How LPC went virtual

Jonathan Corbet
LWN.net
corbet@lwn.net
This is a discussion session!

Please break in with questions, comments, complaints... anytime!
LPC 2020 - Overview

August 25-27, Halifax, Nova Scotia, Canada

The Linux Plumbers Conference for 2020 has been set for August 25-27 at the Halifax Marriott Harbourfront Hotel.

Note that Halifax, NS is in the AST time zone (UTC -4). In August, Linux Plumbers Conference will take place in ADT time zone (UTC -3).

Sponsorship opportunities

Linux Plumbers Conference would not be possible without our sponsors. Many thanks to all the great organizations that have supported Linux Plumbers Conference over the years.

New sponsorship opportunities are available for 2020! We hope that your organization will consider joining our amazing growing list of sponsors this year. Find out more here.

Social Media

Please follow us on Twitter for any last minute updates and to stay connected: @LinuxPlumbers
Oh .... s*** ...
Problem 1: facing the issue
Will people come?
Next problem:
getting out of the existing contracts
Cancel or go online?
How would we do it online?

Proprietary platforms?

...or go for free software?
Video conferencing with Jitsi

By Jonathan Corbet  
March 24, 2020

Spring is coming to the northern hemisphere, and one's thoughts naturally turn to being locked up inside the house and not allowed to go anywhere. That has, in turn, led to an increasing interest in alternative mechanisms for keeping up with family and coworkers, especially video conferencing. There are a number of proprietary video-conferencing services out there; your editor decided to look into what solutions exist in the free-software realm. It turns out that there are a few; the first to be looked at is Jitsi.

Jitsi is, in fact, a collection of components, written mostly in Java (and JavaScript) and released under the Apache license. At the core is Jitsi Videobridge, which implements multi-partyicipant video conferences, and Jitsi Meet, which implements the client side. Various other components live under the hood and are likely to only come to one’s attention if something goes wrong with them. There is also a Jitsi Desktop application, but that has been superseded by the browser interface and is considered “legacy” at this point.

Getting it going

Brave folks can certainly install the whole thing from source; for the rest of us, pre-built packages are limited to the Debian and Ubuntu distributions. Your editor installed that version on a Debian 10 machine sitting out there in the cloud somewhere.

That installation did not go as easily as one might have liked. The (sparse) documentation suggests that the packages will perform the needed web-server setup, but a number of details were left as exercises for the reader. There is support for automating the setup of a Let’s Encrypt certificate, but your editor did not try that. Once the server configuration was properly tweaked, it was possible to connect to the newly established bridge with an ordinary Firefox browser.

Running a single connection to a Jitsi server is a great way to admire one’s own image on the screen, but it lacks something from the full conferencing experience. Unfortunately, when the second user connected, both users were immediately disconnected with a helpful “something went wrong” error, and an even more helpful “error 143” message on the server side. Some searching revealed this useful post stating that the Debian package installs the server in a misconfigured manner; once the suggested change was made, everything worked as expected.

The user experience

The Jitsi client runs fine in a standard web browser, with no need to install any plugins. In the default setup, a new user connecting to the server will see a screen inviting them to create a new meeting, typing any string into the field provides a “room” with that name (or joins a room with that name if it already exists). The curious can see this screen in action on the demo site set up to allow anybody to run a free conference.

After one joins a conference, the behavior is much like the proprietary services out there. It is possible to see a tiled view with all participants, or just the person who is speaking at any given time. There are buttons to mute audio and/or video, a separate area for text chat, and a button to “raise your hand” for attention. One cute feature is the ability to blur the background of one’s outgoing video, though the result is somewhat ethereal.

Video and audio quality both seem to be quite good, though video suffers somewhat when there is a low-bandwidth connection involved. Jitsi claims that all data is encrypted between clients and the video bridge, though it goes through the bridge itself in the clear.

There is a screen-sharing option that can transmit the contents of a single window or the screen as a whole — though the latter leads to amusing effects if the conference itself is on-screen. The web
Video conferencing with Jitsi

By Jonathan Corbet
March 24, 2020

Spring is coming to the northern hemisphere, and one’s thoughts naturally turn to being locked up inside the house and not allowed to go anywhere. That has led to an increasing interest in alternative mechanisms for keeping in touch with fellow coworkers, especially video conferencing. There are a number of proprietary conferencing services out there; your editor decided to look into what solutions exist in the free-software world that turn out there are the first to be looked at is Jitsi.

Jitsi is, in fact a collection of components, written mostly in Java (and JavaScript) and released under the GPL license. At the core is Jitsi VideoBridge, which implements multi-participant video conferences, and Jitsi UI implements the client side. Various other components live under the hood and are likely to only come to your attention if something goes wrong with them. There is also a Jitsi Desktop application, that has been extended for archiving user conferences.

