WHITE PAPER

Quali | TORQUE

Optimizing GPU Allocation for AI Workloads with Dynamic Environments

AI generated Image

Introduction

To deliver high-performing Artificial Intelligence (AI) models and Machine Learning (ML) applications, optimizing resource utilization—especially GPU capacity—is critical.

These workloads often go through different phases, each with varying computational needs. For instance, training a machine learning model demands significant GPU resources, while inference tasks generally require far less. Considering that AI and ML workloads transition through these phases continuously, the ability to manage GPU allocation dynamically is critical to ensure that resources are not wasted while meeting performance requirements.

Quali Torque, a platform that abstracts infrastructure as environments, can manage these GPU resources dynamically. Torque integrates technologies like Kata Containers, Firecracker, and Kubernetes to offer a unified approach for environment provisioning, management, and optimization.

This whitepaper outlines how Torque leverages these technologies to provide dynamic GPU allocation for workloads across their entire lifecycle, improving performance and cutting costs.

The Need for Dynamic GPU Allocation and Context-Aware Optimization

Current Challenges

Over-provisioning and under-utilization

Many organizations allocate GPUs conservatively to support the maximum amount of resources required to ensure performance. This often leads to high costs and resource waste during low-demand phases like inference.

Lack of resource optimization

Without real-time adjustments, GPU resources are often either over-committed or unavailable when needed urgently.

Contextual resource allocation

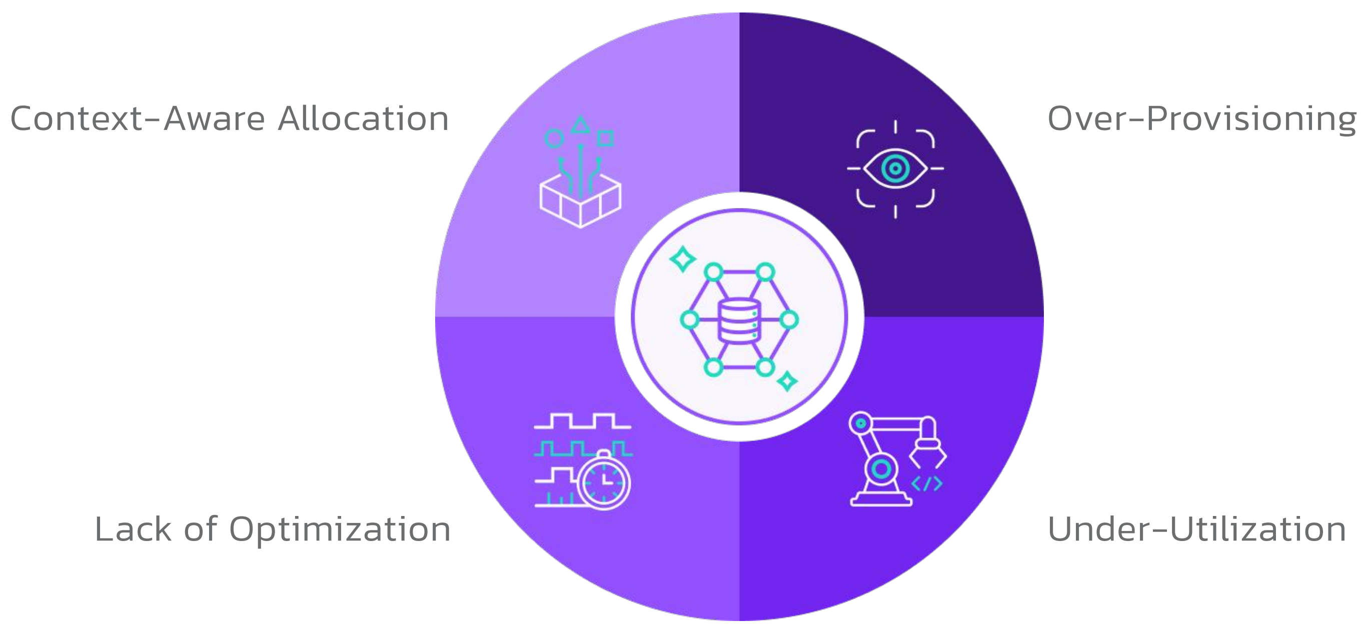
Existing solutions rarely consider the full context of a workload's lifecycle—training, inference, retraining—when allocating resources. This makes it difficult to optimize resource consumption to align with actual workload needs.

