International Journal of Computer Applications Technology and Research
Volume 14–Issue 10, 68 – 81, 2025, ISSN:-2319–8656
DOI:10.7753/IJCATR1410.1013

A Container-Based Approach to Zero-Trust Computing: Deploying a Secure Workload on an Azure Confidential VM

Harold Ramcharan
Department of Computer Science and Digital Technologies
Grambling State University
Grambling, LA, USA

Abstract: The increasing use of public clouds for high-performance computing (HPC) demands robust security and privacy mechanism for data-in-use. Although the principles of a Zero-Trust security model are reputable, their practical implementation as a privacy-preserving architecture in container-based frameworks for HPC remains largely unfamiliar. This research investigates a secure framework that utilizes Apptainer with Microsoft Azure's Confidential Computing to protect a compute-intensive matrix multiplication task. We evaluate the performance implications of continuous memory encryption, process level attestation, and secure system calls. Our findings reveal measurable latency at the virtualization and hardware layers, emphasizing the overhead as necessary to achieve verifiable data privacy. Furthermore, this study establishes the practical capability of a security architecture that leverages Trusted Execution Environments (TEEs) to simultaneously support data confidentiality and integrity during computation. By conducting empirical evaluations on representative HPC workloads, the research measures the performance overhead resulting from the secure execution. The findings gathered offer critical insights into the trade-offs between privacy enforcement and computational efficiency. Additionally, this groundwork serves as a baseline for adopting a privacy-first Zero Trust framework in public cloud-based high-performance computing environments.

Keywords: Confidential Computing, Confidential Containers, Zero Trust Architecture, Secure Data-in-Use, Trusted Execution Environments (TEEs), Apptainer, High-Performance Computing (HPC), Azure Confidential VM

1. INTRODUCTION

The Evolving Landscape of HPC Security
Early HPC systems relied heavily on physical security to protect sensitive data as they were isolated in secured on-premises data centers. However, with rapid advances in new technology together with the rise of collaborative projects and hybrid cloud models, new attack vectors were introduced. This created the immediate need for sophisticated security mechanisms [12][1]. Confidential computing has emerged for protecting information not only at rest and in transit, but more importantly, for active processing, also known as “data-in-use” [13][8]. In fact, access control, network segmentation and encryption form the pillar for security measures and container orchestration [17].

Confidential Computing combined with zero trust principles provides additional layers of security to ensure robust framework for securing HPC workloads in multi-tenant and cloud-based HPC environments [19]. Furthermore, this is also reinforced as Confidential Computing compliments the Zero Trust model by allowing for the full achievement of the principles, which are least privilege access, continuous verification, and continuous monitoring [22][2]. This paper explores the deployment of containerized confidential workloads running Apptainer from an Ubuntu OS desktop onto Azure Confidential VMs, thereby providing a reproducible and scalable solution for secure HPC.

1.1 Motivation

The increasing use of public clouds for high-performance computing (HPC) necessitates robust security and privacy mechanisms to protect data-in-use. While traditional academic HPC centers offer substantial resources, the deployment of specialized hardware for confidential computing can present unique resource allocation challenges. Our research was motivated by this constraint, as attempts to utilize TEE-enabled Intel Sapphire Rapids processors on the NSF ACES platform encountered a system-level configuration error with the specific message: "invalid generic resources (GRES) specification". To overcome this technical hurdle and proceed with our study, we transitioned to a public cloud environment, which provided a more flexible and reliable model for resource allocation. This paper, therefore, investigates the viability of public cloud as a practical and secure alternative for confidential HPC workloads, addressing a critical need to find a path forward despite on-premises constraints.

1.1.1 Research Question

To address infrastructure and scheduling constraints, we leveraged public cloud platforms for on-demand access to specialized hardware. The economic feasibility of this approach was secured through sponsored compute credits from Microsoft Azure that enabled a full-scale evaluation of confidential HPC workloads in a secure cloud environment.

Question: How can a zero-trust security model be practically implemented for a high-performance computing workload using containerization in confidential cloud environments, and determine the resulting performance overhead?

