RECENT PROGRESS ON PARALLEL ELEGANT *

Y. Wang †, M. Borland, H. Shang, R. Soliday, A. Xiao, ANL, Argonne, IL 60439, USA

Abstract

The electron accelerator simulation software `elegant` [1] is being parallelized in a multi-year effort. Recent developments include parallelization of input/output (I/O), frequency map analysis, dynamic aperture search, and position-dependent momentum aperture determination. Parallel frequency map, momentum aperture analysis, and dyanmic aperture search provide rapid turnaround for important determinants of storage ring performance. The development of parallel Self-Describing Data Sets file (SDDS) I/O based on MPI-IO made it possible for parallel `elegant` (`Pelegant`) to take advantage of parallel I/O. Compared with previous versions of `Pelegant` with serial I/O, the new version not only enhances the I/O throughput with good scalability but also provides a feasible way to run simulations with a very large number of particles (e.g., 1 billion particles) by eliminating the memory bottleneck on the master with serial I/O. Another benefit of using parallel I/O is reducing the communication overhead significantly for the tracking of diagnostic optical elements, where the particle information has to be gathered to the master for serial I/O.

INTRODUCTION

The parallel version of `elegant`, `Pelegant`, has proved to be very beneficial to several computationally intensive accelerator research projects. Simulation with a very large number of particles is essential to study detailed performance of advanced accelerators. This was demonstrated in simulations of microbunching for FERMI [2]. In those simulations the maximum number of particles was reached at about 60M when the serial version of SDDS was used, which limited our ability to probe microbunching effects at shorter wavelengths. In the version of `Pelegant` used in those studies, the bottleneck came from the memory usage of the master CPU, which was required to hold all the particle information when simulating a diagnostic element, such as a watch point, where all the particles have to be gathered to master to be written on the disk.

The recent development of parallel SDDS [3] makes it possible for `Pelegant` to take advantage of parallel I/O through MPICH2 [4]. With parallel I/O, a common file is opened by all the processors, but each processor is only responsible for reading/writing the particles allocated to it. This technique improved I/O throughput significantly, especially on some parallel file systems, such as Parallel Virtual File System (PVFS) [5] and General Parallel File System (GPFS) [6]. `Pelegant` can also run on Network File System (NFS) file system, although the I/O performance is not as good as on the parallel file systems. The overall performance of `Pelegant` on all the file systems mentioned above has also been improved due to reduced communication overhead compared with gathering particles to master before writing to the disk with serial I/O. A nice feature of this parallel SDDS I/O is that the output/input files are the same as the files for serial I/O, which is very convenient for data analysis and exchanging data in SDDS format with other related simulation programs.

In this paper, we first describe the effort we made to integrate parallel SDDS with `Pelegant`, then we report the progress made on the parallelization of frequency map, momentum aperture analysis, and dynamic aperture optimization in `Pelegant`.

IMPLEMENTATION OF `PELEGANT` WITH PARALLEL SDDS

The simulation code `elegant` is being gradually parallelized with particle-based domain decomposition to reduce the simulation time for multi-particle beams. Beamline elements are classified in the element dictionary as parallel-capable or serial-only. Particles will be gathered to the master CPU or scattered to slave CPUs when the beam encounters a serial element or a parallelized element [7, 8], respectively. As the majority of the frequently used elements has been parallelized, `Pelegant` has been efficiently used for several important accelerator research projects [2, 9, 10, 11, 12].

Even in cases where one must use beamline elements that are not yet parallel-capable, a very significant performance improvement can be realized. However, for simulations with large numbers of particles, I/O for input, intermediate output, and final output, can consume a significant portion of simulation time. In addition to the communication overhead of gathering particles to the master, memory also becomes a problem when we simulate a very large number of particles with a central process (i.e., Master) holding all the particle information for I/O operations.

To eliminate these bottlenecks, we developed parallel SDDS [3] with MPI-IO recently. The parallel SDDS is derived from the serial version of SDDS [13], which has been successfully applied to several accelerator simulation

* Work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02-06CH11357.

