Global Scheduling on Heterogenous MPSoCs

Emmanuel Grolleau
grolleau@ensma.fr
Based on a work done with:
Antoine Bertout, Joël Goossens, Xavier Poczekajlo, Roy Jamil
Classification of mp platforms

- Identical
- Uniform
- Consistent
- Unrelated
Important results

- Seminal paper in operations research scheduling Lawler & Labetoulle 1978
- Seminal paper in r-t scheduling of S. Baruah 2004
  
  Input: (strictly) periodic independent synchronous implicit deadline tasks, defined by $u_i = C_i/T_i$, $C_i$ defined on a fictional reference core, and for each core $\Pi_j$, a rate $r_{ij}$

- Linear Program: what fraction of core to what task?
- Theorem: the system is feasible if and only if the LP has a solution
Linear Programming problem

Workload assignment

<table>
<thead>
<tr>
<th>Task</th>
<th>Workload</th>
<th>( \Pi_1 )</th>
<th>( \Pi_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_1 )</td>
<td>( x_{11} \times 1 )</td>
<td>( x_{12} \times 2 )</td>
<td></td>
</tr>
<tr>
<td>( \tau_2 )</td>
<td>( 0.5 \times x_{21} )</td>
<td>( x_{22} \times 1 )</td>
<td></td>
</tr>
<tr>
<td>( \tau_3 )</td>
<td>( 2 \times x_{32} )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ x_{21} \times 0.5 \leq 1 \]

\[ \forall \text{core } j, \sum_{\forall \text{task } i} x_{ij} \leq 1 \]

\[ \forall \text{task } i, \sum_{\forall \text{core } j} x_{ij} r_{ij} = u_i \]

\[ \forall \text{task } i, \sum_{\forall \text{core } j} x_{ij} \leq 1 \]
From workload assignment to feasible schedule

This is a Doubly Stochastic (DS) matrix!
- A DS matrix
  - Square, non-negative values, sum of each row and column is 1
- A DS matrix can be expressed as a convex combination of permutation matrices

<table>
<thead>
<tr>
<th>0.3</th>
<th>0.4</th>
<th>0.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
<td>0.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>0.3</th>
<th>0.4</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.4</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>0.2</td>
<td>0</td>
<td>0.7</td>
</tr>
</tbody>
</table>

Note that $1/0.9$ is DS
Convex combination of permutation matrices

\[
\begin{bmatrix}
0.3 & 0.4 & 0.3 \\
0.5 & 0.5 & 0 \\
0.2 & 0.1 & 0.7 \\
\end{bmatrix}
= 0.1 \times
\begin{bmatrix}
0 & 0 & 1 \\
1 & 0 & 0 \\
0 & 1 & 0 \\
\end{bmatrix}
+ 0.2 \times
\begin{bmatrix}
0 & 0 & 1 \\
0 & 1 & 0 \\
1 & 0 & 0 \\
\end{bmatrix}
+ 0.3 \times
\begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{bmatrix}
= 0.3 \times
\begin{bmatrix}
0.3 & 0.4 & 0.2 \\
0.4 & 0.5 & 0 \\
0.2 & 0 & 0.7 \\
\end{bmatrix}
+ 0.4 \times
\begin{bmatrix}
0.3 & 0.4 & 0 \\
0.4 & 0.3 & 0 \\
0 & 0 & 0.7 \\
\end{bmatrix}
+ 0.4 \times
\begin{bmatrix}
0 & 0.4 & 0 \\
0.4 & 0 & 0 \\
0 & 0 & 0.4 \\
\end{bmatrix}
\]
As a real-time scheduling guy, I can interpret it as a schedule...

Each permutation matrix generates a scheduling point
- Preemption and/or migration
From template schedule to schedule

- A template schedule can be repeated on each time unit
- Or stretched between each deadline
- Mirrored every other time
Discussion about the template schedule

- Can always be obtained from a DS matrix (BvN decomposition theorem)
- Obtaining a valid workload assignment matrix is a necessary and sufficient schedulability condition
  - Under the hypothesis of no preemption cost, no migration cost
- The produced off-line schedule supposes an « almost fluid » scheduler, able to preempt/migrate tasks several times per time unit
- Number of permutation matrices = number of scheduling points per template schedule
Minimizing the number of permutation matrices is NP-hard in the strong sense (from 3-Partition in [Dufossé 2015])

Heuristic to reduce migrations/preemptions

- In the previous example, Birkhoff method (minimum non-null value)
- Maximize locally the duration of each assignment?