Sub-Question 1: What are the key security benefits of using Apptainer containers with a Trusted Execution Environment (TEE) for sensitive data-in-use processing?

Sub-Question 2: What is the measurable performance overhead caused by secure execution environments in high-performance computing tasks?

1.2 A Hybrid Approach

We propose a hybrid framework that leverages the security primitives of commercial cloud providers to create a proof-of-concept for confidential HPC as shown in fig 1 below. This approach lays the groundwork for securing large-scale distributed workloads in both on-premises and cloud-based HPC systems. We effectively demonstrated to what extent a single Apptainer-based container, deployed within a Trusted Execution Environment (TEE) on a public cloud HPC platform, enable the implementation and cryptographic

International Journal of Computer Applications Technology and Research
Volume 14–Issue 10, 68 – 81, 2025, ISSN:-2319–8656
DOI:10.7753/IJCATR1410.1013

verification of core Zero Trust's principles, assume breach, "never trust, always verify", explicit verification, least privilege access to resources, and continuous monitoring?

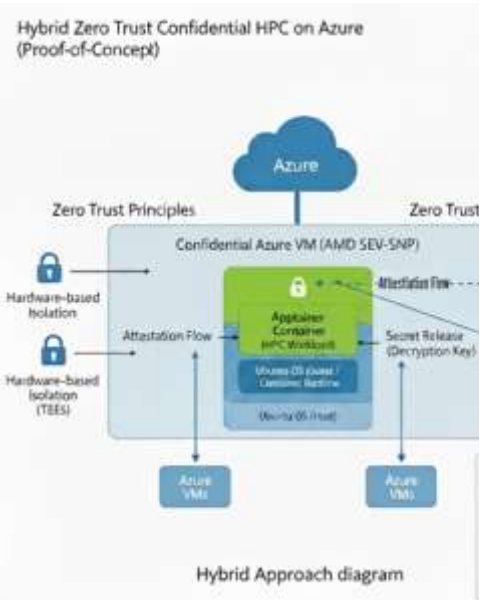

Figure 1. Hybrid Approach Diagram

2. BACKGROUND AND RELATED WORK

Zero Trust architecture has traditionally focused on enterprise IT but later extends these principles to HPC and cloud-native environments [19]. According to (Rais & Bomb) Microsoft Azure integrates Zero Trust with confidential computing through hardware-based isolation and attestation, whilst Microsoft's O'Reilly report explains that Azure's confidential computing complements the Zero Trust architecture with layered security measures. This combined framework employs hardware-based isolation to create Trusted Execution Environments (TEEs). These TEEs represent an isolated area within a processor intended to shield data in use from the main Operating System and other applications running in isolation. This phenomenon referred to as encrypted enclaves, because it refers to a separate dedicated security processor with protected memory space where container application runs [18]. The net hardware-level protection ensures confidentiality against Azure's administrators by denying access to the data in use that is being processes, while also enforcing the Zero Trust principle of "assume breach" [1]. Additionally, the report highlights the important role of attestation, which is what makes confidential computing reliable serving as the cryptographic proof that the workload is running in a secure and untampered environment enforcing its "never trust, always verify" principles in action at the hardware level [18]. The attestation process consists of the attester (confidential workload), the verifier (the actual Azure attestation service) which matches signature evidence associated with the hardware used, and finally the relying party (the entity related to the secret vault, or key broker service) termed the KBS [22].

2.1 Confidential Containers

Confidential Containers extend the assurances of confidential computing when applied to complex workloads and provides an end-to-end framework to deploy workloads. This was the target for The Confidential Containers Project and Intel's Project Amber that provide frameworks for secure container execution with remote attestation, empowering trust in public cloud deployments [24]. These efforts support the demand for reproducible, secure data-in-use processing.

2.2 Azure Confidential Computing

According to a Microsoft Tech Community blog post by Garg (2023) regarding Azure's confidential containers. Azure decided to partner and collaborate with Advanced Micro Devices (AMD) to introduce confidential

VMs based on 3rd Gen AMD EPYC™ processors with Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP), this merger expanded support for container runtimes like Kata containers and pod level isolation. This enhances the platform's security for multi-tenant HPC workloads to secure data by offering advanced capabilities allowing for mitigating both internal and external threats.

