Parallel performance of "Synchrotron Radiation Workshop" code: partially coherent calculations for storage rings and time-dependent calculations for XFELs

He, A., Chubar, O., Rakitin, M., Samoylova, L., Fortmann-Grote, C., et al.

A. He, O. Chubar, M. Rakitin, L. Samoylova, C. Fortmann-Grote, S. Yakubov, A. Buzmakov, "Parallel performance of "Synchrotron Radiation Workshop " code: partially coherent calculations for storage rings and time-dependent calculations for XFELs," Proc. SPIE 11493, Advances in Computational Methods for X-Ray Optics V, 114930H (21 August 2020); doi: 10.1117/12.2567448

Event: SPIE Optical Engineering + Applications, 2020, Online Only

Parallel performance of “Synchrotron Radiation Workshop” code: partially coherent calculations for storage rings and time-dependent calculations for XFELs

A. He*[a], O. Chubar[a], M. Rakitin[a], L. Samoylova[b], C. Fortmann-Grote[b], S. Yakubov[c], A. Buzmakov[d]
[a]National Synchrotron Light Source II, Brookhaven National Laboratory, Upton 11973, NY, USA;
[b]European XFEL GmbH, 22869 Schenefeld, Germany; [c]German Electron Synchrotron DESY,
22607 Hamburg, Germany; [d]FSRC Crystallography and Photonics RAS, Moscow, 119333, Russia

ABSTRACT

The “Synchrotron Radiation Workshop” (SRW) computer code is extensively used for the development of insertion devices (IDs) and X-ray beamlines at the National Synchrotron Light Source II and at other light source facilities. Among frequently used types of SRW calculations are the calculations of spontaneous emission from an ID in a storage ring, physical optics based simulations of propagation of this partially-coherent radiation through a beamline, and the simulations of propagation of 3D time-dependent radiation pulses through instruments of X-ray Free-Electron Lasers (XFELs). The two types of radiation propagation calculations are CPU-intensive, therefore for each of them parallel algorithms have been developed in SRW. For the storage ring related calculations, the parallel processing was implemented using the Message Passing Interface (MPI). For the XFEL calculations, a shared memory approach provided by the Open Multi-Processing (OpenMP) was adopted. The two parallelization methods, and their implementation in SRW, have different advantages and drawbacks: the MPI-parallelization of partially-coherent calculations for storage rings has a good scaling, but over-consumes memory, whereas the OpenMP-parallelization of time-dependent XFEL calculations is memory-efficient, but it can only scale within one multi-core server. We are reporting the results of the efficiency tests of these two types of parallel calculations, obtained for representative optical schemes. The tests were performed on an isolated server as well on a large computer cluster - the US DOE’s NERSC scientific computing facility.

Keywords: SRW, XFEL, storage ring, synchrotron radiation, radiation propagation, partially-coherent calculations, parallel computing, MPI, OpenMP

1. INTRODUCTION

SRW [1, 2] is an open source physical optics computer code for calculation of detailed characteristics of Synchrotron Radiation (SR) generated by relativistic electrons in magnetic fields of arbitrary configuration and for simulation of radiation wavefront propagation through optical systems of beamlines with a high-accuracy, using the methods of physical optics. SRW has been used to help to better employ all great features of spontaneous SR in modern storage rings and self-amplified spontaneous and seeded emission in XFELs. The Fourier optics and compatible numerical methods are used in SRW for the simulation of fully- and partially-coherent radiation propagation through beamlines of storage ring sources [3 - 5] and through instruments of XFELs [6 - 8]. In the simulations for storage rings, the code accurately treats the effects of partial transverse coherence of the radiation beam due to finite electron beam emittance; in the simulations for XFELs, it can perform an accurate calculation of 3D X-ray pulse characteristics from the electric field in the frequency and time domains at any step of the propagation from source to sample. These two types of simulations have proved to be very useful and important for miscellaneous applications at light sources, such as the development and commissioning of new beamlines, source diagnostics, simulation of experiments and experimental data processing. However, both these calculations are also very CPU-intensive.

*ahe@bnl.gov; phone 1 631 344-8848

Advances in Computational Methods for X-Ray Optics V, edited by Oleg Chubar,
Kawal Sawhney, Proc. of SPIE Vol. 11493, 114930H · © 2020 SPIE
CCC code: 0277-786X/20/$21 · doi: 10.1117/12.2567448

To accelerate these simulations, two different parallelization methods were implemented in SRW: an MPI (Message Passing Interface) based method was implemented in the Python part of SRW to speed-up partially-coherent simulations for storage rings [9, 10], and an OpenMP (Open Multi-Processing) based method was implemented in the C++ part of SRW to speed-up time-/frequency-dependent simulations for XFELs. The MPI parallelization was implemented at the National Synchrotron Light Source II, Brookhaven National Laboratory (NSLS-II, BNL) to support the development of new beamlines at the NSLS-II and other modern low-emittance storage ring based light sources, and the OpenMP parallelization was added to the SRW at the European X-ray Free Electron Laser (EuXFEL) to accelerate calculations for instruments of the E-XFEL and other XFELs. The SRW versions supporting these two parallelization methods are extensively used not only through its Python interface, but also through the “WavePropaGator” (WPG) [11] and Sirepo [12] web-based cloud-computing frameworks.

