STM Project

Your task is to implement a (simple) software transactional memory system (STM). Your goal is to out-perform the (simplest-possible) STM implementation given in: `https://github.com/LPD-EPFL/ca15-stm`.[1]

You are free to choose, or devise the algorithm to implement. The following papers describe the concept of STM, plus certain simple STM algorithms. You can find the papers in the `readers/` sub-folder of the repository.

1. Software transactional memory

2. Software Transactional Memory for Dynamic-Sized Data Structures

3. Dynamic Performance Tuning of Word-Based Software Transactional Memory

4. Transactional Locking II

5. NOrec: Streamlining STM by Abolishing Ownership Records

You need to implement your solution by modifying the skeleton code provided in the github repository (i.e., do not create a project from scratch)

1 Deliverables

You need to email your solutions to the TAs. Use the following title for your submission: "CA15 STM".

You will submit a pdf report named `ca15_report_Lastname.pdf` (replace `Lastname` with your family name). The report must contain (i) a detailed description of the algorithm you implemented, (ii) graphs from evaluating your STM using the provided script (read the README file of the project), and (iii) a thorough explanation of the results you obtain (i.e., why does it scale or not). Please include your sciper number in the header of the report.

Additionally, you need to submit your *working* STM implementation as a zip file called `ca15_code_Lastname.zip`. **Do NOT change the folder structure of the skeleton code provided.**

2 Grading

You can get up to 1 bonus point. The bonus points you get will be calculated as the: $min(1, speedup)$, where $speedup$ is the performance speedup your implementation achieves over the provided baseline implementation on the workloads of the `benchmark.sh` script (see the README file of the project).

In other words, if your STM is 30% faster than the baseline on average, you will get 0.3 bonus points. Additionally, the students with the three fastest implementations will present their solutions in one of the seminars.

Notice! You need a correct implementation to get the bonus points.

3 Deadline – Updated!

Sunday, ~~December 13 2015, 23:59~~ January 15 2016, 23:59 – *strict!*

---

[1]Instructions on how to clone a git repository can be found: `https://help.github.com/articles/cloning-a-repository/`

4 Copying

There are many more resources out there. However, there is no tolerance for academic dishonesty. Please refer to the University Policy on cheating and plagiarism. Discussion and group studies are encouraged. However, all submitted material must be the student's individual work.

Example behavior that is considered academic dishonesty:

- Writing code together;

- Copying code from any online resources or previous solutions.

You can talk to the professor or TAs for clarification if you have any questions.
Notice! You might be called by the professor or the TAs to explain your code.