

Universidad Carlos III de Madrid  
Escuela Politécnica Superior  
Máster en Ciencia y Tecnología Informática  
Diseño de Sistemas Distribuidos

**uc3m**

---

Universidad  
**Carlos III**  
de Madrid

Práctica

# **Introducción a la simulación de sistemas distribuidos**

Autor:

Javier Prieto Cepeda, 100307011  
Gustavo Adolfo Chaves, 100372928

Leganés, Madrid, España  
Enero 2018

# Índice general

|   |                        |   |
|---|------------------------|---|
| 1 | Introducción . . . . . | 4 |
| 2 | Parte 1 . . . . .      | 4 |
|   | 2.1 mm1 . . . . .      | 4 |
|   | 2.2 mm2 . . . . .      | 5 |
| 3 | Parte 2: mm3 . . . . . | 6 |
| 4 | Conclusiones . . . . . | 9 |

# Índice de figuras

|   |  |    |
|---|--|----|
| 1 | Plataforma mm1. . . . .                  | 4  |
| 2 | Plataforma mm2. . . . .                  | 5  |
| 3 | Plataforma mm3. . . . .                  | 6  |
| 4 | Comparativa tiempos de servicio. . . . . | 9  |
| 5 | Comparativa tamaños de cola. . . . .     | 10 |

# 1 Introducción

El presente documento corresponde con la memoria asociada a la práctica de la asignatura Diseño de Sistemas Distribuidos del Máster en Ciencia y Tecnología Informática de la Universidad Carlos III de Madrid.

En esta práctica se pretende realizar una primera aproximación a la simulación de sistemas distribuidos utilizando SimGrid. SimGrid es un toolkit que proporciona mecanismos para evaluar algoritmos y heurísticas cluster, grid y P2P. El núcleo de simulación de SimGrid implementa y proporciona interfaces para simular un gran número de modelos diferentes, que pueden ser usados para simular diversos tipos de recursos, tanto computacionales como de redes.

La estructura del documento es la siguiente: En la Sección 2 se explica el diseño de los programas mm1 y mm2, mientras que la Sección 3 corresponde con la descripción de la implementación realizada del programa mm3 y el análisis de los resultados obtenidos. Por último, la Sección 4 finaliza el documento exponiendo las conclusiones obtenidas. Todas las pruebas presentadas en este documento han sido realizadas utilizando una máquina con un procesador Intel(R) Core(TM) i5-3570 CPU @ 3.40GHz con 4 cores.

## 2 Parte 1

Como código de apoyo se proporcionan dos programas implementados en el lenguaje de programación C y que utilizan herramientas proporcionadas por SimGrid. Cada uno de estos programas, constan de un escenario diferente.

### 2.1 mm1

Este programa permite obtener el tiempo medio de respuesta y el tamaño medio de la cola para 10000 peticiones de servicio. La tasa de llegadas es de 5 peticiones por segundo y la tasa de servicio es de 6 peticiones por segundo. El objetivo de este primer programa es estudiar cómo modelar colas M/M/1 en simGrid. El escenario se basa en la plataforma mostrada en la Figura 1:

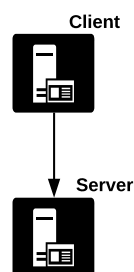


Figura 1: Plataforma mm1.

Esta plataforma está formada por dos máquinas donde:

- **ClientHost:** Máquina cliente de 1 GigaFlop de potencia. Ejecuta una aplicación que se encarga de realizar peticiones a la máquina servidor.
- **ServerHost:** Máquina servidor de 1 GigaFlop de potencia. Ejecuta una aplicación que se encarga de recibir las peticiones de la máquina cliente.
- **Red:** La red que conecta el cliente con el servidor tiene 1Gbps y 0 micro segundos de latencia

