 00:21:56 JST 2020
```

ロケール

日本語のロケールを使用するには locales パッケージをインストールし、dpkg-reconfigure コマンドで設定します。

```
$ sudo apt-get update
$ sudo apt-get install locales
$ sudo dpkg-reconfigure locales
  → Configuring locales
    → ja_JP.UTF-8 を選択
  → Default locale for the system environment
    → ja_JP.UTF-8 を選択
```

サーバの再起動と動作確認

サーバを再起動してタイムゾーンとロケールの変更を全プロセスに反映させます。

```
$ sudo reboot
```

タイムゾーンとロケールの状態を確認します。

```
$ date
2020 年  4 月 18 日 土曜日 00:30:38 JST
```

NTP サーバの確認

GCE では NTP サーバとして chronyd が 標準で動作しています。

```
$ ps ax | grep chrony | grep -v grep
  365 ?        S      0:00 /usr/sbin/chronyd -F -1
  366 ?        S      0:00 /usr/sbin/chronyd -F -1
```

3.3.6 Jitsi のインストール

nginx のインストール

Jitsi サーバとクライアント PC の通信は直接 jitsi-videobridge2 デーモンと行うか、リバースプロキシを経由して jitsi-videobridge2 と通信するかのどちらを選択できます。今回はリバースプロキシを経由することとし、 nginx を利用します。apt-get コマンドで nginx をインストールします。

```
$ sudo apt-get install nginx
$ sudo systemctl start nginx
$ sudo systemctl enable nginx
```

重要なポイントは後述する jitsi-meet パッケージをインストールするより前に nginx をインストールしておくことです。jitsi-meet パッケージはインストール処理で設定ファイルを生成するのですが、 web サーバがインストール済

みの場合はリバースプロキシで動作することを前提とした設定ファイルを自動で生成するようです[*7]。

Jitsi のインストール

Jitsi のインストールで利用するコマンドをインストールます。

```
$ sudo apt-get install wget
```

Jitsi で用意している apt リポジトリを登録します。残念ながら jitsi のパッケージは debian の公式パッケージとして stable 版、unstable 版ともに提供されていません。

```
$ wget -qO - https://download.jitsi.org/jitsi-key.gpg.key | sudo apt-key add -
$ sudo sh -c "echo 'deb https://download.jitsi.org stable/' > /etc/apt/sources.list.d/jitsi-stable.list"
$ sudo apt-get update
```

jitsi-meet パッケージをインストールします。

```
$ sudo apt-get install jitsi-meet
```

jitsi-meet パッケージのインストール時に DNS 名を入力するダイアログが表示されますので、決めておいた DNS 名を入力します。また、SSL 証明書をどうするか質問するダイアログも続けて表示されます。これは「Generate a new self-signed certificate (You will later...」を選択してインストールを行います（後述の手順で Let's Encrypt の SSL 証明書をインストールします)。

jitsi-meet をインストールすると、2020 年 4 月 10 日にリリースされた以下のパッケージがインストールされました。

```
norimitu@videochat2:~$ dpkg -l | grep jitsi
ii  jitsi-meet             2.0.4416-1           all  WebRTC JavaScript video conferences
ii  jitsi-meet-prosody     1.0.3992-1           all  Prosody configuration for Jitsi Meet
ii  jitsi-meet-turnserver  1.0.3992-1           all  Configures coturn to be used with Jitsi Meet
ii  jitsi-meet-web         1.0.3992-1           all  WebRTC JavaScript video conferences
ii  jitsi-meet-web-config  1.0.3992-1           all  Configuration for web serving of Jitsi Meet
ii  jitsi-videobridge2     2.1-169-ga28eb88e-1  all  WebRTC compatible Selective Forwarding Unit (SFU)
```

Let's Encrypt の SSL 証明書のインストール

jitsi-meet パッケージのインストール時に自己署名証明書を生成していますが、ターミナルの出力の中に Let's Encrypt の SSL 証明書を簡単にインストールするスクリプトがあるというメッセージが表示されています。

以下のコマンドを実行して Let's Encrypt の SSL 証明書をインストールします[*8]。

