Distributed Video Analytics across Edge and Cloud using ECHO

Aakash Khochare, Pushkara Ravindra, Siva P. Reddy, Yogesh Simmhan

Indian Institute of Science, Bangalore 560012 India
aakhochare@grads.cds.iisc.ac.in, simmhan@cds.iisc.ac.in

Abstract. Analytics over urban video streams is well suited for distributed computing across Edge, Fog and Cloud. Such streams are network intensive, making it is prohibitive to fully transfer them to the Cloud. Deep Neural Networks have achieved remarkable accuracy in image classification, but are computationally costly on just Edge devices. We propose ECHO as a big data platform to compose IoT dataflows and seamlessly distribute them across Edge and Cloud resources. In this demonstration, we illustrate the capabilities of ECHO for deploying several video analytics applications to support smart city use-cases.

1 Introduction

Internet of Things (IoT) is proliferating sensing and actuation devices in the physical space around us. Smart Cities, a manifestation of IoT, use analytics over streaming data sensed from city infrastructure to make management decisions on public utilities, traffic control, public safety, etc. While such processing has traditionally been limited to either local computation at the data source or centralized computation in the Cloud, analytics over video streams from thousands of cameras in a city challenges these two exclusive approaches.

The rise of deep neural network models is radically advancing computer vision algorithms to match humans in their ability to classify images. Such models can transform video streams into a urban meta-sensor to detect traffic movement, people density, pollution levels, safety violations, etc. But model inferencing is computationally costly, often requiring GPU acceleration, with model training even costlier. The typical approach of moving all the data to the Cloud for scalable analytics is bandwidth-intensive for video streams, and introduces network latencies during decision making. Further, such models are just one part of more complex applications that perform pre-processing and decision-making too.

The availability of distributed Edge and Fog devices as part of smart city deployments with substantial cumulative computing capacity can be leveraged in conjunction with Cloud resources for such urban video analytics applications. This requires an application platform to compose these dataflows, deploy them on distributed resources, and seamlessly manage their online orchestration. ECHO is one such platform that we have developed to address these needs [1].

In this demo, we showcase the ability of the ECHO platform to deploy and manage urban video analytics applications across Edge, Fog and Cloud resources.

2 The ECHO Platform

ECHO [1] is a platform for *Orchestration of Hybrid dataflows across Cloud and Edge* [1]. It allows the user to compose applications as a dataflow of tasks, with support for *hybrid data models* such as streams, micro-batches and files flowing through. An *application manager* deploys these tasks on distributed Edge, Fog and Cloud resources, using a *platform service* that runs on each device. A *scheduler* maps tasks to resources based on their availability maintained in a *registry*. Once deployed, the tasks are orchestrated by an *Apache NiFi* engine on each resource, which we extend for distributed execution. We also support delegation of parts of the dataflow to *external native engines* like *Apache Edgent* for Complex Event Processing (CEP), *Apache Storm* for distributed stream processing, and *Google TensorFlow* for deep learning. ECHO incorporates *dynamic adaptation* to remap tasks onto different resources, on-demand, to meet an application's current needs. In this demo, we extend ECHO with two novel features that we discuss next: *efficient scheduling* and managing *network asymmetry*.

Resource and Energy-aware Scheduling. We have earlier proposed the scheduling of a given dataflow onto Edge and Cloud resources as an optimization problem and solved it using a *Genetic Algorithm (GA) meta-heuristic* for an individual directed acyclic graph (DAG) [2]. Here, we extend this to support dataflows that arrive and depart continuously within the Edge, Fog and Cloud resources, and integrate the scheduler algorithm with ECHO. The optimization problem takes the tasks, their compute latencies, throughput, and energy footprint on different devices, and the network latency and bandwidth between devices as input. It enforces constraints to prevent the compute capacity for a single device from being saturated, and the energy usage on an edge device from draining its battery before it is recharged. The GA represents a task to device mapping as a chromosome, and uses mutations/crossovers to iteratively converge to a valid solution, with the goal of reducing the dataflow's latency.

We extend and use this for ECHO's adaptive scheduler. The app manager passes the user's DAG to the GA scheduler, along with the state of available resources from the registry. The GA reduces the capacity of each resource based on the tasks already running on them and runs incrementally to return the mapping of tasks to resources for the new DAG. The algorithm can later be rerun for adaptive re-balancing in case the dataflow's latency does not match the requirements. We propose to demonstrate the GA scheduler and its rebalancing.

Managing Network Asymmetry. ECHO's app manager invokes the REST platform service on each device to deploy and connect the dataflow tasks. However, this requires that the manager service on the Cloud be able to access the platform service on every resource over the Internet. Edge and Fog devices are often behind firewalls, making them inaccessible from the public Internet. Here, we mitigate this by extending the app manager to support asynchronous message passing to the platform service using an MQTT publish-subscribe broker. The platform service in each resource subscribes to a unique topic in the broker