## 2.2 mm2

Este programa simula una cola M/M/1 de una manera distinta al programa mm1. En este caso en la máquina que hace de servidor (ServerHost-0) se modelan dos procesos: *queue* y *server*. El proceso cliente genera peticiones con un tiempo entre llegadas según una función de distribución exponencial. Estas peticiones llegan al proceso *queue* que las encola en una estructura de datos. El proceso servidor va extrayendo elementos de esta estructura y ejecutando las peticiones localmente. El escenario se basa en la plataforma mostrada en la Figura 2:

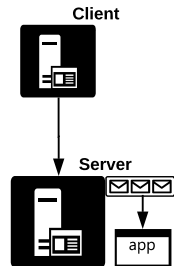


Figura 2: Plataforma mm2.

Esta plataforma está formada por dos máquinas donde:

- **ClientHost:** Máquina cliente de 1 GigaFlop de potencia. Ejecuta una aplicación que se encarga de realizar peticiones a la máquina servidor.
- **ServerHost:** Máquina servidor de 1 GigaFlop de potencia. Ejecuta dos aplicaciones. La primera, es la aplicación *queue* que se encarga de encolar las peticiones que recibe la máquina por parte del cliente. La segunda aplicación, *server*, se encarga de recoger de la cola las peticiones recibidas siguiendo la política FCFS (First Come-First Serve).
- **Red:** La red que conecta el cliente con el servidor tiene 1Gbps y 0 micro segundos de latencia

### 3 Parte 2: mm3

En esta segunda parte se pide modificar el fichero de plataforma para incluir 10 servidores, por lo que la plataforma presentada en el programa mm2 pasa a tener ser la plataforma mostrada en la Figura 3, donde cada servidor tiene las mismas características que el servidor de la parte 1.

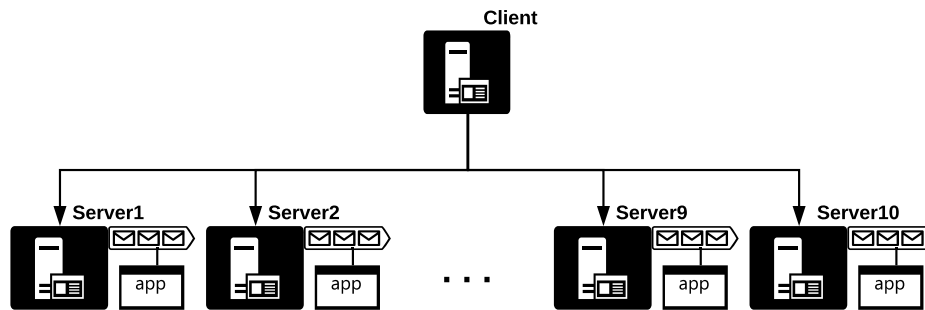


Figura 3: Plataforma mm3.

Además, también se pide modificar el proceso *client* de la máquina cliente para que el reparto de tareas a los servidores se realice mediante cuatro políticas diferentes:

- **Round-Robin:** Para la implementación del reparto Round-Robin, simplemente hemos hecho que el reparto se realice de forma cíclica, es decir, primero al servidor 0, después al 1, etc, hasta llegar al servidor número 9 y después volver al 0 para continuar con el reparto.
- **Random:** Para la implementación del reparto Random, hemos utilizado la función *rand()*, de forma que aplicando el módulo con el número de servidores (10) obtenemos un servidor de forma aleatoria.
- **Smaller queue:** Para la implementación del reparto Smaller queue, hemos implementado una función que se encarga de recorrer las colas de los servidores, y elige el servidor que menos peticiones tiene encoladas para la asignación de la tarea.
- **Power of two choices:** Para la implementación del reparto Power of two choices, hemos implementado una función que se encarga de elegir de forma aleatoria dos servidores de los diez disponibles, y de los dos elige el servidor que tenga un menor número de tareas encoladas.

