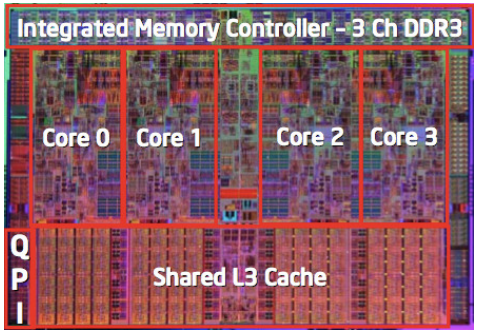
Satisfying Dataflow Programs Constraints on Multicore Architectures

Citi lab PhD day 2014

Manuel Selva
Supervisor: Lionel Morel
Director: Stéphane Frénot
Bull: Frédéric Soinne

27th March 2014

More and more parallelism



Nehalem multicore die
(source: www.intel.com)

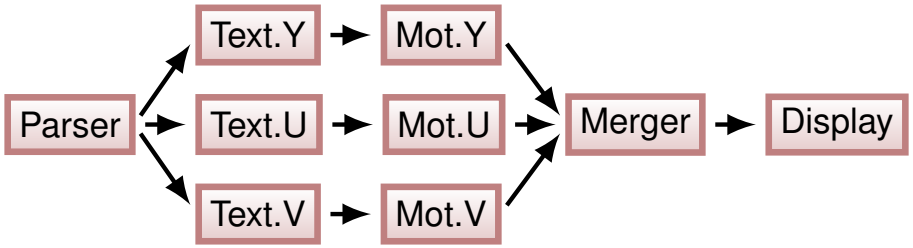
Apple/ARM dual core
(source: www.cultofmac.com)

Multiprocessor motherboard
(source: www.bit-tech.net)

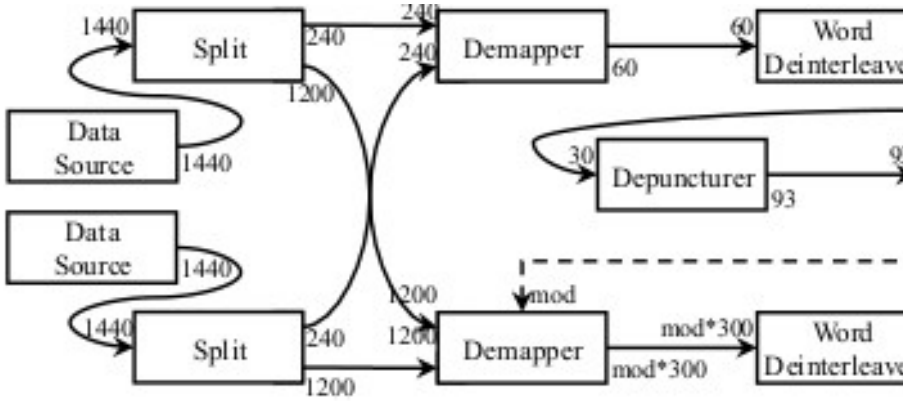
How to program ?

- Threads (Java, C + Pthreads)
- Annotations to sequential code (OpenMP)
- **Dataflow**

Why and what is dataflow programming ?

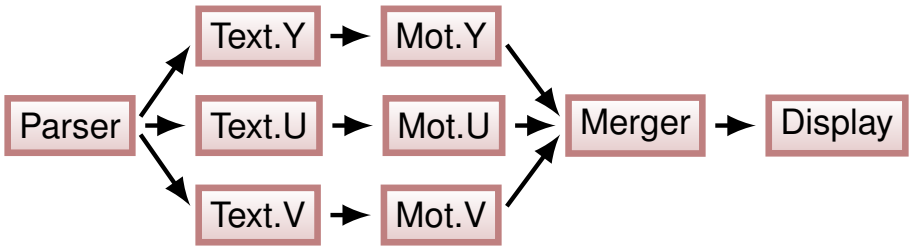


(a) H264 decoding

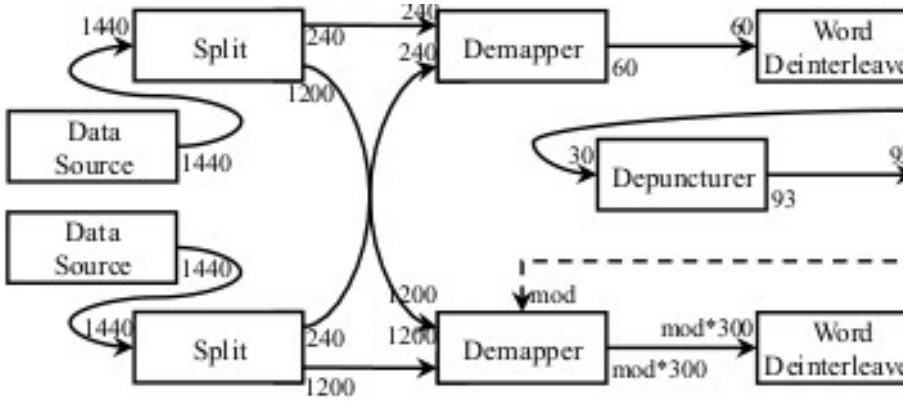


(b) LTE-Adv decoding

Why and what is dataflow programming ?