[1] https://github.com/dream-lab/echo

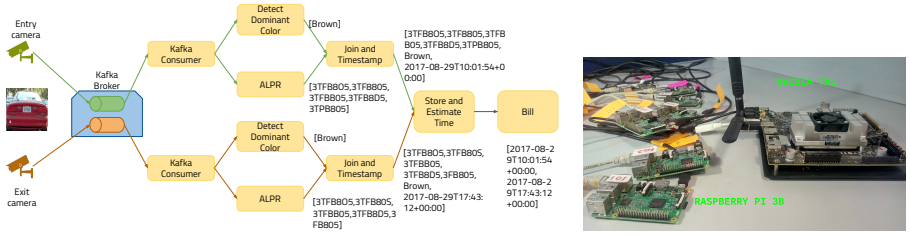


Fig. 1: ALPR dataflow for parking billing (L), and IoT Testbed Devices (R)

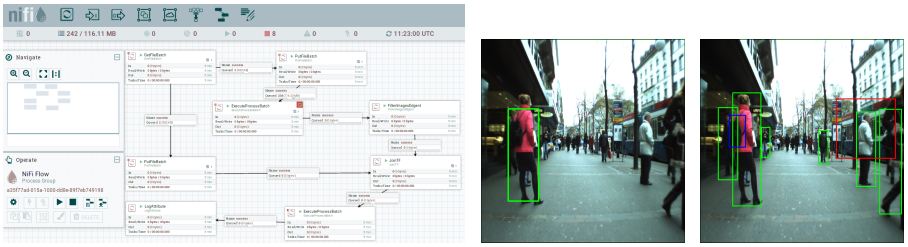
to which the manager publishes control messages for initiating a dataflow deployment. Each deployment session spawns a unique topic which is used to pass request and response JSON messages. Through this pattern, only the broker needs to be in a network location that is visible to all devices. Such a network asymmetry can also affect tasks on different devices that need to pass data items. Besides the existing support for both a push and a pull mechanism between two NiFi engines, we further support a similar broker-based model using Apache Kafka to for scalable transfer of large and fast data streams within the application. We will demonstrate support for such forms of network asymmetry.

3 Video Analytics Applications

We design two representative video analytics dataflows motivated by smart city applications, and demonstrate their execution using ECHO on Edge, Fog and Cloud resources. Our IoT testbed (Fig. 1(R)) where these applications are deployed consists of 12 Raspberry Pi 2B and 3B edge devices, an NVIDIA TX1 and a SoftIron ARM64 Fog servers, four Azure DS1 VMs in Microsoft's South India data center, and one Azure NC6 GPU VM in the US East data center [1].

Automatic Billing of Parking. Automatic License Plate Recognition (ALPR), a popular computer vision analytic, is used in applications like traffic enforcement, congesting pricing and automated toll collection. It is solved in two parts – the license plate region is first detected in an image, and then the characters are extracted from the region using Optical Character Recognition (OCR) [3]. Here, we design a dataflow that uses ALPR for automated time-based billing of vehicles across hundreds of parking lots in a city, when cameras are present at their entry and exit gates. We correlate the time at which a license plate enters and when it exits using ALPR, and bill them on exit. The challenge comes from the ALPR algorithm giving false positives. To address this, we also capture and store the detected color of the vehicle [2], besides the top n estimates of the license plate returned by an OpenALPR task [3] and the timestamp when

[2] https://github.com/fengsp/color-thief-py
[3] OpenALPR library, https://github.com/openalpr/openalpr



(a) YOLO dataflow deployed within NiFi (b) YOLO Tiny (c) YOLO Full

Fig. 2: YOLO dataflow and classified outputs from models

a vehicle enters. When a vehicle exits, these image analytics algorithms are re-run to detect its color along with the estimated plate numbers by ALPR, which are compared using a distance function with the details of vehicles that entered earlier. A match is found if the best distance score is above a threshold, and is used to determine the duration of parking and the bill. This dataflow that will be demonstrated is shown in Fig. 1(L).

Urban Scene Classification. Classification algorithms based on deep learning models associate bounding-boxes and tags to different entities in a given image. The outputs from such models can be used to detect situations of interest in urban environments, such as safety incidents, traffic violation, etc. YOLO is one such popular deep convolutional neural network for object detection that is trained and available on TensorFlow [4]. We demonstrate a novel use of YOLO using a two-level classification of urban scenes in conjunction with an *Apache Edgent CEP engine*, as described in [1]. Running YOLO on a full resolution image frame (608×608) is computationally costly, and TensorFlow achieves a frame-rate of only $1/sec$ even with an NVIDIA K80 GPU. Instead, we use an additional *tiny* YOLO model that operates on a scaled-down image on the Edge or Fog device, and if any interesting tags are detected, forwards a full-resolution video segment to a GPU VM on the Cloud where the full model runs for accurate classification. This also illustrates the use of hybrid engines, TensorFlow and Edgent, for execution within ECHO. A screenshot of the dataflow in NiFi, along with a sample frame classification from the models is shown in Fig. 2.

References

1. P. Ravindra, A. Khochare, S. P. Reddy, S. Sharma, P. Varshney, and Y. Simmhan, "ECHO: An Adaptive Orchestration Platform for Hybrid Dataflows across Cloud and Edge," in *ICSOC*, 2017, To appear.
2. R. Ghosh and Y. Simmhan, "Distributed Scheduling of Event Analytics across Edge and Cloud," *ACM Transactions on Cyber-Physical Systems*, 2017, To Appear.
3. S. Ozbay and E. Ercelebi, "Automatic vehicle identification by plate recognition," *World Academy of Science, Engineering and Technology*, vol. 9, no. 41, 2005.
4. J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *CoRR*, no. 1612.08242, 2016.