También se pide modificar el proceso *queue* de las máquinas servidores para que la ejecución de las tareas en los servidores se realice mediante dos políticas diferentes:

- **Primero el trabajo más corto:** Para la implementación de esta política de ejecución, simplemente hemos hecho que cuando la cola reciba una tarea, la inserte al final y posteriormente ordene la cola de forma ascendente en tiempo de ejecución haciendo uso de la función proporcionada *short\_function()*.

- **Primero el trabajo más largo:** Para la implementación de esta política de ejecución, simplemente hemos hecho que cuando la cola reciba una tarea, la inserte al final y posteriormente ordene la cola de forma descendente en tiempo de ejecución haciendo uso de la función *short\_function\_inverse()* que hemos implementado.

Una vez hecho esto, el enunciado nos pide indicar el tiempo medio de servicio y el tamaño medio de las colas en los servidores para la tasa media de llegadas  $T$  definida inicialmente (5 peticiones por segundo) y para una tasa media de llegadas diez veces mayor (50 peticiones por segundo) para un intervalo de confianza del 95 % con cada una de las cuatro políticas de reparto y además, con cada una de las políticas de ejecución en los servidores. Las Tablas 1, 2, 3, 4 muestran estos resultados, donde para cada política se han tenido en cuenta 10 muestras.

Para cada configuración, se han realizado 10 simulaciones, y los resultados presentados (Tablas 1, 2) indican el tiempo medio de servicio por tarea, el valor mínimo, el valor máximo, la media de las 10 simulaciones, la desviación estándar, el error, y el intervalo de confianza del 95 % (como  $n < 30$ , se ha utilizado una distribución  $t$  de Student).

Tabla 1: Tiempo medio de servicio para una tasa media de llegadas de 5.

| Política                | Mínimo   | Máximo   | Media    | Desv. estándar | Error    | Intervalo (95 %)    |
|-------------------------|----------|----------|----------|----------------|----------|---------------------|
| Round-Robin-FS          | 0,168532 | 0,168932 | 0,168837 | 0,000122563    | 0,000076 | (0,168761-0,168913) |
| Round-Robin-FL          | 0,168267 | 0,168736 | 0,168474 | 0,000195       | 0,000121 | (0,168353-0,168594) |
| Random-FS               | 0,181345 | 0,181932 | 0,181649 | 0,000261107    | 0,000162 | (0,181487-0,181811) |
| Random-FL               | 0,181876 | 0,182357 | 0,182117 | 0,000254       | 0,000157 | (0,181959-0,182274) |
| Smaller queue-FS        | 0,082351 | 0,082781 | 0,082611 | 0,000189111    | 0,000117 | (0,082494-0,082728) |
| Smaller queue-FL        | 0,090634 | 0,090931 | 0,090783 | 0,000156       | 0,000097 | (0,090686-0,090880) |
| Power of two choices-FS | 0,181148 | 0,181982 | 0,181567 | 0,000340493    | 0,000211 | (0,181356-0,181778) |
| Power of two choices-FL | 0,191589 | 0,191871 | 0,191708 | 0,000123       | 0,000076 | (0,191632-0,191784) |

Tabla 2: Tiempo medio de servicio para una tasa media de llegadas de 50.

| Política                | Mínimo   | Máximo   | Media    | Desv. estándar | Error    | Intervalo (95 %)     |
|-------------------------|----------|----------|----------|----------------|----------|----------------------|
| Round-Robin-FS          | 0,399663 | 0,399973 | 0,399733 | 0,000106958    | 0,000066 | (0,3996671-0,399799) |
| Round-Robin-FL          | 1,203923 | 1,204252 | 1,204085 | 0,000148       | 0,000091 | (1,203994-1,204176)  |
| Random-FS               | 0,533176 | 0,533425 | 0,533266 | 0,000119155    | 0,000074 | (0,533192-0,533340)  |
| Random-FL               | 2,406114 | 2,406883 | 2,406432 | 0,000326       | 0,000202 | (2,406230-2,406633)  |
| Smaller queue-FS        | 0,376112 | 0,376551 | 0,376293 | 0,000186159    | 0,000115 | (0,376177-0,376408)  |
| Smaller queue-FL        | 0,459264 | 0,459747 | 0,459561 | 0,000214       | 0,000132 | (0,459429-0,459694)  |
| Power of two choices-FS | 0,352331 | 0,352778 | 0,352522 | 0,000195088    | 0,000121 | (0,352401-0,352643)  |
| Power of two choices-FL | 0,545231 | 0,545632 | 0,545440 | 0,000185       | 0,000115 | (0,545326-0,545555)  |

