INF01113 – Organização de Computadores B– 2019/2
Turmas A e B
Profs. Gabriel Luca Nazar e Luigi Carro

Especificação do Segundo Trabalho Prático

1. Tarefas

Utilizando o simulador *gem5* (modo de simulação para processadores x86), simular a execução de 3 aplicações utilizando diversas configurações de organização.

O grupo deve escrever 3 programas (e.g.: algoritmos de ordenamento ou de busca, algoritmos de processamento como FFT ou CRC, ...). De preferência, eles devem ter características distintas (e.g.: complexidade diferente; proporcionalmente mais ou menos instruções de controle; mais ou menos acessos à memória; ou ainda ser o mesmo programa, mas compilado com *flags* de otimização diferentes no GCC, ou compiladores diferentes etc). Quanto mais complexo for o programa ou a comparação, maior valor terá na avaliação.

Mais especificamente, para o conjunto de configurações que podem ser alteradas no simulador, o grupo deve:

1. Escolher uma configuração fixa para todos os parâmetros. ***Esta configuração fixa deve ter, no mínimo, 3 parâmetros diferentes da configuração original, fornecida como base.***
2. A partir desta configuração fixa, escolher três parâmetros[1] a serem modificados. O grupo deve explicar o porquê da escolha de tais parâmetros a serem modificados e explicar o qual era o efeito esperado. A Figura 1 ilustra as etapas 1 e 2.

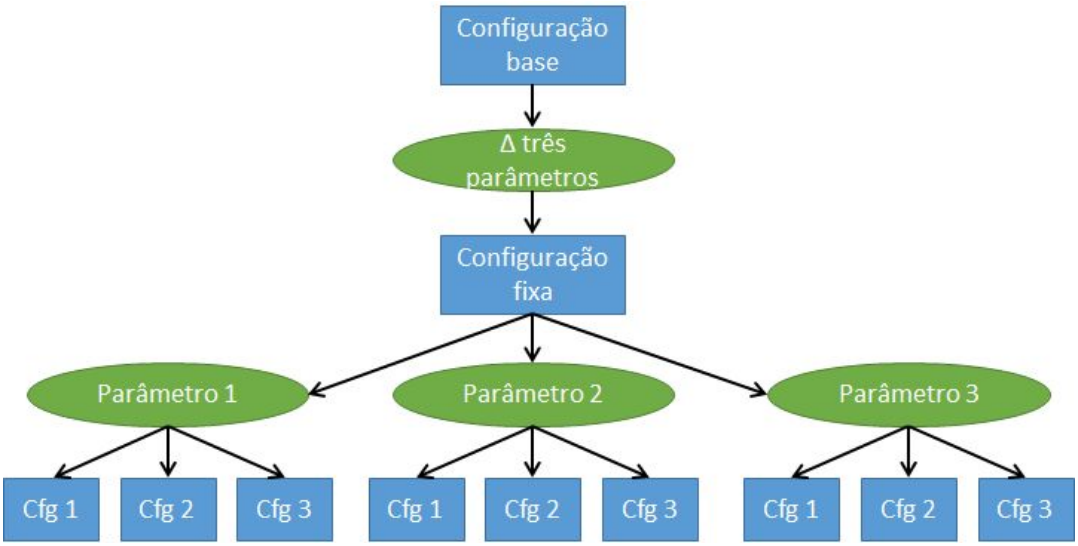

Figura 1 - Etapas 1 e 2.

3. Simular as três aplicações utilizando cada uma das configurações resultantes, inclusive a configuração fixa. Deverão ser coletados, no mínimo, os resultados de simulação: tempo de execução da aplicação e IPC da execução. Outros dados poderão ser coletados para explicar o comportamento dos programas.
4. Deverão ser feitos dois gráficos para cada parâmetro modificado: um demonstrando o IPC e outro demonstrando o tempo de execução. O eixo X de cada gráfico deverá mostrar os níveis do parâmetro e o eixo Y o IPC/tempo de execução. Haverão três linhas, uma para cada Benchmark. Veja um exemplo na Figura 2. No total, serão seis gráficos.
5. Para cada benchmark de cada gráfico, deverá haver uma explicação do porquê da variação de IPC de

[1] Verifique os comentários no arquivo de configuração do simulador para uma lista dos parâmetros que podem ser modificados.

cada um. Relacione os resultados com as características de cada benchmark (como as que foram dadas como sugestão no primeiro parágrafo). **Também deverá haver uma breve discussão sobre o desempenho final (tempo de execução).**

6. Para cada variação no parâmetro, fazer uma média do IPC das três aplicações. Discutir a diferença
7. destas médias, relacionando com as diferentes configurações e o custo-benefício de cada variação (ex: dobrando o tamanho do parâmetro, a performance é dobrada?).

