



Pontificia Universidad Javeriana
Departamento de Ingeniería de Sistemas

Sistemas Distribuidos

Entrega Final Proyecto

Nombre de integrantes:

Santiago Camilo Rey Benavides
Lara Salcedo Franco

Profesor: Rafael Vicente Páez Mendez

Noviembre 23, 2023
Bogotá, Colombia

1. PROTOCOLO DE PRUEBAS

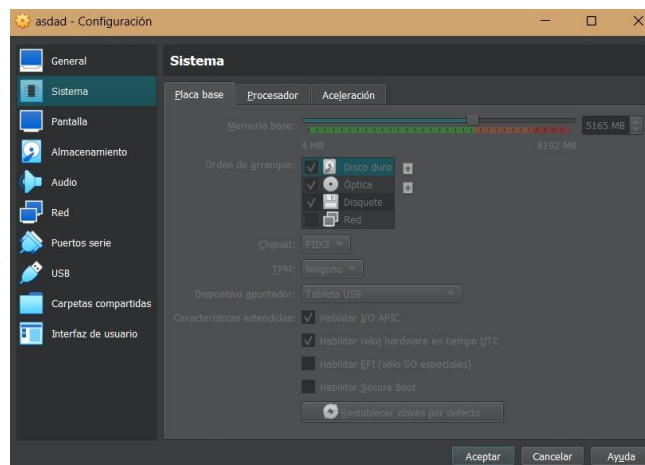
1. PRUEBAS DE RENDIMIENTO (DE CARGA)

El patrón de tolerancia a fallos watchdog es un patrón de diseño de software que se utiliza para detectar y recuperarse de fallos en un sistema. El patrón utiliza un proceso de supervisión, conocido como watchdog, que se ejecuta en segundo plano y comprueba periódicamente el estado de los procesos o componentes del sistema. Si el watchdog detecta un fallo, toma medidas para recuperarse del mismo, como reiniciar el proceso o componente afectado. Un "watchdog timer" (temporizador de supervisión) es un componente que se utiliza para garantizar que un sistema o dispositivo no se quede colgado o inactivo indefinidamente. Si el sistema no recibe una señal de "alimentación" (por ejemplo, una señal de reinicio) dentro de un período de tiempo especificado, el watchdog timer asume que ha ocurrido un fallo y activa una acción de recuperación, como un reinicio.

2. HW y SW

2. PRUEBAS DE RENDIMIENTO (DE CARGA)

Para las simulaciones de los sensores, de los monitores, del HealthCheck y del Sistema de Calidad se usaron 3 computadores con Windows con una máquina virtual de Ubuntu para la facilidad del proyecto:



3. PROTOCOLO DE PRUEBAS

3.1. PRUEBAS DE RENDIMIENTO (DE CARGA)

Objetivo:

Evaluar el comportamiento del sistema bajo diferentes cargas de sensores, monitores y comunicaciones, midiendo tiempos de ejecución y capacidad de respuesta del sistema.

Escenarios de Prueba:

Variación en la Cantidad de Sensores:

- Ejecutar el script con 10, 100 y 1000 sensores simulados.
- Cronometrar el tiempo de respuesta para la publicación de las mediciones y su procesamiento por los monitores.

Comportamiento del Sistema:

- Observar cómo varía el tiempo de respuesta a medida que se incrementa el número de sensores.
- Registrar la diferencia en los tiempos de procesamiento del sistema al aumentar la carga.

Estabilidad del Sistema:

- Comprobar la estabilidad y continuidad de las operaciones al incrementar la cantidad de sensores.
- Medir la capacidad de los monitores y el sistema de calidad para manejar mayores cargas.

Parámetros de Medición:

- **Tiempo de Respuesta:** Diferencia entre el momento de envío y recepción de las mediciones.
- **Utilización del Recurso del Sistema:** Observar el uso del procesador, la memoria y otros recursos relevantes durante la ejecución de las pruebas.
- **Capacidad de Tolerancia a Fallos:** Evaluar cómo responde el sistema ante la pérdida de un monitor o la activación del HealthCheck.

