Edge Video Analytics for Facial Analysis using HPE Edgeline Systems

Getting Started with IDOL Media Server for Facial Detection and Recognition on Edgeline IoT Systems

Authors:

Bill Walton (bill.walton@hpe.com)

Tripathy (aalap.tripathy@hpe.com)

Nathan Lin (jih-tsen.nat.lin@hpe.com)

Technical white paper

Contents

Executive Summary

When it comes to data analytics in the *Internet of Things* (IoT) space, Edge computing has the edge over the traditional cloud computing at the centralized nodes. Edge computing pushes the computing away from the central server and enables data analytics at the source of the data. This reduces data volume moved across the network, minimizes data transmission cost, and enables faster response time for control.

HPE's Edgeline IoT systems, along with *HPE's Intelligent Data Operating Layer* (IDOL) Media Server software, provides the video analytic capability at the IoT Edge. The IDOL Media Server is capable of a wide variety of rich media analytics, such as facial analysis, image classification, optical character recognition, number plate recognition, scene analysis/object detection, etc. Utilizing off-the-shelf camera components, HPE Edgeline servers and appropriate vendor software can create value added solutions for a wide variety of industrial and retail intelligent Edge environments.

This document highlights the step-by-step procedure to get started with IDOL Media server on HPE Edgeline servers, setting up facial analysis workload, the performance analysis of facial analysis algorithms on Edgeline systems v/s a datacenter class system. Two separate follow-on whitepapers build on the basic setup here to cover scene analysis/pedestrian detection and automatic number plate recognition.

Target audience: Solution Architects, HP Technical Presales, Customers.

Document purpose: The purpose of this document is to describe a recommended architecture/solution for video analytics at the Edge, demonstrating a use case of face detection on ELX0 Gateways and IDOL Media Server to technical and non-technical audiences.

NOTE: ELX0 represents the Edgeline series EL10, EL20, EL1000, and EL4000. Refer to section **Analysis and Recommendations** in this document for more performance overview comparison.

Solution Overview

A use case of face detection and face recognition was built utilizing HPE's Edgeline ELX0 IoT Systems and HPE IDOL Media Server ingesting a live video feed from an off-the-shelf security camera. A broader introduction to Edge Video Analytics and several possible use cases is presented in http://www.hpe.com/support/EL-EVA. Although EL20 was used for initial development of this solution (Figure 1), it was extended to the HPE Edgeline EL10, EL20, EL1000 and EL4000 systems for the same use cases and performance analysis (

Figure 2).

As a person walks in front of the camera, HPE IDOL Media Server software running on the Edge device detects his or her face by its distinctive features. The Edge device is configurable so each facial detection can trigger generation of metadata and image file for further processing or to be saved for a record. After training known face images into the Media Server face database, the Media Server is able to distinguish between faces and can respond with an identifier. In this use case, the default setting was used for the Media Server web page displaying the person's name on the screen. This use case demonstrates that complicated image analytics can be realized with HPE's hardware and software solution at the IoT Edge. This technical white paper outlines installation and configuration of IDOL Media server on these systems, interfacing a camera to the system, and concludes with performance analysis studies performed on all HPE Edgeline systems.

Figure 1. Connecting PoE camera, HPE Edgeline EL20 for face detection/recognition analytics.

Figure **2** describes a variation of this solution viewed from the 4-stage generalized IoT architecture. Power over Ethernet (PoE) cameras constitute Stage 1 sensor devices. HPE Edgeline IoT systems HPE EL10, 20 can interface to these devices and perform limited video processing before pushing the results of the analysis over 3G/4G or Wifi to a cloud.

The converged Edge systems HPE EL1000, EL4000 can interface to a much larger number of PoE cameras (via PCIe NIC cards or an external Aruba PoE switch). This enables more rigorous and CPU intensive video analytics in near real time for a wide variety of use cases including face recognition/detection, license plate recognition, vehicle recognition, scene and object analysis, etc.

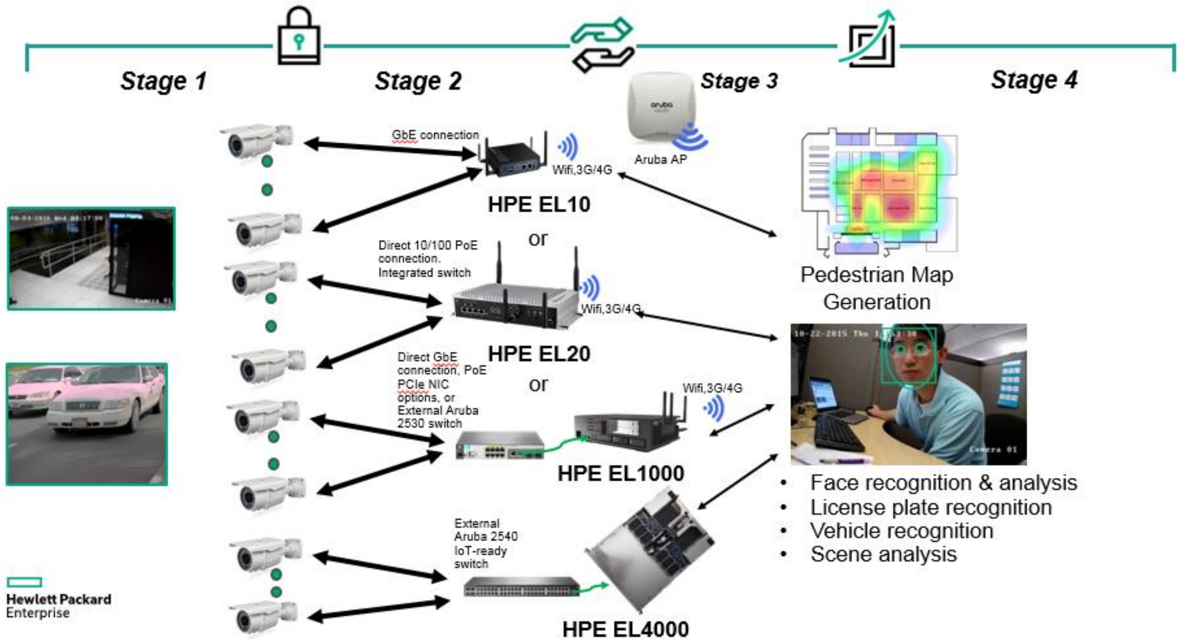


Figure 2. Connecting PoE camera, HPE Edgeline EL20 for face detection/recognition analytics.

Solution Components

Hardware

This solution utilizes the HPE Edgeline family of products as the foundation for an Edge video analytics solution. The need to quickly analyze data and make critical decisions based on real-time video data accentuates the need for Edge computing. HPE has created unique systems that are purpose-built for converging real-time data acquisition, enterprise-class computing, and remote manageability. HPE Edgeline systems are energy-efficient, ruggedized platforms with a broad range of network connectivity and data acquisition options to accommodate even the most complex industrial application. Chassis type, number of servers, number of CPU cores, memory, and storage can all be tailored to site requirements.

- **HPE EL10:** The HPE Edgeline EL10 IoT System is an entry level price/performance optimized ruggedized compute solution for very light weight data aggregation, acquisition and analytics. It is completely fanless and is designed to operate in industrial environments such as manufacturing, smart cities, or oil and gas. This device is configured with an Intel® E3286 dual-core Atom CPU, 4 GB RAM, 32 GB SSD storage, and Wi-Fi connectivity.

Figure 3: HPE Edgeline EL10 IoT System.

- **HPE EL20:** The HPE Edgeline EL20 IoT System is a mid-level ruggedized compute solution designed for light data aggregation and acquisition. This Edgeline device delivers a performance solution that comes optimally configured with an Intel® i5-4300U CPU, 8 GB RAM, 64 GB SSD storage, Wi-Fi connectivity, and an expansive I/O selection including four-port Power over Ethernet (PoE) plus 1 x 8 bit digital input/output (DIO).

Figure 4: HPE Edgeline EL20 IoT System.

- **HPE EL1000:** This rugged, compact Converged Edge System is designed specifically for harsh environments, providing data center-level capabilities at the Edge that delivers immediate insight from IoT data. It can carry datacenter-class compute cartridges with Intel Xeon x86 processors, perform unique integration of precision data capture and control, and is managed with data center-class security and systems management software. In addition it can carry wireless and 3GPP data cards for backhaul traffic. Deep Edge compute capabilities will enable businesses to make real-time decisions, adding value to their operational processes that result in better business outcomes.

Figure 5: HPE Edgeline EL1000 Converged Edge System.

- **HPE EL4000:** The HPE EL4000 chassis supports up to four HPE ProLiant m510 server cartridges, for a total of 64 Intel Xeon cores. This flexible design allows the unit to scale a monitoring solution across a large number of industrial systems or for configuring redundancy within the solution. The EL4000 operates in the same manner as an EL1000 with the exception that it has redundant power options.

Figure 6: HPE Edgeline EL4000 Converged Edge System.

- **Moonshot m510 cartridge:** The HPE ProLiant m510 server cartridge is designed to enhance the performance of many general purpose workloads. The ProLiant m510 server cartridge has one Intel® Xeon® D-1548 (8-core) or D-1587 (16-core) with up to 128GB of ECC protected memory, dual 10Gb Ethernet along with up to 2 (1TB NVMe each) M.2 flash storage modules, and up to one 240GB SATA M.2 SSD for local OS booting. This is used as the compute engine for the EL1000 & 4000 system described earlier.

Figure 7: HPE ProLiant m510 cartridge.

- **Moonshot m710x cartridge:** The HPE ProLiant m710x server cartridge is designed to enhance the performance of video transcoding and HPC workloads. The ProLiant m710x server cartridge has one Intel® Xeon® E3-1585L v5 (4-core) with up to 64 GB of ECC protected memory, dual 10Gb Ethernet along with up to four (1TB NVMe each) M.2 flash storage modules, and up to one 240GB SATA M.2 SSD for local OS booting. This is used as the compute engine for the EL1000 & 4000 system described earlier.

Figure 8: HPE ProLiant m710x cartridge.

- **IP Camera:** This solution used an off-the-shelf 1080p NVR Camera, capable of high definition 1080p (2.1 MEGAPIXEL) video with 1000TVL sensors, 115ft (35m) night vision range with IR cut filter which was rated for both indoor and outdoor use with IP66 weather-resistant rating. This is an 802.3af class 2 device (15W power). Cameras from Axis Communications and Milestone Systems are typically recommended.

Figure 9. Swann NHD820 security camera.

Software

- **HPE Intelligent Data Operating Layer (IDOL):** HPE Intelligent Data Operating Layer (IDOL) server integrates unstructured, semi-structured, and structured information from multiple repositories through an understanding of the content. It delivers a real time environment to automate operations across applications and content, removing all the manual processes involved in getting information to the right people at the right time.

  Key IDOL Components:

  - **Connectors:** Connectors enable automatic content aggregation from any type of local or remote repository (for example, a database, a Web site, or a real-time telephone conversation). Connectors form a unified solution across all information assets within the organization.
  - **Security**: HPE provides the software infrastructure that automates operation on unstructured information. This software infrastructure is based on IDOL server. IDOL server makes it possible for organizations to process digital content automatically and allows applications to communicate with each other. It consists of data operations that integrate information by understanding content, and is therefore data-agnostic. The Intellectual Asset Protection System (IAS) provides an integrated security solution to protect your data:
    - **Front end security.** At the front end, authentication ensures users are allowed to access the system on which result data is displayed.
    - **Back end security.** At the back end, entitlement checking and authentication combine to ensure query results include only documents the user is allowed to view from repositories the user is allowed to access.
    - **Secure communications.** Communications can be encrypted between ACI servers and any applications using the HPE IDOL ACI API. The IDOL Server and its other components can be configured to use Secure Socket Layer (SSL) communications.
  - **Interfaces:**
    - **IDOL Admin** allows administration of an IDOL component. It provides a user interface for many common operations for various IDOL components. IDOL Admin is installed by the IDOL Server installer.

        - **IDOL Site Admin** allows administration of a wider IDOL installation, with multiple servers and components.
        - **IDOL Search Optimizer** allows management of the content of data indexed in IDOL Servers. By creating and modifying different business projects, which results are returned to users, and how the user views them can be defined.
        - **Find** provides a basic end-user search application for IDOL. Find can be installed by using the IDOL Server installer.
        - **ACI API** uses HTTP to allow custom-built applications to communicate with IDOL ACI servers. Several IDOL SDKs are available to allow development of applications with the ACI API.
    - **Distributed Systems:** HPE IDOL distribution solutions facilitate linear scaling of systems through faster action execution and reduction of processing time.
        - **DAH™ (Distributed Action Handler)** enables the distribution of ACI (Autonomy Content Infrastructure) actions to multiple HPE IDOL Servers, providing failover and load balancing.
        - **DIH™ (Distributed Index Handler)** enables distributed indexing of documents into multiple HPE IDOL Servers, providing failover and load balancing.
    - **Multimedia:** HPE IDOL includes components that allow incorporation of information from video, image, and audio sources.
        - **Media Server** analyzes video files and streams, to extract information about their content. Media Server can run analysis operations such as face recognition, speech-to-text, and number plate recognition.
        - **IDOL Speech Server** processes audio files, performing tasks such as speech-to-text transcription, speaker identification, and audio search
- **HPE Media Server:** The Media Server enables video processing and meaningful information extraction about its content. Media Server can be deployed for broadcast monitoring purposes, to identify when individuals or organizations appear in television broadcasts and extract information about these appearances. It may also be used on cataloging an existing archive of audio and video clips. In security and surveillance deployments, Media Server can help security personnel with automated processing, such as reading the number plates on vehicles and automatically identifying suspicious events.

