pepperdata

Pepperdata Saves Large Consumer Internet Company 33% for Apache Spark on Amazon EKS

Pepperdata Capacity Optimizer maximizes the performance of a massive Apache Spark on Amazon EKS environment running Karpenter and YuniKorn

Concern for Growing Apache Spark Costs

Concerned about ever-increasing costs to run Apache Spark application workloads in their growing Amazon EKS environment, the big data platform team at a consumer financial services brand in the Fortune 500 embarked upon an executive-led optimization initiative to lower monthly costs. In addition to its Amazon EKS environment, the company had also pioneered the addition of a series of Kubernetes-adjacent technologies, including **Karpenter for advanced autoscaling** and **YuniKorn for advanced scheduling**.

Pepperdata Proof of Value Delivered the Required Cost Savings

To recapture the notorious compute waste inherent in Spark, the company tapped Pepperdata Capacity Optimizer, a solution that automatically optimizes cost and utilization rates in real time without requiring any application code changes.

Capacity Optimizer immediately addressed the problem of waste inside this customer's Spark applications, a problem not in scope for common optimization solutions such as manual application tuning, Spark Dynamic Allocation, Managed Autoscaling, and Instance Rightsizing. Pepperdata demonstrated its benefits through a two-week Proof of Value comprising two segments:

Segment 1: Capacity Optimizer was deployed to the company's **development cluster** to quickly validate the solution robustness and cost saving results.

Segment 2: Capacity Optimizer was deployed to the company's **production cluster** to double validate Pepperdata's ability to support the company's scale and robustness

In just days, Capacity Optimizer enabled the customer to **run the same workloads with 33 percent less infrastructure**, helping deliver a **corresponding decrease in the cost** as measured by reduced vCPU hours, even with larger memory requirements and workloads with more volatility on average than any other environment previously addressed by Pepperdata. Pepperdata immediately and measurably improved the company's performance/price value while decreasing resources required to run Spark applications on Amazon EKS.

This example dashboard graph from a month of recent production cluster activity showed that Pepperdata added the equivalent of over one million core-hours on top of the base core-hour amount, representing a 30 percent uplift during busy usage time.

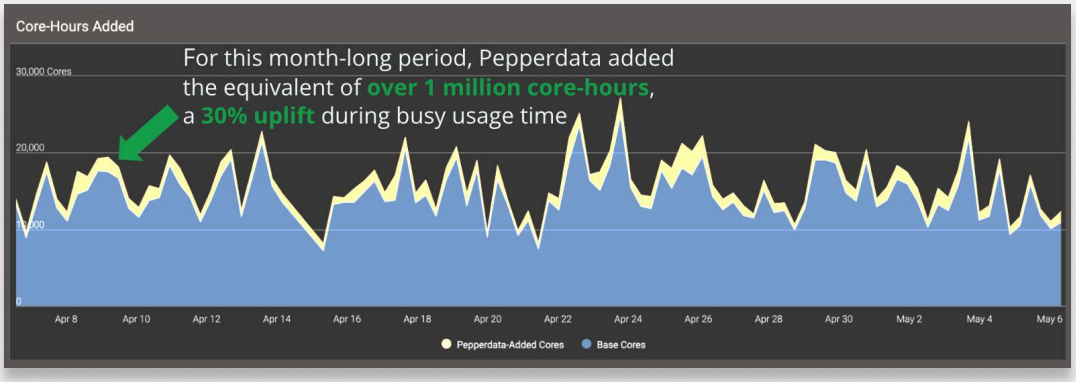

Figure 1: *Pepperdata Capacity Optimizer added the equivalent of over a million core-hours in a month, representing a 30 percent uplift during busy usage time.*

Ultimately, during the two-week POV, Capacity Optimizer achieved the company executive leadership team's cost optimization goal and then continued to **deliver 33 percent cost savings results on an ongoing basis**, which translates into a minimum of **approximately $100,000 per month in reduced costs** for this company.

