MAGPIE User Guide
(version 1.0)

June 2017

Authors: Sophiane Senni, Pierre-Yves Péneau, Abdoulaye Gamatié

Contents

1

About this guide

This document aims at getting started with the MAGPIE[1] framework. A brief presentation of the tool is given followed by a detailed example on how to use MAGPIE for specifying and simulating a typical multicore system.

If you use MAGPIE in your research, we would appreciate a citation to this paper in any publications you produce:
Thibaud Delobelle, Pierre-Yves Péneau, Abdoulaye Gamatié, Florent Bruguier, Sophiane Senni, Gilles Sassatelli and Lionel Torres. *MAGPIE: System-level Evaluation of Manycore Systems with Emerging Memory Technologies*, in EMS: Emerging Memory Solution, March 2017

[1] MAGPIE stands for (Manycore Architecture enerGy and Performance evaluatIon Environment). It is developed in the framework of the French ANR CONTINUUM project (http://www.lirmm.fr/continuum-project) and the GREAT (heteroGeneous integRated magnetic tEchnology using multifunctional stAndardized sTack) H2020 European project (http://www.great-research.eu).

2

Introduction

MAGPIE is built upon different tools in order to enable performance, energy and area evaluation of multicore and manycore architectures. Heterogeneous systems can be explored for instance by considering different kinds of processors or different memory technologies.

At memory level, the NVSim tool is used to estimate the access time, the access energy and the total area of non-volatile memory circuits. This information is extracted and applied so as to explore the impact of different memory technologies at system level thanks to gem5, a computer architecture simulator that is able to simulate a complete system including an operating system (OS).

The gem5 simulator generates a detailed report of the system activity including the number of memory transactions (e.g., number of reads/writes, number of cache hits/misses) and the execution time. This activity information is then used by McPAT, a power and area estimator for multicore and manycore architectures. As a result, MAGPIE can evaluate the performance, energy and area of a complete system including the processor cores, caches, buses, and the memory controller.

After setting up MAGPIE (see MAGPIE Installation Guide), the tree view of the framework should look like as in Figure 2.1.

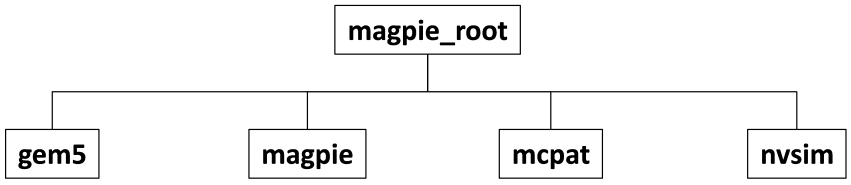

Figure 2.1: MAGPIE's tree view

The **gem5**, **mcpat** and **nvsim** directories contain the evaluation tools invoked by MAGPIE during each simulation. The **magpie** directory contains the scripts to manage a MAGPIE simulation and a configuration file to define the system architecture.

3

Getting started with MAGPIE

3.1 Cross-compiling an application

First of all, you may need to compile the application for the targeted CPU (ARM in this guide). As a simple example, you can write the following program in a file named **hello_magpie.c**:

```c
#include <stdio.h>

int main()
{
   printf("Hello MAGPIE");
   return 0;
}
```

Figure 3.1: A sketch of **hello_magpie.c** file

To compile the application, use the following commands:

```
$ cd /path-to-application-directory/
$ arm-linux-gnueabi-gcc -static -o hello_magpie hello_magpie.c
```

3.2 Copying an application into a disk image

The application has to be copied into the disk image that gem5 will boot up during a full system simulation. Use the following command to mount the disk image:

```
$ sudo mount -o loop,offset=32256 $M5_PATH/disks/linux-aarch32-
  ael.img /mnt/
```

Once the disk image is mounted, create a new directory named **benchmarks**, and copy the application **hello_magpie** into the **benchmarks** directory by using the following commands:

```
$ sudo mkdir /mnt/benchmarks
$ sudo cp /path-to-application-directory/hello_magpie /mnt/benchmarks/
```

Then, you can unmount the disk image with the following command:

```
$ sudo umount /mnt/ && sudo sync
```