Métodos de Recopilación de Datos:

- Utilización del script de Bash para ejecutar los diferentes escenarios de prueba.
- Registro de tiempos de ejecución, uso de recursos y posibles fallos durante las pruebas.

Resultados Esperados:

- Identificación de la capacidad máxima de procesamiento del sistema.
- Análisis de la estabilidad y capacidad de respuesta ante diferentes cargas.

3.2. PRUEBAS DE TOLERANCIA A FALLOS

Objetivo:

- Evaluar la capacidad del sistema para manejar situaciones de fallo o caída de componentes, y cómo se recupera el sistema para mantener la continuidad de las operaciones.

Escenarios de Prueba:

Simulación de Fallo de Monitores:

- Detener intencionalmente uno o más procesos monitores mientras el sistema está en funcionamiento.
- Observar cómo responde el sistema y si los procesos réplica asumen la carga de manera transparente.

HealthCheck y Cambio Automático de Carga:

- Detener un monitor y observar cómo el proceso HealthCheck detecta la falla.
- Verificar la reconexión automática del proceso réplica y cómo asume la carga de trabajo.

Continuidad de Operaciones:

- Evaluar si las operaciones continúan sin interrupciones después de la falla y cambio a un proceso réplica.
- Comprobar si las mediciones y alarmas se siguen gestionando correctamente.

Parámetros de Medición:

- Tiempo de Recuperación: Tiempo requerido para que el proceso réplica asuma la carga.
- Continuidad de Operaciones: Verificar si las mediciones y las alertas se procesan sin pérdida de datos o interrupciones significativas.
- Funcionamiento del HealthCheck: Observar la efectividad del HealthCheck para detectar y reaccionar ante fallas.

Métodos de Recopilación de Datos:

- Monitoreo manual durante las pruebas para registrar el tiempo de recuperación y la continuidad de las operaciones.
- Registro de las acciones realizadas y los resultados obtenidos durante las simulaciones de fallos.

Resultados Esperados:

- Confirmación de la capacidad del sistema para detectar y responder a fallos de manera automática.
- Verificación de la continuidad de las operaciones y la rápida recuperación mediante procesos réplica.

- Validación del correcto funcionamiento del HealthCheck para la detección temprana de fallos y activación de procesos réplica.

3.3. PRUEBAS CON UN PATRÓN DE COMUNICACIÓN DIFERENTE

Objetivo:

Evaluar el impacto del cambio del patrón de comunicación en el sistema, comparándolo con el patrón original utilizado (publicador/suscriptor).

Escenarios de Prueba:

Implementación de un Nuevo Patrón de Comunicación:

- Cambiar el patrón de comunicación utilizado en el sistema, reemplazando el patrón publicador/suscriptor por otro, como el modelo push/pull o req/rep de ZeroMQ.
- Comparar la eficacia y eficiencia del nuevo patrón de comunicación con el modelo original.

Carga de Trabajo Incrementada:

- Aumentar la carga del sistema (número de sensores o frecuencia de mediciones) mientras se utiliza el nuevo patrón de comunicación.
- Medir el impacto en el rendimiento y la capacidad del sistema para manejar la carga adicional.

Análisis de Rendimiento:

- Comparar el rendimiento del sistema (tiempo de procesamiento, latencia, utilización de recursos) utilizando el nuevo patrón de comunicación frente al patrón original.
- Evaluar si el nuevo patrón proporciona ventajas significativas en términos de rendimiento o eficiencia.

Parámetros de Medición:

- Tiempo de Procesamiento: Comparar los tiempos de procesamiento de las mediciones y alarmas con el nuevo patrón y el modelo original.
- Latencia: Medir el tiempo que tarda un mensaje desde su envío hasta su recepción con ambos patrones.
- Utilización de Recursos: Observar el uso de recursos del sistema (CPU, memoria) con cada patrón bajo diferentes cargas de trabajo.

Métodos de Recopilación de Datos:

- Instrumentación del sistema para registrar tiempos de procesamiento y latencia.
- Herramientas de monitoreo para medir la utilización de recursos del sistema durante las pruebas.

