Universidade Federal do Ceará
Campus Quixadá - PCOMP
Análise de Desempenho
Luzia Nogueira - 503740

Relatório TFlashM Benckmark

1. Nome do benchmark: TFlashM Benckmark

2. Descrição do benchmark: O TFlashM benckmark é um benchmark para análise de desempenho de dispositivos de memória flash, especialmente pendrive, sua análise é realizada com base na velocidade de escrita e leitura. É usado para teste da escrita 50MB de carga de trabalho multiplicado pela quantidade de repetição. A leitura também recebe uma quantidade de repetição. A quantidade de repetição é a mesma para os testes de leitura e de escrita e é solicitada pelo programa durante a execução.

3. Detalhes de código do benchmark: Arquivo disponibilizado em anexo

4. Métricas:
   1- Velocidade de Leitura: Tempo necessário para realizar a leitura de determinada quantidade de arquivo.
   2- Velocidade de Escrita: Tempo necessário para realizar a escrita de determinada quantidade de arquivo.

5. Manual de instalação:
   1- O arquivo foi escrito na linguagem de programação python e para executar o mesmo é necessário que a máquina tenha o python3 instalado, caso não tenha instalado utilizar o seguinte comando para instalar: sudo apt-get install python3.
   2- O arquivo utiliza algumas bibliotecas, caso não estejam instaladas é necessário que sejam instaladas, caso contrário irá da erro na compilador do programa. As bibliotecas são as seguintes:
      - subprocess
      - matplotlib

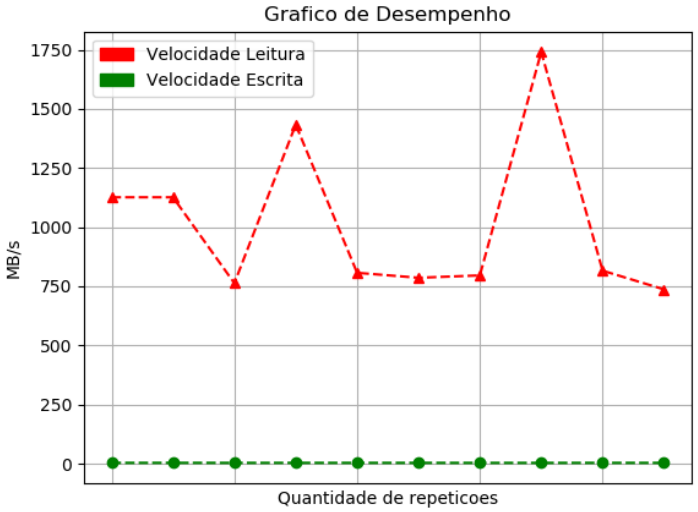
6. Execução
   1. Baixe o scrip.py na maquina
   2. Pluge o pendrive via usb que deseja testar na máquina
   3. Utilize o comando para execução: python3 script.py
   4. Os parâmetros são pedidos durante a execução.
   5. Caso gere erro, verifique se a máquina possui todas as bibliotecas instaladas e se o pendrive esta sendo reconhecido.

7. Exemplos de utilização

```
marcos@mp-notebook:~/Documentos/benchmark_luzia$ python3 script.py
Sist. Arq.     Blocos de 1K    Usado Disponível Uso% Montado em
udev              3991836       0    3991836   0% /dev
tmpfs              802532    9324     793208   2% /run
/dev/sda6       128953508 61096044   61263928  50% /
tmpfs             4012644  158572    3854072   4% /dev/shm
tmpfs                5120       4       5116   1% /run/lock
tmpfs             4012644       0    4012644   0% /sys/fs/cgroup
/dev/sda3          231911   58877     156753  28% /boot
tmpfs              802528      20     802508   1% /run/user/1000
/dev/sdc1        15137792 1029784   14108008   7% /media/marcos/MP

Por favor, informe o caminho do pendrive. Ex: /media/pendrive
/media/marcos/MP
Informe a quantidade de repeticoes desejadas
10
################## BENCHMARK TFLASHM ##################
Tempo total de execução:  1.573676009 s
Tempo Médio de Escrita:  0.15681509999999999 s
Tempo Médio de Leitura:  0.0005525009 s
Velocidade Média de Escrita:  3.28 MB/s
Velocidade Média de Leitura:  1013.22 MB/s
######################Valores por Teste################
Velocidade Escrita - MB/s
[3.3, 3.4, 3.2, 3.3, 3.3, 3.3, 3.2, 3.2, 3.3, 3.3]
Variancia de Escrita:  0.0035999999999999843
Velocidade de Leitura - MB/s
[1126.4, 1126.4, 763.0, 1433.6, 807.0, 786.0, 796.0, 1740.8, 816.0, 737.0]
Variancia de Leitura:  105085.48359999999
############################################################
Obs: Quanto menor a variância mais próximo os valores estão em relação a média.
marcos@mp-notebook:~/Documentos/benchmark_luzia$
```

8. Telas de resultados dos exemplos



9. Um estudo de caso: O teste foi realizado em um pendrive de 32GB memória, antes de ser utilizado o pendrive foi formatado e não contia nenhum arquivo gravado. As imagens acima são foram obtidas nesse teste.

