Whitepaper

Launching & Running Large Language Models on a single Dell server produces outstanding results

Authors:
Ben Fauber, Ph.D.,
Senior AI Research Scientist
and Distinguished Technical Staff

Bhavesh Patel,
Senior Distinguished Engineer

Introduction

Large language models (LLMs) are artificial intelligence (AI) systems that use machine learning (ML) algorithms to process vast amounts of natural language text data. They have become increasingly popular due to their impressive natural language processing (NLP) capabilities.[1] Large pretrained language models are capable of extracting generalizations from vast amounts of text data, which can be utilized for a myriad of downstream applications such as text classification, text summarization, text generation, named entity recognition (NER), text sentiment analysis, and question-answering (Q&A). Additionally, many large language models are multilingual, making them even more versatile in utilizing text datasets across many different languages. This whitepaper will discuss one approach to developing and deploying LLMs on a single Dell PowerEdge server and the impressive results delivered over a traditional HPC architecture approach.

Large Language Models

In language model training, the task is to predict a word in a sequence of words. In 2017, a breakthrough was achieved with the introduction of transformers.[2] Unlike previous deep learning language models, transformers can process all input data at once and assign different weights to distinct parts of the input data based on their position in the language sequence. This feature has led to significant enhancements in large language models, enabling them to handle much larger datasets and infer deeper meaning and context by differentially weighing various parts of the sequence.

Recent advancements in natural language processing have resulted in remarkable large language model outcomes, exemplified by GPT-3/GPT-4 (OpenAI)[2,3] Megatron-Turing (NVIDIA and Microsoft),[4] OPT-175B (Meta),[5] Bloom-176B/Bloomz-176B (BigScience),[6,7] and most recently ChatGPT (OpenAI).[8] Many of these large language models provide state-of-the-art (SOTA) performance on natural language processing benchmark tasks such as:

- GLUE and SuperGLUE: The General Language Understanding Evaluation (GLUE) and SuperGLUE benchmarks include a set of diverse natural language understanding tasks, such as sentiment analysis, natural language inference, commonsense reasoning, textual entailment, and question answering.[9,10]

- SQuAD: The Stanford Question Answering Dataset (SQuAD) includes a set of reading comprehension tasks, where models are asked to answer questions based on a given passage.[11]

- SNLI and MultiNLI: The Stanford Natural Language Inference (SNLI) and the Multi-Genre Natural Language Inference (MultiNLI) corpora are benchmark datasets for natural language inference, where models are asked to determine whether a given hypothesis can be inferred from a given premise.[12,13]

- DREAM: The Dialogue-based Reading Comprehension (DREAM) dataset is a benchmark for evaluating models on answering questions that require reasoning and comprehension in a dialogue context.[14]

The large language models that wield the greatest power are extremely expansive, often boasting more than 100 billion model parameters and necessitating more than 800 GB of memory to activate, making them some of the largest machine learning models ever created. Many of these models require access to high-performance computing

(HPC) infrastructure, including low-latency networking, high-performance storage, and servers equipped with graphics processing units (GPUs), to achieve high throughput with minimal latency. These large language models have displayed remarkable abilities in producing text that is indistinguishable from human writing, tackling intricate inquiries, and even crafting computer code.

It is also noteworthy that large language models do not query the internet or other data sources for their knowledge. Rather, they use information and relationships embedded within their deep neural network (DNN) to provide responses to tasks and prompts. There are ongoing activities at Microsoft to merge their Bing internet search engine with the capabilities of ChatGPT.[15]

Enabling Secure Large Language Models within Your Infrastructure

Enable your organization with large language models secured within your infrastructure to securely interact with your data and employees. Open-source foundational large language models such as BLOOM-176B,[6] BLOOMZ-176B,[7] GPT-J-6B,[16] or OPT-175B[5] can be instantiated within your corporate information technology (IT) infrastructure. The biggest benefit of running a language model within your infrastructure is security: securely manage the model and secure the information it receives and outputs. Additional benefits include faster inference, greater availability and up time, reduced reliance on third-party services, maintain greater control of your own data and infrastructure, cost-effectiveness, and filtering the language model outputs to align them with your organization's domain and voice.

