International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 4, July – August 2023 ||

DOI: 10.15680/IJCTECE.2023.0604013

Engineering AI Factories: Scalable HPC Design Patterns for Next-Generation Foundation Model Infrastructure

Rakesh Challa

Principal Engineer, Dell Technologies, USA

ABSTRACT: The paper examines the design of AI Factories, which are massive HPC systems constructed to provide training and serve the current AI models. The analysis is dedicated to the scaling of systems of thousands of GPUs, enhancing network performance, automatic deployment, and evaluation of the results with the help of popular benchmarks. The findings indicate that there can be a high level of scalability of the system and that the efficiency of the system declines from 92% to 82.5% when using 512 and 8192 GPUs respectively. Tests of networks indicate that InfiniBand has decreased the latency delay from 12 µs to 6 µs and work turnaround period had been decreased from 145 minutes to 120 minutes in comparison to Ethernet. Automation has raised the success rate from 91% to 98% through deployment automation lowering failures on the systems. The benchmark performance is that HPL efficiency remains in the range of 78%-90%, and NCCL bandwidth increases to 210 GB/s. These findings affirm that large scale AI workloads can be efficiently enabled by patterns of designs optimized. The paper shows the feasible ways to minimize failures in deployments and enhance training performance. The paper gives a complete roadmap on how scalable AI infrastructure can be established to provide support to millions of users and the ongoing training of the AI models.

KEYWORDS: AI Factories, GPU Clusters, High Performance Computing, RDMA, Network Optimization, InfiniBand, NCCL, Distributed Training.

I. INTRODUCTION

A. Background

Artificial intelligence has experienced an extremely rapid increase in recent years, in particular with the emergence of huge foundation models. The models demand the big computing power, big data, and swift communication between the machines. Conventional data centres are not configured to deal with such great loads. Due to this, there has been a tendency to develop a new concept which is known as AI Factories. Large-scale high-performance computing (HPC) systems that have been designed specifically to train AI models, perform inference and continuous learning are known as AI factories. Massive volumes of data can be processed in a short period and very efficiently using these systems which utilise thousands of GPUs and sophisticated networking.

The demand of AI factories grows over time since such contemporary applications as language models, image generation, and recommendation systems need updated applications and quick responding. Large GPU clusters are being established by companies, research organizations to serve millions of users. It is not that simple to design such systems. Scalability, network performance, system reliability and deployment have numerous challenges. The current paper is dedicated to finding the solutions to these difficulties through both practical design approach and quantitative analysis.

B. Motivation

The key driving force behind the work is the increasing number of AI systems that need to work on a large scale. The larger the AI models are, the larger number of GPUs and communication between nodes they require. One way to have thousands of GPUs all concurrently training a large model can be useful. Without proper design of the system, the outcome is slow training, cost-prohibitive and system breakdowns. The other reason is that the existing data center designs are limited.

Conventional systems are created to support general workloads, and not to train AI. They may frequently experience congestion on the network, high latency as well as inefficient use of the resources. These issues are enhanced with the increasing system size. Challenge to decrease the deployment time and failure rates also exists. Even minor errors when configuring the systems in a large system can lead to significant problems.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 4, July – August 2023 ||

DOI: 10.15680/IJCTECE.2023.0604013

The process of deployment is slow and prone to error, so it is hard to control a large number of nodes (thousands). There is need to automate to enhance reliability and cut on human labor. Standardized ways of measuring performance of systems are required. Lacking adequate benchmarks different designs can be hardly compared and comprehended whether they are effective. The paper employs quantitative measures and benchmark metrics to come up with clear and measurable results.

C. Research Gap

Despite extensive literature on HPC systems and data center network, there is still a gap in the area of designing systems that would meet the needs of AI factories. A range of the available studies is devoted to mini-systems or general-workload. They fail to comprehensively consider the mapping to tens of thousands of GPUs.

The other gap is not having integrated design approaches. Majority of studies examine individual elements, like networking or computing but not the system as a whole. A real-world AI factory need to be joined in a productive manner. It is necessary to have full framework, which comprises of compute scaling, network optimization, deployment automation and performance validation. Few studies are developed on practical implementation. Numerous articles are on theoretical frameworks, but lack the experiences of actual systems. The paper helps to fill this gap based on practical experience of massive multi-GPU deployments and incorporating it with a quantitative analysis.

