

# Projeto IS

Integração Sistemas



UNIVERSIDADE DE COIMBRA

Francisco Barão 2016238992

Pedro Ribeiro 2016218753

# Index

<b>Conditions of experiment</b>	<b>3</b>
<b>Client-Server</b>	<b>4</b>
<b>Client functions</b>	<b>4</b>
<b>Server functions</b>	<b>4</b>
<b>Time measuring</b>	<b>4</b>
<b>Google Protocol Buffer</b>	<b>5</b>
<b>Data Structure</b>	<b>5</b>
<b>XML</b>	<b>6</b>
<b>Data Structure</b>	<b>6</b>
<b>Trials</b>	<b>7-9</b>
<b>Tables</b>	<b>7-8</b>
<b>Graphs</b>	<b>9</b>
<b>Conclusion</b>	<b>10</b>
<b>Advantages</b>	<b>10</b>

# Conditions of Experiment

Computer characteristics:

- 1) Processor: i7-8565U Quad Core 1.8-4.6 GHz  
Ram: 8GB

Tecnologies:

- Java (version) – 8.0 (jdk: 1.8.0.222)
- gRPC framework to connect client-server services (version) – 1.24.1
- Maven for dependencies – 4.0.0 version
- Google protocol Buffer and XML

How to run:

- Server : `mvn exec:java -Dexec.mainClass=com.assign_1.ProjectIsServer`
- Client : `mvn exec:java -Dexec.mainClass=com.assign_1.ProjectIsClient`

Changeable Parameters:

- Port (Currently on port 5682)
- Number of random generated client/cars per client
- Number of Clients to be requested to the server
- Client/Car parameters to be retrieved

# Client - Server

All Client-server communication is done using gRPC framework which is used to efficiently and easily connect these two.

## Client Functions:

The client starts by creating the message to be sent to the server. This message is either a XML String or a Protocol Buffer message containing Id's of clients to be returned.

The client will use Stub to call a remote function from the server to request owner and car information.

After receiving the information, the client will then deserialize or unmarshal the messages and record the endTime/packageSize.

## Server Functions:

After the server has been created and its service added, the server either generates clients or retrieves them from an object file working as a database. When a client connects the server, it will receive the clients Ids and proceed to find the owners and their cars. When found, the server either creates the Reply message by creating the cars and owners builders and attaching them to the message or it creates the XML String to be sent to the client and recording the startTime after Marshaling.

## Measuring Package Size:

Package Size is measured at the client when he receives the package from the server.

## Measuring Time:

startTime is recorded when the server ends marshalling/serializing the information to be sent to the client.(Fig. 1)

EndTime is recorded when the client finishes unmarshalling/deserializing. (Fig. 2)

Finally we subtract startTime to the endTime.

```
195 private Reply makeResponse(OwnersRequest id_list) {
196     List<Integer> owners = id_list.getIdList();
197     ArrayList<O> reply_owners = new ArrayList<>();
198     Reply rep;
199
200     long startTime = System.currentTimeMillis();
201     timeToFile(startTime);
202     infoToFile(numberOwners, numberCars);
203
204     for (int i : owners) {
```

Fig.1 – startTime on file("ProjectIsServer")

```
81     }
82     sizeToFile(response.getSerializedSize());
83 }
84 long endTime = System.currentTimeMillis();
85 timeToFile(endTime);
86 // Function end
87 }
```

Fig.2 – endTime on file("ProjectIsClient")

# Google Protocol Buffer

## Data structures:

message OwnersRequest - Request message that contains array of client ids to be retrieved from the server.

message Reply – Reply message that contains array of owners and necessary information to be retrieved.

message O – Message defining each owner contained in Reply.

Each owner contains:

-id | name | telephone | address | Cars

message C – Message defining each car contained in an Owner.

Each car contains:

-id | brand | model | engineSize | power | consumption | plate

# XML

## Data Structures:

We used java classes with the javax.xml annotations to create the XML (JAXB). In order to create a list of Owners we had to create another class that contains a list of owners, which each owner has a list of cars. Using the Xml attribute and element annotations, the creation of the XML is automated and contains every piece of information we needed.

OwnerList class that has an ArrayList of Owners.

Owner class has an ArrayList of Cars, an id, a name, a telephone number and an address.

