

Trabalho 2 - Introdução à Computação Científica

Arianne Bortolan

Novembro de 2018

1 Introdução

Este relatório tem como objetivo apresentar os resultados dos testes e otimizações realizadas no código do trabalho 1 e justificar as mesmas.

2 Configurações Hardware

A máquina em que foram realizados os testes foi a i27 do laboratório 4. Sua configuração de hardware, a máquina contém um processador: Intel(R) Core(TM) i5-7500, ele é modelo Kabylake e tem um clock de 3.40GHz. O processador contém 4 núcleos físicos e 1 virtual. O espaço de suas caches L1, L2 e L3 são respectivamente 32kB, 256kB e 6MB.

3 Otimizações Realizadas

As otimizações realizadas foram na função `mult_soma_vetor` e na seção de `mult_soma_vetor` em gradiente conjugado. Na versão 1 do código tínhamos:

```
if(produto != 0.0){
    w= rp/produto;
}else{
    w= rp;
}
mult_soma_vetor(dados_variaveis->vetor_x,xa,v,w,dados_variaveis->N);
mult_soma_vetor(dados_variaveis->vetor_r,dados_variaveis->vetor_r,z,(-w),tam);
```

Enquanto na versão 2 ficou:

```
if(produto != 0.0){
    w= rp/produto;
}else{
    w= rp;
}
```

```

//mult_soma_vetor otimizado
for(cont = 0; cont < tam; cont++){
    dados_variaveis->vetor_x[cont] = xa[cont] + (w * v[cont]);
    r[cont] = r[cont] + ((-w) * z[cont]);
}

```

Essa alteração foi realizada para a diminuição de loops e consequentemente otimização de tempo. Outra alteração foi na própria função `mult_soma_vetor`, realizando o loop unrolling para otimização.

4 Resultados

Os testes foram realizados utilizando scripts de automatização, o primeiro deles `benchmark` automatiza os testes e saídas em loop, o `e_perfctr` executa os comandos enquanto o `plot` faz a plotagem de gráficos. Todos os testes foram realizados na máquina especificada, porém o máximo de variáveis no sistema em que os testes conseguiram ser executados foi 4000, os testes em 8000 o processo foi encerrado por estar executando a tempo demais. Sendo assim todos os gráficos contem dados de 32 até 4000, outro problema foi que por algum motivo o `likwid` não capturou o cache miss e nem a banda de memória das operações 1 na versão 1 do programa, tendo assim somente esses dados na versão 2. Respondendo as perguntas, o código só é capaz de resolver sistemas de até 4000 em tempo hábil (executavel no `dinf`) a interação na versão 2 levaria entre 15 a 20 segundos, enquanto na versão 1 isso leva mais de 1 minuto.

