Symbolic Computation of Latency for Dataflow Graphs

Adnan Bouakaz Pascal Fradet Alain Girault

SYNCHRON International Workshop, Bamberg

December 7th, 2016

Outline

Data-flow models of computation

Stream-processing applications are found in many embedded systems

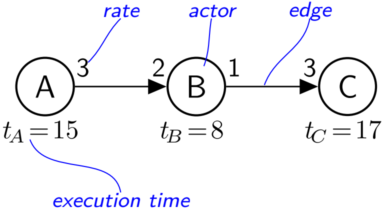
- video codecs, software defined radio, ...
- computationally intensive
- strict quality-of-service requirements
- low energy consumption
- more and more these applications run on many-core platforms

Data-flow models of computation are good at:

- Expressing task-level parallelism
- Achieving efficient implementation
- Guaranteeing performances at compile time:
    - **throughput**: stream oriented applications
    - **latency**: automatic control oriented applications
    - **buffer sizes**: all embedded applications

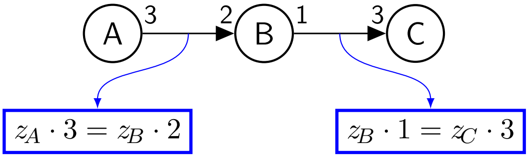
Acyclic Synchronous Data-FLow (SDF) graphs

[Lee and Messerschmitt, Proc. 1987]



Acyclic Synchronous Data-FLow (SDF) graphs

[Lee and Messerschmitt, Proc. 1987]



System of Balance Equations

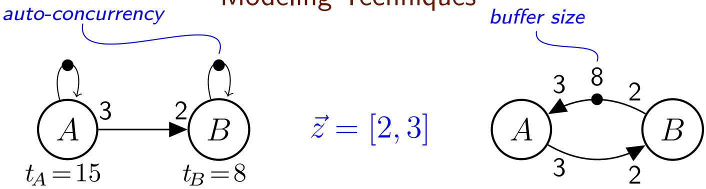
- **Consistent SDF graph G:** this system has a non-null solution
- **Repetition vector of G:** $\vec{z} = [A^2, B^3, C^1]$
- **Iteration** $=$ firing sequence that returns G to its initial state

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} \xrightarrow{A^2} \begin{bmatrix} 6 \\ 0 \end{bmatrix} \xrightarrow{B^3} \begin{bmatrix} 0 \\ 3 \end{bmatrix} \xrightarrow{C^1} \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Scheduling policy

- As Soon As Possible (ASAP) [Sriram and Bhattacharyya 2000]
- No auto-concurrency

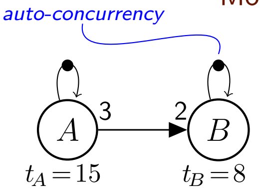
Modeling Techniques



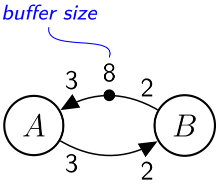
Scheduling policy

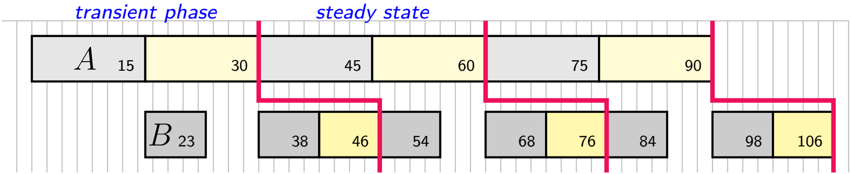
- As Soon As Possible (ASAP) [Sriram and Bhattacharyya 2000]
- No auto-concurrency

Modeling Techniques



$\vec{z} = [2, 3]$

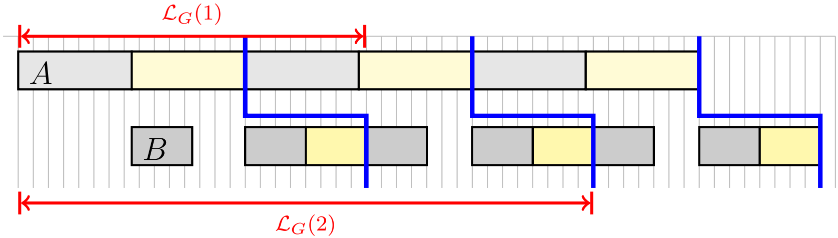




Scheduling policy

Definition: Multi-iteration latency of graph G:

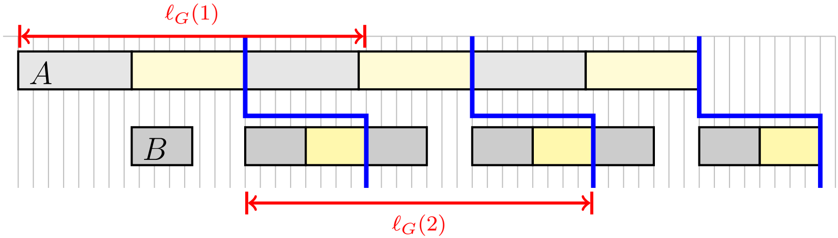
$\mathcal{L}_G(n) =$ the finish time of the n^{th} iteration.



Scheduling policy

Definition: Input-output latency of graph G:

$\ell_G(n) =$ the duration between the start and ending of the n^{th} iteration.



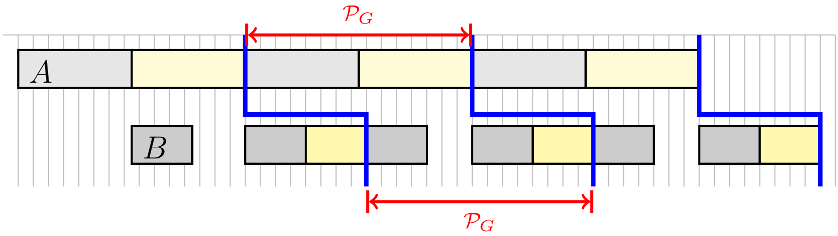
Scheduling policy

Definition: Period of graph G:

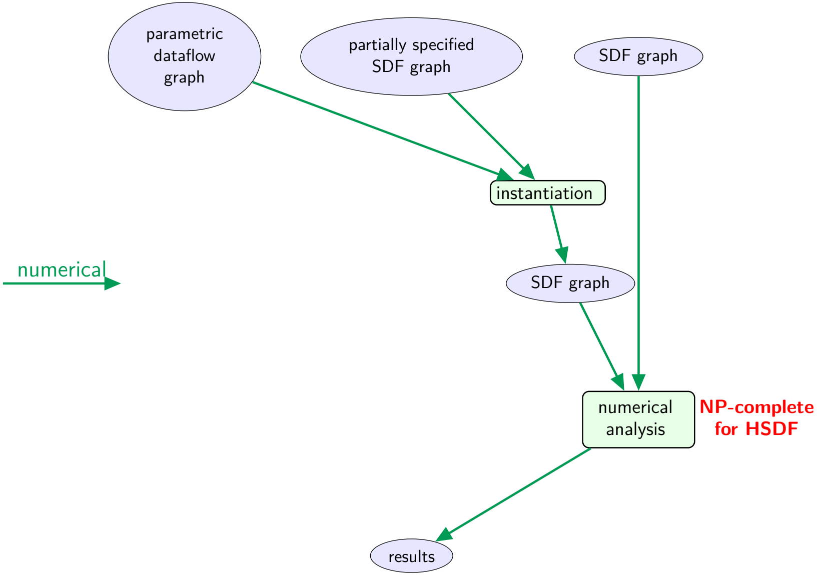
$$\mathcal{P}_G = \text{ the average length of an iteration } = \lim_{n\to\infty} \frac{\mathcal{L}_G(n)}{n}$$

Definition: Throughput of graph G:

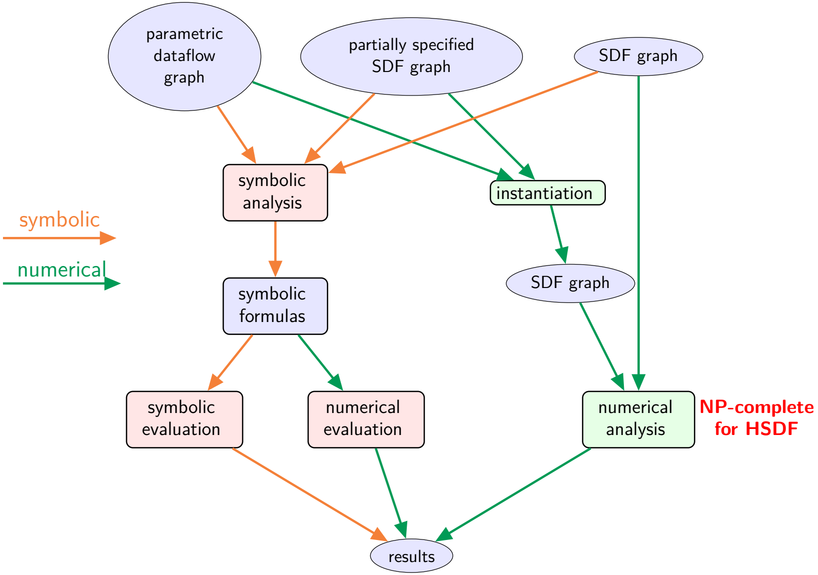
$$\mathcal{T}_G = \frac{1}{\mathcal{P}_G}$$



Symbolic analysis



Symbolic analysis



Outline

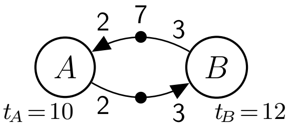
Duality theorem

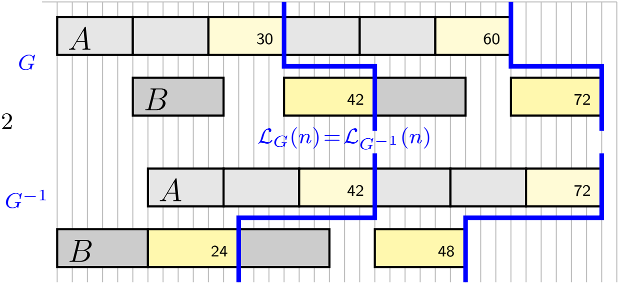
Definition: The dual of an SDF graph G:

G^{-1} is obtained by reversing all edges of G.

Duality theorem:

Let G be any (cyclic or not) live graph and G^{-1} be its dual, then $\mathcal{T}_G = \mathcal{T}_{G^{-1}}$ and $\forall i.\ \mathcal{L}_G(i) = \mathcal{L}_{G^{-1}}(i)$.





Duality theorem

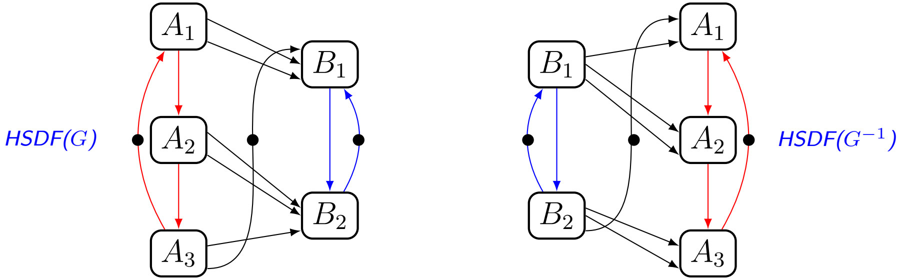
Definition: The dual of an SDF graph G:

G^{-1} is obtained by reversing all edges of G.

Duality theorem:

Let G be any (cyclic or not) live graph and G^{-1} be its dual, then $\mathcal{T}_G = \mathcal{T}_{G^{-1}}$ and $\forall i.\ \mathcal{L}_G(i) = \mathcal{L}_{G^{-1}}(i)$.

- **Proof:** Using SDF-to-HSDF transformation + unfolding:



Outline

1. Introduction
2. Preliminary results
3. Graph $A \xrightarrow{p \quad q} B$
   - Enabling patterns
   - Minimum latency
4. Generalization to chains and acyclic graphs
5. Experiments
6. Conclusion

Preliminaries about graph $A \xrightarrow{p \quad q} B$

- Four parameters: $p, q \in \mathbb{N}^+$ and $t_A, t_B \in \mathbb{R}^+$.
- Repetition vector:

$$\left[z_A = \frac{q}{\gcd(p,q)}, z_B = \frac{p}{\gcd(p,q)}\right]$$

- ASAP period: $\mathcal{P}_G = \max(z_A t_A, z_B t_B)$.