  Key Media Server features**:**

- **Ingest Media:** *Ingestion* is the process of bringing media into Media Server so that it can be processed and analyzed. Media Server can ingest the following media:
  - Image files
  - Office documents such as PDF files that contain embedded images
  - Video files
  - Video from IP streams. Many devices such as IP cameras, network encoders, and IPTV devices can produce IP streams.
  - Video from cameras and third-party video management systems

- **Analyze Media:** Media Server can run many types of analyses, including:
  - Automatic number plate recognition
  - Barcode recognition
  - Color analysis
  - Face detection, face recognition, demographic analysis, and expression analysis
  - Scene analysis
  - Keyframe extraction
  - Object recognition
  - Image classification
  - Optical character recognition
  - Speaker identification
  - Speech-to-text
- **Encode and Stream Video:** When video is ingested Media Server can write the video to disk. Rolling buffers, fixed-size storage areas on disk where the oldest content is discarded to make space for the latest can be configured. A live UDP stream of the content ingested by Media Server can be created.
- **Event Stream Processing:** Media Server can be configured to filter, duplicate, and find combinations of events in analysis results. For example, Event Stream Processing (ESP) rules could be used to identify events where the text "Breaking News" appears in a television broadcast
- **Output Information:** Media Server can output the metadata that it extracts to many formats and systems, including:
  - Connector Framework Server (CFS)
  - IDOL Server
  - Vertica databases
  - XML
  - Milestone XProtect

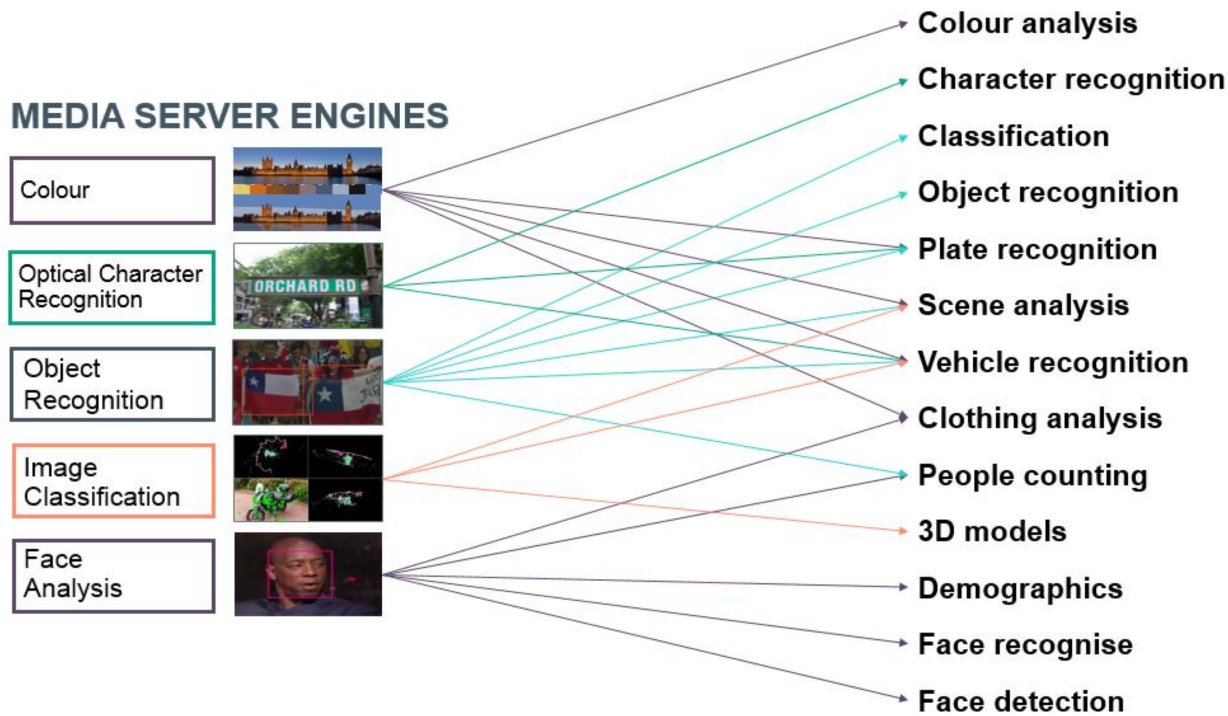


Figure 10: Core Media Server Computer Vision engines & use cases they enable.

Figure 10 shows the core HPE IDOL Media Server video analytic engines for color, character recognition, object recognition, image classification and facial analysis. These hardened core functions are combined to create use cases listed on the right. For example "people counting" in retail or industrial environments will require the combined use of object recognition (for people) and face analysis (to ensure unique people are counted).

Likewise the "scene analysis" use case involves the combined use of the "object recognition", "image classification", and "color" recognition Media Server Engines. "Vehicle recognition" for instance involves the use of multiple media server engines – color, OCR, object and image. Consequently it can perform fairly sophisticated vehicle identification including make, model, year, type, color etc. Facial analysis includes face detection, recognition, demographic identification, clothing analysis, etc.

HPE Media Server provides a configuration file for each use case listed on the right with commonly used options as parameters that can be configured for each. The input stream and the corresponding use case configuration file need to be fed as input into a media server process to trigger its operation. Media Server process management can be done from a web-based GUI or controlled through the use of REST API calls via MMAP (described later).

Scene analysis requires the use of a custom training utility that enables a user to train a model for a custom "object recognition" based on sample inputs. The trained model is only to be used as an input by the corresponding media server engine for the use case desired. The training utility is sophisticated enough to differentiate characteristics such as orientation, shape, size, direction (of movement of an object), speed, color, etc. in a “region of interest” in a certain scene. Media Server can run multiple asynchronous processes as part of its queue. Its performance and behavior is dependent on the specifics of the use case used and the type/capability of machine it is operating on (memory, cores available, etc.).

Best Practices and Configuration Guidance for the Solution

Part 1. Set Up Cameras

1. Install Ubuntu 16.04 LTS onto the EL20. Download at https://www.ubuntu.com/download/desktop.
2. Configure the proxy setting if required.
    - export http proxy=<proxy address>:<port number>
3. Connect the camera to EL20 at the POE port.

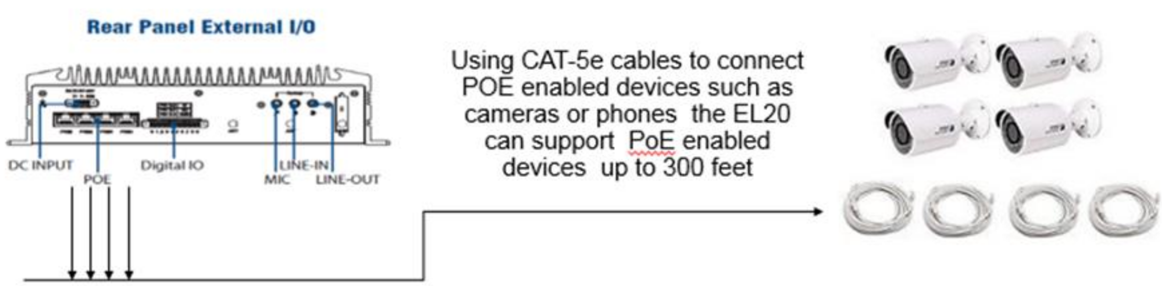


Figure 11. Camera connection to the EL20 PoE port via CAT-5e cables.

4. The HPE EL20 has 4 PoE capable ports directly on the back of the device. They are behind a single Intel I210-IT NIC and a Marvell 88E6352 (7-port PoE switch with only 5 ports used). This gives us “deep (video) data ingestion” capability immediately at the Edge.

5. The IP address for the camera in this use case is set to 192.0.0.63 and subnet mask 255.0.0.0. This may differ depending on the type of PoE camera used. The following diagram depicts the connection of the three EL20 NICs to the corresponding RJ45 connectors on the front and back.

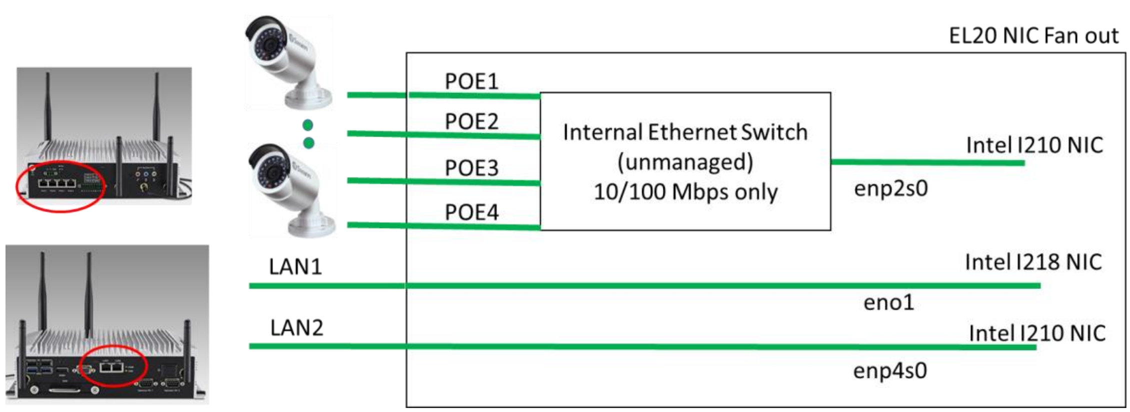


Figure 12. NIC connection inside EL20.

Before connecting the camera to the EL20 POE ports, the POE port address needs to be configured to be within the same subnet as the camera IP address to be able to successfully communicate with them. The four PoE devices connect to the EL20 (to an internal I210 NIC) via an internal unmanaged Ethernet switch. The PoE switch is 10/100 Mbps only and the corresponding Intel I210 NIC it is connected to will always show "connected" and at 10/100 Mbps from an OS. There are two more NICs (I-218,

The three NICs will enumerate in the PCIe space in the following way:

- $ lspci | grep Eth
- 00:19.0 Ethernet controller: Intel Corporation Ethernet Connection I218-LM (rev 04)
- 02:00.0 Ethernet controller: Intel Corporation I210 Gigabit Network Connection (rev 03)
- 04:00.0 Ethernet controller: Intel Corporation I210 Gigabit Network Connection (rev 03)

The interface name identification will differ depending on the type and version of OS installed (Windows and Linux). Even within Linux versions, they might get named eth0, 1, 2 or eno1, enp2s0, enp4s0, etc.

To identify the POE port:

- Check the network interfaces enumerated in terminal with command "ifconfig". In this case eno1 which corresponded to LAN1 was connected to an upstream network and was assigned a static IP address. Enp2s0 was identified as the PoE port and was assigned an IP 192.0.0.1 (as described below)

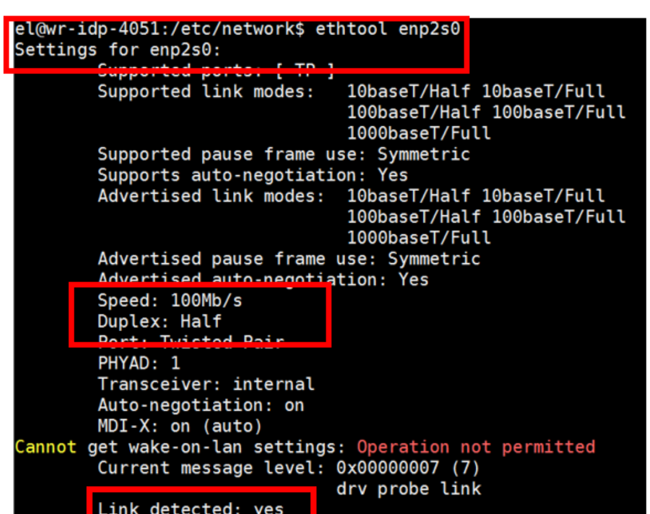
```
el@wr-idp-4051:~$ ifconfig
eno1      Link encap:Ethernet  HWaddr 00:0b:ab:a9:40:51
          inet addr:10.82.12.205  Bcast:10.82.255.255  Mask:255.255.0.0
          inet6 addr: fe80::20b:abff:fea9:4051/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 GB)  TX bytes:0 (0.0 GB)
          Interrupt:20 Memory:f7f00000-f7f20000
```

```
enp2s0    Link encap:Ethernet  HWaddr 00:0b:ab:a8:16:5e
          inet addr:192.0.0.1  Bcast:192.255.255.255  Mask:255.0.0.0
          inet6 addr: fe80::2998:6e83:8312:732c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:369692211 errors:847 dropped:0 overruns:0 frame:847
          TX packets:40957243 errors:3 dropped:0 overruns:0 carrier:11
          collisions:12436352 txqueuelen:1000
          RX bytes:399492486630 (399.4 GB)  TX bytes:2784814006 (2.7 GB)
          Memory:f7c00000-f7cfffff

enp4s0    Link encap:Ethernet  HWaddr 00:0b:ab:a9:40:52
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
          Memory:f7900000-f79fffff
```

- Terminal returns the network interface information. There should be three ports. In this case (Ubuntu 16.04.1), the three ports are named eno1, enp2s0, and enp4s0. One of these three represents the POE interface card on EL20
- To find out which one represents the POE interface, enter command “ethtool” follow by the port name in the Terminal.
- The Ethernet interface with speed at 100Mb/s and link detected: yes, the port is connected and is the POE port because the POE NIC is always connected to the EL20 switch at 10/100 Mb/s as shown above. Check all three network interfaces with the “ethtool” command. Figure 13 below shows one example of the port connected to the internal switch (i.e. enp2s0 as PoE connected NIC) and one example of the port not connected (enp4s0). In this case enp4s0 was not connected physically and it corresponds to LAN2.

Port enp2s0 is connected. It’s the POE port



Port enp4s0 is NOT connected

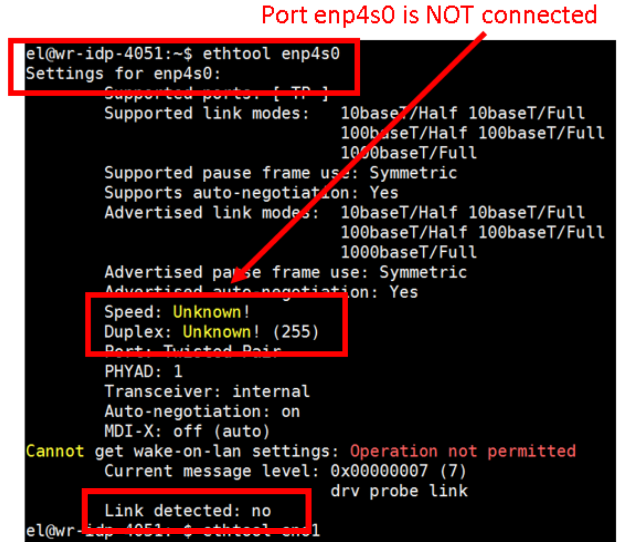
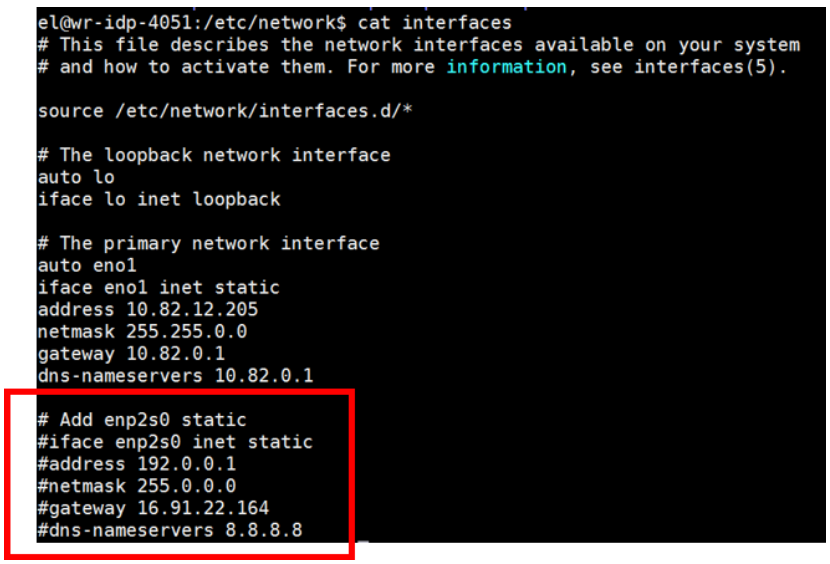


Figure 13. Ethtool response in the terminal window for two Ethernet interfaces on the HPE EL20.

- Lastly, once the POE port is identified, change its IP address to be in the same subnet as the camera IP address. In this example, we assume the camera is at 192.0.0.63 and subnet mask 255.0.0.0. This can be achieved by editing the configuration file in **/etc/network/interfaces**



```
el@wr-idp-4051:/etc/network$ cat interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eno1
iface eno1 inet static
address 10.82.12.205
netmask 255.255.0.0
gateway 10.82.0.1
dns-nameservers 10.82.0.1

# Add enp2s0 static
#iface enp2s0 inet static
#address 192.0.0.1
#netmask 255.0.0.0
#gateway 16.91.22.164
#dns-nameservers 8.8.8.8
```

Figure 14. File content snippet for /etc/network/interfaces.