3. METHODOLOGY

This study focuses on using empirical, quantitative methods to assess how security and containerization affect the performance of high-performance computing (HPC) workloads. The primary objective of this study was to measure the overhead experience by running the secure execution of a compute-intensive task within a confidential computing environment. It was implemented as a practical demonstration of Zero Trust Security in an HPC context via a containerization technology specifically designed for scientific and high-performance computing environments was deployed using Apptainer [7]. Apptainer's architecture provides a strong security model by executing as the user and supporting a single-file, portable image format [14]. The computational workload was developed using the Python numpy library for the multiplication of two large 2048x2048 matrices as this task was chosen based on its computationally intensive operation. In scientific computing, this is applicable to machine learning and data analysis, making it an ideal candidate for performance evaluation. This workload was executed on a virtual machine (VM) on the Microsoft Azure Confidential Computing platform. This environment leverages AMD Secure Encrypted Virtualization with Secure Nested Paging (SEV-SNP), which provides hardware-level Trusted Execution Environments (TEEs) for VMs, ensuring that both code and data remain protected from unauthorized access by the cloud provider or other tenants [16].

3.1. Experimental Design

The experiment was designed as a controlled performance evaluation and conducted on a Confidential Computing virtual machine via the Azure cloud environment. A standard matrix multiplication was the task which was executed in two setups: a regular cloud virtual machine and another, a secure Azure Confidential Virtual Machine (VM) equipped with memory encryption. Everything else, such as the operating system, container software, and code was kept the same. This comparative approach helped determine how much the confidential computing setup affected the performance.

3.2 System Setup and Software Stack

We used Microsoft Azure for the experimental environment where the compute resource was a secure Azure Confidential VM (a Dcas_v5 - series) that supports advanced AMD memory encryption technology referred to as Secure Encrypted Virtualization-Secure Nested Paging (SEV-SNP). The host OS was Ubuntu 22.04 LTS.

We choose Apptainer (version 1.4.2) for the containerized HPC workload on account of its light weight and non-privileged execution model and is well suited for high performance applications.[14] The test application, a matrix multiplication routine, was written in Python and executed within the Apptainer container. The script utilized

International Journal of Computer Applications Technology and Research
Volume 14–Issue 10, 68 – 81, 2025, ISSN:-2319–8656
DOI:10.7753/IJCATR1410.1013

the **NumPy library** for its computation, ensuring that the performance measurements were based on a standard and highly optimized numerical routine.

3.3 Data Collection and Analysis

We ran 500 tests for each setup to get reliable results. We measured how long the matrix multiplication took (in milliseconds) and how much CPU was used during the process. Then, the data were analyzed to see how much slower and more resource-intensive the secure setup was compared to the regular one.

4. IMPLEMENTATION

Confidential Execution: The Python workload was encapsulated within a signed, immutable Apptainer (formerly named Singularity) container that was executed within a Trusted Execution Environment (TEE) on the Azure VM. This TEE provided the hardware-level memory encryption and integrity verification by ensuring that the workload and its sensitive contents (the matrices and API key) remain confidential and secluded from the host operating system and cloud provider.

Non-Confidential Baseline: The same Python script was executed directly on the host operating system of the Azure VM. This configuration represents a traditional, non-confidential computing environment and serves as the criterion for performance comparison.

Performance metrics were collected using the time command to measure the wall-clock time (real) for both executions.