† ywang25@aps.anl.gov

codes. We made significant efforts to integrate Pelegant with parallel SDDS:

1. For large-scale simulation, if one element in the lattice is not parallelized, the gather-scatter procedure will downgrade the performance of Pelegant, and the memory problem will appear. So we parallelized almost all of frequently used elements. A warning message will be given whenever a serial element is tracked.

2. Besides the final output of the simulation, some intermediate outputs from elements have to be adapted with parallel I/O, which was previously done by the master. Such I/O becomes impossible for simulations with extremely large numbers of particles.

3. The statistics information for the particles allocated to each slave will be calculated locally first, then the master will calculate the global statistics results based on the information provided by the slaves. Both the communication overhead and computation load on master are significantly reduced compared with previous versions, where the master gathers all the particles to perform the statistics calculation. The parallelization of the statistics calculation is not straightforward, as elegant provides more than 100 statistical parameters for the properties of the output beam. Parallelization of these statistical calculations requires a number of strategies to meet the different requirements of the specific statistic.

4. Validating results through numerical comparison with the serial version result is a challenge when simulating with a randomly generated beam, as the number of random numbers generated on each of the CPUs is unpredictable, especially for a beam with some cut-off criteria in more than one dimension. elegant provides several different types of beam generation. For certain types of beam distributions, we were able to make elegant generate the same random sequence as Pelegant by using the same set of random seeds as Pelegant and generating beam sequentially with several iterations. For some types of beams, it is essentially impossible in any straightforward way for Pelegant and elegant to generate the same sequence, and we have to do the reconciliation by visualization.

Pelegant with parallel SDDS has been successfully applied to accelerator research and operations at the Advanced Photon Source (APS) at Argonne National Laboratory. To test the performance of the new version of Pelegant, we chose a simulation requiring a significant amount of I/O operations. The system being modeled is a very large energy recovery linac (ERL) upgrade of the APS [11]. It includes a two-pass 7-GeV linac, nine 48-m-long undulators, and numerous transport-line magnets. Beam is accelerated from 10 MeV to 7 GeV and then decelerated through the same linac. Modeling includes rf acceleration with exact time dependence, coherent synchrotron radiation, longitudinal space charge, wakefields, quantum excitation, and beam apertures.

There are nearly 20 watch points in the system, providing valuable information about the phase space at important locations. In addition, statistics are computed at the exit of every beamline element. The input beam is read from an SDDS file. Because of these factors, this simulation would be inefficient using the previous version of Pelegant for a large number of simulation particles, due to both I/O and memory bottlenecks.

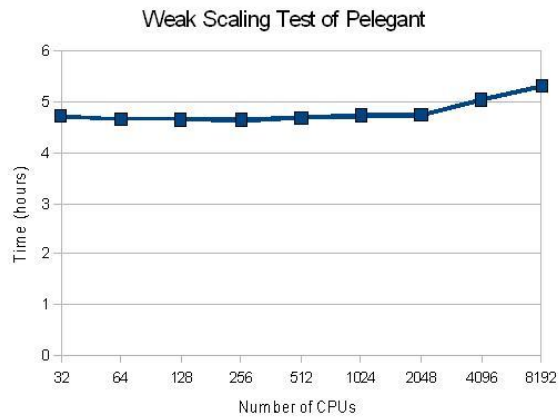

Figure 1: Weak scaling test of Pelegant on NERSC's Cray XT4 Franklin supercomputer.

We did a weak scaling test by increasing the number of particles in proportion to the number of CPU cores (i.e., keeping workload per CPU same) on the Cray XT4 Franklin supercomputer at National Energy Research Scientific Computing Center (NERSC). The performance test started from 448K particles on 32 cores and ended with 115 million particles on 8192 cores. This is significant because 115 million particles is approximately the actual number of electrons in the 19 pC beam.

From Figure 1, we can see that Pelegant achieved optimal performance when the number of cores is less than or equal to 2048. When the number of cores is above 4096, it takes a little bit more time than the test with fewer CPU cores due to the I/O scalability limit and communication overhead. A test with a quarter billion particles was also conducted on the Franklin supercomputer, but we ran out of allocated CPU hours. The simulation with a quarter billion particles is expected to be done in about 6.5 hours, which is still reasonably good from an efficiency point of view. Technically speaking, Pelegant should be able to simulate a billion particles efficiently. We can also increase the number of particles for each core (longer run time is expected) to run Pelegant more efficiently, as the memory requirement for the weak scaling test designed above is just under 1 percent of 2GB memory for each core on the Franklin supercomputer.

