

# PRÁCTICA TEMA 2 - MONITORIZACIÓN

Avaluació del Comportament  
de Sistemes Informàtics

Curso 2024/25

Ignasi Paredes Casasnovas  
Jaume Galmés Ramis

# MONITORIZACIÓN DE LA CPU

## 1. ¿Cuántas CPUs tiene el sistema que se ha monitorizado? ¿De dónde se ha obtenido esa información?

Con el comando: `lscpu | grep "CPU(s):"` obtenemos el siguiente resultado: **CPU(s): 16**, lo que significa que el sistema que hemos monitorizado, cuenta con 16 CPUs.

## 2. ¿Cuál es la utilización media de la CPU en modo usuario, sistema y en global?

Para obtener las medias de utilización de CPU pedidas, hemos usado un comando en bash con TOP para obtener las distintas métricas.

Para obtener Timestamp, %CPU idle, %CPU user, %CPU system:

```
top -b -n1080 -d5 | grep -E -i "top |id," | awk '/top / {Timestamp = $3} /id,/ {CPUIdle = $8; if (CPUIdle == "id,") CPUIdle = "100.0"} /us,/ {CPUUser = $2} /sy,/ {print "Timestamp: ", Timestamp, " | %CPU idle: ", CPUIdle, " | %CPU user: ", CPUUser, " | %CPU system: ", $4}' > monitorizacionCPU.txt
```

De esta manera hemos obtenido un fichero .txt con los datos necesarios de la monitorización. Posteriormente hemos pasado los datos a un archivo .csv con un script en Java, aquí se ha cambiado el %CPU en idle a uso global de CPU restándole 100,00 al idle. Finalmente se ha cambiado a archivo .xlsx, el formato de Excel para calcular la media de cada una de las columnas.

Obtenemos que:

en **modo usuario** = 0,0165740740740741%  $\approx$  0,016574%

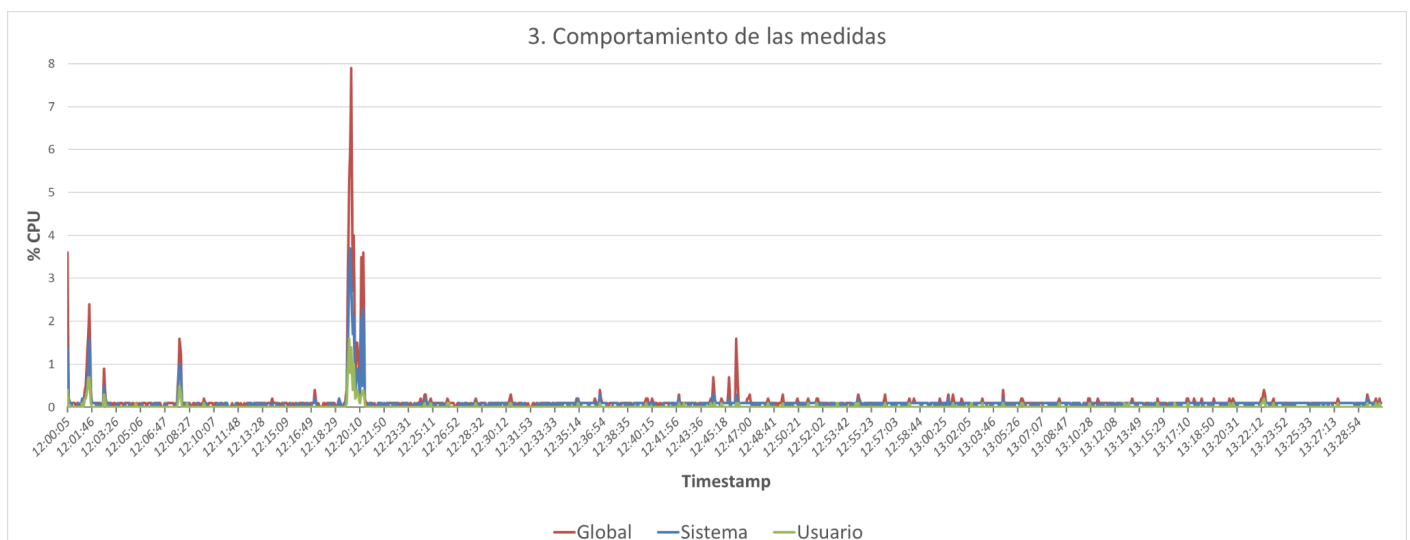
en **modo sistema** = 0,0821296296296287%  $\approx$  0,082130%

y en **modo global** = 0,1494444444444441%  $\approx$  0,149444%

## 3. ¿Cómo se comportan las medidas anteriores a lo largo del tiempo de observación? Muestra las tres métricas de forma gráfica.

