Workload balancing that executes up to 30% faster with IBM DataStage on IBM Cloud Pak for Data

Authors:
Scott Brokaw
Yi Yang Ren

Executive summary

This study established that IBM® DataStage® on IBM Cloud Pak® for Data yields clear performance benefits over traditional binary deployment of IBM InfoSphere® DataStage, due to its best-in-breed parallel engine and dynamic workload balancing feature to dynamically allocate available compute resources to balance workloads. Specifically, during execution windows of resource contention, throughput was up to 30% more for containerized DataStage compared to a traditional symmetric multiprocessing (SMP) DataStage environment.

IBM Cloud Pak for Data is a fully integrated data and AI platform that modernizes how businesses ingest, manage, and understand their data to infuse AI throughout their organizations. Built on a Red Hat® OpenShift® foundation, it enables customers to easily provision and scale deployments across on-premises and any cloud infrastructure. IBM DataStage is a market-leading data integration tool that businesses rely on for mission-critical workloads. Demonstrating performance characteristics of containerized IBM DataStage on IBM Cloud Pak for Data is critical for customers to have confidence that they can continue meeting their SLAs even as they modernize their information architecture.

Introduction

Containers enable unprecedented portability of code and vastly simplify the deployment lifecycle of software applications, including install, patch, scale, and in-place updates. As a result, containerization is increasingly a critical part of modern information architecture because it enables seamless deployment of workloads across hybrid cloud environments. IBM Cloud Pak for Data is a data and AI platform architected for containerized microservices fully integrated with Red Hat OpenShift Container Platform.

As enterprises modernize their information architecture, they need to be confident that mission-critical workloads demanding intensive compute processing will scale and perform in containerized deployments on any environment. IBM conducted this study to assess data integration workload performance for IBM DataStage on IBM Cloud Pak for Data—a market-leading data integration capability that is foundational to all analytics and AI endeavors.

Red Hat OpenShift and containerization

Red Hat OpenShift Container Platform offers a leading enterprise Kubernetes platform focused on security, portability, operations simplicity, and scalability.

Key benefits include:

– Integrated platform including container host, Kubernetes, and application lifecycle management using your choice of infrastructure
– Greater value for operations and development teams throughout the application lifecycle
– Secure, validated container content and services from a wide partner ecosystem
– Faster application development cycles and more frequent software deployments
– Simple installations and upgrades, even in air-gapped environments
– Application portability with lower operational cost across hybrid cloud, multicloud, and edge footprints

IBM DataStage for IBM Cloud Pak for Data

IBM DataStage can perform enterprise-class data integration via extract, transform, and load (ETL) jobs developed by customers. In the context of DataStage, "job" refers to ETL tasks composed of operators called "stages".

Examples of stages include:

- Source and target access for databases, data warehouse appliances, data lakes, applications, and files
- General processing stages such as filter, sort, union, lookup, and aggregators
- Built in and custom transformations
- Copy, move, FTP, and other data movement stages
- Real-time XML, service-oriented architecture (SOA) and message queue processing
- Interface with big data analytics platforms, business rule management system rules, and custom Java applications

The core function set in DataStage includes Transformation, Sort, Lookup, Join, Funnel Sort, Funnel Continuous, Filter, Aggregation, DataSets, Time Function, Number of Delimiters in a String, Sort and Join, String Comparison, Modify Datatype Conversion, Complex Lookup, and Complex Aggregation.

IBM DataStage on IBM Cloud Pak for Data utilizes a best-in-breed parallel engine and a new dynamic workload management feature available starting in IBM Cloud Pak for Data 3.0.1. A key value of this feature is that it dynamically allocates available compute resources to maximize throughout and minimize resource congestion. It enables users to define a dynamic configuration file (APT_CONFIG_FILE) for job execution and easier management of compile and runtime assets. For instance, DataStage job logs are generated in a simple text file, which can be consumed by common log aggregation stacks for searching and indexing. These are key differentiators between containerized DataStage compared to traditional binary deployment.

Test objectives

This study of IBM DataStage on IBM Cloud Pak for Data performance focused on the following objectives:

- Characterize performance during execution windows of severe resource contention.
- Quantify performance differences between IBM Cloud Pak for Data using Massively Parallel Processing (MPP) and DataStage running in a symmetric multiprocessing (SMP) environment for identical workloads.
- Validate that there is no discernible difference in performance behavior of core operations/stages within IBM DataStage.
- Demonstrate performance characteristics between DataStage deployed via traditional binary versus as containerized software.

The study focused on CPU-intensive workloads to represent a typical DataStage job execution. The CPU-intensive workload was represented by job designs that include CPU-intensive operations such as 'Copy' and 'Transformer' to generate an extreme amount of CPU workload and ensure the job was bottlenecked by CPU cycles.

In addition, performance tests were also conducted on core-function workloads that represented typical core DataStage Operators such as Transformation, Sort, Lookup, Join, Funnel Sort, Funnel Continuous, Filter, Aggregation, DataSets, Time Function, Number of Delimiters in a String, Sort and Join, String Comparison, Modify Datatype Conversion, Complex Lookup, and Complex Aggregation.

