Dumitru Potop-Butucaru

November 12, 2022

Experienced permanent researcher (CRHC) at INRIA, France in the Kairos research team.

address: Inria de Paris, 2, rue Simone Iff CS 42112
75589 Paris CEDEX 12, France
e-mail: dumitru.potop@inria.fr
phone: +33-180.49.40.10
web: https://who.rocq.inria.fr/Dumitru.Potop_Butucaru/

1 Degrees

2015 Habilitation (HdR) in Computer Science. EDITE doctoral school. Habilitation mémoir: "Real-time systems compilation". Reviewers: Sanjoy Baruah, Nicolas Halbwachs, and Reinhard von Hanxleden.

2002 PhD in Computer Science. Ecole des Mines de Paris. PhD advisors: Gérard Berry and Robert de Simone. Thesis: "Optimizations for Faster Simulation of Esterel Programs". Eiffel excellence scholarship.

1999 MSc in Computer Science ("Operating Systems and Parallel Computing" master program). University of Bucharest, Faculty of Mathematics, Informatics Department. Ranked 1st (9.80/10 GPA). ERASMUS student at Pierre et Marie Curie University (Paris 6) in the 1998-1999 scholar year ("Numerical Analysis" master program).

1997 BSc in Computer Science. University of Bucharest, Faculty of Mathematics.

2 Employment history (research and teaching)

2005- Permanent researcher at INRIA, France. Junior position (CR2) until 2007, experienced position (CR1/CRCN) from 2007 to 2020, experienced position (CRHC) since 01/01/2021.

2019- Teaching at EIDD. Module "Real-time systems". Volume: 24h lecture + 18 hours lab supervision per year.

2011- Teaching at EPITA. Module "A synchronous approach to embedded real-time systems design". Volume: 16h lecture + 16 hours lab supervision per year from 2016, 6h lecture + 6h lab supervision per year in 2013, 2014, and 2015, 4h lecture + 4h lab supervision per year in 2011 and 2012.

2014 Teaching at Polytech Paris UPMC. Module "A synchronous approach to embedded real-time systems design.". Volume: 12h lecture + 15h lab supervision.

2005 Post-doc in Verimag, Grenoble, France, in the Synchrone team. Funded by the ASSERT European project.

2005 Laboratory instructor at ENSIMAG, Grenoble, France. Volume: 32 hours of lab (software project) supervision.

2003-2004 Post-doc in INRIA/IRISA, Rennes, France, in the S4 team. Funded by the ARTIST and COLUMBUS European projects.

1997-1998 Teaching assistant at the University of Bucharest, Faculty of Mathematics, Department of Informatics. Disciplines: Informatics (1st year Licence level), Data bases (2nd year Master level). Volume: more than 200 hours of lab supervision.

3 Advising

3.1 PhD

3.1.1 Current

- Fabien Siron (CIFRE funding application jointly submitted with KronoSafe accepted in october 2020). Co-supervision with Robert de Simone. Main result: paper [C18]. Topic: "Formal verification methodology for LET real-time applications".

- Hugo Pompougnac (2019-2022). Full supervision (EDITE doctoral school). Funding: ES3CAP. Topic: "Timing-predictable implementation of neural networks". Main results: paper [J7], open-source software mlirlus. Defence scheduled for 9 dec. 2022.

3.1.2 Past

- Keryan Didier (2015-2019). Full supervision (EDITE doctoral school). Funding: ITEA3 ASSUME. PhD thesis: "Contributions to the safe and efficient parallelisation of hard real-time systems". Main results: papers [J6],[C17],[C16],[W6] software Lopht (the shared memory multi-core backend). Current position: Engineer at OCamlPRO.

- Thomas Carle (2011-2014). Full supervision (EDITE doctoral school). Funding: FUI8 PARSEC and INRIA. PhD thesis: "Efficient compilation of embedded control specifications with complex functional and non-functional properties". MSc dissertation: "Application of software pipelining techniques to the optimization of table-based real-time schedules". Main results: papers [J4], [C10], [J5], software Lopht (the time-triggered ARINC 653 back-end), the industrial collaboration with Airbus DS and the CNES. Current position: Assistant professor, IRIT, Toulouse, France.

- Manel Djemal (2010-2014). Co-supervision with A. Munier (EDITE doctoral school). Funding: INRIA CORDI/S. PhD thesis: "Reconciling performance and predictability on a NoC-based MPSoC using off-line scheduling techniques". Main results: papers [C9], [C10], [C11], software Lopht (the many-core back-end). Current position: private sector.

3.2 Post-doctoral

- Jad Khatib (2019-2020). Funding: ES3CAP. Subject: "Parallelization of safety-critical embedded control applications onto many-cores". Results: report [R5].

- Fatma Jebali (2017-2018). Funding: ITEA3 ASSUME. Subject: "Modeling of hardware platforms for simulation and timing analysis". Results: paper [C15]. Current position: CEA.

- Mihail Asavoae (2015-2016). Funding: LEOC CAPACITES project. Subject: "Efficient real-time mapping and timing analysis of many-core software. Application to Kalray MPPA256". Current position: CEA.

- Raul Gorcitz (2013-2016). Funding: CNES (R&T program). Subjects:

  - Evaluation of the Lopht real-time systems compiler on an avionics case study coming from Airbus DS.
  - Extension of Lopht to handle the TTEthernet embedded time-triggered network.