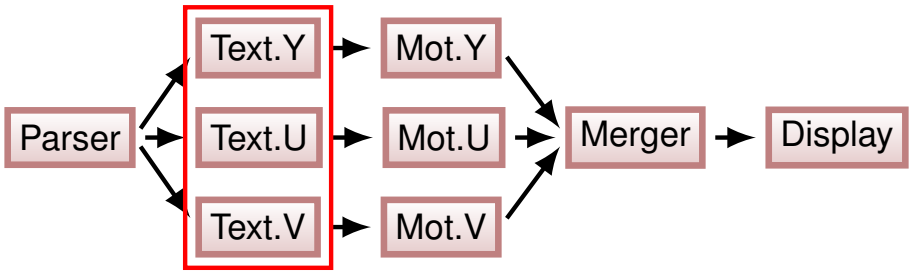
(a) H264 decoding



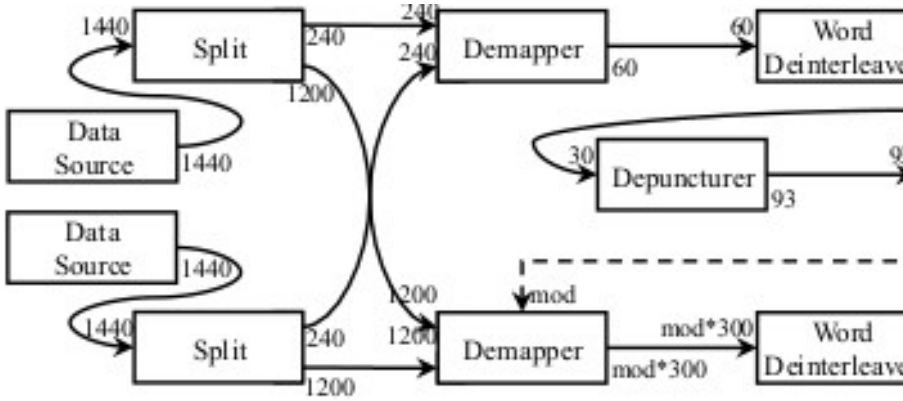
(b) LTE-Adv decoding

- Different kinds of parallelism
  - Actors exchanging data **only** through FIFO channels

Why and what is dataflow programming ?



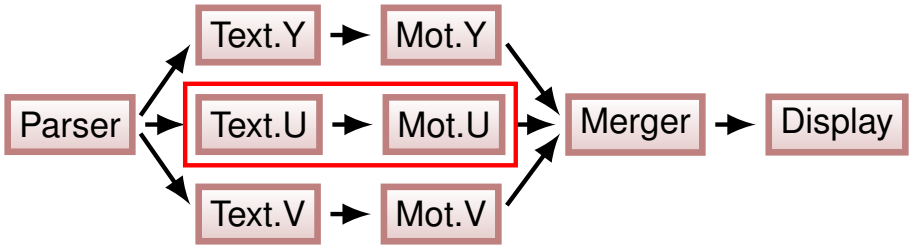
(a) H264 decoding



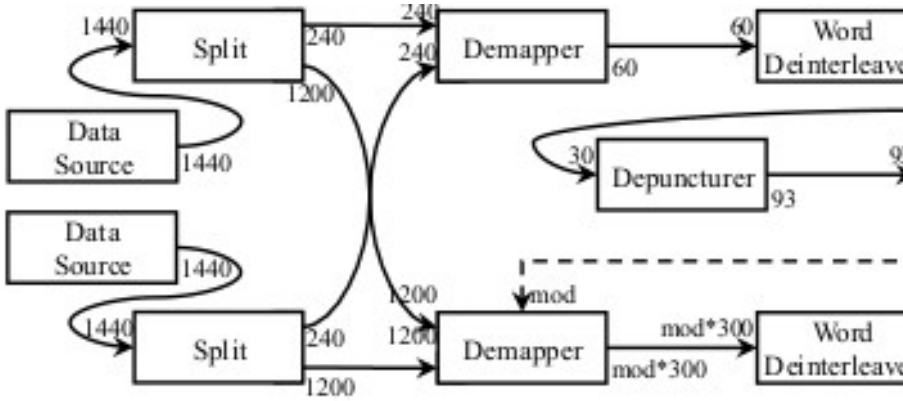
(b) LTE-Adv decoding

- Different kinds of parallelism
  - Actors exchanging data **only** through FIFO channels
  - Task

Why and what is dataflow programming ?



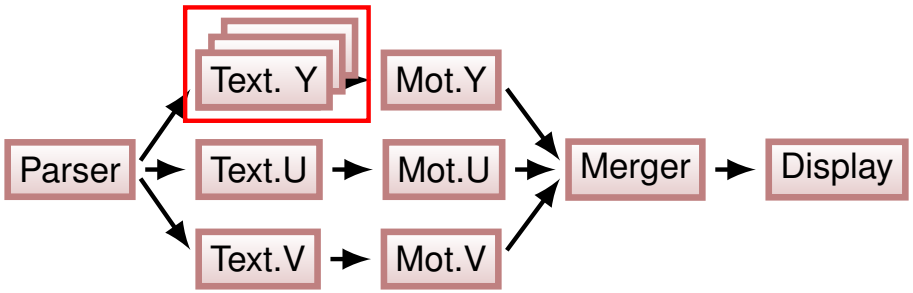
(a) H264 decoding



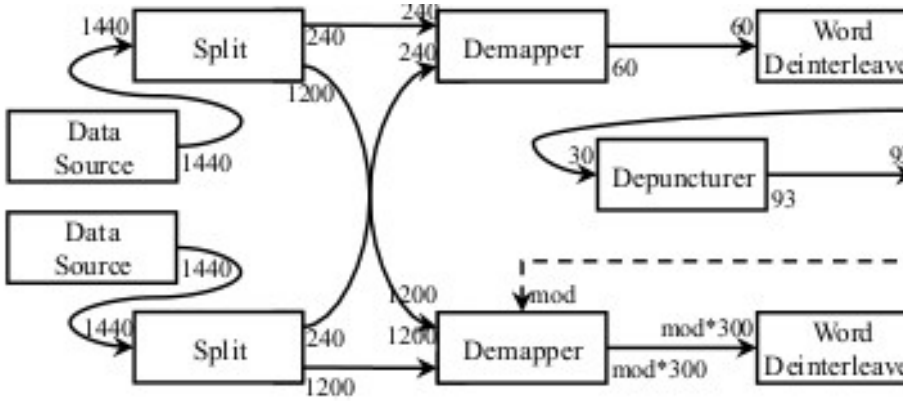
(b) LTE-Adv decoding

- Different kinds of parallelism
  - Actors exchanging data **only** through FIFO channels
  - Task , pipeline

Why and what is dataflow programming ?



