Mapping and Scheduling of Dataflow Graphs - A Systematic Map

Mirza, Usman Mazhar; Arslan, Mehmet Ali; Cedersjö, Gustav; Sulaman, Sardar Muhammad; Janneck, Jörn

2014

Mapping and Scheduling of Dataflow Graphs - A Systematic Map

Usman Mazhar Mirza, Mehmet Ali Arslan, Gustav Cedersjo, Sardar Muhammad Sulaman and Jorn W. Janneck
Dept. of Computer Science, Lund University, 221 00 Lund, Sweden
{Usman_Mazhar.Mirza, Mehmet_Ali.Arslan, Gustav.Cedersjo,
Sardar, Jorn.Janneck}@cs.lth.se

Abstract—**Dataflow is a natural way of modelling streaming applications, such as multimedia, networking and other signal processing applications. In order to cope with the computational and parallelism demands of such streaming applications, multiprocessor systems are replacing uniprocessor systems. Mapping and scheduling these applications on multiprocessor systems are crucial elements for efficient implementation in terms of latency, throughput, power and energy consumption etc. Performance of streaming applications running on multiprocessor systems may widely vary with mapping and scheduling strategy. This paper performs a systematic literature review of available research carried out in the area of mapping and scheduling of dataflow graphs.**

I. Introduction

Modern embedded multimedia systems support many different applications. These systems are required to maintain certain performance (for example, throughput in audio and video processing, networking and various other signal processing applications) while keeping a tight power and energy budget. Such applications are today ubiquitous and become increasingly demanding. In order to meet the computational and parallelism demands of these sort of streaming applications, multi and many core platforms or multiprocessor system on chip (MPSoC) are now replacing uniprocessor systems. Therefore, the race for faster clock has turned into the race for more cores, for more parallelism. A large variety of MPSoC target architectures is currently available or under development. These target architectures are comprised of different types of homogeneous/heterogeneous computing nodes and interconnected structures (e.g. buses and network-on-chip).

Dataflow is a natural way of modelling streaming applications, with its synchronous (SDF), cyclo-static (CSDF) and dynamic (DDF) flavours [1][2]. This way of modelling may exhibit the parallelism inherent in such applications, composed of modules (actors) that can run in parallel, only interacting through FIFO buffers (channels).

Even though dataflow programming paradigm lets the programmer express parallelism explicitly that does not necessarily entail better efficiency. The application has to be run on a target platform, which in turn means that pieces of the program have to be mapped to processing and communication elements. The performance of an application may widely vary with the mapping depending on the type of processing nodes running the actors, and the type of hardware implementing the channels. Even after a decent mapping is found, bits of the program i.e. actions of the actors has to be scheduled correctly and cleverly to result in an efficient implementation. Efficiency has many aspects, such as execution time, throughput, power and energy consumption, communication, etc... Mapping and scheduling are intertwined and crucial elements that affect the efficiency in all these aspects. That is why, it is interesting to explore that what has been done so far in this research area and to find the research gap in this area.

In this article we present a systematic overview of available research carried out on mapping and scheduling of dataflow graphs. A systematic literature review or mapping study is used for the evaluation and interpretation of all available research relevant to a particular research question or area of interest. The review presented in this study is a systematic mapping study, conducted based on the guidelines presented in [3].

The main objective of this study is to find out what research has been carried out in this area. Later we divide the main objective into number of research questions. In addition to the overview of the mapping and scheduling of dataflow graphs, a categorization of the mapping and scheduling of dataflow graphs on different processing architectures is also presented.

Section 2 presents related work in the field of systematic literature review of mapping and scheduling of dataflow graphs. Section 3 presents a brief background about the dataflow graphs and their types. Section 4 discusses the methodology used in this study in detail. Section 5 contains the results of this mapping study and Section 6 presents the validity assessment. Finally, Section 7 concludes this study.

II. Related Work

To the best of our knowledge this paper is the first attempt at a systematic literature review of mapping and scheduling dataflow graphs. In this section we mention two other studies that, even though are not systematic reviews or entirely about dataflow graphs, are related to our work. There are 18 years between these two studies, which indicates a need for more studies on systematic mapping and review in this field.

In [4], Blindell presents an extensive survey on instruction selection. He examines and summarizes a large number of techniques, tackling instruction selection in its various aspects.

Part of this survey covers graph-based approaches, which can be related to dataflow graphs to some extent.