  Results: papers [C12], [C13]. Current position: private sector.

- Zhen Zhang (2013-2014). Funding: ANR HeLP and IRT SystemX/FSF. Subjects:

  - Construction of a predictable many-core platform and its programming. This included the definition of the DSPINpro programmable network-on-chip (NoC) (including the definition of its micro-programming language), the definition of the overall many-core architecture.
  - Translating scheduling tables (the output of the Lopht tool) in the PsyC time-triggered language.

Results: Papers [C9], [C10], [C11]. The DSPINpro programmable NoC and the associated predictable many-core architecture. The Lopht-to-PsyC translator. Current position: Senior research engineer, Huawei, France.

- Virginia Papailiopoulou (2010-2012). Funding: ARC Triade and FUI8 PARSEC. Subjects:
  - Synthesis of concurrent, possibly distributed or multi-threaded code from polychronous (multi-clock synchronous) specifications.
  - Synthesis of real-time implementations onto time-triggered ARINC 653 avionics platforms.

  Results: Paper [W1] and software tool *Syn*[2].

4 Research grants and direct industrial collaborations

4.1 Current

4.1.1 Google grant (2023-2025)

Google grant funding a PhD thesis. The objective is to study the use of synchronous languages in the formalization of a datacenter-oriented execution platform and resource allocation problem.

4.1.2 CAOTIC (Inria PI) (2022-2026)

CAOTIC is an ambitious initiative aimed at pooling and coordinating the efforts of the main French research teams working on the timing analysis of multicore real-time systems, with a focus on interference due to shared resources. The objective is to enable the efficient use of multicore in critical systems. Based on a better understanding of timing anomalies and interferences, taking into account the specificities of applications (structural properties and execution model), and revisiting the links between timing analysis and synthesis processes (code generation, mapping, scheduling), significant progress is targeted in timing analysis models and techniques for critical systems, as well as in methodologies for their application in industry.

4.1.3 Confiance.AI, project EC7 Embedded AI (2021-2024)

We participate to the Confiance.AI program, in the project "EC7–Embedded AI". Confiance.AI is the technological pillar of the Grand Défi "Securing, certifying and enhancing the reliability of systems based on artificial intelligence" launched by the Innovation Council. It is the largest technological research programme in the AIforHumanity plan, which is designed to make France one of

the leading countries in artificial intelligence (AI). Our technical contributions to the project are described in Section 8.1.1.

4.1.4 ES3CAP (2018-2022)

The objective of the (now terminating) ES3CAP to propose a software environment for new-generation multi-core and many-core critical systems. The proposed solution will facilitate the transition from current critical systems architectures (distributed heterogenous, single-core, single-task) to centralized (multi-/many-core) solutions with support for HPC, real-time, safety and security. Target markets are avionics (aviation, drones). defense, and autonomous vehicles. Prototyping within ES3CAP will be performed on Kalray many-cores, but the result should be exploitable on other architectures. Our participation in the project, which is now terminating, has been the definition of a novel method for the analysis of timing interferences on the Kalray Coolidge many-core [R5]. The project funded the post-doc of Jad Khatib and part of the PhD thesis of Hugo Pompougnac.

4.1.5 Airbus direct collaboration (coordinator) (2019,2023)

Direct collaboration and 25k€ grant by Airbus (civil airplane division). The objective was to extend the real-time parallelization technique and the Lopht multi-core back-end developed during the ITEA3 Assume project, to allow parallelization and code generation on consumer multi-cores (and in particular on the T1042 quad-core with POWER architecture). This requires handling architectures that are not timing predictable, and where predictability requires the use of prefetching.

After a period of internal evaluation within Airbus, work will restart in 2023.

4.2 Past

- IRT Saint-Exupéry (Coordinator) (2019)
  Direct collaboration and 25k€ grant by IRT Saint Exupéry (CAPHCA project). The objective was to extend the real-time parallelization technique developed during the ITEA3 Assume project, to allow parallelization and code generation on predictable automotive multi-cores (and in particular the Infineon TC27x family).

- ITEA3 Assume (Inria coordinator) (2015-2018).
  For efficient construction and synthesis of embedded systems, the project provides new tools, standards and methodologies to cover most of the challenges by design. In addition, ASSUME provides a well-integrated sound static analysis solution that allows proving the absence of problems even in a multi-core environment.

- LEOC CAPACITES (2014-2017).
  The objective of CAPACITES is to build a hardware and software plat-

form based on the exploitation of integrated many-core processors and to demonstrate the relevance of these many-core architectures and the implementation in Kalray's MPPA for several industrial applications.

- Airbus DS/CNES (INRIA coordinator) (2012-2016).
  Direct collaboration with Airbus DS (space launcher division) and the CNES. The objective is to study computer-aided techniques for the construction of the real-time software and of the embedded network configuration for embedded control systems such as that of a space launcher. Such techniques must reduce the design and validation costs, while improving the reliability levels by comparison to the Ariane 5 flight program. The CNES funded the 2-year post-doctorate of R. Gorcitz and it bought and lent to INRIA for 1 year a TTEthernet and VxWorks653-based platform.

