SpotSDC: An Information Visualization System To Analyze Silent Data Corruption

Zhimin Li
Scientific Computing and Imaging Institute
zhimin@sci.utah.edu

Harshitha Menon
Lawrence Livermore National Laboratory
harshitha@llnl.gov

Yarden Livnat
Scientific Computing and Imaging Institute
yarden@sci.utah.edu

Kathryn Mohror
Lawrence Livermore National Laboratory
mohror1@llnl.gov

Valerio Pascucci
Scientific Computing and Imaging Institute
pascucci@sci.utah.edu

ABSTRACT

Aggressive technology scaling trends are expected to make the hardware of HPC systems more susceptible to transient faults. Transient faults in hardware may be masked without affecting the program output, cause a program to crash, or lead to silent data corruptions (SDC), where the fault introduces an error in the application that is not detected. Due to the insidious nature of SDCs, understanding a program's SDC resilience characteristics is critical for developing and evaluating software resiliency techniques in order to ensure correct output results from HPC applications. The default method for studying how SDC affects an application is with a fault injection campaign. While fault injection studies can give an overall resiliency profile for an application, without a good visualization tool it is extremely difficult to summarize and highlight critical information obtained. In this work, we design SpotSDC, a visualization system to analyze a program's resilience characteristics to SDC. SpotSDC provides an overview of the SDC impact on an application by highlighting regions of code that are most susceptible to SDC and will have a high impact on the program's output. SpotSDC also enables users to visualize the propagation of error through an application execution. The results from SpotSDC will enable developers to design selective algorithmic error detection and mitigation techniques to protect their applications from SDC.

KEYWORDS

Fault Tolerance, Information Visualization, Transient Error

1 INTRODUCTION

Aggressive technology scaling trends are expected to make the hardware of HPC systems more susceptible to transient faults. Transient faults are caused by cosmic radiation or electrical noise, and can manifest in applications in several ways. It can cause a program to crash, or be masked without affecting the program output. The most insidious outcome is silent data corruption (SDC), where the fault introduces an error in the application that is not detected and can potentially alter HPC simulation results. Due to the insidious nature of SDC, researchers have tried to understand how SDC affects a particular applications. Understanding a program's SDC resilience characteristics is critical for developing and evaluating software resiliency techniques.

The default method for studying how transient faults affect an application is using a fault injection campaign[2]. In a fault injection campaign, a tool injects a fault in an application at a specific instance of time and records the result: crash, mask, or SDC. Without the help of a visualization tool, it is very challenging to summarize the critical information from the fault injection study; and current tools typically provide users only high level statistics such as overall resilience merit for a small number of metrics.

In this work, we design SpotSDC, a visualization system to analyze program's resilience characteristics to SDC. SpotSDC highlights regions of code that are most susceptible to SDC and have a high impact on the program's output and enables researchers to compare the impact of faults on different program regions as well as bit locations in variables. In addition to providing an overall SDC characteristics of the application, SpotSDC also enables users to visualize the propagation of error through an application execution. The results from SpotSDC will enable developers to design selective algorithmic error detection and mitigation techniques to protect their codes from producing erroneous output.

2 SPOTSDC SYSTEM

SpotSDC has several visualization views to help researchers assess the impact of transient faults. The primary views are 1) Overview, 2) SDC analysis of source code regions, and 3) Error propagation view. It loosely follows Schneiderman mantra: "Overview first, zoom, filter, details on demand", where SpotSDC first provides an overall impact of transient fault on the program output, and the user can conduct further detailed analysis by zooming into a specifics about source code component or analyze the error propagation for a specific case.

In this section, we provide details about the individual components in SpotSDC and highlight the main insights obtained from our visualization system. An exhaustive fault injection campaign was performed on a Conjugate Gradient (CG) benchmark, where we used a single bit-flip fault model. The transient faults introduced by our fault injection have three outcomes, namely crash (also denoted as DUE), masked, and SDC. SpotSDC uses the data provided by the fault injection campaign to construct different visualization views to help researchers identify critical resilience properties[1] and behavior of CG in the presence of transient faults.

2.1 Overview Visualization

The *overview* visualization component provides a high level view of the entire application's resilience characteristics. Figure A in the poster shows the *overview* component for CG. This view has a hierarchical structure, where the users can select different contexts for the hierarchy. The *overview* visualization panel provides the following for the selected program components.

- Frequency of transient faults.
- SDC, DUE and Masked rates.
- Distribution of SDC with respect to bit location.