The work by Ahmad et al. [5] surveys algorithms that map and schedule a parallel program represented as a dataflow graph to a set of homogeneous processors, optimizing the completion time. The algorithms are classified according to the number of processors available (*bounded/unbounded*), network topology, and the type of task duplication algorithm they use. The paper presents experiments and comparisons of these algorithms in different scheduling settings. The main difference with our study is that we consider mapping and scheduling of dataflow programs on any architecture types.

III. Background

A dataflow graph is a directed graph with nodes denoting computational kernels (*actors*) and edges denoting data communication (which are usually referred to as *tokens*) between the actors. These are generally implemented as first in, first out (FIFO) buffers to reflect the streaming nature of the application. Each actor has one or more actions that can consume and/or produce tokens. These actions can only be fired when the input tokens they require are present in the respective buffers. Several actors can be grouped together (as in Fig. 1) to construct hierarchically structured dataflow graphs.

Running a program that is represented as a dataflow graph includes at least two steps, namely *mapping* and *scheduling*.

Mapping signifies the process of assigning actors to processing elements (PE) that can run them. This set of decisions can affect performance in almost every aspect. Execution times of actors may differ from PE to PE in heterogeneous systems; in homogeneous systems they can have a limited amount of run-time assigned to them periodically, because of other actors that are mapped to the same PE. Communication between actors is another cost factor that should be taken into account and is also directly affected by mapping. Depending on when it is carried out, mapping can be *static* or *dynamic*. If the actors are mapped before run-time and are not migrated in run-time, the mapping is static. Contrarily, if the mapping decisions are carried out or altered in run-time, the mapping is classified as dynamic.

Scheduling, on the other hand, is the process of ordering the actions of actors, on the PEs they are mapped. Scheduling mainly affects throughput, latency, memory and buffer requirements. The static versus dynamic distinction can be made for scheduling as well. Some types of dataflow graphs (synchronous dataflow (SDF)[1], cyclo-static dataflow (CSDF)[2]) allow scheduling before run-time while others do not.

IV. Methodology

This paper presents a review of available research carried out in the area of mapping and scheduling of dataflow graphs. The work is carried out as a systematic literature review, based on the guidelines presented in [3]. The research objective of this paper is to present an overview of different mapping and scheduling techniques for dataflow graphs, their target architectures, optimization criteria, and used optimization and evaluation methods.

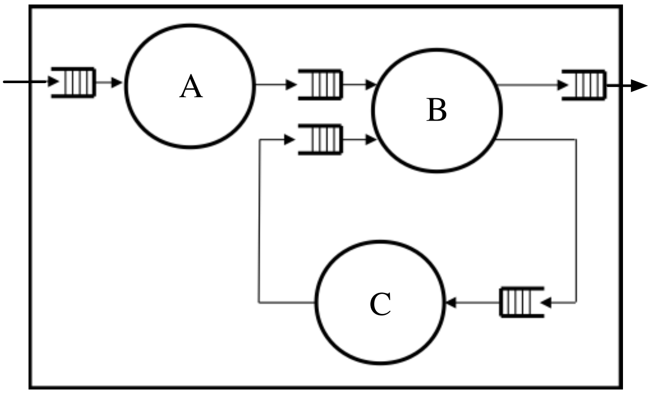


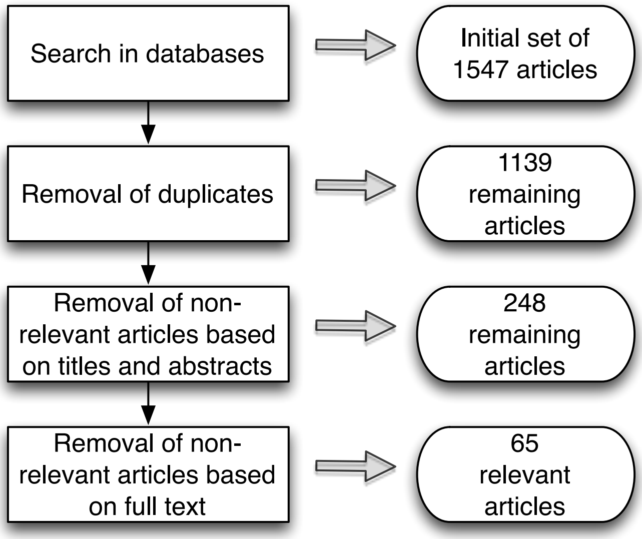
Fig. 1. A three-node dataflow graph. Nodes are connected with FIFO buffers.

