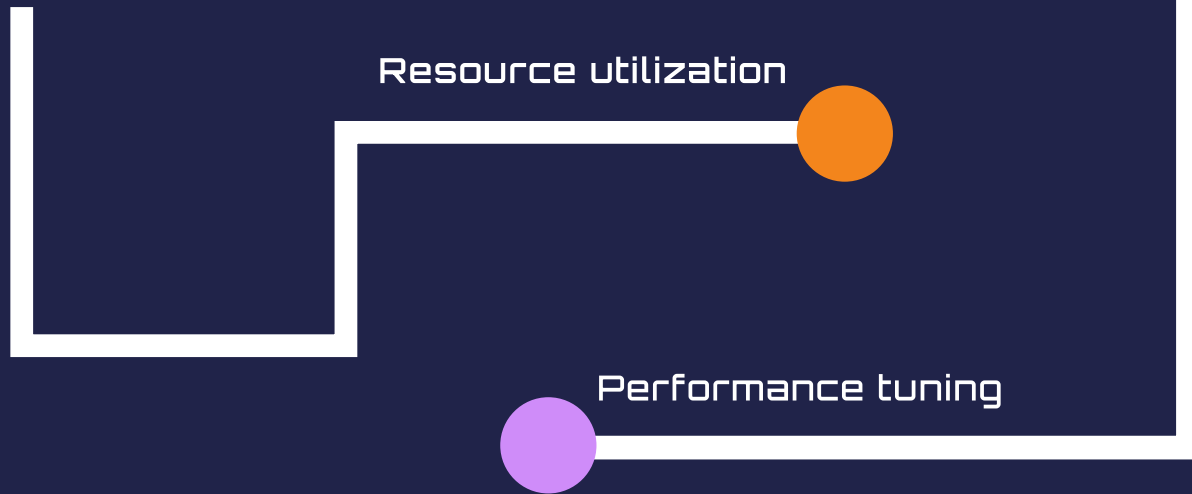


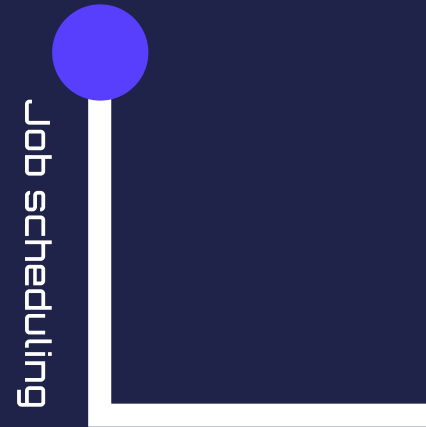
intel Granulate



A Comprehensive Guide to

# BIG DATA

OPTIMIZATION



# Table of Contents

Executive Summary	2
The State of Big Data	3
The 4 Main Challenges of Big Data Optimization	4
Getting to Know Big Data Architecture	7
Optimizing Big Data Platforms	10
Infrastructures and Execution Engines	12
Resource Orchestration Tools Best Practices	15
Optimizing Big Data Workloads with Intel Granulate	18
About Intel Granulate	22
References	23

Automated orchestration

# BIG DATA

Data processing

ETL pipelines

Performance tuning

Scalability

Job scheduling

Resource utilization



# Executive Summary

Data velocity is escalating at unprecedented levels and forward-thinking enterprises have accelerated the adoption of big data solutions. Despite the current economic climate, investments in big data continue to grow.

At the same time, there are continuing concerns about complexity and costs. Frameworks have become incredibly complex, requiring more specialized knowledge and time-consuming management. Spending on big data solutions, especially in the cloud, requires more stringent cost controls and better allocation of resources. Organizations cannot afford the high cost of inefficiency.

Enterprises are still expected to deliver on ambitious growth goals and sales targets despite shrinking budgets. During market downturns, only essential services survive cuts. Businesses will need to do more with less and provide a superior customer experience even with a smaller staff.

This requires relentlessly powerful automation and application performance. Investing in autonomous, continuous optimization will immediately result in lower compute costs for businesses, helping them maintain a high level of service so they can continue to grow regardless of the macroeconomic conditions.

In this eBook, we'll discuss the current state of big data, the main challenges involved in optimizing resource allocation and managing costs, explore several of the more popular infrastructure and execution engines, and best practices for optimizing workloads. Finally, we will present solutions for [optimizing big data workloads](#) and controlling costs.



# The State of Big Data

The big data analytics market was valued at \$271.8 billion in 2023 and is forecast to grow to \$655.5 billion by 2029, a CAGR of 24.4%.[1] Digital transformation across nearly every industry is driving growth as more companies invest in artificial intelligence, machine learning, predictive intelligence, and automation.

