RTGWG Working Group
Internet-Draft
Intended status: Informational
Expires: 1 September 2025

F. Li
Y. Li
R. Meng
R. Huang
Huawei

28 February 2025

The Challenges and Requirements for Routing in Computing Cluster network
draft-li-rtgwg-computing-network-routing-01

Abstract

This document explores the characteristics of computing cluster network and analyzes the challenges of employing the traditional distributed or centralized routing mechanisms in it. In order to achieve the enhanced scalability, improved resilience and optimized performance simultaneously, hybrid routing is briefly examined as a potential way to combine the advantages of distributed and centralized mechanisms. The document further shows the possible future work.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 1 September 2025.

Copyright Notice

Table of Contents

1. Introduction

As artificial intelligence (AI) and deep learning have gained popularity, large-scale AI models such as GPT-3, BERT, and T5 have emerged as focal points in the industry. Due to their extensive data requirements and complex computing performance needs, the support of computing cluster network become necessary. Computing cluster network with high performance serves as the foundation for many advanced scientific and engineering applications. To accommodate the training and inference of large AI models, these networks with high-capacity, low-latency, scalability and reliability are critical, as well as efficient and stable computing power.

This document explores the characteristics of computing cluster network and, based on that, it analyzes the challenges of employing the distributed or centralized routing mechanisms in terms of the scalability, resilience and performance. Hybrid routing combines the advantages of scalability in distributed routing and light configuration and calculation on individual node in centralized routing and is briefly introduced as a potential candidate in computing cluster network.

2. The Overview of Computing Cluster Network

Networks for AI training and inference handle very different traffic compared to traditional data centers. AI training, especially in distributed systems, requires moving huge amounts of data between computing nodes, like model parameters and gradients. This creates a "many-to-many" communication pattern, where all machines talk to each other frequently. In contrast, traditional data centers mostly deal with "client-server" traffic, where users request data from servers, resulting in less intense flows.

Taking LLM (large language model) as an example, the training of large-scale models with billions or even trillions of parameters presents a significant challenge. For instance, within a single computing iteration, the communication volume required for gradient synchronization alone can reach the terabytes (TB) scale. Moreover, the introduction of various parallelization modes and acceleration frameworks adds to the communication demands. The bandwidth of traditional networks insufficient to support the efficient computing power of accelerator (xPU) clusters. To fully leverage the powerful computing resources, it is required to establish a high performance network infrastructure that utilizes the high bandwidth to boost the overall computing power of the clusters.

There are series of innovations explored in the recent researches and developments of computing cluster networks. Traditional data center networks are primarily designed to serve APP/Web applications, where internet access traffic is often far greater than inter-server traffic. While in computing cluster networks, all accelerators must be interconnected through the network. This interconnection often necessitates ultra-high bandwidth and ultra-low latency, leading servers to be interconnected with multiple network interfaces at high speeds and inter-server traffic dominates. This results in a network topology to be optimized for density and distance. Hierarchical topologies like Fat-Tree, Dragonfly [I-D.agt-rtgwg-dragonfly-routing] 3D Torus and proprietary design are used to efficiently connect large clusters, reducing "hops" between distant nodes. Traditional data centers rely on simpler three-tier designs (core-aggregation-access) optimized for cost and ease of maintenance, not performance.

The service traffic in such environments is often periodic and predictable. For example, for a specific AI training model application, the communication traffic for each iteration of the computing is deterministic and would quickly consume the full network bandwidth. Enhancing training efficiency and shortening the training time are critical in AI computing. The network should minimize the idle time of xPU as much as possible. From the point of view of routing, there are primarily two aspects to consider. One is how to

effectively construct ECMP (Equal-Cost Multipath) and Non-Shortest Paths to greatly improve the network bandwidth consumption. The other is how to achieve fast convergence after failures, which is particularly important in AI computing, where latency cause by inefficient reroute can significantly impact training time.