Test configuration and methodology

DataStage deployed on traditional binaries utilizes a symmetric multiprocessing (SMP) environment while containerized DataStage deployed via IBM Cloud Pak for Data and Red Hat OpenShift Container Platform utilized a Massively Parallel Processing (MPP) environment. Both were deployed on x86 infrastructure.

The aggregate CPU and memory resources available to SMP and MPP environments were identical: 8 CPUs and 32GB of memory.

Traditional binary used DataStage 11.7.1.1. Containerized used DataStage on IBM Cloud Pak for Data 3.0.1 with OpenShift version 3.11 and Kubernetes 1.11.0.

This study focused principally on the impacts of severe CPU load on DataStage running in standalone and containerized environments. The test cases in this study were CPU bound to ensure proper resource utilization and comparison of performance characteristics between DataStage binaries running directly on Linux® versus containerized infrastructure.

Comparison of CPU-intensive workload

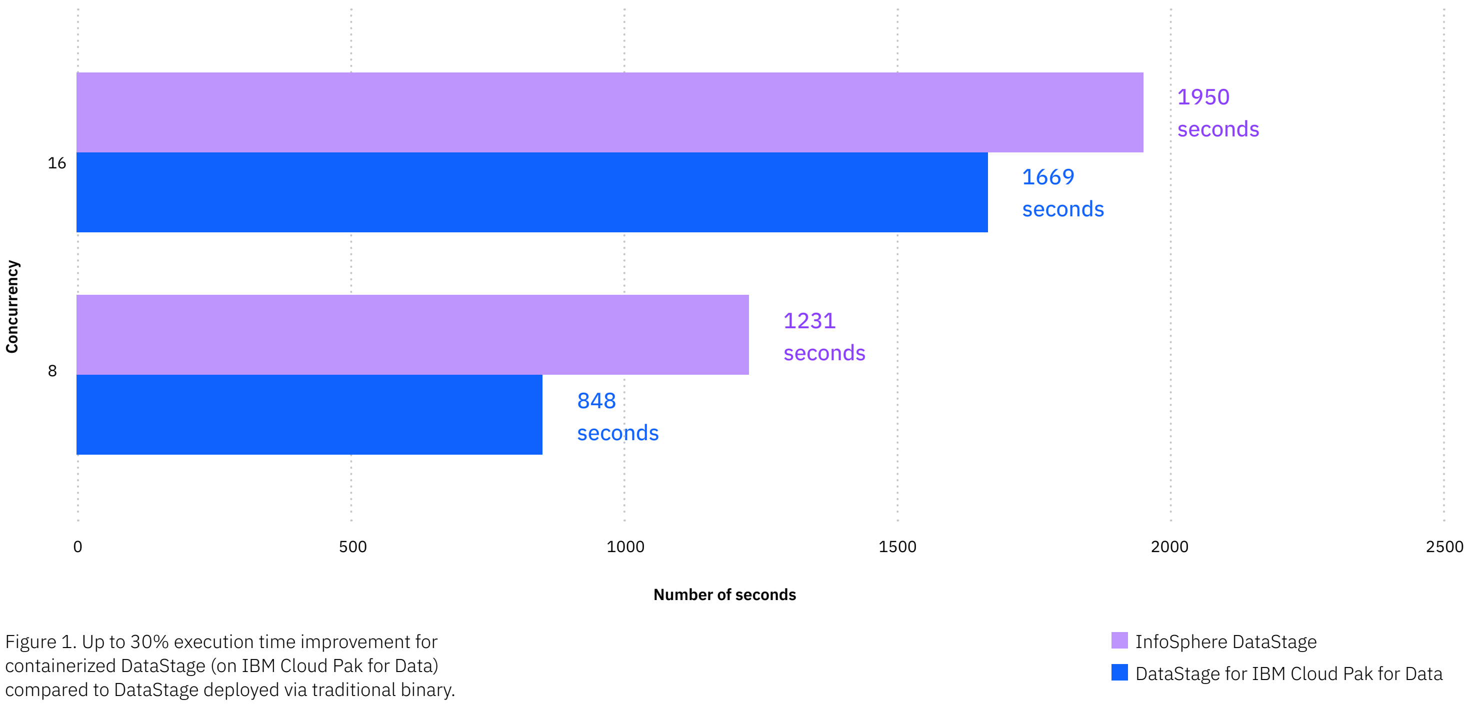

Figure 1. Up to 30% execution time improvement for containerized DataStage (on IBM Cloud Pak for Data) compared to DataStage deployed via traditional binary.

Test results

Test results for the CPU-intensive workload that represents a typical DataStage job are shown in Figure 1.