Business and Technical Requirements

Teams tasked with delivering profitable AI and ML applications need to be able to

- Dynamically scale GPU resources to align with the different stages of these workloads, from build and test to train and retrain.
- Maintain complete context for these stages of workloads to optimize application performance without incurring unnecessary expenses.

Dynamic GPU Allocation Strategies



Benefits of Dynamic GPU Allocation

Cost Efficiency

By dynamically adjusting GPU allocation in real time, Torque prevents costly over-provisioning and under-utilization of resources. In cloud environments where GPU instances are charged based on usage, Torque's ability to right-size resources for each phase of the workload lifecycle leads to significant cost savings. Just-in-time provisioning ensures that GPUs are only allocated when needed, preventing idle capacity and unnecessary expenses.

Accelerated Monetization and Return on Investment

Torque's dynamic GPU allocation accelerates the monetization of GPU investments by maximizing utilization through its self-service and automated infrastructure provisioning, allowing teams to rapidly access and deploy GPU resources exactly when needed, without manual intervention. Organizations can deploy GPUs for high-demand tasks like training or inference exactly when required, ensuring that the resources are fully utilized and directly contribute to business value. In addition to improving cost efficiency, this just-in-time provisioning shortens the time to value for GPU infrastructure, ensuring a faster return on investment (ROI) by eliminating waste and ensuring that every GPU cycle contributes to active workloads.

Enhanced Performance

Torque's continuous optimization ensures that workloads always receive the resources they need, when they need them, resulting in faster training times and lower latency in inference tasks. This allows businesses to meet or exceed performance benchmarks for critical AI/ML models, driving faster insights and more responsive AI systems.

Operational Agility

Torque's centralized operational hub provides full visibility and control over GPU resources, allowing organizations to adjust workload priorities on the fly. The ability to adjust in real-time enhances operational agility, ensuring that resources are directed to the most important tasks as business needs evolve.

Scalability and Flexibility

Torque's architecture scales effortlessly, whether managing a handful of models or hundreds across multiple environments. By providing just-in-time resource provisioning, Torque ensures that each workload is right-sized for the available GPU resources, supporting rapid scaling without over-commitment or performance degradation.

Self-Service and Developer Empowerment

Torque's self-service capabilities enable developers and data scientists to directly provision environments and access GPU resources as needed, without waiting for IT teams to manually allocate resources. This not only accelerates the development lifecycle but also reduces operational bottlenecks, empowering teams to move faster and innovate more effectively.

Use Case Scenarios

To understand the benefits for this approach, it can help to consider real-world examples where GPU workload optimization can play a role in the performance and financial viability of an AI model.

Training Job Submission
A data scientist needs to initiate training for an AI model. Torque provides a simple self-service interface through which the data scientist can easily initiate the training job in just a few clicks, without requiring extensive training or intervention from an IT or engineering administrator.