- Figure 14 above shows the enp2s0 port is set to IP address 192.0.0.1 in the configuration file. This is in the same subnet as the camera IP address (at 192.0.0.63)

- We can further verify the network traffic at IP 192.0.0.1 with Wire Shark. In case of this Swann IP camera, the default IP address was not well documented. Because of this the packets on enp2s0 needed to be snipped initially. Once ARPs were observed from 192.0.0.63, it was revealed that the IP address and subnet the NIC needed to be configured to 192.0.0.1/255.0.0.0. Most IP cameras allow reconfiguring the IP address and subnet mask, the frame rate, the type of encoding, etc. via an embedded webserver interface.

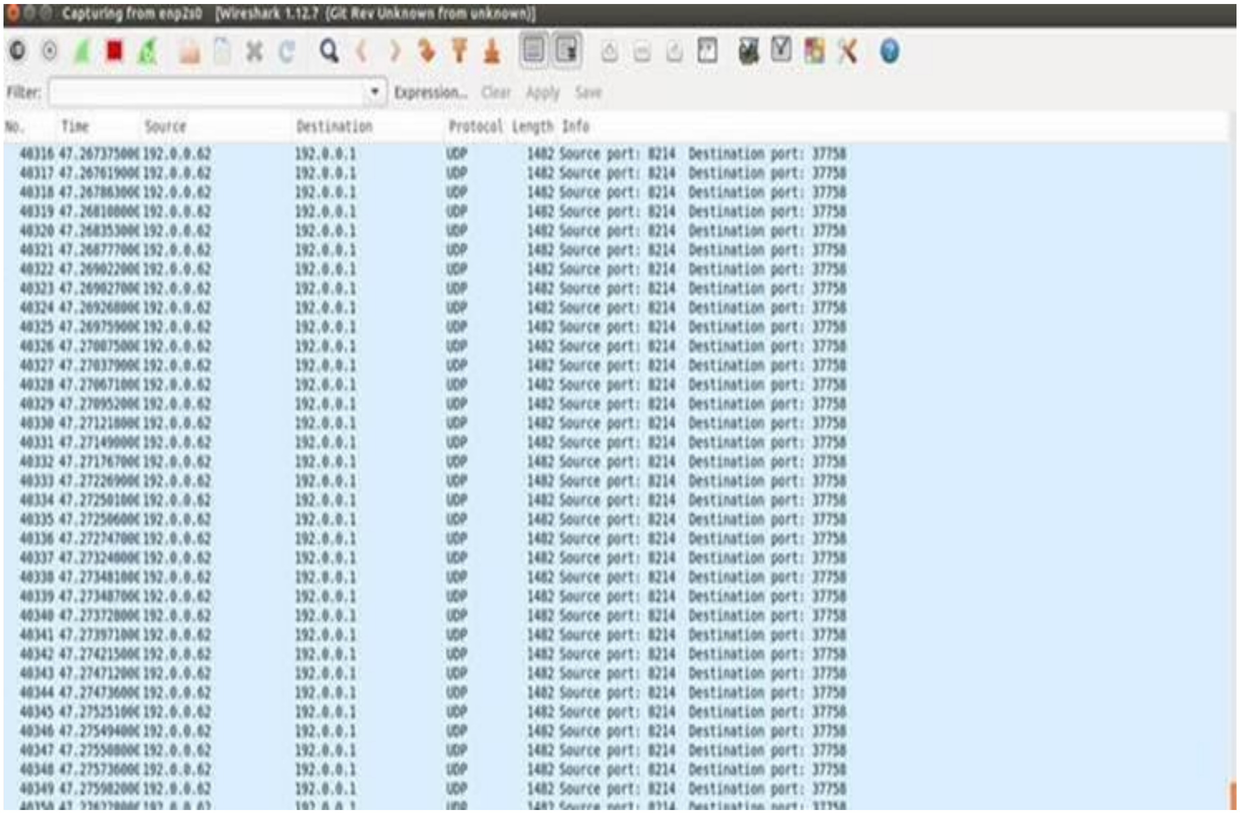


Figure 15. Wire Shark snapshot showing RTSP traffic as UDP frames from camera (192.0.0.62) to I210 NIC on EL20 (192.0.0.1).

NOTE: The EL20 in default operation does not have a DHCP server running. This is not necessary if the connected device can be configured to have a static IP address. A DHCP server can be run on the EL10/20 if required to serve IP addresses separately.

6. Ping 192.0.0.63 to make sure the camera is there and is accessible from the host.

7. Verify video feed using VLC player. This can be downloaded from videolan.org. The camera in this use case has IP address at 192.0.0.63, login user name: admin, and password: 12345 and it offered an RTSP stream. This information can be viewed by opening the media in Network Tab on VLC Media player.

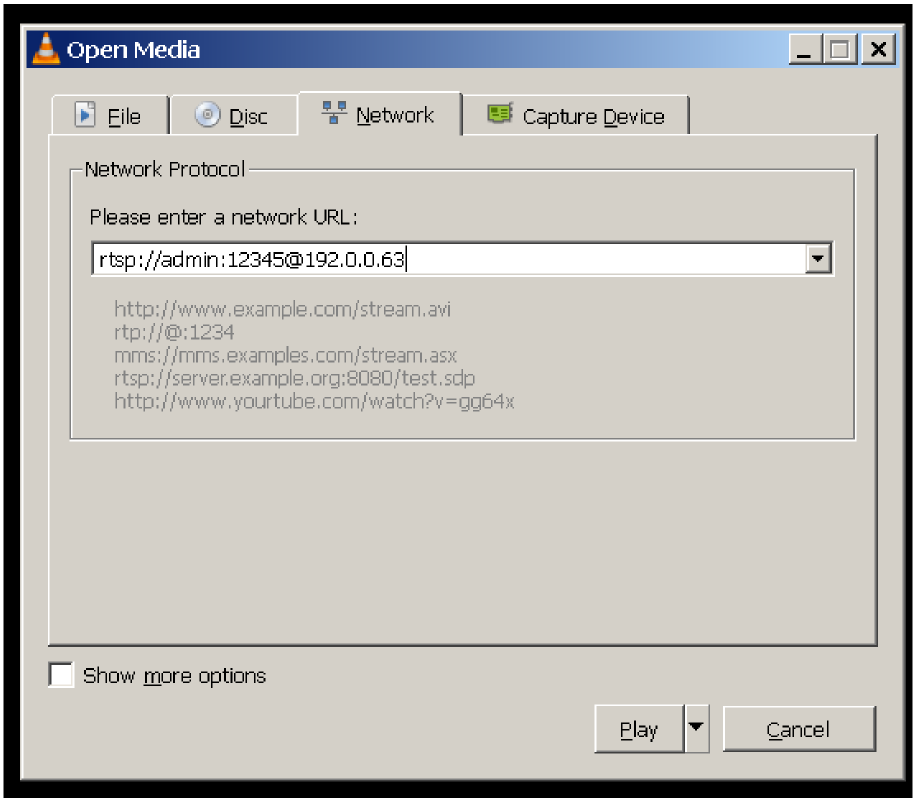


Figure 16. Configuring VLC player to ingest/open a RTSP feed from a directly attached camera.

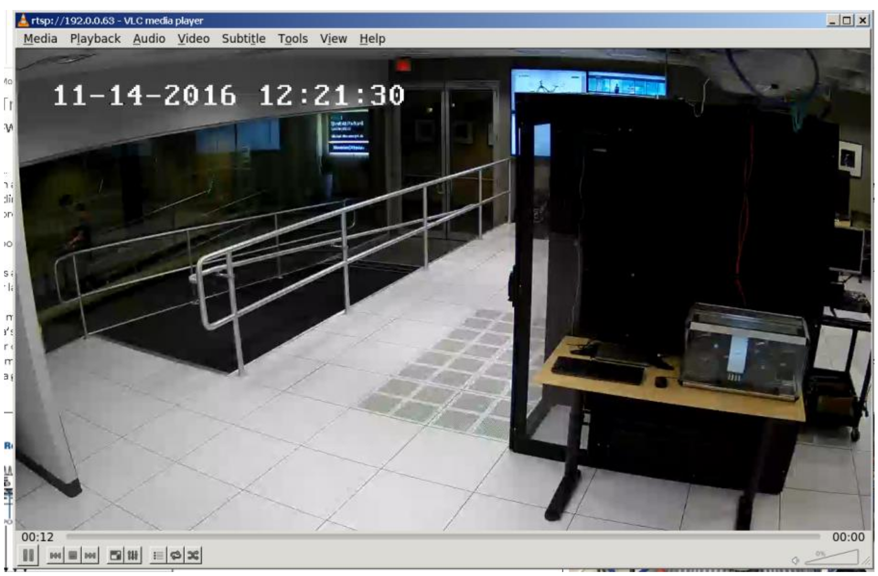


Figure 17. VLC Player displays the camera feed through the PoE connection.

Part 2. Install HPE Media Server

8. PDF Documentations can be found at HPE GitHub.
   - https://github.hpe.com/jih-tsen-nat-lin/MediaServer_Doc

9. Obtain the license server and the media server executables from HPE GitHub.
    - https://github.hpe.com/jih-tsen-nat-lin/MediaServer_Executable
    - https://github.hpe.com/jih-tsen-nat-lin/LicenseServer_Executable
10. Obtain the setup files from HPE GitHub. Refer to Appendix B for more detail.
    - https://github.hpe.com/jih-tsen-nat-lin/Face

Part 3. Install IDOL License Server

11. Obtain the IDOL license. Contact Global Solutions Engineering Support at https://downloads.autonomy.com/login.do
12. Extract the zipped License Server executable files and upload the license server files to directory /opt/licenseserver/ directory on EL20.
13. Rename the IDOL license file, **license.dat**, to **licensekey.dat** and save to the license server file directory in EL20.
14. Set privilege to the licenseserver directory.
    - chmod 777 /opt/licenseserver
15. In the **licenseserver.cfg** file, change all the “localhost” to “*.*.*.*” (total three places in the file). See Appendix B.1.
16. Copy the **start.sh** and **stop.sh** files to directory /opt/licenseserver/ See Appendix B.2 and B.3
17. Give privilege to both script files:
    - chmod 777 start.sh
    - chmod 777 stop.sh
18. Continue to next section for Media Server installation.

Part 4. Install IDOL Media Server

NOTE: For more information, refer to the user manual, MediaServer_11.1_admin_en.pdf

19. Obtain the media server application from the HPE GitHub link.
20. Create a media server directory in EL20.
    - sudo mkdir /opt/MediaServer_11.1.0_LINUX_X86_64
21. Upload the unzipped media server files to the mediaserver directory.
22. Set privilege to **mediaserver.exe**.
    - chmod 777 mediaserver.exe
23. Copy the **mediaserver-activity.html** from the software package into directory /opt/MediaServer_11.1.0_LINUX_X86_64/. See Appendix B.4.

NOTE: mediaserver-activity.html sets the layout of the media server activity page.

24. Save the **mediaserver.cfg** file to directory /opt/MediaServer_11.1.0_LINUX_X86_64/ if not already there. See Appendix B.5.
25. Set privilege for the configuration file.
    - sudo chmod 777mediaserver.cfg
26. Save the **start.sh** and the **stop.sh** files to the media server directory /opt/MediaServer_11.1.0_LINUX_X86_64/. See Appendix B.6 and B.7.
27. Set privilege to the script files.
    - sudo chmod 777 start.sh
    - sudo chmod 777 stop.sh
28. Start the license server and the media server software.
    - Navigate to directory /opt/licenseserver , and run the script
        - ./start.sh
    - Navigate to directory /opt/MediaServer_11.1.0_LINUX_X86_64, and run script
        - ./start.sh
29. To verify the Media Server and the license server are running, open a web browser and navigate to the Media Server Administrator page.
    - http://[IP Address]:14000/action=admin#page/status

30. Click on the **License** tab. **Valid** should have **true**. This means you have the media server operational.

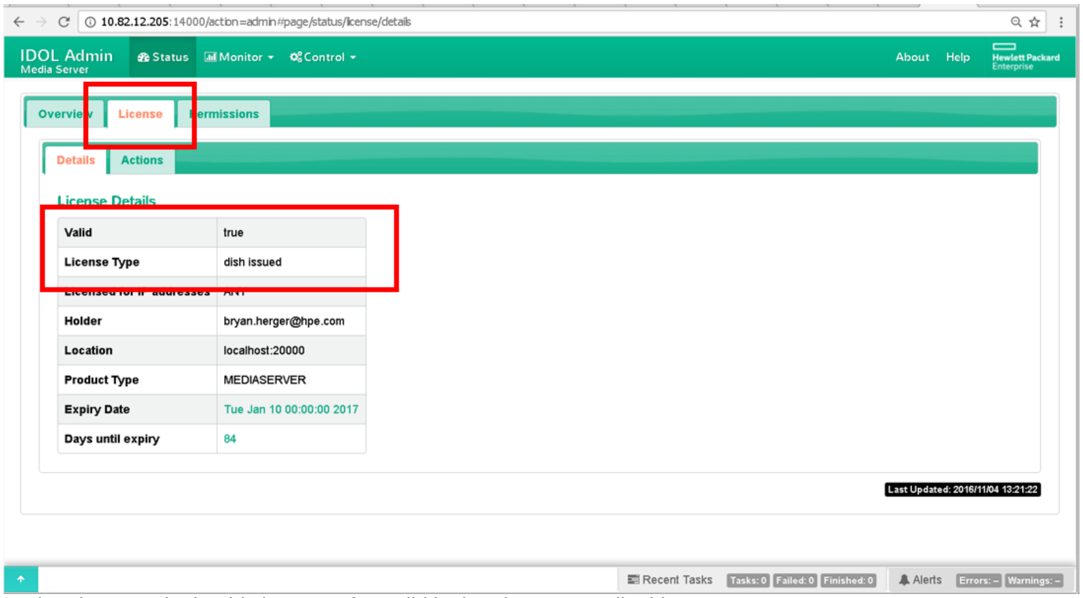


Figure 18. The License tab should show true for Valid in the License Detail table.

NOTE: To stop license server, Navigate to directory /opt/licenseserver, and run script ./stop.sh.

NOTE: To stop Media Server, Navigate to directory /opt/ MediaServer_11.1.0_LINUX_X86_64, and run script ./stop.sh.

Part 5. Set up Face Detection

NOTE: For more information on the content of the configuration file, please refer to the Media Server help page at page at link: http://[IP Address]:14000/action=help. See section **Configuration Parameters → Analysis**. The media server and the license server need to be running to access the help page.

