

Detection of Silent Data Corruptions in Smoothed Particle Hydrodynamics Simulations

Aurélien Cavelan
University of Basel, Switzerland
aurelien.cavelan@unibas.ch

Florina M. Ciorba
University of Basel, Switzerland
florina.ciorba@unibas.ch

Rubén M. Cabezón
University of Basel, Switzerland
ruben.cabezón@unibas.ch

Abstract—Soft errors, such as silent data corruptions (SDCs) hinder the correctness of large-scale scientific applications. Ghost replication (GR) is proposed herein as the first SDCs detector relying on the fast error propagation inherent to applications that employ the smoothed particle hydrodynamics (SPH) method. GR follows a two-steps selective replication scheme. First, an algorithm selects which particles to replicate (or ghost), on a different process. Then, a different algorithm detects SDCs by comparing the data of the selected particles with the data of their ghost. Error propagation is studied for the first time in SPH through a set of performance, error-injection and error-detection experiments. The overhead and scalability of the proposed approach are assessed through a set of strong-scaling experiments conducted on a large HPC system under error-free conditions, using upwards of 3,000 cores. The results show that GR achieves a recall and precision similar to that of full replication methods, at only a fraction of the cost, with detection rates of 91–99.9%, no false-positives, and an overhead of 1–10%.

Index Terms—Soft errors, silent data corruptions (SDC), propagation, replication, detection, smooth particle hydrodynamics (SPH), Exascale

The frequency of errors and failures in large-scale systems increases proportionally to the number of components, and reliability has been identified as a major challenge for Exascale computing [14]. In particular, silent errors, or *silent data corruptions* (SDCs), represent a major threat to the correctness of the results of such calculations. SDCs typically manifest when one or more bits are flipped in the memory of the system. There are several causes for these errors, such as packaging pollution or cosmic rays, among others [1], [11], [16]–[18].

Providing error detectors for scientific applications towards extreme-scale computing has been identified as one of the hardest challenges today [3], [4]. This work concentrates on detecting DRAM errors. A novel SDCs detector is proposed in this work, called *ghost replication* (GR), that is solely based on the *selective replication* of particles in Smooth Particle Hydrodynamics simulations.

A. Smoothed Particle Hydrodynamics

Smoothed particle hydrodynamics (SPH) is a fully Lagrangian meshless method to perform hydrodynamics simulations. Initially devised in the late '70s [6], [8], [9], this technique underwent sustained development [2], [5], [7], [12], [13], [15] and is, nowadays, common in many fields, including computational fluid dynamics, plasma physics, solid mechanics, and astrophysics. Its inherent good conservation

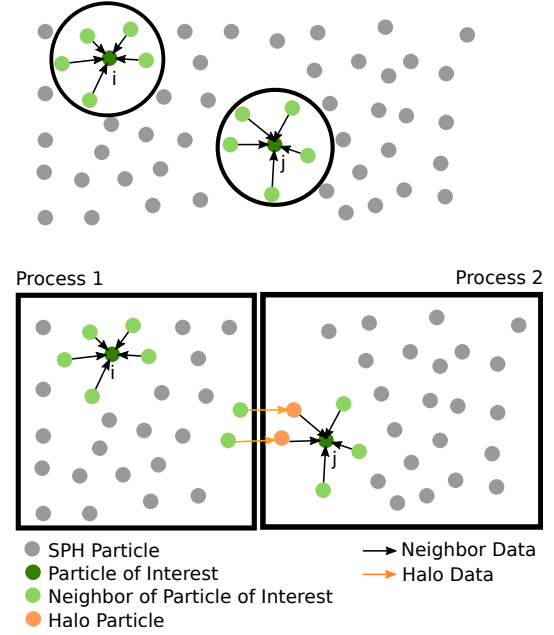


Figure 1: Example of an SPH particle domain partitioned into two sub-domains that are assigned to be computed by two different processes. Two particles are highlighted, i and j , for which the sample set of neighbors are also shown in the circles. Particle j has two neighbors on process 1, and needs a copy of these two neighbors (halo particles) in its own memory.

properties, and its adaptability to distorted geometries, make SPH a common choice to simulate highly-dynamic three-dimensional scenarios. The SPH technique discretizes a fluid in a series of interpolation points (SPH particles) whose distribution follows the mass density of the fluid, and their evolution relies then on a weighted interpolation over close neighboring particles.

