

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

ESCUELA DE CIENCIAS Y SISTEMAS

SISTEMAS OPERATIVOS 1 SECCIÓN A

ING. JESÚS ALBERTO GUZMÁN POLANCO

AUX. GERMAN JOSÉ PAZ CORDÓN

PRIMER SEMESTRE 2023



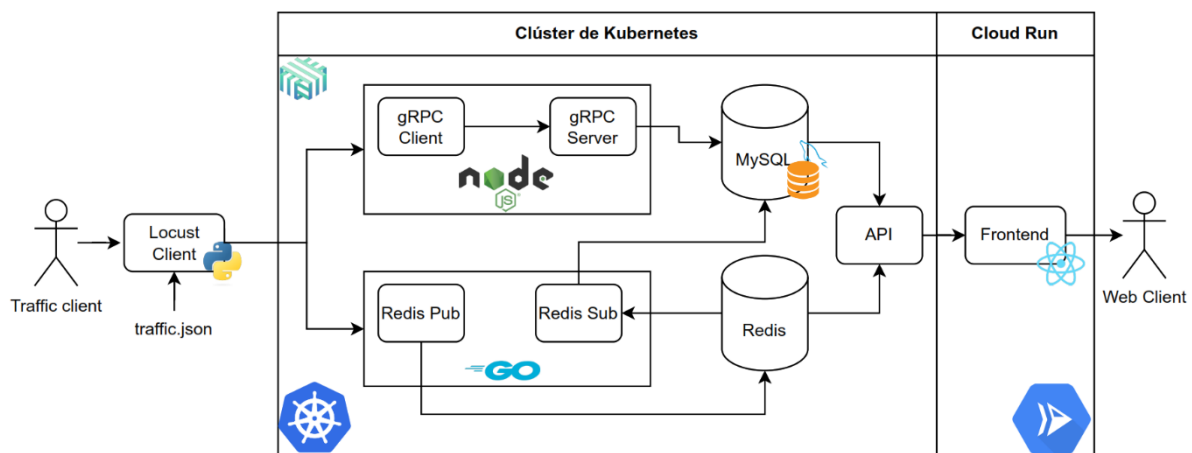
PROYECTO

VISUALIZADOR EN TIEMPO REAL DEL PORCENTAJE DE VOTOS EN ELECCIONES

OBJETIVOS

- Comprender la concurrencia y la teoría del paralelismo para desarrollar sistemas distribuidos.
- Experimentar y probar tecnologías de la nube que ayudan a desarrollar sistemas distribuidos moderno.
- Implementar contenedores y orquestadores en sistemas distribuidos.
- Medir la fiabilidad y el rendimiento en sistemas de alta disponibilidad.

ARQUITECTURA



DESCRIPCIÓN

Se debe crear un sistema distribuido que muestre estadísticas en tiempo real mediante Kubernetes y tecnologías en la nube. También se debe proporcionar un despliegue blue/green, es decir, una división de tráfico de entrada. Este proyecto será aplicado para llevar el control sobre el porcentaje de votos emitidos en las elecciones.

PRIMERA PARTE (Generador de tráfico Locust.io)

Esta parte consiste en la creación de un generador de tráfico de entrada utilizando Locust y Python como lenguaje de programación. Este tráfico será recibido por un balanceador de carga (k8s ingress). Se debe cumplir con la generación de un archivo con el nombre **traffic.json**. El contenido deberán ser los parámetros para enviar aleatoriamente en cada solicitud, estos datos se obtendrán de manera aleatoria. El archivo de aplicación o prueba de Locust debe denominarse como **traffic.py**.

Ejemplo de archivo:

```
[
    {
        "sede": 1,
        "municipio": "Guatemala",
        "departamento": "Guatemala",
        "papeleta": "Blanca",
        "partido": "FCN"
    },
    ...
]
```

Tipo de papeleta:

- **Blanca:** presidente
- **Verde:** diputados
- **Rosada:** alcaldía

Partidos:

- UNE
- VAMOS
- FCN
- UNIONISTA
- VALOR

SEGUNDA PARTE (Docker y Kubernetes)

Esta parte contiene el uso de Docker y la instalación de un clúster de Kubernetes.

Docker: se usará para empaquetar las aplicaciones dentro de contenedores, se recomienda utilizar imágenes sin distribución para reducir el tamaño de estas a lo menor posible. Docker será la herramienta para crear un entorno local para realizar pruebas antes de que las imágenes generadas se implementen en Kubernetes.

