WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

OPEN GPU ENABLED SERVERLESS COMPUTING PLATFORM WITH BLOCKCHAIN

Yamuna S [*1] ,Aravind S[*2], Mahesh Kumar G[*3], Subaram MS[*4]*

[*1]Assistant Professor, Department Of CSE, Meenakshi Sundararajan Engineering College, Chennai, Tamil Nadu, India.
[*2,3,4]Final Year UG Student, Department Of CSE, Meenakshi Sundararajan Engineering College, Chennai, Tamil Nadu, India.

ABSTRACT :

Exploring the synergy of GPU-enabled serverless computing in the realm of distributed image workloads, this study investigates the dynamic integration of Graphics Processing Units (GPUs) within serverless architectures to create an open platform of GPU resources that can be used for a wide variety of tasks. The growing demand for efficient image processing in contemporary applications necessitates scalable and high-performance solutions. Leveraging the parallel processing prowess of GPUs, this research optimizes resource utilization and reduces latency for tasks such as image analysis and rendering. By dynamically allocating GPU resources based on workload requirements, the serverless architecture ensures cost-effectiveness and seamless scalability. Through empirical evaluations and case studies, this work illustrates the efficacy of GPU-enabled serverless computing, providing valuable insights for the deployment of image-intensive applications in diverse computing environments.

Keywords: Analysis, investigation, GPU, serverless

Introduction

In the rapidly evolving landscape of technological advancements, the pervasive influence of GPU-centric applications across sectors such as healthcare, entertainment, and artificial intelligence has precipitated an ever-growing demand for GPU resources for high volume multi threaded data processing. A fundamental aspect of this exploration lies in understanding the architecture of GPUs, particularly NVIDIA GPUs, which have become synonymous with high-performance computing. Modern GPUs are equipped with a myriad of CUDA cores, forming the bedrock of parallel processing capabilities. These cores, organized into streaming multiprocessors (SMs), operate cohesively to execute multiple threads simultaneously. A granular examination of GPU architecture and CUDA cores reveals the inherent parallelism, providing insights into how they enhance the computational efficiency of image processing algorithms.The parallel processing capabilities of CUDA cores are pivotal in addressing the computational demands of image processing, which often involve intricate mathematical computations and intensive data manipulations. As image-centric tasks become more intricate and computationally demanding, harnessing the power of CUDA cores provides a pathway to accelerate these operations and achieve significant performance gains.Simultaneously, the paper delves into the paradigm of serverless computing, a contemporary approach that departs from traditional infrastructure models. Serverless computing operates on an event-driven architecture, allowing for dynamic resource allocation and cost-effective scaling. This aligns seamlessly with the inherent variability and unpredictability of gpu demanding processing workloads. A detailed exploration of serverless computing fundamentals provides a foundational understanding of its role in accommodating distributed image processing tasks.Central to serverless computing is the concept of Function as a Service (FaaS), representing a fine-grained approach to deploying individual functions tailored for specific tasks. FaaS enables modular and flexible deployment, aligning perfectly with the granular nature of image processing operations. This section of the paper explores the integration of FaaS in image processing, illustrating how it amplifies the benefits of serverless computing for handling distributed workloads.The NVIDIA-container toolkit emerges as a critical component in facilitating the deployment of GPU-accelerated applications in a serverless environment. Containers encapsulate the application and its dependencies, ensuring consistent execution across various environments. The toolkit provides a streamlined approach to containerization, simplifying the deployment and management of GPU-accelerated applications within serverless architectures. A closer examination of the NVIDIA-container toolkit sheds light on its role in achieving reproducibility, portability, and scalability in the context of GPU-enabled serverless computing for distributed image processing.Empirical evaluations constitute an integral part of this exploration, offering a practical understanding of the performance implications of GPU-enabled serverless computing for distributed image processing workloads. Real-world use cases and scenarios are meticulously analyzed to showcase the benefits and challenges of implementing this hybrid approach. Through these evaluations, the paper aims to offer actionable insights for researchers, developers, and practitioners grappling with the complexities of contemporary image processing applications.Furthermore, the paper delves into the practical considerations and challenges associated with deploying GPU-enabled serverless computing solutions. It addresses issues such as resource management, load balancing, and scalability, offering recommendations and best practices to guide developers in optimizing their implementations

