Throughput optimization for micro-factories subject to failures

DOBRILA Alexandru

Phd Student, Besançon, Université de Franche-Comté

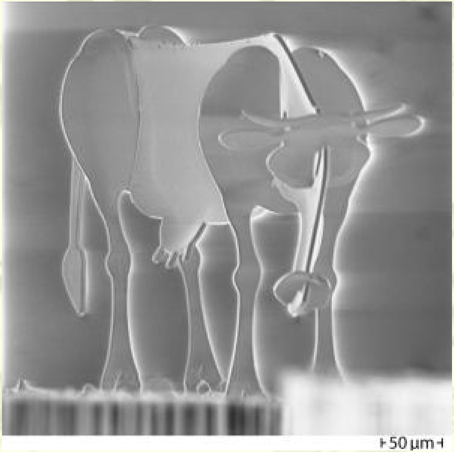
Scheduling Workshop in Knoxville - May 13 2009

Summary

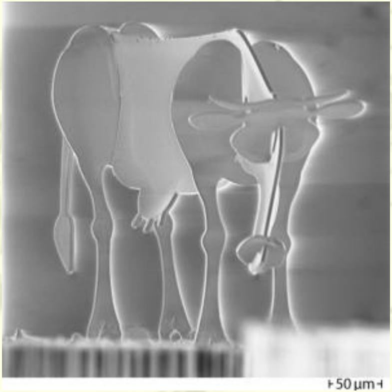
Introduction

- Mapping several tasks onto a set of machines
- Failure attached to tasks not to machines
- A study case of the micro-factory

The micro-factory



The micro-factory



- Pieces composed of micro-metric elements
- Use of dynamic modules of production
- Still in laboratories (mechanical aspects, elementary actuators as piezo-electric beams ...)
- Nothing on scheduling
- Particular DAG (in-tree)

Summary

Application

- a set $\mathcal{N}$ of n tasks: $\mathcal{N} = \{T_1, T_2, \ldots, T_n\}$
- a set $\mathcal{T}$ of p task types with $n \geq p$ and a function $t : [1..n] \rightarrow \mathcal{T}$
- in-tree

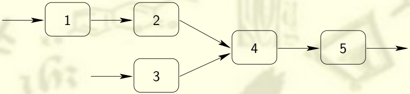


Figure: Example of application.

Platform

- a set $\mathcal{M}$ of m machines: $\mathcal{M} = \{M_1, M_2, \ldots, M_m\}$
- fully connected graph
- machine M_u can perform the task T_i in a time $w_{i,u}$

Failure model

- Failure attached to the task
- $F_i = \frac{a_i}{b_i}$
- $r_i = b_i - a_i$ is the number of successful products
- b_i is called the period of the task
- two tasks of the same type fails with the same rate
  $\forall i, i' \in [1, n] \quad t(i) = t(i') \ \Rightarrow \ f_i = f_{i'}$

Example

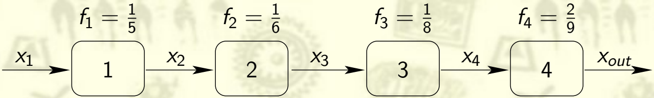


Figure: Example of a linear chain application with failures.

$$x_i = x_{i+1} + a_i \times \left\lceil \frac{x_{i+1}}{r_i} \right\rceil$$

Objective functions

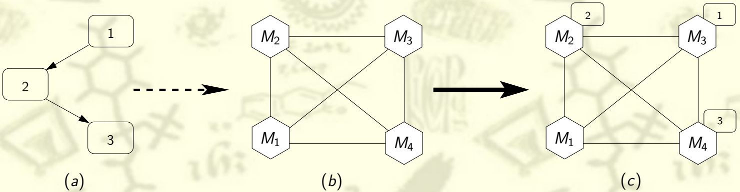
- find an allocation function $a : [1..n] \rightarrow [1..m]$
- possible objectives : reliability, throughput ...
- Period : time between the output of two products
- Average number of products : $\overline{x_i} = \frac{b_i}{r_i} \times \overline{x_{i+1}}$
- $$period(M_u) = \sum_{a(i)=u} \overline{x_i} w_{i,u}$$

Objective functions

- find an allocation function $a : [1..n] \rightarrow [1..m]$
- possible objectives : reliability, throughput ...
- Period : time between the output of two products
- Average number of products : $\overline{x_i} = \frac{b_i}{r_i} \times \overline{x_{i+1}}$
- $$period(M_u) = \sum_{a(i)=u} \overline{x_i} w_{i,u}$$

