A Dataflow Model for Interactive Data-dependent Streaming Applications

Peter Poplavko, Pascal Fradet, Alain Girault
INRIA, Grenoble, France

Ali-Erdem Ozcan
ST Microelectronics Inc., Ottawa, Canada

Abstract. The modern streaming applications – e.g. video surveillance systems – are interactive and data-dependent. They respond to the events from the user and the environment and vary their workload depending on the motion and details in the visual scene. These applications are characterized by virtually infinite number of use cases due to dynamic combination of various coding and filtering algorithms.

The existing dataflow models of computation cannot suitably capture all the system use cases in one model. Other models are either not suitable for streaming applications or cannot be analyzed statically for absence of deadlock and bounded memory.

To fill this gap we present a new dataflow model. On one hand, it extends the advanced features of other statically analyzable dataflow models such as heterochronous dataflow [HDF], parametrical synchronous dataflow [PSDF], and variable-rate dataflow [VRDF]. On the other hand, being at least as simple and intuitive as any of them, it can still model an infinite number of use cases. It also enables efficient quasi-static scheduling and analysis for deadlock and consistency. We describe our model and give practical examples.

In future we plan to develop the methods to manage the resource usage under timing constraints and to work on scheduling techniques for embedded multiprocessors, e.g. using scenario-based approach [SCEN].

References


A Dataflow Model for Interactive Data-dependent Streaming Applications

Peter Poplavko, Pascal Fradet, Alain Girault
INRIA, Grenoble, France
petro.poplavko@inria.fr

Ali-Erdem Ozcan
ST Microelectronics Inc., Ottawa, Canada

Our Model = NDF + Transformations

NDF = Nested Dataflow - (reflects nested-loop structure)

variable communication

NDF > VRDF

- product operator
  \( P \circ Q = \sum_{p=1}^{P} Q(p) \)
  same meaning as multiplication for balance equations
- any node can set any parameter provided there is no deadlock, e.g.

NDF Features

- efficient static analysis
- quasi static scheduling: \((IN) A^\circ B^\circ(OUT)\)
- parameter can potentially change at every firing: e.g.

Transformation Rules (Example)

Transformation Rule 1: Add a Splitter

X

Transformation Rule 2: Add a VOD application

X

Transformation Rule 3: Add Video Scaling

X

Transformation Rule 4: Remove Application

X

Use Case 1: One VOD is running, no scaling

We obtain Use Case 2 from Use Case 1 by applying the following rules:

Use Case 2: Two VODs are running, one of them being scaled