International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V72I4P109

Volume 72 Issue 4, 71-76, April 2024
© 2024 Seventh Sense Research Group®

Original Article

Exploring the Challenges of Serverless Computing in Training Large Language Models

Kushal Walia

Amazon Web Services.

Corresponding Author : waliakushal@gmail.com

Received: 15 February 2024 Revised: 20 March 2024 Accepted: 08 April 2024 Published: 19 April 2024

Abstract *- This paper delves into the exploration of utilizing serverless computing frameworks for the training of Large Language Models (LLMs), a cornerstone of modern artificial intelligence and machine learning advancements. While serverless computing offers significant benefits, including reduced infrastructure costs and enhanced scalability, its application in the context of LLM training introduces a unique set of challenges and limitations. Through an in-depth analysis, this study identifies key obstacles such as statelessness, execution time limits, cold start latency, resource constraints, data management complexities, dependency management, and cost predictability issues that inherently complicate the deployment of LLM training pipelines in a serverless environment. Despite these hurdles, the potential of serverless computing to revolutionize the scalability and cost-efficiency of LLM training remains undeniable. By presenting a balanced view on the feasibility, challenges, and prospective solutions, this paper aims to provide insights into the current state and future possibilities of serverless computing in the realm of large language model training, marking a critical step towards optimizing computational resources in the advancement of AI technologies.*

Keywords *- Artificial Intelligence, Cloud Computing, Generative Pretrained Transformer, Large Language Models, Serverless Computing.*

1. Introduction

Large language models (LLMs) have emerged as a cornerstone of contemporary artificial intelligence (AI) research and applications, driving significant advancements across a variety of domains, including natural language processing (NLP), machine translation, and automated content generation. These models, characterized by their deep learning architectures and massive parameter counts, require substantial computational resources for training on extensive datasets. Traditionally, the training of such models has relied on dedicated hardware or cloud-based virtual machines configured to meet the intensive demands of computational power, memory, and storage (Brown et al., 2020; Devlin et al., 2018).

In parallel, the advent of serverless computing has revolutionized the landscape of cloud computing by offering a model where customers can execute code in response to events without managing the underlying compute resources. This paradigm, often associated with Function as a Service (FaaS) and Backend as a Service (BaaS), promises scalability, flexibility, and cost-efficiency, particularly for applications with variable workload patterns (Baldini et al., 2017). Major cloud providers, including AWS, Google Cloud, and Azure, have rapidly expanded their serverless offerings, highlighting their growing importance in the cloud computing ecosystem.

The potential of leveraging serverless computing for training large language models lies in its ability to dynamically scale computing resources to match the variable computational demands of the training process. Additionally, the pay-as-you-go pricing model of serverless computing can potentially offer cost savings for the training of LLMs, which typically require substantial investment in computational resources. However, the adoption of serverless computing for this purpose is not without challenges. The stateless nature of serverless functions, execution time limits, cold start latencies, resource constraints, and the complexity of managing large datasets in a serverless environment pose significant hurdles (Fox et al., 2017; Wang et al., 2018).

This paper aims to explore the challenges and limitations of using serverless computing architectures for the training of large language models. By examining the unique characteristics of serverless computing and the specific requirements of LLM training, we seek to identify the key obstacles to the adoption of serverless computing in this context and propose potential directions for overcoming these challenges. Through this exploration, we contribute to


This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)

the ongoing discussion on the scalability and efficiency of LLM training, highlighting the role of innovative cloud computing solutions in facilitating the next generation of AI advancements.

2. Background

2.1. Serverless Computing Architecture

Serverless computing, a paradigm shift in cloud computing, abstracts the complexities of server management away from the developer, offering a model where the cloud provider dynamically manages the allocation of machine resources. Predominantly characterized by Function as a Service (FaaS) and Backend as a Service (BaaS), serverless computing enables developers to build and run applications and services without the need to manage infrastructure. FaaS allows developers to execute code snippets in response to events without concerning themselves with the underlying compute resources, while BaaS provides a suite of automatically managed backend services. Major cloud providers, including Amazon Web Services (AWS) Lambda, Google Cloud Functions, and Microsoft Azure Functions, lead the market in offering these services, facilitating a wide array of computing tasks with scalability, high availability, and a pay-for-what-you-use pricing model (Baldini et al., 2017; Roberts & Chapin, 2017).

