2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC) | 978-1-6654-2463-9/21/$31.00 ©2021 IEEE | DOI: 10.1109/COMPSAC51774.2021.00216

Advancing Design and Runtime Management of AI Applications with AI-SPRINT

(Position Paper)

Hamta Sedghani*, Danilo Ardagna*, Matteo Matteucci*, Giulio Angelo Fontana*, Giacomo Verticale*, Fabrizio Amarilli*, Rosa Badia†, Daniele Lezzi†, Ignacio Blanquer‡, André Martin§, Konrad Wawruch¶

*Politecnico di Milano, †Barcelona Super Computing Center, ‡Universitat Politècnica de València
§Dresden University of Technology, ¶7Bulls

***Abstract*—The adoption of Artificial intelligence (AI) technologies is steadily increasing. However, to become fully pervasive, AI needs resources at the edge of the network. The cloud can provide the processing power needed for big data, but edge computing is close to where data are produced and therefore crucial to their timely, flexible, and secure management. In this paper, we introduce the AI-SPRINT project, which will provide solutions to seamlessly design, partition, and run AI applications in computing continuum environments. AI-SPRINT will offer novel tools for AI applications development, secure execution, easy deployment, as well as runtime management and optimization: AI-SPRINT design tools will allow trading-off application performance (in terms of end-to-end latency or throughput), energy efficiency, and AI models accuracy while providing security and privacy guarantees. The runtime environment will support live data protection, architecture enhancement, agile delivery, runtime optimization, and continuous adaptation.**

Keywords: *Cloud computing, fog computing, edge computing; AI and machine learning; Cloud trust security & privacy.*

1. Introduction

Artificial Intelligence (AI) is becoming pervasive today, with the AI software platforms worldwide market forecast to grow significantly through 2023, approaching USD 11.8 billion in revenue at a CAGR of 35.3% [1]. Many of the benefits of this evolution will come from using computing resources at the periphery of the network, i.e., where source data is produced. Many companies are evaluating the use of edge computing for data collection, processing, and online analytics to reduce applications latency and data transfers. A growing number of use cases, e.g., predictive maintenance, machine vision, and healthcare, to name a few, can benefit from AI applications spanning edge-to-cloud infrastructures leveraging resources available at the computing continuum. Edge intelligence, i.e., edge-based inferencing, will become the foundation of all industrial AI applications while most new applications [2] will involve some AI components at various levels of the computing continuum. Training and/or retraining AI models at the edge will provide opportunities to optimize the use of computational resources preserving data privacy and increasing the security of data. However, while solutions to support application development and transparent execution of general applications are becoming available, AI applications still lack solutions to optimally split AI models and application components providing resource efficiency, performance, data privacy, and security guarantees.

In this paper, we introduce AI-SPRINT, a recently funded by the European Commission under the *Horizon 2020* framework. AI-SPRINT will make it possible to seamlessly design and partition AI applications among the current plethora of cloud-based solutions and AI-based sensor devices (i.e., devices with intelligence and data processing capabilities). The AI-SPRINT framework is currently under development and this paper aims at highlighting the main *challenges* and design principles behind it. The remainder of this paper is organized as follows. Section 2 describes AI-SPRINT objectives. In Section 3, we introduce an application inspired by one of the AI-SPRINT use cases. In Section 4, we present AI-SPRINT approach. Related works are discussed in Section 5, while conclusions are finally drawn in Section 6.

2. AI-SPRINT Objectives

AI-SPRINT will overcome current technological challenges for the design and efficient execution of AI applications exploiting resources in the edge-to-cloud continuum such as flexibility, scalability, interoperability, security and privacy, and aims at achieving the following objectives:

1) **Provide design and development tools** for the implementation of AI applications (including machine learning / deep learning applications and large-scale analytics) consuming resources across the computing continuum including multi-clouds, edge servers, and AI enabled sensors. The design environment will include: (1) programming abstractions to hide the communications across components and to transparently implement the parallelization of the compute-intensive

This work is funded by EU Grant no. 101016577.

978-1-6654-2463-9/21/$31.00 ©2021 IEEE
DOI 10.1109/COMPSAC51774.2021.00216

part of the application possibly exploiting specialized resources (e.g., GPUs and AI enabled sensors), (2) quality annotations (e.g., data flow rates, application latency, energy constraints) to express performance, accuracy, privacy, and security constraints.

2) **Deliver tools for secure execution and privacy preservation** to implement solutions providing secure AI models deployment and data processing which combine hardware-enforced confidentiality and integrity with policy-defined privacy. This aims to: (1) ensure confidentiality and integrity of application data across the computing continuum, (2) secure application configuration and execution and data transfers, (3) provide privacy-preserving solutions through AI model partitioning and edge training and retraining.
3) **Develop a runtime environment for application execution and monitoring** that implements policies orchestrating the applications' execution across the computing continuum via: (1) application deployment to accelerate the delivery of applications and AI models, (2) monitoring of edge-to-cloud resources, (3) dynamic, energy aware, and optimal resource allocation to fit workload requirements and data size/streaming throughput variability, and (4) resilient applications' execution with performance guarantees.
4) **Provide advanced solutions for AI architecture enhancement** allowing iterative refinement of AI application architecture design and deployment by: (1) assimilating newly generated data from the field, (2) integrating new AI sensors, (3) adding new privacy, security, and application performance constraints. The final goal is to feed such information to the design tools and redeploy an updated AI model and application to the target environment, exploiting edge resources for model training and retraining.