Building large language models specific to a certain business or vertical with transfer learning requires in-house machine learning expertise or an engagement with a trusted partner. Additionally, high-performance computing (HPC) hardware infrastructure such as multiple graphics processing units (GPUs), high-speed networking, and data storage capabilities are required to instantiate large language models.

Dell Technologies PowerEdge XE9680 Server

The Dell PowerEdge XE9680 is a high-performance server designed and optimized to enable uncompromising performance for artificial intelligence, machine learning, and high-performance computing workloads. Dell PowerEdge is launching our innovative 8-way GPU platform with advanced features and capabilities.

- 8x NVIDIA® H100 80GB 700W SXM GPUs or 8x NVIDIA® A100 80GB 500W SXM GPUs
- 2x Fourth Generation Intel® Xeon® Scalable Processors
- 32x DDR5 DIMMs at 4800MT/s
- 10x PCIe Gen 5 x16 FH Slots
- 8x SAS/NVMe SSD Slots (U.2) and BOSS-N1 with NVMe RAID

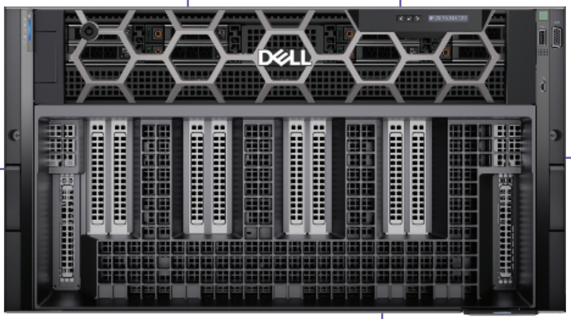
Dell PowerEdge XE9680

2 Socket Capable
- Up to two 4th Generation Intel® Xeon® Scalable processors with up to 56 cores per processor
- 6U air-cooled, up to 35C ambient
- 1200mm rack capable

Support for high-speed and memory capacity
- Up to 32 DDR5 DIMMs
- Up to 4800 MT/s(1DPC) or 4400 MT/s (2DPC)

AI

Large Model Training



I/O
- 10 x 16 PCIe Gen5 slots
- One OCP NIC 3.0
- 2 x 1GbE LOM

Support for up to 16 Drives
- Up to 8 x SAS/SATA/ NVMe Gen4 or 16x E3.S
- Rear Hot-Plug BOSS N -1 (2 x M.2 MVNe) for boot (optional)
- SW RAID/PERC12 support

GPU Optimized
- NVIDIA 8 x H100 SXM5 700W 80GB GPUs
  -or-
- NVIDIA 8 x A100 SXM4 500W 80GB GPUs
- Full NVLINK interconnectivity

	PowerEdge XE9680
CPU	2x Intel® Xeon® 8470 52-core Processor
GPU	8x NVIDIA® H100-SXM-80GB (700W)
System Memory	32x64GB – 2TB
Host NIC	NVIDIA® CX7

The PowerEdge XE9680 6U server is designed for AI, machine learning, and deep learning applications. It features the latest Intel Xeon processors with up to 56 cores, 8 NVIDIA® H100 or A100 GPUs, NVIDIA® NVLink™ technology for GPU-GPU communication, and supports up to 4 TB RDIMM of CPU RAM. The server supports virtualization options like NVIDIA® Multi-Instance GPU (MIG) capability, DDR5, NVLink™, PCIe Gen 5.0, and NVMe SSDs. It also supports NVIDIA® GDS (GPUDirect Storage), which provides a direct data path between GPU memory and storage, increasing system bandwidth and decreasing latency. The server is certified by NVIDIA® and has a secure, efficient, and comprehensive systems management solution with the OpenManage Enterprise console and iDRAC.

The results described in this article used a single XE9680 server with 8 x 80 GB H100 NVIDIA® HGX GPU cards with NVLink™ technology, 2 TB of CPU RAM, and 2 x Intel® Xeon® processors on each server. The XE9680 server was configured with the Ubuntu v22.04 Linux operating system, Anaconda v23.1.0, CUDA v12.1, cuDNN v8.8.1, and the same python dependencies as noted by in the original report of the large language model inference benchmark.[17] A full list of the python dependencies installed on the XE9680 server can be found in the Appendix section of this article.