## As the scale of data adoption increases, so does the complexity

93% of enterprise businesses are using a multi-cloud strategy. 87% have a hybrid cloud environment. In more than half of companies, cloud resources teams are responsible for maintaining their environments and managing costs[2].

**\$271B**

Value of Big Data analytics market in 2023

**28%**

Cloud spend wasted

**93%**

Businesses that use multi-cloud strategy

**87%**

Enterprises that have a hybrid cloud environment

At the same time, however, companies of all sizes are struggling to manage their environments efficiently and contain costs. When asked to rank the top challenges for cloud resources and managing big data, enterprise companies and small to medium-sized businesses (SMBs) both put managing cloud spending at the top of their list.[2]

Cloud infrastructure spending has slowed a bit due to economic factors, but management costs continue to climb.[3] Self-reporting of wasted cloud spend hovers around 28%. Although this is down slightly from the 32% reported in 2022, the number is staggering. Nearly a third of all spending on cloud and data is wasted or unaccounted for.[4]

With all of these wasted resources in the cloud, specifically for data streaming and processing, application optimization has become a top priority for data engineering teams at digital native businesses and enterprises.

# The Four Main Challenges of Big Data Organizations

1

## Complex Data Workloads

97% of data engineers say they are burned out from managing day-to-day tasks.[6]

3

## Scalability

49% of companies say they cannot keep their cloud and data costs under control.[8]

2

## Big Data Visibility

54% say they only have visibility into half of their workloads.[7]

4

## Dynamic Nature of Workloads

28% of cloud infrastructure spending is wasted or unaccounted for.[9]

# The Four Main Challenges of Big Data Organizations



## Complex Data Workloads

Even the most highly skilled data engineers can quickly be overwhelmed by today's complex data workloads. It starts with the sheer amount of data being gathered.

The amount of data collected continues to accelerate and it's more diverse than ever before. It's not just numbers and transactions in databases. As much as 80% of the data gathered is unstructured.[10] In the past, this so-called dark data was left unanalyzed. Today, it's yielding significant insights for companies, but it has significantly increased the challenges of analyzing big data.



## Big Data Visibility

Managing cloud resources, storage, compute resources, and orchestration has become increasingly challenging across larger data sets, more diverse data, and more environments. In one study, 54% of data professionals reported they have visibility into half (or less) of their organization's infrastructure.[7]

A lack of visibility into data workloads and performance makes it difficult to optimize performance and identify workflow bottlenecks. Monitoring, managing, and optimizing performance across complex data workloads can be overwhelming, especially if you don't have strong visibility into everything. Currently, teams often have to juggle multiple resources and visibility tools to keep on top of changes in real-time.



## Scalability

High volume workloads are hard to optimize, especially when you're scaling up and down frequently. Data engineers have to consider large volumes of data in a variety of formats and at high speeds, managing data consistency and partitioning in distributed systems, and addressing issues of data quality, security, and compliance as the data scale.

Engineers must also consider the infrastructure cost and limitations of hardware and networks along with the need to optimize the use of resources such as compute, storage, and networking to handle data processing at scale.



## Dynamic Nature of Workloads

Workloads are constantly changing and evolving. Even minor changes in workloads can negatively impact cluster performance and require retuning of the code and configuration. Changing data structures and formats, spikes or drops in data volume, and ever-evolving use cases and pipeline requirements produce wide variations, sometimes from hour to hour.

Data engineers must be able to manage environments to account for this dynamic nature, yet many data teams view workload configurations as “set it and forget it”. This requires changes in pipelines, volumes, and sources affecting workloads to be handled manually, increasing the work for data teams and leading to errors and delays.[5]

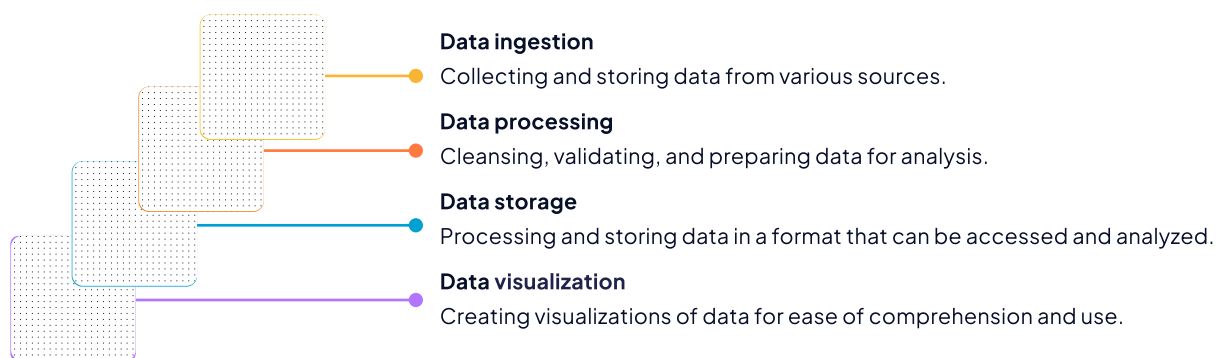
Data engineers also have to manage the entire data and analytics lifecycle, including:



