Optimizing Latency and Reliability of Pipeline Workflow Applications

Anne Benoit  Veronika Rehn-Sonigo  Yves Robert

GRAAL team, LIP  
École Normale Supérieure de Lyon  
France

HCW 2008
Introduction and motivation

- Mapping applications onto parallel platforms
  **Difficult challenge**
- Heterogeneous clusters, fully heterogeneous platforms
  **Even more difficult!**
- Structured programming approach
  - Easier to program (deadlocks, process starvation)
  - Range of well-known paradigms (pipeline, farm)
  - Algorithmic skeleton: help for mapping
Introduction and motivation

- Mapping applications onto parallel platforms
  Difficult challenge

- Heterogeneous clusters, fully heterogeneous platforms
  Even more difficult!

- Structured programming approach
  - Easier to program (deadlocks, process starvation)
  - Range of well-known paradigms (pipeline, farm)
  - Algorithmic skeleton: help for mapping

Mapping pipeline skeletons onto heterogeneous platforms
Introduction and motivation

- Mapping applications onto parallel platforms
  
  Difficult challenge

- Heterogeneous clusters, fully heterogeneous platforms
  
  Even more difficult!

- Structured programming approach
  
  - Easier to program (deadlocks, process starvation)
  - Range of well-known paradigms (pipeline, farm)
  - Algorithmic skeleton: help for mapping

Mapping pipeline skeletons onto heterogeneous platforms
Multi-criteria scheduling of workflows

Workflow

Several consecutive data-sets enter the application graph.

Multi-criteria?

Latency: maximal time elapsed between beginning and end of execution of a data set

Failure: the probability that a processor fails during execution

Bi-criteria!
Multi-criteria scheduling of workflows

Workflow

Several consecutive data-sets enter the application graph.

Multi-criteria?

**Latency:** maximal time elapsed between beginning and end of execution of a data set

**Failure:** the probability that a processor fails during execution

Bi-criteria!
Multi-criteria scheduling of workflows

Workflow

Several consecutive data-sets enter the application graph.

Multi-criteria?

**Latency:** maximal time elapsed between beginning and end of execution of a data set

**Failure:** the probability that a processor fails during execution

Bi-criteria!
Multi-criteria scheduling of workflows

Workflow

Several consecutive data-sets enter the application graph.

Multi-criteria?

**Latency**: maximal time elapsed between beginning and end of execution of a data set

**Failure**: the probability that a processor fails during execution

Bi-criteria!
Rule of the game

- Map each pipeline stage on a single processor
- Goal: minimize latency AND minimize failure probability

Several mapping strategies

The pipeline application
Rule of the game

- Map each pipeline stage on a single processor
- Goal: minimize latency **AND** minimize failure probability
- Several mapping strategies

The pipeline application

\[ S_1 \rightarrow S_2 \rightarrow \ldots \rightarrow S_k \rightarrow \ldots \rightarrow S_n \]
Rule of the game

- Map each pipeline stage on a single processor
- Goal: minimize latency AND minimize failure probability
- Several mapping strategies

One-to-one Mapping
Rule of the game

- Map each pipeline stage on a single processor
- Goal: minimize latency **AND** minimize failure probability
- Several mapping strategies

```
S_1 -> S_2 -> ... -> S_k -> ... -> S_n
```

**Interval Mapping**
Rule of the game

- Map each pipeline stage on a single processor
- Goal: minimize latency **AND** minimize failure probability
- Several mapping strategies

\[ S_1 \rightarrow S_2 \rightarrow \ldots \rightarrow S_k \rightarrow \ldots \rightarrow S_n \]

**General Mapping**
Rule of the game

- Map each pipeline stage on a single processor
- Goal: minimize latency AND minimize failure probability
- Several mapping strategies

**Interval Mapping**

- Replication (one interval onto several processors) in order to increase reliability
Major Contributions

- Definition of bi-criteria mapping
- Complexity results
  - Mono-criterion problems
  - Bi-criteria problems
- Optimal algorithms
Outline

1. Framework
2. Motivating Examples
3. Complexity Results
   - Mono-criterion Problems
   - Bi-criteria Problems
4. Conclusion
The application

- n stages $S_k$, $1 \leq k \leq n$
- $S_k$:
  - receives input of size $\delta_{k-1}$ from $S_{k-1}$
  - performs $w_k$ computations
  - outputs data of size $\delta_k$ to $S_{k+1}$
- $S_0$ and $S_{n+1}$: virtual stages representing the outside world
The platform