Large Language Model Inference Benchmarks

Inferencing and training models with billions to trillions of parameters often faces hardware limitations. To accommodate these models within memory, current solutions make compromises between computation, communication, and development efficiency.

Microsoft has developed an open-source library called DeepSpeed, which improves large model training and inference by enhancing scale, speed, cost, and usability.[18] This library includes a new parallelized optimizer called Zero Redundancy Optimizer (ZeRO) that reduces the resources needed for model and data parallelism, allowing for the training and inference of 100-billion-parameter models.[19] ZeRO removes memory redundancies across data-parallel processes by partitioning model states, which allows per-device memory usage to scale linearly and fit models of arbitrary size. ZeRO offers three stages of acceleration, and stage 3 is the most relevant for inferencing. The large language model inferencing benchmarks for this section of the article were conducted with DeepSpeed ZeRO.

Accelerate is a HuggingFace library that simplifies PyTorch code adaptation for efficient operation on distributed and accelerated hardware.[20] It provides a high-level interface to parallelize and optimize code. The library supports PyTorch fully-sharded data parallelism (FSDP), which distributes optimizer states, gradients, and parameters across multiple devices, enabling larger batch sizes and more efficient use of hardware.[21] In addition to using FSDP for model sharding and distribution, the HuggingFace Accelerate library also supports mixed-precision training and inferencing for even more efficient computation.

PowerEdge XE9680 Server Benchmarking Results

Dell Technologies has demonstrated large language models of comparable scale and performance to GPT-3 can be launched and queried on our PowerEdge XE9680 server platform. The 176-billion parameter open-source large language models include BLOOM-176B and BLOOMZ-176B. Additionally, the 6-billion parameter GPT-J-6B, 7.1-billion parameter BLOOM-7B1,[6] BLOOMZ-7B1,[7] and 20-billion parameter GPT-NeoX-20B[22] open-source language models can also be hosted and queried on the XE9680.

Large language models require sizable memory and GPU capabilities at inference. In this instance, the open-source large language models BLOOM-176B and BLOOM-7B1 were launched, queried, and benchmarked on a single PowerEdge XE9680 server.

HuggingFace released a series of benchmarks for the large language model BLOOM-176B with DeepSpeed ZeRO[20] on the Jean Zay high-performance computing supercomputer.[23] A hardware configuration of 8 x 80 GB A100 NVIDIA® GPU cards with 512 GB of CPU RAM, and I/O networking read speeds of 3GB/s with general parallel file system (GPFS) was used on the Jean Zay system. Those same HuggingFace BLOOM-176B benchmarks were run on a single Dell Technologies XE9680 server, using the same HuggingFace code.[20]

Our PowerEdge XE9680 large language model BLOOM 176B DeepSpeed ZeRO (Figure 1) and HuggingFace Accelerate (Figure 2) benchmarking results both indicated a 50-70% speed-up of inferencing throughput vs. the Jean Zay supercomputer. Our results are especially noteworthy as a single PowerEdge XE9680 outperformed the Jean Zay supercomputer, which continually ranks as one of the fastest supercomputers in the world.[24]

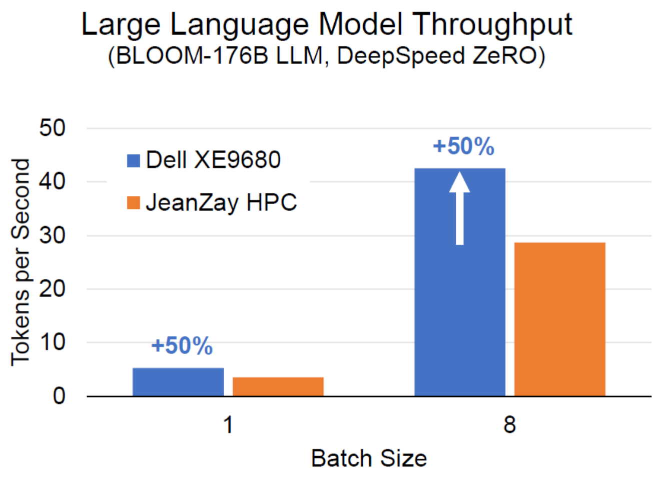


Figure 1. Throughput in tokens per second of the BLOOM-176B large language model (LLM) on the Dell Technologies PowerEdge XE9680 server (blue) vs. the Jean Zay supercomputer (orange) using the same benchmarking code and same model instantiation with Microsoft DeepSpeed ZeRO and float16 precision. In this instance, the XE9680 demonstrated a 50% increase in throughput as compared with the Jean Zay supercomputer, regardless of batch size.

