Poster: Fault-Tolerant Multi-Processor Scheduling with Backup Copy Technique
Petr Dobiáš, Emmanuel Casseau, Oliver Sinnen

To cite this version:
Petr Dobiáš, Emmanuel Casseau, Oliver Sinnen. Poster: Fault-Tolerant Multi-Processor Scheduling with Backup Copy Technique. Conference on Design and Architectures for Signal and Image Processing (DASIP), Sep 2017, Dresden, Germany. hal-01610745

HAL Id: hal-01610745
https://inria.hal.science/hal-01610745
Submitted on 5 Oct 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

HAL est une archive ouverte pluridisciplinaire destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Fault-Tolerant Multi-Processor Scheduling
with Backup Copy Technique

Petr Dobiáš, Emmanuel Casseau and Oliver Sinnen
emmanuel.casseau@irisa.fr

Aim: Development of a run-time, self-adaptive scheduling algorithm for Multi-Processor System-on-Chip taking into account faults and thereby treating unreliable system components

Keywords: Fault-tolerant design, Mapping and scheduling, Multi-processor platform

Context:
- Increasing demand on multi-processor systems for high performance and low power consumption
- Rising susceptibility of system failure because of transistor scaling and diminution of operating voltage

Issues:
- Trade-off between parallel computing (performance) and spatial redundancy on multi-cores (reliability)
- Mapping and scheduling control on multiprocessor platforms

Assumptions:
- P identical processors
- Aperiodic and independent tasks
- Dynamic mapping and scheduling without preemption
- Only one processor failure at any instant of time
- Primary/backup (PB) approach: each task T has two identical copies: primary copy (PC) and backup copy (BC)

**Principle of Mapping and Scheduling Based on the Primary/Backup Approach**

Algorithm to map and schedule tasks
- Map and schedule PC (as soon as possible)
- Map and schedule BC (as late as possible)
- If PC and BC schedules found:
  - YES: Commit the task
  - NO: Reject the task
- After execution of PC, the fault detection mechanism reports:
  - Faulty execution of PC?
    - YES: Wait for results of BC
    - NO: Deallocate BC
- PC and BC schedules found?
  - YES: Commit the task
  - NO: Reject the task

Fault management algorithm
- Faulty execution of PC?
  - YES: Wait for results of BC
  - NO: Deallocate BC

### Table: Task parameters

<table>
<thead>
<tr>
<th>Task attribute</th>
<th>Distribution</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arrival time $a$</td>
<td>Poisson</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Computation time $c$</td>
<td>Uniform</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td>Deadline $d$</td>
<td>Uniform</td>
<td>2$c$</td>
<td>5$c$</td>
</tr>
</tbody>
</table>

### Table: Simulation parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of processors</td>
<td>2-25</td>
</tr>
<tr>
<td>Targeted processor load (TPL)</td>
<td>0.25-2.25</td>
</tr>
</tbody>
</table>

**Results**

- Task rejection rate (TPL=1)
- Processor load (TPL=1)
- Time to next fault in case of permanent fault (TPL=1)