# Getting to Know Big Data Architecture

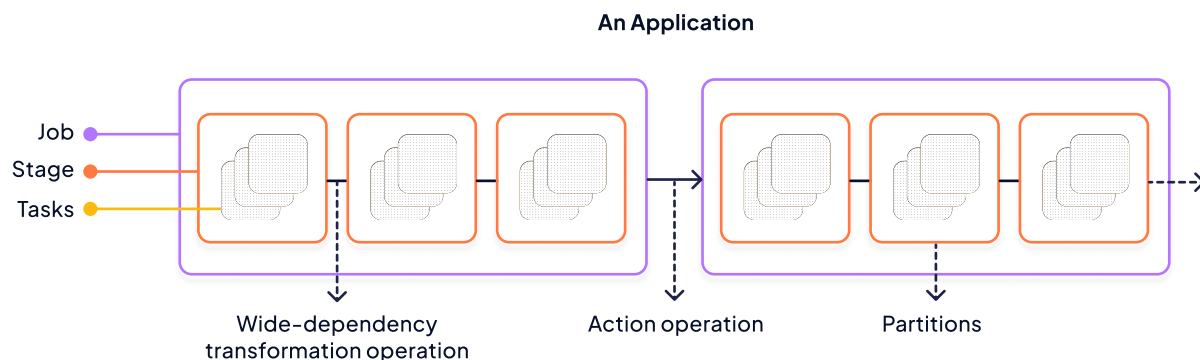
Big data architecture must be capable of handling the complexity, scale, and variety of data while simultaneously supporting the needs of diverse user groups.

There are four main layers in big data architecture:



Big Data applications are mapped to distributed processing to increase parallelism and minimize time to completion.

Each application is divided into staggered Jobs, and each job is divided into staggered stages with wide dependencies. Each stage processes multiple tasks in parallel. A stage is generally a collection of Tasks. A task is a unit of work that can be run on a partition of a distributed dataset and executed on a single executor. All the tasks within a single stage can be executed in parallel.



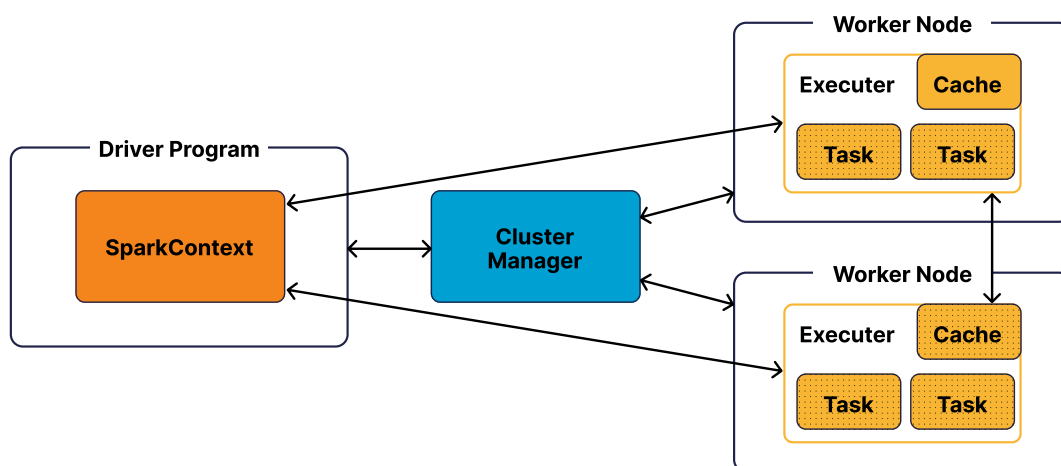
An executor is a single JVM process launched for an application on a worker node. Each application has its executors. A single node can run multiple executors, and executors for an application can be run on multiple worker nodes. The number of executors for a spark application can scale up and down upon tasks backlog time and idle executor time, respectively.

The unit of parallel execution is at the task level. The number of cores per executor determines the parallelism of an executor. A core is a basic computation unit of a CPU, and a CPU may have one or more cores to perform tasks at a given time.

The more cores we have, the more work we can do. In Spark, this controls the number of parallel tasks an executor can run. The more CPU cores an executor provides, the higher level of parallelism can be achieved.

The Cluster Manager schedules and divides resources in the host machine, which forms the cluster. The prime work of the cluster manager is to divide resources across applications. Apache Spark system supports three types of cluster managers: Standalone, YARN, and Mesos.

