Security Issues of GPUs and FPGAs for AI-powered near & far Edge Services

Stylianos Koumoutzelis, George Avdikos, Titos Georgoulakis, Ioannis Giannoulakis and Emmanouil Kafetzakis
EIGHT BELLS Ltd, Athens, Greece

stelios.koumoutzelis@8bellsresearch.com
george.avdikos@8bellsresearch.com
titos.georgoulakis@8bellsresearch.com
giannoul@8bellsresearch.com
mkafetz@8bellsresearch.com

Abstract: Graphics Processing Units (GPUs) and Field Programmable Gate Arrays (FPGAs) are widely applied to cloud and embedded applications in which such devices are applied to near and far edge computing operations. This pool of available devices has a wide range of power/size specifications to support servers ranging from big data centres to small cloudlets, or even down to embedded systems and IoT boards. Overall, the most prominent devices and vendors in the market today are the following Xilinx for FPGA-based accelerators, Nvidia and AMD for GPUs, Intel for FPGA- /GPU-based accelerators. Decreasing the latency and increasing the throughput of Artificial Intelligence Functions (AIF), either for network automation or user applications, requires some sort of parallelization inside such purpose-built hardware acceleration. The AI@EDGE project is developing a Connect-Compute Platform (CCP) in which hardware accelerators (1 Nvidia GPU Tesla V100 (near edge device) and 1 Jetson AGX and 1 Jetson Nano (far edge devices), as well as 2 Xilinx FPGAs Alveo U280+U200 (near edge devices) and 1 Versal VCK190 and 2 Zynq ZCU104) are placed inside a server node and execute edge computing scenarios involving multiple nodes of diverse compute capabilities each, to test various integration approaches, to study orchestration techniques measure AIF deployment efficiency, all while developing certain FPGA/GPU code to accelerate representative AIFs of AI@EDGE. In this paper we compare the power/size/performance specifications of all accelerators and highlight the security issues associated with the cloud and embedded accelerators. This study presents the security issues announced by the vendors with the results of our tests and proposes tests and security functions (policies and objectives) which will be applied to the CCP to increase the security level of CCP. It also considers security issues related with the hardware set-up (accelerators inside server nodes) from the network point of view.

Keywords: GPUs, FPGAs, AIFs, AI/ML, Connect-Compute Platform

1. Introduction

The use of hardware accelerators, such as GPUs, FPGAs, TPUs, and VPUs, has become increasingly prevalent in various applications, especially in cloud and edge computing (*INACCEL* et al. 2018). These devices are designed to perform complex computational tasks, process large amounts of data, and support machine learning algorithms. However, as the reliance on these devices grows, so does the need for addressing the security concerns associated with their use. In many cases, these devices handle sensitive information and critical applications, making them vulnerable to various types of attacks and security breaches. To ensure the safe and secure use of hardware accelerators, it is essential to understand and mitigate the potential security risks involved. This paper examines the security concerns associated with modern applications of GPU edge acceleration, including unauthorized access to sensitive data, tampering with the device or its firmware, and exploitation of vulnerabilities in the software or hardware. We also explore the current best practices and recommendations for mitigating these security risks and ensuring the secure deployment and use of hardware accelerators in modern applications.

2. Embedded systems in edge architectures

Recently, embedded systems have emerged as a popular solution for providing low-latency, real-time processing capabilities for a wide range of tasks, making them particularly useful in edge computing architectures and a preferable choice for many modern applications of GPU and FPGA edge acceleration. GPU and FPGA-based embedded systems are being deployed in a wide range of applications, including IoT devices, edge gateways, and mobile devices, supporting data collection, data processing, and control functions.
Embedded systems of such kind, despite their widespread usage and utility, are vulnerable to various security threats and breaches as they often handle large amounts of sensitive data and perform complex machine learning tasks (Pottie & Kaiser et al). This could possibly make them targets for attackers who may seek to gain unauthorized access to sensitive data or tamper with the system's firmware, exploiting vulnerabilities in the

system's software or hardware. Therefore, it is essential to implement robust security measures in order to ensure the secure deployment and use of GPU and FPGA-based embedded systems in edge computing (Ghodke et al., 2021). This could include the implementation of encryption and authentication techniques to secure sensitive data and prevent unauthorized access, the use of firewalls and intrusion detection systems to prevent attacks, and the conduct of regular security audits to identify and address potential vulnerabilities.

