NOVEMBER 2023 | Omer Dayan & Ekin Karabulut

A THROUGHPUT PERFORMANCE BENCHMARKING:

Pre-training NVIDIA NeMo GPT-3 on Kubernetes with Run:ai

Abstract

This benchmarking study provides an examination of distributed training throughput for GPT-3 models, offering insights into performance and efficiency gains via Kubernetes (K8s). Specifically, we investigate the throughput of two NVIDIA NeMo™ Megatron GPT-3 models, namely the 5B and 126M variants, across one, two and four nodes of an NVIDIA DGX BasePOD™ system. We leverage Kubernetes as our orchestration framework, enhanced by the integration of the Run:ai platform.

Our results demonstrate that Kubernetes exhibits nearly perfect linear scaling behavior for GPT-3 with 5B parameters, while it achieves perfect linear scaling for GPT-3 with 126M parameters. This performance affirms Kubernetes as a potent tool for training GPT-3-like large language models efficiently, especially in distributed settings, empowering researchers and practitioners in advancing language models and AI applications. Kubernetes scripts used in this study are accessible in our repository for researchers and developers who seek to harness the power of Kubernetes for distributed computing to train and optimize language models efficiently.

Table of contents

Introduction

In the rapidly evolving landscape of artificial intelligence, the performance and efficiency of training large-scale language models are of paramount importance. As the demand for sophisticated language understanding models continues to grow, the need for robust and efficient training infrastructure becomes increasingly critical. In this benchmarking whitepaper, we delve into an evaluation of the distributed training throughput of GPT-3 models in Kubernetes, a prominent and widely utilized orchestration platform, boosted by the capabilities of the Run:ai platform.

Our focus extends to evaluating two variants of the NeMo Megatron GPT-3 5B and GPT-3 126M models. With their scale and complexity, training these models require cutting-edge hardware and optimized software frameworks to achieve optimal performance. Therefore, for our hardware infrastructure, we harnessed the power of an NVIDIA DGX BasePOD™ consisting of 4 NVIDIA DGX™ A100 nodes. This hardware configuration, known for its high-performance networking, was instrumental in achieving the training throughput we present in this study. NeMo™ Megatron, a creation of the NVIDIA Applied Deep Learning Research team, represents a GPU-accelerated framework tailored for training and deploying transformer-based Large Language Models (LLMs). We use it for its ability to handle models of up to a trillion parameters, offering a cost-effective and swift path to train generative AI.

At the heart of our investigation lies Kubernetes; an open-source container orchestration platform. It has emerged as a de facto standard for cloud-native infrastructure management and has become the platform of choice for AI companies like OpenAI and Spotify, and new AI Cloud providers like Coreweave. Its dynamic nature and cloud-native architecture make it a prime contender for orchestrating distributed machine learning workloads.

In our experiments, we evaluate the training throughput using Kubernetes across 1, 2, and 4 nodes, examining both models. Our aim is to provide a comparison of training throughput for these models, shedding light on their performance characteristics and the efficiency gains realized through Kubernetes.

Our exploration would not be complete without the integration of the Run:ai platform, a pivotal component in our benchmarking on Kubernetes. Run:ai is an AI compute platform, which enhances the Kubernetes ecosystem with advanced scheduling capabilities, cluster management, and GPU virtualization techniques for AI containerized workloads, thereby streamlining the training and deployment processes of AI models and maximizing availability and utilization of AI compute.

Throughout this whitepaper, we outline the infrastructure setup for the Kubernetes environment. We delve into the benchmarking methodologies, the experimental setup, and the results obtained. Notably, we highlight the custom scripts sourced from the Run:ai k8s-launcher repository, adaptable for Kubernetes deployments—including standalone Kubernetes clusters. We make these scripts readily accessible in our repository, fostering an environment of collaborative exploration.

Data preprocessing

Dataset: The Pile
"The Pile" is a massive dataset that represents one of the largest publicly available collections of diverse natural language data. Compiled and released by OpenAI, "The Pile" is a culmination of various sources like books, articles, websites, and other written material from the internet.

