# Particle-in-Cell Plasma Simulation on Intel Xeon Phi Coprocessors

S. Bastrakov[1], D. Durnov[3], A. Gonoskov[1,2], E. Efimenko[1,2], A. Korzhimanov[1,2], I. Meyerov[1], I. Surmin[1]

[1]Lobachevsky State University of Nizhni Novgorod

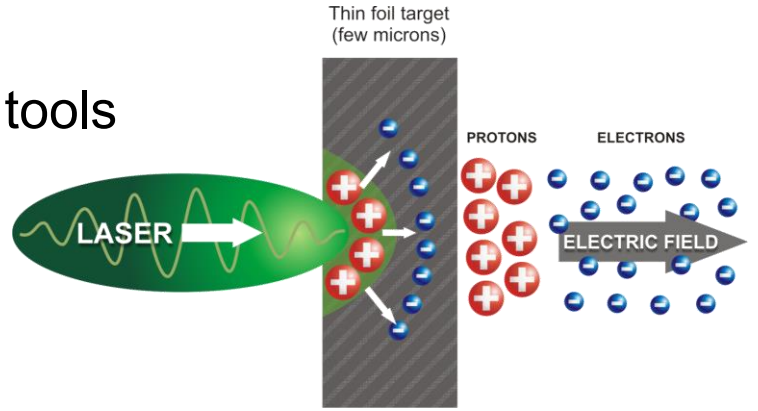
[2]Institute of Applied Physics of the RAS, Nizhni Novgorod

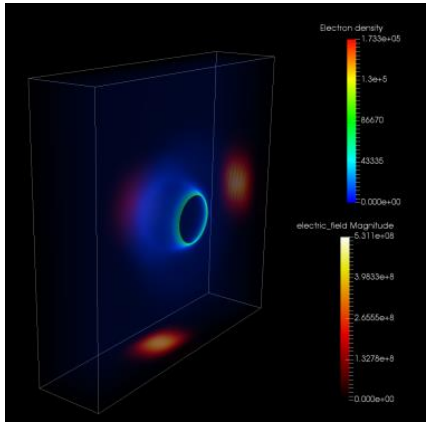
[3]Intel Corporation

Russia

# What's unique about my tuning work

- **Application: PICADOR – innovative, extendable and fully parallel software**
  - Multi-level infrastructure, modular extension interface
  - Optimized computational core
  - Specialized scripting configuration language and visualization tools
  - A wide set of numerical schemes and extensions
  - Dynamic load balancing
- **Application domain**
  - 3D Particle-in-Cell plasma simulation
- **Execution mode**
  - CPU + Intel Xeon Phi in symmetric mode
- **Tools used**
  - Intel Parallel Studio XE, C++
- **Optimization**
  - Intrinsic functions to improve vectorization effectiveness; memory tuning



# Performance

- **1.5x to 2x speedup** on Xeon Phi 5110P vs. Xeon E5-2660
- **Speedup due to optimization:** Xeon 4.2 x, MIC 7.5x
- **Improving memory locality. Speedup:** Xeon 3.1x, MIC 3.4x. Inspired by physical locality of the method. Very generalizable.
- **Improving vectorization with #pragma simd and intrinsic functions (baseline was partially auto-vectorized). Speedup:** Xeon 1.3x, MIC 1.8x. Motivated by Intel VTune Amplifier diagnostics. Generalizable.
- **Reducing thread communication by introducing checkerboard cell traversal order. Speedup:** Xeon 1.1x, MIC 1.3x. Motivated by Intel VTune Amplifier diagnostics. Generalizable, some details specific to the method.

# Insights

- **Learned:**
  - Memory locality is very important, particularly for vectorization
  - Serial code with a negligible share of run time on CPU can become a performance limiting factor on Xeon Phi
- **Recommendation:** designing computational schemes and data structures taking into consideration threading and vectorization potential
- **Intel VTune Amplifier provided valuable insight that invoked optimizations**
- **Biggest surprise:** low vectorization efficiency of loops with indirect array access
- **Key remaining challenge:** improving vectorization efficiency of field interpolation and current deposition phases involving indirect array access
- **Will KNL require a new approach to optimization compared to KNC or minor tuning?**
- **We're thinking of collaborating with HPC application engineers**