ELECTRONICS CLUB

Local Video Transmission and Face Detection

How are we planning on doing it ?

Black Box

Output

What’s in the Black Box ?

The Black Box Can we Any of the Following depending on the use case :

- A Cloud Computer
  - When you need to Feed the Video Through WAN
- A Router
  - Video Through LAN
- Ethernet Cable
  - PAN (Personal Area Network)

Let’s Setup Our RPi in the Meantime :

https://github.com/aswinkumar1999/RPi_Video_Stream

Software Tools we will be using :

- Motion (Dependencies : lsusb)
  - ` sudo apt install motion lsusb `
- Imagezmq (Dependencies : pyzmq , opencv , imutils , ffmpeg)
  - Installing Opencv :
    - Installing ffmpeg
  - ` pip install pyzmq imutils `
  - ` git clone https://github.com/jeffbass/imagezmq.git `
- NodeJS (Dependencies : OpenCV , ffmpeg)

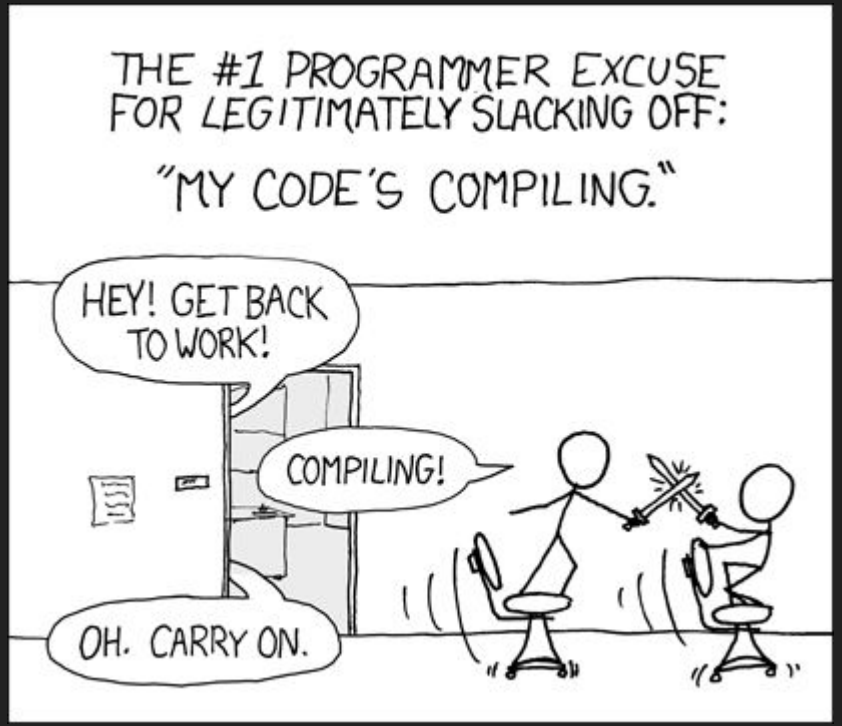
Building from Source

Building all the software you will be using today from source would take a very long time ~4 hours , so we have compiled it for RPi

The key thing is that it's an automated process which transforms "source" files edited by humans into some other files which are actually useful.

Advantages of installing from source:

- You can install the latest version and can always stay updated, whether it be a security patch or a new feature.
- Allows you to trim down the features while installing so as to suit your needs.
- Similarly you can add some features which may not be provided in the binary.
- Install it in a location you wish.
- In case of some software you may provide your hardware specific info for a suitable installation.



OpenCV

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision.[1] Originally developed by Intel, it was later supported by Willow Garage then Itseez (which was later acquired by Intel[2]). The library is cross-platform and free for use under the open-source BSD license.

OpenCV supports the deep learning frameworks TensorFlow, Torch/PyTorch and Caffe.

FFmpeg

FFmpeg is a free and open-source project consisting of a vast software suite of libraries and programs for handling video, audio, and other multimedia files and streams. At its core is the FFmpeg program itself, designed for command-line-based processing of video and audio files, and widely used for format transcoding, basic editing (trimming and concatenation), video scaling, video post-production effects, and standards compliance

Why and What is Video Compression ?

Let’s Assume we have a Full HD 1 minute Video Clip , and Calculate the Storage Space needed to Save it :

1920 * 1080 * 3 * 60 * 24 = Width * Height * Channels (RGB) * Seconds * FPS

1 Pixel Correspond to 1 byte of data storage.

Total = 8957952000 bytes ~ 8.34Gb

We All know A One Minute Recording in Our Phones do not consume 8.34 Gb , So It is Compressed and Stored in our Phones and Computers.

How ? : A Simple Video Compression Algorithm

Let’s Now Assume we have a Full HD 1 minute Video Clip is of a person standing in a place and talking , and let’s assume we do not move and i am only part of 10% of the screen , let us assume the 90% of the screen is constant through the video.