Thanks to the availability of dedicated standard libraries, the MPI and OpenMP parallelization methods are extensively used in computer codes running on modern multi-core server computers; and these methods are also well supported on supercomputers / large computer clusters, such as the National Energy Research Scientific Computing Center (NERSC) [13]. Here we report on performances of the parallel calculations with SRW after dedicated tests on three computer systems with different numbers of cores and threads. Section 2 briefly introduces the hardware configurations of the computer systems. In section 3, the performance of the MPI calculation is analyzed using an example of partially-coherent undulator radiation and its propagation through a microscopy-type beamline in a storage ring light source. In section 4, we describe the results of testing the OpenMP parallelization performance using the example of a 3D Gaussian beam propagation through a complicated multi-crystal Inelastic X-ray Scattering (IXS) beamline that was considered for implementation at an XFEL [8]. All results are summarized and discussed in Section 5.

2. COMPUTER SYSTEMS USED IN THE TESTS

For the parallel performance testing, we used an isolated computer server at NSLS-II equipped with Intel Xeon CPU E5-2699 v4 processors operating at 2.2 GHz frequency. This server contained 44 physical cores (88 logical cores after hyper-threading), allowing to run simulations with up to 88 parallel processes with a reasonable efficiency. We also used the Cori supercomputer at NERSC. This supercomputer consists of two partitions, one with Intel Xeon “Haswell” processors and the other with Intel Xeon Phi “Knights Landing” processors, all on the same Cray “Aries” high speed inter-node network. In our tests, we used the Haswell computer nodes. Each of these nodes has 2 sockets, each socket is populated with a 16-core Intel Xeon E5-2698 v3 CPU operating at 2.3 GHz, making 32 physical cores per server node (64 logical cores after hyper-threading) [13]. The total number of the Haswell computer nodes in Cori is 2388, but in our tests, less than 20 of these nodes were used simultaneously, with a total number of parallel processes reaching 800.

3. PARALLEL PERFORMANCE OF MPI BASED SRW CALCULATIONS

3.1 Partially-coherent emission and propagation calculations for storage ring applications

Many novel experimental techniques used at modern light sources, e.g. X-ray photon correlation spectroscopy, scanning X-ray microscopy, phase-contrast and coherent diffraction imaging and others, strongly rely on X-ray coherence and brightness. With the rapid development of these techniques, the accurate calculation of the wave-optical characteristics of SR in the modern light sources, and the capabilities for simulation of fully- and partially-coherent SR wavefront propagation through optical elements of beamlines become urgently needed. Such capabilities are offered by the SRW code [1, 5].

The calculation of fully-coherent spontaneous SR (electric field at a fixed frequency) from an individual electron and propagation of this electric field through X-ray optical elements of a beamline can be done in SRW relatively quickly, within seconds or minutes on one CPU core, thanks to the use of numerically efficient methods based on Lienard-Wiechert potentials and Fourier optics [1]. The simulation of emission and propagation of partially-coherent radiation from finite-emittance electron beam circulating in a storage ring is, in the general case, a more numerically- intensive task. In SRW, such simulations are done by a function that repeats many times the fully-coherent emission and propagation calculations for individual electrons distributed over the phase space volume occupied by the electron beam [4]. The Monte-Carlo method is used to generate “initial conditions” for the electron trajectories, i.e., initial horizontal and vertical coordinates and trajectory angles at a given location of magnet lattice, and the electron energy. This method allows to accurately calculate X-ray intensity, mutual intensity / cross-spectral density, degree of coherence at any location of a beamline, up to experimental sample or even detector position, taking into account the electron beam emittance, magnet lattice parameters, magnetic field distributions in insertion devices where the SR is produced, multiple details of X-ray optical elements, including their imperfections, and details of experimental samples (in the case of simulation of experiments). To speed-up these CPU-intensive partially-coherent SR emission and propagation simulations, a parallel calculation method has been developed in the Python interface of SRW. This method makes use of the MPI through the mpi4py Python library [14]. The implemented parallel computing algorithm is based on the fact that the processes of spontaneous emission by different electrons in a storage ring are independent from each other, as well as are the processes of propagation of the waves, emitted by different electrons, through optical elements of beamlines. This allows for applying nearly "embarrassingly parallel" computing algorithm with post-processing (e.g. averaging of intensities or other characteristics of radiation originating from different macro-electrons).

We took a microscopy beamline simulation as an example to benchmark the parallel performance of this computation method. The beamline optical layout, as defined in the Sirepo interface of SRW, is shown in Figure 1. It is a simplified version of one of protein crystallography beamlines constructed at NSLS-II [15]. The Python script performing this simulation is included into the set of standard examples distributed with SRW (Python version). The optical layout contains a horizontally-focusing cylindrical mirror (HFM), a secondary source aperture (SSA, for the horizontal plane only) located at the longitudinal position of the horizontal focus, and the vertically- and horizontally-focusing elliptical cylinder mirrors (KBv and KBh) in the K.-B. geometry [16] producing sub-micron X-ray beam spot size on sample at 12.7 keV photon energy (a monochromator is assumed to be ideal and not included to the simulation). The electron beam parameters in this example are those of the NSLS-II at 0.9 nm ("day one") horizontal emittance in the low-beta straight section, and the undulator parameters are those of a canted in-vacuum undulator used at some hard X-ray beamlines of NSLS-II (21 mm period, 1.5 m length, maximum deflection parameter ~1.8).