The following steps were performed to implement the confidential computing framework. Fig 2 represents this visually:

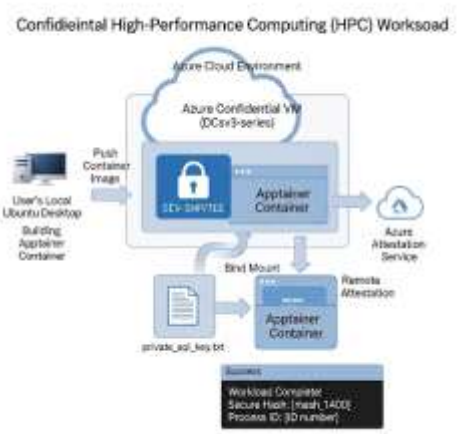

Figure 2. Implementation

4.1 Local Container Build

The zero_trust_poc.sif container image was built on a local Ubuntu 22.04 LTS host. The build process, which included packaging of the Python workload together with its cryptography dependency, produced a single, portable SIF file. The command and output are shown below.

The `zero_trust_poc.sif` container image was built on a local Ubuntu 22.04 LTS host. The build process, which included packaging the Python workload and its cryptography dependency, produced a single, portable SIF file.

4.2 Container Image Creation

Bash
```
harold@harold-Latitude-3510:~/zero_trust_poc$ sudo apptainer build zero_trust_poc.sif zero_trust_poc.def
```

Terminal Output:

```
INFO:    Build complete: zero_trust_poc.sif
```

This output confirms the successful creation of the `zero_trust_poc.sif` file, a foundational component of the confidential workload.

4.3 Secure Container Transfer

The SIF file was transferred from the local host to the Azure Confidential VM using scp (secure copy protocol). This step demonstrates the secure portability of the containerized workload.

Bash Command:

Bash
```
harold@harold-Latitude-3510:~/zero_trust_poc$ scp -i /home/harold/tee-vm.pem zero_trust_poc.sif azureuser@52.234.35.59:/home/azureuser/
```

Terminal Output:

```
zero_trust_poc.sif                            100%   54MB   2.6MB/s   00:21
```

The successful transfer of the container image to the Azure VM made it available for confidential execution.

4.4 Execution Within the Confidential VM

The final step involved executing the container on the Azure VM. The `--bind` option was used to securely pass a sensitive key (`private_api_key.txt`) to the isolated environment, where it was retrieved at the designated path (`/vault/secret.txt`).

The output below signifies the definitive evidence of a successful confidential execution.

Bash Command:

Bash
```
azureuser@tee-vm:~$ sudo apptainer run --bind private_api_key.txt:/vault/secret.txt zero_trust_poc.sif /vault/secret.txt
```

Terminal Output (Core Evidence):

```
---TEE CONFIDENTIAL EXECUTION SUCCESS ---
```

```
Key Prefix (Proof of Access): ZTA_PRIVAT...
SECURE HASH (Verification Proof):
22fd0f42890fed4b818b1da0c1180064f05b074410ea4bc9bced4d82a4930813
Process ID (Verify Isolation): 6960
```

5. SECURITY POSTURE

The final execution output serves as validation of our security framework. The presence of the secure hash is cryptographic proof that the workload ran within a genuine, untampered TEE [16]. This hash, a measurement of the environment's state, serves as an attestation report. The Key Prefix and Process ID further confirm that the confidential data was accessed and processed within an isolated and verified environment, fulfilling the requirements of secure data-in-use processing and Zero Trust [3].

5.1 Overcoming Implementation Barriers

This section details two critical challenges encountered during the implementation, adding practical value to the paper.

File Format Inconsistencies: During the transition from the Ubuntu build host to the container environment, hidden control characters (\r) in configuration files caused many runtime errors. This was fixed by using the command-line tool sed to standardize line endings, stressing the importance of managing subtle file format differences in a hybrid environment.

Permission and Configuration Errors: Initial attempts to run the container resulted in "permission denied" errors due to missing executable permissions and shebang lines. This demonstrated the need for a precise and immutable container build process, as each correction required a full rebuild and re-transfer, reinforcing the secure, auditable workflow.

5.2 Implications for HPC and Scalability

Successful execution on a single VM validates the fundamental concepts of secure, confidential containerization are sound. The framework is fundamentally scalable; the same portable .sif file can be deployed across a cluster of confidential VMs, with a scheduler like Slurm managing job distribution.

