

Reporte de Experimentación

Caso de Uso: Consultar Mesa de Votación

Introducción

En el siguiente documento se describe el procedimiento para validar el software de votaciones desarrollado para la empresa XYZ y el cumplimiento de los atributos de calidad establecidos. Se empieza evaluando lo anterior para el primer caso de uso definido, *Consultar Mesa de votación*.

Para abordar el requerimiento de alta disponibilidad y rendimiento en el caso de uso, se diseñó una arquitectura distribuida haciendo uso de patrones de diseño como *Broker* y *Proxy*.

Así mismo, se tomarán en cuenta los valores de referencia establecidos a continuación:

Consulta mesa:
30 millones
8am - 4pm = 8h = 480 min
 $30m / 480 = 62.500$ en distribución
uniforme.
Con dos horas pico iguales de
32% c/u: 160.000 por minuto
o 2.666 por segundo

Objetivos

- Garantizar una alta disponibilidad del sistema a la hora de permitir a los ciudadanos consultar su mesa de votación desde su respectivo dispositivo de consulta.
- Evaluar el rendimiento de la funcionalidad descrita, especialmente en escenarios de estrés (gran número de consultas simultáneas/falla de alguno de los elementos del sistema).
- Usar métodos cuantitativos que permitan verificar y respaldar el cumplimiento de los objetivos (en términos de atributos de calidad) establecidos para el caso de uso.

Experimentación

Para realizar la experimentación, en primer lugar, se hizo el despliegue de los componentes necesarios para el caso de uso en mano (a través de fat jars, para asegurar el uso de las dependencias y/o versiones correctas), definido por la siguiente especificación:

- Query Server: x104m04

- Broker: x104m05
- Proxy: x104m06 / x104m07
- QueryDevice: x104m08
- QueryTester: x104m09

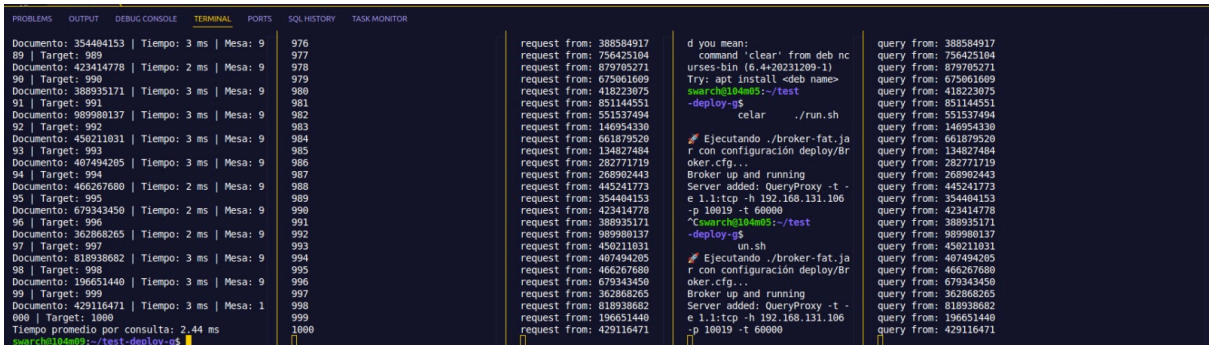
Una vez los componentes se ejecuten correctamente (en el orden definido anteriormente), el QueryTester tiene un método que lee un archivo csv, con 1000 datos y que contiene la siguiente información:

- Cédula
- MesaID
- Target

El QueryTester realiza la consulta de la mesa de votación para cada una de las cédulas de la lista; si durante la ejecución se observa que el target y el mesaID son iguales, la información está siendo obtenida correctamente.

Adicionalmente, por consulta se mide el tiempo que cada una toma y con dichos tiempo, se obtiene el promedio de tiempo para las consultas.

Resultados



# Consultas	Tiempo Promedio [ms]
1000	2.44
# Consultas / segundo	= 1000 / 2.44
	= 409 consultas/s

Conclusiones

La experimentación realizada para el caso de uso Consultar Mesa de Votación evidencia que el sistema cumple con los atributos de calidad definidos, específicamente con el

rendimiento, donde se alcanzó un tiempo promedio de respuesta de 2.44 ms por consulta, lo que implica un throughput de aproximadamente 409 consultas por segundo, un valor alto que demuestra la eficiencia del sistema ante múltiples solicitudes concurrentes.