Como se puede observar en el gráfico, la CPU estuvo con una carga muy baja durante la mayor parte de la prueba, con un 0,1% de uso mayoritariamente, con algunos picos puntuales, el más notorio en el minuto 20 de la prueba donde llega a casi 8% de uso global de CPU.

El uso de CPU dedicado a usuario es la más baja de las tres, el uso de sistema casi siempre es mayor al dedicado al usuario y el global es siempre igual o mayor al uso de usuario y sistema combinados.



**4. ¿Cuál es la sobrecarga provocada por el monitor TOP?**

Para calcular la sobrecarga, necesitamos el tiempo de ejecución del monitor que obtenemos mediante añadir time al principio del comando y quitando la parte del comando de redireccionar los datos al fichero, obtenemos un valor alrededor de 0.175 segundos y su intervalo de medida son 5 segundos en este caso.

La sobrecarga equivale a tiempo de ejecución del monitor entre el intervalo de medida

$$\text{Sobrecarga} = \frac{\text{Tiempo de ejecución del monitor}}{\text{Intervalo de medida}} = \frac{0,175}{5} = 0,035 * 100 = 3,5\%$$

# MONITORIZACIÓN DE LA MEMORIA PRINCIPAL

## 1. ¿Qué capacidad total tiene la memoria principal del sistema? ¿De dónde se ha obtenido ese dato?

El sistema que hemos monitorizado cuenta con **4036748 KB** de memoria total, que hemos obtenido con el comando: `vmstat -s | grep "total memory"`.

## 2. ¿Cuál es la utilización media de la memoria? ¿Y la capacidad media utilizada?

Para obtener las medias de utilización de memoria principal pedidas, hemos usado un comando en bash con `vmstat` para obtener las distintas métricas. Para obtener Timestamp, Capacidad disponible y Capacidad utilizada en cantidad de K y Memoria utilizada en porcentaje:

```
vmstat -n 3 3600 | gawk 'NR > 2 { printf "Timestamp: %s - Capacidad disponible: %s - Capacidad utilizada: %i - %% Memoria utilizada: %.2f\n", strftime("%H:%M:%S"), $4, 4036748-$4-$5-$6, (4036748-$4-$5-$6)/4036748*100 }' > monitorizacionRAM.txt
```

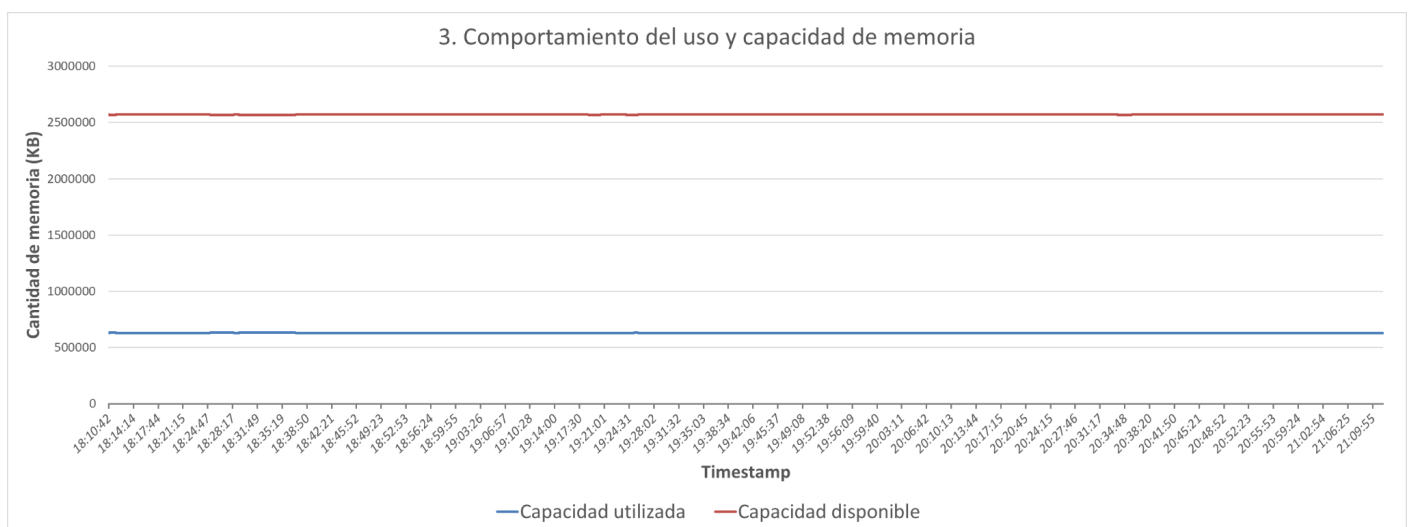
De esta manera hemos obtenido un fichero `.txt` con los datos necesarios de la monitorización. Posteriormente se han pasado los datos a un archivo `.csv` con un script en Java. Finalmente se ha cambiado a archivo `.xlsx`, el formato de Excel para calcular las medias.

