Hybrid-BLonD: Efficient Scale-Out of Beam Longitudinal Dynamics Simulations

Konstantinos Iliakis[†,§] Helga Timko[†] Sotirios Xydis[§]
Dimitrios Soudris[§]

[†]European Organization for Nuclear Research (CERN), Geneva, Switzerland
[§]National Technical University of Athens (NTUA), Athens, Greece

konstantinos.iliakis@cern.ch

Intel Extreme Performance Users Group (IXPUG) Annual Conference,
September 24-27, 2019
Globe of Science and Innovation at CERN, Geneva, Switzerland

The BLonD Simulator Suite

What is BLonD?[1] [2]

- **B**eam **Lon**gitudinal **D**ynamics Simulator.
- Main components: **Beam** and the **Synchrotron**.
- Written in Python and C++.
- Modular structure.
- Thoroughly tested and benchmarked[3].

[1] H. Timko et al. "Beam Longitudinal Dynamics Simulation Suite BLonD". In: *Physical Review Accelerators and Beams (to be published)* (2018)

[2] *CERN Beam Longitudinal Dynamics code BLonD*. 2014. URL: `https://blond.web.cern.ch/` (visited on 03/02/2018)

[3] Helga Timko et al. "Benchmarking the beam longitudinal dynamics code BLonD". In: *Proceedings of the 7th International Particle Accelerator Conference (IPAC 2016): Busan, Korea*. 2016

BLonD Usages

BLonD in Action

- Existing synchrotrons[4] [5] [6]
- Machine upgrades, e.g. LIU project[7].
- Future concepts, e.g. FCC[8].

[4] Vincenzo Forte et al. "Longitudinal injection schemes for the CERN PS Booster at 160 MeV including space charge effects". In: *Procs. of the 6th IPAC in Richmond, VA, USA*. 2015

[5] Markus Schwarz et al. "Flat-Bottom Instabilities in the CERN SPS". In: *10th Int. Partile Accelerator Conf.(IPAC'19), Melbourne, Australia*. JACOW. 2019, pp. 3224–3227

[6] Joël Repond et al. "Simulations of Longitudinal Beam Stabilisation in the CERN SPS With BLonD". In: *Proceedings ICAP2018: Key West, FL, USA*. 2018, TUPAF06

[7] Danilo Quartullo, Simon Albright, Elena Shaposhnikova, et al. "Studies of Longitudinal Beam Stability in CERN PS Booster After Upgrade". In: *IPAC'17, Copenhagen, Denmark, May, 2017*. JACOW. 2017

[8] "The Future Circular Collider study". In: *CERN Courier* 54.3 (2014), pp. 16–18. URL: http://cds.cern.ch/record/2064538

BLonD++

BLonD++[9] Overview

- Single-node, optimized version of BLonD.
- Vectorization, parallelization with OpenMP.
- Optimized libraries, e.g. Intel MKL[10], VDT[11], FFTW[12].
- Top-Down method for performance analysis[13].

[9] Konstantinos Iliakis et al. "BLonD++: performance analysis and optimizations for enabling complex, accurate and fast beam dynamics studies". In: *18th Int. Conf. on Embedded Computer Systems: Architectures, Modeling, and Simulation, Pythagorion, Greece*. 2018. URL: https://doi.org/10.1145/3229631.3229640

[10] *Intel® Math Kernel Library*. 2018. URL: https://software.intel.com/en-us/mkl

[11] Danilo Piparo, Vincenzo Innocente, and Thomas Hauth. "Speeding up HEP experiment software with a library of fast and auto-vectorisable mathematical functions". In: *Journal of Physics: Conference Series* (2014)

[12] Matteo Frigo and Steven G Johnson. "FFTW: An adaptive software architecture for the FFT". In: *Procs. of the 1998 Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP'98*. IEEE. 1998

[13] Ahmad Yasin. "A top-down method for performance analysis and counters architecture". In: *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE. 2014, pp. 35–44

BLonD++

BLonD++[9] Overview

- Single-node, optimized version of BLonD.
- Vectorization, parallelization with OpenMP.
- Optimized libraries, e.g. Intel MKL[10], VDT[11], FFTW[12].
- Top-Down method for performance analysis[13].

Limitations

- Memory-bounded.
- Only vertical scaling.

[9] Konstantinos Iliakis et al. "BLonD++: performance analysis and optimizations for enabling complex, accurate and fast beam dynamics studies". In: *18th Int. Conf. on Embedded Computer Systems: Architectures, Modeling, and Simulation, Pythagorion, Greece*. 2018. URL: https://doi.org/10.1145/3229631.3229640

[10] *Intel® Math Kernel Library*. 2018. URL: https://software.intel.com/en-us/mkl

[11] Danilo Piparo, Vincenzo Innocente, and Thomas Hauth. "Speeding up HEP experiment software with a library of fast and auto-vectorisable mathematical functions". In: *Journal of Physics: Conference Series* (2014)

[12] Matteo Frigo and Steven G Johnson. "FFTW: An adaptive software architecture for the FFT". In: *Procs. of the 1998 Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP'98*. IEEE. 1998

[13] Ahmad Yasin. "A top-down method for performance analysis and counters architecture". In: *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE. 2014, pp. 35–44

Hybrid-BLonD Overview

- MPI-over-OpenMP.
- Horizontal and vertical scaling.
- Target: Computation, Communication, Intra-worker processing.

Hybrid-BLonD Overview

- MPI-over-OpenMP.
- Horizontal and vertical scaling.
- Target: Computation, Communication, Intra-worker processing.

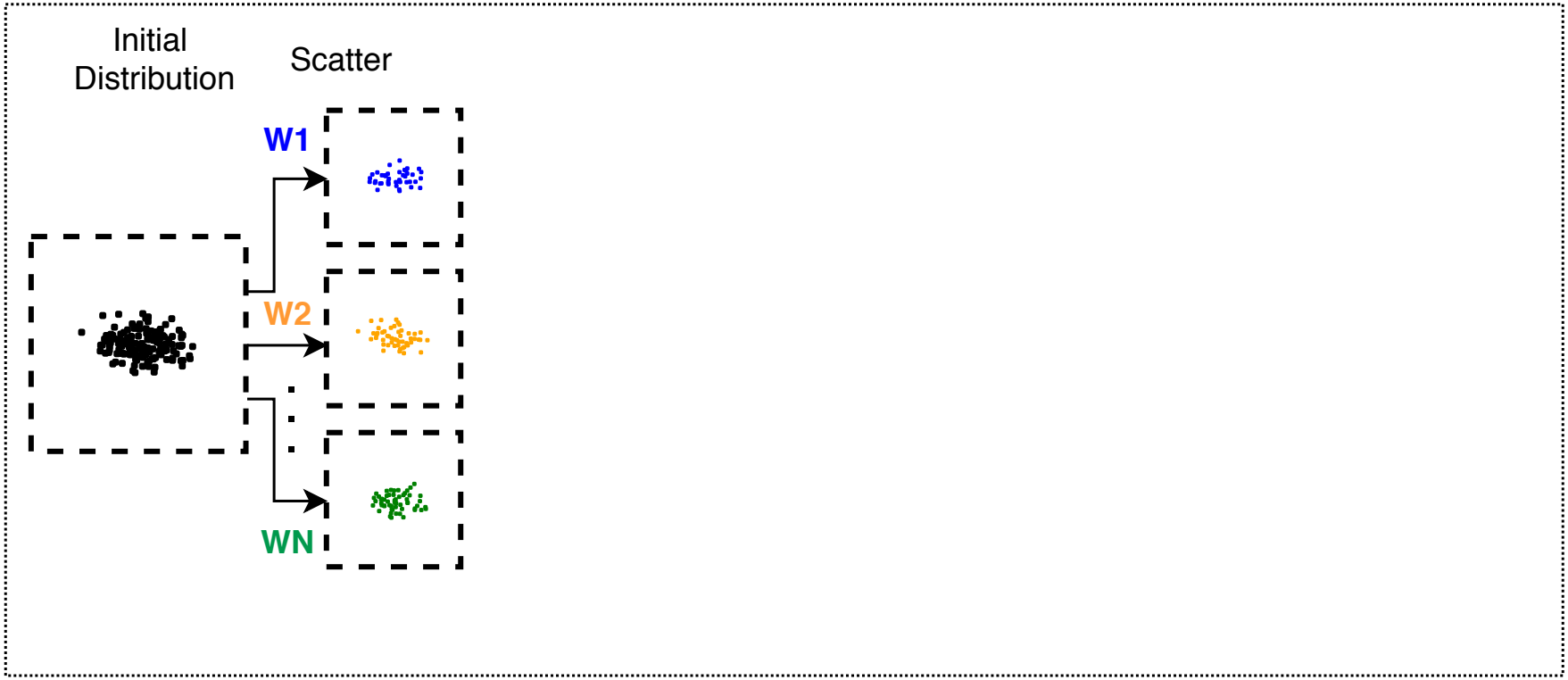


Figure 1: Baseline Hybrid-BLonD work-flow.

Hybrid-BLonD Overview

- MPI-over-OpenMP.
- Horizontal and vertical scaling.
- Target: Computation, Communication, Intra-worker processing.

