

Boost Your Python Performance by up to 37x with the Intel® Distribution for Python^{*,2}

The Intel® Distribution for Python* is optimized for Intel® Xeon® W processors to enhance Python performance, particularly for data science and numerical computing using NumPy and SciPy libraries.



Performance Benefits

Linear Algebra Optimizations

Intel® Distribution for Python* is engineered to maximize the performance of linear algebra operations, which are the backbone of numerical computing and data science. Operations such as matrix multiplication, determinant calculation, inversion, and factorization (including Cholesky, LU, QR, SVD), as well as solving eigenvalue problems, are optimized using Intel® oneAPI Math Kernel Library (oneMKL), fine-tuned for Intel® Xeon® processors, which offer higher core counts, larger caches, and support for advanced vector instructions like AVX-512. This results in significantly faster computations compared to stock Python distributions.^{1,2} While Intel® Core™ processors can also benefit from the Intel® Distribution for Python*, they have fewer cores, less cache memory, and lack advanced vector instructions like AVX-512.

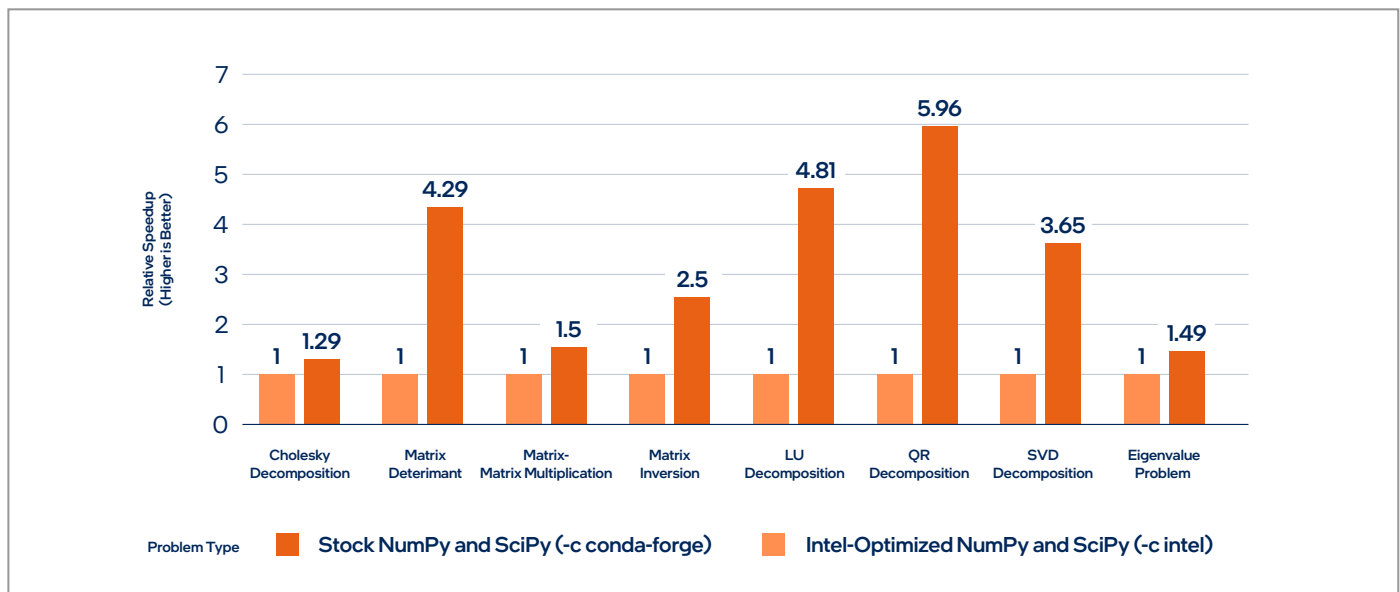


Figure 1. Intel-Optimized NumPy & SciPy Linear Algebra Performance¹

Fourier Transformations (FFT) Optimizations

FFT is a critical component in signal processing, image analysis, and solving partial differential equations. The Intel® Distribution for Python* optimizes 1D, 2D, and 3D FFTs by up to 37x using oneMKL, which is designed to take full advantage of the parallel processing capabilities of Intel® Xeon® processors.

