Efficient Buffer Capacity and Scheduler Setting Computation for Soft Real-Time Stream Processing Applications

Marco Bekooij, Maarten Wiggers, and Jef van Meerbergen
Outline

- Context: real-time stream processing
- Problem statement
- Cyclo-static dataflow representation of an execution trace
- Low complexity FIFO capacity computation algorithm
- Results of artificial test-cases
- Results of an H263 video decoder case-study
- Conclusion
Context

- Real-time stream processing on embedded multiprocessor systems
  - Smart phones
  - Car-radios

- Real-time requirements
  - Hard real-time = guarantee for all streams the throughput and latency
    - Use of worst-case execution times
    - Input and output behavior of tasks known at design time
    - Low resource utilization if variation in execution time of the tasks is high
  - Soft real-time = throughput and latency target
    - Execution times + correlation
    - Data dependent input/output behavior of tasks
    - *Improved resource utilization*
    - Throughput and latency is not guaranteed for every possible stream
    - Deadlock freedom cannot be guaranteed for every stream
Related work

- Throughput analysis for soft real-time applications
  - Simulation at different abstraction levels (CP-T, PV-T, cycle-true)
    - Trade-off between run-time and accuracy
  - Probabilistic analysis (Markov chain, Markov decision process)
    - Long running averages
    - Approximation of correlation to reduce run-time analysis algorithm
    - Undefined accuracy for short time-intervals

- This work: throughput synthesis for soft real-time applications
  - Compute FIFO capacities given throughput constraint
  - Assuming a predictable multiprocessor system
  - For one input stream, it is guaranteed that throughput constraint is met
Process network: H263 video decoder

**YAPI Process Network**
- No firing rules
- Unbounded FIFO buffers
- Untimed

**Dynamic Dataflow (DDF)**
- Implicit firing rules
- Bounded FIFO buffers
- Timed

---

**Nodes**
- VLD
- DQ
- IDCT
- MC

**Connections**
- Motion vectors
- Macro blocks

---

**Symbols**
- Implicit firing rules
- Timed
- Untimed
- Node 1
- Node 2
- Node 3
- Node 4

---

**Logos**
- NXP
YAPI process

while(1){
    ...
    read(FIFO1,x);
    ...
    if(x>0){
        write(FIFO2,z);
    }
}

trace

Cyclo-static dataflow actor

- Firing rules explicit
- Correlation fully captured

Input of buffer computation algorithm!
## Buffer capacity computation

### Run-time(number_of_phases)

<table>
<thead>
<tr>
<th>phases</th>
<th>$10^2$</th>
<th>$10^3$</th>
<th>$10^4$</th>
<th>$10^5$</th>
<th>$10^6$</th>
<th>$10^7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>exact</td>
<td>0.05s</td>
<td>2.09s</td>
<td>218s</td>
<td>&gt;2h</td>
<td>&gt;2h</td>
<td>&gt;2h</td>
</tr>
<tr>
<td>approx</td>
<td>0.02s</td>
<td>0.09s</td>
<td>0.79s</td>
<td>7.7s</td>
<td>78s</td>
<td>780s</td>
</tr>
</tbody>
</table>

Linear complexity in the number of phases: $O(|V|^4 + |V|*|E|*phases)$
Processor sharing (time division multiplex)

\[ \rho(v_i,f) = \phi(v_i,f) + (p-s_i) \left\lceil \frac{\phi(v_i,f)}{s_i} \right\rceil \]

- \( f \) = phase
- \( v_i \) = i-th actor
- \( p \) = time division multiplex period
- \( s_i \) = i-th slice length
- \( \phi(v_i,f) \) = execution time with stalls
- \( \rho(v_i,f) \) = response time

No sharing then response time = execution time
Predictable multiprocessor architecture

- Time division multiplex (TDM) arbitration for every shared resource
- Guaranteed throughput connections in network
  - guaranteed min bandwidth, max latency, in-order delivery, flow control
- A processor can only access the memory in its tile
  - communication assist copies data between network interface FIFO and MEM
Conservative execution time estimate

Execution time ($\phi(v_i,f)$) of a non-blocking code segment including stall cycles:

$$\phi(v_i,f) = \zeta(v_i,f) + \lceil \text{accesses}/9 \rceil \times 10$$

Execution time without stalls (measured with e.g. delay statements):
FIFO capacity computation example
Stretched schedule

Monotonic execution: earlier production ⇒ earlier enable ⇒ earlier production
Stretched schedule

\[ \beta_{pc} = 1 \]

\[ \text{cap} = 2 \text{ tokens} \]
Multiple distance constraints

\[ \beta_{01} = 1 \rightarrow 2 \]

\[ \beta_{02} = 1 \]

\[ \beta_{21} = 1 \]

min-cost max-flow
Capacity buffer $v_0 \rightarrow v_1$

cap=3 tokens

$\beta_{01} = 2$
Artificial test-case: varying response-time

\[ x = \{1,1,...,1\} \]
\[ y = \{1,1,...,1\} \]
\[ r = \{10,4,...,9\} \]
\[ \theta(v_p) = 1000 \]
\[ 1 \leq x_i \leq 10, \text{ uniform distribution} \]
Varying response-time: capacity(MCM)

Throughput = $1 / \text{MCM}$

Similar for varying input output behavior
H263 video decoder case-study

YAPI Process Network

VLD → DQ → IDCT → MC

DDF

VLD → DQ → IDCT → MC

Communication latency network was not taken into account
### FIFO_capacity(desired_MCM)

<table>
<thead>
<tr>
<th>Desired MCM (52*10^6)</th>
<th>n1 VLD→DQ</th>
<th>n2 DQ→IDCT</th>
<th>n3 IDCT→MC</th>
<th>n4 VLD→MC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>86</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>2</td>
<td>8</td>
<td>85</td>
</tr>
<tr>
<td>2</td>
<td>5</td>
<td>2</td>
<td>12</td>
<td>78</td>
</tr>
<tr>
<td>1.5</td>
<td>6</td>
<td>2</td>
<td>43</td>
<td>166</td>
</tr>
<tr>
<td>1</td>
<td>8</td>
<td>2</td>
<td>468</td>
<td>984</td>
</tr>
</tbody>
</table>

Phases = 143010 phases, run-time = 14 sec
Linear bounds not tight ⇒ large buffer

Maximum throughput
Desired MCM=52 *10^6
Conclusion

- Presented a buffer capacity computation technique for soft real-time applications that are described as YAPI process networks
  - Given a throughput constraint
    - Conservative capacities for one stream $\Rightarrow$ correlation is fully taken into account

- Execution trace is encoded in phases of CSDF actors
  - Potentially a large number of phases
  - Low complexity algorithm for computation of conservative buffer capacity estimates
    - Less tight results if desired throughput gets closer to maximum throughput

- H263 video decoder case-study
  - Larger buffer if desired MCM $\Rightarrow$ minimum MCM
  - Run-time was 14 seconds for a CSDF with in total 143010 phases
Questions?
Artificial test-case: varying rates

\[ y = \{1,1,\ldots,1\} \]
\[ r = \{1,1,\ldots,1\} \]
\[ x = \{2,4,\ldots,1\} \]
\[ \theta(v_p) = 1000 \]
\[ 1 \leq x_i \leq 4, \text{ uniform distribution} \]
Varying rates: capacity (MCM)