To automatically execute the application after running a simulation, it is necessary to create a rcS file. This file is loaded by the gem5 simulator and is executed by the shell after the OS boots. You can create the following rcS file named **hello_magpie.rcS** for the **hello_magpie** application. The **hello_magpie.rcS** file has to be placed into the **boot** directory whose the path is **$GEM5/configs/boot/**.

```
#!/bin/sh

# Change directory to the benchmarks directory
cd /benchmarks

# Reset simuation statistics after the OS boots
/sbin/m5 resetstats

# Launch the execution of the application
./hello_magpie

# Exit the gem5 simulation
/sbin/m5 exit
```

Figure 3.2: A sketch of **hello_magpie.rcS** file

Finally, the **hello_magpie** application has to be added to the benchmarks list into the **Benchmarks.py** file located in **$GEM5/config/common/**. Edit the benchmarks list as follows:

```
# All benchmark should be like this :
Benchmarks = {
    # Demo Benchmark
    'hello':            'hello.rcS',
    'hello_magpie':     'hello_magpie.rcS',
}
```

Figure 3.3: Benchmarks list inside the Benchmarks.py file

3.3 Running a simulation

Before running a simulation, you may want to define the system architecture. A configuration file named **config.ini.example** is available as an example in the **magpie** directory. The default system architecture defined in this file corresponds to a 4-core architecture based on the Cortex-A15 processor, with one level of cache (size 32kB) for each core and a 2GB LPDDR3 DRAM main memory.

Copy the **config.ini.example** file and rename it to **hello_magpie_config.ini** by using the following commands:

```
$ cd path-to-magpie-root-directory/magpie/
$ cp config.ini.example hello_magpie_config.ini
```

Edit the following parameters in the **hello_magpie_config.ini** file as follows (a detailed description of each parameter is given in Section 3.4):

- **result-dir**=/path-to-magpie-root-directory/results
- **nvsim-result-dir**=/path-to-magpie-root-directory/results/nvsim
- **nvsim-dir**=/path-to-magpie-root-directory/nvsim
- **gem5-dir**=/path-to-magpie-root-directory/gem5
- **mcpat-dir**=/path-to-magpie-root-directory/mcpat
- **gem5tomcpat-dir**=/path-to-magpie-root-directory/magpie/parser
- **dtb-file**=/path-to-magpie-root-directory/gem5/util/binaries/vexpress.aarch-32.ll_20131205.0-gem5.4cpu.dtb
- **kernel**=/path-to-magpie-root-directory/gem5/util/binaries/vmlinux.aarch-32.ll_20131205.0-gem5
- **disk-image**=/path-to-magpie-root-directory/gem5/util/disks/linux-aarch32-ael-.img
- **benchmark**=hello_magpie

Note: To specify the application you desire to simulate, the **script** parameter can be used instead of the **benchmark** parameter. In this case, edit the **script** parameter with the full path to the **hello_magpie.rcS** file. Note that editing the benchmarks list into the **Benchmarks.py** file (Figure 3.3) is not necessary if the **script** parameter is used.

Now, you can launch your first MAGPIE simulation by using the following commands:

```
$ cd /path-to-magpie-root-directory/magpie/
$ ./magpie.py –configuration-file hello_magpie_config.ini
```

At the end of the simulation, a **results** directory should have been generated. The new tree view under the **magpie_root** directory should look like as in Figure 3.4.

The **gem5**, **mcpat** and **nvsim** directories (located in the **date-time** directory) respectively contain the outputs of the gem5, mcpat and nvsim tools. The remaining files are generated from MAGPIE. A brief description of these files is given below:

- **configuration.txt**:
  Summarizes all the values considered for each parameter in the configuration file.
- **stdout.txt**:
  A log file containing the messages printed by MAGPIE during the simulation.
- **magpie_time.txt**:
  Provides information regarding the time spent by the simulation.