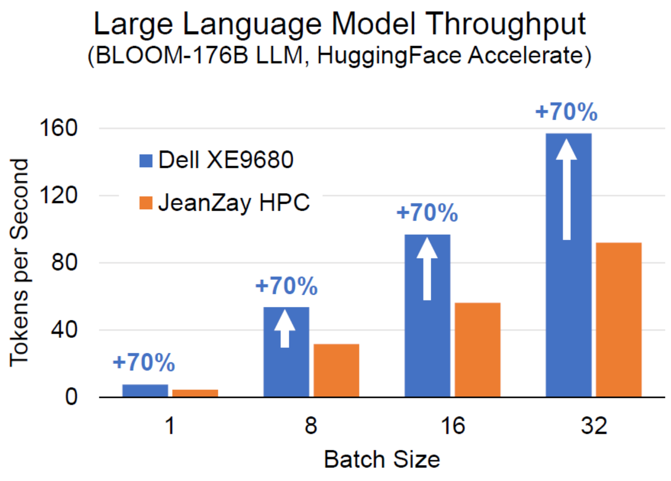


Figure 2. Throughput in tokens per second of the BLOOM-176B large language model (LLM) on the Dell Technologies PowerEdge XE9680 server (blue) vs. the Jean Zay supercomputer (orange) using the same benchmarking code and same model instantiation with HuggingFace Accelerate and float16 precision. In this instance, the XE9680 demonstrated a 70% increase in throughput as compared and the Jean Zay supercomputer, regardless of batch size.

The outcomes indicate that a single PowerEdge XE9680 server has the capability to initiate and operate a vast language model of comparable size and efficiency as ChatGPT within a solitary Dell Technologies server. Furthermore, the results demonstrate better model throughput and performance on publicly available benchmarks in comparison to one of the world's most robust supercomputers.

Impact of Software-Defined Features on Language Model Throughput

The impact of the language model loading method on throughput was assessed using the BLOOM-176B and BLOOM-7B1 open-source language models.[6] The same benchmarking method as used in the above section was also applied in this survey,[17] and the language model loading method for inferencing was varied. It should be noted that attempting to load the BLOOM-176B large language without Microsoft DeepSpeed ZeRO or HuggingFace Accelerate libraries for resource optimization resulted in out-of-memory (OOM) errors.

The BLOOM-176B large language model was launched for inferencing with the Microsoft DeepSpeed ZeRO library on the PowerEdge XE9680 server. The model was launched with float16 precision, and the batch size was varied to assess its impact on the large language model throughput, as assessed by tokens per second (Figure 3). The model scaled well with increases in batch size correlating to a polynomial increase in throughput, with batch sizes of 16 or greater leading to out-of-memory errors (OOM).

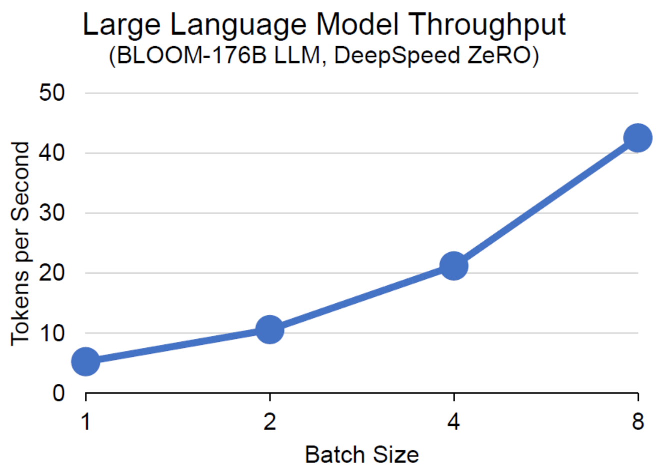


Figure 3. Throughput in tokens per second of the BLOOM-176B large language model (LLM) on the Dell Technologies PowerEdge XE9680 server vs. batch size with float16 precision. The model inferencing package was Microsoft DeepSpeed ZeRO with the benchmarking code.[17]

A similar throughput assessment of the BLOOM-176B large language model was conducted with float16 precision and model inferencing via the HuggingFace Accelerate library (Figure 4). The Accelerate library is optimized for memory distribution, thus the observation that larger batch sizes were permitted with this library versus DeepSpeed ZeRO are in line with the different objectives of the respective inferencing libraries. The model scaled well with increases in batch size correlating to a polynomial increase in throughput, with batch sizes of 64 or greater leading to out-of-memory errors (OOM).

