

```

struct square
{
    __host__ __device__
    float operator() (float x) const
    {
        return x *x;
    }
}

float snrm2_slow(const thrust::device_vector<float>& x)
{
    // without fusion
    device_vector<float> temp(x.size()); transform(x.begin(),
x.end(), temp.begin(), square());

    return sqrt( reduce(temp.begin(), temp.end()) );
}

float snrm2_fast(const thrust::device_vector<float>& x)
{
    // with fusion
    return sqrt( transform_reduce(x.begin(),x.end(),square(),0.0f,
plus<float>()) );
}

```