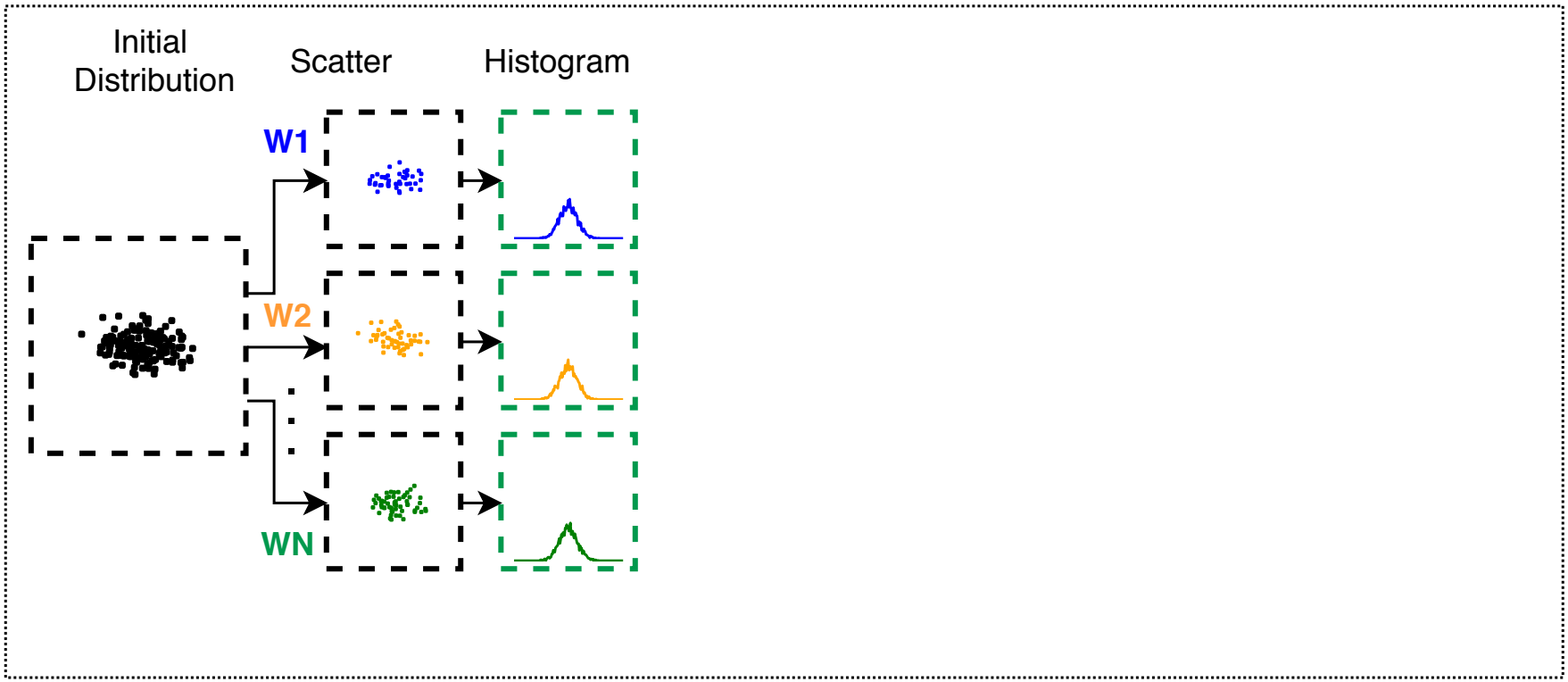


Figure 1: Baseline Hybrid-BLonD work-flow.

Hybrid-BLonD Overview

- MPI-over-OpenMP.
- Horizontal and vertical scaling.
- Target: Computation, Communication, Intra-worker processing.

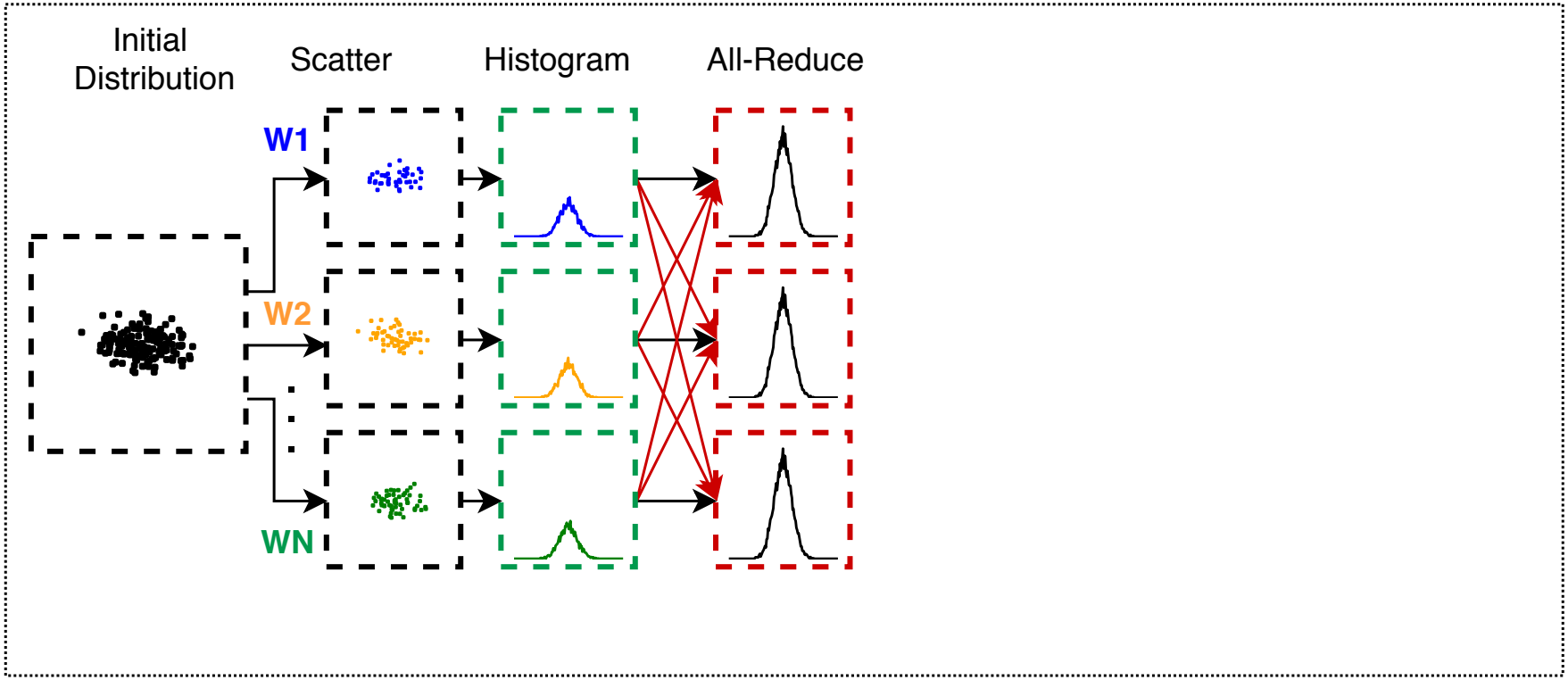


Figure 1: Baseline Hybrid-BLonD work-flow.

Hybrid-BLonD Overview

- MPI-over-OpenMP.
- Horizontal and vertical scaling.
- Target: Computation, Communication, Intra-worker processing.

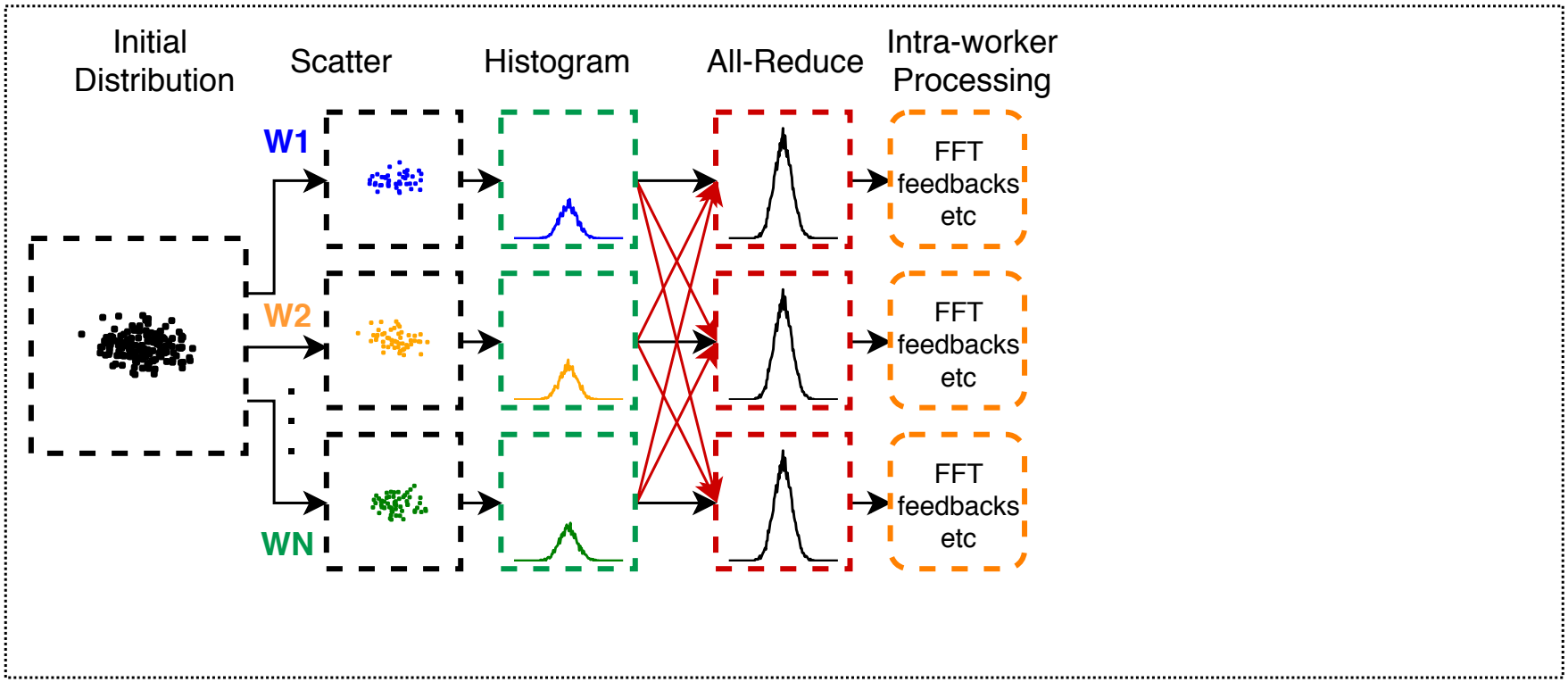


Figure 1: Baseline Hybrid-BLonD work-flow.