- \( p \) processors \( P_u, 1 \leq u \leq p \), fully interconnected
- \( s_u \): speed of processor \( P_u \)
- bidirectional link \( \text{link}_{u,v} : P_u \rightarrow P_v \), bandwidth \( b_{u,v} \)
- \( \text{fp}_u \): failure probability of processor \( P_u \) (independent of duration, meant to run for a long time)
- one-port model: each processor can either send, receive or compute at any time-step
Different platforms

**Fully Homogeneous** – Identical processors \((s_u = s)\) and links \((b_{u,v} = b)\): typical parallel machines

**Communication Homogeneous** – Different-speed processors \((s_u \neq s_v)\), identical links \((b_{u,v} = b)\): networks of workstations, clusters

**Fully Heterogeneous** – Fully heterogeneous architectures, \(s_u \neq s_v\) and \(b_{u,v} \neq b_{u',v'}\): hierarchical platforms, grids
Different platforms

**Fully Homogeneous** – Identical processors \((s_u = s)\) and links \((b_{u,v} = b)\): typical parallel machines

**Failure Homogeneous** – Identically reliable processors \((fp_u = fp_v)\)

**Communication Homogeneous** – Different-speed processors \((s_u \neq s_v)\), identical links \((b_{u,v} = b)\): networks of workstations, clusters

**Fully Heterogeneous** – Fully heterogeneous architectures, \(s_u \neq s_v\) and \(b_{u,v} \neq b_{u',v'}\): hierarchical platforms, grids

**Failure Heterogeneous** – Different failure probabilities \((fp_u \neq fp_v)\)
Mapping problem: **Interval Mapping**

- Partition of \([1..n]\) into \(m\) intervals \(I_j = [d_j, e_j]\) (with \(d_j \leq e_j\) for \(1 \leq j \leq m\), \(d_1 = 1\), \(d_{j+1} = e_j + 1\) for \(1 \leq j \leq m - 1\) and \(e_m = n\))

- Interval \(I_j\) mapped onto set of processors \(P_{\text{alloc}(j)}\)

\[
FP = 1 - \prod_{1 \leq j \leq p} (1 - \prod_{u \in \text{alloc}(j)} fp_u)
\]
Mapping problem: **Interval Mapping**

- Partition of $[1..n]$ into $m$ intervals $I_j = [d_j, e_j]$
  (with $d_j \leq e_j$ for $1 \leq j \leq m$, $d_1 = 1$, $d_{j+1} = e_j + 1$ for $1 \leq j \leq m - 1$ and $e_m = n$)

- Interval $I_j$ mapped onto set of processors $P_{\text{alloc}(j)}$

\[
FP = 1 - \prod_{1 \leq j \leq p} \left( 1 - \prod_{u \in \text{alloc}(j)} \text{fp}_u \right)
\]
Mapping problem: **Interval Mapping**

- Partition of $[1..n]$ into $m$ intervals $I_j = [d_j, e_j]$
  (with $d_j \leq e_j$ for $1 \leq j \leq m$, $d_1 = 1$, $d_{j+1} = e_j + 1$ for $1 \leq j \leq m - 1$ and $e_m = n$)

- Interval $I_j$ mapped onto set of processors $P_{\text{alloc}(j)}$

\[
FP = 1 - \prod_{1 \leq j \leq p} (1 - \prod_{u \in \text{alloc}(j)} fp_u)
\]

\[
\mathcal{L} = \sum_{1 \leq j \leq p} \left\{ k_j \times \frac{\delta_{d_j-1}}{b} + \frac{\sum_{i=d_j}^{e_j} w_i}{\min_{u \in \text{alloc}(j)} (s_u)} \right\} + \frac{\delta_n}{b}
\]
Mapping problem: **Interval Mapping**

- Partition of \([1..n]\) into \(m\) intervals \(I_j = [d_j, e_j]\) (with \(d_j \leq e_j\) for \(1 \leq j \leq m\), \(d_1 = 1\), \(d_{j+1} = e_j + 1\) for \(1 \leq j \leq m - 1\) and \(e_m = n\))

- Interval \(I_j\) mapped onto set of processors \(P_{\text{alloc}(j)}\)

\[
\mathcal{FP} = 1 - \prod_{1 \leq j \leq p} \left(1 - \prod_{u \in \text{alloc}(j)} fp_u \right)
\]

\[
\mathcal{L} = \sum_{u \in \text{alloc}(1)} \frac{\delta_0}{b_{in,u}} + \sum_{1 \leq j \leq p} \max_{u \in \text{alloc}(j)} \left\{ \frac{\sum_{i=d_j}^{e_j} w_i}{s_u} + \sum_{v \in \text{alloc}(j+1)} \frac{\delta_{e_j}}{b_{u,v}} \right\}
\]

Veronika.Sonigo@ens-lyon.fr  
HCW 2008  
Optimizing Latency and Reliability  
10/27
Objective function?

