

# PAIGE: Towards a Hybrid-Edge Design for Privacy-Preserving Intelligent Personal Assistants

Yilei Liang  
yilei.liang@kcl.ac.uk  
King's College London  
London

Dan O'Keeffe  
Daniel.OKeeffe@rhul.ac.uk  
Royal Holloway University of London  
Egham

Nishanth Sastry  
nishanth.sastry@kcl.ac.uk  
King's College London  
London

## ABSTRACT

Intelligent Personal Assistants (IPAs) such as Apple's Siri, Google Now, and Amazon Alexa are becoming an increasingly important class of web-service application. In contrast to keyword-oriented web search, IPAs provide a rich query interface that enables user interaction through images, audio, and natural language queries. However, supporting this interface involves compute-intensive machine-learning inference. To achieve acceptable performance, ML-driven IPAs increasingly depend on specialized hardware accelerators (e.g. GPUs, FPGAs or TPUs), increasing costs for IPA service providers. For end-users, IPAs also present considerable privacy risks given the sensitive nature of the data they capture.

In this paper, we present Privacy Preserving Intelligent Personal Assistant at the EdGE (PAIGE), a hybrid edge-cloud architecture for privacy-preserving Intelligent Personal Assistants. PAIGE's design is founded on the assumption that recent advances in low-cost hardware for machine-learning inference offer an opportunity to offload compute-intensive IPA ML tasks to the network edge. To allow privacy-preserving access to large IPA databases for less compute-intensive pre-processed queries, PAIGE leverages trusted execution environments at the server side. PAIGE's hybrid design allows privacy-preserving hardware acceleration of compute-intensive tasks, while avoiding the need to move potentially large IPA question-answering databases to the edge. As a step towards realising PAIGE, we present a first systematic performance evaluation of existing edge accelerator hardware platforms for a subset of IPA workloads, and show they offer a competitive alternative to existing datacenter alternatives.

## CCS CONCEPTS

- **Computer systems organization** → **Distributed architectures**;
- **Computing methodologies** → **Machine learning**.

## KEYWORDS

Intelligent Personal Assistants, Edge computing, Trusted execution environments

## ACM Reference Format:

Yilei Liang, Dan O'Keeffe, and Nishanth Sastry. 2020. PAIGE: Towards a Hybrid-Edge Design for Privacy-Preserving Intelligent Personal Assistants. In *3rd International Workshop on Edge Systems, Analytics and Networking (EdgeSys '20)*, April 27, 2020, Heraklion, Greece. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3378679.3394536>

## 1 INTRODUCTION

Modern web applications are increasingly dependent on sophisticated machine learning algorithms. This trend is epitomised by the evolution of keyword-oriented web search services into Intelligent Personal Assistants (IPAs), such as Apple's Siri, Google Now, and Microsoft Cortana. IPAs support complex user queries involving image, voice and natural language processing, which in turn depend on computationally intensive machine learning inference.

IPAs and similar advanced machine-learning workloads place considerable stress on traditional data centers, and are driving a trend in datacenter design towards more heterogeneous compute resources (e.g. GPUs, FPGAs, and TPUs). This trend is exemplified by the growing availability of such specialized hardware in major cloud computing platforms. Indeed, recent work has demonstrated that data centers comprised primarily of general purpose CPUs are inefficient for IPA workloads in terms of raw throughput, latency and energy efficiency, and that specialized hardware offers a more cost-effective and scalable solution [12].

Fundamentally however modern cloud-centric IPA designs suffer from two major drawbacks. First, the additional access latency and bandwidth required to ship user data to the cloud is often considerable. Second, they introduce major privacy concerns, as potentially sensitive user data must be shipped to the cloud. For example, in 2019, a reviewer of confidential audio recordings made by Google Assistant leaked those recordings to a Belgian news organisation [11, 23]. Emerging approaches to confidential cloud computing based on hardware-enforced enclaves promise a solution [14], but state-of-the-art enclaves are not yet available on cloud GPUs, FPGAs or TPUs.