Cars class has an id, a brand, a model, engine size, power, consumption, plate and belongs to an owner.

# Trials

## Tables

Input Variables:

-Number of Owners (with 10 cars each owner):

We ran each test 5 times to have more precise values.

Values : 50 | 150 | 300 | 500 | 1000 | 3000 | 5000 | 7500 | 10000

Output Variables:

startTime || endTime || Size

Protocol Buffer					XML				
Owners	Cars	startTime	endTime	Size	Owners	Cars	startTime	endTime	Size
50	10	1,57015E+12	1,57015E+12	18567	50	10	1,57019E+12	1,57019E+12	141346
50	10	1,57015E+12	1,57015E+12	18567	50	10	1,57019E+12	1,57019E+12	141346
50	10	1,57015E+12	1,57015E+12	18567	50	10	1,57019E+12	1,57019E+12	141346
50	10	1,57015E+12	1,57015E+12	18567	50	10	1,57019E+12	1,57019E+12	141346
50	10	1,57015E+12	1,57015E+12	18567	50	10	1,57019E+12	1,57019E+12	141346
150	10	1,57015E+12	1,57015E+12	56003	150	10	1,57019E+12	1,57019E+12	424597
150	10	1,57015E+12	1,57015E+12	56003	150	10	1,57019E+12	1,57019E+12	424597
150	10	1,57015E+12	1,57015E+12	56003	150	10	1,57019E+12	1,57019E+12	424597
150	10	1,57015E+12	1,57015E+12	56003	150	10	1,57019E+12	1,57019E+12	424597
150	10	1,57015E+12	1,57015E+12	56003	150	10	1,57019E+12	1,57019E+12	424597
300	10	1,57015E+12	1,57015E+12	112226	300	10	1,57019E+12	1,57019E+12	850297
300	10	1,57015E+12	1,57015E+12	112226	300	10	1,57019E+12	1,57019E+12	850297
300	10	1,57015E+12	1,57015E+12	112226	300	10	1,57019E+12	1,57019E+12	850297
300	10	1,57015E+12	1,57015E+12	112226	300	10	1,57019E+12	1,57019E+12	850297
300	10	1,57015E+12	1,57015E+12	112226	300	10	1,57019E+12	1,57019E+12	850297
500	10	1,57015E+12	1,57015E+12	187242	500	10	1,57019E+12	1,57019E+12	1417897
500	10	1,57015E+12	1,57015E+12	187242	500	10	1,57019E+12	1,57019E+12	1417897
500	10	1,57015E+12	1,57015E+12	187242	500	10	1,57019E+12	1,57019E+12	1417897
500	10	1,57015E+12	1,57015E+12	187242	500	10	1,57019E+12	1,57019E+12	1417897
500	10	1,57015E+12	1,57015E+12	187242	500	10	1,57019E+12	1,57019E+12	1417897
1000	10	1,57015E+12	1,57015E+12	374742	1000	10	1,57019E+12	1,57019E+12	2836899
1000	10	1,57015E+12	1,57015E+12	374742	1000	10	1,57019E+12	1,57019E+12	2836899
1000	10	1,57015E+12	1,57015E+12	374742	1000	10	1,57019E+12	1,57019E+12	2836899
1000	10	1,57015E+12	1,57015E+12	374742	1000	10	1,57019E+12	1,57019E+12	2836899
1000	10	1,57015E+12	1,57015E+12	374742	1000	10	1,57019E+12	1,57019E+12	2836899
3000	10	1,57019E+12	1,57019E+12	1138362	3000	10	1,57019E+12	1,57019E+12	8534899
3000	10	1,57019E+12	1,57019E+12	1138362	3000	10	1,57019E+12	1,57019E+12	8534899
3000	10	1,57019E+12	1,57019E+12	1138362	3000	10	1,57019E+12	1,57019E+12	8534899
3000	10	1,57019E+12	1,57019E+12	1138362	3000	10	1,57019E+12	1,57019E+12	8534899
3000	10	1,57019E+12	1,57019E+12	1138362	3000	10	1,57019E+12	1,57019E+12	8534899
5000	10	1,57019E+12	1,57019E+12	1908340	5000	10	1,57019E+12	1,57019E+12	14232899
5000	10	1,57019E+12	1,57019E+12	1908340	5000	10	1,57019E+12	1,57019E+12	14232899
5000	10	1,57019E+12	1,57019E+12	1908340	5000	10	1,57019E+12	1,57019E+12	14232899
5000	10	1,57019E+12	1,57019E+12	1908340	5000	10	1,57019E+12	1,57019E+12	14232899
5000	10	1,57019E+12	1,57019E+12	1908340	5000	10	1,57019E+12	1,57019E+12	14232899
7500	10	1,57019E+12	1,57019E+12	2870819	7500	10	1,57019E+12	1,57019E+12	21355399
7500	10	1,57019E+12	1,57019E+12	2870819	7500	10	1,57019E+12	1,57019E+12	21355399
7500	10	1,57019E+12	1,57019E+12	2870819	7500	10	1,57019E+12	1,57019E+12	21355399
7500	10	1,57019E+12	1,57019E+12	2870819	7500	10	1,57019E+12	1,57019E+12	21355399
7500	10	1,57019E+12	1,57019E+12	2870819	7500	10	1,57019E+12	1,57019E+12	21355399
10000	10	1,57019E+12	1,57019E+12	3833285	10000	10	1,57019E+12	1,57019E+12	28477900
10000	10	1,57019E+12	1,57019E+12	3833285	10000	10	1,57019E+12	1,57019E+12	28477900
10000	10	1,57019E+12	1,57019E+12	3833285	10000	10	1,57019E+12	1,57019E+12	28477900
10000	10	1,57019E+12	1,57019E+12	3833285	10000	10	1,57019E+12	1,57019E+12	28477900
10000	10	1,57019E+12	1,57019E+12	3833285	10000	10	1,57019E+12	1,57019E+12	28477900