Mono-criterion

- Minimize $L$
- Minimize $FP$

Bi-criteria

- How to define it?
  Minimize $\alpha \cdot L + \beta \cdot FP$?
- Values which are not comparable
- Minimize $L$ for a fixed failure probability
- Minimize $FP$ for a fixed latency
Objective function?

Mono-criterion

- Minimize $\mathcal{L}$
- Minimize $FP$

Bi-criteria

- How to define it?
  Minimize $\alpha \mathcal{L} + \beta FP$?
- Values which are not comparable
- Minimize $\mathcal{L}$ for a fixed failure probability
- Minimize $FP$ for a fixed latency
Objective function?

Mono-criterion

- Minimize $\mathcal{L}$
- Minimize $FP$

Bi-criteria

- How to define it?
  Minimize $\alpha \mathcal{L} + \beta FP$?
- Values which are not comparable
  - Minimize $\mathcal{L}$ for a fixed failure probability
  - Minimize $FP$ for a fixed latency
Objective function?

Mono-criterion

- Minimize $\mathcal{L}$
- Minimize $FP$

Bi-criteria

- How to define it?
  Minimize $\alpha \mathcal{L} + \beta FP$
- Values which are not comparable
- Minimize $\mathcal{L}$ for a fixed failure probability
- Minimize $FP$ for a fixed latency
Outline

1. Framework

2. Motivating Examples

3. Complexity Results
   - Mono-criterion Problems
   - Bi-criteria Problems

4. Conclusion
Mono-criterion - Interval Mapping

Minimize $\mathcal{L}$

Comm. Hom. Platform

Hetero. Platform
Mono-criterion - Interval Mapping

Minimize $\mathcal{L}$

Comm. Hom. Platform

Hetero. Platform

Veronika.Sonigo@ens-lyon.fr HCW 2008
Optimizing Latency and Reliability 13/27
Mono-criterion - Interval Mapping

Minimize $\mathcal{L}$

Comm. Hom. Platform

Hetero. Platform
Bi-criteria - Interval Mapping

Minimize $FP$ with fixed latency
Communication homogeneous - Failure heterogeneous

Fixed latency: 22

$w_1 = 1$
$w_2 = 100$

$s = 1, fp = 0.1$
$s = 100, fp = 0.8$
Bi-criteria - Interval Mapping

Minimize \( FP \) with fixed latency

Communication homogeneous - Failure heterogeneous

Fixed latency: 22

\[ w_1 = 1, \quad w_2 = 100 \]

\[ 10 + 101 \gg 22 \]

\( s = 1, \; fp = 0.1 \)

\( s = 100, \; fp = 0.8 \)
Bi-criteria - Interval Mapping

Minimize $FP$ with fixed latency
Communication homogeneous - Failure heterogeneous

Fixed latency: 22

$10 \rightarrow S_1 \rightarrow 1 \rightarrow S_2 \rightarrow 0$

$w_1 = 1 \quad w_2 = 100$

$20 + 101/100 < 22$

$FP = (1 - (1 - 0.8^2)) = 0.64$
Minimize $FP$ with fixed latency
Communication homogeneous - Failure heterogeneous

Fixed latency: 22

$w_1 = 1, w_2 = 100$

$30 + 101/100 > 22$

$s = 1, fp = 0.1$

$s = 100, fp = 0.8$
Minimize $FP$ with fixed latency
Communication homogeneous - Failure heterogeneous

Fixed latency: 22

$10 + 1/1 + 10 \times 1 + 100/100 = 22$

$FP : 1-(1-0.1)\times(1-0.8^{10}) < 0.2$
Outline

1. Framework
2. Motivating Examples
3. Complexity Results
   - Mono-criterion Problems
   - Bi-criteria Problems
4. Conclusion
Mono-criterion Problems

Minimize the failure probability?

**Theorem 1**

Minimizing the failure probability can be done in polynomial time.

- Replicate the whole pipeline as a single interval.
- Use all processors.
- True for all platform types.
Mono-criterion Problems

Minimize the failure probability?

**Theorem 1**

Minimizing the failure probability can be done in polynomial time.

- Replicate the whole pipeline as a single interval.
- Use all processors.
- True for all platform types.
Mono-criterion Problems

Minimize the failure probability?