- FUI P (2012-2015).
  The goal of Project P is to support the model-driven engineering of high-integrity embedded real-time systems by providing an open code generation framework able to verify the semantic consistency of system specifications, generate optimized source code for multiple programming (Ada, C/C++) and syntesis (VHDL, SystemC) languages, and support a multi-domain certification process.

- IRT SystemX/Project FSF (2013-2014).
  The FSF project targets the railway sector. In this context, the objective is to organize operational hardware and software resources so that the execution platform attains the performance, security-of-operation, availability and safety goals for a given system.

- FUI8 PARSEC (INRIA coordinator) (2011-2014).
  The project provided development tools for critical real-time distributed systems requiring certification according to the most stringent standards such as DO-178B (avionics), IEC 61508 (transportation) or Common Criteria for Information Technology Security Evaluation.

- ANR HeLP (2010-2013).
  The HELP project focused on functional and non-functional high-level models for the design of low-power embedded systems.

- ITEA2 OPENPROD (2009-2012).
  Development of an open model-driven development, modelling and simulation (M&S) environment that integrates Eclipse with open-source modelling and simulation tools such as OpenModelica and industrial M&S tools and applications.

- ARC Triade (2009-2010).
  The overall objective is to best map complex applications onto heterogeneous architectures, the former comprising latent (logical) concurrency while the latter offer (physical) intrinsic parallelism with contrived communication structures.

- ANR MeMVaTEx (2005-2008).
  Definition of a methodology for the modeling and the validation of requirements trough a design flow for embedded real-time systems.
- Older projects: IST ARTIST2 NoE, ARTIST, COLUMBUS, HIDES associated team (with Columbia University, NY, USA)

5 Full list of publications

5.1 Book

B1 D. Potop-Butucaru, S. Edwards, G. Berry : Compiling Esterel. Springer, July 2007. ISBN 978-0-387-70626-9.

5.2 Book chapters

BC6 D. Potop-Butucaru, Y. Sorel: Approche synchrone et ordonnancement. In M. Chetto (ed.), "Ordonnancement dans les systèmes temps réel", ISTE, June 2014.

BC5 D. Potop-Butucaru, Y. Sorel: Synchronous Approach and Scheduling. In M. Chetto (ed.), "Real-time Systems Scheduling 2: Focuses", Wiley-ISTE, 2014.

BC4 D. Potop-Butucaru, R. de Simone, Y. Sorel: From Synchronous Specifications to Statically-Scheduled Hard Real-Time Implementations. In S. Shukla, J.-P. Talpin (eds.), Synthesis of Embedded Software. Springer, 2010. ISBN: 978-1-4419-6399-4

BC3 D. Potop-Butucaru, R. De Simone, J.-P. Talpin: The synchronous hypothesis and polychronous languages. In R. Zurawski (ed.), The Embedded Systems Handbook, 2nd edition, CRC Press, 2009.

BC2 D. Potop-Butucaru, R. De Simone, J.-P. Talpin: The synchronous hypothesis and synchronous languages. In R. Zurawski, ed., The Embedded Systems Handbook, CRC Press, 2005.

BC1 D. Potop-Butucaru, R. de Simone : Optimizations for Faster Execution of Esterel Programs. In R. Gupta, P. LeGuernic, S. Shukla, and J.-P. Talpin, eds. Formal Methods and Models for System Design, Kluwer Academic Publishers, 2004.

5.3 Peer-reviewed Journals

J7 H. Pompougnac, U. Beaugnon, A. Cohen, D. Potop-Butucaru: Weaving Synchronous Reactions into the Fabric of SSA-form Compilers. ACM TACO 19(2), 2022.

J6 K. Didier, D. Potop-Butucaru, G. Iooss, A. Cohen, J. Souyris, P. Baufreton, A. Graillat: Correct-by-Construction Parallelization of Hard Real-Time Avionics Applications on Off-the-Shelf Predictable Hardware. ACM TACO 16(3), 2019. (preprint).

J5 T. Carle, D. Potop-Butucaru, Y. Sorel, D. Lesens: From dataflow specification to multiprocessor partitioned time-triggered real-time implementation. In LITES 2(2) (2015). (preprint).

J4 T. Carle, D. Potop-Butucaru: Predicate-aware, makespan-preserving software pipelining of scheduling tables. In ACM TACO 11(1): 12 (2014). (preprint).

J3 D. Potop-Butucaru, Y. Sorel, R. de Simone, J.-P. Talpin : From Concurrent Multi-Clock Programs to Deterministic Asynchronous Implementations. In Fundamenta Informaticae, 108(1-2):91-118, IOS Press, 2011. (preprint).

J2 D. Potop-Butucaru, B. Caillaud : Correct-by-Construction Asynchronous Implementation of Modular Synchronous Specifications. In Fundamenta Informaticae, 78(1):131-159, IOS Press, 2007. (preprint).

J1 D. Potop-Butucaru, B. Caillaud, A. Benveniste : Concurrency in Synchronous Systems. In Formal Methods in System Design, 28(2):111-130, 2006. (preprint).

5.4 Peer-reviewed Conference and Workshop Proceedings

5.4.1 International Conferences

C18 F. Siron, D. Potop-Butucaru, R. de Simone, D. Chabrol, A. Methni. The synchronous Logical Execution Time paradigm. In ERTS 2022 - Embedded real time systems, Jun 2022, Toulouse, France. (preprint).