Fig.3 – Protocol Buffer table values

Fig.4 – XML table values

From the tables (Fig. 3 and Fig. 4 ) we calculated the average times (startTime. EndTime and also Time which consists of endTime - startTime) and the average packages' size of the serilazitations in question.

Protocol Buffer:

50	10	1,570147E+12	1,57015E+12	6,540015E-08	18567
150	10	1,57015E+12	1,57015E+12	1,01E-07	56003
300	10	1,57015E+12	1,57015E+12	1,333999E-07	112226
500	10	1,57015E+12	1,57015E+12	2,136001E-07	187242
1000	10	1,57015E+12	1,57015E+12	3,397998E-07	374742
3000	10	1,57019E+12	1,57019E+12	6,372002E-07	1138362
5000	10	1,57019E+12	1,57019E+12	7,218E-07	1908340
7500	10	1,57019E+12	1,57019E+12	1,151E-06	2870819
10000	10	1,57019E+12	1,57019E+12	1,4832E-06	3833285

Fig.5 – Protocol Buffer average values table

XML:

50	10	1,570187E+12	1,57019E+12	1,13E-07	141346
150	10	1,57019E+12	1,57019E+12	6,9E-08	424597
300	10	1,57019E+12	1,57019E+12	1,148E-07	850297
500	10	1,57019E+12	1,57019E+12	2,04E-07	1417897
1000	10	1,57019E+12	1,57019E+12	3,554001E-07	2836899
3000	10	1,57019E+12	1,57019E+12	4,117E-06	8534899
5000	10	1,57019E+12	1,57019E+12	4,56E-06	14232899
7500	10	1,57019E+12	1,57019E+12	9,9114E-06	21355399
10000	10	1,57019E+12	1,57019E+12	1,731E-05	28477900

Fig.6 – XML average values table

From this Tables(Fig.5 and Fig.6), we were already able to see that for the same input values, XML takes longer to marshal/unmarshal and the packages are way bigger than Protocol Buffer. With the Graphs below, we hoped to improve our understanding of these results.