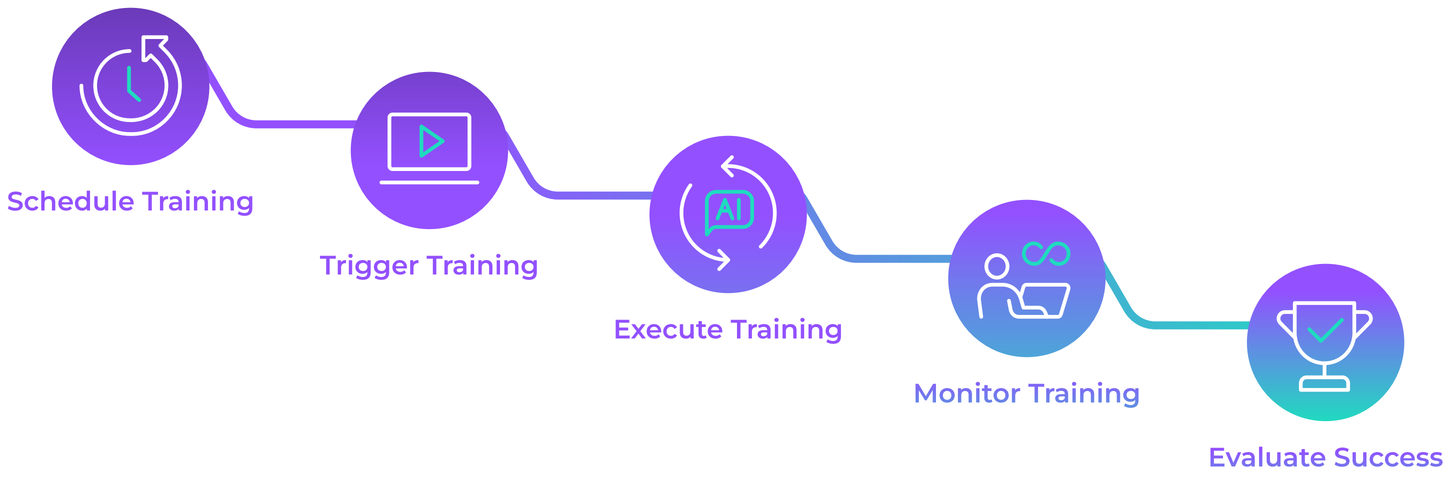
GPU Scaling to Support Training
To support the high GPU capacity demands to perform training for an AI model, Torque dynamically adjusts GPU allocation automatically.

Automation for AI Model Training
If needed, Torque can automate the initiation of training for AI models based on custom schedules or in response to events, while providing visibility for data scientists to understand when training occurred and whether it was successful. This eliminates manual work required to perform routine training processes.

Transition to Inference
As the model transitions from training to the inference phase, Torque reduces GPU allocation, freeing up resources for other tasks while maintaining performance. This prevents wasted costs from over-provisioned resources to support the inference phase.

Continuous Model Retraining
When models are retrained based on new data, Torque dynamically increases GPU resources during retraining to ensure a quick turnaround. This occurs without disrupting other ongoing tasks like inference, providing an efficient, real-time response to workload changes.

Cross-Workload Optimization
In environments running multiple AI models, Torque optimizes GPU usage across workloads. For instance, if a high-priority real-time inference task requires immediate resources, Torque reallocates GPUs from lower-priority tasks, ensuring optimal performance for critical workloads.



Key Technologies Enabling Dynamic GPU Allocation

Torque’s Environment-as-a-Service Model:

Torque manages workloads as dynamic environments, applying custom policies for individual environments that control GPU allocation at every phase of the workload’s lifecycle. By maintaining a holistic view of these environments, Torque ensures resources are always appropriately allocated based on real-time requirements and cost constraints and business priorities.

Torque’s role as the orchestration, deployment, and control layer makes it easier to assign context throughout the various stages of a workload, thereby enabling intelligent decision making about resource utilization.

