|  |  |  |
| --- | --- | --- |
| Z:\Meu Drive\03. Doutorado\30. Doutorado IC-Unicamp\2022-2\Technical Report Template\logo-unicamp-name-line-blk-red-0120.png | **Universidade Estadual de Campinas**  **Instituto de Computação**  **Arquitetura de Computadores II – MO601**  **Prof. Rodolfo Jardim de Azevedo** | Z:\Meu Drive\03. Doutorado\30. Doutorado IC-Unicamp\2022-2\Technical Report Template\logo-ic-unicamp-slant-tint-beg-sky-ora-0120.png |

**Projeto 3**

**Experimentar ferramentas e coletar dados**

**Rubens de Castro Pereira**

**RA 217146**

Campinas – SP

Maio de 2023

**Índice**

[1 Introdução 3](#_Toc136266721)

[2 Ambiente de Experimentação 3](#_Toc136266722)

[3 Ferramentas experimentadas 4](#_Toc136266723)

[3.1 SPEC CPU 2017 benchmark \* 4](#_Toc136266724)

[3.2 Simulador multi-core Sniper \* 7](#_Toc136266725)

[3.3 Perf profiler \* 9](#_Toc136266726)

[3.4 PARSEC Benchmark Suite 3.0 \* 11](#_Toc136266727)

[3.5 Rodinia benchmark \* 16](#_Toc136266728)

[3.6 Intel Pin 19](#_Toc136266729)

[3.7 Dinero cache simulator 21](#_Toc136266730)

[4 Considerações sobre o aprendizado nesse projeto 24](#_Toc136266731)

# Introdução

Esse trabalho tem o propósito de utilizar algumas ferramentas de avaliação de arquitetura de computadores com a coleta de dados da execução de *benchmarks* e programas que exploram aspectos como tempo de processamento, número de instruções executadas e uso de memória RAM e cache. As ferramentas utilizadas foram SPEC CPU 2017, simulador multi-core Sniper, Perf profiler, Parsec benchmark, Rodinia benchmark, Intel Pin e Dinero cache simulator.

Os resultados obtidos na execução das ferramentas estão organizados no repositório Github por meio do link <https://github.com/rubenscp/RCP-MO601-Project-03>.

A Seção 2 apresenta o ambiente de experimentação, a Seção 3 detalhada a execução e resultados alcançados em cada ferramenta, a Seção 4 descreve considerações sobre o aprendizado neste projeto e a Seção 5 apresentas as conclusões do trabalho.

# Ambiente de Experimentação

O computador utilizado nesse trabalho será denominado como “Laptop Rubens” e o sistema operacional base é o Windows 10 Pro 22H2, contudo para a execução de todas as ferramentas foi utilizado o *Windows Subsystem for Linux* (WSL).

Os detalhes da configuração do Laptop Rubens são descritos a seguir:

* Notebook HP Pavilion dm4
* Memória RAM: 16 Gbytes
* SSD: 1 TBytes
* Sistema Operacional utilizado no Windows Subsystem for Linux:
  + Ubuntu 22.04.2 LTS (GNU/Linux 5.15.90.1-microsoft-standard-WSL2 x86\_64)
* CPU:
  + Model name: Intel(R) Core(TM) i7-2620M CPU @ 2.70GHz
  + Architecture: x86\_64
  + CPU op-mode(s): 32-bit, 64-bit
  + Address sizes: 36 bits physical, 48 bits virtual
  + Byte Order: Little Endian
  + CPU(s): 4
  + Vendor ID: GenuineIntel
  + CPU family: 6
  + Thread(s) per core: 2
  + Core(s) per socket: 2
  + L1d cache: 64 KiB (2 instances)
  + L1i cache: 64 KiB (2 instances)
  + L2 cache: 512 KiB (2 instances)
  + L3 cache: 4 MiB (1 instance)

# Ferramentas experimentadas

As ferramentas utilizadas para avaliações em arquitetura de computadores foram definidas previamente na especificação do projeto cujos detalhes de cada execução são apresentados nas próximas subseções.

## SPEC CPU 2017 benchmark \*

O SPEC CPU 2017 é um pacote de benchmark que contém a próxima geração de SPECs (*Standard Performance Evaluation Corporation*), pacotes de processamento intensivo de CPU para medição e comparação de desempenho computacional, sobrecarregando o processador do sistema, memória e compilador. Esta ferramenta oferece 4 suites para benchmark considerando velocidade (*speed*) e throughput (*rate*) para números inteiros e em ponto flutuante: intspeed, fpspeed, intrate e fprate.

A Tabela 1 apresenta o resumo da experimentação do SPEC CPU 2017 no Laptop Rubens indicando os parâmetros da execução, a duração da execução e a métrica final de execução produzida pela ferramenta. Os resultados detalhados desse experimento podem ser consultados na seção [SPEC CPPU 2017 do repositório Github](https://github.com/rubenscp/RCP-MO601-Project-03/tree/main/spec_cpu_2017/results%20laptop%20Rubens).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Resultados da execução do SPEC CPU 2017** | | | | | | |
| **Suíte** | **Cópias** | **Threads** | **Nº Iterações** | **Qtde de Benchmarks** | **Tempo de execução** | **Métrica Final (base)** |
| intspeed | 4 | 4 | 3 | 9 | 17993 s - 4,99 hs | **3,42** |
| intspeed | 8 | 8 | 3 | 10 | 18438 s – 5,12 hs | **3,35** |
| intspeed | 16 | 16 | 3 | 10 | 32523 s - 9,03 hs | **1,96** |
| intrate | 4 | 4 | 3 | 10 | 38073 s - 10,57 hs | **5,32** |
| intrate | 8 | 8 | 3 | 9 | 65121 s – 18,08 hs | **5,03** |
| fpspeed | 4 | 4 | 3 | 9 | 79708 s - 22,14 hs | **3,11** |
| fpspeed | 4 | 8 | 3 | 9 | ???????????? | **??????** |
| fprate | 4 | 4 | 3 | 13 | 58396 s - 16,22 hs | **6,25** |
| fprate | 8 | 1 | 3 | 14 | 124885 s - 34,69 hs | **5,82** |
| Duração total das execuções | | | | | ~~435137 s – 120,87 hs~~ |  |

Tabela 1. Suites executadas na ferramenta SPEC CPU 2017 com seus parâmetros da execução, o tempo de execução e a métrica final da execução.