C17 P. Baufreton, V. Bregeon, K. Didier, G. Iooss, D. Potop-Butucaru, J. Souyris: Efficient fine-grain parallelism in shared memory for real-time avionics. In Proceedings ERTS 2020, Toulouse, France.

C16 K. Didier, A. Cohen, A. Gauffriau, D. Potop-Butucaru: Sheep in wolf's clothing: Implementation models for data-flow multi-threaded software. In Proceedings ACSD 2019. (preprint).

C15 F. Jebali, D. Potop-Butucaru: Ensuring consistency between cycle-accurate and instruction set simulators. In Proceedings ACSD 2018. (preprint).

C14 A. Cohen, V. Perrelle, D. Potop-Butucaru, M. Pouzet, E. Soubiran, Z. Zhang: Hard real-time and mixed time criticality on off-the-shelf embedded multi-cores. In Proceedings ERTS2, 2016.

C13 R. Gorcitz, E. Kofman, T. Carle, D. Potop-Butucaru, R. de Simone: On the scalability of constraint solving for static/off-line real-time scheduling. In Proceedings FORMATS 2015. (preprint).

C12 R.A. Gorcitz, D. Monchaux, T. Carle, D. Potop-Butucaru, Y. Sorel, D. Lesens. Automatic implementation of TTEthernet-based time-triggered avionics applications. In Proceedings DASIA 2015. (preprint).

C11 T. Carle, M. Djemal, D. Potop-Butucaru, R. de Simone, Zhen Zhang: Static Mapping of Real-Time Applications onto Massively Parallel Processor Arrays. In Proceedings ACSD 2014. (preprint).

C10 T. Carle, M. Djemal, D. Genius, F. Pêcheux, D. Potop-Butucaru, R. de Simone, F. Wajsbürt, Zhen Zhang: Reconciling performance and predictability on a many-core through off-line mapping. In Proceedings ReCoSoC 2014. (preprint).

C9 M. Djemal, R. de Simone, F. Pêcheux, F. Wajsbürt, D. Potop-Butucaru, Zhen Zhang: Programmable routers for efficient mapping of applications onto NoC-based MPSoCs. In Proceedings DASIP 2012.(preprint).

C8 D. Potop-Butucaru, A. Azim, S. Fischmeister: Semantics-preserving implementation of synchronous specifications over dynamic TDMA distributed architectures. In Proceedings EMSOFT 2010, Phoenix, AZ, USA. (preprint).

C7 D. Potop-Butucaru, R. de Simone, Y. Sorel, J.-P. Talpin: From Concurrent Multi-clock Programs to Deterministic Asynchronous Implementations. In Proceedings ACSD'09, Augsburg, Germany. (preprint).

C6 D. Potop-Butucaru, R. de Simone, Y. Sorel, and J.-P. Talpin: Clock-driven distributed real-time implementation of endochronous synchronous specifications. In Proceedings EMSOFT'09, Grenoble, France. (preprint).

C5 D. Potop-Butucaru, R. de Simone, Y. Sorel: Necessary and sufficient conditions for deterministic desynchronization. In Proceedings EMSOFT'O7, Salzburg, Austria. (preprint).

C4 D. Potop-Butucaru, B. Caillaud: Correct-by-construction asynchronous implementation of modular synchronous specifications. In Proceedings ACSD'05, Saint Malo, France, 2005. (preprint).

C3 J.-P. Talpin, D. Potop-Butucaru, J. Ouy, B. Caillaud: From multi-clocked synchronous processes to latency-insensitive modules. In Proceedings EMSOFT'05, Jersey City, NJ, USA, 2005. (preprint).

C2 D. Potop-Butucaru, B. Caillaud, A. Benveniste: Concurrency in synchronous systems. In Proceedings ACSD'04, Hamilton, ON, Canada. (preprint).

C1 D. Potop-Butucaru, R. de Simone: Optimizations for Faster Execution of Esterel Programs. In Proceedings Memocode'03, Mont St. Michel, France. (preprint).

5.4.2 International Workshops

W6 J. Souyris, K. Didier, D. Potop, G. Iooss, A. Cohen, T. Bourke, M. Pouzet: Automatic Parallelization from Lustre Models in Avionics (short paper). In Proceedings ERTS2 2018. (preprint)

W5 A. Cohen, V. Perrelle, D. Potop-Butucaru, E. Soubiran and Zhen Zhang: Mixed-criticality in Railway Systems: A Case Study on Signaling Application. In Proceedings WMCIS 2014. (preprint)

W4 D. Potop-Butucaru, I. Puaut: Integrated Worst-Case Execution Time Estimation of Multicore Applications. In Proceedings WCET 2013. (preprint).

W3 V. Papailiopoulou, D. Potop-Butucaru, Y. Sorel, R. De Simone, L. Besnard, and J.-P. Talpin: From Design-Time Concurrency to Effective Implementation Parallelism: The Multi-Clock Reactive Case. In Proceedings ESLsyn 2011. (preprint).

W2 S. Dasgupta, D. Potop-Butucaru, B. Caillaud, A. Yakovlev: Moving from Weakly Endochronous Systems to Delay-Insensitive Circuits. Electr. Notes Theor. Comput. Sci. 146(2). Proceedings FMGALS 2006. (preprint).

