Part 2: PyPy

Bookmark this page

Profiling & Performance

Part 2: PyPy

PyPy is an alternative Python interpreter focused on performance. It uses an optimizing just-in-time compiler which is particularly well suited to long-running programs.

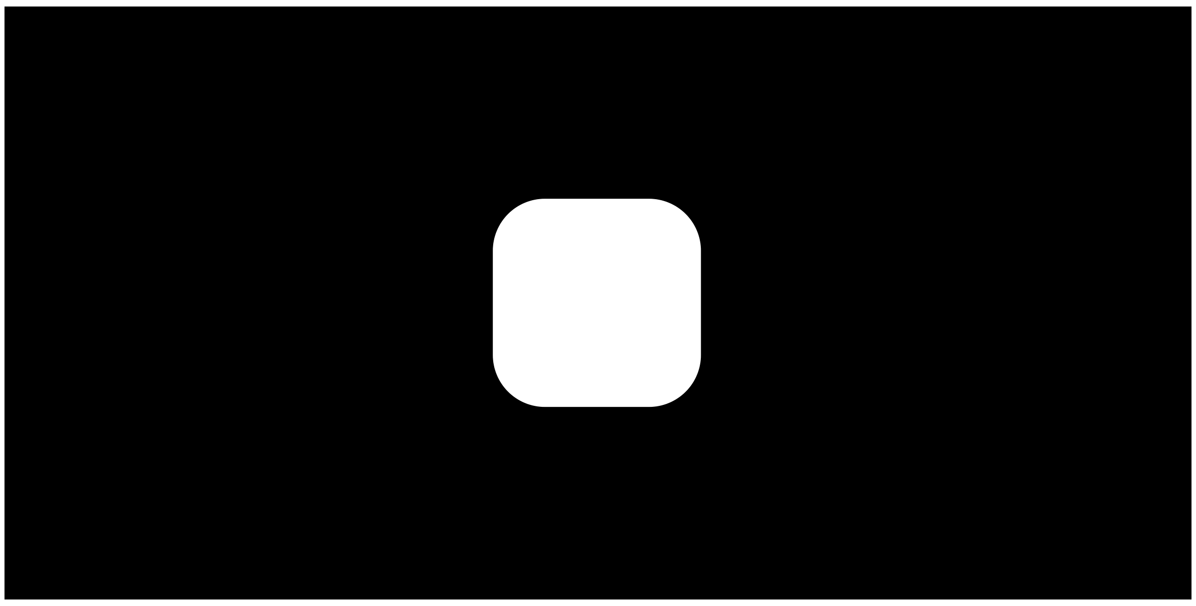
With very large data sets Python sometimes chokes whereas PyPy keeps going.

PyPy is best evaluated directly vis-a-via the standard CPython reference interpreter. Profile code under standard Python and then under PyPy.

PyPy can improve code that is factored out to functions, the more granular the better in some cases. Note that in the video, during the last timing test with a large number of iterations, PyPy runs faster with the factored code than the unfactored code. Remember to rely on timing and profiling to determine the best structure for your use case.

https://github.com/rriehle/ProfilingPerformance/tree/master/source/scripts

PYPY



Start of transcript. Skip to the end.

[INTRO MUSIC PLAYING]

Now we're going to start talking about ways

to improve your code that are effectively global changes.

They affect the entire context in which your code is running.

And the specific one that we're going to look at in this segment is PyPy.

This is an alternative interpreter to the

Video

Download video file

Transcripts

Download SubRip (.srt) file

Download Text (.txt) file

< Previous

Next >