

Evaluation de performance

Compte rendu TP2 :

Réalisé par :

Hazem Mejri et Nada Ben Slimen

https://github.com/nadabsn/Eval_performance

Année universitaire : 2020/2021

ISITCOM

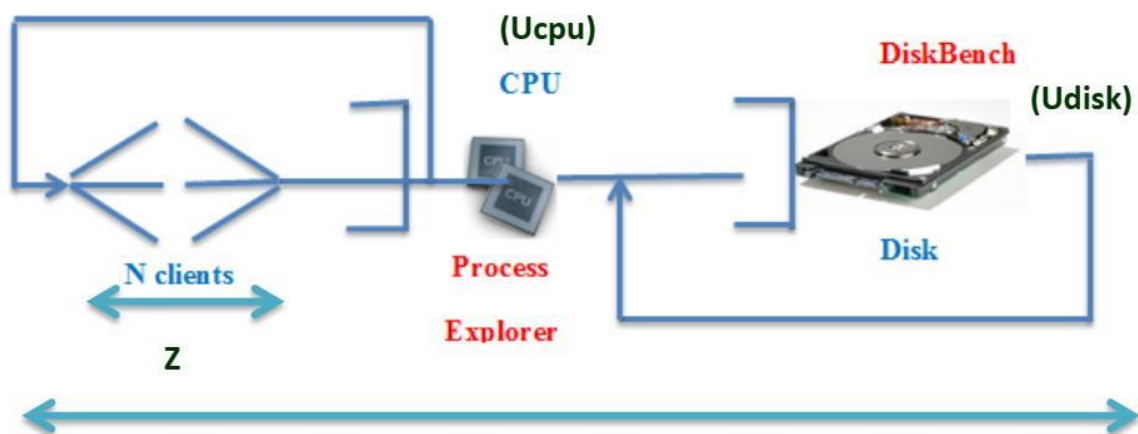
OBJECTIFS DU TP :

L'objectif de ce TP est de se familiariser avec la modélisation d'un système informatique :

- Faire une caractérisation de la charge au niveau du système.
- Estimer et analyser le débit et le temps de réponse.

PRESENTATION DU SYSTEME

Le système fermé étudié dans ce TP consiste à répondre aux demandes de N clients en termes d'exécution des tâches sur le microprocesseur (CPU) et en termes d'accès au disque Dur (Disk). La demande de chaque client passe par le microprocesseur pour avoir par la suite l'accès au disque Dur. Une fois l'accès au disque est terminé, le CPU s'occupe du renvoi de la réponse au client.



Les outils :

- **Diskbench** : C'est un utilitaire conçu pour calculer précisément la vitesse réelle de Disque Dur.
- **Process Explorer** : C'est un gestionnaire évolué qui permet de visualiser/gérer les processus et aussi de donner le temps d'accès de ces processus à la CPU.
- **Jupyter notebook** : L'éditeur et le compilateur du code python.

TRAVAIL DEMANDE

L'application que nous avons créé permet de lancer automatiquement le logiciel « **ProcessExplorer** » et de préciser le nombre de fois pour lancer les « **diskbench** » simultanément afin de capturer les **TCPU** et **Tdisk**.