Getting it going

Bravo folks can install all the thing from source, for the rest of us, pre-built packages are linked on Debian and Ubuntu distributions. Your editor installed that version on a Debian 10 machine sitting out the cloud somewhere.

That installation did not go as easily as one might have liked. The (sparse) documentation suggests that the packages will perform the needed web-server setup, but a number of details were left as exercises for the reader. There is support for automating the setup of a Let’s Encrypt certificate, but your editor did not try that. Once server configuration was properly tweaked, it was possible to connect to the newly established bridge with ordinary Firefox browsers.

Running a single connection to a Jitsi server is a great way to admire one’s own image on the screen, but something from the full conference experience. Unfortunately, when the second user connected, both user immediately disconnected with a helpful “something went wrong” error, and an even more helpful “error message on the server board. Some searching revealed this useful post stating that the Debian package itself a misconfigured manner; once the suggested change was made, everything worked as expected.

The user experience

The Jitsi client runs fine in a standard web browser, with no need to install any plugins. In the default setup, selecting a video stream to view in a new window, typing any string in the address field provided creates a “room” with that name (or joins a room with that name if it already exists). The client screen in action on the demo site set up to allow anyone to run a free conference is shown.

After one joins a conference, the behavior is much like the proprietary services out there. It is possible to view with all participants, or just the person who is speaking at any given time. There are buttons to mute and/or video, a separate area for text chat, and a button to “raise your hand” for attention. One cute feature is the ability to blur the background of one’s outgoing video, though the result is somewhat ethereal.

Video and audio quality both seem to be quite good, though video suffers somewhat when there is a low-bandwidth connection involved. Jitsi claims that all data is encrypted between clients and the video bridge, though it goes through the bridge itself in the clear.

There is a screen-sharing option that can transmit the contents of a single window or the screen as a whole — although the latter leads to amusing effects if the conference itself is on-screen. The web

Video conferencing with BigBlueButton

By Jonathan Corbet
April 10, 2020

While social distancing often comes naturally to free-software developers, there are still times when we wish to talk to each other. In the absence of community conferences, the next-best alternative is often video conferencing. While video conferencing tends to be held using centralized, proprietary systems, there are free alternatives as well. LWN recently looked at Jitsi but this effort did not stop there; next on the list is BigBlueButton, a system that is oriented toward the needs of online educators but is applicable beyond that use case. BigBlueButton is not a new project; it has been under development since 2007. That history shows in a number of ways; for example, the actual conferencing component was originally implemented in Flash and has only recently been supplemented by an HTML5/WebRTC-based solution. The code is licensed under the Lesser GPL, the web site contains a copy which version 3 or later. The code itself is a massive collection of Java, Scala, and JavaScript (at least) code — almost 1,800 directories worth.

Installing BigBlueButton

The heavy weightage of BigBlueButton shows in the installation instructions, which go on for some length. There is also a script to do all of that work, along with a suggestion to use the time-honored "wget | bash" execution method. Your editor used the script, but only after eyeballing it (thus guaranteeing that it is secure) and making a change as described below.

The system’s server requirements are quite specific; in particular, only the Ubuntu 16.04 release is supported. BigBlueButton wants at least 8GB of memory ("with swap enabled") and four CPUs. The installation script will refuse to run on any other distribution or if the system has less than 4GB of installed memory. Your editor capitulated on Ubuntu 16.04 but, being of a generally parsimonious nature, tweaked the script to make it accept the low-end 2GB virtual server used to test the system.

The Ubuntu 16.04 requirement is justified in the name of stability, and one can easily imagine that the developers wouldn’t want to support this towering stack of software on multiple platforms. But that is an ancient distribution at this point, and it is losing support in one year: there is going to have to be a mass migration of BigBlueButton systems over the next twelve months.

The installation script takes about a half hour to get its job done. This work includes setting up a number of third-party repositories for needed components; the installation instructions correctly note that "the default version of ffmpeg in Ubuntu 16.04 is old", for example, so the script obtains a more current version from elsewhere. By the end of the installation, not only is the software installed, but the script hands one a TLS certificate for the server from Let’s Encrypt. Those trying this at home should note that BigBlueButton can take several minutes to actually get started after being launched; they should not assume quickly that the installation has failed.

While BigBlueButton handles the conferencing tasks, it does not concern itself with front-end tasks like managing "rooms" or authenticating users. There is, instead, a reasonably well-documented API that is intended to be used by the front end. Given its roots in the educational community, it is not entirely surprising that applications like Moodle use this API to integrate with BigBlueButton. For those wanting to run a standalone system, there is a front end called Greenlight that can optionally be installed with the BigBlueButton installation script.

