WHITE PAPER

Data Center
Data and Artificial Intelligence (AI)

Intel® Add-On Software for IBM Cloud Pak* for Data

Introduction

Competing and innovating with data requires end-to-end optimization throughout the entire data life cycle—from data ingestion at the edge, to staging and preparing data for analytics, to achieving actionable insights with artificial intelligence (AI). At every stage of the data life cycle, Intel's data-centric platform optimized for your software environments and libraries provides high-performance solutions. Intel® add-on software packages are available for IBM Cloud Pak* for Data.

What Is IBM Cloud Pak* for Data?

IBM Cloud Pak for Data is a hybrid cloud data and AI platform that delivers an information architecture for AI. IBM Cloud Pak for Data enables you to unlock the value of all your data on a unified, cloud-native platform so you can collect, organize, and analyze your data, then infuse AI to generate business insights. IBM Cloud Pak for Data runs on Red Hat OpenShift* Container Platform, combining the power of the industry's leading container platform with an open, containerized advanced analytics solution.

What Is the Deep Learning Reference Stack?

The Deep Learning Reference Stack is an integrated, high-performance open source stack optimized for Intel® Xeon® Scalable processor-based platforms. This open source stack provides AI developers with easy access to all features and functionality of Intel® platforms.

Intel® Software Packages

Intel offers a number of add-on software packages to go with IBM Cloud Pak* for Data that are already optimized for Intel® architecture, saving you up to weeks of time in configuring and tuning software for your workloads. These add-ons are free and available as containers from the IBM Cloud Pak for Data library. They offer excellent performance and can be implemented quickly on Intel architecture. Click a button on the right to learn more about each optimized add-on.

Intel® Add-On Software for Deep Learning Reference Stack

Deep Learning Reference Stack for TensorFlow

Partner | Premium

A Deep Learning Reference Stack for TensorFlow that is optimized for Intel architecture. Requires Intel® AVX-512.

Deep Learning Reference Stack for PyTorch

Partner | Premium

A Deep Learning Reference Stack for PyTorch that is optimized for Intel architecture. Requires Intel® AVX-512.

Other Intel Add-On Software for Analytics

Analytics Zoo for Apache Spark

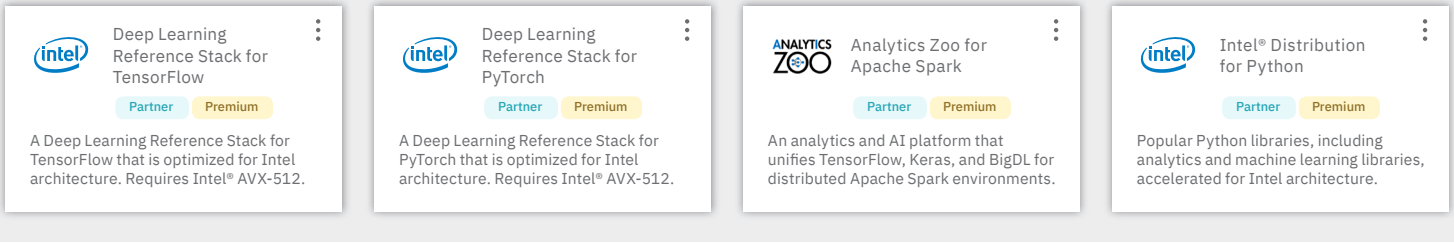
Partner | Premium

An analytics and AI platform that unifies TensorFlow, Keras, and BigDL for distributed Apache Spark environments.

Intel® Distribution for Python

Partner | Premium

Popular Python libraries, including analytics and machine learning libraries, accelerated for Intel architecture.



The Deep Learning Reference Stack is highly tuned and built for cloud-native environments. The stack includes highly optimized software components across the operating system (Clear Linux* OS), deep-learning frameworks (TensorFlow*, PyTorch*), deep-learning libraries (Intel® Math Kernel Library for Deep Neural Networks (Intel® MKL-DNN)), and other software components. It enables AI developers to quickly prototype by helping to reduce complexity associated with integrating multiple software components, while still giving them the flexibility to customize their solutions. The Deep Learning Reference Stack is integrated with Kubernetes* and Kubeflow* and can be used in either a single or multinode architecture, providing options for development and deployment of deep-learning workloads (see Figure 1).

