Politechnika
Warszawska

Cloud Computing

Calculating chances of winning for each set of cards in BlackJack

Kacper Kowalczyk

1 Introduction

The main purpose of this project was to show the difference between running calculations on one local computer and carring the same calculations on few instances using Cluster Computing frameworks. In this very project I have used Apache Spark which is a well known Cluster Computing framework. I have also used Amazon Web Service (AWS), a very extensive service that among other options allows user to carry out calculations on their servers instances, which leads to reducing the time of calculations.

2 Project Overview

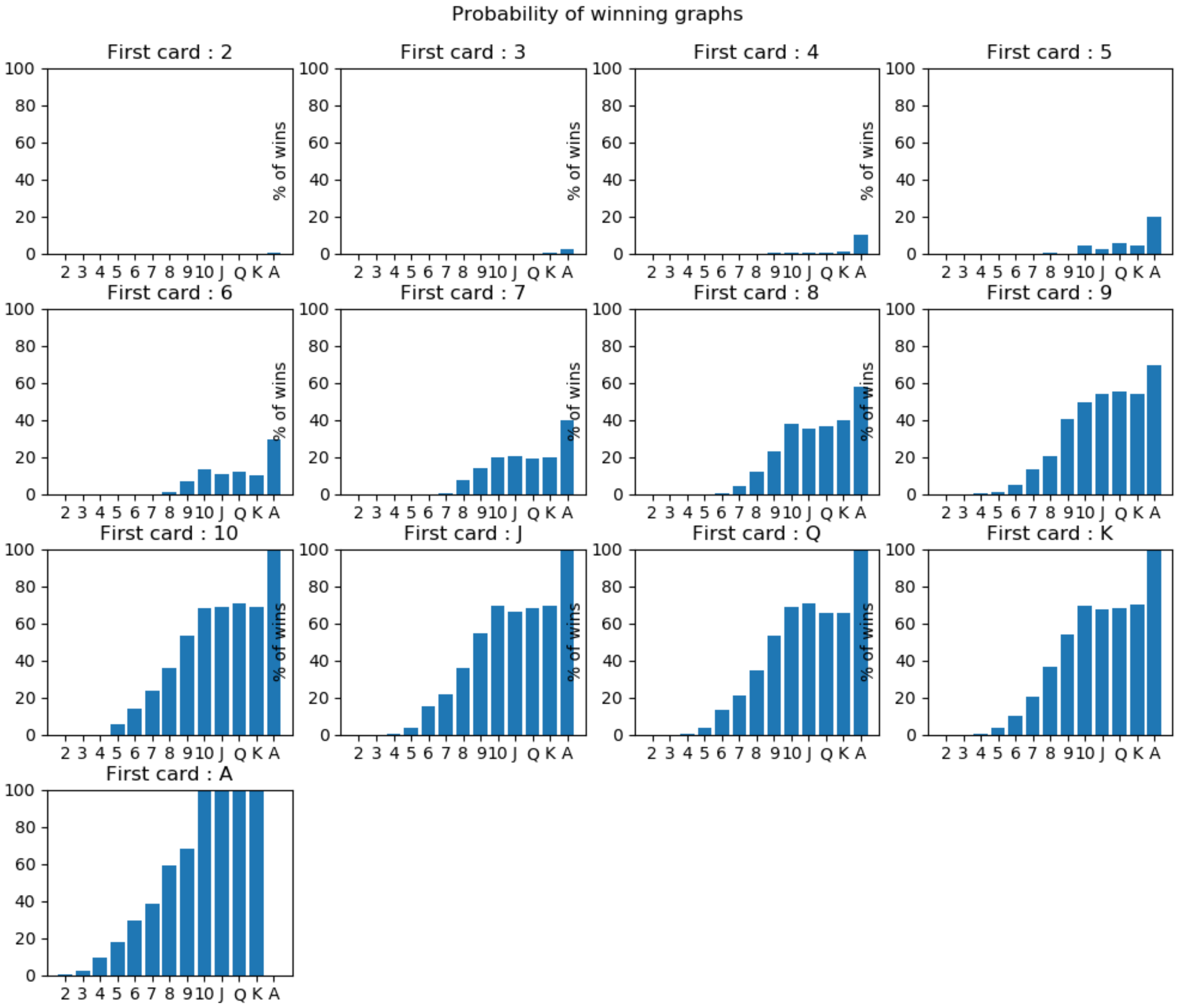
I decided to make a project that involves one of the things I like - Black Jack.

2.1 Quick Black Jack Overview

Black Jack is one of the most popular gambling, card games. In this game you are playin against casino one vs one. The main goal is to get more points than the casino but no more that 21 points. Each card has a specific number of points. At the beginning of the game player draws two cards and after that player can say that he does not want more cards or if he thinks that he can get more points without getting more than 21 points he can draw until he will be satisfied or if he will have more than 21 points. After that its casino's turn and if it has more or as many points as the player, the player loses.