In the SRW function performing partially-coherent emission and propagation calculations, there are two parameters allowing to tune the accuracy and the execution time of the parallel partially-coherent calculations. One of these parameters that we denote here as "n_{tot}" is the total number of macro-electrons to be used in the calculation, the other one, that we denote here as "n_{avg}" is the number of macro-electrons to be used in calculation in each "worker" process for averaging the propagated radiation intensity data before sending it to the "main" process at every communication with it. In the following testing, the MPI calculations with different number of processes were performed for different values of n_{tot} and n_{avg} parameters.

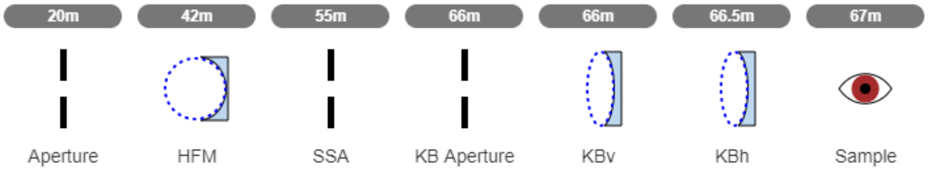

Figure 1. Optical layout of a microscopy beamline defined in Sirepo and used for testing the efficiency of parallel partially-coherent calculations.

3.2 Partially-coherent calculation results

To study the impact of a total number of macro-electrons on the simulation convergence, we repeated the entire partially-coherent emission and propagation calculation of the microscopy beamline with n_{tot} = 5000, 10000, 20000, 30000, 40000, 50000, 500000. The calculation results are illustrated in Figure 2, which shows the intensity distribution of the 12.7 keV radiation on the sample plane at fully open secondary source aperture for n_{tot} = 5,000 (Figure 2-a) and for n_{tot} = 500,000 (Figure 2-b). As one can see from Figure 2, both intensity distributions are in good qualitative agreement. In the vertical plane (at the horizontal position x = 0) the intensity distributions are close even numerically (see graphs on the right in Figure 2-a,b). However, in the horizontal plane (where the electron beam emittance is much larger), the intensity distribution obtained for n_{tot} = 5,000 contains some numerical noise, which is not present in the calculation result for n_{tot} = 500,000 (see graphs on the middle in Figure 2-a,b). In Figure 3, we show the horizontal (3-a) and vertical (3-b) cuts of the intensity distribution by the horizontal and vertical median planes (i.e. at y = 0 and x = 0 respectively) for the whole set of values of n_{tot} from 5,000 to 500,000. The calculations illustrated in Figures 2 and 3 were carried out at NERSC supercomputer. For the case of n_{tot} = 500,000, we requested a totally 1280 cores on 40 nodes, and it took ~70 minutes to complete the calculation.

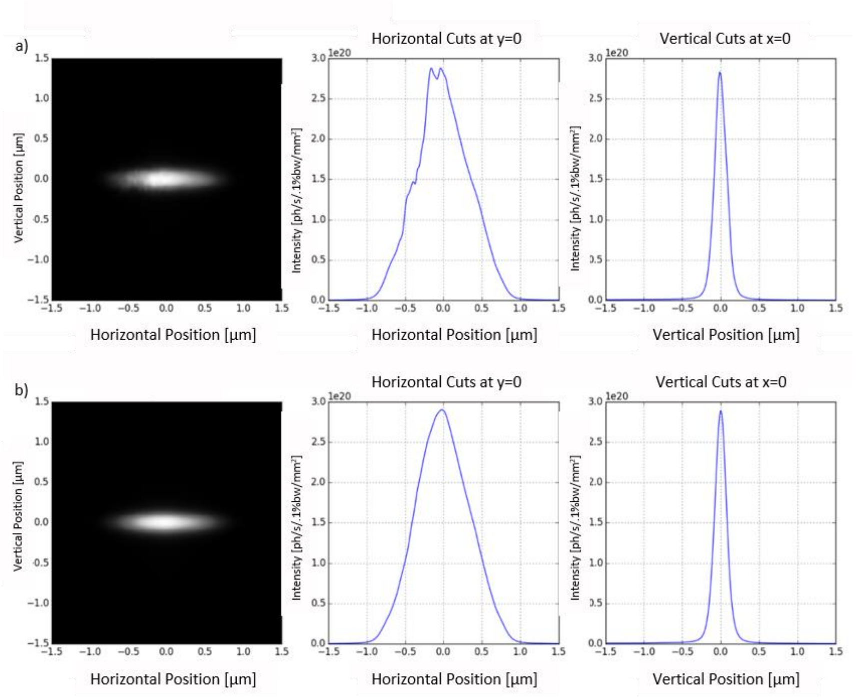

Figure 2. Intensity distributions at the photon energy of 12.7 keV calculated in the sample plane, for the simulation of the partially-coherent emission and propagation of the microscopy beamline with different numbers of macro-electrons: a) $n_{tot} = 5000$, $n_{avg} = 50$ and b) $n_{tot} = 500{,}000$, $n_{avg} = 50$. The left column plots represent 2D intensity distributions, and the center and the right column are the intensity cuts by the horizontal (y = 0) and vertical (x = 0) mid-planes respectively.

