Worst-case performance analysis for SDF-based dataflow MoCs

Mladen Skelin[1]

[1]Norwegian University of Science and Technology, Norway

May 4th 2015

Outline I

Models of Computation (MoC)[10]

— Models of Computation (MoCs) describe system behaviour by defining
  - System components and an execution model for components and
  - System communication model, i.e. how the information is exchanged between components

Models of Computation (MoC)

- Given a system, the challenge is to find a good MoC for that system
- This is the key to efficient design and analysis

Models of Computation (MoC)

— Time plays a role
— The Von Neumann MoC?
— Thread based concurrency models?

Models of Computation (MoC)

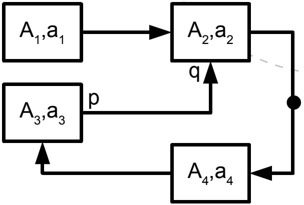
— For control-dominated and reactive systems
  - State-based models are appropriate, monitor inputs, change state and set outputs

— For data-dominated systems
  - Actor-based dataflow models ar appropriate, transform input data streams into output data streams

Outline I

Dataflow MoCs

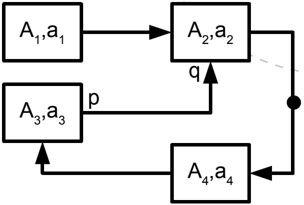
- Instantiated as directed graphs
- Nodes are called *actors*
  - Describe some functionality
- Edges are called *channels*
  - Describe actor dependencies
- Token
  - A quantum of information exchanged




```
fire {
        ...
        token = InPort.Read();
        ...
        OutPort.Write(token);
        ...
}
```

Dataflow MoCs

- Actor firing
  - Quantum of computation
  - Token consumption and production
- Rates
  - Token production and consumption numbers
  - Actors as processes, token sequences as signals
  - Describe signal partitioning
- Actor firing delay
  - Actor execution time




```
fire {
        ...
        token = InPort.Read();
        ...
        OutPort.Write(token);
        ...
}
```

Dataflow Advantages

- Simple graphical representation
- Elegantly expresses parallelism
- An intuitive system representation for signal processing and multimedia designers
- Best software practices: modularity and code reuse
- Software synthesis from dataflow graphs (front end)

Dataflow MoC categories

— Static dataflow MoCs

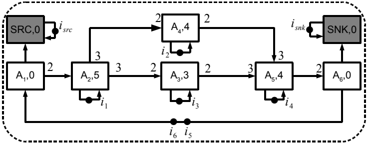
— Dynamic dataflow MoCs

— The former can describe dynamic application, the latter cannot

Outline I

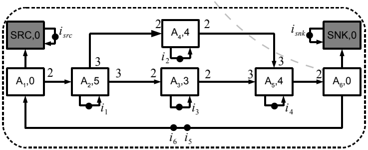
Synchronous dataflow (SDF)

- Introduced by Edward Lee et al., UC Berkley, 1987 [9]
- Most widely used dataflow MoC in general
- Rates and actor firing delays are known at compile time
- Channels as FIFOs



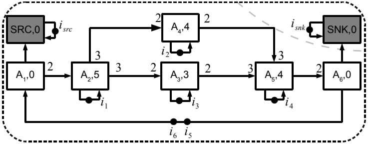
Synchronous dataflow (SDF)

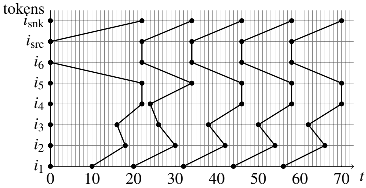
- Initial tokens
  - Typically understood to be an initial condition for an execution, rather than part of the execution
- Progresses in iterations
  - A sequence of actors' firings $\mathscr{I}$ that brings the SDF graph (SDFG) to its initial state
  - Typically a coherent set of calculations, e.g. decoding a video frame
  - Graph repetition vector Γ



Synchronous dataflow (SDF)

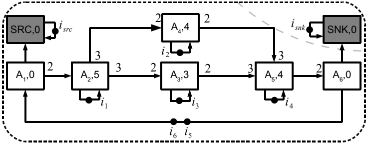
- Graph repetition vector: $\Gamma(SRC, A_2, A_3, A_4, A_5, A_6, SNK) = (1,1,2,3,3,2,1,1)$
- Graph iteration: $\mathscr{I} = A_1^1 SRC_1^1 A_2^2 A_3^3 A_4^3 A_5^2 A_6^1 SNK_2^1$
- Self-timed execution
- Ultimately periodic

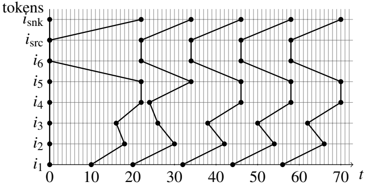




Synchronous dataflow (SDF)

