Thesis Overview:

Fault Tolerance in Multicore Clusters. Techniques to Balance Performance and Dependability

Hugo Meyer

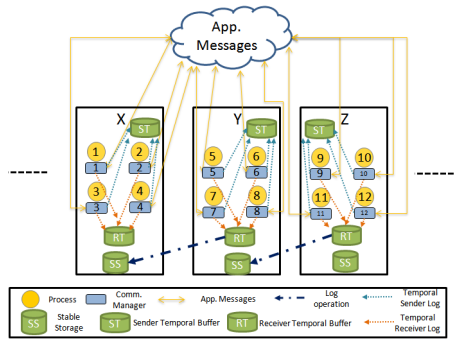
Computer Architecture and Operating Systems Department (CAOS)
Universitat Autònoma de Barcelona, Barcelona, Spain
Advisor: Ph.D. Dolores Rexachs
hugo.meyer@caos.uab.es, dolores.rexachs@uab.es
Supported by the MICINN Spain, under contract TIN2007-64974 and TIN2011-24384.

In High Performance Computing (HPC) the demand for more performance is satisfied by increasing the number of components. With the growing scale of HPC applications has came an increase in the number of interruptions as a consequence of hardware failures. The remarkable decrease of Mean Times Between Failures (MTBF) in current systems encourages the research of suitable Fault Tolerance (FT) solutions which makes it possible to guarantee the successful completion of parallel applications. By executing applications on HPC systems, we aim to improve the performance despite the failures that may affect systems. Our research focuses on analyzing and reducing the impact of scalable FT techniques based on rollback-recovery (e.g. uncoordinated checkpoint). As message logging is normally the main source of overhead when using uncoordinated checkpoint approaches, our research focuses on analyzing and reducing the impact of current pessimistic receiver-based message logging techniques. Taking into account the advent of multicore machines, our main contributions aim to make an efficient use of the parallel environment considering the interaction between applications processes and fault tolerance tasks. The main contributions of this research are described below.

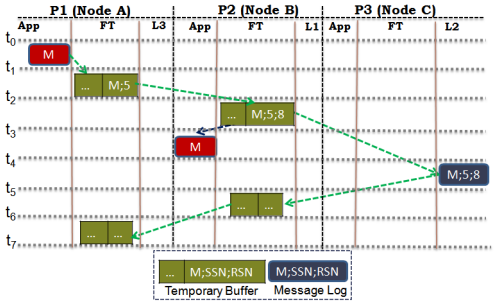
Hybrid Message Pessimistic Logging protocol -HM$_{PL}$ The HM_{PL} [1] focuses on combining the fast recovery feature of pessimistic receiver-based message logging with the low protection overhead introduced by pessimistic sender-based message logging. Fig. 1a shows the design of the HM_{PL} and Fig. 1b shows its operation mechanisms. Senders (P1) save messages in a temporary buffer before sending them. These messages may be used in certain failure scenarios to allow processes to fully recover. When processes receive messages (P2), they store these messages in a temporary buffer and continue with the normal execution without waiting for the messages to be fully saved in a stable storage at another location (P3). The main benefit of this approach is that it reduces the impact in the critical path of applications by removing the blocking behavior of pessimistic approaches guaranteeing that non-failed processes will not rollback.

Methodology to Determine Suitable FT task configuration The main goal of this methodology [2] (Fig. 1c) is to determine the configuration of message logging (loggers) which generates the least disturbance to the parallel application. We first extract the behavior of the application by using a kernel or a signature of it. Then, we characterize how the application interacts with different message logging process mapping. We can leave the distribution of loggers to the Operating System (OS), make an Homogeneous Distribution or to save resources for the loggers (Own FT Resources). Finally, we select the configuration that introduces less overhead and execute the parallel application.

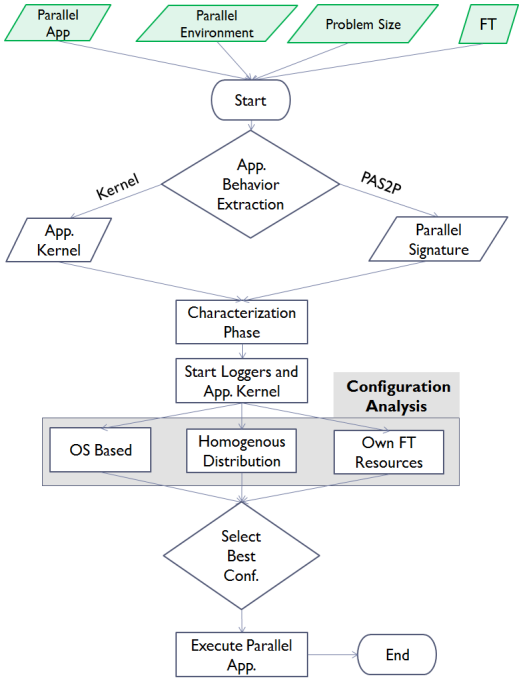
Methodology to increase Performability of Single Process Multiple Data (SPMD) applications Taking into account that many scientific applications are written using the SPMD paradigm, we



(a) Hybrid Message Pessimistic Logging Design. Temporary buffers and stable storage are distributed among the system.



(b) Hybrid Message Pessimistic Logging Operation Mechanism.



(c) Methodology to Determine Suitable FT tasks Configuration.

Figure 1: Techniques to balance Performance and Dependability.

have proposed a novel method which allows us to obtain the maximum speedup under a defined efficiency threshold taking into account the impact of a fault tolerance strategy when executing on multicore clusters [3], [4]. We determine analytically the number of computational cores and the ideal number of tiles (Supertile), which permit us to obtain a suitable balance between speedup, efficiency and dependability. Our method manages the overheads of message logging by overlapping them with computation of the internal tiles.

References

[1] H. Meyer, D. Rexachs, and E. Luque, "Hybrid Message Logging. Combining advantages of Sender-based and Receiver-based Approaches," *Procedia Computer Science*, vol. 29, no. 0, pp. 2380 – 2390, 2014, 2014 International Conference on Computational Science. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050914003998

[2] ——, "Managing Receiver-Based Message Logging Overheads in Parallel Applications," *XIX Congreso Argentino de Ciencias de la Computación. Mar del Plata, Argentina*, pp. 204–213, 2013.

[3] H. Meyer, R. Muresano, D. Rexachs, and E. Luque, "Tuning SPMD Applications in order to Increase Performability," *The 11th IEEE International Symposium on Parallel and Distributed Processing with Applications, Melbourne, Australia*, pp. 1170–1178, 2013.

[4] ——, "A Framework to write Performability-Aware SPMD Applications," *The 2013 International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, USA*, pp. 350–356, 2013.