Greenlight, as it turns out, is a Ruby-on-Rails application, adding nicely to the menagerie of languages running on the server. And, while the installation script loads BigBlueButton directly onto the server, Greenlight gets installed as a Docker container. That makes management a little more interesting; after some digging, it turned out that the way to create an initial administrative account on the server is a command like this:

docker exec greenlight-v2 script execute rake \ user:create["root", "bb-admin@lwn.net", "password", "admin"]

Running this command promptly throw the server into an existential out-of-memory crisis, making it clear that the project was written when it was all 4GB is necessary — even before a single video conference is established. Once this

null
Decision: BigBlueButton
The worries begin

“We recommend no single sessions exceed one hundred (100) users”

https://docs.bigbluebutton.org/support/faq.html
Dealing with scalability issues

Talk to consultants (useless)
Two “town hall” events
Host the OSPM conference
Dealing with scalability issues

Talk to consultants (useless)
Two “town hall” tests
Host the OSPM conference

Eliminate plenary events
Add the YouTube streams
Encourage listen-only mode
Use massive servers
Footguns
Complexity
The front end

The default front-end is Greenlight
Not a great conference experience

So...
Write our own, how hard can it be?
Welcome Jonathan Corbet!

This is the entry portal for the Linux Plumbers Conference 2020 meeting rooms. Look below for the list of what is happening currently.

The timetable below is focused on current (or near-future) sessions. Times are given in both UTC and our best guess as to your local time (which appears to offset by UTC by ~360 minutes). Click on the "join" button that will appear for any sessions that are currently ongoing.

The full schedule is always available on the main Linux Plumbers Conference site.

Thank you for participating in LPC 2020; please do not hesitate to contact us at contact@linuxplumbersconf.org should you have any difficulties, questions, or comments.

<table>
<thead>
<tr>
<th>Time</th>
<th>Session Description</th>
<th>Location</th>
<th>Speakers</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:39</td>
<td>Build strategies</td>
<td>Android MC</td>
<td></td>
</tr>
<tr>
<td>09:44</td>
<td>Build strategies</td>
<td>Android MC</td>
<td></td>
</tr>
<tr>
<td>09:59</td>
<td>Build strategies</td>
<td>Android MC</td>
<td></td>
</tr>
<tr>
<td>10:09</td>
<td>Build strategies</td>
<td>Android MC</td>
<td></td>
</tr>
<tr>
<td>10:14</td>
<td>Build strategies</td>
<td>Android MC</td>
<td></td>
</tr>
<tr>
<td>10:29</td>
<td>Build strategies</td>
<td>Android MC</td>
<td></td>
</tr>
<tr>
<td>10:44</td>
<td>Build strategies</td>
<td>Android MC</td>
<td></td>
</tr>
<tr>
<td>10:59</td>
<td>Build strategies</td>
<td>Android MC</td>
<td></td>
</tr>
</tbody>
</table>
Version 0.01 released today

git://git.lwn.net/lpcfe.git
Are we creating yet another damn video call?
Preserving the LPC experience

Room protocols for discussions
Hackrooms
The RocketChat system
Other questions

When do the sessions run?
  How long?
  Time zones?

Conclusions:
  Half days
  Optimize for Americas / Europe
Other questions

How many people do we let in?
 Ended up ~950!
Do we charge for entry?

Yes, $50

Help ensure that registrants are serious
If we had to do this again

Gain some deep-down BBB expertise
If we had to do this again

Gain some deep-down BBB expertise
Worry less about number of attendees
If we had to do this again

Gain some deep-down BBB expertise
Worry less about number of attendees
Find a way to spread the schedule pain
If we had to do this again

Gain some deep-down BBB expertise
Worry less about number of attendees
Find a way to spread the schedule pain
Don’t go nuts with the server sizing
If we had to do this again

Gain some deep-down BBB expertise
Worry less about number of attendees
Find a way to spread the schedule pain
Don’t go nuts with the server sizing
Better integrate BBB and chat
If we had to do this again

Gain some deep-down BBB expertise
Worry less about number of attendees
Find a way to spread the schedule pain
Don’t go nuts with the server sizing
Better integrate BBB and chat
Stick with free-software solutions!
What else?
Thanks to the committee

Laura Abbott
Elena Zannoni
Kate Stewart
James Bottomley
Christian Brauner
Jonathan Corbet
Guy Lunardi
Paul McKenney
Ted Ts’o
Steve Rostedt
David Woodhouse
Eternal LPC?

We have this infrastructure and know-how

Set up a monthly LPC day
   Microconfs for groups that need one