This containerized software stack is easy to install and run and can potentially save weeks of time in adjusting and customizing the software. Depending on your needs, you can use the Deep Learning Reference Stack for TensorFlow* and/or the Deep Learning Reference Stack for PyTorch*, which are already optimized for your Intel hardware. They are free of charge with your IBM Cloud Pak for Data platform.

- **TensorFlow** is the leading machine-learning and deep-learning framework for Python*.
- **PyTorch** is a machine-learning library for Python.

Both of these are available as individual add-ons in the Analytics category of the IBM Cloud Pak for Data library.

Key Benefits of the Deep Learning Reference Stack

The Deep Learning Reference Stack offers many benefits to help you deliver highly optimized, tuned, and well-performing AI applications:

- Performance of data science operations can be improved with container images tuned for hardware acceleration from Intel® architecture.
- Tailored images are optimized and tested together for various use cases.
- Reduced complexity associated with software components allows for quick prototyping.
- Supports customized solutions.
- Deployment is easy from Docker Hub*.
- Supports flexible deployment models for cloud service providers and on-premises installations.

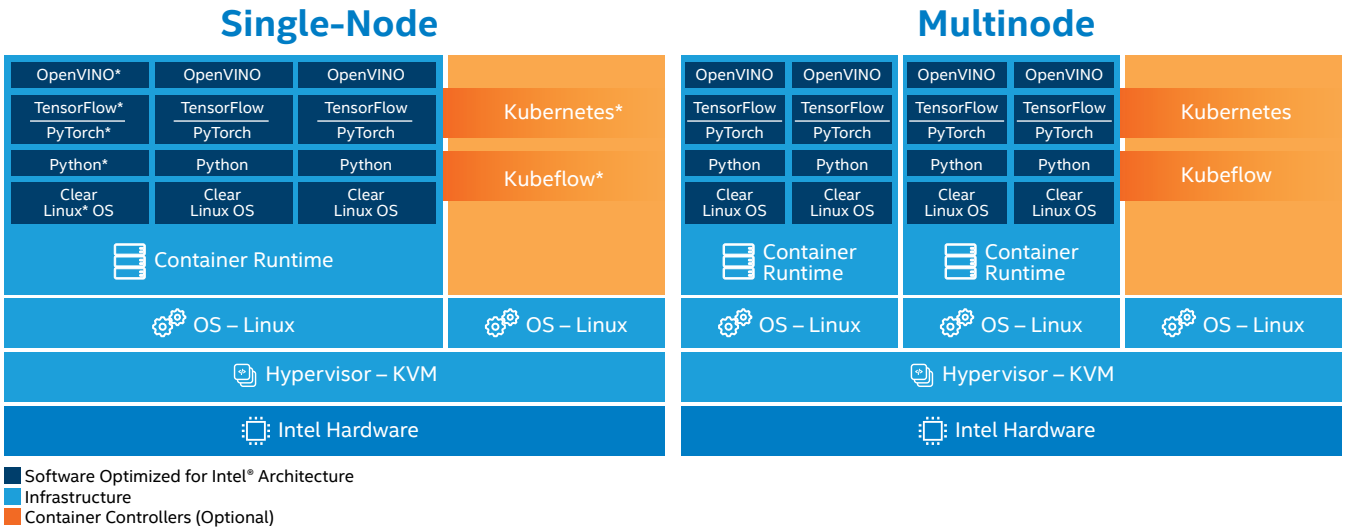
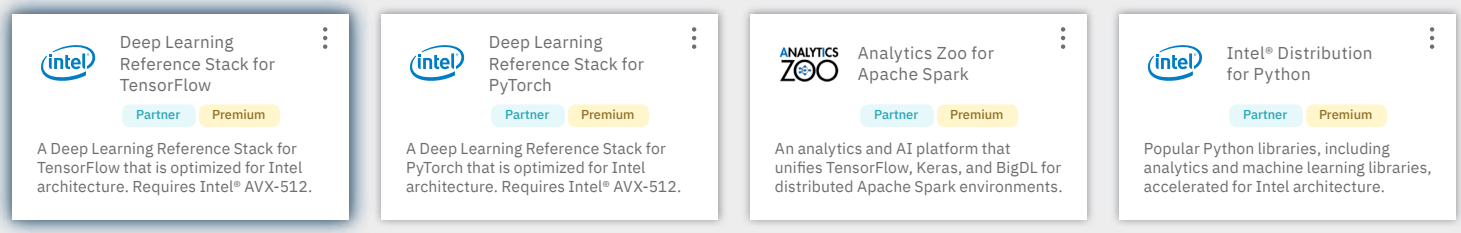


Figure 1. The Deep Learning Reference Stack can be used in either a single-node or multinode architecture.