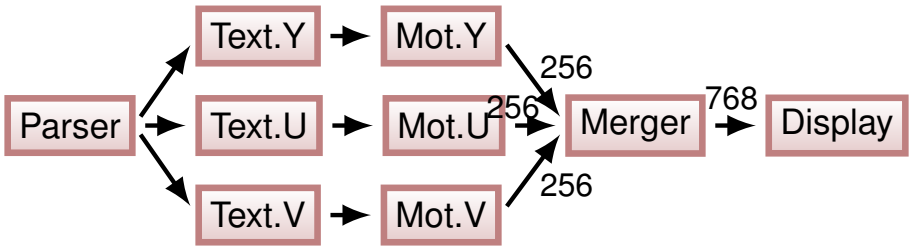
(a) H264 decoding



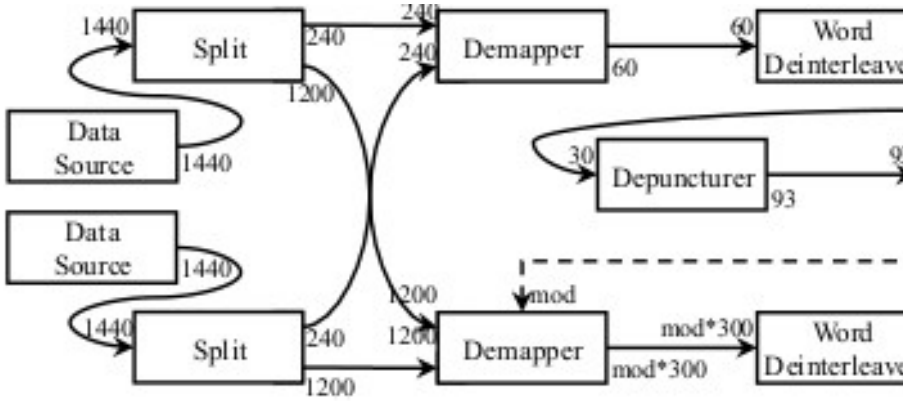
(b) LTE-Adv decoding

- Different kinds of parallelism
  - Actors exchanging data **only** through FIFO channels
  - Task , pipeline , data

Why and what is dataflow programming ?



(a) H264 decoding



(b) LTE-Adv decoding

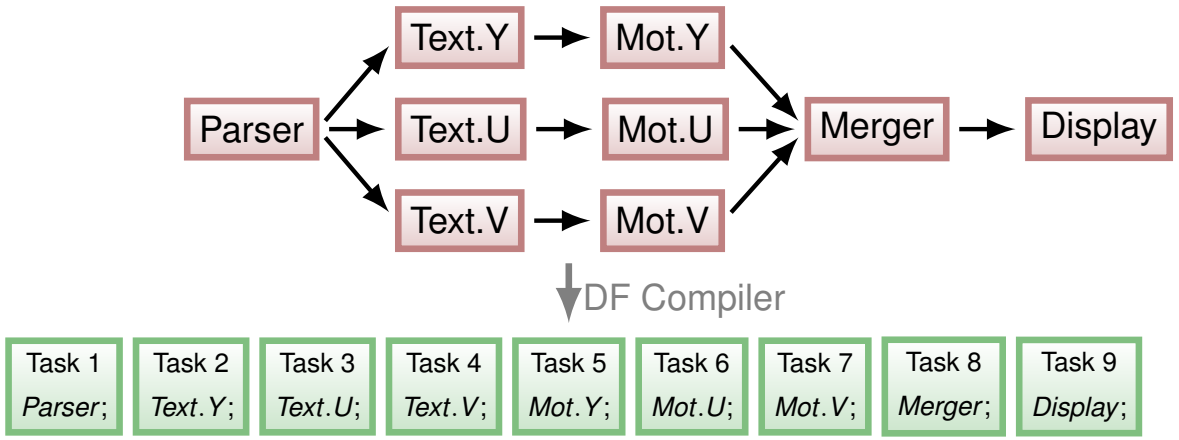
- Different kinds of parallelism
  - Actors exchanging data **only** through FIFO channels
  - Task , pipeline , data

Many models

Static analyses ← SDF ... ... (DDF) → Expressiveness

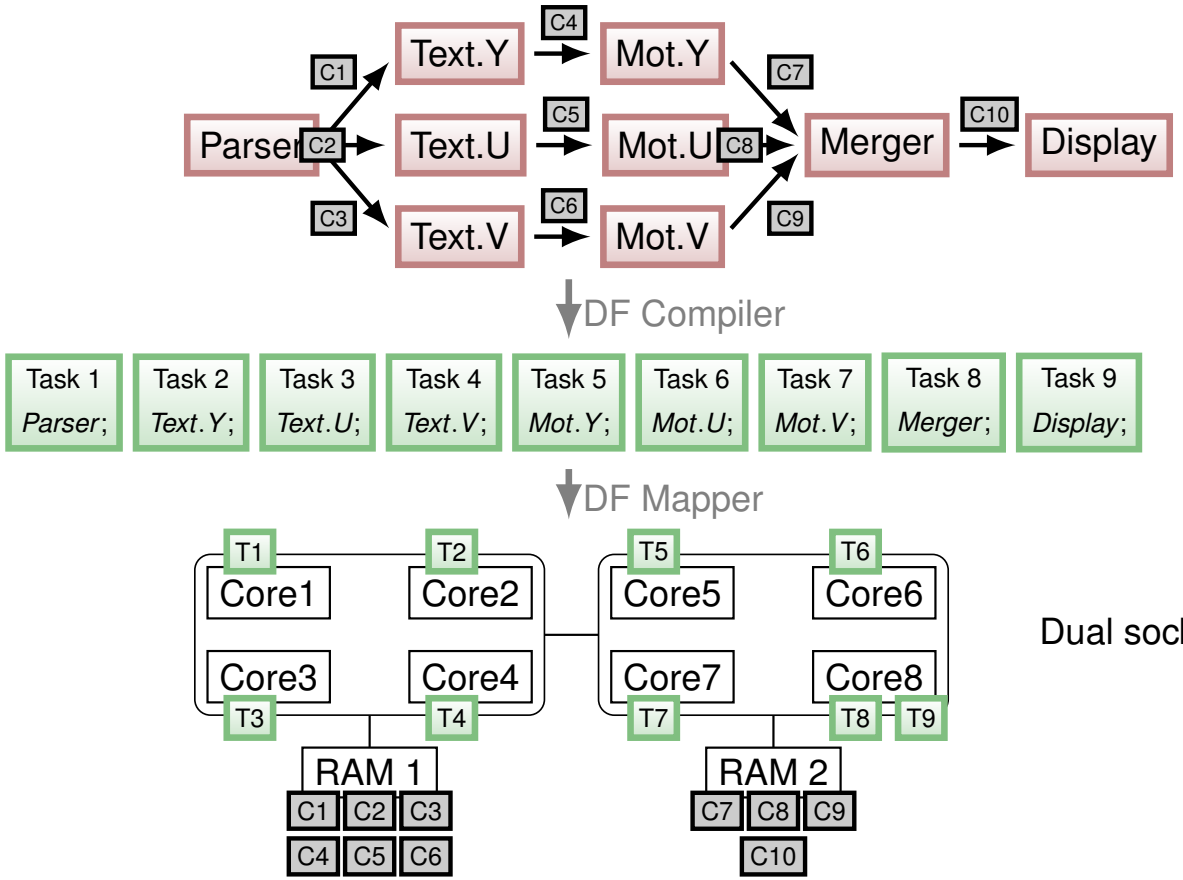
How to execute dataflow programs ?