AI-SPRINT solutions will be validated across industrial and health domains thanks to three use cases including farming 4.0, maintenance & inspection, and personalized healthcare in line with EU priority areas for AI investments [3].

The tools developed in the AI-SPRINT project will help AI application developers to easily implement new applications. System integrators will be provided with powerful yet flexible tools to develop multi-cloud systems including resources at the full computing continuum stack involving classic components and AI models. Cloud providers will benefit from tools for simplifying resource management also at the edge layer and for offering new PaaS AI services thanks to the availability of novel open source design tools.

3. AI-SPRINT Running Example

In order to provide the intuition of the AI-SPRINT solutions' benefits, we introduce an application inspired by one of the AI-SPRINT use cases related to the maintenance and inspection of a wind farm as depicted in Fig. 1(a). The application software includes six components that today need to be developed independently and integrated by spending a significant effort. Conversely, Fig. 1(b) shows the development path that can be followed when the AI-SPRINT toolchain will be available.

The first software component, C_1, is in charge of taking a picture from the airborne drone, controlled remotely by a human operator. The second component, C_2, is a classifier, which checks if the image quality is sufficient for further processing and otherwise it triggers the acquisition of a new picture. C_3 is a complex, computer-vision-based application that, given a model of the wind farm, positions the picture on the farm itself and guides the operator to identify the next element to be examined. This component has the goal of monitoring the inspection campaign and guarantees that this covers the complete site. C_4 and C_5 are two additional AI modules whose goals are to: (1) provide an initial feedback to the operator and notify if the inspected part, e.g., the blade, has a problem (this is performed by C_4 and it could possibly require to take additional pictures); and (2) identify the specific problem, e.g., a crack (performed by C_5). Finally, C_6 is a large-scale analytic application, which identifies patterns in part degradation by considering external factors (e.g., historical weather data, usage of a specific turbine, size of the plant, etc.) and it is periodically run also considering the data gathered across all the windfarms in a geographic area.

A possible deployment runs C_1 and C_2 on the drone, C_3 on the edge server located in the operator's van, while C_4, C_5 and C_6 are executed on the cloud. Since C_4 and C_5 require significant computation, they are run on expensive GPU-based cloud VMs, which are started on demand by C_3 logic. Unfortunately, C_4, C_5, and C_6 are not under the administrative control of the data owners, hence cloud tenants or powerful adversaries with root access can potentially access image data. C_4 and C_5 are implemented via classifiers based on Convolutional Neural Networks (CNNs) whose architecture was iteratively identified by software developers with a classic trial and error process until the model accuracy was considered satisfactory. Being a complex activity, application developers do not usually explore a large number of solutions and the computational requirements of the final identified CNNs need to be matched *a posteriori* with the cloud VMs capacity and application performance constraints (e.g., by verifying *a posteriori* that C_5 inference time for a single image is less than 50 ms).

Similarly, it is also difficult to provide an upper bound to the time required to retrain the AI model in case the application is extended to cope with additional damage classes. Model retraining, and to some extent also the initial training of C_4 and C_5 CNNs, is challenging from the privacy perspective if it is run on cloud, whereas it could only achieve best effort performance if the organization local infrastructure has limited capacity.

4. AI-SPRINT approach

AI-SPRINT provides novel design tools and a runtime environment for fast development, optimization, and man-

