Constrained Global Scheduling of Streaming Applications on MPSoCs

Jun Zhu, Ingo Sander, and Axel Jantsch

{junz,ingo,axel}@kth.se

Royal Institute of Technology (KTH), Stockholm, Sweden

ASP-DAC '10. January 20, 2010

Synchronous data flow (SDF) preliminaries

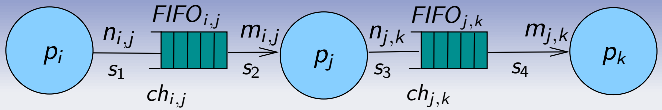


Figure: An example streaming application in SDF model.

- Process, channel, FIFO, and the "synchronous" name
- Computation $T = [t_{C,i}, t_{C,j}, t_{C,k}]$ and storage $\Gamma = [\gamma_{i,j}, \gamma_{j,k}]$
- Consistency, e.g., $r_i \cdot n_{i,j} = r_j \cdot m_{i,j}$ in $ch_{i,j}$; avoid auto-concurrency

Architecture model

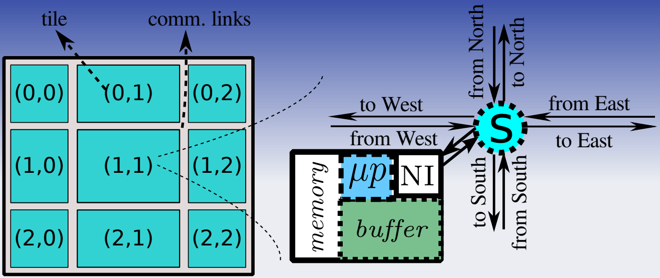


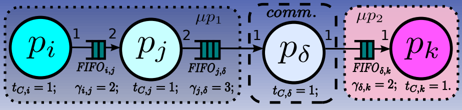
Figure: A 3×3 regular 2-D mesh MPSoC architecture.

- Each tile consists of a processor (μp), an application *memory*, a token *buffer*, and a network interface (NI).
- Hard real-time NoC infrastructure: switches (s) and links.

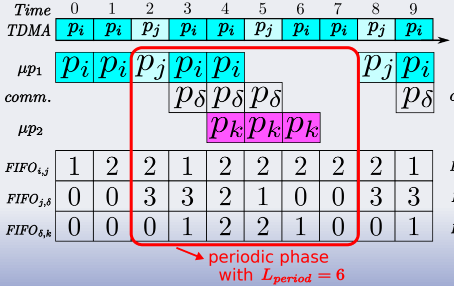
Problem statement: the analysis of routing, scheduling, and buffer properties, based a **fixed** allocation from processes to tiles.

Motivation – differnt scheduling schemes

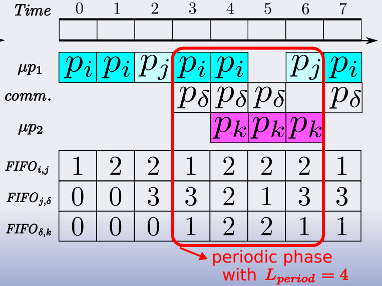
(0) Application allocated onto buffer constrained dual-processor.



(1) TDM scheduling scheme.



(2) Throughput optimized scheduling.



- OH in TDM-like schemes causes sub-optimal: throughput vs. buffer
- Contributions: **compu. & comm.** global scheduling for MPSoCs

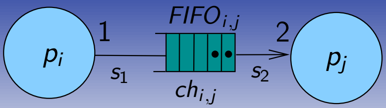
Outline

1. Related Work
2. Constraint based Analysis
3. Experimental Results
4. Conclusion

Related Work

- **Heuristic** scheduling of **task graphes** on distributed systems with comm. protocols optimization [Eles et al., TVLSI '00];
- For SDF models, to construct heuristic periodic admissible parallel schedules (PAPS) on multi-processors [Lee and Messerschmitt, 1987a], **without** constraints on buffers and throughput.
- Stuijk et al. [DAC '07] propose a mapping and **TDMA/list** scheduling design flow for throughput constrained SDF applications on MPSoCs, which is argued about the OH and lacking of global optimization on performance metrics.
- Inspiring work: model-checker determines the minimal deadlock-free buffer of SDF models (**no computation constraints**) [Geilen et al., DAC '05]; Liu et al. [RTSS '08] use SAT-solver to explore the mapping and scheduling of homogeneous SDF (**HSDF**) models on multi-processors to **maximize** application throughput.

