SPM Project: BSP

Lorenzo Bellomo, 531423

AA 2018/2019

1 Introduction

In this report the *Bulk Synchronous Pattern* (BSP) will be analysed both from a theoretical viewpoint and from a practical one.
The outline of the report will be the following:

- Section 2 (Parallel Architecture Design)
- Section 3 (Performance Modeling)
- Section 4 (Implementation Structure and Details)
- Section 5 (Experimental Evaluation)

2 Parallel Architecture Design

BSP is by nature a parallel bridging model. The main idea is that the process is divided in three phases (as shown in figure 1):

- *Local computation*: The processors act independently and compute their task on a different partition of the input data.
- *Communication*: In this phase processors are allowed to send data to the other ones. It is important to notice that phase 1 and 2 (super step and communication) may overlap in case of uneven input workload.
- *Synchronization*: In this phase all the processors synchronize and wait that the communication phase is over for each worker (*barrier*).

This whole process made of 3 phases composes a *super step*.

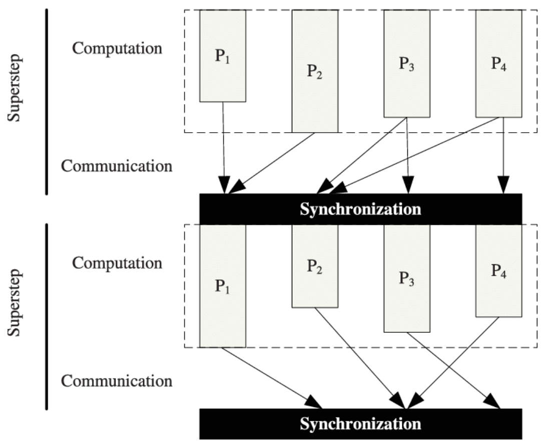

Figure 1: BSP structure

3 Performance Modeling

In order to theoretically model the performances of a BSP algorithm some values are going to be defined:

- nw is the number of workers (BSP processors).
- k is the number of super steps
- w_i is the time spent by the *i-th* worker in the compute phase.
- m_i is the number of maximum messages receiver by the *i-th* worker.
- c is the cost of sending or receiving one message.
- B is the cost of the synchronization step (*barrier*).
- I is the initialization process cost.
- F is the finalization process cost

The cost of a super step (SS) can be approximated (upper bounded) as follows:

$$SS_j = \max_{i=1}^{nw} w_i + \max_{i=1}^{nw} cm_i + B$$

Which eventually gives:

$$tot\_cost = I + \sum_{j=1}^{k} SS_j + F$$

3.1 Opportunities for parallelism

According to the model provided, the opportunities for parallelization reside in the *super step cost*, which ideally should decrease linearly by increasing the number of parallel activities. In particular, it is safe to assume that $w_i = \Theta(n/nw)\,\forall i$, while instead both the communication cost and the barrier cost are pure overhead, which grows with nw.
In this particular problem, the parallel design is already provided, but the goal of the implementation is to minimize the overheads, with particular focus on the *communication* one, which can be assumed to be the most costly.
For what regards the user provided business logic code, which regards both the computation phase and the communication one, the scalability of the provided *BSP* depends on the goodness of the business logic code.

4 Implementation Structure and Details

5 Experimental Evaluation