

Мультипроцессинг

Python

Содержание

- процессы и потоки
- threading
- process
- joblib
- шаринг ресурсов
- Map Reduce

Зачем нужна многопоточность?

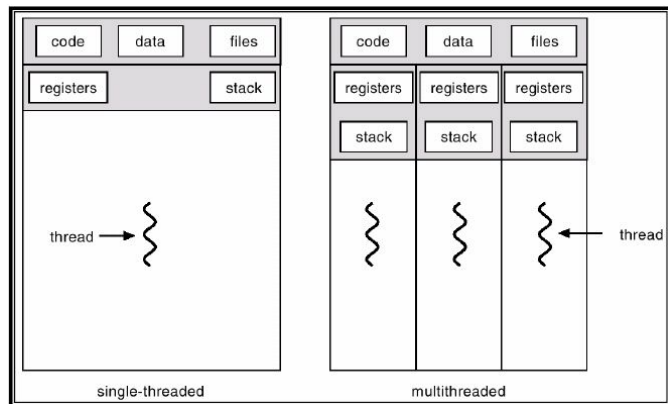
- Производительность
- Многозадачность (графические интерфейсы)
- Асинхронная обработка событий (запрос-обработка-ответ)
-

Процессы и потоки

- Процесс -- экземпляр программы во время выполнения, независимый объект, которому выделены системные ресурсы
 - грубо говоря, процесс = программа
 - используют разное (виртуальное) адресное пространство
 - у каждого процесса разные системные ресурсы (дескрипторы открытых файлов)

Хотим одновременно выполнить несколько задач? Создадим несколько процессов

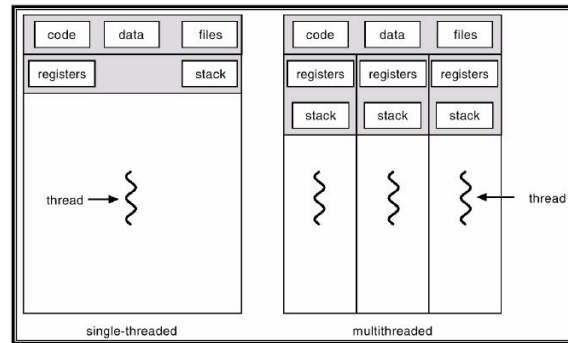
- Поток



Процессы и потоки

- Процесс -- экземпляр программы во время выполнения, независимый объект, которому выделены системные ресурсы
 - грубо говоря, процесс = программа
 - используют разное (виртуальное) адресное пространство
 - у каждого процесса разные системные ресурсы (дескрипторы открытых файлов)
- Поток -- определенный способ выполнения процесса. Каждый поток последовательно выполняет инструкции процесса, которому он принадлежит, параллельно* с другими потоками этого процесса.
 - поток выполняется внутри процесса, разделяя с ним системные ресурсы и адресное пространство

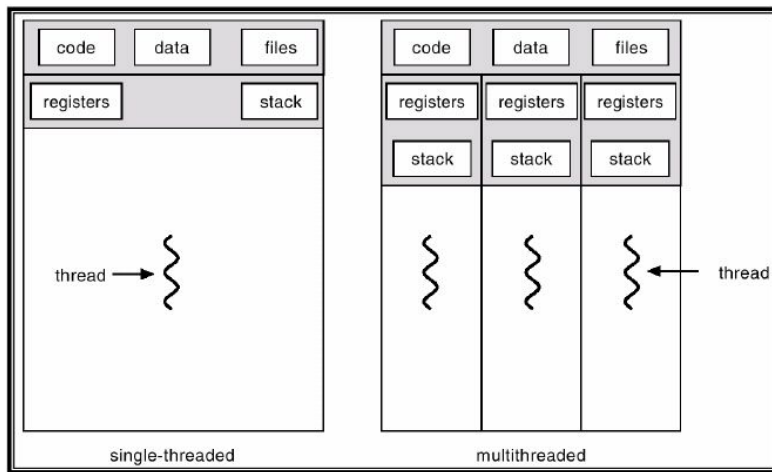
Хотим одновременно выполнить несколько задач, которым требуется доступ к разделяемому состоянию? Создадим несколько потоков!