Recently, edge computing architectures have received much attention as a potential alternative to cloud-centric systems. Edge computing processes user data with resources available at the network edge, potentially alleviating the aforementioned drawbacks of the cloud. In the context of IPAs however, the performance tradeoffs and cost-effectiveness of an edge computing architecture remain relatively unexplored. In particular, the compute intensive nature of IPA query processing presents a major challenge for edge-centric IPAs, which typically assume relatively resource constrained edge devices. Furthermore, many IPA queries require access to large

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*EdgeSys '20*, April 27, 2020, Heraklion, Greece

© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-7132-2/20/04...\$15.00

<https://doi.org/10.1145/3378679.3394536>

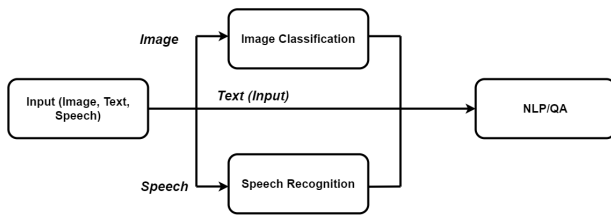


Figure 1: IPA workflow

databases, e.g. for question-answering (QA), that would be difficult to migrate completely to the edge.

We observe that there has been a recent proliferation of low cost machine learning accelerators designed for edge environments [10, 13, 17]. These devices typically sacrifice some performance in comparison to datacenter accelerators, but are cheap (~\$100) and energy efficient. To date however there has been no systematic study as to whether such devices could support an edge IPA design competitive with cloud-centric IPAs for real-world IPA workloads.

We propose PAIGE<sup>1</sup>, a hybrid edge-cloud architecture to support privacy-preserving Intelligent Personal Assistants. PAIGEs design offloads compute-intensive tasks to low-cost machine learning accelerators deployed at the network edge and under user control. Edge-based ML inference preserves end-user privacy for ML tasks, while edge accelerators ensure acceptable performance. To support privacy-preserving access to large IPA question-answering databases, PAIGE forwards pre-processed queries to the cloud, where they are executed securely inside an Intel SGX enclave [8, 14] against an encrypted database. Since these pre-processed queries are typically less compute-intensive, the inability of enclaves to protect accelerator computation (e.g. on GPUs) is less relevant.

As an initial step towards validation PAIGE’s design, we present evidence of the suitability of edge ML accelerators to underpin an edge based IPA architecture. We perform a comprehensive analysis of performance and accuracy trade-offs for a range of edge accelerators with respect to a variety of different ML model architectures. Our results indicate that edge accelerators offer a competitive alternative to datacenter devices in terms of inference latency and energy consumption. We anticipate PAIGE’s design will be of interest to new and existing IPA service providers, privacy conscious users of IPA services, and even organisations wishing to deploy their own privacy-preserving IPA services.

## 2 BACKGROUND

### 2.1 Intelligent Personal Assistants (IPAs)

IPAs are an emerging class of ML-driven application that process user queries containing voice, image, and other contextual information (e.g. location) in order to answer user questions, make recommendations, and even perform actions. Initially, user interaction with IPAs was primarily through mobile devices (e.g. Apple Siri, Google Now, Microsoft Cortana), but with the growth of the IoT IPA interaction is increasingly occurring through smart-home devices (e.g. Amazon Alexa, Google Home).

Figure 1 shows a typical workflow for an IPA application, and illustrates its dependence on a variety of ML driven components, including image analysis, automated speech recognition (ASR), and natural language processing (NLP). After queries have been pre-processed by one or more ML components, the IPA system may execute a database query against a question-answering system, or perform some automated action.

As argued in previous work [12], IPA workloads present a severe scalability challenge for traditional datacenter design, as they require substantial compute resources to service each request. In comparison to text-based web search workloads, IPA workloads are instead more efficiently served using specialised accelerators (e.g. GPUs, FPGAs, TPUs) instead of multi-core servers. Furthermore, the range of ML tasks required to support an IPA application means that different accelerators may be more cost-effective for different parts of the query pipeline (e.g. FPGAs for NLP vs. GPUs for images).

### 2.2 Privacy-Preserving Cloud ML

Sending IPA queries to the cloud not only introduces privacy concerns but also increases latency, which may affect usability of IPA queries which are latency-sensitive. To counteract the privacy implications, several techniques have been proposed for performing privacy-preserving computation over encrypted data. In particular, hardware-enforced trusted execution environments (TEEs) such as Intel SGX *enclaves* [1, 3, 8, 14] allow applications to ensure confidentiality and integrity, even if the OS, hypervisor or BIOS are compromised. Enclaves also protect against attackers with physical access, assuming the CPU package is not breached.