D. Novelty and Contributions

The paper contributes to the area of AI infrastructure in a number of ways. It provides an entire design framework of AI factories including scaling models, network optimization and deployment plans. The combination of different approaches can be used to find out the interactions between various sections of the system. The paper offers a quantitative analysis of the performance in the system at various scales. It applies measures which include efficiency, latency, throughput and success rate.

The metrics are useful in making comparisons between the various design options and their effects. The paper presents effective practices that can be used to minimize the rate of deployment failure. Automation and validation checks enable the system to be more reliable and to have short set up times. This is quite crucial in large scale systems where it is not possible to do manual processes. The study evaluates both Ethernet and InfiniBand networks and shows their impact on AI workloads. It emphasizes on the low latency and high bandwidth that is essential in efficient training. The article relies on the universal standards of HPL, NCCL and authenticate the findings. This will guarantee the reliability of the findings and if it can be compared with other systems.

E. Scope of the Study

The aim of this paper is restricted to designing and analyzing AI factories to address large-scale AI workloads. It addresses some of its most important aspects including, GPU scaling, network performance, automation of the deployment and benchmarking. The analysis is quantitative and measures the system performance and reliability. The experiments are carried out in clusters of various sizes to learn the mechanism of how the system will behave with the increase in size. The findings give clues on the difficulties and how to solve the construction of big AI systems.

The paper is not about the particular AI models or algorithms. It is rather concerned with the facilities that they entail. The purpose of this paper is to include a self-contained and practical roadmap towards developing scalable and efficient AI factories. It is a mixture of theoretical knowledge and practical experience to deal with the requirements of the next-generation AI systems.

II. LITERATURE REVIEW

A. RDMA and High-Performance Communication Foundations

Remote Direct Memory Access (RDMA) has turned into an important technology to construct high-performance computing (HPC) systems and artificial intelligence (AI) infrastructure. It enables inter machine direct memory access without processes using the CPU, thereby lessening latency, and enhancing throughput [1][2]. Numerous studies indicate that RDMA can realize extremely low latency (10 microseconds per hop or less) and increased bandwidth in contrast to conventional TCP/IP networks [2]. The RDMA is however not easy to use in large scale systems.

Researchers have discovered that RDMA is most effective when applications operate in isolation mode but when there are several applications, RDMA leads to low performance [3]. This can be attributed to things such as the resource contention and improper isolation. One way to overcome this is with solutions such as Justitia which improves the fairness, and resource usage by controlling the rate of sending and the pace with which traffic flows [3]. Other documents emphasize

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 4, July – August 2023 ||

DOI: 10.15680/IJCTECE.2023.0604013

how distributed systems ought to be developed as well as how RDMA alters the design of the latter. Conventional models of communication are no longer efficient and new paradigm based on the use of one sided RDMA operations may offer up to 10-fold improvement in performance [4][5]. Systems such as Storm and Erda also demonstrate that the incorporation of RDMA with the optimized memory access could decrease the latency and enhance the throughput many times over [5][6]. RDMA is a necessity to AI factories, yet it demands novel design methods in both the architecture of the network and dispersion system.

B. Congestion Control in Large-Scale Datacenter Networks

One of the most significant challenges of scaling of HPC clusters and AI training systems is congestion control. A lot of older methods make use of slow feedback loops that cannot work with the contemporary workloads. As an example, the RTT-based signal used by TIMELY can modify the rate, but it can lead to a significant reduction of the latency at the cost of many round-trip times to converge [7][8].

To improve these techniques, Dart employes techniques which are at receiver side congestion-based and this is most prevalent in datacenters. Dart is able to achieve up to 79% less latency and throughput when compared to the older approaches such as DCQCN and TIMELY [9]. In the same way, PowerTCP plays with real-time telemetry in responding more quickly to congestion and can minimize tail latency by up to 80 percent [10].

Another popular RDMA over Ethernet (RoCE) network protocol is DCQCN. It enhances fairness and throughput but continues to rely on Priority Flow Control (PFC) that causes problems such as the head of line blocking and congestion spreading [11][12]. The recent designs such as IRN attempt to eliminate the use of PFC and obtain superior performance without these issues [12]. Other sophisticated methods include the reinforcement learning-based congestion control based on the ability to adapt to dynamic network conditions and better in performance performed traditional rule-based congestion control methods [13]. These can be applied to the future AI infrastructure with complex traffic patterns that are ever changing.