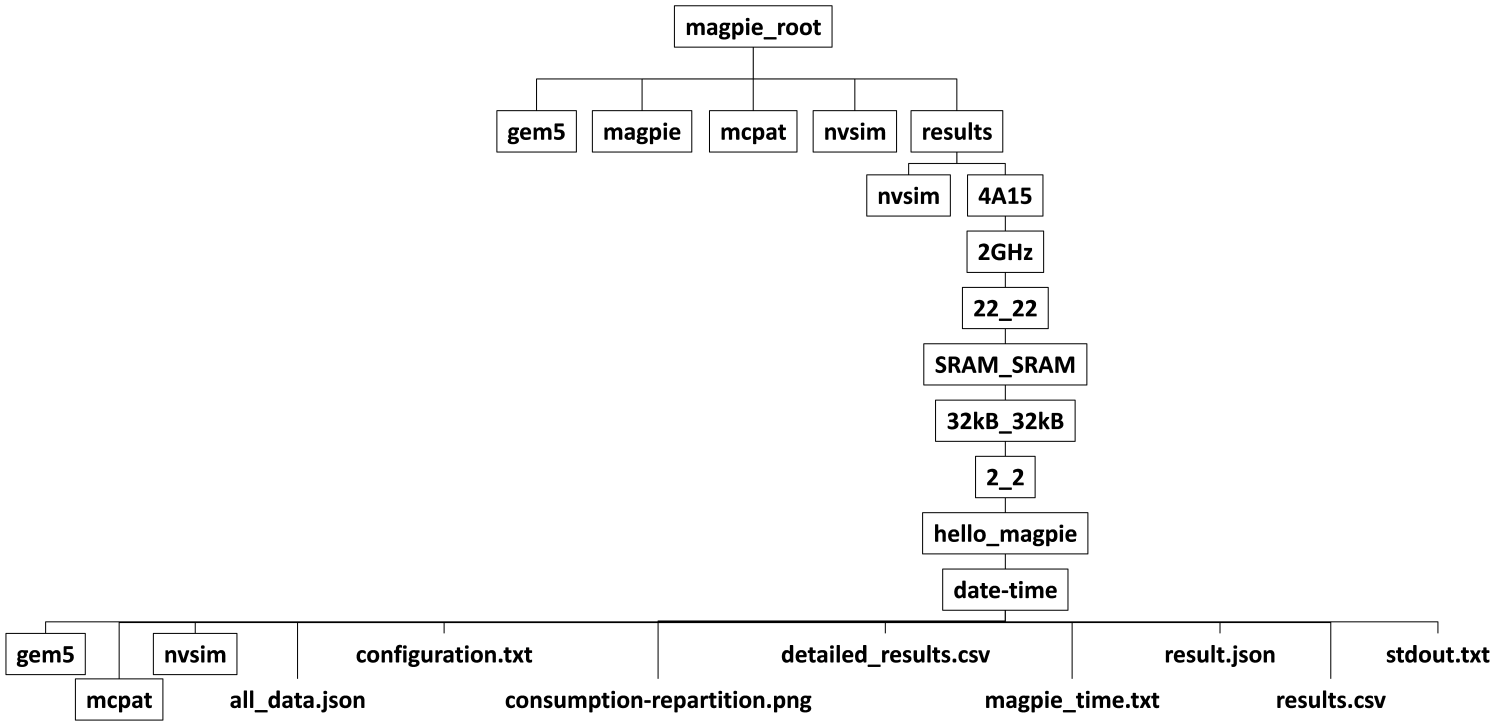

Figure 3.4: Organization of MAGPIE results tree

- **results.csv**:
  Contains the total performance, energy and area results of the simulated system.
- **detailed_results.csv**:
  In addition to the information given in **results.csv** , this file details the energy and area results of each part of the system (i.e. cores, cache memories, buses, memory controller).
- **all_data.json**:
  Contains detailed information results of the MAGPIE simulation.
- **result.json**:
  A subset of data contained in **all_data.json**.
- **consumption-repartition.png**:
  A figure showing the energy consumption of each part of the system.

3.4 MAGPIE configuration file description

The description of each parameter in the MAGPIE configuration file is given below. This configuration is just an example. A complete description with all possible values is given in **config.ini.example**.