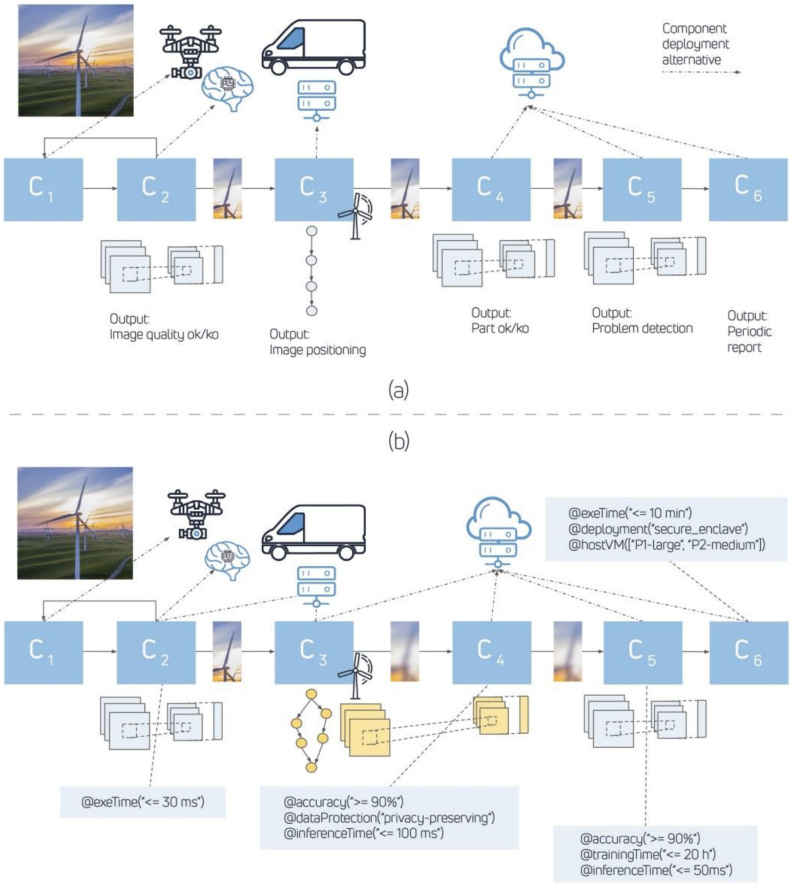

Figure 1. AI-SPRINT solution applied to the use case of maintenance and inspection of a wind farm. (a) Components required by the application (b) Design approach enabled by AI-SPRINT toolchain.

agement of AI applications running across a cloud-edge computing continuum. According to the current assets of the partners, we present the general architecture of AI-SPRINT in Fig. 2.

4.1. AI-SPRINT design tools

AI-SPRINT design-time solutions include tools for: (1) annotating AI applications with non-functional/QoS constraints and guide code parallelization; (2) evaluating *a priori* performance of AI applications both at training and at inference time; (3) identifying and automating the search for the most accurate deep neural network starting from labeled dataset; (4) automatically exploring multiple candidate deployments for the application components maximizing resource efficiency and minimising the cloud usage cost.

4.1.1. Design & programming abstractions. Application components will be annotated with design abstractions allowing to specify: i) intra- and intercomponents parallelism, ii) constraints on application performance, security, privacy, and underlying resources, and iii) deployment alternatives that will be automatically explored by the envisioned solutions. For example, in Fig. 1-b, C_3 is a complex application and internal parallelism can be exploited to improve application execution time and to reduce the van edge server required capacity. The base for the AI-SPRINT programming layer will be the PyCOMPSs programming framework[1] that aims to facilitate the parallelisation of existing Python applications. PyCOMPSs offers a simple programming model based on sequential development in which the developer is mainly responsible for: i) identifying the functions to be executed as asynchronous parallel tasks; ii) annotating them with standard Python decorators. New annotations will be introduced to allow predicating on components performance and to specify constraints on the target deployment. For example, C_2 can be automatically allocated by the design tools on the drone or on the van edge server according to the application performance constraints (e.g., image processing time less than 30 ms). On the other hand, to enforce a high-security level for the execution of C_6, another annotation will force C_6 code to run in a secure enclave. A number of ML algorithms implemented with PyCOMPSs are already

1. https://www.bsc.es/research-and-development/software-and-apps/software-list/comp-superscalar

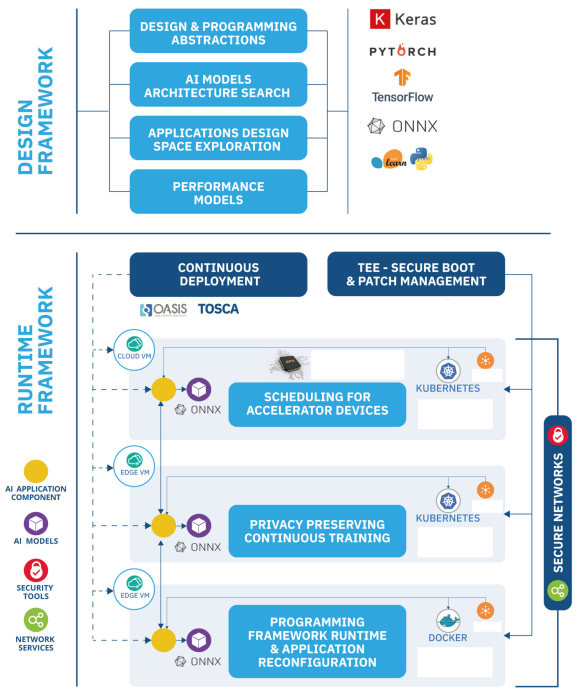

Figure 2. AI-SPRINT Architecture

available as a Distributed Computing Library[2] inspired by scikit-learn, that eases the task of developing applications providing a common interface in all algorithms. Furthermore, as a parallel framework, PyCOMPSs exploits inter-node and intra-node parallelism and executes neural networks tasks (like Tensorflow or Keras) as external processes.