*Utilización media de memoria* = 15,5590805555557%  $\approx$  15,559%

*Capacidad media utilizada* = 628077,177777778 KB  $\approx$  628077,178 KB

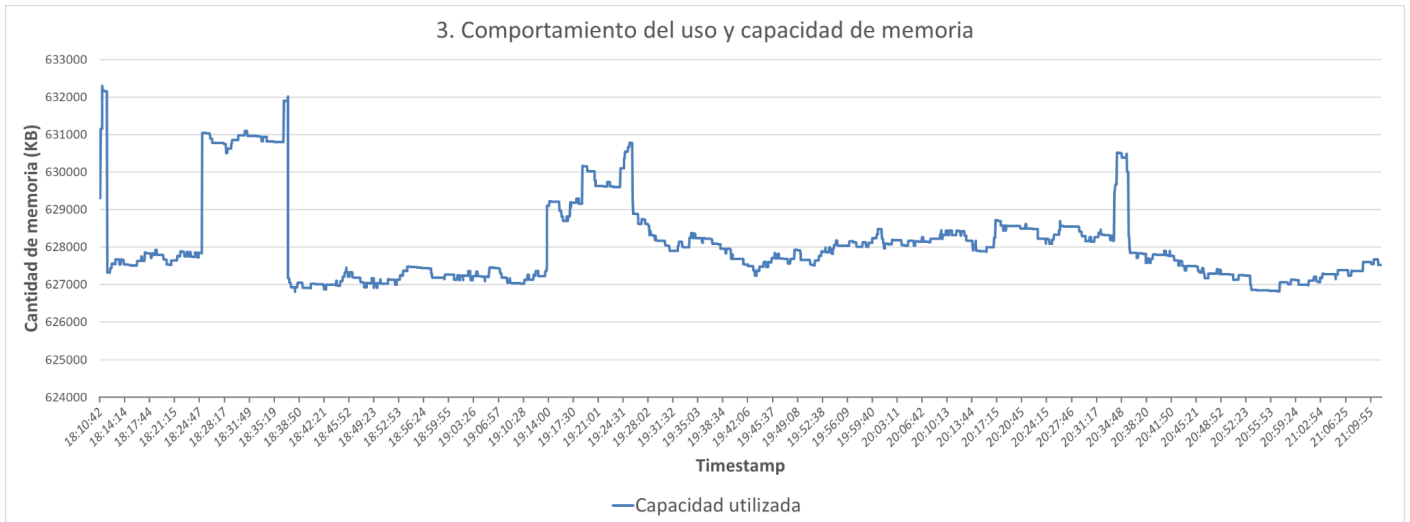
## 3. ¿Cómo se comporta la utilización de la memoria y la capacidad utilizada? Representa estas métricas gráficamente.

A grandes rasgos, el uso de memoria y la capacidad utilizada no se mueve considerablemente, es decir, se mantiene en los mismos niveles.



Ahora bien, si los vemos por separados y nos acercamos a la región donde se producen las oscilaciones, se notan mejor las pequeñas variaciones que ha tenido a través del tiempo. Observamos también que la capacidad disponible es el invertido de la capacidad utilizada. Los picos de uso de memoria más notorios están en los lapsos de tiempo entre el minuto 16 a 26, 63 a 75 y 144 a 146 de la prueba.

