LLNL-TR-648366

pF3D Proposals

S. Langer

January 12, 2014

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Quarterly Update
Simulations of Laser-Plasma Interaction in National Ignition Facility Experiments.
INCITE 2013 Award
P.I. Steve Langer

Report on Project Milestones

- Provide status on each of your project's simulations milestones as outlined in your original proposal

  We completed a 3 quad simulation of a NIF experiment on Mira during 4Q13. The run simulated Stimulated Raman Scattering (SRS) and Stimulated Brilliouin Scattering (SBS). The simulation extended 0.75 mm along the laser propagation direction. The SRS comes in bursts in this simulation. The simulation was run for 64.9 ps, long enough for several SRS bursts. SRS and SBS can interact through depletion of the incident laser light. This simulation ran long enough to allow the slower growing SBS time to build up. The simulation showed that, for these conditions, SRS is much stronger than SBS.

- List major accomplishments thus far in CY2013. Please include scientific and computational details of simulations undertaken, including images if possible.

  pF3D has run reliably with 256K MPI processes (32 MPI processes per node on 8K nodes) on Mira for the past several months. Other than a few problems with the parallel file system, there have been no failed runs. pF3D delivers good compute, message passing, and I/O performance on 8K nodes.

  Tests during the past quarter using an updated version of pF3D demonstrated that pF3De can now run reliably using 1024K MPI processes (32 MPI processes per node on 32K nodes). The message passing performance drops noticeably between 8K and 32K nodes. This was expected based on past experience with default MPI rank to torus mappings on BG/L and BG/P systems. The cure on a BG/Q should be the same – generate a scalable custom mapping which is aware of the message passing characteristics of pF3D. We should start running using these mappings during 1Q14.

  The amount of time spent in the yorick code steering framework increases noticeably when going from 256K to 1024K MPI processes, but it is still at a tolerable level. Two new parallel operations are currently being added to the framework and should allow us to remove the remaining non-scalable portions from the yorick coding. These changes should also be completed during 1Q14.

  We have continued testing and tuning OpenMP directives in pF3D and are able to use 8 hardware threads per process efficiently. Pure MPI performance with

1024K processes is good, so we feel no urgency in putting the mixed MPI+OpenMP code into production use. Testing will continue with the intent to be ready for production use of OpenMP in the summer of 2014.

Project Productivity
Primary

- Publications –
- Presentations –

Secondary

- Journal Covers, Awards, Honors, Popularizations
- Technical Accomplishments – Please list technical accomplishments such as development of reusable code resulting in a new tool, new algorithm design ideas or programming methodologies, formal software releases, etc.
- Other, for example: Simulation results used in outreach initiatives/students graduated or postdocs deployed; Journal Covers; Awards/Honors –

- Highlights – the center creates (concise, short, highly visible) bi-weekly center highlights to submit to DOE—is your project ready, willing, and able to contribute a highlight?

Center Feedback

- Please answer as applicable: Has the support received from the following been beneficial to your project team? Cite examples if possible
  - User Assistance Center– **Tim Williams has done a very good job of helping us**.
  - Scientific Computing Group
  - Visualization and Analysis Team
- Any additional feedback from your project team for the ALCF?

Code Description and Characterization

- Name and provide a description of the primary codes used by your project

  pF3D is a massively parallel laser-plasma interaction code. It solves wave equations for the laser light and one or more backscattered light waves. It also solves for the amplitude of plasma waves (ion-acoustic and/or electron-plasma).

- What are the typical production run sizes that your team plans to undertake in the coming year?

  We will use most of our 2014 allocation for 16k node runs. The runs will fit on 16k nodes and our efficiency would drop somewhat if we used 32k nodes. We plan to run 2-3 single quad runs to begin investigating backscatter in newer NIF experiments like rugby hohlraums. We also plan to do a scaling study up to full system runs on Mira, but the bulk of the CPU time we use will be for 16k production runs.

- What languages and libraries (scientific, I/O, etc.) are used in each code?

  pF3D is written in C. The only scientific library that consumes significant amounts of time is FFTW.

- If possible and useful, please indicate which of the following algorithmic motifs appear in each of your major production codes.