An assessment of the BLOOM-7B1 language model was conducted with float16 precision and model inferencing via the Microsoft DeepSpeed ZeRO library to assess the impact of batch size on model throughput (Figure 5). The BLOOM-7B1 model is about 1/25th the size of BLOOM-176B, thus the smaller model could scale to larger batch sizes than those observed with BLOOM-176B (Figure 4). BLOOM-7B1 scaled well with increases in batch size correlating to a polynomial increase in throughput, with batch sizes of 1024 or greater leading to out-of-memory errors (OOM).

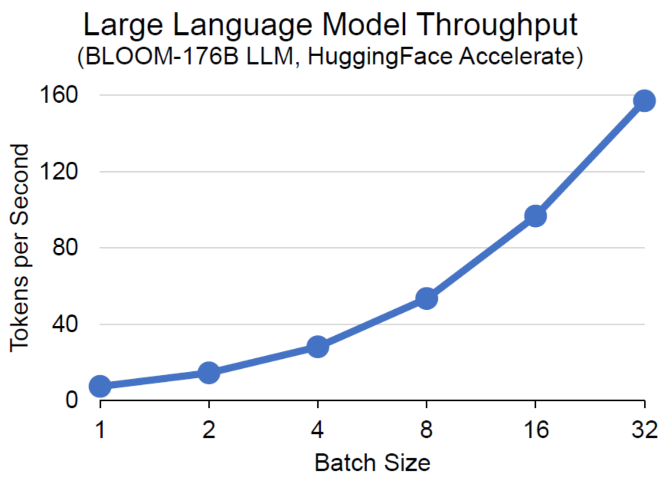


Figure 4. Throughput in tokens per second of the BLOOM-176B large language model (LLM) on the Dell Technologies PowerEdge XE9680 server vs. batch size with float16 precision. The model inferencing package was HuggingFace Accelerate with the benchmarking code.[17]

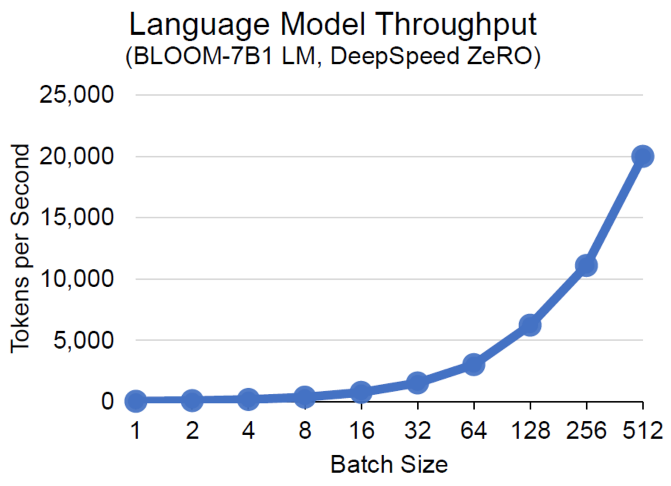


Figure 5. Throughput in tokens per second of the BLOOM-7B1 language model (LM) on the Dell Technologies PowerEdge XE9680 server vs. batch size with float16 precision. The model inferencing package was Microsoft DeepSpeed ZeRO with the benchmarking code.[17]

The BLOOM-7B1 language model was launched for inferencing with the HuggingFace Accelerate library on the PowerEdge XE9680 server. The model was launched using two separate precisions, on two separate run instances: float16 and int8. For each of these instances, the batch size was varied and the language model throughput, as assessed by tokens per second, was analyzed for each precision of the language model (Figure 6).
The BLOOM-7B1 language model with HuggingFace Accelarate and int8 precision allowed for similar token per second throughput to the float16 precision at smaller batch sizes of 4 or less (Figure 6). As the batch size increased, the float16 precision outperformed int8, resulting in several fold greater tokens per second language model throughput with float16 precision. The float16 precision allowed for greater batch sizes before reaching the limits of the system memory. The float16 instance achieved out-of-memory errors (OOM) with batch sizes of 1024 or greater, whereas the int8 instance achieved out-of-memory errors at batch sizes of 512 or greater.