In the initial phase of the study, a review protocol was developed. The protocol consists of research questions, search strategy, selection criteria and procedure, validity assessment, extraction of data, and data synthesis strategies.

This work is carried out as a planned study and following sequential steps were performed:

- Defining the research questions.
- Selection of sources for search of available relevant articles.
- Defining the search query and performing the search on the sources selected in the previous step – (Resulted in 1547 articles).
- Removing the duplicate articles using EndNote reference manager and by manual search – (Resulted in 1139 remaining articles).
- Defining the inclusion and exclusion criteria and initial selection based on titles, keywords, and reading abstracts according to the defined inclusion/exclusion criteria – (Leaving 248 remaining articles).
- Final selection of the articles based on reading of full text of the articles carefully according to the same criteria defined in previous step – (Resulting in 65 final articles: for this work we limit our scope to SDF graphs only because it would be impossible to cover mapping/scheduling research for all type of dataflow graphs in review like this, due to large number of relevant work done in this area).
- Analysis of the results for the classification of the mapping and scheduling techniques presented in the final list of articles.

The steps involved in the selection of the relevant articles for this study are summarized in Fig. 2. Each step is described in detail in the following subsections and during each step special measures were taken to improve the research validity.



Search in databases
Initial set of 1547 articles
Removal of duplicates
1139 remaining articles
Removal of non-relevant articles based on titles and abstracts
248 remaining articles
Removal of non-relevant articles based on full text
65 relevant articles

Fig. 2. Identification and selection of articles.

A. Research questions

As described already, that the objective of this work is to present an overview of different available mapping and scheduling techniques for dataflow graphs. We broke down this general goal to the following research questions:

- What mapping and scheduling techniques are available for dataflow graphs in the research literature?
- What different kind of architectures are targeted by the researchers for mapping and scheduling of dataflow applications?
- What is the mostly targeted optimization criteria of the available mapping and scheduling techniques?
- What are the different methods used for solving the problem?

B. Search strategy

1) Searched resources: The following databases were searched (through Engineering Village[1]) for relevant research:

- INSPEC: This database is provided by Elsevier Engineering Information Inc. and the Institute of Electrical Engineers (IEE). It includes articles from 1969 to present.
- COMPENDEX: This database is provided by Elsevier Engineering Information Inc. It includes papers from 1969 to present.

The above mentioned databases provide a broad coverage of the area of interest, i.e., “mapping and scheduling of dataflows”, and they include articles from the main conferences, journals, and publishers (IEEE, ACM, Springer, etc.).

2) Search query: After a number of iterations, the following search query was considered a good compromise between finding as many of the relevant articles as possible, and returning a manageable number of results:

```
(((mapping WN all) OR
(scheduling WN all))
AND
((dataflow WN all) OR
(data?flow WN all) OR
(static dataflow WN all) OR
(static data?flow WN all) OR
(synchronous dataflow WN all) OR
(synchronous?data?flow WN all) OR
(cyclo?static dataflow WN all) OR
(cyclo?static data?flow WN all) OR
(SDF WN all) OR
(CSDF WN all) OR
(DDF WN all)))
```

The search string has two main parts separated by an AND clause. The first part of the search string excludes articles that are not about ‘mapping’ or ‘scheduling’. The second part of the search string excludes articles that do not discuss dataflow. It also includes different synonyms used for the term dataflow. The ‘?’-character is a wildcard representing one character, included because we want to identify both ‘-’ and ‘ ’.

C. Inclusion and exclusion criteria

When articles were identified using the search string, presented in previous subsection, from the databases, it was needed to identify and remove the duplicate articles. This was accomplished using EndNote reference manager and by manual search for the duplicate articles. At this stage, it was necessary to manually remove the non-relevant articles from the selection. This was carried out first based on the titles, keywords, and abstracts, and finally based on careful reading of the full text. As described earlier, the inclusion and exclusion criteria were defined during the design of the review protocol. The manual selection of articles was done based on this inclusion and exclusion criteria that is as follows:

- Articles not about mapping/scheduling of dataflow graphs were excluded from the selection.
- Articles presenting the work about design of dataflow architectures/machines were excluded from the selection.
- Articles about modelling of concurrent applications were excluded from the selection.
- Articles about graph transformation techniques were excluded from the selection.
- Articles about dataflow languages and their semantics were excluded from the selection.
- Articles about hardware synthesis were excluded from the selection.
- Articles about task graph mapping/scheduling were excluded from the selection.
- Articles not about SDF graphs were excluded from the selection – (This exclusion criteria is just to limit the scope of this paper, for future work we want to extend this work for all type of dataflow graphs).