31. Copy two files, **Face.bat** and **Face.cfg** onto the system. See Appendix B.8 and B.9. They can be downloaded from HPE GitHub.
    - The **Face.bat** file calls the **Face.cfg** configuration file
    - See below for the content for the **Face.bat** file

      curl "http://localhost:14000/a=process&source=rtsp://<Camera login user name>:<Camera login credential>@<camera IP>&configpath=/opt/videos/Face.cfg&Persist=true"
      el@wr-idp-4051:/opt/videos$ ls

    - ***source=rtsp://<Camera login user name>:<Camera login credential>@<camera IP>*** sets RSTP address for video ingestion
    - ***configpath*** sets the path to the **Face.cfg** file
    - **Face.cfg** sets the configurations for face detection and face recognition

[FaceDetect]
Type=facedetect
ColorAnalysis=true
NumParallel=4
SizeUnit=pixel
Orientation=any
SampleInterval=20
[FaceRecognize]
Type=FaceRecognize
Input=FaceDetect.ResultWithSource
SyncDatabase=TRUE
Database=HPE

32. To run face detection, navigate in Terminal to the directory that contains the **Face.bat** file. Run the script.
    - ./Face.bat
33. The terminal should return **SUCCESS**.

```
el@wr-idp-4051:/opt/videos$ ./Face.bat
<?xml version='1.0' encoding='UTF-8' ?><autnresponse xmlns:autn='http://schemas.autonomy.com/aci/'><action>PROCESS</action
><response>SUCCESS</response><responsedata>
<token>MTI3LjAuMS4xOjE0MDAwOlBST0NFU1M6MTQ3ODc5MjkyNDU5NTAwMDEyMDUwODM3</token></responsedata></autnresponse>
el@wr-idp-4051:/opt/videos$
```

Figure 19. IDOL Media Server should return XML response SUCCESS if the batchfile runs succesfully in the Terminal.

34. For face detection, it works out of the box.
35. To test face detection, open a web browser and navigate to the Media Server activity page at http://[IP Address>]:14000/action=activity.

36. Click **Connect**. Click on **FaceDetect**, and in **ResultWithSource** tab you should see face detection results.

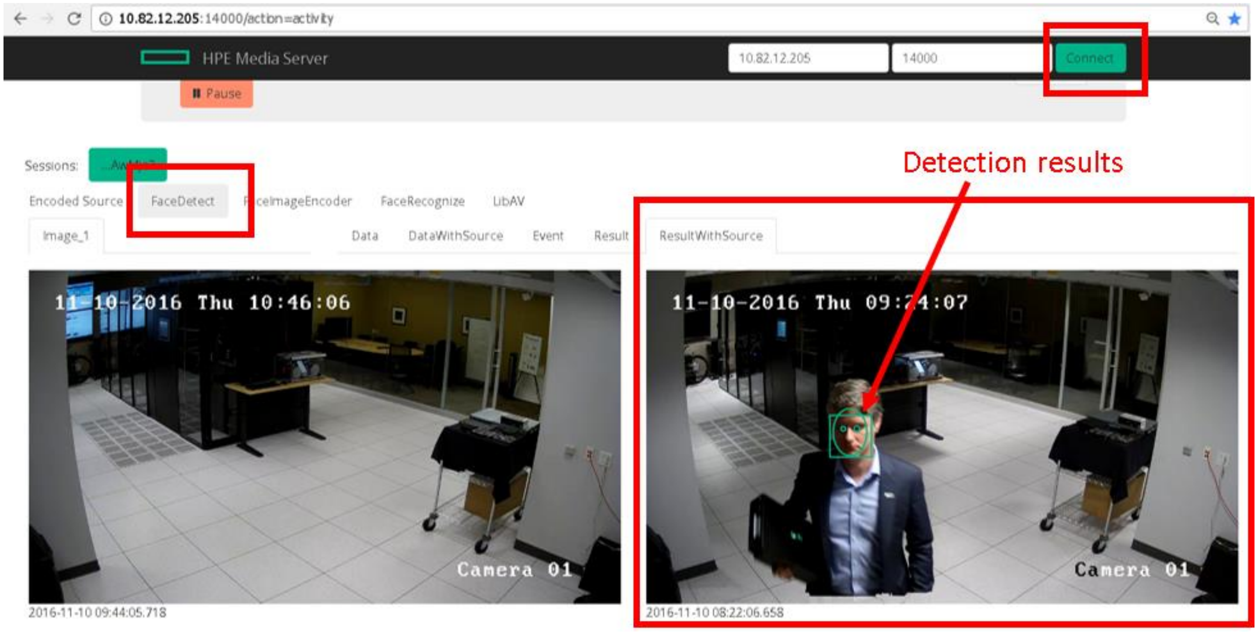


Figure 20. Face detection results live in the Media Server Activity page.

Part 6. Training for Face Recognition

NOTE: For more information, refer to the user manual, MediaServer_11.1_admin_en.pdf.

NOTE: Providing multiple images can improve the machine learning. Face images are in JPEG format. Face image examples are shown below:

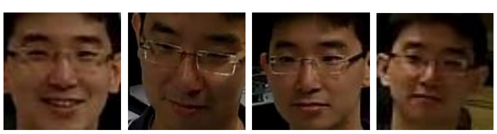


Figure 21. Example of face images used for face recognition training.

37. If not started, start the license server and the Media Server.
38. Create Face database.
    - In a web browser, navigate to http://[Media Server IP]:14000/action=admin#page/status
    - Click on **Control**, and then **Databases**
    - Click on **Add New Database**

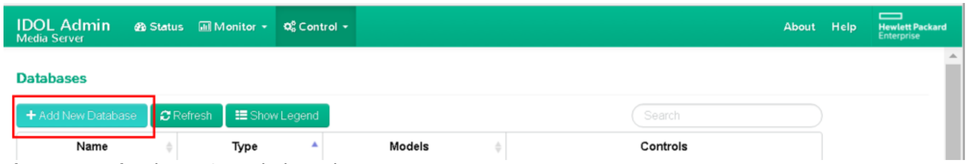


Figure 22. User interface menu for the IDOL Admin web page.

- Enter **Name** of the database and Select **Face** type

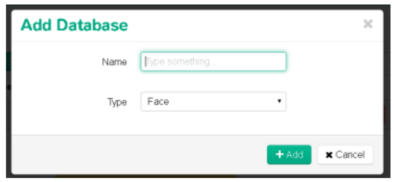


Figure 23. The pop-up menu for adding new database.

39. Create face models. There are two methods:

Method 1: Initiate training with Media Server's web menu

- In web browser, navigate to http://[Media Server IP]:14000/action=admin#page/status
- Click on **Control**, and then **Models**

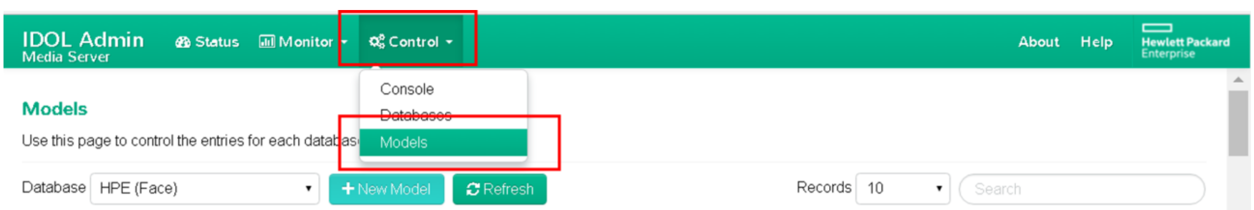


Figure 24. Access the Models menu in the IDOL Admin web page.

- Click **New Model**

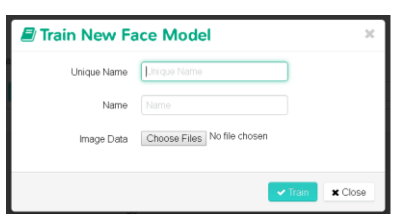


Figure 25. The pop-up menu for training new face models.

- In the **Train New Face Model** window, enter the person's name in both **Unique Name** and **Name** fields, then click on **Choose Files** to select the face images of the person; then Click **Train**
- The new model should show up in the face database. This indicates training success. If the face does not show in the face database, the training failed

Method 2: Training using curl call in the terminal

- In terminal, navigate to the directory containing the face images for training
- Enter the curl command in terminal to send face training request to the Media Server

  curl http://[Media Server IP]:14000 -F action=TrainFace -F database=<Database Name> -F imagedata=@<Face Image File names with Extension> -F identifier=<Unique Identifier> -F imagelabels=<Images Labels> -F metadata=fullname:"<Full Name>"

- **Example**: Train face for Nathan_Doe using 8 face images with Media Server running on the local host

  curl http://localhost:14000 -F action=TrainFace -F database=FaceDataBase -F imagedata=@nathan1.jpg, nathan2.jpg, nathan3.jpg, nathan4.jpg, nathan5.jpg, nathan6.jpg, nathan7.jpg, nathan8.jpg -F identifier=Nathan -F imagelabels=1,2,3,4,5,6,7,8 -F metadata=fullname:"Nathan_Doe"

- In terminal, a **SUCCESS** message should be displayed
- The new model should show up in the face database. This indicates training success. If the entry not show in the face database, the training failed

40. To test face recognition, open a web browser and navigate to the Media Server activity page at:
    - http://[IP Address]:14000/action=activity.
41. Click **Connect**. Click on **FaceRecognize**, and in the **ResultWithSource** tab face results should be displayed.

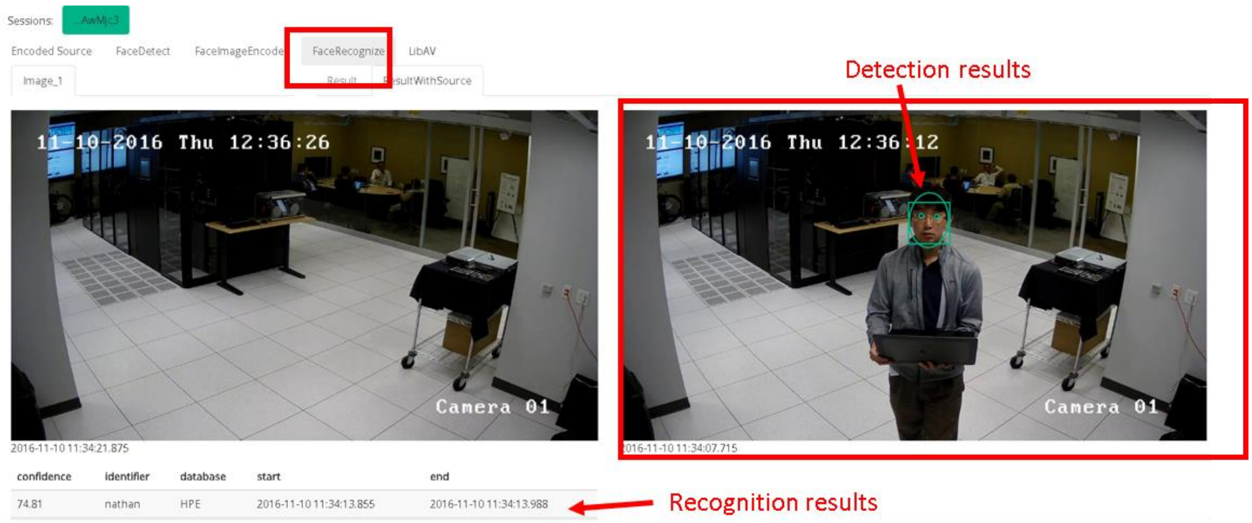


Figure 26. Face recognition results live in the Media Server Activity page.

Part 7. Demo - Run Face Detection and Face Recognition Live

42. If not yet started, start the license server and the media server software.
    - Navigate to directory /opt/licenseserver, and run script
        - ./start.sh
    - Navigate to directory /opt/MediaServer_11.1.0_LINUX_X86_64, and run script
        - ./start.sh
43. Execute **Face.bat**. The terminal should return **SUCCESS**.
44. Open web browser and navigate to the Media Server activity page at http://[IP Address]:14000/action=activity.
45. Click **Connect**. Click on **FaceDetect**, and in **ResultWithSource** tab to see the face detection results.
46. Click **Connect**. Click on **FaceRecognize**, and in the **ResultWithSource** tab to see the face recognition results.

Part 8. Demo - Run Face Detection and Face Recognition on a Pre-recorded Video

NOTE: Download the **FaceVideo.bat** file, the **Face.cfg** file, and the pre-recorded video from HPE GitHub. See Appendix B.9, B.10 and B.11 for more information.

47. If not yet started, start the license server and the media server software.
    - Navigate to directory /opt/licenseserver, and run script
        - ./start.sh
    - Navigate to directory /opt/MediaServer_11.1.0_LINUX_X86_64, and run script
        - ./start.sh
48. Execute **FaceVideo.bat**. The terminal should return **SUCCESS**.

    NOTE: FaceVideo.bat specifies the path to the pre-recorded video at directory at */home/el/Documents/vlc/FaceVideo2.mp4.* This must be changed to the location of the pre-recorded video file if the demo option is used.

49. Open web browser and navigate to the Media Server activity page at:
    - http://[IP Address]:14000/action=activity.
50. Click **Connect**. Click on **FaceDetect**, and in **ResultWithSource** tab to see the face detection results.
51. Click **Connect**. Click on **FaceRecognize**, and in the **ResultWithSource** tab to see the face recognition results.

Part 9. To kill process running on the media server

52. Any update in the configuration file requires restarting the process in Media Server. First kill the running process. Navigate to the admin asynchronous queues page at:
    - http://[IP Address]:14000/action=admin#page/async-queues/PROCESS.
53. Copy the process token.

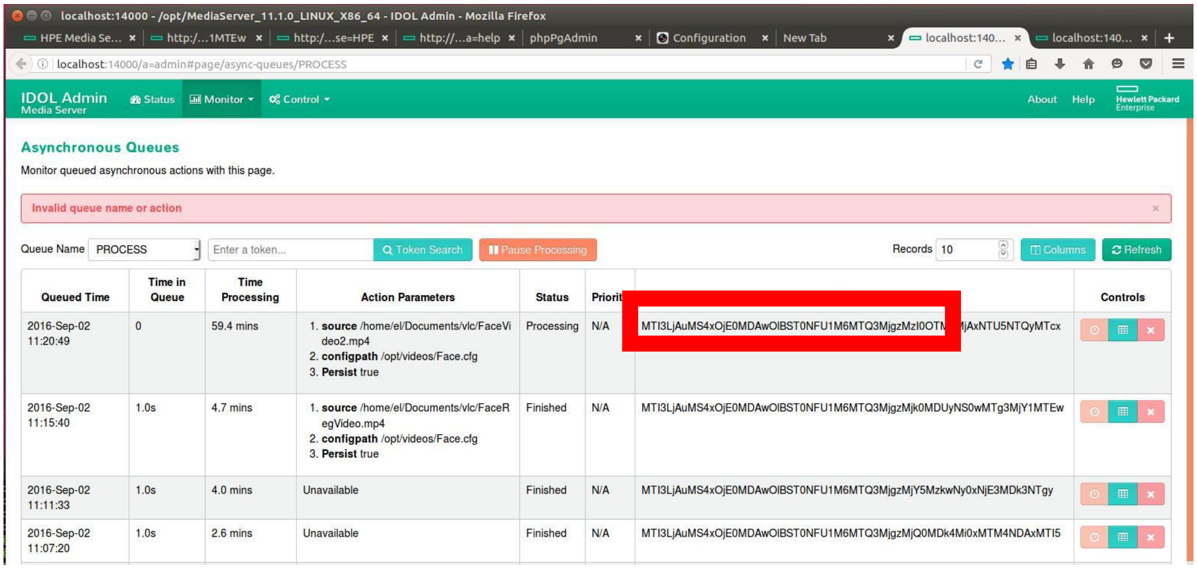


Figure 27. The IDOL Admin Monitor pages shows the job process ID.

