NUMERICAL MODELING OF SPACE PLASMA FLOWS / ASTRONUM-2007
ASP Conference Series, Vol. 385, ©2008
Nikolai V. Pogorelov, Edouard Audit, and Gary P. Zank, eds.

Challenges of Extreme Computing using the FLASH code

A. Dubey[1], R. Fisher[1,2], C. Graziani[1,2], G. C. Jordan, IV[1,2], D. Q. Lamb[1,2], L.B. Reid[1], P. Rich[1], D. Sheeler[4], D. Townsley[2,3], and K. Weide[1]

Abstract. FLASH is a modular, adaptive mesh, parallel simulation framework capable of handling the compressible, reactive flows found in many astrophysical environments. FLASH was recently used in a series of three-dimensional simulations of the gravitationally confined detonation mechanism for Type Ia supernovae on IBM SP-5 platforms at LLNL and NERSC on up to 768 processors. The simulations used multiple physics components, such as hydrodynamics, Newtonian self-gravity, and a sub-grid flame model on an Eulerian adaptive grid. The simulations also used Lagrangian tracer particles to track the nucleosynthetic history. These simulations were a huge challenge not only because of the computational complexity of the problem, but also because of resource constraints. Some components, such as the gravity solver and flame model, had to be algorithmically optimized to increase time efficiency. Other components, such as the tracer particles distribution, grid management, and refinement patterns, were carefully tuned to optimize the memory usage and parallel efficiency.

1. Introduction

The Flash Center, funded by the DOE ASC alliance program, studies detonation of Type Ia supernovae through computer simulations. The FLASH framework used in these simulations is a multiphysics, adaptive mesh code developed and maintained by the Flash Center at the University of Chicago. FLASH is a modern framework which follows good software engineering practices and has a wide user base outside the Flash Center. In this paper, we focus on the application of primary interest to the scientists at the Center, namely the simulations of the deflagration to detonation phase of the Type Ia supernovae. In particular, we discuss the simulations recently carried out to examine the gravitationally confined detonation (GCD) model in 3D (Jordan et al. 2007). The model was originally proposed by Plewa et al. (2004) on the basis of two-dimensional simulations and later scrutinized by other groups (Röpke et al. 2007b). However, extending this model to three dimensions has been a massive challenge because of huge resource requirements and the inherent complexity of the physics.

[1] Center for Astrophysical Thermonuclear Flashes, University of Chicago, Chicago, IL, USA

[2] Department of Astronomy and Astrophysics, University of Chicago, Chicago, IL, USA

[3] Joint Institute for Nuclear Astrophysics, University of Chicago, Chicago, IL, USA

[4] Argonne National Laboratory, Argonne, IL, USA

In this paper, we first describe the GCD mechanism, focusing on the various physical processes. Then we discuss the computational challenges of the adaptive mesh and the numerical algorithms used to treat essential physics, including burning, gravity, and Lagrangian tracer particles. We address how limited resources necessitated computational optimizations, and discuss the many "behind the scenes" activities essential for the success of a large-scale cutting edge calculation.

2. The GCD Model for Type Ia supernovae

Type Ia supernovae (SNe Ia) are thought to be white dwarf stars in binary systems that explode due to a thermonuclear runaway. The companion star transfers matter onto the white dwarf. The increasing mass of the white dwarf raises the temperature and density in its center, eventually igniting at one or more points in the core of the star. Following ignition, the regions of burning grow, and soon become Rayleigh-Taylor unstable. This instability initiates a phase of buoyantly-driven turbulent nuclear burning via an ordinary flame that is referred to as the "deflagration phase." However, while it is the nuclear energy which powers the SN Ia explosion, it is the radioactive decay of ^{56}Ni that makes the SN Ia visible. Pure deflagration models do not generate enough ^{56}Ni to account for the observed light curves and spectra. For this reason, most scientists in the field now think that the deflagration wave must somehow turn into a detonation wave that incinerates the entire star (Khokhlov 1991; Höflich et al. 2002; Gamezo et al. 2004; Plewa et al. 2004; Plewa 2007; Röpke & Niemeyer 2007; Röpke et al. 2007a). The resulting "detonation phase" releases more nuclear energy and creates more ^{56}Ni, producing SNe Ia with kinetic energies and luminosities similar to the observations.