4.1.2. Performance models. AI-SPRINT will provide also a performance modelling approach based on Machine Learning (ML) to predict the execution time of inference or training tasks of AI models deployed across the computing continuum. The tools will automate the AI application performance profiling and identify the ML model providing the highest performance prediction accuracy supporting model selection and hyper-parameters tuning. Preliminary results in [4], [5], [6] have shown that ML models allow to achieve good accuracy (with average percentage error between 5 and 15%) in cloud environments. AI-SPRINT will extend the use of such models to consider AI-based sensors and deep networks partitioned and deployed across computing continua.

4.1.3. AI models architecture search. AI-SPRINT will develop solutions to enable developers with limited ML expertise to train high-quality models specific to their needs also in terms of Quality of Service (QoS) requirements. In particular, we focus on deep neural networks, which now have become very popular and the reference framework for ML due to their human level accuracy in domains such as images and sequences (e.g., time series) understanding. Despite the compelling arguments for using neural networks as a general template for solving ML problems, training these models and designing the right network topology is still a matter of art craft and designer experience. Indeed, the design still requires specifying the parameters of a typically large network architecture with several layers and units, and then solve a difficult non-convex optimization problem. AI-SPRINT will provide tools for developing learning as a service solution, that, starting from a training set with labelled training examples (images or temporal data series which are of interest for use cases) will automatically identify the most accurate deep neural network which provides execution time guarantees. For example, component C_5 (see Fig. 1-b) will be automatically identified in a way that the model accuracy (on a cross-validation set) will be at least 90% keeping the inference time below 50ms under a target production environment. Moreover, AI-SPRINT will automatically identify, e.g., the minimum cost target deployment able to support the training activity in less than 20 hours (given the available training set). Finally, AI-SPRINT will develop solutions for breaking down large deep networks for cooperative, privacy preserving analytics. To this end, instead of performing the whole model inference on the cloud, AI-SPRINT will explore solutions where the edge servers or the AI-enabled sensors run the initial layers of the neural network, and then send the output to the cloud to feed the remaining layers and produce the final result. For example, since C_4 is annotated with *privacy protection*, the AI model will be split into two components (orange elements in Fig. 1-b) so that some feature pre-processing can be performed on the van edge server and the images sent to the cloud will be obfuscated (blurred) and will not allow, even to the cloud provider, to reconstruct the original picture.

4.1.4. Applications design space exploration. AI models architecture search, as the deployment of non-deep models/application components, will be subject to an additional step of design space exploration where multiple alternative candidate deployments will be evaluated. As an example, in Fig. 1-b component C_6 can be deployed on large VMs at provider P_1 or medium instances at provider P_2. Since the number of solutions may become extremely large depending on the number of possible providers and components allocation alternatives, AI-SPRINT will ease the application design by providing an automated tool to derive, given the QoS constraints, optimal component placement maximizing resource efficiency while minimizing the cloud usage cost.

4.2. AI-SPRINT runtime environment

To run applications transparently on a heterogeneous architecture, we need a runtime environment which manages the different components and data of the applications and reacts to system perturbations and requirement changes. The AI-SPRINT runtime environment will include tools to: i) support the continuous deployment of AI applications; ii)

2. https://github.com/bsc-wdc/dislib

support application components concurrent execution reacting to node failures and identifying their optimal placement and resource capacity; iii) trigger automated model retraining, leveraging on solutions for training and retraining at the edge; iv) optimize the scheduling and assignment of accelerator devices among competing training jobs.

4.2.1. Continuous deployment. Continuous deployment is the process where development, testing, delivery and progressive deployment of an application is considered as a whole, reducing operation and development costs. Nowadays, once an application is in production, IaaS users have to manage the virtual infrastructures where their applications run, which usually imposes significant overhead and need for skills. IaaS users have to choose the VM image from a catalogue that best fits their needs and configure software (i.e., application servers and binaries). As application requirements might change, new application versions are developed and need to be re-deployed. This is especially important in a scenario where data is continuously retrieved, and AI models are retrained. Given the heterogeneity of the edge layer, it is necessary to provide streamlined mechanisms to guide the process of configuring computing resources across the computing continuum. A large number of different cloud management platforms are available today to automate application deployment on cloud systems [7]. Despite the advantage of being able to choose among different providers, this diversity poses an important drawback caused by the use of non-interoperable systems. To avoid cloud lock-in and to facilitate application deployment at the edge, AI-SPRINT will leverage IM[3] and provide a cloud-edge orchestrator enabling the automatic deployment of AI application models and components, without manual provisioning.

AI-SPRINT will use Docker containers to automate deployment and provide the level of isolation needed to enforce performance constraints with minimal overhead. Both private and public clouds will be valid targets. Applications will be described as OASIS TOSCA templates[4] describing the topology of their components and their software dependencies. Application templates will support restrictions for the deployment on multiple heterogeneous resources (i.e., including hardware accelerators, e.g., GPGPUs) also at the edge layer, by specific attributes (e.g., the target image, performance constraints or privacy requirements), which will be instantiated and managed by a single interaction with the AI-SPRINT framework.