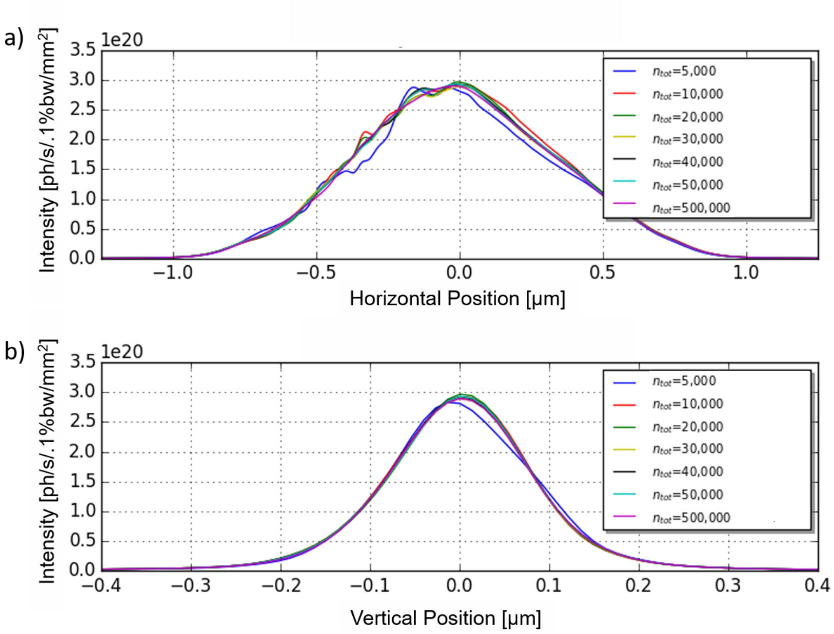

Figure 3. Variation of the intensity distributions at 12.7 keV photon energy calculated in the sample plane, for the simulation of the partially-coherent emission and propagation for the microscopy beamline with n_{tot} varying from 5,000 to 500,000: a) horizontal intensity distribution cuts at y = 0; b) vertical intensity distribution cuts at x = 0.

3.3 Parallel performance of MPI calculations

We first tested the execution time of the partially-coherent simulations for the microscopy beamline on an isolated computer server at NSLS-II (containing 44 physical cores, i.e. 88 after the hyper-threading) vs. the number of requested processes (cores). In the simulation, we set n_{tot} = 5,000 and n_{avg} = 5. The results of these tests are shown in Figure 4. The parallel computation speed-up rate in this case for a small number of processes is ~0.64, and it reduces to zero with the increase of the number of processes, reaching saturation at ~60 processes (which is larger than the number of physical cores and lower than the number of logical cores after the hyper-treading). This is likely to be explained by architecture of the server used in the tests.

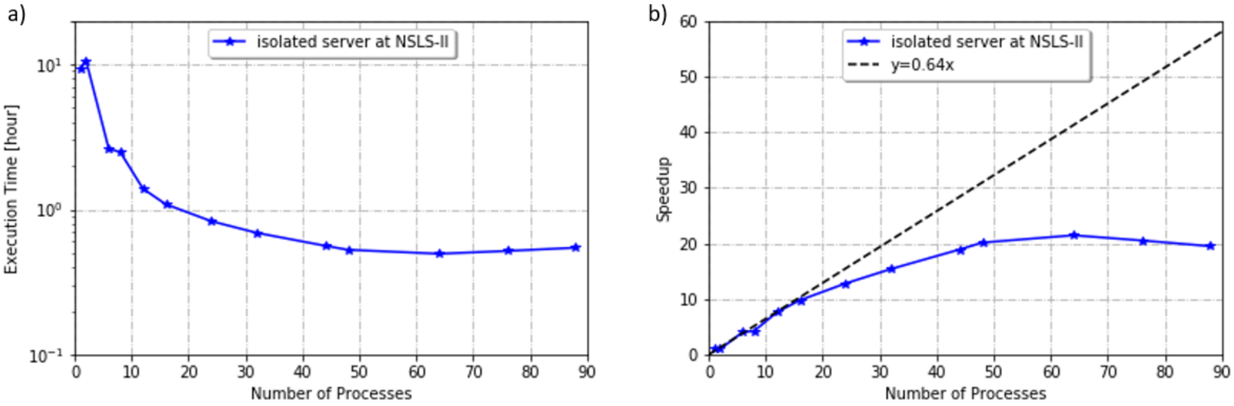

Figure 4. Parallel performance of MPI based calculations on an isolated server at NSLS-II with n_{tot} = 5,000 and n_{avg} = 5: a) dependence of the execution time on the number of processes (cores) used in the calculation; b) the corresponding calculation speed-up compared to the sequential execution.

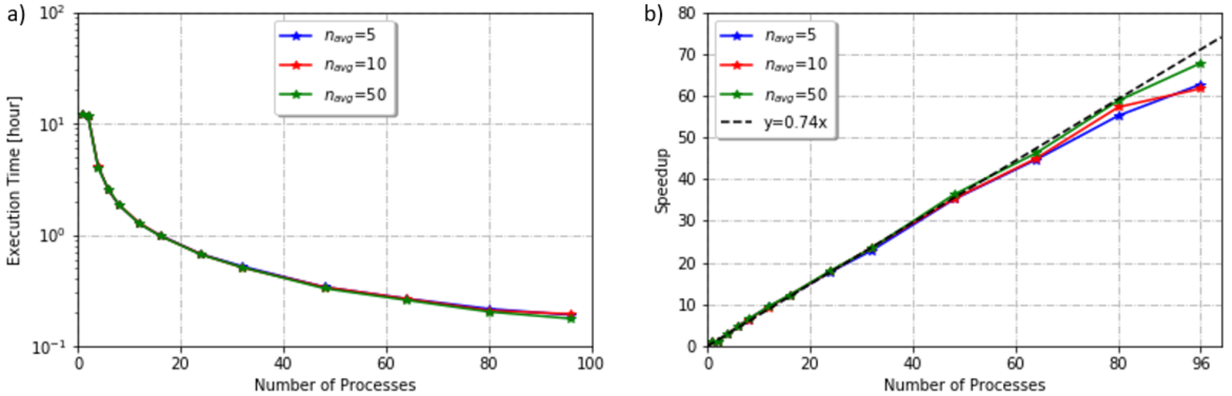

Figure 5. Parallel performance of MPI based calculations at NERSC supercomputer for n_{tot} = 5,000 at n_{avg} = 5, 10, 50: (a) dependence of the execution time on number of processes (cores) used in the calculation; (b) the corresponding calculation speed-up compared to the sequential (single process) execution.