Event Models [Zhu et al., DATE '09]



time tag	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
p_i	1	1	0								1	1	0		
p_j	4	3	2	1	0						4	3	2	1	0
$FIFO_{i,j}$	3	4	4	4	2	2	2	2	2	2	3	4	4	4	2

Cumulative functions [Cruz1995]:

Arrival function $R_{i,j}(t) = \sum_0^t s_1$

Service function $C_{i,j}(t) = \sum_0^t s_2$

Output function $R'_{i,j}(t) = R_{i,j}(t)$

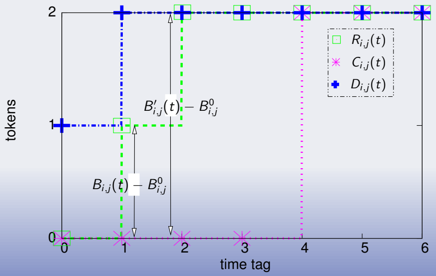
Demand function

$$D_{i,j}(t) = \begin{cases} \sum_0^t s_1 + n_{i,j}, & p_i \ computing \\ \sum_0^t s_1, & p_i \ stalling \end{cases}$$

Buffer Properties:

Buffer backlog $B_{i,j}(t)$

Buffer requirement $B'_{i,j}(t)$



J. Zhu, I. Sander, and A. Jantsch.
Buffer minimization of real-time streaming applications scheduling on hybrid CPU/FPGA architectures.
DATE '09.

Constraint based analysis

Our declarative scheduling method considers

- Execution semantics of SDF models [Zhu et al., DATE '09]: e.g., static token rates, asynchronous buffer
- Resource constraints: binding applications onto target MPSoC platforms
  - application to architecture mapping (be aware)
  - computation scheduling
  - communication routing and flow control
  - performance metrics, e.g., buffer cost and real-time constraints

Mapping

Mapping decision $\alpha_{i,\mu p_n}$: denotes the presence of p_i on μp_n

Constraint (Single residence)

$$\sum_{\mu p_n \in \mathbf{U}} \alpha_{i,\mu p_n} = 1, \quad \forall p_i \in \mathbf{P} \tag{1}$$

Although mapping is fixed, scheduling needs to be mapping-decision aware

Constraint (Mapping & scheduling association)

Sequential execution of processes on each processor, and the mapping and scheduling association:

$$\sum_{p_i \in \mathbf{P}} \alpha_{i,\mu p_n} W_j(t) \in \{0,1\}, \ \forall \mu p_n \in \mathbf{U}, t \in \mathbb{N}_0 \tag{2}$$

in which $W_j(t)$ denotes the **1-0** *(computing) status of p_j.*

$$W_j(t) = max(L_j(t), L_j(t+\Delta_t)), \quad \forall \Delta_t \in [1, t_{C,j}] \tag{3}$$

$$L_j(t+1) = \frac{C_j(t+1) - C_j(t)}{m_{i,j}} \in \{0,1\}$$

Template based Buffering Analysis - 1

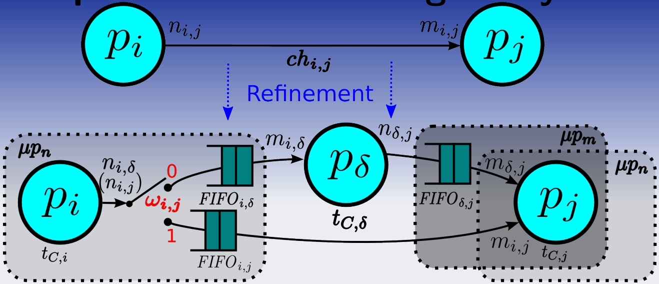


Figure: A template of producer-consumer refinement. $\omega_{i,j}$ (correlated with α) denotes whether $ch_{i,j}$ is as intra-processor channel.

