

# Distributed Systems (CSE431) Monsoon 2016

## Assignment-1

**Due: 25<sup>th</sup> August, 9 pm**

Task: To write programs and perform comparison between Json and Google Protobuf for serialization and deserialization of given set of records. Comparison is to be done with respect to time and rates of serialization/deserialization.

Input Record Format:

<Name>,<RollNo>:<Course1>,<Marks1>:<Course2>,<Marks2>:....:<CourseN>,<MarksN>

***Example:***

Bhavesh,201301116:cs210,85:cs155,94

Dhruva, 201301151:cs210,20:cs155,47

### **1. Json format:**

Convert the given records to json format as shown in result.json (sample attached). An illustration for the input records above is as shown:

```
[{
  "Name": "Bhavesh", "CourseMarks": [{
    "CourseScore": 85, "CourseName": "cs125"
  }, {
    "CourseScore": 94, "CourseName": "cs210"
  }],
  "RollNo": 165
}, {
  "Name": "Dhruva", "CourseMarks": [{
    "CourseScore": 20, "CourseName": "cs125"
  }, {
```

```
        "CourseScore": 37, "CourseName": "cs210"
    },
    "RollNo": 169
}]
```

***Naming convention for files:***

Serialize the records and write it to file: **result.json**

Deserialize the records back to the original format (as in input) to the file: **output\_json.txt**

## **2. Google protobuf format:**

Convert the records to the given Google protobuf format (**Result.proto**)

***Naming convention for files:***

Serialize the records and write it to file: **result\_protobuf**

Deserialize the records back to the original format (as in input) to the file: **output\_protobuf.txt**

***Useful Links:***

- Follow the instructions and tutorials given here for installing and understanding google protobuf:  
<https://developers.google.com/protocolbuffers/?hl=en>
- Use this for validating result.json: <http://jsonlint.com/>

***Submission Format:***

- Put all your code in a folder "Assignment1"
- Write a script "run.sh" that uses following options/flags:
  - -c : to compile the code
  - -s -j <INPUT\_FILE> : to serialize the given input records to json format and write it to "result.json"
  - -s -p <INPUT\_FILE> : to serialize the given input records to protobuf format and write it to "result\_protobuf"
  - -d -j <JSON\_FILE> : to deserialize the given json file and write plaintext records to "output\_json.txt"

- -d -p <PROTOBUF\_FILE> : to deserialize the given protobuf file and write plaintext records to "output\_protobuf.txt"
- -t -j <INPUT\_FILE> : to perform metric measurement (time/rate) on the given input file with json as intermediate format and print it.
- -t -p <INPUT\_FILE> : to perform metric measurement (time/rate) on the given input file with protobuf as intermediate format and print it.
- Submit archive as "Assignment1.tar.gz".

### Key Points:

- Languages allowed: Java/C++/Python
- Do not add any additional special characters or white spaces while deserializing (result.json) or formatting. 'output\_json.txt' and 'output\_protobuf.txt' should match exactly with the input file provided.
- Time should be measured in milliseconds (ms)
- Along with total time, print rate of serialization/deserialization e.g: 10Kbps
- Rate of Serialization/Deserialization is defined :  $\text{Amount of Data Converted} / \text{Time taken for conversion}$
- The Time taken for conversion need not include the Time taken for File I/O operations.
- You might want to take a look at Ordered Dictionaries to match the output format to the one given for comparison
- Result.proto, a sample input file (sample) and its corresponding result.json is attached for reference.
- **Plagiarism will be penalised.**