4.2.2. Programming Framework Runtime. AI-SPRINT will also provide a runtime system in charge of supporting concurrent code execution, automatically detecting and enforcing the data dependencies among components and spawning parallel tasks to the available resources, which can be nodes in an edge cluster or clouds. The solution will leverage the PyCOMPSs runtime, which already provides scalability and elasticity features of edge and cloud resources. In particular, PyCOMPSs is already able to: i) manage distribution, parallelism and heterogeneity in the edge resources transparently to the application programmer; ii) handle data regardless of persistency by supporting a single and unified data model; iii) use edge devices as workers to migrate components initially deployed in the cloud and/or to react to failures of nodes running at any layer of the computing continuum. Within AI-SPRINT, PyCOMPSs runtime will be extended to support the serverless computing paradigm, and application components will be run as event-triggered functions orchestrated within a workflow. This will allow a flexible and elastic execution of AI applications. For example, given the edge server current load, C_3 in Fig. 1-b can be dynamically allocated and run as a function on the edge server or in the cloud reducing operation costs and still providing execution time guarantees for the overall application. Application components migration will leverage Krake[5] which provides a central point to manage application components at any layer of the computing continuum identifying the "best" layer depending on user-defined metrics, re-evaluating the placement at regular intervals and automatically scaling-up and down resources. Components to resource assignment will be based on multiple criteria (e.g., battery level, network latency, availability of accelerator devices boosting component execution, etc.).

Finally, application components will be continuously monitored by a monitoring platform which will be able to gather metrics at every layer (including the application level, possibly including AI inference accuracy) of the computing continuum and the ESPER[6] complex event processing engine will integrate and correlate application end-toend performance with the monitoring events and respective measurements at the full cloud-edge stack.

4.2.3. Privacy preserving continuous training. AI applications frequently operate in dynamic environments that change over time and models must be continually updated to capture the most recent data trends; AI-SPRINT will provide solutions to facilitate AI model continuous training. In particular, AI-SPRINT will provide a general-purpose framework towards continuous integration of AI model updates and code changes into the deployment environment, advancing the features provided by most of today's learning frameworks which perform training offline, without any system support for continual model updating. Model retraining (periodic or triggered by large drifts) will be supported. Two possible kinds of updates are foreseen: i) the AI model structure is unchanged (only weight matrices and internal state initialization, in the case of models with memory, are updated); ii) the AI model structure changes because of retriggering of the AI model architecture search. Edge training and retraining in AI-SPRINT will leverage federated learning algorithms [8], [9], [10] where different

3. https://www.egi.eu/services/cloud-compute/

4. http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.3/TOSCA-Simple-Profile-v1.3.pdf

5. https://gitlab.com/rak-n-rok/krake

6. http://www.espertech.com/esper/

learners exchange only part of the information which can be extracted from the data in order to improve a model globally, but without making explicit the data used to train it. Error losses and corresponding gradients will be computed as close as possible to the data sources so as to distribute the least amount of information which is not strictly required for the training or retraining task leveraging AI model partitioning between the edge and cloud layers (C_3-C_4 and C_3-C_5 CNN submodels in Fig. 1). For example, referring to the example in Fig. 1-b, during the training, C_3 submodel forward pass will be processed on an edge server which will send to the cloud only preprocessed data (blurred images in Fig. 1). The cloud, in turn, will compute the loss function and C_4 parameters gradients (which will be sent back to the edge server) without accessing the raw images.

4.2.4. Scheduling for accelerator devices. Deep neural network training is a computationally intensive process. A promising approach to reduce the processing time is using one or more GPUs, which allows to achieve from 5 up to 40x time improvement when compared to CPU-only deployments. Training NNs on multiple GPUs makes it also possible to handle larger amounts of data and increases the accuracy of the trained models. Despite all of these advantages, the cost of GPU-based systems is usually high both in private and public clouds (e.g., GPU-based VMs time unit cost is 5-8x higher than the corresponding CPU based VMs [6]). The efficient use of GPUs is hence an important problem that AI-SPRINT will address. Moreover, to increase the possibility to share expensive and specialized resources, cloud infrastructures are shifting away from traditional configurations, where hardware accelerators are installed locally in individual servers, toward the resource disaggregation paradigm, in an attempt to expose accelerators as pooled network resources that can be accessed across all the system nodes. As a consequence, cloud infrastructures can increase resource utilization by allocating spare fragmented resources to remote training applications. Virtualization and resource disaggregation simplify the complexity and enhance the flexibility of cloud systems management but pose new challenges on how to obtain the best performance. Indeed, in such environments, applications have limited access and view of the resources, since disaggregation technologies abstract the view of the hardware topology and its characteristics and allow resource access through fine-grained resource partitioning. Hence, to fully exploit the resource efficiency, an intelligent system able to orchestrate the resources with a high-level view of the system is needed. AI-SPRINT will face these problems by proposing advanced techniques to solve the joint resource planning (i.e., how many GPUs to assign to a training job) and scheduling problems (i.e., determine the job ordering leading to optimal resource access) both for private and public clouds. The final goal is to obtain the efficient use of shared public cloud VMs/private infrastructures while observing the completion time goals of training jobs (taking into account, possibly, AI model partitioning between cloud and edge servers). GPU sharing and access mechanisms will be based on rCUDA[7], which allows multiple devices that connect through a network to a single rCUDA server, to share the physical GPUs among the different concurrent requests.