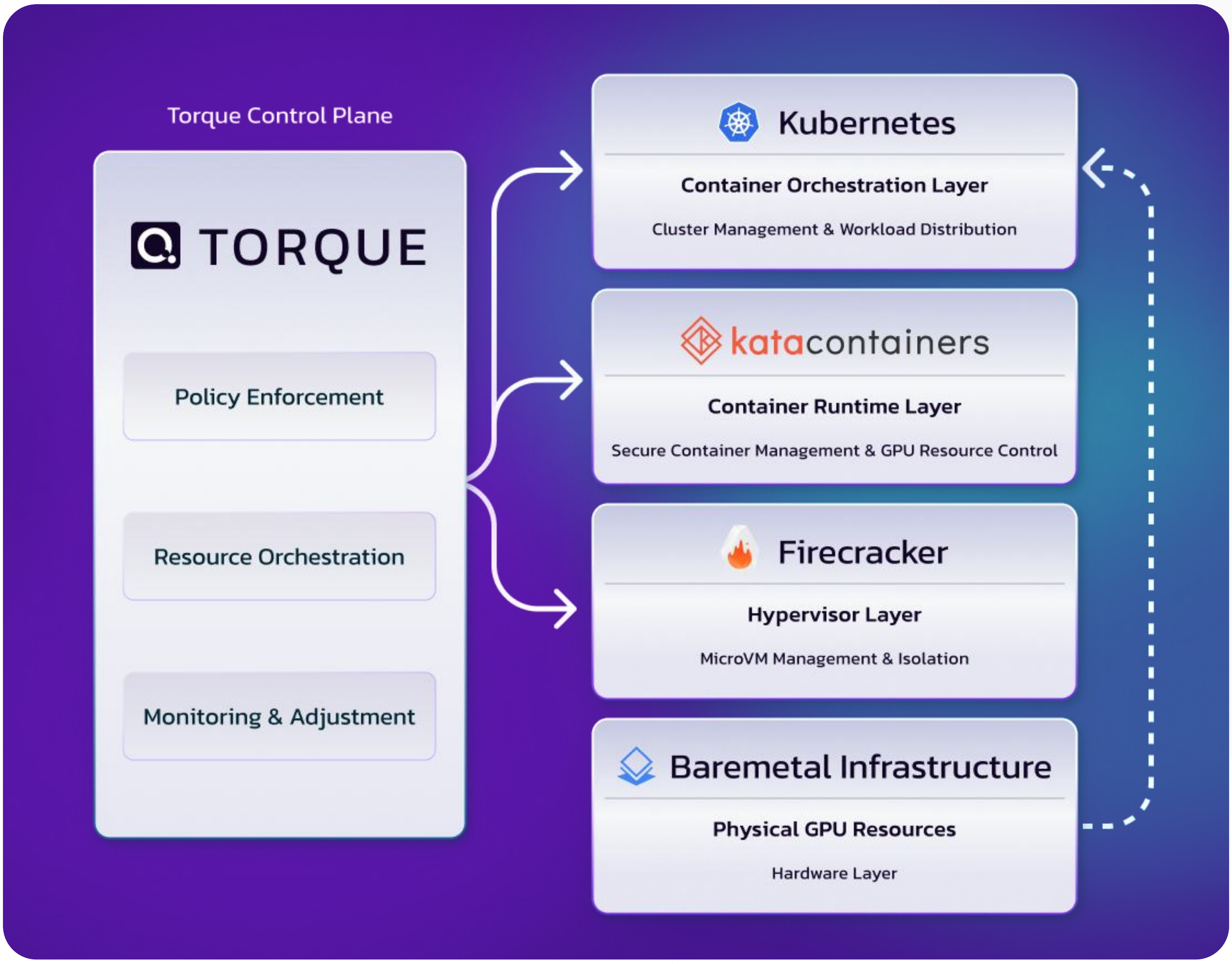
Firecracker and Kata Containers

Firecracker provides lightweight, secure microVMs, allowing the rapid and efficient launch of isolated GPU-enabled environments. It is designed to be highly resource-efficient, making it ideal for ephemeral workloads that require secure, on-demand compute power.

Kata Containers complement this by providing the isolation and flexibility of virtual machines, while retaining the speed and lightweight nature of containers.