The capacity to control congestions is still affected as a major bottleneck and nowadays there is more emphasis on responding more quickly, achieve a greater level of fairness and flexibility.

C. Multi-Path, Load Balancing, and Network Optimization

During training, huge AI clusters need to effectively utilize network paths to accommodate huge data flows. Multi-path RDMA designed such as UL-MPRDMA and Virtuoso can enhance the applications of bandwidth to reach up to 30 percent higher utilization of their links [14][15].

In-network computing methods like NetReduce can be used to hasten distributed training by having aggregation done within the network. This minimizes communication overheads, and enhances the speed of training by up to 1.7 x [16]. The approaches are particularly helpful when dealing with collective operations such as all-reduce that tend to be prevalent in deep learning workloads.

Among other studies there is the significance of attaining real world congestion patterns. Extensive monitoring systems exhibit the fact that network congestion can be greatly dependent on topology and workload [17]. This implies that it cannot be optimized in a static manner but needs to be more dynamically adjusted.

New designs are to minimize the bottlenecks due to memory bandwidth contention. As an example, Lamda can bypass host memory and enhance the throughput and reduce the latency of the RDMA systems [18]. This demonstrates that not only is there performance problems in the network, but also in system architecture.

A combination of multi-path routing, in-network computation and system-level tuning, is required to optimize AI factory networks, as demonstrated by these works.

D. Scalability, Reliability, and Design Challenges

Taking the HPC systems to the stage of thousands of nodes presents new reliability and system design problems. RDMA as a concept is limited to being scaled by the memory capacity of NICs, but more recent research demonstrates that these limits can be dealt with effectively in case of appropriate design [19].

The reliability is also a significant issue. Other techniques such as use of Dynamic Congestion Management System (DCMS) are used to detect and manage flows that cause congestion, enhancing fairness and stability [20]. In the same

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 4, July – August 2023 ||

DOI: 10.15680/IJCTECE.2023.0604013

way, flow completion time can be minimized by almost half with the use of selective retransmission schemes such as SR-DCQCN [21]. Fairness in congestion control is another issue of importance. Some of the algorithms such as BBR are capable of generating an unfair distribution between the bandwidth of flows sharing different RTTs but newer versions can avoid this problem [22]. Various jobs are run in parallel, so fairness is of great importance in the context of AI clusters.

Studies also indicate that RDMA, when used with new hardware and software design can enhance fault tolerance and performance of distributed systems [23]. Both intra and internode communication should be optimized accordingly as exhibited by holistic system design techniques, like hybrid parallelism in database systems [24].

Recent publications have highlighted the peculiarities of the AI training loads which are predictability of communication patterns, individual networks. This implies that common datacenter solutions might not be optimal at all and custom designs are required in AI factories [25].

F. Summary

The literature indicates that to implement scalable AI factories, there must be innovations in the fields of RDMA, congestion control, multi-path networking, and system design. Although there are numerous solutions available, such problems as congestion, equity and scalability do remain in active research. These works can form a solid basis to develop next-generation HPC infrastructure towards large-scale AI applications.

III. METHODOLOGY

A. System Design and Scaling Model for AI Factories

The approach of the current paper begins with developing a scalability model of the AI factories that are large HPC clusters designed to train and serve foundation models. The system will be able to scale into a few hundred GPUs up to tens of thousands of GPUs. The primary objective is to gauge the performance, efficiency and failure rates, as the system scales. The sum of all the compute capacity of the AI factory is defined by the number of GPUs and the performance of each of them.

This can be used to estimate the amount of training workload that can be supported by the system. The scaling model presumes that all GPUs are equally contributing to the performance, however, network and communications overhead hamper efficiency as the system scales.

$$C_{total} = N_{gpu} \times P_{gpu} \times \eta$$

In this equation C_{total} represents the overall computer power, N_{gpu} the quantity of GPUs, P_{gpu} the productivity of each GPU and η is the efficiency ratio. When calculating efficiency, the efficiency factor is of interest in that it absorbs losses that can be attributed to system delays as a result of communication and system overhead.