In AI networks, a single node failure can disrupt a training job that might have been running for days or weeks. To avoid this, AI networks are designed with strong fault tolerance, allowing them to quickly recover from failures. In synchronous training (e.g., data-parallel approaches), all nodes must wait for the slowest device to finish computation before updating model parameters. As clusters scale, straggler nodes (due to hardware variance or network delays or failure) disproportionately slow down the entire system. Traditional data centers, while also reliable, are less sensitive to short interruptions, as most tasks can be restarted without significant consequences.

Therefore, the following key aspects are particularly crucial in the design and implementation of a computing cluster network with high performance:

1. Scalability: Computing cluster network must be scalable to accommodate the growing size and complexity of AI models. This includes the ability to handle increased data traffic and the expansion of the network infrastructure without sacrificing performance.

2. Reliability: Given the critical nature of AI computations, the network must be highly reliable, with mechanisms in place to ensure continuous operation and rapid recovery from failures.

3. Optimized Routing: The network should employ advanced routing algorithms that can efficiently route traffic to minimize latency and maximize throughput. This includes the use of BGP and other routing protocols that can adapt to changing network conditions and optimize path selection.

3. State of the Art on Routing in Computing Cluster Network

The routing mechanisms in computing clusters mainly fall into two realms, i.e. distributed routing solutions and centralized routing solutions, each of which is extensively studied in the academies and widely deployed in the industry.

For the distributed routing in AI networks, BGP is a popular protocol because it scales efficiently for large clusters by avoiding full-topology synchronization, reducing overhead. Its policy-driven

control optimizes bandwidth allocation for latency-sensitive tasks like distributed training, while link-state IGPs struggle with policy flexibility and high computing and storage overheads when maintaining the global topology (each node needs to store the LSDB of the entire network). Both [BGPinDC] and [HPN] use the BGP protocols for the DCN or large language model (LLM) training clusters. [BGPinDC] chooses BGP as the fundamental mechanism in the favor of its good scalability, flexibility of policy control. BGP-based data center routing design are made for better usage and performance, such as ASN allocating mechanism, using BGP federations to enable ASN reuse in different DCN/clusters, configuration templates for eliminating misconfigurations, route summarization for saving hardware resources, etc. [HPN] deploys BGP for their 2-tier and dual-plane architecture. Besides the basic layer3 forwarding, BGP is leveraged especially for the failure handling in their non-stacked dual-ToR architecture. [HPN] mentions that BGP is not used on the host, for the reason that all hosts taking part in the BGP updating procedure would greatly slow down the BGP convergence speed.

Centralized routing normally uses Software-Defined Networking (SDN) concept to construct the control and management systems, in which the centralized controller collects link status information and sends control instructions to each network element under control. Orion [ORION] and Borg [BORG] are two typical systems. Hierarchical multi-layers controllers are deployed for solving the scalability issue. [SIDR] short for Scalable Intent-Driven Routing, also provides a single control plane architecture, in which three main components, SIDR supervisor, SIDR fabric controller (SFC) and SIDR daemon are provided. The SIDR supervisor and SFC perform the hierarchically control of the network, while SIDR daemon running on network elements works as proxies for messages exchanging between control plane and network elements.

4. Challenges for Distributed Routing in Computing Cluster Network

Though BGP is a commonly used routing protocol in traditional data centers [rfc7938], it faces the challenges of the routing convergence and configurations when being used in the computing cluster network.

4.1. Slow Routing Convergence

In the computing cluster network, routing convergence is crucial for the AI training jobs, as well as HPC workloads. During the BGP convergence period, there may be blackholes in the network, then packets would be get dropped. Losing of exchanging data, especially the gradient data, or the temporary calculating results data between layers, may cause the tasks deployed on different GPUs an illusion of malfunction of the corresponding GPUs. The AI job may launch the

backup GPU and resort to the last checkpoints, which degrades the AI jobs performance dramatically. It is reported that monthly link failure ratio is very high in the operating cluster, general larger than 0.05% [HPN], then in the case of large-scale computing clusters, BGP converges frequently. The shorter of the BGP convergence period, the less impact on the AI jobs' performance.