Some constaints on FIFOs can be formalized, e.g.,

Property (Buffer usage)

The FIFO usages of $FIFO_{i,\delta}$ at time tag t is denoted as $B_{i,\delta}(t)$:

$$B_{i,\delta}(t)=D_{i,\delta}(t)-\neg\omega_{i,j}(C_{i,\delta}(t)-B^0_{i,\delta})-\omega_{i,j}(C_{i,j}(t)-B^0_{i,j}) \quad (4)$$

in which $B^0_{i,\delta}(t)$ is the initial data tokens.

Template based Buffering Analysis - 2

Different FIFOs on the same tile can be implemented as either disjoint or shared buffer partitions.

Property (Buffer cost $\gamma_{Sum'}$ – disjoint partition)

$$\gamma_{Sum'} = \sum_{i,j,\delta} (\omega_{i,j}\gamma_{i,j} + \neg\omega_{i,j}(\gamma_{i,\delta} + \gamma_{\delta,j})),\ \forall p_i, p_j \in \mathbf{P} \quad (5)$$

$$\text{where} \quad \gamma_{i,j} \geqslant B_{i,j}(t), \gamma_{i,\delta} \geqslant B_{i,\delta}(t), \gamma_{\delta,j} \geqslant B_{\delta,j}(t),\ \forall t \in \mathbb{N}_0$$

in which $\gamma_{i,j}$, $\gamma_{i,\delta}$, and $\gamma_{\delta,j}$ denote the sizes of FIFOs in the template.

Property (Buffer cost $\gamma_{Sum''}$ – shared partition)

$\gamma_{Sum''}$ is the sum of the shared buffer space on individual tiles.

$$\gamma_{Sum''} = \sum_{n} \gamma_{tile_n} \quad (6)$$

$$\text{where } \gamma_{tile,n} \geqslant \sum_{i,j} (a_{i,\mu p_n} B_{i,\delta}(t) + a_{j,\mu p_n} B_{\delta,j}(t) + a_{i,\mu p_n} \omega_{i,j} B_{i,j}(t))$$

The total buffer cost $\gamma_{Sum'} \geq \gamma_{Sum''}$.

Routing and flow control - 1

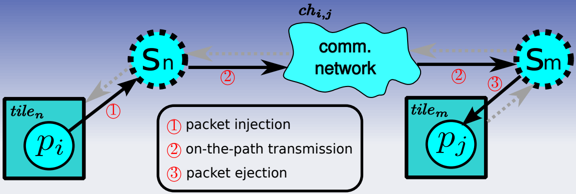


Figure: Three phases to route a packet in inter-tile channel $ch_{i,j}$.

- Assuming deterministic X-Y routing on NoC, routing decisions are in β^r and β^c on **rows** and **columns** with conrresponding constraints.
- In hard-real time NoC (contention-free), a packet (data token) reserves a circuit switching before the transmission finishes, e.g., in Æthereal [Goossens et al., '05 IEEE Des. Test].

Routing and flow control - 2

- Different application flows sharing the same NoC links are scheduled temporally to avoid contention in: packet injection (see Eq. 7), ejection, and inter-tile traffic flow (for rows of links see Eq. 8)

Constraint (Contention-free traffic flow scheduling)

$$\sum_{i} \alpha_{i,\mu p_n} W_\delta(t) \in \{0,1\}, \tag{7}$$

$$\sum_{i,j} \beta^r_{ch_{i,j},l_k} W_\delta(t) \in \{0,\pm 1\}, \sum_{i,j} |\beta^r_{ch_{i,j},l_k}| W_\delta(t) \in \{0,1,2\}, \tag{8}$$

$$\forall p_i, p_j \in \mathbf{P}, \mu p_n \in \mathbf{U}, t \in \mathbb{N}_0$$

with $W_\delta(t)$ to denote the data transmission **1-0** *(working or idle) status on NoC modeled by p_δ.*

Real-time constraints

The efficient execution means the streaming services are delivered on-demand to the end-user, neither too fast nor too slow.

