About Logging, Repository and External Libraries

ASL Exercise Session

Week 3

Repository Questions

- GitLab managed by ISG, not the ASL team
  - While there is no size limit at the moment, ISG can manage storage at their will
  - We checked with them and **up to ~1GB** you should have no issues
- Submit <u>only the logs referenced in your report </u>+ compress if necessary
- You are responsible that all the data is submitted by the time of the deadline

  (The less data to move, less things to go wrong)

Don’t forget to instrument the code!

- The middleware code needs to be instrumented for each request to collect (at least) the following metrics:
  - $\boldsymbol{T_{mw}}$: Time spent between receiving a request and sending back the response.
  - $\boldsymbol{T_{queue}}$: Time each request spends in a queue (Time from enqueue to time of dequeue).
  - $\boldsymbol{T_{server}}$: Time each request spend in the server (time from sending request to the server to time of receiving server response).
  - $\boldsymbol{F_{success}}$: A Flag indicating if the request succeeded or not.

Measurements with Sampling

- ***NOTE:*** As logging *each request* will produce very large log files, we recommend you to sample your measurements:
  - Log only *ONE Request out of N request*.

    Where N ≤ 100. (Recommended N = 100).
  - Sample reads and writes separately (i.e. sample every 100th read and every 100th write operation).

How to produce logs?

- One design option (could be done differently):
  - For every request ***you need a data structure*** to store the client socket so you can send the response back to the correct client.
  - Add to this data structure the ***logging information*** (i.e. ***events time stamps***).
  - The thread which sends the response back to the client should log the measurements collected for that request.

Logging: simple does it

- Simplest way is to use Java logging utility:

  https://docs.oracle.com/javase/8/docs/technotes/guides/logging/overview.html

- Your logging needs are simple.
  - Create a logger once:

```
Logger myLogger = Logger.getLogger(String name);
myLogger.setLevel(INFO);
myLogger.addHandler( new FileHandler(“readsData.log”) );
```

  - Create string message

```
final String logMsg = String.format(“%d, %d”, time1, time2);
```

  - Log the message

```
myLogger.info(logMsg);
```

External Libraries

- You are allowed to use Log4j, if you decide not to use java.util.logging.
- BUT at your own responsability:
  - YOU NEED to change the Ant build.xml file in your project to include the log4j library.
  - Add the log4j jar file to your project repository so we can build your project and test it.
  - Use this link for further details on how to use log4j

http://www.codejava.net/coding/how-to-configure-log4j-as-logging-mechanism-in-java