Umeå University
Department of Computing Science

Parallel Programming 7.5 p
5DV152

Exercises, Chapter/Topic 1

Submitted 2017-01-26
Author: Lorenz Gerber (dv15lgr@cs.umu.se lozger03@student.umu.se)
Instructor: Lars Karlsson / Mikael Ränner

Contents

1 Introduction

This report is part of the mandatory coursework. It describes the solutions for several chosen exercises from the course book [**?**].

2 2.4 - Counting pages

2^{20}

3 2.8 - Hardware multithreading and caches

Caching operates on whole cache lines. Hence if the chance that another process/thread changes something in a specific cache line increases with cache size and number of processes/threads. The specific situation that can happen is called 'false sharing': When one process changes data in a cache line, there is no possibilty to check or know for another process which data exactly was changed. It can very well be the case that to the current process unrelated data was changed and a cache reload would not be needed. However, cache has to be reloaded.

4 2.10 - Communication overhead

The calculations can be split in two parts: the instructions and the communication. For both a) and b) the instructions will take the same time:
`instructions / cores / instructions_per_second`
Hence $10^{12} \div 1000 \div 10^6 = 1000sec$. For communications, the time is calculated as
`messages_to_send * time_per_message`
while messages to send is described as $10^9(p-1)$.

a) if sending one message takes 10^{-9} seconds, the communication with 1000 cores will take: $10^9(1000-1) \times 10^{-9} = 99'900sec$, together with the actual calculation, $100'900sec = 28.03h$.

b) if sending one message takes $^{-3}$ seconds , the communication with 1000 cores will take: $10^9(1000-1) \times 10^{-3} = 99'900'000'000sec$, together with the actual calculation, $99'900'001'000sec = 3'167.8years$.

5 2.16 - Speedup efficiency

6 2.19 - Scalability

7 2.20 - Linear speedup and strong scalability

8 2.23 - Alternative algorithm for computing histogram