❖ *Code python de l'interface demandée :*

```
import tkinter as tk
from tkinter import Label, Entry
import os, _thread
from PIL import Image, ImageTk
def run_proc():

    os.system(r"D:\NADA\Downloads\Compressed\procexp.exe")

def para_window():

    win_2 = tk.Tk()
    win_2.title("SETTING")
    win_2.geometry("400x300")
    frame_2 = tk.Frame(win_2)
    frame_2.pack()
    txt_2=Label(frame_2,text="Combien de Disk voulez-vous ?")
    txt_2.grid(row=0,column=3,pady=20,padx=100)
    box=Entry(frame_2)
    box.grid(row=1,column=3)
    btn_valide = tk.Button(frame_2,text="Valider",fg="red")
    btn_valide.grid(row=2,column=3,pady=10,padx=100)

    try:

        def open_disk():
            os.system(r"D:\NADA\Downloads\Compressed\DiskBench.exe")

        def run_disk():
            n=int(float(box.get()))
```

```

def run_disk():
    n=int(float(box.get()))

    for i in range(1,n+1):
        thread.start_new_thread(open_disk, ())
    win_2.destroy()

    btn_valide.config(command=run_disk)

except ValueError:
    return "nope"

win=tk.Tk()
win.title("Evaluation de performance")
win.geometry("800x250")
frame=tk.Frame(win)
frame.pack()
logo = tk.PhotoImage(master=win ,file=r"D:\NADA\Downloads\logo_isitcomm.gif")
btn_proc=tk.Button(frame,text="PROCESS EXPLORER",fg="red",height="2" ,command=run_proc)
btn_disk=tk.Button(frame,text="DISK BENCH",fg="green",height="2" ,command=para_window)

btn_proc.grid(row=3,column=3,pady=10,padx=10)
btn_disk.grid(row=2,column=3,pady=10,padx=10)

txt=Label(frame,text="Réalisé par BEN SLIMEN Nada et MEJRI Hazem", fg = "dark blue", bg = "light blue",font = "Helvetica 16")
w1 = tk.Label(frame, image=logo)
txt.grid(row=0,column=3,pady=10,padx=100)
w1.grid(row=0,column=0,pady=15,padx=10)
win.mainloop()

```

Résultat de l'exécution :

