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

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

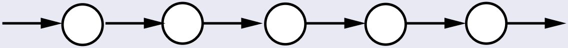
Introduction and motivation

- 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?

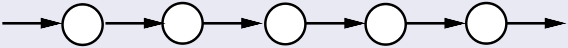
Latency: maximal time elapsed between beginning and end of execution of a data set

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?

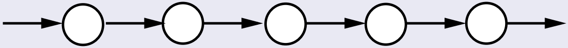
Latency: maximal time elapsed between beginning and end of execution of a data set

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?

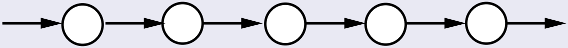
Latency: maximal time elapsed between beginning and end of execution of a data set

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?

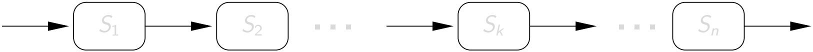
Latency: maximal time elapsed between beginning and end of execution of a data set

Failure: the probability that a processor fails during execution

Bi-criteria!

Rule of the game

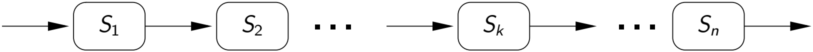
- Map each pipeline stage on a single processor
- Goal: minimize latency AND minimize failure probability
- Several mapping strategies



The pipeline application

Rule of the game

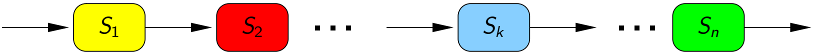
- Map each pipeline stage on a single processor
- Goal: minimize latency AND minimize failure probability
- Several mapping strategies



The pipeline application

Rule of the game

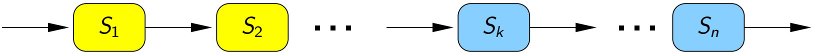
- Map each pipeline stage on a single processor
- Goal: minimize latency AND minimize failure probability
- Several mapping strategies



ONE-TO-ONE MAPPING

Rule of the game

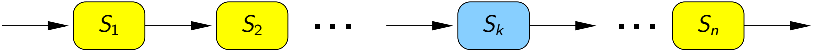
- Map each pipeline stage on a single processor
- Goal: minimize latency AND minimize failure probability
- Several mapping strategies



INTERVAL MAPPING

Rule of the game

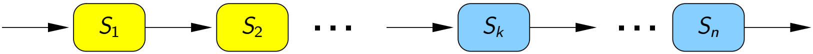
- Map each pipeline stage on a single processor
- Goal: minimize latency AND minimize failure probability
- Several mapping strategies



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

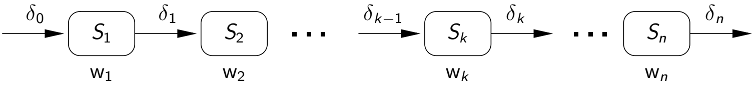
Major Contributions

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

Outline

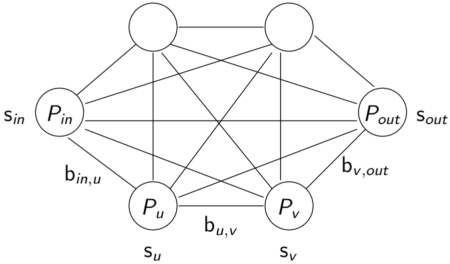
1. Framework
2. Motivating Examples
3. Complexity Results
   - Mono-criterion Problems
   - Bi-criteria Problems
4. Conclusion

The application



- n stages $\mathcal{S}_k$, $1 \leq k \leq$ n
- $\mathcal{S}_k$:
  - receives input of size δ_{k-1} from $\mathcal{S}_{k-1}$
  - performs w_k computations
  - outputs data of size δ_k to $\mathcal{S}_{k+1}$
- $\mathcal{S}_0$ and $\mathcal{S}_{n+1}$: virtual stages representing the outside world

The platform



- p processors P_u, $1 \leq u \leq \text{p}$, fully interconnected
- s_u: speed of processor P_u
- bidirectional link $\text{link}_{u,v} : P_u \rightarrow P_v$, bandwidth $b_{u,v}$
- 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 $e_m = \text{n}$)
- Interval I_j mapped onto set of processors $P_{\text{alloc}(j)}$

$$\mathcal{FP} = 1 - \prod_{1 \leq j \leq p} (1 - \prod_{u \in \text{alloc}(j)} \text{fp}_u)$$

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

$$\mathcal{FP} = 1 - \prod_{1 \leq j \leq p} (1 - \prod_{u \in \text{alloc}(j)} \text{fp}_u)$$