2.2. Large Language Model Training

The training of large language models (LLMs) involves deep learning techniques to develop models capable of understanding and generating human-like text. These models, such as GPT (Generative Pretrained Transformer) and BERT (Bidirectional Encoder Representations from Transformers), have millions to billions of parameters and require extensive computational resources for training on large datasets. The training process involves backpropagation and gradient descent algorithms, running iteratively over vast amounts of text data to adjust the model parameters for better performance on language tasks (Devlin et al., 2018; Brown et al., 2020). This process is resource-intensive, requiring significant computational power, memory, and storage, traditionally met by dedicated GPU clusters or cloud-based virtual machines configured for high-performance computing tasks.

2.3. Traditional Infrastructure for LLM Training

The conventional approach to training LLMs has relied on dedicated hardware or virtualized environments that can provide the necessary computing, memory, and network resources. Dedicated GPU clusters, often in on-premises data centers or provisioned through cloud services, are commonly used to meet the high demand for parallel processing capabilities essential for efficiently training LLMs. These setups offer the advantage of dedicated resources and potentially lower latency but come with higher upfront costs and the complexity of managing and scaling physical infrastructure. Cloud-based virtual machines and managed AI services offer more flexibility and scalability but can also incur significant costs, especially for training models over extended periods (Hazelwood et al., 2018; Lample & Conneau, 2019).

The juxtaposition of serverless computing's dynamic scalability and cost-efficiency against the traditional, resource-intensive model of LLM training sets the stage for an intriguing exploration of how these seemingly disparate technologies can intersect. This background lays the foundation for understanding the complexities and potential of leveraging serverless computing for the training of large language models, a subject of growing interest as the fields of AI and cloud computing continue to evolve.

3. Challenges in Using Serverless for LLM Training

The integration of serverless computing architectures for the training of large language models (LLMs) presents several unique challenges that stem from both the inherent characteristics of serverless computing and the specific requirements of LLM training. These challenges pose significant barriers to the efficient and effective use of serverless technologies for this purpose.

3.1. Statelessness and Execution Time Limits

Serverless computing environments, such as AWS Lambda or Google Cloud Functions, are designed to be stateless. This means that each function execution is independent, with no inherent way to maintain state between invocations. For LLM training, which requires iterative updates to model parameters over extended periods, this statelessness poses a significant challenge. Models must either be stored externally and loaded for each invocation, introducing significant overhead, or divided into smaller tasks that can complete within a single invocation, potentially complicating the training process and reducing efficiency (Fox et al., 2017; Wang et al., 2018).

Moreover, serverless functions are subject to execution time limits imposed by cloud providers, typically ranging from a few minutes to 15 minutes. Given that LLM training can take hours to weeks, these time constraints necessitate a complex orchestration of function invocations to continue the training process over time, complicating the training architecture and potentially increasing overhead (Baldini et al., 2017).

3.2. Cold Start Latency

Cold starts occur when a serverless function is invoked after being idle, requiring the cloud provider to allocate resources and bootstrap the runtime environment before executing the function. This latency can significantly impact the performance of LLM training tasks, particularly when

functions are invoked frequently during the training process. The variability in startup times adds an additional layer of unpredictability and inefficiency, making it difficult to maintain consistent training performance (McGrath & Brenner, 2017).

3.3. Resource Limitations and Scalability

While serverless computing offers the ability to scale function instances automatically, each instance is subject to resource limitations, including CPU power, memory, and temporary disk space. These limitations are often significantly lower than what can be achieved with dedicated servers or virtual machines optimized for high-performance computing. LLM training, with its high demand for computational resources and large in-memory data sets, may hit these resource ceilings quickly, leading to suboptimal training performance or the need for complex workarounds to distribute training across multiple function instances (Lloyd et al., 2018).

3.4. Data Management and Transfer

Training LLMs require access to large datasets, often several gigabytes to terabytes in size. Serverless functions, however, are designed to operate on smaller, event-triggered inputs and outputs. The challenge of managing and transferring large volumes of data in a serverless environment includes not only the technical limitations on data payload sizes for function invocations but also the network latency and costs associated with moving large amounts of data to and from the serverless environment and external storage solutions (Jonas et al., 2019).

3.5. Dependency Management

LLM training often requires complex software dependencies, including deep learning frameworks and libraries. In a serverless environment, there are limits on the size of the deployment package, which includes the function code and its dependencies. Managing these dependencies within the size constraints and ensuring they are properly initialized at runtime can be a complex and time-consuming task, potentially limiting the choice of tools and frameworks available for LLM training in a serverless context (Hellerstein et al., 2018).

3.6. Cost Predictability and Optimization

While serverless computing operates on a pay-as-you-go model, which in theory can offer cost savings for compute-intensive tasks like LLM training, predicting and optimizing costs in practice can be challenging. The dynamic scaling of serverless functions in response to training tasks can lead to unpredictable costs, especially when functions are inefficiently triggered or run longer than necessary.