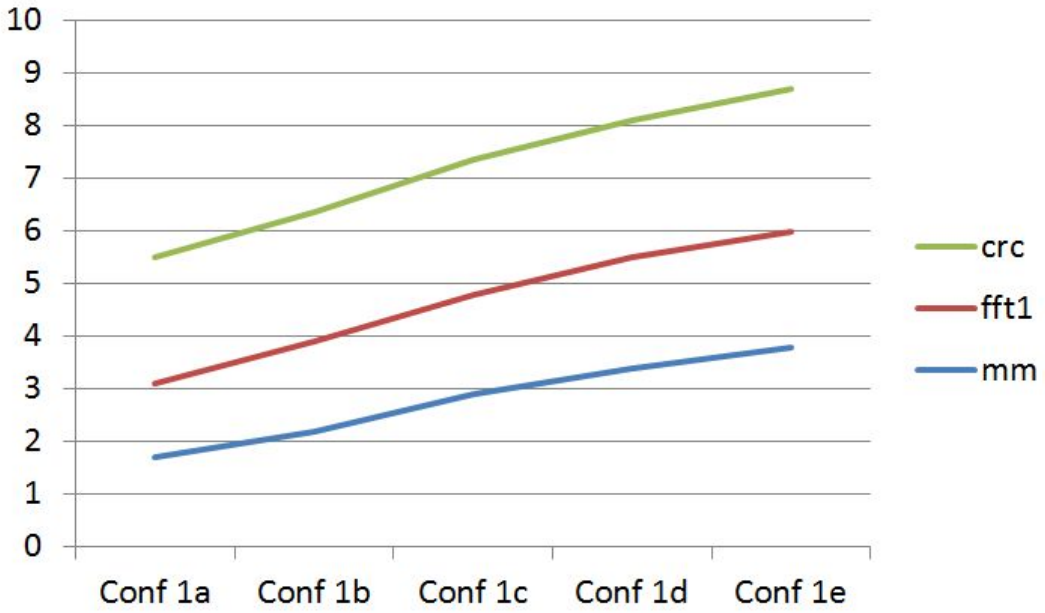

Figura 2 – Como um gráfico deve ser entregue. Eixo X: Cada modificação no parâmetro escolhido. Eixo Y: IPC

2. Instruções para execução da simulação

O simulador que deve ser utilizado é o gem5. O simulador é de código aberto e pode ser baixado aqui. Para este trabalho, porém, é oferecida uma máquina virtual com todo o ambiente já preparado**. O link para download da máquina virtual (.zip) e passo-a-passo para usar o gem5 dentro da máquina virtual estão no moodle da disciplina.**

Cuidado

- Note que simulação é um processo lento, podendo tomar um tempo que é até 100.000 vezes maior que o tempo de execução do programa em uma máquina real. Dimensione as entradas de seu programa de acordo.

Resultados da simulação

Ao concluir a simulação, se tudo der certo, será exibido na tela
1) A saída (stdout) do programa
2) Resultados da simulação (log do simulador).

Confira na Figura 3 um exemplo de execução correta. O log do simulador é exibido como uma lista na forma <variável, valor, comentário>. Alguns dados de interesse, nesse trabalho, são os seguintes:
sim_seconds : Tempo (segundos) que foi simulado. É o tempo que o programa usaria se executasse no processador que foi modelado.
sim_insts : Quantidade de instruções que foram simuladas.
system.cpu.numCycles : Quantidade de ciclos da execução do programa, ou seja, quantos ciclos usaria se executasse no processador modelado.t
System.cpu.ipc : IPC (instruções por ciclo) da execução.