Hybrid-BLonD Overview

- MPI-over-OpenMP.
- Horizontal and vertical scaling.
- Target: Computation, Communication, Intra-worker processing.

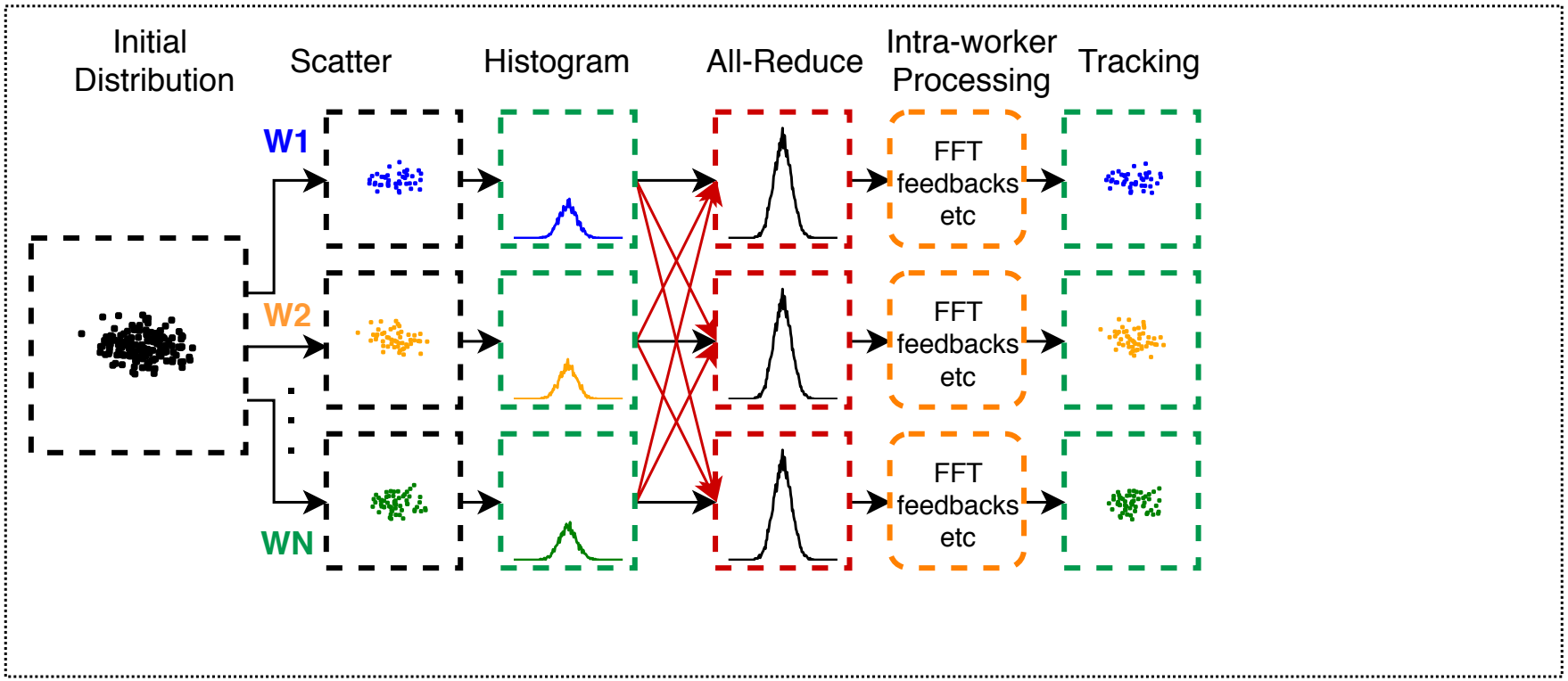


Figure 1: Baseline Hybrid-BLonD work-flow.

Hybrid-BLonD Overview

- MPI-over-OpenMP.
- Horizontal and vertical scaling.
- Target: Computation, Communication, Intra-worker processing.

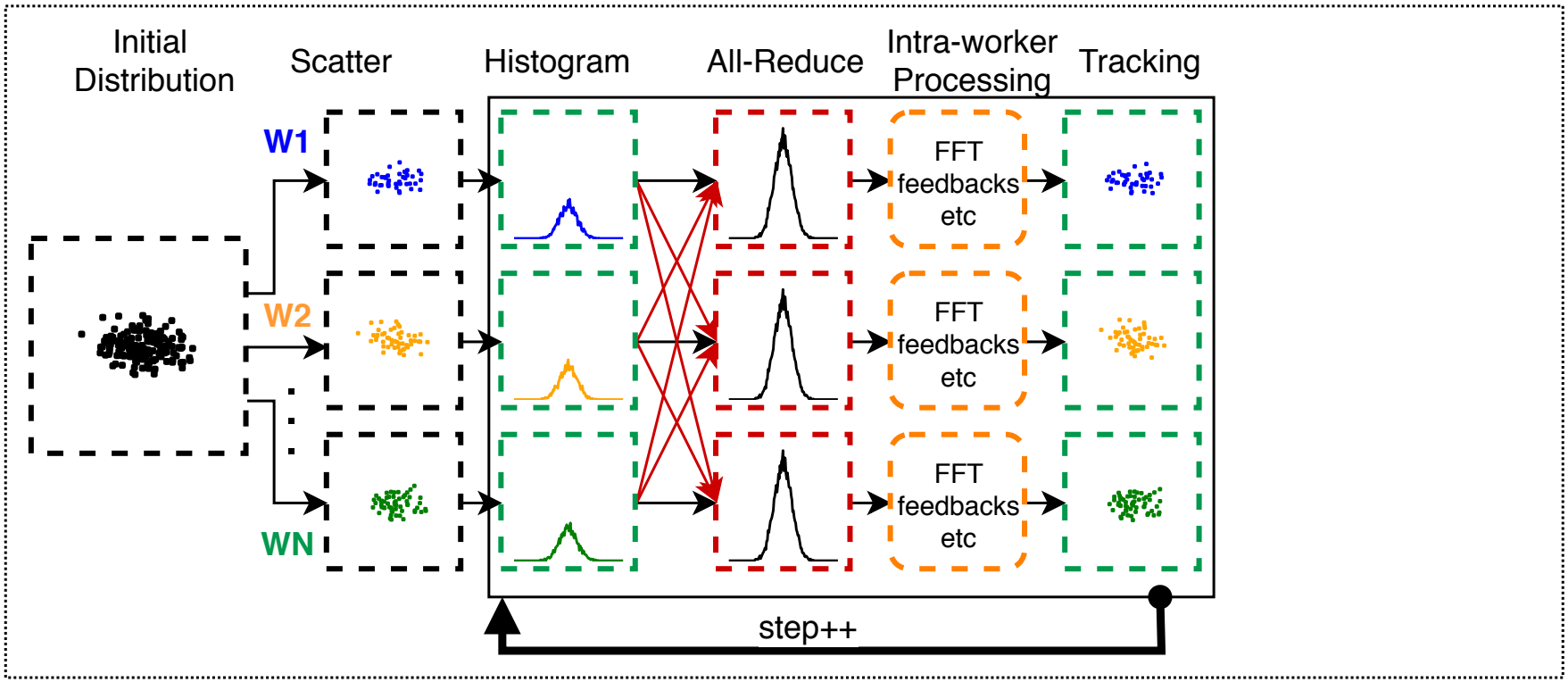


Figure 1: Baseline Hybrid-BLonD work-flow.

Hybrid-BLonD Overview

- MPI-over-OpenMP.
- Horizontal and vertical scaling.
- Target: Computation, Communication, Intra-worker processing.

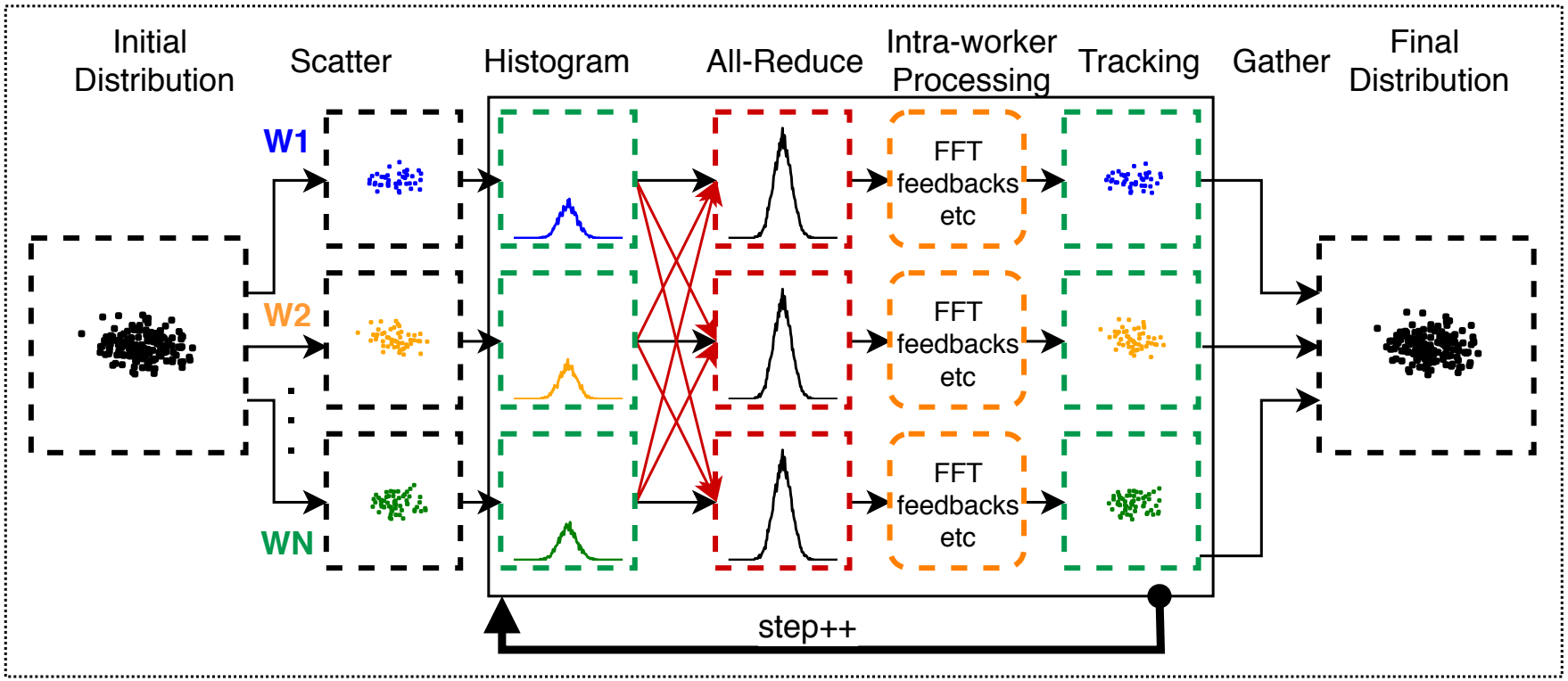


Figure 1: Baseline Hybrid-BLonD work-flow.

