



HAL
open science

MAGPIE: System-level Evaluation of Manycore Systems with Emerging Memory Technologies

Thibaud Delobelle, Pierre-Yves Péneau, Abdoulaye Gamatié, Florent
Bruguier, Sophiane Senni, Gilles Sassatelli, Lionel Torres

► **To cite this version:**

Thibaud Delobelle, Pierre-Yves Péneau, Abdoulaye Gamatié, Florent Bruguier, Sophiane Senni, et al..
MAGPIE: System-level Evaluation of Manycore Systems with Emerging Memory Technologies. EMS:
Emerging Memory Solutions, Mar 2017, Lausanne, Switzerland. lirmm-01467328

HAL Id: lirmm-01467328

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01467328v1>

Submitted on 1 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MAGPIE: System-level Evaluation of Manycore Systems with Emerging Memory Technologies

Thibaud Delobelle, Pierre-Yves Péneau, Abdoulaye Gamatié, Florent Bruguier,
Sophiane Senni, Gilles Sassatelli, and Lionel Torres
LIRMM - CNRS - University of Montpellier

I. INTRODUCTION

The design complexity of high-performance embedded systems is getting very challenging as they keep on integrating more cores with innovative communication and memory technologies, in order to efficiently achieve various functionalities. This trend is typically observed in chips embedded within smartphones such as the Exynos 5 Octa one [1]. It relies on the recent ARM big.LITTLE technology composed of eight cores: four Cortex-A15 high-performance cores for heavy workloads and four Cortex-A7 low-power cores for light workloads.

Traditional memories are known as a major obstacle to energy-proportionality and their static power consumption is becoming the most important energy budget. Integrating emerging non-volatile memory (NVM) technologies [2] in system design is a very promising way to reach a complete energy-proportionality.

In this paper, we present a holistic design evaluation tool, named MAGPIE explorer, which aims at manycore systems integrating emerging NVM technologies for energy-efficiency. MAGPIE stands for *Manycore Architecture energy and Performance evaluation Environment*. It is built upon three mature and popular tools: the gem5 full-system simulator [3], the McPAT [4] and NVSim [5] power/energy and area estimation tools for CMOS, SOI and NVM technologies. MAGPIE explorer enables a designer to specify input design specifications and run a seamless evaluation flow that automatically produces results including application outputs, performance numbers, area, power and energy consumption.

II. MAGPIE FRAMEWORK

A. Evaluation workflow

MAGPIE explorer relies on a generic evaluation flow depicted in Fig. 1. The inputs of the flow comprises information related to the software and hardware parts of the system. The software-related inputs include a gem5 execution script file for each workload/application to be executed, together with the underlying operating system supported by the considered full-system simulator. From the hardware perspective, a number of parameters of the target manycore architectures are required: types and number of cores, memory hierarchy and its technology-specific properties, and the interconnect type.

Provided the above input information, MAGPIE proceeds through four main steps as follows:

- 1) Platform components calibration: the basic parameters of all hardware components are setup in the full-system simulator. Typically, the operating frequency of cores, the memory size, and access latencies at different levels of the memory hierarchy are defined.
- 2) Manycore system execution: the full system simulation of the user-customized system is performed to obtain detailed execution statistics. Note that since the inputs of the MAGPIE flow can specify at the same time different system design choices, several simulation instances can be launched in parallel.
- 3) Surface and energy estimation: based on detailed execution statistics generated by gem5, the surface, power and energy consumption of the simulated system are estimated. For instance, the dynamic energy of a cache memory is determined based on its collected read/write activity events and the energy consumption of elementary memory access.
- 4) Postprocessing for graphical renderings: as a major goal of MAGPIE is to assist the user in design space exploration, the final evaluation metrics can be reported in both textual and graphical formats. Then the user can quickly gather insights from each evaluated design.

B. Implementation

We seamlessly combine the gem5 simulator with NVSim and McPAT. These tools are briefly described below.

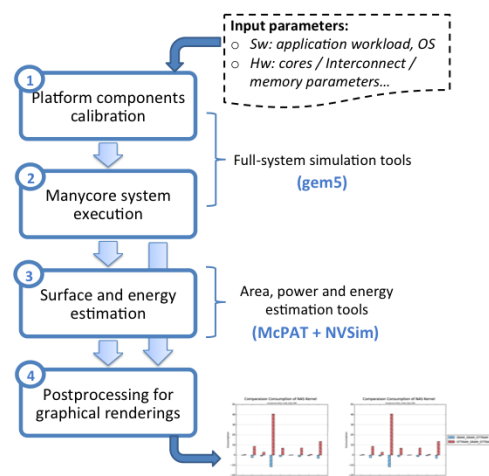


Fig. 1. MAGPIE evaluation flow.