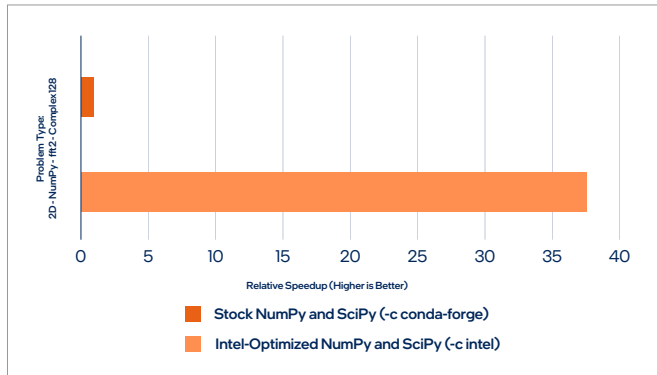


Figure 2. Intel-Optimized NumPy & SciPy Fast Fourier Transformation (FFT) Performance²

Why Intel® Distribution for Python* Matters

The optimizations provided by the Intel® Distribution for Python* are crucial for data scientists and researchers who require the utmost computational efficiency for their high-performance computing tasks. By leveraging the advanced features of Intel® Xeon® processors, users can achieve faster computation times, as evidenced in the charts above, leading to quicker insights and the ability to tackle larger, more complex problems.

Conclusion

The Intel® Distribution for Python* is the optimal choice for data science and numerical computing workloads running on Intel® Xeon® processors. With its many extensions and optimizations for core scaling, AVX-512, linear algebra, FFTs, and composable parallelism, it greatly outperforms stock Python distributions.^{1,2} Users leveraging Intel® Xeon® processors can expect significant performance gains, enabling them to accelerate their research and development in data science and AI.



How Do I Get Started?

Download the Intel® Distribution for Python* [here](#). Learn more about the packages included inside the [Intel® Distribution for Python*](#). To learn more about AI and Machine Learning development tools, check out [Intel's resources](#) to help you prepare, build, deploy, and scale your AI solutions.

- **Start with NumPy:** For most data preparation tasks involving numerical data, NumPy is the go-to library due to its versatility and built-in functionality. Additionally, the Data Parallel Extension for NumPy can leverage multiple cores.
- **Consider SciPy for Specialized Needs:** If you require advanced mathematical functions, optimization techniques, or specific scientific domain tools not readily available in NumPy, then SciPy is the right choice.

Need a workstation based on Intel® Xeon® W processors?

- [Dell Precision workstations:](#) Shop mobile, rack, and tower workstations from Dell. See the top workstation deals today.
- [Lenovo ThinkStation workstations:](#) Lenovo P-series workstations offer professional power and ultimate versatility.
- [HP Z workstations:](#) From mini to full tower, HP Z workstations deliver professional-grade performance.

¹ Performance results are based on testing by Intel as of June 13th, 2024 and may not reflect all publicly available updates. Configuration details and Workload Setup: Reference Validation Platform (FHFDCRB1.86B.0056.D23.2403040627) Processor: Intel® Xeon® w9-3595X processor (60 core), 385W, Total Memory: 256GB (8x32GB) DDR5-4800 Mt/s, Storage: Samsung 980 Pro 1TB PCIe Gen 5.0, GPU: Nvidia RTX A4000, Test: NumPy/SciPy – iBench for Python 3.9.19 (Used for comparing Standard Python 3.9.19 vs Intel® Distribution of Python* 3.9.19)

² Performance results are based on testing by Intel as of June 13th, 2024, and may not reflect all publicly available updates. Configuration details and Workload Setup: Reference Validation Platform (FHFDCRB1.86B.0056.D23.2403040627) Processor: Intel® Xeon® w9-3595X processor (60 Core), 385W, Total Memory: 256GB (8x32GB) DDR5-4800 Mt/s, Storage: Samsung 980 Pro 1TB PCIe Gen 5.0, GPU: Nvidia RTX A4000, Test: NumPy/SciPy – iBench for Python 3.9.19.

Performance varies by use, configuration, and other factors. Learn more at www.intel.com/PerformanceIndex.

Performance results are based on testing as of dates shown in configurations and may not reflect all publicly available updates. See backup for configuration details. No product or component can be absolutely secure.

Your costs and results may vary.

Intel technologies may require enabled hardware, software, or service activation.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.