Task-Parallelism

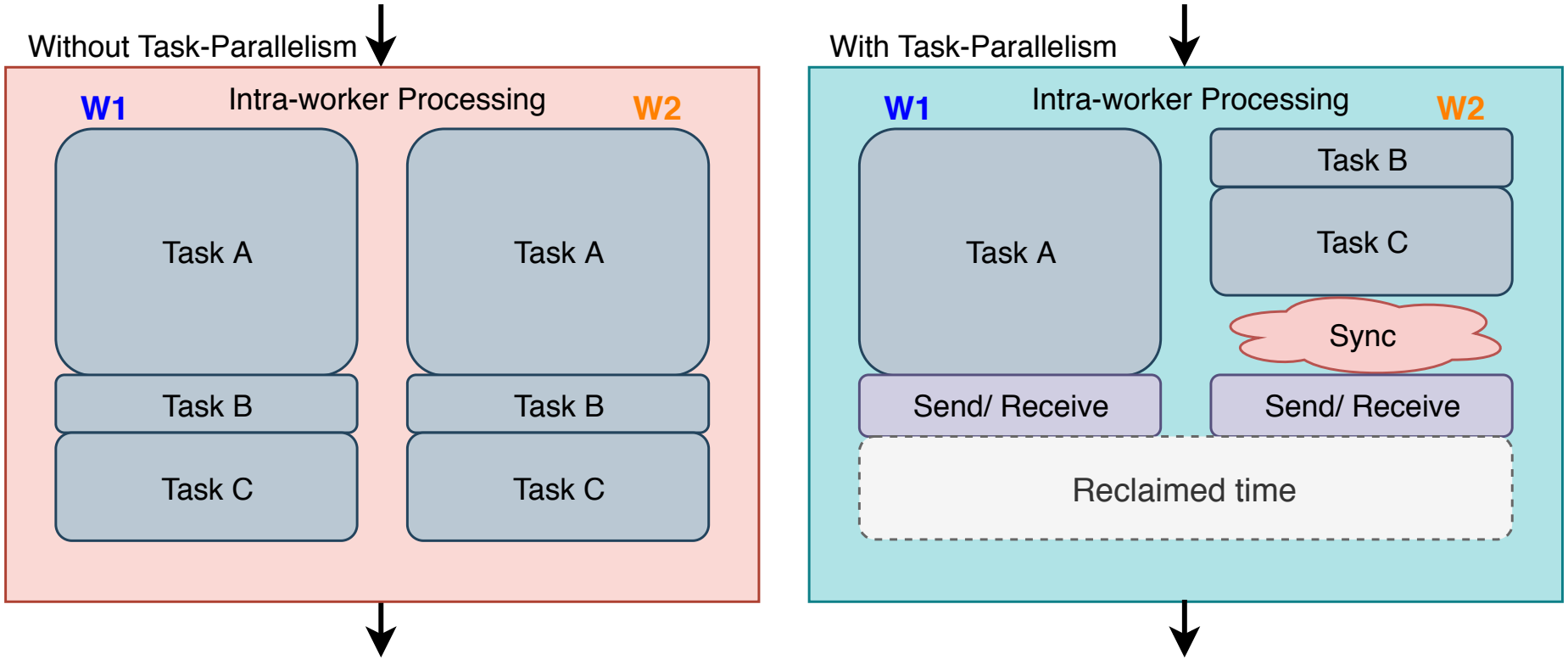


Figure 2: Intra-node processing w/ and w/o task-parallelism.

- Data-parallelism across nodes.
- Task-parallelism intra-node.
- Imbalanced workload $\Rightarrow$ lost time.

Approximate-Computing

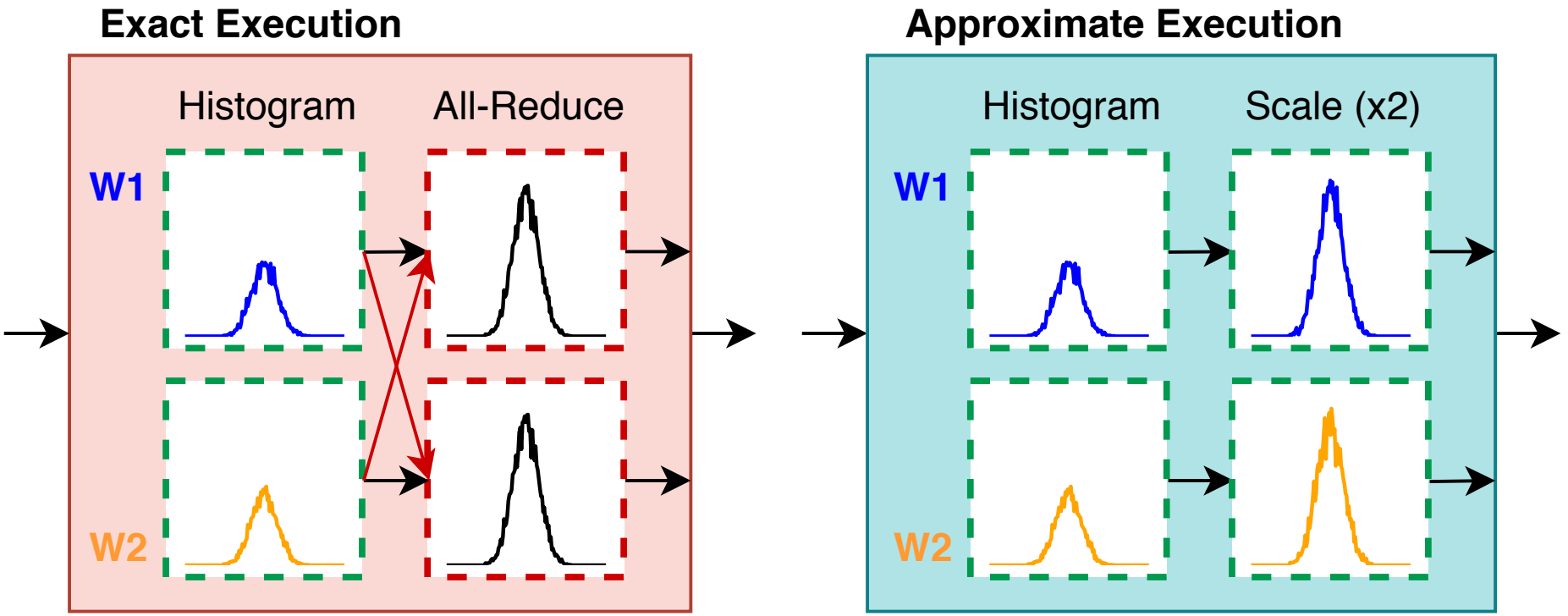
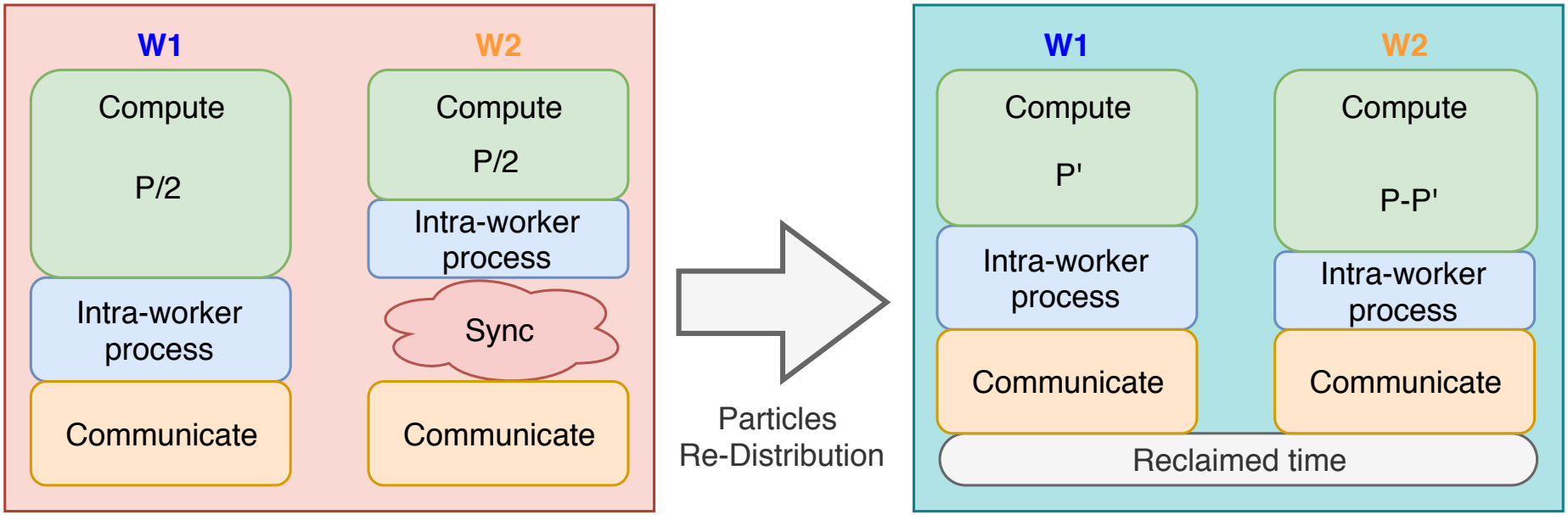


Figure 3: Representative distribution sub-set approximation.

