

Sistemas Distribuidos

**PR 2 – Report for Phase 4.2**



Raúl Pastor Clemente

[*rpastorc@uoc.edu*](mailto:rpastorc@uoc.edu)

Group ID 309571699

Contenido

[Introducción 2](#_Toc40449833)

[Fase 4.2 3](#_Toc40449834)

[Pruebas locales 3](#_Toc40449835)

[Prueba en entorno con máquinas virtuales 4](#_Toc40449836)

[Conclusiones de las pruebas 6](#_Toc40449837)

# Introducción

Una vez conseguido que la fase 4 funcione, he realizado algunas pruebas utilizando diversas combinaciones de aquellos parámetros sobre los que tenemos control para comprobar cómo afectan al desarrollo del experimento.

Para ello, he utilizado el mismo setup que en el resto de la práctica: Eclipse y el plugin C/C++ Remote Launcher.

Se ha creado un nuevo grupo de lanzamiento que contiene 10 ejecuciones de la fase 4. Una vez se inicia el grupo, estas ejecuciones se realizan de forma secuencial, una tras otra.

Al terminar cada batería de ejecuciones, se comprueban los resultados almacenados en la carpeta “/results” y se cambian los parámetros contenidos en el fichero “config.properties” antes de realizar la siguiente tanta de pruebas.

Los resultados de cada prueba se pueden encontrar junto a este informe para su consulta.

|  |
| --- |
|  |

# Fase 4.2

Para esta fase, he decidido crear cinco grupos distintos de configuraciones (desde el archivo *config.properties*) y ejecutarlos con distintos números de nodos para ver cómo afectan al funcionamiento del protocolo TSAE.

Las pruebas se han realizado en ejecuciones con 5, 10 y 15 nodos. En total se han realizado **150 pruebas**, que han tardado cerca de **20 horas** en completarse.

Los resultados de las pruebas para cada configuración pueden encontrarse junto a este documento.

Por desgracia, no he podido realizar pruebas en un entorno distribuido (con varias máquinas virtuales) porque carecer de la infraestructura necesaria para llevarlas a cabo.

## Pruebas locales

**Pruebas realizadas con 5 nodos**

|  |  |
| --- | --- |
| Configuración | Resultado (éxito/total) |
| a) Conf. defecto | 8/10 |
| b) Conf. defecto + probReconnect=0.5 | 10/10 |
| c) Conf. defecto + probReconnect=1.0 | 10/10 |
| d) Conf. Defecto + executionStop=360 | 9/10 |
| e) Conf. Defecto + executionStop=360 + probDisconnect=0.0 | 10/10 |

**Pruebas realizadas con 10 nodos**

|  |  |
| --- | --- |
| Configuración | Resultado (éxito/total) |
| f) Conf. defecto | 2/10 |
| g) Conf. defecto + probReconnect=0.5 | 4/10 |
| h) Conf. defecto + probReconnect=1.0 | 10/10 |
| i) Conf. Defecto + executionStop=360 | 4/10 |
| j) Conf. Defecto + executionStop=360 + probDisconnect=0.0 | 8/10 |