## Graphs

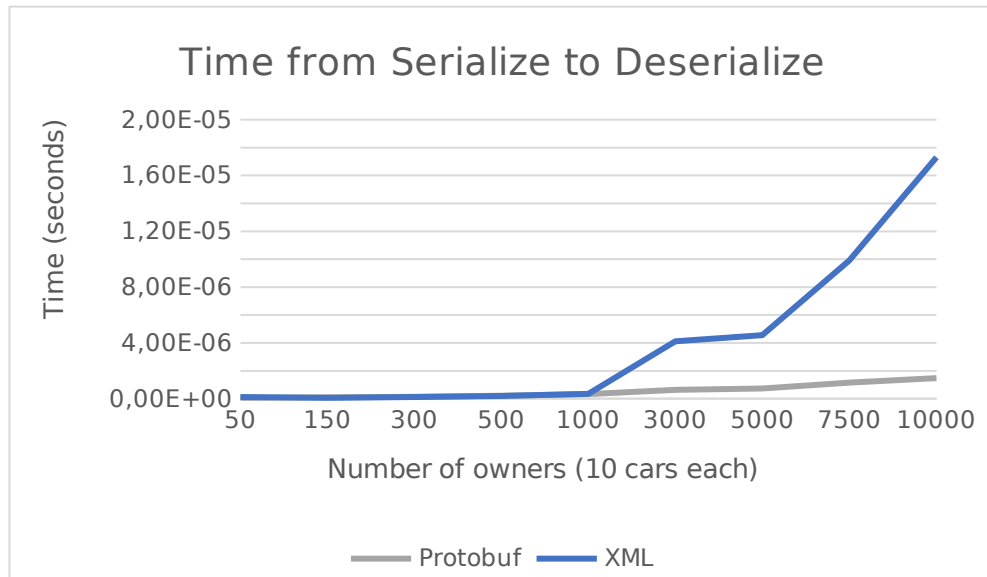


Fig.7 – Time comparison of Protobuffer and XML

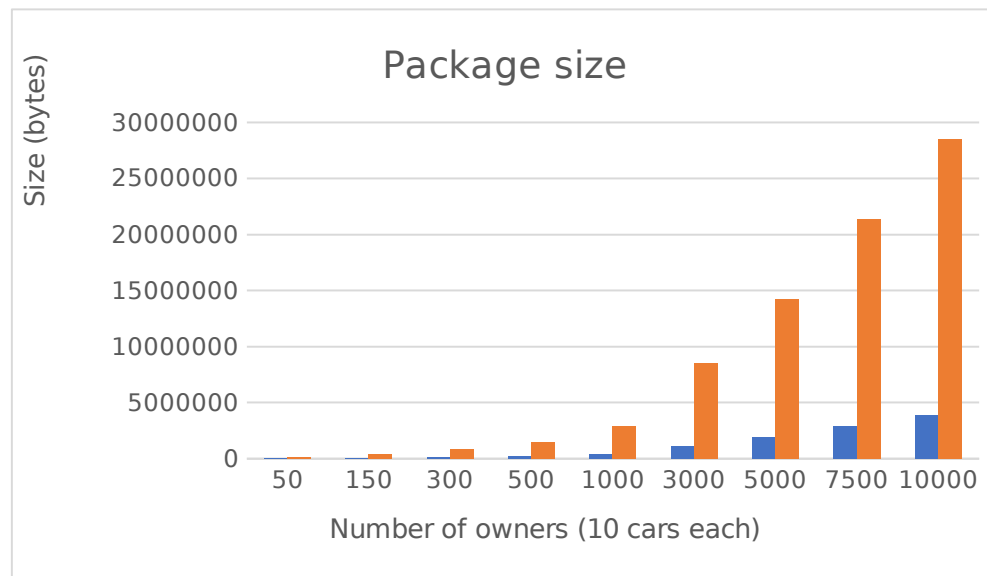


Fig.8 – Package size comparison of Protobuffer and XML

# Conclusion

From the graphs (Fig 7. and Fig 8.) we are able to visualize how badly XML performs compared to protocol buffer. In regards to Marshaling/Unmarshaling times, for a low amount of objects, the results are similar. However, as the number of owners increases we can easily see that XML gets exponentially bigger in size and takes longer to send.

This is because XML needs to be transformed to a string before being sent (which already is an extra step) and it is also human readable, which means it needs more information than protobuf (which is not human readable). The difference is only visible with bigger requests, clarified via the charts.

## Advantages

The combination of the results and our experience with both serialization methods, lead us to the following advantages of protocol buffer over XML:

- Protocol Buffer is Simpler to implement since most classes are generated automatically
- It is 5 to 10 times smaller in size
- For bigger packages, it is at least 10 times faster