4.2.5. Trusted Execution Environments. AI-SPRINT will provide tools for the deployment of compute instances enabled with Trusted Execution Environments (TEEs). Trusted execution of applications in computing continua poses many challenges: i) application entities must be orchestrated in a way that only trusted parties are allowed to establish communication channels with each other, i.e., end-to-end encryption; ii) code attestation and verification mechanisms are required to ensure that only the correct code/binary is being executed, as well as secure patch management where only code changes which are verifiable and auditable are performed; iii) the orchestration framework must support heterogeneous environments that provide TEEs of different vendors such as Intel SGX, ARM Trustzone[8], etc.; iv) since there exists no trust relationship among the various stakeholders as well as privileged users such as the cloud provider, data and code must be protected in a way only owners can access and modify. In order to provide security guarantees across all layers, mechanisms such as secure boot are essential to also attest and guarantee that the correct operating system (OS) is booted. In order to address these challenges, AI-SPRINT, starting from SCONE[9] will develop an orchestration framework that provides secure application execution using TEEs. The framework will enable software developers to run unmodified code in TEEs supporting various vendors as well as providing a cross-compiler framework for translating legacy applications into secure AI-SPRINT ones. In order to provide strong end-to-end encryption, the framework will provide mechanisms for key generation, exchange as well as remote attestation and verification. Moreover, the framework will also integrate secure booting mechanisms to ensure that the OS applications are running on can be attested to provide an extra level of security. To reflect continuous integration of fast-evolving applications, the framework will be equipped with patch management mechanisms providing verifiable mappings of binary code changes to trusted source code repositories in a way the executed code can be always trusted. Finally, an integrated policy enforcement mechanism ensures that only data owners can access and modify their training or inference data.

4.2.6. Secure Networks. The creation and lifecycle management of secure network paths in a computing continuum requires the coordination of multiple routers across multiple sites interconnected by transit networks, across multiple administrative domains, and with different security characteristics. Providing trusted and reliable communications requires dealing with many challenges, such as: (1) untrusted transit

7. http://www.rcuda.net
8. http://www.sierraware.com/open-source-ARM-TrustZone.html
9. https://sconecontainers.github.io

networks; (2) network resources changing over time due to node mobility; (3) different nodes and domains having different capabilities in terms of security; (4) conflicting policies due to the co-existence of multiple tenants; (5) network failures and disruptions.

To address these challenges, AI-SPRINT will leverage and extend the ONOS controller[10] [11] and its ecosystem of technologies for network control and management. AI-SPRINT will make it possible to seamlessly configure and manage a secure, private edge cloud service. The AI-SPRINT Secure Networks component will leverage network programmability to minimize provisioning delays and to implement rich security policies while minimizing traffic tromboning. Additionally, it will manage the setup of the relevant Virtual Network Functions, steer secure tunnels through them, and will ensure that the network paths comply with the security policies defined at design time including when traffic is rerouted in case of failure.

In deploying the network functions and paths, the component will optimize the deployment taking into account the availability of programmable network devices, such as programmable switches or smart NICs supporting the P4 programming language[11] [12].

5. Related work

In this section, we review some recent relevant works related to AI cloud-edge application design, runtime management and security. An important step of the application design is performance modeling. The performance analysis of computing continuum applications can be tackled from different perspectives. The most traditional ones rely on analytical models [13] and simulation [14]. Some recent studies have exploited ML models for performance prediction of large systems [15], [16], since a black box approach has several benefits (e.g., no required knowledge of the system internals, fast estimation time). Moreover, AI model architecture search is a hot topic in AI cloud-edge application design both for industry and academia. Works can be classified according to the exploration paradigm adopted in searching for the best topology suited for solving a specific problem; in general, most of the approaches use genetic algorithms [17] and reinforcement learning [18]. After the first attempt to learn dense neural models [19], most of the research focus is now on CNNs for image recognition with model accuracy as a target, although more recent approaches have started considering a multi-objective perspective trading-off accuracy with energy consumption [20] also for edge devices [21] and in a federated setting [10].