- Assumption: Representative particle distribution sub-set.
- Completely disengages workers.

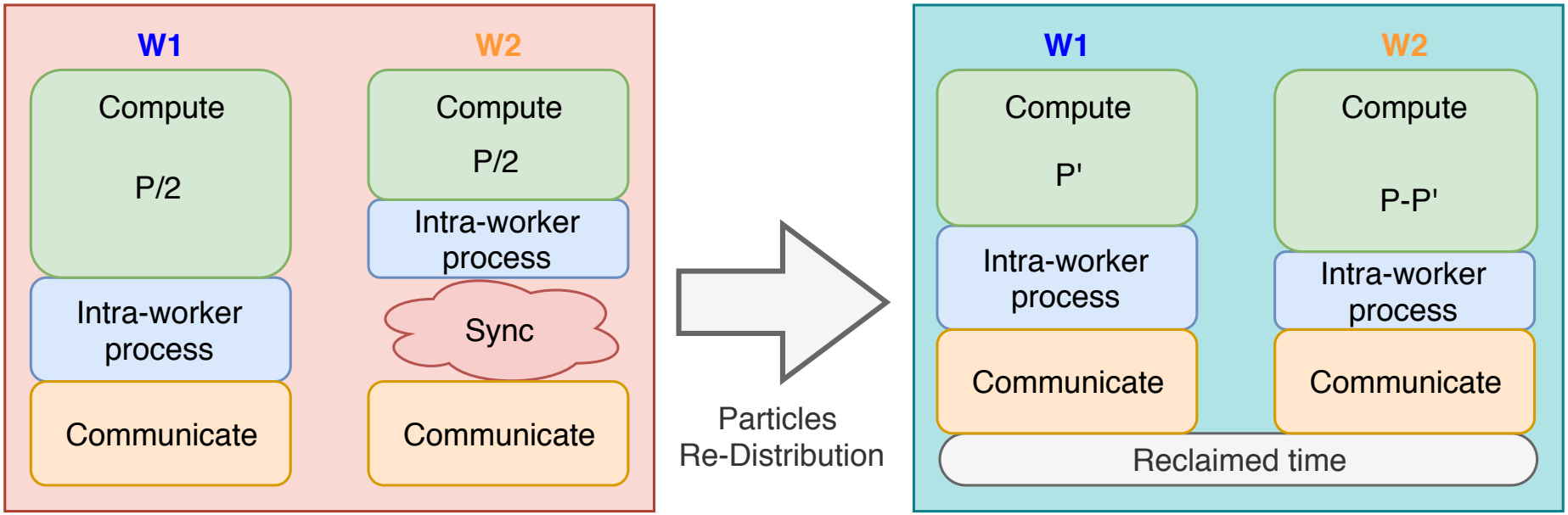
Dynamic-Load-Balancing



P: Total number of particles

Figure 4: Particle re-distribution to balance load across workers.

Dynamic-Load-Balancing



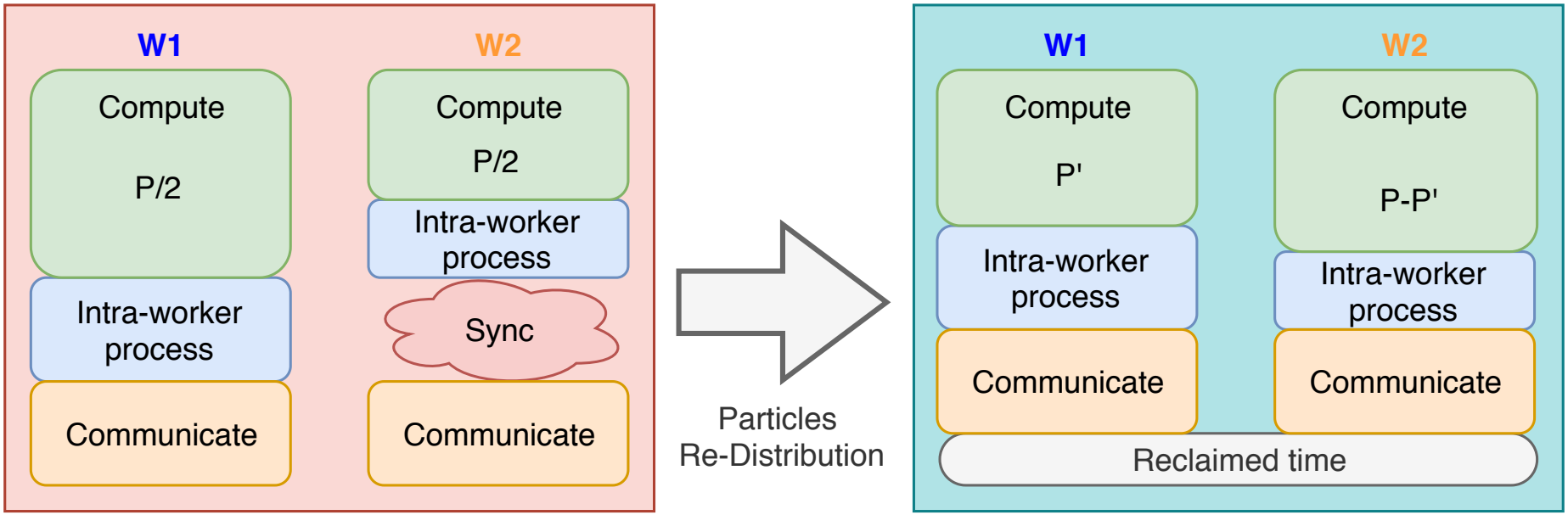
P: Total number of particles

Figure 4: Particle re-distribution to balance load across workers.

Assumptions

1. $T_{comp} \propto P$.
2. $T_{comm} = const$, $T_{intra} = const$.
3. Perfect load balance $\Leftrightarrow T_{sync} \rightarrow min$.
4. Workers exhibit similar behavior for long periods.

Dynamic-Load-Balancing



P: Total number of particles

Figure 4: Particle re-distribution to balance load across workers.

Assumptions

1. $T_{comp} \propto P$.
2. $T_{comm} = const$, $T_{intra} = const$.
3. Perfect load balance $\Leftrightarrow T_{sync} \rightarrow min$.
4. Workers exhibit similar behavior for long periods.

Set-Up

Cluster & Test-cases

- 314 nodes, Total cores: $\approx$ 6k.
- Intel Xeon®, dual-socket, 10 cores per socket @ 2.2GHz, 25MB L3.
- Infiniband interconnect.
- MVAPICH-2.3 (also benchmarked MPICH-3, OPENMPI-3).
- 3 Realistic test-cases: **LHC**, **SPS** and **PS**[14].

[14] CERN Accelerator Complex: Large Hadron Collider (LHC), Super Proton Synchrotron (SPS), Proton Synchrotron (PS)

Hybrid-BLonD Evaluation

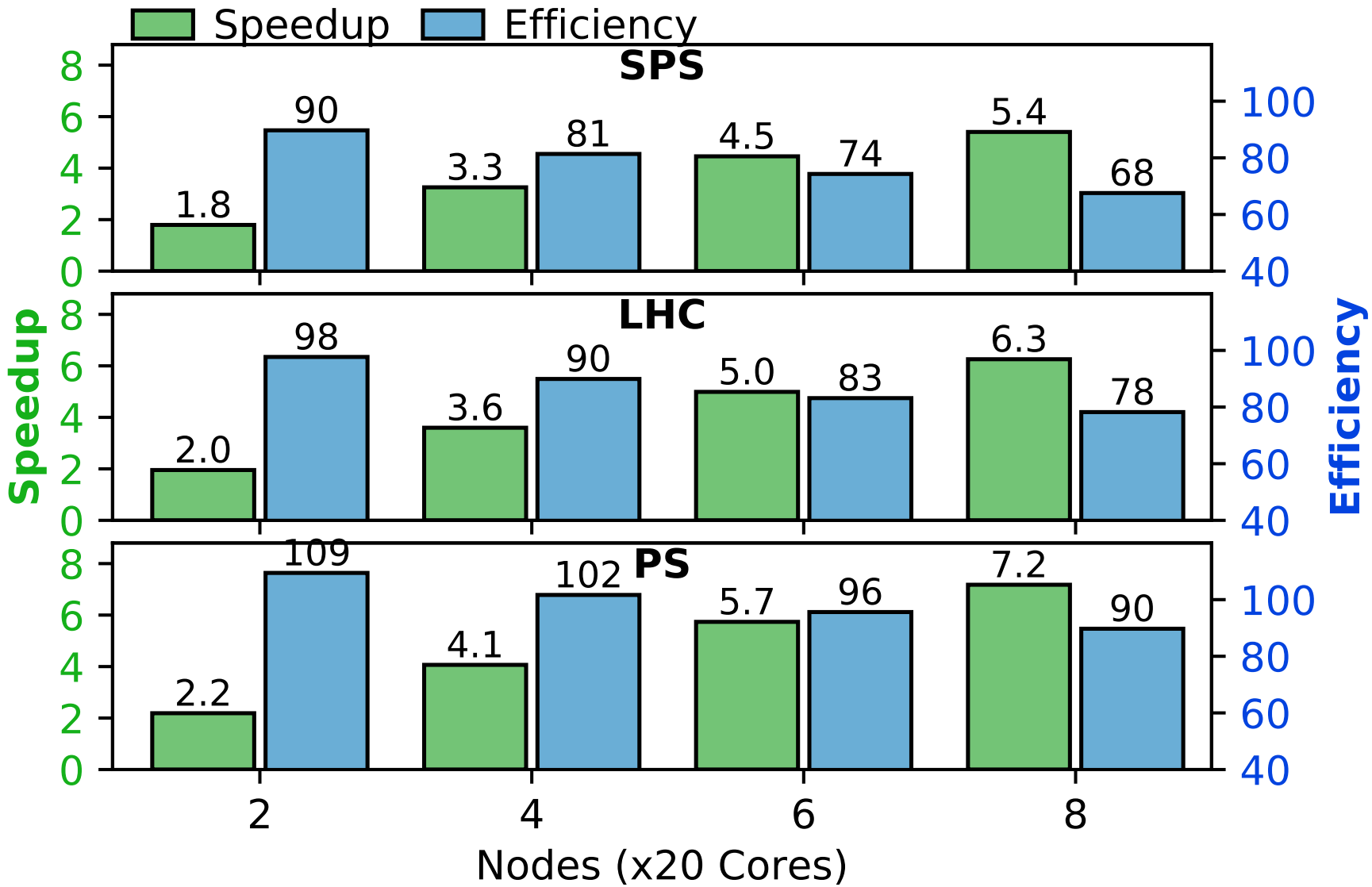


Figure 5: Speedup: Hybrid-BLonD against single-node BLonD++.
Efficiency: Speedup normalized to theoretical peak.