Additionally, the costs associated with data transfer and storage, particularly for large datasets required for LLM training, can add significant and sometimes unexpected expenses to the overall cost of training (Yussupov et al., 2019).

4. Case Studies on Serverless Training of Large Language Models

The exploration into serverless architectures for training large language models (LLMs) presents an intriguing opportunity to leverage the scalability and cost-efficiency of cloud computing. However, it also introduces a set of unique challenges due to the inherent limitations of serverless environments. Through detailed case studies on the serverless training of a BERT model on AWS Lambda and training GPT-3-like models on Google Cloud Functions, I uncover the complexities, optimization strategies, and potential alternatives that can pave the way for more effective implementations in the future.

4.1. Case Study 1: Serverless Training of a BERT Model on AWS Lambda

Context and Setup: Attempting to train a BERT (Bidirectional Encoder Representations from Transformers) model using AWS Lambda sought to investigate serverless computing's viability for NLP tasks. The experiment involved breaking down the BERT training process into smaller, manageable tasks that fit within the constraints of AWS Lambda, such as execution time and resource limitations.

4.1.1. Challenges Statelessness and Execution Time Limits

The serverless nature of AWS Lambda, with its stateless operations and strict execution time limits, significantly hindered continuous training processes, necessitating innovative strategies to maintain model state across invocations.

Resource Limitations

AWS Lambda is designed for short-term, serverless computing tasks. It has limitations on execution time (up to 15 minutes per invocation), memory (up to 10,240 MB), and storage (512 MB of ephemeral disk space per invocation). Training BERT, a large model requiring substantial computational resources, exceeds these limits.

Complexity of BERT

BERT-Large, for example, has 340 million parameters. Training such a model requires significant GPU or TPU resources over an extended period. This is far beyond what AWS Lambda is designed to handle.

Data Management and Transfer

Efficiently managing and transferring large training datasets within the serverless environment proved difficult due to AWS Lambda's limited temporary storage and the need for frequent data transfers. BERT is trained on a large corpus of text data. Managing this data within the constraints of AWS Lambda would be impractical.

Optimization Strategies

Incremental Training Approach: By breaking down the training process into smaller, incremental steps, the model could be trained within the Lambda execution time limits, though this approach required careful management of training data and state.

Optimized Data Handling

Strategies for optimizing data transfer and storage, such as compressing training data and utilizing AWS's more efficient data storage and transfer services, helped mitigate some of the challenges associated with data management.

4.1.2. Alternative Approach

Training BERT or similar models typically occur on cloud instances equipped with GPUs or TPUs, such as AWS EC2 instances with GPU support. For a rough cost and time estimate, let's consider using a popular instance type for machine learning tasks.

Estimation Instance Type

For example, using AWS p3.2xlarge instances with one NVIDIA V100 GPU.

Time to Train

Training BERT from scratch can take several days to weeks. As an estimate, it could take approximately 4-5 days of continuous training on a cloud instance with 4-8 GPUs.

Cost

AWS EC2 pricing varies by region and instance type. p3.2xlarge instances cost around $3.06 per hour on demand in the US East (N. Virginia) region. However, using Spot Instances can reduce costs by up to 70-90%. If I assume 4 days of continuous training on a p3.2xlarge instance, with 96 hours of training and $3.06 per hour, the total could would be around $293.76 (96 hours * $3.06/hour = $293.76).

4.2. Case Study 2: Training GPT-3-like Models on Google Cloud Functions

Context and Setup: The ambitious goal of training GPT-3-like models using Google Cloud Functions aimed to test the boundaries of serverless computing's capabilities in handling extremely large and complex models. This approach required segmenting the model training process to fit within the constraints of Google Cloud Functions.

4.2.1. Challenges

Scalability vs. Resource Limitations

Despite the ability to scale horizontally, the resource limitations per Cloud Function instance significantly impacted the ability to train large models efficiently.

Complex Dependency Management

The large size of GPT-3-like models and their dependencies exceeded the deployment package size limits for Cloud Functions, complicating the setup and execution of the training process.

Execution Time and State Management

The stateless nature and execution time constraints of Cloud Functions presented major obstacles to continuous, long-duration model training sessions.

