Назад к неделе 1 Уроки этот курс: Parallel Programming in Java Пред. Дальше

1 Task-level Parallelism

1.3 Computation Graphs, Work, Span, Ideal Parallelism

Lecture Summary: In this lecture, we learned about Computation Graphs (CGs), which model the execution of a parallel program as a partially ordered set. Specifically, a CG consists of:

- A set of *vertices* or *nodes*, in which each node represents a *step* consisting of an arbitrary sequential computation.
- A set of *directed edges* that represent ordering constraints among steps.

For *fork–join* programs, it is useful to partition the edges into three cases:

1. *Continue* edges that capture sequencing of steps within a task.

2. *Fork* edges that connect a fork operation to the first step of child tasks.

3. *Join* edges that connect the last step of a task to all *join* operations on that task.

CGs can be used to define *data races*, an important class of bugs in parallel programs. We say that a data race occurs on location *L* in a computation graph, *G*, if there exist steps *S*1 and *S*2 in *G* such that there is no path of directed edges from *S*1 to *S*2 or from *S*2 to *S*1 in *G*, and both *S*1 and *S*2 read or write *L* (with at least one of the accesses being a write, since two parallel reads do not pose a problem).

CGs can also be used to reason about the *ideal parallelism* of a parallel program as follows:

- Define *WORK(G)* to be the sum of the execution times of all nodes in CG *G*,
- Define *SPAN(G)* to be the length of a longest path in *G*, when adding up the execution times of all nodes in the path. The longest paths are known as *critical paths*, so *SPAN* also represents the *critical path length* (CPL) of *G*.

Given the above definitions of *WORK* and *SPAN*, we define the *ideal parallelism* of Computation Graph *G* as the ratio, *WORK(G)/SPAN(G)*. The ideal parallelism is an upper limit on the speedup factor that can be obtained from parallel execution of nodes in computation graph *G*. Note that ideal parallelism is only a function of the parallel program, and does not depend on the actual parallelism available in a physical computer.

Optional Reading:

1. Wikipedia article on Analysis of parallel algorithms

Пометить как выполненное