

Tsunami and Storm Surge Simulation Using Low Power Architectures

Concept and Evaluation

Dominik Schoenwetter¹, Alexander Ditter¹, Bruno Kleinert¹, Arne Hendricks¹, Vadym Aizinger², Harald Koestler³ and Dietmar Fey¹

¹Chair of Computer Science 3 (Computer Architecture), Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Martensstr. 3, 91058 Erlangen, Germany

²Chair of Applied Mathematics (AM1), Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Cauerstr. 11, 91058 Erlangen, Germany

³Chair of Computer Science 10 (System Simulation), Friedrich-Alexander-University Erlangen-Nürnberg (FAU), Cauerstr. 11, 91058 Erlangen, Germany

Keywords: Multiscale Simulation, Environmental Modeling, Crisis Modeling and Simulation.

Abstract: Performing a tsunami or storm surge simulation in real time is a highly challenging research topic that calls for a collaboration between mathematicians and computer scientists. One must combine mathematical models with numerical methods and rely on computational performance and code parallelization to produce accurate simulation results as fast as possible. The traditional modeling approaches require a lot of computing power and significant amounts of electrical energy; they are also highly dependent on uninterrupted access to a reliable power supply. This paper presents a concept how to develop suitable low power hardware architectures for tsunami and storm surge simulations based on cooperative software and hardware simulation. The main goal is to be able – if necessary – to perform simulations in-situ and battery-powered. For flood warning systems installed in regions with weak or unreliable power and computing infrastructure, this would significantly decrease the risk of failure at the most critical moments.

1 INTRODUCTION

Accurately predicting floods in endangered coastal regions precipitated by catastrophic geophysical events such as earthquakes, landslides, hurricanes, etc. requires accurate numerical simulations that utilize two- or three-dimensional regional or global ocean models. The computational resources needed to run these models at a sufficient spatial resolution and in real time, i.e., with a hard upper bound for the allocated time to produce a usable solution, greatly exceeds the capabilities of conventional work stations. This currently precludes the use of accurate simulation software in tsunami and flood warning systems installed in areas with weak or unreliable computational, communication, and power infrastructure – which is the case in many of the affected regions.

Currently, a number of less sophisticated approaches are being offered for installation in flood warning systems. These solutions are relying on one of the following techniques: (i) simulation at a coarser grid resolution, (ii) using simpler or less accurate nu-

merical methods, (iii) searching in a database of pre-computed scenarios, or (iv) running the simulation remotely (e.g., in a cloud). All these alternatives increase the risk of the flood prediction being either too late or too inaccurate incurring a potentially high cost in terms of human lives and property damage.

In this paper, we present a concept for an affordable, reliable, and energy-efficient flood simulation system designed to mitigate the aforementioned problems of current systems. We analyze the requirements for such a system in terms of performance, power efficiency, and reliability with the ultimate goal of designing a combined hardware/software-system capable of carrying out a flood simulation in parallel in an entirely battery-powered manner using low power compute units.

In order to speed up the development of such a system, we rely on its simulation as a first step. This simulation must not only cover the functional properties, but also provide an estimate of the system's energy consumptions and other relevant non-functional requirements. Even though our algorithm is highly

scalable (Aizinger et al., 2013), we do not consider the use of either dedicated GPU-based compute units, nor GPGPUs, as we currently do not have any reliable power models or simulations tools for them. In addition, we want to focus our study on processor architectures available in the open-source form, which also precludes usage of commercially available GPUs.

Our simulation relies on 2D/3D shallow-water solver UTBEST/UTBEST3D that is based on the discontinuous Galerkin finite element method (Aizinger and Dawson, 2002; Dawson and Aizinger, 2005). We decided on this type of numerical algorithm for a number of reasons: It runs on unstructured meshes which allows to utilize computational grids with optimal spatial resolution in the areas of interest; the code has shown excellent parallel scalability; the implementation possesses adaptive refinement capabilities, thus capable of automatically increasing the mesh resolution in the critical locations not known in advance; the user can choose between different approximation orders on the same mesh providing a simple way to get the optimal accuracy for a given performance/energy cost.