Nomenclature

AMQP-Advance message queue protocol
MSA-Micro Service Architecture
GPU- Graphic Processing Unit

Methodology :

Serverless computing is an event driven resource allocation model, where necessary resources are allocated only when a particular function is invoked by various means of trigger.It is also considered as Function as a Service(FAAS), where small programs are written in form of small functions and these functions are executed in an isolated environment. Isolation of resources for these functions usually happens through containerization mechanisms like docker,containerd etc. These functions primarily utilize CPU resources and not GPU resources. This makes the use case of functions limited. Exposing isolated containers to underlying system GPU resources can open doors for new means of computation. This happened through nvidia-docker toolkit, which is a toolkit provided by Nvidia to enable the containers to access underlying GPU resources.

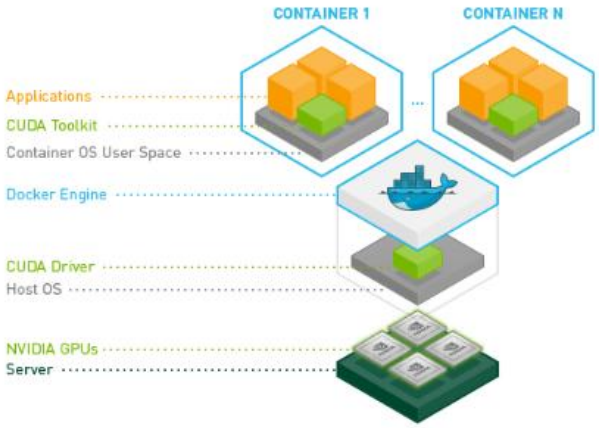

Fig . 1 - Nvidia-container toolkit architecture.

With the integration of the Nvidia-container toolkit into existing serverless computing frameworks like IronFunctions, we can provide a GPU enabled serverless computing platform. But to make the platform open (i.e.) allows anybody to add their system as a processing machine, we will need more than that. The primary issue that arises in making the platform open is Proof of Work i.e. identifying the correct system that executed the task and providing rewards to that machine. This is done by integrating Blockchain into the system. A private blockchain is set up using Hyper Ledger Fabric which is an open source framework to build and deploy private blockchain networks.

System Description

The system primarily comprises three components, a controller, a provider and the blockchain network. The controller is the system that acts as an entry point for any task request that is submitted to the system.The controller listens for http requests to execute particular tasks and when a request comes to execute the task the task data is fetched and is sent to the provider. The Controller maintains the list of providers that are connected to the network and whenever a task is requested , a suitable provider to execute the task is identified and is sent to that provider. A similar entry is also persisted in the blockchain.The controller uses rabbitmq message queue to transmit the job details to the provider system.

Table 1 - Hardware Requirements

Requirements	Specifications
GPU	Nvidia GPU
CPU	4 vCPU
RAM	8GB
OS	Linux

Table 2 - Software Requirements

Requirements	Specification
USER PORTAL	HTML,,CSS,JS
MESSAGE QUEUE	RABBIT MQ
CODING LANGUAGE	PYTHON, JAVA
CONTAINERISATION	DOCKER

System architecture

The provider system which receives the job request, spins a docker container with nvidia-toolkit support and then redirects the job to the particular container to execute the job and destroys the container. Upon completion of the task the result is sent back to the controller in the message queue which the controller sends as result for the http request.All the activities that happen in the request are recorded in a privately deployed blockchain network using hyperledger fabric, which is used for monetary transactions between the task scheduler and compute provider. It also acts as a ledger for all the actions in the network to ensure proof of work.