Doing the Math Again :

1920*1080*3*0.9 = ~5.3Mb

1920 * 1080 * 3 * 60 * 24 *0.1 = Width * Height * Channels (RGB) * Seconds * FPS

~ 834Mb + 5.3Mb = ~ 840mb

Some More Probably ?

Now as I am standing , I would probably move a bit to the right / left , let’s say that happens once every second.

Doing the Math :

1920*1080*3*0.9 = ~5.3Mb

1920 * 1080 * 3 * 60 *0.1 = Width * Height * Channels (RGB) * Seconds * FPS

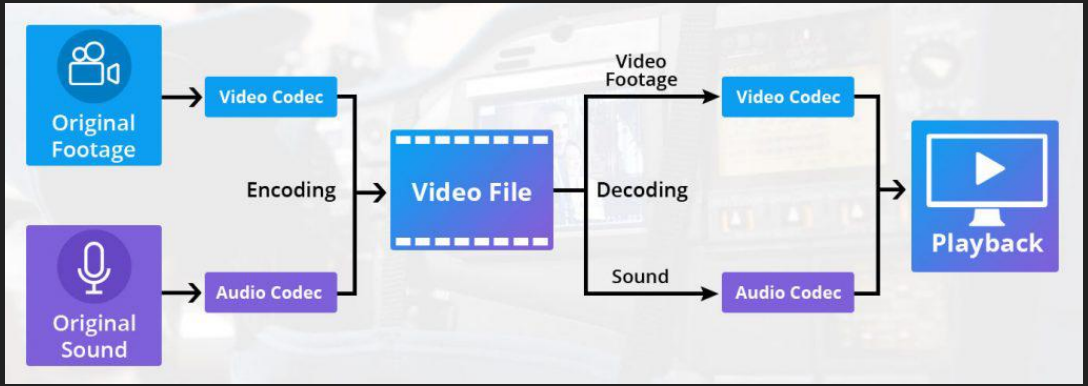
~ 34Mb + 5.3Mb = ~ 40mb

Conclusion

That was a Very Simple Analogy for Video Compression where the Actual Methods are very Math Intensive and is outside the scope of this Course.

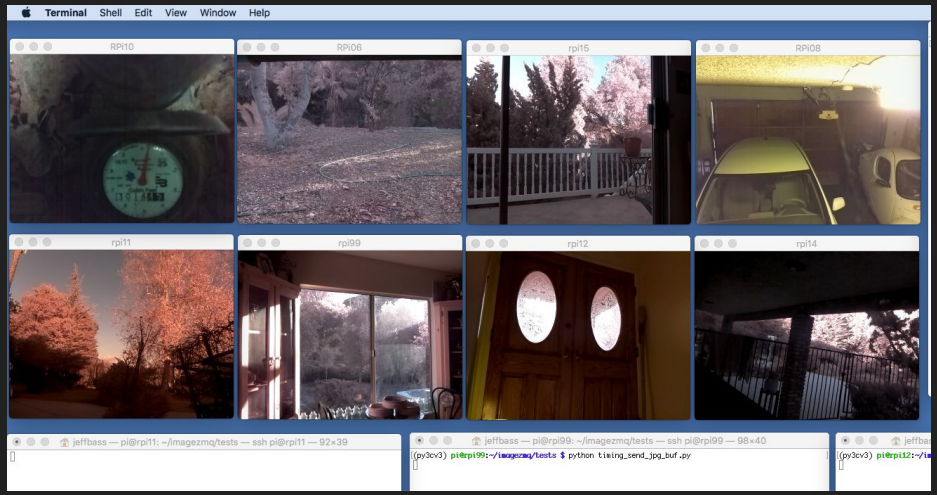
But this is how Youtube Provides you Videos even if you are stuck with a Bad Internet Connection.

Here We will be using H.264 Codec for Video Encoding and Decoding



ImageZMQ

imagezmq is a set of Python classes that transport OpenCV images from one computer to another using PyZMQ messaging. For example, here is a screen on a Mac computer showing simultaneous video streams from 8 Raspberry Pi cameras:



Screenshot of a Computer Displaying Video Feeds from 8 RPi's simultaneously

ZMQ

ZeroMQ (also known as ØMQ, 0MQ, or zmq) looks like an embeddable networking library but acts like a concurrency framework. It gives you sockets that carry atomic messages across various transports like in-process, inter-process, TCP, and multicast. You can connect sockets N-to-N with patterns like fan-out, pub-sub, task distribution, and request-reply. It's fast enough to be the fabric for clustered products. Its asynchronous I/O model gives you scalable multicore applications, built as asynchronous message-processing tasks. It has a score of language APIs and runs on most operating systems.

What is Distributed Computing ?

DECEMBER 2, 2010 WEBLOG

US Air Force connects 1,760 PlayStation 3's to build supercomputer

by Lisa Zyga , Phys.org

The Condor Cluster consists of 1,760 Sony PlayStation 3's, and is the US Department of Defense's fast…

(Phys.org)—About the 33rd largest supercomputer in the world right now is the US Air Force Research Laboratory's (AFRL) newest system, which has a core made of 1,760 Sony PlayStation 3 (PS3) consoles. In addition to its large capacity, the so-called "Condor Cluster" is capable of performing 500 trillion floating point operations per second (TFLOPS), making it the fastest interactive computer in the entire US Defense Department.