- Advantages of SDF
  - Compile-time predictable/analyzable
  - High implementation efficiency
- Disadvantages of SDF
  - Low expressive power





Outline I

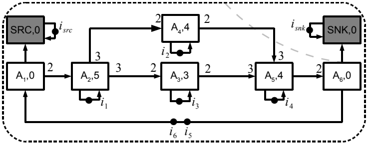
Parametrized dataflow

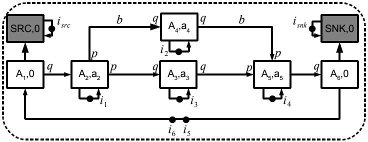
— Integrate dynamic parameters and run-time adaptation of parameters in a structured way into a certain dataflow MoC called the *base model*

— Base models with well-defined concept of a graph iteration, e.g. SDF

— SDF-based parametrized dataflow (SDF-PDF)

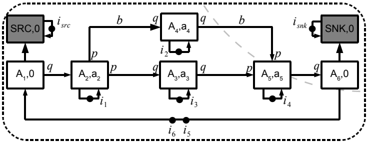
- SDF parametrization



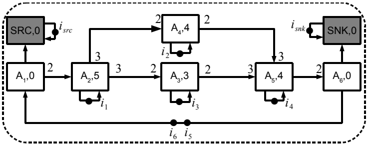


SDF-PDF

- Parameter value become known at run-time, i.e. SDF-PDFG is configured at run-time
- SDF-PDFG configuration c is obtained by assigning values to all parameters of the SDF-PDFG
- When we apply c to the SDF-PDFG, an SDFG emerges
- This SDFG is called the instance of the original SDF-PDFG, denoted $\iota_G(c)$

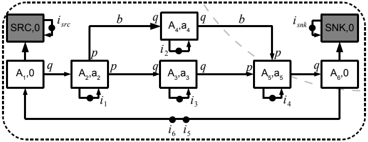


$$\downarrow c = \{p = 3, q = 2, b = \texttt{tt}, a_2 = 5, a_3 = 3, a_4 = 4, a_5 = 4\}$$

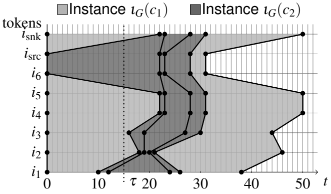


SDF-PDF

- SDF-PDFG domain $X = \{c_k\}, k \in \mathbb{N}_{>0}$
- Graph repetition vector: $\Gamma(SRC, A_2, A_3, A_4, A_5, A_6, SNK) = (1, 1, q, p, p, q, 1, 1)$
- Graph iteration: $\mathscr{I} = A_1^1 SRC_1^1 A_2^p A_3^q A_4^q A_5^p A_6^1 SNK_2^1$
- An SDF-PDF graph (SDF-PDFG) progresses in base model iterations
- Parameter values change between iterations



$\downarrow c_1 = \{p = 3, q = 2, b = \texttt{tt}, a_2 = 5, a_3 = 3, a_4 = 4, a_5 = 4\}, c_2 = \{p = 1, q = 1, b = \texttt{ff}, a_2 = 2, a_3 = 3, a_4 = 1, a_5 = 1\}$



SDF-PDF

- SDF-PDF is a collective name for parametrized dataflow MoCs where SDF serves as the base models and not another MoC
- Parametrized SDF (PSDF) [3], schedulable parametric dataflow (SPDF) [4], boolean parametric dataflow (BPDF) [2] and variable rate dataflow (VRDF) [11] are example of such models

Outline I

Throughput

- The amount of work a system can perform over a period of time
- We consider dataflow models with well-defined concept of a graph iteration
- Throughput Th is the long run-average number of completed iterations per time-unit

Worst-case throughput of an SDF-PDFG

— One iteration of an SDF-PDFG corresponds to one iteration of an arbitrary SDFG instance

Worst-case throughput (SDF-PDFG)

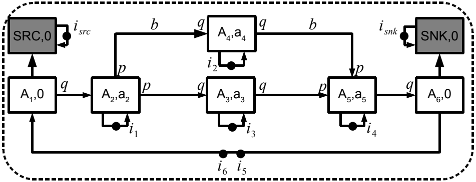
The worst-case throughput of an SDF-PDFG G is given as

$$Th(G) = \min_{\sigma \in \Sigma} \liminf_{k \to \infty} \frac{\sigma_k}{C_k^\sigma}$$

where σ is a sequence of SDFG instances, Σ is the set of all admissible sequences of SDFG instances, σ_k is the first k elements of σ and C_k^σ is the completion time of σ_k.

