Ивченков Дмитрий М3234

Лабораторная работа номер 6(шесть)

Данные системы:

```
maybebabyenjoyer@LAPTOP-EOQIRLHS:/mnt/d/OS-Lite/HW-5$ top -b -n 1 | head -n 5 | tail -n 2
MiB Mem :   1906.1 total,   1381.7 free,    327.4 used,    197.0 buff/cache
MiB Swap:   1024.0 total,   1024.0 free,      0.0 used.   1437.3 avail Mem
```

```
maybebabyenjoyer@LAPTOP-EOQIRLHS:/mnt/d/OS-Lite/HW-6$ lscpu
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   48 bits physical, 48 bits virtual
CPU(s):                          2
On-line CPU(s) list:             0,1
Thread(s) per core:              2
Core(s) per socket:              1
Socket(s):                       1
Vendor ID:                       AuthenticAMD
CPU family:                      23
Model:                           104
Model name:                      AMD Ryzen 5 5500U with Radeon Graphics
Stepping:                        1
CPU MHz:                         2096.066
BogoMIPS:                        4192.13
Hypervisor vendor:               Microsoft
Virtualization type:             full
L1d cache:                       32 KiB
L1i cache:                       32 KiB
L2 cache:                        512 KiB
L3 cache:                        4 MiB
```

ЧАСТЬ ПЕРВАЯ.

Сразу скажу что я не собирал каждый датасет 3 часа (зачем? Интересный вопрос). У нас в итоге будет отличие на константу по времени +- шумы от того что данных мало, но в целом вывод очевиден. Перейдем к делу.

ОСНОВА(xd.sh):

```
c=$1
for i in {1..100}
do
    for j in {1..100}
    do
        c=$(($c + $2))
    done
done
```

Последовательно(1.sh)

```
#!/bin/bash
N=$1
for i in $(seq 1 $N)
do
    bash xd.sh i i
done
```

Параллельно(2.sh)

```
#!/bin/bash
N=$1
for i in $(seq 1 $N)
do
    bash xd.sh i i &
done
wait
```

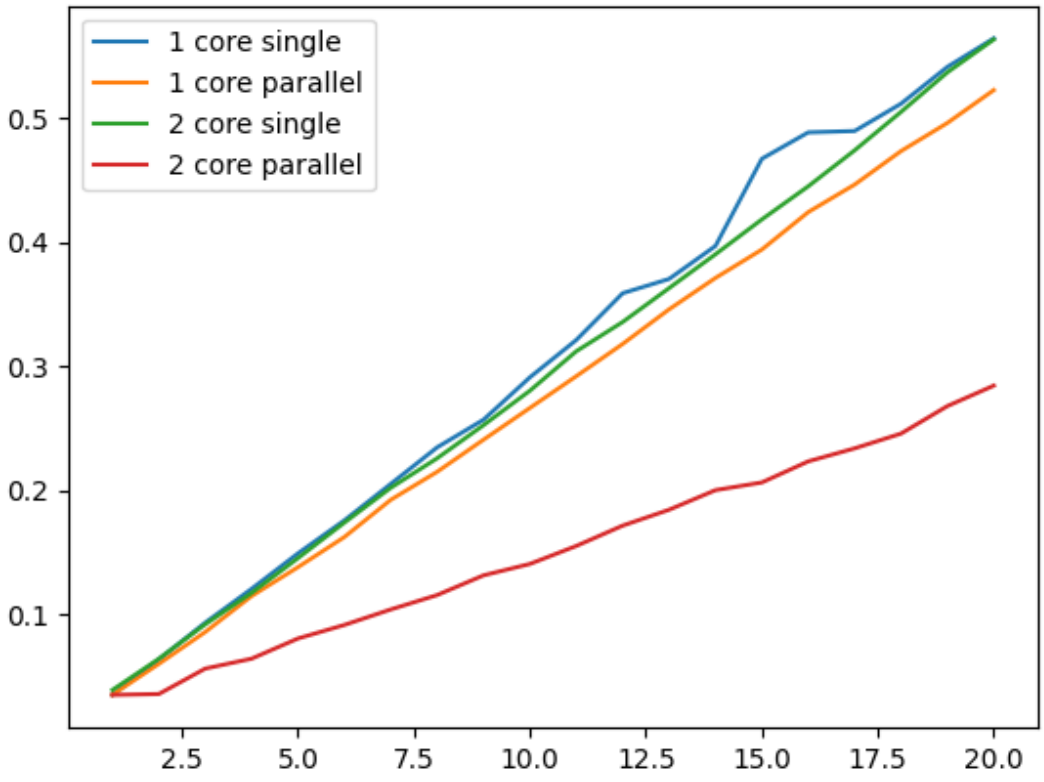
Запускатор последовательный(run1.sh)

```
for N in {1..20}
do
    echo "N = $N" >> c.txt
    for j in {1..10}
    do
        { time bash 1.sh $N >> c.txt ; } >> c.txt 2>&1
    done
done
```

