Throughput Analysis of Synchronous Data Flow Graphs

A.H. Ghamarian, M.C.W Geilen, S. Stuijk, T. Basten, A.J.M. Moonen, M. Bekooij, B. Theelen and M. Mousavi

Department of Electrical Engineering
Electronic Systems

Outline

- Introduction
- Synchronous Data Flow Graphs (SDFGs)
- Throughput Definition
- Throughput analysis based on state space
- Traditional approach for throughput analysis
- Experimental results
- Conclusion

Introduction

- Streaming Multimedia and DSP Applications
  - Throughput
  - …
- SDFGs
  - Modeling and analysis
  - Single and multiprocessor platforms

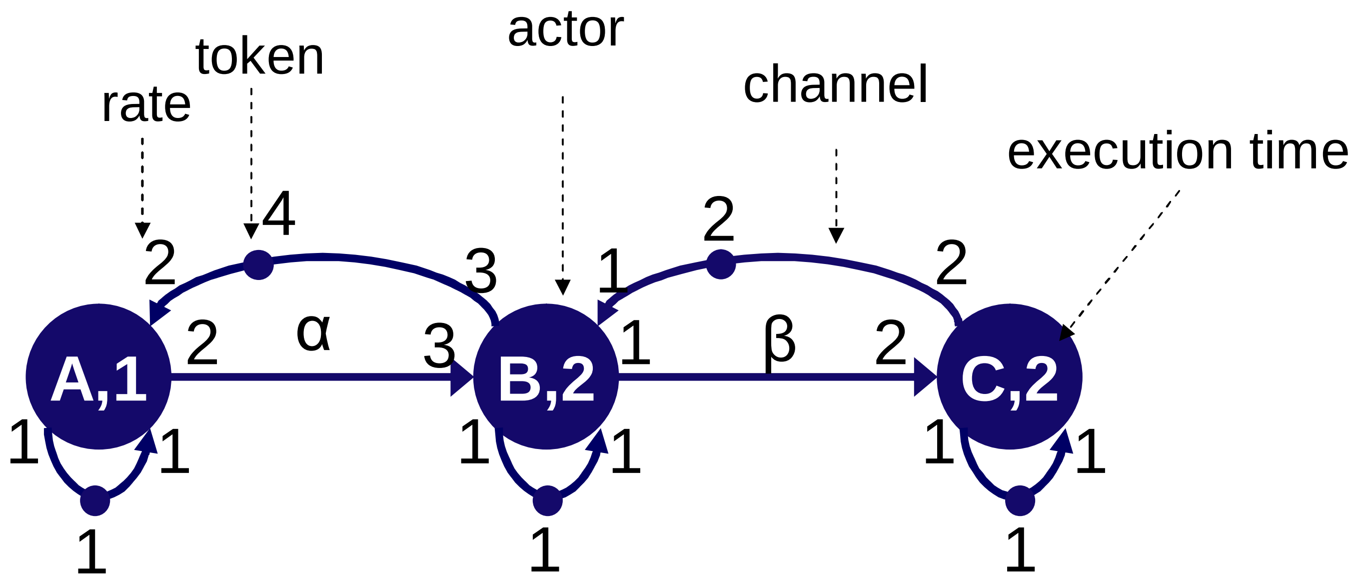
Open question: Efficient throughput analysis