A Tabela 2 apresenta a comparação das métricas produzidas no Laptop Rubens e de outros computadores selecionados a partir da lista de resultados disponíveis no site oficial da ferramenta SPEC CPU 2017 (<https://www.spec.org/cpu2017/results/cpu2017.html>). Os computadores selecionados são aqueles que mais se aproximam das características do computador Laptop Rubens a fim de que as comparações possam ser equilibradas e justas. Destaca-se que as métricas finais de Laptop Rubens foram obtidas em execuções sem exclusiva do computador, isto é, diversas outras tarefas não relacionadas aos benchmarks eram executadas simultaneamente.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Suite** | **Threads** | **Métrica obtida do Laptop Rubens** | **Outros computadores** | **Métrica** |
| intspeed | 4 | int\_base: 3,42 | SuperWorkstation 5039C-T (X11SCA , Intel Core i3-8100) | int\_base: 7,58 |
| intspeed | 8 | Int\_base: 3,35 | SuperWorkstation 5039C-T (X11SCA , Intel Core i7-9700K) | int\_base: 10,6 |
| intspeed | 16 | int\_base: 1,96 | Não localizado computador equivalente com thread = 16 | --- |
| intrate | 4 | int\_base: 5,32 | ASUS Z170M-PLUS Motherboard (Intel Core i7-6700K) | int\_base: 23,5 |
| intrate | 8 | int\_base: 5,03 | SuperWorkstation 5039C-T (X11SCA , Intel Core i7-9700K) | int\_base: 44,8 |
| fpspeed | 4 | fp\_base: 3,11 | SuperWorkstation 5039C-T (X11SCA , Intel Core i7-9700K) | fp\_base: 32,2 |
| fpspeed | 8 | fp\_base: ??? | SuperWorkstation 5039C-T (X11SCA , Intel Core i7-9700K) | fp\_base: 32,2 |
| fprate | 4 | fp\_base: 6,25 | SuperWorkstation 5039C-T (X11SCA , Intel Core i7-9700K) | fp\_base: 42,6 |
| fprate | 8 | fp\_base: 5,82 | SuperWorkstation 5039C-T (X11SCA , Intel Core i7-9700K) | fp\_base: 42,6 |

Tabela 2. Comparação das métricas dos benchmarks executados no laptop Rubens e outros computadores.

A Tabela 3 apresenta os comandos com os parâmetros utilizados na execução de cada uma das suítes.

|  |  |
| --- | --- |
| **Suite** | **Comando para execução do SPEC CPU 2017** |
| intspeed | runcpu --config=rubens-try1 --noreportable --iterations=3 600.perlbench\_s 602.gcc\_s 605.mcf\_s 620.omnetpp\_s 623.xalancbmk\_s 625.x264\_s 631.deepsjeng\_s 641.leela\_s 648.exchange2\_s 998.specrand\_is |
| intrate | runcpu --config=rubens-try1 --reportable --iterations=3 intrate |
| fpspeed | runcpu --config=rubens-try1 --noreportable --iterations=3 603.bwaves\_s 607.cactuBSSN\_s 619.lbm\_s 621.wrf\_s 628.pop2\_s 638.imagick\_s 644.nab\_s 649.fotonik3d\_s 654.roms\_s 996.specrand\_fs |
| fprate | runcpu --config=rubens-try1 --reportable --iterations=3 fprate |

Tabela 3. Comandos SPEC CPU 2017 executados para as suites intspeed, intrate, fpspeed e fprate.

## Simulador multi-core Sniper \*

Sniper é uma ferramenta de simulação de código voltada para a modelagem e análise do desempenho de sistemas multi-core explorando o comportamento do sistema para sua otimização.

A Tabela 4 apresenta os comandos utilizados na execução dos programas de teste que acompanham a ferramenta Sniper e o resultado da execução.

Os resultados detalhados desse experimento podem ser consultados na seção [Sniper do repositório Github](https://github.com/rubenscp/RCP-MO601-Project-03/tree/main/sniper).

|  |  |  |
| --- | --- | --- |
| **Comando para execução do programa** | **Resultado da execução do programa** | **Arquivo com o resultado da execução** |
| make run > sniper-result-api.txt | sucesso | sniper-result-api.txt |
| make run > sniper-result-dvfs.txt | sucesso | sniper-result-dvfs.txt |
| make run > sniper-result-fft.txt | sucesso | sniper-result-fft.txt |
| make run > sniper-result-fft-dvfs.txt | sucesso | sniper-result-fft-dvfs.txt |
| make run > sniper-result-fft-hetero.txt | sucesso | sniper-result-fft-hetero.txt |
| make run > sniper-result-fft-hetero-cfg.txt | erro | sniper-result-fft-hetero-cfg.txt |
| make run > sniper-result-fft-marker.txt | erro | sniper-result-fft-marker.txt |
| make run > sniper-result-fork.txt | sucesso | sniper-result-fork.txt |
| make run > sniper-result-shared.txt | sem programa fonte | sniper-result-shared.txt |
| make run > sniper-result-signal.txt | erro | sniper-result-signal.txt |
| make run > sniper-result-smc.txt | erro | sniper-result-smc.txt |
| make run > sniper-result-sniper-in-sniper.txt | erro | sniper-result-sniper-in-sniper.txt |
| make run > sniper-result-spinloop.txt | sucesso | sniper-result-spinloop.txt |
| make run > sniper-result-true.txt | sucesso | sniper-result-true.txt |
|  |  |  |
| **Programas adicionais** | | |
| ../../run-sniper ./RADIX > sniper-result-radix.txt | sucesso | sniper-result-RADIX.txt |
| ../../run-sniper ./CHOLESKY tk14.0 > sniper-result-CHOLESKY-tk14.O.txt | Sucesso | sniper-result-CHOLESKY-tk14.O.txt |
| ../../run-sniper ./CHOLESKY d750.O > sniper-result-CHOLESKY-d750.O.txt | Sucesso | sniper-result-CHOLESKY-d750.O.txt |

Tabela 4. Comandos Sniper executados nos benchmarks.

A Tabela 5 indica os programas selecionados com os tempos de execução nativo e pelo simulador Sniper e o cálculo do slowdown de simulação. A Tabela 6 apresenta outras métricas produzidas pelo Sniper.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Programas selecionados** | **Número de instruções executadas** | **Tempo de execução no simulador Sniper “Total Time”**  **(TSni)** | **Tempo de execução nativo “Total Time”**  **(TNat)** | **Slowdown de simulação**  **(TSni / TNat)** |
| radix | 46,6M | 2430 | 2585 | **0,940** |
| cholesky tk14.0 | 44,3M | 2906 | 31135 | **0,093** |
| cholesky d750.o | 309,8M | 42077 | 158097 | **0,266** |
| fft | 1,6M | 249 | 276 | **0,902** |

Tabela 5. Cálculo de slowdown para programas selecionados.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Programas** | **Número de Instruções Executadas**  **(Milhão)** | **Instruções por Ciclo**  **(IPC)** | **Ciclos**  **(Milhão)** | **Tempo total com inicialização** | **Tempo decorrido**  **(s)** |  | **Leaving ROI Time**  **(s)** | **Velocidade da Simulação**  **(KIPS)** |
| radix | 46,6M | 0,45 | 104,4M | 39118 | 267,41 |  | 267,37 | 174,5 |
| cholesky tk14.O | 44,3M | 2,30 | 19,3M | 7085 | 704,18 |  | 704,33 | 62,9 |
| cholesky d750.O | 309,8M | 1,84 | 167,9M | 62965 | 4628,97 |  | 4632,49 | 66,9 |
| fft | 1,6M | 1,45 | 1,1M | 370 | 36,19 |  | 2,92 | 404,9 |

Tabela 6. Outras métricas de desempenho coletadas pelo Sniper.