The rest of the paper is organized as follows: Section 2 provides an overview of the state-of-the-art in the use of low power architectures for high performance computing and energy optimization. Section 3 presents our concept for the simulation based evaluation of suitable low power architectures. A preliminary evaluation of available multi- and many-core hardware simulation environments and their suitability for our purposes is given in section 4. The paper concludes with a summary and outlook of future work.

2 RELATED WORK

There is a significant body of research in the field of utilizing low power architectures for high performance computing (HPC) and in the optimization of energy efficiency for HPC applications.

Rajovic et al. investigated the usage of low power ARM¹ architectures and SoCs (System on Chips) as means to reduce the cost of high performance computing (Rajovic et al., 2013a). They conclude that low power ARM-based SoCs have promising characteristics for high performance computing.

In 2013, Goeddeke et al. presented a paper on energy-to-solution comparisons between different architectures for different classes of numerical methods for partial differential equations (Göddeke et al., 2013).

¹<http://www.arm.com/>

They showed that energy to solution and energy per time step improvements up to a factor of three are possible when using the ARM-based Tibidabo cluster (Rajovic et al., 2014), instead of an x86-based cluster. The x86 cluster used for the reference measurements was the Nehalem sub-cluster of the LiDong machine provided by TU Dortmund (ITMC TU Dortmund, 2015).

A study comparing the performance as well as the energy consumption of different low power and general-purpose architectures was published by Castro et al. (Castro et al., 2013). Based on the Traveling-Salesman problem (Applegate et al., 2011), they investigated time to solution and energy to solution for an Intel Xeon E5-4640 Sandy Bridge-EP, a low power Kalray MPPA-256 many-core processor (KALRAY Corporation, 2015), as well as for a low power CARMA board from SECO (NVIDIA Corporation, 2015a). The results show, that the CARMA board and the MPPA-256 many-core processor achieve better results than the Xeon 5 measured in terms of energy to solution. Concerning the time to solution, the Xeon 5 performed better than the CARMA board but not as good as the low power MPPA-256 many-core processor.

A work considering low power processors and accelerators in terms of energy aware high performance computing was published in 2013 (Rajovic et al., 2013b). There, a number of different HPC micro-benchmarks was used to determine the energy to solution. The architectures evaluated were NVIDIA Tegra 2 (NVIDIA Corporation, 2015b) and Tegra 3 (NVIDIA Corporation, 2015c) SoCs. The results show that drastic energy to solution improvements are possible on the newer Tegra 3 SoC in comparison to the Tegra 2 SoC (reduction of 67% on average). Furthermore, the authors conclude that the usage of integrated GPUs in low power architectures, such as Tegra 2 and Tegra 3, can improve the overall energy efficiency.

3 CONCEPT

Our goal is to develop an integrated hardware/software system that can satisfy (i) the functional requirements, i.e., computational performance, accuracy, and efficiency as well as (ii) the non-functional requirements, such as the energy efficiency and cost effectiveness. The project requirements are formulated in this slightly unusual vein, where the non-functional requirements are given comparable importance to the functional ones. This is caused by the fact that the operating environment

of the system sets a number of rather rigid constraints for the entire solution.

The flow of events in this setting provides the most important boundary condition. As soon as remote sensors provide a warning and supply data for a geophysical event (e.g., an earthquake) capable of causing flooding, the time remaining until the landfall of the wave is clearly defined. The available information then needs to be processed as fast as possible in order to predict the magnitude of the flood and identify the affected areas. In addition, from this moment on, the availability of uninterrupted power supply is critical in order to complete the flood simulation. One has to note here, that this hard upper bound for the time-to-solution can strongly vary on a case-by-case basis depending on a variety of factors (distance from shore, wave speed, area and topography of affected regions, etc.). If one also includes other non-functional requirements such as the size and speed of the available computational hardware, power source (external or battery), this clearly motivates the need of a highly adaptive flood modeling software in the warning system. Many of the regions at risk from such catastrophic events cannot boast of either good power infrastructure or reliable communication networks. Thus the minimum set of requirements must include the following two: (i) the hardware platform must be able – if such need arises – to complete the simulation on battery power (ii) the computation must be carried out locally.

Since the algorithm can be executed in highly parallel manner, we will focus on parallel computer architectures. As mentioned in the introduction, GPGPU computing is not in the scope of our work. We will simulate the different functional as well as non-functional properties of the computation of the flood simulation on different virtual hardware platforms first. This enables us to produce a preliminary decision on which hardware to use, thus saving resources and time during the development phase. With regard to the non-functional properties of the system, we need to know the total computation time and to obtain an estimate of the energy consumption for the entire run.

