

TER n°10

Implementation and performance evaluation of multi-threaded Heap structures using transactionnal memory

Olivier Dalle

e-mail: olivier.dalle@inria.fr

Nombre d'étudiants souhaités: 2 à 4

Description du sujet

Heap structures are critical for the performance of many applications. One good exemple is Discrete-Event Simulation (DES), in which the events which represent the history of the system are stored randomly in such a structure during the simulation, but they have to be processed in in strict increasing order (of occurence time) to follow the chronology.

In order to speed-up and/or scale-up the execution of such DES, various algorithms have been proposed for distributing simulations on multiple computers and keep the global synchronisation, using message passing. The advent of many-core architctures opens new perspectives for the parallelization of such algorithms on multiple cores, using multiple threads and shared-memory.

The goal of this TER is to implement and evaluate the performance of such a multi-threaded heap algorithm based on Transactional Memory. Transactional Memory is recent technique proposed to replace the use of locks and mutual exclusion in concurrent algorithms running on a shared-memory architecture. As its name suggests, it borrows the idea of transactions to the database world : when a concurrent action is needed, a transaction is initiated in memory; if the action completes without conflict (with other hreads), the transaction is committed and the new memory state is kept; if the action generates a conflict, some of the conflicting transactions have to be rolled-back (ie. the new memory state is dropped) and restarted. This new technique was proposed a few years ago, but it was only virtually available by means of software emulation. Major actors, such as Intel and IBM have recently started to build or announce hardware support for Transactional Memory in their latests products (Eg. IBM BlueGene/Q super computer already has it[1], and the next generation of Intel Processors is announced with an extension to instruction set for the support of TM[2]).

Work to do: The goal of this TER is to investigate the use of such a TM software emulation library to implement a multi-threaded Heap data structure. The library we chose is TBoost.STM[3,4], a proposed extension to the Boost C++ library.

1. implement or retrieve various multi-threaded heap data-structures in C++ using algorithms:
 - without transactional memory , as found in the literature
 - with transactional memory, custom designed
2. run performance comparisons between the various implementations
 - Build a performance benchmark
 - Run experiments on a multi-core computer

Lieu

Prérequis

C/C++ programming, experience with Posix threads and concurrent programming (locks, semaphores)

Informations complémentaires

Reference: [1] Peter Bright. "IBM's new transactional memory: make-or-break time for multithreaded revolution." ARS Technica, Aug 31 2011.

<http://arstechnica.com/gadgets/2011/08/ibms-new-transactional-memory-make-or-break-time-for-multithreaded-revolution/>

[2] Peter Bright. "Transactional memory going mainstream with Intel Haswell." ARS Technica, Feb 2012.

<http://arstechnica.com/business/2012/02/transactional-memory-going-mainstream-with-intel-haswell/>