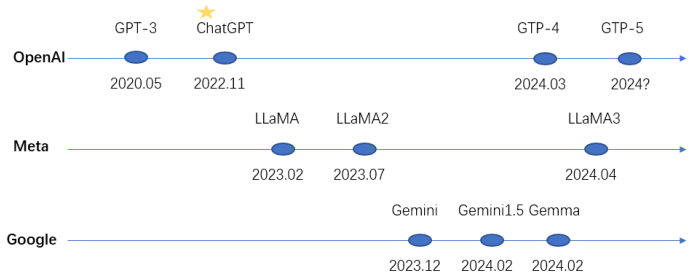
Contributed Text to AICN report

In AICN report draft v0.1, "scale" is listed as one topic in the chapter"Requirements and Challenges of AI computing Networks". This document describes scale issue and proposes text to AICN report.

In the last 4 years, industry giants and startups have accelerated AI large models development, such as GPT from OpenAI, LLaMA from Meta and Gemini/Gemma from Google.

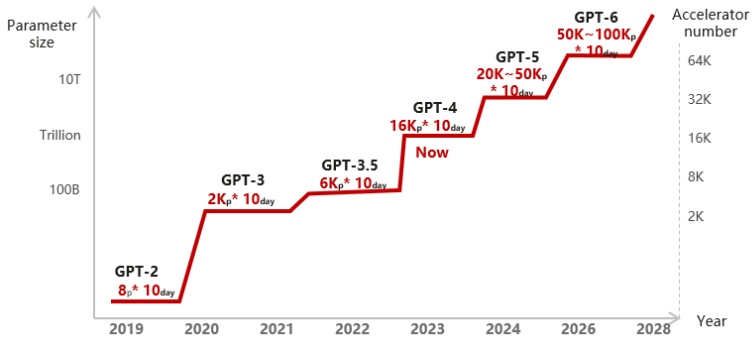


The evolvement of models follows the LLM scaling law, increasing the size of model to get better and better performance. Take GPT series as example. GPT3 is 175B parameters with 300B tokens. When it comes to GPT4 2 years later, parameter size grows to 1.8T with 13T tokens. GPT5 is not released yet, but It is stated to be much more powerful than GPT4 by OpenAI. The parameter size is estimated to be close to 10T with 30T tokens.

The larger the model is, the more computing power is required to train the model. Roughly, F=6PD. F is computing power, P is parameter size, and D is token number. [1] So, compared with the model 4 years ago, the computing power needs increases 10000 times.

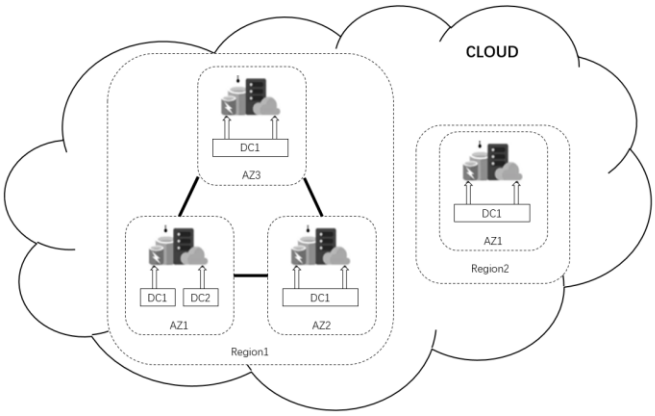
Although the accelerators used for AI training are also constantly evolving, like Nvidia's latest GPU B200 which reaches 2.5PFLOPS[5], almost 8 times that of its previous product A100, the speed of evolvement is lag behind the growth of required computing power. So it is seen that the scale of AI cluster increases from thousands of accelerators to tens of thousands accelerators, even to hundreds of thousands accelerators[2].

Below figure depicts the development of model size, cluster scale, and its relationship. The x-axis is the year of development. The y-axis on the left is parameter size of AI models. The y-axis on the right is the number of accelerators in AI cluster. To train GPT-3 within 10 days, it needs about 2000 accelerators 4 years ago. Now, the mainstream large-scale AI cluster is built with tens of thousands accelerators. In future, it is expected to use hundreds of thousands accelerators to train larger models, such as GPT-6.



With the scale of AI cluster growing bigger and bigger, power consumption and network cost become challenges.