The bar graph in Figure 1 represents the total execution time of the DataStage job in seconds. The test was run with two different degrees of concurrency: 8 and 16 simultaneous job workloads. The data volume used for this test was 6.3 GB for each job instance—totaling 50.4 GB of data volume for the 8 concurrent job workload test and 100.8 GB of data volume for the 16 concurrent job workload test. In both cases, the results clearly demonstrate a performance improvement of up to 30% for running the same workloads on the containerized environment.

The study also examined core standalone functions/operations within IBM DataStage and compared the execution time in each environment. The study concluded that there was no discernible difference in performance for these standalone core functions and options between DataStage running on standalone binaries versus DataStage running in containers. Functions such as Sort, Lookup, Join, Funnel (Sort and Continuous), Filter, Aggregation, and Transformer were tested with 2.1 GB of data volume. Additional tests were done with nearly 300 GB of data volume with functions/operations such as: Dataset, Time Function, Number of Delimiters in a String, Sort and Join, String Comparison, Modify Type Conversion, Complex Lookup, and Complex Aggregation.

Conclusions and recommendations

Based on the results of this study, we conclude that data integration with DataStage in a distributed MPP containerized architecture yields clear performance benefits over SMP in traditional binary deployment. Specifically, comparison of CPU-intensive data integration workloads showed a notable improvement in execution time: up to 30% improvement in overall runtime execution when utilizing IBM DataStage on IBM Cloud Pak for Data. This performance improvement is attributed to DataStage more evenly balancing and distributing workloads on a containerized environment. DataStage achieves this balance by utilizing the Kubernetes engine within Red Hat OpenShift Container Platform for resource management of the compute pods available for DataStage job execution. DataStage leverages a dynamic runtime configuration file allowing the data engineer to specify the degree of parallelism/partitioning to run the job with, without requiring the engineer to dictate what instances the job will run on. This permits the DataStage engine to effectively schedule the workload across the available compute resources in the containerized environment.

The study also confirmed the expected behavior of core function/operation performance within DataStage. We conclude that there was no discernible performance difference in individual, isolated functions/operations within DataStage. This validates expectations and provides a data point for validation and confidence in running critical DataStage workloads in containers versus traditional standalone binaries.

This behavior also enables DataStage to elastically manage the running compute resources in the environment, scaling up or down the amount of compute pod replica resources based on workload. This feature, *available only in containerized DataStage*, allows for the preservation of hardware resources in a given Red Hat OpenShift cluster so that these resources can be used for other applications or microservices. DataStage adapts to given workloads and elastically expands the available resource footprint to accommodate fluctuations in workload assignments.[1] DataStage manages and maintains a desired state of minimum and maximum replicas for compute pods with configurable intervals that control the frequency at which DataStage will scale down the available replicas if no job workload is present. For customers that require minimal to no startup time, set the minimum and maximum replicas to be the same value to disable the elastic scaling capability but maintain the dynamic workload balancing aspects of DataStage.

Other considerations

Although this study focused on CPU utilization and performance characteristics of workloads that were CPU-bound, there are other resources to take into consideration when configuring and using DataStage in a containerized environment.

1. The first resource is network throughput speed available within the cluster. It is important to ensure that a strong network connection exists between the worker nodes in the Red Hat OpenShift cluster when DataStage is running in an MPP configuration (a core tenet of DataStage in a containerized environment) that involves re-partitioning of datasets. We recommend that you ensure network bottlenecks are eliminated from the system design, forcing the constraints instead on CPU or memory capacity, which can be scaled and further partitioned using the power of the DataStage Parallel Engine.
2. The second resource is disk performance that is available to DataStage. DataStage has two primary disk needs:
   - Runtime requirements for scratch disk and temporary storage
   - Resource disk for persisted data

Scratch disk and temporary storage needs to leverage a High Input/Output Operations Per Second (IOPS) disk. This disk can be configured with an access mode of ReadWriteOnce. This storage will most commonly be backed by a Solid State Drive (SSD) or local disk on the Red Hat OpenShift worker node that can be leveraged through the default EmptyDir volume created by the Kubernetes statefulset deployment. If additional performant storage is required, an explicit local disk can be leveraged as part of the deployment. It is important to ensure that a slow shared network disk such as NFS is not used for scratch and temporary storage. Resource disk is a shared disk utilized for persisted data such as DataSets. This disk must be configured with an access mode of ReadWriteMany, as multiple compute pod instances will need to concurrently access this disk volume object. While IOPS of this disk is important for performance of import/export file operations, it is not as critical to maintain a fast IOPS disk as it is for scratch and temporary storage where usage is pervasive across critical functions and operations.

Want to learn more about IBM DataStage on IBM Cloud Pak for Data and how deploying on containers can save your organization time and operational costs? Click here to watch the community webinar.

IBM Corporation
New Orchard Road
Armonk, NY 10504

Produced in the United States of America
November 2020

1 There is a 1-2 minute startup time of the compute pod when a new replica is required. This startup time can impact the overall runtime of the submitted job executions, but is limited to the initial container start, especially in situations where job execution workload drives elastic scaling.

O3KRGXEG