Deep Learning Reference Stack for TensorFlow*

The Deep Learning Reference Stack for TensorFlow* can be found in the Analytics category

TensorFlow is a popular machine-learning framework for deep learning, and it demands efficient utilization of computational resources. In order to take full advantage of Intel architecture and to achieve maximum performance, the TensorFlow framework has been optimized using Intel MKL-DNN primitives. Other optimizations include:

- Parallelization within a given layer or operation
- Parallelization across layers
- Balanced use of prefetching
- Cache blocking techniques
- Data formats that promote spatial and temporal locality

The Deep Learning Reference Stack for TensorFlow, running on a 2nd Generation Intel Xeon Scalable processor-based platform, produces significantly more inference throughput[1] than the non-optimized version. Throughput gains for the Deep Learning Reference Stack with the ResNet50 model are shown in Figure 2:

- With a batch size of 128, optimized TensorFlow produces 11.70x more throughput than non-optimized TensorFlow.
- With a batch size of 64, optimized TensorFlow produces 11.77x more throughput than non-optimized TensorFlow.
- With a batch size of 32, optimized TensorFlow produces 11.89x more throughput than non-optimized TensorFlow.

The performance gains are even more substantial with the more complex Inceptionv4 model within the Deep Learning Reference Stack running on a 2nd Generation Intel Xeon Scalable processor-based platform, as shown in Figure 3:[2]

- With a batch size of 128, optimized TensorFlow produces 12.66x more throughput than non-optimized TensorFlow.
- With a batch size of 64, optimized TensorFlow produces 12.96x more throughput than non-optimized TensorFlow.
- With a batch size of 32, optimized TensorFlow produces 12.72x more throughput than non-optimized TensorFlow.

ResNet50 Performance Comparison

Deep Learning Reference Stack for TensorFlow*

HIGHER IS BETTER

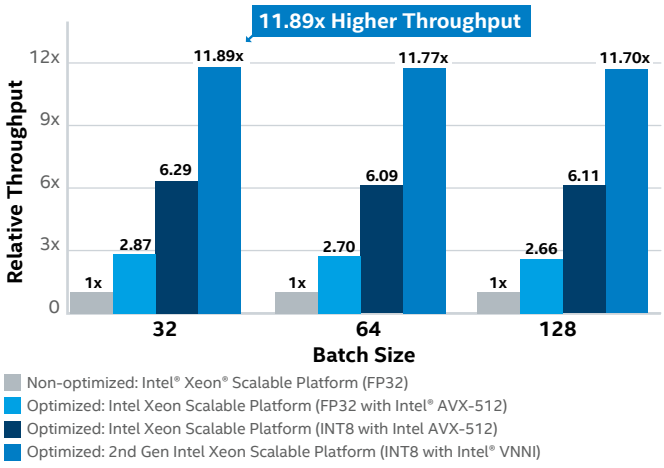


Figure 2. ResNet50 model inference produces up to 11.89x more throughput with the Deep Learning Reference Stack for TensorFlow* running on a 2nd Generation Intel® Xeon® Scalable processor-based platform, compared to a non-optimized platform.[1]