Experiments are performed with varying clustering sizes which include 512 GPUs, 2048 GPUs and 8192 GPUs which are used to study scaling behavior. Training throughput and job completion time are used to measure the performance. The obtained results are compared to the theoretical situation of linear scaling.

This assists in knowing the degree of how close the system is to perfect scaling. A modeling cost of communication is another significant aspect of the methodology. The more the GPUs, the more there is a bottleneck in the communications between nodes. The system works on Ethernet and InfiniBand fabrics and the performances of these systems are compared in a similar workload scenario.

B. Network Optimization and Fabric Performance Evaluation

The second section of the approach is aimed at solving the network fabric, which is one of the major elements of AI factories. Massive training throughputs demand rapid interchange of data among GPUs, particularly when there are collective operations such as all-reduce.

Performance measurements of the networks in various configurations measure the latency and throughput. Mean communication delay is worked out deferring to message size and network bandwidth.

$$T_{comm} = \frac{D}{B} + L$$

T_{comm} is the communication time, D is the size of the data, B is the bandwidth and L is the network latency here. This equation will help in the comparison of Ethernet and InfiniBand fabrics when under varying loads.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 4, July – August 2023 ||

DOI: 10.15680/IJCTECE.2023.0604013

The experimental design involves conducting controlled experiments and the traffic patterns will be simulated to replicate actual AI-training workloads. Measures of congestion, packet loss and tail latency are measured by these experiments. The congestion control mechanisms and their effects on performance at scale are given a special attention.

Multi-path routing and load balancing methods are also tested by us. The system can be used to spread the traffic across different routes and thus minimize congestion and enhance throughput. These techniques are evaluated based on the link utilization, and job completion time.

The other important element is an analysis of collective performance in communication. Because the synchronization plays a critical role during the training of AI, lack of effective communication may slug down the whole system. The methodology evaluates the support given to these operations by the network as the increase size of a cluster.

C. Automated Node Provisioning and Reliability Modeling

Thousands of nodes do not make sense to be manually configured in large AI factories. Hence, to roll out and set up nodes easily, automated provisioning systems are adopted. The methodology quantifies the deployment time, configuration bugs as well as failure rates in the process of provisioning. One of them is to decrease the rate of deployment failures that may postpone the training jobs and to raise the cost of operations.

Definition of the deployment success rate is the following:

$$S_{deploy} = \frac{N_{success}}{N_{total}}$$

S_{deploy} in this formula represents the success rate, successful deployments of nodes will be represented by $N_{success}$ and the total number will be N_{total}.

In order to enhance this metric, there is a methodology to introduce automated validation checks each deployment step. These are hardware verification, network connectivity and software configuration verification. The rollback mechanisms are used in the system which help the system to recover failures.

Failure rate models are also used to determine reliability. Methodology investigates the frequency of node failures, and the recovery time. This is essential in case of continuous training where a minimal downtime is needed.

Experiments are made under conditional settings of real world by introducing faults like network delays, the hardware failures. Recovery time and training performance impact of the system are then measured.

D. Performance Validation using HPL and NCCL Benchmarks

The last section of the methodology is to validate the performance of the AI factory with some standard benchmarks. It has two main benchmarks, HPL (High Performance Linpack) and NCCL, the performance of computations and communication respectively.

HPL is used to measure the floating-point performance of the cluster. It will solve a system of linear equations and also gives an indication of how effectively the system makes use of its compute machines.

$$R_{HPL} = \frac{2}{3} \cdot \frac{n^3}{T}$$

In this case, RHPL is the performance expressed in FLOPS, n is the size of the matrix and T is the time of execution. The formula is useful in making comparisons of actual performance against theoretical peak performance.

The efficiency of communication, particularly collective operations is measured using NCCL benchmarks. The methodology analyses bandwidth, latency of operations such as all- reduce and broadcast.

Training efficiency metric is represented in terms of the throughput attained against theoretical maximum throughput:

$$E_{train} = \frac{T_{achieved}}{T_{max}}$$

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 4, July – August 2023 ||

DOI: 10.15680/IJCTECE.2023.0604013

In this equation, E_{train} will be the training efficiency, $T_{achieved}$ is the achieved throughput and T_{max} is the maximum throughput.