✓ Solving a Linear Bottleneck Assignment Problem at each step
✓ Polynomial time using Hungarian method
### S. Baruah’s strategy

<table>
<thead>
<tr>
<th>Workload assignment</th>
<th>$\Pi_1$</th>
<th>$\Pi_2$</th>
<th>$\text{Latency}_1$</th>
<th>$\text{Latency}_2$</th>
<th>$\text{Latency}_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>$x_{11}$</td>
<td>$x_{12}$</td>
<td>$l_1 = 1 - (x_{11} + x_{12})$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>$x_{21}$</td>
<td>$x_{22}$</td>
<td></td>
<td>$l_2 = 1 - (x_{21} + x_{22})$</td>
<td></td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>$x_{32}$</td>
<td>$x_{11}$</td>
<td></td>
<td></td>
<td>$l_3 = 1 - (x_{31} + x_{32})$</td>
</tr>
<tr>
<td>Idle$_1$</td>
<td>$i_1$</td>
<td>$x_{11}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idle$_2$</td>
<td>$x_{12}$</td>
<td>$x_{21}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\Sigma = 1$ for all idle states.

Latency equations:

- $l_1 = 1 - (x_{11} + x_{12})$
- $l_2 = 1 - (x_{21} + x_{22})$
- $l_3 = 1 - (x_{31} + x_{32})$
An "as conservative" template schedule as possible

- Square node = urgent task or full processor, assign absolutely
- Circle node = non urgent, non full, assign if necessary

![Diagram](image)

Arbitrary marriage
Reverse construction of template schedule

<table>
<thead>
<tr>
<th>Workload assignment</th>
<th>$\Pi_1$</th>
<th>$\Pi_2$</th>
<th>$\text{Latency}_1$</th>
<th>$\text{Latency}_2$</th>
<th>$\text{Latency}_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>$x_{11}$</td>
<td>$x_{12}$</td>
<td>$l_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>$x_{21}$</td>
<td>$x_{22}$</td>
<td></td>
<td>$l_2$</td>
<td></td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>$x_{32}$</td>
<td></td>
<td></td>
<td></td>
<td>$l_3$</td>
</tr>
<tr>
<td>Idle$_1$</td>
<td></td>
<td></td>
<td>$i_1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idle$_2$</td>
<td></td>
<td></td>
<td></td>
<td>$x_{11}$</td>
<td>$x_{21}$</td>
</tr>
</tbody>
</table>

Smallest non null value

To do
Reverse construction of template schedule / contd

- $\Pi_1$ is now full, like $\Pi_2$, and they will remain full until the beginning of the template schedule.
- $\tau_1$ is now urgent, and will remain urgent until the beginning.
Marriage problem

- Find a marriage such that each square node is married
- Circle nodes are "spare nodes"

- Could we invent a new "marriage in the nobility" problem?
Baruah’s method

\[
\begin{align*}
\tau_1 & \quad \Pi_1 \\
\tau_2 & \quad \Pi_2 \\
\tau_3 & \quad \Pi_1 \\
\tau_2 & \quad \Pi_2 \\
\tau_3 & \quad \Pi_2
\end{align*}
\]

\[
\begin{align*}
\tau_1 & \quad \Pi_1 \\
\tau_2 & \quad \Pi_2 \\
\tau_3 & \quad \Pi_2
\end{align*}
\]

\[
\begin{align*}
\tau_1 & \quad \Pi_1 \\
\tau_2 & \quad \Pi_2 \\
\tau_3 & \quad \Pi_2
\end{align*}
\]

\[
\begin{align*}
\tau_1 & \quad \Pi_1 \\
\tau_2 & \quad \Pi_2 \\
\tau_3 & \quad \Pi_2
\end{align*}
\]

\[
\begin{align*}
\tau_1 & \quad \Pi_1 \\
\tau_2 & \quad \Pi_2 \\
\tau_3 & \quad \Pi_2
\end{align*}
\]
Cleaning method

- Only important nodes can be the endpoints of two edges.
- An non cyclic even length path has a non important node as one of its extermetiies.
- A cycle can only have an even length path.

- Number any path starting from an important node.
- Cut the edges with an even number.

Diagram:

- Nodes A, B, C, D, E, F.
- Edges connecting nodes with numbers 1 to 5.
- Nodes A and B are connected with an edge.
- Nodes A and D are connected with a red cross.
- Nodes D and C are connected with a red cross.
### Workload assignment

<table>
<thead>
<tr>
<th>Workload assignment</th>
<th>$\Pi_1$</th>
<th>$\Pi_2$</th>
<th>Latency$_1$</th>
<th>Latency$_2$</th>
<th>Latency$_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau_1$</td>
<td>$x_{11}$</td>
<td>$x_{12}$</td>
<td>$l_1$</td>
<td>$l_2$</td>
<td>$l_3$</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>$x_{21}$</td>
<td>$x_{22}$</td>
<td>$x_{32}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>$i_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idle$_1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idle$_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Diagram

- $\tau_1$ to $\Pi_1$
- $\tau_3$ to $\Pi_2$
- $\tau_2$ to $\Pi_1$

### Latency

- $l_1$, $l_2$, $l_3$
- $x_{11}$, $x_{21}$, $x_{22}$, $x_{32}$
- $i_1$

### To do

- $\Pi_1$
- $\Pi_2$

CAPITAL Workshop, 06/04/21
Tasks + rates on cores

Workload assignment

Template schedule

Tasks + rates on cores

- $u_1$
- $u_2$
- $u_3$

Rates on cores:

- $r_{i1}$
- $r_{i2}$

Workload assignment:

<table>
<thead>
<tr>
<th>$\tau_1$</th>
<th>$\Pi_1$</th>
<th>$\Pi_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x_{11}$</td>
<td>$x_{12}$</td>
</tr>
<tr>
<td>$\tau_2$</td>
<td>$x_{21}$</td>
<td>$x_{22}$</td>
</tr>
<tr>
<td>$\tau_3$</td>
<td>$x_{31}$</td>
<td>$x_{32}$</td>
</tr>
</tbody>
</table>

Template schedule:

BvN
Remarks on the LP

- The objective function can be any
  \[ \text{LP Load Obj: min } \sum_j \sum_i x_{ij} \]

- Room left for e.g. energy or heat dissipation optimization

- Constraints can be added

  - \[ \forall \text{task } i, \forall \text{core } j, b_{ij} \in \{0,1\} \]
  - \[ b_{ij} \text{ is } 1 \text{iff } \tau_i \text{ uses } \Pi_j, 0 \text{ else} \]
  - \[ \text{ILP Mig Obj: min } \sum_j \sum_i b_{ij} \]
### LP Feas vs. LP Load

<table>
<thead>
<tr>
<th>C_i</th>
<th>T_i</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ₁</td>
<td>5</td>
</tr>
<tr>
<td>τ₂</td>
<td>5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rates</th>
<th>π₁</th>
<th>π₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>τ₁</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>τ₂</td>
<td>10</td>
<td>1</td>
</tr>
</tbody>
</table>

**LP Feas**

\[
\begin{bmatrix}
\frac{5}{11} & \frac{5}{11} \\
\frac{5}{11} & \frac{5}{11}
\end{bmatrix}
\]

**LP Load**

\[
\begin{bmatrix}
0,5 & 0,5 \\
0 & 0
\end{bmatrix}
\]
BvN, LBAP or Conservative decomposition?

- Metrics: number of migrations & preemptions
- And (experimentally) the winner is...

- The conservative decomposition

- Traveler salesman problem
Heterogeneous MPSoCs: a growing trend

- **TI Sitara AM57x**
  - Embedded computing, robotics, avionics, medical imaging, etc.

- **Samsung Exynos 9 9820**
  - Smartphone

- **NXP i.MX 8 QuadMax**
  - Automotive, etc.
Clustered platform

Rather than having a heterogeneous platform

Consider a set of clusters of identical cores
A system is feasible iff LP has a solution \( \Rightarrow \) always possible to build a DS matrix

- Less variables (rate \( r_{ij} \) per cluster)
  - ILP for inter-cluster migrations minimization smaller

- Inter-cluster \( \neq \) Intra-cluster migration
  - Experimentally 10 to 70\( \mu \)s vs. 1 to 2 \( \mu \)s on i.mx8 and STM32MP1
Comparison flat vs. clustered

Average number of clustered workload assignments on 2 clusters

Average number of clustered workload assignments on 5 clusters

Legend:
- LP-Feas
- LP-Load
- LP-CFeas
- LP-CLoad
- ILP-Cmig
- Hetero-split
### Performance gain

#### Average execution time (in seconds)

<table>
<thead>
<tr>
<th></th>
<th>2 clusters</th>
<th>5 clusters</th>
</tr>
</thead>
<tbody>
<tr>
<td>LP-Feas</td>
<td>0.013</td>
<td>0.464</td>
</tr>
<tr>
<td>LP-Load</td>
<td>0.012</td>
<td>0.562</td>
</tr>
<tr>
<td>LP-CFeas</td>
<td>0.002</td>
<td>0.027</td>
</tr>
<tr>
<td>LP-CLoad</td>
<td>0.002</td>
<td>0.029</td>
</tr>
<tr>
<td>Hetero-split</td>
<td>0.007</td>
<td>N/A</td>
</tr>
<tr>
<td>ILP-Mig</td>
<td>0.061</td>
<td>N/A</td>
</tr>
<tr>
<td>ILP-CMig</td>
<td>0.023</td>
<td>0.156</td>
</tr>
</tbody>
</table>
How (un)realistic is the model?

- Zero cost for preemption & migration => already NP-hard for uniprocessor
- « If any portion of a thread is executed for 5% of the time on a core of rate 2, it executes for 10% »
  - What if the first half of a thread uses intensively integers, and the second half uses intensively floats?

- Limited to implicit deadlines, strictly periodic tasks
  - Sporadic tasks with explicit deadlines?
- Offline schedule hard to implement
  - Dynamic schedule à la U-EDF?
Conclusion

- Are used and will be more in the future
- Energy saving possibilities
  - DVFS, DPM
- Global scheduling
  - Can use up to 100% of the platform
  - Can be seen as saving more energy in the future
- Lots to do...

CAPITAL Workshop, 06/04/21