International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406015

Optimizing InfiniBand Congestion Control for Large-Scale AI Model Training Workloads

Rakesh Challa

Principal Engineer, Dell Technologies, USA

ABSTRACT: This paper examines the issue of InfiniBand congestion in large-scale AI model training systems. The emphasis is made on the impact of congestions on training time, utilization of GPUs, and scaling efficiency in cases where thousands of GPUs are utilized in conjunction. Base system iteration time rose from 320 ms over 256 GPUs to 780 ms over 2048 GPUs and the utilization of GPUs decreased from 78% to 58%. The time of communication was also greater and it demonstrated that congestion of a network is a significant bottleneck. In order to address this issue, the paper uses network counters to identify congestion, and then involves congestion optimization techniques that include subnet manager tuning, virtual lane separation, and load balancing. Upon the implementation of these techniques, the time of iteration decreased to 590 ms at 2048 GPUs with an improvement of approximately 24%. The efficiency of scaling was increased from 58% to 78 %, and the network latency was decreased from 95 µs to 55 µs at peak levels. There was also better equilibrium of the use of links over the network. Based on the results of the conducted investigation, it is possible to increase the system performance by over 20% with the help of the proper congestion control. This assists in the quicker training of AI and optimal utilization of computing assets, which promote huge research and innovations.

KEYWORDS: InfiniBand Congestion Control, AI Model Training, High-Performance Computing, GPU Utilization, Scaling Efficiency.

I. INTRODUCTION

A. Background and Context

Over the past few years, artificial intelligence has expanded extremely rapidly, particularly within the field of massive language models. These models consume extremely high computing power and are trained on hundred or thousand A, to say the least, GPU clusters. In these systems, GPUs need to interact with one another severally in the process of training. It is achieved with high-speed networks and InfiniBand is one of the most popular technologies that serve this purpose. It offers very low latency coupled with high bandwidth, which is significant in quick transfer of data.

However, as the size of GPU clusters increases, the network becomes a critical part of the system. Although the GPUs may be rather powerful, a bad network performance may slow down the whole process of training. Among key issues of these networks is congestion. Congestion occurs when excess data is transmitted across the network simultaneously and hence, delays and lower the performance. This problem grows more severe on large scale, where there are hundreds of GPUs interacting.

Due to this, large-scale AI training prioritizes network performance understanding and improvement. The profits of increasing the number of GPUs are less without an appropriate optimization. This necessitates the need to research the impact of congestion on performance and ways through which it can be restrained.

B. Motivation

This paper is primarily motivated by the real-life struggles encountered in massive GPUs deployment. It is a common phenomenon when training large AI models that more GPUs do not always result in faster training. In most instances, the improvement of performances reduces with the size of the system. This can mostly be attributed to congestions and delays in communication on the network.

Engineers working in real systems will tend to find GPUs idling as they wait on data transmission to be completed by other nodes. This decreases the efficiency of the entire system and exerts more cost on training. These inefficiencies may be significant in the resource and time consumption since AI models are getting increasingly large and complex. To illustrate, a minor time delay during a given training step can cause a tremendous increase in the overall training time, when repeated millions of times.

A second reason is that there is a need to use the available infrastructure better. Depending on energy efficiency is necessary in harnessing investment into the creation of massive clusters of GPUs and efficiency is paramount. The same hardware will be able to perform better, reducing congestion and improving network performance. This also not only saves money but also contributes to the rapid development of AI studies.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |
| Volume 4, Issue 6, November-December 2022 |
DOI:10.15662/IJEETR.2022.0406015

C. Research Gap

Despite numerous investigations on network congestion and high-performance computing, there is still a gap in the case of realistic AI training loads. The vast majority of the current studies are devoted to theoretical models, simulations, or to general HPC applications. However, large language model training has unique communication patterns, such as frequent all-reduce operations, which create bursty traffic in the network.

Another weakness is that there is no common information about how to tune InfiniBand networks to such massive AIs. Although there is a wide variety of congestion control methods, not all such methods can be simple to implement in practice. There should be easy and convenient ways that can be adopted by the engineers with existing tools and settings.