## Perf profiler \*

Perf profiler é uma ferramenta Linux que coleta e analisa dados de desempenho de programas ou do sistema operacional.

A Tabela 7 apresenta os comandos com os parâmetros utilizados na execução de cada um dos programas.

|  |  |
| --- | --- |
| **Comando para execução do programa Perf** | **Arquivo com o resultado da execução** |
| perf stat ./api > results/perf-result-api.txt | perf-result-api.txt |
| perf stat ./dvfs > results/perf-result-dvfs.txt | perf-result-dvfs.txt |
| perf stat ./fft > results/perf-result-fft.txt | perf-result-fft.txt |
| perf stat ./fork > results/perf-result-fork.txt | perf-result-fork.txt |
| perf stat ./signal > results/perf-result-signal.txt | perf-result-signal.txt |
| perf stat ./smc > results/perf-result-smc.txt | perf-result-smc.txt |
| perf stat ./spinloop > results/perf-result-spinloop.txt | perf-result-spinloop.txt |
| perf stat ./true > results/perf-result-true.txt | perf-result-true.txt |
| perf stat ./RADIX > results/perf-result-RADIX.txt | perf-result-RADIX.txt |
| perf stat ./CHOLESKY tk14.O > results/perf-result-CHOLESKY-tk14.O.txt | perf-result-CHOLESKY-tk14.O.txt |
| perf stat ./CHOLESKY d750.O > results/perf-result-CHOLESKY-d750.O.txt | perf-result-CHOLESKY-d750.O.txt |

Tabela 7. Comandos Perf profiler executados nos programas.

A Tabela 8 apresenta as mesmas métricas coletadas de forma nativa pelo Perf aplicados aos programas RADIX, CHOLESKY e FFT, sendo as mesmas métricas coletadas pelo Sniper.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Programas** | **Número de Instruções Executadas** | **Instruções por Ciclo**  **(IPC)** | **Ciclos** | **Tempo total com inicialização** | **Tempo decorrido**  **(s)** |
| radix | 49058550 ~ 49M | 0,42 | 118135743 ~ 118M | 231299 | 0,244433302 |
| cholesky tk14.O | 47273748 ~ 47M | 1,09 | 43349489 ~ 43,5M | 56411 | 0,085269899 |
| cholesky d750.O | 318793875 ~ 318M | 1,27 | 250775918 ~ 250M | 179758 | 0,209806402 |
| fft | 2793932 ~ 2,8M | 0,66 | 4238960 ~ 4,2M | 6133 | 0,021505200 |

Tabela 8. Métricas coletadas pelo Perf.

A comparação entre as métricas produzidas pelo Sniper e Perf, observou-se o seguinte:

* para a métrica “número de instruções executadas”, Perf apresentou valores maiores em todos os programas;
* para as métricas “instruções por ciclo (IPC)” e “ciclo”, Perf apresentou valores menores do que Sniper em todos os programas;
* para a métrica “tempo total com inicialização”, Perf apontou apresentou maiores do que Sniper em todos os programas;
* para a métrica “tempo decorrido”, Perf apresentou valores menores do que Sniper em todos os programas.

Dessa forma, as diferenças nos valores das métricas entre Perf e Sniper ocorrem devido ao Perf coletar dados de uma execução nativa do programa e Sniper coletar dados de uma execução simulada do programa.

## PARSEC Benchmark Suite 3.0 \*

O PARSEC (*Princeton Application Repository for Shared-Memory Computers*) é um conjunto de benchmark composto por programas *multithread* com o propósito de possibilitar estudos de desempenho em computadores com múltiplos processadores.

A Tabela 9 apresenta a compilação dos pacotes de benchmark oferecidos no PARSEC com o resultado indicando sucesso ou os erros apresentados no processo de compilação (build).

|  |  |  |
| --- | --- | --- |
| **Pacote** | **Comando para compilação** | **Resultado** |
| blackscholes | parsecmgmt -a build -p blackscholes | Compilou com sucesso. |
| bodytrack | parsecmgmt -a build -p bodytrack | Compilou com sucesso. |
| facesim | parsecmgmt -a build -p facesim | make[2]: \*\*\* [/usr/local/parsec-3.0/pkgs/apps/facesim/obj/amd64-linux.gcc/Public\_Library/Makefile.common:407: obj/Collisions\_And\_Interactions/COLLISION\_BODY\_LIST\_3D.o] Error 1  make[2]: Leaving directory '/usr/local/parsec-3.0/pkgs/apps/facesim/obj/amd64-linux.gcc/Public\_Library'  make[1]: \*\*\* No rule to make target '/usr/local/parsec-3.0/pkgs/apps/facesim/obj/amd64-linux.gcc/lib/libPhysBAM.a', needed by 'facesim'. Stop.  make[1]: Leaving directory '/usr/local/parsec-3.0/pkgs/apps/facesim/obj/amd64-linux.gcc/Benchmarks/facesim'  make: \*\*\* [Makefile:16: all] Error 2  [PARSEC] Error: 'env version=pthreads PHYSBAM=/usr/local/parsec-3.0/pkgs/apps/facesim/obj/amd64-linux.gcc CXXFLAGS=-O3 -g -funroll-loops -fprefetch-loop-arrays -fpermissive -fno-exceptions -std=c++11 -static-libgcc -Wl,--hash-style=both,--as-needed -DPARSEC\_VERSION=3.0-beta-20150206 -fexceptions /usr/bin/make' failed. |
| ferret | parsecmgmt -a build -p ferret | make: \*\*\* [Makefile:108: /usr/local/parsec-3.0/pkgs/apps/ferret/obj/amd64-linux.gcc/parsec/obj/LSH\_query.o] Error 1  [PARSEC] Error: 'env version=pthreads CFLAGS=-I/usr/local/parsec-3.0/pkgs/libs/gsl/inst/amd64-linux.gcc/include -I/usr/local/parsec-3.0/pkgs/libs/libjpeg/inst/amd64-linux.gcc/include -O3 -g -funroll-loops -fprefetch-loop-arrays -static-libgcc -Wl,--hash-style=both,--as-needed -DPARSEC\_VERSION=3.0-beta-20150206 LDFLAGS=-L/usr/local/parsec-3.0/pkgs/libs/gsl/inst/amd64-linux.gcc/lib -L/usr/local/parsec-3.0/pkgs/libs/libjpeg/inst/amd64-linux.gcc/lib -L/usr/lib64 -L/usr/lib /usr/bin/make' failed. |
| fluidanimate | parsecmgmt -a build -p fluidanimate | Compilou com sucesso. |
| freqmine | parsecmgmt -a build -p freqmine | Compilou com sucesso. |
| raytrace | parsecmgmt -a build -p raytrace | No package 'xext' found  Consider adjusting the PKG\_CONFIG\_PATH environment variable if you  installed software in a non-standard prefix.  Alternatively, you may set the environment variables XLIBGL\_CFLAGS  and XLIBGL\_LIBS to avoid the need to call pkg-config.  See the pkg-config man page for more details.  [PARSEC] Error: 'env ./configure --with-driver=xlib --enable-glut --enable-static --disable-shared --prefix=/usr/local/parsec-3.0/pkgs/libs/mesa/inst/amd64-linux.gcc' failed. |
| swaptions | parsecmgmt -a build -p swaptions | | ^~~~~~  make[1]: \*\*\* [../../build/Makefile.tbbmalloc:70: proxy.o] Error 1  make[1]: Leaving directory '/usr/local/parsec-3.0/pkgs/libs/tbblib/obj/amd64-linux.gcc/build/linux\_intel64\_gcc\_cc11.3.0\_libc2.35\_kernel5.15.90.1\_debug'  make: \*\*\* [Makefile:49: tbbmalloc] Error 2  [PARSEC] Error: 'env compiler=gcc PATH=/usr/bin:/usr/local/parsec-3.0/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin:/usr/local/parsec-3.0/bin CXXFLAGS=-O3 -g -funroll-loops -fprefetch-loop-arrays -fpermissive -fno-exceptions -static-libgcc -Wl,--hash-style=both,--as-needed -DPARSEC\_VERSION=3.0-beta-20150206 -fexceptions /usr/bin/make' failed. |
| vips | parsecmgmt -a build -p vips | Compilou com sucesso. |