Latency

— The time delay between the stimulus and its effects
— In dataflow,stimuli and their effects are actor firings
— One is typically interested in the maximal time-span between firings of actors that represent the input and output of the system
— These actors typically have a repetition vector entry of one
— If not, we consider groups of firing of those actors that contain as may firings as their repetition vector entries
— We record these groups using the *SRC* and *SNK* actors



Worst-case latency of an SDF-PDFG

Worst-case latency (SDF-PDFG)

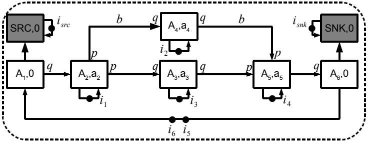
The worst-case latency of an SDF-PDFG G is given as

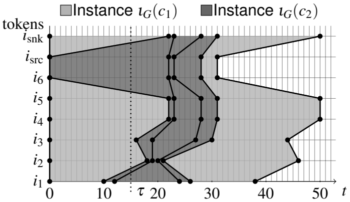
$$L(G) = \max_{\sigma \in \Sigma} \max_{k \in \mathbb{N}_{>0}} \left(F^{\sigma}_{SNK,k} - F^{\sigma}_{SRC,k} \right)$$

where σ is a sequence of SDFG instances, Σ is the set of all admissible sequences of SDFG instances and $F^{\sigma}_{A,k}$ is the completion time of the k-th firing of an SDF-PDFG actor A.

Challenges

- SDF-PDFG actors execute in parallel within a graph iteration;
- SDF-PDFG iterations overlap, i.e. they are pipelined;
- SDF-PDFG iterations are inter-dependent, i.e. synchronized by the availability of of the initial tokens; and
- SDF-PDFG is a dynamic dataflow structure, i.e. properties of consecutive iterations may drastically differ.

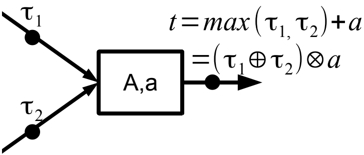




Outline I

Synchronization and delay

- Self-timed execution of SDFGs
  - Synchronization
  - Delay
- Synchronization corresponds to the $\max$ operator, while delay correponds to the $+$ operator $\Rightarrow$ Max-plus algebra



Max-plus algebra

- A semiring defined over $\mathbb{R}_{\max} = \mathbb{R} \cup \{-\infty\}$
- $a \oplus b = \max(a, b)$ and $a \otimes b = a + b$
- $-\infty \oplus a = a \oplus -\infty = a$ and $a \otimes -\infty = -\infty \otimes a = -\infty$
- $A \oplus B$ is defined by $[A \oplus B]_{ij} = a_{ij} \oplus b_{ij}$
- $A \otimes B$ is defined by $[A \otimes B]_{ij} = \bigoplus_{j=1}^{k} a_{ij} \otimes b_{jk}$
- We use $a \otimes c$ or $c \otimes a$ to denote the vector $[a_i + c]$
- $\otimes$ symbol in the exponent indicates a matrix power, $c \in \mathbb{R}$, $c^{\otimes n} = n \cdot c$

Max-plus for SDF

- SDFGs progress in iterations
- Define a vector $\gamma(k)$ that records the production times of initial tokens after the k-th SDFG iteration
- Then, the evolution of the graph can be described using the following Max-plus linear equation

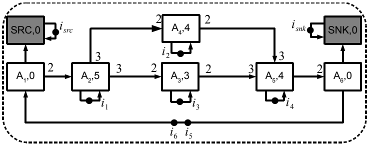
$$\gamma(k+1) = M_G \otimes \gamma(k)$$

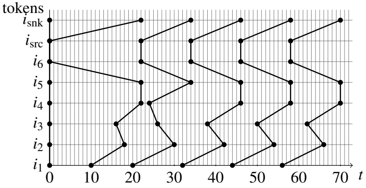
- M_G is the Max-plus characteristic matrix of an SDFG and is obtained by symbolic simulation of the graph for a single iteration [6]

Max-plus for SDF

$$\gamma(k) = [t_{i_1}, t_{i_2}, t_{i_3}, t_{i_4}, t_{i_5}, t_{i_6}, t_{i_{\text{src}}}, t_{i_{\text{snk}}}]^T, \quad \gamma(0) = [0,0,0,0,0,0,0,0]^T$$

$$\gamma(1) = \underbrace{\begin{bmatrix} 10 & -\infty & -\infty & -\infty & -\infty & 10 & -\infty & -\infty \\ 18 & 12 & -\infty & -\infty & -\infty & 18 & -\infty & -\infty \\ 16 & -\infty & 9 & -\infty & -\infty & 16 & -\infty & -\infty \\ 22 & 16 & 14 & 8 & -\infty & 22 & -\infty & -\infty \\ 22 & 16 & 14 & 8 & -\infty & 22 & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & 0 & -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty & -\infty & -\infty & 0 & 0 & -\infty \\ 22 & 16 & 14 & 8 & -\infty & 22 & -\infty & 0 \end{bmatrix}}_{M_G} \otimes \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \max(10,10) \\ \max(18,12,18) \\ \max(16,9,16) \\ \max(22,16,14,8,22) \\ \max(22,16,14,8,22) \\ \max(0) \\ \max(0,0) \\ \max(22,16,14,18,22) \end{bmatrix} = \begin{bmatrix} 10 \\ 18 \\ 16 \\ 22 \\ 22 \\ 0 \\ 0 \\ 22 \end{bmatrix}$$