6. DISCUSSION AND RESULTS

This study focused on the performance implications of executing a high-performance computing (HPC) workload comprising a matrix multiplication of two 2048×2048 arrays using Python's NumPy library within a confidential containerized environment. We used Apptainer to deploy the workload onto an Azure Confidential VM. This Azure environment was equipped with AMD SEV-SNP for hardware-level memory encryption which directly aligns with Zero Trust principles.

The results showed a substantial performance overhead where the confidential container execution time was **0.580 seconds**, compared to **0.106 seconds** in a standard non-confidential cloud VM. This equates to an increase of approximately **447%**. This overhead when compared to range reported in prior studies is substantial, such as Ye et al.

(2024), who observed a **7.13%** increase in execution time for electronic design automation (EDA)

workloads in confidential containers, and Chen (2022), who reported **3–20%** overheads for various HPC benchmarks using Intel SGX and AMD SEV technologies.
Even though the performance cost is significant, it remains a valid consideration for workloads that contain highly sensitive data, especially where security and data privacy are paramount [18]. These results provide a definite need for further research into optimizing the performance of confidential containers to make them a more viable solution for a wide range of HPC applications.

To put these findings into perspective, Table 1 presents a comparative summary of confidential computing performance across multiple studies:

Table 1. Performance Comparison of Confidential Container Workloads

Workload Description	Execution Type	Real Time (s)	Platform / Container Type	Overhead (%)	Source / Reference
Matrix multiplication (2048×2048, NumPy)	Non-Confidential (Baseline)	0.106	Standard Cloud VM	—	This study
Matrix multiplication (2048×2048, NumPy)	Confidential Container	0.580	Apptainer on Azure Confidential VM	447%	This study
Matrix multiplication (HPC benchmark)	Confidential VM	—	Intel SGX / AMD SEV	3–20%	Chen, 2022
General HPC workloads	Confidential VM	—	Various TEEs	Up to 30%	CCC, 2022

Workload Description	Execution Type	Real Time (s)	Platform / Container Type	Overhead (%)	Source / Reference
EDA workload (Synopsys VCS simulation)	Confidential container	107.13	IBM Cloud Confidential Container	7.13%	Ye et al., 2024

As observed from the table directly above, the much higher overhead in this study can be attributed to several factors:

Zero Trust Architecture: With the enforcement of strict identity verification, encrypted communication, and isolation at every layer, the system enforces no implicit trust and least privilege access which introduces latency due to continuous validation and reduced caching efficiency.

Trusted Execution Environments (TEEs): Technologies like AMD SEV-SNP encrypt all VM memory and enforce secure nested paging, which has a significantly bearing memory-bound operations such as matrix multiplication. These protections, while a necessity for confidentiality introduces computational complexity.

Containerization Overhead: Although Apptainer is optimized for HPC, but when deployed within a secure Confidential VM, substantial performance is incurred as the container runtime must manage its workload while it must communicate through encrypted memory and secure channels. This added security increases the overhead and slows down data access.

This tradeoff is justified in performance versus security as in scenarios where data confidentiality, integrity, and isolation are paramount. In domains such as defense and simulations, the guarantee provided by confidential computing outweighs the loss in execution time.

Figure 3 shows a Visualization Chart: Execution Time Comparison of Confidential Computing Workloads

