Introduction

For our project, we decided to compare the performance of running Spark on Azure vs running a regular Azure SQL database. We are using a dataset of NBA shots, which has a record for every individual shot attempted in the NBA since 1996.

After scraping our data, we uploaded a CSV with all of our records to both a regular SQL Server database and a Databricks Cluster, both set up through Azure. In order to benchmark the two setups and compare their runtimes, we then created seven different queries to analyze the data and compare the two.

Setup

Setting up Azure Databricks required us to create and set up a workspace. From there, we created a cluster for each of us to share. Unfortunately, the free tier had a quota on the number of nodes that you can use, so we did bite the bullet and used a pay-as-you-go subscription that charges based on the amount of hours spent using a given resource in Azure.

However, once we were able to create the cluster, we were both able to simultaneously work on a shared Jupyter notebook. This is where we placed our queries, to run with Spark.

The standard SQL Server database was configured in a similar way. Using the standard pay-as-you-go subscription, we took advantage of Azure's free tier for SQL Databases. We set up our database with only 1 vCore and the free 32 GB of storage.

Data/Queries

Our dataset originated from a CSV file containing millions of records on individual shots taken in each NBA game since 1996. Our queries stem from seven different questions we had about

our data, and we were curious to run them and retrieve not only the query results, but also the runtime results. The questions we set out to answer include:

1. Have popular “regions” of the court shifted over time?
2. What is the average field goal percentage for home teams vs away teams? How has this changed over time?
3. What players attempt the most 3 point shots? For these players, what percentage of the total shots do they use performing Layup Shots, Jump Shots, etc.?
4. What section of the court does each player shoot the most from? Which section does each player have the highest field goal percentage?
5. What is the average shot distance per period?
6. What team has the highest field goal percentage for any team in any year? Which team has the lowest?
7. What is the average points per attempt for each player with a minimum of 300 attempts? How about the average points per attempt during the last 5 minutes of a game (minimum 200 attempts)?

Results

So far, we have been able to construct Spark queries in Azure Databricks. For question 3, our query utilized several subqueries, since we first needed to filter for players who attempted the most 3 point shots. Next, we had to join this subquery on the original data so that we could get the total number of shots attempted for each player and shot type. After this, we had to do one more join so that we could get the total number of shots for each player, regardless of the shot type. Finally, from there, we calculated the percentage as a new column, then pivoted the

percentage values into their own column, such that there was a column for each shot type. Since this query had a lot of joins, it is a prime example to use for comparing runtimes.

Running this using Spark SQL resulted in a runtime of 2.93 seconds, which is pretty decent considering the size of the data and the number of joins we were performing. We performed a similar process for question 4, utilizing multiple joins to create our result. The runtime for query 4 took 1.29 seconds using Spark SQL, so we can see that these queries are running quickly.

However, we wanted to compare this with a normal database query, in order to really illustrate the benefits of Spark. We ran this query on our regular Azure database and the runtime was drastically different, taking 67 seconds to execute. While we knew Spark was beneficial, we didn't realize just how drastic the difference would be.

Next Steps

As of this moment, we have constructed and run all queries on Azure Databricks using Spark SQL commands, and we have run one query on the normal SQL database. We are still planning to run the remaining queries on the other database, although we will need to tweak some of the queries as there is not a one-to-one compatibility between the different SQL dialects that each environment uses. This does make it a bit tricky since we want to make our queries as identical as possible for a fair comparison. Once we are able to run all of the queries, we will focus on making visualizations and performing statistical analysis to measure just how much of an impact on performance Spark has.