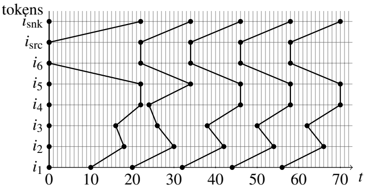


Max-plus for SDF

- Sequence $\bar{\gamma} = \gamma(1), \gamma(2), \ldots$ is ultimately periodic
- From the cyclicity theorem of Max-plus algebra it follows that for every $k \geq t(M_G)$

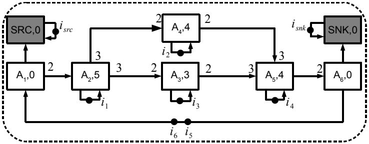
$$\gamma(k+\nu) = \lambda^{\otimes \nu} \otimes \gamma(k)$$

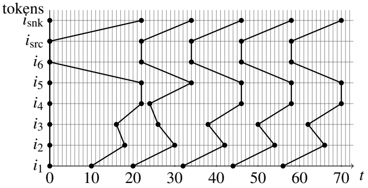
where ν is the cyclicity of M_G, λ the eigenvalue of M_G and $t(M_G)$ the transient time of M_G



Throughput of an SDFG

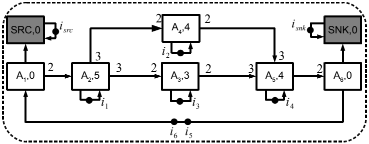
- For an SDFG G, $Th(G) = 1/\lambda$
- The value of λ corresponds to the maximum cycle mean (MCM) of the communication graph of M_G
- $Th(G) = 1/12$ *iterations per time-unit*

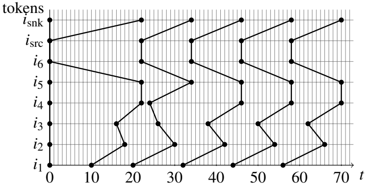




Latency of an SDFG

- Let S_γ be the set of the first $t(M_G)+\nu$ elements of $\bar{\gamma} = \gamma(1), \gamma(2), \ldots$
- For an SDFG G, $L(G) = \max_{\gamma \in S_\gamma} \left([\gamma]_{i_{\mathrm{snk}}} - [\gamma]_{i_{\mathrm{src}}}\right)$
- $L(G) = 22$ *time-units*

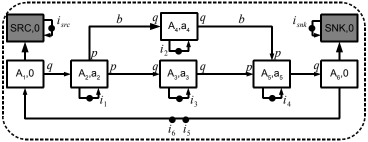


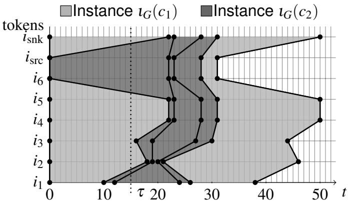


Outline I

Max-plus for SDF-PDF

- An SDF-PDFG progresses in SDFG instance iterations, i.e. one SDF-PDFG iteration corresponds to an iteration of an arbitrary SDFG instance $\iota_G(c)$ of that SDF-PDFG
- Thus, $\gamma(k+1) = M_{\iota_G(c)} \otimes \gamma(k)$
- Can we actually compute all $\{M_{\iota_G(c)}\}$ where $c \in X$?





Max-plus for SDF-PDF

- We would like to obtain a parametrized Max-plus characteristic matrix of an SDF-PDFG $M(c)$ where $c \in X$
- $M(c)$ for a concrete c becomes $M_{\iota_G(c)}$
- Symbolic simulation of the graph for a single iteration?
- Analytical simulation of the graph for a single iteration?
- The goal is to express the dependence of $\gamma(k+1)$ on $\gamma(k)$

$$X \xrightarrow{p \quad \iota \quad q} Y$$

$$\tau(Y,n) = \tau\left(X, \lceil\frac{n \cdot q - \iota}{p}\rceil\right) \otimes y$$