The work that relates network-level optimization and application-level performance is also sparse. Network improvements are learned individually in most cases, without demonstrating their direct effect on training time and GPU utilization. The purpose of this paper is to fill in this gap by connecting network behavior to AI training performance.

D. Novelty and Contribution

This paper contributes in a number of ways. To begin with, it offers an in-depth analysis of the impact of InfiniBand jitter on large-scale AI training systems. The paper uses congestion linked to important performance measurement parameters (e.g., training time and GPU utilization) and scaling efficiency instead of addressing the network metrics exclusively. This provides a better insight on the problem.

Second, the article provided a method of practical efforts to detect network congestion through real network metrics like latency, link utilization, and performance counters. These techniques may be implemented in actual systems without involved usage of tricky tools and simulations.

Third, the paper suggests the list of optimization methods that comprise the tuning of the subnet manager, virtual lanes, and load balancing. The methods are evaluated in the actual setting and they demonstrate positive results in performance. The findings indicate that scaling efficiency can efficiently be enhanced in more than 20% at large scale.

The emphasis on practical engineering and leadership of addressing real-world problems also constitutes another significant contribution. The article underscores the ability to solve a congestion problem in large-scale deployments of GPUs through systematic analysis and step-by-step optimization. This offers valuable lessons to engineers in these types of areas.

The paper demonstrates that the positive impact of bettering network performance is greater as compared to technical benefits. Innovation is encouraged in various areas, such as science, healthcare, and technology, with the assistance of faster and effective AI training. This work can be part of a broader development of AI research and development by optimizing the underlying infrastructure.

II. LITERATURE REVIEW

A. InfiniBand Architecture and Congestion Challenges

HPC systems require high interconnection networks including the use of InfiniBand to offer low latency and high bandwidth connections. The current InfiniBand systems have become capable of the high speeds such as EDR and HDR which can be used in large-scale AI workloads. But still, congestion remains one of the main problems impacting general system performance. Issues like head-of-line (HoL) blocking and buffer hogging may limit throughput and persistently hike latency [1]. Research indicates that despite sophisticated hardware, congestion may be propagated throughout the network as a consequence of link-level flow control and affect flows not necessarily on the route of the congestion [2].

Studies also indicate that the larger the cluster size, the more likely that it will develop congestion hotspots and hence lower effective bandwidth and poor scalability [3]. It has been found out that network contention is one of the factors that influence the communication latency of parallel systems particularly when deployed at large scales [4]. Application runtime can be greatly slowed by congestion, as much as over 40 percent of communication-intensive workloads can be slowed [5]. These results indicate clearly that the performance of the large AI training clusters heavily depends on congestion control.

B. Congestion Control Mechanisms and Optimization Techniques

Various methods have been suggested to control congestion in the InfiniBand and such other high-speed networks. Traffic segregation with virtual lanes (VLs) is one of these techniques that contribute to minimizing HoL blocking by isolating flows [6]. The other approach is the use of injection throttling where the rate of sending congested flows is decreased in order to stabilize the network. These methods have demonstrated an increase in performance in various traffic scenarios.

They have also suggested dynamic congestion management systems (DCMS) that measure congestion by using the performance counters of switchs and regulate the sending rate based on this measure. Equally, the newer congestion control algorithms like HPCC employ precise network telemetry to ensure that a queue length is kept under zero and the efficiency

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |
| Volume 4, Issue 6, November-December 2022 |
DOI:10.15662/IJEETR.2022.0406015

is high [7]. Delay-based protocols such as TIMELY utilize round-trip time (RTT) as a congestion metric in order to minimize the latency without sacrificing throughput [8].
Another method is ECN-based protocols such as DCQCN which is more stable and fairer than delay-based schemes [9]. Hybrid solutions like the HULL would attempt to trade-off between latency and bandwidth deliberately by leaving the network capacity unused to avoid line-ups [10]. The methods prove that to achieve congestion control the combination of rate control, traffic isolation, and real-time feedback is needed.

