Assignment-3
SUMMARY OF CILK PAPER

Multi threading is a popular way to implement asynchronous and concurrent applications. Cilk is C based runtime system for multi threaded programming. It emphasizes the application programmer to think of work and critical path , abstractions which can be used to characterize the performance of an algorithm independent of the machine configuration.

The most important part of cilk is the work stealing scheduler. A program in cilk consists of a set of procedures which are broken into a sequence of threads which form the vertices of a dag each of which is non-blocking and runs to completion once they are started.

Cilk uses a explicit continuation passing form of programming where a parent thread spawns children and successors which wait for the children to complete , whereas the parents themselves complete execution. The most important data structure in cilk is a closure which holds a pointer to the thread, slots for specific arguments and a join counter. A closure does not execute unless its ready ie it has received all arguments and join counter is set to zero. Another important data type is a continuation which holds a pointer to a closure and a number representing the slot in the closure where the values should be written.

Cilk provides primitives to
- spawn a thread - thread T (args) {stmts }
- spawn a successor – spawn_next T(args)
- send argument values to the slot number mentioned in the continuation of the closures – send_argument(k, value)

Cilk uses the technique of work stealing where a processor who runs out of tasks selects another victim processor and steals the shallowest thread from the victims spawn tree. When a processor finds that its ready queue is empty it does the following steps
- Selects a victim processor at random
- Checks if the victims ready queue is empty. If its is go back to Step 1
- Steal the shallowest thread from the victims ready queue.

Work stealing is implemented with a request reply protocol between the processors. Always the shallowest thread is stolen for the reason that shallow threads have a lot of work to be done and stealing large work reduces the number of steals and the overhead of communication.

The paper describes the performance measurement of Cilk where they show relationships between efficiency and thread length and speedup and average paralleism. They show that for applications with longer threads have higher efficiency than ones with shorter threads. Their results show when the average parallelism is large the speedup is higher and when the parallelism is less speed up also is less.

The paper also proves a bound on Space, Time and Communication for the "fully strict" class of programs which are the programs which returns arguments only to the parents successor.