Failure prediction for large scale system: What are the missing pieces?

Ana Gainaru

University of Illinois at Urbana-Champaign

The last few years have been a fertile ground for the development of many scientific and data-intensive applications in all fields of science and industry. These applications provide an indispensable mean of understanding and solving complex problems through simulation and data analysis. As large-scale systems evolve towards post-petascale computing to accommodate applications' increasing demands for computational capabilities, many new challenges need to be faced, among which fault tolerance is a crucial one [1, 2]. With failure rates predicted in the order of tens of minutes [6] for the exascale era and applications running for extended periods of time over a large number of nodes, an assumption about complete reliability is highly unrealistic. Because processes from scientific applications are, in general, highly coupled, even more pressure is put on the fault tolerance protocol since a failure to one of the processes could eventually lead to the crash of the entire application.

By far the most popular fault tolerance technique to deal with application failures is the Checkpoint-Restart strategy. Unfortunately, classical checkpointing, as used today, will be prohibitively expensive for exascale systems [5]. A complement to this classical approach is failure avoidance, by which the occurrence of a fault is predicted and proactive measures are taken. In general, failure prediction is based on the observation that there is a fault-errors-failure propagation graph [3]. The fault generates a chain of errors that could be observable at the system level that end with the failure which is observed at the application level and usually is represented by an application interruption or performance degradation.

Over the years, different methods have been developed that deal with failure prediction in the HPC community [3], methods that have been used extensively on different HPC systems and presented a variety of results. In our previous work we introduced the concept of signal analysis in the context of event analysis, which allowed us to characterize the behaviour of different events and to analyze them separately depending on their individual behaviour [4]. All results show that failure prediction is a theoretical viable solution for future fault tolerance techniques. One of such solution is called proactive checkpointing by which an application saves its state only when a failure is predicted.

Proactive checkpointing alone cannot systematically avoid re-executing the application from scratch if failures are not perfectly predicted. Unfortunately, the complexity of an HPC system makes the correlations between its events probabilistic and so it makes perfect failure prediction almost impossible. Moreover, recent research shows that a large set of failures do not generate any precursor events in the system [1]. Therefore, it becomes increasingly important to find ways to combine failure prediction with existing fault tolerance techniques. One solution is a combination of failure prediction in the form of proactive checkpointing with the classical periodic checkpointing.

All experiments for failure prediction methods have been the result of the analysis of past generation HPC machines in simulated online environments. The scale of today's systems has increase by two orders of magnitude, with predictions for exascale showing an even higher increase [2]. Moreover, simulated online predictions assume tuning the parameters of all prediction modules in the

offline phase in order to achieve the best possible results in the online phase. While this methods show prediction results that could theoretically be achieved in real scenarios, they do not reflect the reality of running in realtime and predicting failures using best local parameters.

In this poster, we introduce a methodology for truly online predictions and we show that by using this model, prediction is possible and gives good results on small systems. Moreover, we investigate the differences between current large-scale systems and previous smaller systems and how this difference affects prediction. For this purpose we study the feasibility of online failure prediction methods on the Blue Waters system. With a sustained performance of 1 Petaflop on a range of real-world science and engineering applications, the Blue Waters supercomputer is representative for today's large scale systems and provides new insights into the performance and results of current fault predictors.

Also, we implemented a hybrid checkpoint strategy by integrating our previous work with a multilevel checkpointing protocol, in order to study the overheads of these fault tolerance techniques on applications' execution. We investigate the difference in the overhead encountered by this approach on a small and large system and how this is translated in the decrease of the waste given by the protocol. We show to what extent current failure prediction method are possible on Blue Waters and what are the challenges in achieving an effective fault prevention mechanism.

References

[1] Inter-Agency Workshop on HPC Resilience at Extreme Scale. `http://institute.lanl.gov/resilience/docs/Inter-AgencyResilienceReport.pdf`, 2012. [Accessed on July 2013].

[2] U.S. Department of Energy Fault Management Workshop. `http://shadow.dyndns.info/publications/geist12department.pdf`, 2012. [Accessed on July 2013].

[3] F. Salfner et al. A survey of online failure prediction methods. *Computing Surveys*, 42:1–42, 2010.

[4] A. Gainaru, F. Cappello, M. Snir, and W. Kramer. Fault prediction under the microscope: A closer look into hpc systems. In *Proceedings of the International Supercomputing Conference*, 2012.

[5] W. M. Jones, J. T. Daly, and N. DeBardeleben. Application monitoring and checkpointing in HPC: looking towards exascale systems. *Proceedings of the 50th Annual Southeast Regional Conference*, pages 262–267, 2012.

[6] M. Snir, W. Gropp, and P. Kogge. Exascale Research: Preparing for the Post-Moore Era. *Computer Science Whitepapers*.