Inceptionv4 Performance Comparison

Deep Learning Reference Stack for TensorFlow*

HIGHER IS BETTER

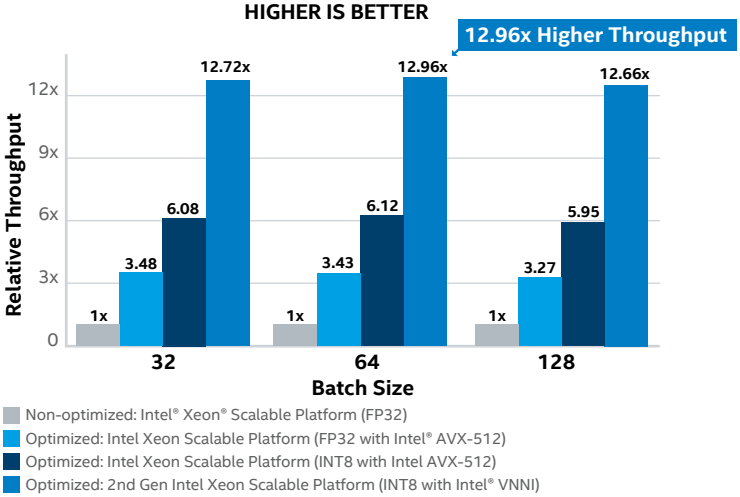
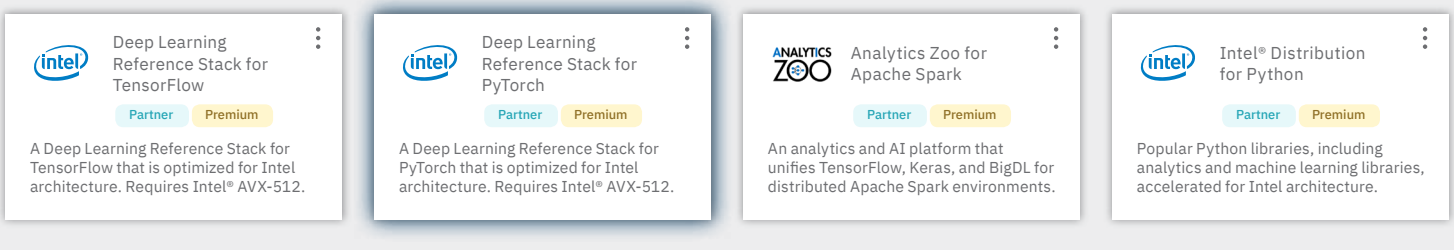


Figure 3. Inceptionv4 model inference can run up to 12.96x faster with the Deep Learning Reference Stack for TensorFlow* running on a 2nd Generation Intel® Xeon® Scalable processor-based platform, compared to a non-optimized platform.[2]



Deep Learning Reference Stack for PyTorch*

The Deep Learning Reference Stack for PyTorch* can be found in the Analytics category

PyTorch is a Torch*-based open source machine-learning library for Python that developers can use across deep-learning applications such as natural language processing. Developers use this scientific computing package as a replacement for NumPy*.

Intel Xeon Scalable processors feature Intel® Advanced Vector Extensions 512 (Intel® AVX-512), which benefits existing deep-learning applications because Intel MKL-DNN optimizations are abstracted and integrated directly into the PyTorch framework. Developers can take advantage of this technology with minimum changes to their code.

The Deep Learning Reference Stack for PyTorch runs significantly faster[3] on an Intel Xeon Scalable processor-based platform than the non-optimized version of PyTorch (see Figure 4).

- With a batch size of 128, optimized PyTorch runs 47.72x faster than non-optimized PyTorch.
- With a batch size of 64, optimized PyTorch runs 47.34x faster than non-optimized PyTorch.
- With a batch size of 32, optimized PyTorch runs 46.12x faster than non-optimized PyTorch.

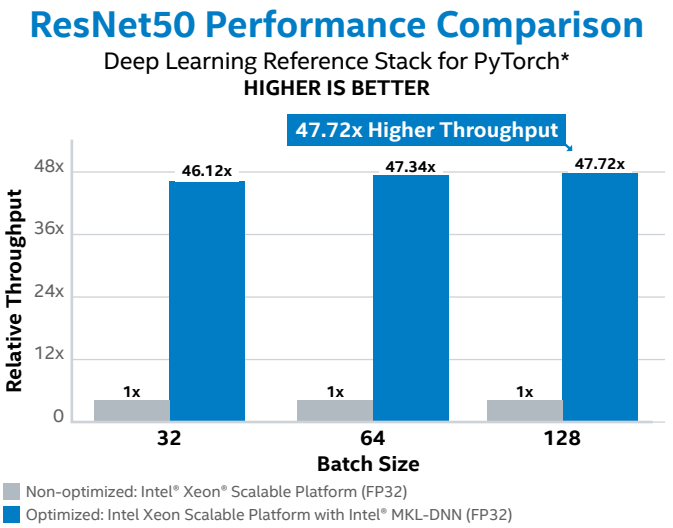
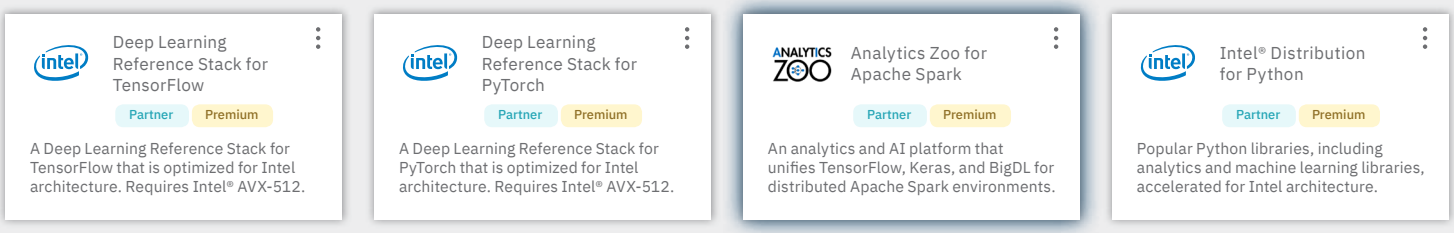