Tabela 9. Resultado da compilação dos pacotes do PARSEC.

O PARSEC possibilita definir a região de interesse (ROI – *Region Of Interest*) baseada em seis tipos de entrada possíveis na execução dos benchmarks. As entradas são: *test, simdev, simsmall, simmedium, simlarge* e *native*. Além disso, foram realizadas execuções considerando diferentes valores para o parâmetro ”–n” que indica número mínimo de threads na execução. O valor default de “n” é 1.

Os testes realizados no experimento utilizaram todas as entradas nos benchmarks executados, cujos comandos de execução e indicação dos arquivos de resultados são apresentados na Tabela 10.

Os resultados detalhados desse experimento podem ser consultados na seção [Parsec do repositório Github](https://github.com/rubenscp/RCP-MO601-Project-03/tree/main/parsec/result).

|  |  |  |  |
| --- | --- | --- | --- |
| **Núm. da Execução** | **Pacote** | **Entrada** | **Comando de execução do pacote de Benchmark** |
| 001 | blackscholes | test | parsecmgmt -a run -p blackscholes -i test > results/exec-001-blackscholes-test.txt |
| 002 | blackscholes | simdev | parsecmgmt -a run -p blackscholes -i simdev > results/exec-002-blackscholes-simdev.txt |
| 003 | blackscholes | simsmall | parsecmgmt -a run -p blackscholes -i simsmall > results/exec-003-blackscholes-simsmall.txt |
| 004 | blackscholes | simlarge | parsecmgmt -a run -p blackscholes -i simlarge > results/exec-004-blackscholes-simlarge.txt |
| 005 | blackscholes | native | parsecmgmt -a run -p blackscholes -i native > results/exec-005-blackscholes-native.txt |
| 006 | vips | test | parsecmgmt -a run -p vips -i test > results/exec-006-vips-test.txt |
| 007 | vips | simdev | parsecmgmt -a run -p vips -i simdev > results/exec-007-vips-simdev.txt |
| 008 | vips | simsmall | parsecmgmt -a run -p vips -i simsmall > results/exec-008-vips-simsmall.txt |
| 009 | vips | simlarge | parsecmgmt -a run -p vips -i simlarge > results/exec-009-vips-simlarge.txt |
| 010 | vips | native | parsecmgmt -a run -p vips -i native > results/exec-010-vips-native.txt |
| 011 | bodytrack | test | parsecmgmt -a run -p bodytrack -i test > results/exec-011-bodytrack-test.txt |
| 012 | bodytrack | simdev | parsecmgmt -a run -p bodytrack -i simdev > results/exec-012-bodytrack-simdev.txt |
| 013 | bodytrack | simsmall | parsecmgmt -a run -p bodytrack -i simsmall > results/exec-013-bodytrack-simsmall.txt |
| 014 | bodytrack | simlarge | parsecmgmt -a run -p bodytrack -i simlarge > results/exec-014-bodytrack-simlarge.txt |
| 015 | bodytrack | native | parsecmgmt -a run -p bodytrack -i native > results/exec-015-bodytrack-native.txt |
| 016 | fluidanimate | test | parsecmgmt -a run -p fluidanimate -i test > results/exec-016-fluidanimate-test.txt |
| 017 | fluidanimate | simdev | parsecmgmt -a run -p fluidanimate -i simdev > results/exec-017-fluidanimate-simdev.txt |
| 018 | fluidanimate | simsmall | parsecmgmt -a run -p fluidanimate -i simsmall > results/exec-018-fluidanimate-simsmall.txt |
| 019 | fluidanimate | simlarge | parsecmgmt -a run -p fluidanimate -i simlarge > results/exec-019-fluidanimate-simlarge.txt |
| 020 | fluidanimate | native | parsecmgmt -a run -p fluidanimate -i native > results/exec-020-fluidanimate-native.txt |
| 021 | freqmine | test | parsecmgmt -a run -p freqmine -i test > results/exec-021-freqmine-test.txt |
| 022 | freqmine | simdev | parsecmgmt -a run -p freqmine -i simdev > results/exec-022-freqmine-simdev.txt |
| 023 | freqmine | simsmall | parsecmgmt -a run -p freqmine -i simsmall > results/exec-023-freqmine-simsmall.txt |
| 024 | freqmine | simlarge | parsecmgmt -a run -p freqmine -i simlarge > results/exec-024-freqmine-simlarge.txt |
| 025 | freqmine | native | parsecmgmt -a run -p freqmine -i native > results/exec-025-freqmine-native.txt |
| 026 | splash2 | test | parsecmgmt -a run -p splash2 -i test > results/exec-026-splash2-test.txt |
| 027 | splash2 | simdev | parsecmgmt -a run -p splash2 -i simdev > results/exec-027-splash2-simdev.txt |
| 028 | splash2 | simsmall | parsecmgmt -a run -p splash2 -i simsmall > results/exec-028-splash2-simsmall.txt |
| 029 | splash2 | simlarge | parsecmgmt -a run -p splash2 -i simlarge > results/exec-029-splash2-simlarge.txt |
| 030 | splash2 | native | parsecmgmt -a run -p splash2 -i native > results/exec-030-splash2-native.txt |
| 031 | splash2x | test | parsecmgmt -a run -p splash2x -i test > results/exec-031-splash2x-test.txt |
| 032 | splash2x | simdev | parsecmgmt -a run -p splash2x -i simdev > results/exec-032-splash2x-simdev.txt |
| 033 | splash2x | simsmall | parsecmgmt -a run -p splash2x -i simsmall > results/exec-033-splash2x-simsmall.txt |
| 034 | splash2x | simlarge | parsecmgmt -a run -p splash2x -i simlarge > results/exec-034-splash2x-simlarge.txt |
| 035 | splash2x | native | parsecmgmt -a run -p splash2x -i native > results/exec-035-splash2x-native.txt |
| 051 | blackscholes | native | parsecmgmt -a run -p blackscholes -i native -n 2 > results/exec-051-blackscholes-native-n2.txt |
| 052 | blackscholes | native | parsecmgmt -a run -p blackscholes -i native -n 4 > results/exec-052-blackscholes-native-n4.txt |
| 053 | blackscholes | native | parsecmgmt -a run -p blackscholes -i native -n 8 > results/exec-053-blackscholes-native-n8.txt |
| 054 | blackscholes | native | parsecmgmt -a run -p blackscholes -i native -n 16 > results/exec-054-blackscholes-native-n16.txt |
| 055 | vips | native | parsecmgmt -a run -p vips -i native -n 2 > results/exec-055-vips-native-n2.txt |
| 056 | vips | native | parsecmgmt -a run -p vips -i native -n 4 > results/exec-056-vips-native-n4.txt |
| 057 | vips | native | parsecmgmt -a run -p vips -i native -n 8 > results/exec-057-vips-native-n8.txt |
| 058 | vips | native | parsecmgmt -a run -p vips -i native -n 16 > results/exec-058-vips-native-n16.txt |
| 059 | bodytrack | native | parsecmgmt -a run -p bodytrack -i native -n 2 > results/exec-059-bodytrack-native-n2.txt |
| 060 | bodytrack | native | parsecmgmt -a run -p bodytrack -i native -n 4 > results/exec-060-bodytrack-native-n4.txt |
| 061 | bodytrack | native | parsecmgmt -a run -p bodytrack -i native -n 8 > results/exec-061-bodytrack-native-n8.txt |
| 062 | bodytrack | native | parsecmgmt -a run -p bodytrack -i native -n 16 > results/exec-062-bodytrack-native-n16.txt |
| 063 | fluidanimate | native | parsecmgmt -a run -p fluidanimate -i native -n 2 > results/exec-063-fluidanimate-native-n2.txt |
| 064 | fluidanimate | native | parsecmgmt -a run -p fluidanimate -i native -n 4 > results/exec-064-fluidanimate-native-n4.txt |
| 065 | fluidanimate | native | parsecmgmt -a run -p fluidanimate -i native -n 8 > results/exec-065-fluidanimate-native-n8.txt |
| 066 | fluidanimate | native | parsecmgmt -a run -p fluidanimate -i native -n 16 > results/exec-066-fluidanimate-native-n16.txt |
| 067 | freqmine | native | parsecmgmt -a run -p freqmine -i native -n 2 > results/exec-067-freqmine-native-n2.txt |
| 068 | freqmine | native | parsecmgmt -a run -p freqmine -i native -n 4 > results/exec-068-freqmine-native-n4.txt |
| 069 | freqmine | native | parsecmgmt -a run -p freqmine -i native -n 8 > results/exec-069-freqmine-native-n8.txt |
| 070 | freqmine | native | parsecmgmt -a run -p freqmine -i native -n 16 > results/exec-070-freqmine-native-n16.txt |
| 071 | splash2 | native | parsecmgmt -a run -p splash2 -i native -n 2 > results/exec-071-splash2-native-n2.txt |
| 072 | splash2 | native | parsecmgmt -a run -p splash2 -i native -n 4 > results/exec-072-splash2-native-n4.txt |
| 073 | splash2 | native | parsecmgmt -a run -p splash2 -i native -n 8 > results/exec-073-splash2-native-n8.txt |
| 074 | splash2 | native | parsecmgmt -a run -p splash2 -i native -n 16 > results/exec-074-splash2-native-n16.txt |
| 075 | splash2x | native | parsecmgmt -a run -p splash2x -i native -n 2 > results/exec-075-splash2x-native-n2.txt |
| 076 | splash2x | native | parsecmgmt -a run -p splash2x -i native -n 4 > results/exec-076-splash2x-native-n4.txt |
| 077 | splash2x | native | parsecmgmt -a run -p splash2x -i native -n 8 > results/exec-077-splash2x-native-n8.txt |
| 078 | splash2x | native | parsecmgmt -a run -p splash2x -i native -n 16 > results/exec-078-splash2x-native-n16.txt |