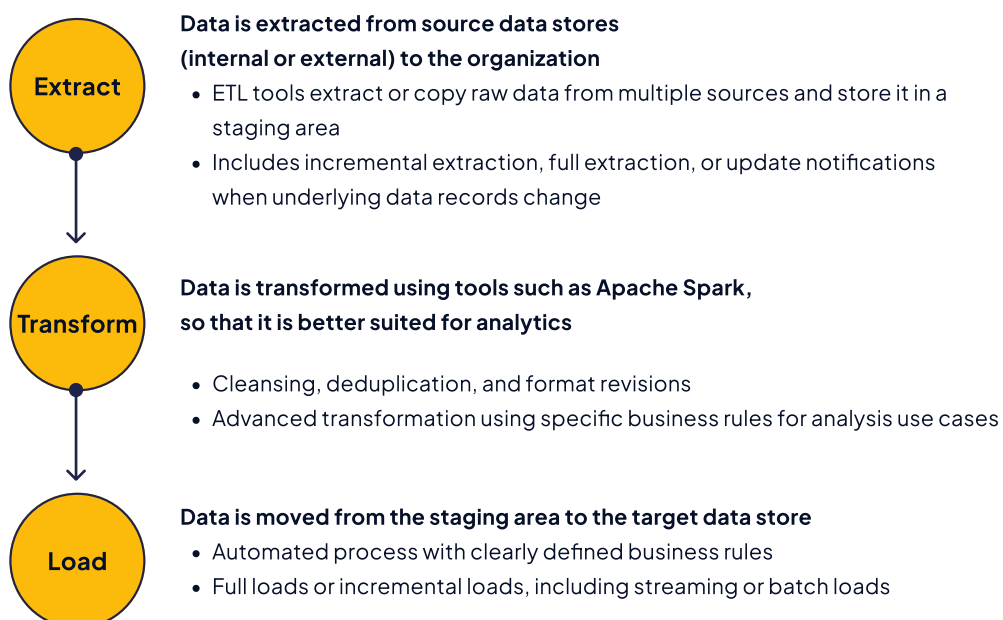
The most common big data infrastructure is based on Spark and YARN. Spark is a data processing framework that can perform tasks on large data sets and distribute data. YARN is the resource negotiator, the middle layer between Spark and the Hadoop file system.



## ETL Pipelines

Poor data quality, data integrity, and duplicate data are constant concerns for data engineers. Building an efficient and well-architected data pipeline requires a robust data architecture and consistent application. Generally, this requires an extract, transform, load (ETL) mechanism to orchestrate data transformation across multiple steps.

In a big data project, the collected data serves many different use cases across departments. Each department likely has its own ETL pipeline that transforms the incoming data before loading it into a database.



Data engineers also employ extract, load, and transform (ELT) that mixes the order of operations. In this case, a staging area is not required. Instead, data is mapped to the target data lake or warehouse. Data can be loaded directly into the target store before processing.

ETL works well for high-volume and unstructured datasets that require frequent loading.[11] Planning for analytics can also be completed after data extraction and storage have been completed, leaving the bulk of the transformation for the analytics stage.



# Optimizing Big Data Platforms

Optimizing big data platforms requires a well-architected infrastructure and consistent adherence to business rules. Otherwise, data can be stuck in disparate locations and egress fees can eat up your budget.



## De-Silo Data

Efficient data stores eliminate data silos throughout an organization. You must be able to ingest any data from any source and analyze it in a central platform. Data lakes allow you to store data at scale. Created as a data fabric, you can access data without having to move it across multiple cloud service providers to get it into your analytics platform.



## Utilize Cost Controls

As more data migrates to the cloud, costs continue to rise. The more often you move and transform data for analysis, the more expensive it gets. Simplifying your data architecture can help reduce expenses, but this requires careful monitoring and automated tools for optimization. Today's environments are typically too complex to manage manually.



## Automate Orchestration

Automating orchestration enables data engineers to provision, configure, and manage infrastructure and software services, including containers, virtual machines, and microservices.

**Tools like Intel Granulate can streamline and simplify complex processes, allowing data teams to focus on high-value tasks and letting Intel Granulate's algorithms find the right solutions to optimize cost and performance.**

Automated orchestration improves the efficiency and reliability of software development and deployment, reducing the risk of errors and improving time to market. This allows greater scalability, especially for large-scale IT environments that require rapid changes.



## Structure and Use-Case Agnostic

Your data architecture must be able to handle data that is structured, unstructured, or semi-structured equally well. As data analytics can range across a wide variety of use cases, data must be operational for each use case to extract optimal value. This broadens data usage and helps future-proof your data.