4 Our result

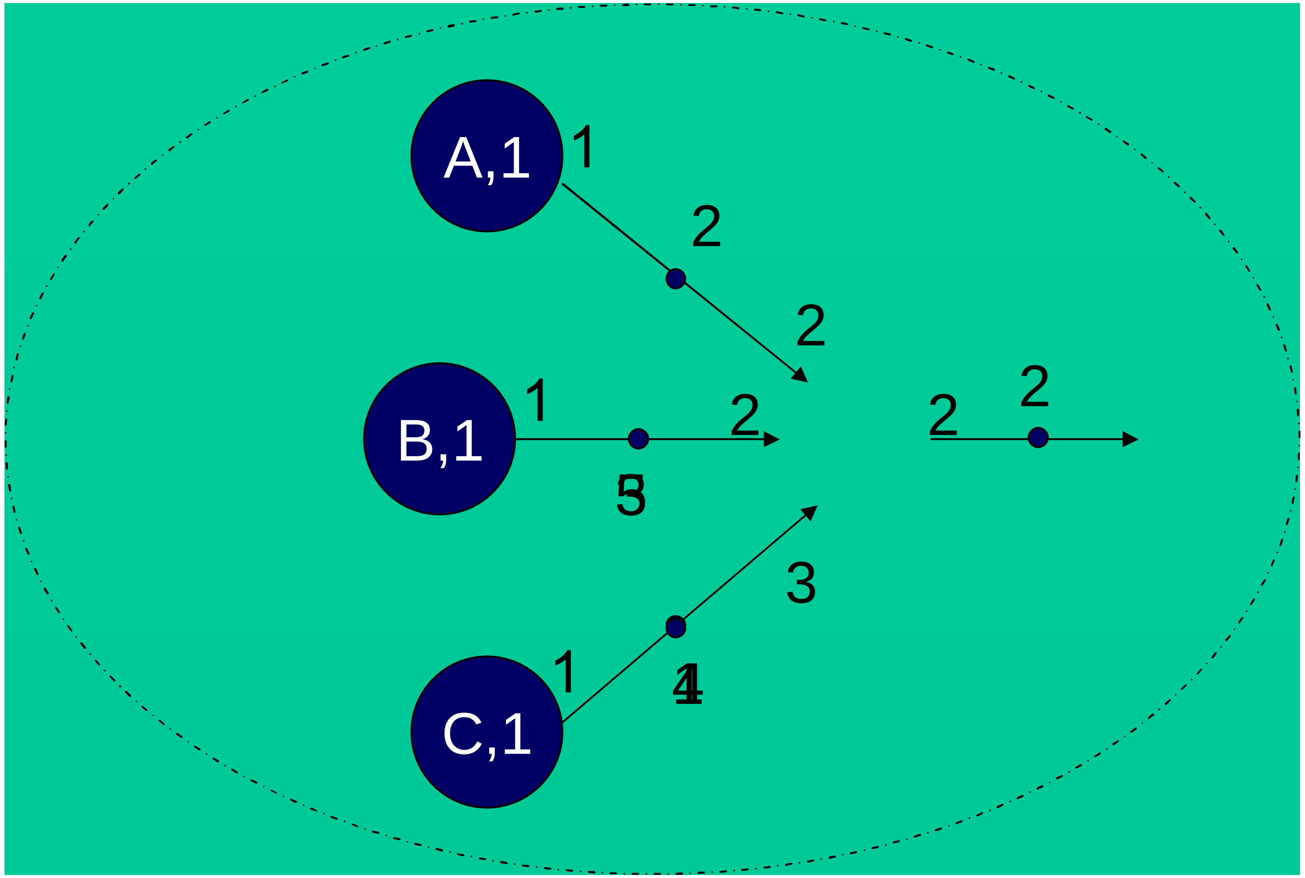
	State Space	Dastan Gupta	Howard	Young Tarjan Orlin
MP3 dec.	$11. 10^{-3}$	$1. 10^{-3}$	$1. 10^{-3}$	$1. 10^{-3}$
Modem	1.10^{-3}	$82. 10^{-3}$	$81. 10^{-3}$	$81. 10^{-3}$
Sample Rate	$1. 10^{-3}$	>1800	>1800	>1800
Satellite	$4. 10^{-3}$	>1800	>1800	>1800
H263 dec.	4	>1800	>1800	>1800

Runtimes of various throughput analysis methods (in seconds)

Synchronous Data Flow Graphs [Lee87]