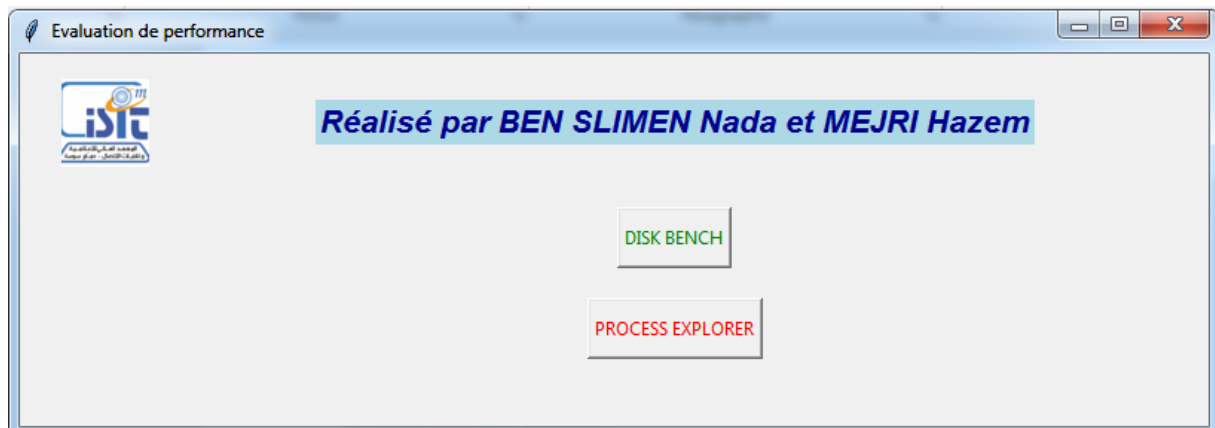


Figure 1 Interface 1

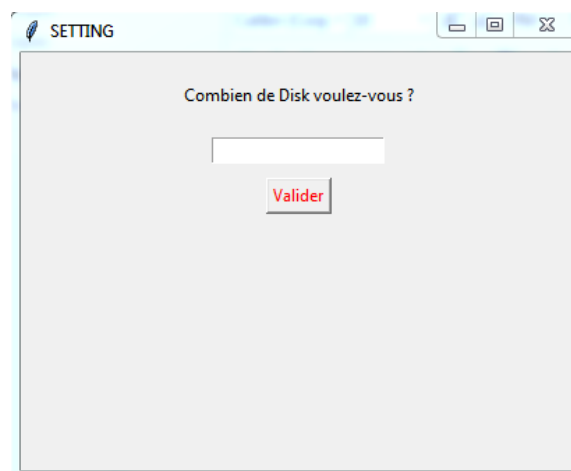
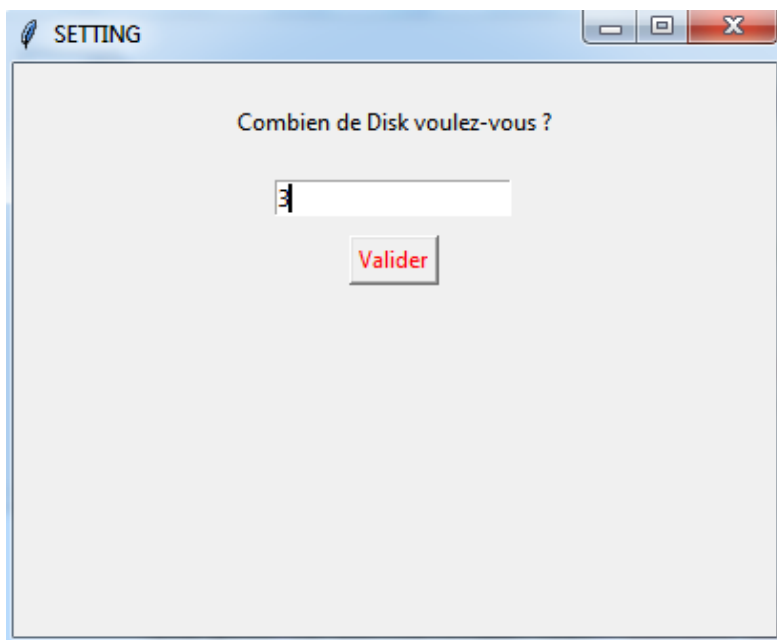


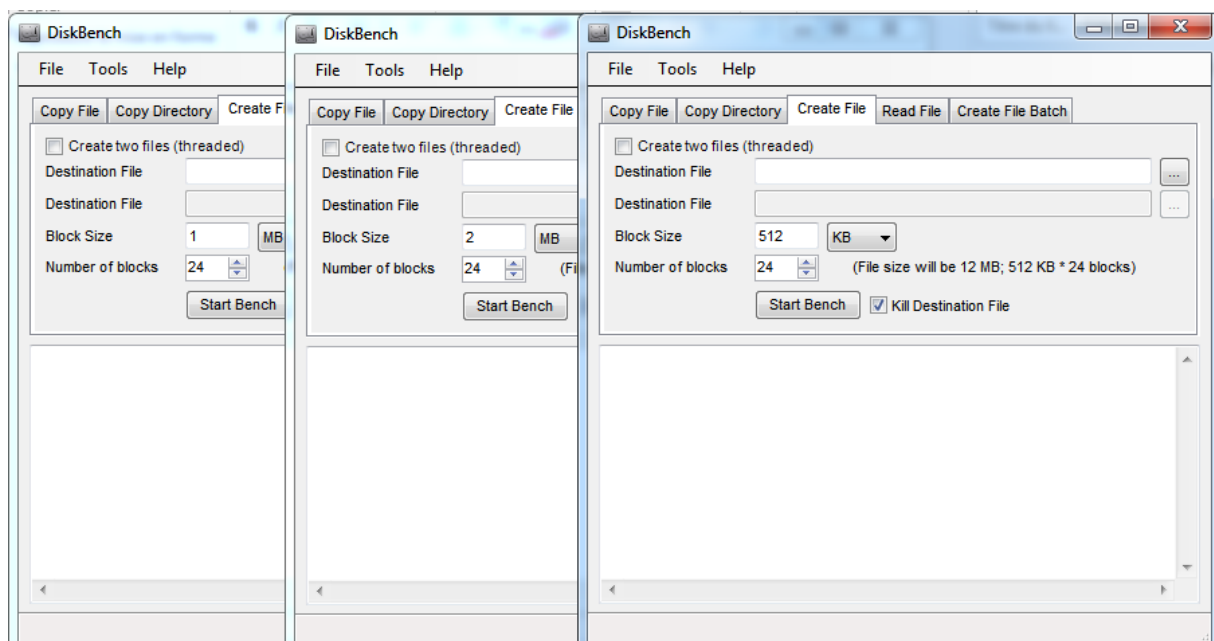
Figure 2 Interface 2

1- Pour ce test, on fixe N=3 pour lancer le DiskBench 3 fois.