There are many factors contributing to the BGP convergence, such as the minimum time interval between BGP updates or advertisements, the complexity for path selection, etc. Accordingly, there are many proposals to ensure the BGP converge in a shorter time period respectively. Some solutions set the MRAI (MinRouteAdvertisementInterval) timer to be zero, as well as some methods reducing the path selecting complexity in a way by limiting the searching space. Allocating the Autonomous System Number (ASN) in a suitable way, can greatly alleviate the path selection hunting procedures, thus making the BGP converge more quickly.

For the spine-leaf topology in Figure 1, there are six network elements, i.e. switches. Two spine switches, S1 and S2, connect to four leaf switches, L1, L2, L3 and L4. If each switch is assigned a unique AS number, then S1 will receive multiple BGP update messages for the IP prefix attached to L1, each containing different AS_PATH attributes information as following. S1 would save all the routes generated from the BGP updates in the RIB and generate the final optimal route in the FIB. If there are lots of spine and leaf switches in the network, S1 will be overburdened by the route computing and cost of the routing entry storage, eventually resulting in bad BGP convergence.

AS_PATH info 1: prefix-L1-S1
AS_PATH info 2: prefix-L1-S2-L2-S1
AS_PATH info 3: prefix-L1-S2-L3-S1
AS_PATH info 4: prefix-L1-S2-L4-S1

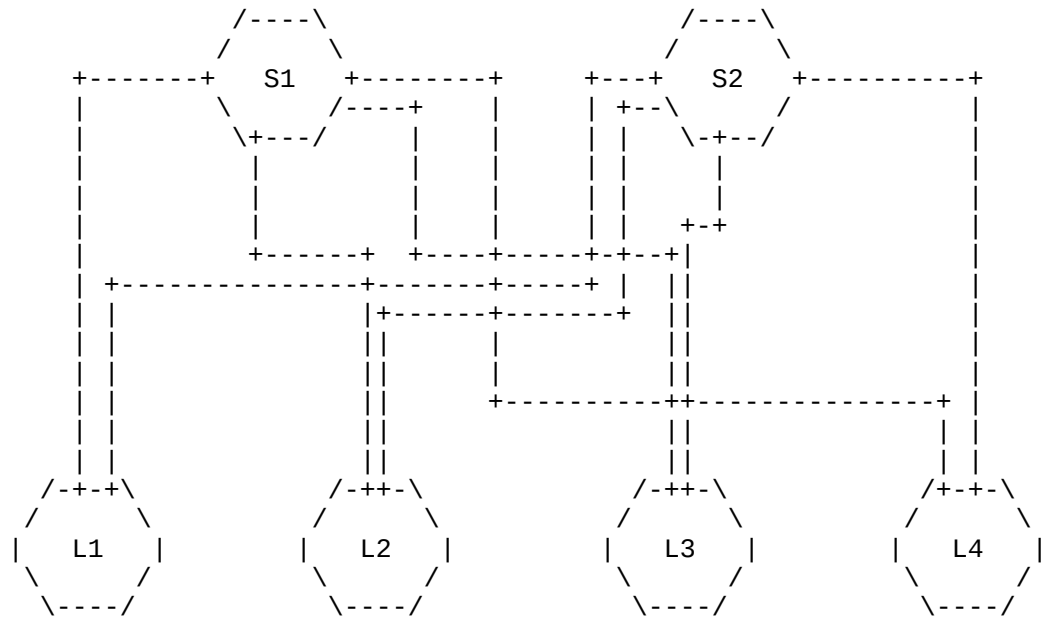

Figure 1: Spine Leaf Topology

Some solution reduces the route computing complexity in S1 by assigning the same AS number to all the switches in the spine layer, then S1 would only receive one BGP updates, as L2, L3 and L4 will not forward BGP updates received from and to the same AS. This greatly releases the burden of S1, but loses the redundant paths, which may be of high value when these paths are used as backup paths, or non-ECMP paths for fully utilizing the bandwidth provided by the connecting network. Many solution uses the non-ECMP paths, i.e. both the shortest path and no-shortest path, then it is normal way to steer the traffic along a non-shortest (non-ECMP) path to acquire flatter pipe with more bandwidth between two nodes regardless of the more forwarding hops. This may greatly reduce the BGP messages needed for BGP to converge, and enable a better BGP convergence performance, but at the cost of sacrificing path diversity.

