Performance evaluation of parallel sorting algorithms on large data sets

Nicola Desogus, Paolo Giangrandi, Fabio Luporini

November 26, 2010

Contents

1 Introduction

This project has the aim of studying the well known problem of sorting *atomic* items (i.e. they occupy O(1) space) within the context of parallel computing. All the standard sequential sorting algorithms can be re-designed to exploit parallel machines; as a matter of fact, literature is not poor of this kind of works. There are some sequential algorithms (such as mergesort or quicksort) that can be easily adapted to the new parallel context; nevertheless, other ones needs a trickier re-engineering. On the other hand, algorithms that performs very well on sequential machines could not be so good in their new parallel version. Thus, the first objective of this project is to sperimentally compare the performance of the most known parallel sorting algorithms.

In this scenario things get even more complicated from the fact that we have to manage very large data sets, i.e. our parallel algorithms sort terabytes of data. This means that some of the theoretical results achieved in the field of *external memory sorting* will have to be compared with the ones that we will sperimentally obtain. Further, in order to keep implementation complexity limitated, we decide to not extends our algorithms for exploiting the presence of multi-disks eventually provided by the parallel architecture.

In this complicated context the role of the test environment becomes crucial. Obvioulsy, we can not say neither that our algorithms will scale the same way on every possible parallel machine nor that results are meaningful only for a specifc machine. Hence, analyzing the results, we will have to consider even some fundamental architectural aspects. At a first glance, we have to take care of at least two macroscopic aspects: first, the possibility that the parallel machine is a hierarchycal systems (e.g. a cluster of shared memory nodes); second, the interconnection network of the nodes. In case of a hierarchycal system, if we will be able to find a way for exploiting the presence of shared memory, then we might achieve a very significative gain in terms of performance. Hence, depending both on the way we will exploit shared memory and the properties of the interconnection structure (bandwith, latency), we might get performance results more or less close to the ones we expected.

We will address all this issues both from a theoretical point of view and by implementing a structured framework. This document is organized as follows: first, we briefly explain which algorithms will be parallelized; second, we describe all the characteristcs of our framework; then, after having described the test environment, we will detail the implementation of our algorithms and, for each of them, we will analyze their performance. Finally, we will compare all the achieved results.