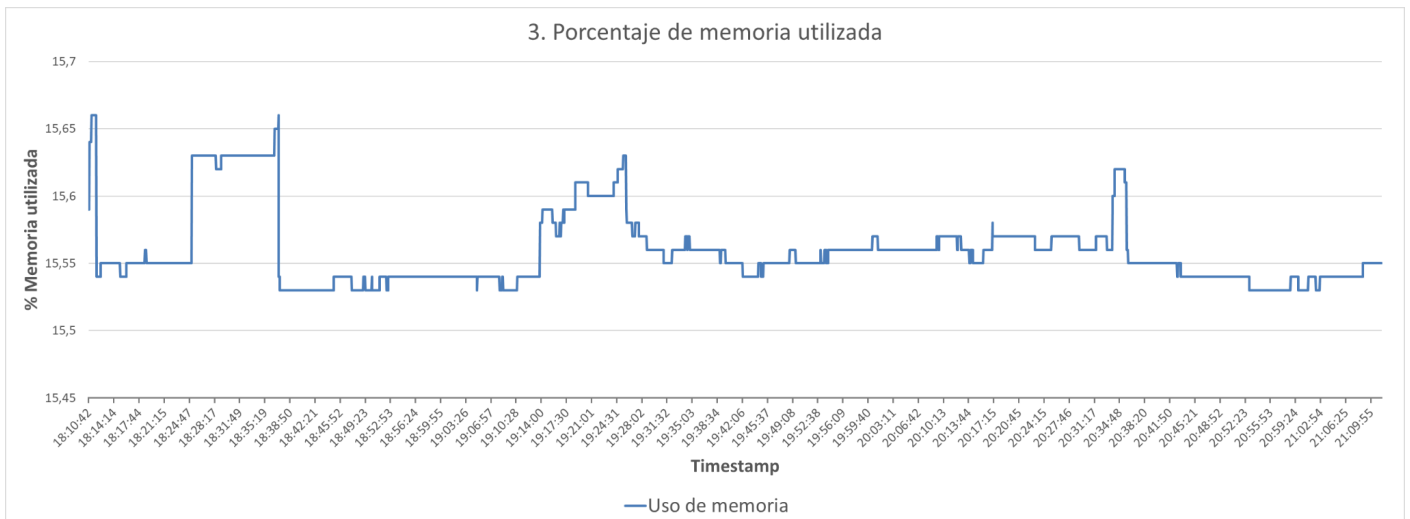
## Capacidad utilizada:



## Capacidad disponible:



## Porcentaje de uso de memoria:



#### 4. ¿Cuál es la sobrecarga provocada por el monitor VMSTAT?

Para calcular la sobrecarga, necesitamos el tiempo de ejecución del monitor que obtenemos mediante añadir time al principio del comando y quitando la parte del comando de redireccionar los datos al fichero, obtenemos un valor alrededor de 0.005 segundos y su intervalo de medida son 3 segundos en este caso.

La sobrecarga equivale a tiempo de ejecución del monitor entre el intervalo de medida

$$\text{Sobrecarga} = \frac{\text{Tiempo de ejecución del monitor}}{\text{Intervalo de medida}} = \frac{0.005s}{3s} = 0,00166666666666667 * 100 = 0,166666666666667 \%$$

Observamos una sobrecarga notoriamente menor que el monitor top a pesar de que el vmstat se haga en intervalos más cortos.

## MONITORIZACIÓN EN PARALELO

El comando en BASH utilizado para obtener la monitorización en paralelo es la siguiente:

```
top -b -n1440 -d5 | grep -E -i "top -| id, |KiB Mem : " | awk '/top -/ {Timestamp = $3} / id, / {CPUIdle = $8; if (CPUIdle == "id,") CPUIdle = "100.0"; ; gsub(", ", " ", CPUIdle); CPUIdle = CPUIdle + 0; CPUTotal = 100 - CPUIdle}; /KiB Mem :/ {mUsed = $8; mUsed = mUsed + 0; pmUsed = mUsed / 4036748 * 100; printf "Timestamp: %s | %% global CPU: %.1f | Capacidad de memoria utilizada: %d | %% Memoria utilizada: %.2f\n", Timestamp, CPUTotal, mUsed, pmUsed}' > monitorizacionParalelo.txt
```

Aquí se usa un solo monitor top para obtener en paralelo las métricas tanto de la CPU como de la memoria principal, que son guardadas al fichero .txt. Posteriormente se han pasado los datos a un archivo .csv con un script en Java, para finalmente cambiarlo a archivo .xlsx.

