Preface

As modern society relies on the fault-free operation of complex computing systems, system fault-tolerance has become a matter of course. Common agreement exists that large software systems always contain faults and precautions must be taken to avoid system failure. Failure of hardware components often is caused by external factors that can be neither predicted, avoided, nor corrected. Therefore, mechanisms are needed that guarantee correct service in the presence of failure of system components, be it software or hardware elements. Commonly used are redundancy patterns. These can be either redundancy in space or redundancy in time. Redundancy in space means the inclusion of additional hardware or software modules in the system that can replace a failed component. Different types of redundancy in space exist, such as cold, warm and hot stand-by. Redundancy in time includes methods such as restart, rejuvenation and checkpointing, where execution of tasks is repeated, or the task environment is restarted, either following a system failure or, preventively, during normal operation.

This book is concerned with methods of redundancy in time that need to be issued at the right moment. In particular we address the question of choosing the right time for the different fault-tolerance mechanisms. This includes a brief introduction to the regarded methods, i.e., restart, rejuvenation and checkpointing and aspects of their practical implementation in real-world systems. But the focus of the book is on selecting the right time, or timeout, for restart, rejuvenation and checkpointing. In general, this is the timeout selection problem.

Selecting the right timeout is a problem that is subject to a number of uncertainties. It is, in general, unknown when the system will fail. Furthermore, upcoming busy and idle periods of the system are not known, as is future user interaction with the system. Due to the many uncertainties, the timeout selection problem lends itself for a stochastic treatment. Consequently, many stochastic models addressing the timeout selection problem in restart, rejuvenation and checkpointing have been proposed. This book gives an overview of existing stochastic models of restart, rejuvenation and checkpointing.

The second part of this book treats a stochastic model of restart in various facets. Restart operates on program, or application level. If a task does not complete within a given time it is suspected to have failed and is, consequently, aborted and restarted. The timeout after which to abort and restart the task must be carefully chosen
because if it is too short the task might be aborted just before completion while if it is chosen too long one must wait unnecessarily.

The second method is software rejuvenation. Software rejuvenation restarts the operating environment of a task in order to prevent failures. Rejuvenation is a proactive fault-tolerance treatment. Hence it is issued before the system fails. This is particularly challenging, as it implies assumptions as to when the system would fail if no measures were taken. Ideally, the rejuvenation interval would always end just before the system fails. A conservative choice of the rejuvenation interval will select short intervals. But rejuvenation comes with a cost of saving the operating environment and all processes, restarting the system and reinitialising the operating environment and all processes. If rejuvenation is performed too often the rejuvenation cost accumulates unnecessarily, while if the system is rejuvenated at too long intervals it will often fail, rendering rejuvenation ineffective.

The third method, checkpointing, is the most complex mechanism of the three as it has a preventive component, saving a checkpoint, and a reactive component, rollback recovery. Checkpointing systems save the system state in regular or irregular time intervals. Upon failure the system recovers by rolling back to the most recent checkpoint. The work performed since the most recent checkpoint is lost with a failure. If checkpoints are taken too frequently the interrupt and save operation incurs too high a cost, while if checkpoint intervals are too long much work is lost upon system failure.

For all three methods similar trade-offs exist. The fault-tolerance mechanisms come at a cost that must be traded against the cost of a potential system failure. If the timeouts are well chosen the fault-tolerance mechanism will avoid a failure and be worth-while. The trade-offs can be evaluated and optimised using stochastic models. The focus of this book is to collect, summarise and compare those stochastic models. This can be seen as a first step towards understanding and solving the generic timeout selection problem.

This book is based on the author’s habilitation thesis at Humboldt-University in 2008. The habilitation thesis, and hence this book, would not be as it is without the careful and thorough reading from the first to the last page of Mirek Malek. I would like to thank him for his efforts. His many valuable comments helped to improve this text tremendously. I am thankful to Boudewijn Haverkort and Miklos Telek as they agreed on reviewing and commenting on the thesis. Miklos Telek even came to Berlin for the habilitation lecture, even though it was on yet another topic.

I thank the members of the habilitation committee for their invested time and consideration. It was a pleasure to work with the gifted students at Humboldt-University over the five years I spent there until completion of my habilitation thesis. I particularly thank Willi Engel for being a member of the habilitation committee and Philipp Reinecke for the many little shell scripts he quickly set up and for his dedicated work in the restart project. I thank Steffen Tscharpke for technical support and Christine Henze for help in all administrative matters. I am grateful to my former colleagues at Humboldt-University for many fervid discussions that broadened my view.

Aad van Moorsel has been a colleague on the restart project for many years. He is the perfect person to write a foreword and I am happy he agreed to do so.
I want to take the opportunity to thank Jochen Schiller and his group for giving me such a warm welcome and making me feel at home immediately at Freie Universität. This made me anticipate future work related to restart and also broadened my horizon.

I would like to thank Springer Verlag and Ralf Gerstner in particular for agreeing to publish this book.

Finally, the most crucial support throughout the past years came from my family. I thank Tobias Zepter and our children Louise, Margarete, Elisabeth, Samuel and Jakob just for being there and being my delight every day.

Berlin, Germany, 
October 2009
Stochastic Models for Fault Tolerance
Restart, Rejuvenation and Checkpointing
Wolter, K.
2010, XVI, 269 p., Hardcover
ISBN: 978-3-642-11256-0