The final criterion was needed to limit the scope of this mapping study. It would be impossible to cover map-

[1] http://www.engineeringvillage2.org

ping/scheduling research for all type of dataflow graphs in one review like this, because of the large number of relevant work done in this area.

D. Selection of relevant articles

The search query presented in Section IV-B retrieved 1547 articles when searched on $5^{th} August$ 2014, and we decided to continue systematic review with these results. For these results, the titles, author names, keywords, and abstracts were downloaded for the initial selection of the relevant articles. From this initial list, 408 duplicate articles were found and removed using EndNote Reference manager and by manual search.

In each step of the selection process (based on the titles, keywords, and abstracts, and finally based on careful reading of the full text), the criteria presented in previous subsection were used to manually remove the non-relevant articles. This resulted in the following three groups of articles at each selection step:

- **Relevant:** Articles that clearly fulfilled the criteria presented in previous subsection.
- **Non relevant:** Articles that are out of the scope of this study according to the selection criteria.
- **Possibly relevant:** Those articles for which there is not enough information to determine that those are relevant or not. This list was rechecked by the co-authors for the selection. After the reselection (based on titles, keywords, and abstracts), the selected articles were then included in the relevant articles for further selection in the next step.

The first effort to remove the irrelevant articles was based on the titles, keywords and the abstracts. This selection resulted in a list of 248 relevant articles (this selection includes the reselection step by the co-authors for the possibly relevant articles as explained above).

For the final step of selection process, the full text of the remaining relevant articles was downloaded. After the careful reading of the full text of the selected relevant articles (selected in the previous selection step), 65 articles were found relevant and were included in the final list for the review. The final list of 65 relevant articles can be found at http://goo.gl/G1e3bR.

E. Data extraction and synthesis

In the final step of the selection, i.e., selection based on full text of the articles, number of relevant attributes were also extracted from each of the articles with respect to research questions presented in Section IV-A. The mapping and scheduling techniques were classified as static and dynamic and there were number of different target architectures. The results based on extracted data of the selected articles are discussed in the next section.

V. Results

In this section we present the results of this study. Fig. 3 shows a histogram over the publication year of the papers included in the final review.

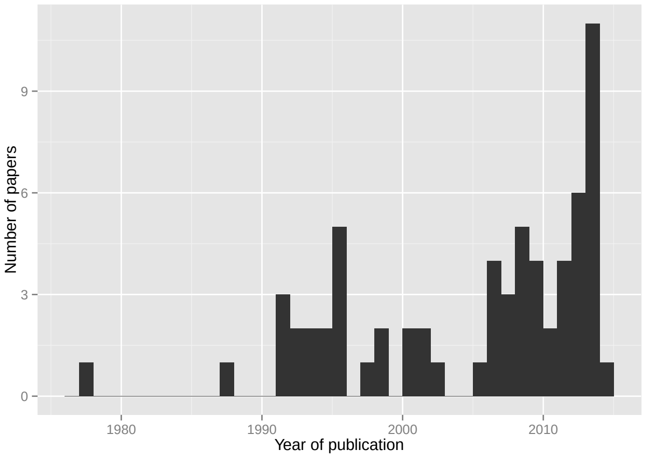


Fig. 3. Histogram over the publication year of the papers included in the final review.

TABLE I
Target architectures

Architecture	Number of papers
Single	17
Homogeneous	22
Heterogeneous	14
Other	4

A. Mapping and scheduling techniques

To get an overview of the first question, “What mapping and scheduling techniques are available for dataflow graphs in the research literature?”, we studied when mapping and scheduling decisions were made, if they were made at design-time or compile time versus at runtime. There were 29 papers where mapping of actors to the resources was done before runtime, and 5 papers where the mapping was at least partially made at runtime. There were 45 papers that used static scheduling, and 12 papers that used dynamic scheduling.

B. Target architectures

To answer the second question, “What different kind of architectures are targeted by the researchers for mapping and scheduling of dataflow applications?”, we categorized the architectures in four categories: *single processor*, *homogeneous processor architecture*, *heterogeneous architecture*, and *other*. Table I shows the number of paper targeting each category of processor architectures.