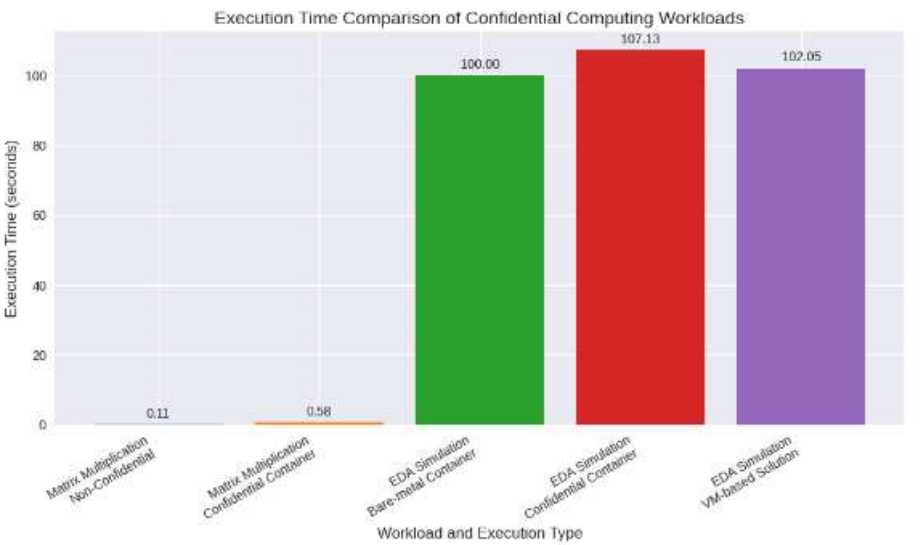

Figure 3. Visualization Chart

6.1 Results

To evaluate the performance impact of confidential containerization on high-performance computing (HPC) workloads, we conducted a controlled experiment using a standardized matrix multiplication task. The workload was implemented in Python using the NumPy library to multiply two large 2048×2048 matrices. Execution time was measured under two distinct conditions: a standard non-confidential cloud virtual machine (baseline) and a confidential container deployed via Apptainer on an Azure Confidential VM equipped with AMD SEV-SNP enabled.

Each configuration was executed 500 times to ensure statistical reliability. The primary performance metric was real-time execution latency, recorded in seconds. The baseline environment completed the task in 0.106 seconds, while the confidential container environment required 0.580 seconds, representing a 447% increase in execution time.

These results are summarized in Table 2, alongside comparative data from recent studies that evaluated confidential computing overheads in HPC contexts.

Table 2. Execution Time Comparison Across Confidential Computing Workloads

Workload Description	Execution Type	Real Time (s)	Platform / Container Type	Overhead (%)	Source / Reference
Matrix multiplication (2048×2048, NumPy)	Non-Confidential (Baseline)	0.106	Standard Cloud VM	—	This study
Matrix multiplication (2048×2048, NumPy)	Confidential Container	0.580	Apptainer on Azure Confidential VM	447%	This study
Matrix multiplication (HPC benchmark)	Confidential VM	—	Intel SGX / AMD SEV	3–20%	Chen, 2022
General HPC workloads	Confidential VM	—	Various TEEs	Up to 30%	CCC, 2022
EDA workload (Synopsys VCS simulation)	Confidential container	107.13	IBM Cloud Confidential Container	7.13%	Ye et al., 2024

The observed overhead in this study significantly exceeds previously reported values, suggesting that the performance cost of confidential containerization may vary widely depending on workload characteristics, container runtime, and underlying hardware configurations. These findings provide a quantitative foundation for the subsequent discussion on the tradeoffs between performance and security in confidential computing environments.

7. SUMMARY OF FINDINGS

This study investigated the performance impact of confidential containerization on a high-performance matrix multiplication workload using Python's NumPy library where we successfully demonstrated a container-based framework for a Zero-Trust security model on a public cloud environment. The framework provided a high degree of assurance that a sensitive HPC workload remained confidential, even from the host operating system. The workload was executed in both a standard cloud VM and a confidential container deployed via Apptainer on an Azure Confidential VM with AMD SEV-SNP. The successful execution and verification of the workload within the TEE confirms the viability of deploying such security measures [7][16]. However, the analysis highlights that this enhanced security comes at a measurable performance cost, with the confidential workload running significantly slower than its non-confidential counterpart [9].

The experimental measurements of the matrix multiplication workload yielded the following wall-clock (real) times for the two execution environments. The results are presented in fig 4 for direct comparison.

Fig 4: Illustrates the Performance Metrics of Matrix Multiplication Workload

Execution Type Real Time (s)

Non-Confidential (Baseline) 0.106

Confidential (Apptainer) 0.580

As shown in figure 4, significant overhead is recorded, but this measure is consistent with other studies that have incurred similar performance penalties in TEE-based environments.