In the "gravitationally confined detonation" (GCD) model proposed by Plewa et al. (2004), off-center ignition at one or more points produces a hot bubble of ash that rises rapidly and breaks through the surface of the star, spreads rapidly across the stellar surface, converges at the opposite point, and initiates a detonation (Plewa et al. 2004; Jordan et al. 2007).

3. Computational Challenges

The physical processes involved in the Type Ia supernovae deflagration include hydrodynamics, self-gravity, and nuclear burning. In addition, Lagrangian tracer particles are used to collect the thermodynamic history of mass elements to calculate nucleosynthesis.

In the GCD computation, a huge challenge is posed by the adaptive mesh process of regridding when the refinement pattern changes as a result of the rapidly moving burning front. Here the communications can be long range, and a large volume of data may need to be moved to other processors. From the time that the expanding bubble nears the surface of the star until reaching detonation conditions, regridding is typically needed every 2–3 timesteps to capture the physics accurately. Also, since there is a direct correlation between the grid resolution and the amount of computation in a single timestep, the refinement process must be carefully managed to avoid unnecessary work.

In the following sections we describe the computational challenges posed by the individual physics solvers and the limitations imposed by finite resources.

3.1. Physics Solvers

The hydrodynamics solver is based on the directionally split piecewise-parabolic method (PPM) (Colella & Woodward 1984). The method is computationally intensive, but has modest communication requirements because it is an explicit scheme. The hydrodynamics module is the oldest physics component of FLASH, and has been optimized for single processing element performance in the past (Calder et al. 2000).

The thermonuclear flame is treated with an Advection-Diffusion-Reaction (ADR) model with a reactive ash in nuclear statistical equilibrium (Khokhlov 1995; Calder et al. 2007; Townsley et al. 2007). This model acts as a flame front that has been artificially broadened to the mesh scale. The reason why the model is necessary is because the thickness of the physical nuclear flame front is in the millimeter to centimeter range, while the spacing of the mesh at its highest resolution is in kilometers. Hence a range of several orders of magnitude in the spatial scales eliminates any possibility of direct numerical simulation of the flame. The flame calculation can influence parallel performance by causing load imbalance. The processors holding the portions of the physical domain with the flame front have significantly more computations than do the rest of the processors.

Newtonian self-gravity is solved using a multipole scheme (ASC Flash Center 2007). Here, the time to calculate the gravitational potential increases at a superlinear rate as the number of included moments is increased. Because the rising bubble is highly directional, higher order moments are necessary, thus making the gravity solver expensive. The Lagrangian tracer particles track the thermodynamic history of Lagrangian mass elements throughout the simulation. The code utilizes a simplified nuclear burning scheme to treat the key reactions for the release of nuclear energy during the simulation. More extensive nuclear networks tracking additional species are handled in the post-processing of the Lagrangian tracer particles. The tracer particles' movement is independent of the Eulerian grid used for all other physics solvers. The range of communications is directly related to spatial locality maintained by the AMR. Additionally, the tracer particles can influence the parallel performance by causing load imbalance if many of them congregate on a few processors.

3.2. Available Resources

In addition to the computational challenges posed by the complexity of the physical processes and the numerical software, the very large simulations also face the challenge of running on highly specialized massively parallel computers. These high performance machines typically have a useful lifecycle of 3–4 years. They are not produced in large quantities, and their architecture is closely tied to cutting-edge research in computer science. Consequently, these machine usually lack the robustness of commodity workstations and clusters, and their system software is relatively less well tested. Often the applications running on these systems stress test the system in a way that vendors do not. This environment presents great challenges to application frameworks like FLASH, which must run on multiple such platforms.