- **result-dir**:
  Define the path to the MAGPIE simulation results.
- **nvsim-result-dir**:
  Define the path to the NVSim simulation results for future reuse.
- **nvsim-dir**:
  Define the path to NVSim directory.
- **gem5-dir**:
  Define the path to gem5 directory.
- **mcpat-dir**:
  Define the path to McPAT directory.
- **gem5tomcpat-dir**:
  Define the path to gem5tomcpat directory.
- **dtb-file**:
  Define the path to the dtb file which describe the underlying hardware.
- **kernel**:
  Define the path to the kernel.
- **disk-image**:
  Define the path to the disk image which contains your applications.
- **checkpoint-dir**:
  Define the path to the gem5 checkpoint to restore the simulation.
- **machine-type**:
  Define the machine type visible from the operating system.

- **benchmark**:
  Define the application to run. The available applications are listed inside the **Benchmarks.py** file located in **/path–to-gem5-directory/configs/common/**. You must have created a rcS file for each application (refer to Section 3.2).
- **script**:
  Define the application to run by giving the full path to the rcS file.
- **cacheline-size**:
  Define the cache line size in bytes.
- **mem-type**:
  Define the main memory controller model.
- **mem-size**:
  Define the main memory size.
- **technology-node**:
  Define the technology node (in nanometers) of the system.
- **num-cpus**:
  Define the number of cores.
- **cpu-type**:
  Define the type of core.
  Possible values: A15, A7, arm_detailed, detailed, minor.
- **cpu-clock**:
  Define the frequency of the cores.
- **l1i-type**:
  Define the memory technology for the L1 instruction cache.
  Possible values: SRAM, STTRAM, RRAM, PCRAM.
- **l1i-size**:
  Define the L1 instruction cache size.
- **l1i-assoc**:
  Define the associativity of the L1 instruction cache.
- **l1i-technology**:
  Define the technology node (in nanometers) for the L1 instruction cache.
  Possible values: 200, 130, 120, 90, 65, 45, 32, 22.
- **l1d-type**:
  Define the memory technology for the L1 data cache.
  Possible values: SRAM, STTRAM, RRAM, PCRAM.
- **l1d-size**:
  Define the L1 data cache size.
- **l1d-assoc**:
  Define the associativity of the L1 data cache.
- **l1d-technology**:
  Define the technology node (in nanometers) for the L1 data cache.
  Possible values: 200, 130, 120, 90, 65, 45, 32, 22.
- **l2**:
  Activate the L2 cache.

- **l2-type**:
  Define the memory technology for the L2 cache. This option is taken into account only if the option l2 is uncommented.
- **l2-size**:
  Define the L2 cache size. This option is taken into account only if the option l2 is uncommented.
- **l2-assoc**:
  Define the associativity of the L2 cache. This option is taken into account only if the option l2 is uncommented.
- **l2-technology**:
  Define the technology node for the L2 cache. This option is taken into account only if the option l2 is uncommented.
- **big-little**:
  Activate the heterogeneous mode. In this mode, a big.LITTLE architecture is considered for simulation. If this option is commented, the homogeneous mode is considered.
- **num-cpus-big**:
  Define the number of cores in heterogeneous mode for the big cluster.
- **cpu-type-big**:
  Define the type of core in heterogeneous mode for the big cluster.
  Possible values: A15, A7, arm_detailed, detailed, minor.
- **cpu-clock-big**:
  Define the frequency of the cores in heterogeneous mode for the big cluster.
- **num-cpus-little**:
  Define the number of cores in heterogeneous mode for the little cluster.
- **cpu-type-little**:
  Define the type of core in heterogeneous mode for the little cluster.
  Possible values: A15, A7, arm_detailed, detailed, minor.
- **cpu-clock-little**:
  Define the frequency of the cores in heterogeneous mode for the little cluster.
- **l1i-type-big**:
  Define the memory technology for the L1 instruction cache in heterogeneous mode for the big cluster.
  Possible values: SRAM, STTRAM, RRAM, PCRAM.
- **l1i-size-big**:
  Define the L1 instruction cache size in heterogeneous mode for the big cluster.
- **l1i-assoc-big**:
  Define the associativity of the L1 instruction cache in heterogeneous mode for the big cluster.
- **l1i-technology-big**:
  Define the technology node (in nanometers) for the L1 instruction cache in heterogeneous mode for the big cluster.
  Possible values: 200, 130, 120, 90, 65, 45, 32, 22.