54. In a blank web page, enter the following http link to kill the process. Enter the token information at **<Token>**.

 http://[IPaddress]:14000/a=queueinfo&queuename=process&queueaction=stop&token=<Token>

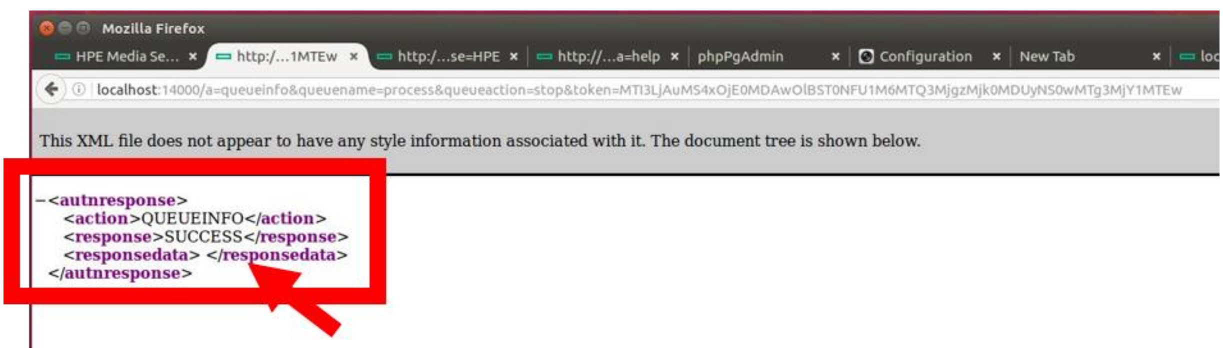


Figure 28. The XML response should show SUCCESS for a job successfully terminated.

55. It should respond SUCCESS.
56. Check in the media server admin page if the process is stopped.

Analysis and Recommendations

This section outlines the results of studies undertaken to understand the capability of HPE Edgeline IoT Systems processing the Media Server Face recognition and detection workloads.

Table 1 lists the configuration details for each of the systems that were under this performance study; it includes the EL10 and the EL20 from the HPE Edgeline product line, the m510-8 core, the m510-16 core, and the m710x from the HPE ProLiant product line, which goes inside the Edgeline EL1000 and EL4000 systems, and lastly the DL360 Gen9 server from the ProLiant product line, in which is a reference for comparison.

The HPE EL10 and EL20 are examples of the “light” edge nodes. Specifically, the EL10 holds a two-core Intel Atom E3826 CPU with 4GB memory, while the EL20 has Intel i5 4300U CPU with 8 GB of memory. The EL1000 and EL4000 systems, on the other hand, are examples of the “heavy” edge nodes. With the m510 and m710x cartridges, the EL1000 and EL4000 systems have significantly more processing power, memory capacity and storage capability comparing to that of the light-edge devices. In detail, the EL1000 can either hold a single m510 or m710x cartridge, whereas the EL4000 can hold up to four cartridges.

The HPE DL360 Gen9 system has two Intel Xeon E5-2640 v3 processors and 128 GB memory. This type of system is commonly found in a data center environment, and it servers to facilitate a comparison between what can be done at the edge with an Edgeline product family and what can be done in the data center with industry-grade hardware. All tests were performed on HPE Media Server software, version 11.0.0 for Linux (x86_64) running in the CentOS 7.2 operating system.

Table 1: IoT Systems under test.

SERVER	CPU	# CORES	INSTALLED MEMORY	MAXIMUM MEMORY
HPE EL10	Intel Atom E3826	2	4 GB	8 GB
HPE EL20	Intel i5-4300U v3	2	8 GB	8 GB
HPE EL1000 with m510 – 8 core	Intel Xeon D 1548 v4	8	80 GB*	128 GB
HPE EL1000 with m510 – 16 core	Intel Xeon D 1587 v4	16	80 GB*	128 GB
HPE EL1000 with m710x	Intel Xeon E3 1585L v5	4	64 GB	64 GB
Reference HPE DL360 gen9	2 x Intel Xeon E5 2640 v3	2x8	128 GB	768 GB

The study mainly focused at the CPU utilization and the network traffic at each case of1, 2, 3, and 4 video streams ingested into the Media Server. These streams are required to be decoded and rendered. This is independent of any video analytic which is performed subsequently on them. Furthermore, all setup used in this study utilized only the default encoder; there was not any special purpose hardware blocks, which may be present on some processors. Video streams were 1080p (1920x1080) at 30 FPS with H.264 encoding, and max bitrate set to 4096 Kbps.

Table 2 shows the utilization for CPU and for memory for up to four PoE H.264 encoded streams on the HPE EL20. These are live video decoding and rendering on the system. This represents the overhead that can be expected to be encountered when performing video analytics on the same system. The experiment results this section were carried out on pre-recorded streams. Media Server decodes the video stream with standard open source *libav* decoder.

Table 2. CPU, Memory, Network Utilization in ingesting and decoding H.264 streams from PoE cameras (HPE EL20 example).

NUMBER OF STREAMS	CPU UTILIZATION	MEMORY UTILIZATION	NETWORK UTILIZATION
1 PoE Stream	~20%	5.4GB	~150 KB/s
2 PoE Streams	~36%	5.6GB	~502 KB/s
3 PoE Streams	~62%	5.7GB	~1021 KB/s
4 PoE Streams	~75%	5.8GB	~1500 KB/s

Each Media Server vision analytic has its own custom configuration options/parameters depending on the use case. In performance evaluation, a parameter to be explored is called a factor, and the different values the parameter can take are called levels. In this study, it was not possible to perform a full-factorial design due to the lack of availability of ground truth data, which means that not all combinations of the selected factors and levels were explored to gain better insight into which factors and levels have the most significant effect on the application's behavior. The set of tests documented here are only relative runs of the IDOL software running on the servers above for a pre-recorded video stream which was created in a lab environment (with participants permission).

The only parameters which varied were the number of video streams injected into Media Server's processes in parallel. Only the run time when varying the number of instances from 1-30 was measured; while simultaneously monitoring the number of records processed and records dropped; the CPU utilization, memory usage, number of interrupts handled, the number of context switches encountered. When the number of records dropped reached an arbitrarily high threshold, a limit has been reached for a given platform and no further modifications can be done without significant changes to hardware or algorithm used to be able to sustain a higher number of streams.

20 people participated in the recording of the 20 minute reference video. The face recognition training for these people was done with 4 different snapshots of each person's face.

The technical terms of the metrics reported are defined here:

Run Time – Wall clock time used by a given configuration to process the workload.

Frames Dropped – Number of video frames dropped by Media Server while processing the workload.

Precision – The fraction of retrieved instances that are relevant.

Recall – The fraction of relevant instances that were retrieved.

F-Measure – Weighted harmonic mean of a system's precision and recall.

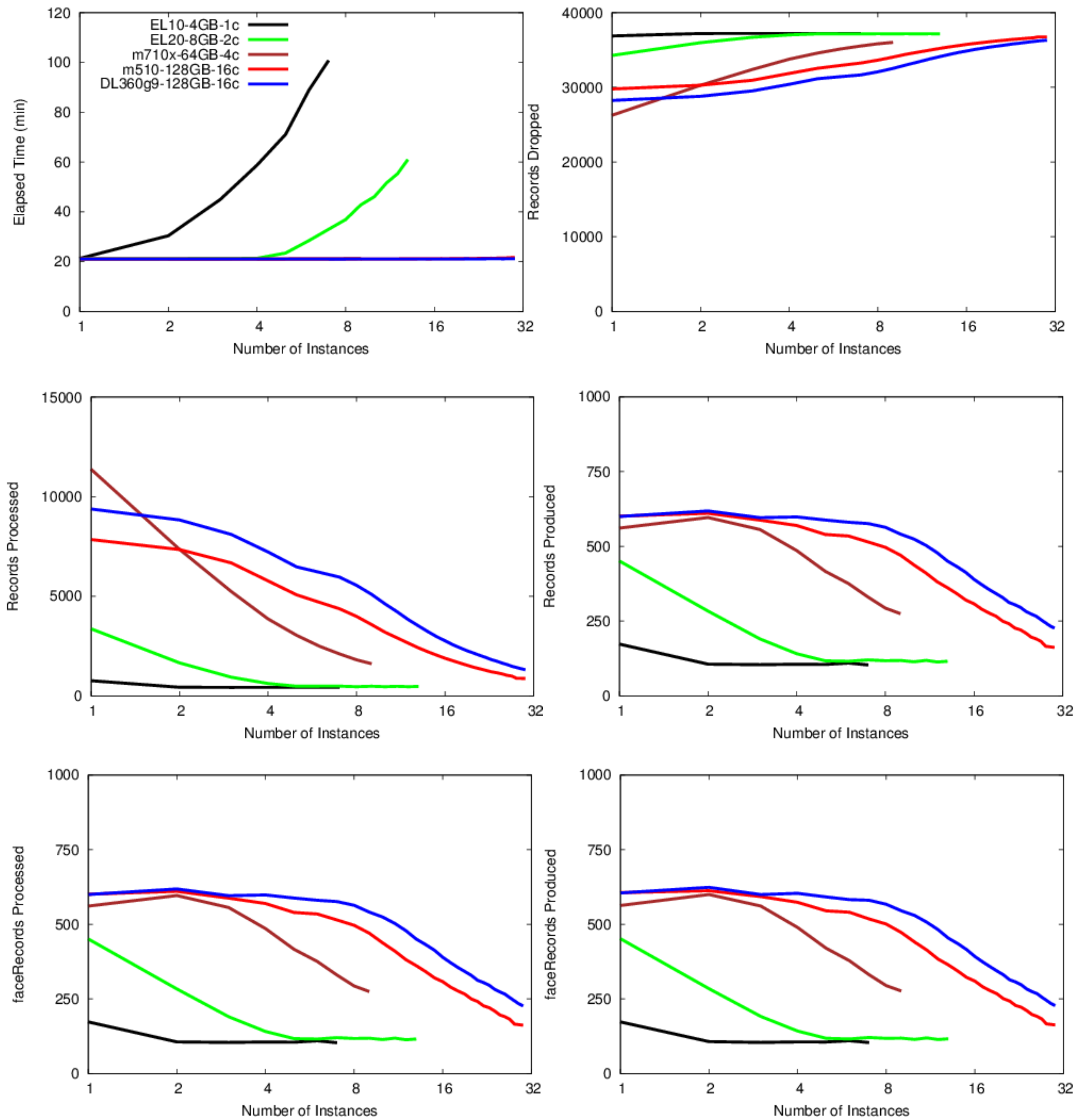


Figure 29. Elapsed time, records dropped, records processed, records produced with varying instances (#streams) on systems under test.

Figure 29 shows the resulting elapsed time, records dropped and records processed for face detection on each of the system. This is followed by records produced, *faceRecords* processed, and *faceRecords* produced for the face recognition. All statistics were as reported by the Media Server.

From the elapsed time graph, it was observed the EL10 unable to reliably process more than one video stream; a fairly large number of frames got dropped. It struggled to keep up with face recognition- thereby exponentially increasing its run time as more streams are required of it. It was also observed the EL10 can process 7 concurrent instances, while the EL20 can process 13. With hyper threading enabled, the 2 core in the EL20 system can be considered as 4 parallel threads. It was able to keep up real-time face detection/recognition for up to 4 parallel duplicated streams, although with fairly high dropped records.

The m710x and m510 (16-core variant) form the compute nodes for the EL1000 and the EL4000 were able to keep up near real-time behavior much better with progressively increasing dropped records as the number of instances increased. Total 9 concurrent instances on m710x (a 4-core Xeon E3 core) were able to be run. Beyond that, some instances started to fail. Media Server runs legacy (feature comparison) algorithms for face detection which are known to require at least one core per stream for real-time detection.

The statistics for the “detection” phase showed a high number of records dropped, approximately 10,000, while those for “recognition” phase with approximately 500. The same statistics were observed on the reference DL360 Gen 9 system. This may be an artifact of how the face detection engine operates. Once a face has been detected, Media Server ignores the remaining frames and drops them. There are additional configurations which help improving drop rate. It needs further investigation.

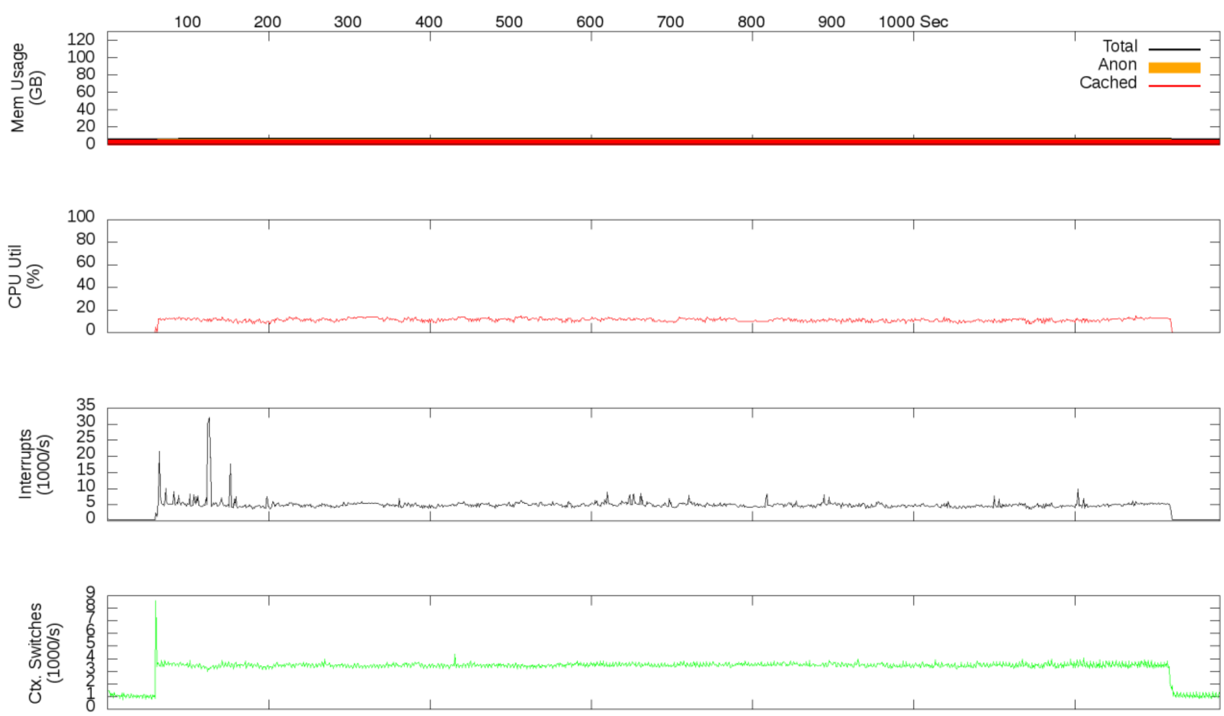


Figure 30. CPU Utilization on a reference Intel Xeon E5 2640 v3 server (16 cores) for face detection/recognition.1 instance- relatively low CPU usage.