Como se puede ver, los peores resultados se obtienen en aquellas políticas que tienen componentes

de aleatoriedad (Random-FL, Power of two choices), lo cual tiene sentido puesto que no se realiza una elección objetiva del servidor al cual enviar las tareas. Además, se puede ver como la política de ejecución *Primero tarea más larga* tiene peor comportamiento que la política *Primero tarea más corta*, lo que provoca que la tasa de servicio sea mayor al ejecutar aquellas tareas que requieren un mayor tiempo de ejecución, provocando posibles cuellos de botella.

Además para cada configuración se presentan los resultados (Tablas 4, 3) que indican el tamaño medio de las colas, el valor mínimo, el valor máximo, la media de las 10 simulaciones, la desviación estándar, el error, y el intervalo de confianza del 95 % (como  $n < 30$ , se ha utilizado una distribución t de Student).

Tabla 3: Tamaño medio de las colas para una tasa media de llegadas de 5.

| Política                | Mínimo   | Máximo   | Media    | Desv. estándar | Error    | Intervalo (95 %)    |
|-------------------------|----------|----------|----------|----------------|----------|---------------------|
| Round-Robin-FS          | 0        | 0        | 0        | 0              | 0        | (0-0)               |
| Round-Robin-FL          | 0        | 0        | 0        | 0              | 0        | (0-0)               |
| Random-FS               | 0,003774 | 0,008815 | 0,006295 | 0,002656624    | 0,001647 | (0,004648-0,007941) |
| Random-FL               | 0,005928 | 0,008132 | 0,007030 | 0,001162       | 0,000720 | (0,006310-0,007750) |
| Smaller queue-FS        | 0,022374 | 0,034722 | 0,028548 | 0,006507904    | 0,004034 | (0,024515-0,032582) |
| Smaller queue-FL        | 0,028543 | 0,028643 | 0,028593 | 0,000053       | 0,000033 | (0,028560-0,028626) |
| Power of two choices-FS | 0,004130 | 0,004975 | 0,004552 | 0,000445091    | 0,000276 | (0,004276-0,004828) |
| Power of two choices-FL | 0,003023 | 0,003201 | 0,003112 | 0,000094       | 0,000058 | (0,003054-0,003170) |

Tabla 4: Tamaño medio de las colas para una tasa media de llegadas de 50.

| Política                | Mínimo    | Máximo    | Media     | Desv. estándar | Error    | Intervalo (95 %)      |
|-------------------------|-----------|-----------|-----------|----------------|----------|-----------------------|
| Round-Robin-FS          | 0,860321  | 0,860891  | 0,860633  | 0,00024979     | 0,000155 | (0,860478-0,860788)   |
| Round-Robin-FL          | 4,992347  | 4,998340  | 4,994790  | 0,003056       | 0,001894 | (4,992896-4,996684)   |
| Random-FS               | 1,819175  | 1,819765  | 1,819465  | 0,000268589    | 0,000166 | (1,8192984-1,819631)  |
| Random-FL               | 10,880123 | 10,880889 | 10,881463 | 0,000561       | 0,000348 | (10,880542-10,881237) |
| Smaller queue-FS        | 0,442398  | 0,442851  | 0,442607  | 0,000199109    | 0,000123 | (0,442483-0,442730)   |
| Smaller queue-FL        | 0,813267  | 0,813872  | 0,813617  | 0,000264       | 0,000164 | (0,813453-0,813780)   |
| Power of two choices-FS | 0,530571  | 0,530909  | 0,530694  | 0,000151655    | 0,000094 | (0,530600-0,530788)   |
| Power of two choices-FL | 1,230237  | 1,230578  | 1,230441  | 0,000149       | 0,000093 | (1,230348-1,230533)   |