- Compilation to synchronized tasks respecting data dependencies



How to execute dataflow programs ?

- Compilation to synchronized tasks respecting data dependencies
- Mapping of tasks and channels to hardware



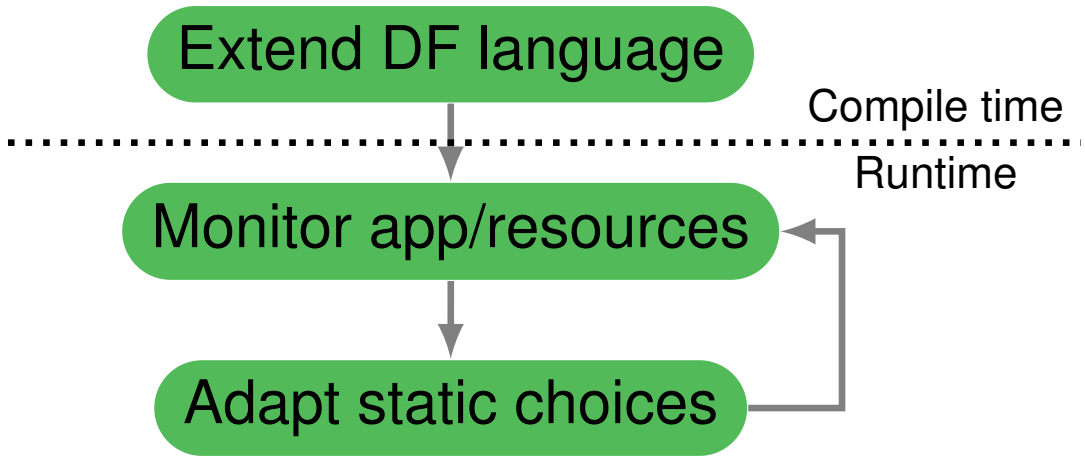
Dual socket processor

Proposition

Motivations

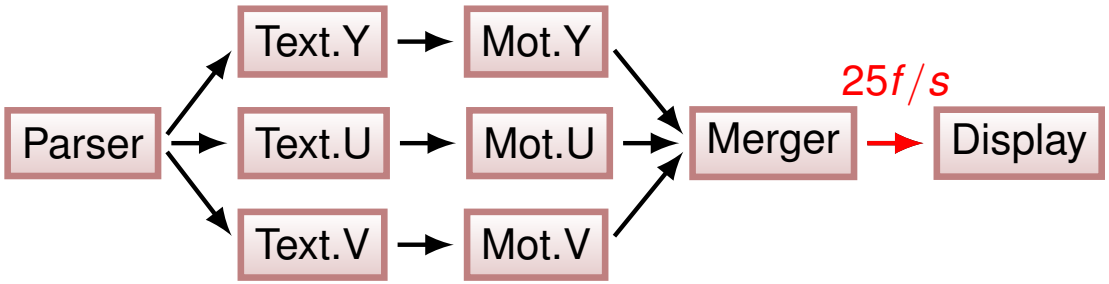
- DF applications with throughput constraints
- Mapping satisfying constraints requires:
  - Actors internal execution time
  - Concurrent applications
  - DF actors consumption/production rates

Goals



Contrib 1: Languages/compilers extensions

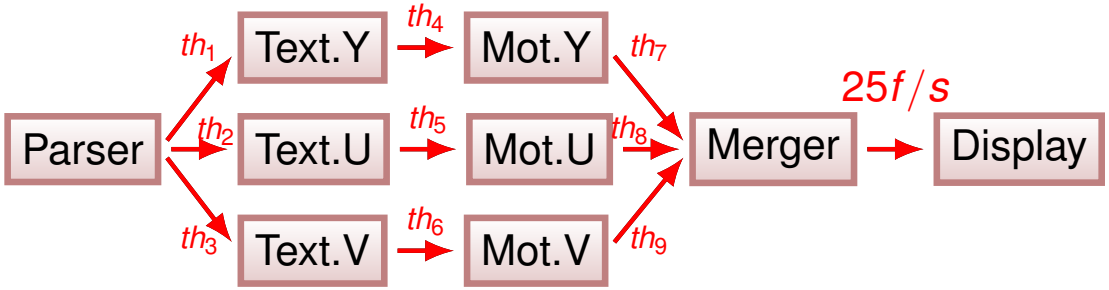
- Languages extensions taken into account in compilation flow [9, 10]



H264 graph with throughput constraint

Contrib 1: Languages/compilers extensions

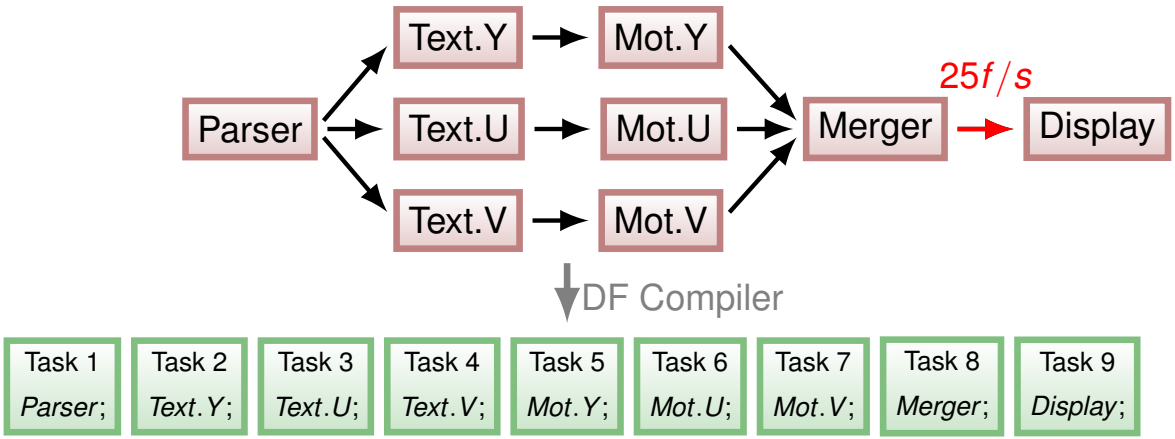
- Languages extensions taken into account in compilation flow [9, 10]



