Avesha

An Avesha Vision
Raj Nair, Cheng Wu
December 2024

# Optimize Your ML and Data Workloads

## Complexity of ML Workload Optimization

Despite the recent advancements in machine learning job scheduling and workload management tools such as KubeFlow, Ray and AirFlow, it's still challenging to efficiently use GPUs and CPUs while keeping costs down and maintaining automation. The main problem is that the basic hardware (like GPUs and CPUs) doesn't automatically adjust to what each task needs. Which means that there is a gap between non-adaptive low-level computation hardware resource management and high-level workload orchestration, which specifies only the workload tasks and their order of execution. Ideally, resource allocation should use dynamic policies, based on the actual needs of the current and upcoming tasks. This paper looks at how making resource allocation more flexible could improve GPU usage, and how to measure, what KPIs to use, if these improvements are working well.

ML tasks must be executed to completion if the downstream tasks depend on their outputs, making just-in-time notification or prediction of their completion key to reducing resource idle time. Accurate prediction of "task completion time" in turn enables proactive pre-loading of the next tasks to further streamline the continuity of pipeline execution. Tasks that are defined to run in parallel will execute in parallel only if their dependencies allow it, implying a failure to allocate resources concurrently in time could inadvertently cause pipeline delays. Scheduling is further complicated by workload specific quota controls imposed by many organizations to ensure that workloads of high priority receive a fair share of resources, irrespective of task level prioritization. For example, a high-priority workload can preempt GPU resources from a lower priority task, even if it needs only CPU resources for its initial ETL processing, leaving the preempted GPU resources idle.

Mismatches between the ML pipeline dependencies and ML job scheduling logic can lead to idle GPUs or unnecessary delays. Tasks with higher priority may still be blocked by dependencies, and this could potentially lead to wasted resources if those higher-priority tasks cannot yet begin execution. This is a common challenge in workload orchestration systems, where resource allocation and dependency management must work together efficiently.

Efficient and fair GPU sharing and isolation for large MLOps installations are challenging. ML pipelines are often distributed or large enough to force resource isolation into separate Kubernetes clusters with complex Kubernetes namespace and pipeline security issues that are difficult to manage, let alone automate. Kubernetes schedulers and resource pools work within the scope of a single cluster. Hence, a multi-cluster approach to resource management is an important unaddressed problem that is key to a well-balanced load and capacity distribution.

The confluence of data analytics and AI makes mixed-GPU-CPU workloads a new norm and heightens the importance of a common resource allocation scheme under the common framework of Kubernetes. The mixture of GPU and CPU as well as increasingly number of specialized GPUs for domain specific models or inference makes it mandatory to devise a common computing rating system an inevitable infrastructure as a service standard that makes it easy to compare different service providers based on a well-defined industry-recognized open-source models rather than simply using current raw capacity numbers. For example, one could measure the resources used (counted as costs for CPU-mins and GPU-mins for a specific CPU/GPU instance type) per million tokens for the latest llama model as way to rate the effectiveness of a compute infrastructure. It is important to note that some models and model optimizers (for inference workloads) can run purely on CPUs very efficiently. Hence, a mixed model might require only CPUs. The normalization to costs makes this a very effective benchmark to compare compute service providers.

## An Innovative Workload Optimized Approach for Elastic Computation Services

One way of achieving ML workload automation, optimization and cost reduction lies in enabling a just-in-time orchestration of the workload pipeline. It is important that the orchestration matches pipeline inter-dependencies and its task resource requirements, in such a way that dynamic GPU allocation across heterogenous CPU and GPU pools is possible.

Most industrial workload tools use a Directed Acyclic Graph (DAG) to specify orchestration of complex, reproducible workflows for machine learning, but differ in a few key execution details including: (1) handling of pipeline inter-dependencies, (2) how and when task and workload priorities are set, (3) how much task wait time is allowed (related to opportunity cost) before a new task is started, (4) as well as specification of the target computation hardware that meets the resource requirements.

A universal DAG plug-in abstraction layer may be conceived to unify and insert these observed or implied configuration attributes into a set of orchestration insights so that a unified dynamic hardware allocator can serve as a smart agent to dynamically allocate GPUs and manage job scheduling efficiently.

We define a mechanism called Elastic GPU Service (EGS) that can adapt GPU job resource scheduling to ensure that the following requirements are met:

1. Enable look-ahead preload of next tasks based on prediction or observation of the completion time of the previous task in sequence.
2. Preempting tasks of equal priority at the end of a job queue in favor of those with equal priority but with an output dependency, ie Prioritize tasks with output dependencies over equal-priority tasks in the queue.
3. Preempting tasks in the pipeline that have reached usage limits.