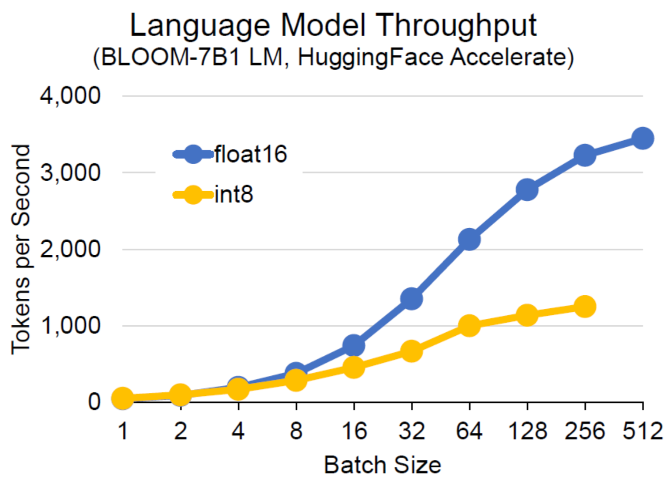


Figure 6. Throughput in tokens per second of the BLOOM-7B1 language model (LM) on the Dell Technologies PowerEdge XE9680 server vs. batch size for both float16 (blue) and int8 (gold) precision. The model inferencing package was HuggingFace Accelerate with the benchmarking code.[17]

Next, the impact of available GPUs on the language model throughput was assessed using the BLOOM-7B1 open-source language model.[6] The same benchmarking method as used above was applied in this survey,[17] and only the number of available GPUs was varied. A linear relationship was observed between the tokens per second throughput of the language model and the number of available GPUs when float16 precision and the Microsoft DeepSpeed ZeRO model loading procedure were employed (Figure 7).

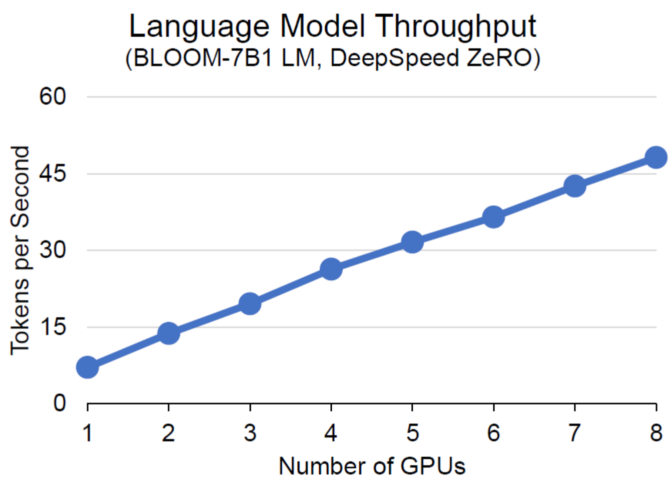


Figure 7. Throughput in tokens per second of the BLOOM-7B1 language model (LM) on the Dell Technologies PowerEdge XE9680 server (blue) vs. number of available GPUs using the model benchmarking code[17] and model instantiation with Microsoft DeepSpeed ZeRO and float16 precision.

In contrast to the Microsoft DeepSpeed ZeRO linear scaling result (Figure 7), varying the language model loading procedure to the HuggingFace Accelerate library resulted in a non-linear relationship between the model throughput and number of available GPUs (Figure 8). This finding is consistent with the goals of Microsoft DeepSpeed ZeRO and HuggingFace Accelerate, as DeepSpeed ZeRO is configured for near-linear workload scaling, whereas Accelerate attempts to distribute large models across all available memory for inferencing with limited memory burden.