The power requirements for large-scale AI clusters are substantial due to the significant computational resources(mainly accelerators) needed. A report by Schneider Electric points out the introduction of new GPU generations has led to a significant increase in power consumption, despite yielding higher productivity gains. An AI cluster with 22,000 H100 GPUs demands 31 megawatts of power, which is about 2 times to A100 GPUs. Considering a typical public cloud deployment, as shown in below figure. It consists regions, AZs(available zone) and DCs(data center). An AZ is a logical data center, backed by one or more physical data centers. Each AZ has separate power, networking and connectivity. Deploying large scale AI cluster in an AZ is restrained by the power supply of the district. It is difficult to find a place available of huge power supply when AI cluster becomes bigger and bigger. Microsoft has already met the problem. Microsoft engineer complains that they cannot put more than 100K H100s in a single state without bring down the power grid. [2]



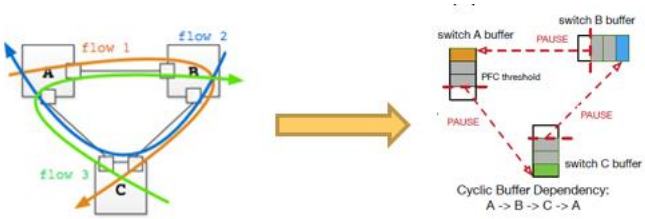
It has to do AI training across different datacenters or AZs. Therefore, long distance DCI transmission is involved. When training is running within a single location, it is relatively easier to control the traffic in AI cluster. The focus is how to get large bandwidth. People build regular topology, use non-blocking network, and run single job in the AI cluster. But when the training job is across different locations, those friendly pre-conditions will disappear. First, cross DCs/AZs AI traffic is mixed with other application traffic. The burst of AI huge traffic can easily conflict with other applications' traffic, causing incast congestion. Second, distance of DCI is much longer. It can be tens of kilometers or hundreds of

kilometers depending on where the datacenters are located. Current congestion control will react slowly because it takes longer time to sense the congestion. And the communication time is increasing which decreases accelerator utilization. Third, it turns to be over-subscribed network. The oversubscription ratio is usually 1:10. The convergent point may become bottleneck.

Another challenge is the network cost of AI cluster. Fat-tree topology, while popular in traditional datacenter network encounters cost issues when scaled up for large AI clusters. Fat-tree is good for diverse traffic pattern. Its structured approach considers the worst case patterns. But that is not efficient for AI traffic which is predictable. As network scale grows, the hierarchical nature of fat-tree requires a significant number of switches and extensive cabling, leading to increased hardware expenses. According to HOTI 2023 keynote speech by Nvidia[4], network cost per node(<5k endpoints) is above 160$ while dragonfly topology is only 80$. The constant value may have changed, but the ratio remains the same. Moreover, the complexity of managing such a vast network adds to operational costs, as adding or removing nodes affects the overall topology and may require significant reconfiguration.

This has led industry to explore optimized topologies in order to have a more cost-effective model for AI cluster. Such works include direct topologies and optical interconnects. Dragonfly/Dragonfly+ and torus are typical direct topologies. They are scalable and flexible, requiring fewer switches than the traditional fat-tree topologies. Optical interconnects provide higher bandwidth, lower latency, and improved energy efficiency compared to traditional electrical interconnects. Some examples are 3D torus and optical spine switches used by google[6], and Nvidia proposed Dragonfly+ that uses OCS (optical path switching) to provide interconnect between groups[4].

But those optimized topologies have their own challenges when deployed for Ethernet/IP traffic. It must support adaptive routing when direct links between 2 endpoints are congested. The paths are not regular, that increases risk of PFC deadlock. PFC deadlock is caused by CBD(cyclic buffer dependency), as shown in the figure. Flow 1, 2 and 3 use the same priority queue. The paths of the 3 flows overlap. Once any one of switches A, B, and C starts PFC due to congestion, PFC deadlock may occur. In fat-tree, it uses equal-cost paths and does not form CBD. Only when a link failure happens, the traffic is re-routed causing CBD. It is easy to identify the re-routing by observing the direction of traffic path. 802.1Qcz[7] provides topology recognition function for fat-tree topology assisting to determine traffic path direction, thereby breaking CBD. However, the topology recognition is not applicable for the optimized topologies. One reason is direct topologies do not have layers as fat-tree. Another reason is optical interconnects may be reconfigured on demand to suit traffic pattern which changes the topology. Furthermore, the traffic re-routing is no longer an indicator of CBD. Because traffic paths are not regular in optimized topologies and adaptive routing makes it more complicated. Developing a method adaptive to different topologies is critical to solve PFC deadlock issue.



[1] https://zhuanlan.zhihu.com/p/688178908
[2] https://mp.weixin.qq.com/s/y6B9vM13byV8KT9A3fRiDA
[3] https://www.powerelectronicsnews.com/schneider-electric-predicts-substantial-energy-consumption-for-ai-workloads-globally/
[4] https://www.youtube.com/watch?v=napEsaJ5hMU
[5] https://www.theregister.com/2024/03/18/nvidia_turns_up_the_ai/
[6] https://arxiv.org/pdf/2304.01433