B. Ghost Replication

To detect SDCs in SPH simulations, this work introduces *ghost particles*. A ghost particle is an SPH particle, that has been selected on a given process, to be replicated on a target process, different from the original one. Replication is achieved by (1) copying the data associated with the selected particle, i.e., initial values and a copy of the selected particle's neighbors, that are not already on the target process, and (2)

by simply replicating all computations associated with the original particle on the target process.

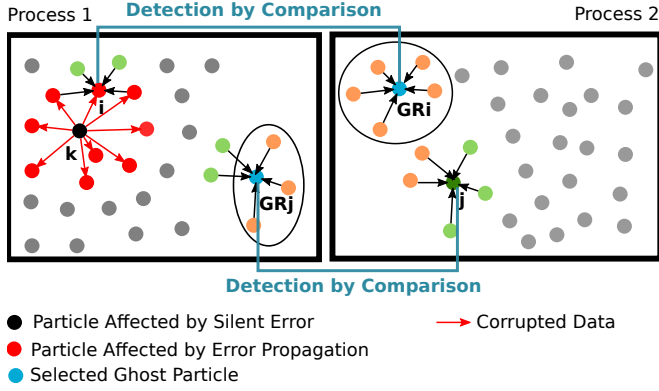


Figure 2: An error strikes particle k on process 1, and rapidly propagates to neighboring particles, including particle i . The ghost particle of i , GR_i , being on a different process, is not affected by the error on particle k . The error is detected by comparing the data of particle i against the data of its ghost, particle GR_i , before halo particles are exchanged.

In parallel SPH implementations [10], a process is typically assigned a list of particles to process. Ghost particles are simply added to the list of particles assigned to the ghost target process, to be computed together with the other non-replicated particles. The main difference is that ghost particles are never communicated to other processes (except for detection), and they never interact with other particles on the ghost target process. Figure 2 shows two such ghost particles, GR_i and GR_j , where GR_i is the ghost of particle i (from process 1 in Figure 1) and GR_j is the ghost of particle j (from process 2 in Figure 1), each replicated on the other process.

Given that an original particle and its ghost represent the same particle, their data are expected to remain identical at any point during the simulation. Moreover, the data of the original particle can be compared against the data of its ghost with *bit-level resolution*. If an error propagates to either the original particle or its ghost, even by causing a single bit to be different, it will be detected via a simple comparison.

Ghost particles need to be selected such that all particles in a process have *at least* one neighbor that has a corresponding ghost particle on another process. Note that the overhead of GR on a process is proportional to the number of ghost particles that need to be computed in addition to the original ones.

C. Experiments

Experiments were conducted by incorporating GR in a production SPH code and by running two test simulations commonly used in astrophysics. A first set of error-injection and error-detection experiments was performed to assess the detection capabilities of GR. A subsequent set of performance evaluation experiments was performed, without injecting errors, to assess the overhead of GR for the two test simulations

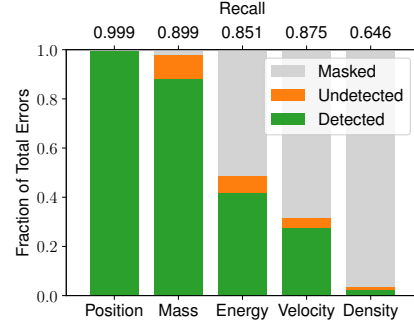
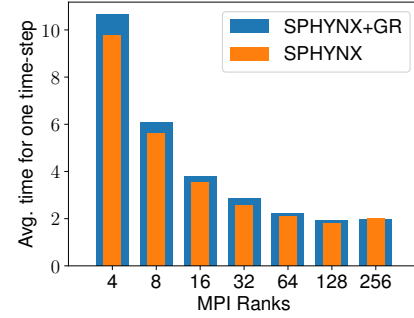
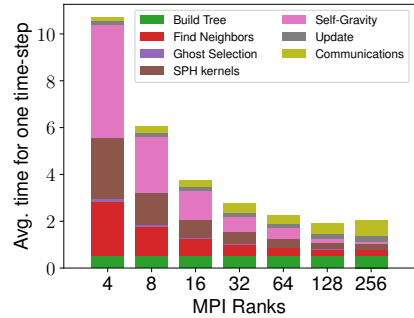


Figure 3: Fraction of errors detected, undetected and masked for five datasets.