```
$ sudo bash /usr/share/jitsi-meet/scripts/install-letsencrypt-cert.sh
```

ここまでインストール作業を終えて web ブラウザで https://vc2.pcdennokan.wjg.jp/ へ接続すると Jitsi のトップ画面（図 1）を表示できます。

（オプション設定）NAT 環境の後ろで Jitsi サーバを動かす設定

Jitsi サーバを置くネットワーク構成によっては、サーバがグローバル IP アドレスを直接持たず NAT してインターネットに接続する構成の場合があります（GCE や Docker を利用する場合は NAT 構成になります)。その場合は jitsi-videobridge2 デーモンに設定を加える必要があります[*9]。

[*7] https://www.scaleway.com/en/docs/setting-up-jitsi-meet-videoconferencing-on-debian-buster/

[*8] DNS の A レコードが事前に登録されていない場合は Let's Encrypt の SSL 証明書の取得処理がエラーになりますのでご注意ください。

[*9] イントラネット環境やグローバル IP を直接サーバに割り当てるネットワーク構成の場合はこの設定変更は不要です。

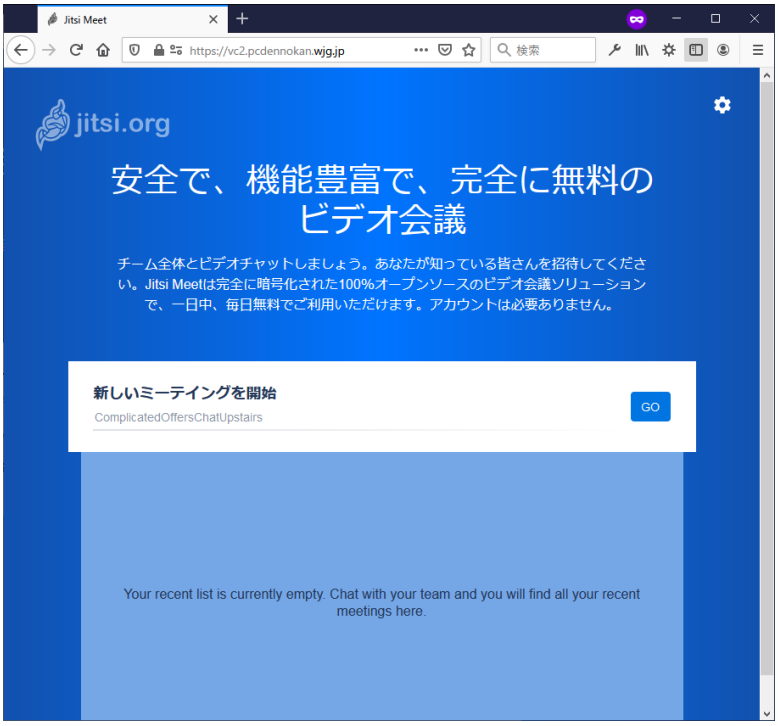


図 1　Jitsi サーバの web のトップ画面

```
$ sudo vi /etc/jitsi/videobridge/sip-communicator.properties
    (snip)
org.ice4j.ice.harvest.NAT_HARVESTER_LOCAL_ADDRESS=10.146.0.6
org.ice4j.ice.harvest.NAT_HARVESTER_PUBLIC_ADDRESS=34.84.236.29
    (snip)
```

jitsi-videobridge2 デーモンを再起動して設定を反映します。

```
$ sudo systemctl restart jitsi-videobridge2
```

NAT 設定の説明は以下 URL に情報があります。

`https://github.com/jitsi/jitsi-meet/blob/master/doc/quick-install.md#advanced-configuration`

3.3.7　ビデオ会議を行う

ここまで設定してようやく Jitsi を使ってビデオ会議ができるようになりました。

以下の URL へアクセスし、画面中央の「新しいミーティングを開始」に会議室名を入力するとビデオ会議の URL へジャンプして会議室ができます。会議に参加する方にはその URL を連絡し、Web ブラウザまたはモバイルアプリで URL を開くことで会議に参加することができます。

- https://vc2.pcdennokan.wjg.jp/

3.4　Jitsi のカスタマイズについて

3.4.1　設定ファイルの場所

Jitsi の設定ファイルは ”/etc/jitsi” 配下及び ”/usr/share/jitsi-*” 配下のディレクトリにあります。主な設定ファイルを以下に示します。

- /etc/jitsi/meet/vc2.pcdennokan.wjg.jp-config.js
  （インストール時に入力した DNS 名を含むファイル名が生成されます）

- /usr/share/jitsi-meet/interface_config.js
- /usr/share/jitsi-meet/libs/external_api.min.js

3.4.2 最近の会議室一覧の非表示

Jitsi のトップ画面を開くと最近作成された会議室の一覧が表示されます。ただ、会議室の存在を知る人を制限するには誰でも会議室の URL がわかる状態になっているのは好ましくありません。

/usr/share/jitsi-meet/interface_config.js の ”RECENT_LIST_ENABLED” パラメータを変更すると Jitsi の最近の会議室一覧の表示をしないようにできます。