Процессы и потоки

Выполнением нескольких процессов и потоков управляет ОС

Поочередно отдает процессорное время потоку или процессу (сколько-то циклов в зависимости от важности)



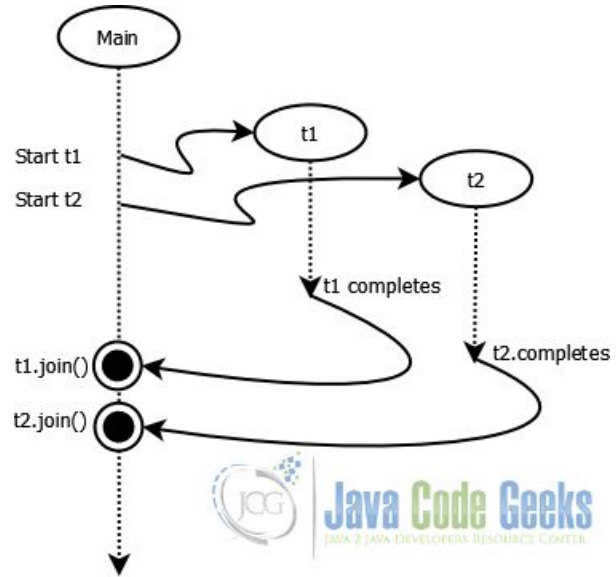
threading

В Python есть потоки - они системные - ими управляет ОС.

В модуле threading есть класс Thread который позволяет создавать потоки.

В ноутбук!

threading



GIL

GIL (global interpreter lock)

мьютекс, который гарантирует, что в каждый момент времени только один поток имеет доступ к внутреннему состоянию интерпретатора.

Зачем?

чтобы избежать проблем: гонки за ресурсы или порчи данных.
(есть проблемы когда есть доступ к общему ресурсу (данным или устройству), который не должен быть одновременно использован более чем одним потоком исполнения)

multiprocessing

У каждого процесса - свой GIL

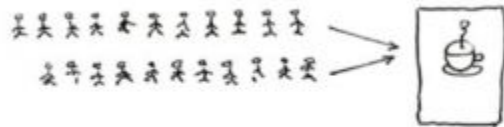
если нужно общение между процессами можно использовать multiprocessing.Pipe (сокеты)

или

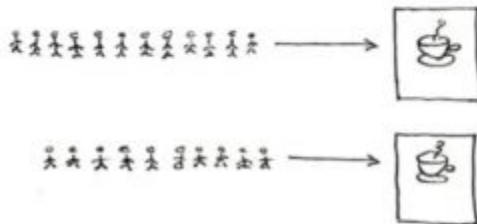
потокбезопасные структуры - queue.*Queue

