

# Потоки и процессы. Celery. MapReduce

@pvavilin

6 ноября 2022 г.

# Outline

# Процесс

Это программа, находящаяся в режиме выполнения. Операционная система подгружает в оперативную память с каждым процессом

- Саму программу
- Данные к программе
- Стек программы

Переключение между процессами происходит на уровне ядра.

# Поток

- Каждый процесс состоит из минимум одного потока.
- Потоки разделяют общее адресное пространство процесса.

Переключение между потоками может происходить как на уровне ядра, так и на уровне пользователя (процесса).

# Что можно узнать про процесс?

```
ps alx  
ls -l /proc/<PID>/
```

# создание процессов

Для создания нового процесса используются системные вызовы копирования процесса:

`fork` UNIX-системы

`CreateProcess` Win2k-системы

# CPU-bound / IO-bound задачи

**CPU-bound** задачи, которые активно используют CPU.  
Арифметические операции, матричные  
вычисления, поиск строк, и т.д.

**IO-bound** задачи, связанные с вводом-выводом данных.  
Работа с сетью, с файловыми системами, с  
пользовательским вводом

# GIL

## Python/ceval.c

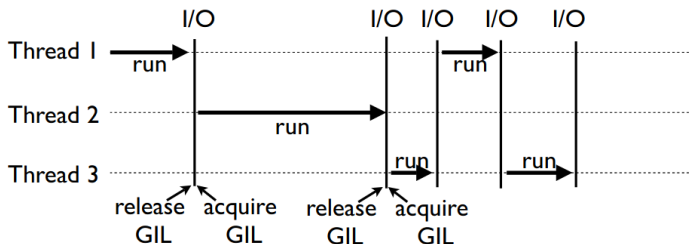
```
/* This is the GIL */  
static PyThread_type_lock  
    interpreter_lock = 0;
```



# GIL

GIL гарантирует интерпретатору, что только один *поток* может быть запущен в текущий момент. Это сделано для безопасной работы управления памятью, вызова расширений написанных на других языках (на C).

GIL



- `sys.getcheckinterval()` # -> Python2
- `sys.getswitchinterval()` # -> Python3

# GIL

GIL замедляет CPU-bound задачи. Старая реализация GIL очень плохо работала с *CPU-bound + IO-bound* задачами. Пример, да и новая не лучше.

# Практика

GitHub

## Дополнительная литература

- GIL
- UnderstandingGIL.pdf
- Groking The GIL
- GIL и его влияние на многопоточность в Python
- multiprocessing

# Что такое Celery?

## Официальная документация

*Celery* это брокер задач, который позволяет в фоновом, асинхронном режиме выполнять задачи в отдельных процессах/тредах и/или на других машинах.

# Практика запуска задач на Celery

```
pip install celery  
apt install rabbitmq-server
```

- Можно описывать сложные последовательности

# Что такое map-reduce

Это процесс решения больших задач при помощи разбивки данных на части и решения задач с частями данных на разных машинах. MapReduce состоит из обязательных шагов:

- 1 Map — разбить данные на блоки (присвоить каждой записи некоторый ключ блока)
- 2 Shuffle — присвоить каждому блоку некоторый ключ (*не-уникальный* между всеми блоками)
- 3 Reduce — для каждого ключа выполнить некоторую функцию над всеми данными в этом ключе



# Практика запуска map-reduce на pyspark

*тестовая сборка для работы с Hadoop (надо дополнительно поставить python на namenode)*

- mapper.py
- reducer.py
- создадим директорию в HDFS для данных и вывода (на NameNode)

```
hdfs dfs -mkdir /d
```

## Практика запуска map-reduce на pyspark

### ■ запуск на NameNode

```
hdfs dfs -rmkdir \
    --ignore-fail-on-non-empty \
    /d/out
hadoop jar /opt/hadoop-2.7.4/share \
    /hadoop/tools/lib/ \
    hadoop-streaming-2.7.4.jar \
    -files /root/mapper.py, \
    /root/reducer.py \
    -mapper /root/mapper.py \
    -reducer /root/reducer.py \
    -input /d/out/98.txt \
    -output /d/out
hdfs dfs -cat /d/out/part-00000
```

# Вопросы-ответы

