# Spark Parallel Execution

**Spark information:**

m4.xlarge
8 vCore, 16 GiB memory, EBS only storage
EBS Storage:32 GiB
1 Master node
2/4 Core nodes

Architecture: x86_64
CPU op-mode(s): 32-bit, 64-bit
Byte Order: Little Endian
CPU(s): 4
On-line CPU(s) list: 0-3
Thread(s) per core: 2
Core(s) per socket: 2
Socket(s): 1
NUMA node(s): 1
Vendor ID: GenuineIntel
CPU family: 6
Model: 79
Model name: Intel(R) Xeon(R) CPU E5-2686 v4 @ 2.30GHz
Stepping: 1
CPU MHz: 2300.022
BogoMIPS: 4600.04
Hypervisor vendor: Xen
Virtualization type: full
L1d cache: 32K
L1i cache: 32K
L2 cache: 256K
L3 cache: 46080K
NUMA node0 CPU(s): 0-3

Amazon Linux AMI release 2017.03
4.4.35-33.55.amzn1.x86_64
Python 2.7.12

'--build=x86_64-redhat-linux-gnu' '--host=x86_64-redhat-linux-gnu' '--target=x86_64-amazon-linux-gnu' '--program-prefix=' '--prefix=/usr' '--exec-prefix=/usr' '--bindir=/usr/bin' '--sbindir=/usr/sbin' '--sysconfdir=/etc' '--datadir=/usr/share' '--includedir=/usr/include' '--libdir=/usr/lib64' '--libexecdir=/usr/libexec' '--localstatedir=/var' '--sharedstatedir=/var/lib' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--enable-ipv6' '--enable-shared' '--enable-unicode=ucs4' '--with-dbmliborder=gdbm:ndbm:bdb' '--with-system-expat' '--with-system-ffi' '--with-dtrace' '--with-tapset-install-dir=/usr/share/systemtap/tapset' '--with-valgrind' 'build_alias=x86_64-redhat-linux-gnu' 'host_alias=x86_64-redhat-linux-gnu' 'target_alias=x86_64-amazon-linux-gnu' 'CC=gcc' 'CFLAGS=-O2 -g -pipe -Wall -Wp,-D_FORTIFY_SOURCE=2 -fexceptions -fstack-protector --param=ssp-buffer-size=4 -m64 -mtune=generic -D_GNU_SOURCE -fPIC -fwrapv ' 'LDFLAGS= ' 'CPPFLAGS= ' 'PKG_CONFIG_PATH=%{_PKG_CONFIG_PATH}:/usr/lib64/pkgconfig:/usr/share/pkgconfig'

**Description of the experiment:**

I used the Distributed Grep Spark script and the large version of the movielens data set (file ratings.csv) to show the ratings with 5.0 stars.

First I evaluated performance and speed-up in the local mode installation of Spark on an m4.xlarge instance.

Local Mode:

62.726634 s setMaster(local[1])
37.210488 s setMaster(local[2])
36.469688 s setMaster(local[3])
36.341190 s setMaster(local[4])

Later I evaluated performance and speed-up on a cluster with 2 and 4 m4.xlarge instances and 1 and 2 cores per node.

Spark Cluster:

2 nodes:
213.601880 s spark-submit --num-executors 1 --executor-cores 1 P21_spark.py
142.108002 s spark-submit --num-executors 1 --executor-cores 2 P21_spark.py
228.467354 s spark-submit --num-executors 2 --executor-cores 1 P21_spark.py
114.128206 s spark-submit --num-executors 2 --executor-cores 2 P21_spark.py

4 nodes:
126.397747 s spark-submit --num-executors 3 --executor-cores 1 P21_spark.py
55.482909 s spark-submit --num-executors 3 --executor-cores 2 P21_spark.py
82.195508 s spark-submit --num-executors 4 --executor-cores 1 P21_spark.py
121.731950 s spark-submit --num-executors 4 --executor-cores 2 P21_spark.py

Later I tried tuning three different parameters of the Spark configuration on the larger cluster with 4 nodes.

Tuning for cluster with 4 nodes:

21.989862 s spark-submit --num-executors 4 --executor-cores 2 --executor-memory 2g P21_spark.py
24.527546 s spark-submit --num-executors 4 --executor-cores 2 --driver-memory 2g P21_spark.py
23.451891 s spark-submit --num-executors 4 --executor-cores 2 --driver-cores 2 P21_spark.py

**Discussion about performance, speed-up and tuning:**

In the local mode on a virtual machine the significant change in performance occurs when changing from 1 core to 2 cores. Increasing the number of cores even further does not influence performance due to the limited number of cores actually available (only 2 cores).

In the cluster mode with 2 nodes increasing the number of executor cores results in highly improved performance. Increasing the number of executors only slightly influences performance.
In the cluster mode with 4 nodes both changing the number of executors and the number of executor cores influences performance significantly.

Tuning different configuration parameters proved that increasing the executor and the driver memory (both from 1g to 2g) as well as adding more driver cores (2 instead of 1) leads to significant performance improvements.