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 $e_m = \mathrm{n}$)
- Interval I_j mapped onto set of processors $P_{\mathrm{alloc}(j)}$

$$\mathcal{FP} = 1 - \prod_{1 \leq j \leq p} (1 - \prod_{u \in \mathrm{alloc}(j)} \mathrm{fp}_u)$$

$$\mathcal{L} = \sum_{1 \leq j \leq p} \left\{ k_j \times \frac{\delta_{d_j - 1}}{\mathrm{b}} + \frac{\sum_{i=d_j}^{e_j} \mathrm{w}_i}{\min_{u \in \mathrm{alloc}(j)}(\mathrm{s}_u)} \right\} + \frac{\delta_n}{\mathrm{b}}$$

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 $P_{\text{alloc}(j)}$

$$\mathcal{FP} = 1 - \prod_{1 \leq j \leq p} (1 - \prod_{u \in \text{alloc}(j)} \text{fp}_u)$$

$$\mathcal{L} = \sum_{u \in \text{alloc}(1)} \frac{\delta_0}{\text{b}_{in,u}} + \sum_{1 \leq j \leq p} \max_{u \in \text{alloc}(j)} \left\{ \frac{\sum_{i=d_j}^{e_j} \text{w}_i}{\text{s}_u} + \sum_{v \in \text{alloc}(j+1)} \frac{\delta_{e_j}}{\text{b}_{u,v}} \right\}$$

Objective function?

Mono-criterion

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

Bi-criteria

- How to define it?
  Minimize $\alpha.\mathcal{L} + \beta.\mathcal{FP}$?
- Values which are not comparable

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

Objective function?

Mono-criterion

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

Bi-criteria

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

Objective function?

Mono-criterion

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

Bi-criteria

- How to define it?
  Minimize $\alpha.\mathcal{L} + \beta.\mathcal{FP}$?
- Values which are not comparable

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

Objective function?

Mono-criterion

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

Bi-criteria

- How to define it?
  Minimize $\alpha.\mathcal{L} + \beta.\mathcal{FP}$?
- Values which are not comparable

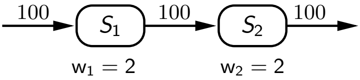
- Minimize $\mathcal{L}$ for a fixed failure probability
- Minimize $\mathcal{FP}$ for a fixed latency