Kubernetes

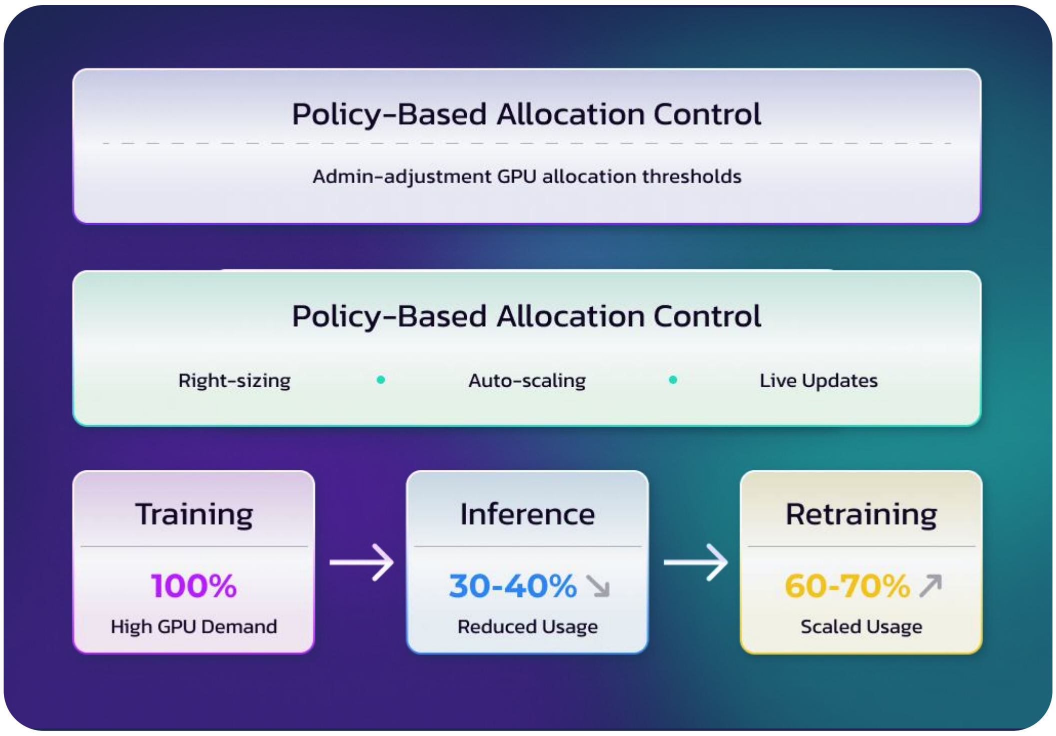
Kubernetes orchestrates the containerized workloads and ensures that resources, particularly GPUs, are allocated, scaled, and reallocated based on demand. Kubernetes' GPU plugin supports the dynamic scheduling of GPUs for ML/AI tasks.



Dynamic GPU Allocation: Workflow and Architecture with Torque

Architecture Overview

- Torque integrates with Kubernetes, Firecracker, and Kata Containers to manage AI/ML workloads as dynamic environments.
- GPU capacity is allocated and adjusted dynamically across different stages of the AI/ML lifecycle, such as training, retraining, inference, and model maintenance. The key feature here is Torque's optimization layer, which continuously monitors workloads and adjusts GPU allocation in real-time based on the specific requirements of the task at hand.



Torque's Optimization Layer and Operational Hub

Optimization Layer

Torque's optimization layer dynamically manages GPU allocation across all live workloads by continuously tracking performance and resource usage. It ensures that GPU resources are right-sized for the current phase of each workload (e.g., training, retraining, or inference), maximizing efficiency while minimizing waste. This real-time adjustment helps align resource allocation with the specific demands of each phase.

Context-Aware Prioritization

Torque adapts GPU allocation based on the operational context and workload phase. For example, during a model's training phase, significant GPU resources are allocated to accelerate the process. When the model shifts to inference, requiring fewer resources, Torque scales back GPU allocation. In cases where retraining a critical model (e.g., fraud detection) takes priority, Torque reallocates resources to expedite retraining without impacting lower-priority inference tasks. This ensures that GPU resources are continuously optimized based on workload priority and lifecycle phase

Operational Hub

Torque's operational hub provides centralized visibility and control over all workloads, enabling businesses to enforce GPU allocation policies based on real-time performance and shifting business priorities. This hub facilitates seamless optimization and prioritization of GPU usage across multiple workloads, ensuring an automated, efficient resource allocation process.

Conclusion: Unlocking the Full Potential of AI Workloads with Torque

Torque offers a revolutionary approach to GPU resource management for AI/ML workloads. By representing workloads as dynamic environments, applying real-time policies, and optimizing GPU allocation across workload phases, Torque enables organizations to maximize the performance of their AI systems while reducing operational costs. The integration of Firecracker, Kata Containers, and Kubernetes ensures that these environments are lightweight, secure, and scalable, delivering unparalleled flexibility and efficiency in resource management.

References

- Kubernetes GPU Scheduling Documentation
- Firecracker MicroVM Technical Overview
- Kata Containers Architecture Guide
- Torque Dynamic Environment Management