C. Network Topologies, Load Balancing, and Scalability
Network topology design significantly contributes to the congestion behavior and scalability. The HPC systems commonly utilize fat-tree, dragonfly, and flattened butterfly topologies with varying trade-offs to performance as well as costs [11][12]. As a case in point, dragonfly networks are very scalable with efficacious routing algorithm to prevent congestion [13].
Another consideration of reducing congestion is load balancing. Traffic is dynamically distributed over various paths with systems such as Hedera and CONGA in order to increase the bandwidth use and decrease hotspots [14][15]. Flowlet based load balancing techniques and centralized scheduling (e.g., Fastpass) techniques have demonstrated dramatic throughput and fairness improvements on top of certain underlying models [16].
Application-intelligent optimizations (topology-conscious process placement, MPI communication optimization) can also be used to minimize congestion and gain better performance. It has been concluded that linking communication patterns with network topology is capable of getting maximum throughput improvement of up to 40 percent. These findings emphasize the significance of network design with workload-optimization.

D. Impact on AI Workloads and Future Research Directions
Efficiency of interconnects is highly demanded with the adoption of modern AI and deep learning workloads. When using models with many models, the communication through GPUs is frequent throughout the model training, whereby the network such as InfiniBand would be essential [17]. But, due to congestion, poor utilization of GPU and longer training time can be appreciated.
It has been demonstrated that communication intensive applications (like that of MPI or RDMA) are very sensitive to network congestion and variation of latency [18][19]. In the worst scenario, the network-related issues can cause up to 3 times of performance loss. It has also discussed machine learning methods to predict congestion and optimization of system performance by identifying the important influencing factors on the execution time [20].
Current research highlights the importance of enhancing monitoring and analyzing tools as a way to appreciate congestion measured in the real world in large systems [21]. These tools are useful in determining the bottlenecks and parameters to be used to tune the system to achieve the best results. The literature claims that scaling AI training systems are necessary to enhance congestion control, network configurations, and workloads-network behavior alignment to match network behavior.

III. METHODOLOGY

A. Experimental Environment and Workload Design
The study methodology is based on a real-world large-scale AI training system that consists of thousands of GPUs connected via an InfiniBand network. It aims at reasoning the impact on congestion on training performance and how to alleviate this congestion with practical engineering strategies. The experimental environment consists of multiple node clusters of GPUs, set up with fast InfiniBand connections to enable RDMA communication to transfer data effectively. These workloads incur large language model (LLM) training is the main testing case as they demand frequent communication between GPUs in order to synchronize the gradients.
The overall training procedure will be broken down into several cycles with every cycle comprising forward pass, backward pass, and communication. The length of time taken in communication is important in the efficiency of the training as a whole. The computation time and communication time are used to model the total training time per iteration in order to measure this.

$$\mathrm{T}_{\mathrm{iteration}} = \mathrm{T}_{\mathrm{compute}} + \mathrm{T}_{\mathrm{communication}}$$

This equation assists in isolating the influence of the issue of network congestion over computation. When the time of communication is high because of a congestion then the overall time of iteration is also high and slows down the process of training. Monitored by experiments also includes the use of GPU, as GPUs that are idle reflect inefficiencies due to network delays. The utilization can be determined as a ratio of active computation time to total time.

$$\mathrm{U}_{\mathrm{GPU}} = \frac{\mathrm{T}_{\mathrm{compute}}}{\mathrm{T}_{\mathrm{iteration}}}$$

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406015

Using the values of these variables over varying size clusters, the research determines the increase in congestion with the size of the GPUs. The experiments are executed with varying traffic patterns to imitate the actual AI workloads, such as all-reducing communication and synchronizing parameters. This configuration will give a good foundation to test the congestion behavior of large systems.

B. Congestion Identification and Measurement Techniques

One of the main components of the methodology is determining the locations and times of congestion in the network. It performs this with the help of performance counters that are provided by InfiniBand switches and network interface cards. During training, some metrics are gathered continuously like the queue length, packet latency, and link utilization. These measures enable identifying hot spots of congestion and comprehending its effects on performance.