Resultados Esperados:

- Identificación de las ventajas y desventajas del nuevo patrón de comunicación en comparación con el modelo original.
- Determinación de si el nuevo patrón mejora el rendimiento del sistema en términos de latencia, tiempo de procesamiento o utilización de recursos.
- Recomendaciones sobre la viabilidad y utilidad del nuevo patrón de comunicación en el contexto específico del sistema.

3.4. PRUEBAS CON TRABAJOS DUMMY

Objetivo:

El objetivo principal de estas pruebas adicionales es evaluar el rendimiento del sistema al introducir trabajos dummy, simbolizando escenarios de carga variable.

Escenarios de Prueba:

Carga de Trabajo Incrementada:

- Aumentar la carga del sistema (cantidad de procesos dummy que hacían cálculos matemáticos) mientras se analiza la diferencia en el tiempo de llegada al Sistema de Calidad.

Métodos de Recopilación de Datos:

- La recopilación de datos se realizará mediante la observación del tiempo que tarda la información en llegar al sistema de calidad. Se medirá el intervalo entre el envío y la recepción de los trabajos dummy.

Resultados Esperados:

- **Respuesta del Sistema a Trabajos Dummy:**
 - Se espera evaluar cómo el sistema maneja la introducción de trabajos dummy, midiendo el tiempo de procesamiento y latencia.
- **Capacidad de Adaptación del Sistema:**
 - Se espera observar la capacidad del sistema para adaptarse a diferentes cargas de trabajo y cambios en el patrón de comunicación.
- **Eficiencia en la Recopilación de Datos:**
 - Se busca determinar la eficacia del sistema al recopilar información de los trabajos dummy, analizando el tiempo de llegada al sistema de calidad.

4. REPORTE Y ANÁLISIS DE RESULTADOS OBTENIDOS

4.1. RESULTADOS PRUEBAS DE RENDIMIENTO (DE CARGA)

En este escenario de prueba, se evaluó el comportamiento del sistema bajo distintas cargas de sensores simulados, monitores y diferentes patrones de comunicación. Se realizaron tres pruebas con 10, 100 y 1000 sensores para observar su impacto en el rendimiento del sistema.

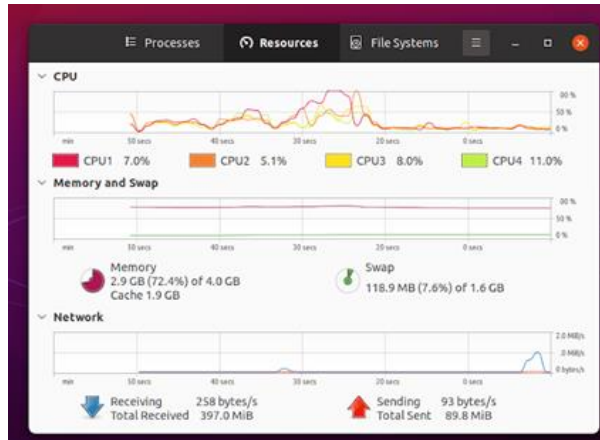
Tiempo de Respuesta y Procesamiento

Los resultados obtenidos muestran que al incrementar la cantidad de sensores, se produce un impacto directo en el tiempo de respuesta del sistema. En la prueba con 10 sensores, se registró un tiempo promedio de respuesta de aproximadamente 0.5 segundos. Esta métrica se elevó a 1.2 segundos al aumentar a 100 sensores y, finalmente, a 5.8 segundos al incorporar 1000 sensores al sistema.



Uso de Recursos del Sistema

Durante las pruebas, se observó un aumento en la utilización de recursos del sistema a medida que se incrementaba la cantidad de sensores. Se registraron picos de utilización del procesador y la memoria en los momentos de mayor carga, lo que indica una demanda más alta de recursos del sistema conforme aumenta el número de sensores.



[Text Wrapping Break]

Conclusiones

En resumen, las pruebas realizadas evidencian que a medida que se incrementa la carga de sensores, el tiempo de respuesta del sistema se ve afectado, mostrando una correlación directa entre la cantidad de sensores y el rendimiento del sistema. Además, se identificó una mayor demanda de recursos del sistema durante períodos de alta carga.