Max-plus for SDF-PDF

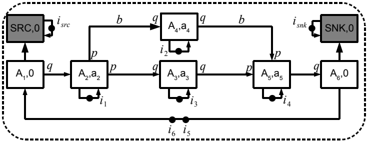
- $\gamma(k+1) = [t'_{i_1}, t'_{i_2}, t'_{i_3}, t'_{i_4}, t'_{i_5}, t'_{i_6}, t'_{i_{\text{src}}}, t'_{i_{\text{snk}}}]^T$
- $\gamma(k) = [t_{i_1}, t_{i_2}, t_{i_3}, t_{i_4}, t_{i_5}, t_{i_6}, t_{i_{\text{src}}}, t_{i_{\text{snk}}}]^T$
- t_{i_1} corresponds to the vector $[0, -\infty, -\infty, -\infty, -\infty, -\infty, -\infty, -\infty]^T$ all the way to $t_{i_{\text{snk}}}$ that corresponds to the vector $[-\infty, -\infty, -\infty, -\infty, -\infty, -\infty, -\infty, 0]^T$
- The iteration schedule: $\mathscr{I} = A_1^1 SRC_1^1 A_2^p A_3^q A_4^q A_5^p A_6^1 SNK_2^1$

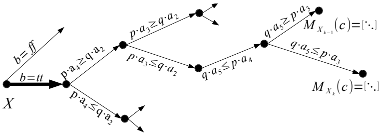
$$X \xrightarrow{p \quad \iota \quad q} Y$$

$$\tau(Y,n) = \tau\left(X, \lceil \frac{n \cdot q - \iota}{p} \rceil\right) \otimes y$$

Max-plus for SDF-PDF

- Max-plus superposition
- Max-plus convolution
- Conditional channels





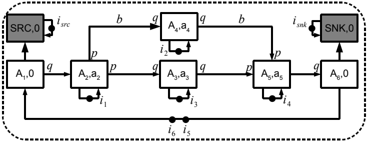
Max-plus for SDF-PDF

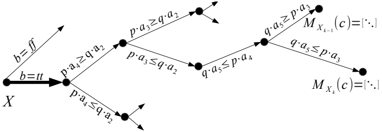
— Fire A_1

$$\tau(A_1,1) = t_{i_6} \otimes 0$$
$$= [-\infty, -\infty, -\infty, -\infty, -\infty, 0, -\infty, -\infty] \otimes \gamma(k)$$

— Fire A_2

$$\tau(A_2,n) = t_{i_6} \otimes {a_2}^{\otimes n} \oplus t_{i_1} \otimes {a_2}^{\otimes n}$$
$$= [{a_2}^{\otimes n}, -\infty, -\infty, -\infty, -\infty, {a_2}^{\otimes n}, -\infty, -\infty] \otimes \gamma(k)$$

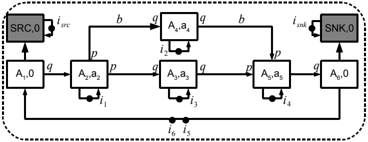


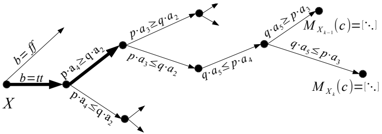


Max-plus for SDF-PDF

— Fire A_4

$$\tau(A_4,n) = [a_2^{\otimes(\frac{q}{p}+1)} \otimes a_4^{\otimes n}, a_4^{\otimes n}, -\infty, -\infty, -\infty, a_2^{\otimes(\frac{q}{p}+1)} \otimes a_4^{\otimes n}, -\infty, -\infty] \otimes \gamma(k) \qquad (1)$$

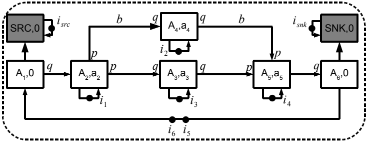


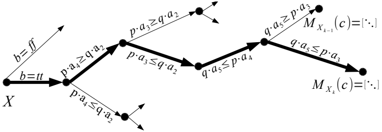


Max-plus for SDF-PDF

— After completing the iteration, we collect the results into the vector $\gamma(k+1) = [t'_{i_1}, t'_{i_2}, t'_{i_3}, t'_{i_4}, t'_{i_5}, t'_{i_6}, t'_{i_{\mathrm{src}}}, t'_{i_{\mathrm{snk}}}]^T$ where

$$\begin{aligned} t'_{i_1} &= \tau(A_2, q), \quad & t'_{i_2} &= \tau(A_4, p) \\ t'_{i_3} &= \tau(A_3, p), \quad & t'_{i_4} &= \tau(A_5, q) \\ t'_{i_5} &= \tau(A_6, 1), \quad & t'_{i_6} &= t_{i_5} \\ t'_{i_{\mathrm{src}}} &= \tau(SRC, 1), \quad & t'_{i_{\mathrm{snk}}} &= \tau(SNK, 1) \end{aligned}$$