Enclave code and data reside in a region of protected physical memory inaccessible to non-enclave code. While cache-resident, enclave code and data are guarded by CPU access controls. An on-chip memory encryption engine (MEE) encrypts and decrypts cache lines written to and fetched from DRAM. Finally, a remote party can verify the integrity of an enclave using an attestation protocol.

In the context of machine-learning applications, a major drawback of enclaves is they are not yet available on cloud GPUs, FPGAs or TPUs. ML models that benefit from these accelerators must therefore pay a substantial performance penalty when executing inside CPU enclaves. Slalom [24] combines enclaves with a cryptographic blinding technique to delegate all linear layers in a deep neural network to a GPU. Although their technique improves performance in comparison to an enclave only technique, it still remains much slower than evaluating on a GPU with no security guarantees.

### 2.3 Edge ML

As an alternative to cloud-centric systems, *edge computing* pushes computation closer to the user at the edge of the network, potentially alleviating datacenter network access latency and bandwidth concerns [20, 21]. Furthermore, some edge computing architectures perform all computation on user-owned devices, mitigating many privacy risks (e.g. [5] [6], [18]).

A key challenge for edge computing architectures in the context of compute-intensive ML-driven applications such as IPAs is that edge resources are typically less powerful than those available in

<sup>1</sup>Privacy-preserving intelligent personal Assistant at the EdGe

a datacenter. To date, several approaches have been proposed to alleviate this issue.

**ML Model Compression:** One approach to executing ML models at the edge is to sacrifice some of the accuracy provided by state-of-the-art ML models in order to reduce model size and inference latency. A variety of techniques have been proposed to achieve this, including quantization, pruning and compression [7, 16]. However, on CPUs with the computational capacity typical of many edge devices or gateways (e.g. a Raspberry Pi), achieving acceptable accuracy and latency is still challenging.

**Edge Accelerators:** A complementary approach to model compression is to exploit specialized hardware for edge ML inference. Similar to datacenter environments, where GPUs, FPGAs and TPUs have been proposed to accelerate machine learning applications, a variety of accelerators for edge environments are now commercially available [10, 13, 17]. Although not as powerful as their datacenter counterparts, these devices are inexpensive (100\$ vs. 1K\$+) and typically highly energy-efficient. Their low cost makes them viable for consumer devices (e.g. in smart homes), but to date there has been no comprehensive study as to their performance and cost-effectiveness as part of an edge-based IPA architecture.

**Hybrid/Distributed Edge:** Instead of relying on individual edge devices, several researchers have suggested harnessing the combined resources of multiple edge devices. For example, *hybrid edge* architectures combine user-owned devices with resources available at an edge gateway or other ISP-owned infrastructure, and perhaps even the cloud [5, 6]. However, this approach offers limited benefits in terms of protecting user privacy. A *distributed edge* architecture combines instead the resources of multiple user-owned devices [18]. Although this architecture is better for user privacy, existing solutions do not explicitly target ML heavy workloads such as IPAs (e.g. by incorporating edge accelerators or model compression). Furthermore, in some cases the output of the ML components must subsequently be used to query a database (e.g. as part of a question-answering system). If the database is very large, storing it at the edge may not be an option.

**Federated learning:** Finally, federated learning [4, 15, 22] protects user privacy during training of machine learning models. Federated learning aggregates gradient updates locally before uploading updates or partial models to the cloud where they can be combined in a privacy preserving manner. Instead of protecting privacy during training, our goal instead is to perform privacy-preserving inference using pre-trained models. Federated learning could however be used as a complementary technique to create models without compromising privacy.

## 2.4 Edge IPA Requirements

In summary, although a combination of some or all of the above approaches to Edge ML are potentially applicable to IPA systems, there does not yet exist a systematic evaluation as to the feasibility of an Edge IPA architecture in comparison to a cloud-based design. Concretely, we define the following requirements for an Edge IPA architecture:

- (1) **Responsiveness (R1):** User IPA queries must execute with low latency (e.g. sub-second).