Problem statement

What is $\theta_{A,B}$ the min. size of channel $A \longrightarrow B$ s.t. the ASAP execution achieves the max. throughput?

Solution

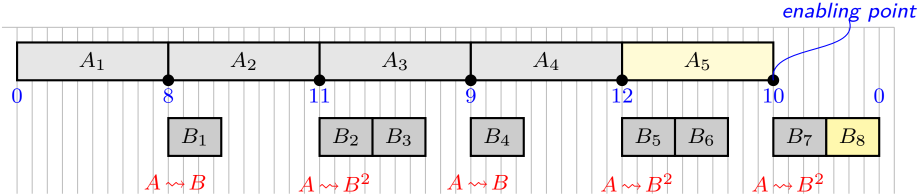
- $p + q - \gcd(p,q) < \theta_{A,B} \leq 2(p + q - \gcd(p,q))$
- **Proof:** 18 cases in total: $p, q \to 6$ cases; $t_A, t_B \to 3$ cases

Enabling patterns

- A time-independent analytic and parametric characterization of the data-dependency $A \rightarrow B$ that covers one iteration.

- **Example:**

  Graph $A \xrightarrow{8 \quad 5} B$ with $t_A = 20$ and $t_B = 7$



$$A^i \rightsquigarrow B^j \Leftrightarrow i \text{ firings of } A \text{ enables } j \text{ firings of } B.$$

Unfolded pattern:

$$A \rightsquigarrow B; A \rightsquigarrow B^2; A \rightsquigarrow B; A \rightsquigarrow B^2; A \rightsquigarrow B^2$$

Enabling patterns

- **Unfolded pattern:**

$$\underbrace{A \rightsquigarrow B; A \rightsquigarrow B^2;}_{\text{block}} \underbrace{A \rightsquigarrow B; A \rightsquigarrow B^2; A \rightsquigarrow B^2}_{\text{block}}$$

Enabling patterns

- **Unfolded pattern:**

$$\underbrace{A \rightsquigarrow B; A \rightsquigarrow B^2;}_{\text{block}} \underbrace{A \rightsquigarrow B; A \rightsquigarrow B^2; A \rightsquigarrow B^2}_{\text{block}}$$

- **Factorized pattern:**

$$\left[A \rightsquigarrow B; [A \rightsquigarrow B^2]^{f_i}\right]^{i=1..2} \text{ with } f_1 = 1, f_2 = 2$$

Enabling patterns

- **Unfolded pattern:**

$$\underbrace{A \rightsquigarrow B;\, A \rightsquigarrow B^2;}_{\text{block}}\, \underbrace{A \rightsquigarrow B;\, A \rightsquigarrow B^2;\, A \rightsquigarrow B^2}_{\text{block}}$$

- **Factorized pattern:**

$$\left[A \rightsquigarrow B;\, [A \rightsquigarrow B^2]^{f_i}\right]^{i=1\cdot\cdot 2} \text{ with } f_1 = 1, f_2 = 2$$

- **General case:**

$$\left[A \rightsquigarrow B^k;\, \left[A \rightsquigarrow B^{k+1}\right]^{\alpha_j}\right]^{j=1\cdot\cdot \frac{q-r}{\gcd(p,q)}}$$

with $p = kq + r$ and $\alpha_j = \left\lfloor \frac{jr}{q-r} \right\rfloor - \left\lfloor \frac{(j-1)r}{q-r} \right\rfloor$

Enabling patterns

Case A. $p \geq q$

Let $p = kq + r$ with $0 \leq r < q$

Case A.1. $r = 0$

$$A \rightsquigarrow B^k$$

Case A.2. $q \leq 2r$

$$\left[A \rightsquigarrow B^k ; \left[A \rightsquigarrow B^{k+1}\right]^{\alpha_j}\right]^{j=1..\frac{q-r}{\gcd(p,q)}}$$