Optimization Strategies

Distributed Training and Microservices Architecture: Utilizing a distributed training approach and microservices architecture allows for training different parts of the model in parallel across multiple Cloud Functions. However, this significantly increased the complexity of the training process. These case studies illuminate the challenges and potential strategies for using serverless computing in the training of LLMs. While serverless architectures offer promising benefits in terms of scalability and cost, significant hurdles remain, particularly for state management, resource limitations, and data handling. A hybrid approach, combining serverless functions with traditional cloud computing resources, presents a viable path forward, enabling the exploitation of serverless advantages for certain aspects of the training process while overcoming its inherent limitations for large-scale model training.

5. Discussion

The exploration into the utilization of serverless architectures for the training of large language models (LLMs) sheds light on both the innovative possibilities and significant challenges of this approach. Through the lens of the detailed case studies on AWS Lambda and Google Cloud Functions, I have identified key challenges, including statelessness, execution time limits, resource constraints, data management difficulties, dependency management complexities, and the unpredictability of costs. These challenges highlight the current limitations of serverless computing when applied to the computationally intensive and resource-demanding process of training LLMs like BERT and GPT-3.

5.1. Reflections on the Challenges

The statelessness and execution time limits of serverless functions, as illustrated in the case studies, necessitate complex workarounds for maintaining model state across invocations and segmenting the training process into smaller tasks. While these strategies enable the continuation of training beyond the time constraints, they introduce significant overhead and complexity, potentially impacting the efficiency and scalability of the training process.

Resource limitations present another critical hurdle, as serverless functions are not designed for the high memory and compute requirements of LLM training. This constraint not only limits the size of the models that can be trained but

also affects the batch size and learning rate, potentially leading to slower convergence and reduced model performance.

Data management and transfer challenges, highlighted in both case studies, underscore the difficulties of handling large datasets in serverless environments. The overhead of transferring data between external storage and serverless functions can significantly slow down the training process, emphasizing the need for optimized data handling and caching strategies. Dependency management in serverless environments complicates the deployment of LLM training tasks due to size limits on deployment packages and the cold start problem. This issue underscores the necessity for lightweight dependencies and efficient initialization of the training environment. Cost predictability and optimization emerge as concerns, given the variable computational demands of LLM training and the potential for inefficient resource utilization in a serverless setup. While serverless computing offers a pay-as-you-go model, managing and optimizing costs for large-scale machine learning tasks requires careful planning and monitoring (Yussupov et al., 2019).

5.2. Potential Solutions

The discussion of challenges and optimization strategies leads to the consideration of hybrid models that combine serverless computing with traditional cloud resources. Such an approach leverages the scalability and cost-efficiency of serverless for specific tasks (e.g., data pre-processing, model evaluation) while utilizing more powerful and stateful compute resources for the core training process. This hybrid model offers a pragmatic solution, balancing the strengths of serverless computing with the demands of LLM training.

Further research and development in serverless technologies could address some of the current limitations. Innovations in serverless architectures that provide longer execution times, higher resource limits, and improved state management capabilities could make serverless computing more feasible for LLM training. Additionally, advancements in data transfer technologies and dependency management could alleviate some of the current challenges associated with data handling and software dependencies in serverless environments.

The exploration of serverless computing for training LLMs highlights significant challenges but also reveals the potential for innovative solutions and hybrid approaches. As serverless technologies continue to evolve, there is a promising path forward for more efficiently leveraging these architectures in machine learning and AI research. The continued collaboration between cloud providers, researchers, and practitioners will be crucial in overcoming the current limitations and unlocking the full potential of serverless computing for training large language models.

6. Future Directions

The exploration of serverless computing for training Large Language Models (LLMs) has unveiled a rich landscape of challenges and opportunities. As we look to the future, several key directions emerge, promising to address the current limitations and harness the full potential of serverless architectures in the fields of machine learning and artificial intelligence.

6.1. Advancements in Serverless Computing Architectures

Future advancements in serverless computing architectures are crucial to overcoming the challenges identified in training LLMs. Innovations that extend execution time limits, enhance state management capabilities, and provide more generous resource allocations will be particularly impactful. Such improvements could make serverless computing a more viable platform for computationally intensive tasks, including the training of state-of-the-art LLMs.

6.2. Hybrid and Flexible Computing Models

The development of hybrid and flexible computing models that seamlessly integrate serverless computing with traditional cloud and dedicated hardware resources represents a promising direction. These models would offer the best of both worlds: the scalability and cost-efficiency of serverless for suitable tasks, combined with the computational power and statefulness of dedicated resources where necessary. Enhanced orchestration tools and platforms that facilitate the dynamic allocation of workloads based on their computational requirements and cost considerations will be key to realizing this vision.

