Minimizing Synchronization in Parallel Nested Loops

Reza Rafeh, Centre for Business, Information Technology and Enterprise, Waikato Institute of Technology, Hamilton, New Zealand, Reza.Rafeh@wintec.ac.nz

Mohammad Hossein Roosta, Department of Computer Engineering, Islamic Azad University, Malayer Branch, Malayer, Iran

Abstract

Although computing power increases continuously, the need for high computational speed in many scientific applications is growing too. As a result, implementation of parallel applications has gained more attention. Since nested loops are the most time-consuming parts of most programs, we propose a method for scheduling uniform nested loops to processors based on the equation of a straight line which includes the maximum possible number of dependence vectors. Experimental results show that the proposed method imposes a lower communication between processors compared with similar methods.

Background

Parallel processing is a form of computing in which many instructions are executed simultaneously. The process of parallelization in general consists of three steps as follows:

- Decomposing the application into tasks
- Analyzing the dependencies between the decomposed tasks
- Scheduling tasks into the target parallel or distributed system

Generally, there are two types of dependency:

- Data dependencies
- Control dependencies

There are two types of nested loops based on dependencies:

- DOALL loops: Nested loops with no dependency.
- DOACROSS loops: Nested loops with dependencies which are divided into two categories:
  - Uniform: A loop in which the pattern of dependencies remains constant during its execution.
  - Non-uniform: A loop in which the pattern of dependencies may have variations during execution.

Proposed Algorithm

Methods

```c
for(int i=k; i<N; i++)
for(int j=1; j<M; j++){
    A[i][j] = 5 * B[i-1][j-3];
    A[i][j] = A[i][j] + 4*B[i-4][j-3];
}
```

Proposed Algorithm

Impact: Assumption which is satisfied with the points of set $S := \{(x_0, y_0), \ldots, (x_N, y_N)\}$

- Compute: the equation of the best straight line (BSLS).

```
BSLS = \text{Find the equation with the largest coefficient}
```

Results

We compared our algorithm (BSLS) with Chain Pattern Scheduling (CSP) method in terms of communication.

Conclusions

We proposed a new algorithm, named BSLS, to reduce the communication cost of uniform nested loops which facilitates parallelizing such loops. To enhance data locality, BSLS finds the best straight line which encompasses the maximum number of dependence vectors. Chains are considered as lines parallel with the best straight line in the iteration space. Chains are assigned to processors periodically. Our experimental results show that BSLS imposes lower communication cost than Chain Pattern Scheduling (CSP).

References