Tabela 10. Comandos PARSEC para execução dos benchmarks com as entradas possíveis.

A Tabela 11 apresenta os tempos de execução dos programas executados pelo Parsec com entrada “native” combinado com variado número de threads.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Programa** | **Execução do programa com tipo de entrada “native” combinado com os diversos números de Threads (N)** | | | | |
| **N = 1** | **N = 2** | **N = 4** | **N = 8** | **N = 16** |
| blackscholes | real 1m39.122s  user 1m38.267s  sys 0m0.830s | real 6m0.260s  user 3m54.314s  sys 0m6.030s | real 3m52.404s  user 3m21.764s  sys 0m4.591s | real 3m12.120s  user 3m19.701s  sys 0m4.644s | real 2m57.154s  user 3m20.085s  sys 0m4.987s |
| vips | real 2m10.307s  user 2m8.549s  sys 0m3.862s | real 9m50.830s  user 5m58.435s  sys 0m36.888s | real 5m52.170s  user 5m41.750s  sys 0m19.538s | real 4m55.799s  user 5m56.183s  sys 0m24.917s | real 4m31.167s  user 6m23.196s  sys 0m46.789s |
| Incluir os demais programas | ... | ... | … | … | … |

Tabela 11. Execução do Parsec nos programas de benchmark com o parâmetro de entrada “-i native” combinado com os diversos valores de número de Threads “-n “ apresentando os tempos de execução.

Fazer um gráfico quatro aplicações, variando o valor de N no eixo X e o tempo real no Y. Usar N=1 a referencia (baseline)

## Rodinia benchmark \*

O Rodinia Benchmark é uma ferramenta destinada a infraestrutura de computação heterogênea com implementações com OpenMP, OpenCL e CUDA.

A Tabela 12 apresenta a lista dos programas que foram compilados com sucesso em cada implementação.

|  |  |  |
| --- | --- | --- |
| **CUDA (make CUDA)** | **OPENMP (make OMP)** | **OPENCL (make OPENCL)** |
| backprop | backprop | OCL\_particlefilter\_double |
| bfs | bfs | OCL\_particlefilter\_naive |
| dwt2d | euler3d\_cpu | OCL\_particlefilter\_single |
| gaussian | euler3d\_cpu\_double | backprop |
| heartwall | heartwall | gaussian |
| hotspot | hotspot | heartwall |
| kmeans | kmeans | hotspot |
| leukocyte | lavaMD | kmeans |
| needle | leukocyte | lavaMD |
| nn | lud\_omp | leukocyte |
| pathfinder | needle | lud |
| sc\_gpu | nn | nn |
| srad\_v1 | particle\_filter | nw |
| srad\_v2 | pathfinder | srad |
|  | pre\_euler3d\_cpu |  |
|  | pre\_euler3d\_cpu\_double |  |
|  | sc\_omp |  |
|  | srad\_v1 |  |
|  | srad\_v2 |  |

Tabela 12. Lista de programas que foram compilados com sucesso no ambiente da ferramenta Rodinia Benchmark.