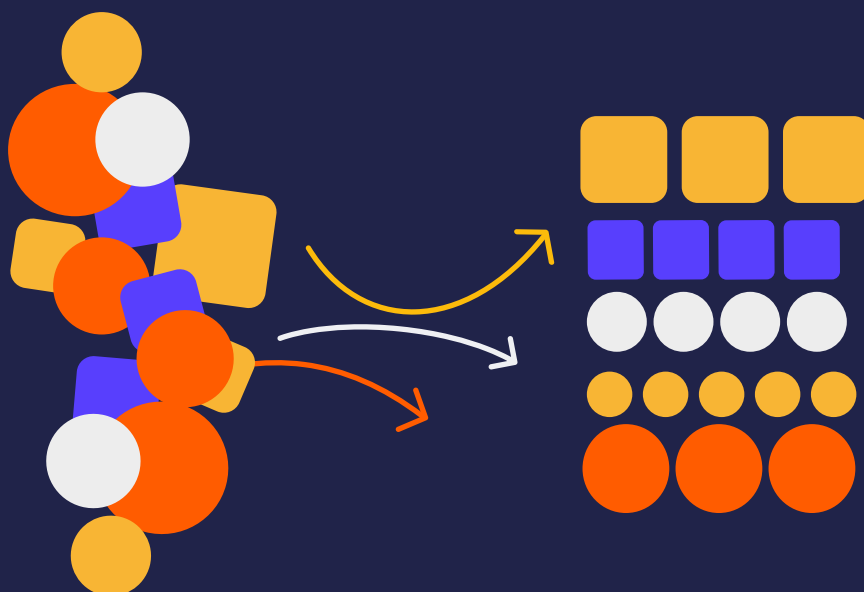


## Monitor and Tune Performance

Identifying slowdowns, bottlenecks, and performance issues is crucial to tuning performance. Big data platforms need constant monitoring for:

- **Resource utilization** to surface constraint that may impact performance
- **Network performance**, such as issues with latency and throughput
- **Data pipelines** to ensure seamless ingest, processing, and storage

Common tools include Apache Hadoop, which allows you to monitor resource allocation across clusters, and Spark UI (part of Apache Spark), for monitoring Spark jobs and performance.



# Infrastructure and Execution Engines

Managing the infrastructure requires execution engines, the software components that enable the processing and analysis of big data, including:

- **Data storage**
- **Distributed computing systems**
- **Data processing frameworks**



## Hadoop

Apache Hadoop is the most popular big data framework with a market share of nearly 19%, enabling distribution of parallel processing of massive data sets across clusters.[12]

The open-source software framework includes the Hadoop Distributed File System (HDFS), which provides a distributed storage system, and Amazon EMR, which is a programming model for distributed computing of Hadoop clusters.

Hadoop is fault-tolerant, so it can continue to function if nodes fail. Hadoop also requires users to manage dependencies between different big data frameworks and tools, which can be challenging for data teams. Performance tuning can also be time-consuming.

The native Hadoop interface is not very intuitive, so there can be a big learning curve. Also, Hadoop does not support real-time data processing, however, it can be combined with Apache Spark or Flink to support real-time processing.[13]



## Amazon EMR

Amazon EMR (Formerly called Elastic MapReduce) provides a simplified way to process data on Amazon Web Services (AWS) by breaking it down into smaller chunks and then processing these chunks in parallel across a distributed compute environment. Independent tasks can be processed across multiple nodes in a cluster without outputs being aggregated to produce final outputs.

Amazon provides the EMR File System (EMRFS) to run clusters on demand based on persistent HDFS data in Amazon S3. When the job is done, users can terminate the cluster and store the data in Amazon S3, paying only for the actual time the cluster was running.[14]

Like other platforms, EMR can be challenging to manage resource allocation, performance tuning, and dependency management. Another challenge is cluster sizing. EMR allows data teams to add or remove nodes from clusters as needed. This makes it scalable, but finding optimal cluster sizes for given workloads is key to efficiency.



## Spark

Apache Spark also provides a flexible platform for processing big data. Spark SQL provides an interface for working with structured data and Spark Streaming for real-time streaming data processing. Because Spark processes data in-memory, it can achieve extremely high processing speeds compared to many other data processing frameworks.

Spark is fairly resource intensive. It works by portioning data across clusters, so it requires optimal portioning to maximize processing. Finding the right partitioning strategy can be challenging as is tuning parameters for optimal value.



## Databricks

Databricks is built on top of Spark, providing an integrated workspace for data scientists, engineers, and AI/ML specialists to work collaboratively on projects. It includes tools for real-time streaming analytics, data visualization, and machine learning.

Compared to some other platforms, Databricks is generally considered easier to use, but it can also be expensive, particularly for businesses below the enterprise level. It's a managed service, which also limits control of the infrastructure which may reduce the ability to customize the platform for unique requirements.