La sobrecarga del monitor mencionado equivale a su tiempo de ejecución entre el intervalo de medida

$$\text{Sobrecarga} = \frac{\text{Tiempo de ejecución del monitor}}{\text{Intervalo de medida}} = \frac{0,185}{5} = 0,037 * 100 = 3,7 \%$$

Cabe mencionar que como alternativa al comando anterior y a modo de prueba, también hemos programado el siguiente comando en BASH donde se usan los monitores top y vmstat para obtener la monitorización:

```
for v in $(seq 1 1440); do
# Marca de tiempo inicial
start_time=$(date +%s)

# Obtener timestamp
timestamp=$(top -b -n1 -d1 | grep "top -" | awk '{print $3}')

# Obtener uso de CPU (segunda muestra de top)
globalCPU=$(top -b -n2 -d1 | grep "%Cpu(s): " | awk 'NR > 1{print $8}')
if [[ "$globalCPU" == "id," ]]; then
globalCPU="100,0"
fi

# Obtener memoria usada en MB y porcentaje
memPercentage=$(vmstat -n 1 1 | gawk 'NR > 2 { printf "Capacidad utilizada: %i - %% Memoria utilizada: %.2f\n",
4036748-$4-$5-$6, (4036748-$4-$5-$6)/4036748*100 }')

# Mostrar resultado
echo "$timestamp: $globalCPU: $memPercentage"

# Calcular el tiempo transcurrido y ajustar el sleep
elapsed_time=$(( $(date +%s) - start_time ))
sleep_time=$((5 - elapsed_time))

# Evitar sleep negativo
[ $sleep_time -gt 0 ] && sleep $sleep_time
done > pruebaTiempoParalelo.txt
```

En este código se usan varios comandos con los monitores para obtener las diferentes métricas a medir. Para que se ejecute cada 5 segundos exactamente, usamos diferentes variables para que de este modo cada iteración del bucle for se realice siempre dentro del mismo intervalo ya que por el contrario podría hacer la medición en menos de 5 segundos, provocando que la monitorización acabe antes de las 2 horas.

Para calcular la sobrecarga, necesitamos el tiempo de ejecución del monitor que obtenemos mediante añadir time al principio de los comandos que contienen monitores, nótese que en el segundo comando top obtenemos la segunda medición de top, ya que la primera no es útil, por tanto tenemos que eliminar 1 segundo de esta métrica, que es el tiempo entre la primera medida de la primera monitorización y la segunda. De esta manera obtenemos un valor alrededor de 0,364 segundos (0,17 + 0,19 + 0,004) de tiempo de ejecución del monitor y su intervalo de medida son 5 segundos en este caso.

La sobrecarga equivale a tiempo de ejecución del monitor entre el intervalo de medida

$$\text{Sobrecarga} = \frac{\text{Tiempo de ejecución del monitor}}{\text{Intervalo de medida}} = \frac{0,364}{5} = 0,0728 * 100 = 7,28 \%$$



## PREGUNTA VOLUNTARIA

Expresar la fórmula de la sobrecarga de la monitorización cuando el monitor es basado en 'detección de eventos'.

La sobrecarga en un monitor software cuando la ejecución de las instrucciones del monitor se lleva a cabo en el procesador del sistema monitorizado, se obtiene de la siguiente forma:

$$\text{Sobrecarga Software} = \frac{n * \text{Tiempo de ejecución del monitor}}{n * \text{Intervalo de medida}} = \frac{\text{Tiempo de ejecución del monitor}}{\text{Intervalo de medida}} \cdot \frac{n * \text{Tiempo de ejecución del monitor}}{\text{Tiempo total}}$$

Pero esto es en el caso donde el tiempo de ejecución del monitor es en un intervalo de medida constante. En cambio en el monitor basado en 'detección de eventos' tiene un intervalo irregular, aunque sigue teniendo siempre el mismo tiempo de ejecución del monitor al igual que en el caso de los monitores software, por tanto podemos adaptar la definición anterior de la siguiente manera:

$$\text{Sobrecarga Eventos} = \frac{n * \text{Tiempo de ejecución del monitor}}{\text{Tiempo total}}$$