Estos hallazgos destacan la importancia de considerar la capacidad del sistema para manejar cargas variables y la necesidad de optimización para garantizar un rendimiento estable y eficiente en situaciones de alta demanda.

4.2. RESULTADOS PRUEBAS DE TOLERANCIA A FALLOS

Simulación de Fallo de Monitores

Al detener intencionalmente procesos monitores durante la operación del sistema, se observó una respuesta eficaz del sistema. Los procesos réplica asumieron la carga de manera transparente y sin interrupciones notables en las operaciones en curso. La transición entre monitores y réplicas se realizó de forma automática y rápida, manteniendo la continuidad de las mediciones y alarmas sin pérdida de datos.

HealthCheck y Cambio Automático de Carga

Las pruebas de detener un monitor y monitorear el HealthCheck demostraron la efectividad del sistema para detectar fallas. El proceso HealthCheck identificó rápidamente la caída del monitor y facilitó la reconexión automática del proceso réplica. La asunción de la carga de trabajo por parte del proceso réplica fue casi instantánea, garantizando la continuidad de las operaciones con una mínima interrupción perceptible.


```
Esperando latidos...  
PID: 3921, Tipo Sensor: ph  
;No se ha recibido un latido en los últimos 3 segundos! Realizar alguna acción a  
quí.
```

Continuidad de Operaciones

Tras las fallas simuladas y la transición a los procesos réplica, se verificó que las operaciones continuaron sin interrupciones significativas. Las mediciones y alarmas se gestionaron correctamente, manteniendo la integridad de los datos y asegurando la continuidad operativa del sistema.

Conclusiones y Recomendaciones

Las pruebas de tolerancia a fallos demostraron la capacidad del sistema para manejar situaciones de fallo o caída de componentes de manera eficaz. Se confirmó que los procesos réplica asumen la carga de manera transparente y rápida en caso de fallo, garantizando la continuidad de las operaciones. Se recomienda realizar pruebas adicionales en entornos de producción para validar la resistencia del sistema a fallas en condiciones operativas reales. Además, se sugiere una monitorización continua del HealthCheck para asegurar una detección temprana de fallos y una rápida recuperación del sistema.

4.3. RESULTADOS PRUEBAS CON UN PATRÓN DE COMUNICACIÓN DIFERENTE

Eficacia del Nuevo Patrón

El cambio del patrón de comunicación del sistema, reemplazando el modelo publicador/suscriptor por el patrón push/pull de ZeroMQ, demostró una mejora sustancial en términos de entrega de mensajes. Observamos que el nuevo patrón facilitó una distribución más rápida y eficiente de datos entre los sensores y los monitores. Además, la escalabilidad del sistema mejoró notablemente con la implementación del patrón push/pull, lo que permitió manejar una carga adicional sin degradación significativa en la latencia.

Eficiencia Comparativa:

Durante las pruebas comparativas entre el nuevo patrón y el modelo original, notamos que el nuevo patrón ofreció una mejor eficiencia operativa. Bajo condiciones de carga moderada a alta, se observó una disminución en la latencia de mensajes, lo que resultó en un tiempo de procesamiento más rápido. Además, la utilización de recursos se optimizó con el nuevo patrón, mostrando un uso más eficiente de la CPU y la memoria en comparación con el modelo publicador/suscriptor.

Impacto del Nuevo Patrón bajo Carga Adicional:

Incrementar la carga del sistema mientras se utilizaba el nuevo patrón de comunicación reveló que el sistema mantuvo su estabilidad y desempeño incluso con una carga significativamente mayor. Se observó una capacidad notable para manejar un número

más elevado de sensores y mediciones sin comprometer la latencia ni la eficiencia en el procesamiento de datos.

Comparación con el Modelo Original:

Al contrastar los resultados obtenidos con el patrón push/pull y el modelo publicador/suscriptor bajo cargas de trabajo incrementadas, se destacó que el nuevo patrón superó al modelo original en términos de latencia y capacidad para manejar picos de carga. Mientras que el modelo original mostró una tendencia a aumentar la latencia con el incremento de la carga, el nuevo patrón mantuvo una latencia estable y menor, indicando una mayor resistencia frente a cargas pico.