Code Name	**Dense Linear Algebra**	**Sparse Linear Algebra**	**Monte Carlo**	**FFTs**	**Particles**	**Structured Grids**	**Unstructured Grids**	**AMR**
pF3D				X		X		

NIF Laser-Plasma Simulations
Sequoia CCC-2

P.I.: Steven Langer
Username=langer1
Work Phone: (925) 423-1358
langer1@llnl.gov
LLNL/WCI/AX Div

Programmatic justification

The NIF laser must operate at a high intensity to enable high energy density physics (HEDP) experiments. Stimulated Brillouin Scattering (SBS) and Stimulated Raman Scattering (SRS) are the two most important types of laser-plasma instability (LPI) for NIF experiments. The intensities are high enough that a significant amount of laser energy may be backscattered by SRS and SBS. Energy scattered back out of the hohlraum does not heat the hohlraum, lowering the x-ray drive. Different backscatter levels in the inner and outer beam cones will change the drive symmetry. Understanding LPI is thus part of understanding hohlraum drive. Denise Hinkel and other members of the LPI team have published a number of papers based on pF3D simulations of NIF experiments.

LPI is also a facility concern for NIF as it moves to running a wider variety of HEDP experiments. There have been two NIF experiments where the SBS levels were high enough to damage mirrors in the NIF beam lines. NIF currently requires new experimental platforms to slowly increase the intensity to avoid generating high levels of SBS. New experimental platforms have become common as NIF shoots a wider range of HEDP experiments. Slowly “walking the intensity up” is a significant problem for an experimentalist who only has 6 shots for his whole campaign. A good predictive capability for LPI would allow experimentalists to more quickly reach the desired laser power.

The high level goal of our CCC proposal is to continue developing a predictive LPI capability for NIF experiments. Our proposal includes one large simulation to look at the details of how LPI is generated in a well-studied NIC experiment. Smaller simulations will be run to begin investigating LPI in some of the new hohlraum designs currently being investigated at NIF.

Our first goal for Sequoia CCC-2 is to run a pF3D simulation of backscattered light from three interacting NIF quads with two different colors of Stimulated Raman Scattering (SRS) backscattered light. Data from NIF experiments often shows SRS with a broad range of frequencies at the same time. The SRS frequency depends on the density and temperature, so this means that SRS is unstable in multiple locations within the plasma. We will investigate the implications of the broad SRS spectrum by using two SRS frequencies in a single pF3D simulation. The first frequency will be the one with the greatest linear gain and the second will be chosen to match conditions where longer wavelength SRS is generated in the higher density plasma close to the hohlraum wall. This will be the largest pF3D simulation run to date. This simulation will help us understand how different SRS frequencies occurring in multiple quads “compete” for the energy in the laser beam.

One rugby hohlraum experiment produced SBS levels high enough to damage a NIF mirror. Near vacuum hohlraums experiments have produced very low levels of SRS. This is in contrast to the high levels of SRS and low levels of SBS in standard NIC hohlraums. This broader range of LPI behavior at NIF affords us the opportunity to improve our confidence in the predictive capabilities of pF3D by exercising it in a wider variety of conditions. We will start this work during CCC-2 by running a few single quad simulations. Single quad pF3D simulations are much less expensive than three quad simulations and are a good choice for exploring backscatter generation in experiments where the HYDRA simulations are still undergoing significant revisions.

Our CCC-2 simulations will also be used to help develop new pF3D capabilities. Some of the SRS light is absorbed as it propagates back towards the laser entrance hole. Such absorbed light is not detectable by diagnostics measuring the light scattered out of the hohlraum. This will shift where the laser energy is effectively absorbed and change the drive symmetry. We will develop pF3D diagnostics of SRS light absorption to help understand the impact of this effect on symmetry.

NIF ignition experiments typically have slightly different laser wavelengths for the 23 and 30 degree cones. pF3D currently assumes all quads have the same wavelength. We plan to add a “multi-color” capability to pF3D and test it during CCC-2. If the new capability works well, we will use it in CCC-3 simulations.

Readiness Justification

pF3D has demonstrated excellent scaling in the past on 192k Blue Gene/L processors, 128k Blue Gene/P processors, and 64k Cielo processors. Tests run on Sequoia in the spring of 2013 demonstrated good scaling up to 256k cores. The tests also demonstrated that the per-node throughput for pF3D nearly doubles when going from one to two hardware threads per core.

