

MAGPIE Installation Guide

(version 1.0)

June 2017

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

Contents

1	About this guide	3
2	Framework installation	4
2.1	Dependencies	4
2.2	MAGPIE utilities setup	5
2.3	Gem5 setup	5
2.4	NVSim setup	7
2.5	McPAT setup	7
2.6	MAGPIE build tree	7

About this guide

This document provides all the steps to set up the MAGPIE¹ framework. This guide has been tested on a machine running Ubuntu 16.04 LTS. It is expected to work on most Linux distributions.

¹MAGPIE 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

Framework installation

As a preliminary step, it is recommended that the user creates a new directory named `magpie_root`, as follows:

```
$ mkdir magpie_root
```

2.1 Dependencies

Use the command `sudo apt-get install <package-name>` to install the following packages:

- python
- python-dev
- python-jinja2
- python-matplotlib
- python-pydot (optional)
- python-psutil (optional)
- g++
- gcc
- scons
- swig
- zlib1g-dev
- m4
- libc6-armel-cross
- libc6-dev-armel-cross
- binutils-arm-linux-gnueabi
- libncurses5-dev
- gcc-4.9-arm-linux-gnueabi
- gcc-4.9-multilib
- g++-4.9-arm-linux-gnueabi
- g++-4.9-multilib

- device-tree-compiler

```
$ sudo apt-get update
$ sudo apt-get install python python-dev python-jinja2 python-matplotlib
python-pydot python-psutil g++ gcc scons swig zlib1g-dev m4 libc6-armel-cross
libc6-dev-armel-cross binutils-arm-linux-gnueabi libncurses5-dev gcc-4.9-arm-
linux-gnueabi gcc-4.9-multilib g++-4.9-multilib device-tree-compiler
```

The `gcc-4.9-arm-linux-gnueabi` package provides the cross-compiler for ARM processors. Once this package is installed, the command line to invoke the cross-compiler is `arm-linux-gnueabi-gcc-4.9`. Use the following command to create a symbolic link named `arm-linux-gnueabi-gcc` which refers to the arm cross-compiler:

```
$ sudo ln -s /path-to-cross-compiler/arm-linux-gnueabi-gcc-4.9 /path-to-cross-
compiler/arm-linux-gnueabi-gcc
```

To get the path to the cross-compiler, the user can launch the command `whereis arm-linux-gnueabi-gcc-4.9`.

2.2 MAGPIE utilities setup

These utilities consist of a set of scripts that automate the complete simulation of a specified system.

Download the MAGPIE utilities within the `magpie_root` directory:

```
$ cd /path-to-magpie-root-directory/
$ git clone https://link/magpie.git
```

2.3 Gem5 setup

Download the gem5 simulator into the `magpie_root` directory:

```
$ cd /path-to-magpie-root-directory/
$ git clone https://gem5.googlesource.com/public/gem5
```

Then, get the suitable version of gem5 for the MAGPIE framework and apply the `gem5.diff` patch located in `/path-to-magpie-root-directory/magpie/patches`, by using the following commands (the warning messages can be ignored once the patch is applied):

```
$ cd /path-to-magpie-root-directory/gem5/
$ git checkout 2f3c467
$ git apply -p1 < /path-to-magpie-root-directory/magpie/patches/gem5.diff
```

Compile gem5 by building a basic ARM system:

```
$ cd /path-to-magpie-root-directory/gem5/
$ scons build/ARM/gem5.fast -j N //N is the number of threads you want to use for compiling
```

Set the new environment variable named GEM5 by using the following command:

```
For bash shell:
$ export GEM5=/path-to-magpie-root-directory/gem5
For tcsh shell:
$ setenv GEM5 /path-to-magpie-root-directory/gem5
```

Note that it is recommended to add the selected command in your favorite default shell configuration file.

Download the pre-compiled kernel and disk images for ARM full system simulation, and place them into \$GEM5/util directory by using the following commands:

```
$ wget -P /tmp/ http://www.gem5.org/dist/current/arm/aarch-system-2014-10.tar.xz
$ tar -C $GEM5/util -xf /tmp/aarch-system-2014-10.tar.xz
```

Note: it is possible to compile your own kernel for ARM full system. Further references are available in http://gem5.org/ARM_Kernel

Set the new environment variable named M5_PATH by using the following command.

```
For bash shell:
$ export M5_PATH=$GEM5/util
For tcsh shell:
$ setenv M5_PATH $GEM5/util
```

Here also, it is recommended to add the selected command in your favorite default shell configuration file.

2.4 NVSim setup

Download the NVSim simulator into the **magpie_root** directory. To get NVSim, follow the instructions at <http://www.nvsim.org>. Once NVSim is downloaded, apply the **nvsim.diff** patch located in **/path-to-magpie-root-directory/magpie/patches** (the warning messages can be ignored once the patch is applied). Finally, compile NVSim by applying the following commands:

```
$ cd /path-to-magpie-root-directory/nvsim/  
$ patch -p1 < /path-to-magpie-root-directory/magpie/patches/nvsim.diff  
$ make
```

2.5 McPAT setup

Download the McPAT simulator into the **magpie_root** directory by using the commands below:

```
$ wget -P /tmp/ https://storage.googleapis.com/google-code-archive-  
source/v2/code.google.com/mcpat/source-archive.zip  
$ unzip /tmp/source-archive.zip -d /path-to-magpie-root-directory/
```

Note: MAGPIE has been tested with McPAT version 1.3 (Feb. 2015). Check the version number in **/path-to-magpie-root-directory/mcpat/version.h**.

Apply the **mcpat.diff** patch located in **/path-to-magpie-root-directory/magpie/patches** (ignore the warning messages once the patch is applied), then compile McPAT:

```
$ cd /path-to-magpie-root-directory/mcpat/  
$ patch -p1 < /path-to-magpie-root-directory/magpie/patches/mcpat.diff  
$ make
```

2.6 MAGPIE build tree

Once the MAGPIE framework is set up, the tree view from the **magpie_root** directory should look like the structure shown in Figure 2.1.

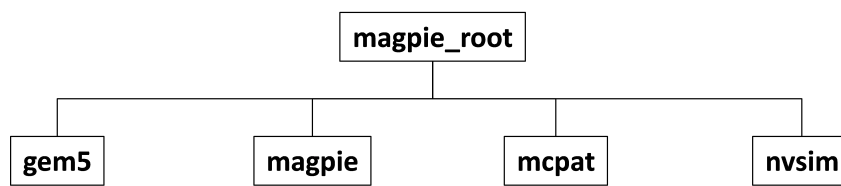


Figure 2.1: MAGPIE's tree view