Firing of an Actor

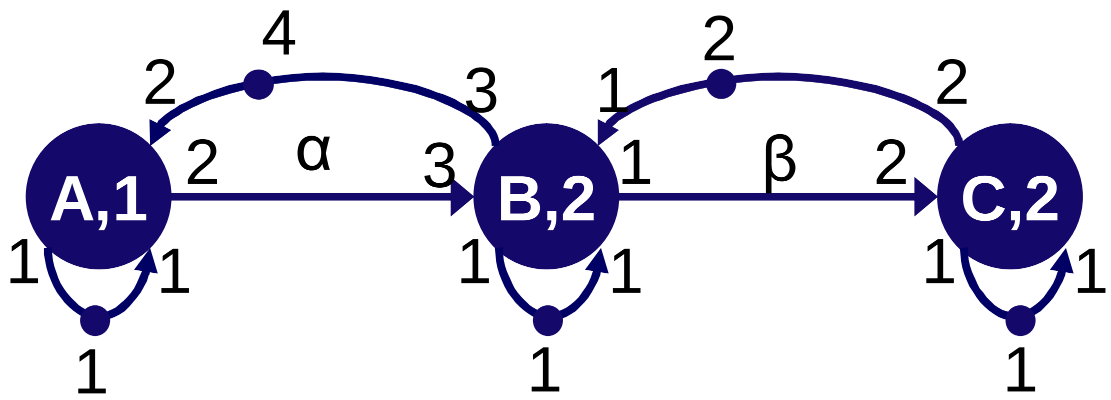


7 Execution and self-timed execution

- Execution is a ***finite*** or ***infinite*** sequence of actor firings.
- ***Self-timed*** execution is the set of firings in which each actor fires ***as soon as*** it can fire.

Assumptions on SDFGs

- Consistent
- Strongly connected



2.q(A)=3.q(B)

1.q(B)=2.q(C)

…

Repetition vector q = {(A, 3), (B, 2), (C, 1)}

An SDF graph is ***consistent*** if q(A) >0 for all actors A

Throughput Definition

- Actor throughput:

  The average number of firings of one actor per time unit.

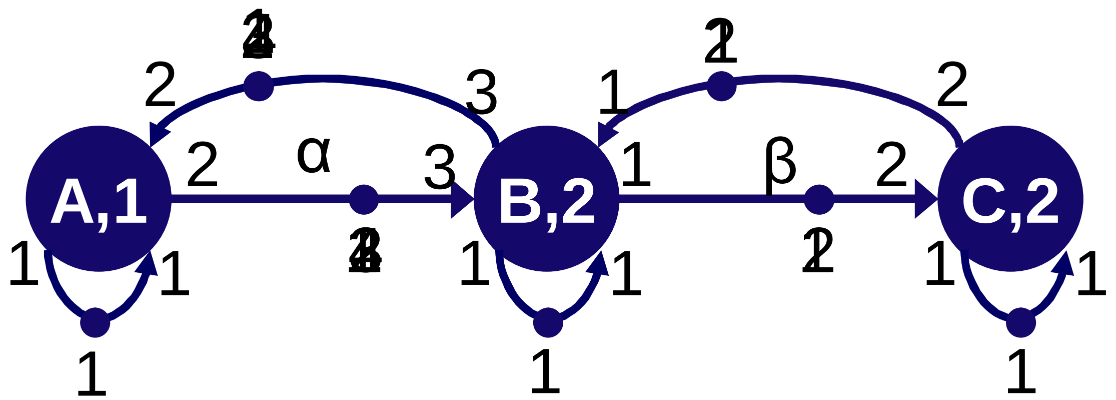
$$Th(a) = \lim_{k \to \infty} \frac{k \text{ firings of } a}{\text{end time of these firings}}.$$