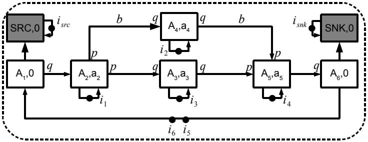


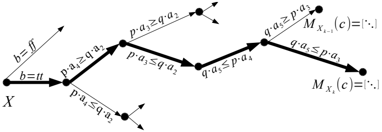


Max-plus for SDF-PDF

— We obtain

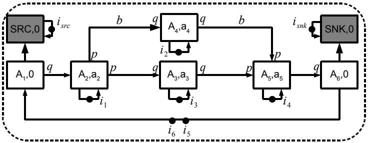
$$M_{X_k}(c) = \begin{bmatrix}
a_2^{\otimes q} & -\infty & -\infty & -\infty & -\infty & a_2^{\otimes q} & -\infty & -\infty \\
a_2^{\otimes(\frac{q}{p}+1)} \otimes a_4^{\otimes p} & a_4^{\otimes p} & -\infty & -\infty & -\infty & a_2^{\otimes(\frac{q}{p}+1)} \otimes a_4^{\otimes p} & -\infty & -\infty \\
a_2^{\otimes(q+1)} \otimes a_3 & -\infty & a_3^{\otimes p} & -\infty & -\infty & a_2^{\otimes(q+1)} \otimes a_3 & -\infty & -\infty \\
a_2^{\otimes(\frac{q}{p}+1)} \otimes a_4^{\otimes(p+1)} \otimes a_5 & a_4^{\otimes(p+1)} \otimes a_5 & a_3^{\otimes(p+1)} \otimes a_5 & a_5^{\otimes p} & -\infty & a_2^{\otimes(\frac{q}{p}+1)} \otimes a_4^{\otimes(p+1)} \otimes a_5 & -\infty & -\infty \\
a_2^{\otimes(\frac{q}{p}+1)} \otimes a_4^{\otimes(p+1)} \otimes a_5 & a_4^{\otimes(p+1)} \otimes a_5 & a_3^{\otimes(p+1)} \otimes a_5 & a_5^{\otimes p} & -\infty & a_2^{\otimes(\frac{q}{p}+1)} \otimes a_4^{\otimes(p+1)} \otimes a_5 & -\infty & -\infty \\
-\infty & -\infty & -\infty & -\infty & 0 & -\infty & -\infty & -\infty \\
-\infty & -\infty & -\infty & -\infty & -\infty & 0 & 0 & -\infty \\
a_2^{\otimes(\frac{q}{p}+1)} \otimes a_4^{\otimes(p+1)} \otimes a_5 & a_4^{\otimes(p+1)} \otimes a_5 & a_3^{\otimes(p+1)} \otimes a_5 & a_5^{\otimes p} & -\infty & a_2^{\otimes(\frac{q}{p}+1)} \otimes a_4^{\otimes(p+1)} \otimes a_5 & -\infty & 0
\end{bmatrix}$$

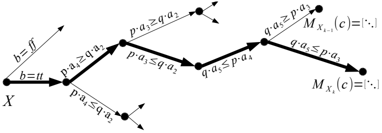




Max-plus for SDF-PDF

- Initial X has to be partitioned in to a set of subregions $\{X_k\}$ that we call natural subdomains of an SDF-PDFG
- Complexity issues
- Conservativeness of $M_{X_k}(c)$
  - $x \leq \lceil x \rceil < x + 1$ with respect to $\tau(Y,n) = \tau\left(X, \left\lceil \frac{n \cdot q - \iota}{p} \right\rceil\right) \otimes y$ and monotonicity of SDF



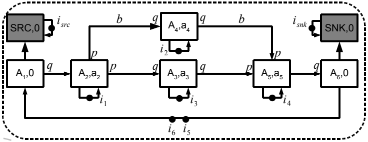


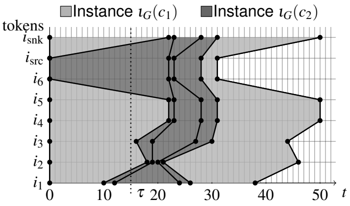
Outline I

Worst-case evaluation of an SDF-PDFG

- For and SDF-PDFG, $\gamma(k+1) = M_{\iota_G(c)} \otimes \gamma(k)$, i.e. $\gamma(k+1) = M_{X_k}(c) \otimes \gamma(k)$ s.t. $c \in X_k$
- SDF-PDFG, FSM-SADF [7] and Max-plus automata [5]
- Instances occur in an arbitrary order
- Worst-case evaluation Max-plus matrix:

