Project Report

CONCURRENCY CONTROL: IMPLEMENTING A TRANSACTION MANAGER

Sujay Natarajan
Harsha Kosuru

Overall Status:

Project successfully completes all the implementations zgt_tx and zgt_tm methods. There are no unfinished portions of the project. The code gives detailed comments about the functionality and logic. In addition this report contains some description about how the project was implemented and what hurdles we faced while working on it.

File Descriptions:

There are no additional files created for this project. All the files that came along with the project's package are unchanged.

Division of Labor:

All parts of this project was implemented under the equal efforts of both of us partners. There are no portions of the project that we implemented individually as each of us has gone over the algorithm and implemented the same code simultaneously on the same machine. To finish this project we spent around 3 weeks.

Logical Errors and solutions:

1. The first logical error we made was to create a node structure in the begintx and in many other methods where necessary. After we looked over the remaining method bodies and going through the new description file about each method that was uploaded on

blackboard we figured out that we need to have a node structure in begintx and in many other methods.

2. The second logical error we encountered was to add pthread_create function where necessary. This caused our tx file to not be able to access the required variables from tm object. We fixed this by going through the tm's BeginTx and understanding what pthread_create function actually does.
3. The third logical error we faced was the usage of zgt_Tx global variable instead of actually creating a new tx node. This caused our program to throw invalid output because there was never any tx created with tx->tid. We fixed this problem by creating new tx node where necessary and passing tx-> to get_tx method that returned a tx if the tx is already created.
4. Our fourth logical error was to mention add p and v operations where necessary which made our program hang. We didn't use this because we were using start_operation and finish_operation instead. Later when we found out that the latter operations are not the same as p and v operations, we fixed this problem by adding them.

Difficulties:

We encountered difficulties in understanding the semaphore logic, that is, how to lock and unlock the managers and y we had to do that but after the lectures were delivered we were able to understand the logic and the pdfs provided more clarity.