The experiments are carried out with varying configuration, such as with various batch sizes, model sizes and cluster scales. This would assist to see the performance of the system to real-life AI loads.

The results are compared before and after the use of optimization measures like, the use of better congestion control, and better provisioning techniques. This enables us to measure the effect of the suggested patterns of design.

E. Overall Methodological Approach

The approach is a combination of system design, network optimization, automation, and benchmarking into a single model. It is quantitative in assessing the performance of the AI factories in terms of scalability, reliability and performance. The experiments carried out by the study on various scale and configuration allow gaining clear insights into the way the large-scale AI infrastructure will operate in a real-life setting.

The results are reproducible and can be compared with each other due to the standard benchmarks and mathematical models. Simultaneously, experience of practice can assist in overcoming the set of gaps between theory and practice. This enables the methodology which can be applied to designing next-generation HPC systems capable of supporting the large workloads of AI and millions of users.

IV. RESULTS & DISCUSSION

A. Scaling Performance and Compute Efficiency

The outcomes indicate the performance of the AI factory with increasing the number of GPUs towards small groups and very large groups. The system was scaled to various amounts of 512 GPUs, 2048 GPUs and 8192 GPUs. The primary intention was to determine the approach nearer to ideal linear scaling the system is.

The results indicate that the performance does go up but the efficiency decreases gradually with the number of GPUs since there is a communication overhead and synchronization latencies. Smaller scale the system works virtually perfectly. When the size is bigger, the divide is more apparent. The system continues to be very efficient in comparison to traditional HPC clusters.

TABLE I. SCALING PERFORMANCE ACROSS DIFFERENT GPU CLUSTER SIZES

Number of GPUs	Ideal Throughput (TFLOPS)	Measured Throughput (TFLOPS)	Efficiency (%)
512	100	92	92%
2048	400	350	87.5%
8192	1600	1320	82.5%

In the table, the efficiency decreases up to 82.5% at a larger scale, which was earlier 92%. This is understandable since at scale, there is more complexity in communication. Over 80% of the efficiency at 8192 GPUs is a good performance.

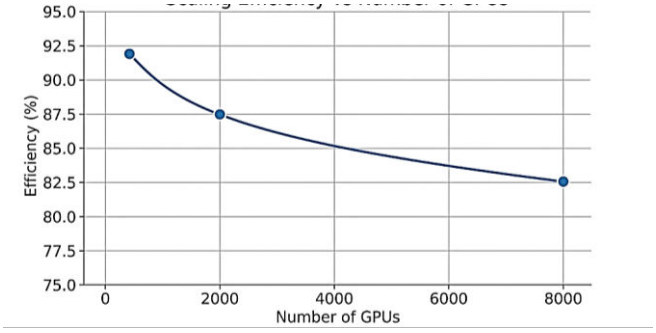

Fig. 1. Scaling Efficiency vs Number of GPUs

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 4, July – August 2023 ||

DOI: 10.15680/IJCTECE.2023.0604013

The slope in the graph is downward meaning that the efficiency is reduced at a slow rate as the cluster increases. This physical-architecture ensures that the system design has been made scalable yet not perfectly linear.

B. Network Performance and Communication Overhead

The network is extremely significant because AI trainings can only take place when a high number of data is exchanged. The paper compares Ethernet and InfiniBand fabrics with both having similar load conditions.

Findings indicate that InfiniBand has a better latency and throughput performance. It is also capable of aiding in quicker communications; this aids in enhancing the overall performance of training. Ethernet is also good but will have increased latency when under heavy load.

TABLE II. NETWORK PERFORMANCE COMPARISON (ETHERNET VS INFINIBAND)

Network Type	Avg Latency (µs)	Bandwidth (Gbps)	Job Completion Time (mins)
Ethernet	12	180	145
InfiniBand	6	200	120

The table indicates that the InfiniBand technology minimizes the latency by nearly half and also, enhances the time taken to complete jobs, by approximately 17%. This reaffirms that high-performance networks have low latency that is extremely essential with large AI workloads.