3. Securing Embedded Accelerators

As discussed in the previous section, the increasing reliance on hardware accelerators to process large amounts of sensitive data and critical applications, has made these devices potential targets for security breaches. This section delves into the risks associated with unauthorized access and tampering with hardware accelerators and their firmware.

Specifically, tampering with hardware accelerators can occur through the use of malware, spyware, or other malicious software that infects the device and compromises its functionality. Attackers may also resort to physical tampering, such as through the use of rogue devices or unauthorized modifications, to gain access to sensitive data or compromise the device's firmware. The consequences of tampering can be severe, potentially leading to the failure of the entire system and exposing sensitive data to potential misuse.

In addition to malware and physical tampering, attackers may also exploit vulnerabilities in the software or firmware of hardware accelerators to gain unauthorized access to sensitive information. These vulnerabilities can range from simple programming errors to more sophisticated attacks that exploit weaknesses in the system's design. For example, side-channel attacks can exploit variations in power consumption or electromagnetic radiation emitted by the device to extract sensitive data.

To mitigate the security risks associated with tampering with hardware accelerators, it is essential to implement robust security measures. This includes regular updates to software and firmware, as well as the use of secure boot mechanisms that verify the integrity of firmware before it is loaded. Additionally, physical security measures, such as tamper-evident seals and secure storage, can help prevent unauthorized access to the device.

4. Exploitation of vulnerabilities in software and hardware-based attacks

Hardware accelerator weaknesses can be popular soft spots for malicious actors in order to initiate attacks through software-based methods. Although these attackers cannot implement new functions through techniques such as buffer overflow or code injection, they can manipulate inputs, including but not limited to configuration parameters or memory values, hence taking advantage of design flaws. For instance, they may exploit weaknesses (Joshi & Jain et al.) in the accelerator controller to execute various attacks, or target vulnerabilities in the System on a Chip (SoC) design.

Another vulnerability commonly found in hardware accelerators lies on the system bus, as a third party could potentially manipulate it in a way that allows them to execute unauthorized operations on the accelerators. If the system lacks proper security measures, this could result in compromised execution and possibly lead to access to sensitive information either through the output port or by exploiting the shared memory space. Thus, it is important to have proper verification and protection in place to ensure the secure execution and outcome of the accelerators.

Even though the accelerator's specifications may be secure, the actual hardware implementation can still be hacked or generally exploited through physical attacks (Pottie & Kaiser et al). These potentially malicious actions exploit the weaknesses of the implementation and can therefore cause critical failure of protection mechanisms. Side-channel attacks are used to steal secret information from embedded devices and cloud servers, such as in the case of Advanced Encryption Standard (AES) algorithm, which is mathematically secure but vulnerable to physical attacks like power and timing attacks. To counteract side-channel attacks, accelerators can help improve security, for example, by providing constant execution time to make timing attacks ineffective. Intel has already added AES-NI instructions as an example (Ghodke et al., 2021). Accelerators need to be protected from various attacks, including fault-based and side-channel attacks. If not protected properly, a circuit separate from the rest of the processor can become an easy target for precise power side-channel attacks, resulting in the compromise of key information.

5. Proposed testing and security functions

The focus here is on hardware installations which are applied at near and far edge computing applications which run AI/ML operations that require heavy computational workload. A test rig for the purposes of AI@EDGE project involved the deployment of servers at Kernel-based Virtual Machines (Ubuntu-Server) with the CPU/RAM resources specified above and a Kubernetes cluster that consists of the aforementioned VMs and is able to manage the additional custom resources, more specifically a Jetson AGX (GPU) first, and an Alveo U280 (FPGA) at a second stage either for near or far edge applications. Verification process and safety under attacks can be tested and potentially ensured if the virtual machines keep the guest operating systems patched and protect their environment. One can consider disabling unnecessary functionalities in order to minimize the use of the virtual machine console, thus conforming to modern guidelines used as best practices by several actors in the hardware acceleration community.

