Introduction

- Today
- Moore's law
- Future
- Task-based parallelism
- About this presentation
Task-based Programming Model

- OpenMP
  - Gcc
  - Icc
  - Mcc
  - Sun CC
- Cilk++
- Wool
- Others (Intel TBB, Microsoft Task Parallel Library, X10 activities)

Task-generation (OpenMP)

```c
int fib(int n) {
    int x, y;
    if (n < 2)
        return n;
    else {
        #pragma omp task shared(x)
        x = fib(n-1);
        #pragma omp task shared(y)
        y = fib(n-2);
        #pragma omp taskwait
        return x + y;
    }
}
```
Task-Generation(Cilk++)

```c++
int fib(int n) {
    int x, y;
    if (n < 2)
        return n;
    else {
        x = cilk_spawn fib(n-1);
        y = cilk_spawn fib(n-2);
        cilk_sync;
        return x + y;
    }
}
```

Task-Generation(Wool)

```c++
TASK_1 (int, fib, int, n) {
    if (n < 2)
        return n;
    else {
        int x, y;
        SPAWN( fib, n-1 );
        y = CALL( fib, n-2 );
        x = SYNC( fib );
        return x + y;
    }
}
```
Methodology

- Two quad-core AMD64 system
- Program executed ten times.
- Measurement as compared to serial/gcc
- Median value used
- Compilation

Benchmarks

- Five benchmarks
- FFT, nQueens, Multisort, SparseLU, Strassen
- Barcelona OpenMP task suite
FFT

- Fast Fourier Transform
- Task-grainsize differs on vector before going serial (64k – 16)

nQueens

- Nqueen problem
- Task-grainsize differ on allowed task-depth (4 – 12)
Multisort

- Multisort benchmark
- Task-grainsize differ on list-size before going serial (512k – 16)

SparseLU

- SparseLU benchmark
Strassen

- Strassen benchmark
- Task-grainsize differ with sub-matrix before serial(256 – 16)

Conclusion

- Performance comparison
- Wool shows good promise
- OpenMP not yet complete for finegrained tasks.
Future work

- Identify and explain
- Possible further improvements

Questions