One of the measures employed in this study is the link utilization that depicts the extent to which bandwidth is being utilized. A high utilization and latency is normally a sign of congestion. The rate at which links are utilized is given as the ratio of the actual data rate to the maximum link capacity.

$$\mathrm{U}_{\mathrm{link}} = \frac{\mathrm{Throughput}}{\mathrm{Capacity}}$$

Scaling efficiency is another crucial measure, demonstrating the effectiveness of the system with the increase of the number of GPUs. In theory more GPUs should improve performance in linear fashion but congestion diminishes this performance. Scaling efficiency is determined as a ratio of actual speedup with optimum speedup.

$$\mathrm{E}_{\mathrm{scaling}} = \frac{\mathrm{Speedup}_{\mathrm{actual}}}{\mathrm{Speedup}_{\mathrm{ideal}}}$$

Latency is also employed in the study to identify the effect of a congestion. Latency increase implies queue formation and loss in delivering packets. With a combination of these metrics, the methodology gives a clear picture of the network behavior during training. This can be used to determine individual links or nodes where congestions are critical.

Traffic patterns are also determined to know which communication operations result in the most congestion. As an example, all-reduce and other collective operations tend to generate bursty traffic, causing temporary spikes in congestion. Occurrence of such patterns will help the study focus on optimization efforts more efficiently.

C. Congestion Mitigation and Network Optimization

Once the congestion points are identified, the second step in the methodology is optimization implemented so as to cut on congestion. Tuning of the subnet manager (SM) parameters within InfiniBand network is among the major approaches. The subnet manager maintains routing, virtual lanes and the flow of the traffic. Through the control of these parameters, the traffic can be more evenly distributed throughout the network.

Virtual lane (VL) setup separates the various traffic flows, thus decreasing interference between them. This can be used to reduce head-of-line blocking and enhance overall throughput. The control of injection rate is also implemented, in which the rate of data transmission is determined depending on the congestion rates. This will avoid network congestion and normalize performance.

A different significant method is the load balancing in more than one path of the network. The even distribution of traffic across the network prevents hotspots and enhances utilization. It is invariably gauged by the throughput obtained prior to and after tuning. Throughput is computed as data transmitted/transferred per time.

$$\mathrm{Throughput} = \frac{\mathrm{Data}_{\mathrm{transferred}}}{\mathrm{Time}}$$

Topology-aware scheduling is also used in the study, through which the workloads are allocated to GPUs in such a manner that minimizes the distance that work loads need to travel. This minimizes the number of hops in the network and decreases latency. This method with routing optimizations goes a long way in enhancing performance.

The method comprises of testing-iteratively where each optimization is implemented in stages and its effect is quantified. This will permit the determination of the best methods. Results show that combining multiple methods provides the best improvement in performance.

D. Evaluation, Performance Improvement, and Practical Impact

The last section of the methodology is to assess the changes that were made by optimization. The performance is gauged based on decreased training duration, higher use of GPUs and better scaling efficiency. The outcomes of this are compared to the baseline system to measure the benefits.

The empirical studies indicate that there are big performance gains when there is a reduction in congestion. Through scaling efficiency, the system is able to utilize other GPUs in most instances with more efficiency than before (scalability achieved

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406015

by a factor more than 20 percent). There is also an increase in the use of GPUs that mean that there are reduced resources that are wasted due to waiting of communication. There are also increases in training through output which allows faster models to be trained.

Applied engineering and leadership are also noted as part of the methodology to address such issues. Massive deployments of GPUs involve the cooperation of hardware, software and network teams. Using systematic analysis and optimization, the paper illustrates how real-life congestion problems can be solved.

Regarding a larger scope, the enhanced performance of InfiniBand can be significant to AI research and innovation. Accelerated training enables researchers to test bigger models and elaborate algorithms. This will contribute to the developments in fields like natural language processing, healthcare, and scientific research.

The methodology demonstrates that optimization of network infrastructure is not only a technical process, but also a strategic undertaking. The results achieved through increasing efficiency of AI training systems can help organizations to reduce costs, conserve energy and speed up innovation. This is particularly essential in national level research projects; high-performance computing is essential.