We ran into a problem during CCC-1 on Sequoia where large pf3d simulations would “hang”. In one test case, the simulation would run for days without any problem using 256k processes, but would hang within a few hours when using 512k processes. We were able, with the assistance of Dong Ahn, to use stat to determine what was going wrong in the 512k process simulation.

pF3D runs under the yorick interpreter. Yorick’s interpreted language is used for pF3D input decks, the time step loop, and parallel I/O. Yorick’s MPI-based message passing is used to coordinate all 3 tasks.

A synchronization construct that worked well under version one of yorick's message passing package suffered from a race condition under version two. This problem has now been corrected, and simulations run reliably with 1024k processes on 32k nodes.

The message passing interconnect on Sequoia is fast enough that we were able to use default mappings from MPI rank to torus location during CCC-1. Message passing rates drop significantly when using default mappings with more than 256k processes. We saw similar drops in message rates when using default mappings on BG/L and BG/P systems. Using custom mappings tailored to the message passing characteristics of pF3D allowed us to achieve very good scaling on those systems. We plan to use Rubik to generate custom mappings for the 5D torus on Sequoia for our CCC-2 runs. We expect custom mappings to be very effective on Sequoia.

We developed an MPI+OpenMP version of pF3D in case we ran into problems using a very large number of MPI processes on Sequoia. We recently completed a scaling study from 2k to 32k nodes on Sequoia. Restart and compute times were nearly constant from 2k to 16k nodes. Switching to custom mappings should solve the problems with our message rate for large jobs. We therefore plan to run the pure MPI version of pF3D during CCC-2.

Requested Resources

We plan to run our large 3 quad simulation using 16k nodes. This provides good batch queue turnaround time and good efficiency. We estimate that a three quad simulation with 2 SRS frequencies will take roughly 6 Sequoia days to simulate 40 ps.

We plan to run one simulation as a DAT using 64-96k nodes. We will request the DAT during the second half of CCC-2 after we complete the message passing, I/O, and compute optimizations we are currently working on. This DAT will assess how well we can strong scale pf3d simulations. The DAT will be for no more than 12 hours (<0.5 Sequoia days).

We also plan to run 2 or 3 single quad simulations to scope out the issues involved in simulating recent experiments like rugby hohlraums, near vacuum hohlraums, etc. These simulations should require a total of roughly one Sequoia day. We will also need roughly 0.5 Sequoia days for code development work. ***Our total request is thus for 8 Sequoia days***.

A checkpoint restart dump will be roughly 130 TB in size. We anticipate having up to 8 dumps on disk at one time between the current run and dumps from earlier runs that are retained for comparison purposes. We anticipate accumulating roughly 120 TB per run for visualization and data analysis purposes during the 3 quad run. Our total disk footprint will be roughly 1.2 PB. We would like to save 100-200 TB to archival storage. We do not anticipate any WAN transfers.

We will run pF3D during this CCC. The primary contact for pF3D is Steve Langer, 925-423-1358.

We will use Visit for visualization.

Proposed Project Members

Steve Langer, langer1, langer1@llnl.gov, 925-423-1358
Bruce Langdon, Langdon, langdon1@llnl.gov, 925-422-5444
Denise Hinkel, dhinkel, hinkel1@llnl.gov, 925-423-2626
Dave Strozzi, strozzi2, strozzi2@llnl.gov, 925-424-4720

DOE CSGF Summer Practicum Project Plan
Student - Eileen Martin, Stanford.
Supervisor - Steve Langer, LLNL

DESCRIBE PROJECT

The processing speed of computers is growing faster than their memory bandwidth and capacity. Some applications will be unable to properly exploit these new computers unless something is done to mitigate the memory constraints.

This project will investigate the use of compression to increase the effective size and bandwidth of DRAM memory. Lossless compression ratios are low due to the quasi-random character of the low order bits of floating point words. This project will employ lossy compression schemes. Results of using lossy compression in test problems run with 3 different physics simulation codes were reported by Laney et al. at SuperComputing 2013 (SC13). Compression ratios up to 4X could be used in the test problems without compromising the quality of the key physics results. Compression does not produce a speedup on current systems because software compression takes longer than the time saved due to lower DRAM transfer times. One of the goals of this project is to better understand the requirements for the dedicated on chip compression hardware that would make lossy compression viable for use in production simulations.

