Paper Review - MAST: Global Scheduling of ML Training across Geo-Distributed Datacenters at Hyperscale

Sai Venkat Malreddy

10 oktober 2024

1 Idea and Contribution

Mast is a global scheduling abstraction for ML workloads, which takes away the control from user to run their workloads in specific region in order to prioritise the whole organization workloads to address the issue of regional mismatch between workload demand and hardware supply. MAST reduced the ratio of demand to supply from 2.63 to 0.98 and balancing the load uniformly over the regions targeting the maximum utilization of he GPU's available. Authors achieved this by implementing the below paradigm in building the MAST: Temporal Decoupling, Scope Decoupling and Exhaustive Search.

2 Positive Comments

- **High Resource Allocation Efficiency**
  MAST achieved a GPU allocation rate of 98%, maximizing resource utilization across Meta's global datacenters. This is crucial as GPUs are expensive and in high demand, so this level of efficiency directly contributes to cost savings and performance.

- **Effective Resource Management Through Soft-Balance Strategy**
  A key positive aspect of the system is its use of the soft-balance strategy for managing scarce GPU resources. Instead of enforcing strict quotas, which could block high-priority jobs, the system allows some flexibility in GPU allocation. This approach penalizes overloaded regions while allowing over subscription, ensuring high-priority workloads get resources by preempting lower-priority ones. As a result, the system maximizes GPU utilization while maintaining a balance across regions, making it efficient and adaptable to real-world demands at scale

- **Scope Decoupling**
  The scope decoupling architecture of the job scheduling system is highly effective due to its scalability, efficiency, and reduced complexity. By separating job queue management, resource allocation, and container orchestration into distinct levels, the system can efficiently handle large-scale workloads across geo-distributed datacenters.erscale machine learning training workloads.

3 Negative Comments

- **Feedback Mechanisms**
  The current architecture may lack robust feedback mechanisms for monitoring and adjusting resource allocations dynamically. Without timely feedback, the system may struggle to adapt to changing workload patterns or unforeseen resource failures

- **Calculation of Priorities**
  While the paper does not explicitly quantify the time taken for the calculation of priorities for each job, it is reasonable to infer that the complexity and volume of calculations involved in determining priority and credit could lead to increased latency. This effect is particularly pronounced in environments with a high number of workloads, where the cumulative impact of scheduling delays could hinder the system's efficiency and responsiveness

4 Questions unanswered about the paper

- **Impact of different workloads - short lived(batching) vs long training**