Kubernetes: se encargará de la orquestación de los contenedores de las diferentes partes de la aplicación. Para el uso este tipo de tecnologías, el proyecto está diseñado para crear un entorno y una aplicación básica desplegada en la nube.

Rutas: el objetivo es comparar los tiempos de respuesta y el rendimiento entre las diferentes rutas. Las cuales utilizaran la tecnología de RPC y Redis Pub/Sub.

Primera ruta de acceso:

1. Generador de Tráfico
2. grpc-client
3. grpc-server
4. Escribir en la base de datos de MySQL.

Segunda ruta de acceso:

1. Generador de Tráfico
2. Redis-pub
3. Redis-sub
4. Escribir en la base de datos de MySQL.

Para la división de tráfico en el proyecto se basa en la idea de dividir el 50% de tráfico a la primera ruta y el otro 50% a la segunda ruta de acceso.

TERCERA PARTE (RPC, Pub/Sub)

La idea principal en esta parte es crear una forma de alto rendimiento para escribir datos en la MySQL, utilizando la comunicación RPC. El objetivo es comparar el rendimiento entre comparación con la otra ruta. La implementación de la primera ruta de acceso consiste en recibir el tráfico de entrada por medio del gRPC-client para luego enviar la data al gRPC-server y finalmente almacenar los datos en la base de datos. La segunda ruta conlleva un flujo similar, los datos de entrada son recibidos por Redis Pub y almacenados en una base de datos NoSQL (Redis), luego serán extraídos por Redis Sub para finalmente ser almacenados en la base de datos de MySQL.

CUARTA PARTE (Bases de Datos)

Debido a la naturaleza del sistema y la de los datos. Se pide utilizar dos tipos de bases de datos, una relacional y la otra NoSQL. Donde la primera será MySQL que deberá utilizarse para almacenar datos persistentes y Redis para implementar contadores y algunos datos caché que solamente se utilicen para mostrar datos o análisis en tiempo real. Queda a decisión del estudiante la implementación y utilización de las bases de datos.

QUINTA PARTE (Web App)

Para la última parte se debe crear un sitio web para mostrar en tiempo real los datos cargados, React para su desarrollo. Se puede usar websockets en NodeJS u otro lenguaje para mostrar la fecha en tiempo real. Se debe crear una página principal o Dashboard para mostrar los siguientes datos:

- Recopilación de datos almacenados en MySQL.
- Top 3 de departamentos con mayores votos para presidente, en MySQL.
- Gráfico circular del porcentaje de votos por partido, según municipio, y departamento, en MySQL.
- Gráfico de barras que muestre las 5 sedes con mayores votos almacenados en Redis.
- Últimos 5 votos almacenados en Redis.

El frontend de la aplicación debe ser desplegado y visualizado por medio de Cloud Run.

MONITOREO

El proyecto deberá implementar la observabilidad en la red y las respuestas asociadas a los diferentes pods por medio del uso de Linkerd.

RESTRICCIONES

- El proyecto se desarrollará en grupos de 2 estudiantes.
- La aplicación debe tener un aspecto profesional.
- La ruta de gRPC debe ser realizada con NodeJS.
- La ruta de Redis Pub/Sub debe ser realizada con Go.
- La interfaz gráfica debe ser realizada con React.
- El lenguaje de la API para conexión con el dashboard queda a elección del estudiante.
- Cualquier copia parcial o total tendrá una nota de 0 puntos y será reportada a la Escuela de Ciencias y Sistemas para que proceda como indica el reglamento.
- Utilizar un repositorio de GitHub, el nombre del directorio debe ser:
proyecto
- El código fuente debe ser administrado por medio de un repositorio de github, al momento de la calificación se bajará la última versión.
- Agregar al auxiliar al repositorio: GermanJosePazCordon
- No se aceptarán entregas tardías.

ENTREGABLES

- Enlace del repositorio de GitHub.
- Manual técnico con explicación de todos los componentes utilizados en el proyecto y manual de usuario con las instrucciones necesarias para utilizar la aplicación. Estos manuales deben ser archivos .md (Markdown).

FORMA DE ENTREGA

- Mediante UEDI, subiendo el enlace del repositorio. Solamente una persona del grupo debe realizar la entrega.

La entrega se debe realizar antes de las 23:59 del 30 de abril de 2023.