Cilk: An Efficient Multithreaded Runtime System

Cilk is an algorithmic multithreaded runtime system which is based on the principle that a programmer should concentrate on structuring the program to expose parallelism and locality without worrying about scheduling the computation to run efficiently on a given hardware platform. The Cilk runtime system takes care of details like scheduling, load balancing, and communication protocols guaranteeing efficient and predictable performance.

A Cilk computation unfolds as a directed acyclic graph (DAG) composed of a collection of tasks. Each task is decomposed to a set of subtasks up to a desired level where the tasks represent a basic unit of execution that could produce some output. Each task is executed as a non-blocking thread, which means that it can run to completion without waiting or suspending once it has been invoked. Since threads cannot block in the Cilk model, a thread must additionally spawn a successor thread to receive the children's return values when they are produced. Return values, and other values sent from one thread to another, induce data dependencies among the threads, where a thread receiving a value cannot begin until another thread sends the value.

The authors present various data structures to represent the components of the DAG. They define a data structure called "closure" that represents the tasks in the DAG. It consists of a slot for each of the arguments that it waits for (if it does), and a join counter indicating the number of missing arguments that need to be supplied before the task is ready to run. To run a ready closure, the Cilk scheduler invokes the thread using the closure itself as its sole argument. The system supports a strategy to facilitate communication between threads called as explicit continuation passing, wherein a continuation is essentially a global reference to an empty argument slot of a closure, implemented as a compound data structure containing a pointer to a closure and an offset that designates one of the closure's argument slots.

Then the authors talk about the "work-stealing scheduler" that is used in Cilk. Each processor maintains a local ready queue which is an array in which the Lth element contains a linked list of all ready closures having level L where L corresponds to the number of spawns on the path from the root to the spawn tree. If the scheduler tries to remove a thread from an empty ready queue, the processor becomes a "work-stealer" and extracts a thread from the non-empty queue of one of the other processors.

The efficiency of the Cilk scheduler has been demonstrated both empirically and analytically. Applications include 'N queens' backtrack search program, protein folding, graphic rendering and the Socrates chess program, which won third prize in the 1994 ACM International Computer Chess Championship. But the system has some shortcomings with respect to its inability to deal with traditional parallel applications that can be programmed effectively in, for example, a message-passing, data-parallel, or single-threaded, shared-memory style.