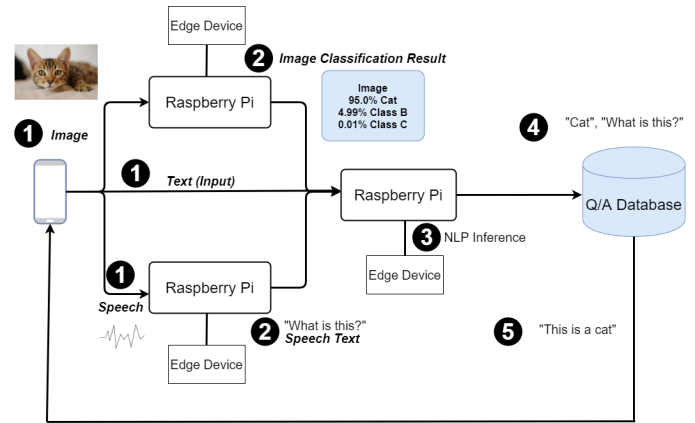


Figure 2: Edge IPA workflow

- (2) **Energy-efficiency (R2):** To ensure cost-effectiveness, energy expenditure of both user and server side components should not be prohibitive.
- (3) **User privacy (R3):** End users should not be required to reveal their queries or other sensitive data to the cloud.

## 3 DESIGN

Based on the requirements outlined in §2.4, we present PAIGE, a hybrid-edge architecture that enables privacy-preserving, cost-effective but responsive Intelligent Personal Assistants. PAIGE's design differs in two main aspects from existing IPAs. *Edge accelerators* execute compute-intensive ML tasks at the edge, preserving user privacy while retaining good performance. *Trusted enclaves* execute pre-processed user queries against encrypted databases at the server-side, avoiding the need to store large databases at the edge while still preserving user privacy.

**PAIGE Overview:** Fig. 2 gives a high-level overview of PAIGE's design. Users submit queries either using voice commands or through a mobile device, and may contain additional image or text components (Step ①). Queries are then offloaded by PAIGE to one or more local compute accelerators (Step ②). These accelerators are typically attached to e.g. one or more gateway devices (e.g. a Raspberry Pi) under the user's control. After pre-processing, the output of the ML models may result in an immediate pre-programmed action (Step ③). Alternatively, the pre-processed query may be encrypted and forwarded to a secure question-answering database (SecQADB) located in the cloud. Queries to the database are executed in a privacy-preserving manner inside a trusted execution environment (e.g. Intel SGX enclaves) (Step ④). PAIGE does not currently mandate a specific enclave-based database, and is compatible with existing proposals [2, 19]. Finally, PAIGE returns the result to the user (Step ⑤).

PAIGE's design offers a balance between privacy and performance. However, we note that for pre-processed queries that must be submitted to the SecQADB, a round trip to the datacenter is still required. We discuss options for addressing this limitation, including caching parts of the SecQADB near the edge, in §5.

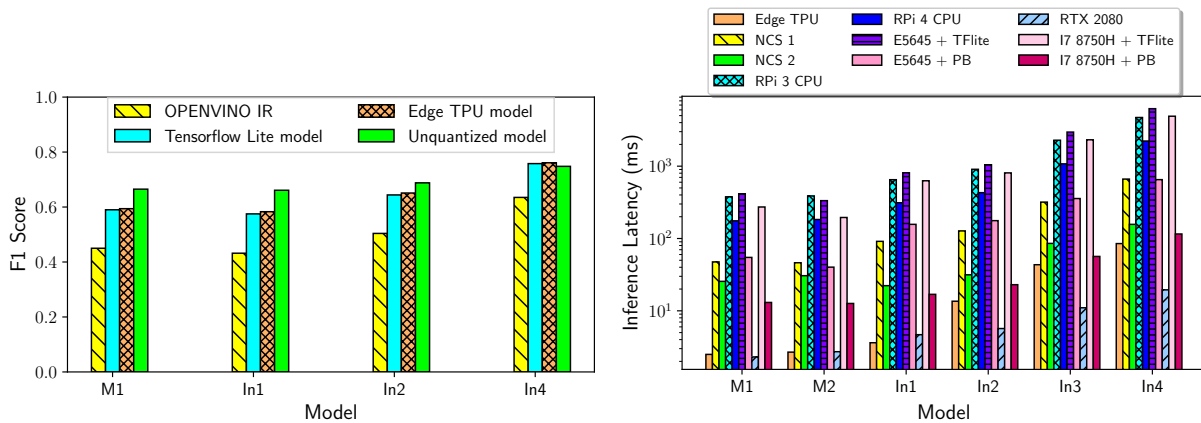


Figure 3: (a) Accuracy and (b) Inference latency across different platforms and for different ML models (Mobilenet V1-2 (M1-2) and Inception V1-4 (In1-4)).