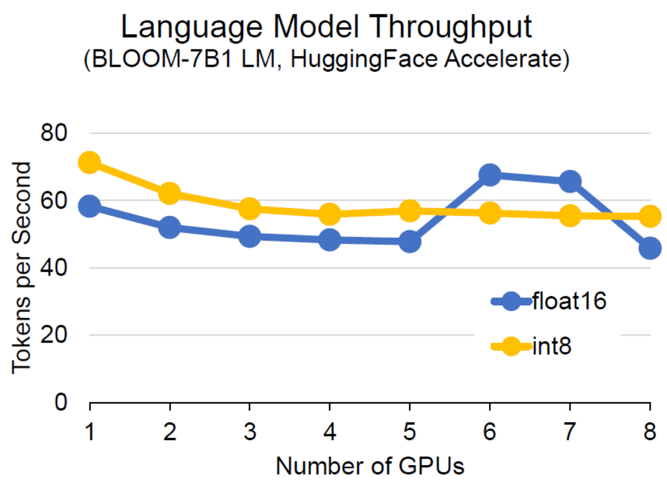


Figure 8. Throughput in tokens per second of the BLOOM-7B1 language model (LM) on the Dell Technologies PowerEdge XE9680 server vs. number of available GPUs using the model benchmarking code[17] and model instantiation with HuggingFace Accelerate and either float16 (blue) or int8 (gold) precision.

CONCLUSION

Our study has shown that the Dell Technologies PowerEdge XE9680 server can significantly enhance the performance of large language model inference and achieve impressive benchmarks, outperforming previous GPU architectures. We found that a single PowerEdge XE9680 server has the capability to create and run language models similar in size and capabilities to ChatGPT within a single Dell Technologies server. Our results also indicate superior model throughput and performance on publicly available benchmarks when compared to one of the world's most powerful supercomputers. Additionally, our study highlights the impact of language model throughput on the inference model loading library, model precision, batch size, and available GPUs. This information will be valuable for those interested in building large language models within their own secure infrastructure.

REFERENCES

1. Fauber, B. “Unleashing the power of large language models like ChatGPT for your business.” 2023, Dell Technologies white paper, https://www.delltechnologies.com/asset/en-us/solutions/infrastructure-solutions/industry-market/unleashing-the-power-of-large-language-models-fauber.pdf (accessed 11Apr2023).
2. Brown, et al. “Language models are few-shot learners.” Conference and Workshop on Neural Information Processing Systems Conference (NeurIPS) 2019.
3. OpenAI Reseaerch. “GPT-4 Technical Report.” 2023, arxiv.org/pdf/2303.08774
4. Alvi, et al. “Using DeepSpeed and Megatron to train Megatron-Turing NLG 530B, the world’s largest and most powerful generative language model.” 2021, https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model
5. Facebook AI Research. “Democratizing access to large-scale language models with OPT-175B” 2022, https://ai.facebook.com/blog/democratizing-access-to-large-scale-language-models-with-opt-175b/
6. BigScience Workshop, et al. “BLOOM: A 176B-parameter open-access multilingual language model.” 2022, arxiv.org/abs/2211.05100
7. Muennighoff, N. et al. “Cross lingual generalization through multitask finetuning.” 2022, https://arxiv.org/abs/2211.01786
8. Open AI Research. “ChatGPT.” 2022, https://chat.openai.com/
9. Wang, et al. “GLUE: A multi-task benchmark and analysis platform for natural language understanding.” International Conference on Learning Representations Conference (ICLR) 2019.
10. Wang, et al. “SuperGLUE: A stickier benchmark for general-purpose language understanding systems.” Conference Workshop on Neural Information Processing Systems Conference (NeurIPS) 2019.
11. Rajpurkar, et al. “SQuAD: 100,000+ questions for machine comprehension of text.” Proceedings of the Conference on Empirical Methods in Natural Language Processing Conference (EMNLP) 2016.
12. Bowman, et al. “A large, annotated corpus for learning natural language inference.” Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP) 2015.
13. Williams, et al. “A broad-coverage challenge corpus for sentence understanding through inference.” Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL) 2018.
14. Sun, et al. “DREAM: A challenge dataset and models for dialogue-based reading comprehension.” Transactions of the Association for Computational Linguistics (ACL) 2019.
15. Kan, M. “Free Sydney? Don’t worry, longer chats will return to Bing, Microsoft says.” 21Feb2023, https://www.pcmag.com/news/free-sydney-dont-worry-longer-chats-will-return-to-bing-microsoft-says
16. EleutherAI, et al. “GPT-J-6B.” 2021, https://huggingface.co/EleutherAI/gpt-j-6b (accessed 11Apr2023).
17. Bekman, S. and Gugger, S. “Incredibly fast BLOOM inference with DeepSpeed and Accelerate.” September 16, 2022, https://huggingface.co/blog/bloom-inference-pytorch-scripts
18. Rasley, J. et al. “DeepSpeed: System optimizations enable training deep learning models with over 100 billion parameters.” International Conference on Knowledge Discovery & Data Mining (KDD) 2020, Tutorial.
19. Rajbhandari, S. et al. “ZeRO: Memory optimizations toward training trillion parameter models.” 2019, arxiv.org/abs/1910.02054
20. Bekman, S. et al. “Fast Inference Solutions for BLOOM.” https://github.com/huggingface/transformers-bloom-inference (accessed 14Apr2023).
21. Ott, M. et al. “Fully sharded data parallel: Faster AI training with fewer GPUs.” 15July2021, https://engineering.fb.com/2021/07/15/open-source/fsdp/
22. Black, S. et al. “GPT-NeoX-20B: An open-source autoregressive language model.” 2022, arxiv.org/abs/2204.06745
23. IDRIS, national computing centre for the Centre National de la Recherche Scientifique. “Jean Zay: Introduction.” http://www.idris.fr/eng/jean-zay/jean-zay-presentation-eng.html (accessed 14Apr2023).
24. Top500.org “Top 500, The List: CNRS/IDRIS-GENCI » Jean Zay.” https://www.top500.org/system/179699/ (accessed 14Apr2023).

