Umeå University
Department of Computing Science

C Programming and Unix 7.5 p
5DV088

mfind

Submitted 2016-10-17
Author: Lorenz Gerber (dv15lgr@cs.umu.se lozger03@student.umu.se)
Instructor: Mikael Ränner / Filip Åberg / Jonathan Westin / Mattias Åsander

Contents

1 *mfind* and Thread Safety 1
2 Performance Assessment 1
References 1

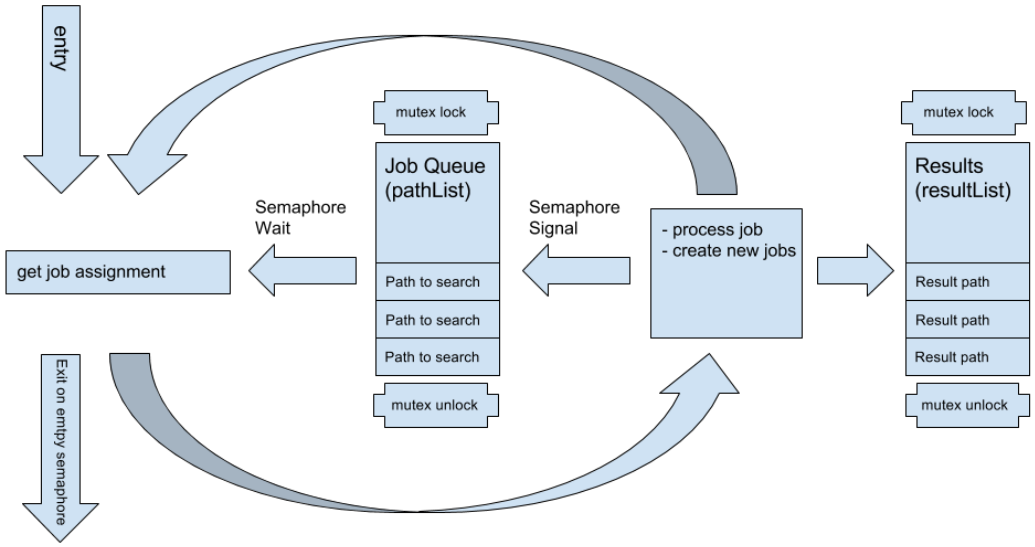

Figure 1: *This figure shows a schematic view of the mfind problem.*

1 *mfind* and Thread Safety

The scheme in figure 1 shows the most important aspects of the mfind system and how it was implemented here. It can be basically seen as a producer consumer system, however, each thread can be both producer and consumer. The most important source of synchronization is the semaphore which distributes the jobs to the threads. When there are no more jobs available. The 'do while' loop looks futile on the first view, however, new jobs can be added after passing the 'no wait' semaphore until checking the value of the semaphore in the 'while' expression.

The current construction is thread safe: The last thread in the loop can not leave until all jobs, also freshly self created are processed. In certain cases, this could lead to an uneven workload. This could be addressed by further synchronizing the threads, however this would also affect performance for most general cases.

2 Performance Assessment

2(2)

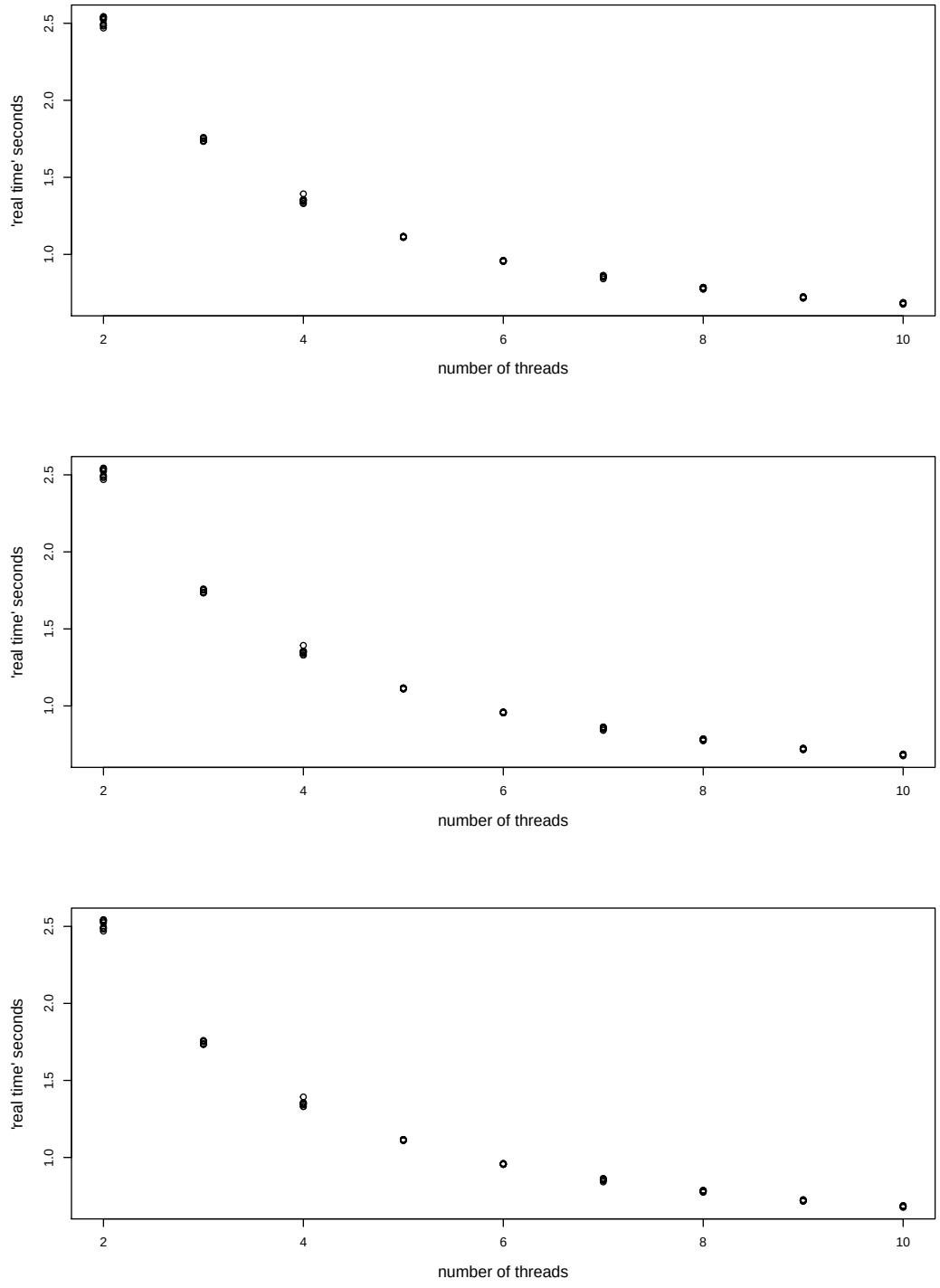

Figure 2: *This figure shows the execution time in relation to number of threads.*