W1 D. Potop-Butucaru: The Kahn Principle for Networks of Synchronous Endochronous Programs. In Proceedings FMGALS2003. (preprint)

5.4.3 National Conferences

NC1 D. Potop-Butucaru: An Object-Oriented Distributed Programming Interface. In the Annals of the University of Bucharest, Mathematics-Informatics, 1997.

NC2 D. Potop-Butucaru: On an implementation of the Brunel language (in Romanian), In the Scientific Bulletin of the University of Pitesti, Romania, 1997.

5.5 Other

5.5.1 Theses

HDR D. Potop-Butucaru: Real-time systems compilation. Habilitation mémoir. EDITE doctoral school. Nov. 2015, France. (PDF).

PHD D. Potop-Butucaru: Optimizations for faster simulation of Esterel programs. PhD thesis. Ecole des Mines de Paris, Nov. 2002, France. (PS.GZ).

5.5.2 Magazine papers

M1 D. Potop-Butucaru: Quand la Raspberry Pi se met à l'avionique. In Open Silicium, 2015.

5.5.3 Significant research reports

R5 D. Potop-Butucaru, J. Khatib, P. Baufreton. Bounding memory access interferences on the Kalray MPPA3 compute cluster. Research report RR-9404 2021.

R4 G. Iooss, M. Pouzet, A. Cohen, D. Potop-Butucaru, J. Souyris, V. Bregeon, P. Baufreton: 1-Synchronous Programming of Large Scale, Multi-Periodic Real-Time Applications with Functional Degrees of Freedom. Research report hal-02495471, 2020.

R3 V. Papailiopoulou, D. Potop-Butucaru, Y. Sorel, R. de Simone, L. Besnard, J.-P. Talpin: From concurrent multi-clock programs to concurrent multi-threaded implementations. Research report INRIA RR-7577, 2011.

R2 D. Potop-Butucaru, R. de Simone, Y. Sorel: Deterministic execution of synchronous programs in an asynchronous environment. A compositional necessary and sufficient condition. Research report INRIA RR-6656, 2008.

R1 D. Potop-Butucaru: Fast Redundancy Elimination Using High-Level Structural Information from Esterel. Research report INRIA RR-4330, 2001.

Full list of research reports.

6 Significant talks

6.1 Invited talks

IT4 "Correct-by-Construction Parallelization of Hard Real-Time Avionics Applications on Off-the-Shelf Predictable Hardware". In HiPEAC 2020, presentation of journal paper [J6]. Bologna, Italy, January 2020. (slides).

IT3 "Real-Time Systems Compilation". Part of the Chronos seminar organized by Gérard Berry at Collège de France, Paris, France, June 2015. (slides).

IT2 "Predicate-aware, makespan-preserving software pipelining of scheduling tables". In HiPEAC 2015, presentation of journal paper [J4]. Amsterdam, Netherlands, January 2015. (slides).

IT1 "Efficient predictability in Manycore Systems for Real-Time" (with F. Pêcheux). In the 7th Spring School on Architectures (ARCHI'13), Le Col de Porte, France, 2013. (slides).

6.2 Other seminaries

ST4 "Reconciling performance and predictability on a many-core through offline mapping". In RePP 2014: Reconciling Performance and Predictability, Grenoble, France.

ST3 “Programmable routers for effcient mapping of appli- cations onto NoC-based MPSoCs”. In Colloque du GDR SOC-SIP du CNRS, 2012.

ST2 “Concurrency in Synchronous Systems”. In UK Asynchronous Forum, Manchester, 2004.

ST1 Regular participation and presentation in the Synchron series of workshops dedicated to synchronous languages and formalisms (from 1999 on).

7 Scientific collaborations

7.1 Current

7.1.1 Google, on embedded Machine Learning (ML)

Joint scientific work with Albert Cohen's research group at Google. It aims at defining a new approach to connect two major systems specification and implementation traditions: high-performance computing (HPC) with real-time embedded (RTE) systems. The lack of unification between these two fields is one key difficulty in the design of high-performance embedded systems, such as autonomous transportation systems or predictive control systems based on digital twins. The problem is easily stated: while HPC/ML compilation frameworks such as TVM, Glow or MLIR address the problem of specifying and very efficiently compiling HPC applications working on stored data, they lack the ability of specifying reactive systems, whose execution is organized along the time line as a continuous interaction with their environments. Our solution is based on a formal, algorithmic and tooled integration of general-purpose reactive programming into HPC/ML compilation frameworks. First results are provided in our first paper [J7], the open-source software mlirlus. Work is synchronized with our participation in the Confiance.AI project.

7.1.2 INRIA, Paris, team PARKAS

Joint work on the definition of a compilation flow going all the way from multi-period applications programmed in the Lustre language (or extensions thereof) to running parallel real-time implementations on various platforms (bare metal multi- and ,many-cores, OpenCL). The result is an integrated compilation method where the front-end (conversion to intermediate representation, normalization) is handled by extensions of the Heptagon compiler developed by the PARKAS team, while the back-end is the Lopht tool I designed. The method is described in joint publications [J6,C17,C16,C14,W6,W5,R4]. Joint participation in projects Collaboration with Airbus, ES3CAP, ASSUME,SystemX/Project FSF.