## Kafka

Apache Kafka is a distributed event store that enables real-time data pipelines and streaming applications for HDFS. It is designed for large-scale streams with fault-tolerant stream processing, message relay, and data retention. As such, Kafka enables low-latency data processing and is efficient for real-time streams.

However, it uses a significant amount of resources, including memory and compute resources, and is not considered the best choice for complex data processing that requires machine learning or complex algorithms.



## Tez

Apache Tez is a distributed data processing framework built on top of Hadoop. An application framework for executing complex data processing, Tez dynamically generates a directed acyclic graph (DAG) in data processing steps to optimize execution.[15]

Tez provides a flexible programming model that allows data teams to customize the execution of data processing tasks. Tez is also compatible with a range of Hadoop tools, but has a steep learning curve, and is resource-intensive. There is also limited support for real-time processing as Tez is optimized for batch processing.



## Dataproc

Dataproc is a managed service offered by the Google Cloud Platform (GCP) for processing and analyzing large datasets. It supports open-source frameworks such as Hadoop and Spark, allowing users to create clusters with a few clicks.

One of the key advantages of Dataproc is that you can migrate code using Spark for on-prem deployment or other cloud providers to GCP without requiring any modification. Dataproc decouples storage and compute, unlike Hadoop itself which unifies storage based on HDFS and various compute engines on top of cluster nodes.

Dataproc can automatically create and configure cluster resources to reduce manual setup, however, tuning for optimal performance is both time-consuming and complex.



### Resource management

Hadoop clusters contain multiple nodes, and it's important to distribute workloads evenly across all the nodes. Managing resources efficiently is especially challenging in multi-tenant environments.



### Job scheduling

Job scheduling gets complex quickly. When clusters are not optimized, jobs can get backed up. Balancing priority, resource requirements, and data location and regionalization can be a challenge.



### Performance tuning:

Identifying and resolving bottlenecks, such as utilization, memory, disk I/O, and network latency requires optimization of configuration parameters, hardware resources, and network topology.



### Data management:

Data storage and retrieval mechanisms must be designed and managed efficiently to minimize data replication and egress fees.

# Resource Orchestration Tools Best Practices

Several resource orchestration tools can help in managing big data environments. Here, we'll discuss YARN, Kubernetes, and Mesos along with some considerations and best practices.

## YARN

In Hadoop's large-scale distributed operating system, YARN (Yet Another Resource Negotiator) is responsible for allocating system resources to applications running in Hadoop clusters and scheduling tasks on different node clusters. YARN enables Hadoop to process and run data for batch processing, stream processing, interactive processing, and graph processing which are stored in HDFS.

YARN separates the processing layer from the resource layer to increase efficiency, which was an issue in the early versions of Hadoop. When client machines make a query or fetch code for analysis, the resource manager allocates and manages the resources. All nodes have node managers that monitor their resource usage. The master node requests container resources from the node managers when job requests come in— the resources then go back to YARN.

YARN is a resource manager; however, it does monitor resource usage. So, if an HDFS node process takes resources on a machine allocated to YARN, this can create system issues due to overcommitted resources.

The default for most YARN versions is to tune YARN for equal containers as a minimum container size. This can waste resources. For example, many MapReduce applications are small, requiring only a small amount of RAM to run. This wastes the remaining resources that were reserved in YARN.

## Best Practices Tip

Monitor and manage the lower and upper bounds of resource requirements for each mapper and reducer of the job to set minimum/maximum allocation units.[16]

## K8s (Kubernetes)

Kubernetes is an open-source container orchestration system for automating the deployment, scaling, and management of containerized applications.[17] Kubernetes' goal is to make it simple to deploy and operate swarms of containers for applications by creating an abstraction layer on available node clusters.

Similar to managing a microservice, DevOps teams can focus on building applications rather than scaling servers. Once you define your CPU capacity and individual container requirements, the Kubernetes engine allocates and monitors resources. Kubernetes, also known as K8s, balances the incoming traffic load to containers and swarms evenly to help reduce the load from individual containers, eliminating the need for additional servers to handle this process.

Kubernetes also automatically provisions and fits containers into nodes to optimize resources. DevOps engineers must know Kubernetes, but also have a deep understanding of distributed applications, logging, and cloud computing to deploy Kubernetes efficiently.

Configuring and setting up Kubernetes can be complex and time-consuming. Custom resource definitions (CRDs) can help manage big data workloads to ensure they are managed consistently and predictably. Users will want to explore persistent volumes for data storage to store data outside of containers, especially stateful applications such as databases.

### Best Practices Tip

Leverage Kubernetes Operators to automate complex data analytics workloads and such tasks as scaling, upgrading, and monitoring. You'll also want to isolate the data analytics workload using Kubernetes namespace. This keeps data analytics workloads isolated from any other workloads running on the same cluster to prevent interference or limit performance.