6.3. Optimization Techniques for Serverless ML Training

There is a need for continued research into optimization techniques specifically tailored for machine learning training in serverless environments. This includes innovations in model checkpointing, data caching, and incremental training approaches that minimize the overhead and maximize the efficiency of serverless function invocations. Furthermore, developing serverless-specific frameworks and libraries that abstract away some of the complexities of managing state and dependencies could significantly lower the barrier to entry for utilizing serverless computing in ML tasks.

6.4. Improved Data Management Solutions

The challenge of data management in serverless computing calls for improved solutions that facilitate efficient data storage, access, and transfer. Advances in distributed file systems, data streaming technologies, and serverless databases could offer more efficient ways to handle the large datasets typical of LLM training. Additionally, tighter integration between serverless platforms and data storage services, possibly through new data transfer protocols or networking technologies, could reduce latency and bandwidth constraints.

6.5. Cost Management and Optimization Tools

As cost predictability and optimization remain challenges for serverless computing, the development of more sophisticated cost management and optimization tools will be essential. These tools should provide real-time monitoring and predictive analytics to help users understand and forecast their spending, identify inefficiencies, and automatically adjust resource utilization to optimize costs without compromising on performance.

6.6. Policy and Standards Development

Finally, the establishment of policies and standards around the use of serverless computing for machine learning could facilitate broader adoption and interoperability. Guidelines on best practices, security, privacy, and compliance issues specific to serverless ML training could help organizations navigate the complexities of deploying these technologies responsibly and effectively.

7. Conclusion

The exploration of serverless computing for the training of Large Language Models (LLMs) illuminates both the potential benefits and the significant challenges of this approach. Through an in-depth examination, including case studies on AWS Lambda and Google Cloud Functions, I have identified critical barriers such as statelessness, execution time limits, resource constraints, data management challenges, dependency management issues, and the intricacies of cost optimization. These findings underscore the complexity and nuances of leveraging serverless architectures for computationally intensive tasks like LLM training.

Despite these challenges, our discussion also highlights the emergence of innovative solutions and alternative approaches that demonstrate the evolving nature of serverless computing in the machine learning domain. The development of hybrid models, which combine the scalability and efficiency of serverless functions with the robust computational resources of traditional cloud or dedicated environments, presents a pragmatic path forward. Such models can exploit the strengths of serverless computing for specific aspects of the training pipeline while overcoming its inherent limitations for the core computational tasks involved in training large models. Looking ahead, the future directions outlined emphasize the need for advancements in serverless architectures, optimization techniques, data management solutions, cost management tools, and the establishment of best practices and standards. These developments promise to enhance the feasibility and efficiency of serverless computing for machine learning applications, including the training of LLMs.

In conclusion, while the current landscape presents significant hurdles to the widespread adoption of serverless computing for LLM training, the ongoing innovations in cloud computing and machine learning technologies hold great promise. The dynamic interplay between evolving serverless architectures and the growing demands of AI research foreshadows a future where the scalability, flexibility, and cost-efficiency of serverless computing can be fully harnessed for the advancement of machine learning and artificial intelligence. As we continue to navigate these challenges and explore new solutions, the potential for serverless computing to revolutionize the training of large language models remains a compelling and exciting prospect.

References

[1] Loana Baldini et al., "Serverless Computing: Current Trends and Open Problems," *Research Advances in Cloud Computing*, pp. 1-20, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[2] Tom B. Brown et al., "Language Models are Few-Shot Learners," *arXiv*, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[3] Jacob Devlin et al., "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," *arXiv*, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] Michael Armbrust et al., "*Above the Clouds: A Berkeley View of Cloud Computing*," Electrical Engineering and Computer Sciences, University of California at Berkeley, Technical Report, 2009. [Google Scholar] [Publisher Link]

[5] Kim Hazelwood et al., "Applied Machine Learning at Facebook: A Datacenter Infrastructure Perspective," *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, Vienna, Austria, pp. 620-629, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Michael Roberts, and John Chapin, Designing a Serverless Web Application, AWS Whitepaper, 2017. [Online]. Available: https://aws.amazon.com/whitepapers/serverless-architectures-with-aws-lambda

[7] Liang Wang et al., "Peeking Behind the Curtains of Serverless Platforms," *Proceeding of the 2018 USENIX Annual Technical Conference*, *(USENIX, ATC 18,*) Boston, MA, USA, 2018. [Google Scholar] [Publisher Link]

[8] Guillaume Lample, and Alexis Conneau, "Cross-lingual Language Model Pretraining," *arXiv*, 2019. [CrossRef] [Google Scholar] [Publisher Link]