7.2 Past – National

7.2.1 IRISA, Rennes, team ALF

Joint work with I. Puaut on the definition of a novel worst-case execution time (WCET) analysis technique for parallel multi-core applications. Implemenation of this technique in the Heptane WCET analysis tool of I. Puaut. Results: The extension of Heptane to multi-cores. Joint publication [W4]. Joint participation in project CAPACITES, based on the previous results.

7.2.2 LIP6 laboratory, ALSOC team

Joint work with F. Pêcheux, F. Wajbürst, and D. Genius on defining a many-core platform with strong support for timing predictability and for static (off-line) mapping of real-time applications. Joint supervision with A. Munier of M. Djemal's PhD thesis. Post-doc of Zhen Zhang (first year). Publications [C9,C10]. The DSPINpro programmable NoC and the associated predictable many-core architecture. The definition of this platform allowed the definition of efficient mapping techniques ensuring both efficiency and predictability, as described in [C11]. Joint presentation [IT1].

7.2.3 INRIA, Rennes, team Espresso

Joint work with J.-P. Talpin on the application of the theory of weak endochrony to synchronous programs written in the Signal/Polychrony language. The objective was to allow the synthesis of multi-threaded/distributed code that preserves the semantics of the initial synchronous program. Results: papers [J3,W3], tool Syn^2, post-doc of V. Papailiopoulou.

7.3 Past - International

7.3.1 University of Waterloo, Canada

Joint work with S. Fischmeister on the automatic mapping of time-triggered distributed applications. Automatic synthesis of application-specific clock synchronization protocols with reduced communication needs. Connecting the Lopht compiler to the time-triggered execution platform defined by S. Fischmeister. Joint publication [C8]. One-week visit of S. Fischmeister to INRIA in 2009.

7.3.2 University of Newcastle, UK

Joint work with A. Yakovlev on applying the theory of weakly endochronous systems, defined in [J1,J2], to the automatic synthesis of asynchronous circuits from synchronous high-level specifications. Joint publication [W2]. One-week visit to visit to Newcastle University in 2004.

7.3.3 Columbia University, USA

Joint work with S. Edwards on the efficient compilation of the Esterel language. We co-authored (with Gérard Berry) book [B1]. I have visited Columbia University in 2005 for one week, and the Synopsys company (in Mountain View, California) in 2000, for one week.

8 Software

8.1 Active development

8.1.1 mlirlus

While machine learning (ML) compilation frameworks such as TVM, Glow, or MLIR concentrate the existing know-how in high-performance (HPC) compilation for virtually every execution platform, they lack a key ingredient needed in the high-performance embedded systems of the future: the ability to represent reactive control and real-time aspects of a system. They do not provide first-class representation and reasoning for systems with a cyclic execution model, synchronization with external time references (logical or physical), synchronization with other systems, tasks and I/O with multiple periods and execution modes.

In current ML practice, this poses problems in multiple ML settings, for reasons related to either specification or implementation. Thus, in recurrent neural networks (RNNs) and in reinforcement learning (RL) the application itself is a feedback control loop, that is naturally modeled as a reactive system. In more traditional convolutional networks, input acquisition and convolution computation can be scheduled in time, using buffers (stateful computation) in order to reduce the memory footprint of the application.

Beyond this ML algorithmic view, the field of embedded ML, where ML components are placed in the feedback loop of real-time embedded control applications,[1] is becoming both a reality and an industrial necessity.

Existing ML frameworks cannot adequately handle the specification and the implementation of reactive aspects. Specific work-arounds are defined for each class of problems mentioned above. Thus, RL agents are directly programmed in Python and interpreted, since ML compilers cannot synthesize reactive code. Ad-hoc changes to the Python code of frameworks such as Keras are also a solution when trying to modulate memory usage in specific classes of networks. Most interesting, the intuitively reactive feedback loops of recurrent neural networks (RNNs)[2] are unrolled over a finite time horizon at both specification and implementation level. Thus, instead of a reactive application working on streams, ML frameworks manipulate non-reactive, stateless applications work-

[1] In autonomous transportation, model-predictive maintenance and control based on digital twins.

[2] Such as those involved in keyword spotting (as used in applications such as Alexa, Siri or Google Assistant) or in video segmentation applications.

ing on vectors of the size of the time horizon. This approach poses problems whenever an embedded implementation is desired: high memory needs, high time latency, high computational cost, difficult to predict accuracy...

To solve this problem, instead of these *ad hoc* workarounds discussed above, we propose the direct integration of general-purpose reactiveness into the specification formalisms and compilers of ML frameworks. This solution is disruptive with respect to the current practice of both ML scientific mainstream and to its industrial practice.

We have provided a first implementation of this approach as an extension of MLIR. MLIR is an ML compiler framework built for modularity and extensibility, distributed along with LLVM compiler infrastructure.

Our extensions are organized as a new intermediate representation and compiler called mlirlus. Following MLIR convention, these extensions come under the form of two dialects:

lus is a high-level reactive programming dialect based on Lustre. Specification of application control at lus dialect level comes with formal correctness checking ensuring the absence of infinite or undefined behaviors. In turn, this ensures that embedded implementation is possible in bounded memory space and in bounded execution time. The lus dialect follows a dataflow programming paradigm allowing the natural specification of ML specifications with complex control and integration with signal processing pipelines.

sync is a low-level reactive programming dialect. It directly extends the Static Single Assignment (SSA) form which stands at the core of MLIR with reactive primitives allowing synchronization with other functions and with external time references, I/O with multiple periods and execution modes.

