

Why NumPy is Faster than an array list?

 [Try to read now!](#)

NumPy is often faster than regular Python lists(arrays) for some reasons.

Here are six reasons why numPy is faster:

1. Contiguous memory allocation

NumPy arrays have a fixed size and a homogeneous data type, which allows them to be stored in a contiguous block of memory. This allows for efficient memory access and enables fast element-wise operations.

2. Vectorized operations ☆☆

NumPy provides a wide range of vectorized operations that can be applied to entire arrays instead of operating on individual elements. These operations are implemented in optimized **C** programming code, making them faster than writing equivalent operations in pure Python.

3. Optimized C implementation

The core functionality of NumPy is implemented in highly optimized **C programming** code, which takes advantage of low-level optimizations and CPU-specific instructions. This results in faster execution compared to Python's interpreted nature.

4. Caching and locality of reference

Due to the contiguous memory layout, NumPy arrays can make better use of the CPU cache, resulting in improved memory access patterns and reduced cache misses. This can have a significant impact on performance, especially for large arrays

5. Efficient indexing and slicing

NumPy provides powerful indexing and slicing capabilities, allowing for efficient extraction of subsets of data without the need for explicit loops. This can lead to significant performance gains compared to manually iterating over elements in a Python list.

6. Functionality: - **SciPy** and **NumPy** have optimized functions such as linear algebra operations built in.

Conclusion

Overall, the combination of efficient memory handling, vectorized operations, and optimized C implementation makes NumPy faster than Python lists for numerical computations and array manipulations. However, it's important to note that for small datasets or non-numeric operations, the performance difference between NumPy and Python lists may not be significant.

Infinite fun with numPy!

epythonlab