Case A.3. $q > 2r$

$$\left[\left[A \rightsquigarrow B^k\right]^{\beta_j} ; A \rightsquigarrow B^{k+1}\right]^{j=1..\frac{r}{\gcd(p,q)}}$$

$$\alpha_j = \left\lfloor \frac{jr}{q-r} \right\rfloor - \left\lfloor \frac{(j-1)r}{q-r} \right\rfloor$$

$$\beta_j = \left\lceil \frac{jq}{r} \right\rceil - \left\lceil \frac{(j-1)q}{r} \right\rceil - 1$$

Case B. $p < q$

Let $q = kp + r$ with $0 \leq r < p$

Case B.1. $r = 0$

$$A^k \rightsquigarrow B$$

Case B.2. $p \geq 2r$

$$\left[A^{k+1} \rightsquigarrow B ; \left[A^k \rightsquigarrow B\right]^{\gamma_j}\right]^{j=1..\frac{r}{\gcd(p,q)}}$$

Case B.3. $p < 2r$

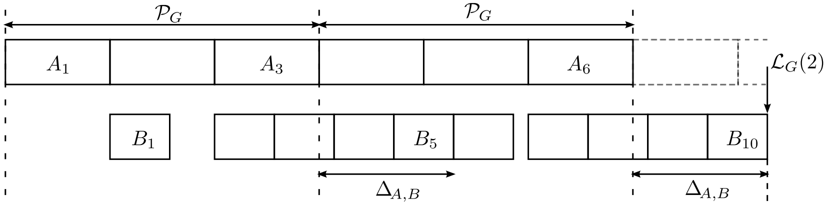
$$\left[\left[A^{k+1} \rightsquigarrow B\right]^{\lambda_j} ; A^k \rightsquigarrow B\right]^{j=1..\frac{p-r}{\gcd(p,q)}}$$

$$\gamma_j = \left\lceil \frac{jp}{r} \right\rceil - \left\lfloor \frac{(j-1)p}{r} \right\rfloor - 1$$

$$\lambda_j = \left\lceil \frac{jr}{p-r} \right\rceil - \left\lceil \frac{(j-1)r}{p-r} \right\rceil$$

Multi-iteration latency: **Case** $z_A t_A \geq z_B t_B$

- A imposes a higher load than B
- A never gets idle $\Longrightarrow \mathcal{P}_G = z_A t_A$
- $\mathcal{L}_G(n) = n\mathcal{P}_G + \Delta_{A,B} \Longleftrightarrow \frac{\mathcal{L}_G(n)}{n} = \frac{n\mathcal{P}_G + \Delta_{A,B}}{n} = \mathcal{P}_G + \frac{\Delta_{A,B}}{n} \geq \mathcal{P}_G$
- $\Delta_{A,B}$ is the remaining execution time for actor B after actor A has finished its firings of the n^{th} iteration
- $\Delta_{A,B}$ is constant over all iterations so $\lim_{n\to+\infty} \frac{\Delta_{A,B}}{n} = 0$



(graph $A \xrightarrow{5 \quad 3} B$ with $T_A = 14$ and $t_B = 8$)

Multi-iteration latency: **Case** $z_A t_A \geq z_B t_B$

Case I.

$$\Delta_{A,B} = \left\lceil \frac{p}{q} \right\rceil t_B$$

Case II.1.

$$\Delta_{A,B} = t_A + \left\lceil \frac{r}{q-r} \right\rceil \Big((k+1)\, t_B - t_A \Big)$$

Case II.2.

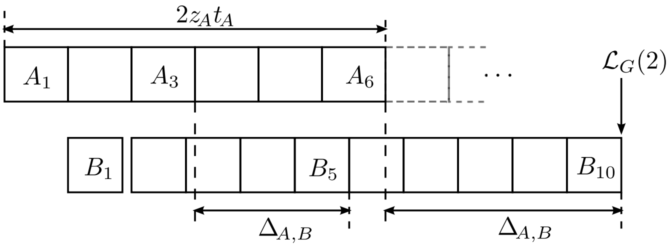
$$\Delta_{A,B} = t_B + \left\lceil \frac{p-r}{r} \right\rceil (t_B - k t_A)$$

Case III.

...