Both dialects freely combine with the data processing operations already present in MLIR (in dialects such as tensorflow or linalg]), thus allowing joint specification of all aspects of an embedded system - high-performance data processing and interaction with the environment.

Such specifications are compiled by mlirlus to reactive implementations. No loss of efficiency has been identified with respect to the classical non-reactive ML compilation flow.

The basic scientific underpinnings of mlirlus is described in publication [J7]. A snapshot (including use cases) is available on the public website mlirlus. The development of mlirlus is the main output of Hugo Pompougnac's PhD thesis, of our scientific collaboration with Google, and of our participation to the Confiance.AI programme, which put us in contact with the French industrial community of embedded AI. Our current effort is focused in two directions:

- Promoting the lus dialect for the industrial specification of high-performance embedded applications featuring deep neural networks (DNNs) and/or signal processing running under complex reactive control.
- Improving code generation.

8.1.2 Lopht – Logical to Physical Time Compiler

APP deposits:

- IDDN.FR.001.090041.000.S.P.2016.000.10600, deposit date 24/02/2016, for the distributed time-triggered back-end.
- IDDN.FR.001.090043.000.S.P.2016.000.10600, deposit date 24/02/2016, for the many-core back-end.

Technology Readiness Level: TRL4, evaluated on a large scale Airbus use case during the ASSUME project.

Lopht is a compiler for real-time embedded systems. It takes as input 3 pieces of specification:

- A functional specification provided under the form of a data-flow synchronous program written in Lustre. The Heptagon dialect of Lustre and its compiler are used as front-end by Lopht, but importing is also possible from the Scade, SynDEx, Signal/Polychrony or industrial specifications based on Lustre [J6,R4,C14].
- A specification of the target implementation platform. The architecture specification languages of Lopht allow the modeling and use of the following platforms:
  - Manycore processors with distributed shared memory and strong support for predictability (PhD thesis of K. Didier). Code generation is currently possible for the compute clusters of the Kalray MPPA v2 (Bostan) and for the many-core developed jointly with the Lip6 team (DSPINpro). Support for Kalray MPPA v2 (Coolidge) is under development (Post-doc of J. Khatib).
  - Consumer multi-core processors with POWER or ARM architecture (T1042, ARMv7). Support for predictable multi-cores of the Infineon TC27x family is under development (collaboration with IRT Saint-Exupéry).
  - Distributed time-triggered systems of several types:
    - IMA systems where each processors run under an ARINC653-compliant OS (PhD thesis of T. Carle) and inter-processor communication is done using the time-triggered communication network TTEthernet (post-doctorate of R. Gorcitz).
    - ASTERIOS-based systems programmed in the PsyC language (collaboration with Z. Zhang in the IRT SystemX/FSF project).
    - Systems programmed in the Network Code language [C8] (collaboration with S. Fischmeister from U. Waterloo).
- A non-functional specification defining requirements of several types: real-time (release dates, deadlines, period), criticality and partitioning, preemptability [J5].

Lopht performs the automatic allocation, real-time multiprocessor scheduling, and optimized code generation. It outputs full implementations, including executable code and platform configuration code, that are functionally correct and respect all the non-functional requirements (including real-time requirements).

Its scheduling algorithms ensure an efficient use of platform resources (processor cores, communication media, memory) through the use of advanced optimizations: efficient handling of execution conditions and modes, software pipelining to improve computation throughput, post-scheduling optimization of the number of context switches, task clubbing to reduce implementation complexity, etc. To ensure the respect of real-time requirements, the many-core back-end pilots the use of the worst-case execution time (WCET) analysis tool (ait from AbsInt). By doing this, and by using a precise timing model for the platform, it eliminates the need to adjust the WCET values through the addition of margins to the WCET values that are usually both large and without formal safety guarantees.

All algorithms are scalable heuristics providing good results on case studies and synthetic examples (exact scheduling techniques do not scale for such complex systems [C13]).

General-purpose optimization techniques included in Lopht are presented in [C6] (handling of execution conditions) and [J4] (software pipelining). The algorithms for taking into account complex non-functional requirements are presented in [J5]. The shared-memory and manycore-specific scheduling and code generation algorithms are presented in [J6],[C17],[C16],[C11],[C10],[W6]. The ARINC 653-specific algorithmics are presented in [J5],[C12]. Avionics case studies in [J6],[J5], mixed criticality rail case study in [C14],[W5].

8.2 Past/dormant

8.2.1 xMASTime - hardware modeling language for timing analysis

The xMAS micro-architecture modeling language has been introduced by Intel to facilitate the formal representation and analysis of on-chip interconnect fabrics. Our new domain-specific language xMAStime, inspired by xMAS, allows the modeling of full micro-architectures comprising certain classes of CPU pipelines, caches, and RAM. Given an in-order pipeline model in xMAStime, it allows the automatic generation of both a Cycle-Accurate, Bit-Accurate (CABA) hardware simulator and a timed instruction set simulator where time is accounted with safe upper bounds, as in the pipeline analysis step of Worst-Case Execution Time (WCET) analysis. The approach relies on the theory of endochronous systems ([J1],[J2],[C2-5]), which allows us to ensure functional equivalence and timing consistency between the two generated simulators, using a delay-insensitivity argument. xMAStime is implemented over Lucid Synchrone – a dataflow synchronous language featuring a higher order type system and type inference, which facilitate the definition of our DSL. We used the new DSL to model and synthesize simulation code for a full-fledged MIPS32-based

architecture. Preliminary results are presented in [C15].