In Figure 1 of this abstract, we show an *overview* view that gives the overall impact of transient faults on CG. The stacked bar chart on the right shows the total number of fault injections conducted on CG, and the overall SDC rate for this application. The chart on the left is called the *BitStackChart*, which indicates the distribution of transient fault outcomes with respect to the bit location. Here we use the IEEE Standard 754 floating point representation, and the x-axis shows the bit location for double precision floating-point format having 52 bits of mantissa, 11 bits of exponent and 1 sign bit. At each bit location in the *BitStackChart*, a stacked bar chart shows the distribution of SDC, DUE and Masked due to a transient fault (bit flip) at that bit location. Specifically, figure 1 shows that 45% of the single bit flips at bit location 62 result in SDC (shown in orange), 10% result in DUE (blue), and 45% are masked by the program (green). The *overview* panel of SpotSDC provides the following main insight: The majority of the SDC is caused by bit flips in the exponent and higher mantissa bits.

2.2 Source Code View

Figure 2 is a subsection of the source code view showing the SDC analysis of line 57 in function `dot` of CG. The view highlights critical regions of the code, where an error can significantly corrupt the outcome of the application, and also provides information with respect to loop iterations by color coding the iteration number next to the source line. The gray color indicates no execution happen at that iteration. The main insight for Figure 2 is that faults in later iterations have less impact on program results in CG.

2.3 Propagation View

In Figure 3 of this abstract we show how a transient error propagates through CG. The top panel (1) shows where the error was introduced (depicted by the red triangle), and the relative error in variables as the execution progresses. The middle panel (2) zooms into the period of execution which is under the zoom lens to show more detail. Finally, the bottom panel (3) shows how the error propagates from one variable to another within that time interval. As the visualization shows, the error starts at line 33 in variable `r`, and propagates to line 25 in the array variable `Ap`.

3 CONCLUSION

We designed SpotSDC, a domain-specific visualization system aimed towards helping users understand resilience properties of HPC applications under the influence of transient faults. SpotSDC has different views to provide summary as well as fine-grain details using data from fault injection campaigns. It highlights critical regions of the code, and provide valuable insights to the users regarding an application's SDC characteristics. It also allows users to understand how the error propagates through an application. The results from SpotSDC will provide insights to developers and thereby enable them to design and evaluate fault resilience techniques.

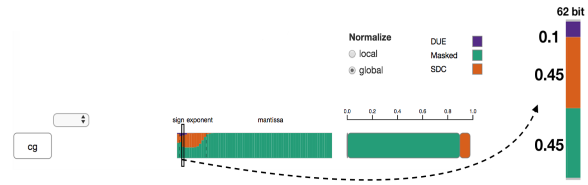

Figure 1: SpotSDC shows the overall transient fault impact on a conjugate gradient benchmark (CG).

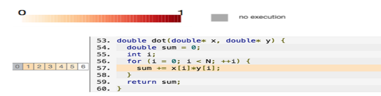

Figure 2: Source code shows critical line's SDC rate and different iterations' SDC ratio

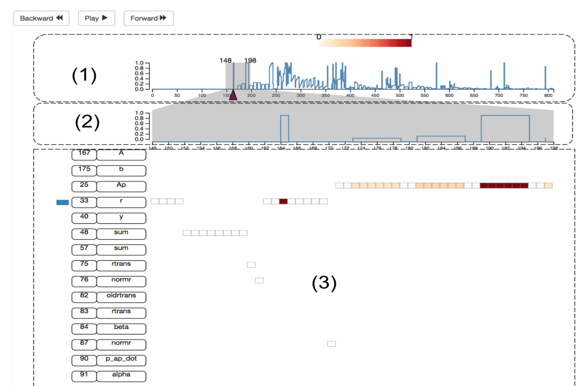

Figure 3: Error propagation view shows a transient fault propagate through a program execution.

ACKNOWLEDGMENTS

The authors would like to thank Shusen Liu, Dan Maljovec for their suggestion during the program development. This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number(s) DE- SC0014098, program manager Lucy Nowell. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-POST-755344).

REFERENCES

[1] Harshitha Menon and Kathryn Mohror. 2018. D is CV ar: discovering critical variables using algorithmic differentiation for transient faults. In *Proceedings of the 23rd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*. ACM, 195–206.

[2] Haissam Ziade, Rafic A Ayoubi, Raoul Velazco, and others. A survey on fault injection techniques. (????).