Multi-iteration latency: **Case** $z_A t_A < z_B t_B$

- B imposes a higher load than A
- B never gets idle in the steady state (untrue in transient)
- $\Delta_{A,B}$ may not constant over all iterations and diverges to infinity if the buffer is unbounded
- Better solution: compute $\Delta_{A,B}$ with the duality theorem
- $\mathcal{L}_G(n) = \mathcal{L}_{G^{-1}}(n) = n\mathcal{P}_{G^{-1}} + \Delta_{B,A}$



(graph $A \xrightarrow{5 \quad 3} B$ with $T_A = 14$ and $t_B = 12$)

Input-output latency

- **Case** $z_A t_A \geq z_B t_B$
  - A imposes the highest load $\Longrightarrow \mathcal{P}_G = z_A t_A$
  - $\ell_G(n)$ is equal to the finish time of the n^{th} iteration minus the start time of the first firing of A in the n^{th} iteration
  - $\ell_G(n) = \mathcal{L}_G(n) - (n-1)z_A t_A = \mathcal{L}_G(n) - (n-1)\mathcal{P}_G = \mathcal{P}_G + \Delta_{A,B}$
  - Hence $\ell_G = \mathcal{P}_G + \Delta_{A,B} = \mathcal{L}_G(1)$
- **Case** $z_A t_A < z_B t_B$
  - B imposes the highest load
  - Unbounded buffer: $\ell_G(n) = \mathcal{L}_G(n) - (n-1)z_A t_A$
  - It diverges with n!
  - Bounded buffer: We compute an over-approximation with a (backward) linearization technique

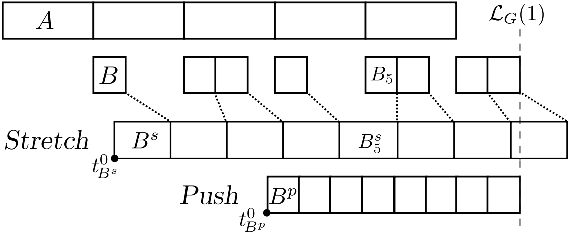
Outline

Multi-iteration latency of chain $A \xrightarrow{p \quad q} B \xrightarrow{p' \quad q'} C$

- Forward linearization B
- First analyse the graph $A \xrightarrow{p \quad q} B$
- If B does not fire continuously, then build a fictive actor B^u s.t.:

$$\forall i.\ f_B(i) \leq f_{B^u}(i) \quad \wedge \quad \exists i.\ f_B(i) = f_{B^u}(i)$$

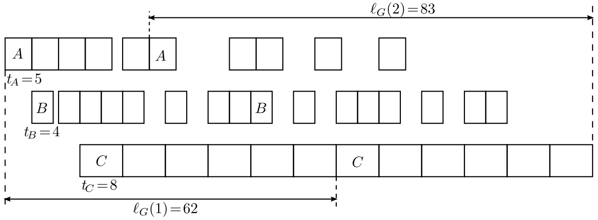
- Then analyse the graph $B^u \xrightarrow{p' \quad q'} C$
- Finally combine the two schedules



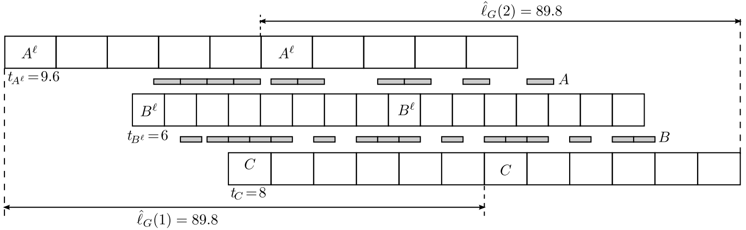
Multi-iteration latency of acyclic graphs

- Acyclic graph G seen as a set of *maximal chains* $\mathcal{G}(G)$ (chains from a source actor to a sink actor)
- **Property:** $\forall i.\ \mathcal{L}_G(i) = \max_{g \in \mathcal{G}(G)} \{\mathcal{L}_g(i)\}$
- **Proof:** transform G into HSDF then unfold i times
- Compute the multi-iteration latency of each maximal chain

Input-output latency for the chain for $A \xrightarrow{p \quad q} B \xrightarrow{p' \quad q'} C$



Linearized schedule: (backward linearization)



Conclusion: $\ell_G = 83$ and $\hat{\ell}_G = 89.8$ so we over-approximate by 8.2%