With the intention of promoting advances in natural language processing (NLP) and machine learning (ML), the dataset spans an extensive range of topics, languages, and writing styles. It contains over hundreds of gigabytes of text data, making it a valuable resource for training state-of-the-art language models and developing AI systems capable of comprehending and generating human-like language. Its size and diversity enable researchers and developers to address a wide array of NLP challenges, leading to the advancement of various language-related applications across multiple domains.
The Pile is provided as 30 shards of 15 GB size each. For our benchmark purposes we will download, extract and preprocess only 2 shards.

Data Locality
The preprocessed data is available to all the nodes in the cluster through NFS server with Read/Write spec of approximately 500 GB/s.
We preprocess the data such that the length of every sequence is 2048 tokens.

Cluster preparation

Run:ai and Kubernetes
Kubernetes (K8s) is an open-source container orchestration platform designed to automate the deployment, scaling, and management of containerized applications. Originally developed by Google and now maintained by the Cloud Native Computing Foundation (CNCF), Kubernetes provides a powerful and flexible solution for running distributed applications in a cloud-native environment. With Kubernetes, developers can package their applications in containers, such as Docker, along with all their dependencies and configurations, ensuring consistent deployment across various environments.
Run:ai is installed on top of Kubernetes.
The cluster consists of:

- NFS Server
- 4 x NVIDIA DGX A100 Nodes, with a total of 32 x NVIDIA A100 Tensor Core GPUs with 80 GB of GPU memory each
- 8 x 200 Gb HDR NVIDIA InfiniBand connectivity per node

Benchmarking setup

Models

We will run the benchmarking on NVIDIA NeMo GPT 3 models in the following sizes:

- 126M parameters
- 5B parameters

Run

For running on Kubernetes we will use specifically written scripts and tools that can be found in Run:ai k8s-launcher repository.

Throughput calculation

The throughput stands for the amount of tokens processed per second over the job's entire runtime. We measure the training throughput by the following equation based on step train time:

$$\text{Tokens per second} = \frac{\text{Global Batch Size} \times \text{Sequence Length}}{\text{Seconds per Step}}$$

Results

In our experiments, we conducted training using different node configurations: 1, 2, and 4 nodes, each equipped with 8, 16, and 32 GPUs, respectively. Our main expectation was to see how throughput scales as we increase the number of nodes and GPUs on Kubernetes with Run:ai. To create a baseline for comparison, we used perfect linear scaling as a reference point, which demonstrates the theoretical performance if scaling were perfectly linear and then compared it with the throughput achieved. The linear factor we used helps us understand how the throughput scales with the number of nodes.

GPT-3 – 5B parameters

		1 Node/ 8 GPUs	2 Nodes / 16 GPUs	4 Nodes / 32 GPUs
Kubernetes + Run:ai	Tokens Per Second	48,131	95545	182,791
	Linear Factor / Perfect Linear Factor	1 / 1	1.98 / 2	3.78 / 4
	Perfect Linear Scale	48,131	96262	192,524

Table 1: Throughput Comparison of GPT-3 with 5B parameters for 1 Node, 2 Nodes and 4 Nodes

Starting with the GPT-3 5B, we measured the rate at which tokens were processed per second for each setup and then interpolated these measurements. Detailed values can be found in **Table 1**, and visualizations are provided in **Figure 1**. Our investigation revealed that Kubernetes showed almost linear scaling behavior for each scenario. For instance, when the number of nodes doubled from 1 to 2, it achieved a throughput factor of 1.98. With 4 nodes, it exhibited a scaling factor of 3.79.