## 4 EDGE IPA EVALUATION

As a step towards demonstrating the feasibility PAIGE’s design, we next present an evaluation of the accuracy, inference latency, and energy-efficiency of a variety of different edge hardware platforms (EdgeTPU, Neural Compute Stick 1 & 2, CPU and GPU) and ML models (MobileNet V1-2 and Inception V1-4) relevant to IPA systems.

Our evaluation aims to answer the following questions: (i) How accurate are models for edge platforms? (ii) Which platform has the lowest inference latency? (iii) How cost-effective are edge platforms in terms of energy efficiency?

### 4.1 Experimental setup

**Hardware platforms:** We deploy our benchmark on Raspberry Pi 4 model B (4G RAM, Quad core ARM v8 64-bit CPU) with USB plugin accelerators (EdgeTPU and NCS 1 & 2). For comparison, we also deploy our benchmark on a Laptop with 6 physical cores CPU (I7 8750H with 2.20 GHz base frequency and 4.10 GHz max turbo frequency), RTX 2080 MAX-Q GPU (2944 shading units, 8GB GDDR6) and 16GB RAM, and a server with Intel Xeon E5645 (6 physical cores with 2.20 GHz base frequency and 2.67 GHz max turbo frequency) and 48 GB RAM.

**Workload:** As a representative workload, we evaluate the performance of image classification. We leave evaluation of other ML workloads relevant to IPAs (e.g. for ASR or NLP) to future work. The dataset that we used for the evaluation benchmark is ImageNet 2012 Evaluation set (ILSVRC2012\_img\_val)[9]. The models that we use are from the official TensorFlow pre-trained model list<sup>2</sup> converted as appropriate to different formats (e.g. IR for Neural Compute Stick, TF Lite or EdgeTPU TF Lite).

### 4.2 Model Accuracy

For our first experiment we evaluate the accuracy of models for different edge platforms. As the models are quantized for the edge accelerators (EdgeTPU and NCS 1 & 2), there is a trade-off between

accuracy and inference time. For each platform, we record the results in a confusion matrix and from this calculate the F1 Score. We ignore Mobilenet V2 and Inception V3 as their official pre-trained models (as quantized for the EdgeTPU and NCS) are buggy and we have been unable to obtain an accurate result. The results for the remaining models are shown in Fig. 3(a).

We find that, as expected, unquantized models do typically have better performance than quantized models. EdgeTPU exhibits a 7.1% drop in F1 score for Mobilenet V1, 7.8% for Inception V1 and 3.8% for Inception V2 in comparison to unquantized models. OpenVINO IR models performs worse than EdgeTPU, with a 21.5% drop in F1 score for Mobilenet V1, 22.9% for Inception V1 and 18% for Inception V2. One exception to the overall trend is Inception V4, for which all platforms except OpenVINO IR (for Neural Compute Stick) have a similar F1 Score. We believe that the quantization from 32 bits float to 8 bits integer will not impact the accuracy that much for large models. Overall, the results show that especially for Edge TPUs, the accuracy reduction is comparable to that achieved on other more sophisticated hardware.

### 4.3 Inference Latency

Our second experiment evaluates the inference latency of the different platforms and models. As mentioned above, we now have smaller models for the edge devices compared to the traditional model. For this benchmark, we developed a test harness to classify a set of images and monitor the speed of processing of the image. We record the total inference time for each experiment and from this compute the average inference latency per image. We define two different operation modes (i) *default* mode – a single TensorFlow session is used for the whole experiment, and (ii) *request* mode – a new TensorFlow session is created for each image. We include results for request mode to capture the potential performance overhead of switching between different models sharing the same device.