Theorem 1
Minimizing the failure probability can be done in polynomial time.

- Replicate the whole pipeline as a single interval.
- Use all processors.
- True for all platform types.
Mono-criterion Problems

Minimize the latency?

**Theorem 2**

Minimizing the latency can be done in polynomial time on *Communication Homogeneous* platforms.

Idea:

- Latency is optimized by suppressing all communications.
- Replication increases latency (additional communication).

Map whole pipeline on fastest processor.
Mono-criterion Problems

Minimize the latency?

Theorem 2

Minimizing the latency can be done in polynomial time on *Communication Homogeneous* platforms.

Idea:

- Latency is optimized by suppressing all communications.
- Replication increases latency (additional communication).

Map whole pipeline on fastest processor.
Minimize the latency?

Theorem 2

Minimizing the latency can be done in polynomial time on Communication Homogeneous platforms.

Idea:

- Latency is optimized by suppressing all communications.
- Replication increases latency (additional communication).

Map whole pipeline on fastest processor.
Mono-criterion Problems

Minimize the latency?
What about *Fully Heterogeneous* platforms?

Remember example:

![Diagram showing two stages S1 and S2 with one-to-one mappings.]

**Theorem 3**

Minimizing the latency is NP-hard on *Fully Heterogeneous* platforms for one-to-one mappings.
Mono-criterion Problems

But ... considering general mappings ...

Theorem 4
Minimizing the latency is polynomial on *Fully Heterogeneous* platforms for general mappings.
Mono-criterion Problems

But ... considering general mappings ...

Theorem 4

Minimizing the latency is polynomial on *Fully Heterogeneous* platforms for general mappings.

Optimal mapping: Shortest path in the graph.
Mono-criterion Problems

But ... considering general mappings ...

Theorem 4
Minimizing the latency is polynomial on *Fully Heterogeneous* platforms for general mappings.

Optimal mapping: Shortest path in the graph.

Interval mapping: still an open problem
Bi-criteria Problems

\[ 1 - (1 - fp^{a+b}) \leq 1 - ((1 - fp^a)(1 - fp^b)) \]

**Lemma**

On *Fully Homogeneous* and *Communication Homogeneous-Failure Homogeneous* platforms, there is a mapping of the pipeline as a single interval which minimizes the failure probability under a fixed latency threshold, and there is a mapping of the pipeline as a single interval which minimizes the latency under a fixed failure probability threshold.
Bi-criteria Problems

\[ 1 - (1 - fp^{a+b}) \leq 1 - ((1 - fp^a)(1 - fp^b)) \]

Lemma

On *Fully Homogeneous* and *Communication Homogeneous-Failure Homogeneous* platforms, there is a mapping of the pipeline as a single interval which minimizes the failure probability under a fixed latency threshold, and there is a mapping of the pipeline as a single interval which minimizes the latency under a fixed failure probability threshold.
Bi-criteria Problems

\[1 - (1 - fp^{a+b}) \leq 1 - ((1 - fp^a)(1 - fp^b))\]

Lemma

On *Fully Homogeneous* and *Communication Homogeneous-Failure Homogeneous* platforms, there is a mapping of the pipeline as a single interval which minimizes the failure probability under a fixed latency threshold, and there is a mapping of the pipeline as a single interval which minimizes the latency under a fixed failure probability threshold.
Fully Homogeneous platforms

Minimize $FP$ for a fixed latency $\mathcal{L}$

Algorithm 1

begin

Find $k$ maximum, such that

$$k \times \frac{\delta_0}{b} + \sum_{1 \leq j \leq n} \frac{w_j}{s} + \frac{\delta_n}{b} \leq \mathcal{L}$$

Replicate the whole pipeline as a single interval onto the $k$ (most reliable) processors

end
Fully Homogeneous platforms

Minimize $FP$ for a fixed latency $\mathcal{L}$

Algorithm 1

begin

Find $k$ maximum, such that

$$k \times \frac{\delta_0}{b} + \frac{\sum_{1 \leq j \leq n} w_j}{s} + \frac{\delta_n}{b} \leq \mathcal{L}$$

Replicate the whole pipeline as a single interval onto the $k$ (most reliable) processors

end
**Fully Homogeneous platforms**

Minimize $\mathcal{L}$ for a fixed failure probability $FP$

**Algorithm 2**

```plaintext
begin
  Find $k$ minimum, such that
  
  $$1 - (1 - fp^k) \leq FP$$

  Replicate the whole pipeline as a single interval onto the $k$ (most reliable) processors

end
```
Fully Homogeneous platforms

Minimize $\mathcal{L}$ for a fixed failure probability $FP$

Algorithm 2

\begin{verbatim}
begin
  Find $k$ minimum, such that
  \[1 - (1 - fp^k) \leq FP\]
  Replicate the whole pipeline as a single interval onto the $k$ (most reliable) processors
end
\end{verbatim}
Other Platform Configurations

**Communication Homogeneous platforms - Failure Homogeneous**

Slightly modified *Fully Homogeneous* algorithms are optimal.

