PRO3D – Programming for Future 3D Manycore Architectures

Christian Fabre
CEA LETI
bâtiment CTL
7 av. de palestine
Z.I. de mayencin
38610 Gières, France
christian.fabre1@cea.fr

Iuliana Bacivarov
ETH Zürich
Computer Engineering and
Networks Laboratory
Gloriastrasse 35
8092 Zürich, Switzerland
bacivarov@tik.ee.ethz.ch

Ananda Basu
VERIMAG
Centre Equation
2, av. de vignate
38610 Gières, France
ananda.basu@imag.fr

Martino Ruggiero
University of Bologna
DEIS
Viale Risorgimento 2
40136 Bologna, Italy
martino.ruggiero@unibo.it

David Atienza
EPFL
ESL
ELG 130, Bld. ELG, Sta. 11
1015 Lausanne, Switzerland
david.atienza@epfl.ch

Éric Flamand
STMicroelectronics
AST
12 rue jules-horowitz
38000 Grenoble, France
eric.flamand@st.com

ABSTRACT

PRO3D tackles two 3D technologies and their consequences on stacked architectures and software stack: through silicon vias (TSV) and liquid cooling. 3D memory hierarchies and the thermal impact of software on the 3D stack are mainly explored. The PRO3D software development flow is based on a rigorous assembly of software components and monitors the thermal integrity of the 3D stack. PRO3D experiments are mainly targeted on P2012, an industrial embedded manycore platform.

Categories and Subject Descriptors

B.3.3 [**Memory Structures**]: Performance Analysis and Design Aids — *Design, Performance*; B.4.3 [**Input/Output & Data Communication**]: Subsystems Interconnections — *Physical structures, Topology*; B.4.4 [**Input/Output & Data Communication**]: Performance Analysis & Design Aids — *Formal models, Simulation, Worst-case analysis, Verification*; D.2.4 [**Software Engineering**]: Software/Program Verification — *Model checking, Statistical methods*; D.3.2 [**Programming Languages**]: [Data-flow, Concurrent, distributed, and parallel.]

1. INTRODUCTION

The shift to parallel architectures is not at all the consequence of a scientific breakthrough. It is a consequence of hitting technology walls that prevented from pushing forward the efficient implementation of traditional uniprocessor designs. Future manycores will benefit tremendously from 3-dimensional (3D) integration technology that enables the spatial distribution of both computation and storage. Such

INA-OCMC '12, January 25, 2012, Paris, France

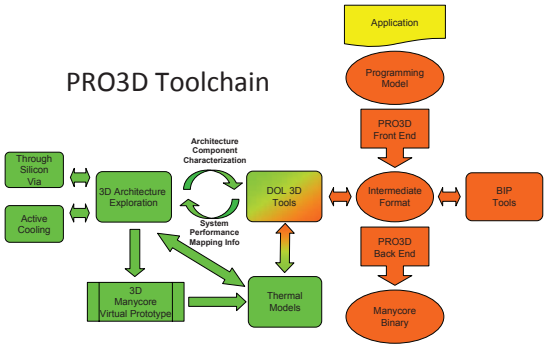

Figure 1: The PRO3D Tool Chain

oportunities requires a holistic approach starting with programming manycores, 3D architecture exploration, and fabrication technologies. While exploring design avenues offered by 3D opportunities, important challenges must be tackled: thermal management from system-level and strong requirements on software quality like composability and 3D aware distribution.

The PRO3D tool-chain presented in Fig. 1 is used first to explore the 3D design space and to select the most promising alternatives while taking into account key issues of 3D stacks, like memory hierarchies and thermal management. Application to 3D manycore mappings are actively investigated, considering both performance and temperature. The runtime environment is actually extended to provide active cooling and thermal monitoring of the 3D stack.

2. 3D ARCHITECTURAL EXPLORATION

Instruction caching, data memory and DRAM interfacing play a fundamental role when dealing with massively parallel systems, since they must provide the required memory bandwidth, complying with tight constraints in terms of size and complexity. The interconnection plays also a key role in these systems, being responsible not only for communication between the various elements but also for the management

of fundamental mechanisms such as shared resources utilization and performance scalability. Clearly, detailed design space exploration and analysis are needed to evaluate how micro-architectural differences in 3D memory hierarchy and communication architecture may affect the overall system behavior and IPC.