Figure 4. Inference performance of the Deep Learning Reference Stack for PyTorch* is up to 47.72x faster than the non-optimized version of PyTorch running on an Intel® Xeon® Scalable processor-based platform.[3]



Analytics Zoo for Apache Spark*

The Analytics Zoo for Apache Spark* can be found in the Analytics category

Analytics Zoo is an open source distributed deep-learning library for Apache Spark*. It has direct access to stored data and provides unified analytics in addition to an AI platform that seamlessly unites Spark, TensorFlow, Keras*, and BigDL into an integrated pipeline. The entire pipeline can then transparently scale out to an existing Hadoop*/Spark cluster where the data is stored for distributed training or inference.

Analytics Zoo includes:

- Deep-learning model development using TensorFlow or Keras.
- Distributed training/inference on Spark and BigDL.
- The ability to execute deep-learning applications dynamically while sharing existing Hadoop/Spark clusters with other workloads.
- Data wrangling and analysis using PySpark*.

All of this takes place within a single unified pipeline and in a user-transparent fashion.

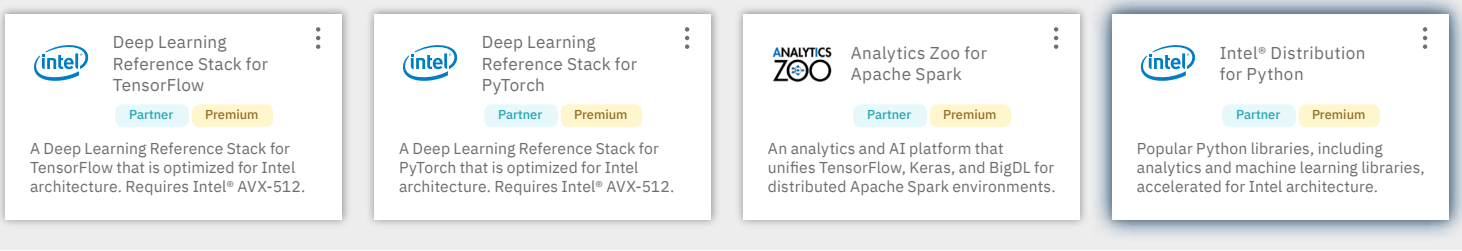
In addition, Analytics Zoo also provides a rich set of analytics and AI support for the end-to-end pipeline, including:

- Easy-to-use abstractions and APIs (for example, transfer learning support, autograd operations, Spark DataFrame and machine-learning pipeline support, and online model serving)
- Common feature engineering operations (for example, image, text, and 3D images)
- Built-in deep-learning models (for example, object detection, image classification, text classification, recommendation, anomaly detection, text matching, and sequence-to-sequence)
- A range of use cases (for example, anomaly detection, sentiment analysis, fraud detection, and image similarity)

Analytic Zoo for Apache Spark is optimized for 2nd Generation Intel Xeon Scalable processors, and is a great add-on for your existing Apache Spark infrastructure. This add-on is available in the Analytics category of IBM Cloud Pak for Data add-ons.

For more information about Analytics Zoo, see github.com/intel-analytics/analytics-zoo.

Seamlessly scale your AI models to big data clusters with thousands of nodes for distributed training or inference. Built on top of the open source platform of Apache Spark*—leveraging TensorFlow*, Keras, and BigDL—this unified analytics and AI platform has an extensible architecture to support more libraries and frameworks.