In order to obtain optimal results for our simulations, we need to limit the number of possible computer architectures. E.g., though the Intel x86 architecture is known to deliver high computation performance, this architecture is not in our scope due to its high power consumption when running high computation loads. Yet, along with the improvements of smart phones, the ARM CPU platform has strongly evolved as well and turned into a high performance and low power consumption architecture. It enables

today's devices to run on battery power for a long time under relatively high computational loads.

Another possible architecture is the well-known sparc-v8 architecture. Here we choose the LEON processor (an open source sparc processor developed by AEROFLEX Gaisler), as there have been successful investigations on estimation of non-functional properties for it. Furthermore, the sparc-v8 powered LEON processor is well suited to our purposes due to its fault tolerance, high configurability, and a relatively cost-effective licensing. Therefore we concentrate our investigations on the promising ARM platform as well as the LEON processor family.

4 PRELIMINARY EVALUATION OF SIMULATION ENVIRONMENTS

We intend to focus on building heterogeneous as well as homogeneous low power multi- and many-core architectures. Therefore, virtual environments that enable the simulation of those systems are of interest for us and have to be evaluated with respect to their suitability for our purposes. There are three further key aspects that are of importance for our choice: Simulation performance, the capability of power estimation and modeling, and last but not least, the availability of low power processor models. The next paragraphs briefly describe available multi- and many-core simulation environments and discuss their suitability for our purposes; Table 1 gives an overview of chief parameters of each simulation environment relevant in the context of our project.

An open-source multi- and many-core simulator is Graphite (Miller et al., 2010); it offers the possibility to simulate hundreds or even thousands of cores. Graphite is not a complete cycle-accurate simulator, it uses different techniques to provide accurate performance results. The simulation environment offers processors, a memory subsystem, cache models as well as a network for realizing interconnections. All these models use further analytical timing models to guarantee accurate results. Two processor models are supported, *iocoom* (in-order core model with out-of-order memory completion) and *simple* (in-order core model that adds all latencies). Power modeling for the processor, the caches, and the network (Kurian et al., 2014) is also supported. However, the focus of Graphite is not on embedded systems and low power architectures.

Based on the Graphite simulation infrastructure Carlson et al. (Carlson et al., 2011) developed Sniper.

Table 1: Comparison of different simulation environments for our purposes.

Simulation Environment	Multi- and Many-Core Simulations Possible	Simulation Speed ¹	Power Estimation Possible	Availability of Low Power Processor Models
Graphite	yes	mid to high	yes	none
Sniper	yes	high	yes	none
SoCLib	yes	slow to mid	yes	mid
HORNET	yes	slow to mid	yes	very low
gem5	yes	slow to mid	yes	mid
QEMU	yes	very high	no	low
OVP	yes	very high	yes	high

¹ Speed of instruction accurate simulation is used as the reference value and thus for classification of simulation speed

Sniper enhances Graphite by an interval simulation approach, a more detailed timing model, and operating system modeling improvements. It allows faster and more precise simulations for exploring homogeneous and heterogeneous multi- and many-core architectures than Graphite. Sniper supports power modeling and estimation using McPAT (Li et al., 2013) and custom DRAM power models (Heirman et al., 2012). Unfortunately, Sniper is an x86-tool only.

An environment that focuses on virtual prototyping of multi-processor system on chips (MP-SoC) is SoCLib (SocLib Project, 2015). SoCLib provides a wide range of processor and peripheral models, for example MIPS32 and ARM. Furthermore, the usage of real-time operating systems like eCos is supported. This environment enables simulations at cycle- and bit-accurate level. Since all models are written in SystemC (Accellera Systems Initiative, 2015), the ability to simulate at transaction level is provided as well. To supply the power and energy estimations, Atitallah et al. (Atitallah et al., 2006) developed energy models for different hardware components that can be used in conjunction with SoCLib. In terms of simulation speed, the cycle accurate simulation level is very slow in comparison to the instruction accurate level: Weaver and McKee (Weaver and McKee, 2008) showed that discrepancies of hours up to days are possible. As a consequence, cycle accurate simulations are not an option for the simulation of large many-core systems in the context of our project.