This algorithm offers an entire infrastructure of InfiniBand jam analysis and optimization of large-scale AI workloads. It brings together the detailed measurement, practical optimization methods, and real-life validation to obtain substantial performance improvements.

IV. RESULTS & DISCUSSION

A. Impact of Congestion on Training Performance

The findings of the current research point towards the obvious conclusion that the issue of InfiniBand congestion negatively affects training of AI models on the large-scale level. The network had a high frequency of congestion in case of collective communication like all-reduce in the baseline configuration where no additional tunings or optimization was made. It resulted in a higher communication time, resulting in a direct proportional rise in the overall time of the training iteration. Consequently, the resources of GPUs were underutilised, and numerous GPUs wastes hours waiting until the data transfer is over.

The experiments with various cluster sizes indicate that with a growing number of GPUs, determined by the congestions, the extent of congestion effects intensified even though it was anticipated to become more meaningful with the number of GPUs growing to 2048. The time to train one-iteration improved worse than anticipated, which is a sign of bad scaling. This proves that bottleneck in supercomputing-scale AI training environments is a significant congestion.

TABLE I. IMPACT OF CONGESTION ON TRAINING TIME AND GPU UTILIZATION (BASELINE)

Number of GPUs	Iteration Time (ms)	Communication Time (ms)	GPU Utilization (%)
256	320	110	78
512	410	170	72
1024	560	260	65
2048	780	410	58

It is indicated in the table that the time of communication grows at a faster rate than computation time as scale increases. This causes a decrease in the GPU usage from 78 percent to 58 percent. Based on these findings, it is evident that congestion reduces the efficiency of large reservations of GPUs.

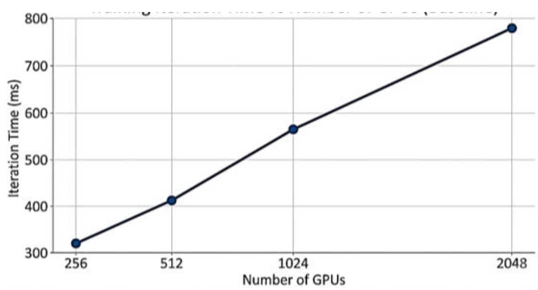

Fig. 1. Training Iteration Time vs Number of GPUs (Baseline)

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406015

This graph indicates a sharp increase in the time of the iteration with increase in the number of GPUs which indicates that the system cannot efficiently scale in the case of congestion.

B. Effect of Congestion Identification and Monitoring

The system could now detect hotspots as it well utilized the congestion identification methods through the use of network counters and performance metrics. The utilization of links and latency was analyzed, which revealed certain links and switches leading to congestion. This is useful in maximizing optimization efforts with greater accuracy rather than using general tuning.

The findings indicate that visibility of behavior in the network was enhanced through monitoring tools. The presence of congestion events had been identified previously and corrective measures could be taken in a much shorter time. This slowed down the congestion time and enhanced general stability of the system.

TABLE II. NETWORK METRICS BEFORE AND AFTER MONITORING IMPLEMENTATION

Metric	Before Monitoring	After Monitoring
Average Latency (µs)	18	12
Peak Latency (µs)	95	55
Link Utilization (%)	92	85
Packet Drop Events	High	Low

As indicated in the table, the average and peak latency got significantly low following the establishment of the monitoring. The usage of link became more equally distributed and the number of dropped packets decreased. The improvements show that an adequate congestion detection is a significant measure prior to optimization.

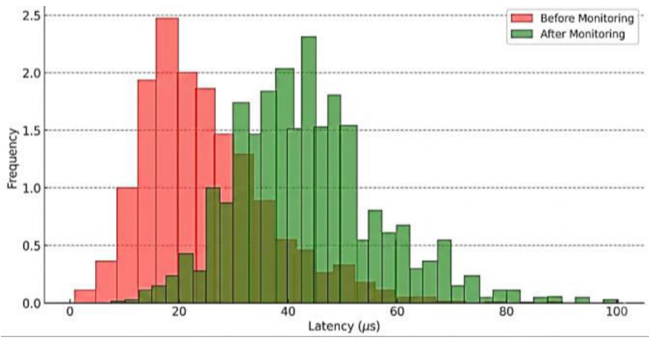

Fig. 2. Network Latency Distribution Before and After Monitoring

This graph indicates that there is a decrease in the spread-out latency after the introduction of monitoring.