Another machine challenge is that the available resources are rarely enough for the interesting scientific problems. Before running a really large computation, it is sometimes necessary to do several smaller pathfinder runs to determine the trade-offs that would not significantly compromise the quality of the results obtained. We estimated the computational resources needed for the three-dimensional GCD model calculation based on the findings of the earlier two-dimensional simulations. Before the optimizations described in the next section, each time step was taking on average 1.33 seconds per block. The simulation ran on IBM power5 uP at LLNL with 768 processors, and 4 GB of memory per processor. At this resolution, with very conservative refinement, we expected the simulation to take approximately 60 hours. However, a few hours into the calculation it became very clear that our estimates were off by several hours, and a real probability of running out of memory arose. In the next section we describe some of the bottlenecks encountered, and the optimizations and trade-offs that made the simulation possible.

4. Optimizations and Trade-offs

The calculation of gravitational potential using the multipole solver was the biggest initial bottleneck in each timestep. Multipole solvers need additional moments as the asymmetry of the domain increases. In the GCD calculation, the rising hot bubble introduced enough asymmetry such that up to 8th order moments were required to resolve the gravitational potential. However, carefully aligning one arbitrary coordinate axis with the rising bubble produced an axisymmetric problem and therefore eliminated many off-axis moments, thereby reducing the calculation by a factor of five. In a highly fluctuating density field, the accuracy of the gravitational potential calculation with a multipole method depends upon the degree of subsampling, and the solver in FLASH used a high degree of precision by default. A careful analysis of the density field of the GCD model revealed that the subsampling rate could be reduced by a factor of six without significantly changing the value of the calculated potential, saving a factor of 16 in the three-dimensional gravity calculation. These two optimizations together reduced the potential calculation time by a factor of nearly 80.

The next bottleneck was the flame model. An analysis of data gathered from all the processors revealed that most of the calculation was congregated in very few processors. The expensive flame calculation is very localized for most of the simulation, since it follows the flame front. Additionally, since FLASH tries to keep spatially close blocks together, the blocks with the flame front tend to congregate on a few processors thereby leading to load imbalance. We determined that the Nuclear Statistical Equilibrium could be precomputed for the whole problem, allowing us to change the iterative burning calculation into a relatively quick table lookup which also improved load balance. This change saved a factor of four in the flame calculation.

An additional crisis occurred when Lagrangian tracer particles caused memory overflow. At initialization, the tracer particles are randomly distributed throughout the domain in proportion to density, and all processors get roughly similar numbers of particles. However, as the bubble rises, the domain refines aggressively around the bubble, and many low resolution blocks cluster on few processors. The particles travel with their corresponding physical blocks, and hence collect in the same few processors. In the deflagration phase, this phenomenon caused enough imbalance in the distribution of particles that some processors catastrophically ran out of memory. By forcing some blocks to refine in regions without the bubble, we were able to solve this problem.

Even with all the optimizations described above, the savings were not enough to complete the simulation within the allocated resources. During execution, the number of blocks was increasing more rapidly than expected, and would have caused the simulation to run out of memory long before detonation conditions were reached. Near real-time visualization of the density and temperature snapshots indicated that physically unimportant small scale structures were causing unnecessary refinement. However, the same visual inspection of data also revealed small scale structures forming in other regions that were not able to capture the intricate physics and needed to further refine. Thus by aggressively controlling refinement, we channeled the use of memory into scientifically important regions. The most important cost saving came from the pathfinder runs that indicated that with a carefully managed refinement pattern a simulation with 8 kilometer resolution would be sufficient to capture all the important physics in the simulation.

In parallel with the optimizations in the code, the team was doing extensive analysis of the target platform's dynamic memory behavior to estimate the maximum allowable size of the data for the simulation run. Virtual memory on the IBM machines is managed in a large page mode by default, which operates in sticky mode; effectively a page once brought into the memory stays in the memory. If there are many instances of memory allocation and deallocation of varying but large sizes, the heap use becomes very inefficient, causing failure even though there is theoretically enough memory. This situation occurred in the GCD run and we had to use temporally less efficient small page mode, which marginally increased the run time.