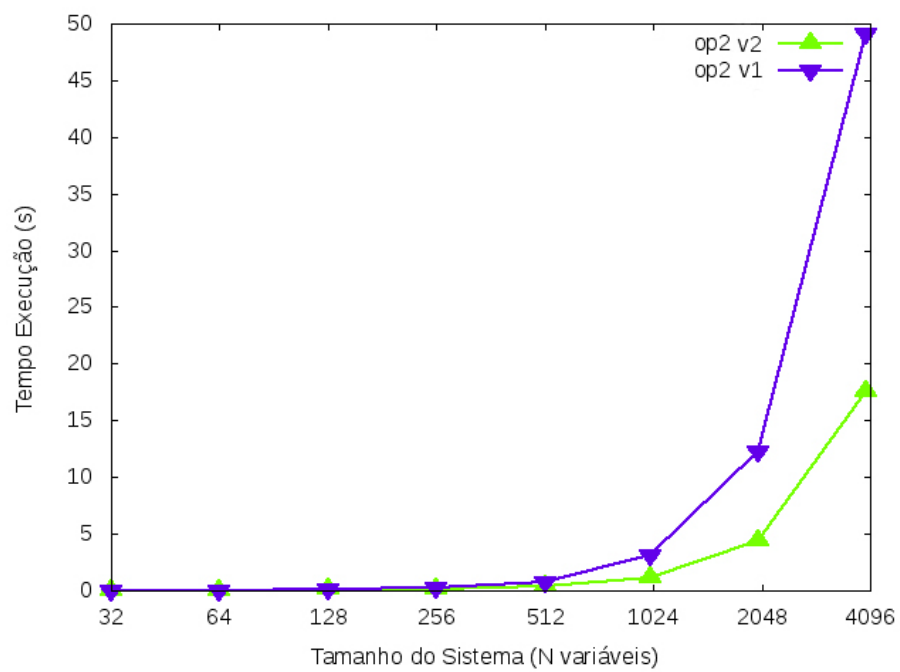


Figure 1: Gráfico Tempo de Execução V1 e V2

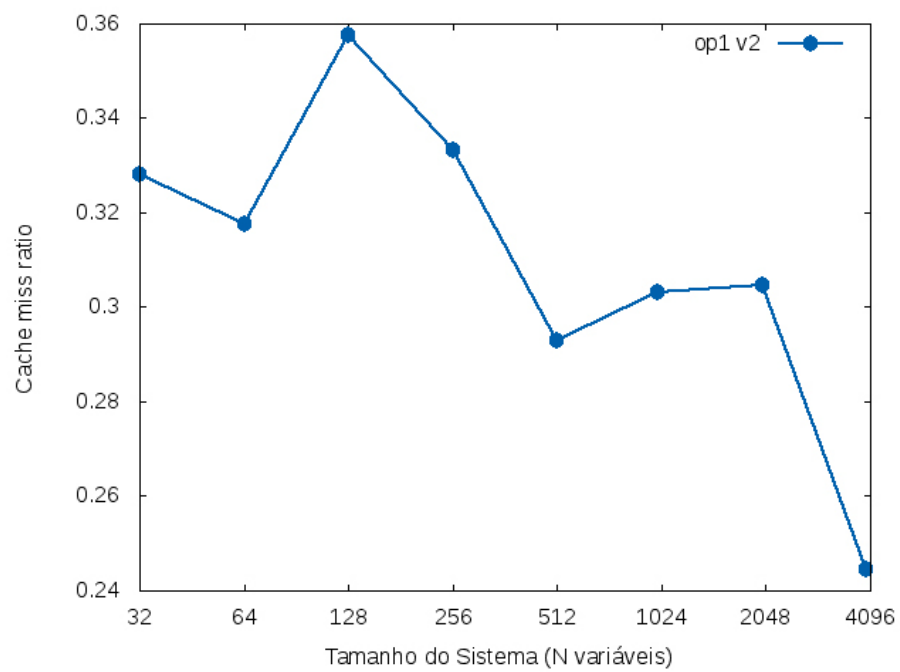


Figure 2: Gráfico de cache miss em V2

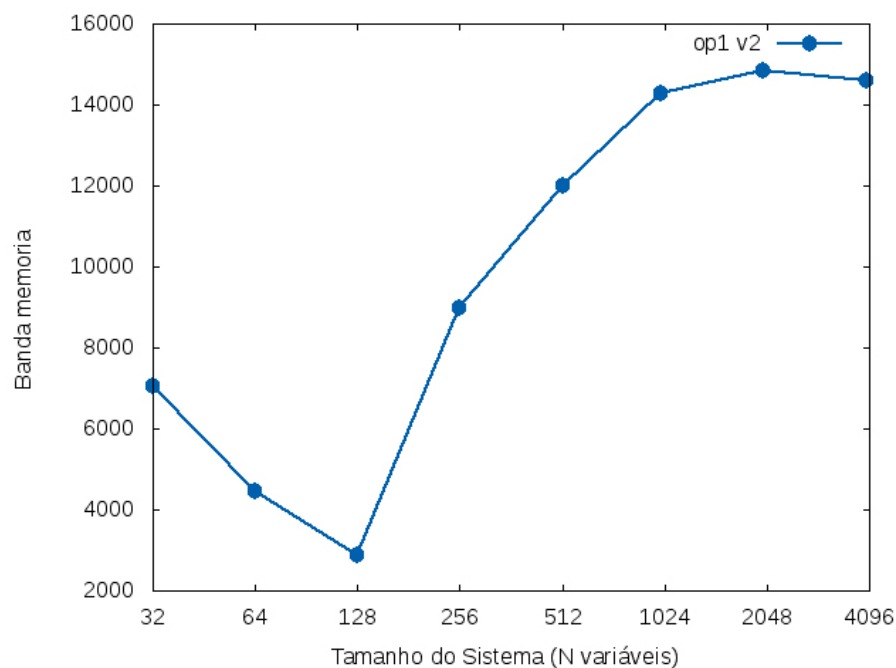


Figure 3: Gráfico de banda de memória usada em V2

5 Conclusão

As pequenas mudanças demonstraram uma melhora no desempenho no quesito tempo de execução embora não foi possível comparar os outros dados como cache miss e banda de memória tendo assim resultados não tão satisfatórios com relação aos testes e melhorias entre a versão 1 original do código e a versão 2 com as alterações relatadas. Novas melhorias e outros testes resultariam em melhoras significativas ao programa.