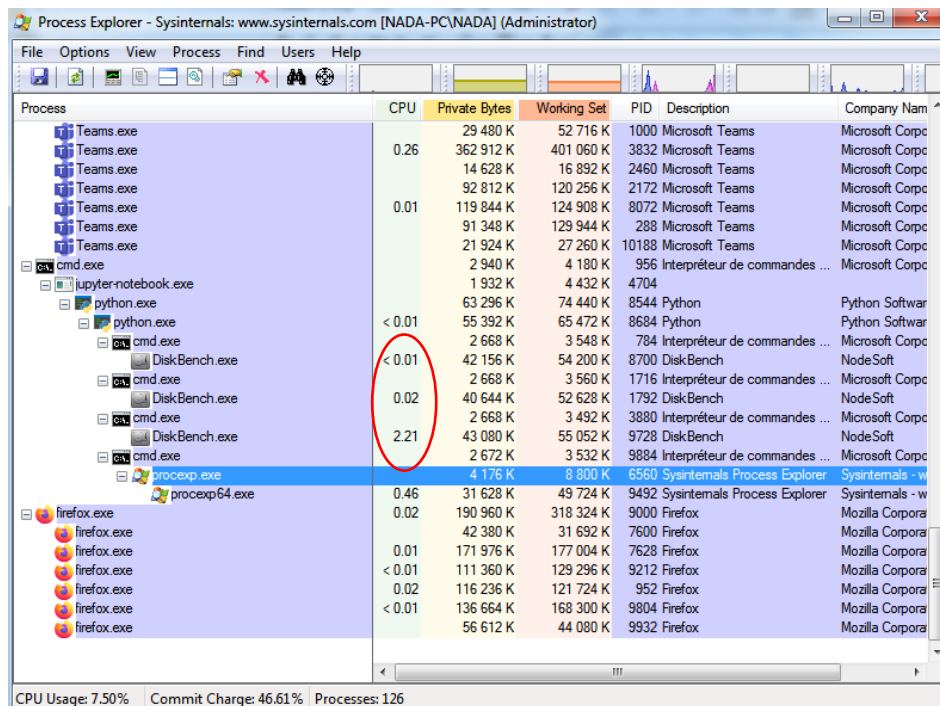


A travers l'outil « DiskBench » on crée simultanément 3 fichiers ayant chacun 24 blocks :

- Taille block fichier 1 : 512 ko
- Taille block fichier 2 : 1 Mo
- Taille block fichier 3 : 2 Mo



A travers l'outil Process Explorer on récupère le « busy time » associé à la demande de chaque client.



2- On répète l'opération durant 15 minutes (Le Temps d'observation globale T) tout en comptant le nombre d'achèvement des tâches demandées par les 3 clients. Ce nombre représentera le C₀ (total number of completion).

	Client 1		Client 2		Client 3	
	TCPU	Tdisk	TCPU	Tdisk	TCPU	Tdisk
C₀=1	1.44	69	2.12	123	0.29	38
C₀=2	1.38	73	2.07	113	0.29	54
C₀=3	1.47	68	0.28	119	0.97	34
C₀=4	0.26	69	2.06	118	1.10	44
C₀=5	0.28	65	0.29	128	0.29	43
C₀=6	1.17	59	1.73	130	1.10	39
C₀=7	1.49	70	0.30	121	1.13	41
C₀=8	0.29	62	2.31	136	0.58	37
C₀=9	0.31	67	2.03	112	0.29	35
C₀=10	0.25	72	0.22	124	1.18	45
C₀=11	1.51	59	0.27	219	0.01	34
C₀=12	1.38	66	1.98	114	0.89	33