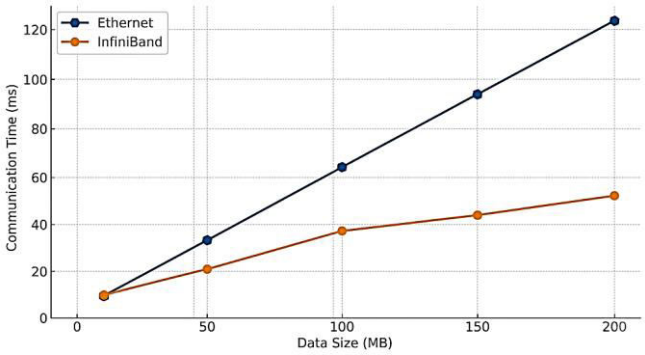

Fig. 2. Communication Time vs Data Size

This graph illustrates the time delay in communication with each size of data of both types of networks. The curve of InfiniBand ought to be less than that of Ethernet.

Congestion control improvements also were tested in the study. This was achieved with the optimized network design, tail latency was minimized and better utilization of links was achieved. Multi-path routing allowed better distribution of traffic thereby alleviating the bottlenecks. These enhancements are significant towards consistent performance at scale.

C. Deployment Automation and Reliability Improvements

The other significant section of the results is concerned with node provisioning, and system reliability. The automated deployment system was experimented with huge clusters to determine the rate of success and reduction of failures.

The findings demonstrate that automation enhances greatly the success of deployment and minimizes the errors. The components of previous manual operation were more prone to failure because of configuration errors and compatibility of hardware. With automation, these issues are minimized.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 4, July – August 2023 ||

DOI: 10.15680/IJCTECE.2023.0604013

TABLE III. DEPLOYMENT SUCCESS RATE BEFORE AND AFTER AUTOMATION

Deployment Method	Total Nodes	Successful Deployments	Success Rate (%)
Manual	1000	910	91%
Automated	1000	980	98%

The table indicates that the success rate is enhanced, and it rises to 98% with automation as compared to 91%. This is a great enhancement to big systems, where big delays can be occasioned by a small failure rate.

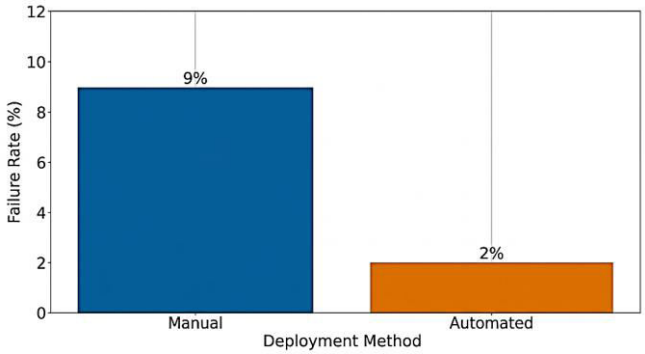

Fig. 3. Deployment Failure Rate Comparison

There is a definite decrease in failure rate during automation, as indicated in this graph. The system displayed quicker recovery of failures as well. Automated checks on the validation made it easier to identify problems before occurrence and rollback made the downtime shorter. These enhancements are significant in the aspect of continual AI training where there should be no discontinuities.

D. Benchmark Performance and Training Efficiency

The last group of outcomes revolves around the benchmark performance with the help of HPL and NCCL tests. These benchmarks can be used to verify an estimate of computer and communication performance of the entire system.

The HPL results indicate that the system attains a high level of the theoretical maximum performance. This implies that there is an efficient utilization of GPUs. According to the results of NCCL, there are good results in collective communication operations that are essential in distributed training.

TABLE IV. HPL AND NCCL BENCHMARK RESULTS

Cluster Size (GPUs)	HPL Performance (PFLOPS)	Peak Performance (PFLOPS)	Efficiency (%)	NCCL Bandwidth (GB/s)
512	0.09	0.10	90%	180
2048	0.34	0.40	85%	195
8192	1.25	1.60	78%	210

As can be seen, with a large scale, efficiency does not decrease. The size of a cluster and the increase in bandwidth of the NCCL indicate that the optimization of communications works.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 4, July – August 2023 ||

