

Assignment 4

(10% of Final Grade)

1 Fault Tolerance Models

Consider the following fault tolerance models: Recovery Blocks, Retry Blocks, Flat Transactions, Nested Transactions, Atomic Actions, Coordinated Atomic Actions, Open Multithreaded Transactions, N-Version Programming, N-Copy Programming.

State for each model if it supports:

- Forward error recovery
- Backward error recovery
- Recursion
- Structured exception handling
- Concurrency (be specific, e.g. if it is intra-model concurrency and/or inter-model concurrency)

Answer for each model if it can tolerate:

- Crash failures of the underlying hardware
- Transient faults in the environment
- Design faults in the algorithms / code

2 Terminology

Classify each of the following practices as:

- Fault avoidance
- Fault tolerance
- Fault forecasting
- Fault removal
- Robustness (sequential systems slide, page 3 and 4)

In case the answer is fault tolerance, specify if the practice can tolerate hardware failures and/or software failures, and if it is effective against transient faults or permanent faults.

Note: In case of doubt, choose the most appropriate one.

1. Using a strongly typed programming language.
2. Comparing the output of diverse implementations of a given specification running on separate hardware, and choosing the best result
3. Rigorous model-driven software development (requirements elicitation, analysis, architecture and design)
4. During development of a piece of software, using results obtained from one implementation of a given specification to test a different implementation of the same specification (back-to-back testing)
5. Gather software metrics during development
6. Structured programming
7. Invocation of alternate routine when primary exceeds time limit
8. Use of default input data when input data is out of range
9. Proof of correctness of a set of data manipulations
10. Upon detecting an error, restore a previously valid state and try again

3 Redo/NoUndo Recovery

Write pseudo-code that implements the `preOperation`, `postOperation`, `abortTransaction` and `commitTransaction` methods of the Redo/NoUndo recovery manager in the OPTIMA framework (similar to what has been shown in class for Undo/NoRedo recovery on slides 43 to 45). What cache policy can be used in combination with Redo/NoUndo recovery?

4 Duality of Fault-Tolerant System Structures

In the paper [SMR93], the authors show that the “Object and Transaction” model and the “Process and Conversation” model are duals of each other. In class, we looked at different concepts and their respective duals in the two models (hybrid systems slides, page 5).

Choose at least one concept from the transaction or conversation world (i.e. from one of the models Flat Transactions, Nested Transactions, Atomic Actions, Coordinated Atomic Actions, Open Multithreaded Transactions) that we looked at in class but that is not mentioned on page 5, and map it to the other model.

[SMR93] S. Shrivastava, L. Mancini and B. Randell: “The Duality of Fault-Tolerant System Structures”, Software Practice and Experience, Volume 23(7), pages 773 - 798, July 1993.

Grading

The grade will be based on correctness of your answers.

Hand-In

Please hand in a hard copy of your solution by Wednesday April 11th in class, or by sending an email to Wisam.Alabed@mail.mcgill.ca and Joerg.Kienzle@mcgill.ca!