

MAGPIE TUTORIAL

Application to real workload

Abdoulaye Gamatié, Pierre-Yves Péneau

LIRMM / CNRS-UM, Montpellier

ComPAS Conference, June 2017, Sophia-Antipolis

Other contributors: S. Senni, T. Delobelle, Florent Bruguier, L. Torres, G. Sassatelli

Application to real workload

- At this stage, you are able to:
 - Cross-compile an application
 - Create a disk image with this application
 - Automate the application execution with a `rcS` file
 - Configure MAGPIE to define your architecture
 - Run MAGPIE and analyze the outputs
- Let's try this with two well-known benchmark suites
 - Polybench¹
 - NAS parallel benchmark²

¹ L. N. Pouchet. "Polybench: The polyhedral benchmark suite". URL: <http://www.cs.ucla.edu/pouchet/software/polybench>, 2012

² D. H Bailey et al. "The NAS parallel benchmarks". *The International Journal of Supercomputing Applications*, 1991

Polybench configuration with laptop

- Prepare the compilation

```
$ cd $POLY && perl utilities/makefile-gen.pl -cfg
```

- Edit `config.mk` with appropriate changes (compiler, static flag)
 - Add `-DMINI_DATASET` in `CFLAGS=` to speedup the simulation

- Compile with

```
$ perl utililies/run-all.pl $POLY
```

- Copy few applications on your disk image in `$HOME/disks`
- Write your own `rcS` in `$HOME/app`
- Take a new checkpoint in `$CHKPT`

Name	2mm	covariance	heat-3d	lu	nussinov
Sim. time (minutes)	3'	3'	2'30	2'30	2'

Polybench configuration without laptop

- For those without laptop, we provide a disk image with pre-compiled binaries
 - `$M5_PATH/disks/linux-aarch32-ael-filled.img`
- Applications are in `/benchmark/polybench-c-4.2`
- Applications' names are listed in the table below
- Write your own rcS in `$HOME/app`
- Take a new checkpoint in `$CHKPT`

Name	2mm	covariance	heat-3d	lu	nussinov
Sim. time (minutes)	3'	3'	2'30	2'30	2'

Explorations with Polybench

- 1 core, Cortex-A15, full SRAM L1 caches, no L2, no L3,
- Same with L1 instruction and data in STTRAM 45nm
- Observe the execution time and EDP
- Add a L2 cache

NAS parallel benchmark configuration

- Prepare the compilation

```
$ cd $NAS && cp config/make.def.template \
  config/make.def
```

- Edit `$NAS/config/make.def` with appropriate compiler and flags
 - Add `-fopenmp` in `*_INC` and `*_LIB` to for OpenMP support
 - Don't forget `-static` !
- Compile **IS** application:

```
$ make CLASS=S is # Output in $NAS/bin
```

- Modify the disk image

NAS parallel benchmark configuration (no laptop)

- Pre-filled disk image in `$M5_PATH/disks/linux-aarch32-ael-filled.img`
- Binaries are in `/benchmark/NPB3.3.1`
- Application' name: `is.S.x`
- Execution time : around 30 minutes

Explorations with NAS

- Take two checkpoints in \$CHKPT:
 - 2 cores
 - 4 cores
- Heterogeneous exploration:
 - 1 big core (A15) + 1 LITTLE core (A7)
 - 2 big core (A15) + 2 LITTLE core (A7)
 - Activate L2 cache for both
- Retry with NVM on caches