

Здравейте, ако четете това вероятно имате интерес да вземете курсов проект по КА при мен.

Курсовият проект ще се състои в изграждане и симулация на съвременна архитектура с помощта на симулаторът SniperSim. За начало той трябва да бъде инсталиран. В текущата папка ще намерите архивен файл - SNIPERSIM.zip в който са описани стъпките за инсталацията, както и инсталационните скриптове (Ако скриптовете не работят, моля свържете се с мен).

След като го инсталирате отивате в папката benchmarks и го тествате със следната команда:

```
snipersim@snipersim-VirtualBox:~/snipersim/sniper-7.4/benchmarks$ ./run-sniper -p splash2-fft -i test -n 4 -c gainestown
[SPLASH] Benchmarks to run: fft
```

Трябва да получите резултат подобен на:

```
[SPLASH] [----- Beginning of output -----]
[SNIPER] Warning: Unable to use physical addresses for shared memory simulation.
[SNIPER] Start
[SNIPER] -----
[SNIPER] Sniper using SIFT/trace-driven frontend
[SNIPER] Running pre-ROI region in  CACHE_ONLY mode
[SNIPER] Running application ROI in DETAILED mode
[SNIPER] Running post-ROI region in FAST_FORWARD mode
[SNIPER] -----
[RECORD-TRACE] Using the Pin frontend (sift/recorder)

FFT with Blocking Transpose
  1024 Complex Doubles
   1 Processors
  65536 Cache lines
   16 Byte line size
  4096 Bytes per page

[HOOKS] Entering ROI
[SNIPER] Enabling performance models
[SNIPER] Setting instrumentation mode to DETAILED
[SNIPER] Disabling performance models
[SNIPER] Leaving ROI after 0.68 seconds
[SNIPER] Simulated 0.3M instructions, 1.4M cycles, 0.25 IPC
[SNIPER] Simulation speed 494.7 KIPS (494.7 KIPS / target core - 2021.6ns/instr)
[SNIPER] Sampling: executed 12.17% of simulated time in detailed mode
[SNIPER] Setting instrumentation mode to FAST_FORWARD
[TRACE:0] -- DONE --
[HOOKS] Leaving ROI

      PROCESS STATISTICS
Proc  Computation  Transpose  Transpose
      Time       Time       Fraction
   0         61         8         0.13115

      TIMING INFORMATION
Start time           : -1844408243
Initialization finish time : -1844407867
Overall finish time   : -1844407806
Total time with initialization : 437
Total time without initialization : 61
Overall transpose time : 8
Overall transpose fraction : 0.13115

[SNIPER] End
[SNIPER] Elapsed time: 4.37 seconds
[SPLASH] [----- End of output -----]
[SPLASH] Done.
```

Флаговете от командния ред можете да видите с командата ./sniper-sim

```
snipersim@snipersim-VirtualBox:~/sniperSim/sniper-7.4/benchmarks$ ./run-sniper
Run benchmark under the Sniper simulator
Usage:
./run-sniper -p <program> -i <inputsizes (test)> -n <ncores (1)> -m <machines (1)> -d <outputdir (.)> -c <config-file> -r <sniper-root-dir> -g <options>
Benchmarks:
cpu2006:
perlbench bzip2 gcc bwaves gamesss mcf milc zeusmp gromacs cactusADM leslie3d namd gobmk dealII soplex povray calculix hmmer sjeng
GemsFDTD libquantum h264ref tonto lbm omnetpp astar wrf sphinx3 xalancbmk
jikes:
lusearch jythion antlr pmd avrora fop luindex xalan bloat eclipse lusearch.fix hsqldb sunflow pjjb2005
local:
pi
npb:
bt cg dc ep ft is lu mg sp ua
parsec:

splash2:
barnes cholesky fft fft_00 fft_01 fft_02 fft_03 fft_forever fft_rep2 fmm lu.cont lu.ncont ocean.cont ocean.ncont radiosity radix
raytrace raytrace_opt volrend water.nsq water.sp barnes-scale fft-scale fmm-scale lu.cont-scale lu.ncont-scale ocean.cont-scale radix
-scale water.nsq-scale
```

-**p** е програмата за симулация. В понеже ще използваме само splash2 пакетът, синтаксисът е splash2-<benchmarkName> (в примера това е fft) (може да зададете и цял път, ако искате).

-**i** Обем на симулационните данни (test,small,medium,big). Ние ще използваме **small**.

-**n** броят на ядрата на които ще се изпълнява програмата - нека да отговаря на броят на ядрата във вашата конфигурация.

-**c** – Конфигурацията която ще се симулира. Тук поставяте вашата новосъздадена конфигурация.

След като пуснете всеки от бенчмарковете(програмите), резултатът от симулацията ще се изведе в текстови файл sim.out. Той изглежда така:

The screenshot shows a web browser interface for SniperSim benchmarks. The main directory view lists folders like cpu2006, jikes, local, npb, parsec, splash2, and tools, along with files like cc-sniper.bldconf, install_cpu2006, Makefile, run-sniper, sim.cfg, sim.info, and sim.out. A detailed view of the sim.out file is shown in the foreground, displaying performance metrics for four cores (Core 0 to Core 3).