Runtime management is among the most important issues to run the applications on a heterogeneous architecture. Cloud orchestration is the process needed to manage the entire lifecycle of a cloud application. Our previous work in the field is IM [7], a cloud orchestration runtime that deploys complex and customized virtual infrastructures on multiple back-ends, including most popular public clouds and on-premise solutions. From the programming framework runtime perspective, optimizing and provisioning edge resources is critical because of the edge limited capacity and several approaches have been proposed to tackle this issue. Some of them (e.g., [22]) focus on domain-specific solutions. Another approach refers to the assignment of multi-task applications in heterogeneous clouds (e.g., [23], [24]). A third possible set of approaches exploits decentralized resource management systems [25]. Scheduling is another significant topic in runtime management. There has been considerable work done in the domain of GPU scheduling for HPC systems to improve load balancing and performance of CPU and GPUs [26] but there are few solutions for AI training applications. Authors in [27] proposed Gandiva, a cluster scheduling framework able to improve latency of training deep learning models in GPU clusters. [28] presents a topology-aware placement strategy to schedule deep learning jobs on a Power8 machine based on NVLink while [29] proposes Harmony, a deep learning-based scheduler addressing explicitly performance interference among DL jobs and minimizes average job completion time. Significant GPU utilization increase can be obtained by means of virtualization [30] and remote access [31] which allow to transparently share the GPUs in a server among many applications running in different nodes of the cluster.

To guarantee privacy and security at the node and network levels, we must address these elements both in design tools and in the runtime environment. A Trusted Execution Environment (TEE) is a secure part of a general-purpose CPU that guarantees confidentiality and integrity of code and data loaded into it. A number of frameworks exist that exploit the TEE functionality in several ways. For example, Haven [32] uses Intel SGX to execute unmodified Windows applications, such as Microsoft SQL Server, inside a secure enclave. Another relevant framework is provided by VC3 [33] which uses Intel SGX to execute trustworthy MapReduce applications in an untrusted cloud environment. A number of open source frameworks exist that support the use of Intel SGX such as OpenEnclave [34], [35]. From the network perspective, recent works have explored various directions relevant to AI-SPRINT: guaranteeing consistent updates throughout the network is a critical issue for patch management [36]; policy checking by injecting probe packets makes it possible to assess the behaviour of hybrid networks [37]; automatic conflict resolution is a necessary tool to minimize the impact of security policies on network performance [38].

6. Conclusion

This paper presents the research agenda and main challenges that will be faced by the AI-SPRINT H2020 project. AI-SPRINT will develop a novel framework for developing and operating AI applications and, since its very beginning, will foster open source and the early adoption of its tools

10. https://opennetworking.org/onos/
11. https://p4.org/

by end-users. Core benefits resulting from such a novel tech toolbox include:

- Reduction of skill shortage and steep learning curves in the development of AI software on computing continuum ecosystems through open source tools, models, methods and methodologies.
- Shorter time to market for AI applications that meet performance, security and privacy requirements.
- Reduction of costs for software developers to design and evolve AI applications running at the edge, by defining novel solutions to select optimal architectures.
- Reduction of costs to operate AI applications and increased value for cloud providers and end-users, by relying on a runtime environment with efficient and energy aware use of computing continuum resources.

References

[1] D. Schubmehl, "Worldwide Artificial Intelligence Software Platforms Forecast, 2019–2023," https://www.idc.com/getdoc.jsp?containerId=US44170119, 2019.

[2] D. Woods, "Why AI At The Edge Is The Next Goldmine," https://www.forbes.com/sites/danwoods/2018/04/04/why-ai-at-the-edge-is-the-next-goldmine, 2018.

[3] "White Paper on Artificial Intelligence: a European approach to excellence and trust," https://ec.europa.eu/info/publications/white-paper-artificial-intelligence-european-approach-excellence-and-trust_en, 2020.

[4] A. Maros, F. Murai, A. P. Couto da Silva, J. M. Almeida, M. Lattuada, E. Gianniti, M. Hosseini, and D. Ardagna, "Machine learning for performance prediction of spark cloud applications," in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*, 2019, pp. 99–106.

[5] E. Gianniti, M. Ciavotta, and D. Ardagna, "Optimizing quality-aware big data applications in the cloud," *IEEE Transactions on Cloud Computing*, 2018.

[6] E. Gianniti, L. Zhang, and D. Ardagna, "Performance Prediction of GPU-based Deep Learning Applications," in *CLOSER 2019*, 2019.

[7] M. Caballer, I. Blanquer, G. Moltó, and C. de Alfonso, "Dynamic management of virtual infrastructures," *Journal of Grid Computing*, vol. 13, no. 1, pp. 53–70, 2015.

[8] W. Y. B. Lim *et al.*, "Federated learning in mobile edge networks: A comprehensive survey." *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[9] K. Bonawitz *et al.*, "Towards federated learning at scale: System design," in *SysML*, 2019.

[10] C. He *et al.*, "Fednas: Federated deep learning via neural architecture search," in *CVPR 2020*, 2020.

[11] G. M. Berde Pankaj *et al.*, "Onos: towards an open, distributed sdn os," in *Proceedings of the third workshop on Hot topics in software defined networking*, 2014, pp. 1–6.