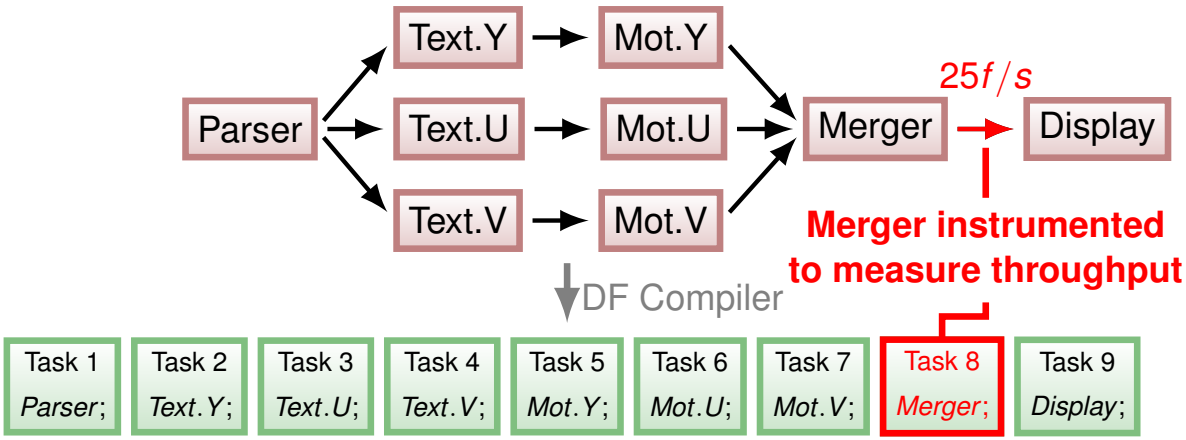
H264 graph with throughput constraint

- Propagate this value in SDF languages
  - Determine actors acceptable exec time