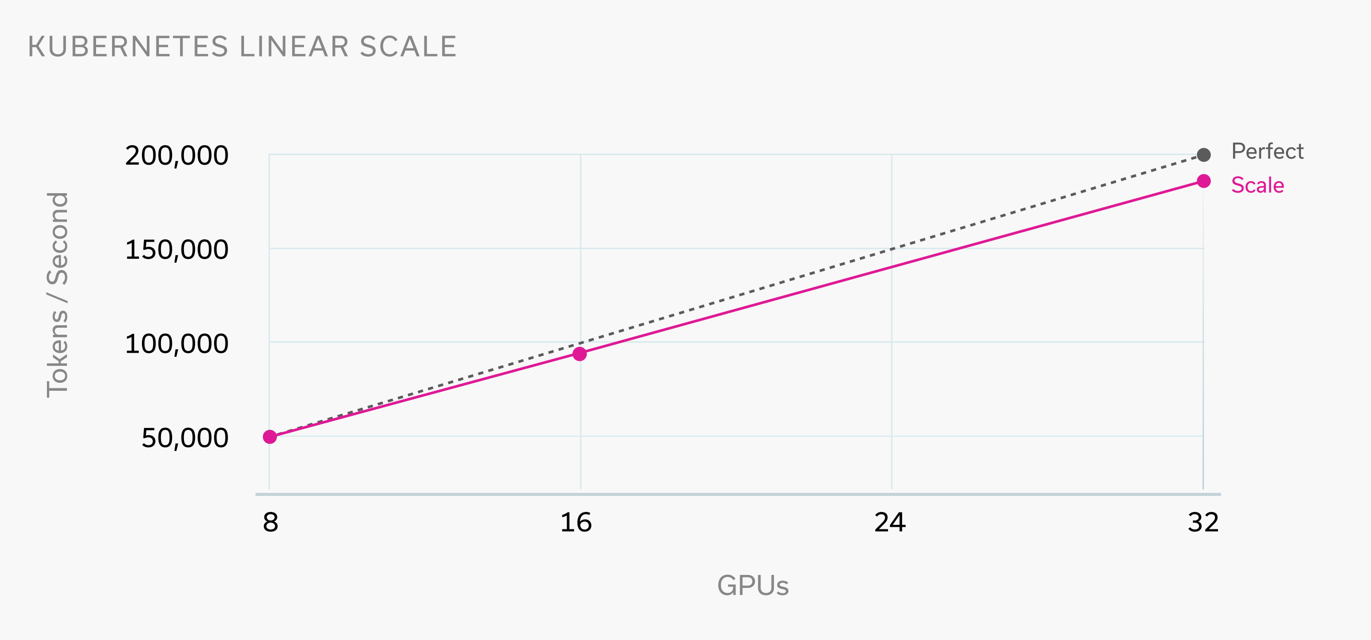

Figure 1*:* Tokens per seconds for 8 - 32 GPUs on Kubernetes with Run:ai (GPT-3 5B parameters)

GPT–3 – 126M Parameters

		1 Node/ 8 GPUs	2 Nodes / 16 GPUs	4 Nodes / 32 GPUs
Kubernetes + Run:ai	Tokens Per Second	892,208	1,724,417	3,568,835
	Linear Factor / Perfect Linear Factor	1 / 1	1.93 / 2	4 / 4
	Perfect Linear Scale	892,208	1,784,416	3,568,832

Table 2: Throughput Comparison of GPT-3 with 126M parameters

Turning to the GPT-3 126M, our subsequent experiments demonstrated nearly linear scalability in throughput when using Kubernetes with Run:ai again. The linear factor calculated for this scenario was 1.93 for 2 nodes and 4 for 4 nodes. For the case with 4 nodes, we noticed a slightly higher value in the throughput of Kubernetes in comparison to linear scale. These results lead to the conclusion that Kubernetes is particularly efficient for training GPT-3 with 126M parameters, while GPT-3 with 5B parameters still achieve throughput very close to the ideal linear scale.

