Hello Mirek

To help with parallel programming in U++ I have uploaded some benchmarks in new package
Bazaar/OpenMP_demp. I hope it will happen as positive NTL vs. STL comparison :)

About storing partial results, it could be interesting to see demo "Pi" at it uses reduction() clause to
handle temporary results between cores.

```
static double pi_device() {
 double x, y;
 long count = 0, i;

 // Parallel loop with reduction for calculating PI
 #pragma omp parallel for private(i, x, y) shared (samples) reduction(+:count)
 for (i = 0; i < samples; ++i) {
  x = Random(1000000)/1000000.;
  y = Random(1000000)/1000000.;
  if (sqrt(x*x + y*y) < 1)
   count++;
 }
 return 4.0 * count / samples;
}
```