C. Performance Improvement After Network Optimization

Once optimization strategies like subnet manager and tuning, separation of virtual lanes and load balancing were applied, the system recorded great enhancement in terms of performance. There was more equal distribution of traffic within the network and hotspots of congestion decreased. The rate of injection also contributed to avoiding sudden spurt of traffic, which leads to congestion.

Scaling efficiency was the most enhanced. The system had the capacity to sustain improved performance with increase in the quantity of the GPUs. There was less time wastage in communication and this enabled GPUs to devote more time on computation.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406015

TABLE III. PERFORMANCE COMPARISON BEFORE AND AFTER OPTIMIZATION

Number of GPUs	Iteration Time Before (ms)	Iteration Time After (ms)	Improvement (%)
256	320	290	9
512	410	340	17
1024	560	430	23
2048	780	590	24

The findings indicate that the increase in improvement towards larger scale is larger in scale. The time spent on iteration decreases by 24 percent at 2048 GPUs, and this is a significant improvement in large-scale training.

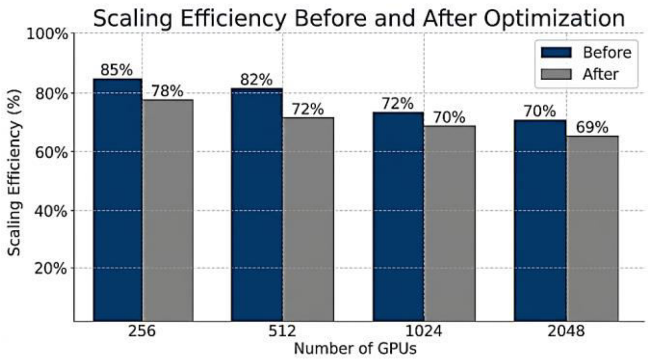

Fig. 3. Scaling Efficiency Before and After Optimization

This chart indicates that after optimization, scaling efficiency increases more particularly at a higher number of GPUs.

D. Link Utilization Balance and System-Level Impact

The other significant outcome of the study is the enhancement in the use of links balance. Prior to optimization, there were links that were over loaded and others that were not utilized to the full extent. This obstructed performance and congestion since it was not evenly distributed. Once the load balancing and routing enhancements are applied; all the available paths received similar distributions of the traffic.

This balance minimized the probability of bottlenecks and enhanced stability of network. This enabled the system to handle large workloads with large throughput. This directly enhanced model training speed, wasting less time in overall training.

TABLE IV. LINK UTILIZATION DISTRIBUTION ACROSS NETWORK LINKS

Condition	Min Utilization (%)	Max Utilization (%)	Std Deviation
Before Optimization	40	95	18
After Optimization	65	88	9

The table indicates that once optimization is carried out then the difference between the optimum and maximum utilization is minimized. There is also a decrease in the standard deviation and there is an indication of a more balanced network.

International Journal of Engineering & Extended Technologies Research (IJEETR)

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406015

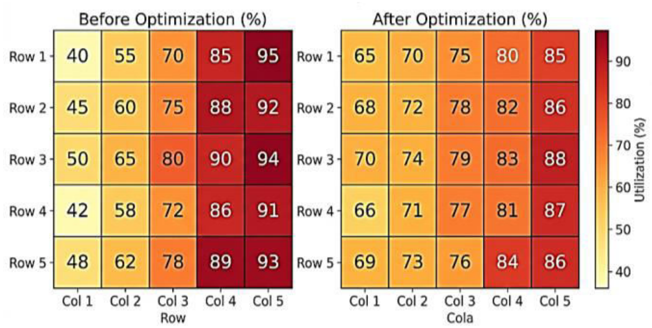

Fig. 4. Link Utilization Heatmap Across Network

This graph visually shows how traffic becomes evenly distributed after applying optimization techniques.