5. Conclusions

In this paper we have reported on challenges in running expansive scientific simulations on large scale machines. To successfully complete the simulations, it is not enough to simply have a scientific code and access to a large computer. In addition to the resources being inadequate for the problem of interest, the lack of robustness in system software and specialized nature of system hardware makes these simulations extraordinarily challenging. The Flash Center has been successful in running these simulations because we approach them with a multipronged strategy. The strategy includes simpler pathfinder runs that inform us on the features of the physics formulation that are critical to the simulation. These preliminary runs also indicate the pattern of refinement that would maximize the effective capture of small scale physics that has most impact on the results. In this instance, the findings from the pathfinder runs allowed us to reduce the maximum resolution of the mesh by a factor of two in each dimension, thereby reducing the amount of work by at least a factor of eight.

Monitoring the output and resource usage while the simulation proceeds provides information necessary to adjust the parameters of the run. Specifically, in the GCD model simulation, monitoring the resource usage led to optimization of the gravity, conversion of the flame calculation to table lookup, and modification of refinement criteria to prevent bunching of particles in a few processors. These optimizations together reduced the overall execution time per timestep by a factor of two. Additionally experimenting with the hardware resources allowed us to identify the small page mode of virtual memory in the machine as being more suitable for the simulation.

The successful simulation of three dimensional GCD model has a significant impact on the scientific knowledge of Type Ia supernovae detonation mechanism. Observational data of SN Ia indicate that there must be a deflagration to detonation transition to account for the observed spectrum; only the GCD model spontaneously exhibits this effect. A paper by Jordan IV et al. (2007) in this volume shows snapshots of important phases in the simulation of the GCD model.

Acknowledgments. This work is supported by the U.S. Department of Energy under Grant No. B523820 to the Center for Astrophysical Thermonuclear Flashes at the University of Chicago.

References

ASC Flash Center 2007, FLASH User's Guide, 3rd edn., University of Chicago, http://flash.uchicago.edu/website/codesupport/flash3_ug_beta/

Calder, A.C., Curtis, B.C., Dursi, L.J., Fryxell, B., Henry, G., MacNeice, P., Olson, K., Ricker, P., Rosner, R., Timmes, F.X., Tufo, H.M., Truran, J.W., & Zingale, M. 2000, in Supercomputing, ACM/IEEE 2000 Conference

Calder, A.C., Townsley, D.M., Seitenzahl, I.R., Peng, F., Messer, O.E.B., Vladimirova, N., Brown, E.F., Truran, J.W., & Lamb, D.Q. 2007, ApJ, 656, 313

Colella, P. & Woodward, P.R. 1984, J. Comput. Phys., 54, 174

Gamezo, V.N., Khokhlov, A.M., & Oran, E.S. 2004, Phys.Rev.Lett, 92, 211102

Höflich, P., Gerardy, C.L., Fesen, R.A., & Sakai, S. 2002, ApJ, 568, 791

Jordan, G.I., Fisher, R., Townsley, D., Calder, A., Graziani, C., Asida, S., Lamb, D., & Truran, J. 2007, ArXiv:astro-ph/0703573, submitted to ApJ Letters

Jordan IV, G., Fisher, R., Townsley, D., Calder, A., Graziani, C., Asida, S., Lamb, D., & Truran, J. 2007, in Proceedings of ASTRONUM-2007, this volume.

Khokhlov, A.M. 1991, A&A, 245, L25

—. 1995, ApJ, 449, 695

Plewa, T. 2007, ApJ, 657, 942

Plewa, T., Calder, A.C., & Lamb, D.Q. 2004, ApJ, 612, L37

Röpke, F.K. & Niemeyer, J.C. 2007, A&A, 464, 683

Röpke, F.K., Hillebrandt, W., Schmidt, W., Niemeyer, J.C., Blinnikov, S.I., & Mazzali, P.A. 2007a, ArXiv:astro-ph/0707.1024, 707, accepted for publication in ApJ

Röpke, F., Woosley, S., & Hillebrandt, W. 2007b, ApJ, 660, 1344

Townsley, D.M., Calder, A.C., Asida, S.M., Seitenzahl, I.R., Peng, F., Vladimirova, N., Lamb, D.Q., & Truran, J.W. 2007, ArXiv:astro-ph/0706.1094, 706