Hybrid-BLonD Evaluation

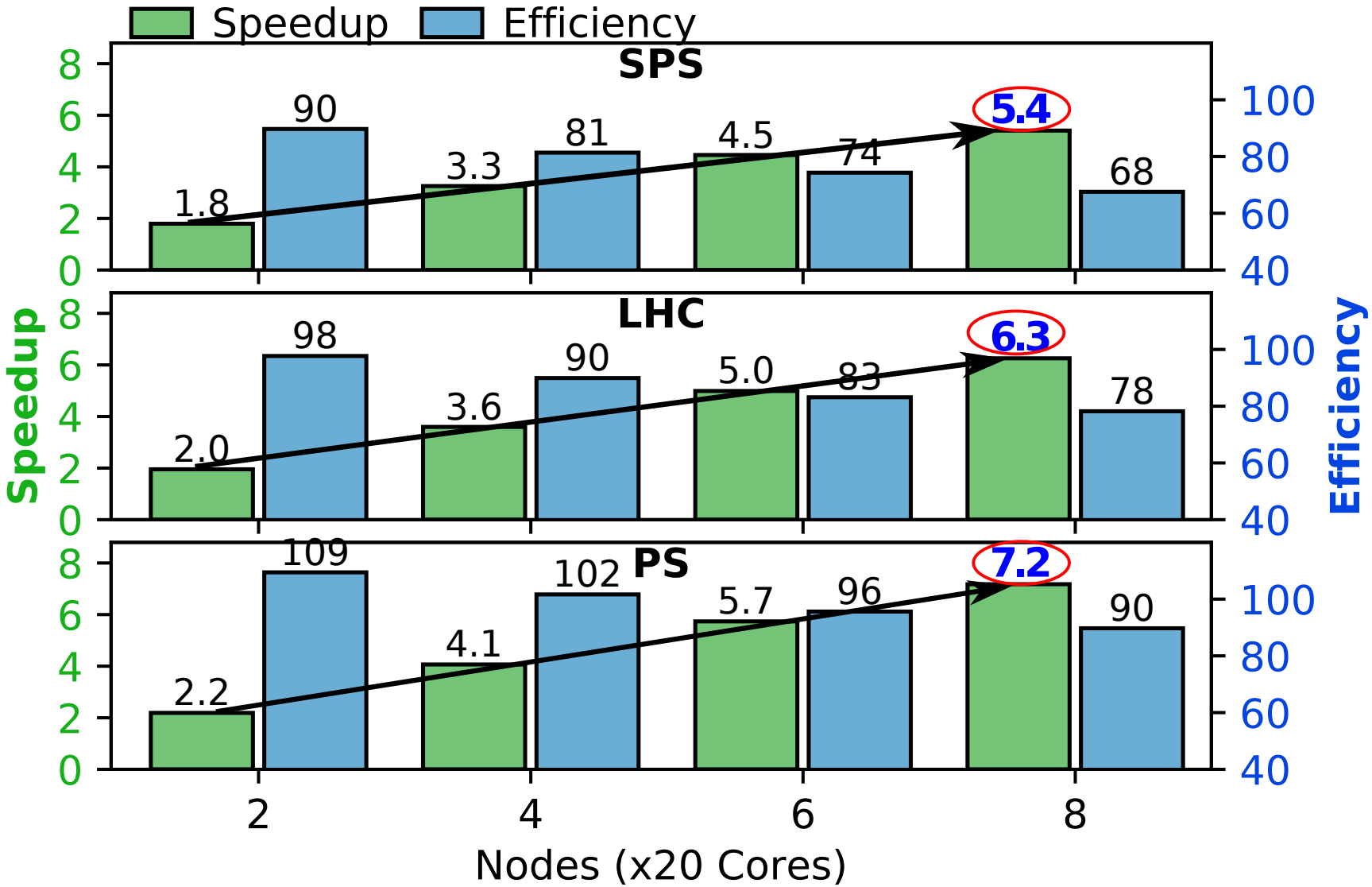


Figure 5: Speedup: Hybrid-BLonD against single-node BLonD++.
Efficiency: Speedup normalized to theoretical peak.

Hybrid-BLonD Evaluation

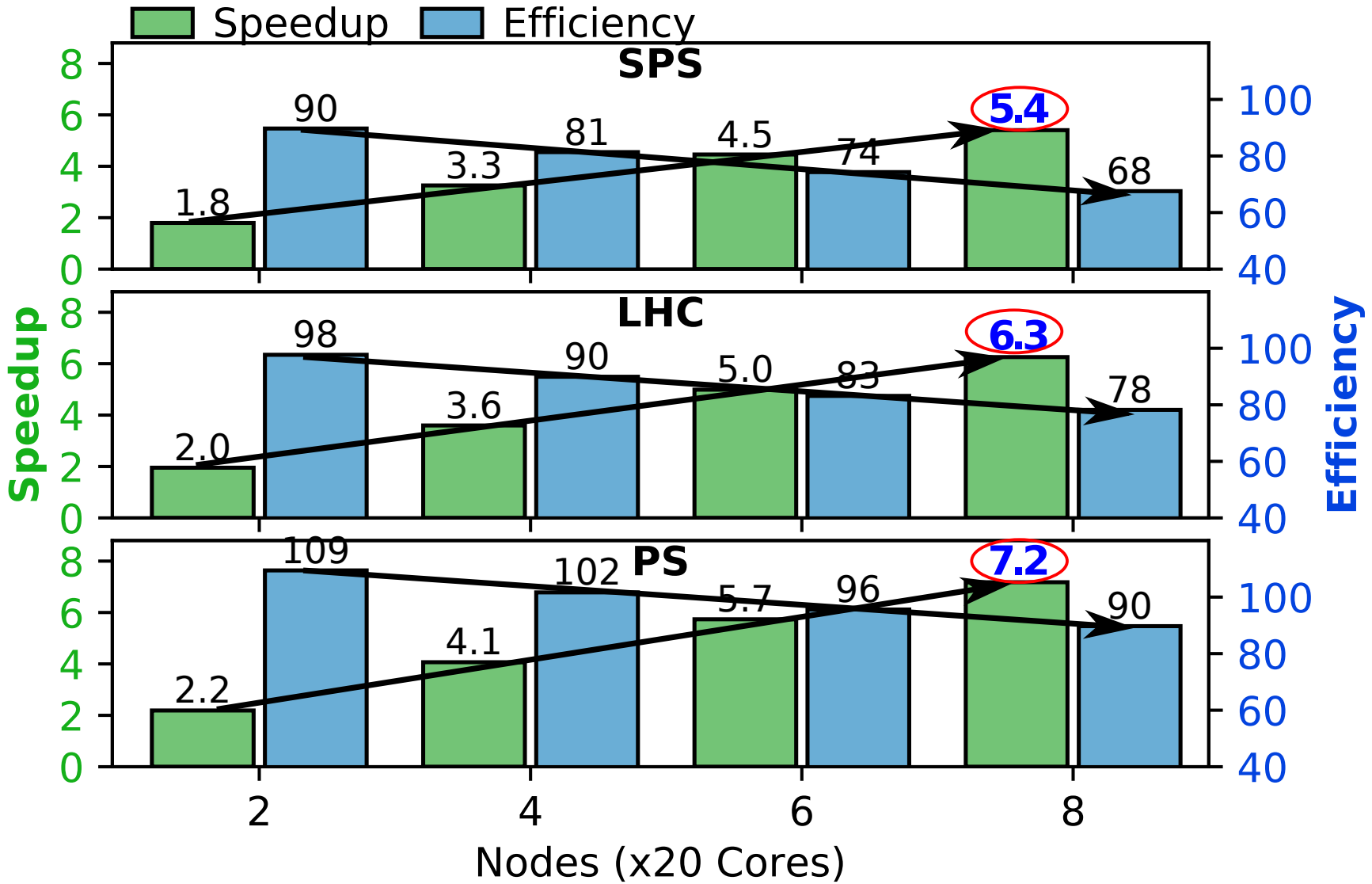


Figure 5: Speedup: Hybrid-BLonD against single-node BLonD++.
Efficiency: Speedup normalized to theoretical peak.