PARALLELIZATION OF FREQUENCY-MAP ANALYSIS

The frequency-map analysis command in elegant is very useful to quickly identify resonances in a circular accelerator. The task is to track particles with a grid of starting coordinates and determine frequencies of x and y mo-

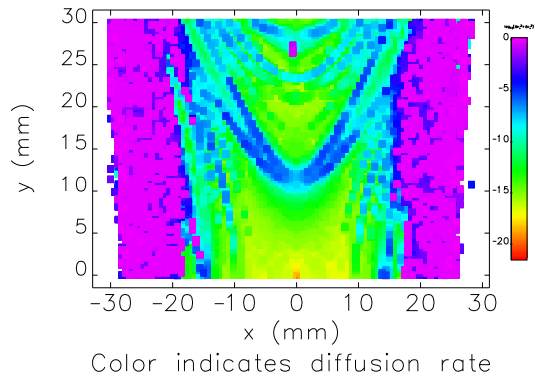

Figure 2: Result of parallel frequency map analysis with Pelegant.

tion. In the parallel version, the simulation is done by distributing the starting coordinates to different CPUs and tracking the particles simultaneously. We chose 2D domain decomposition to parallelize this operation. As there is no inter-process communication requirement for the tracking of each grid point, and the final frequency map results from all the processors are written with parallel I/O, a very good speedup of this type of simulation can be achieved. For example, we set the number of grid points in the x direction to 100 and the number in the y direction to 80. The simulation was done with 3 hours and 35 minutes with serial elegant, while it took 3.5 minutes on 100 CPUs on a cluster at APS. To use this operation efficiently, a user should choose the total grid points to be a multiple of the number of CPUs.

Figure 2 shows the frequency map for the APS Particle Accumulator Ring (PAR) lattice as computed with Pelegant. With this parallelized command, the user can choose very fine grids and get the frequency-map analysis result very quickly. This result took under 10 minutes on an 8-core desktop.

PARALLELIZATION OF MOMENTUM APERTURE SIMULATION

The position-dependent momentum aperture[14] is an important aspect for storage ring optimization, as it strongly affects the Touschek lifetime. The purpose of this simulation is to determine the range of momentum deviation for which the particle will survive as a function of starting longitudinal position. The end of each element will be used as the longitudinal position to scan, and hundreds of passes must be tracked for each starting longitudinal position. For a damping ring with a large number of elements, such as the International Linear Collider (ILC) damping ring, it would take several days or weeks to finish one simulation on a single processor. For this type simulation, we chose element decomposition for parallelization. The lattice is partitioned to several segments that are distributed to different CPUs. An equal number of elements are allocated to each processor. The workload is largely independent of where the simulation starts, so the static load balancing will be good enough for efficiency considerations.

We did a test with 780 elements, which is a small portion of elements from the ILC damping ring, on a cluster at APS. The simulation was done in 55 hours and 50 minutes with the serial elegant. It took 7 hours and 13 minutes on 8 CPUs, and 17 minutes on 250 CPUs. This is another application where Pelegant allows finishing large simulations quickly enough to provide a useful design tool for day-to-day work.

DYNAMIC APERTURE OPTIMIZATION WITH PELEGANT

Dynamic aperture is another important aspect of storage ring optimization. We have successfully parallelized the line search mode to find the dynamic aperture. The line mode searches for the aperture boundary starting from the origin and moving outward. The exploration starts from (0, 0) to $(x_{max} * sin(\theta), y_{max} * cos(\theta))$, where θ takes values from $-\pi/2$ to $\pi/2$. The area of the dynamic aperture is given in the output file with a parameter called "Area"[15]. The easiest way to parallelize this application is to distribute the workload to different processors with different θs. In this case, the maximum number of CPUs would be limited to the number of lines to search. As the number of grid points in a line is usually greater than the number of lines to search, a better approach would be distributing different grid points of a line for each θ to different CPUs. First, the origin point is tracked by all the processors. If the particle in the origin survives from tracking, then the next n_c grid points close to the origin will be distributed to each CPU, where n_c is the total number of CPUs. This procedure will continue until a particle is lost for a grid point, or the boundary of the searching area is reached on a CPU. The last surviving grid point on each CPU will be returned to the master CPU. Finally, Pelegant finds the closest point to the origin from all the returned points, then searches the dynamic aperture boundary point for the next line.