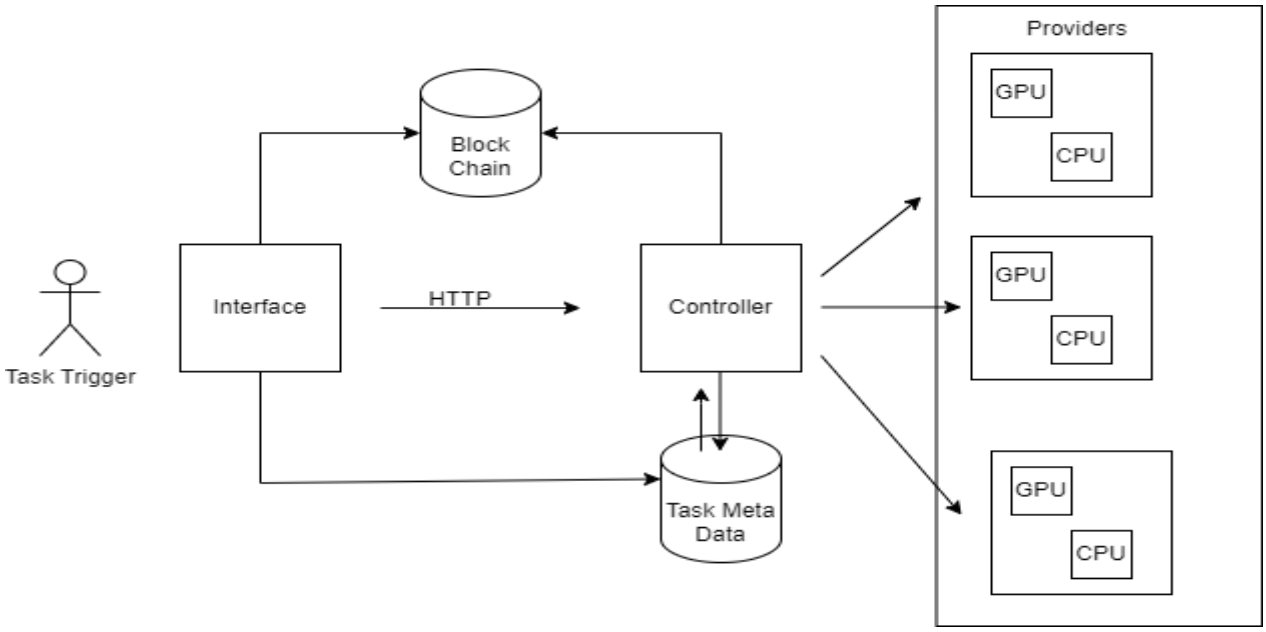

Fig .1 - Architecture

1.3.1 Image Building

Any workload that needs to be executed in the network should be first in the form of a docker image which can be uploaded to the docker hub.The docker image should consist of the necessary software and necessary assets to execute the task. The url of the image uploaded in docker hub should be provided in the web interface. The url will be used to pull the image and spin container out of it to execute the task in the provider.

1.3.2 Provider Addition

A provider with a nice computer with GPU, should install the software provider. This software will consist of a shell script which will perform a one time installation of necessary software like docker, nvidia-container toolkit etc.The user can add this computer to the network whenever he wants using the software and can remove it from the network whenever he wants.The provider software will subscribe to a channel in rabbitmq and will execute tasks that come in the channel. The rabbitmq acts in a RPC pattern for synchronous request handling.

Result and Discussion

The proposed system was tested for the task of extracting text from images using the Tesseract Optical Character recognition engine. The System took around 20-30 seconds on average to execute the task, The primary parts that contributed to the above latency is the function which calculates the best node to execute the task and to transmit the task to the node using rabbitmq message queue. In order to bring the latency down, changes can be made in implementing efficient algorithms to find the best node to execute the task by considering a variety of factors. Currently a cluster based round robin is used while the cluster is constructed based on a model of GPU.