- $Th(a).q(a) = Th(b).q(b).$

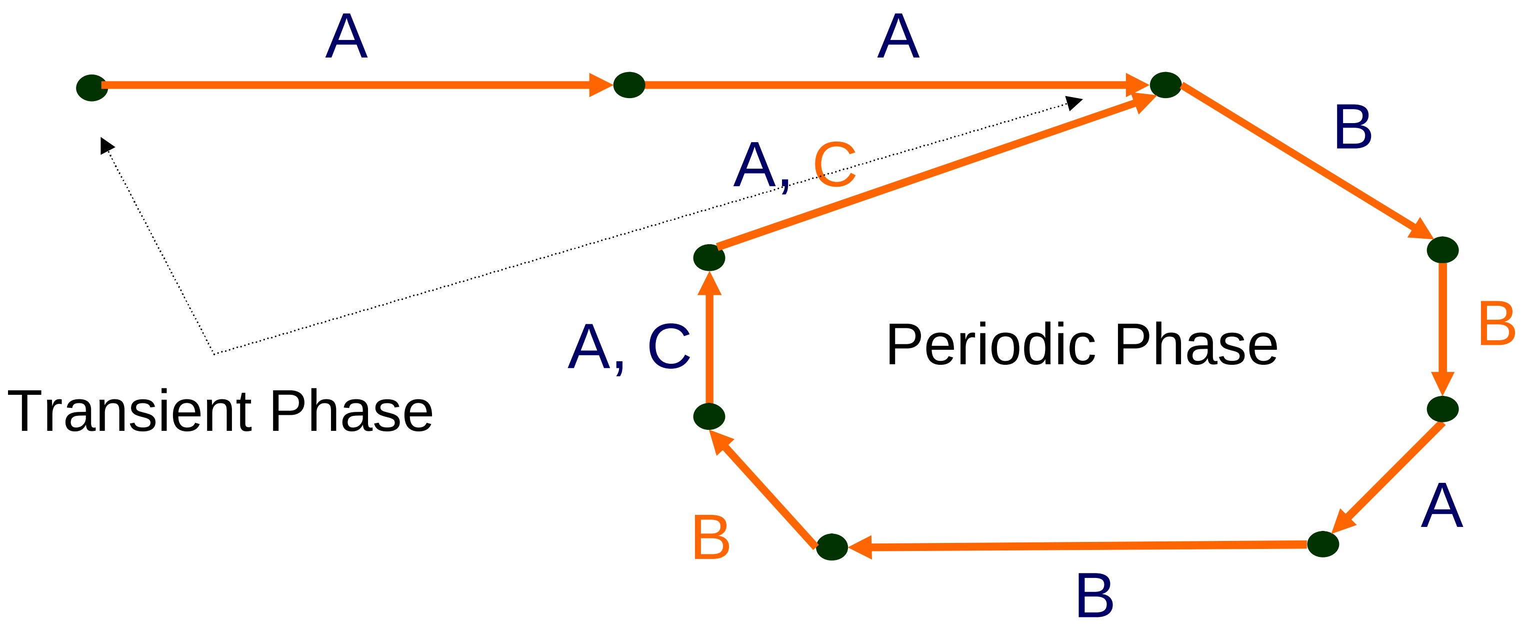
  Graph throughput: $\frac{Th(a)}{q(a)}.$

- Maximal throughput: self-timed execution

10 State Space of Self-timed Execution



State: distribution of tokens + remaining execution times



State-based Throughput Analysis

- Consistent and strongly connected SDFG, self-timed execution:

  Always a ***transient phase*** and a ***periodic phase***.

- Throughput analysis: ***Only periodic phase*** relevant

- Iteration: A ***set*** of firings in which each actor fires as many as indicated by the ***repetition vector***.

- Periodic phase: a ***whole number*** (possibly 0) of ***iterations***.

- Efficient implementation:

  Store one state for an arbitrary actor ***in each iteration***

12 Traditional Method for Throughput Analysis

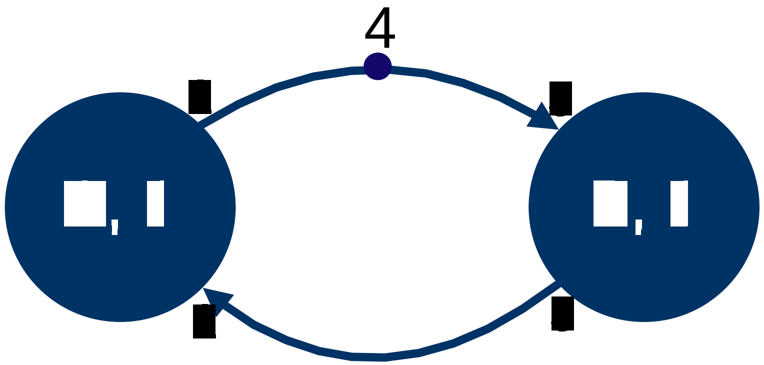
- Conversion of SDFG to Homogeneous SDFG (HSDFG)
- Compute throughput of the HSDFG