Furthermore, because Capacity Optimizer works continuously in real time, **no application changes or manual interventions were required**. The customer could have chosen to implement a recommendation, but none were ever necessary since Capacity Optimizer optimizes autonomously. For this customer, that meant dozens of engineering hours freed up each week for innovation and higher value activities.

How Pepperdata Capacity Optimizer Immediately Reduced Costs by 33 Percent for Apache Spark on Amazon EKS

Installed in about an hour, Capacity Optimizer increased node-level utilization in real time without the need for application changes. This type of optimization is known as **Continuous Intelligent Tuning**. Continuous Intelligent Tuning enables the scheduler to make use of the node resources that are allocated but not used. Capacity Optimizer makes the allocated but unused resources visible to the scheduler so that it can launch more workloads on existing nodes before new nodes are added.

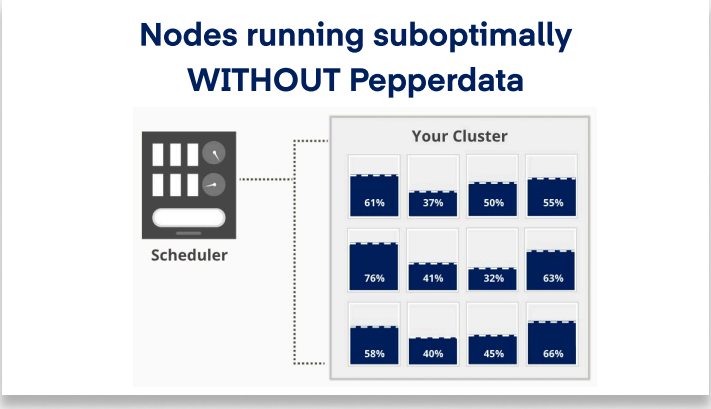

Figure 2: *Without Pepperdata, the scheduler does not see that allocated node resources are not fully utilized, leading to wasted capacity, reduced efficiency, and increased cloud cost.*

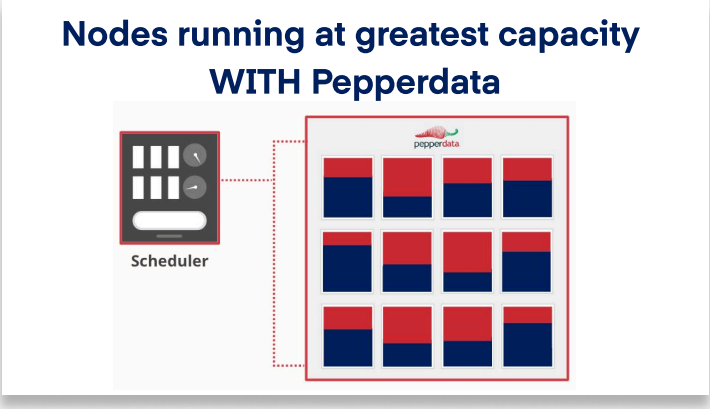

Figure 3: Pepperdata increases capacity, utilization, and savings *by communicating to the scheduler that nodes still have allocated but unused resources, allowing the scheduler to add more jobs to existing nodes.*

Without Capacity Optimizer, the cluster scheduler is unaware that allocated node resources are not fully utilized, leading to wasted capacity and cost. In most environments, typical cluster resource usage is between 30 and 65 percent. **Pepperdata can increase utilization to the 85 percent level or above** by automatically and continuously communicating to the scheduler that nodes still have allocated but unused resources, allowing the scheduler to add more jobs to existing nodes. Capacity Optimizer also optimizes autoscaling by ensuring that new instances are launched only when the existing instances are fully utilized.

Capacity Optimizer Supports Leading-Edge Kubernetes Technologies

As the company turned to Capacity Optimizer to optimize its Kubernetes environment, the cutting-edge Kubernetes-adjacent technologies that the company had implemented presented an opportunity for Pepperdata to enhance and extend the cost-saving and efficiency-enhancing capabilities of Capacity Optimizer.