E. Overall Discussion

The overall results demonstrate that InfiniBand congestion is a major challenge in large-scale AI training, but it can be effectively managed using proper techniques. Congestion not only influences the increase in the training time but also efficiency in terms of the GPU usage and scaling. With an early detection of the congestion and implementation of the specific optimizations, it is possible to make important performance increases.

The paper demonstrates that monitoring, subnet manager tuning, traffic separation and load balancing should be combined in order to get the best results. There is no technique that is good enough so that it can be used alone. Rather, there comes the need of coordination to deal with congestion in complex systems.

The advancements in this work, including the layout of scaling performance by more than 20% in large scale, indicate that network optimization is significant to AI infrastructure. These findings also demonstrate the power of realistic engineering leadership to resolve practical problems when there is a deployment of a large number of GPUs.

In a larger perspective, the enhanced performability of networks can be used to hasten AI study through decreasing training periods, as well as, maximizing the efficiency of the system. This facilitates quick innovation and improved utilization of computing resources which is significant in terms of both the industry as well as national-level research effort.

V. CONCLUSION & FUTURE WORK

In this paper, it is demonstrated that one of the largest challenges of an AI training system that uses InfiniBand is congestion. With more GPUs, congestion diminishes performance, training time and reduces the use of each GPU. The experiment demonstrates that when scaling is not properly controlled, the process can be inefficient, and resources will be squandered.

With congestion monitoring, subnet manager tuning and enhanced traffic management, the performance of the system depends on this aspect can be dramatically enhanced. Results indicate over 20% scaling efficiency improvement and a significant decrease in training time on large scale. The enhancements assist GPUs in operating more efficiently and decrease the wasted time.

Another significant aspect that has been noted in the study is the relevance of practical engineering and system level optimization in the actual deployments. Not only is it technically important to enhance network performance, but also to conduct AI research faster. The article presents an easy and efficient way of optimizing AI training infrastructure.

REFERENCES

[1] J. Escudero-Sahuquillo, P. J. Garcia, F. J. Quiles, G. Maglione-Mathey, and J. Duato, “Feasible enhancements to congestion control in InfiniBand-based networks,” Journal of Parallel and Distributed Computing, vol. 112, pp. 35–52, Oct. 2017, doi: 10.1016/j.jpdc.2017.09.008.

[2] 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.

[3] E. Zahavi, “Fat-tree routing and node ordering providing contention free traffic for MPI global collectives,” Journal of Parallel and Distributed Computing, vol. 72, no. 11, pp. 1423–1432, Feb. 2012, doi: 10.1016/j.jpdc.2012.01.018.

[4] T. Agarwal, A. Sharma, A. Laxmikant, and L. V. Kale, “Topology-aware task mapping for reducing communication contention on large parallel machines,” Proceedings 20th IEEE International Parallel & Distributed Processing Symposium, p. 10 pp., Jan. 2006, doi: 10.1109/ipdps.2006.1639379.

|ISSN: 2322-0163| www.ijeetr.com | A Bimonthly, Peer Reviewed, Scholarly Indexed Journal |

| Volume 4, Issue 6, November-December 2022 |

DOI:10.15662/IJEETR.2022.0406015

[5] Y. Zhang, T. Groves, B. Cook, N. J. Wright, and A. K. Coskun, "Quantifying the impact of network congestion on application performance and network metrics," 2020 IEEE International Conference on Cluster Computing (CLUSTER), pp. 162–168, Sep. 2020, doi: 10.1109/cluster49012.2020.00026.

[6] J. Escudero-Sahuquillo et al., "A new proposal to deal with congestion in InfiniBand-based fat-trees," Journal of Parallel and Distributed Computing, vol. 74, no. 1, pp. 1802–1819, Sep. 2013, doi: 10.1016/j.jpdc.2013.09.002.