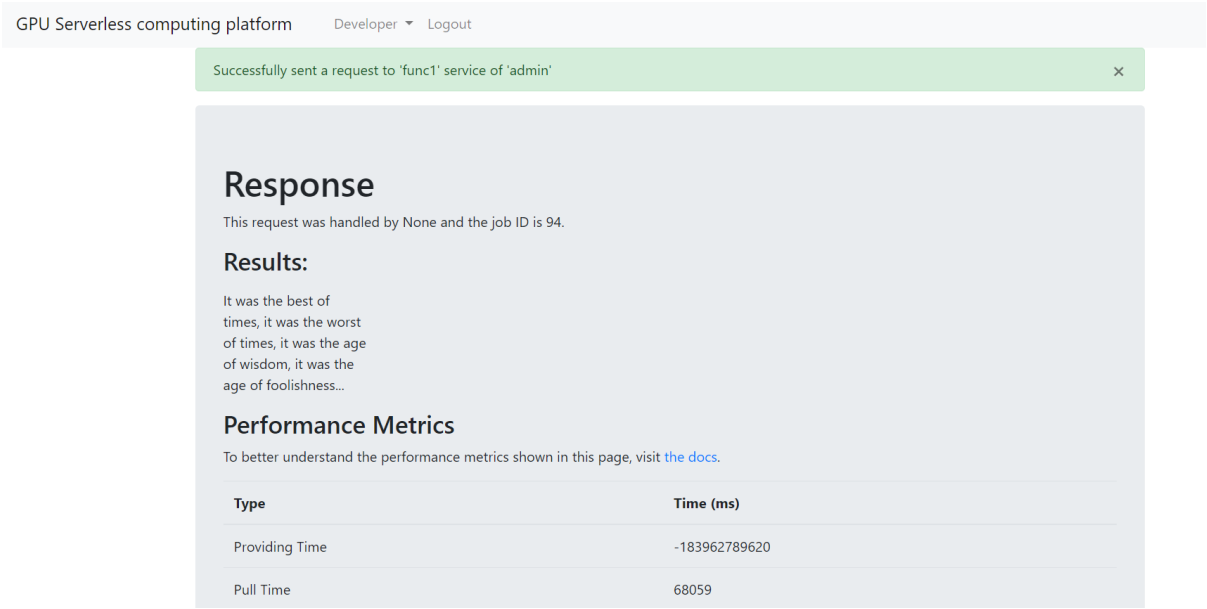

Fig . 3 - Task execution result

Conclusion

In conclusion, the proposed system design harmonizes GPU-enabled serverless computing with containerization and the NVIDIA-container toolkit, offering a robust solution for distributed image processing. By segmenting image tasks within a serverless framework, the design achieves modularity and scalability. The introduction of an additional container isolation layer ensures secure and efficient GPU resource utilization. Orchestrating serverless functions triggering specific containers enhances parallelism, optimizing image workload processing. This innovative combination not only streamlines deployment but also contributes to seamless orchestration. Emphasizing adaptability and efficiency, this design presents a promising pathway for the future of scalable and efficient image processing applications.

REFERENCES :

1. S.Ghaemi, H. Khazaei and P. Musilek,(2020)"ChainFaaS: An Open Blockchain-Based Serverless Platform," 2020 in IEEE Access, vol. 8, pp.
2. R. A. P. Rajan,(2018)"Serverless Architecture - A Revolution in Cloud Computing," 2018 Tenth International Conference on Advanced Computing (ICoAC), Chennai, India, 2018, pp. 88-93, doi: 10.1109/ICoAC44903.2018.8939081.
3. M. Golec, D. Chowdhury, S. Jaglan, S. S. Gill and S. Uhlig, (2022)"AI BLOCK: Blockchain based Lightweight Framework for Serverless Computing using AI," 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Taormina, Italy, 2022
4. M. M. Martinez and S. R. Pandey,(2022)"Predictive function placement for distributed serverless environments," 2022 25th Conference on Innovation in Clouds, Internet and Networks (ICIN), Paris, France,
5. N. S. Kumar and S. S. Selvakumara(2022),"Serverless Computing Platforms Performance and Scalability Implementation Analysis," 2022 International Conference on Computer, Power and Communications (ICCPC), Chennai, India
6. B. Jambunathan and K. Yoganathan, "Architecture Decision on using Microservices or Serverless Functions with Containers,2018 International Conference on Current Trends towards Converging Technologies (ICCTCT), Coimbatore, India
7. J. L. Vázquez-Poletti and I. M. Llorente, "Serverless Computing: From Planet Mars to the Cloud," 2018,in Computing in Science & Engineering, vol. 20, no. 6, pp. 73-79, 1 Nov.-Dec. 2018
8. J. Cho and Y. Kim, "A Design of Serverless Computing Service for Edge Clouds," 2021 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, Republic of, 2021

9. N. S. Kumar and S. S. Selvakumara, "Serverless Computing Platforms Performance and Scalability Implementation Analysis," 2022 International Conference on Computer, Power and Communications (ICCPC), Chennai, India, 2022, pp. 598-602
10. N. Sisyukov, O. S. Yulmetova and V. A. Kuznecov, "GPU Accelerated Industrial Data Analysis in Private Cloud Environment," 2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Saint Petersburg and Moscow, Russia, 2019, pp. 348-352, doi: 10.1109/EIConRus.2019.8656751.