4. Enablement of mixed-GPUs or mixed-CPUs based on per task workload requirement. A mixed mode workload must not tie up GPU resources if is currently working on a CPU task. Further, the CPU resources must be optimally utilized based with predictive scaling to minimize latencies that will impact DAG pipeline performance.

5. Enable near parallel allocation and loading for tasks that must be completed simultaneously.

6. Provide operator control per workload or per stage to forced abort operations.

Conceptually, the EGS serves as normalizing proxy by adapting workload wide priorities and configured inter-dependencies to the Kubernetes scheduler or any third-party job scheduler constrained by their limited visibility to only jobs and their respective priority, not workloads or their inter-dependencies.

By leveraging infrastructure and model inputs such as temperature, model size, optimization metrics, loss functions, network topology, and DAG graphs, enterprises can create and maintain dynamic GPU allocations and efficient workload mappings.

Predictive analysis based on historical execution patterns allows for intelligent allocation, addressing imbalanced GPU availability and ensuring balanced resource utilization. This approach eliminates manual workload management, enabling enterprises to focus on innovation.

Finally, an approach combining Dynamic Resource Allocation (DRA, a newly introduced Kubernetes feature) with KubeSlice (a CNCF Sandbox project) automates and optimizes job groups, DAG paths, and data flow.This approach addresses multi-cluster ML orchestration challenges, ensuring precise execution and monitoring, to achieve secure inter-cluster data flow, muti-tenancy of workloads with usage quota enforceability.

## Intelligent GPU & CPU Execution

This transformative approach empowers enterprises to maximize throughput and minimize inefficiencies -- offering a scalable and automated solution to meet the demands of resource-constrained environments. By predicting and adapting GPU allocation strategies, EGS addresses the growing need for efficiency, scalability, and cost control.

### Core Benefits:

1. **Cost Efficiency**
   Businesses can ensure GPU usage to align with their business priorities. Predictive GPU allocation enables prioritization of mission-critical workloads while reducing idle resources, significantly lowering operational expenses. This is achieved by GPU cluster time-slicing, which allows dynamic provisioning and reallocation of resources.

2. **Enhanced Observability**
   Real-time dashboards provide visibility into GPU status, resource allocation, and workflow efficiency, enabling automated remediation (e.g. on temperature or memory or power).

3. **Improved Throughput**
   Optimized job completion rates result in higher utilization and reduced idle time across GPU workloads. EGS has shown to result in up to 44% improved throughput.

4. **Automated Remediation**
   Continuous monitoring of GPU nodes and health checks minimizes manual intervention. Dynamic reconfiguration of nodes ensures adaptability to changing workload conditions.

5. **Scalability and Modularity**
   Features such as observability and cost control are independently deployable, allowing businesses to adopt capabilities incrementally. CPU resources may not be as expensive as GPU resources – however, there are certain data-preparation operations that only need CPUs and any latencies waiting for these resources will be detrimental to pipeline execution performance. Tools like Smart Scaler can increase CPU utilization to 90% and Smart Karpenter can reduce node latencies significantly through predictive scaling,

### Is your infrastructure working hard enough for you?

The most relevant KPI for your IT investment must be the Wastage Ratio (WR) defined here as the percentage of times when you have idle and available resources when there is at least one revenue-earning workload that is blocked for all or a part of the said resources. The shocking reality is that this WR ratio may not necessarily be zero – something you need to discover for your infrastructure. The closer to zero, you can get WR, the more optimal your infrastructure is working for you. Of course, this KPI does depend on how - both GPU and CPU resources– are utilized as well as the way workloads are orchestrated. Yet, it can inform how efficient you are for the needs of your own organization. Modern tools like EGS with its comprehensive observability and optimized scheduling together with Smart Karpenter and Smart Scaler can assist you in your journey to reach closer to the goal of zero WR.

## Conclusion

EGS redefines GPU resource management by delivering cost-efficiency, real-time observability, and high throughput. By combining advanced scheduling, dynamic provisioning, and seamless ML framework integration, it empowers enterprises to overcome inefficiencies and unlock the full potential of their GPU infrastructure. This solution addresses the challenges of mixed GPU-CPU workload management, enabling businesses to scale AI-driven operations with confidence and agility. As a result, enterprises can achieve reduced costs, optimized resource utilization, and a competitive edge in innovation.