Another in-efficiency use case of BGP updates is constructing the multiple paths for ECMP scenarios. In figure 1, L2 needs to get two BGP updates advertising reachability to prefix attached to L1, respectively from S1 and S2, in order to formulate the two ECMP paths, L2->S1->L1->prefix and L2->S2->L1->prefix. If the computing cluster is a large scale one, lots of BGP updates would be sent out and process, and this would greatly impact BGP convergence as well.

4.2. Complex BGP Configurations

Running BGP in computing cluster network, lots of configurations need to be carried out. Configurating the network is a challenging and tedious work with the increasing of the network scale, complexity and its dynamism attributes. These configurations mainly fall into two categories. The first type is BGP routing protocol configurations, which includes configuring the parameters of the AS number, prefixes, router ID, BGP peer addressing information, etc. The second type is BGP policy configurations, which includes the route importing and exporting filters, traffic steering policies for drain/undrain operations, policies for traffic load balancing, redundancy and path preferences, etc. Most of the configurations in the first type may remain static during the lifetime of computer cluster network. On the other hand, it is expected that some dynamic changes are common, such as the drain policy configuration differs time-to-time because of upgrading or failures of different network elements. It is error-prone and time-consuming to do the BGP configurations.

Computing cluster network operators are trying different ways to find an easier BGP configurations methods. Some are defining configuration templates for different network elements, and it is easy to do configuration just by filling different parameters in the corresponding templates. Some solutions are defining a high-level language, which can express the desired behavior in an Intent way. Then a compiler or generator will convert the Intent expressions into network element configurations automatically. All these efforts reduce the configuration complexity to some extent, but configuring the network is still needed. Once configuration is performed, drawbacks of possible misconfigurations and non-realtime effectiveness of configurations still exist.

5. Challenges for Centralized Routing in Computing Cluster Network

An SDN-like (Software Defined Networking) centralized routing mechanism for computing cluster network is another common approach in market. The centralized controller connects to all the network elements under control through either an in-band network, i.e. the same network as the forwarding of data packets and control packets, or an out-of-band network, i.e. the network dedicated only for control or management packets. The controller and network elements exchange the control information over the connections. The network element can report its neighboring information, link up/down status, etc. to the controller. The controller can install the forwarding table entries, policies, or configurations to the network elements.

Generally, the centralized controller can formulate the global network topology and do the end to end path calculation based on the communication demands. Then, the forwarding table entries of each network elements along the path will be installed accordingly by the controller. The computing cluster network is enabled with the forwarding capability in this centralized way of control.

The centralized control for the computing cluster network has glorious advantages. First and foremost, the controller greatly alleviates the burden of the control capability of the network elements. No routing protocol or path calculating algorithms is running in the network elements. Secondly, as no protocols is deployed in the network elements, no protocol specific configurations are needed any more. Configuring the network is a daunting work. Thirdly, as the controller has the whole picture of the computing cluster network, it is easy to work out the optimal forwarding path for all the flows.

The centralized approach has the scaling problem. A single controller is unlikely to manage hundreds of thousands of elements in the computing clusters. Even though hierarchical controller architecture is proposed in some solutions, coordination among the controllers is non-trivial. The time consumption of the path calculation in the single controller is also crucial for the network control. The running algorithm should be efficient enough in face of large scalability. For the cluster network failure or recovery events, it goes through the handling procedures step by step. Network elements reports the event to the controller, then controller works out the solutions and install the instructions to the network. This long processing path greatly induces the lag between the event and its handling, which is bad for the convergence time. Extra attention should be paid to the stability of the connection between the controller and network elements. Losing the connection would let the cluster network out of control, which would never happen in the distributed routing solution world.

6. Hybrid Routing in Computing Cluster Network

Computing cluster networks make a significant change from traditional data center networks. There is a trend to combine the advantages of distributed and centralized routing to achieve the enhanced scalability, improved resilience and optimized performance.