No obstante, si bien PostgreSQL ha demostrado ser suficiente en esta prueba, el uso de una base de datos NoSQL como Cassandra podría aportar ventajas significativas en escenarios de mayor escala, dado a su alta resiliencia ante fallos, su optimización para escritura y lectura distribuida, alta disponibilidad y escalabilidad.

Reporte de Experimentación

Caso de Uso: Votar

Introducción

En el siguiente documento se describe el procedimiento para validar el software de votaciones desarrollado para la empresa XYZ y el cumplimiento de los atributos de calidad establecidos. Se evalúa, en esta sección, lo anterior para el segundo caso de uso definido, *Votar*.

Para abordar el requerimiento de confiabilidad, alta disponibilidad y unicidad; se diseñó una arquitectura distribuida haciendo uso de patrones de diseño como *Broker*, *Proxy* y *Reliable Messaging*.

Así mismo, se tomarán en cuenta los valores de referencia establecidos a continuación:

Votación:
30 millones
 $x2/3=20$ millones
8am - 4pm = 8h = 480 min
 $20m / 480 = 41.666$ en
distribución uniforme.
Con dos horas pico iguales de
32% c/u: 106.666 por minuto
o 1.777 por segundo

Objetivos

- Garantizar una alta disponibilidad del sistema a la hora de permitir a los ciudadanos votar en su respectiva mesa de votación.
- Garantizar que los votos realizados sean únicos y que no haya pérdida de información.
- Evaluar la confiabilidad en la comunicación dentro del sistema.
- Usar métodos cuantitativos que permitan verificar y respaldar el cumplimiento de los objetivos (en términos de atributos de calidad) establecidos para el caso de uso.

Experimentación

Para realizar la experimentación, en primer lugar, se hizo el despliegue de los componentes necesarios para el caso de uso en mano (a través de fat jars, para asegurar el uso de las dependencias y/o versiones correctas), definido por la siguiente especificación:

- QueryServer: x104m04
- Server: x104m03
- Broker: x104m05
- QueryProxy: x104m06
- ProxySync: x104m30

- VotingSite: x104m29
- VotingTable: x104m31
- LoadTester: x104m28

Una vez los componentes se ejecuten correctamente, el LoadTester consume la interfaz provista *vote*, del nodo VotingTable. Para dicho método, se utilizan las cédulas de los votos de la mesa 5 (la cual es la VotingTable que fue desplegada para evitar rechazos de votos por duplicación y por mesa incorrecta) en cada llamado del método *vote*. Se utilizó una muestra de 980 cédulas.

Adicionalmente, por voto se mide el tiempo que cada una toma y con dichos tiempos, se obtiene el promedio de tiempo para los votos.

Finalmente, se hizo dicha experimentación para dos escenarios:

- Ningún ciudadano de la mesa 5 ha votado, por lo que todos los votos son aceptados.
- Todos los ciudadanos de la mesa 5 han votado, por lo que todos los votos son rechazados.

Resultados

Escenario 1: Aceptación Total de los Votos

# Votos	Tiempo Promedio [ms]
980	3.18
# Votos / segundo	= $980 / 3.18$
	= 308 votos/s

Escenario 2: Rechazo Total de los Votos

# Votos	Tiempo Promedio [ms]
980	3.02
# Votos / segundo	= $980 / 3.02$
	= 324.5 votos/s

Conclusiones

La experimentación del caso de uso Votar demuestra que el sistema de votación implementado cumple de manera satisfactoria con los atributos de calidad establecidos, particularmente en cuanto a confiabilidad, alta disponibilidad y unicidad de los votos.

Los resultados muestran un comportamiento estable del sistema en ambos escenarios evaluados:

- En el escenario de aceptación total, el sistema logró un tiempo promedio de respuesta de 3.18 ms por voto, con un throughput de 308 votos por segundo.
- En el escenario de rechazo total, el tiempo promedio fue ligeramente inferior, 3.02 ms por voto, con un throughput de aproximadamente 324.5 votos por segundo. Dicho escenario también permite observar que se mantiene la unicidad de los votos, al rechazar la acción para cédulas que previamente habían realizado la acción de votar.

Los valores obtenidos evidencian que el sistema no solo responde eficientemente ante una gran cantidad de solicitudes simultáneas, sino que también maneja correctamente la lógica de validación del voto sin afectar significativamente el rendimiento, lo cual es crucial para garantizar la unicidad y exactitud del proceso electoral.