Como se puede ver, los resultados obtenidos para una tasa de llegadas de 5 tareas por segundo no son concluyentes, de forma que para la política de reparto *Round-Robin* las colas nunca llegan a acumular más de una tarea. En cambio, para una tasa de llegadas de 50 tareas por segundo, se puede observar como las configuraciones que tienen aleatoriedad (Random, Power of two choices) acumulan un mayor número de tareas en las colas. En la Tabla 2 se puede ver como estos valores se corresponden con una mayor tasa de servicio por tarea, lo cual tiene sentido ya que no se optimiza el reparto de tareas en



los servidores en función de su carga. Otro factor que provoca un mayor tamaño de tareas encoladas, se puede ver en la política de ejecución FS (Primero tarea más larga), ya que al ejecutar con mayor prioridad aquellas tareas que requieren un mayor tiempo de ejecución se produce un cuello de botella que provoca el crecimiento de las colas.

Además, para apoyar estos resultados se han realizado simulaciones variando el número de tareas desde 100 hasta 10.000.000, de forma que se pueda observar el impacto que tiene un mayor número de tareas en el sistema tanto en la tasa media de servicio como en el tamaño medio de las colas.

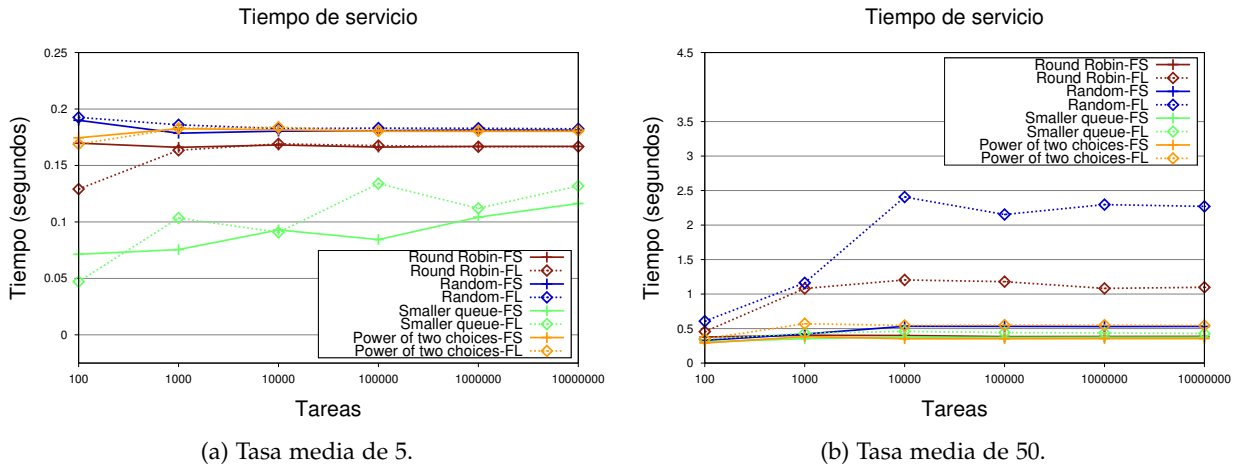


Figura 4: Comparativa tiempos de servicio.