We further tested the execution time of the partially-coherent simulations for the microscopy beamline at NERSC. In these tests, we used the two fixed values of the total number of macro-electrons, n_{tot}, equal to 5,000 and 50,000, and then measured the execution time with different numbers of requested cores. First, we set n_{tot} = 5,000 and scanned the execution time of the simulation at a different number of requested cores, from 1 to 96, with different number of macro-electrons simulated and averaged in each worker process (defined by n_{avg}). Figure 5 illustrates the parallel performance at NERSC with n_{tot} = 5,000 and n_{avg} = 5, 10, 50.

For the MPI based parallel calculation in the SRW function performing partially-coherent emission and propagation simulation, the exact total number of macro-electrons to be used in a simulation is decided by the values of n_{tot}, n_{avg} and the number of requested processes (cores), to make the total numbers of macro-electrons treated by each worker process equal. Therefore, for some cases, the actual number of macro-electrons contributed to the radiation intensity was less than the input setting of n_{tot}. In such cases we scaled the execution time by taking the actual number of macro-electrons into account. The results of these tests (after the scaling) are shown in Figure 5-a. We can see that with the increase of the number of processes (cores), the execution time has decreased from ~12 hours (when running sequentially on 1 core) to ~11 minutes (when running on 96 cores). Figure 5-a also illustrates that the execution time of running on 1 core is nearly the same as that on 2 cores. That’s because MPI uses the main-worker system, when one (main) process is assigned to manage the entire calculation, and the other (worker) processes execute sub-tasks in parallel (being guided by the main). A more straightforward illustration of the parallel performance can be found in Figure 5-b which shows that the rate of speed-up is about 0.74. We note that, ideally, this rate would be approaching 1 (at a large number of processes). The fact that it is less than 1 can be explained by contributions to the execution time from communications and some non-parallelized operations, e.g. when main receives data from workers, averages and saves it to intermediate files, etc. Figure 5-b shows that n_{avg} doesn’t have much effect on the simulation speed when the number of requested cores is less than 55. With the increase of the number of processes, the speed-up curves for different n_{avg} gradually separate, with the speed-up being somewhat smaller for smaller values of n_{avg}. However, at $n_{avg} = 50$, the speed-up rate becomes almost fully restored back to “normal”. This happens because the larger is the value of n_{avg}, the fewer times the main needs to receive and average data from workers, spending less time on communication and non-parallelized operations. This trend becomes more obvious at further increase of the number of parallel processes (see below).

We then increased the value of n_{tot} to 50,000, and requested more cores to further study the parallel performance of the MPI calculation. We scanned the execution time of the simulations with the number of requested processes (cores) from 64 to 768. In each simulation, we used different numbers of macro-electrons for averaging intensity in each worker process: n_{avg} = 5, 10, 50, 100. Figure 6 illustrates the parallel performance of the MPI calculations, obtained from these tests executed at NERSC. The obtained speed-up was by more than a factor of 350 compared to the sequential calculation. For example, for the case of $n_{avg} = 100$ the calculation on 768 cores completed in 20 minutes instead of 120 hours that were required for the same calculation executed sequentially.

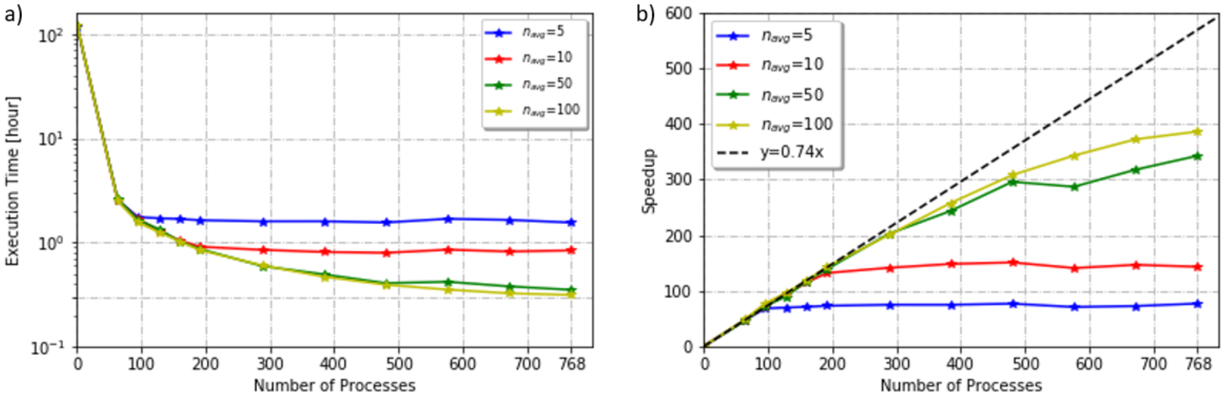

Figure 6. Parallel performance of MPI based calculations at NERSC for n_{tot} = 50,000 at n_{avg} = 5, 10, 50, 100: a) dependence of the execution time on number of processes (cores) used in the calculation; b) the corresponding calculation speed-up compared to the sequential execution.