Запускатор параллельный(run2.sh)

```
for N in {1..20}
do
    echo "N = $N" >> xd4.txt
    for j in {1..10}
    do
        { time bash 2.sh $N >> xd4.txt ; } >> xd4.txt 2>&1
    done
done
```

Графики:

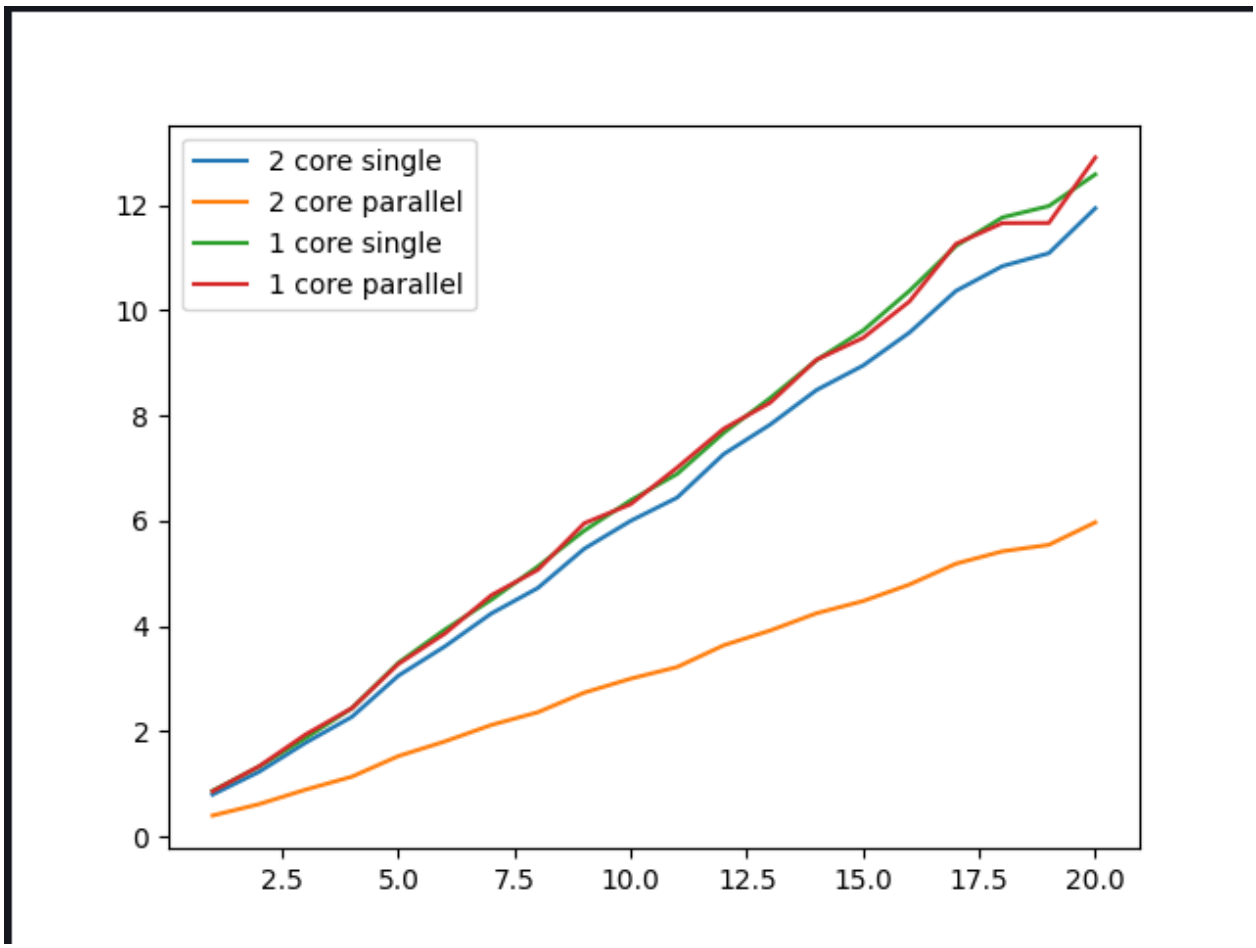


Итоги. Подписаться.

Мы видим что все, за исключением параллелки на двух процессорах эквивалентно. (Они одинаковые, смещение вызвано помехами при сборе датасета так как данных поменьше). Вполне очевидно почему. Параллелка круто!

ЧАСТЬ ВТОРАЯ

Там таки тоже самое не особо понимаю смысла делать две части у этой лабы. Скриптики залью вместе с отчетом, а теперь к графикам.



Ровно такая же ситуация как в первом эксперименте, 3 первых работают одинаково, параллелка на 2+ процессорах лучше. Ладно. Перейдем к выводам.

Вывод: Параллелить это хорошо. Но не на одном процессоре, так как они просто исполняются по очереди. А когда их два они ну делаются между процессорами, заканчивают работу примерно в одно время и так далее. Ускорили примерно в два раза как раз (ожидаемо). Круто!