Impact of memory allocation on the performance of a delegation synchronization algorithm

Written by Riyane SID-LAKHDAR (M1 MoSIG)
Supervised by Thomas ROPARS (LIG, team ERODS)

May 09, 2016

Abstract

The main objective of our research is to study the impact of the dynamic memory allocation on multithreaded application. We also aim extend the scope of the "critical section" algorithm described within the publication ***"On the Performance of Delegation over Cache-Coherent Shared Memory"***[1]

During the previous period (part-time), we have accomplished a theoretical study of some existing memory allocation strategies. We have described their architectures, and highlighted their advantages and drawbacks for multithreading purposes.

This document describes the objective and the work plan of the full-time period. It may refer to this previous part of the research.

[1] Published by Darko Petrovic, Thomas Ropars and Andre Schiper, from the "Ecole Polytechnique Federale de Lausanne" (EPFL), Switzerland 2014

1 Objective of the full time period

Thanks to the theoretical study previously accomplished, we have been able to produce a "state of the art" of different allocation strategies implemented within existing (and heterogeneous)) memory allocators.
Our first objective is to determine a set of metrics that allows to compare all the studied memory allocators for multithreading purposes. This metrics should help to establish a hierarchy between these different allocators (within a scope that we have to determine).

Second, we aim to have a practical approach of these implemented strategies. The objective is to run performance tests using each one of the studied allocator. These tests are designed to show the impact of the allocator on the performance of a given application. The purpose is also to show the impact of the hardware architecture and the kernel allocation policies on the performance of these allocators.

Using the most efficient allocator previously picked, we finally want to improve the accuracy of the experiments described in the publication previously mentioned1. Indeed, the authors of this publication have described a new critical section algorithm based on the "delegation" principle. The performance tests exposed in this paper have been realized using a *"fake allocator"* (ideal but non scalable). Our objective is to determine the impact of a *"real allocator"* on the efficiency of this algorithm.

2 Work plan

To achieve the previous objectives, our plan is to go through the following steps:

- Implement a set of custom test designed to stress the considered dynamic memory allocators within a given environment.
- Implement a platform designed to load the custom tests, link them to the specific allocator libraries, and run separately each test. The purpose of this platform is to run all the tests with respect to the specific "fairness" and "isolation" rules designed for the experiments (see section "Material and Methods" of the report).
  Thanks to this platform, we expect to run the custom **tests on different and heterogeneous machines (hardware and software)**. Thus, this platform needs to be portable, hence to manage the dynamic recompilation of the custom test as well as the allocator libraries. It also needs to recognize the host processor architecture (NUMA) in order to adapt the execution rules to the environment.
  Finally, this platform needs to assess the execution statistics and produce result files that could be accessible and reusable. Thus, it needs to dynamically and on remote access the different execution metrics in order to highlight and compare specific trends and behaviors.
- Analyze the experimental results and conclude about the impact of the hardware and software environment on the performance of the memory allocator. This analyze will also lead to confirm or infirm the improvement of the critical section algorithm previously described in the publication: ***"On the Performance of Delegation over Cache-Coherent Shared Memory"*** 1