Outros dados interessantes:
host_seconds : Quanto tempo (segundos) a simulação usou na máquina rodando o simulador.
system.cpu.dcache.overall_miss_rate::total : Taxa de miss da cache de dados
system.cpu.icache.overall_miss_rate::total : Taxa de miss da cache de instruções
system.cpu.l2cache.overall_miss_rate::total : Taxa de miss da cache L2
system.cpu.l3cache.overall_miss_rate::total : Taxa de miss da cache L3
system.cpu.commit.op_class_0::TIPO : Classe de cada instrução completada. CUIDADO: branches são contados como operações de inteiros.
System.cpu.commit.branches : Quantidade de branches completados.
system.cpu.branchPred.condPredicted : Quantidade de branches "previstos" pelo preditor de desvios
system.cpu.branchPred.condIncorrect : Quantidade de branches que foram preditos incorretamente (a razão entre essas duas grandezas fornece a taxa de acertos do preditor de desvios).

```
* Executando o gem5...
* gem5 --outdir=m5out MySimulation.py -c hello
************************************************************************

gem5 Simulator System. http://gem5.org
gem5 is copyrighted software; use the --copyright option for details.

gem5 compiled Feb 16 2016 16:35:34
gem5 started May 18 2016 17:31:02
gem5 executing on simulacaolse3
command line: gem5 --outdir=m5out MySimulation.py -c hello


Global frequency set at 1000000000000 ticks per second
warn: DRAM device capacity (8192 Mbytes) does not match the address range assigned (512 Mbytes)
0: system.remote_gdb.listener: listening for remote gdb on port 7000

************************************************************************
* Início da simulação...
* Redirecionando stdout...
************************************************************************
info: Entering event queue @ 0. Starting simulation...
Hello world!

Fim da simulação.
Tick atual: 24206000. Motivo: target called exit()


************************************************************************
* Resultados da simulação
************************************************************************

sim_seconds 0.000024 # Number of seconds simulated
sim_ticks 24206000 # Number of ticks simulated
final_tick 24206000 # Number of ticks from beginning of simulation (restored from checkpoints and never reset)
sim_freq 1000000000000 # Frequency of simulated ticks
host_inst_rate 32839 # Simulator instruction rate (inst/s)
host_op_rate 65622 # Simulator op (including micro ops) rate (op/s)
host_tick_rate 167696036 # Simulator tick rate (ticks/s)
```

Figura 3. Exemplo de execução correta de um programa no simulador.

3. Método de entrega

Terá de ser feita uma apresentação em Powerpoint com no máximo 10 slides (incluindo título), que irá ser apresentada ao restante da turma ou ao professor. Se for o caso, a ordem de apresentação se dará por sorteio, feito no mesmo dia da apresentação; ou definida pelo professor previamente.

4. Data de Apresentação

Encontra-se no plano de ensino.

A. Apêndice A - Instalação do gem5 na máquina pessoal

É possível instalar e executar o gem5 na sua máquina pessoal. Porém, **dada a complexidade de configuração da ferramenta, este procedimento não é recomendado**.

Se ainda assim preferir este método, a versão mais recente do código está disponível em https://github.com/gem5/gem5.

Para instalar o gem5 e efetuar a simulação na sua máquina pessoal, alguns cuidados têm de ser tomados. São dois problemas:

1. a versão do gem5 que roda no nosso servidor (@simulacaolse3.inf.ufrgs.br) é diferente da que está disponível no git, e os desenvolvedores tem o habito de mudar os nomes das variáveis de uma versão pra outra do código :) Logo, os scripts disponibilizados no site não são compatíveis com as versões mais recentes.
2. a interface web utilizada na simulacaolse3 não mostra todos os scripts utilizados. Isso foi feito para simplificar a utilização.

No moodle está disponível uma versão "atualizada" dos scripts necessários para a simulação. São dois:

1. O3System.py : Define as caches, a CPU e o sistema. É, basicamente, os mesmos três arquivos disponibilizados no site em um só.
2. single_thread_sim.py : É o arquivo que configura a simulação, instancia o O3System e o programa a ser simulado (esse é o que optamos por fazer rodar behind-the-scenes para simplificar a vida).

Compile o gem5 utilizando o comando
scons -Q build/X86/gem5.opt-

(se estiver em uma máquina multicore, pode utilizar também o parâmetro -j <num_threads> para efetuar o processo de compilação mais rapidamente com múltiplas threads).

Extraia o conteúdo do zip dentro da pasta configs do gem5. Vai ficar o caminho configs/minhas_configs com os dois arquivos dentro.

Chamem a simulação com o seguinte comando

<binário do gem5> <opções do binário do gem5> <script de simulação em python> <opções do script de simulação em python>

no meu caso ficou:

build/X86/gem5.opt configs/minhas_configs/single_thread_sim.py -c tests/test-progs/hello/bin/x86/linux/hello

onde

<binário do gem5> = build/X86/gem5.fast
<opções do binário do gem5> = NULL
<script de simulação em python> = configs/minhas_configs/single_thread_sim.py
<opções do script de simulação em python> = -c tests/test-progs/hello/bin/x86/linux/hello

(façam o teste executando o hello world que vem pré-compilado, antes de qualquer coisa, para ter certeza de que o script está funcionando).

Estes scripts foram testados sob o commit dbeeb9693c8ab364fca7cf01817c9628252af652 do código, de

14.07.2017. Para usar esta versão do codigo, faça

git checkout dbeeb9693c8ab364fca7cf01817c9628252af652