Dataflow Analysis Revisited

Twan Basten
a.a.basten@tue.nl

‘Knowing is not understanding.’
Charles Kettering

2 Acknowledgements

Joint work with:
• Marc Geilen
• Sander Stuijk
• AmirHossein Ghamarian
• Bart Theelen
• …

Projects:
• NWO PROMES
• EC FP6 Betsy
• EC FP7 MNEMEE

3 Overview

• Embedded Multi-media
• Analysis of Synchronous Dataflow Graphs
  • Throughput Analysis
  • Throughput-Storage Trade-off Analysis
  • Parametric Throughput Analysis
  • Scenario-aware Dataflow
• Multiprocessor System-on-Chip Design
• Looking Forward

4 Embedded Multi-media

Trends
Concurrency
Connectedness
Interaction
Variability

emb. mm → throughput → storage → parameters → scenarios → mpsocs → the future
Embedded Multi-media

Approach: Models of Computation (MoCs)

on the interface between science and engineering

essential for predictability

<table>
<thead>
<tr>
<th>Basic dimensions</th>
<th>Aspects</th>
<th>Derivatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>processing</td>
<td>functionality, concurrency, timing, energy, quality</td>
<td>execution time, energy dissipation, perceived quality</td>
</tr>
<tr>
<td>communication</td>
<td>functionality, concurrency, timing, energy, quality</td>
<td>reconfiguration time, ...</td>
</tr>
<tr>
<td>storage</td>
<td>functionality, concurrency, timing, energy, quality</td>
<td>...</td>
</tr>
</tbody>
</table>
9 Essential Challenges

predictability and efficiency

... are contradictory

Kahn Process Networks (KPN)  
Synchronous Data Flow (SDF)  
Homogeneous SDF (HSDF)

10 Overview

- Embedded Multi-media
- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
    - Throughput-Storage Trade-off Analysis
    - Parametric Throughput Analysis
    - Scenario-aware Dataflow
  - Multiprocessor System-on-Chip Design
  - Looking Forward

11 Synchronous Data Flow Graphs

(SDFGs [Lee 1986])

Each actor fires as soon as it can fire
Known to maximize throughput

12 Self-timed Execution

Throughput: average number of actor firings over time
### Our Throughput Analysis Method

**State:** distribution of tokens + remaining execution times  

Throughput can be calculated from the periodic phase

**Transmit Phase**

Consider a subset of the firings of one actor only to detect a recurrent state

**Efficient implementation**

### Traditional Throughput Analysis

- Convert the SDFG to a Homogeneous SDFG (HSDFG)
- Compute the throughput of the HSDFG

Disadvantage: conversion may lead to an explosion in the size of the graph

### Throughput Analysis: Our Result

<table>
<thead>
<tr>
<th></th>
<th>State Space</th>
<th>Dasdan Gupta</th>
<th>Howard</th>
<th>Young</th>
<th>Tarjan</th>
<th>Orlin</th>
</tr>
</thead>
<tbody>
<tr>
<td>MP3 dec.</td>
<td>1. $10^{-3}$</td>
<td>1. $10^{-3}$</td>
<td>1. $10^{-3}$</td>
<td>1. $10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modem</td>
<td>1. $10^{-3}$</td>
<td>82. $10^{-3}$</td>
<td>81. $10^{-3}$</td>
<td>81. $10^{-3}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sample Rate</td>
<td>2. $10^{-3}$</td>
<td>&gt;1800</td>
<td>&gt;1800</td>
<td>&gt;1800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Satellite</td>
<td>56. $10^{-3}$</td>
<td>&gt;1800</td>
<td>&gt;1800</td>
<td>&gt;1800</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H.263 decoder</td>
<td>10. $10^{-3}$</td>
<td>&gt;1800</td>
<td>&gt;1800</td>
<td>&gt;1800</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Runtimes of various throughput analysis methods (in seconds)

### Conclusions Throughput Analysis

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

- Embedded Multi-media
- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
    - Throughput-Storage Trade-off Analysis
  - Parametric Throughput Analysis
  - Scenario-aware dataflow
- Multiprocessor System-on-Chip Design
- Looking Forward

Trade-off: Storage vs. Throughput

DAC 2006, IEEE T on Comp 2008

Storage Distribution

Tokens on channels must be stored in memory.

- Separate memory for each channel.
- Backward channels used to model storage space.

Problem Definition

Find all minimal storage distributions for any possible throughput.
### Design-Space Exploration Algorithm

Given an SDFG $G$ with storage distribution $\delta$

1. Compute throughput and abstract dependency graph $\Delta$ for $G$ with $\delta$
2. For each channel $c$ with a storage dependency in $\Delta$
3. Create new storage distribution by enlarging $c$
4. Repeat steps 1-2

