

Institute for Computer Science



Frederik Hennecke

Impact of GIL-less Cpython

Table of contents

- 1 Introduction
- 2 Compatibility Insights
- 3 Performance Benchmarks
- 4 Analysis
- 5 Challenges and Next Steps

What's New In Python 3.13

JIT compiler

- Clearer error messages
- Minor upgrades to Python's static type system
- Improvements made to the interactive interpreter (REPL)
- Experimental support for free-threaded mode (PEP 703)

Recap: Why GIL-less CPython Matters

- Definition of GIL: Mutex in CPython that limits Python to execute one thread at a time.
- Why it exists: Simplifies memory management and avoids race conditions in CPython's reference counting.
- Prevents multi-threaded performance gains in CPU-bound workloads.
- Threads benefit mostly from I/O-bound tasks (e.g., file operations, web requests).

Porting Python Packages to Support Free-Threading

(For C, C++, Cython, Rust)

- Declare compatibility
- Ensure thread safety
- Test thoroughly
- Handle global state
- Verify dependencies
- In pure Python: Mostly, no changes are required.

Declaring NoGIL Compatibility

How to explicitly declare your library supports NoGIL:

- C/C++: Use Py_mod_gil or PyUnstable_Module_SetGIL.
- pybind11: Add gil_not_used .
- Cython: Set freethreading_compatible=True.
- Rust (PyO3): Use gil_used = false .

Compatibility List

- Python 3.13 Readiness
- Shows Python 3.13 support for the 360 most downloaded packages on PyPI
- 129 green packages (35.8%) support Python 3.13;
- 231 uncolored packages (64.2%) don't explicitly support Python 3.13 yet.

Performance Benchmarks: Setup

Four benchmark categories:

- ▶ 3.12.8 : Newest Python version before 3.13
- ► 3.13.1-gil : Compiled with the old threading model (GIL)
- ▶ 3.13.1gil_enabled : Compiled with new threading model, GIL enabled
- ► 3.13.1gil_disabled : Compiled with new threading model, GIL disabled

1,2,4,8 threads

Tests show the average value of 5 runs each.

Introduction	Compatibility Insights	Performance Benchmarks	Analysis	Challenges and Next Steps
00	000	000000000000000000000000000000000000000	00	000

Performance Benchmarks: Numpy Matrix Multiplication



Introduction	Compatibility Insights	Performance Benchmarks	Analysis	Challenges and Next Steps
00	000	00000000000000	00	000

Performance Benchmarks: Numpy Matrix Multiplication



Performance Boost for matrix-np

Introduction	Compatibility Insights	Performance Benchmarks	Analysis	Challenges and Next Steps
00	000	000000000000000000000000000000000000000	00	000

Performance Benchmarks: Pure Python Matrix Multiplication



Introduction Compatibility Insights OCO	Analysis 00	Challenges and Next Steps
---	----------------	---------------------------

Performance Benchmarks: Pure Python Matrix Multiplication



Performance Boost for matrix-inline





Introduction 00	Compatibility Insights	Performance Benchmarks	Analysis 00	Challenges and Next Steps

Performance Benchmarks: Pandas (Filter, Mean, Merge, Lambda)



Introduction 00	Compatibility Insights	Performance Benchmarks	Analysis 00	Challenges and Next Steps

Performance Benchmarks: Strings



Introduction Compatibility Insights Performance Benchmarks Analysis 00 000 000000000000000000000000000000000000	Challenges and Next Steps
---	---------------------------

Performance Benchmarks: Strings



Introduction 00	Compatibility Insights	Performance Benchmarks ○○○○○○○○●○○○○○○	Analysis 00	Challenges and Next Steps

Performance Benchmarks: Compress, Decompress



Introduction 00	Compatibility Insights	Performance Benchmarks	Analysis 00	Challenges and Next Steps

Performance Benchmarks: Compress, Decompress



Introduction 00	Compatibility Insights	Performance Benchmarks	Analysis 00	Challenges and Next Steps

Performance Benchmarks: Dictionary (Intersection, Union)



Introduction 00	Compatibility Insights	Performance Benchmarks	Analysis 00	Challenges and Next Steps $\circ \circ \circ$

Performance Benchmarks: Dictionary (Intersection, Union)



Introduction 00	Compatibility Insights	Performance Benchmarks	Analysis 00	Challenges and Next Steps

Performance Benchmarks: I/O



Introduction 00	Compatibility Insights	Performance Benchmarks	Analysis 00	Challenges and Next Steps

Performance Benchmarks: I/O



Introduction	Compatibility Insights	Performance Benchmarks ○○○○○○○○○○○○○●○	Analysis 00	Challenges and Next Steps

Performance Benchmarks: PyTorch



Introduction 00	Compatibility Insights	Performance Benchmarks ○○○○○○○○○○○○●	Analysis 00	Challenges and Next Steps

Performance Benchmarks: PyTorch



Performance Benchmarks

(GIL) Worse Performance for:

- Numpy
- String operations (not in the graphs)
- GIL) Better Performance for:
 - Pure Python Matrix Multiplication
 - Pandas
 - Compress, Decompress
 - Dictionary (Intersection, Union)

Introduction	Compatibility Insights	Performance Benchmarks	Analysis	Challenges and Next Steps
00	000	0000000000000000	0	000



Introduction 00	Compatibility Insights	Performance Benchmarks	Analysis ○●	Challenges and Next Steps



▶ Better scalability when working with large data operations.



▶ Better scalability when working with large data operations.

Challenges:

Benefits:

▶ Better scalability when working with large data operations.

Challenges:

▶ Worse performance for the new interpreter with GIL enabled.

Benefits:

▶ Better scalability when working with large data operations.

Challenges:

- ▶ Worse performance for the new interpreter with GIL enabled.
- Increased synchronization costs for some workloads.

Challenges and Next Steps

- More tests on communication overhead.
- Benchmark Python 3.14 to look for further improvements (especially the bug fix for string operations)

Introduction

Conclusion

Key Takeaways:

- GIL-less Python delivers performance gains for some pure-python workloads.
- Libraries like Numpy do not get any noticeable performance improvements.
- Compatibility issues are manageable but require ecosystem-wide cooperation.

Introduction	Compatibility Insights	Performance Benchmarks	Analysis
00	000	0000000000000000	00

Challenges and Next Steps ○○●

References

Porting Python Packages to Support Free-Threading. 2024. URL: https://py-free-threading.github.io/porting/. Python 3.13 Readiness. 2025. URL: https://pyreadiness.org/3.13/. What's New In Python 3.13. 2025. URL: https://docs.python.org/3/whatsnew/3.13.html.