An Efficient Fault-tolerant Scheduling Algorithm for Real-time Tasks with Precedence Constraints in Heterogeneous Systems

Xiao Qin    Hong Jiang    David R. Swanson

Department of Computer Science and Engineering

University of Nebraska-Lincoln
Overview

- Applied to embedded real-time heterogeneous systems
- Directed acyclic graph (DAG)
- Fault tolerance
- Reliability
Fault Tolerance ≠ Reliability

P1 FR = 1\times 10^{-6}

P2 FR = 5\times 10^{-6}

P3 FR = 8\times 10^{-6}
computational heterogeneity

- V1: E:20,8,10 D:55
- V2: E:6,18,8 D:72
- V3: E:10,22,7 D:70
- V4: E:9,12,10 D:80
- V5: E:12,8,10 D:75
- V6: E:12,24,10 D:95
Communicational heterogeneity

Weight = 1/Bandwidth

Weight = 1

Weight = 3

Determine communication time

V1: p1 → V2
2 * 1 = 2

V1: p2 → V2
2 * 3 = 6

V1: p3 → V2
2 * 3 = 6
Reliability cost = Failure Rate*Execution Time

Reliability heterogeneity

Failure rate p1 = 0.5*10^{-6}

Failure rate p2 = 1.5*10^{-6}

Failure rate p3 = 1.0*10^{-6}

V2: E:6,18,8

p1 \rightarrow p2: 0.5*10^{-6}*6 = 3*10^{-6}

p2 \rightarrow p3: 1.5*10^{-6}*18 = 2.7*10^{-5}

p3 \rightarrow p1: 1.0*10^{-6}*8 = 8*10^{-6}
Reliability cost = Failure Rate * Execution Time (cont.)

V1: E: 20, 8, 10

V2: E: 6, 18, 8
V3: E: 10, 22, 7

p1: Failure rate = 0.5 * 10^{-6}
V1: 0.5 * 10^{-6} * 20 = 10 * 10^{-6}

p3: Failure rate = 1.0 * 10^{-6}
V2: 1.0 * 10^{-6} * 8 = 8 * 10^{-6}
V3: 1.0 * 10^{-6} * 7 = 7 * 10^{-6}

Reliability cost of the schedule = 10 * 10^{-6} + 8 * 10^{-6} + 7 * 10^{-6} = 25 * 10^{-6}
Reliability cost = Failure Rate*Execution Time (cont.)

V1
E:20,8,10

V2
E:6,18,8

V3
E:10,22,7

p3
Failure rate=1.0*10^{-6}

V1:1.0*10^{-6}*10=10*10^{-6}

V2:0.5*10^{-6}*6=3*10^{-6}

V3:0.5*10^{-6}*10=5*10^{-6}

Reliability cost of the schedule
=10*10^{-6}+3*10^{-6}+5*10^{-6}
=18*10^{-6}
Fault Tolerance

![Diagram of fault tolerance]

- Primary copies
- Backup copies
- Overlap
Strong Primary Copy

$p_1$ fails, $\nu_i^P$ executes. $\nu_j^P$ cannot receive message from $\nu_i^B$, $\nu_j^B$ must execute instead of $\nu_j^P$. 
Principle 1

1. $v_i^P$ and $v_j^P$ are strong primary copies
2. $v_i^P$ and $v_j^P$ are scheduled on two different processors.

$v_i^B$ is not execution-preceding $v_j^B$.
1. $v_i^B$, is not schedule-preceding $v_j^p$
2. $v_i^p$ is a strong primary copy.

$v_j^B$ can not be scheduled on the processor on which $v_i^p$ is scheduled.
Principles 3

1. $v_i$ is the predecessor of $v_j$.
2. $v_i^P$ and $v_j^P$ are scheduled on the same processor.
3. $v_i^P$ is the strong primary copy.

In this case, $v_i^B$ is not execution-preceding $v_j^P$. 
Step 1: Sort Task

1. Deadlines
2. Precedence constrains

Task sequence = (V1, V3, V2, V5, V4, V6)
Step 2: Schedule Primary copies

1. Can be completed before deadline?
2. Reliability cost is minimized?
Step 2: Schedule Primary copies (cont.)
Step 3: Schedule Backup Copies

1. Determine candidate processors

2. For all candidate processors:
   2.1 Can be completed before deadline?
   2.2 Reliability cost is minimized?
Step 3: Schedule Backup Copies (cont.)
Step 3: Schedule Backup Copies (cont.)
Related Work

Algorithm eFRCRD

P1


P2

20
Related Work (cont.)


Performance Evaluation (cont.)

\[
\text{Schedulability} = \frac{\text{Number of jobs with feasible schedules}}{\text{Total number of tested jobs}}
\]

\[
\text{Reliability} = e^{-(\text{Reliability Cost})}
\]

\[
\text{Performability} = \text{Reliability} \times \text{Schedulability}
\]
Performance Evaluation (cont.)

Schedulability

Schedulability as a function of N. Common deadline = 100, m = 16, MIN_F = 0.5*10^-6, MAX_F = 3.0*10^-6, EX = [1, 20]
Performance Evaluation (cont.)

Reliability

Reliability as function of MAX_F. N = 50, m = 20,
MIN_F = 1*10^-6, EX = [500, 1500]
Performance Evaluation (cont.)

Heterogeneity Level

PF of btree and 4-ary tree as a function of heterogeneity level. \( H=[1,100] \), \( N=150 \), \( m=20 \), alpha=20
Conclusions

- Static Scheduling
- Real-time
- Fault tolerance
- Tasks with Precedence Constraints
- Heterogeneous Systems
Thank You