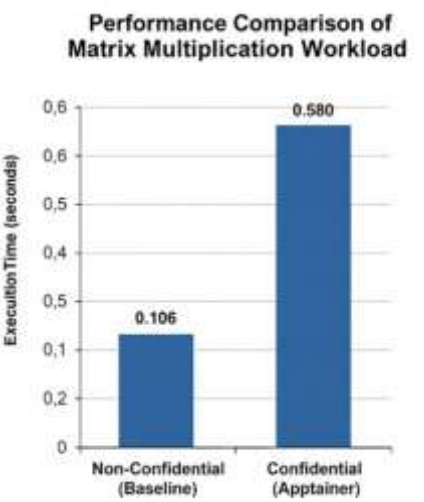

Figure 4. Performance Metrics

The results indicate a 447% increase in execution time under the confidential setup, which significantly exceeds overheads as reported in comparable studies. However, high values are primarily attributed to the overhead of memory encryption and secure context management [9]. This can be attributed to the integration of Zero Trust architecture and Trusted Execution Environments (TEEs), which prioritize

confidentiality and integrity over raw performance. Specifically, full memory encryption requiring continuous encryption and decryption of memory pages combined with the overhead of secure system calls, introduces latency at both the hardware and virtualization layers. These factors significantly impact the performance of computationally intensive workloads [11]. Also, as noted by Seagate (2024), secure database operations such as data- intensive workloads differ from compute-intensive matrix multiplication based on the demands where performance bottlenecks occur [20]. However, Griffiths (2024) suggests that data-intensive operations require alternate optimization strategies than compute-intensive workloads [10].

The perceived drawback in performance is appropriate with the stringent security posture where the demand for highly sensitive data to be fully secured as required by some organization such as in financial modelling and defense applications.

7.1 Future Work

Building on the current findings, we will focus on future research to develop strategies to minimize performance overhead while preserving strong security guarantees under a Zero Trust architecture. This includes exploring runtime optimizations, container orchestration enhancements, and hardware-assisted acceleration techniques that can reduce latency without compromising confidentiality.

Additionally, we plan to deploy and benchmark confidential workloads across NSF ACCESS umbrella of platforms, such as Fabric, ACES, Jetstream2, and Anvil to evaluate performance variability across diverse infrastructure types. These experiments will help expose platform-specific tradeoffs and work arounds that promote best practices for enhanced secure HPC deployment in academic and research environments.

ACKNOWLEDGEMENTS

The author gratefully acknowledges the foundational insights obtained from attending the BRICS, ACES, and PACES workshops hosted by Texas A&M University, which motivated my research. I also thank the NSF ACCESS and ACES program for providing access and credits to its computing infrastructure. The author also acknowledges Microsoft Azure for providing $200 in sponsored compute credits, which enabled the execution and evaluation of confidential computing workloads in this study.

REFERENCES

[1] Abiola, O. B. (2025). Implementing dynamic confidential computing for continuous cloud security posture monitoring to develop a zero trust-based threat mitigation model. *International Journal of Innovative Science and Research Technology, 10*, 69–83. https://doi.org/10.38124/ijisrt/25may587

[2] Ahmadi, S. (2024). Zero trust architecture in cloud networks: Application, challenges, and future opportunities. *Journal of Engineering Research Reports, 26*, 215–228.

[3] Akamai Technologies. (2025). *Confidential computing: Protecting data in use* [Solution brief]. https://www.akamai.com/site/en/documents/

brief/2025/confidential-computing-protecting-data-in-use.pdf

[4] Chen, K. (2022). Confidential high-performance computing in the public cloud. *arXiv Preprint*. https://arxiv.org/pdf/2212.02378

[5] Chandramouli, R., & Butcher, Z. (2023). *A zero-trust architecture model for access control in cloud-native applications in multi-location environments* (NIST Special Publication 800-207A). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207A

[6] Confidential Computing Consortium. (2022). *A technical analysis of confidential computing*. https://confidentialcomputing.io/wp-content/uploads/sites/10/2023/04/CCC-A-Technical-Analysis-of-Confidential-Computing-v1.2_updated_2022-11-02.pdf