We can also see from Figure 6 that for the calculations with large numbers of processes (cores), the impact of the n_{avg} parameter on the parallel performance becomes very significant. E.g. at $n_{avg} = 5$ the speed-up never exceeds the factor of 80 (see blue line in Figure 6-b), whereas at $n_{avg} = 100$, it is close to a factor of 400 (for 768 processes, see yellow line in Figure 6-b). The trend for the “saturation” of the parallel performance of this MPI based calculation with the increase of the number of parallel processes at fixed n_{avg} is explained by gradual increase of the portion of time spent on the main - worker communications and non-parallelized processing. The increase of n_{avg} allows to reduce this “overhead” and the trend to saturation quite significantly. This confirms the validity and high efficiency of the MPI based parallel processing algorithm implemented for partially-coherent calculations in SRW.

4. PARALLEL PERFORMANCE OF OPENMP BASED SRW CALCULATIONS

4.1 Time-/frequency-dependent X-ray pulse propagation calculations for XFEL applications

The efficient use of precise time- and frequency-domain characteristics of radiation pulses generated by XFEL facilities in user experiments demands for the accurate simulation of the processes of Self-Amplified Spontaneous Emission (SASE) and seeded XFEL emission, pulse propagation through X-ray optical elements along instruments, their interaction with samples and propagation of the scattered radiation to detectors. The high-accuracy calculation of XFEL emission can be done by such proven software tools as GENESIS [17] and FAST [18]. These codes generate electric field of radiation pulse at the output of XFEL undulator as a function of transverse coordinates and time or frequency. This electric field can then be used by SRW code for the physical optics based simulation of the pulse propagation. Such calculations are supported in SRW by using the Fourier-optics and compatible wavefront propagation methods, as well as the methods for "extraction" and analysis of XFEL radiation characteristics in the frequency and in time domains [6]. In simple cases, one can also use e.g. electric field of a fully coherent Gaussian beam as input for the wavefront propagation calculations. The electric field propagation calculations are usually performed in the frequency domain, at a number of frequency values. In some cases, e.g. for SASE pulses, this number of frequency values needs to be high (e.g. several thousands), with each frequency "slice" of the electric field being sampled on a transverse grid containing thousands of points vs. horizontal and vertical positions. When executed sequentially, such CPU-intensive and memory-consuming pulse propagation calculation may take quite some time (e.g. hours).

To speed-up X-ray pulse propagation simulations for XFEL applications, an OpenMP based parallelization method was implemented in the core C++ part of SRW by the European XFEL. These developments were merged with the open source version of the SRW code [2]. To avoid any potential conflicts with the MPI based parallel partially-coherent calculations for storage rings (see previous section), it was decided to introduce a special compilation option of the SRW C++ library for producing the OpenMP parallelized version optimized for the XFEL simulations. This compilation option, controlled by a preprocessor directive, is available both in the SRW Microsoft Visual C++ project for Windows and in the Makefile that is used for compiling SRW for Linux / MacOS (with GCC or compatible compilers).

To test the parallel performance of OpenMP based calculations in SRW, we took an example that was set up earlier to simulate XFEL pulse propagation through a complicated IXS instrument [8] containing a large number of crystals and compound refractive lenses. In these calculations, most CPU time is spent on simulation of the 3D electric field propagation through the X-ray optics of the instrument / beamline at multiple frequencies, and on changing the field representation between the frequency and time domains. Some execution time is also spent on saving (sequentially) the calculation results to files, however, this time was excluded from the parallel performance test results presented below. To study performance of the OpenMP based parallelization, we ran the same pulse propagation calculation, with the input pulse simulated by a 3D Gaussian beam, using different number of threads, that was set via the environmental variable OMP_NUM_THREADS. One server of NSLS-II and one node of the NERSC Cori cluster were used in these tests.

4.2 Calculated X-ray pulse characteristics at sample

Before starting the performance tests of the OpenMP based calculation, we made sure that the physical results generated by the parallel calculation (both with the version of SRW library produced by the European XFEL and the version compiled after the code merging) are identical to the results produced by the sequential calculation. Some of these calculation results, describing temporal, spectral, and spatial characteristics of the 3D X-ray pulse after the propagation from XFEL exit to sample position, are presented in Figure 7.

4.3 Parallel performance of OpenMP calculations

Figure 8 illustrates the parallel performance test results of the OpenMP based calculations of the EXFEL_IXS simulation obtained at the NERSC Cori's Haswell node and a server at NSLS-II (see section 2 for the description of the hardware). Figure 8-a shows the dependence of the execution time on the number of requested threads, and Figure 8-b presents the same results in terms of the speed-up of parallel calculations with respect to the sequential calculation with only one thread. As one can see from Figure 8-b, the rate of the speed-up is about 0.8 when the number of requested threads is smaller than 10; however, at a larger number of threads, the speed-up rate gradually drops down ~0. The maximal obtained speed-up rate is about 20.

We note that since the time when the tests of the efficiency of the OpenMP parallelization described here were done, further developments and fixes were implemented in the SRW code, which increased the efficiency and scaling of this type of parallel calculations. However, these improvements did not change the principal limitation of the OpenMP based parallelization method – its scalability only within the maximum number of threads supported by one server.

