

Assignment 1

Write client-server programs that read/write records from/to a server. Each team (same as CUDA project team) member is supposed to write a program that uses one of the following communication methods – the members of a team must write programs that use different methods:

- ✓ Socket
- ✓ RPC (Remote Procedure Call)
- ✓ RMI (Remote Method Invocation)

Measure the mean response time for each method, and compare your results with the other team members'. Do experiments for all combinations of the following dimensions:

- ✓ Server Load (the number of client processes with each running on a different machine)
 - 1 and 2 (Mandatory); 4 and 8 (Optional)
- ✓ Access Operation Type
 - Read and Write
- ✓ Database File Access Pattern
 - Random and Sequential

For this comparison, draw a line chart for each combination of Access Operation Type and Database File Access Pattern (with a line for each communication method) - use "Server Load" as the x-axis title and "Mean Response Time" as the y-axis title. Also, provide a table that shows the mean response times and the standard deviations in each case.

About Submission

- Tar/zip the files that contain the source code (with each author's name on the corresponding part) and a team report
- Name the resulting tar/zip file as "DIP17_HW1_TeamNo.[zip|tar];" e.g, "DIP17_HW1_1.zip."
- **Submit this file to ETL until 11:59 pm on October 31**

4541.662A Distributed Information Processing (2017 Fall)

- ✓ Code
 - Indent and comment the source code
 - Make the code compilable and runnable
- ✓ Report - make brief, clear statements
 - Explain the experiment environments
 - Mention parameter values used in the experiments
 - Discuss comparison results by using charts and tables
 - Explain the results by using the corresponding pseudo code
 - Conclude the report
 - Writing in Korean is ok, but writing in English is preferred.

Use the provided programs to generate database-record and client-access-trace files – default parameter values are defined in the code.

- ✓ **dbgen** for generating a database-record file

```
$ ./dbgen -h

Usage:  ./dbgen  -o <output_file_name>  -rtn <record_num>  -dsz
<data_size>
        -rtn <record_num> : total number of db records
```

- The resulting file contains records, each consisting of the key and data fields. A key is of 4-byte integer, and data is of 1024-byte character, by default.

- ✓ **tracegen** for generating a client-access-trace file

```
$ ./tracegen -h

Usage:  ./tracegen  -o <outputfile> [-s|-r] -rtn <record_num>  -dsz
<data_size> -rn <req_num>
        -s : sequential access to db
        -r : random access to db
        -rtn <record_num> : total number of db records
        -dsz <data_size> : size of record
        -rn <req_num> : number of requests
```

- ※ You may download these programs from the link under this HW instruction file
- ※ (Compressed file name: as1_util.tar.gz)