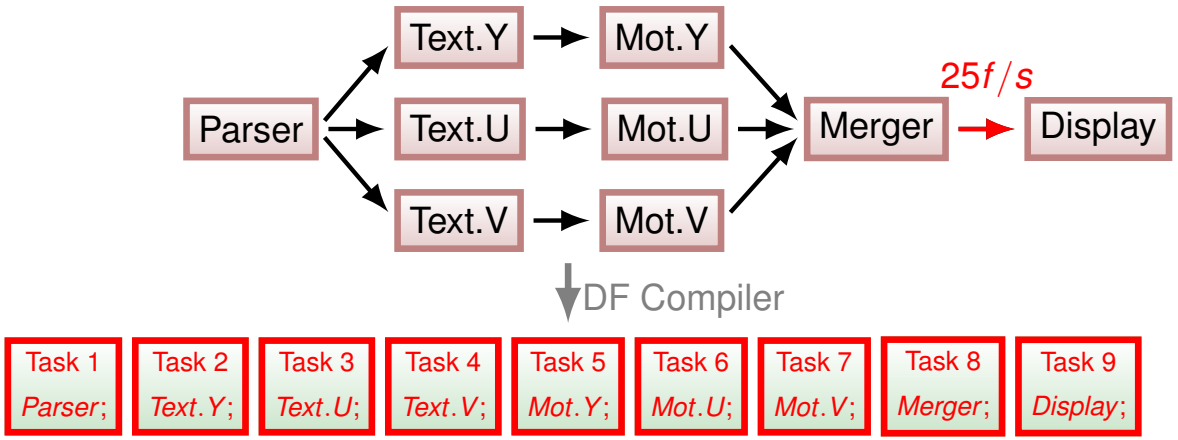
Contrib II - Fine grain monitoring



Contrib II - Fine grain monitoring

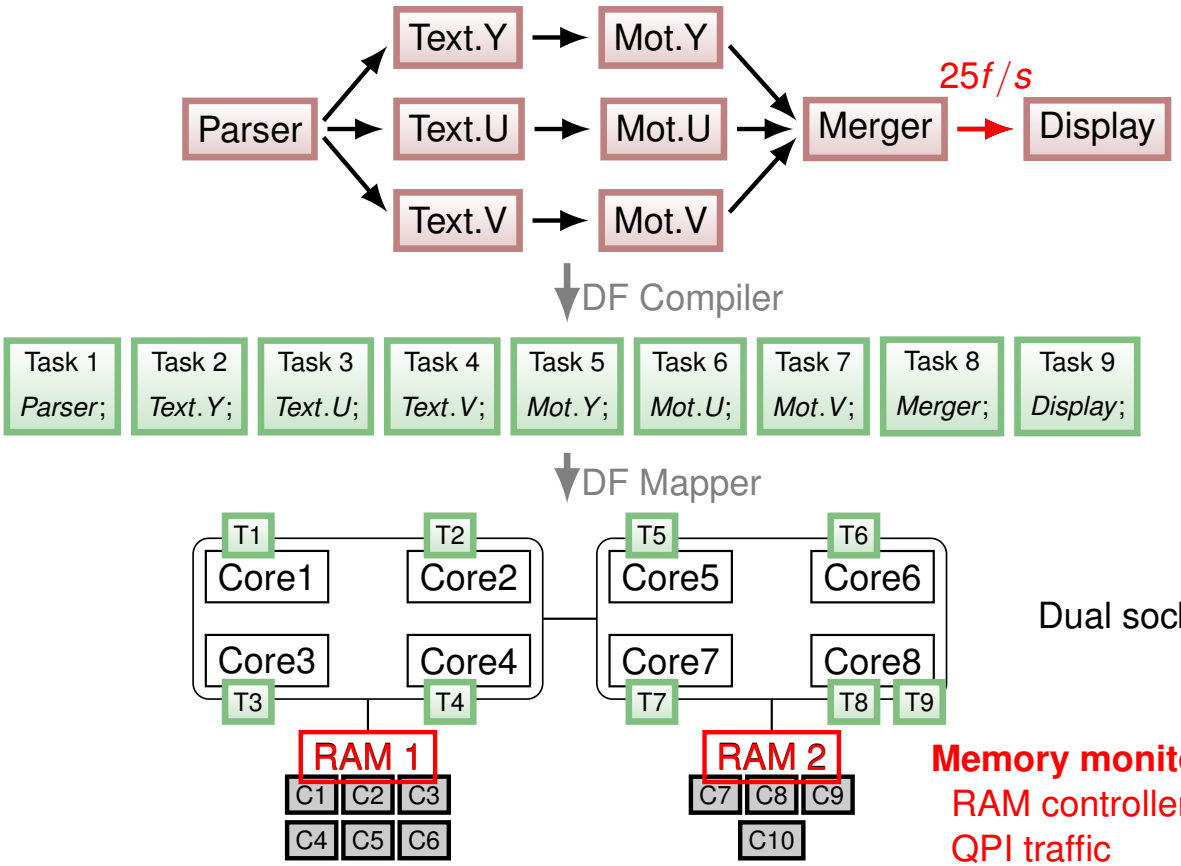


Contrib II - Fine grain monitoring



Tasks instrumented to measure actors execution times

Contrib II - Fine grain monitoring

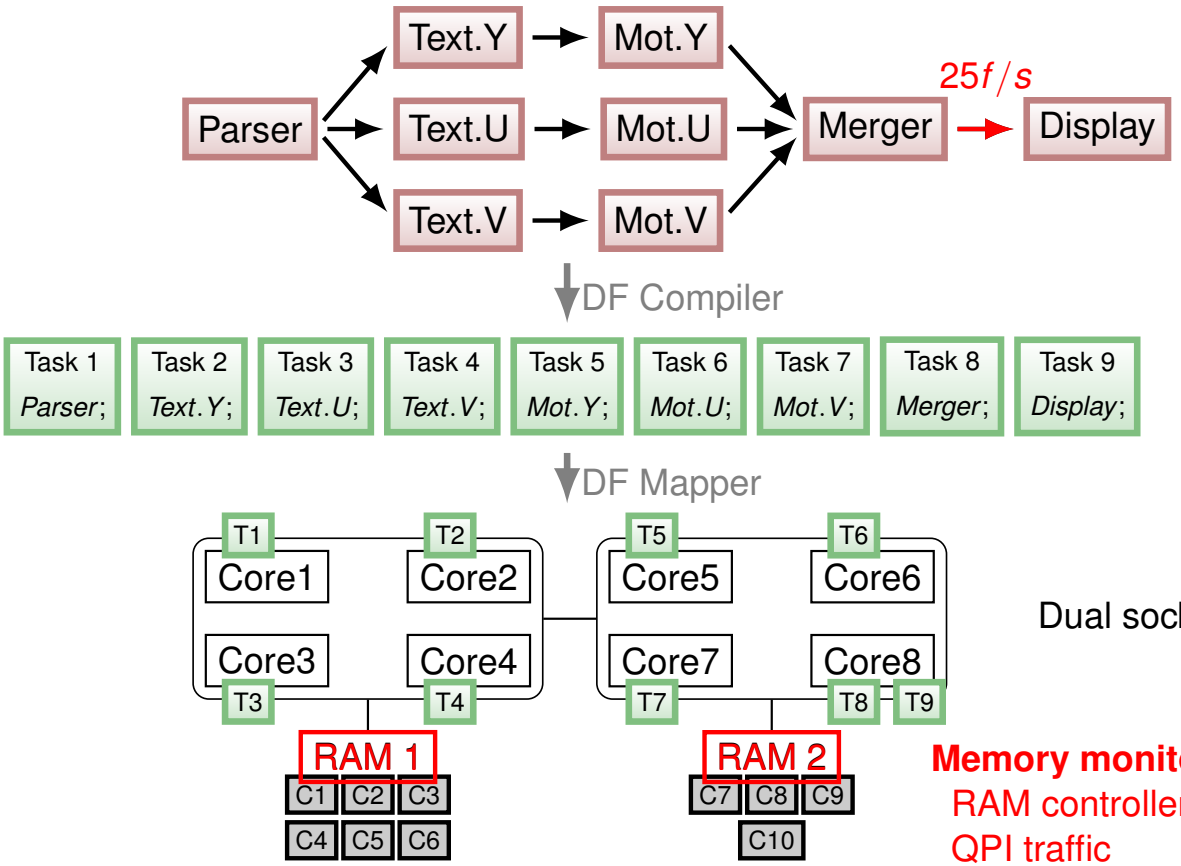


Dual socket processor

Memory monitoring using PMU:

- RAM controllers load
- QPI traffic

Contrib II - Fine grain monitoring



Dual socket processor

Memory monitoring using PMU:

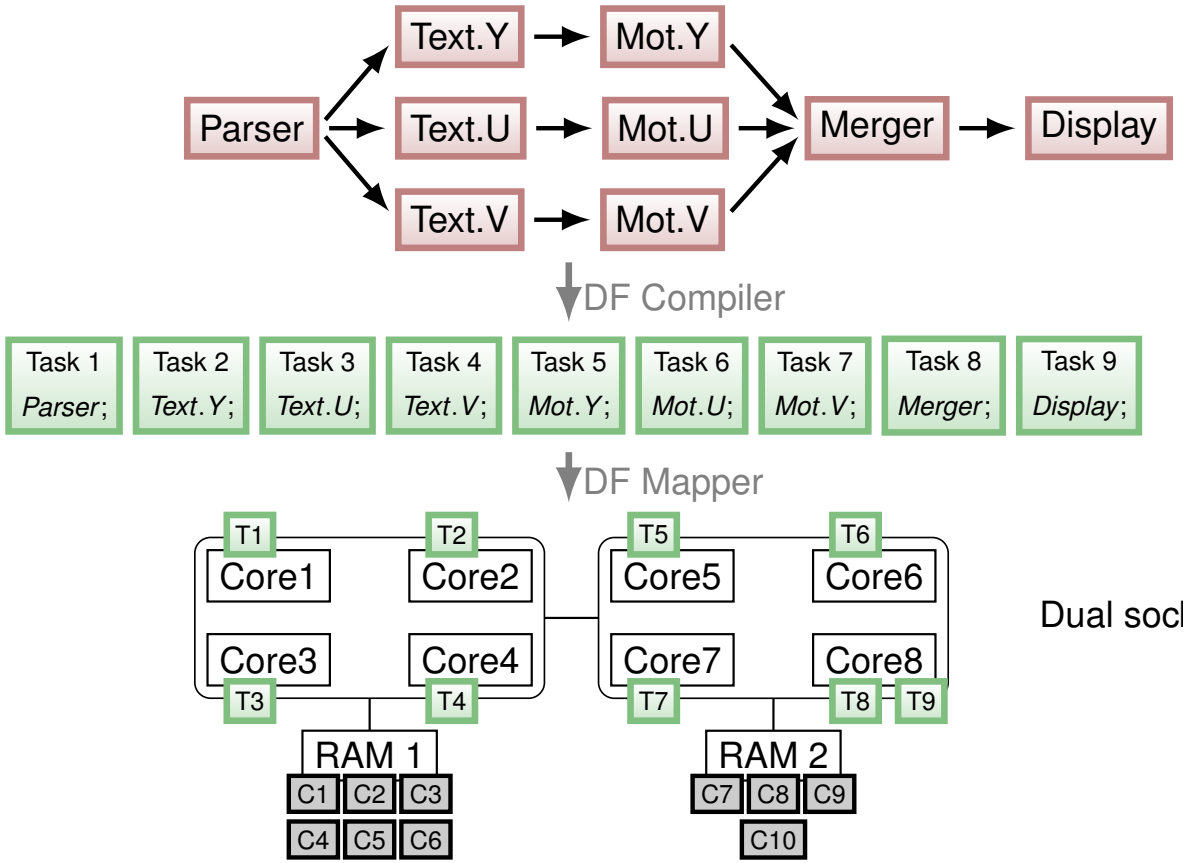
- RAM controllers load
- QPI traffic

Conclusions

Are we facing cores load imbalance ?

Are the actors too slow because of memory latencies ?