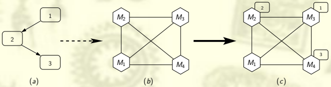
Rules of the game

- One-to-one mapping

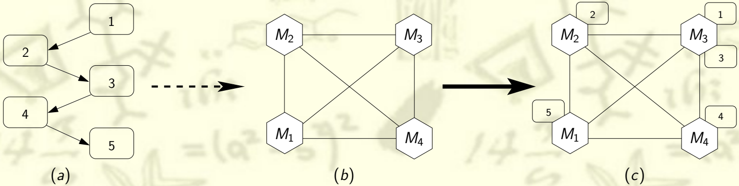


Rules of the game

- One-to-one mapping



- Specialized mapping



t(1)=t(3)=t(5)=1 and t(2)=t(4)=2

Summary

Heuristics H1 and H2

H1 - Random assignment

- One constraint : respect the specialized mapping

H2 - Task group heuristic

- Uses all possible machines
- Create p groups of tasks, putting all tasks of the same type in the same group
- while $m > p$, split the biggest group in two and distribute the load to another machine

H3, H4 and H5 - Binary search heuristics

H3 - Potential optimization

- Assign to the machine a set of tasks that it is efficient for.
- Make the best use of each machine

H4 - Fastest machine

- For a given task, we choose the fastest machine available

H5 - Heterogeneity level

- Sort the machines by their heterogeneity level
- Assign first the more heterogeneous ones

Summary

Configuration

- m is the number of machines
- p the number of types
- n the number of tasks
- average value of 50 simulations where the $w_{i,u}$ randomly chosen between 100 and 1000 ms,
- failure rates f_i $(1 \leq i \leq n)$ randomly chosen between 0.5 and 2 % (i.e., 1/200 and 1/50)

Configuration

- m is the number of machines
- p the number of types
- n the number of tasks
- average value of 50 simulations where the $w_{i,u}$ randomly chosen between 100 and 1000 ms,
- failure rates f_i $(1 \leq i \leq n)$ randomly chosen between 0.5 and 2 % (i.e., 1/200 and 1/50)

m and p fixed - Behavior of H1

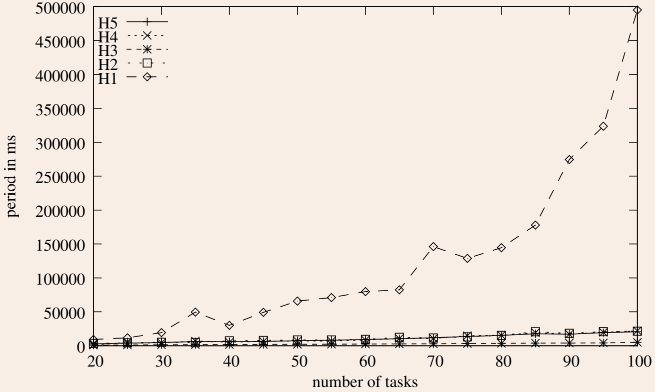


Figure: $m = 10,\ p = 5.$

m and p fixed - Platform heterogeneity

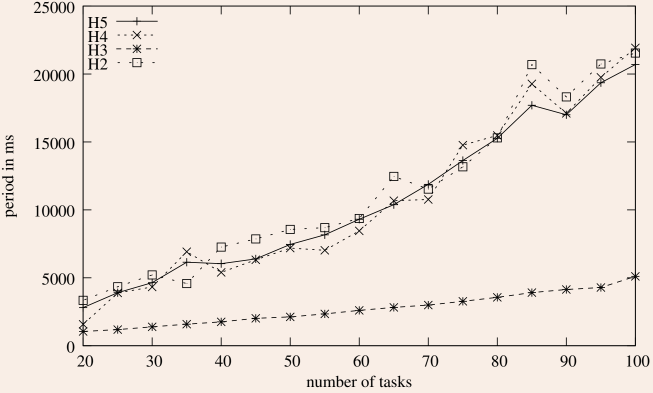


Figure: $m = 10$, $p = 5$. $100 < w_{i,u} < 1000$.

m and p fixed - Platform heterogeneity

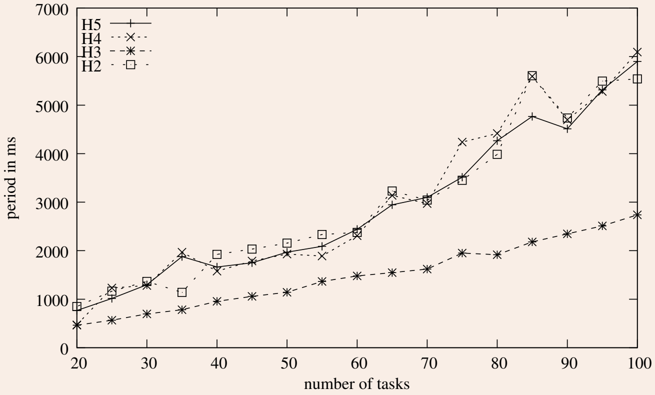


Figure: $m = 10$, $p = 5$. $100 < w_{i,u} < 200$.