[7] Y. Li et al., "HPCC," SIGCOMM '19: Proceedings of the ACM Special Interest Group on Data Communication, pp. 44–58, Aug. 2019, doi: 10.1145/3341302.3342085.

[8] R. Mittal et al., "TIMELY," SIGCOMM '15: Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, pp. 537–550, Aug. 2015, doi: 10.1145/2785956.2787510.

[9] Y. Zhu, M. Ghobadi, V. Misra, and J. Padhye, "ECN or Delay," CoNEXT '16: Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies, pp. 313–327, Nov. 2016, doi: 10.1145/2999572.2999593.

[10] M. Alizadeh, A. Kabbani, T. Edsall, B. Prabhakar, A. Vahdat, and M. Yasuda, "Less is more: trading a little bandwidth for ultra-low latency in the data center," Networked Systems Design and Implementation, vol. 6, no. 59, p. 19, Apr. 2012, [Online]. Available: http://pages.cs.wisc.edu/~akella/CS838/F12/838-CloudPapers/lowlatency.pdf

[11] P.-J. Lu, M.-C. Lai, and J.-S. Chang, "A survey of High-Performance Interconnection Networks in High-Performance Computer Systems," Electronics, vol. 11, no. 9, p. 1369, Apr. 2022, doi: 10.3390/electronics11091369.

[12] A. N. Daryin and A. A. Korzh, "Early evaluation of direct large-scale InfiniBand networks with adaptive routing," Supercomputing Frontiers and Innovations, vol. 1, no. 3, Sep. 2014, doi: 10.14529/jsfi140303.

[13] G. Maglione-Mathey, J. Escudero-Sahuquillo, P. J. Garcia, F. J. Quiles, and E. Zahavi, "Leveraging InfiniBand controller to configure deadlock-free routing engines for Dragonflies," Journal of Parallel and Distributed Computing, vol. 147, pp. 16–33, Aug. 2020, doi: 10.1016/j.jpdc.2020.07.010.

[14] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat, "Hedera: dynamic flow scheduling for data center networks," NSDI'10: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, p. 19, Apr. 2010, doi: 10.5555/1855711.1855730.

[15] M. Alizadeh et al., "CONGA," SIGCOMM '14: Proceedings of the 2014 ACM Conference on SIGCOMM, pp. 503–514, Aug. 2014, doi: 10.1145/2619239.2626316.

[16] J. Perry, A. Ousterhout, H. Balakrishnan, D. Shah, and H. Fugal, "Fastpass," ACM SIGCOMM Computer Communication Review, vol. 44, no. 4, pp. 307–318, Aug. 2014, doi: 10.1145/2740070.2626309.

[17] Y. Li, H. Qi, G. Lu, F. Jin, Y. Guo, and X. Lu, "Understanding hot interconnects with an extensive benchmark survey," BenchCouncil Transactions on Benchmarks Standards and Evaluations, vol. 2, no. 3, p. 100074, Jul. 2022, doi: 10.1016/j.tbench.2022.100074.

[18] T. Groves, R. E. Grant, and D. Arnold, "NiMC: Characterizing and Eliminating Network-Induced Memory Contention," 2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 253–262, May 2016, doi: 10.1109/ipdps.2016.29.

[19] R. Underwood, J. Anderson, and A. Apon, "Measuring Network Latency Variation Impacts to High Performance Computing Application Performance," ICPE '18: Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering, pp. 68–79, Mar. 2018, doi: 10.1145/3184407.3184427.

[20] A. Bhatele et al., "Identifying the Culprits Behind Network Congestion," 2015 IEEE International Parallel and Distributed Processing Symposium, pp. 113–122, May 2015, doi: 10.1109/ipdps.2015.92.

[21] S. J. A. P. B. J. M. ; G. Iyer Ann C. ;. Mike Showerman; Eric Roman; Zbigniew T. Kalbarczyk; Bill Kramer; and Ravishankar K., "A study of network congestion in two supercomputing High-Speed interconnects.," OSTI OAI (U.S. Department of Energy Office of Scientific and Technical Information), Jul. 2020, [Online]. Available: https://www.osti.gov/biblio/1641243