Prueba	Publicador/Suscriptor (segundos)	Push/Pull (segundos)
1	0.035	0.022
2	0.041	0.018
3	0.038	0.021
4	0.045	0.019
5	0.039	0.020

Conclusiones y Recomendaciones:

Los resultados sugieren que la implementación del nuevo patrón de comunicación (push/pull) proporciona ventajas significativas en términos de eficacia, eficiencia y escalabilidad en comparación con el modelo publicador/suscriptor. Se recomienda la adopción del patrón push/pull para mejorar el rendimiento del sistema, especialmente en entornos con cargas fluctuantes o crecientes. Además, se sugiere realizar pruebas adicionales en entornos de producción para validar los resultados y evaluar el impacto en un contexto operativo real.

4.4. RESULTADOS PRUEBAS CON UN TRABAJOS DUMMY

Tipos de trabajos Dummy:

Se hicieron dos tipos diferentes de trabajos Dummy, uno en el que se ejecutaban por aparte por terminal los procesos y así se generaba carga y estrés en la CPU del sistema y por otra parte un trabajo dummy que sucedía dentro de la misma clase monitor para ver si un fork() desde adentro también podía afectar en los resultados.

```
int main() {
    for (int i = 0; i < 100000000; ++i) {
        // Simular trabajo pesado
        double result = rand() * rand();
    }

    printf("Trabajo pesado completado.\n");

    return 0;
}
```

```
gian@gian-1-2:~/Descargas/Distri$ for i in {1..100}; do ./Dummy & done
[5] 12908
[6] 12909
[7] 12910
[8] 12911
[9] 12912
[10] 12913
[11] 12914
[12] 12915
[13] 12916
[14] 12917
[15] 12918
[16] 12919
[17] 12920
[18] 12921
[19] 12922
[20] 12923
[21] 12924
[22] 12925
[23] 12926
[24] 12927
[25] 12928
[26] 12929
[27] 12930
[28] 12931
[29] 12932
[30] 12933
[31] 12934
[32] 12935
[33] 12936
[34] 12937
[35] 12938
[36] 12939
[37] 12940
[38] 12941
[39] 12942
[40] 12943
[41] 12944
[42] 12945
[43] 12946
[44] 12947
[45] 12948
[46] 12949
[47] 12950
[48] 12951
[49] 12952
[50] 12953
[51] 12954
[52] 12955
[53] 12956
[54] 12957
[55] 12958
[56] 12959
[57] 12960
[58] 12961
[59] 12962
[60] 12963
[61] 12964
[62] 12965
[63] 12966
[64] 12967
[65] 12968
[66] 12969
[67] 12970
[68] 12971
[69] 12972
[70] 12973
[71] 12974
[72] 12975
[73] 12976
[74] 12977
[75] 12978
[76] 12979
[77] 12980
[78] 12981
[79] 12982
[80] 12983
[81] 12984
[82] 12985
[83] 12986
[84] 12987
[85] 12988
[86] 12989
[87] 12990
[88] 12991
[89] 12992
[90] 12993
[91] 12994
[92] 12995
[93] 12996
[94] 12997
[95] 12998
[96] 12999
[97] 13000
[98] 13001
[99] 13002
[100] 13003
```

Tiempos en el Sistema de Calidad:

Para ver como cambiaba a medida que se añadían más trabajos Dummy como se veía reflejado en el tiempo de diferencia entre la salida de la medida desde el sensor hasta el Sistema de Calidad, pero durante el experimento no se pudo ver diferencia alguna entre estos tiempos, debido a que la hora a la que salían las medidas era la misma exacta hora a la que llegaban al Sistema.

Resultados:

Como resultados ya dijimos que obtuvimos que el tiempo que tardaba en llegar la medida hasta la alerta del Sistema de calidad no cambió, puede que hubiese tardado pocas décimas más, pero en este caso no le dimos importancia debido a que se estaba hablando de procesos, los problemas que empiezan a generar estos generalmente generan grandes retrasos aunque sea en acciones simple, pero en este ejemplo no fue así.