A cycle-level multi- and many-core simulator for Network on Chip (NoC) architectures is HORNET (Lis et al., 2011). The simulator provides a variety of memory hierarchies, interconnect geometries as well as accurate power modeling. HORNET can operate in full multi-core mode, i.e., using a built-in MIPS core simulator in addition to the network model. Unfortunately, HORNET only offers one single-cycle in-order MIPS core. For increasing simulation performance, a loose synchronization mechanism is supported. As a result of loose synchronization, accuracy

of performance measurements suffers.

The gem5 simulation environment (Binkert et al., 2011) combines the benefits of the M5 (Binkert et al., 2006) and the GEMS ((GEMS Development Team, 2015)) environments. M5 is a configurable simulation environment offering multiple ISAs (instruction set architectures) as well as various CPU models. The CPUs can be configured to operate on different levels of detail and accuracy. In combination with GEMS, gem5 provides a detailed and flexible memory system as well as interconnection models. A wide range of instruction set architectures (e.g. x86, ALPHA, ARM or MIPS) is supported by gem5. For a short time, power modeling and estimation for low power ARM architectures is also possible (Endo et al., 2015). This simulation environment is not designed to be pure instruction accurate and targets low power architectures only partially.

QEMU (Bellard, 2005) is an emulator and virtual machine (VM) for the Intel x86 architecture, that can also emulate and virtualize a variety of systems of differing architectures. When used as an emulator, QEMU operates on an instruction accurate level. Typically, QEMU is used as a VM in hosting centres, but can also be used as a debugging platform for embedded ARM systems. QEMU is not meant to be an extensible framework, even though it is possible to implement new platforms. Among the emulated ARM platforms are, e.g., Nokia N810 tablet or ARM Versatile Express for Cortex-A9. QEMU does not support power measurements and estimations.

The instruction accurate simulation technology from Open Virtual Platforms (OVP) was developed for high performance simulation of low power multi- and many-core architectures. Simulations can run 100s of MIPS, often faster than in real time. Debugging applications, which run on the virtual hardware, as well as analysis of virtual platforms containing multiple processor and peripheral models is possible. A wide range of older and current processor models is available, e.g. for ARM, MIPS, Renesas,

or PowerPC processor families. A number of predefined platforms (Altera Cyclone V SoC, MIPS malta, etc.) is also available for the system. Furthermore, the OVP simulation technology offers the ability to create new processor architectures and other platform components (Imperas Software Limited, 2014). The OVP simulator supports measuring instruction counts within a program, thus permitting in-depth analysis of specific code fragments. Also power modeling for selected processors is possible using OVP – as introduced in (Rosa et al., 2014).

5 CONCLUSION

In this paper, we present a concept that enables the determination of suitable low power multi- and many-core architectures for tsunami and storm surge simulation. The realization of the concept relies on a virtual environment that enables emulation and simulation of different low power multi- and many-core hardware architectures. For that reason, we conducted a preliminary investigation of available multi- and many-core simulation environments. Three aspects played an important role in our choice: Simulation performance, the ability to estimate and model power consumption, and the availability of low power processor models in the simulation system. As can be seen from Table 1, OVP appears to be the best solution for the realization of our concept.

6 FUTURE WORK

Preliminary estimations of time-to-solution and energy consumption are necessary to improve design decisions when developing hardware architectures. This is especially true in the case of a power-aware flood warning system. As already discussed, cycle accurate simulation is not an option for developing a many core system due to the poor simulation speed. However, some work has already been done on utilizing high-level functional simulation enhanced by a mechanistic extension, thus including non-functional properties such as time and energy consumption on given hardware. Functional Simulators, such as Instruction Set Simulators (ISS) can be considered as interpreters of binary executables, simulating the internal registers of a given system. Our general approach is to estimate the energy- and time-to-solution by classifying a given instruction set into instruction categories, particularly regarding their non-functional characteristics. This can be done by using micro-benchmarks as well as existing data (Berschneider

et al., 2014). The process of categorizing and collecting the information can be regarded as an initial training phase. Once complete, the obtained information from each single category can thus be used to analyze an instruction mix with regard to its mean energy consumption, computation time, or other non-functional properties. The ISS is used to provide an instruction mix out of compiled binary executable for a given toolchain. This approach works well for simple embedded architectures with simple pipeline design, no caches, and in-order execution. Evaluations where already performed for different, mostly compute bound, algorithms, and the results where very promising so far (mean relative estimation errors of under 5%). Since the flood simulation algorithm is also compute bound, it seems also promising to choose this approach for our system as well. However, further research has to be done to include effects of more complicated pipeline structures, one or more data/instruction caches and possible out-of-order execution.