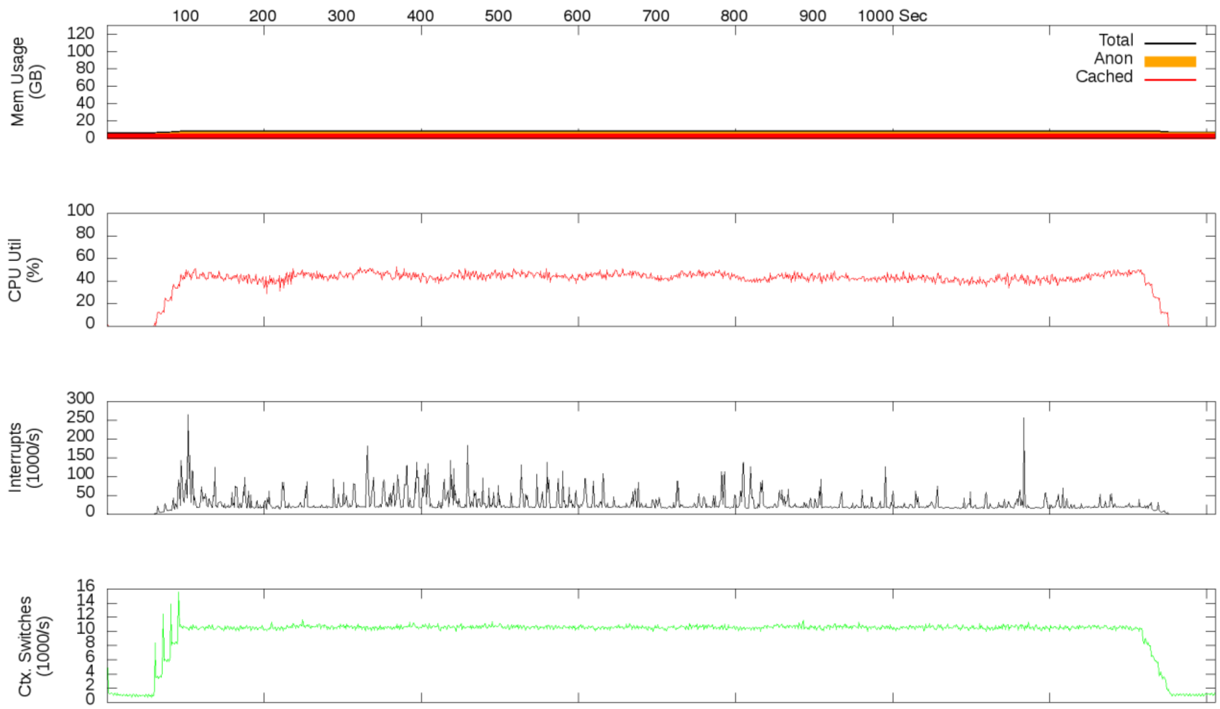


Figure 31. CPU Utilization on a reference Intel Xeon E5 2640 v3 server (16 cores) for face detection/recognition. 4 instances (beyond this records produced starts to drop).

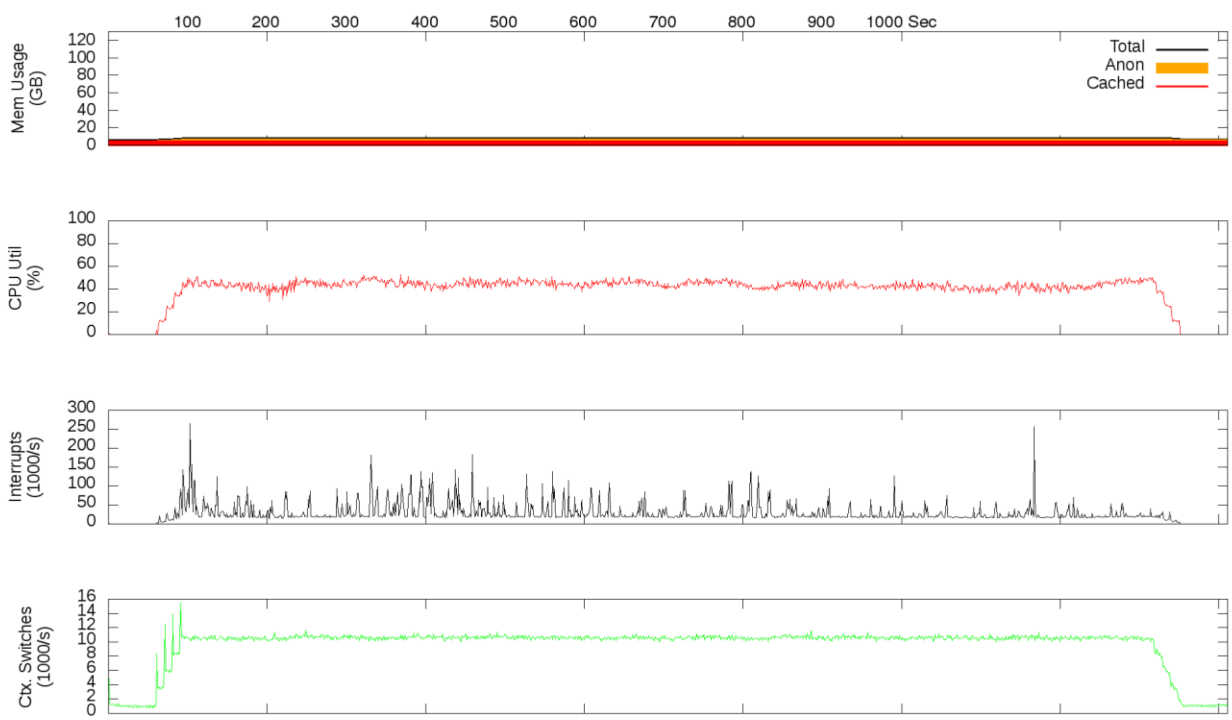


Figure 32. CPU Utilization on a reference Intel Xeon E5 2640 v3 server (16 cores) for face detection/recognition. 12 instances- CPU utilization at 100%.

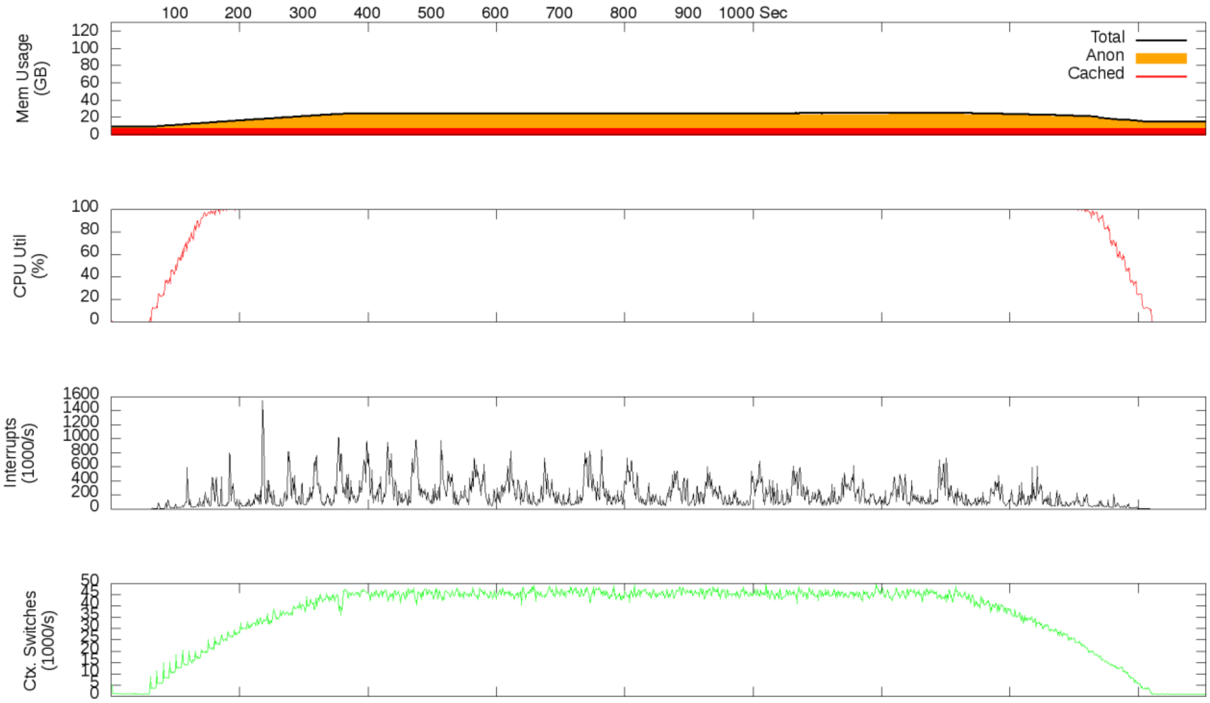


Figure 33. CPU Utilization on a reference Intel Xeon E5 2640 v3 server (16 cores) for face detection/recognition. 30 instances- CPU fully utilized throughout.

Figures 30 through 33 show the CPU utilization, memory usage, and context switches encountered in case of a reference datacenter system, which is used to baseline the Edgeline systems performance. One instance (baseline) took approximately 21 minutes to run; 605 face records produced. It maintained 600 face records produced with up to four concurrent instances (also up to 8 instances). Although it can be pushed to about 30 instances (beyond which it *segfaults*) and have similar average running time, only about 228 face records produced on average as it is observed the in *RecordsProduced* graph in Figure 29. Up to 12 parallel streams are reasonable to expect for face detection/recognition on an un-optimized reference system built for the datacenter. Further optimizations and tuning can be done depending on the use case involved.

Error! Reference source not found. shows the same results on the EL10 and the EL20 respectively. On the EL10, after one instance, the average running time grows rapidly. For the reference video under test with only one instance, the software produced only 173 face records (less than1/3 of what a DL360 system can produce). Refer to the Records Produced graph in Figure 29 for more detail.

The EL20, when operating on one stream, produces 452 face records. With four concurrent instances running, it produced 142 face records on average. Beyond four instances, its average running time grows rapidly. Please refer to the Record Produced graph in Figure 29 for more detail. This indicates that for face recognition, the EL10 could only handle up to one stream, while the EL20 could not handle no more than two streams.

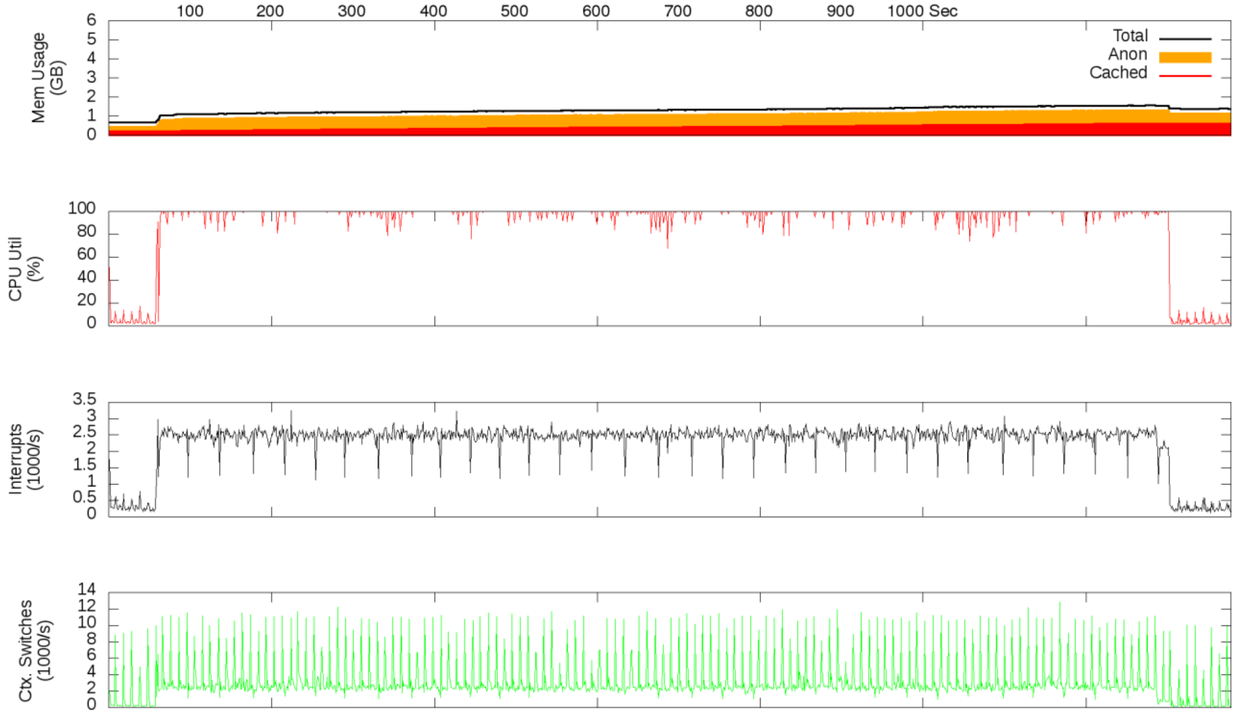


Figure 33. CPU Utilization on HPE Edgeline EL10 Face detection/recognition workloads. 1 instance- CPU is 100% utilized.

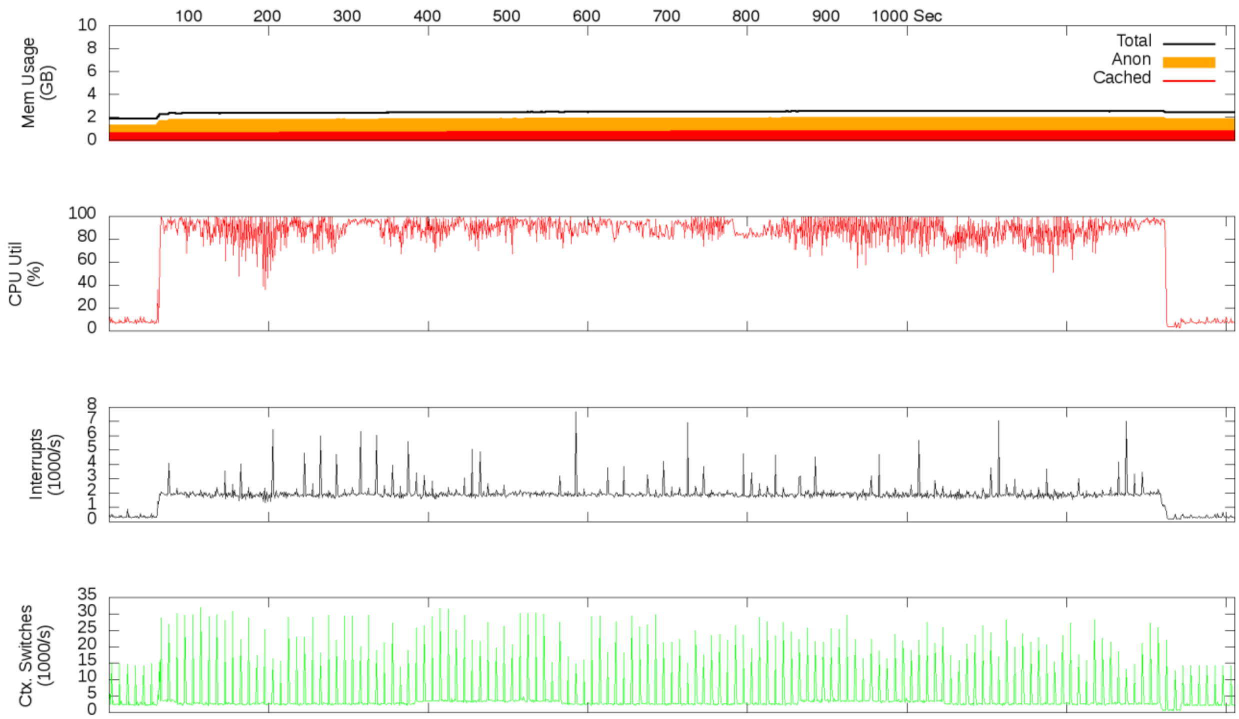


Figure 34. CPU Utilization on HPE Edgeline EL20 Face detection/recognition workloads. 1 instance- CPU is mostly busy.