**All minimal storage distributions are found when starting from storage distribution $\langle 0, \ldots, 0 \rangle$**

### Experimental Results

<table>
<thead>
<tr>
<th>Example</th>
<th>Satellite</th>
<th>MP3</th>
<th>H.263</th>
</tr>
</thead>
<tbody>
<tr>
<td>#actors</td>
<td>3</td>
<td>22</td>
<td>13</td>
</tr>
<tr>
<td>#channels</td>
<td>2</td>
<td>26</td>
<td>12</td>
</tr>
<tr>
<td>min throughput &gt; 0</td>
<td>1/7</td>
<td>0.18</td>
<td>1/137</td>
</tr>
<tr>
<td>Distr. size</td>
<td>6</td>
<td>1542</td>
<td>12</td>
</tr>
<tr>
<td>Max throughput</td>
<td>1/10000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distr. size</td>
<td>8006</td>
<td></td>
<td></td>
</tr>
<tr>
<td>#Pareto points</td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>#Distributions</td>
<td>5</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Exec. time</td>
<td>1ms</td>
<td>7ms</td>
<td>2ms</td>
</tr>
</tbody>
</table>
**Conclusions Trade-off Analysis**

- Exact minimum memory requirements for any possible throughput.
- Technique to search space of storage distributions efficiently.
- Experiments show feasibility of the approach.
- Technique can be combined with heuristics to prune the search space if necessary.

**Overview**

- Embedded Multi-media
- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
  - Throughput-Storage Trade-off Analysis
    - Parametric Throughput Analysis
  - Scenario-aware dataflow
- Multiprocessor System-on-Chip Design
- Looking Forward

**Parametric Throughput Analysis**

- H.263 decoder
- Divide and Conquer: divide space in throughput regions
- through expression
  \((\max (593 \text{ IQ} + 594 \text{ IDCT} + \text{ MC}, \text{ VLD} + 594 \text{ IQ} + 593 \text{ IDCT}))^{11}\)

**Overview**

- Embedded Multi-media
- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
  - Throughput-Storage Trade-off Analysis
    - Parametric Throughput Analysis
    - Scenario-aware dataflow
- Multiprocessor System-on-Chip Design
- Looking Forward
Scenario-aware H.263 Decoder Model
(FSM-based Scenario-Aware DataFlow (SADF))

Execution time

<table>
<thead>
<tr>
<th></th>
<th>VLD</th>
<th>IDCT</th>
<th>MC</th>
<th>FD</th>
</tr>
</thead>
<tbody>
<tr>
<td>P₀</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>others</td>
<td>40</td>
<td>17</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Parameterized rate

Detector

Tokens

x={30,40,50,60,70,80,99}

Rate

a | b | c | d | e
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>99</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Scenario-aware Performance Analysis

- SDF-based throughput analysis considers worst-case actor times

- Scenario-aware throughput analysis considers worst-case actor times per scenario, but it must consider scenario transitions

- Analyzing all scenario transitions separately can be avoided

- Separate scenario transitions with an invariant reference schedule

- For every scenario, our method allows to compute the bound on the completion time of an iteration

Overview

- Embedded Multi-media

- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
  - Throughput-Storage Trade-off Analysis
  - Parametric Throughput Analysis
  - Scenario-aware dataflow

- Multiprocessor System-on-Chip Design

- Looking Forward

SDF-based MP-SoC Compilation
33 SDF-based MP-SoC Compilation

Application SDF
Throughput constraint

(4 steps)
Memory dimensioning

(2 steps)
Constraint refinement
Platform graph

(4 steps)
Tile binding and scheduling
Interconnect graph

(3 steps)
Interconnect routing and scheduling
MP-SoC configuration

34 Horizon: Scenario-aware Flow

Application source code with timing constraints

Scenario identification and characterization
FSM-based SDF
Timing constraints

Memory dimensioning
Set of FSM-based SDFs
Timing constraints

Constraint refinement
Set of FSM-based SDFs
Timing constraints

Tile binding and scheduling
Set of FSM-based SDFs
Timing constraints

Interconnect routing and scheduling
Set of MP-SoC configurations / run-time manager

35 SDF3 Toolkit

www.es.ele.tue.nl/sdf3

36 Overview

- Embedded Multi-media
- Analysis of Synchronous Dataflow Graphs
  - Throughput Analysis
  - Throughput-Storage Trade-off Analysis
  - Parametric Throughput Analysis
- Scenario-aware Dataflow
- Multiprocessor System-on-Chip Design
- Looking Forward
### Dataflow MoCs Expressiveness Hierarchy

- **RPN**: Reactive Process Networks
- **KPN**: Kahn Process Networks
- **DF**: DataFlow
- **DDF**: Dynamic DF
- **BDF**: Boolean DF
- **CSDF**: Cyclo-Static DF
- **SDF**: Synchronous DF
- **HSDF**: Homogeneous SDF

- BDF and larger: turing complete
- Better notions of expressiveness needed

### Challenges

- Suitable notions of expressivity
- Expressivity while maintaining analyzability and synthesizability
- Abstraction without losing accuracy
- Composability and compositionality
- MoCs for non-functional aspects
- Unification of MoCs
- Multi-objective trade-off analysis
- Parametric analysis
- Model-driven design and synthesis flows
- Model-driven run-time systems

### Thank you!

Questions?

More info: [www.es.ele.tue.nl/~tbasten/](http://www.es.ele.tue.nl/~tbasten/)

> "An understanding of the natural world and what's in it is a source of not only a great curiosity but great fulfillment."
> David Attenborough