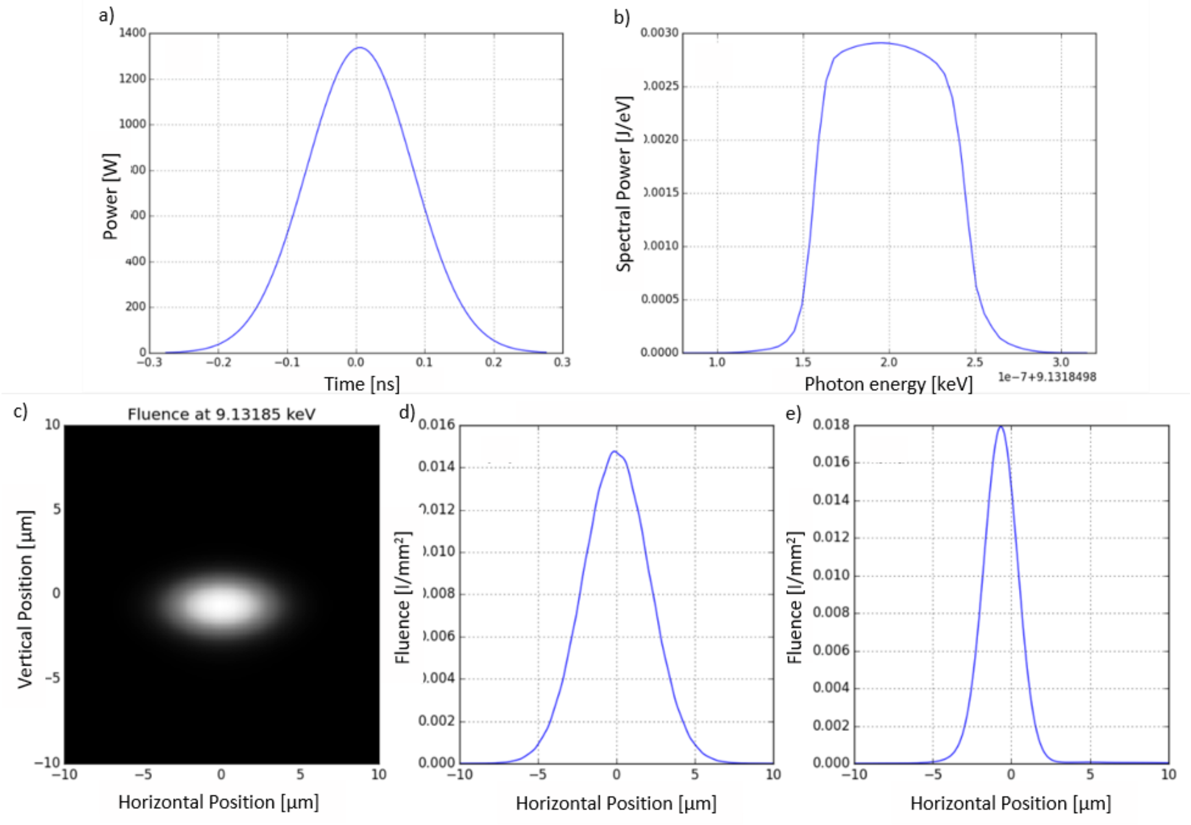

Figure 7. Calculated temporal, spectral, spatial distributions of the radiation pulse in the sample plane of a hypothetic IXS instrument at an XFEL: a) power vs. time (pulse duration is ~160 ps at the FWHM); b) photon energy spectrum (the bandwidth is ~80 µeV at the FWHM); c) image plot of the 2D spectral fluence distribution at the peak photon energy (the FWHM dimensions are 5.5 µm (h) x 2.5 µm (v)); d) horizontal cut through the maximum of the fluence distribution; and e) vertical cut.

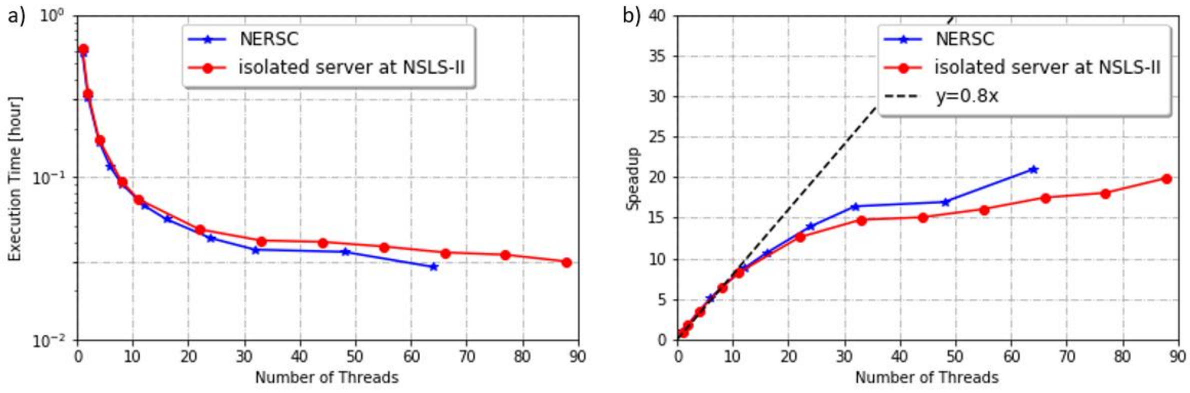

Figure 8. The parallel performance of OpenMP based calculations at NERSC and on a server at NSLS-II: a) dependence of the execution time on the number of threads used in the calculation; b) the corresponding calculation speed-up compared to the sequential execution.