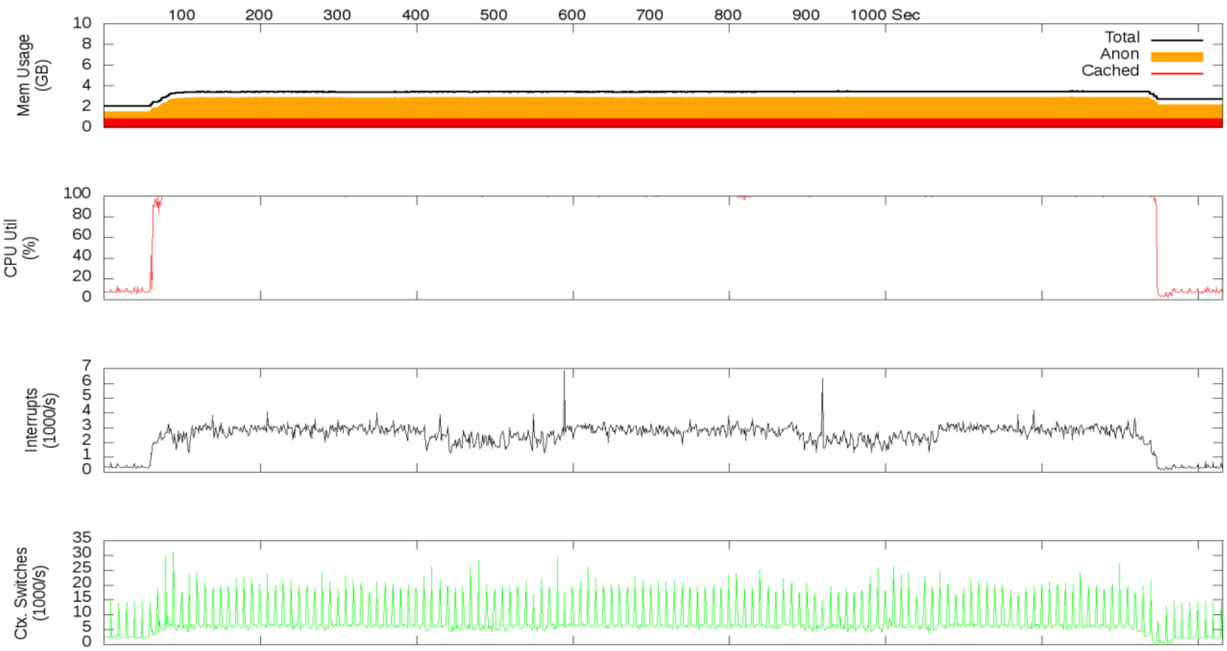


Figure 34. CPU Utilization on HPE Edgeline EL20 Face detection/recognition workloads. 3 instance- CPU is pinned at 100% utilization.

Error! Reference source not found. presents the results on an HPE ProLiant m510-16 core system. One compute cartridge constitutes an EL1000 Converged Edge IoT system for Edge Analytics. Four such compute cartridges can be used in the EL4000 Converged Edge IoT system for more CPU intensive Edge analytics use cases. The m510-16 core, when operating on one instance, produced 606 face records at 10% CPU utilization. This about the equivalent to the DL360 datacenter class Xeon E5 system. 30 instances resulted in average running time less than 22 minutes while produced 163 face records on average. As shown in *RecordsProduced* graph in Figure 29, while this is slightly lower than that of a DL360 Xeon E5 system, it is approximately equivalent.

The HPE ProLiant m510-16 core system can safely be used to perform face detection/recognition for up to 10 instances, and the m510-8 core variant for up to 5 instances as long as they are provided with sufficient amount of memory. Since the EL4000 can carry up to four m510-16 core cartridges or four m510-8 core cartridges, these workloads can run independently at the edge of the network (i.e. close to the camera's). It is therefore possible to expect at least 40 streams with the m510-16 cores and 20 streams with the m510-8 cores running the face detection/recognition algorithm. The result of face detection is typically a cropped snapshot of a person's face and in case of face recognition is an identifier (name) of a pre-trained face. Therefore there will be considerable savings in just retaining only those artifacts instead of the entire video stream.

For data storage on these video stream, there are two options: Frist the on-board flash storage on the compute node itself can be used up to 2 TB capacity or the data can be stored onto onboard/hot plug storage drives on the EL4000 itself, which is available as a separate SKU. Second, an external Video Management Solution (VMS) such as the product from Milestone can be leveraged. It includes a cost-effective bulk storage option, recording servers, and potentially backed by StoreVirtual VSA running on HPE Edgeline 4000 for higher capacity (provides ISCSI or NFS mount points). In case even more additional storage is required, an external D6000 JBOD can be connected to the HPE EL4000 using a Smart Array P441 on the PCIe slot.

Since it appeared from **Error! Reference source not found.** that a Xeon E5 2x8 core system was saturating at 12 instance streams when running the OS & the Media server software, no attempts were made at virtualizing the cores. All results indicated were ran on bare-metal.

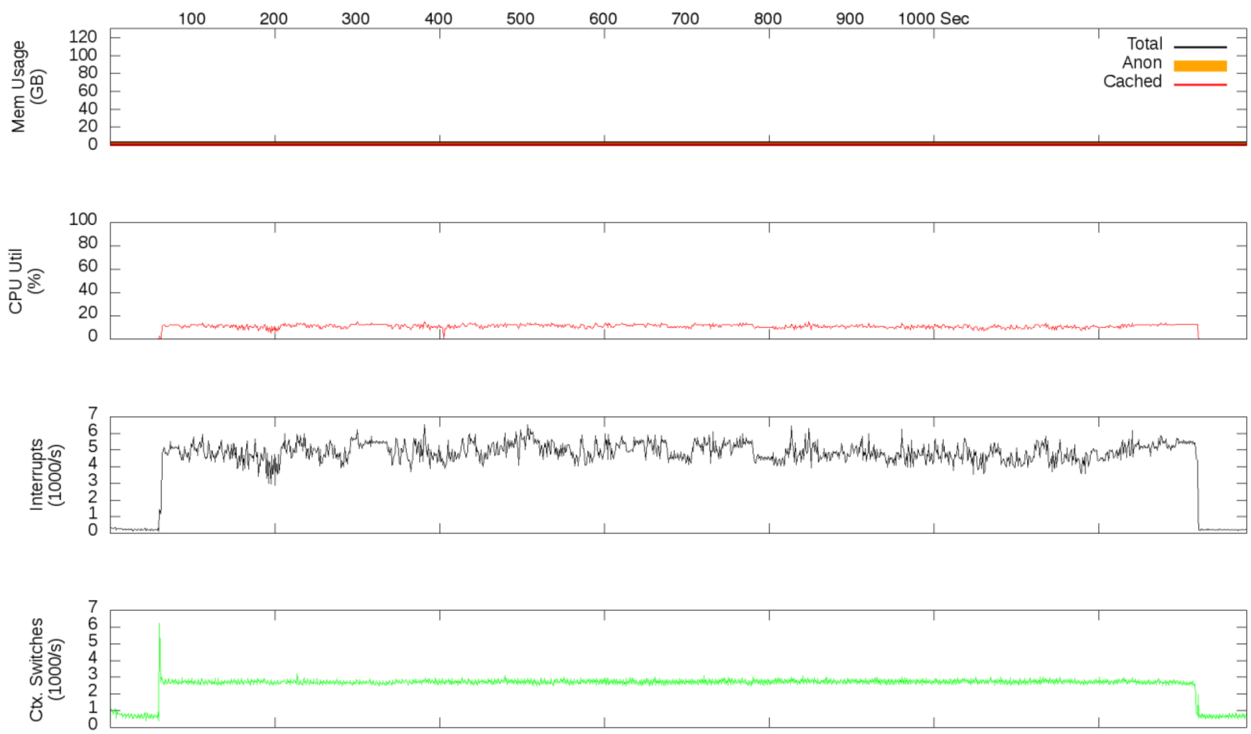


Figure 35. CPU CPU Utilization on Intel Broadwell-D 1587 v4 (HPE Edgeline Converged Edge System EL1000 and EL4000) for Face detection/recognition workloads. 1 instance- CPU is barely utilized.

.

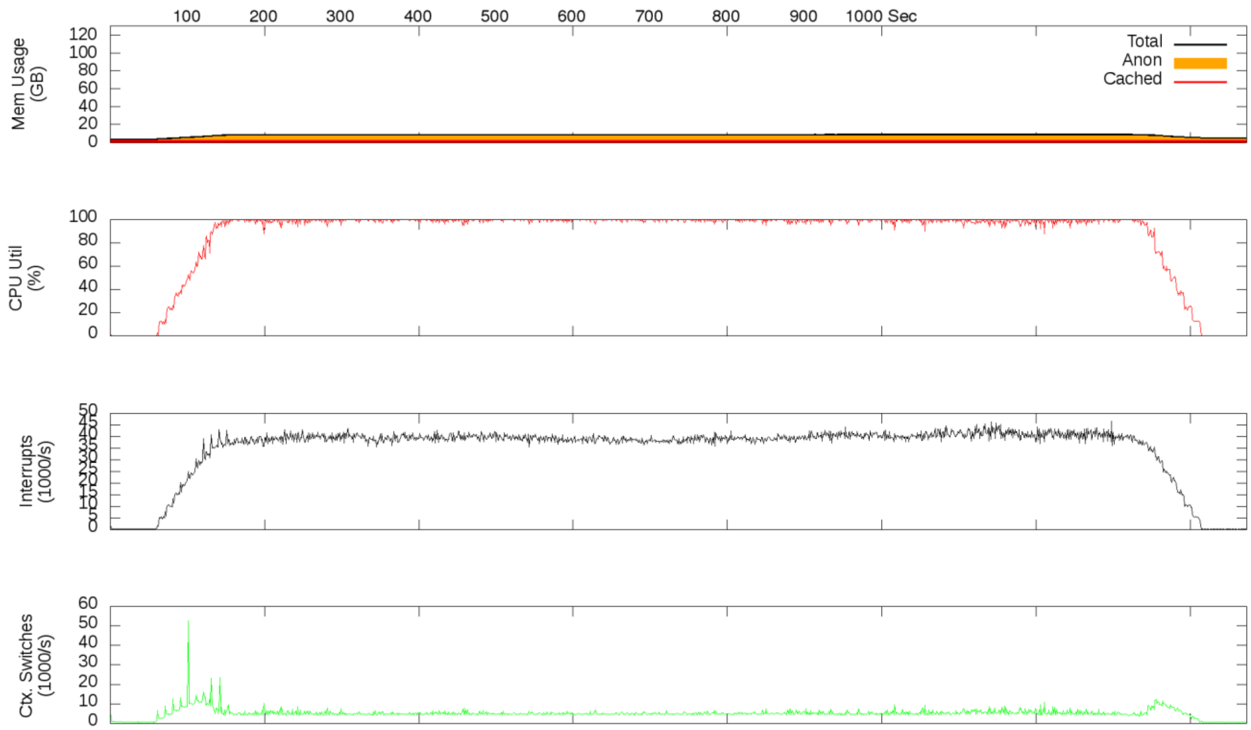


Figure 36. CPU CPU Utilization on Intel Broadwell-D 1587 v4 (HPE Edgeline Converged Edge System EL1000 and EL4000) for Face detection/recognition workloads. 10 instances- CPU is 100% utilized.

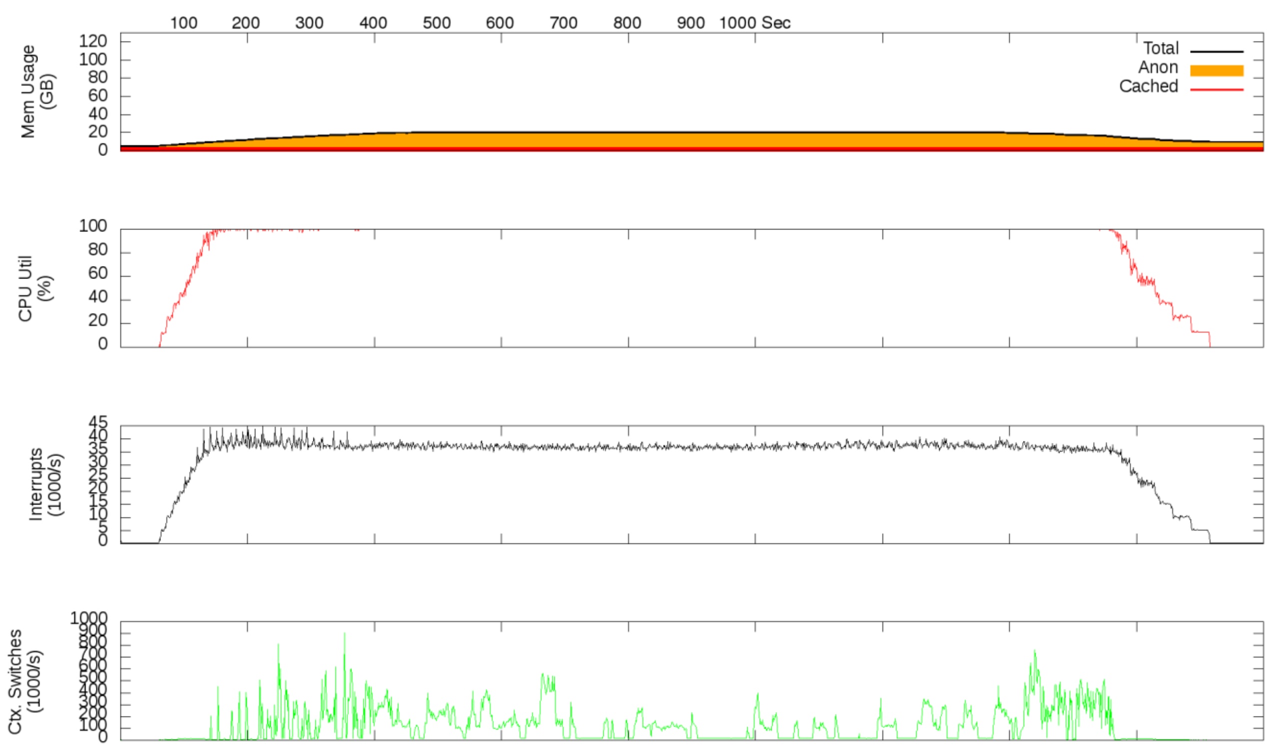


Figure 37. CPU CPU Utilization on Intel Broadwell-D 1587 v4 (HPE Edgeline Converged Edge System EL1000 and EL4000) for Face detection/recognition workloads. 30 instances- CPU is pinned at 100% utilization.

Table 3 shows the same results for 1 and 4 instances on m710x. One instance produced 563 face records. 9 instances had same average running time, but produced only 276 face records on average. As seen in Figure 29 records processed graph, this is lower than that achievable by the m510-16 core and D360. No more than 4 streams should be run on a HPE ProLiant m710x 4 core system for face detection/recognition workloads.

These results are along the expectation of 1 core per stream for face detection/recognition workloads for Media Server. The current algorithms for face detection are CPU centric and benefit from a larger number of SMP cores on a given system but only up to a certain extent. However, it was also determined that just having a higher number of cores alone is not sufficient since things do get bottlenecked nonetheless (CPU bound). Therefore a large number of mid-size cores densely packed may be a better alternative for video analytics workloads.