$C_0=13$	0.29	65	0.33	128	1.05	38
$C_0=14$	0.28	68	0.23	122	0.67	32
$C_0=15$	0.63	64	0.27	119	0.98	39
$C_0=16$	0.97	76	2.32	148	0.97	35

C_0 : Le nombre de fois que le client a été servi : *Total number of Completion*

⇒ $C_0=16$

a- On calcule maintenant le « Busy Time » total de CPU et de disque dur :

- CPU :

$T_{CPU} \text{ (Client 1)} = 13.4\text{ms}$

$T_{CPU} \text{ (Client 2)} = 18.81\text{ms}$

$T_{CPU} \text{ (Client 3)} = 11.79\text{ms}$

⇒ $T_{CPU} = 44\text{ms}$

- Disque dur :

$T_{disk} \text{ (Client 1)} = 1072\text{ms}$

$T_{disk} \text{ (Client 2)} = 2074\text{ms}$

$T_{disk} \text{ (Client 3)} = 621\text{ms}$

⇒ $T_{disk} = 3767\text{ms} = 3.8\text{s}$

b- On calcule la moyenne de « Thinking Time » : Z

Z est la moyenne des Z_i déterminés dans le tableau précédent :

$$Z = \frac{\sum Z_i}{\text{Nbre des } Z_i} = \frac{Z_1 + Z_2 + \dots + Z_n}{C_0 - 1}$$

AN :

$$Z = (1 + 1 + 1 \dots + 1) / 15 = 1.06\text{min}$$

c- On déduit donc le temps d'utilisation $UCPU$ et U_{disk} :

- $UCPU = T_{CPU}/T = 44.10^{-3} / (15 \cdot 60) = 0.05ms$
- $U_{disk} = T_{disk}/T = 3767.10^{-3} / (15 \cdot 60) = 4.18ms$

d- On déduit la demande globale D_i avec $i=\{CPU, Disk\}$ et $D_i = (U_i \cdot T)/C_0$

$$DCPU = U_{CPU}/C_0 = 0.05 \cdot 10^{-3} \cdot 15 \cdot 60 / 16 = 0.003$$

$$D_{disk} = U_{disk}/C_0 = 4.18 \cdot 10^{-3} \cdot 15 \cdot 60 / 16 = 0.24$$

3- L'analyse asymptotique :

C'est une analyse qui nous permet de déterminer les bornes minimales et maximales pour le débit et le temps de réponse.

Pour un système fermé la formule de **débit** est :

$$\frac{N}{ND + Z} \leq X(N) \leq \min\left(\frac{1}{D_{max}}; \frac{N}{D + Z}\right)$$

On a :

- $N = 3$ clients
- $Z = 63.6s$
- $D_{max} = D_{disk} = 0.24$
- $D = D_{cpu} + D_{disk} = 0.003 + 0.24 = 0.243$

AN :

$$\frac{N}{N \cdot D + Z} = \frac{3}{3 \cdot 0.243 + 63.6} = 0.045$$

$$\frac{1}{D_{max}} = \frac{1}{0.24} = 4.16$$

$$\frac{N}{D + Z} = \frac{3}{0.243 + 63.6} = 0.045$$

Le temps de réponse est :

$$\max(ND_{max} - Z; D) \leq R(N) \leq ND$$

AN :

$$\max(ND_{max} - Z; D) = \max(3 \cdot 0.24 - 63.6; 0.243) = 0.243$$

$$ND = 3 \cdot 0.243 = 0.729$$

❖ **Le code matlab :**

```
Z=63.6  
Dmax=0.24  
D=0.243  
N=[1:0.1:100]  
H=(N.*Dmax)-Z;  
L=1/Dmax;  
F=N.*D  
plot(N,H,'g',N,L,'b',N,F,'r')  
%plot(N,L,'*r')  
grid on
```

❖ **La simulation :**

