Chapter 4

Automatic code generation of SPMD parallel programs

Previous studies have shown that CTADEL can generate highly optimized serial code for several schemes in a weather forecast model. In this chapter we show how to extend CTADEL so that it can automatically generate SPMD parallel programs. We applied this technique to generate parallel code for the Shallow-Water equations which have been discussed in Chapter 2 and performed experiments on the DAS-3 system [67].

4.1 Parallel programming models

Many parallel programming models have been developed. In general, popular models which are used in many parallel programs include Master-Slave, Data Pipelining, Divide and Conquer, and Single Program Multiple Data (SPMD). These models are presented in detail in [11]. In the following paragraphs, we give a brief description of each.

Master-Slave

The master-slave model consists of a master process and multiple slave processes. The master decomposes the problem into small tasks, distributes these tasks to slave processes, and gathers and combines the results from slave processes to a final result. The slave receives a task, executes it, and sends the result to the master. The required communication is only between the master and the slaves. This model is suitable for applications where a sequential algorithm can be executed simultaneously and independently on different processes, each with different input data. This model is also suitable if there are several different algorithms, perhaps even all operating on the same input data.

Data Pipelining

In the data pipelining model, a number of processes form a virtual pipeline. A continuous data stream is fed into the pipeline, and the processes execute at different pipeline stages simultaneously in an overlapped fashion. This model is usually applied in data reduction or image processing applications.

Divide and Conquer

In the divide and conquer approach, a problem is split into several subproblems, where each is solved independently and their results are combined to give the final result. Different from the master-slave approach, where the communication is needed between the master and the slaves, the divide and conquer method does not require any communication between the processes because the subproblems in this approach are independent.

Single Program Multiple Data

The Single Program Multiple Data model (SPMD) is the most widely used method [4]. Within a SPMD program, each process executes the same code but on a different part of data. This involves the splitting of application data among the available processes. This type of parallelism is also referred to as geometric parallelism, domain decomposition, or data parallelism. SPMD applications can be very efficient if data are well distributed over processes and the problem is homogeneous.

The choice of a parallel model depends on the type of parallelism inherent in the problem, which reflects the structure of either the application or its data. We chose the SPMD parallel programming model as base for our parallel code generation, because of the following reasons:

- SPMD is widely used for the parallelization of scientific models [4].

- SPMD works particularly well for problems which can be partitioned into static chunks which interact only with their nearest neighbors. Atmospheric models, for example, tend to be suitable for this type of parallelization [5].

- Because data is split among the processes, the SPMD model is well-suited for platform that has a distributed memory. Hence, a grid platform is also suitable for the SPMD model.