3D integration reflects to a tremendous increase in heat dissipation per unit area, as power density grows with the number of layers we pack in a given volume. This clearly results in higher chip temperatures and thermal stress, hence, limits the performance and reliability of the 3D stack. Conventional heat sinks, air cooling and microchannel cold-plates (i.e. back-side heat removal strategies) prove to be insufficient for 3D ICs and can only scale partially with the die size. Inter-tier microchannel-based liquid cooling has instead been proved to scale with the number of dies, but it has to be compatible with area-array TSVs in order to be capable of removing heat from multi-processor 3D ICs.

System-level architectural explorations and thermal issues have so far been addressed independently at different levels of the system design. Hence, new methodologies that address the heat removal problem concurrently at all stages and levels of the 3D chip design need to be developed and to be exploited by high-level software programming frameworks. Designers of upcoming 3D chip with microchannel will need new distinctive tools for thermal-aware 3D architectural exploration, enabling a cooling-aware design of 3D ICs. PRO3D has developed a flexible virtual platform infrastructure (VPI) for modelling and analysis of 3D integrated architectures and memory systems, as well as accurate thermal models for calculating the costs of operating the liquid cooling, determining the overall energy budget and performing run-time thermal management. MPARM [11] has been used as main VPI tool for design space explorations. It is a virtual SoC platform based on the SystemC simulation kernel, which could be used to model both HW & SW of complex systems. The system architecture simulated by the default MPARM distribution is represented by a homogeneous multicore system based on shared bus communication. During PRO3D, MPARM has been enhanced with several HW parametric models of the main micro-architectural components of a 3D integrated interconnect and memory hierarchy [3, 12], and a support of modular plug-ins for thermal models interfacing. The new 3D models are highly parametric, flexible and customizable.

PRO3D has produced 3D-ICE, a compact transient thermal model (CTTM) for liquid cooling that provides fast and accurate thermal simulations of 3D ICs with inter-tier microchannel cooling [17]. 3D-ICE can accurately predict the temporal evolution of chip temperatures when system parameters (heat dissipation, coolant flow rate, etc.) change during dynamic thermal management. We have validated the accuracy of the model with a commercial computational fluid dynamics simulation tool as well as measurement results from a 3D test IC and have foudn a maximum error of 3.4 % in temperature. PRO3D has also defined and characterized (electrically and thermally) a 3D integration process flow that combines TSV and microchannels fabrication for liquid cooling of multiple tiers [14]. These high-level technology models of complete 3D stacks have been successfully used to validate the effects of the cooling methods while executing benchmarks in the VPI [7].

3. MAPPING APPLICATIONS TO 3D PLATFORMS

In order to actually exploit the huge computational power offered by 3D integration, application software must be mapped efficiently to the architecture. Mapping involves distributing the execution of target application on parallel architectures and scheduling the components that share resources. The Distributed Operation Layer (DOL) is a framework for efficient mapping and scheduling [16, 6]. However, because mapping greatly depends on the underlying architecture and design constraints, 3D features explored in PRO3D have been added to DOL. The DOL design flow adopts synchronous data flow (SDF) to specify the application behavior independently of the communication. The architecture is modeled as an abstract representation of the underlying 3D platform, including the available execution resources and their parameters, to represent at system-level all architectural information useful for mapping decisions. The separation between computation and communication on the one hand and between application and architecture on the other hand, enables fast exploration of different mappings. The mapping is derived as a result of an automatic design space exploration (DSE) process that balances key design parameters. For instance, for PRO3D designs we are interested both in performance, reflected by execution latency on distributed components, and system maximal temperature, as the effect of application activity, hardware instantiation, and locality of system components.

The typical design practice today for thermal management is to adopt dynamic solutions, at run-time. If not planned at system-level such strategies might lead to unpredictable behavior, and might even cause poorer performance of the system, because of the (unpredictable) reduction of system frequency or runtime overhead. One strategy is to include at system-level worst-case analysis (WCA) mechanisms for both performance and temperature that provide guarantees on system behavior. These WCA mechanisms can be used as such, to propose system level mappings, or to complement/relax dynamic thermal management.