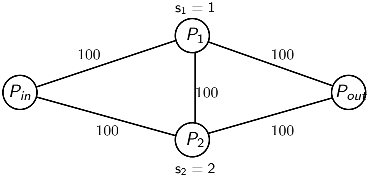
Outline

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

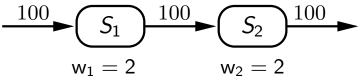
Mono-criterion - Interval Mapping

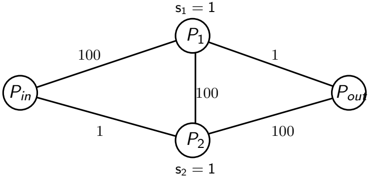
Minimize $\mathcal{L}$





Comm. Hom. Platform

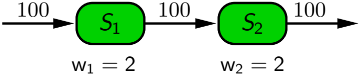


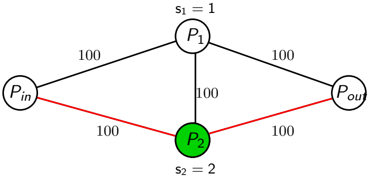


Hetero. Platform

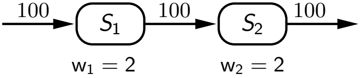
Mono-criterion - Interval Mapping

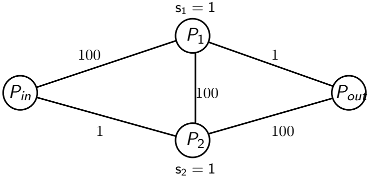
Minimize $\mathcal{L}$





Comm. Hom. Platform

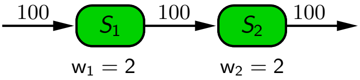


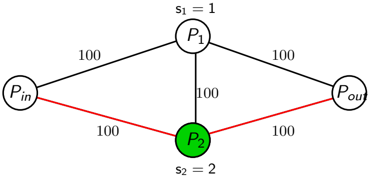


Hetero. Platform

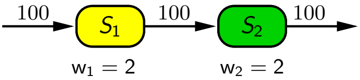
Mono-criterion - Interval Mapping

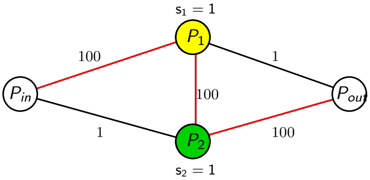
Minimize $\mathcal{L}$





Comm. Hom. Platform





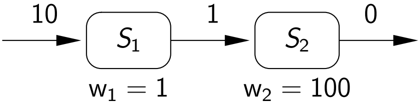
Hetero. Platform

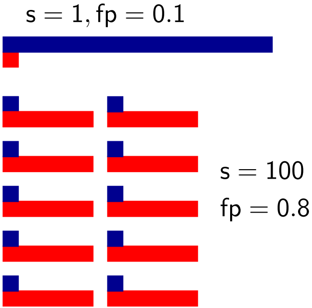
Bi-criteria - Interval Mapping

Minimize $\mathcal{FP}$ with fixed latency

Communication homogeneous - Failure heterogeneous

Fixed latency: 22

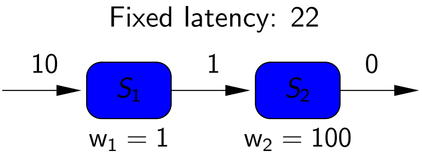




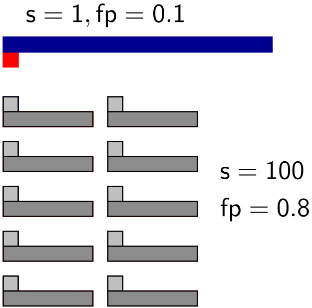
Bi-criteria - Interval Mapping

Minimize $\mathcal{FP}$ with fixed latency

Communication homogeneous - Failure heterogeneous



$10 + 101 \gg 22$

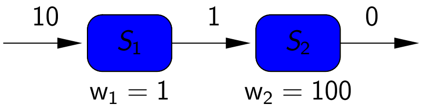


Bi-criteria - Interval Mapping

Minimize $\mathcal{FP}$ with fixed latency

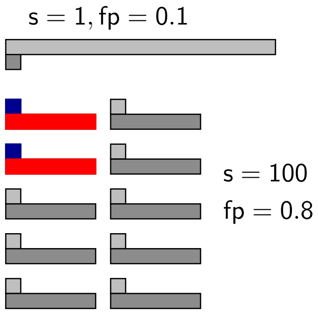
Communication homogeneous - Failure heterogeneous

Fixed latency: 22



$20 + 101/100 < 22$

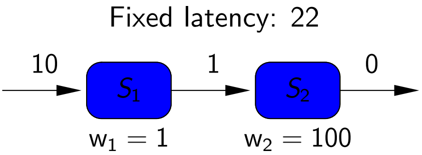
$\mathcal{FP} = (1 - (1 - 0.8^2)) = 0.64$



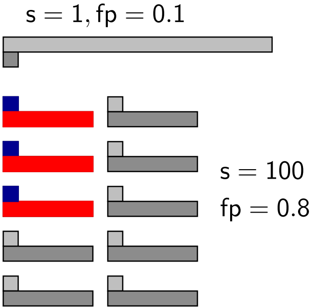
Bi-criteria - Interval Mapping

Minimize $\mathcal{FP}$ with fixed latency

Communication homogeneous - Failure heterogeneous



$30 + 101/100 > 22$

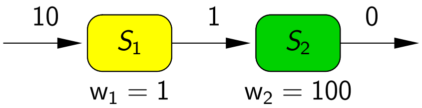


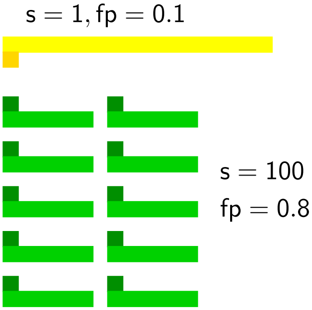
Bi-criteria - Interval Mapping

Minimize $\mathcal{FP}$ with fixed latency

Communication homogeneous - Failure heterogeneous

Fixed latency: 22





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

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

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.

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).

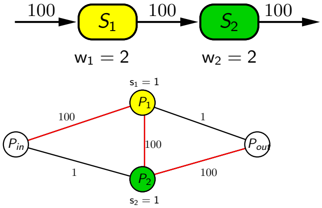
Map whole pipeline on fastest processor.

Mono-criterion Problems

Minimize the latency?

What about *Fully Heterogeneous* platforms?