REFERENCES

- Accellera Systems Initiative (2015). Official systemc website. <http://www.systemc.org>. Last visit on 02.02.2015.
- Aizinger, V. and Dawson, C. (2002). A discontinuous galerkin method for two-dimensional flow and transport in shallow water. *Advances in Water Resources*, 25(1):67–84.
- Aizinger, V., Proft, J., Dawson, C., Pothina, D., and Negusse, S. (2013). A three-dimensional discontinuous galerkin model applied to the baroclinic simulation of corpus christi bay. *Ocean Dynamics*, 63(1):89113.
- Applegate, D., Bixby, R., Chvátal, V., and Cook, W. (2011). *The Traveling Salesman Problem: A Computational Study: A Computational Study*. Princeton Series in Applied Mathematics. Princeton University Press.
- Atitallah, R., Niar, S., Greiner, A., Meftali, S., and Dekeyser, J. (2006). Estimating energy consumption for an mpoc architectural exploration. In Grass, W., Sick, B., and Waldschmidt, K., editors, *Architecture of Computing Systems - ARCS 2006*, volume 3894 of *Lecture Notes in Computer Science*, pages 298–310. Springer Berlin Heidelberg.
- Bellard, F. (2005). QEMU, a Fast and Portable Dynamic Translator. In *USENIX Annual Technical Conference, FREENIX Track*, pages 41–46.
- Berschneider, S., Herglotz, C., Reichenbach, M., Fey, D., and Kaup, A. (2014). Estimating video decoding energies and processing times utilizing virtual hardware. In *Proc. 3PMCES Workshop. Design, Automation & Test in Europe (DATE)*.
- Binkert, N., Beckmann, B., Black, G., Reinhardt, S. K., Saidi, A., Basu, A., Hestness, J., Hower, D. R., Kr-