A Tabela 13 apresenta a execução de benchmarks com alguns resultados detalhados ou o nome do arquivo de resultado devido ao seu tamanho excessivo. Para a execução de cada benchmark, basta acessar a respectiva pasta e executar o comando “./run” e os resultados da execução estão armazenados no arquivo “result.txt”.

|  |  |  |
| --- | --- | --- |
| **Implementação** | **Benchmark** | **Resultado** |
| OPENMP | bfs | result.txt |
| OPENMP | cfd (euler3d) | 409.637 segundos  root@NotebookRubens:/usr/local/rodinia\_3.1/openmp/cfd# ./run  Starting...  Compute time: 409.637  Saving solution...  Saved solution...  Cleaning up...  Done... |
| OPENMP | heartwall | result.txt |
| OPENMP | hotspot | output.out |
| OPENMP | kmeans | root@NotebookRubens:/usr/local/rodinia\_3.1/openmp/kmeans# ./run  ./run: line 1: ./kmeans\_serial/kmeans: No such file or directory  I/O completed  num of threads = 4  number of Clusters 5  number of Attributes 34  Time for process: 4.266001 |
| OPENMP | lavaMD | root@NotebookRubens:/usr/local/rodinia\_3.1/openmp/lavaMD# ./run  Configuration used: cores = 4, boxes1d = 10  Time spent in different stages of CPU/MCPU KERNEL:  0.000000000000 s, 0.000000000000 % : CPU/MCPU: VARIABLES  0.000014000000 s, 0.000279933040 % : MCPU: SET DEVICE  0.000000000000 s, 0.000000000000 % : CPU/MCPU: INPUTS  5.001182079315 s, 99.999717712402 % : CPU/MCPU: KERNEL  Total time:  5.001195907593 s |
| OPENMP | leukocyte | root@NotebookRubens:/usr/local/rodinia\_3.1/openmp/leukocyte# ./run  Num of threads: 4  Detecting cells in frame 0  Cells detected: 36  Detection runtime  -----------------  GICOV computation: 0.52551 seconds  GICOV dilation: 0.21413 seconds  Total: 0.79247 seconds  Tracking cells across 5 frames  Processing frame 5 / 5  Tracking runtime (average per frame):  ------------------------------------  MGVF computation: 14.68158 seconds  Snake evolution: 0.02456 seconds  Total: 4.09308 seconds  Total application run time: 21.25787 seconds |
| OPENMP | nn | root@NotebookRubens:/usr/local/rodinia\_3.1/openmp/nn# ./run  The 5 nearest neighbors are:  1974 12 22 18 24 JOYCE 30.6 89.9 80 593 --> 0.608276  2003 8 27 12 10 TONY 29.9 89.4 160 286 --> 0.608275  1997 11 14 12 24 HELENE 30.5 89.8 134 529 --> 0.538515  1980 10 22 18 3 ISAAC 30.1 90.4 110 778 --> 0.412312  1988 12 27 0 18 TONY 30.0 89.8 113 39 --> 0.199997  total time : 0.527607023716 s |
| OPENMP | particle\_filter | Result.txt |
| OPENMP | pathfinder | o.out |
| OPENMP | srad\_v1 | root@NotebookRubens:/usr/local/rodinia\_3.1/openmp/srad/srad\_v1# ./run  Time spent in different stages of the application:  0.000000000000 s, 0.000000000000 % : SETUP VARIABLES  0.000021000000 s, 0.001276622177 % : READ COMMAND LINE PARAMETERS  0.131821006536 s, 8.013600349426 % : READ IMAGE FROM FILE  0.002430000110 s, 0.147723421454 % : RESIZE IMAGE  0.000082999999 s, 0.005045697093 % : SETUP, MEMORY ALLOCATION  0.016366999596 s, 0.994974911213 % : EXTRACT IMAGE  1.328287959099 s, 80.748657226562 % : COMPUTE  0.005131000187 s, 0.311921358109 % : COMPRESS IMAGE  0.160110995173 s, 9.733392715454 % : SAVE IMAGE INTO FILE  0.000714000023 s, 0.043405152857 % : FREE MEMORY  Total time: 1.644966006279 s |
| OPENMP | srad\_v2 | root@NotebookRubens:/usr/local/rodinia\_3.1/openmp/srad/srad\_v2# ./run  Randomizing the input matrix  Start the SRAD main loop  Computation Done |

Tabela 13. Benchmarks executados com os resultados.

A Tabela 14 apresenta comparações de alguns benchmarks que foram executados nas três implementações CUDA, OpenMP e OpenCL.

|  |  |  |
| --- | --- | --- |
| **Comparação de Benchmarks entre as Implementações CUDA, OpenMP e OpenCL** | | |
| **Hotspot** | | |
| **CUDA** | **OpenMP** | **OpenCL** |
| root@NotebookRubens:/usr/local/rodinia\_3.1/cuda/hotspot# ./run  WG size of kernel = 16 X 16  pyramidHeight: 2  gridSize: [512, 512]  border:[2, 2]  blockGrid:[43, 43]  targetBlock:[12, 12]  Start computing the transient temperature  Ending simulation | root@NotebookRubens:/usr/local/rodinia\_3.1/openmp/hotspot# ./run  Start computing the transient temperature  Ending simulation  Total time: 0.045 seconds | Erro de execução |
| **BFS** | | |
| **CUDA** | **OpenMP** | **OpenCL** |
| root@NotebookRubens:/usr/local/rodinia\_3.1/cuda/bfs# ./run  Reading File  Read File  Copied Everything to GPU memory  Start traversing the tree  Kernel Executed 1 times  Result stored in result.txt >> 1.000.000 lines | root@NotebookRubens:/usr/local/rodinia\_3.1/openmp/bfs# ./run  Reading File  Start traversing the tree  Compute time: 0.633491  Result stored in result.txt >> 1.000.000 lines | Erro de compilação |
| **HeartWall** | | |
| **CUDA** | **OpenMP** | **OpenCL** |
| root@NotebookRubens:/usr/local/rodinia\_3.1/cuda/heartwall# ./run  WG size of kernel = 256  frame progress: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19  Resultados no arquivo result.txt | root@NotebookRubens:/usr/local/rodinia\_3.1/openmp/heartwall# ./run  num of threads: 4  frame progress: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19  Resultados no arquivo result.txt | Erro de execução |
| **particlefilter** | | |
| **CUDA** | **OpenMP** | **OpenCL** |
| Erro de compilação | video sequence took 0.043539  time to get neighbors took: 0.000005  time to get weightstook: 0.014813  time to set arrays took: 0.000106  time to set error took: 0.000682  time to get likelihoods took: 0.002394  time to get exp took: 0.000109  time to sum weights took: 0.000008  time to normalize weights took: 0.000004  time to move object took: 0.000008  xe: 64.523185  ye: 64.469547  0.702991  ...  time to calc cum sum took: 0.000033  time to calc u took: 0.011697  time to calc new array x and y took: 0.061382  time to reset weights took: 0.000047  time to set error took: 0.006374  time to get likelihoods took: 0.008245  time to get exp took: 0.011114  time to sum weights took: 0.011051  time to normalize weights took: 0.008430  time to move object took: 0.016451  xe: 48.546698  ye: 72.385056  17.581630  time to calc cum sum took: 0.000034  time to calc u took: 0.013806  time to calc new array x and y took: 0.053608  time to reset weights took: 0.000045  particle filter took 0.937339  entire program took 0.980878 | root@notebookrubens:/usr/local/rodinia\_3.1/opencl/particlefilter# ./run  video sequence took 0.063222  error: clgetplatformids(1,\*,0) failed  particle filter took 0.694592  entire program took 0.757814  video sequence took 0.031961  error: clgetplatformids(1,\*,0) failed  particle filter took 0.632983  entire program took 0.664944 |