Remember example:



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

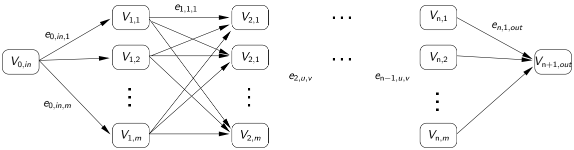
Minimizing the latency is polynomial on *Fully Heterogeneous* platforms for general mappings.

Mono-criterion Problems

But ... considering general mappings ...

Theorem 4

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



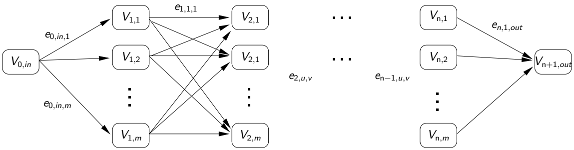
Optimal mapping: Shortest path in the graph.

Mono-criterion Problems

But ... considering general mappings ...

Theorem 4

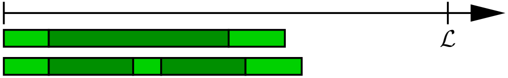
Minimizing the latency is polynomial on *Fully Heterogeneous* platforms for general mappings.



Optimal mapping: Shortest path in the graph.

Interval mapping: still an open problem

Bi-criteria Problems

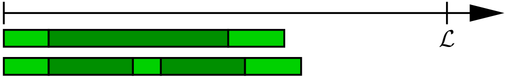


$$1 - (1 - \mathrm{fp}^{a+b}) \leq 1 - ((1 - \mathrm{fp}^{a})(1 - \mathrm{fp}^{b}))$$

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

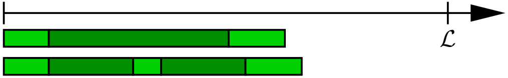


$$1 - (1 - \text{fp}^{a+b}) \leq 1 - ((1 - \text{fp}^{a})(1 - \text{fp}^{b}))$$

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



$$1 - (1 - \text{fp}^{a+b}) \leq 1 - ((1 - \text{fp}^{a})(1 - \text{fp}^{b}))$$

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

Algorithm 1

begin

Find k maximum, such that

$$k \times \frac{\delta_0}{\mathrm{b}} + \frac{\sum_{1 \le j \le n} \mathrm{w}_j}{\mathrm{s}} + \frac{\delta_n}{\mathrm{b}} \le \mathcal{L}$$

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

end

Fully Homogeneous platforms

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

Algorithm 1

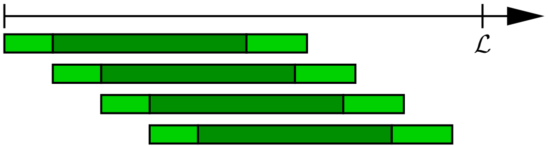
begin

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 $\mathcal{FP}$

Algorithm 2

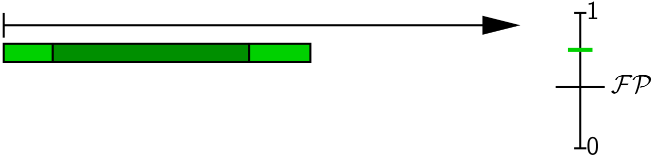
begin

Find k minimum, such that

$$1 - (1 - \text{fp}^k) \leq \mathcal{FP}$$

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 $\mathcal{FP}$

Algorithm 2

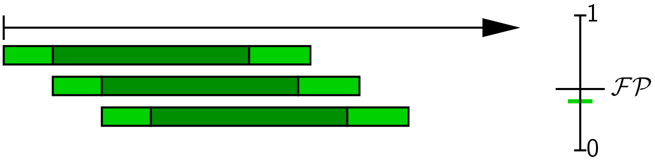
begin

Find k minimum, such that

$$1 - (1 - \text{fp}^k) \leq \mathcal{FP}$$

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

end



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.

Outline

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

		Hom.	Com. Hom.	Hetero.
Mono-crit.	$\mathcal{L}$	polyn.	polyn.	?
	$\mathcal{FP}$	polyn.	polyn.	polyn.
Bi-crit.	$\mathcal{L}$ - $\mathcal{FP}$ hom	polyn.	polyn.	NP
	$\mathcal{L}$ - $\mathcal{FP}$ het	polyn.	?	NP

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

Conclusion

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

Interval Mapping

		Hom.	Com. Hom.	Hetero.
Mono-crit.	$\mathcal{L}$	polyn.	polyn.	?
	$\mathcal{FP}$	polyn.	polyn.	polyn.
Bi-crit.	$\mathcal{L}$ - $\mathcal{FP}$ hom	polyn.	polyn.	NP
	$\mathcal{L}$ - $\mathcal{FP}$ het	polyn.	?	NP

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