Constraint

(Application output throughput) After some start-up time period τ_0 ($\tau_0 > 0$) with no stable output tokens, a specified output throughput ρ_{out} should be sustained at the application sink process p_j.

$$C_j(\tau_0 + k \cdot L_{period}) - C_j(\tau_0) = k \cdot J \cdot r_j \cdot m_{i,j}, \ \forall k \in \mathbb{N}_0 \qquad (9)$$

Empirically, $L_{period} = \lceil \frac{J \cdot r_k \cdot m_{i,j}}{\rho_{out}} \rceil, J \in \mathbb{N} \backslash \{\infty\}$, in which J is incremental and leads to an valid L_{period}.

Furthermore, throughput is guaranteed by periodic phase checking.

Constraint programming techniques

- CP (**Gecode**) has been successful to solve NP-complete problems.
- Problems are defined by: variables, domains, and constraints.
- Some efficient modeling techniques in solutions finding:
  - **Redundant variables and constraints.** E.g., redundant mapping decision variables.
  - **Domain and constraints reduction.** E.g., lower bounds for FIFOs, constraints are checked in $t \in [0, \tau_0 + L_{period}]$ once periodic phase happens.
  - **Branching and exploration.** To construct the search tree, the branching variables are prioritized as follows: $\gamma_{Sum'}$ ($\gamma_{Sum''}$), γ_{tile_n}, and C. Empirically, the exploration starts with minimal values for all variables

Experimental Results - 1

Table: Comparison of scenarios with varying OH. (3×3 platform).

specification				*SCP-OH*[a]		*SCP*	
app.	#[b]	*thru. req.*	*J*	$\gamma_{Sum'}(\gamma_{Sum''})$	*time*[c]	$\gamma_{Sum'}(\gamma_{Sum''})$	*time*[c]
Modem	2	3.125e-2	1	- [d]	352	98(45)	3.0e3
	2	1.667e-2	1	98(46)	5.0e3	92(41)	1.4e3
Wireless	2	2.5e-2	1	-	422	121(53)	4.7e4
	2	1.7e-2	1	123(49)	6.4e5	120(48)	1.2e3

[a] 50% OH in computation latency, plus 100% OH in communication latency.
[b] The number of application instances mapped onto platform.
[c] The solutions finding time (*ms*), branched and explored with $\gamma_{Sum'}$ and γ.
[d] The problem is infeasible.

Experimental Results - 2

Table: Comparison of scheduling methods (2×2 platform).

specification			PAPS			SCP		
app.	#	thru. req.	J	$\gamma_{Sum'}(\gamma_{Sum''})$	time	J	$\gamma_{Sum'}(\gamma_{Sum''})$	time'(time") [a]
Bipartite	1	1.101e-1	3	510(510)	120	3	36(31)	2.6e3(2.3e5)
	1	1.096e-1	2	352(352)	50	1	36(31)	1.8e3(1.4e5)
	1	1.082e-1	1	194(194)	20	1	36(28)	1.9e3(2.1e5)
Cd2dat	2	2.462e-1	-	-	-	1	68(34)	1.9e3(4.7e4)
	2	1.553e-1	2	2926(1504)	430	1	66(34)	1.8e3(3.3e5)
	2	1.550e-1	1	1472(762)	120	1	66(34)	1.9e3(3.3e5)
H263	2	2.103e-4	-	-	-	1	9512(9506)	9.7e3(2.3e5)
	2	2.102e-4	2	19016(19012)	2.0e5	1	9512(9506)	9.5e3(2.3e5)
	2	2.101e-4	1	9512(9508)	2.4e4	1	9512(9506)	9.2e3(2.1e5)

[a] The solutions finding time (*ms*) branched and explored with two buffer measurements.

Conclusions & future work

Conclusions:

1. We present a constraint based global scheduling framework for SDF streaming application on NoC based MPSoCs.
2. The global scheduling for both processors computing and communication flow control has been achieved.
3. It shows higher predictable application throughput and with minimized buffer cost.

Future work:

1. Caused by the complexity of scheduling, we plan to combine heuristics with our constraint based techniques, e.g., using heuristics to explore the search tree.
2. To explore parallel search with multiple threads in *Gecode* on multi-core workstations.

Thanks for your attention!

Questions?