Fig. 3 shows the average inference latency for a single image. The results show that edge devices are far superior to server computers and recent CPUs in default mode at the image classification machine learning task – even close to recent GPUs for small models. For

<sup>2</sup>[https://www.tensorflow.org/lite/guide/hosted\\_models?hl=en](https://www.tensorflow.org/lite/guide/hosted_models?hl=en)

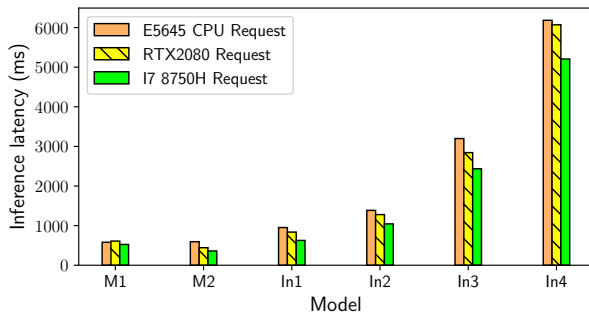


Figure 4: Inference latency in request mode

example, for MobileNet V2 the EdgeTPU has an average latency of 2.685ms in comparison to 2.723ms for the GPU and 12.647ms for the I7 8750H, the best performing CPU. For larger models such as Inception V3 and Inception V4, GPUs achieve better performance, with for example the EdgeTPU resulting in 75% and 77% slowdowns in comparison. In request mode, both the server and laptop are much slower than other devices as there is a substantial overhead in initializing a new session.

We also found that Tensorflow Lite models have worse performance on server/laptop compared to Raspberry Pi. This is because Tensorflow Lite operators are optimized for the ARM architecture so that performance on the X86/64 architecture is not as good as the ARM-based Raspberry Pi.

It is remarkable that edge TPUs have such a competitive performance for well-tuned image recognition tasks. However, this does not mean that there is a comparable difference in other ML tasks, and more evaluation is needed to confirm the generality of this result.

#### 4.4 Energy Benchmark

In this benchmark, we evaluate how much energy each platform requires to perform inference with comparable accuracy (F1 score). As described in previous sections, smaller models imply some accuracy will be sacrificed. We therefore evaluate the power consumption for models that have a similar F1 score rather than the same model. For all our energy experiments, we record the difference in power between idle state and inference state. To record energy consumption, we use USB power measurement tools (JUWEI UM24C Power Meter) for the Edge Devices, Intel Power Gadget for the Laptop CPU, and Nvidia-smi for the GPU.

In Figures 5 and 6, we demonstrate the power consumption for image classification on EdgeTPU and Laptop in default mode. In all cases the corresponding F1 Score is between 0.64 and 0.66. We find that edge devices are substantially more energy-efficient than both CPUs and GPUs for the same F1 score, with for example the GPU consuming 29x more energy per image in comparison to the EdgeTPU. Although the maximum power consumption in request mode is lower than default mode, the overall energy consumption is still higher than default mode.

### 5 DISCUSSION

The main claim of the paper is that low-powered Raspberry Pis, equipped with special purpose Tensor Processing Units (TPUs), can provide similar or superior performance to traditional servers with

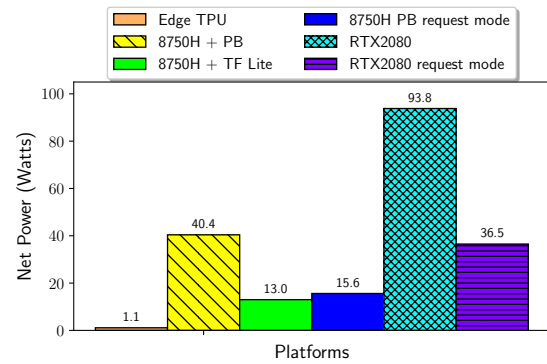


Figure 5: Power consumption increase for different platforms (Net Power = Avg. Power - Idle Power).

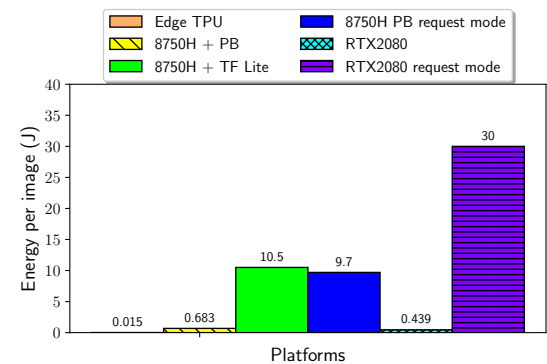


Figure 6: Energy consumption per image

GPUs. This implies that by deploying special purpose hardware such as TPUs, edge-based architectures and solutions can be extremely competitive, and yield privacy benefits, in comparison to centralised counterparts.