(a) Execution time per time-step (EC)



(b) Execution break down for SPHYNX+GR by computational workflow step within a single time-step (EC)

Figure 4: Average time needed to complete one time-step for the EC test with the original code (SPHYNX) and the modified version (SPHYNX+GR) (a), and break down by workflow step for SPHYNX+GR only (b).

on the Piz Daint supercomputer¹. These results show that GR is particularly suited for detecting SDCs in SPH simulations, with GR being able to detect 91 – 99.9% of injected errors, with no false-positives and a typical execution overhead of 1 – 10%.

ACKNOWLEDGEMENT

This work has been supported by the Swiss Platform for Advanced Scientific Computing (PASC) project SPH-EXA and by a grant from the Swiss National Supercomputing Centre (CSCS) under project ID c16.

¹<https://www.cscs.ch/computers/piz-daint/>

REFERENCES

- [1] L. Bautista-Gomez, F. Zulkararov, O. Unsal, and S. McIntosh-Smith. Unprotected computing: A large-scale study of dram raw error rate on a supercomputer. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '16*, pages 55:1–55:11, Piscataway, NJ, USA, 2016. IEEE Press.
- [2] R. M. Cabezón, D. García-Senz, and A. Relaño. A one-parameter family of interpolating kernels for smoothed particle hydrodynamics studies. *Journal of Computational Physics*, 227:8523–8540, Oct. 2008.
- [3] F. Cappello, A. Geist, W. Gropp, S. Kale, B. Kramer, and M. Snir. Toward Exascale Resilience. *Int. J. High Performance Computing Applications*, 23(4):374–388, 2009.
- [4] F. Cappello, A. Geist, W. Gropp, S. Kale, B. Kramer, and M. Snir. Toward Exascale Resilience: 2014 update. *Supercomputing frontiers and innovations*, 1(1), 2014.
- [5] D. García-Senz, R. M. Cabezón, and J. A. Escartín. Improving smoothed particle hydrodynamics with an integral approach to calculating gradients. *Astronomy & Astrophysics*, 538:A9, Feb. 2012.
- [6] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics - Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, Nov. 1977.
- [7] D. Le Touzé, G. Oger, and B. Alessandrini. Smoothed Particle Hydrodynamics simulation of fast ship flows. In *Proc. 27th Symposium on Naval Hydrodynamics*, 2008.
- [8] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:1013–1024, Dec. 1977.
- [9] J. J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30(1):543–574, 1992.
- [10] G. Oger, D. L. Touzé, D. Guibert, M. de Leffe, J. Biddiscombe, J. Soumagne, and J.-G. Piccinalli. On distributed memory mpi-based parallelization of sph codes in massive hpc context. *Computer Physics Communications*, 200:1 – 14, 2016.
- [11] T. O’Gorman. The effect of cosmic rays on the soft error rate of a DRAM at ground level. *IEEE Trans. Electron Devices*, 41(4):553–557, 1994.
- [12] D. J. Price. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231:759–794, Feb. 2012.
- [13] S. Rosswog. Boosting the accuracy of SPH techniques: Newtonian and special-relativistic tests. *Monthly Notices of the Royal Astronomical Society*, 448:3628–3664, Apr. 2015.
- [14] M. Snir and al. Addressing failures in exascale computing. *Int. J. High Perform. Comput. Appl.*, 28(2):129–173, 2014.
- [15] V. Springel. Smoothed Particle Hydrodynamics in Astrophysics. *Annual Reviews of Astronomy and Astrophysics*, 48:391–430, Sept. 2010.
- [16] J. Ziegler, H. Muhlfield, C. Montrose, H. Curtis, T. O’Gorman, and J. Ross. Accelerated testing for cosmic soft-error rate. *IBM J. Res. Dev.*, 40(1):51–72, 1996.
- [17] J. Ziegler, M. Nelson, J. Shell, R. Peterson, C. Gelderloos, H. Muhlfield, and C. Montrose. Cosmic ray soft error rates of 16-Mb DRAM memory chips. *IEEE Journal of Solid-State Circuits*, 33(2):246–252, 1998.
- [18] J. F. Ziegler, H. W. Curtis, H. P. Muhlfield, C. J. Montrose, and B. Chin. IBM experiments in soft fails in computer electronics. *IBM J. Res. Dev.*, 40(1):3–18, 1996.