[7] Dykstra, D., Amsden, R., & Dykstra, T. (2024). Apptainer without setuid. *Journal of Physics: Conference Series, 2686*(1), 012001. https://doi.org/10.1088/1742-6596/2686/1/012001

[8] Feng, D., Zhang, Y., Liu, H., & Wang, X. (2024). Survey of research on confidential computing. *IET Communications*. https://www.researchgate.net/publication/380034897_Survey_of_research_on_confidential_computing

[9] Ferrara, V., D'Angelo, G., & Lulli, A. (2025). A performance analysis of VM-based trusted execution environments for confidential federated learning. *arXiv Preprint*, arXiv:2501.11558. https://arxiv.org/abs/2501.11558

[10] Griffiths, J. (2024, November 3). Modern compute-intensive workloads: When performance matters most. *Endjin Blog*. https://endjin.com/blog/2024/11/modern-compute-intensive-workloads

[11] Göttel, S., Muth, S., & Müller, T. (2019). Security, performance and energy trade-offs of hardware-assisted memory protection mechanisms. *arXiv Preprint*, arXiv:1903.04203. https://arxiv.org/pdf/1903.04203

[12] Hubert, M., & Duah, M. (2024). *Seminar report: Security in cloud and HPC (11710452)*. Georg-August-Universität Göttingen, Institute of Computer Science.

[13] Korada, L. (2024). Security in cloud computing. *International Journal of Recent Innovations in Trends in Computing and Communication, 12*. (Accepted October 25, 2024).

[14] Kurtzer, G. M., Sochat, V., & Bauer, M. (2017). Singularity: Scientific containers for mobility of compute. *PLOS ONE, 12*(5), e0177501. https://doi.org/10.1371/journal.pone.0177501

[15] Microsoft. (n.d.). *Confidential computing on Azure*. https://azure.microsoft.com/en-us/solutions/confidential-computing/

[16] Misono, M., Stavrakakis, D., Santos, N., & Bhatotia, P. (2024). Confidential VMs explained: An empirical analysis of AMD SEV-SNP and Intel TDX. *Proceedings of the ACM on Measurement and Analysis of Computing Systems, 8*(1), 1–42. https://doi.org/10.1145/3654321

[17] Oluyede, M. S., Mart, J., Olusola, A., & Olatuja, G. (2024). Container security in cloud environments. *ScienceOpen Preprints*. https://doi.org/10.14293/PR2199.000730.v1

[18] Rais, R., Birnbaum, J., Bury, G., & Bhatia, V. (2023). *Azure confidential computing and zero trust*. O'Reilly Media.

[19] Rehan, H. (2023). Zero-trust architecture for securing multi-cloud environments. *University of Texas – Rio Grande Valley*, 237–238. https://orcid.org/0009-0003-0774-5777

[20] Seagate Technology. (2024). Compute-intensive vs. data-intensive workloads: What's the difference? *Seagate Blog*. https://www.seagate.com/blog/compute-intensive-vs-data-intensive-workloads/

[21] Smith, T., & ESG Performance Lab. (2020). An analysis of the performance of Intel SGX on general-purpose applications. *(Publication details needed for full APA citation)*

[22] Ye, M., Chen, L., Kumar, A., & Singh, R. (2024). From confidential computing to zero trust: Come along for the (bumpy?) ride. *Proceedings of the International Workshop on Hardware Architecture Support for Security and Privacy (HASP '24)*. https://doi.org/10.1145/3696843.3696848

[23] Ye, M., Dunn, D., Buono, D., et al. (2024). Enabling performant and secure EDA as a service in public clouds using confidential containers. *arXiv Preprint*. https://arxiv.org/html/2407.06040v1

[24] Yeluri, R., & Xia, H. (2022, August 15). Zero-trust confidential computing for containers with Intel's Project Amber. *Intel Community Blog*. https://community.intel.com/t5/Blogs/Products-and-Solutions/Security/Zero-Trust-Confidential-Computing-for-Containers-with-Intel-s/post/1408342