Tabela 14. Comparação de benchmarks nas três implementações CUDA, OpenMP e OpenCL.

* Se tiver hardware suficiente, rodar as múltiplas versões do programa e comparar o desempenho no mesmo computador.
* Para múltiplas configurações do mesmo programa, indicar as diferenças de desempenho.
* Executar o Rodínia nos três programas abaixo e comparar o desempenho:
  + RADIX
  + CHOLESKY
  + FFT

## Intel Pin

A Tabela 15 apresenta os comandos de execução dos programas e o resultado da execução. Antes deve-se acessar a pasta “/usr/local/pin-3.27-98718-gbeaa5d51e-gcc-linux/source/tools” no ambiente de experimentação.

|  |  |  |
| --- | --- | --- |
| **PinTool programa** | **Programa selecionado** | **Comando de execução do pacote de Benchmark** |
| opcodemix | RADIX | ../../pin -t SimpleExamples/obj-intel64/opcodemix.so -- ./programs\_selected/radix/RADIX |
| opcodemix | CHOLESKY tk14.O | ../../pin -t SimpleExamples/obj-intel64/opcodemix.so -- ./programs\_selected/cholesky/CHOLESKY tk14.O |
| opcodemix | CHOLESKY tk14.O | ../../pin -t SimpleExamples/obj-intel64/opcodemix.so -- ./programs\_selected/cholesky/CHOLESKY d750.O |
| opcodemix | FFT | ../../pin -t SimpleExamples/obj-intel64/opcodemix.so -- ./programs\_selected/fft/fft |

Tabela 15. Comandos para execução do PinTools nos programas selecionados com os respectivos resultados.

|  |  |  |
| --- | --- | --- |
| **PinTool programa** | **Programa selecionado** | **Resultado da Execução** |
| opcodemix | RADIX | Integer Radix Sort  262144 Keys  1 Processors  Radix = 1024  Max key = 524288  [HOOKS] Entering ROI  [HOOKS] Leaving ROI  PROCESS STATISTICS  Total Rank Sort  Proc Time Time Time  0 112299 66758 20663  TIMING INFORMATION  Start time : 667317197  Initialization finish time : 668397616  Overall finish time : 668509915  Total time with initialization : 1192718  Total time without initialization : 112299 |
| opcodemix | CHOLESKY tk14.O |  |
| opcodemix | CHOLESKY tk14.O |  |
| opcodemix | FFT | FFT with Blocking Transpose  1024 Complex Doubles  1 Processors  65536 Cache lines  16 Byte line size  4096 Bytes per page  PROCESS STATISTICS  Computation Transpose Transpose  Proc Time Time Fraction  0 133392 41462 0.31083  TIMING INFORMATION  Start time : 938681432  Initialization finish time : 939364208  Overall finish time : 939497600  Total time with initialization : 816168  Total time without initialization : 133392  Overall transpose time : 41462  Overall transpose fraction : 0.31083 |

## Dinero cache simulator

A ferramenta Dinero é um simulador de cache de 4ª geração de simuladores.

Os programas utilizados nessa ferramenta foram o RADIX e o fft. Vários parâmetros foram avaliados considerando valores distintos para cache L1 (instrução e data), combinados com cache L2 e L3 (unificadas).

A Tabela 16 apresenta os comandos utilizados na execução dos programas RADIX e fft com os diversos parâmetros de execução relacionados às caches L1, L2 e L3.

|  |  |  |
| --- | --- | --- |
| **Programa RADIX** | | |
| **Execução** | **Comando de execução** | **Arquivo com o resultado da execução** |
| 001 | ./dineroIV-tar -l1-isize 1k -l1-dsize 1k -l1-ibsize 32 -l1-dbsize 32 -informat p < RADIX > dinero-result-RADIX-001.txt | dinero-result-RADIX-001.txt |
| 002 | ./dineroIV-tar -l1-isize 2k -l1-dsize 2k -l1-ibsize 16 -l1-dbsize 16 -informat p < RADIX > dinero-result-RADIX-002.txt | dinero-result-RADIX-002.txt |
| 003 | ./dineroIV-tar -l1-isize 4k -l1-dsize 4k -l1-ibsize 8 -l1-dbsize 8 -informat p < RADIX > dinero-result-RADIX-003.txt | dinero-result-RADIX-003.txt |
| 004 | ./dineroIV-tar -l1-isize 8k -l1-dsize 8k -l1-ibsize 4 -l1-dbsize 4 -informat p < RADIX > dinero-result-RADIX-004.txt | dinero-result-RADIX-004.txt |
| 005 | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -informat p < RADIX > dinero-result-RADIX-005.txt | dinero-result-RADIX-005.txt |
| 006 | ./dineroIV-tar -l1-isize 32k -l1-dsize 32k -l1-ibsize 1 -l1-dbsize 1 -informat p < RADIX > dinero-result-RADIX-006.txt | dinero-result-RADIX-006.txt |
| 007 | ./dineroIV-tar -l1-isize 1k -l1-dsize 1k -l1-ibsize 32 -l1-dbsize 32 -l1-iassoc 8 -l1-dassoc 8 -informat p < RADIX > dinero-result-RADIX-007.txt | dinero-result-RADIX-007.txt |
| 008 | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -informat p < RADIX > dinero-result-RADIX-008.txt | dinero-result-RADIX-008.txt |
| 009 | ./dineroIV-tar -l1-isize 32k -l1-dsize 32k -l1-ibsize 1 -l1-dbsize 1 -l1-iassoc 8 -l1-dassoc 8 -informat p < RADIX > dinero-result-RADIX-009.txt | dinero-result-RADIX-009.txt |
| 010 | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 512k -l2-ubsize 1 -informat p < RADIX > dinero-result-RADIX-010.txt | dinero-result-RADIX-010.txt |
| 011 | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 256k -l2-ubsize 2 -informat p < RADIX > dinero-result-RADIX-011.txt | dinero-result-RADIX-011.txt |
| 012 | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 128k -l2-ubsize 4 -informat p < RADIX > dinero-result-RADIX-012.txt | dinero-result-RADIX-012.txt |
| 013 | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 128k -l2-ubsize 4 -l2-uassoc 8 -l3-usize 1m -l3-ubsize 4 -l3-uassoc 8 -informat p < RADIX > dinero-result-RADIX-013.txt | dinero-result-RADIX-013.txt |
| 014 | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 128k -l2-ubsize 4 -l2-uassoc 8 -l3-usize 2m -l3-ubsize 4 -l3-uassoc 8 -informat p < RADIX > dinero-result-RADIX-014.txt | dinero-result-RADIX-014.txt |
| 015 | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 128k -l2-ubsize 4 -l2-uassoc 8 -l3-usize 4m -l3-ubsize 1 -l3-uassoc 8 -informat p < RADIX > dinero-result-RADIX-015.txt | dinero-result-RADIX-015.txt |