Kubernetes often isn't the best choice to run Big Data. It was initially designed for stateless apps, and complex integration between the components can create challenges. Big data workloads tend to be more resource-intensive and stateful.

[Download: A Comprehensive Guide to Kubernetes Optimization](#)

## Apache Mesos

Apache Mesos is an open-source cluster manager for resource isolation and sharing across data frameworks.

Mesos operates by abstracting the entire data center into a pool of compute resources for dynamic allocation across distributed applications. This enables users to run multiple applications, frameworks, and services on the same set of resources to reduce costs and manage resource utilization. Mesos can dynamically acquire and release resources to help reduce waste while sharing commodity clusters between multiple diverse frameworks.[18]

In essence, Mesos works the opposite of virtualization. While virtualization splits a physical resource into multiple virtualization resources, Mesos join multiple physical resources into a single virtual resource.[19]



### Best Practices Tip

Configure Mesos to optimize resource utilization and make sure to run the right Mesos schedule for workloads. For big data analytics workloads, the Mesos framework for Apache Spark is a popular choice. It provides integration with Spark's cluster manager and can be used to run Spark workloads on Mesos. You can also configure Mesos to use fine-grained resource sharing, which allows multiple applications to share the same resources.

Users will also want to closely monitor resource utilization, task scheduling, and other performance closely monitor metrics to identify bottlenecks and optimize the system.



# Optimizing Big Data Workloads with Intel Granulate

Each of these frameworks, infrastructure and execution engines, and resource optimization tools has pros and cons. As you can see, however, things get complex quickly and none of them have a holistic way to manage resources effectively across an organization's on-prem and cloud resources to produce optimal results.

**Intel Granulate provides the continuous automation and big data optimization solution organizations need to manage workloads and costs efficiently by minimizing wasted resources, increasing time to completion, and accelerating pipeline throughput.**

Intel Granulate improves application performance and can cut costs by up to 45% with no code changes involved, overcoming the challenges inherent in big data with industry-leading solutions.

Intel Granulate works as the final stage in the optimization journey, working on the runtime level, to provide value add after any other solutions.



# Intel Granulate Overcomes Challenges In Optimizing Big Data Workloads



## Challenge



## Solution

### Efficiency

Optimizing big data workloads is challenging and requires a lot of resources.

Using Intel Granulate, applications run more efficiently, minimizing CPU and memory resources, reducing job completion time, and lowering costs.

### Visibility

Lack of visibility into big data workload performance harms data processing strategies because it makes it difficult to optimize the performance of data processing tasks and identify bottlenecks in the workflow.

With the gCenter dashboard, you can get a full view of your data workload performance, resource utilization, and costs.

Visibility includes CPU utilization, thread activity, memory usage, application performance, and system resource usage. This helps identify and mitigate bottlenecks that inhibit performance.

### Scalability

High volumes of workloads are hard to optimize at scale and often require manual configuration.

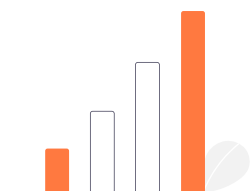
By automating the optimization process, Intel Granulate ensures that the workloads remain efficient, even when scaling rapidly.

### Dynamic

Even minor changes in workloads can hurt cluster performance and require retuning of code and configuration.

Intel Granulate autonomously and continuously optimizes data workloads despite their dynamic nature, so that data engineering teams don't have to spend their time making manual changes.

SOURCE: Intel Granulate



Throughput  
increase



CPU  
reduction



Response  
time reduction

# Intel Granulate supports all infrastructure

Whether you're running on-prem, in the cloud, or with hybrid data solutions.

Intel Granulate supports all infrastructure, including EMR, Dataproc, HDInsight, Cloudera, MapReduce, Spark, PySpark, along with the most popular data engines, resource orchestration tools, and cloud service providers.

## Execution Engines



## Platform



CLUSTERA



## Resource Orchestration



## Single, multi & hybrid cloud




With Intel Granulate, data science, data engineering, and data analytics teams can complete more jobs in less time, improving performance in a variety of ways.

**Intel Granulate's approach to Big Data optimization works on two levels at the same time.**

On the runtime level, Intel Granulate applies the most efficient crypto and compression acceleration libraries, memory arenas Profile-Guided Optimization (PGO), and JVM runtime optimizations.

Intel Granulate also tunes YARN resource allocation based on CPU and memory utilization, optimizes Spark executor dynamic allocation based on job patterns and predictive idle heuristics, and optimizes the cluster autoscaler.