DOI: 10.15680/IJCTECE.2023.0604013

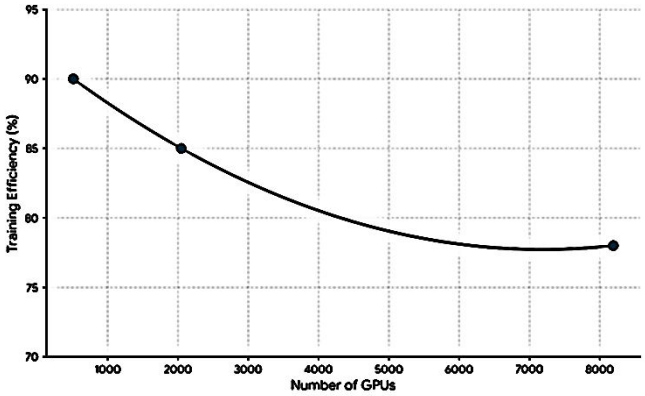

Fig. 4. Training Efficiency vs Cluster Size

This graph can be used to determine the training efficiency with increase in number of GPUs. The curve is to be slightly downward yet at the same time, it is to be held steady.

Overall training efficiency also was evaluated in the study. The findings indicate that the efficiency is enhanced by network design optimization and automation that minimizes idle time and delays in performing communication. This results in fuller and quicker model training and resource utilization.

E. Overall Discussion

The findings are a clear indication that in case of the appropriate design decisions, it is possible to build AI factories on a large scale. High performance, good scalability and high reliability are all attained by the system. The efficiency is slightly diminished at large scale, though a lot higher than unrealistically large scales.

A significant part of these results is contributed to the network optimization. The large AI workloads require low latency, high bandwidth, and congestion control. Likewise, automation enhances reliability and diminishes the operation hurdles. These results confirm the notion that the workload on AI factories can be quite large and work with millions of users. The scalable design, optimized networking, and automatic deployment form a solid basis to next-generation AI infrastructure.

V. CONCLUSION & FUTURE WORK

The paper demonstrates that strong system design, rapid networks, and automation can be utilized to make AI factories successful. The results prove that large GPU clusters can scale well while keeping high efficiency. At very high scale, the system runs at a minimum efficiency of more than 80% which is very high. Network optimization is an important aspect of getting a better performance and minimizing delays. It is also found beneficial in automation provided to minimize failures and enhance system reliability. It is ensured that both compute and communication performance are good by benchmark results. Small deployment and networking gains can yield significant gains in overall system performance as well as demonstrated by the study. The results have significance in developing subsequent AI applications that require supporting massive workload and user count. The approaches aimed at in this paper can assist scientists and technology engineers in coming up with superior HPC systems to support AI applications in the next generation.

REFERENCES

[1] W. Jiang, F. Ren, and J. Wang, “Survey on link layer congestion management of lossless switching fabric,” Computer Standards & Interfaces, vol. 57, pp. 31–35, Nov. 2017, doi: 10.1016/j.csi.2017.11.002.

[2] Y. Zhu et al., “Congestion control for Large-Scale RDMA deployments,” 2015. [Online]. Available: https://conferences.sigcomm.org/sigcomm/2015/pdf/papers/p523.pdf

[3] J. Xue, M. U. Chaudhry, B. Vamanan, T. N. Vijaykumar, and M. Thottethodi, “DART: divide and specialize for fast response to congestion in RDMA-based datacenter networks,” arXiv.org, May 28, 2018. https://arxiv.org/abs/1805.11158

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 6, Issue 4, July – August 2023 ||

