

# BLonD Meeting

Konstantinos Iliakis

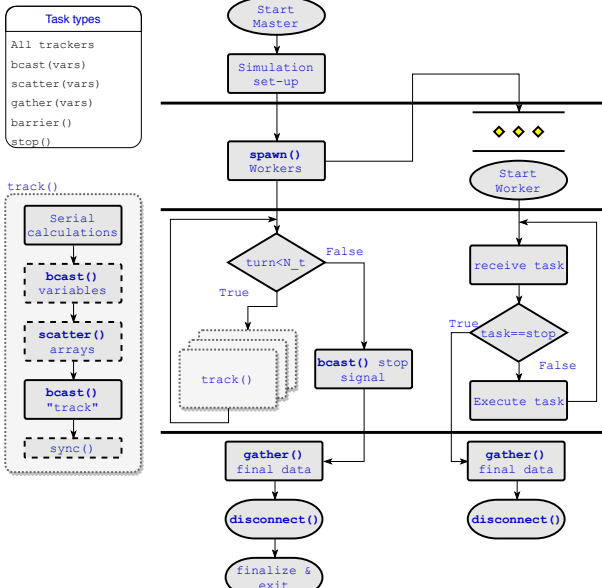


May 18, 2018

# Table of Contents

- 1 Distributed BLonD
- 2 Benchmarking
- 3 Other Notes

# Overview

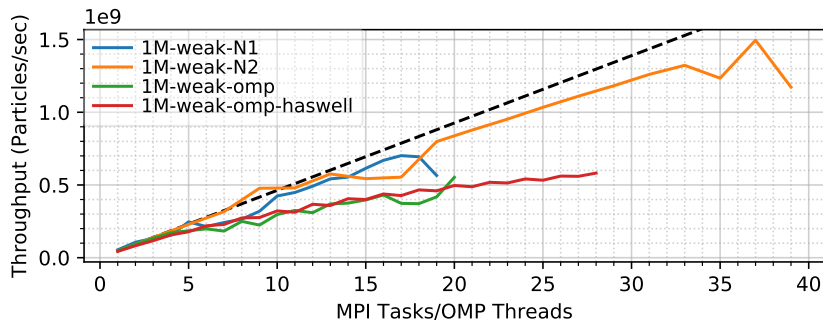


- The master process **only** co-ordinates the workers.
- Each worker is assigned a part of the beam.
- Workers rarely intra-communicate.
- Beam coordinates are not exchanged during the simulation.
- The beam profile is reduced globally and broadcasted to all workers every turn.

# Test-case configuration

- Simple test-case with kick, drift and slice.
- Main file: EX\_01\_Acceleration.py.
- Configuration:
  - 2000 Turns
  - Weak scaling: 1M Particles / Worker
  - Strong scaling: 10M Particles per node.
  - 500 slices per 1M particles (2K particles per slice).
  - No hyper-threading (1 task/thread per physical core).
  - Each node: 2x10 cores (Intel Xeon Broadwell).
  - OpenMP & MPI versions both compiled with the `-parallel` flag.

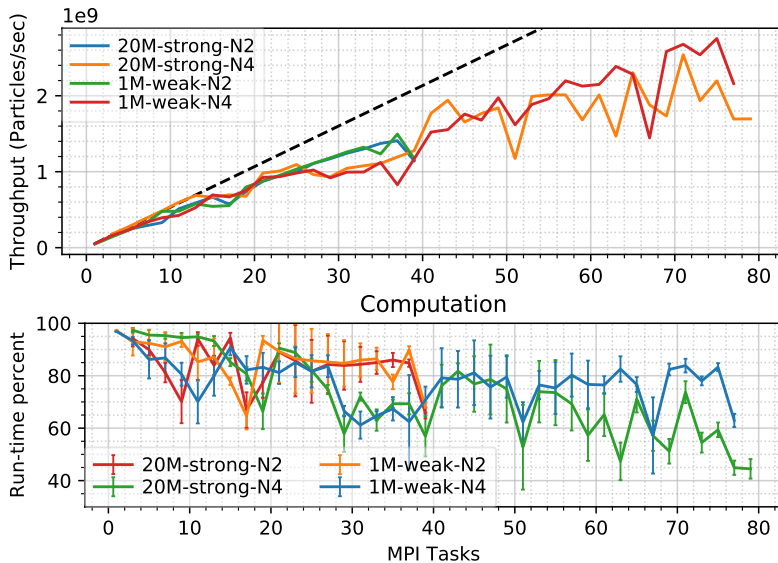
# MPI vs OpenMP



Version	Max Speedup	Threads	Efficiency
OMP-haswell	12.0	28	0.43
OMP-cluster <sup>1</sup>	11.4	20	0.57
MPI-N1	14.5	17	0.85
MPI-N2	31.0	37	0.84
MPI-N4	57.0	75	0.76

<sup>1</sup>Used as the single-core reference

# Multi-node MPI



# Next steps

- Minimize communication: static work-flow, more work per task.
- Relax synchronization: non-blocking communication.
- Hybrid implementation: 1 MPI task/node spawning multiple OpenMP threads.
- Need finer control over the mapping of physical cores to tasks:
  - Ex1: 2 nodes, 2 tasks, 20 cores per task  $\rightarrow \sim 52\%$  utilization.
  - Ex2: 2 nodes, 3 tasks, 20 cores per task  $\rightarrow 52 \leq util. \leq 100$
- Benchmark a more complete simulation, with induced voltage calculation.
- Context manager to handle the MPI initialization, data exchange, functions implementation, etc..

# Other Notes

- BLoND paper accepted for presentation in Samos in July.
- For the most optimized version of BLoND be sure to use the current master. Updated lately:
  - `interp_kick()`
  - `tracker:rf_volt_calc()`
- To be updated:
  - FFTs (in impedances and profile) with `fftw3`
    - windows: Pre-compiled binary from `fftw`.
    - linux: Pre-installed in `blond/work/public/install`.
  - `beam_feedback:beam_phase()`
  - Separatrix losses.



# Thank you for your attention