[12] P. Bosshart, D. Daly, Gibb *et al.*, "P4: Programming protocol-independent packet processors," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 87–95, 2014.

[13] D. J. Dubois and G. Casale, "Performance prediction for burstable cloud resources," in *VALUETOOLS 2016*. ACM, 2017, p. 217–218.

[14] G. Kecskemeti, G. Casale, D. N. Jha, J. Lyon, and R. Ranjan, "Modelling and simulation challenges in internet of things," *IEEE Cloud Computing*, vol. 4, no. 1, pp. 62–69, 2017.

[15] S. Venkataraman *et al.*, "Ernest: Efficient performance prediction for large-scale advanced analytics," in *NSDI 2016*. ACM, 2016, p. 363–378.

[16] X. Pan *et al.*, "Hemingway: Modeling distributed optimization algorithms," in *NIPS Workshop on Machine Learning Systems*, 2016.

[17] H. Liu *et al.*, "Hierarchical representations for efficient architecture search," in *ICLR*, 2018.

[18] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," in *ICLR*, 2017.

[19] K. O. Stanley, D. B. D'Ambrosio, and J. Gauci, "A hypercube-based encoding for evolving large-scale neural networks," *Artificial Life*, vol. 15, no. 2, pp. 185–212, 2009.

[20] R. Mammadli, F. Wolf, and A. Jannesari, "The art of getting deep neural networks in shape," *ACM Transactions on Architecture and Code Optimization*, vol. 15, no. 4, pp. 1–21, 2019.

[21] L. Cai *et al.*, "TEA-DNN: the Quest for Time-Energy-Accuracy Co-optimized Deep Neural Networks," https://arxiv.org/abs/1811.12065, 2019.

[22] S. Yi *et al.*, "Lavea: latency-aware video analytics on edge computing platform," in *SEC*, 2017, pp. 1–13.

[23] S. Olena, N. Matteo, S. Stefan, B. Michael, and L. Philipp, "Optimized iot service placement in the fog," *Service Oriented Computing and Applications*, vol. 11, no. 4, pp. 427–443, 2017.

[24] A. Das *et al.*, "Performance optimization for edge-cloud serverless platforms via dynamic task placement," in *CCGRID*, 2020.

[25] D. Puthal, R. Ranjan, A. Nanda, P. Nanda, P. P. Jayaraman, and A. Y. Zomaya, "Secure authentication and load balancing of distributed edge datacenters," *Journal of Parallel and Distributed Computing*, vol. 124, pp. 60–69, 2019.

[26] O. Kayiran *et al.*, "Managing gpu concurrency in heterogeneous architectures," in *MICRO*, 2014.

[27] W. Xiao, R. Bhardwaj, R. Ramjee *et al.*, "Gandiva: Introspective cluster scheduling for deep learning," in *13th (OSDI 18)*. Carlsbad, CA: USENIX Association, Oct. 2018, pp. 595–610.

[28] M. Amaral, J. Polo, D. Carrera *et al.*, "Topology-aware gpu scheduling for learning workloads in cloud environments," in *SC '17: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2017.

[29] Y. Bao, Y. Peng, and C. Wu, "Deep learning-based job placement in distributed machine learning clusters," in *INFOCOM*, 2019.

[30] C.-H. Hong, I. T. A. Spence, and D. S. Nikolopoulos, "Gpu virtualization and scheduling methods: A comprehensive survey," *ACM Computing Surveys*, vol. 50, no. 3, p. 1–37, 2017.

[31] C. Reaño *et al.*, "Local and remote gpus perform similar with edr 100g infiniband," in *Middleware Industry*, 2015.

[32] A. Baumann, M. Peinado, and G. Hunt, "Shielding applications from an untrusted cloud with haven," *ACM Transactions on Computer Systems*, vol. 33, no. 3, 2015.

[33] F. Schuster, M. Costa *et al.*, "Vc3: Trustworthy data analytics in the cloud using sgx," in *2015 IEEE Symposium on Security and Privacy*, 2015, pp. 38–54.

[34] "Open Enclave SDK," https://openenclave.io/sdk/, 2020.

[35] "OpenSGX: An open platform for Intel SGX," .https://github.com/sslab-gatech/opensgx, 2020.

[36] K.-T. Foerster, S. Schmid, and S. Vissicchio, "Survey of consistent software-defined network updates," *Survey of Consistent Software-Defined Network Updates*, vol. 21, no. 2, pp. 1435–1461, 2018.

[37] P. Zhang, C. Zhang, and C. Hu, "Fast data plane testing for software-defined networks with rulechecker," *IEEE/ACM Transactions on Networking*, vol. 27, no. 1, pp. 173–186, 2019.

[38] Q. Li, Y. Chen, P. P. C. Lee, M. Xu, and K. Ren, "Security policy violations in sdn data plane," *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1715–1727, 2018.