

Timing Results and Analysis

After implementing servers and clients for both gRPC and XML-RPC, gRPC was a bit more complex. With gRPC, the server expected 1 gRPC message, and it sent 1 gRPC message at the end of the method. Since we had helper functions on our server side, the gRPC methods had to do some extra work to be able to deal with or even send data back to the client. With XML-RPC, we could just call one of our helper functions and send the data right back if we wanted to, along with some additional checks for return values, etc. However, we made the XML-RPC server/client a bit more user friendly and had nicer output on the client side.

XML-RPC was definitely slower between the two, in terms of the time it took from when a RPC was called and when the response was returned back to the client. We created a program to test the speed of each of the RPCs for both gRPC and XML-RPC, and recorded the median times for each iteration. After 3, we found the median times of our recorded times and compared the results. The main one that stuck out was getting all products from the server, which resulted in a **0.208445959 second** difference. This was the largest difference between the same RPC that was called between the two frameworks. Then, the smallest difference between an RPC called on both was creating a new order, which resulted in only a **0.0651542174 second** difference. Overall, gRPC always had the faster times.

If we were to use one of these frameworks for a commercial application, we would use gRPC, primarily due to its speed over XML-RPC. In addition, you are able to be more explicit with what data you are expecting to receive and send on both the server and client sides, which we could have even optimized further for quicker speeds for this project. Attached on the second page are the median timing results for each RPC on both gRPC and XML-RPC, along with the calculations for time differences.

Median times by RPC call (gRPC):

Add Product: 0.0297302450000001
Get Product: 0.029912351000000115
Manufacturer: 0.00028159000000016476
Update Product: 0.0315410044999993
Get In-stock: 0.00024622600000001604
Get All: 0.0002392595000002551
Create Order: 0.030247275499999837
Get Order: 0.030177553000000135
Amend Order: 0.03060449100000029
Unpaid/Unshipped: 0.00025883449999852814

Median times by RPC call (XML-RPC):

Add Product: 0.06855864249999999
Get Product: 0.06809160900000011
Manufacturer: 0.07155132299999956
Update Product: 0.06525908799999991
Get In-stock: 0.11989150299999984
Get All: 0.20844595999999882
Create Order: 0.06515421749999994
Get Order: 0.06960150399999954
Amend Order: 0.06720605349999964
Unpaid/Unshipped: 0.06840443199999946

XMLrGPC

Add Product: 0.06855864249999999 - 0.0297302450000001 = 0.0685586424

Get Product: 0.06809160900000011 - 0.029912351000000115 = 0.068091609

Manufacturer: 0.07155132299999956 - 0.00028159000000016476 = 0.0715513229

Update Pr: 0.06525908799999991 - 0.0315410044999993 = 0.0652590879

Get In-stock: 0.11989150299999984 - 0.00024622600000001604 = 0.119891502

Get All: 0.20844595999999882 - 0.0002392595000002551 = 0.208445959

Create Order: 0.06515421749999994 - 0.030247275499999837 = 0.0651542174

Get Order: 0.06960150399999954 - 0.030177553000000135 = 0.0696015039

Amend Order: 0.06720605349999964 - 0.03060449100000029 = 0.0672060534

Unpaid/Unshipped: 0.06840443199999946 - 0.00025883449999852814 = 0.0684044319