Anexo I - Códigos

```
import time
import os
import subprocess
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import numpy as np


def main():
    def testEscrita(path):
        proc = subprocess.Popen(
            "dd if=/dev/zero of="+path+"/arquivo bs=5k count=100", shell=True,
            stdout=subprocess.PIPE, stderr=subprocess.PIPE, stdin=subprocess.PIPE)
        output, err = proc.communicate()
        r1 = err.decode("utf-8").split("\n")[2]
        t1 = r1.split(" ")[7]
        s1 = float(r1.split(" ")[9].replace(",", "."))
        unid = r1.split(" ")[10]
        if (unid == "kB/s"):
            s1 = s1/1024
        return t1, s1

    def testLeitura(path):
        proc = subprocess.Popen(
            "dd if="+path+"/arquivo of=/dev/null bs=4k", shell=True,
            stdout=subprocess.PIPE, stderr=subprocess.PIPE, stdin=subprocess.PIPE)
        output, err = proc.communicate()
        r1 = err.decode("utf-8").split("\n")[2]
        t1 = r1.split(" ")[7]
        s1 = float(r1.split(" ")[9].replace(",", "."))
        unid = r1.split(" ")[10]
        if (unid == "GB/s"):
            s1 = s1*1024
        return t1, s1

    def selectPath():
        path = os.path.curdir
        if os.name == 'posix':
            result = subprocess.check_output("df")
            print(result.decode("utf-8"))
            path = input(
                "Por favor, informe o caminho do pendrive. Ex: /media/pendrive \n")
            rep = int(input("Informe a quantidade de repeticoes desejadas\n"))
        if os.name == 'nt':
            path = input("Entre com a Letra do Disco. Ex: D")
            path = path.split(":")
```

```
    if not os.path.exists(path):
        print('Caminho invalido')
        selectPath()

    return path, rep

def plotarGraf(VelEsc, VelLeit):
    plt.plot(VelEsc, 'go')  # green bolinha
    plt.plot(VelEsc, 'k--', color='green')  # linha pontilha

    plt.plot(VelLeit, 'r^')  # red triangulo
    plt.plot(VelLeit, 'k--', color='red')  # linha tracejada

    plt.title("Grafico de Desempenho")
    plt.tick_params(axis='x', which='both', bottom=False,
                    top=False, labelbottom=False)
    red_patch = mpatches.Patch(color='red', label='Velocidade Leitura')
    green_patch = mpatches.Patch(color='green', label='Velocidade Escrita')
    plt.legend(handles=[red_patch, green_patch])

    plt.grid(True)
    plt.xlabel("Quantidade de repeticoes")
    plt.ylabel("MB/s")
    plt.show()

path, rep = selectPath()
i = rep
ArrayVelEscrita = []
ArrayVelLeitura = []
tempoTotalE = 0
tempoTotalL = 0

while(i > 0):
    tempoE, velocidadeE = testEscrita(path)
    tempoTotalE = float(tempoE.replace(",", ".")) + tempoTotalE
    ArrayVelEscrita.append(velocidadeE)

    tempoL, velocidadeL = testLeitura(path)
    tempoTotalL = float(tempoL.replace(",", ".")) + tempoTotalL
    ArrayVelLeitura.append(velocidadeL)
    i -= 1

EscritaMedia = np.mean(ArrayVelEscrita)
LeituraMedia = np.mean(ArrayVelLeitura)
VarianciaEscrita= np.var(ArrayVelEscrita)
VarianciaLeitura = np.var(ArrayVelLeitura)

print("#################### BENCHMARK TFLASHM ####################")
print("Tempo total de execução: ", (tempoTotalE + tempoTotalL), "s")
```

```
    print("Tempo Médio de Escrita: ", tempoTotalE/rep, "s")
    print("Tempo Médio de Leitura: ", tempoTotalL/rep, "s")
    print("Velocidade Média de Escrita: ", EscritaMedia, "MB/s")
    print("Velocidade Média de Leitura: ", LeituraMedia, "MB/s")

    print("########################Valores por Teste###################")
    print("Velocidade Escrita - MB/s")
    print(ArrayVelEscrita)
    print("Variancia de Escrita: ", VarianciaEscrita)

    print("Velocidade de Leitura - MB/s")
    print(ArrayVelLeitura)
    print("Variancia de Leitura: ", VarianciaLeitura)
    print("############################################################")

      print("Obs: Quanto menor a variância mais próximo os valores estão em relação a média.")

    plotarGraf(ArrayVelEscrita, ArrayVelLeitura)

if __name__ == "__main__":
    main()
```