One WCA model that has been successfully applied in DOL for the analysis of real systems is the modular performance analysis (MPA) [5, 6, 13, 10, 8]. The main idea is to use a compositional approach that splits up the system into actors with minimal interactions. After characterizing separately each of the actors, and describing the composition between actors, real-time calculus is applied to analyze the entire system behavior. This WCA framework provides design guarantees that will be transferred along the entire design sequence and used by designers when doing refinements and taking design decisions along the design flow. The example in Fig. 2 shows an MJPEG application with five parallel processes mapped onto three processors of the MPARM platform [11], see [8] for a complete description. The conclusion is that the structural, parametric, and spatial location of actors and architectural elements do matter. For instance, just by taking different allocation decisions, differences of about $3K$ can be noticed in this simple example, while the latency remains the same. By including all these factors in

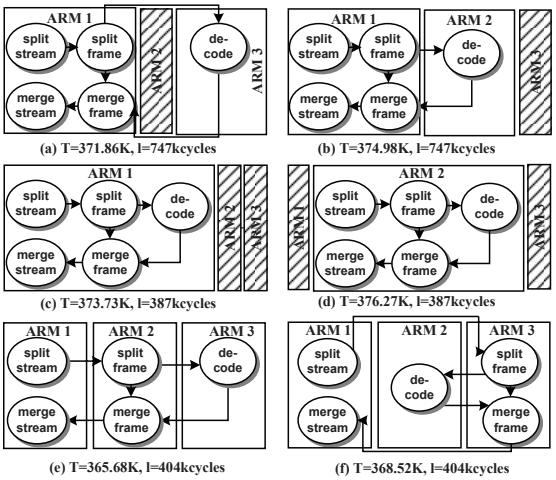

Figure 2: Different Performance for Different Application Mappings.

a formal system description, therefore allowing for formal analysis models generation and calibration with platform-dependent parameters, optimal designs can already be identified at system-level even when considering the non-trivial dependency between performance and maximum temperature.

4. GENERATION & SIMULATION OF THE SYSTEM-MODEL

The PRO3D system construction method [4] starts from a DOL [16, 6] specificationn and is both rigorous and allows fine-grain analysis of system dynamics. It is rigorous because it is based on formal BIP models [1] with precise semantics that can be analyzed by using formal techniques. A system model in BIP is derived by progressively integrating constraints induced on an application software by the underlying hardware. It is obtained, in a compositional and incremental manner, from BIP models of the application software and respectively, the hardware platform, by application of source-to-source transformations that are proven correct-by-construction [4]. The system model describes the behavior of the mixed HW/SW system and can be simulated and formally verified using the BIP toolset.

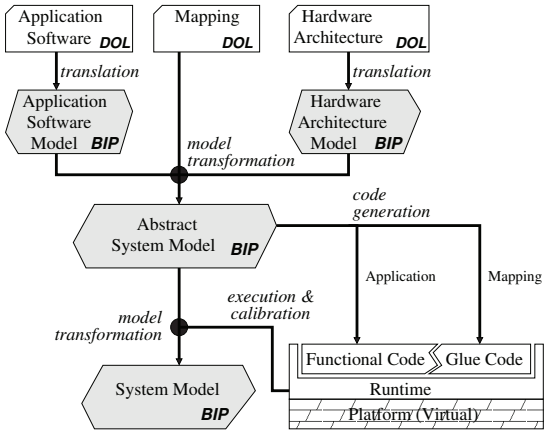

Figure 3: System Model Construction & Code Generation

The method for the construction of mixed HW/SW system models is illustrated in Fig. 3. It takes as inputs: (i) the (untimed) application software, (ii) the (timed) hardware architecture and (iii) the mapping between them described in DOL. It proceeds in two main steps. The first step is the construction of the *abstract system model*. This model represents the behavior of the application software running on the hardware platform according to the mapping, but without taking into account execution times for the software actions. In the second step, the (bounds for) execution times are obtained by running every software process in isolation on the platform. These bounds are injected into the abstract system model and lead to the *system model*. This final model allows for the accurate estimation through simulation of real-time characteristics (response times, delays, latencies, throughput, etc.) and indicators regarding resource usage (bus conflicts, memory conflicts, etc.).