Pepperdata Augments the Benefits of Karpenter

To simplify and automate the management of its massive cluster, the company chose to use Karpenter, one of the fastest growing and most widely adopted Kubernetes add-ons. Karpenter automates node provisioning and scaling, which optimizes resource utilization and ultimately minimizes infrastructure costs.

Although Karpenter is excellent in providing cost optimization through rightsizing compute resources, it doesn't address the problem that Apache Spark applications tend to waste the resources that are provisioned for them. Spark is well known to be inherently resource-intensive in its design, especially with regard to memory. Even with the most efficient, Karpenter-optimized infrastructure in the world, overprovisioned Spark applications will use infrastructure resources inefficiently. Pepperdata uniquely addresses this inefficiency through Real-Time Cost Optimization, providing a complementary benefit to the use of Karpenter.

Pepperdata Contributes to YuniKorn to Drive Further Efficiencies with Optimized Scheduling

For more efficient scheduling in their large-scale Amazon EKS cluster, the company uses YuniKorn. YuniKorn is an open-source scheduler that dynamically adjusts resource allocations based on both developer-configured priorities and workload demands to optimize resource utilization and performance for heterogeneous workloads on large scale, multi-tenant, cloud-native environments. YuniKorn integrates with Kubernetes to either complement or replace the default Kubernetes scheduler.

Despite its incredible power and flexibility, YuniKorn is still early in its evolution. Pepperdata worked closely with this customer to ensure flawless operation of Capacity Optimizer alongside the YuniKorn scheduler. As part of this work, the Pepperdata engineering team discovered a stability issue and spent the time to create and contribute an improvement to the YuniKorn codebase, working with the vibrant and active open-source community to improve the overall robustness of YuniKorn.

Above and Beyond: Pepperdata Achieves a Tenfold Footprint Decrease for its Bundled Prometheus

Prometheus is a powerful and versatile yet remarkably simple open-source application for event monitoring and alerting, capable of reliably handling the billions of metrics generated in dynamic cloud environments. Prometheus has become a de facto standard for monitoring Kubernetes environments, including the large-scale environment operated by this customer.

In this customer's environment, Pepperdata made use of Prometheus as a bundled component to provide key optimization metrics to Capacity Optimizer. Pepperdata developed a specialized configuration for Prometheus that reduced the memory footprint by nearly tenfold. Despite this customer's very large-scale environment, Pepperdata Capacity Optimizer with Prometheus as a bundled component exhibited a very small memory footprint, which improved stability and reliability.

Outsized Gains in a Massive, Innovative Kubernetes Environment

Pepperdata's work with this customer delivered significant cost savings through powerful, real-time, and continuous optimization across Apache Spark workloads in a highly-scaled and innovative Amazon EKS environment that was also running Karpenter and YuniKorn.

Working safely and autonomously in the background, Capacity Optimizer addressed the issue of in-application waste to increase efficiency while also eliminating the hassle and uncertainty of manual tuning. Capacity Optimizer delivered this customer significant and measurable ROI from both the 33 percent cost reduction and the additional efficiency gains they achieved. And by removing the need for manual tweaking and tuning of applications, Pepperdata additionally empowered this company's developers to focus on production and innovation.

About Pepperdata

Pepperdata is the only cost optimization solution that delivers up to 47% greater cost savings—continuously and in real-time—on Amazon EMR and EKS with no application changes or manual tuning. Our customers include the largest, most complex, and highly-scaled clusters in the world, at top enterprises such as Citibank, Autodesk, Royal Bank of Canada, and those in the Fortune 5. For more information, visit pepperdata.com.

Pepperdata, Inc.
530 Lakeside Drive, Suite 170
Sunnyvale, CA 94085
USA

Start a Free PoV
www.pepperdata.com

Send an Email
eval@pepperdata.com