Synchrotron Radiation Benchmarking

Konstantinos Iliakis

PhD Candidate
CERN, CH - NTUA, GR
konstantinos.iliakis@cern.ch

Supervisors:

Dr. Helga Timko, CERN
Dr. Sotirios Xydis, NTUA
Dr. Dimitrios Soudris, NTUA

March 27, 2020

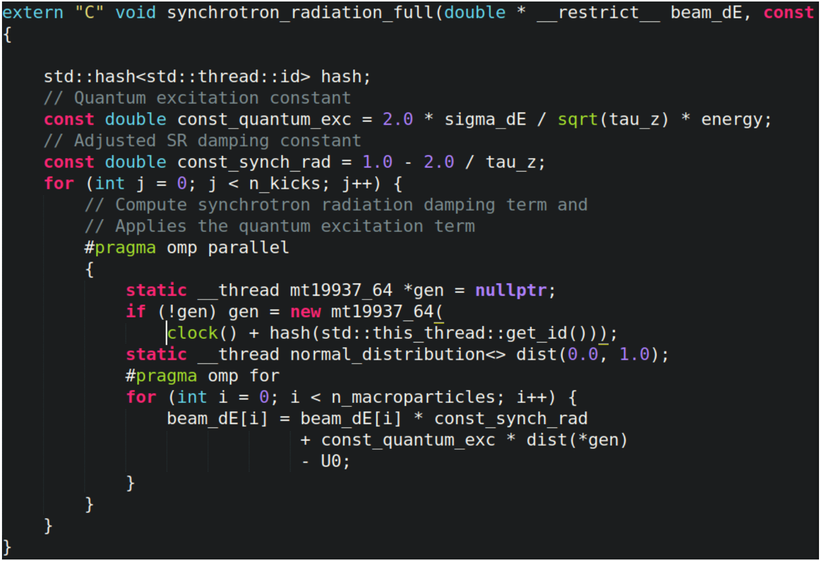
Synchrotron radiation (with quantum excitation) single-thread performance

Table 1: Synchrotron radiation implementations. 16M particles, single-thread, n_kicks=1, 100 iteratoins, 5 runs.

Description	**Run time (sec)**	**Speedup**
std, non-parallel dist gen, separate loops	70.24 ± 0.47	1.0×
boost, non-parallel dist gen, separate loops	19.41 ± 0.56	3.6×
std, parallel dist gen, single loop	65.00 ± 0.59	1.1×
boost, parallel dist gen, single loop	16.50 ± 0.63	4.3×

Bottom line: Use **Boost** if possible (**does not require installation**). Small performance gain by merging together all loops.

Best implementation, boost/std, single loop, scalable



```
extern "C" void synchrotron_radiation_full(double * __restrict__ beam_dE, const
{

    std::hash<std::thread::id> hash;
    // Quantum excitation constant
    const double const_quantum_exc = 2.0 * sigma_dE / sqrt(tau_z) * energy;
    // Adjusted SR damping constant
    const double const_synch_rad = 1.0 - 2.0 / tau_z;
    for (int j = 0; j < n_kicks; j++) {
        // Compute synchrotron radiation damping term and
        // Applies the quantum excitation term
        #pragma omp parallel
        {
            static __thread mt19937_64 *gen = nullptr;
            if (!gen) gen = new mt19937_64(
                clock() + hash(std::this_thread::get_id()));
            static __thread normal_distribution<> dist(0.0, 1.0);
            #pragma omp for
            for (int i = 0; i < n_macroparticles; i++) {
                beam_dE[i] = beam_dE[i] * const_synch_rad
                             + const_quantum_exc * dist(*gen)
                             - U0;
            }
        }
    }
}
```

Scalability testing

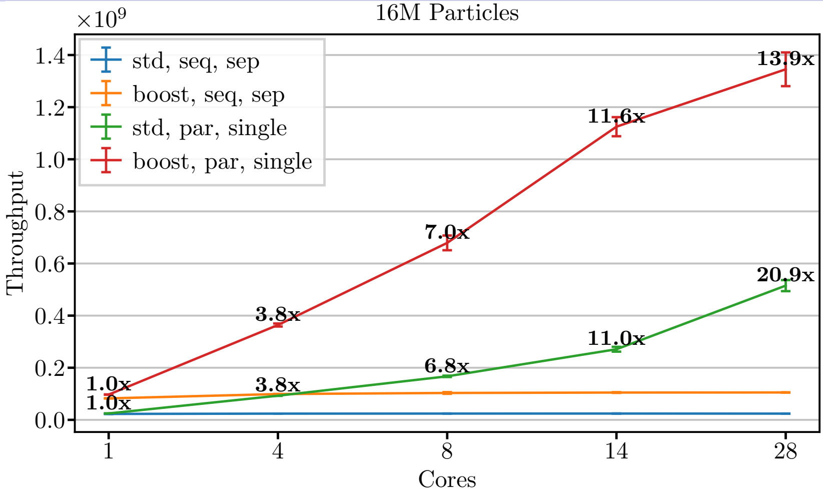
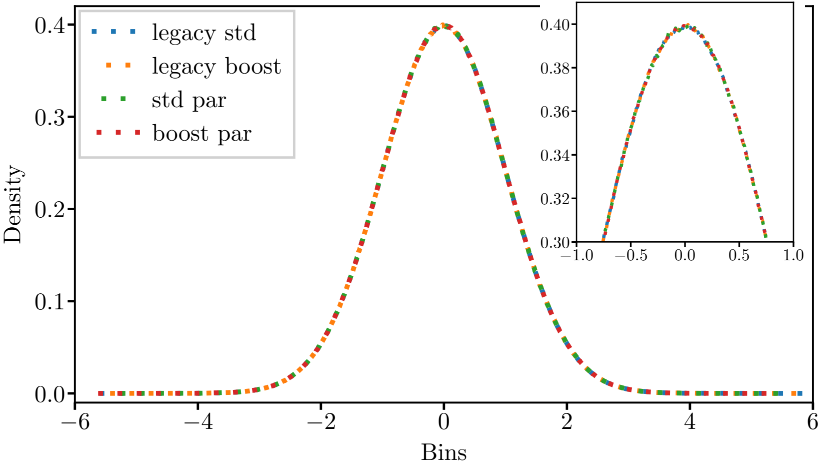


Figure 2: Scalability of the various synchrotron radiation implementations.

Bottom line: Old versions not scalable. New STD and Boost versions scale reasonably.

Normal distribution quality



Looks good to me. If needed it can be tested further.

Pull Request #164

- https://github.com/blond-admin/BLonD/pull/164
- Updated the std and boost implementations.
- Added unittests that cross compare the output of python and C++ synchrotron radiation tracking.

Questions

CERN

Microprocessors Laboratory MicroLAB

BLonD