5. SUMMARY AND DISCUSSION

SRW computer code supports both MPI and OpenMP based parallel calculations. The MPI calculation is programmed in the Python part of the code to simulate the spontaneous emission and propagation of partially-coherent X-rays through a beamline of a storage ring source. The OpenMP parallelization is implemented in the C++ part of the code to simultaneously calculate wavefront propagation at different photon energies ("slices" of a 3D pulse) for XFEL sources and instruments. We tested the parallelization performances of the MPI and the OpenMP calculations by running two SRW simulation scripts on isolated servers of NSLS-II and on DOE's NERSC cluster. The MPI based SRW calculation was shown to scale well with the number of processes used: the tests were performed with up to 768 processes and the obtained speed-up was by more than a factor of ~350 compared to a sequential calculation, with the obtained maximal speed-up rate ~0.74. In the case of the OpenMP based SRW calculation, the obtained maximal speed-up was by a factor of 15-20 with the maximal rate of ~0.8. The obtained smaller scaling of the OpenMP calculation seems to be typical for this parallelization method: it uses the shared memory multi-threading approach that is efficient only at the execution on one computer.

The parallel performance of SRW could possibly be further improved. In the case of pure MPI based calculations, this can be done by reducing the number of non-parallelized operations and / or moving these operations from Python to C++ part of the code to increase their CPU-efficiency. The performance of the OpenMP based parallelization could possibly be improved by using a hybrid method, with combined use of the OpenMP and MPI libraries. An independent future effort could consist of parallelizing some portions of the code for the execution on GPU.

ACKNOWLEDGMENTS

Authors are very grateful to G. Geloni (European XFEL), D. Bruhwiler (RadiaSoft LLC) for the collaboration on SRW code development and applications. The work was supported in part by US DOE contracts DE-SC0012704 and DE-SC0011237 and by US DOE BES Field Work Proposal PS-017. This research used the resources of the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the US Department of Energy under the Contract No. DE-AC02-05CH11231.

REFERENCES

[1] Chubar, O. and Elleaume, P., "Accurate and Efficient Computation of Synchrotron Radiation in the Near Field Region", Proc. of EPAC-98, pp.1177-1179 (1998).

[2] https://github.com/ochubar/SRW/

[3] Chubar, O., Elleaume, P., and Snigirev, A., "Phase Analysis and Focusing of Synchrotron Radiation", Nucl. Instrum. and Methods A435, 495-508 (1999).

[4] Chubar, O., Elleaume, P., Kuznetsov, S., Snigirev, A., "Physical Optics Computer Code Optimized for Synchrotron Radiation", Proc. SPIE 4769, 145 (2002).

[5] Chubar, O. et al., "Development of partially-coherent wavefront propagation simulation methods for 3rd and 4th generation synchrotron radiation sources," Proc. Of SPIE Vol. 8141, 814107 (2011)

[6] Chubar, O. et al., "Time-dependent FEL wavefront propagation calculations: Fourier optics approach", Nucl. Instr. and Meth. A593, 30 (2008).

[7] Roling, S. et al., "Time-dependent wave front propagation simulation of a hard X-ray split-and-delay unit: Towards a measurement of the temporal coherence properties of X-ray free electron lasers", Phys. Rev. ST Accel. Beams 17, 110705 (2014)

[8] Chubar, O. et al., "Ultra-high-resolution inelastic X-ray scattering at high-repetition-rate self-seeded X-ray free-electron lasers", J. Synchrotron Rad. 23, pp. 410-424 (2016).

[9] Chubar, O.et al., "Wavefront propagation simulations for beamlines and experiments with 'Synchrotron Radiation Workshop'," J. Phys.: Conf. Ser. 425, 162001 (2013).

[10] Laundy, D. et al., "Parallel simulations of partially coherent wavefront propagation from a finite emittance electron beam," J. Phys.: Conf. Ser. 425, 162002 (2013)

[11] Samoylova, L., Buzmakov, A., Chubar, O., Sinn, H., "WavePropaGator: interactive framework for X-ray free-electron laser optics design and simulations", J. Appl. Crystallogr., 49(Pt 4), pp. 1347-1355 (2016).

[12] Rakitin, M.S., Moeller, P., Nagler, R., Nash, B., Bruhwiler, D.L., Smalyuk, D., Zhernenkov, M., and Chubar, O., "Sirepo: an open-source cloud-based software interface for X-ray source and optics simulations", J. Synchrotron Rad. 25, 1877-1892 (2018).

[13] http://www.nersc.gov

[14] https://pypi.org/project/mpi4py/

[15] Fuchs, M.R. et al., "NSLS-II biomedical beamlines for micro-crystallography, FMX, and for highly automated crystallography, AMX: new opportunities for advanced data collection", Proc. of SRI-2015, AIP Conf. Proc. 1741, 030006 (2016).

[16] Kirkpatric, P.; Baez, A. V., "Formation of optical images by x-rays". Journal of the Optical Society of America. 38 (9), 766–74 (1948).

[17] Reiche, S., "GENESIS 1.3: a fully 3D time-dependent FEL simulation code", Nucl. Instrum. and Methods A429, 243-248 (1999).

[18] Saldin, E. L., Schneidmiller, E. A., Yurkov, M. V., "FAST: a three-dimensional time-dependent FELsimulation code", Nucl. Instrum. and Methods A429, 233-237 (1999).