We also develop an infrastructure for generating code from the BIP system model, that can be executed on a target platform and is supported by a portable runtime implemented for different platform: e.g. MPARM and P2012 (VPI) [2, 15]. The runtime provides generic API for thread management, memory allocation, communication and synchronization. The generated code is not bound to any particular platform and consists of the functional code and the glue code. The functional code describes the application tasks. For each task, a C file is generated that contains the description of the data and a thread routine describing the behavior of the task. The behavior consists of computation statements and communication calls, that are special API provided by the runtime. The functional code is generated directly from the BIP model of the application, as shown Fig. 3. The glue code implements the main routine that handles the allocation of threads to cores and the allocation of data to memories. Threads are created and allocated to processors according to the task mapping description. Data allocation consists of allocation of the thread stacks and allocation of the FIFO queues for communication. All these operations are implemented by API provided by the runtime. The generation of the glue code is performend by taking into account the mapping description, as shown in figure 3.

The code generator has been fully integrated into a toolchain and connected to the BIP system model generation flow. The generated code is compiled by the native platform compiler. The compiled code is linked with the runtime library to produce the binary image for execution on the native simulator. For PRO3D experiments with the generated code, we have used the Native Programming Layer (NPL), a common runtime implemented for both MPARM and P2012. The generated code has been tested on *mpsim* (MPARM cycle-accurate simulator), *Gepop* (P2012 posix simulator) and P2012 TLM simulator.

5. P2012: A MANYCORE PLATFORM

P2012 is a modular architcture depicted Fig. 4. Each cluster has up to 16 cores in SMP and communication engines to connect user defined HW IPs. Fabric-level communication is based on an asynchronous Network-on-Chip (NoC) organized in a 2D mesh. The routers of this NoC are implemented in asynchronous (clock-less) logic. They pro-

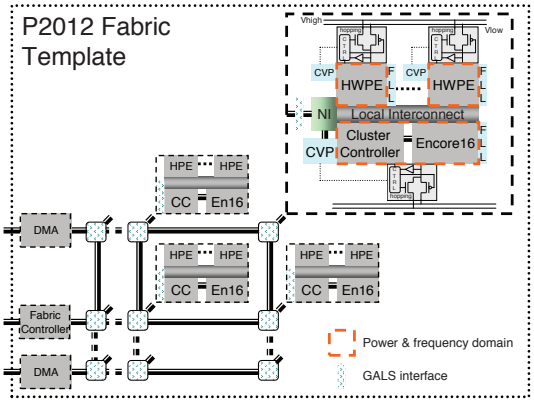

Figure 4: The P2012 Fabric Template.

vide a natural Globally Asynchronous Locally Synchronous (GALS) scheme by isolating the clusters logically. Following the GALS interface, a Network Interface (NI) is the logical link between a cluster and the NoC. It also gives access to the main Clock, Variability and Power (CVP) controller, to control a power management harness. P2012 within PRO3D will be based on a 3-tier stacking: a bottom SoC carrier for the general purpose host and IOs, a P2012 computing die, and a memory die. This will include VPI thermal modeling extensions to exercise the whole SW development flow.

6. CONCLUSIONS

Addressing two key challenges of future 3D manycores, i.e., TSV and active cooling, PRO3D proposes a consistent thermal-aware software framework for these systems. PRO3D tools are tackling all design steps from 3D system specification, system-level design space exploration, 3D stack thermal modelling, 3D architecture exploration, and efficient SW code production, ultimately targeting the industrial embedded platform P2012.

7. THE PRO3D CONSORTIUM

The PRO3D partners are: **CEA**, Commissariat à l'énergie atomique et aux énergies alternatives (coordinator), France; **VERIMAG**, represented by Université Joseph Fourier Grenoble 1, France; **ETHZ**, Eidgenössische Technische Hochschule Zürich, Switzerland; **UNIBO**, Università di Bologna, Italy; **STM**, STMicroelectronics, France; **EPFL**, École polytechnique fédérale de Lausanne, Switzerland. PRO3D lasts from Jan. 2010 to Jun. 2012.

8. ACKNOWLEDGMENTS

PRO3D is funded by the EU under FP7 GA n° 248776.

9. ADDITIONAL AUTHORS

Ahmed Jerraya, Jean-Pierre Krimm & Julien Mottin, CEA LETI. Lothar Thiele, Hoeseok Yang, Pratyush Kumar & Devesh Chokshi, ETH Zürich. Diego Melpignano, STM. Andrea Marongiu & Luca Benini, Univ. of Bologna. Paraskevas Bourgos, Marius Bozga & Saddek Bensalem, VERIMAG.

10. REFERENCES

[1] A. Basu, S. Bensalem, M. Bozga, J. Combaz, M. Jaber, T.-H. Nguyen, and J. Sifakis. Rigorous component-based design using the BIP framework. *IEEE Software, Special Edition – Software Components: Beyond Programming*, June 2011.