The advantage of low-powered RPi+EdgeTPU-based solutions in terms of lower energy consumption is expected, but to our surprise, we also found that inference can be much faster with TPUs rather than with GPUs. We believe this is because recompiling machine learning models to run on the EdgeTPU greatly simplifies the models, allowing them to be executed faster. We show that this simplification comes at a cost of slightly lower accuracy, precision and recall. However, we need to understand this trade-off better: under what conditions, and for what kinds of models, will the EdgeTPU version be a “better” option than the traditional GPU-based version?

Leading on from this initial effort, one research thread to explore is whether the specialised support for eight-bit tensor multiplication offered by devices such as the EdgeTPU can offer a general purpose primitive that can be exploited more widely in other kinds of systems, and how this might allow more sophisticated edge systems. Widening further, it would be interesting to study whether we can build “good enough” approximations at the edge to today’s centralised systems, and whether/how this may allow us to realise novel promises of edge computing, such as increased privacy.

Improved inference speed gives more time for other sub-operations that may be involved in delivering some end to end functionality (e.g., quicker image recognition may allow higher network latencies,

which may in turn enable more centralised processing, or processing over slower networks [5]. However, more research is needed on how inference speedup can alter latency budgets of different application workflows.

## 5.1 Open issues

This is still initial work. Many issues are as yet unaddressed: We have only tested a selected set of *image recognition* models, and our results so far indicate that edge-based architectures using EdgeTPUs can be superior to traditional servers with GPUs. However, it remains to be seen whether the same holds true for other machine learning applications. Even for the IPA applications we target, a number of other kinds of ML models are needed (for example, it could be that voice recognition models see a huge fall in F1-score/accuracy when moved to EdgeTPUs).

Assuming that all components of an IPA can be realised effectively and efficiently on RPIs with TPUs attached, we will still need a distributed edge architecture, as each TPU will be specialised for a specific model, and will not be able to run the entire gamut of ML operations for IPA queries (e.g., one node will run voice recognition; the next node in the chain will have to run question understanding; the next node will have to actually answer the query, and so on). An important piece of “pure systems” work (as opposed to ML-based systems work) remains to be done, on how to design such a distributed architecture efficiently.

Currently, we assume that pre-processed queries that require access to a question answering database will be forwarded to the cloud and executed inside an enclave. Another interesting line of research is whether a distributed database architecture could help to reduce the additional latency needed to access the cloud (e.g. by locating subsets of the database content inside an enclave nearer to the edge).

Finally, as mentioned above, we do not yet understand the precise nature of the trade-off between ML performance (F1-score, accuracy etc) and increased speed of inference. Therefore, a big open issue is: under what circumstances will this trade-off work. Is there a “pareto frontier” between ML performance and inference speedup that can be explored, whereby some ML performance can be sacrificed for faster inference, but still not degrade the performance of the whole system?

## 6 CONCLUSION

We present PAIGE, a hybrid edge-cloud architecture for privacy-preserving IPAs. PAIGE offloads compute intensive IPA ML tasks to the network edge, exploiting recent advances in low-cost hardware for machine-learning inference to improve performance and protect user privacy. To support privacy preserving access to large IPA databases for pre-processed IPA queries, PAIGE leverages hardware enclaves (e.g. Intel SGX) at the server side. As an initial step towards validating PAIGE’s design, we present a first systematic performance evaluation of existing edge accelerator hardware platforms for a subset of IPA workloads and for a variety of ML model architectures. Our results indicate that edge accelerators offer a competitive alternative to datacenter devices in terms of inference latency and energy consumption. Finally, we discuss several open

issues that PAIGE must address to make privacy-preserving high-performance IPAs a reality.

## ACKNOWLEDGEMENTS

The authors wish to acknowledge funding from the King’s Undergraduate Research Fellowship Scheme and a gift from Vodafone R&D for work on Mobile Edge Computation in 5G. The research leading to these results also relates in part to the EDGE-UCATE project. This replicator project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 731677.