```
# vi /usr/share/jitsi-meet/interface_config.js
  (snip)

  RECENT_LIST_ENABLED: false,

  (snip)
```

3.4.3 ストリーミングのビデオの画質調整

Jitsi のビデオ会議の画質はサーバ側で HD（720p、1280x720px）がデフォルトに設定されています[*10]。大変きれいなのですが、パケットの転送量が多くなるため通信帯域が細い場合に遅延が起こる、クライアント PC と Jitsi サーバの負荷が高くなる、Jitsi サーバのトラフィックにかかる料金が気になる、などストリーミングならではの悩みが出てきます。

config.js の設定変更を行うとビデオ会議の画質を変更することができます。”resolution” 値はクライアント PC の Web カメラの解像度の設定値であり、”constraints” のオブジェクトの値はクライアント PC が視聴する他の参加者のビデオの画質の設定値です。

以下の config.js はクライアント PC の Web カメラの解像度の画質を「480p」とし、クライアント PC が視聴する他の参加者のビデオの画質を「480p〜240p」とする設定例です。

```
# vi /etc/jitsi/meet/vc2.pcdennokan.wjg.jp-config.js
    (snip)

    // Sets the preferred resolution (height) for local video. Defaults to 720.
    resolution: 480,

    // w3c spec-compliant video constraints to use for video capture. Currently
    // used by browsers that return true from lib-jitsi-meet's
    // util#browser#usesNewGumFlow. The constraints are independent from
    // this config's resolution value. Defaults to requesting an ideal
    // resolution of 720p.
    constraints: {
        video: {
            height: {
                ideal: 480,
                max: 480,
                min: 240
            }
        }
    },

    (snip)
```

なお config.js は https://vc2.pcdennokan.wjg.jp/config.js の URL でクライアント PC へ配信されます。

3.5 2020 年 3 月 Debian 勉強会の Jitsi サーバのリソース使用量

3.5.1 サーバ環境

2020 年 3 月に開催した Debian 勉強会の後半は Jitsi サーバを試験的に利用して 15 名で BoF を 1 時間程度行いました。その時のサーバ環境、サーバリソースの使用量を公開します。

Jitsi サーバは以下の環境で用意していました。

[*10] ノート PC に搭載している Web カメラのほとんどは HD（720p、1280x720px）の品質のものが多いです。

- クラウドサービス
  - Google Compute Engine (GCE)
- サーバのスペック
  - マシンタイプ: N1 標準マシンタイプ n1-standard-1
  - vCPU: 1 コア
  - RAM: 3.75 GB
  - ネットワーク下り帯域幅: 2 Gbps
  - Disk: 10 GB
  - OS: Debian 9 Stretch[*11]

Jitsi アプリケーションのバージョンは以下でした（2020 年 4 月 18 日現在の最新版より少し古いバージョンです）。