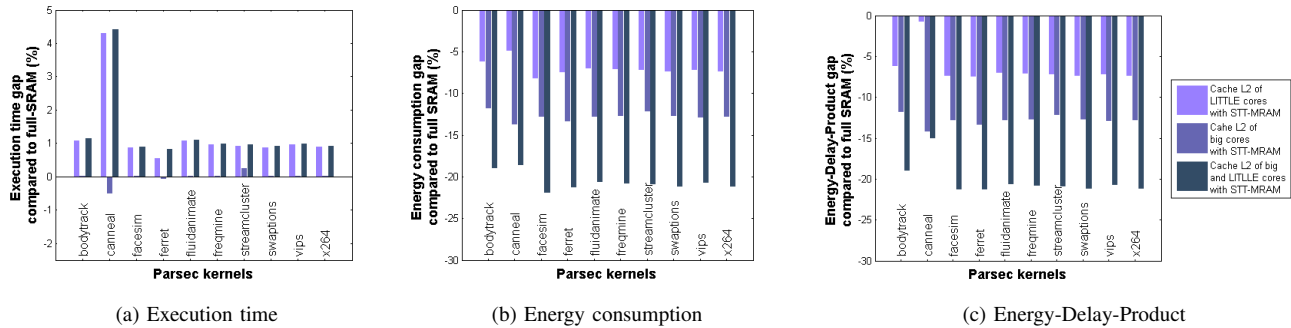


Fig. 2. Comparison of execution time, energy and EDP gap for Parsec: L2 caches in STT-MRAM versus full SRAM L2 caches (reference configuration)

1) *Considered simulation and estimation tools:* gem5 [3] provides an accurate evaluation of system performance [6] thanks to its high configurability for a fine grained architecture modeling. McPAT [4] is a power, area and timing modeling framework for multithreaded, multicore, and manycore architectures which doesn't address NVMs. So, we also use NVSim [5], which is a circuit-level estimator for NVM performance, energy, and area estimations.

2) *Integration within MAGPIE explorer:* MAGPIE explorer defines several Python script programs that automate the whole flow. Its inputs are first read and used for an automatic calibration of the hardware architecture components in gem5. For NVMs, the NVSim tool is invoked by a script to calculate the corresponding read/write latencies based on the desired memory type, memory size, associativity and technology node, as specified in the inputs. Then, gem5 is automatically configured with the computed NVM access latencies. For this purpose, we modified gem5 so as to enable the configuration of memories with asymmetric read and write latencies, such as NVMs. Afterwards, the specified system execution scenarios are run in parallel by automatically triggering the corresponding number of gem5 simulation instances.

Each gem5 simulation instance produces the execution statistics file related its design scenario. From these files, all data required by NVSim and McPAT are automatically extracted by another script. As these files can be huge, the script has been defined in such a way that it optimizes the reading of generated gem5 files. Then, it invokes the two estimation tools on the extracted data to generate the area, power and energy consumption for each captured design scenario. The results are stored in textual files.

Finally, the above textual files are post-processed by several scripts for generating various user-friendly renderings: CSV files and graphical plots that compare the performance, area, power and energy evaluation of the different scenarios.

C. Outcomes

The Fig. 2 depicts results of a study based on an Exynos 5 Octa (5422) chip model including four LITTLE and four big cores. For 10 kernels of the Parsec 3.0 benchmark suite [7], we evaluated 4 system scenarios playing with L2 configurations.

This is equivalent to 40 evaluation instances. The overall evaluation comprising the generation of results takes about 2 hours in our machine (95% for gem5 simulation). With a non-automated approach, calibrating gem5 and exploiting its outputs with McPAT and NVSim to compute the same results takes at least 2 hours per evaluation instance (in addition to gem5 simulation time), assuming no mistake is made during value extraction from the generated statistics files. A manual approach would require around 2 hours for gem5 parallel simulations plus 80 hours for generating all results, i.e., an exploration time increase factor of 41 compared to MAGPIE explorer. With large input sets for kernel execution, our tool demonstrates an effort decrease factor of 6 compared to a manual approach. More generally, the effort in non-automated approach grows with the integration of NVMs in more cache memories as the designer must carefully select the right area and energy estimation tool with gem5-produced statistics.

III. CONCLUSIONS

This paper presented MAGPIE explorer, a design evaluation framework that aims at energy-efficient manycore systems integrating emerging NVM technologies. From design inputs provided by a user, a seamless evaluation flow automatically produces results including application outputs, performance numbers, area, power and energy consumption. Beyond the numerous useful information computed by MAGPIE, the design exploration efforts are significantly reduced by a factor between 6 and 41 on the case study, compared to similar non-automated error-prone approaches. MAGPIE therefore increases the productivity of designers.

REFERENCES

- [1] Samsung, "Exynos Octa SoC," <https://http://www.samsung.com/>, 2015.
- [2] S. Mittal, J. S. Vetter, and D. Li, "A survey of architectural approaches for managing embedded dram and non-volatile on-chip caches," *IEEE TPDS*, vol. 26, no. 6, pp. 1524 – 1537, June 2015.
- [3] "The gem5 simulator," <http://www.gem5.org>, 2016.
- [4] HP Labs, "McPAT Tool," <http://www.hpl.hp.com/research/mcpat/>, 2008.
- [5] X. Dong et al., "NVSim Tool," <http://nvsim.org>, 2016.
- [6] A. Butko, R. Garibotti, L. Ost, and G. Sassatelli, "Accuracy evaluation of gem5 simulator system," in *7th Int'l Workshop on Reconfigurable and Communication-Centric Systems-on-Chip (ReCoSoC)*, 2012, pp. 1–7.
- [7] Princeton University, "Parsec 3.0," <http://parsec.cs.princeton.edu>, 2016.