[2] L. Benini, D. Bertozzi, A. Bogliolo, F. Menichelli, and M. Olivieri. MPARM: Exploring the Multi-Processor SoC Design Space with SystemC. *J. VLSI Signal Process*, 41(2):169–182, 2005.

[3] D. Bortolotti, F. Paterna, C. Pinto, A. Marongiu, M. Ruggiero, and L. Benini. Exploring instruction caching strategies for tightly-coupled shared-memory clusters. In *In Int. Symp. on Systems-on-Chip*, 2011.

[4] P. Bourgos, A. Basu, M. Bozga, S. Bensalem, J. Sifakis, and K. Huang. Rigorous system level modeling and analysis of mixed HW/SW systems. In *Proceedings of MEMOCODE*, pages 11–20. IEEE/ACM, 2011.

[5] W. Haid, M. Keller, K. Huang, I. Bacivarov, and L. Thiele. Generation and calibration of compositional performance analysis models for multi-processor systems. In *Proc. Intl Conference on Systems, Architectures, Modeling and Simulation*, pages 92–99, Samos, Greece, 2009. IEEE.

[6] K. Huang, W. Haid, I. Bacivarov, M. Keller, and L. Thiele. Embedding Formal Performance Analysis into the Design Cycle of MPSoCs for Real-time Streaming Applications. 2011.

[7] A. Marongiu, P. Burgio, and L. Benini. Vertical stealing: robust, locality-aware do-all workload distribution for 3D MPSoCs. In V. Kathail, R. Tatge, and R. Barua, editors, *CASES*, pages 207–216. ACM, 2010.

[8] P. Marwedel, J. Teich, G. Kouveli, I. Bacivarov, L. Thiele, S. Ha, C. Lee, Q. Xu, and L. Huang. Mapping of Applications to MPSoCs. CODES+ISSS'11, 2011. Taiwan.

[9] PRO3D – Programming for Future 3D Multicore Architectures, 2010. `http://pro3d.eu`.

[10] D. Rai, H. Yang, I. Bacivarov, J.-J. Chen, and L. Thiele. Worst-case temperature analysis for real-time systems. DATE11, Grenoble, France, 2011.

[11] M. Ruggiero, F. Angiolini, F. Poletti, D. Bertozzi, L. Benini, and R. Zafalon. Scalability analysis of evolving SoC interconnect protocols. In *Int. Symp. on Systems-on-Chip*, pages 169–172, 2004.

[12] M. M. Sabry, M. Ruggiero, and P. G. Del Valle. Performance and energy trade-offs analysis of L2 on-chip cache architectures for embedded MPSoCs. In *Proceedings of the 20th symposium on Great lakes symposium on VLSI*, GLSVLSI '10, pages 305–310, New York, NY, USA, 2010. ACM.

[13] L. Schor, H. Yang, I. Bacivarov, and L. Thiele. Worst-Case Temperature Analysis for Different Resource Availabilities: A Case Study. In *Proc. International Workshop on Power and Timing Modeling, Optimization, and Simulation (PATMOS)*, volume 6951 of *LNCS*, pages 288–297. Springer, 2011.

[14] A. Sridhar, A. Vincenzi, M. Ruggiero, T. Brunschwiler, and D. Atienza. Compact transient thermal model for 3D ICs with liquid cooling via enhanced heat transfer cavity geometries. In *Thermal Investigations of ICs and Systems (THERMINIC), 2010 16th International Workshop on*, pages 1–6, 2010.

[15] STM and CEA. Platform 2012: A Manycore Programmable Accelerator for Ultra-Efficient Embedded Computing in Nanometer Technology, Nov. 2010. Whitepaper.

[16] L. Thiele, I. Bacivarov, W. Haid, and K. Huang. Mapping Applications to Tiled Multiprocessor Embedded Systems. In *Proc. Int'l Conf. on Application of Concurrency to System Design (ACSD)*, pages 29–40, 2007.

[17] A. Vincenzi, A. Sridhar, M. Ruggiero, and D. Atienza. Fast thermal simulation of 2D/3D integrated circuits exploiting neural networks and GPUs. In *Proceedings of the 17th IEEE/ACM international symposium on low-power electronics and design*, ISLPED '11, pages 151–156, Piscataway, NJ, USA, 2011. IEEE Press.