Intel® Distribution for Python*

The Intel® Distribution for Python* can be found in the Analytics category

This downloadable add-on is a set of Python data science packages that take advantage of 2nd Generation Intel Xeon Scalable processors. These easy-to-use, high-performance Python packages include prebuilt accelerated solutions for data analytics and drop-in replacements for your existing Python with little or no code changes required.

Intel® Distribution for Python* provides:

- Accelerated NumPy and SciPy* routines that use Intel MKL and enhanced thread scheduling with Intel® Threading Building Blocks (Intel® TBB).
- Accelerated scikit-learn* and daal4py routines that use Intel® Data Analytics Acceleration Library (Intel® DAAL) and Intel® Message Passing Interface Library (Intel® MPI Library).
- Access to Priority Support—a direct connection to Intel engineers for technical questions.

Intel's libraries, tools, and runtimes help accelerate the entire analytics process from preprocessing through machine learning and scale out.

Figure 5 shows that Intel® optimizations help improve Python linear algebra efficiency compared to the non-optimized version for Python, approaching native C code performance efficiency on Intel Xeon Scalable processors.[4]

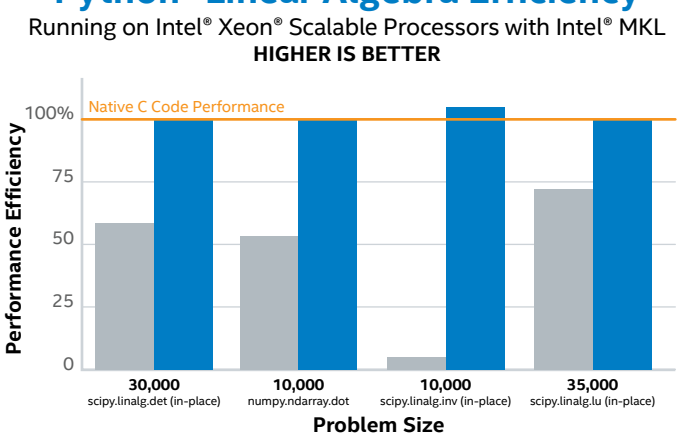
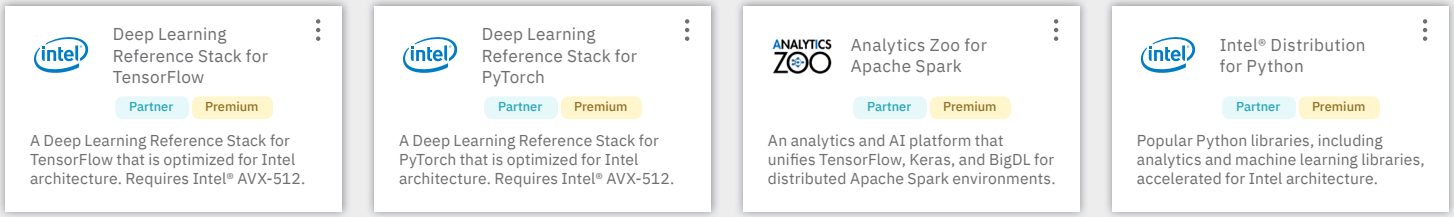


Figure 5. Optimizations for Intel® architecture improve Python* linear algebra efficiency to nearly native C code speeds when running on Intel® Xeon® Scalable processors.[4]

The Intel® Distribution for Python* is a ready-to-use, integrated package that delivers faster application performance on Intel® platforms.



Conclusion

Intel architecture-optimized add-ons for IBM Cloud Pak for Data help you to customize your system and take advantage of features that are built into your Intel Xeon Scalable processors, using free tools built specifically for them. These tools run faster than non-optimized versions and can save you weeks of configuration tasks.

The following Intel architecture-optimized add-ons are available for IBM Cloud Pak for Data:

- Deep Learning Reference Stack for TensorFlow
- Deep Learning Reference Stack for PyTorch
- Analytics Zoo for Apache Spark
- Intel® Distribution for Python

Find the solution that is right for your organization. Contact your Intel representative or visit ibm.com/analytics/products.

[1] **Workload: ResNet50.** Performance results are based on Intel testing as of April 29, 2019 and may not reflect all publicly available security updates.