C. Optimization criteria

The third question was “What is the mostly targeted optimization criteria of the available mapping and scheduling techniques?”. The two most targeted criteria are to minimize *data memory size* and to maximize *throughput*. Minimizing data memory size includes all kinds of techniques for reducing the size of the data memory, including, most notably, buffer minimization but also techniques for sharing buffer memory. Table II shows the number of papers that addresses the most common optimization objectives.

TABLE II
OPTIMIZATION CRITERIA

Optimization criterion	Number of papers
Data memory size	24
Throughput	23
Program memory size	11
Latency	8
Energy	5
Cost	3
Resources	3

TABLE III
SOLVING METHOD

Technique	Number of papers
Evolutionary algorithms	8
Constraint programming	6
Integer linear programming	5
Satisfiability solving	2

D. Solving method

To answer the fourth question, “What are the different methods used for solving the problem?”, we identified general methods for solving optimization problems. Table III shows the most common techniques.

VI. VALIDITY ASSESSMENT

The main objective of this research is to summarize the available research in the field of mapping and scheduling of dataflow graphs. An important threat to the validity of this study is that it cannot be guaranteed that all possible relevant articles in this field have been included in the study. First of all, only research published in English was included for practical reasons. Secondly, some lesser known journals or conferences are not available in the searched databases, and were therefore not searched in this study. Also, articles for which the full text was not available were excluded from this study. This mostly affects older articles. Thirdly, it is likely that some relevant articles were rejected by the search string, since it is impossible to define a search string that finds absolutely all relevant articles without returning an unmanageable number of false positives. Finally, it is of course also possible that relevant articles were incorrectly rejected during the manual selection process from over one thousand articles to the final selection list of articles.

In order to reduce the risk of incorrect rejection of an article during the selection process, the co-authors of this article cross-checked the selection in each step. Whenever there was doubt about whether to include an article or not, the article was retained for the next step of the selection process. After initial selection process based on the title and keywords, the second author of this article cross checked 100 randomly selected articles from the initial list, and suggested a few additions and removals of articles.

After the second selection process based on titles, keywords and abstracts (performed by the first author), the resulted list of articles was divided among first three authors of this study for the election and the extraction of data.

Whenever there was a doubt in selection of an article it was retained for the next step, where more information was available to decide the relevance of an article with more accuracy.

By taking the above mentioned measures for the validity of this study we are more confident that most of the relevant articles for this study have been identified and included in the final list of articles.

VII. CONCLUSIONS

Based on this literature review of available research in the area of mapping and scheduling of SDF graphs, it can be concluded that most articles focused on static mapping and/or scheduling techniques. Only few papers focused on dynamic mapping and/or scheduling techniques. Further, it can be concluded that most research was concerned about minimizing the memory requirements and maximizing the throughput. Also available work equally targeted the uniprocessor and multiprocessor platforms.

Based on our literature review of available research we conclude that there is a need to conduct research about quantitative comparison of the available mapping and scheduling techniques for dataflow graphs. This can be interesting to see that how different techniques performs in comparison to each other. This study presents the systematic literature review of SDF graphs only and in future we plan to extend this work for all type of dataflow graphs.

REFERENCES

[1] E.A. Lee and D.G. Messerschmitt. Synchronous data flow. *Proceedings of the IEEE*, 75(9):1235 – 1245, sept. 1987.

[2] J.T. Buck and E.A. Lee. Scheduling dynamic dataflow graphs with bounded memory using the token flow model. In *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, volume 1, pages 429 –432 vol.1, april 1993.

[3] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. *Technical Report Keele University and University of Durham*, 2.3, 2007.

[4] Gabriel Santin Hjort Blindell. Survey on instruction selection: An extensive and modern literature study. Technical report, Stockholm, Sweden, October 4 2013. ISBN: 978-91-7501-898-0.

[5] Ishfaq Ahmad, Yu-Kwong Kwok, and Min-You Wu. Analysis, evaluation, and comparison of algorithms for scheduling task graphs on parallel processors. pages 207 – 213, Beijing, China, 1996. Arbitrary process network (APN) scheduling algorithms;Bounded number of processors (BNP) scheduling algorithms;Directed acyclic graph (DAG);Task duplication based (TDB) scheduling algorithms;Task graphs;Unbounded number of clusters (UNC) scheduling algorithms;.