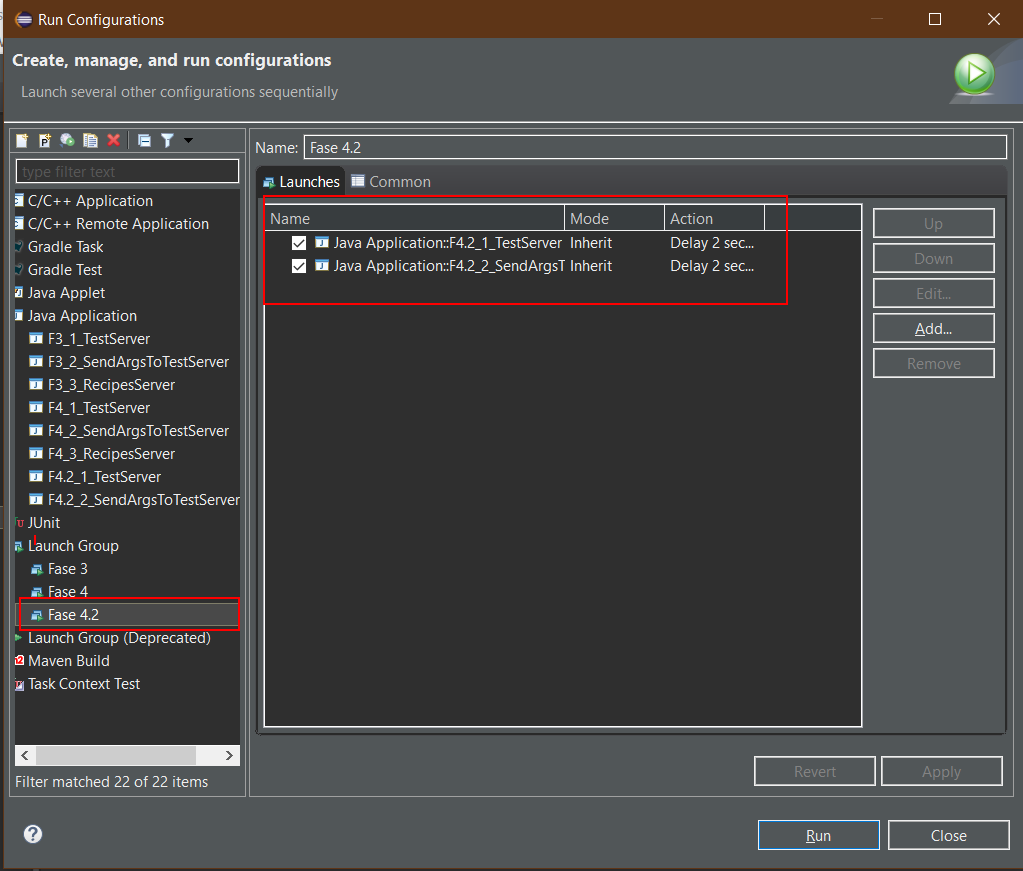
**Pruebas realizadas con 15 nodos**

|  |  |
| --- | --- |
| Configuración | Resultado (éxito/total) |
| k) Conf. defecto | 2/10 |
| l) Conf. defecto + probReconnect=0.5 | 1/10 |
| m) Conf. defecto + probReconnect=1.0 | 7/10 |
| n) Conf. Defecto + executionStop=360 | 0/10 |
| o) Conf. Defecto + executionStop=360 + probDisconnect=0.0 | 10/10 |

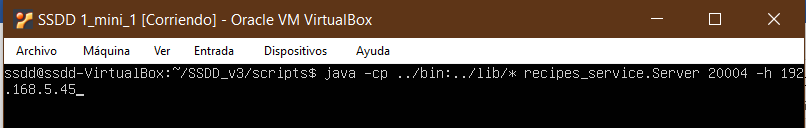
## Prueba en entorno con máquinas virtuales

Para realizar esta prueba he utilizado un host físico y cinco máquinas virtuales Ubuntu 18.04 sin entorno gráfico, funcionando bajo VirtualBox 6.

En el host lanzamos el TestServer con la siguiente configuración:

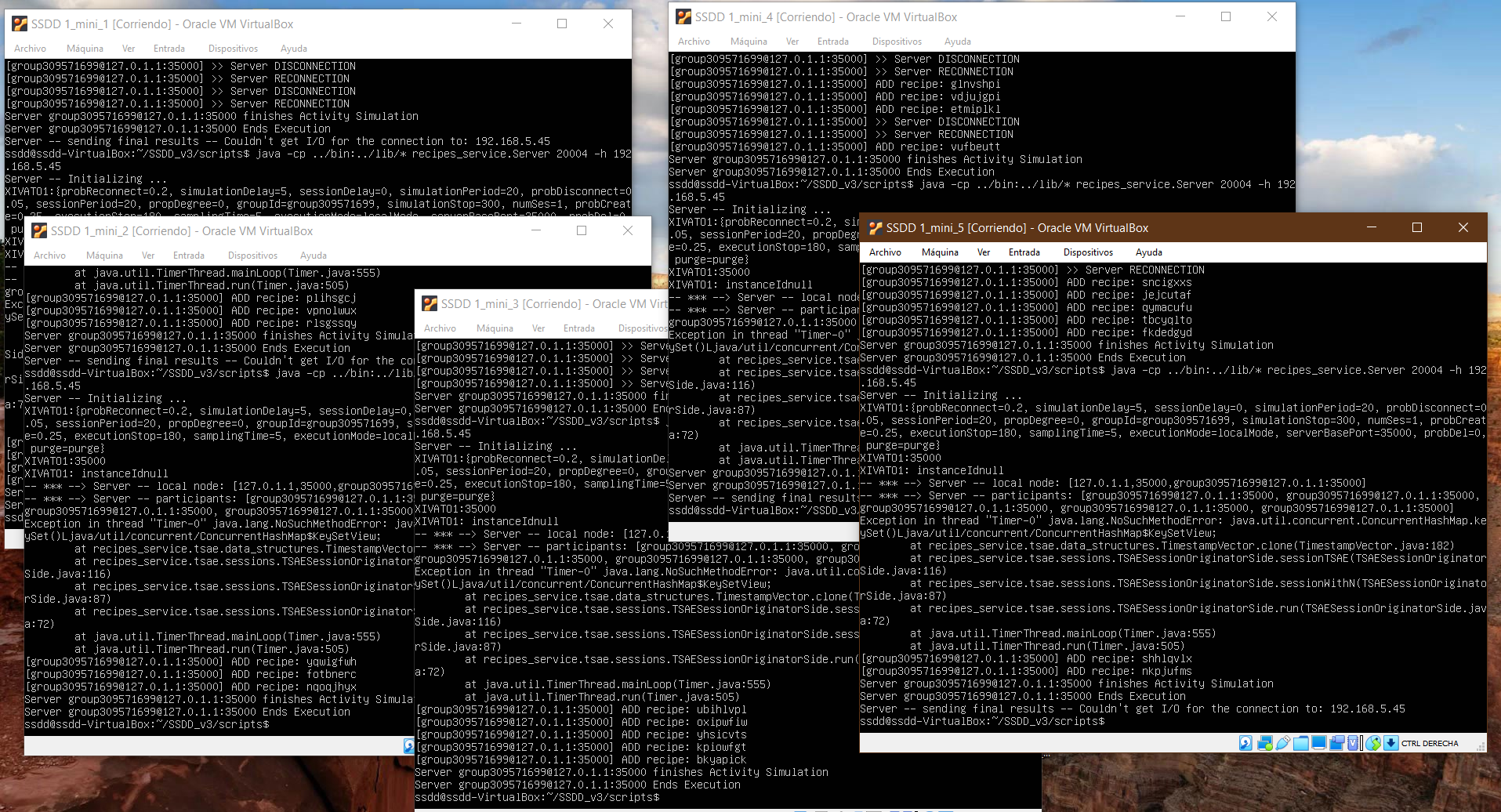


Mientras que en las máquinas virtuales lanzamos el proyecto con el siguiente comando:



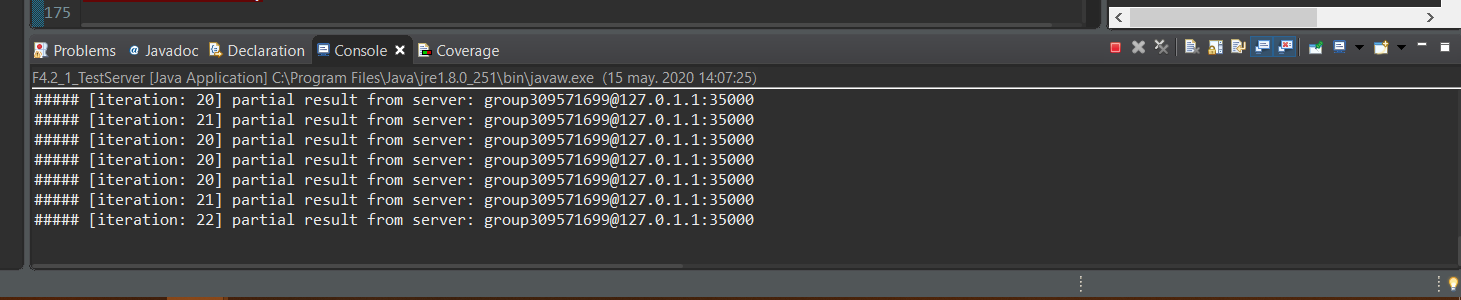
Eso sí, en “*config.properties*” he tenido que dejar el parámetro “*exucutionMode*” en “*localMode*”, porque si lo cambio a “*remoteMode*” obtengo una excepción de I/O en los nodos participantes.

