Reliability Analysis of Software TMR System

Nahmsuk Oh
RATS Seminar
Jul 20, 1998

Outline
- Preliminaries
- Reliability modeling
  - Simplex system
  - Simplex system (discontinuous time execution)
  - Software TMR system
  - Software TMR with switching overhead
- Simulation result
- Conclusion

Preliminaries
- Reliability $R(t)$
  The conditional probability that the system operates correctly throughout the interval of time $[0,t]$, given that the system was in correct state at time 0.
- Assuming constant failure rate $\lambda(t) = \lambda$
- Mean time to failure (MTTF)
  $MTTF = \int_0^t R(t)dt = \frac{1}{\lambda}$

Discrete time $R[n]$
- Discrete time reliability function $R[n]$
  The conditional probability that the system operates correctly throughout the interval of cycles from 0 to $n$, given that the system was in error free state at cycle 0.
- No error in one cycle of period $T$
  $p = R(t)|_{t=T} = e^{-\lambda T}$
- No error in $N$ cycles
  $R[N] = P(\text{no error in N cycles}) = \prod_{i=1}^{N} P(\text{no error in ith cycles}) = p^N$

Simplex System

Simplex System (discontinuous execution time)
Reliability Analysis of Software TMR System

Simplex System
(discontinuous execution time)

\[ R(2N) = P(\text{no error in odd cycles}) = P(\text{no error in cycle 1})P(\text{no error in cycle 3}) \]
\[ = P(\text{no error in cycle 1}) \cdot P(\text{no error in cycle 2N - 1}) \]
\[ \propto \prod_{i=1}^{2N-1} P(\text{no error in cycle } i) \]
\[ = \prod_{i=1}^{2N-1} p \]
\[ = p^{2N-1} \]

Assumptions
- Independence of errors in each cycle
- Three copies of code exist.
- Only memory is triplicated
- No design change needed (just expand memory)

Error Model
- Transient errors in the processor:
  ALU, bus, combinational logic, register file...
- Bit flips in memory
  Three copies of memory can tolerate them.

Software TMR

Hardware TMR

Software TMR

Software TMR

Software TMR

Software TMR

Software TMR

Software TMR

Reliability Analysis of Software TMR System

Software TMR with switching overhead

$q = R_{os}\{L\} + p^L$

$\text{total time} = 3NT + 3\frac{NTL}{M}$

$R_{ps} = \text{P(at least two tasks have no error)} \text{P(no error in task switching)}$

$= \left[\prod_{i=1}^{L} R_{\text{cycle}} \right] \prod_{l=1}^{k} R_{\text{cycle}}$

$\text{A random number generator:}$

- generates a number between 0 and 1. If it is greater than $p$, simulated as an error.

- $k$ clock cycle mission:
  - $k$ random numbers are generated. If one of them is greater than $p$, simulated as an error during the mission.

- Software TMR:
  - Each of three tasks has $k$ clock cycles.
  - More than two tasks have errors, system fails.
Conclusion

- The reliability of software TMR system (with three copies of memory) approaches to the reliability of classical hardware TMR as the task switching overhead is minimized (about 1%).