**Communication Homogeneous platforms - Failure Heterogeneous**

Lemma does not hold anymore.
Remember example.
Open problem

**Fully Heterogeneous platforms**

On *Fully Heterogeneous* platforms, the bi-criteria (decision problems associated to the) optimization problems are NP-hard.
**Other Platform Configurations**

*Communication Homogeneous platforms - Failure Homogeneous*

Slightly modified *Fully Homogeneous* algorithms are optimal.

*Communication Homogeneous platforms - Failure Heterogeneous*

Lemma does not hold anymore.
Remember example.

**Open problem**

*Fully Heterogeneous platforms*

On *Fully Heterogeneous* platforms, the bi-criteria (decision problems associated to the) optimization problems are NP-hard.
Other Platform Configurations

**Communication Homogeneous platforms - Failure Homogeneous**
Slightly modified *Fully Homogeneous* algorithms are optimal.

**Communication Homogeneous platforms - Failure Heterogeneous**
Lemma does not hold anymore.
Remember example.
Open problem

**Fully Heterogeneous platforms**
On *Fully Heterogeneous* platforms, the bi-criteria (decision problems associated to the) optimization problems are NP-hard.
1. Framework

2. Motivating Examples

3. Complexity Results
   - Mono-criterion Problems
   - Bi-criteria Problems

4. Conclusion
Related work

**Subhlok and Vondran**  Latency and throughput optimization on pipeline graphs (homogeneous platforms only)

**Benoit et al.**  Extension of the work of Subholk and Vondran

**Mapping pipelined computations onto clusters and grids**  DAG

[Taura et al.], DataCutter [Saltz et al.]

**Energy-aware mapping of pipelined computations**  [Melhem et al.], three-criteria optimization

**Mapping pipelined computations onto special-purpose architectures**  FPGA arrays [Fabiani et al.]. Fault-tolerance for embedded systems [Zhu et al.]

**Real World Application**  Motion-JPEG
Conclusion

- Bi-criteria mapping problem: latency and reliability
- Pipeline structured workflow applications
- Complexity study

### Interval Mapping

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Mono-crit.</td>
<td>(\mathcal{L})</td>
<td>polyn.</td>
<td>polyn.</td>
</tr>
<tr>
<td></td>
<td>(\mathcal{FP})</td>
<td>polyn.</td>
<td>polyn.</td>
</tr>
<tr>
<td>Bi-crit.</td>
<td>(\mathcal{L} - \mathcal{FP}) hom</td>
<td>polyn.</td>
<td>polyn.</td>
</tr>
<tr>
<td></td>
<td>(\mathcal{L} - \mathcal{FP}) het</td>
<td>polyn.</td>
<td>?</td>
</tr>
</tbody>
</table>

\(\text{min } \mathcal{L}\), one-to-one mapping: NP

\(\text{min } \mathcal{L}\), general mapping: polynomial
## Conclusion

- Bi-criteria mapping problem: latency and reliability
- Pipeline structured workflow applications
- Complexity study

### Interval Mapping

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{L}$</td>
<td>polyn.</td>
<td>polyn.</td>
<td>$\mathbb{?}$</td>
</tr>
<tr>
<td>$\mathbb{FP}$</td>
<td>polyn.</td>
<td>polyn.</td>
<td>polyn.</td>
</tr>
<tr>
<td>$\mathcal{L} \cdot \mathbb{FP}$</td>
<td>polyn.</td>
<td>polyn.</td>
<td>NP</td>
</tr>
<tr>
<td>$\mathcal{L} \cdot \mathbb{FP}$</td>
<td>polyn.</td>
<td>$\mathbb{?}$</td>
<td>NP</td>
</tr>
</tbody>
</table>

- $\min \mathcal{L}$, one-to-one mapping: NP
- $\min \mathcal{L}$, general mapping: polynomial
Future work

Theory

- Extension to fork, fork-join and tree workflows
- Multi-criteria: throughput in addition to reliability and latency

Practice

- Design of multi-criteria heuristics
- Comparison of effective performance against theoretical performance
- Real experiments on heterogeneous clusters with different applications, using MPI