2.2 The Code

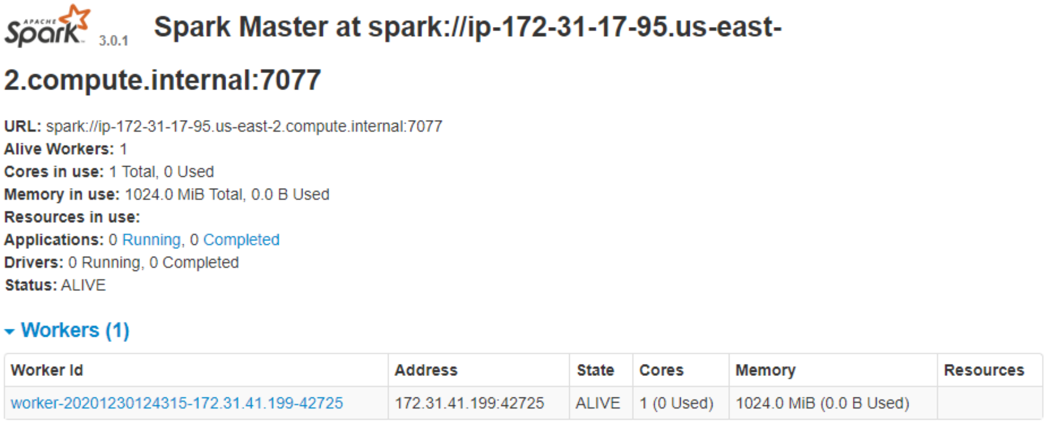
Summing the paragraph above the code that would calculate chances for winning in a situation that player could draw multiple times would be very expanded. Keeping in mind that this was not the most important part of the whole project I decided to make the code a little bit easier to write. I assumed that player does not draw cards after the first two. In other words player does not look at his cards, he just draws them and wait until casino makes their move. The final version of code works as follows : for each set of two cards in hand of a player the code plays the game as casino and checks the result. User can set how many iterations for each set of cards should the code consider. The more iterations the more accurate the calculations are. Of course for some sets of cards this idea is not the best e.g. If a player gets cards : two and three he gets 5 points and casino will always win in this situation. But when it comes to better cards the calculations make more sense. Below I presented the graphs for chances of winning. I also uploaded a bigges image of graphs to my Github reposytory.



Rysunek. 1: Probability of winning for each set of cards (1000 iterations)

2.3 Cloud Computing

In this paragraph I will try to briefly describe the process of configuring the AWS and Apache. In the end I decided to make calculations with two slaves connected to master instance because adding more would make the calculations faster but it would be just repeating same configurations again and again for each slave instance. Every instance was updated and then I installed on every single one of them Java and Spark. After that I have started them and connected to a cluster. Here is a screenshot of one worker attached to the cluster.



Spark 3.0.1 **Spark Master at spark://ip-172-31-17-95.us-east-2.compute.internal:7077**

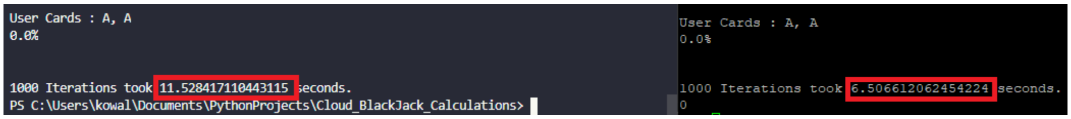
URL: spark://ip-172-31-17-95.us-east-2.compute.internal:7077
Alive Workers: 1
Cores in use: 1 Total, 0 Used
Memory in use: 1024.0 MiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (1)

Worker Id	Address	State	Cores	Memory	Resources
worker-20201230124315-172.31.41.199-42725	172.31.41.199:42725	ALIVE	1 (0 Used)	1024.0 MiB (0.0 B Used)	

Rysunek. 2: Spark Configuration

After repeating this process I ran the code on my PySpark console on the master instance.



```
User Cards : A, A
0.0%


1000 Iterations took 11.528417110443115 seconds.
PS C:\Users\kowal\Documents\PythonProjects\Cloud_BlackJack_Calculations>
```

```
User Cards : A, A
0.0%


1000 Iterations took 6.506612062454224 seconds.
0
```

Rysunek. 3: Comparing calculations time

2.4 Conclusions

As we can see the calculations on my local PC were about two times slower than calculations carried out using Cloud Computing. I only used two slaves, but I suppose that using more slaves would speed the calculations up more. Cloud computing is a very strong tool when it comes to conducting calculations. In my case that was just a few seconds, but it is not hard to imagine that sometimes a calculations could take many, many more time and then reducing this time by even a half would make a great difference.