CONCURRENCY | PARALLELISM

Concurrent = Two Queues One Coffee Machine

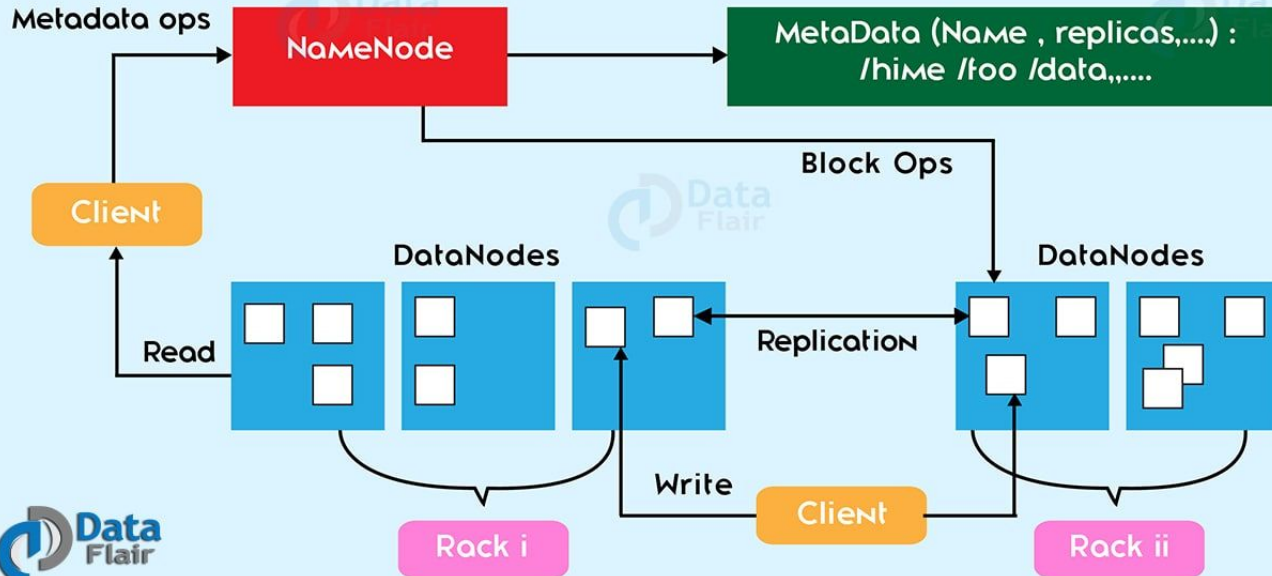


Parallel = Two Queues Two Coffee Machines

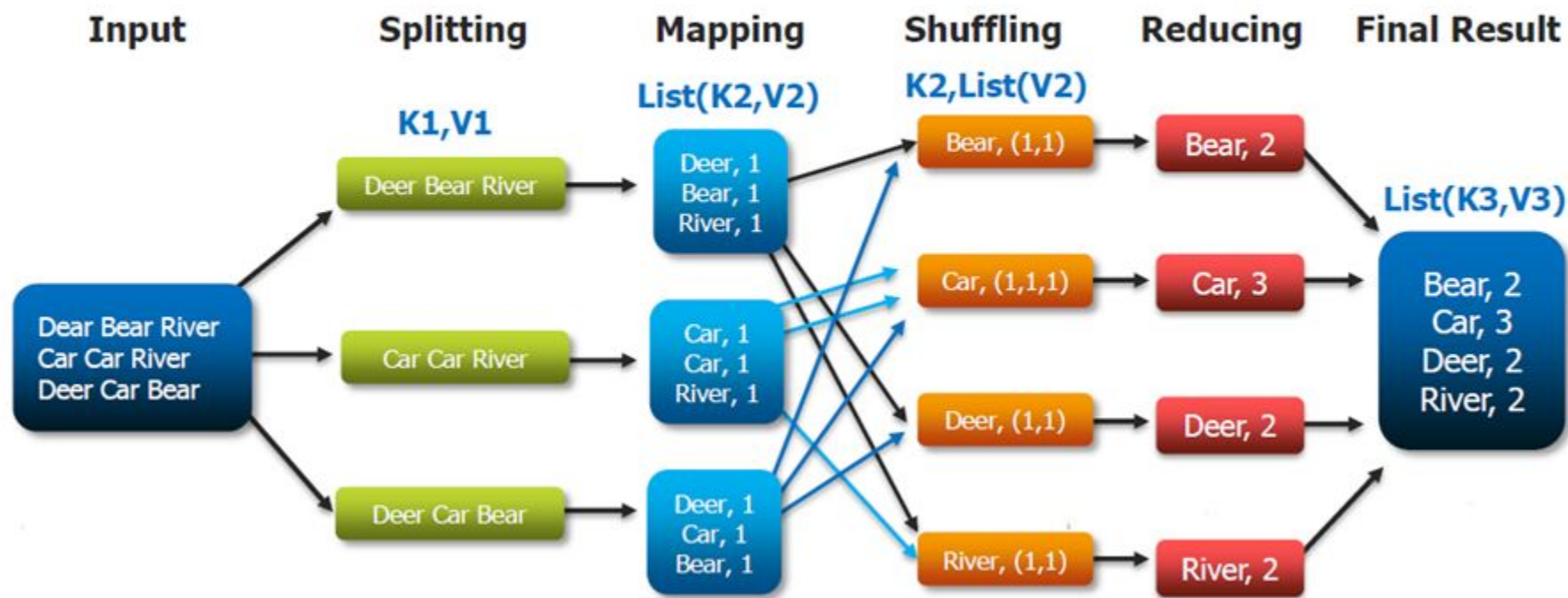




HDFS Architecture



The Overall MapReduce Word Count Process



<https://compscicenter.ru/courses/python/2015-autumn/classes/1562/>

<https://docs.python.org/3.7/library/concurrency.html>

<https://habr.com/post/421625/>