Especially in the case of FPGAs, the process of their programming consists of loading a bitstream file into the FPGA. This bitstream file contains the binary sequence for the FPGA design. These files are usually vulnerable to the same security threats as regular software including malware, unauthorized copy and Intellectual Property (IP) theft. Cyber criminals can duplicate or reverse engineer FPGA applications by reading the internal memory, intercepting the bitstream, or removing the protective cover of the chip. To a favorable end (Wang, Zhang, & Li et al.), application developers do not reveal their design to FPGA vendors or suppliers. As a result, attackers cannot discover any application-related information by attacking the FPGA vendor. In addition, FPGA manufacturers do not use metallization for programming. Thus, reverse engineering techniques where attackers identify the silicon mask layer by decapping the chip do not work.

In the case of GPUs (Maurice, C Neumann, & Francillon et al., 2014), Jiang et al. show a correlation timing attack on GPUs to completely recover the AES key. They use 128-bit “electronic codebook” (ECB) mode AES encryption with T-tables, which uses a 16B key to encrypt a 16B block. Each thread performs one block-encryption and the keys are stored in the constant cache. All threads of a warp simultaneously read the same round-key. T-tables are stored in GIM since access to them leads to different memory requests which are serialized and hence, storing them in the constant cache would lead to resource wastage. They record time in two ways: clean measurement, where the warp execution time inside a kernel can be measured, and noisy measurement, where the latency of incoming/outgoing messages can be measured. ECB of different bit lengths are applied and a security function (policy) is formulated. As in the case of Intel AES-NI, beyond improving performance, the new instructions help address recently discovered side channel attacks on AES. AES-NI instructions perform the decryption and encryption completely in hardware without the need for software lookup tables. Therefore, using AES-NI can lower the risk of side-channel attacks as well as greatly improve AES performance.

6. Conclusions

In the highly competitive market of today, the choice of a suitable computing platform poses a significant challenge, as the variables that revolve around it are characterized by great complexity. Hence, metrics such as its performance, energy efficiency, security and ease-of-use carry important weight in such a choice. However, these metrics are often at odds with each other. Evidently, performance-inefficient security solutions may constitute a GPU as an unattractive platform for hardware acceleration of compute-intensive applications. Security issues for FPGAs may have been identified in the past and treated by vendors and researchers, but there have been no common practices established in order for them to be examined (or even treated) until now. The reason for this is that only in the last few years GPUs have been building functional blocks in cloud applications (and not only for edge computing standalone applications). As an increasing number of security vulnerabilities of GPUs come to light, both vendors and users should now carefully weigh the performance advantages of GPUs vis-a-vis their security loopholes. An additional level of security when FPGAs and GPUs are operating in Kernel-based Virtual Machines (Ubuntu-Server) with the CPU/RAM resources specified and deployed in a Kubernetes cluster. Security functions (policies) are formulated when ECB of different bit lengths are applied. Also, it has been shown that AES-NI instructions perform the decryption and encryption completely in hardware without the need for software lookup tables.

Acknowledgement

This work is funded by the Horizon Project AI@EDGE (https://aiatedge.eu/)

References:

Boukhtouta, A., Katabi, D. and Kaafar, H. "Security in Hardware Accelerators for Cloud and Edge Computing".

Ghodke, V. (2021, February 25th). “An Overview — Systems-On-Chips (SOCs) And Their Security Risks”. Security Risks in Systems-On-Chip (SOCs).

INACCEL, Available: https://inaccel.com/cpu-gpu-fpga-or-tpu-which-one-to-choose-for-my-machine-learning-training/

Jiang, H., Fei Y., and Kaeli D., (2016) "A complete key recovery timing attack on a GPU," High Performance Computer Architecture (HPCA), p. 394–405

Joshi, D. and Jain R., "Security and Privacy Considerations for Hardware Accelerators in Cloud and Edge Computing".

Maurice, C., Neumann, O. H. C. and Francillon A., (2014) "Confidentiality issues on a GPU in a virtualized environment," International Conference on Financial Cryptography and Data Security, p. 119–135

NVIDIA, Available: https://www.nvidia.com/en-us/edge-computing/.

Pilato, C., Garg, S. and Wu K., "Securing Hardware Accelerators: a New Challenge for High-Level Synthesis," IEEE EMBEDDED SYSTEMS LETTERS.

Pottie, J. and Kaiser, W. "Security and Privacy in Hardware Accelerators for Cloud and Edge Computing".

Ramakrishnan, K. and Sekar, R. "Securing Hardware Accelerators in Cloud and Edge Computing".

Wang, X., Zhang, Y. and Li X., "Securing Hardware Accelerators in Cloud and Edge Computing".