

GIL - Co to jest i jak z tym walczyć

Marek Piechula

Clearcode

Python Developer

marpiechula@gmail.com

Plan



1. Czym jest GIL
2. Wady / Zalety GIL
3. Alternatywy
 - 3.1 Procesy
 - 3.2 Asyncio / greenlet
 - 3.3 Jython / Iron python
 - 3.4 Pypy-stm
 - 3.5 Cython

Czym jest GIL



GIL (global interpreter lock) zabezpiecza nasz pythonowy kod
Przed hazardem lub/oraz utratą danych.
Zadaniem GIL-a jest też przełączanie wątków
(szczególnie gdy uruchomiony wątek czeka na sygnał).

Wady / zalety GIL

- + Chroni nasz kod bez hazardem danych
- + Przełącza wątkami w razie blokady
- Nie działa z wielordzeniowymi procesorami

Przykład!

Alternatywy

- Procsy
- Asyncio / greenlet
- Jython / Iron python
- Pypy-stm
- Cython

Procesy

- + Zamiast wątku – niezależny proces którym zarządza jądro systemu
- Następuje fork – cała kopia procesu rodzica
- Procesy nie mają wspólnej pamięci

Przykład!

asyncio / greenlet



- + Rezygnują z wątków w zamian za zadania (tasks) lub mikrowątki
- + Przeznaczone dla wielu operacji sieciowych / IO
- Współbieżność != Równoległość

Jython / Iron python



- + kompilują kod do javy / .NET, które mają swoją obsługę wątków
- Występuje hazard danych
- Wolniejsze od Cpython
- Problemy z bibliotekami

Przykład!

Pypy-stm



- + Bardzo szybki JiT, który gwarantuje pypy
- + Użyta STM (Software transaction memory) emuluje GIL
- Beta

Przykład!

Cython



- + jest bardzo szybki (kompiluje się do C)
- + dyrektywa nogil eliminuje użycie GIL w funkcji cython
- To już nie jest python :(
- Wymaga kompilatora

Przykład!

Źródła



- <https://wiki.python.org/moin/GlobalInterpreterLock>
- <https://docs.python.org/2/c-api/init.html#thread-state-and-the-global-interpreter-lock>
- http://cython.readthedocs.io/en/latest/src/userguide/external_C_code.html
- <http://www.dabeaz.com/python/UnderstandingGIL.pdf>
- <http://doc.pypy.org/en/latest/stm.html>

Q & A

Marek Piechula

Clearcode

Python Developer

marpiechula@gmail.com