m and p fixed - Platform heterogeneity

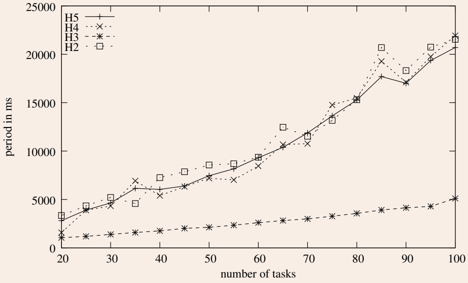


Figure: $m = 10,\ p = 5.\ 100 < w_{i,u} < 1000$

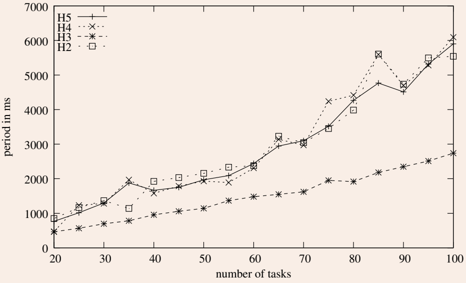


Figure: $m = 10,\ p = 5.\ 100 < w_{i,u} < 200.$

m and *p* fixed - Size of groups

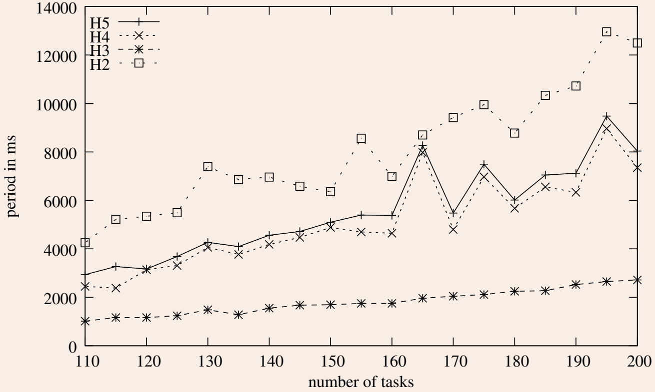


Figure: $m = 100$, $p = 90$.

m and p fixed - Size of groups

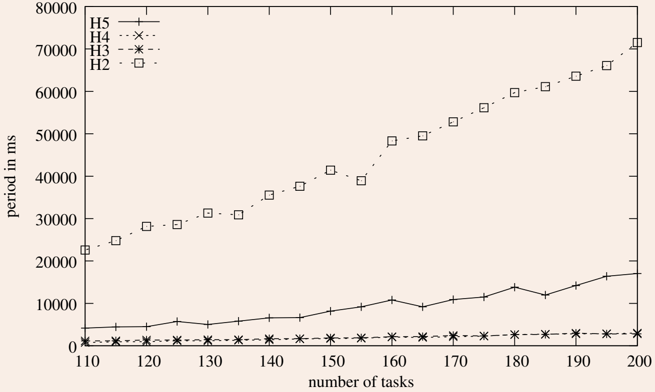


Figure: $m = 100$, $p = 5$.

m and n fixed

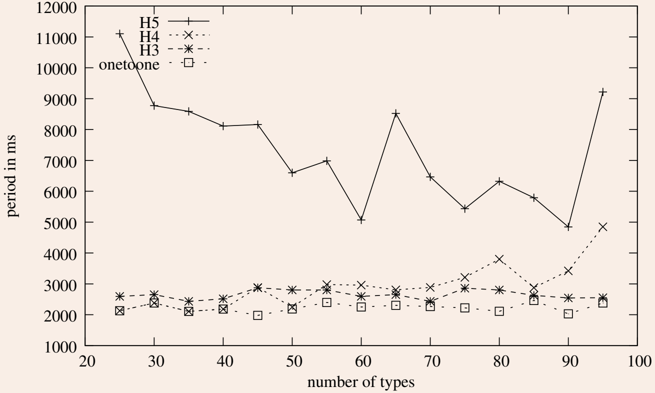


Figure: $m = n = 100$, with $w_{i,u} = w_{i,u'}$.

Summary

Future works

- $f_i \longrightarrow f_{i,u}$
- A task could be executed by different machines
- Consider the *general* mapping, with reconfiguration cost

Questions ...