Hybrid-BLonD Evaluation

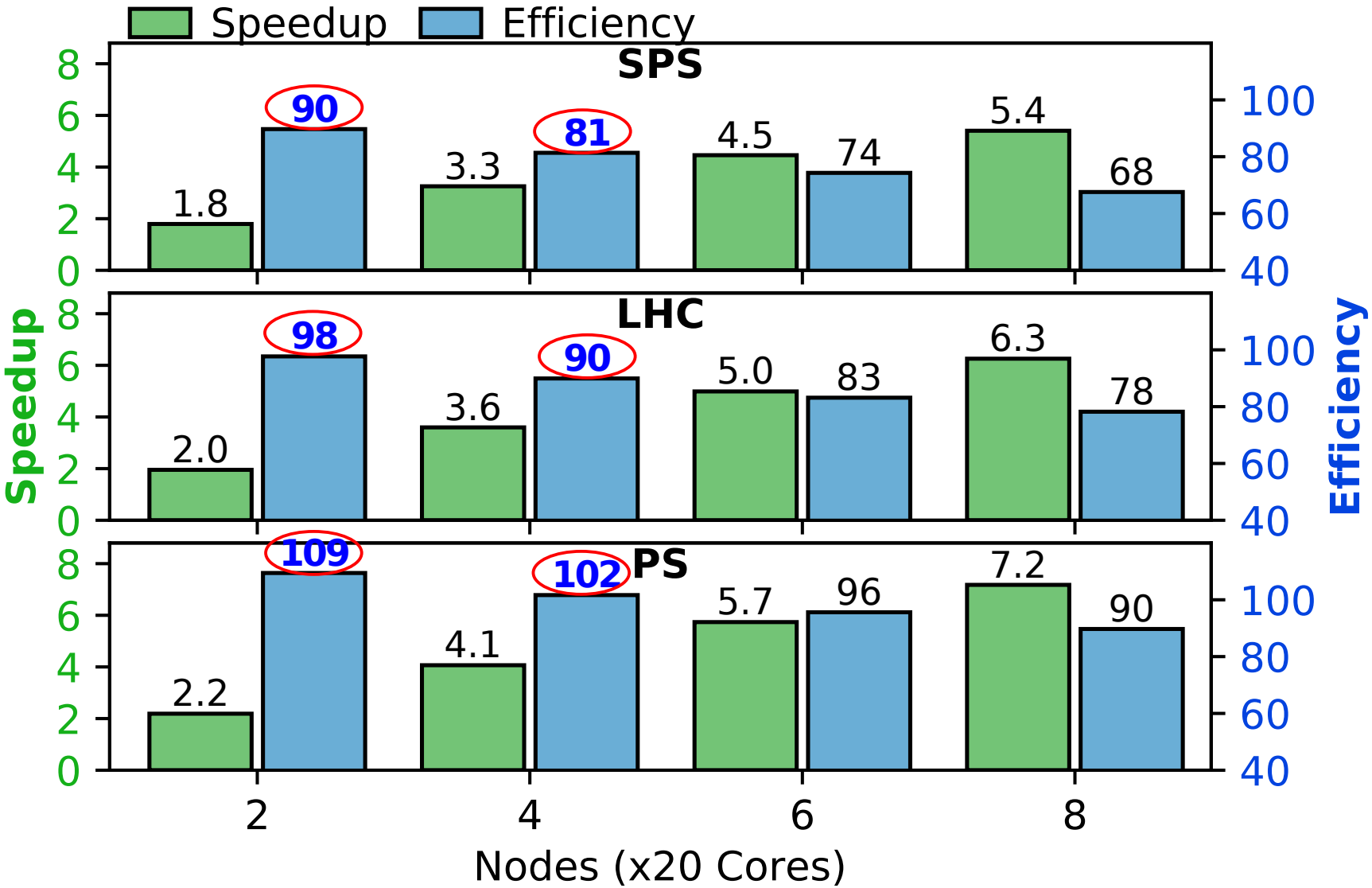


Figure 5: Speedup: Hybrid-BLonD against single-node BLonD++.
Efficiency: Speedup normalized to theoretical peak.

Hybrid-BLonD Incremental Effect Analysis

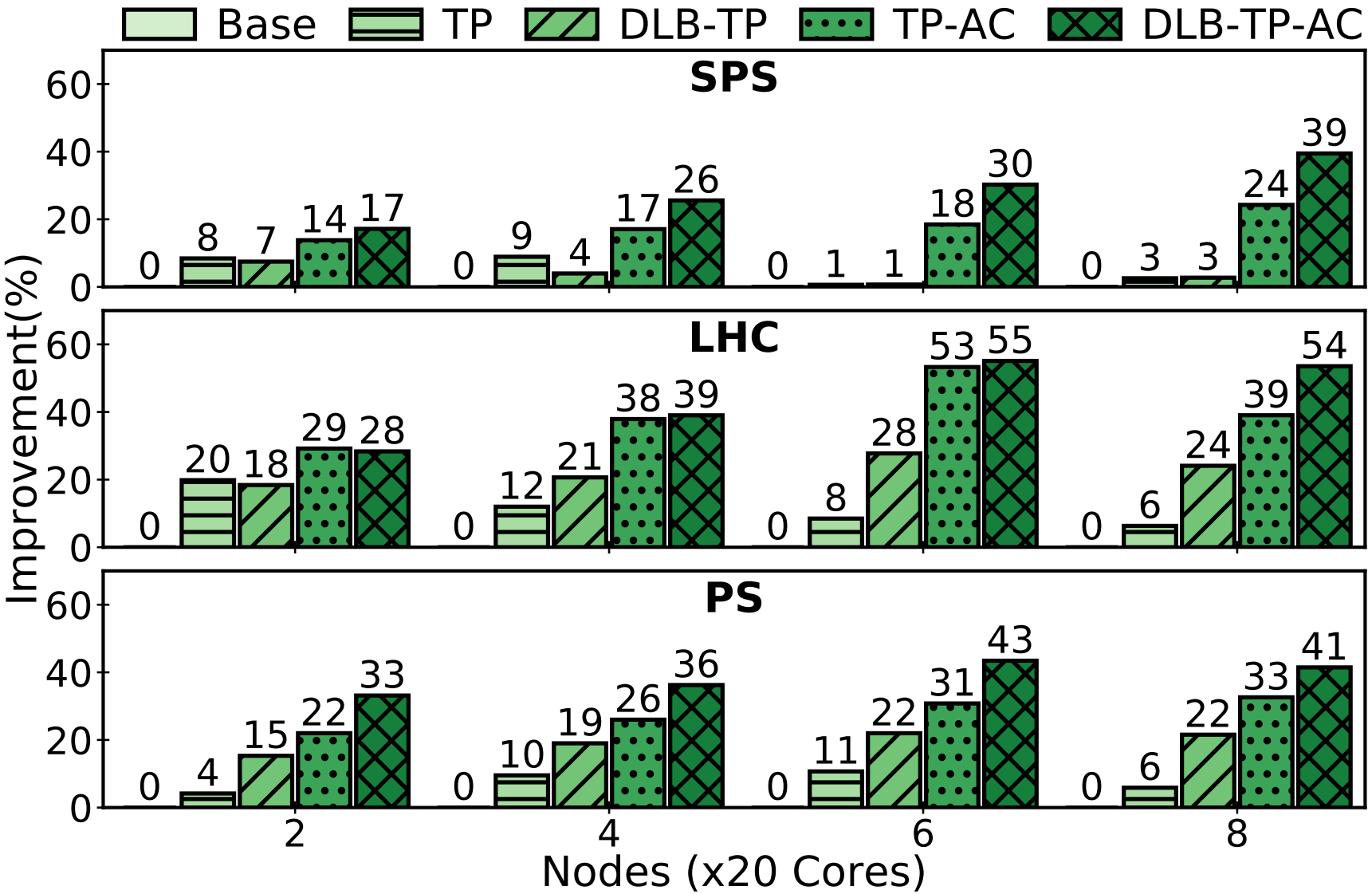


Figure 6: Incremental performance improvement compared to baseline Hybrid-BLonD.
TP: Task-Parallelism, **AC**: Approximate Computing, **DLB**: Dynamic-Load-Balancing