$$M = \bigoplus_{c_k \in X} M_{\iota_G(c_k)}$$

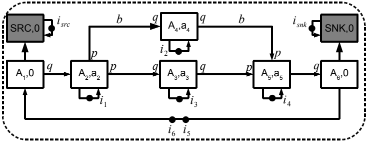


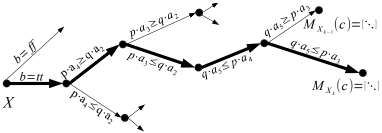


Worst-case evaluation of an SDF-PDFG

- X is too large to be enumerated
- Worst-case evaluation Max-plus matrix re-written:

$$M = \bigoplus_{X_k \in X} \bigoplus_{c \in X_k} M_{X_k}(c)$$





Worst-case evaluation of an SDF-PDFG

— Worst-case evaluation Max-plus matrix re-written:

$$M = \bigoplus_{X_k \in X} \bigoplus_{c \in X_k} M_{X_k}(c)$$

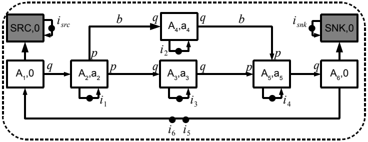
— Obtaining $\bigoplus_{c \in X_k} M_{X_k}(c)$:

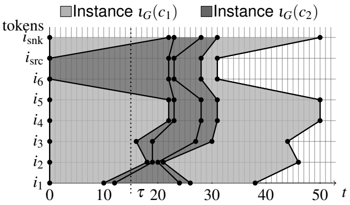
foreach (i,j) s.t. $[M_{X_k}(c)]_{ij} \neq -\infty$ **do**

$$\begin{aligned} \underset{p}{\text{maximize}} \quad & [M_{X_k}(c)]_{ij} \\ \text{subject to} \quad & c \in X_k. \end{aligned}$$

Worst-case throughput of an SDF-PDFG

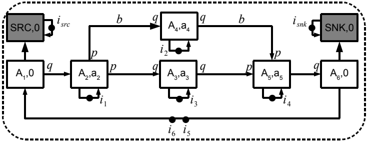
- For an SDF-PDFG G, $Th(G) = 1/\lambda$
- The value of λ corresponds to the maximum cycle mean (MCM) of the communication graph of M

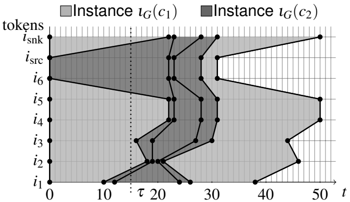




Worst-case latency of an SDF-PDFG

- Let S_γ be the set of the first $t(M) + \nu$ elements of $\bar{\gamma} = \gamma(1), \gamma(2), \ldots$
- For an SDF-PDFG G, $L(G) = \max\limits_{\gamma \in S_\gamma} \left([\gamma]_{i_{\text{snk}}} - [\gamma]_{i_{\text{src}}} \right)$





Discussion

— If all parameters are constrained to intervals, then isn't it simpler to create a "worst-case SDFG" from an SDF-PDFG by taking the interval upper endpoints for values of the parameters?

— The answer is yes, because SDF is monotonic

— However, what if parameters expose interdependencies?

— Rates and firing delays

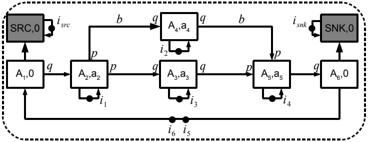
— Then, taking the upper endpoints might result in serious over-approximations

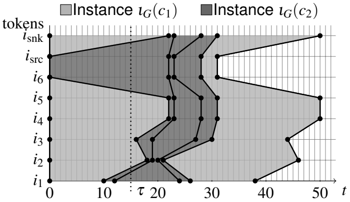
— These reduce design's optimization potential

Discussion

— Assume:

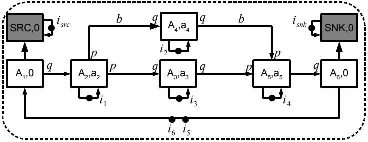
$$X = X_k \cap \{p = w_1 \cdot w_2, w_1 + w_2 = 2 \cdot x_1 - x_2, \\ p \in [1,10], q \in [1,10], w_1 \in [1,3], w_2 \in [1,4], \\ x_1 \in [1,3], x_2 \in [1,5], a_2 \in [1,7], a_4 = 4, \\ a_3 \in [1,5], a_5 = 4\}$$

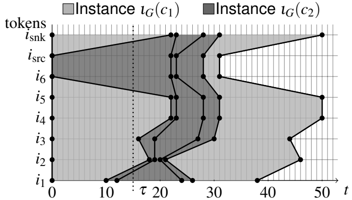




Discussion

Comparison			
Metric	**SDF**	**SDF-PDF**	$\Delta_{rel}[\%]$
Throughput [*it. per time-unit*]	$1/70$	$1/24$	291.66
Latency[*time-units*]	79.0	42.5	46.20