With Intel Granulate, you can:



**Reduce processing costs across Spark, MapReduce, Kafka, Yarn, and more.**

**Improve performance by continuously optimizing application runtime and resource allocation.**

**Unburden your R&D team with continuous orchestration without code changes.**

**Meet your SLAs with faster job completion time.**

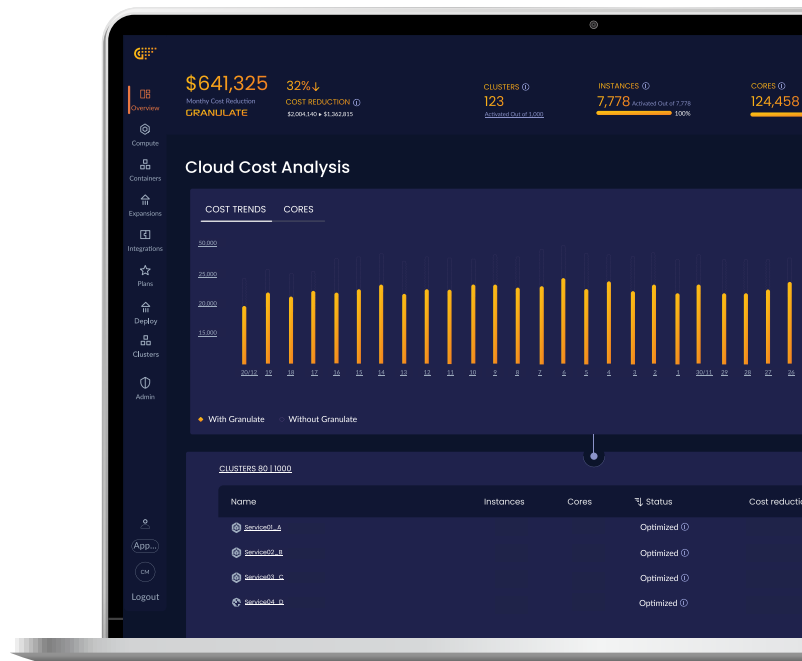
# About Intel Granulate

Intel Granulate, an Intel company, empowers enterprises and digital native businesses with real-time, continuous application performance optimization and capacity management, on any type of workload, resulting in cloud and on-prem compute cost reduction.

**Available in the AWS, GCP, Microsoft Azure and Red Hat marketplaces, the AI-driven technology operates on the runtime level to optimize workloads and capacity management automatically and continuously without the need for code alterations.**

Intel Granulate offers a suite of cloud and on-prem optimization solutions, supporting containerized architecture, big data infrastructures, such as Spark, MapReduce, and Kafka, as well as resource management tools like Kubernetes and YARN. Intel Granulate provides DevOps teams with optimization solutions for all major runtimes, such as Python, Java, Scala, and Go. Customers are seeing improvements in their job completion time, throughput, response time, and carbon footprint while realizing up to 45% cost savings.

[Book a demo](#)



## References

- [1] <https://www.fortunebusinessinsights.com/big-data-analytics-market-106179>
- [2] <https://abdalslam.com/hybrid-cloud-storage-statistics>
- [3] <https://www.canalys.com/newsroom/global-cloud-services-Q4-2022>
- [4] <https://www.infoworld.com/article/3689813/cloud-trends-2023-cost-management-surpasses-security-as-top-priority.html>
- [5] <https://granulate.io/blog/overcome-4-challenges-big-data-workload-optimization/>
- [6] <https://www.businesswire.com/news/home/20211019005858/en/Data-Engineers-Are-Burned-Out-and-Calling-for-DataOps>
- [7] <https://www.forbes.com/sites/forbesbusinessdevelopmentcouncil/2022/12/22/how-to-get-the-most-from-your-multi-cloud-strategy-four-steps-to-avoid-chaos/?sh=68b7f3674aef>
- [8] <https://venturebeat.com/data-infrastructure/report-49-of-businesses-struggle-to-control-cloud-spend/>
- [9] <https://www.infoworld.com/article/3689813/cloud-trends-2023-cost-management-surpasses-security-as-top-priority.html>
- [10] <https://www.baselinemag.com/analytics-big-data/structured-vs-unstructured-data-including-examples-of-both/>
- [11] <https://aws.amazon.com/what-is/etl/>
- [12] <https://6sense.com/tech/big-data-analytics/apache-hadoop-market-share>
- [13] <https://granulate.io/blog/hadoop-ultimate-guide-2023/#YARN>
- [14] <https://granulate.io/blog/hadoop-ultimate-guide-2023/#YARN>
- [15] <https://web.eecs.umich.edu/~mosharaf/Readings/Tez.pdf>
- [16] <http://www.openkb.info/2015/06/best-practise-for-yarn-resource.html>
- [17] <https://kubernetes.io/>
- [18] <https://people.eecs.berkeley.edu/~alig/papers/mesos.pdf>
- [19] <http://iankent.uk/blog/a-quick-introduction-to-apache-mesos/>

**intel** Granulate

Visit us at [Intel Granulate.io](https://intelgranulate.io) to learn more