	Core 0	Core 1	Core 2	Core 3
Instructions	97556	95203	94336	93806
Cycles	544935	544935	544935	544935
IPC	0.18	0.17	0.17	0.17
Time (ns)	204863	204863	204863	204863
Idle time (ns)	166371	176205	175972	179214
Idle time (%)	81.2%	86.0%	85.9%	87.5%
Branch predictor stats				
num correct	8428	8125	7985	7900
num incorrect	650	568	569	584
misprediction rate	7.16%	6.53%	6.65%	6.88%
mpki	6.66	5.97	6.03	6.23
TLB Summary				
I-TLB				
num accesses	8176	8026	7924	7857
num misses	38	39	41	40
miss rate	0.46%	0.49%	0.52%	0.51%
mpki	0.39	0.41	0.43	0.43
D-TLB				
num accesses	24806	23478	23158	22890
num misses	71	73	72	65
miss rate	0.29%	0.31%	0.31%	0.28%
mpki	0.73	0.77	0.76	0.69
L2 TLB				
num accesses	109	112	113	105
num misses	95	102	105	102
miss rate	87.16%	91.07%	92.92%	97.14%
mpki	0.97	1.07	1.11	1.09

Тук са вашите резултати.

Нека те да бъдат сравнени с оригиналната gainestown архитектура, да бъдат създадени таблици в които да се документират разликите в получените параметри. Да се изчисли и представи в проценти подобрението/понижението на вашата архитектура спрямо базовата gainestown.

(бенчмарковете се намират в sniperSim/benchmarks папката)

Бенчмарковете които работят и трябва да бъдат тествани са следните:

<splash2->barnes, fmm, ocean.cont, ocean.ncont, radiosity, raytrace, water.nsq, water.sp, cholesky, fft, lu, lu.cont, radix.

Ако има проблем с тях свържете се с мен.

За създаването на конфигурация:

(конфигурациите се намират в sniperSim/config папката)

1) Избирате си процесор, който не е вече взет от друг студент. Тази информация ще намерите в описанието на папката в Мудъл. Избирате процесора от линка:

https://www.agner.org/optimize/instruction_tables.pdf (страница 8) **(Задължително се свържете с мен за да ви запиша)**

Принцип на работа: Конфигурациите (gainestown, който ще използваме за база) се извикват на йерархичен принцип. Когато се напише -с gainestown, първо се вика base.cfg, върху нея за записва nehalem.cfg, върху която се записва gainestown.cfg(при някои от тези конфигурации има дублиращи се параметри. Взява се параметърът, присъстващ в най-високото ниво на йерархията(или с други думи последният записан)). **Затова първо изпълнете тестовете с базовата конфигурация(gainestown), документирайте ги и чак след това правете промени.** (или копирайте файловете и ги сменяйте при нужда).

Как да си намерите параметрите на вашият процесор:

Голяма част от симулационните характеристики са търговска тайна, понеже процесорите на AMD и Intel са closed-source. За щастие съществуват изследвания.

Нека config файловете да се променят по следния начин (може да промените и само най-горният, като той ще презапише тези на по-ниско ниво(base и nehalem)). Създайте си свой собствен файл като копирате gainestown.

[general] -> да не се пипа

[log] -> да не се пипа

[progress_trace] -> да не се пипа

[clock_skew_minimization] -> да не се пипа

[clock_skew_minimization/barrier] -> да не се пипа

[perf_model/core] -> намерете от сайта на производителя(това е публична информация)

[perf_model/core/interval_timer] -> тези обикновено се пазят в тайна от производителите, но има направени изследвания - <https://www.agner.org/optimize/microarchitecture.pdf>

[perf_model/core/static_instruction_costs] -> https://www.agner.org/optimize/instruction_tables.pdf

[perf_model/branch_predictor] -> <https://www.agner.org/optimize/microarchitecture.pdf>

[perf_model/tlb] -> <https://unix.stackexchange.com/questions/113585/how-to-find-the-page-size-associativity-and-tlb-size-and-number-of-entries> , ако нямате процесора, размера на tlb е публична информация.Потърсете модела + tlb в някоя търсачка.

[perf_model/l1_dcache] -> <https://www.agner.org/optimize/microarchitecture.pdf>

[perf_model/l2_cache] -> <https://www.agner.org/optimize/microarchitecture.pdf>

[perf_model/l3_cache] -> <https://www.agner.org/optimize/microarchitecture.pdf>

[perf_model/l4_cache] -> да не се пипа (освен ако нямате l4 cache)

[perf_model/llc] -> last level cache, да се променя при необходимост и компетентност.

[perf_model/fast_forward] -> да не се пипа

[perf_model/fast_forward/oneipc] -> да не се пипа

[core] -> да не се пипа

[core/light_cache] -> Това не знам за какво е. Да се променя при компетентност.

[core/hook_periodic_ins] -> да не се пипа

[caching_protocol] -> Да се променя само при налични данни, обикновено е mesi.

[perf_model/dram*] -> Ram паметта е външна за чипа, нека студента да избере модел, или да остави базовия.

***** под гореспоменатите параметри нищо да не се пипа, освен планировчика (scheduler). Той се определя от операционната система, което означава, че е софтуерен елемент. Променете го по ваша преценка. (pinned, dynamic, big-small) са част от вариантите.