[Primus] is one of such hybrid routing solutions. It uses centralized controllers to collect/disseminate the network’ s link-states (LS), and offload the actual routing calculation onto each switch. With the observation that the routing changes can be classified into a few fixed patterns which have regular or modular

topologies, it simplify each switch’ s routing calculation into a table-lookup manner by comparing LS changes with pre-installed base topology and updating routing paths according to predefined rules. So it enjoys the faster per-switch routing calculation time and the efficient controller fault-tolerance.

There are different approaches for hybrid routing from research. One of the key strategies is how to properly divide the routing functions between the centralized controller and individual network nodes. The considerations for a good hybrid routing framework in computing cluster network include the followings.

1. Fully use of the prior knowledge like topology and its modularity, available paths and traffic pattern. Such information is helpful in regular address allocation aligned with cluster modularity, routing path pre-computation and predefined rules installation.

2. Quickly detect network failures and reroute traffic to alternative paths. This minimizes the impact of failures on the overall system performance and ensures that the cluster remains operational with minimal disruption.

3. Minimize per-node routing calculation especially when fault occurs for better scalability. Installation of the predefined rule to help route convergence can be useful. Deploy lightweight decision modules on local nodes to handle burst traffic or link failure recovery.

4. Centralized controller is helpful to provide the global view and traffic planning. However single point of route computation and failure handling should be avoided. The central controller can periodically collect state feedback from distributed nodes (e.g., link quality, load changes), reoptimizes rules, and incrementally updates them, enabling a "centralized planning + distributed execution" closed-loop system. Consistency between centralized policies and distributed local states should be ensured.

Hybrid routing is a possible candidate to improve the routing system in computing cluster networks for better scalability and availability. The authors expect more detailed elaboration of hybrid routing in future IETF work.

7. Security Considerations

TBD.

8. IANA Considerations

This document does not request any IANA allocations.

9. References

9.1. Normative References

[rfc2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[rfc8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

9.2. Informative References

[BGPinDC] Abhashkumar, A., "Running BGP in Data Centers at Scale", April 2021, <https://www.usenix.org/conference/nsdi21/presentation/abhashkumar>.

[BORG] Verma, A., "Largescale cluster management at Google with Borg", April 2015, <https://static.googleusercontent.com/media/research.google.com/zh-CN//pubs/archive/43438.pdf>.

[HPN] Qian, K., "Alibaba HPN, A Data Center Network for Large Language Model Training", August 2024, <https://regmedia.co.uk/2024/06/27/supplied_alibaba_hpn_paper_2.pdf>.

[I-D.agt-rtgwg-dragonfly-routing] Afanasiev, D., Roman, and J. Tantsura, "Routing in Dragonfly+ Topologies", Work in Progress, Internet-Draft, draft-agt-rtgwg-dragonfly-routing-01, 4 March 2024, <https://datatracker.ietf.org/doc/html/draft-agt-rtgwg-dragonfly-routing-01>.

[ORION] Ferguson, A., "Orion, Google's Software-Defined Networking Control Plane", April 2021, <https://www.usenix.org/conference/nsdi21/presentation/ferguson>.

[Primus] Lin, F., "Fast, Scalable and Robust Centralized Routing for Data Center Networks", December 2023, <https://dl.acm.org/doi/pdf/10.1109/TNET.2023.3259541>.

[rfc7938] Lapukhov, P., Premji, A., and J. Mitchell, Ed., "Use of BGP for Routing in Large-Scale Data Centers", RFC 7938, DOI 10.17487/RFC7938, August 2016, <https://www.rfc-editor.org/rfc/rfc7938>.

[SIDR] Callaghan, S., "AWS journey towards intent driven network infrastructure", December 2023, <https://d1.awsstatic.com/events/Summits/reinvent2023/NET401-R_AWS-journey-toward-intent-driven-network-infrastructure-REPEAT.pdf>.

Authors' Addresses

Fengkai Li
Huawei
Email: lifengkai@huawei.com

Yizhou Li
Huawei
Email: liyizhou@huawei.com

Rui Meng
Huawei
Email: mengrui@huawei.com

Rachel Huang
Huawei
Email: rachel.huang@huawei.com