A Resource Minimizing Scheduling Algorithm with Ensuring the Deadline and Reliability in Heterogeneous Systems

Laiping Zhao\textsuperscript{1}

Yizhi Ren\textsuperscript{2}, Kouichi Sakurai\textsuperscript{1}
\textsuperscript{1}Department of Informatics, Kyushu University, Japan
\textsuperscript{2}School of Software Engineering, Hangzhou Dianzi University, China

The first author is supported by the governmental scholarship from China Scholarship Council.
Fault tolerant technologies

- Two approaches for fault tolerance
  - Primary/Backup (Qin & Jiang, 2006; Zheng et al. 2009)
    - The backup is activated after the primary fails.
  - Active replication (Benoit et al. 2008, 2009; Zhao et al. 2010)
    - Simultaneously assign $f+1$ replicas to different processors to tolerate $f$ failures.
A comparison

<table>
<thead>
<tr>
<th></th>
<th>Primary/Backup</th>
<th>Active replication</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Advantages</strong></td>
<td>Less resource usage</td>
<td>Quick response</td>
</tr>
<tr>
<td><strong>Disadvantages</strong></td>
<td>Increase the execution time</td>
<td>Waste many resources</td>
</tr>
<tr>
<td><strong>Target system</strong></td>
<td>System with high system load</td>
<td>Hard time critical system with abundant resources</td>
</tr>
</tbody>
</table>

Energy consumption
Economic cost
System throughput
etc...
Outline

◆ Placing replicas to reliable processors
  ➢ Reliability
  ➢ Scheduling with meeting deadline
  ➢ Task priority

◆ DRR scheduling
◆ Experiment
◆ Conclusion and future works
Active replication scheme with dynamic number of replicas (ARD):
- Providing the same reliability as active replication (AR);
- Using less replicas;
- Ensuring the deadline.
Placing replicas to reliable processors

Workflow

AR (Tolerate 2 failures)

Workflow

ARD
Placing replicas to reliable processors

- Providing the same reliability:

- How to provide a comparable reliability with less replicas?
  - Select processors with higher reliability.
Reliability

- Approaches for reliability analysis/prediction exist:
  - Javadi, Kondo, Vincent and Anderson (2009)
  - Narasimhan et al. (2005)
  - Oliner, Sahoo, Moreira, Gupta and Sivasubramaniam (2004)
  - Rood and Lewis (2010)

- For simplification,
  - Qin and Jiang (2006); Jin et al. (2009) etc.
  - Assume the failure distribution follows a Poisson process.
Reliability

- Reliability consists of computation and communication reliability.
- **Computation reliability**
  \[ f(k, \lambda) = \frac{\lambda^k e^{-\lambda}}{k!} \]
- **Communication reliability**
  - Processors in two tracks (e.g. 1 and 3)
    - Two possible communication paths
  - Processors in the same track (e.g. 1 and 2)
    - One communication path
scheduler

\[ R(AR) = R(ARD) \]

The subreliability for a task should not less than: \[ r = \sqrt[n]{R} \], (\( n \) is the number of tasks in a workflow).
Scheduling with meeting deadline

- Breadth First Search (BFS) based subdeadline assignment:

  - Subdeadline of the exit tasks equal to the overall deadline $T$.
  - Parents’ subdeadline
    
    = Children’s subdeadline – communication – execution;
  - Select the minimum value if multiple results are gotten.
Task priority

- Sorting tasks from a workflow into order, while ensuring the precedence relationship.

- Two approaches:
  - **Bottom level (bl)**
    - \( bl(x) = exe(x) + \text{comm}(x,y) + bl(y), \) \( y \) is a child of \( x \).
    - HEFT (Topcuoglu et al. 2002); MaxRe(Zhao et al. 2010)
  
  - **Top level + Bottom level (tl+bl)**
    - \( tl(x) = tl(z) + exe(z) + \text{comm}(z,x), \) \( z \) is a parent of \( x \).
    - FTSA(Benoit et al. 2008); CAFT(Benoit et al. 2009); CPOP(Topcuoglu et al. 2002).
FTSA(bl) (Benoit et al. 2008) has a better performance than FTSA(tl+bl). Therefore, we use the tl value in our scheduling.
Outline

◆ Placing replicas to reliable processors
  ➢ Reliability
  ➢ Scheduling with meeting deadline
  ➢ Task priority

◆ DRR scheduling

◆ Experiment results

◆ Conclusion and future works
Deadline-Reliability-Resource-aware (DRR) scheduling:

DRR algorithm
Deadline-Reliability-Resource-aware (DRR) scheduling:

DRR algorithm
If a task subdeadline is violated by its finish time, the job will be rejected.

- Proved by apagoge.

For any entry task $x$, $ET(x) < subdeadline(x)$ cannot guarantee the job’s successful completeness within deadline $T$.

- Because of processor contention.
Processor contention results in deadline violation.

- In the case of ready time > arrive time,
  Subdeadline(t3) – ET(t3) < ET(t1),
- The job will be rejected.
Outline

- Placing replicas to reliable processors
  - Reliability
  - Scheduling with meeting deadline
  - Task priority
- DRR scheduling
- Experiments
- Conclusion and future works
Experiments

- **Simulation**
  - Randomly generated graphs
  - Heterogeneous system

- **Evaluation metrics:**
  - *Computation cost*
    - Defined as the overall computation time of all processors.
  - *Communication cost*
    - Defined as the overall communication time of all communication links.
Experiments

- Computation resource usage:

DRR and MaxRe save 50% computation resources.
Experiments

- Communication resource usage:

DRR and MaxRe save 70% communication resources.
Conclusion and future works

◆ This work improve the active replication scheme:
  - Achieves corresponding reliability with less resources;
  - Guarantee the deadline constraint.

◆ The future work:
  - A comprehensive study of both primary/backup and active replication based scheduling.
Thanks for the listening
Q&A