8.2.2 grc2c – Fast Esterel language compiler

Optimizing compiler for the Esterel synchronous programming language. The compiler produces fast C code for simulation and implementation while maintaining a small code size. The compiler was included in the EsterelStudio IDE developed by Esterel Technologies (today, part of this technology is property of Synopsys, inc.). The compilation technique is discussed *in extenso* in references [B1,C1,BC1,PHD].

8.2.3 Extension of the Heptane WCET analyzer to handle parallel multi-core code

Heptane is a state-of-the-art worst-case execution time (WCET) analysis tool developed by a team lead by I. Puaut.

Together with I. Puaut, I have extended the analysis algorithms of Heptane to allow the computation of *tight* estimates of the worst-case response time (WCRTs) of parallel non-preemptive applications running on multicore platforms. The proposed technique is termed *integrated* in the sense it estimates jointly WCETs and WCRTs. We demonstrated that using such an integrated approach allows to obtain tighter response times than the more classical *decoupled* approaches, that compute WCRTs based on the composition of WCETs estimated on code portions considered in isolation. We showed that the proposed approach outperforms a baseline WCRT estimation approach on two embedded control applications, by 21% in average.

Results are presented in [W4].

8.2.4 DSPINpro – Programmable Network-on-Chip

In the framework of our Lip6/ALSOC collaboration, we extended the DSPIN 2D mesh network-on-chip (NoC) developed at LIP6. We replaced the fair arbitration modules of the DSPIN NoC routers with static, micro-programmable arbiters that can enforce a given packet routing sequence, as specified by small programs. We advocated the desired level of expressiveness/complexity for such simple configuration programs. The result is named Programmable DSPIN, or DSPINpro. To improve the efficiency and predictability of our DSPINpro-based architecture, and thus facilitate real-time mapping, we also constrained and standardized the structure of the computing tiles connected to the DSPINpro NoC, as well as the software architecture of our implementations. On the hardware side, we constrained the type and number of tile components (CPUs, RAM banks, DMA units ...). On the software architecture side, we constrained the memory organization, we imposed that all computations are performed on local tile data, with specific "send" operations being in charge of all inter-tile data transfers (along with a hardware lock mecanism), and we required the explicit placement of input and output data of the tile on memory banks.

Results are presented in M. Djemal's PhD thesis, and in publications [C9], [C10], [C11].

8.2.5 Syn^2- Synchronization Analysis and Synthesis for Multi-clock Applications

We have defined a full design flow starting from high-level DSLs and going all the way to the generation of deterministic concurrent (multi-threaded) executable code for simulation or (possibly distributed) implementation. The flow uses the Signal/Polychrony-SME environment as a front-end. It adds a new back-end, based on the theory of weakly endochronous systems, which allows the automatic detection of potential parallelism for use in the generation of concurrent code. The theory of weakly endochronous systems is introduced in references [J1,J2,J3], a description of the algorithms is given in references [C7], and a description of the methodology is provided in reference [W3].

8.2.6 scsimplify – Digital circuit optimizer

Optimizer for digital synchronous circuits (netlists) produced by Esterel. By exploiting the hierarchy of the initial Esterel specification, this tool significantly improves the state encoding (including the number of registers) of the circuit. The optimization technique is discussed in references [R1,B1].

9 Community service

9.1 PhD juries

- Pierre Donat-Bouillud (examiner). Doctoral school EDITE/Sorbonne, 2019.
- Lina Marsso (examiner) Doctoral school Univ. Grenoble-Alpes/Inria/GrenobleINP, 2019.
- Jad Khatib (examiner) Doctoral school EDITE/UPMC, 2018.
- Pierre Guillou (examiner). Doctoral school Ecole des Mines de Paris, 2016.
- José Echeveste (examiner). Doctoral school EDITE, Paris, 2015.
- Chan Ngo (examiner). Doctoral school Mathématiques, informatique, signal, électronique et télécommunications, Rennes, 2014.
- Léonard Gérard (reviewer). Doctoral school Paris-Sud, 2013.

9.2 Program committees (member)

- EMSOFT 2017-2018
- FDL 2019-2020
- LCTES 2015
- ACSD 2007,2010-2014,2016-2019
- RTNS 2015-2016
- Memocode 2013-2014
- ESLsyn 2011-2015
- APRES 2011-2013
- ISED 2011-2012

9.3 Scientific vulgarization

- Presentation of computer science basics to high school students visiting INRIA.
- Magazine publication [M1].

9.4 Administration

2007-2017 Member of the selection commission that awards INRIA post-doctoral scholarships and INRIA funding for sabbatical vising academics ("détachements, délégations").

1997-1998 As a teaching assistant at the University of Bucharest I have also coordinated the establishment of the course planning, managed the temporary teaching assignments, and administered the teachers' computer network of the Faculty of Mathematics, Informatics department

1994-1995 Elected student member of the Council of the Faculty of Mathematics of the University of Bucharest.