## REFERENCES

- [1] Sergej Arnaudov, Bohdan Trach, Franz Gregor, Thomas Knauth, Andre Martin, Christian Priebe, Joshua Lind, Divya Muthukumar, Dan O’keeffe, Mark L Stillwell, et al. SCONE: Secure Linux containers with Intel SGX. In *OSDI ’16*.
- [2] Maurice Bailleu, Jörg Thalheim, Pramod Bhatotia, Christof Fetzer, Michio Honda, and Kapil Vaswani. SPEICHER: Securing LSM-based key-value stores using shielded execution. In *FAST ’19*.
- [3] Andrew Baumann, Marcus Peinado, and Galen Hunt. Shielding applications from an untrusted cloud with Haven. *ACM Transactions on Computer Systems (TOCS)*, 33(3):8, 2015.
- [4] Keith Bonawitz, Hubert Eichner, et al. Towards Federated Learning at Scale: System Design. In *SysML ’19*.
- [5] Alejandro Cartas, Martin Kocour, Aravindh Raman, Ilias Leontiadis, Jordi Luque, Nishanth Sastry, Jose Nuñez Martinez, Diego Perino, and Carlos Segura. A reality check on inference at mobile networks edge. In *EdgeSys ’19*.
- [6] Zhuo Chen, Wenlu Hu, Junjue Wang, et al. An empirical study of latency in an emerging class of edge computing applications for wearable cognitive assistance. In *Proceedings of the Second ACM/IEEE Symposium on Edge Computing, SEC ’17*, New York, NY, USA, 2017. Association for Computing Machinery.
- [7] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. A survey of model compression and acceleration for deep neural networks. *CoRR*, abs/1710.09282, 2017.
- [8] Victor Costan and Srinivas Devadas. Intel SGX explained. *IACR Cryptology ePrint Archive*, 2016(086):1–118, 2016.
- [9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [10] Google. Edge TPU. <https://cloud.google.com/edge-tpu>, Accessed March 2020.
- [11] Google. More information about our processes to safeguard speech data. <https://www.blog.google/products/assistant/more-information-about-our-processes-safeguard-speech-data/>, Accessed March 2020.
- [12] Johann Hauswald, Michael A. Laurenzano, et al. Sirius: An open end-to-end voice and vision personal assistant and its implications for future warehouse scale computers. In *ASPLOS ’15*, New York, NY, USA. Association for Computing Machinery.
- [13] Intel. Intel Neural Compute Stick. <https://software.intel.com/en-us/neural-compute-stick>, Accessed March 2020.
- [14] Intel. Intel Software Guard Extensions SDK. <https://software.intel.com/en-us/sgx/sdk>, Accessed March 2020.
- [15] Jakub Konečný, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *CoRR*, abs/1610.02527, 2016.
- [16] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *CoRR*, abs/1806.08342, 2018.
- [17] Nvidia. Nvidia Jetson Nano. <https://developer.nvidia.com/buy-jetson>, Accessed March 2020.
- [18] Dan O’Keeffe, Theodoros Salonidis, and Peter Pietzuch. Frontier: Resilient edge processing for the internet of things. *Proc. VLDB Endow.*, 11(10):1178–1191, June 2018.
- [19] Christian Priebe, Kapil Vaswani, and Manuel Costa. Enclavedb: A secure database using SGX. *IEEE S&P ’18*.
- [20] Aravindh Raman, Nishanth Sastry, et al. Care to share? an empirical analysis of capacity enhancement by sharing at the edge. In *WirelessEdge ’18*.
- [21] Aravindh Raman, Nishanth Sastry, Arjuna Sathiaselan, Jigna Chandaria, and Andrew Secker. Wi-stitch: Content delivery in converged edge networks. *SIGCOMM Computer Communications Review*, 47(5), 2017.
- [22] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *CCS ’15*.
- [23] The Register. Google very angry after a contractor leaks over a thousand assistant recordings. <https://gizmodo.com/google-very-angry-after-contractor-leaks-over-a-thousand-1836308139>, Accessed March 2020.
- [24] Florian Tramèr and Dan Boneh. Slalom: Fast, verifiable and private execution of neural networks in trusted hardware. In *ICLR ’19*.