|  |  |  |
| --- | --- | --- |
|  | **Programa FFT** | |
|  | **Comando de execução** | **Arquivo com o resultado da execução** |
|  | ./dineroIV-tar -l1-isize 1k -l1-dsize 1k -l1-ibsize 32 -l1-dbsize 32 -informat p < fft > dinero-result-fft-001.txt | dinero-result-fft-001.txt |
|  | ./dineroIV-tar -l1-isize 2k -l1-dsize 2k -l1-ibsize 16 -l1-dbsize 16 -informat p < fft > dinero-result-fft-002.txt | dinero-result-fft-002.txt |
|  | ./dineroIV-tar -l1-isize 4k -l1-dsize 4k -l1-ibsize 8 -l1-dbsize 8 -informat p < fft > dinero-result-fft-003.txt | dinero-result-fft-003.txt |
|  | ./dineroIV-tar -l1-isize 8k -l1-dsize 8k -l1-ibsize 4 -l1-dbsize 4 -informat p < fft > dinero-result-fft-004.txt | dinero-result-fft-004.txt |
|  | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -informat p < fft > dinero-result-fft-005.txt | dinero-result-fft-005.txt |
|  | ./dineroIV-tar -l1-isize 32k -l1-dsize 32k -l1-ibsize 1 -l1-dbsize 1 -informat p < fft > dinero-result-fft-006.txt | dinero-result-fft-006.txt |
|  | ./dineroIV-tar -l1-isize 1k -l1-dsize 1k -l1-ibsize 32 -l1-dbsize 32 -l1-iassoc 8 -l1-dassoc 8 -informat p < fft > dinero-result-fft-007.txt | dinero-result-fft-007.txt |
|  | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -informat p < fft > dinero-result-fft-008.txt | dinero-result-fft-008.txt |
|  | ./dineroIV-tar -l1-isize 32k -l1-dsize 32k -l1-ibsize 1 -l1-dbsize 1 -l1-iassoc 8 -l1-dassoc 8 -informat p < fft > dinero-result-fft-009.txt | dinero-result-fft-009.txt |
|  | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 512k -l2-ubsize 1 -informat p < fft > dinero-result-fft-010.txt | dinero-result-fft-010.txt |
|  | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 256k -l2-ubsize 2 -informat p < fft > dinero-result-fft-011.txt | dinero-result-fft-011.txt |
|  | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 128k -l2-ubsize 4 -informat p < fft > dinero-result-fft-012.txt | dinero-result-fft-012.txt |
|  | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 128k -l2-ubsize 4 -l2-uassoc 8 -l3-usize 1m -l3-ubsize 4 -l3-uassoc 8 -informat p < fft > dinero-result-fft-013.txt | dinero-result-fft-013.txt |
|  | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 128k -l2-ubsize 4 -l2-uassoc 8 -l3-usize 2m -l3-ubsize 4 -l3-uassoc 8 -informat p < fft > dinero-result-fft-014.txt | dinero-result-fft-014.txt |
|  | ./dineroIV-tar -l1-isize 16k -l1-dsize 16k -l1-ibsize 2 -l1-dbsize 2 -l1-iassoc 8 -l1-dassoc 8 -l2-usize 128k -l2-ubsize 4 -l2-uassoc 8 -l3-usize 4m -l3-ubsize 1 -l3-uassoc 8 -informat p < fft > dinero-result-fft-015.txt | dinero-result-fft-015.txt |

Tabela 16. Comandos Dinero para execução dos programas RADIX e FFT com variados parâmetros de execução relacionados às caches L1, L2 e L3.

Os resultados detalhados desse experimento podem ser consultados na seção [Dinero do repositório Github](https://github.com/rubenscp/RCP-MO601-Project-03/tree/main/dinero/results).

**Decisão sobre a melhor configuração de cache entre as testadas.**

* **A melhor configuração de cache seria aquela com menor índice de miss?**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Programa RADIX**  **Métrica “*Total demand miss rate*”** | | | | | |
| **Execução** | **L1-icache**  **(instruction)** | **L1-dcache**  **(data)** | **L2-ucache**  **(unified)** | **L3-ucache**  **(unified)** | **Existe algum critério geral ??** |
| 001 | 0,1175 | 0,6308 | --- | --- | ??? |
| 007 | 0,1177 | 0,5039 | --- | --- | ??? |
| 010 | 0,7340 | 0,3908 | 0,9483 | --- | ??? |
| 015 | 0,7340 | 0,3908 | 0,4723 | 0,8739 | ??? |

Tabela 17. Avaliação da métrica “Demand miss rate” produzida pelo Dinero na execução do programa RADIX.

# Considerações sobre o aprendizado nesse projeto

Esse projeto propôs um desafio diferente dos projetos anteriores devido ao uso de ferramentas de terceiros as quais deveriam ser instaladas, configuradas, compiladas e por fim, executadas no ambiente operacional escolhido. Algumas dessas ferramentas foram disponibilizadas para determinadas versões de compiladores e bibliotecas. Devido a essa diversidade de versões, surgiram vários desafios que precisam ser transpostos para se iniciar a tarefa de coleta dados e avaliar desempenho de benchmarks e programas. No meu caso, estimo que me dediquei a mais de 50% do tempo do projeto em tarefas de configuração e compilação das ferramentas, algo que poderia ser reduzido caso existisse documentação mais amigável e direcionada a determinado sistema operacional em uma certa versão. Apesar desse tempo, muito aprendizado foi obtido ao se conseguir ter todas as sete ferramentas aptas para o uso nas avaliações.

Outro aspecto de aprendizagem é o conhecimento que se consolida com experimentos reais de simulação demonstrando a importância na área de arquitetura de computadores para a inovação. Essas ferramentas de simulação auxiliam novos projetos de processadores e de seus componentes na etapa de validação.

Por fim, creio que ao final desse projeto consegui ter uma visão mesmo que superficial dessas ferramentas com possibilidade de aprofundamento em uma ou outra ferramenta caso necessário.