- **l1d-type-big**:
  Define the memory technology for the L1 data cache in heterogeneous mode for the big cluster.
  Possible values: SRAM, STTRAM, RRAM, PCRAM.
- **l1d-size-big**:
  Define the L1 data cache size in heterogeneous mode for the big cluster.
- **l1d-assoc-big**:
  Define the associativity of the L1 data cache in heterogeneous mode for the big cluster.
- **l1d-technology-big**:
  Define the technology node (in nanometers) for the L1 data cache in heterogeneous mode for the big cluster.
  Possible values: 200, 130, 120, 90, 65, 45, 32, 22.
- **l2-type-big**:
  Define the memory technology for the L2 cache in heterogeneous mode for the big cluster. This option is taken into account only if the option l2 is uncommented.
- **l2-size-big**:
  Define the L2 cache size in heterogeneous mode for the big cluster. This option is taken into account only if the option l2 is uncommented.
- **l2-assoc-big**:
  Define the associativity of the L2 cache in heterogeneous mode for the big cluster. This option is taken into account only if the option l2 is uncommented.
- **l2-technology-big**:
  Define the technology node for the L2 cache in heterogeneous mode for the big cluster. This option is taken into account only if the option l2 is uncommented.
- **l1i-type-little**:
  Define the memory technology for the L1 instruction cache in heterogeneous mode for the little cluster.
  Possible values: SRAM, STTRAM, RRAM, PCRAM.
- **l1i-size-little**:
  Define the L1 instruction cache size in heterogeneous mode for the little cluster.
- **l1i-assoc-little**:
  Define the associativity of the L1 instruction cache in heterogeneous mode for the little cluster.
- **l1i-technology-little**:
  Define the technology node (in nanometers) for the L1 instruction cache in heterogeneous mode for the little cluster.
  Possible values: 200, 130, 120, 90, 65, 45, 32, 22.
- **l1d-type-little**:
  Define the memory technology for the L1 data cache in heterogeneous mode for the little cluster.
  Possible values: SRAM, STTRAM, RRAM, PCRAM.
- **l1d-size-little**:
  Define the L1 data cache size in heterogeneous mode for the little cluster.
- **l1d-assoc-little**:
  Define the associativity of the L1 data cache in heterogeneous mode for the little cluster.

- **l1d-technology-little**:
  Define the technology node (in nanometers) for the L1 data cache in heterogeneous mode for the little cluster.
  Possible values: 200, 130, 120, 90, 65, 45, 32, 22.
- **l2-type-little**:
  Define the memory technology for the L2 cache in heterogeneous mode for the little cluster. This option is taken into account only if the option l2 is uncommented.
- **l2-size-little**:
  Define the L2 cache size in heterogeneous mode for the little cluster. This option is taken into account only if the option l2 is uncommented.
- **l2-assoc-little**:
  Define the associativity of the L2 cache in heterogeneous mode for the little cluster. This option is taken into account only if the option l2 is uncommented.
- **l2-technology-little**:
  Define the technology node for the L2 cache in heterogeneous mode for the little cluster. This option is taken into account only if the option l2 is uncommented.
- **l3**:
  Activate the L3 cache.
- **l3-type**:
  Define the memory technology for the L3 cache regardless of the mode (homogeneous or heterogeneous). This option is taken into account only if the option l3 is uncommented.
- **l3-size**:
  Define the L3 cache size regardless of the mode (homogeneous or heterogeneous). This option is taken into account only if the option l3 is uncommented.
- **l3-assoc**:
  Define the associativity of the L3 cache regardless of the mode (homogeneous or heterogeneous). This option is taken into account only if the option l3 is uncommented.
- **l3-technology**:
  Define the technology node for the L3 cache regardless of the mode (homogeneous or heterogeneous). This option is taken into account only if the option l3 is uncommented.