The implementation mentioned above will not reach optimal performance as one processor has no knowledge of whether the other processors have reached the boundary of the aperture. But it still achieved very good efficiency when the number of grid points in each line is a multiple of the number of CPUs. Figure 3 shows an example the APS storage ring 400-turn dynamic apertures for 20 error ensembles using Pelegant. This took under 70 minutes on an 8-core desktop.

CONCLUSION

The capability of Pelegant running large-scale simulation has been significantly enhanced after successful integration with parallel SDDS, which eliminates the bottleneck caused by serial I/O. The program shows good

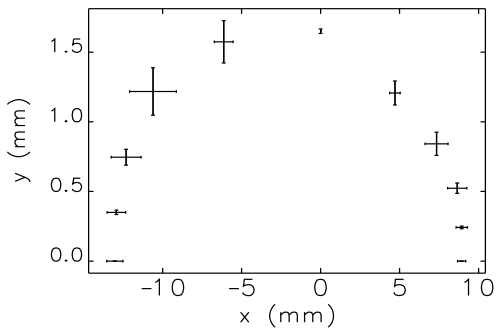

Figure 3: Dynamic aperture for 20 error ensembles for the APS storage ring.

scalability with a very large number of CPU cores on the Franklin supercomputer at NERSC. Simulation with hundreds of millions of particles can be done efficiently within reasonable time. Several important operations in `elegant`, such as frequency map analysis, dynamic aperture search and momentum aperture determination, can now be run on a multi-processor desktop or cluster with `Pelegant`.

ACKNOWLEDGMENT

These studies are partially supported by the SciDAC projects "Community Petascale Project for Accelerator Science and Simulation (ComPASS)" and "Frontiers in Accelerator Design: Advanced Modeling for Next-Generation BES Accelerator". The ComPASS project is supported under the SciDAC program by the U.S. Department of Energy Office of High Energy Physics, Office of Nuclear Physics, Office of Basic Energy Sciences, and Office of Advanced Scientific Computing Research. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

REFERENCES

[1] M. Borland, "elegant: A Flexible SDDS-compliant Code for Accelerator Simulation," APS LS-287, (2000).

[2] M. Borland, "Modeling of the microbunching instability," Phys. Rev. ST Accel. Beams 11, 030701 (2008).

[3] H. Shang, Y. Wang, R. Soliday, M. Borland, "Parallel SDDS: A Scientific High-Performance I/O Interface," these proceedings.

[4] http://www.mcs.anl.gov/research/projects/mpich2/

[5] http://www.pvfs.org

[6] http://www.ibm.com/systems/clusters/software/gpfs.html

[7] Y. Wang and M. Borland, "Pelegant: A Parallel Accelerator Simulation Code for Electron Generation and Tracking," Proceedings of 12th Advanced Accelerator Concepts Workshop, (2006).

[8] Y. Wang and M. Borland, "Implementation and Performance of Parallelized Elegant," Proc. PAC07, (2007)

[9] M. Borland and V. Sajaev, Proc. PAC05, 3886-3888 (2005).

[10] R. Bartolini, *et al.*, Proc. EPAC06, 160-162.

[11] M. Borland *et al.*, "Direct Methods of Optimization of Storage Ring Dynamic and Momentum Aperture," Proc. PAC09, to be published.

[12] M. Borland, *et al.*, "Possible Upgrade of the Advanced Photon Source with an Energy Recovery Linac," Proc. PAC09, to be published.

[13] M. Borland, "A Self-Describing File Protocol for Simulation Integration and Shared Postprocessors," Proc. PAC 95, 2184-2186 (2006).

[14] M. Belgroune *et al.*, "Refined Tracking Procedure for the SOLEIL Energy Acceptance Calculation," Proc PAC 2003, 896-898 (2003).

[15] http://www.aps.anl.gov/asd/oag/manuals/elegant_latest/elegant.html