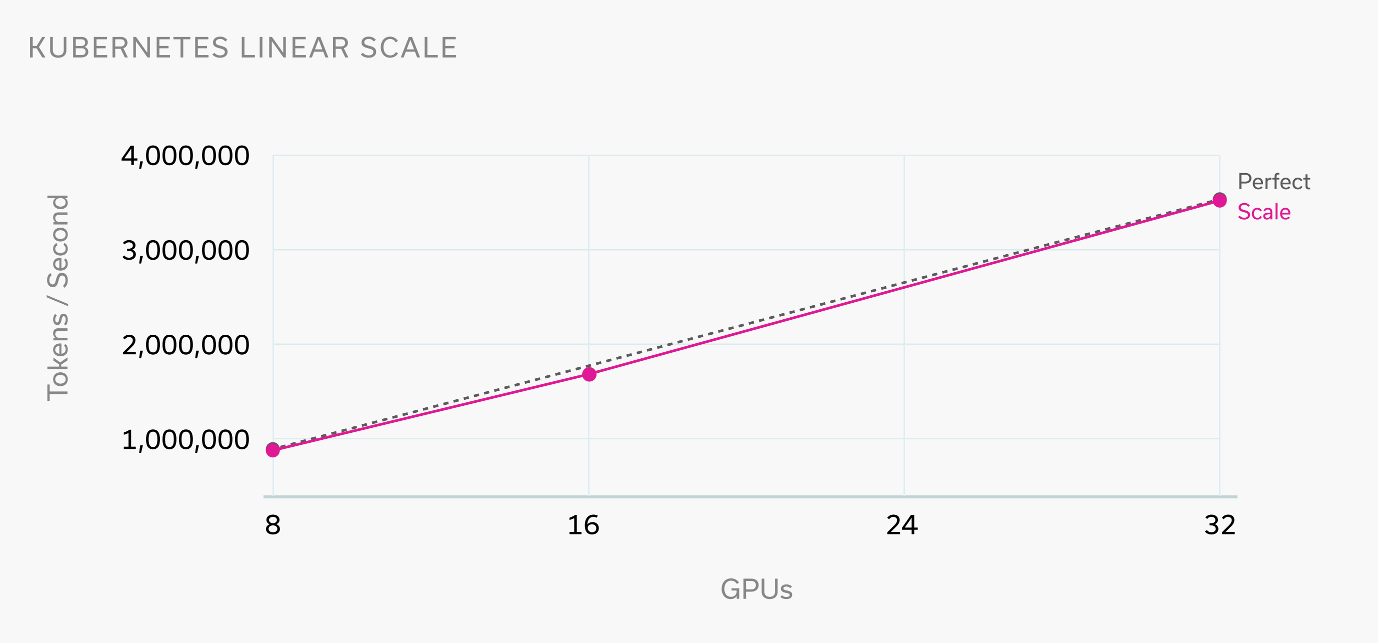

Figure 2: Tokens per seconds for 8 - 32 GPUs on Kubernetes with Run:ai (GPT-3 126M parameters)

Conclusion

In this benchmarking study, we focused on Kubernetes as the central platform, supported by Run:ai, to evaluate the distributed training throughput of GPT-3 models.

Our experiments involved various node configurations of an NVIDIA DGX BasePOD™ system to understand how throughput scales. For the GPT-3 model with 5 billion parameters, Kubernetes showed impressive scalability, closely following linear scaling. With 4 nodes, it achieved a significant scaling factor of 3.79.

When examining the GPT-3 model with 126 million parameters, Kubernetes once again displayed near-linear scalability. In the 4-node scenario, it even exceeded linear scaling. These results affirm Kubernetes as an efficient choice for training large language models.

In summary, our research highlights Kubernetes as a powerful tool for training large language models such as GPT-3 efficiently. Researchers and practitioners can use these insights to make informed decisions about infrastructure and configurations, advancing language models and AI applications.
Additionally, we're committed to ensuring the reproducibility of our experiments. As such, we've made our k8s launcher scripts available in our repository. This toolkit expands possibilities by providing a comprehensive suite of tools and scripts tailored for NVIDIA NeMo models. Designed to facilitate tasks from pretraining to fine-tuning and evaluation of expansive language models, this toolkit enhances the training experience for researchers and developers. By harnessing Kubernetes for distributed computing, we aim to simplify training processes and empower machine learning practitioners, fostering advancements in the field collectively.