Appendix

All results in this article were collected from a Dell Technologies PowerEdge XE9680 server. The server was configured with 8 x 80 GB H100 NVIDIA® HGX GPU cards with NVLink™ technology, 2 TB of CPU RAM, and 2 x Intel® Xeon® processors. The server was configured bare metal with the Ubuntu v22.04 Linux operating system, Anaconda v23.1.0, CUDA v12.1, cuDNN v8.8.1, and the same python dependencies as noted by in the original report of the large language model inference benchmark.17 The python dependencies included accelerate v0.18.0, bitsandbytes v0.37.2, deepspeed v0.8.3, torch v2.0.0+cu118, and transformers v4.27.4. The BLOOM-176B and BLOOM-7B1 language model benchmarking code[17] was imported directly via git clone from https://github.com/huggingface/transformers-bloom-inference.

The language model benchmarking code could be executed in either the Jupyter Notebook (IPYNB) environment, or directly via the python3 command line interface (CLI). The benchmarks were conducted in the same manner as described in reference 17. Benchmarks were conducted with all 8 GPUs available, and either the model loading method (either Microsoft DeepSpeed ZeRO or HuggingFace Accelerate), the precision of the model (either float16 or int8), or the batch size of the language model (bs = {1, 2, 4, 8, 16, 32, 64, 128, 256, 512}) were altered. Only one variable was altered at a time. The model throughput per token, including the tokenizing step, were reported as outputs.

As examples, the following python3 CLI commands were used for benchmarking the BLOOM-176B language models with 8 GPUs, batch size=1, float16 precision, and varying the model loading method between Microsoft DeepSpeed ZeRO and HuggingFace Accelerate, respectively:

```
deepspeed --num_gpus 8 transformers-bloom-inference/bloom-inference-scripts/bloom-ds-zero-inference.py --name
bigscience/bloom --batch_size 1 --benchmark
```

```
CUDA_VISIBLE_DEVICES=0,1,2,3,4,5,6,7 python transformers-bloom-inference/bloom-inference-scripts/bloom-
accelerate-inference.py --name bigscience/bloom --dtype float16 --batch_size 1 –benchmark
```

Further analyses were conducted with the BLOOM-176B and BLOOM-7B1 language models to assess the model throughput per token, including the tokenizing step, as the precision, model inference loading method, batch size, and number of available GPUs were changed.

Learn more about
Dell solutions

Contact a Dell
Technologies Expert

View more resources

Join the conversation with
#PowerEdge
#GenerativeAI

DELL Technologies