Hybrid-BLonD Incremental Effect Analysis

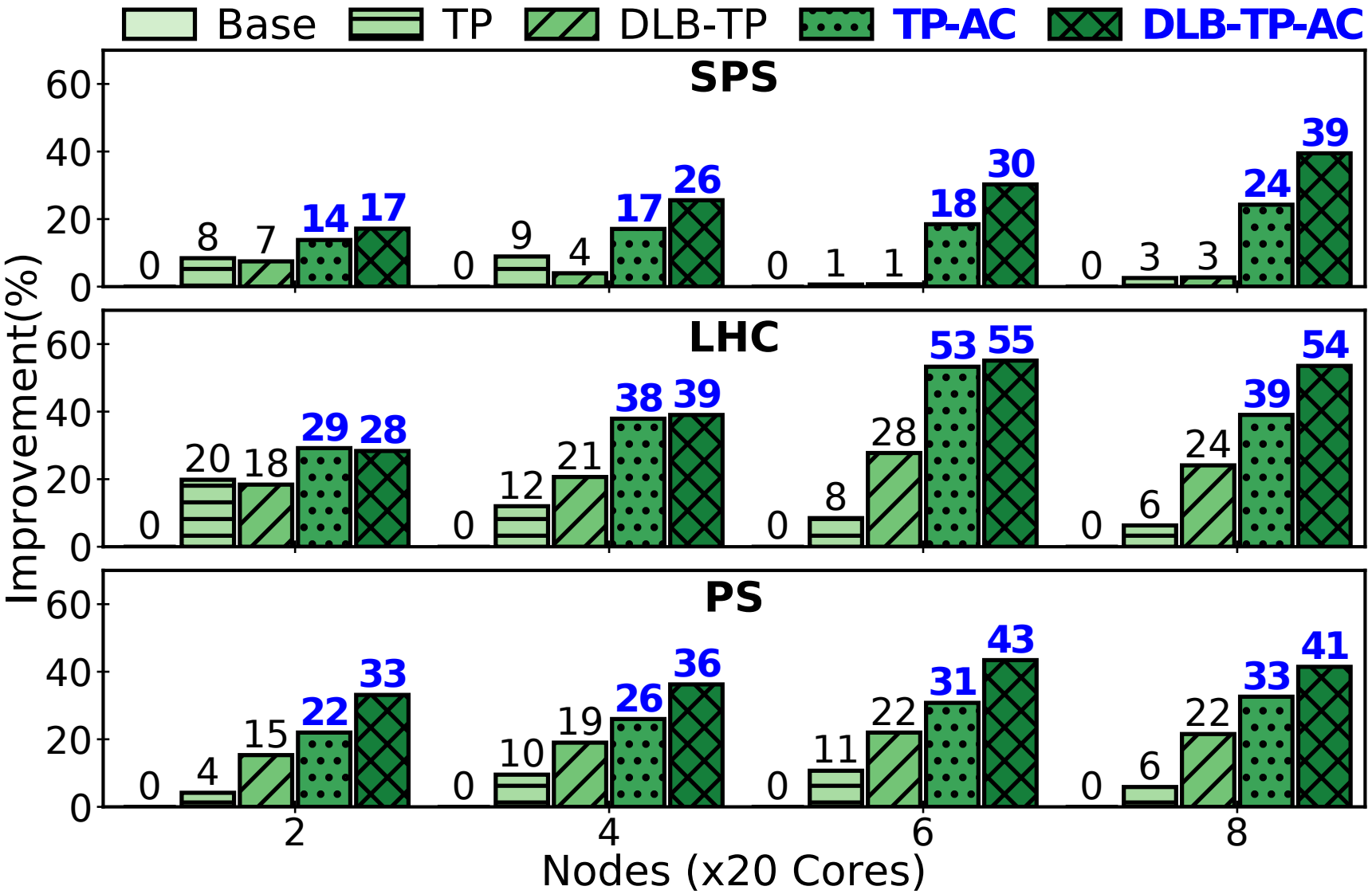


Figure 6: Incremental performance improvement compared to baseline Hybrid-BLonD.
TP: Task-Parallelism, **AC**: Approximate Computing, **DLB**: Dynamic-Load-Balancing

Hybrid-BLonD Incremental Effect Analysis

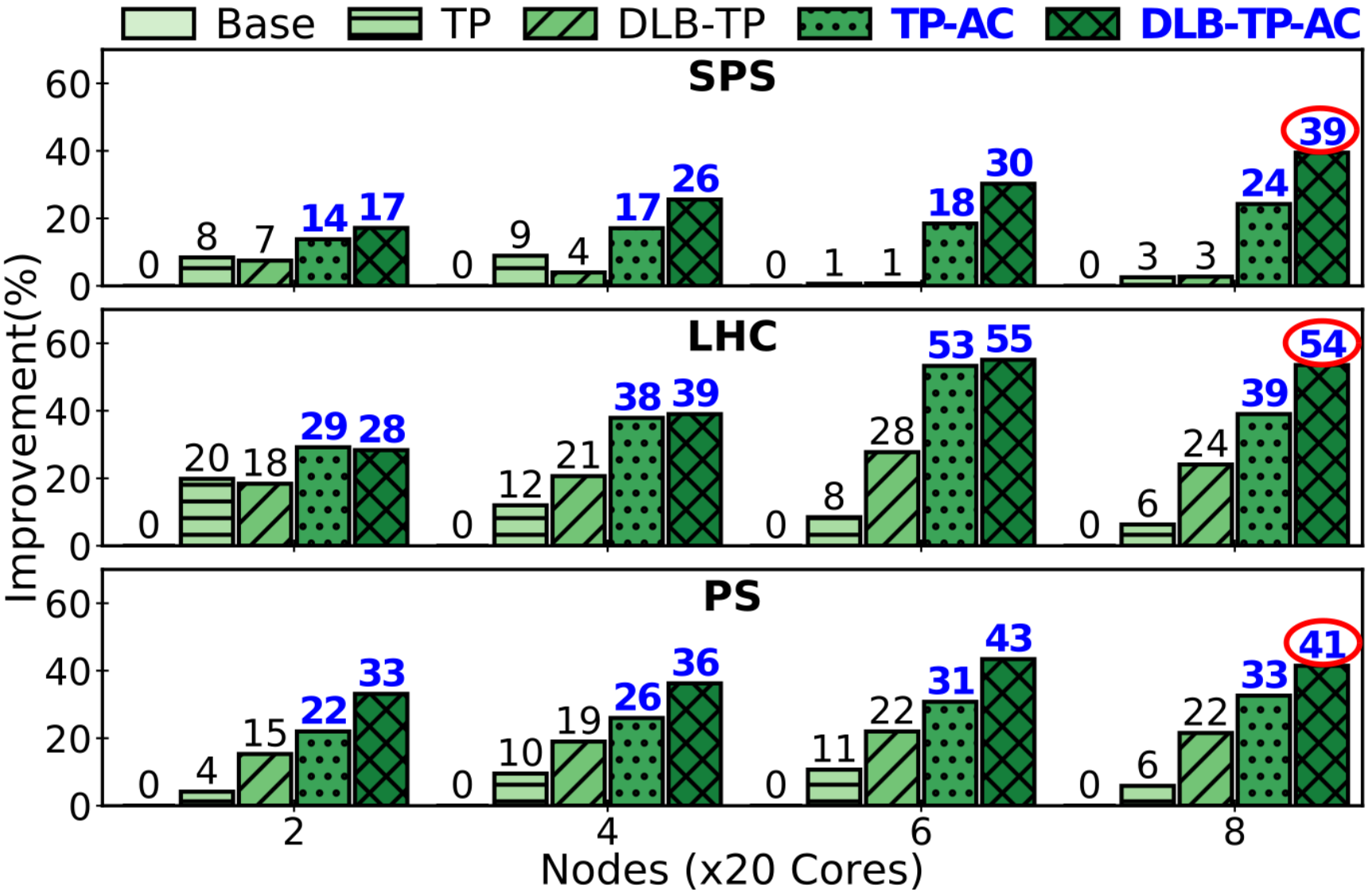


Figure 6: Incremental performance improvement compared to baseline Hybrid-BLonD.
TP: Task-Parallelism, **AC**: Approximate Computing, **DLB**: Dynamic-Load-Balancing

Lessons Learned

- Profile, Profile, Profile![15] [16] [17]

[15] James Reinders. "VTune performance analyzer essentials". In: *Intel Press* (2005)

[16] Philip J Mucci et al. "PAPI: A portable interface to hardware performance counters". In: *Proceedings of the department of defense HPCMP users group conference*. Vol. 710. 1999

[17] *Linux profiling with performance counters*. 2019. URL: `https://perf.wiki.kernel.org/index.php/Main_Page` (visited on 09/03/2019)

[18] Mehrzad Samadi et al. "Paraprox: Pattern-based approximation for data parallel applications". In: *ACM SIGPLAN Notices* 49.4 (2014), pp. 35–50

[19] Daya S Khudia et al. "Rumba: An online quality management system for approximate computing". In: *Computer Architecture (ISCA), 2015 ACM/IEEE 42nd Annual International Symposium on*. IEEE. 2015, pp. 554–566

[20] Qian Zhang and Qiang Xu. "ApproxIt: A Quality Management Framework of Approximate Computing for Iterative Methods". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2017)

[21] Michael A Laurenzano et al. "Input responsiveness: using canary inputs to dynamically steer approximation". In: *ACM SIGPLAN Notices* 51.6 (2016), pp. 161–176

Lessons Learned

- Profile, Profile, Profile![15] [16] [17]
- Invest in visualization tools.

[15] James Reinders. "VTune performance analyzer essentials". In: *Intel Press* (2005)

[16] Philip J Mucci et al. "PAPI: A portable interface to hardware performance counters". In: *Proceedings of the department of defense HPCMP users group conference*. Vol. 710. 1999

[17] *Linux profiling with performance counters*. 2019. URL: `https://perf.wiki.kernel.org/index.php/Main_Page` (visited on 09/03/2019)

[18] Mehrzad Samadi et al. "Paraprox: Pattern-based approximation for data parallel applications". In: *ACM SIGPLAN Notices* 49.4 (2014), pp. 35–50

[19] Daya S Khudia et al. "Rumba: An online quality management system for approximate computing". In: *Computer Architecture (ISCA), 2015 ACM/IEEE 42nd Annual International Symposium on*. IEEE. 2015, pp. 554–566

[20] Qian Zhang and Qiang Xu. "ApproxIt: A Quality Management Framework of Approximate Computing for Iterative Methods". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2017)

[21] Michael A Laurenzano et al. "Input responsiveness: using canary inputs to dynamically steer approximation". In: *ACM SIGPLAN Notices* 51.6 (2016), pp. 161–176

Lessons Learned

- Profile, Profile, Profile![15] [16] [17]
- Invest in visualization tools.
- Approximate computing.[18] [19] [20] [21]

[15] James Reinders. "VTune performance analyzer essentials". In: *Intel Press* (2005)

[16] Philip J Mucci et al. "PAPI: A portable interface to hardware performance counters". In: *Proceedings of the department of defense HPCMP users group conference*. Vol. 710. 1999

[17] *Linux profiling with performance counters*. 2019. URL: https://perf.wiki.kernel.org/index.php/Main_Page (visited on 09/03/2019)

[18] Mehrzad Samadi et al. "Paraprox: Pattern-based approximation for data parallel applications". In: *ACM SIGPLAN Notices* 49.4 (2014), pp. 35–50

[19] Daya S Khudia et al. "Rumba: An online quality management system for approximate computing". In: *Computer Architecture (ISCA), 2015 ACM/IEEE 42nd Annual International Symposium on*. IEEE. 2015, pp. 554–566

[20] Qian Zhang and Qiang Xu. "ApproxIt: A Quality Management Framework of Approximate Computing for Iterative Methods". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2017)

[21] Michael A Laurenzano et al. "Input responsiveness: using canary inputs to dynamically steer approximation". In: *ACM SIGPLAN Notices* 51.6 (2016), pp. 161–176

Conclusions

- Presented typical HPC techniques:
  - Mixed data and task-parallelism.
  - Approximate computing.
  - Dynamic-load-balancing scheme.
- 8-node speed-up **up to 7.2x**, efficiency **up to 90%**.
- Hybrid-BLonD enables **multi-bunch**, **fine-resolution**, **high-complexity** simulations.

Thank you for your attention!