2nd Generation Intel® Xeon® Scalable Platform: 2x Intel® Xeon® Platinum 8280 Processor (2.7 GHz, 28-core), HT On, Turbo On, 384 GB memory (12 x 32 GB DDR4 @ 2933 MHz), 7 TB NVMe SSD SSDPE2KE076T8, Clear Linux 30700, BIOS SE5C620.86B.0D.01.0271.120720180605, ucode (0x4000013), Linux 4.19.65-69.LTS 2018, OpenVINO™ 2019_R1.1, AIXPRT CP2 (Community Preview), Benchmark: principledtechnologies.com/benchmarkxprt/aixprt, Workload: ResNet-50, Compiler: GCC v9.1.1, Intel® MKL-DNN v0.19.

Intel® Xeon® Scalable Platform: 2x Intel® Xeon® Platinum 8180 Processor (2.5 GHz, 28-core), HT On, Turbo On, 384 GB memory (12 x 32 GB DDR4 @ 2633 MHz), 1 TB NVMe SSD SSDPE2KX010T7, Clear Linux 30700, BIOS SE5C620.86B.02.01.0008.031920191559, ucode (0x200005e), Linux 4.19.65-69.LTS 2018, OpenVINO™ 2019_R1.1, AIXPRT CP2 (Community Preview), Benchmark: principledtechnologies.com/benchmarkxprt/aixprt, Workload: ResNet-50, Compiler: GCC v9.1.1, Intel® MKL DNN v0.19.

[2] **Workload: Inceptionv4.** Performance results are based on Intel testing as of April 29, 2019 and may not reflect all publicly available security updates.

2nd Generation Intel® Xeon® Scalable Platform: 2x Intel® Xeon® Platinum 8280 Processor (2.7 GHz, 28-core), HT On, Turbo On, 384 GB memory (12 x 32 GB DDR4 @ 2933 MHz), 7 TB NVMe SSD SSDPE2KE076T8, Clear Linux 30700, BIOS SE5C620.86B.0D.01.0271.120720180605, ucode (0x4000013), Linux 4.19.65-69.LTS 2018, OpenVINO™ 2019_R1.1, AIXPRT CP2 (Community Preview), Benchmark: principledtechnologies.com/benchmarkxprt/aixprt, Workload: ResNet-50, Compiler: GCC v9.1.1, Intel® MKL-DNN v0.19.

Intel® Xeon® Scalable Platform: 2x Intel® Xeon® Platinum 8180 Processor (2.5 GHz, 28-core), HT On, Turbo On, 384 GB memory (12 x 32 GB DDR4 @ 2633 MHz), 1 TB NVMe SSD SSDPE2KX010T7, Clear Linux 30700, BIOS SE5C620.86B.02.01.0008.031920191559, ucode (0x200005e), Linux 4.19.65-69.LTS 2018, OpenVINO™ 2019_R1.1, AIXPRT CP2 (Community Preview), Benchmark: principledtechnologies.com/benchmarkxprt/aixprt, Workload: ResNet-50, Compiler: GCC v9.1.1, Intel® MKL-DNN v0.19.

[3] **Workload: ResNet50.** Performance results are based on Intel testing as of April 29, 2019 and may not reflect all publicly available security updates. 2x Intel® Xeon® Gold 8168 processor (2.7 GHz, 24-core), 192 GB memory (12 x 16 GB DDR4 @ 2666 MHz), 1 TB NVMe SSD SSDPE2KX010T701, Clear Linux* 27910.

[4] Performance results are based on Intel testing as of July 9, 2019 and may not reflect all publicly available security updates.

Non-Optimized: Python 3.6.6 hc3d631a_0 installed from conda, numpy 1.15, numba 0.39.0, llvmlite 0.24.0, scipy 1.1.0, scikit-learn 0.19.2 installed from pip.

Optimized: Intel® Distribution for Python* 2019 Gold: Python 3.6.5 Intel_11, numpy 1.14.3 intel_py36_5, mkl 2019.0 intel_101, mkl_fft 1.0.2 intel_np114py36_6, mkl_random 1.0.1 intel_np114py36_6, numba 0.39.0 intel_np114py36_0, llvmlite 0.24.0 intel_py36_0, scipy 1.1.0 intel_np114py36_6, scikit-learn 0.19.1 intel_np114py36_35; OS: CentOS* Linux* 7.3.1611, kernel 3.10.0-514.el7.x86_64; Hardware: Intel® Xeon® Gold 6140 processor @ 2.30 GHz (2 sockets, 18 cores/socket, HT OFF), 256 GB DDR4 RAM, 16 DIMMs of 16 GB @ 2666 MHz.