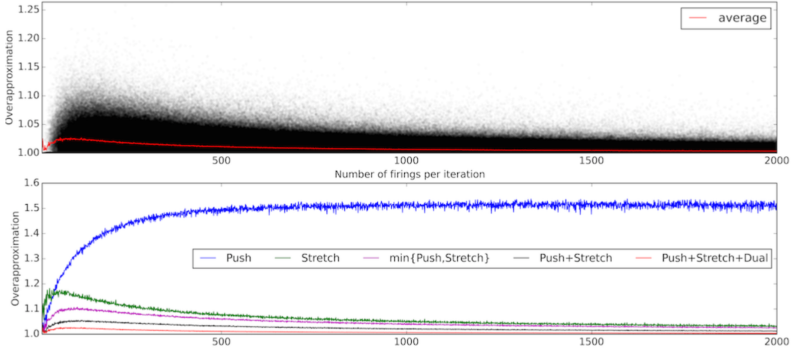
Outline

Multi-iteration latency computation for real benchmarks

graph	$\mathcal{P}_G$	$\mathcal{L}_G(1)$	$\hat{\mathcal{L}}_G(1)/\mathcal{L}_G(1)$	$\hat{\mathcal{L}}_G(2)/\mathcal{L}_G(2)$
modem	32	62	1	1
sample rate converter	960	1000	1.022	1.011
H.263 decoder	332046	369508	1	1
FFT	78844	94229	1	1
TDE	17740800	19314069	1	1

Multi-iteration latency for randomly generated chains

- Randomly generated chains of 10 actors
- $p, q \in [1, 10]$ and $t_X \in [1, 100]$
- Total number of firings per iteration $< 2 \times 10^3$
- We report $\dfrac{\hat{\mathcal{L}}_{A_1 \to A_{10}}}{\mathcal{L}_G(1)} = \dfrac{\text{approximate}}{\text{exact}}$

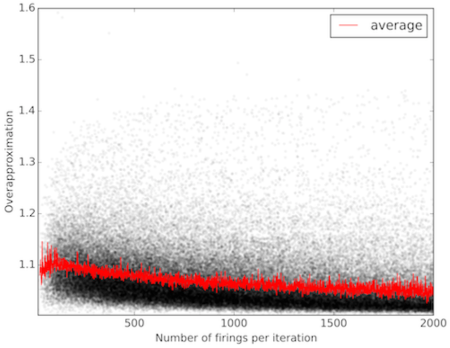


Input-output latency for randomly generated chains

- Randomly generated chains of 9 actors
- $p, q \in [1, 10]$ and $t_X \in [1, 100]$
- Total number of firings per iteration $< 2 \times 10^3$
- A_9 imposes the highest load
- Each channel size $A_i \xrightarrow{p \quad q} A_{i+1}$ is equal to $2(p + q - \gcd(p, q))$

We report

$$\frac{\hat{\ell}_G}{\ell_G} = \frac{\text{approximate}}{\text{exact}}$$



Outline

1. Introduction
2. Preliminary results
3. Graph $A \xrightarrow{p \quad q} B$
4. Generalization to chains and acyclic graphs
5. Experiments
6. Conclusion

Related work

- **[Geilen 2011]** and **[Skelin et al. 2014]**: $(\max, +)$ algebra to compute the token timestamp vector with the eigenvalue of the transition matrix

  $\Longrightarrow$ Requires the ceiling operator to be simplified

- **[Ghamarian et al. 2008]**: parametric throughput analysis for SDF graphs with bounded parametric execution times of actors *but constant rates*

  $\Longrightarrow$ Parameter space divided into a set of convex polyhedra (throughput regions), each with a throughput expression

- **[Damavandpeyma et al. 2012]**: Extension to scenario-aware dataflow (SADF)

- **[Bodin et al. 2013]**: lower bounds of the maximum throughput to compute strictly periodic schedules instead of ASAP schedules

  $\Longrightarrow$ Can handle *some* cyclic graphs, but usually our linearization methods provide better results

Conclusion

- We presented:
  - An exact analytic solution for the $A \xrightarrow{p \quad q} B$ SDF graph using enabling patterns
  - A safe generalization to acyclic graphs using forward and backward linearization
- Still to solve:

Symbolic analysis of cyclic dataflow graphs