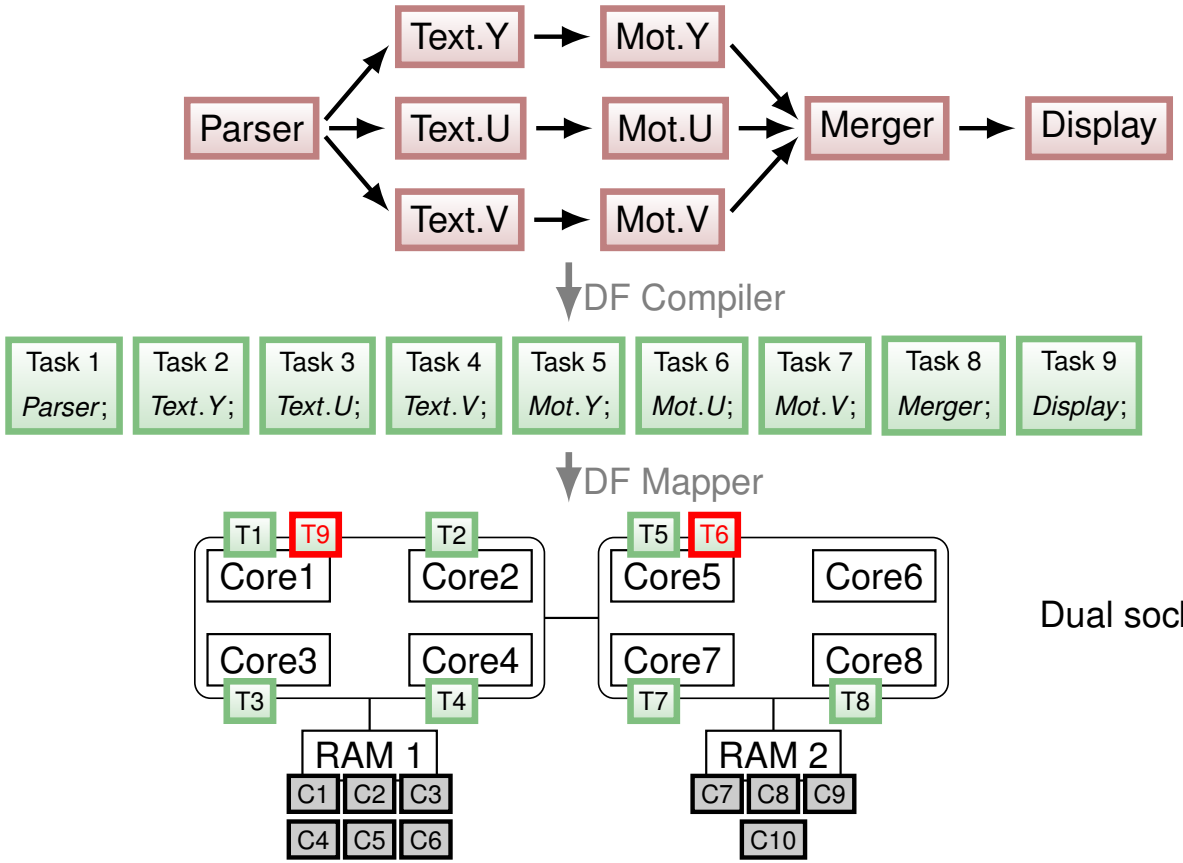
Contrib III - Dataflow adaptations



Dual socket processor

Contrib III - Dataflow adaptations

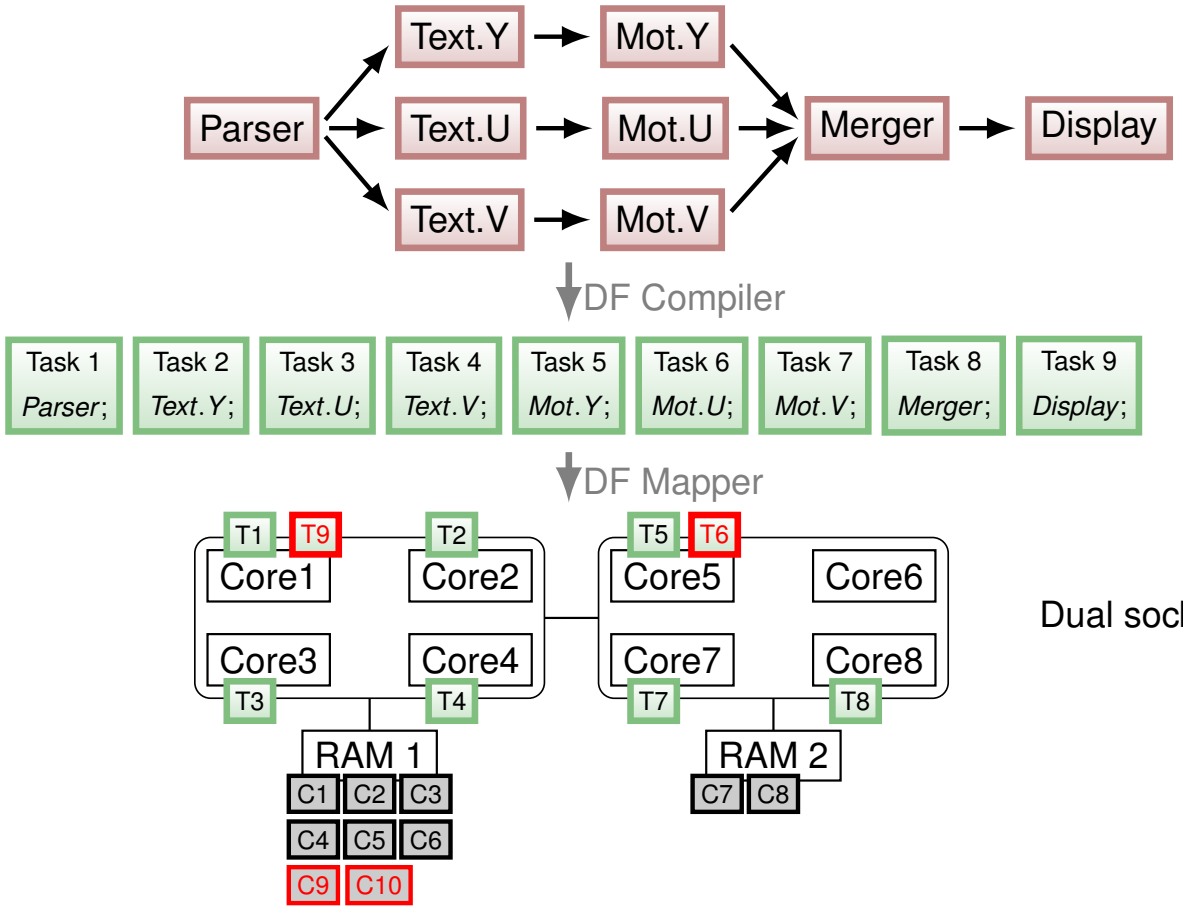
- Cpu load balancing



Dual socket processor

Contrib III - Dataflow adaptations

- Cpu load balancing
- Memory load balancing



Dual socket processor

Conclusion

Dynamic framework for DF programs

- Applicative monitoring
- Hardware monitoring
- Runtime adaptations making profit of DF information

Current work

- Finishing implementation in Streamit

State of the art

- **DF** compilation [7, 13, 4]
- **DF** theoretical throughput analysis [3, 11]
- **DF** adaptation [12, 8, 5, 1]
- **Non-DF** NUMA adaptations [6, 2]

Choi, Y., Li, C.-H., Silva, D. D., Bivens, A., and Schenfeld, E.
Adaptive task duplication using on-line bottleneck detection for streaming applications.
In Proceedings of the 9th Conference on Computing Frontiers (New York, NY, USA, 2012), CF '12, ACM, pp. 163–172.

Dashti, M., Fedorova, A., Funston, J., Gaud, F., Lachaize, R., Lepers, B., Quema, V., and Roth, M.
Traffic management: A holistic approach to memory placement on numa systems.
In Proceedings of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems (New York, NY, USA, 2013), ASPLOS '13, ACM, pp. 381–394.

Ghamarian, A.-H., Geilen, M. C. W., Stuijk, S., Basten, T., Moonen, A. J. M., Bekooij, M., Theelen, B., and Mousavi, M.
Throughput analysis of synchronous data flow graphs.
In Application of Concurrency to System Design, 2006. ACSD 2006. Sixth International Conference on (2006), pp. 25–36.

Gordon, M. I.
Compiler Techniques for Scalable Performance of Stream Progra

PhD thesis, MIT, 2010.

Hormati, A. H., Choi, Y., Kudlur, M., Rabbah, R., Mudge, T., and Mahlke, S.
Flextream: Adaptive compilation of streaming applications for heterogeneous architectures.

In Proceedings of the 2009 18th International Conference on Parallel Architectures and Compilation Techniques (2009), pp. 214–223.

LACHAIZE, R., LEPERS, B., AND QUÉMA, V.
Memprof: A memory profiler for numa multicore systems.
In Proceedings of the 2012 USENIX Conference on Annual Technical Conference (Berkeley, CA, USA, 2012), USENIX ATC'12, USENIX Association, pp. 5–5.

LEE, E. A., AND MESSERSCHMITT, D.
Synchronous data flow.
Proceedings of the IEEE 75, 9 (sept. 1987), 1235 – 1245.

MIN, C., AND EOM, Y. I.
Danbi: Dynamic scheduling of irregular stream programs for many-core systems.
In Proceedings of the 22Nd International Conference on Parallel Architectures and Compilation Techniques

(Piscataway, NJ, USA, 2013), PACT '13, IEEE Press, pp. 189–200.

Selva, M., Morel, L., Marquet, K., and Frénot, S. Extending dataflow programs with throughput properties. In Proceedings of the First International Workshop on Many-core Embedded Systems (New York, NY, USA, 2013), MES '13, ACM, pp. 54–57.

Selva, M., Morel, L., Marquet, K., and Frénot, S. Qos monitoring system for dataflow programs. In Proceedings of the Conférence d'informatique en Parallélisme, Architecture et Système (ComPAS) (2013), CFSE track.

Stuijk, S., Basten, T., Geilen, M. C. W., and Corporaal, H.
Multiprocessor resource allocation for throughput-constrained synchronous dataflow graphs.
In Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE (2007), pp. 777–782.

Tan, C.
A hybrid static/dynamic approach to scheduling stream programs, 2009.

Thies, W.
Language and compiler support for stream programs.
PhD thesis, MIT, 2009.