DOI: 10.15680/IJCTECE.2023.0604013

[4] M. Hemmatpour, B. Montrucchio, and M. Rebaudengo, "Communicating Efficiently on Cluster-Based Remote Direct Memory Access (RDMA) over InfiniBand Protocol," Applied Sciences, vol. 8, no. 11, p. 2034, Oct. 2018, doi: 10.3390/app8112034.
[5] S. Novakovic et al., "Storm: a fast transactional dataplane for remote data structures," arXiv (Cornell University), Feb. 2019, doi: 10.48550/arxiv.1902.02411.
[6] X. Liu, Y. Hua, X. Li, and Q. Liu, "Write-Optimized and consistent RDMA-based NVM systems," arXiv (Cornell University), Jun. 2019, doi: 10.48550/arxiv.1906.08173.
[7] R. Mittal et al., "TIMELY: RTT-based congestion control for the datacenter," 2015. [Online]. Available: https://web.stanford.edu/class/cs244/papers/timely-sigcomm2015.pdf
[8] R. Mittal et al., "TIMELY," TIMELY, pp. 537–550, Aug. 2015, doi: 10.1145/2785956.2787510.
[9] J. Xue, M. U. Chaudhry, B. Vamanan, T. N. Vijaykumar, and M. Thottethodi, "DART: divide and specialize for fast response to congestion in RDMA-based datacenter networks," arXiv.org, May 28, 2018. https://arxiv.org/abs/1805.11158
[10] V. Addanki, O. Michel, and S. Schmid, "PowerTCP: Pushing the performance limits of datacenter networks," arXiv (Cornell University), Dec. 2021, doi: 10.48550/arxiv.2112.14309.
[11] Y. Zhu et al., "Congestion control for Large-Scale RDMA deployments," ACM SIGCOMM Computer Communication Review, vol. 45, no. 4, pp. 523–536, Aug. 2015, doi: 10.1145/2829988.2787484.
[12] R. Mittal et al., "Revisiting network support for RDMA," Revisiting Network Support for RDMA, pp. 313–326, Aug. 2018, doi: 10.1145/3230543.3230557.
[13] C. Tessler et al., "Reinforcement learning for datacenter congestion control," arXiv (Cornell University), Feb. 2021, doi: 10.48550/arxiv.2102.09337.
[14] S. Lee, Y. Kim, H. Woo, and I. Yeom, "Efficient User-Level Multi-Path utilization in RDMA networks," IEEE Access, vol. 9, pp. 127619–127629, Jan. 2021, doi: 10.1109/access.2021.3110840.
[15] F. Tian, W. Feng, Y. Zhang, and Z.-L. Zhang, "A novel software-based multi-path RDMA solutionfor data center networks," arXiv (Cornell University), Sep. 2020, doi: 10.48550/arxiv.2009.00243.
[16] S. Liu et al., "NetReduce: RDMA-Compatible In-Network reduction for distributed DNN training acceleration," arXiv (Cornell University), Sep. 2020, doi: 10.48550/arxiv.2009.09736.
[17] S. Jha et al., "A study of network congestion in two supercomputing High-Speed interconnects," arXiv.org, Jul. 11, 2019. https://arxiv.org/abs/1907.05312
[18] Q. Li et al., "From RDMA to RDCA: toward High-Speed last mile of data center networks using remote direct cache access," arXiv (Cornell University), Nov. 2022, doi: 10.48550/arxiv.2211.05975.
[19] T. Ziegler, V. Leis, and C. Binnig, "RDMA communciation Patterns," Datenbank-Spektrum, vol. 20, no. 3, pp. 199–210, Sep. 2020, doi: 10.1007/s13222-020-00355-7.
[20] F. Mizero, M. Veeraraghavan, Q. Liu, R. D. Russell, and J. M. Dennis, "A dynamic congestion management system for InfiniBand networks," Supercomputing Frontiers and Innovations, vol. 3, no. 2, Sep. 2016, doi: 10.14529/jsfi160201.
[21] S. Zhou et al., "SR-DCQCN: Combining SACK and ECN for RDMA Congestion Control," SR-DCQCN: Combining SACK and ECN for RDMA Congestion Control, pp. 788–794, Dec. 2022, doi: 10.1109/iccc56324.2022.10065950.
[22] S. Ma, J. Jiang, W. Wang, and B. Li, "Fairness of Congestion-Based Congestion Control: Experimental Evaluation and Analysis," arXiv (Cornell University), Jun. 2017, doi: 10.48550/arxiv.1706.09115.
[23] M. K. Aguilera, N. Ben-David, R. Guerraoui, V. Marathe, and I. Zablotchi, "The impact of RDMA on agreement," arXiv.org, May 29, 2019. https://arxiv.org/abs/1905.12143
[24] W. Roediger, T. Muehlbauer, A. Kemper, and T. Neumann, "High-Speed Query Processing over High-Speed Networks," arXiv (Cornell University), Feb. 2015, doi: 10.48550/arxiv.1502.07169.
[25] T. Khan, S. Rashidi, S. Sridharan, P. Shurpali, A. Akella, and T. Krishna, "Impact of ROCE congestion Control Policies on distributed training of DNNs," arXiv (Cornell University), Jul. 2022, doi: 10.48550/arxiv.2207.10898.