The supercomputer, which is located in Rome, New York, was formally presented yesterday at a ribbon-cutting ceremony. It will be used by Air Force centers across the country for tasks such as radar enhancement, pattern recognition, satellite imagery processing, and artificial intelligence research.

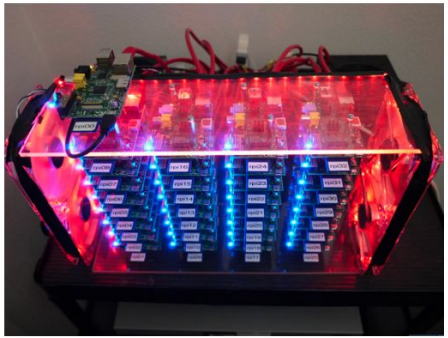
Its speed allows it to analyze ultra-high-resolution images very quickly - at a rate of billions of pixels per minute - to greatly reduce the amount of time required. Due in part to the video game consoles' cutting-edge graphics capabilities, the supercomputer also has improved algorithms that can better identify blurred flying

Build your own supercomputer out of Raspberry Pi boards

Who says you need a few million bucks to build a supercomputer? Joshua Kiepert put together a Linux-powered Beowulf cluster with Raspberry Pi computers for less than $2,000.

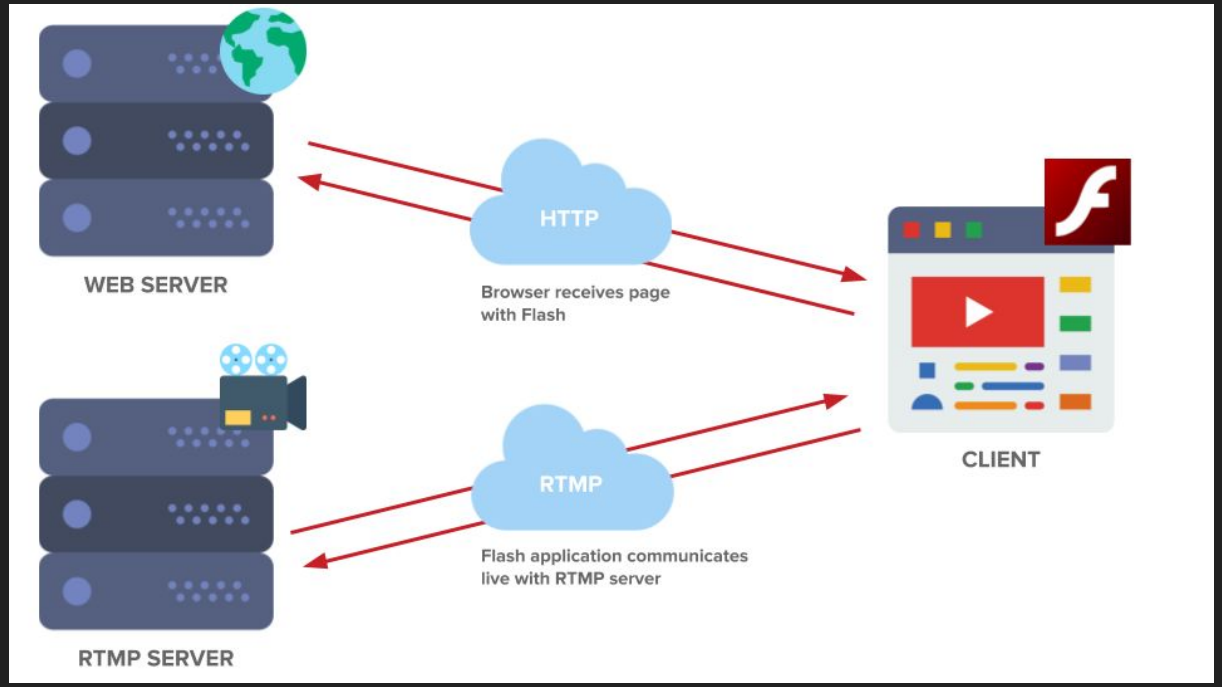
By Steven J. Vaughan-Nichols for Linux and Open Source | May 23, 2013 -- 21:43 GMT (0313 IST) | Topic: SMBs

When you think do-it-yourself (DIY) computing , you probably think of setting up a screaming gaming computer or putting together the best possible components for the least amount of money. You're almost certainly not considering putting together a supercomputer. Maybe you should. Joshua Kiepert, a doctoral student at Boise State's Electrical and Computer Engineering department, has managed to create a mini-supercomputer using Raspberry Pi (RPi) computers for less than $2,000.



Say hello to a homebrew Raspberry Pi-based supercomputer.

RTMP Stream


WEB SERVER
HTTP
Browser receives page
with Flash
CLIENT
RTMP
Flash application communicates
live with RTMP server
RTMP SERVER