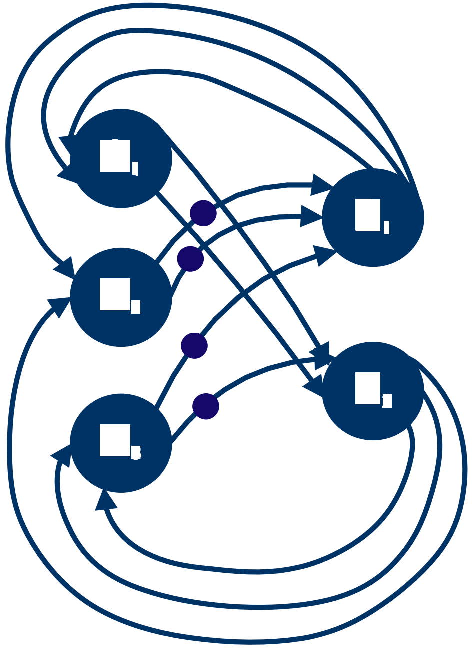
- Disadvantage:

  Conversion to HSDFGs may lead to an explosion in size of the graph

Traditional Method for Throughput Analysis

SDFG to HSDFG conversion





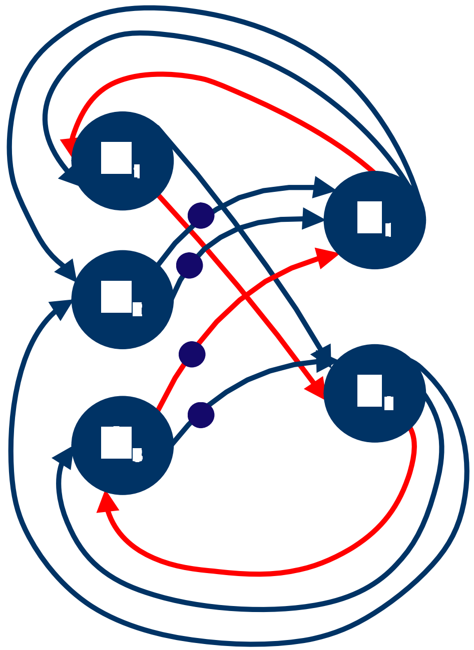
Repetition vector   q = {(A, 3), (B,2)}

Throughput of any SDFG equals the throughput of its equivalent HSDFG.

Traditional Method for Throughput Analysis

Theorem[Reiter66]: The throughput of an HSDFG can be calculated by:

$$\max_{cycles\, C} \frac{Weight\, of\ C}{number\, of\ tokens\ on\ C}$$



Efficient algorithms exist

15 **Experimental setup**

- Real applications
- Synthetic graphs
  - DSP-like
  - Large HSDFG
  - Long transient phase

16 Random Synthetic Graphs: DSP

	Average [s]
SDFG-HSDFG Conversion	47.92
Dasdan-Gupta	271.10^{-3}
Howard	1.10^{-3}
Young-Tarjan-Orlin	1.10^{-3}
Dasdan-Gupta (total)	48
Howard (total)	48
Young-Tarjan-Orlin (total)	48
State Space	$<1.10^{-3}$

Random Synthetic Graphs: Large HSDFG

Avg size: 8100

	Average [s]	Unsolved
SDFG-HSDFG Conversion	220	54
Dasdan-Gupta (MCM)	2	0
Howard (MCM)	9.10^{-3}	0
Young-Tarjan-Orlin (MCM)	8.10^{-3}	0
Dasdan-Gupta (total)	222	0
Howard (total)	220	0
Young-Tarjan-Orlin (total)	220	0
State Space	$<1.10^{-3}$	0

Random Synthetic Graphs: Long Transient Phase

	Average [s]
SDFG-HSDFG Conversion	-
Dasdan-Gupta (total)	252.10^{-3}
Howard (total)	250.10^{-3}
Young-Tarjan-Orlin (total)	250.10^{-3}
State Space	912.10^{-3}

Conclusions

- The first method to perform throughput analysis directly on SDFGs
- Experimental results show that our technique outperforms all other existing techniques
- Can be easily integrated in already existing simulation tools

Questions