```
$ dpkg -l | grep jitsi
ii  jitsi-meet             1.0.4101-1  all    WebRTC JavaScript video conferences
ii  jitsi-meet-prosody     1.0.3729-1  all    Prosody configuration for Jitsi Meet
ii  jitsi-meet-web         1.0.3729-1  all    WebRTC JavaScript video conferences
ii  jitsi-meet-web-config  1.0.3729-1  all    Configuration for web serving of Jitsi Meet
ii  jitsi-videobridge      1126-1      amd64  WebRTC compatible Selective Forwarding Unit (SFU)
```

3.5.2 CPU 利用率

サーバの CPU 利用率のグラフを 図 2 に示します。ピーク時で 86% の使用率でした。CPU コア数は 1 コアの割り当てのため、意外と処理できていた感じがあります。

なお、当日サーバ上で top コマンドの表示を確認していたときはのロードアベレージが 6 程度だった記憶があります。

図 2　Jitsi サーバの CPU 利用率のグラフ（vCPU 割り当て数：1）

3.5.3 受信トラフィック

サーバの受信（上り）トラフィックのグラフを 図 3 に示します。ピーク時で 1.653 MiB/s（= 13.22 Mbps） の受信トラフィックがあり、15 台のクライアント PC から受信したストリーミングデータの合計値となります。ただ、参加者は利用するクライアント PC の Web カメラを無効にしていた方が多く音声データのみを送信している人が多かったと推測しています。

[*11] Web の情報では Debian 9 のインストール情報が多く見つかり、時間的制約から Debian 9 を選んだ経緯があります。

図 3　Jitsi サーバの受信トラフィックのグラフ

3.5.4　送信トラフィック

サーバの送信（下り）トラフィックのグラフを 図 4 示します。ピーク時で 6.828 MiB/s（= 54.62 Mbps） の送信トラフィックがあり、15 台のクライアント PC へ送信したストリーミングデータの合計値となります。会議の参加者の目線で見ると、参加者が利用するクライアント PC の Web カメラを無効にしていた方が多かった状態でピーク時に一人あたり 3.66 Mbps の受信トラフィックが出ていた計算となります。

仮に会議中に 55 Mbps の送信トラフィックが常に出ていると仮定するした場合、1 GB の送信トラフィックを 2 分 30 秒で消費します。120 分の勉強会で利用した場合は約 48 GB の送信トラフィックが発生する計算になります。

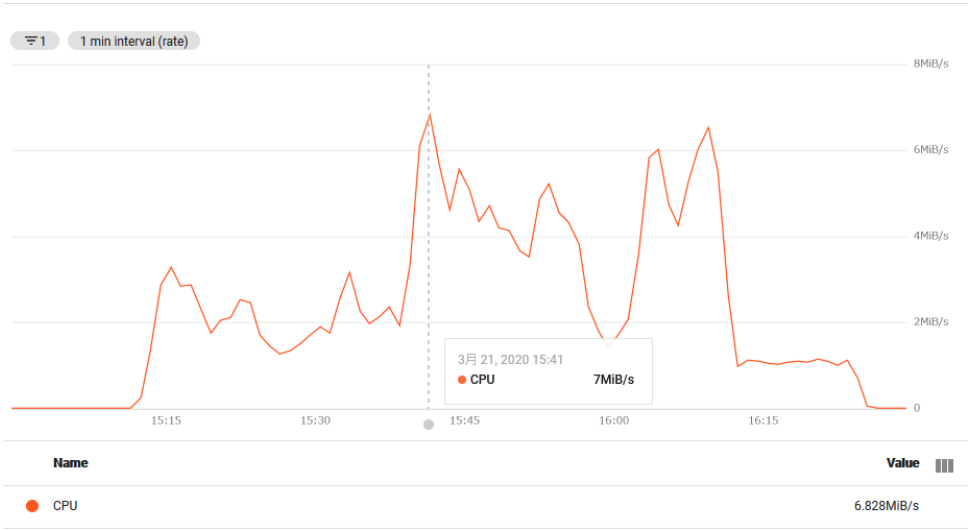
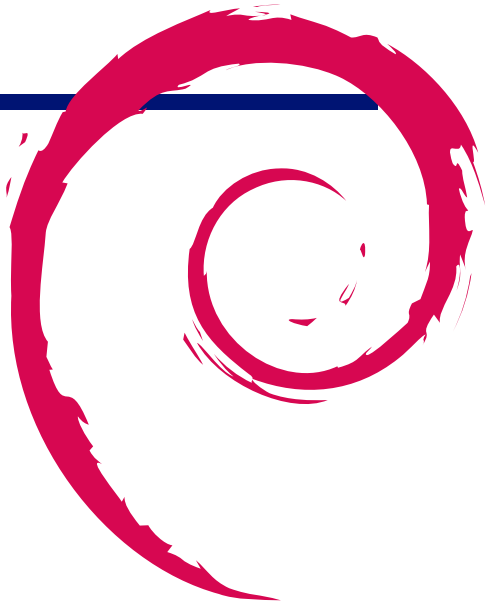


図 4　Jitsi サーバの送信トラフィックのグラフ

3.5.5　おわりに

Jitsi サーバを実際に構築してオンラインのビデオ会議で使ってみました。Jitsi サーバの快適な動作に必要なサーバ性能やネットワーク帯域はクライアント PC のカメラの解像度やフレームレートによっても変わってきます。自分好みの設定に調整できること、イントラネットに自分でビデオ会議サーバを作れること、そしてソースコードの書き換えもできること、これがオープンソースの醍醐味ですのでぜひとも Jitsi を利用してみてください。



4 メモ

Debian 勉強会資料

2020 年 4 月 18 日　　初版第 1 刷発行

東京エリア Debian 勉強会 （編集・印刷・発行）