Como podemos ver en las Figuras 4a y 4b, los peores resultados y la peor escalabilidad del sistema recae en las configuraciones que tienen la política de ejecución FL, tal y como se ha explicado anteriormente. Además, el peor comportamiento se puede ver en la configuración Random-FL, la cual además de tener la política de ejecución FL tiene una componente aleatoria en la elección de los servidores que provoca posibles cuellos de botella. Para una tasa de llegadas de 5 tareas por segundo, los resultados no son concluyentes. También hay que destacar, como la política de asignación de tareas a servidores *Power of two choices* logra mejorar en gran medida el rendimiento de la política *Random*, ya que al seleccionar de los dos servidores obtenidos aleatoriamente aquel que tiene la cola más pequeña, se reduce la carga y cuellos de botella del sistema.

En cuanto al tamaño de las colas, en las Figuras 5a y 5b se puede ver como una vez más, el peor comportamiento se obtiene en las configuraciones que tienen la política de ejecución FL y componentes aleatorias en la asignación de servidores a las tareas, provocando cuellos de botella.

## 4 Conclusiones

Esta práctica ha servido para familiarizarnos con la simulación de sistemas distribuidos y para entender el comportamiento de diferentes procesos que ejecutan en servidores. Personalmente, uno de los

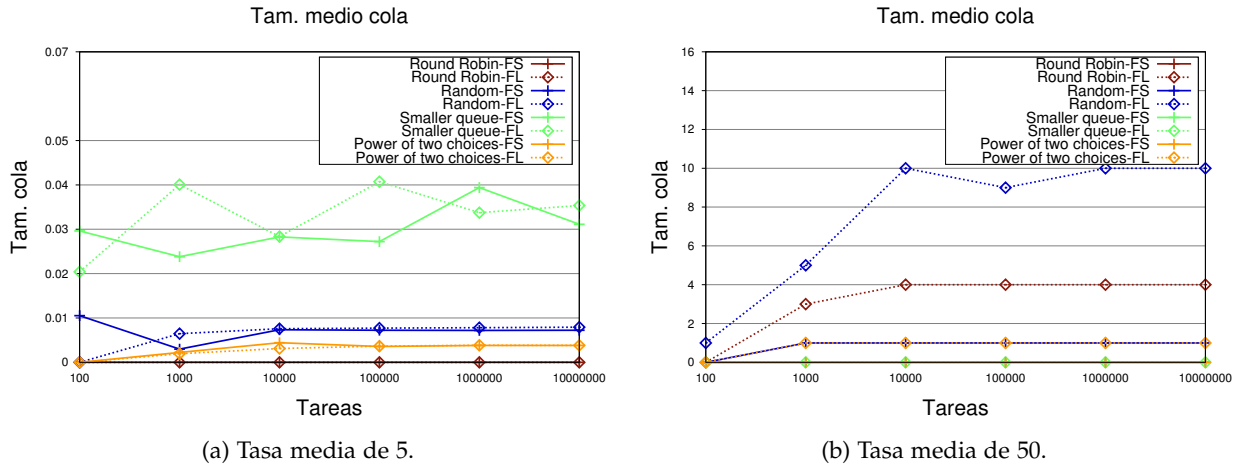


Figura 5: Comparativa tamaños de cola.

integrantes del grupo tenía experiencia con el desarrollo de aplicaciones distribuidas y concurrentes y el otro no. Sin embargo, ambos hemos trabajado conjuntamente y hemos complementado los conocimientos a la perfección.

Por otro lado, de la práctica destacamos la comparativa entre políticas con componentes aleatorias y políticas más eficientes como Round-Robin o Smaller queue. También resulta llamativo como la política de ejecución *Primero tarea más larga* provoca cuellos de botella en el sistema, lo cual resulta coherente y lógico puesto que la tasa de servicio aumenta y las peticiones nuevas se deben ir encolando puesto que el servidor se encuentra ocupado. Sin embargo, no se ha podido ver en mayor profundidad las diferencias entre el resto de configuraciones debido a que la simulación no era demasiado compleja.