This project will extend the earlier work by performing a detailed study of lossy compression in pF3D, a massively parallel code which simulates backscattering of laser light in experiments at the National Ignition Facility. The SC13 paper used simplified pF3D test problems to permit quick turnaround. This project will use more realistic problems and run them for more time steps to determine acceptable compression levels in "production runs".

pF3D solves coupled PDEs on a 3D Cartesian grid. It uses operator splitting, which basically means that each term in the PDEs is implemented as a separate function which performs a partial update on one or more field arrays. The time step loop consists of successive calls to all the operators in the PDEs.

The first step will be to modify pf3d so that lossy compression is integrated in the same way it would be on a system with hardware compression. This will involve changing functions so that they no longer operate on a full spatial domain at once. The domain will instead be split into blocks small enough that all variables fit in cache. A function will load blocks into cache, operate on them, and store updated values back into DRAM. The answers will be validated and the impact of this change on performance will be measured.

Hardware compression/decompression (when it becomes available) will occur as part of the transfer of blocks between DRAM and cache operation. The next step will be to add lossy software compression to the block transfer callsr phase. Test problems will be re-run to examine the impact on performance and accuracy.

If time is available, the loop over blocks will be "hoisted" out of individual functions and wrapped around the calls to the operators in the main time step loop. This will significantly increase cache utilization. Performance will once again be measured.

A model for the performance of pF3D will be created based on the performance measurements indicated above. This will allow us to estimtate the performance of pF3D on future computer systems and help us determine the rate at which compression must occur to be useful.

DESCRIBE MENTORING

I will oversee Eileen's research and provide advice on modifying pF3D. I will explain how to run parallel jobs on LLNL computers and set up pF3D input decks. I will also describe how hardware characteristics impact the performance of pF3D ("parallel performance 101"). In particular, I will discuss how memory bandwidth impacts application performance.

This will provide Eileen with the background she needs to start modifying pF3D. I will provide a lot of assistance when adding "blocking" to the first function, but Eileen will do most of the work on the remaining functions. Some of the performance critical functions in pF3D have been extracted and inserted into a standalone framework which is used to investigate performance optimizations. The first few functions will be tested in this "kernel app" before they are moved into pF3D proper.

The code modifications which add blocking should not change the answers produced by pF3D. Eileen will develop a validation framework which can be used to verify the addition of blocking, but will also work later on when lossy compression is added.

Comparing the performance and memory traffic between the original pF3D and the "blocked pF3D" will provide an introduction to gathering performance data and serve as a starting point for developing a performance model for pF3D.

Work with lossy compression begins after the addition of blocking is complete. Some functions perform zone-by-zone updates. Lossy compression can easily be added to these functions at the points where blocks are gathered from DRAM or scattered back to DRAM.

Other functions use finite difference stencils and thus require data from adjacent blocks ("guard cells"). I will explain the use of guard cells and suggest methods for decompressing guard cells without decompressing entire blocks. Eileen may need to try several approaches to loading guard cell data without requiring large amounts of extra compression and decompression. The first approach to be considered will probably be to compress "chunks" of zones which are much smaller than the blocks operated on by the functions. Guard cells will be populated by decompressing neighboring chunks instead of entire neighboring blocks.

pF3D will be instrumented to track the number of bytes compressed and decompressed by each function, how much time is spent in computation for each block, and how much time is spent on compression and decompression. This will provide the input data for a model of pF3D performance on future computer systems. The inputs will be the compute performance of the future system relative to a current system, the memory bandwidth, and the rate at which blocks can be compressed and decompressed. I will outline how performance models work, but Eileen will develop the model.

Peter Lindstrom of LLNL will provide advice and mentoring on lossy compression algorithms. Peter may implement a new lossy compression for this project.

DESCRIBE EXPECTATIONS

I expect that Eileen will complete the addition of "blocking" to pF3D before the end of the summer. There may or may not be time to add efficient handling of guard cells. Moving the loop over blocks out into the time step loop is definitely a stretch goal. I expect that Eileen will develop a simple performance model for pF3D, but there may not be time for a model which includes the impact of guard cells.

Eileen should learn a number of things over the summer. She will be comfortable running parallel simulations at LLNL and have a good understanding of how memory characteristics impact the performance of simulation codes. She will gain experience gathering hardware performance data from an instrumented code and using it to derive performance metrics. She will have a basic knowledge of how to create a performance model for a code and set the adjustable parameters based on measured performance data. Eileen will also gain familiarity with validating a code based on high-level physics results rather than zone-by-zone differences..