

IT & DATA SCIENCE

Introducing Run:ai's CPU Scheduling: Improved Productivity and Utilization for CPU-only Clusters



Table of contents

Introduction	2
The Challenge with Kubernetes CPU Scheduling	3
Static Quotas and Efficiency	3
Limitations in Advanced Scheduling Strategies	3
Gaps in Handling Batch Distributed Computing Workloads	3
Introducing Run:ai's CPU Scheduling Solution	4
Dynamic Quotas for Enhanced Productivity	4
Multilayered Fairness and Hierarchical Resource Sharing	4
Elastic Scheduling for Adaptive Workloads	4
Unlocking the Benefits: Improved Resource Utilization and User Productivity	5
Conclusion	6



Introduction

Distributed computing frameworks such as Apache Spark, Dask, and Ray, have become cornerstones of modern data analytics, machine learning, and data processing workflows. These frameworks are leveraged for tasks ranging from machine learning model training and data analytics to running complex data pipelines. As their utilization surges, scheduling strategies become more important for user productivity, utilization, and resource allocation.

Run:ai has extended its advanced scheduling capabilities to address the challenges organizations face when orchestrating CPU workloads within their Kubernetes environments. The growing adoption of Kubernetes for orchestrating container workloads brings both convenience and challenges, particularly in the context of distributed computing workloads running on shared clusters of compute.

 [Run:ai's CPU Scheduling Demo](#)



The Challenge with Kubernetes CPU Scheduling

Traditionally, organizations have sought to centralize their compute resources, enabling various teams and users to access and run their workflows efficiently. Kubernetes, while a powerful container orchestration tool, still presents several challenges when it comes [to scheduling workloads on shared clusters of compute resources](#).

> Static Quotas and Efficiency

Traditional Kubernetes resource management relies on fixed quotas for CPU and memory allocation to different teams or users. However, these static quotas often lead to inefficiencies as users and teams risk being trapped by static limits, regardless of available resources in the cluster.

In busy times quotas can be too high, potentially letting users starve other users and introduce long idle times. During quieter times quotas can be too low, preventing users from exploiting idle resources and running more workloads. The lack of dynamic quotas not only lowers cluster utilization but also hampers user productivity.

> Limitations in Advanced Scheduling Strategies

Kubernetes scheduler has limited support in advanced scheduling strategies like bin packing and consolidation for batch workloads which optimize resource usage by placing and moving workloads to run on as few nodes as possible. Instead, the Kubernetes scheduler by default spreads workloads across nodes, resulting in suboptimal resource utilization and reducing the number of available resources for running big distributed workloads.

> Gaps in Handling Batch Distributed Computing Workloads

The default Kubernetes scheduler [lacks important scheduling capabilities](#) needed for running batch distributed computing workloads. Distributed workloads consist of multiple pods that need to be scheduled and orchestrated together. However, [the default scheduler lacks Gang Scheduling capabilities](#) and schedules pods individually, leading to uneven distribution and potentially resource wastage. Non-elastic scheduling further compounds these problems as workloads that have the potential to scale based on resource availability, like with Apache Spark and Ray, cannot dynamically adjust to accommodate changing demands. This leads to inefficient resource utilization, as workloads either remain constrained within their original boundaries or risk overloading the cluster.



Introducing Run:ai's CPU Scheduling Solution

Recognizing the shortcomings of existing approaches, Run:ai has engineered a robust solution to tackle the intricate challenge of CPU scheduling. By integrating its advanced scheduler into Kubernetes environments, Run:ai empowers organizations to optimize resource allocation, boost productivity, and achieve higher utilization rates.

Dynamic Quotas for Enhanced Productivity


Run:ai's scheduler introduces the concept of dynamic quotas. Instead of rigidly defined quotas, users and teams are allocated guaranteed resources that they can access at all times. Moreover, if additional resources become available in the cluster, users can opportunistically access them, further enhancing their productivity and workloads' efficiency. This approach ensures users receive resources when needed, even if their guaranteed quotas are exceeded temporarily.

Multilayered Fairness and Hierarchical Resource Sharing

The scheduler supports multiple hierarchical teams and departments, allowing resources to be shared and prioritized across different levels. A fairness algorithm, [dominant resource fairness \(DRF\)](#), ensures that shared CPU compute and memory resources are allocated fairly based on policies, priorities, and usage. Users can then access resources not only within their team but also borrow resources from other teams or departments if they are underutilized. This adaptable resource sharing fosters a fair and efficient distribution of cluster resources while ensuring priority access for high-priority tasks.

Multilayered Fairness and Hierarchical Resource Sharing

In scenarios where workloads can dynamically adjust their resource needs, the scheduler optimally manages resource allocation. When resources are scarce, the scheduler can automatically shrink resource-intensive workloads, making space for other workloads. These workloads can then automatically expand when resources become available again, enhancing cluster efficiency. This capability prevents unnecessary workload preemptions and ensures efficient utilization of resources even as workloads dynamically expand and contract.



Unlocking the Benefits: Improved Resource Utilization and User Productivity

By adopting Run:ai's CPU Scheduling, enterprises can unlock a range of benefits, including:

- 1 Enhanced Cluster Utilization**

Run:ai's dynamic quotas and elastic scheduling prevent underutilization of cluster resources, ensuring every available CPU and memory unit is leveraged effectively.
- 2 Higher Productivity**

Users and teams experience increased productivity by accessing more resources when needed and running their workloads more efficiently without being limited by rigid quotas.
- 3 Reduced Dependency on IT**

Allowing users to go above their quota in a self-service manner reduces users' reliance on IT administrators to manually adjust quotas and resources, freeing up IT to focus on more strategic tasks.
- 4 Single Software Stack**

Enterprises can streamline their infrastructure by consolidating resources into a single shared environment and focusing on a single cloud-native software stack to maximize the value of their existing investments and simplify operations and management overhead.
- 5 Fair and Equitable Resource Allocation**

Multilayered fairness ensures priority access to critical tasks while optimizing resource distribution among teams, projects, and departments.
- 6 Efficient Distributed Computing**

Run:ai's scheduler understands the unique demands of distributed computing workloads, ensuring optimal packing strategies and efficient execution.
- 7 Visibility into Infrastructure Resources**

With Run:ai administrators can get centralized visibility into how resources are being utilized and allocated at the user, team, department, project and job level, across clouds and on premises from a single location.

Conclusion

In a world where computational demands continue to grow, scheduling emerges as a key ingredient of efficiency and innovation, enabling organizations to harness the full potential of their CPU resources while ensuring equitable access and maximum user productivity. By addressing the limitations of static quotas, introducing dynamic quotas, ensuring fairness in resource allocation, and enabling elastic scheduling, Run:ai's Scheduler brings to distributed computing in the enterprise more efficiency and user productivity.



About Run:ai

Run:ai is an AI management platform for MLOps, Data Science, and DevOps teams. In addition to helping these teams access and utilize their GPU resources more effectively, it also has a powerful set of features that can abstract infrastructure complexities and simplify the process of training and deploying models. With or without a GPU shortage, Run:ai enables data scientists to focus on innovation without having to worry about resource limitations.

Read more about how Run:ai supports
data scientists here

www.run.ai/runai-for-data-science