- ishna, T., Sardashti, S., Sen, R., Sewell, K., Shoaib, M., Vaish, N., Hill, M. D., and Wood, D. A. (2011). The gem5 simulator. *SIGARCH Comput. Archit. News*, 39(2):1–7.
- Binkert, N. L., Dreslinski, R. G., Hsu, L. R., Lim, K. T., Saidi, A. G., and Reinhardt, S. K. (2006). The m5 simulator: Modeling networked systems. *IEEE Micro*, 26:52–60.
- Carlson, T. E., Heirman, W., and Eeckhout, L. (2011). Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pages 52:1–52:12.
- Castro, M., Franceschini, E., Nguélé, T. M., and Méhaut, J.-F. (2013). Analysis of computing and energy performance of multicore, numa, and manycore platforms for an irregular application. In *Proceedings of the 3rd Workshop on Irregular Applications: Architectures and Algorithms*, IA3 '13, pages 5:1–5:8, New York, NY, USA. ACM.
- Dawson, C. and Aizinger, V. (2005). A discontinuous galerkin method for three-dimensional shallow water equations. *Journal of Scientific Computing*, 22(1-3):245–267.
- Endo, F. A., Couroussé, D., and Charles, H.-P. (2015). Micro-architectural simulation of embedded core heterogeneity with gem5 and mcpat. In *Proceedings of the 2015 Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools*, RAPIDO '15, pages 7:1–7:6, New York, NY, USA. ACM.
- GEMS Development Team (2015). Official gems website. <http://research.cs.wisc.edu/gems/>. Last visit on 02.02.2015.
- Göddeke, D., Komatitsch, D., Geveler, M., Ribbrock, D., Rajovic, N., Puzovic, N., and Ramirez, A. (2013). Energy efficiency vs. performance of the numerical solution of pdes: An application study on a low-power arm-based cluster. *J. Comput. Phys.*, 237:132–150.
- Heirman, W., Sarkar, S., Carlson, T. E., Hur, I., and Eeckhout, L. (2012). Power-aware multi-core simulation for early design stage hardware/software co-optimization. In *International Conference on Parallel Architectures and Compilation Techniques (PACT)*.
- Imperas Software Limited (2014). *OVP Guide to Using Processor Models*. Imperas Buildings, North Weston, Thame, Oxfordshire, OX9 2HA, UK. Version 0.5, docs@imperas.com.
- ITMC TU Dortmund (2015). Official lido website. <https://www.itmc.uni-dortmund.de/dienste/hochleistungsrechnen/lido.html>. Last visit on 26.03.2015.
- KALRAY Corporation (2015). Official kalray mppa processor website. <http://www.kalray-inc.com/kalray/products/#processors>. Last visit on 31.03.2015.
- Kurian, G., Neuman, S., Bezerra, G., Giovinazzo, A., Devadas, S., and Miller, J. (2014). Power modeling and other new features in the graphite simulator. In *Performance Analysis of Systems and Software (ISPASS), 2014 IEEE International Symposium on*, pages 132–134.
- Li, S., Ahn, J. H., Strong, R. D., Brockman, J. B., Tullsen, D. M., and Jouppi, N. P. (2013). The mcpat framework for multicore and manycore architectures: Simultaneously modeling power, area, and timing. *ACM Trans. Archit. Code Optim.*, 10(1):5:1–5:29.
- Lis, M., Ren, P., Cho, M. H., Shim, K. S., Fletcher, C., Khan, O., and Devadas, S. (2011). Scalable, accurate multicore simulation in the 1000-core era. In *Performance Analysis of Systems and Software (ISPASS), 2011 IEEE International Symposium on*, pages 175–185.
- Miller, J., Kasture, H., Kurian, G., Gruenwald, C., Beckmann, N., Celio, C., Eastep, J., and Agarwal, A. (2010). Graphite: A distributed parallel simulator for multicores. In *High Performance Computer Architecture (HPCA), 2010 IEEE 16th International Symposium on*, pages 1–12.
- NVIDIA Corporation (2015a). Official nvidia seco development kit website. <https://developer.nvidia.com/seco-development-kit>. Last visit on 31.03.2015.
- NVIDIA Corporation (2015b). Official nvidia tegra 2 website. <http://www.nvidia.com/object/tegra-superchip.html>. Last visit on 27.03.2015.
- NVIDIA Corporation (2015c). Official nvidia tegra 3 website. <http://www.nvidia.com/object/tegra-3-processor.html>. Last visit on 27.03.2015.
- Rajovic, N., Carpenter, P. M., Gelado, I., Puzovic, N., Ramirez, A., and Valero, M. (2013a). Supercomputing with commodity cpus: Are mobile socs ready for hpc? In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, SC '13*, pages 40:1–40:12, New York, NY, USA. ACM.
- Rajovic, N., Rico, A., Puzovic, N., Adeniyi-Jones, C., and Ramirez, A. (2014). Tibidabo: Making the case for an arm-based {HPC} system. *Future Generation Computer Systems*, 36(0):322 – 334. Special Section: Intelligent Big Data Processing Special Section: Behavior Data Security Issues in Network Information Propagation Special Section: Energy-efficiency in Large Distributed Computing Architectures Special Section: eScience Infrastructure and Applications.
- Rajovic, N., Rico, A., Vipond, J., Gelado, I., Puzovic, N., and Ramirez, A. (2013b). Experiences with mobile processors for energy efficient hpc. In *Proceedings of the Conference on Design, Automation and Test in Europe, DATE '13*, pages 464–468, San Jose, CA, USA. EDA Consortium.
- Rosa, F., Ost, L., Raupp, T., Moraes, F., and Reis, R. (2014). Fast energy evaluation of embedded applications for many-core systems. In *Power and Timing Modeling, Optimization and Simulation (PATMOS), 2014 24th International Workshop on*, pages 1–6.
- SocLib Project (2015). Official soclib developer website. <http://www.soclib.fr/trac/dev>. Last visit on 01.02.2015.
- Weaver, V. M. and McKee, S. A. (2008). Are cycle accurate simulations a waste of time? In *Proc. 7th Workshop on Duplicating, Deconstructing, and Debunking*.