How to do reductions

Dmitry Denisenko
Doing reductions in a highly parallel system often involves writing very complicated code, involving barriers and local memory.

Using Altera SDK for OpenCL you can write reductions in the natural way. The compiler will take care of generating efficient hardware by pipelining the loop:

```c
kernel void do_sum(global int *a, int n, global int *result) {
    int local_result = 0;
    for (int i = 0; i < n; i++) {
        local_result += a[i];
    }
    *result = local_result;
}
```
Check the Optimization Report (located in `<kernel>/`<kernel>.log) to verify that the loop is well pipelined.


Exercise:
- Change all data types from `int` to `float` in the previous example.
- What does the Optimization Report say for this kernel?
- Using the Best Practices Guide, fix the kernel to get optimal pipelining.