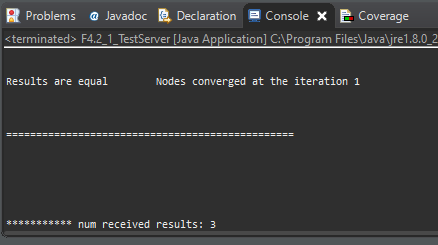
Sin embargo, el resultado es que el experimento funciona exactamente igual que en local. El TestServer envía a los nodos la configuración y la lista de participantes al principio de la ejecución. Acto seguido, los participantes comienzan a crear y eliminar recetas y establecer sesiones TSAE entre ellos.



La excepción está causada por haber compilado con Java 8 y después ejecutado en Java 7. Se trata de un problema documentado y que no afecta al funcionamiento del programa, por lo que se puede ignorar.

Finalmente, tras terminar el tiempo de simulación de actividad, los participantes comienzan a enviar los resultados al servidor.





## Conclusiones de las pruebas

El protocolo TSAE es un protocolo de consistencia eventual, lo que significa que se alcanza un estado consistente entre los nodos una vez que se han hecho una cantidad de sesiones anti-entropía que oscila entre un mínimo de (*n*-2) sesiones y un máximo de (*n*-1)+(*n*-2)+…+(*n*-(*n*-1)) sesiones, siendo *n* el número de nodos participantes, tal y como ya vimos en la última pregunta teórica de la PR1.

En la implementación de TSAE que se utiliza en la práctica no tenemos ningún control sobre el funcionamiento del protocolo, a excepción de los parámetros contenidos en el fichero “*config.properties*”. Los nodos simulan conexiones y desconexiones según los parámetros de este archivo.

Un punto crítico de la implementación es que el tiempo del que disponen los nodos para alcanzar un estado consistente, que inicialmente está limitado a 180 segundos, por lo que puede suceder que no se pueda alcanzar un estado consistente entre los nodos.

Otro punto crítico está definido por la frecuencia con la que los nodos se desconectan y se vuelven a conectar. Modificando los parámetros para reducir la posibilidad de que los nodos se desconecten podemos observar una notable variación la tasa de éxito.

Si aumentamos el tiempo del que disponen los nodos para sincronizarse, vemos que la tasa de éxito generalmente aumenta, puesto que los nodos disponen de más tiempo para sincronizarse.

Asimismo, la cantidad de nodos que participan también resulta ser un factor crítico, tal y como puede observarse en los resultados de las tablas según el número de participantes. A más nodos participantes más difícil es que todos estén sincronizados al finalizar la sesión.

Por el contrario, la tasa de éxito disminuye al aumentar la cantidad de nodos que participan, ya que al aumentar la variable n aumenta la cantidad de sesiones anti-entropía necesarias para lograr alcanzar un estado consistente.

En los resultados que acompañan a este informa puede observarse que en los casos de fallo los nodos conflictivos suelen estar unas pocas actualizaciones por detrás del resto.

**Concluimos pues que el crítico para el protocolo TSAE consiste en el número de nodos que se reconectan, puesto que la información que éstos contienen al volver a la simulación está desactualizada, y este hecho puede dar lugar a conflictos con las operaciones y a problemas a la hora de sincronizar, ya que serán necesarias más sesiones TSAE para que el estado de los nodos sea consistente.**

**Esta conclusión coincide plenamente con la del paper “*The Performance of Weak-consistency Replication Protocols*” de Golding y Long, donde determinan que la principal desventaja de este tipo de protocolos radica en que los nodos o réplicas pueden transmitir información desactualizada, lo que según sus pruebas causa retrasos en la propagación de los datos.**