The HPE ProLiant m710x has an integrated GPU (IRIS Pro P580). This can provide high performance video transcoding and media accelerator capabilities. However this is not leveraged in the current distribution of the IDOL Media Server software because it is a CPU only workload.

Table 3. CPU Utilization on Intel Skylake-E3 1585 v5 (HPE Edgeline Converged Edge System EL1000 and EL4000) for Face detection/recognition workloads with varying instances.

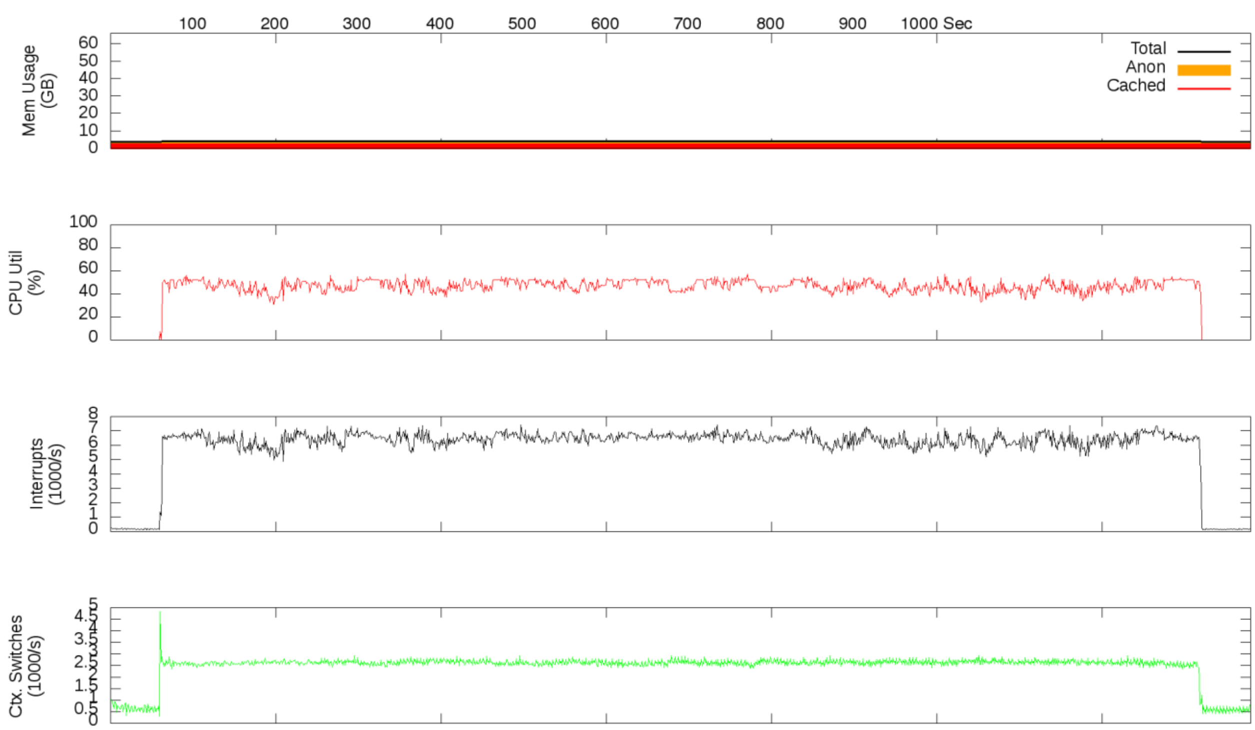


Figure 38. CPU Utilization on Intel Skylake-E3 1585 v5 (HPE Edgeline Converged Edge System EL1000 and EL4000) for Face detection/recognition workloads. 1 instance. CPU at 40% utilization.

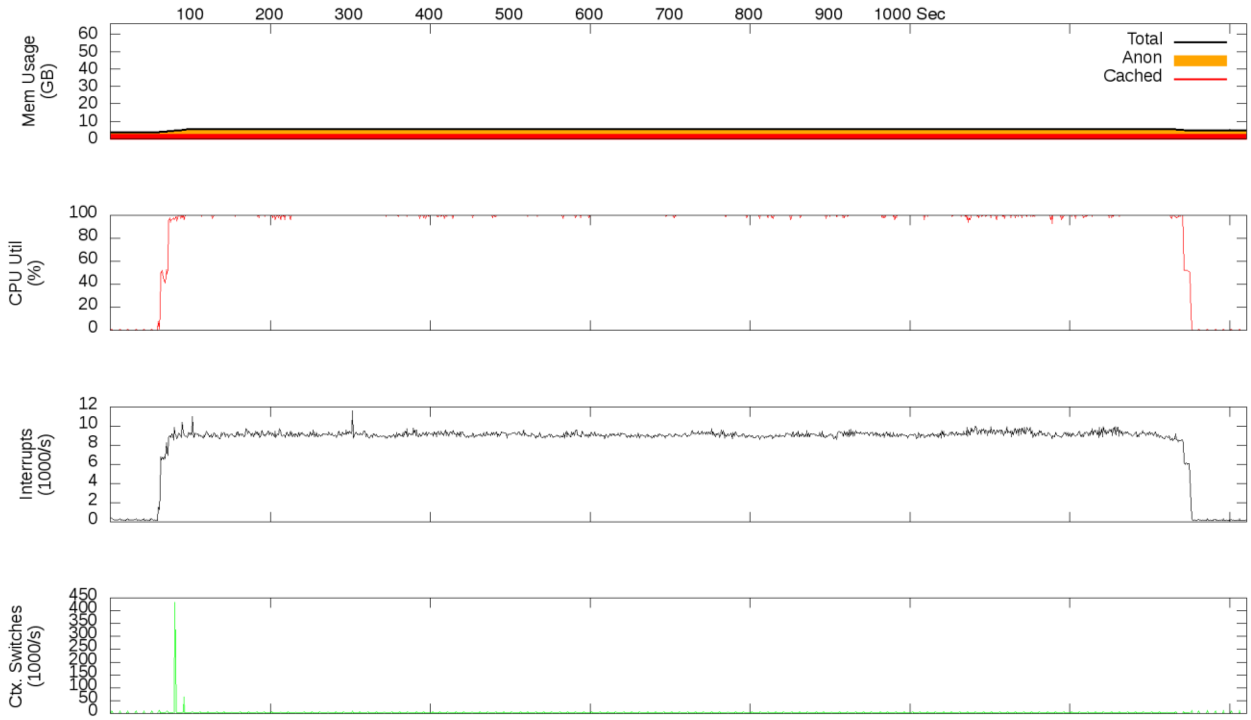


Figure 39. CPU Utilization on Intel Skylake-E3 1585 v5 (HPE Edgeline Converged Edge System EL1000 and EL4000) for Face detection/recognition workloads. 4 instances. CPU pinned at 100% utilization.

Table 4 lists the summary of expected performance of the HPE Edgeline IoT systems for face detection/recognition use case. The number of streams reported are with minimum frame loss. The number of streams for scene analysis and facial recognition are reported when analytic run-time was equal to real-time. This includes a fair number of dropped records, so should be treated as a best-case for all servers under test. The reference 2xIntel Xeon E5 2640 ran bare-metal without any virtual machines.

The limited CPU and memory resources of the EL10 and EL20 servers mean that it is not adequately equipped for running the face detection or the face recognition application without additional tuning (using number of face records produced as a metric). EL10 may be able to support one instance (had O(10) drops) and the EL20 should be able to support up to two instances with considerable frame loss for real-time performance. If that is relaxed for certain use cases, it is more likely.

The m510-8 core can run 5 face detection/face recognition streams whereas m510-16 core can run 10 streams. The m710x can run up to 4 streams without considerable dropped frames. Experiments on 2xIntel Xeon E5 2640 server (data center) indicate that using a “beefier” server does not automatically mean that more instances of face detection and face recognition can be supported. It can support up to 12 instances, similar to the m510 – 16core.

Our preliminary investigation suggests face recognition performs best with two physical CPU cores/instances (stream), such as two instances for m710x, eight instances for m510 – 16 core and eight instances for DL360. 16 core datacenter system may provide optimum performance.

The 1U tall EL4000 chassis can carry 4 independent m510 – 16 core compute nodes thereby providing at least 40 face detection/face recognition instances with similar accuracy comparable to the performance of the datacenter class HPE DL360 and with approximately 100W reduction in power consumption. The considerably smaller EL1000 can call 1 such compute node and can boast of data center class performance at the IoT Edge when performing workloads such as face detection/face recognition.

This exemplifies HPE's innovation in introducing the new Edgeline Servers as a new "product category" to enable the intelligent Edge – *more performance right at the point of data ingestion*. HPE Edgeline IoT systems can not only stand up to the requirements of a tough video analytic workload like face detection/recognition but also provide exceptional power/performance efficiency at the IoT Edge.

These results indicate that HPE has introduced data center-class computing on a product intended for the IoT Edge (same as or better performance). Assuming that equal investment were to be made for required compute infrastructure (either at the datacenter or at the Edge), investing in an Edgeline IoT system instead will enable the same compute without the overhead/cost of data movement to the datacenter (or cloud), thereby enabling cost efficiencies and exceptional value for money for HPE's customers.

Table 4: IoT Systems summary performance for Scene Analysis, Face recognition & ANPR (real-time).

SERVER	# CORES	INSTALLED MEMORY	#STREAMS FACE RECOGNITION MAX	#STREAMS FACE RECOGNITION OPTIMUM
HPE EL10	2	4 GB	~1	~1
HPE EL20	2	8 GB	~2	~1
HPE EL1000 (m510 – 8 core)	8	80 GB*	5	~4
HPE EL1000 (m510 – 16 core)	16	128 GB	10	~8
HPE EL1000 with m710x	4	64 GB	4	~2
HPE EL4000 (4x m510 – 8 core)	4x8	4x80GB*	20	~16
HPE EL4000 (4x m510-16 core)	4x16	4x128 GB	40	~32
HPE EL4000 (4x m710x – 4 core)	4x4	4x64 GB	16	~8
Reference HPE DL360 gen9	2x8	128 GB	12	~8

Summary

This solution demonstrates HPE's Media Server's capability in face detection and face recognition at the Edge and is written in the form of a How-To guide for ease of reproduction. Face detection and face recognition can be deployed quickly with minimal setup and configuration by combining HPE's Edgeline EL-X0 IoT systems and off-the-shelf security cameras, open source operating system and the IDOL Media Server software.

The Media Server video analytics can be further developed for business and security applications, such as marketing tools, security and monitoring systems, etc. Face detection/recognition at the Edge can be particularly valuable in large community events, such as where the ability to detect/recognize someone on a watch-list well in advance (near real-time) is critical to safety.

A solution architecture as described in this paper can provide much needed capability augmentation to law enforcement officers by reducing the amount of information they need to triage, is easy to setup and dismantle, provides data capture, analysis, and retention capability in a single SKU.

This technical white paper describes a solution design and testing performed in 2016 as a joint effort between HPE BEST Solutions architecture, IDOL Media Server R&D Team, HPE DCIG Labs & HPE WW Edgeline IoT GBU.

Implementing a Proof-of-Concept

As a matter of best practice for all deployments, HPE recommends implementing a proof-of-concept using a test environment that matches as closely as possible the planned production environment. In this way, appropriate performance and scalability characterizations can be obtained. For help with a proof-of-concept, contact an HPE Services representative (hpe.com/us/en/services/consulting.html) or your HPE partner.

Appendix A: Bill of Materials (Required)

Table 1a. Bill of Materials.

QTY	PART NUMBER	DESCRIPTION
1	847910-B21	HPE EL10 Intel Atom Dual Core 1.46GHz 4GB RAM 32GB SSD
1	847976-B21	HPE EL20 Intel i5 4300U Dual Core 1.9GHz 8GB RAM 64GB SSD 4POE
1		HPE EL1000 ProLiant m710x Intel Xeon E3-1585L 3.4GHz 16GB RAM 240GB SSD
	853995-B21	• HPE EL1000 1Gb System
	720478-B21	• HPE 500W Flex Slot Platinum Hot Plug Power Supply Kit
	833105-B21	• HPE ProLiant m710x 1P E3-1585Lv5 CPU Configure-to-order Server Cartridge
	863953-B21	• HPE Moonshot 16GB DDR4 DIMM
	866844-B21	• HPE Moonshot 240GB M.2 SATA 2242 Solid State Device
	867893-B21	• HPE m710x Front I/O Cable Kit
1		HPE EL4000 ProLiant m710x Intel Xeon E3-1585L 3.4GHz 16GB RAM 240GB SSD
	847535-B21	• HPE Edgeline EL4000 10GbE Switch System
	720479-B21	• HPE 800W Flex Slot Platinum Hot Plug Power Supply Kit
	833105-B21	• HPE ProLiant m710x 1P E3-1585Lv5 CPU Configure-to-order Server Cartridge
	863953-B21	• HPE Moonshot 16GB DDR4 DIMM
	866844-B21	• HPE Moonshot 240GB M.2 SATA 2242 Solid State Device

QTY	PART NUMBER	DESCRIPTION
	867893-B21	• HPEm710x Front I/O Cable Kit
1	Swann NHD820	NHD-820 1080p high definition network surveillance camera

	RECOMMENDED OPTIONS	
1	P441	Smart Array on the PCIe slot for HPE EL4000

Operating system
Ubuntu 16.01 LTS

Appendix B: Code Locations

B.1 licenseserver.cfg
https://github.hpe.com/jih-tsen-nat-lin/licenseServer_Script

B.2 shart.sh (for License Server)
https://github.hpe.com/jih-tsen-nat-lin/licenseServer_Script

B.3 stop.sh (for License Server)
https://github.hpe.com/jih-tsen-nat-lin/licenseServer_Script

B.4 mediaserver-activity.html
https://github.hpe.com/jih-tsen-nat-lin/MediaServer_Script

B.5 mediaserver.cfg (for Media Server)
https://github.hpe.com/jih-tsen-nat-lin/MediaServer_Script

B.6 stop.sh (for Media Server)
https://github.hpe.com/jih-tsen-nat-lin/MediaServer_Script

B.7 start.sh (for Media Server)
https://github.hpe.com/jih-tsen-nat-lin/MediaServer_Script

B.8 Face.bat
https://github.hpe.com/jih-tsen-nat-lin/Face/tree/master/Face

B.9 Face.cfg
https://github.hpe.com/jih-tsen-nat-lin/Face/tree/master/Face

B.10 FaceVideo.bat
https://github.hpe.com/jih-tsen-nat-lin/Face/tree/master/Video

B.11 Pre-recorded Video
https://github.hpe.com/jih-tsen-nat-lin/Face/tree/master/Video

Resources and additional links

HPE Converged Infrastructure Library
hpe.com/info/convergedinfrastructure

HPE Servers
hpe.com/servers

HPE Storage
hpe.com/storage

HPE Networking
hpe.com/networking

HPE Technology Consulting Services
hpe.com/us/en/services/consulting.html

HPE Edgeline

http://www8.hp.com/us/en/products/server-edgeline/product-detail.html?oid=1008670391

To help us improve our documents, please provide feedback at hpe.com/contact/feedback.

Sign up for updates

★ Rate this document

Hewlett Packard Enterprise

4AA4-xxxxENW, Month 20XX, Rev. #