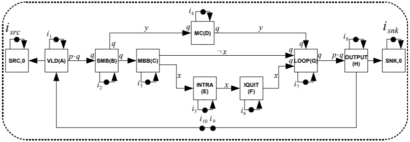


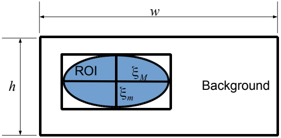


Outline I

VC-1 decoder

- VC-1 decoder [1]
- ROI coding [8]



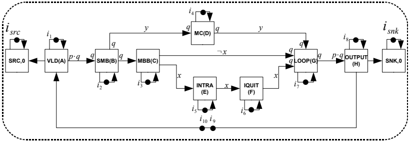


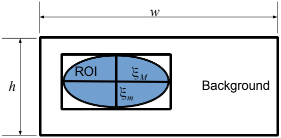
$$
\begin{aligned}
X = \{ & p = (2 \cdot \xi_M \cdot 2 \cdot \xi_m)/(16 \cdot 16), p \in [1, P], q \in [1, 16] \\
& p' \geq \mu/100 \cdot P, p + p' \leq P \\
& o^2 = 4 \cdot \pi^2 (\xi_M^2 + \xi_m^2), o \geq O, \varepsilon^2 \cdot \xi_M^2 = \xi_M^2 - \xi_m^2, \varepsilon = E, 2 \cdot \xi_M \leq w, 2 \cdot \xi_m \leq h \\
& x \wedge \neg y \mid \neg x \wedge y \mid x \wedge y, \\
& a = 7400, b = 10, c = 10, d = 1937, e = 288, f = 365, g = 4074, h = 10 \}
\end{aligned}
$$

VC-1 decoder

- SDTV input format with signal type 480i 16:9 and resolution 720×480 pixels, $P = 1620$
- $O = 700$, $E = 0.6$ and $\mu = 30$

Comparison			
Metric	**SDF**	**SDF-PDF**	$\Delta_{\mathrm{rel}}[\%]$
Throughput [*sl. per time-unit*]	$1.05699 \cdot 10^{-7}$	$1.74727 \cdot 10^{-7}$	65.30
Latency[*time-units*]	$9,465,192.0$	$8,041,792.0$	15.03





Outline I

Questions

Thanks! Questions?

Outline I

References I

V. Bebelis, P. Fradet, and A. Girault.
A framework to schedule parametric dataflow applications on many-core platforms.
In *Proceedings of the 2014 SIGPLAN/SIGBED Conference on Languages, Compilers and Tools for Embedded Systems*, LCTES '14, pages 125–134, New York, NY, USA, 2014. ACM.

V. Bebelis, P. Fradet, A. Girault, and B. Lavigueur.
BPDF: A statically analyzable dataflow model with integer and boolean parameters.
In *Proceedings of the Eleventh ACM International Conference on Embedded Software*, EMSOFT '13, pages 3:1–3:10, Piscataway, NJ, USA, 2013. IEEE Press.

References II

- B. Bhattacharya and S.S. Bhattacharyya.
  Parameterized dataflow modeling of dsp systems.
  In *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, volume 6, pages 3362–3365 vol.6, 2000.
- P. Fradet, A. Girault, and P. Poplavko.
  SPDF: A schedulable parametric data-flow MoC.
  In *Design, Automation Test in Europe Conference Exhibition (DATE), 2012*, pages 769–774, March 2012.

References III

S. Gaubert.
Performance evaluation of (max,+) automata.
Automatic Control, IEEE Transactions on, 40(12):2014–2025, Dec 1995.

M. Geilen.
Synchronous dataflow scenarios.
ACM Trans. Embed. Comput. Syst., 10(2):16:1–16:31, January 2011.

References IV

- M. Geilen and S. Stuijk.
  Worst-case performance analysis of synchronous dataflow scenarios.
  In *Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2010 IEEE/ACM/IFIP International Conference on*, pages 125–134, Oct 2010.
- D. Grois and O. Hadar.
  Recent advances in region-of-interest video coding.
  Recent Advances on Video Coding, 2011.
- E.A. Lee and D.G. Messerschmitt.
  Synchronous data flow.
  Proceedings of the IEEE, 75(9):1235–1245, Sept 1987.

References V

T. Stefanov.
System level MPSoC design with Daedalus, 2015.

M.H. Wiggers, M.J.G. Bekooij, and G.J.M. Smit.
Buffer capacity computation for throughput constrained streaming applications with data-dependent inter-task communication.
In *Real-Time and Embedded Technology and Applications Symposium, 2008. RTAS '08. IEEE*, pages 183–194, April 2008.