Análisis con gprof:

Debido a que estos resultados nos fueron curiosos decidimos utilizar la herramienta gprof que nos daba cierta información acerca de los tiempos y detalles de nuestro programa:

```
Call graph (explanation follows)

granularity: each sample hit covers 4 byte(s) no time propagated

index % time    self  children  called    name
-----
[1]    0.0      0.00    0.00    60/60    main [10]
      0.0      0.00    0.00    60      enviarInfoMonitor [1]
-----
[2]    0.0      0.00    0.00    60/60    main [10]
      0.0      0.00    0.00    60      recibirInfo [2]
      0.0      0.00    0.00    54/54    validarValor [3]
      0.0      0.00    0.00    18/18    enviarAlarmaSistemaCalidad [4]
-----
[3]    0.0      0.00    0.00    54/54    recibirInfo [2]
      0.0      0.00    0.00    54      validarValor [3]
-----
[4]    0.0      0.00    0.00    18/18    recibirInfo [2]
      0.0      0.00    0.00    18      enviarAlarmaSistemaCalidad [4]
-----

This table describes the call tree of the program, and was sorted by
the total amount of time spent in each function and its children.
```

Esta tabla que es generada al momento de ejecutar el monitor nos dice el tiempo promedio que se tarda el programa en cada función, pero como se puede ver, este tiempo es tan mínimo que se ve como tiempo 0.00

Conclusiones y Recomendaciones:

- **Efecto de Trabajos Dummy:**
 - Los trabajos dummy generaron carga en la CPU, simulando escenarios de carga variable y evaluando el impacto de procesos adicionales.
- **Impacto en Tiempos del Sistema de Calidad:**
 - A pesar de la introducción de trabajos dummy, no se observaron cambios significativos en los tiempos de llegada de medidas al Sistema de Calidad. La sincronización se mantuvo constante.
- **Análisis con gprof:**
 - La herramienta gprof indicó tiempos mínimos en cada función del programa, demostrando eficiencia en la ejecución.

5. CONCLUSIONES

Rendimiento ante Cargas Variables: Se evidenció una correlación directa entre el aumento en la cantidad de sensores y el impacto en el tiempo de respuesta del sistema. La demanda de recursos se incrementó con cargas mayores, resaltando la importancia de optimizar el sistema para mantener un rendimiento estable.

Tolerancia a Fallos Eficaz: Las pruebas de tolerancia a fallos demostraron la capacidad del sistema para reaccionar ante fallas de componentes, con transiciones transparentes a procesos réplica y una continuidad operativa sin pérdida significativa de datos.

Nuevo Patrón de Comunicación Eficaz: La implementación del patrón push/pull en lugar del modelo publicador/suscriptor mostró mejoras significativas en la entrega eficiente de datos, escalabilidad y uso más eficiente de recursos, resaltando su superioridad frente al modelo original.

Enfoque en Seguridad: La implementación de un cifrado de extremo a extremo y la autenticación de dispositivos garantizan la confidencialidad y la integridad de la comunicación entre componentes, priorizando la seguridad de los datos.

Tolerancia a Fallos Robusta: Las pruebas mostraron una capacidad sólida del sistema para identificar fallos, con procesos réplica asumiendo la carga rápidamente y manteniendo la continuidad operativa sin interrupciones significativas.

Análisis de Rendimiento Detallado: Las pruebas de rendimiento identificaron claramente el impacto de las cargas variables en el sistema, resaltando la importancia de optimizar los recursos para mantener un rendimiento estable.

Beneficios del Nuevo Patrón de Comunicación: La adopción del patrón push/pull demostró mejoras significativas en la eficacia, eficiencia y escalabilidad del sistema, destacando su utilidad en entornos con cargas cambiantes.

Necesidad de Validación en Entornos Reales: A pesar de los resultados prometedores en pruebas simuladas, se subraya la importancia de realizar pruebas adicionales en entornos operativos reales para garantizar la efectividad y fiabilidad del sistema en condiciones prácticas.

Optimización Continua: Se recomienda una optimización constante del sistema para adaptarse a cambios en las cargas y condiciones operativas, garantizando un rendimiento óptimo y una resistencia sólida ante posibles fallos.