Sistemas Distribuidos 1/2023
Proyecto de Laboratorio

Profesora:
Marcela Rivera (marcela.rivera.c@usach.cl)

I. Objetivos Generales

Este laboratorio tiene como objetivo utilizar herramientas de software, tales como máquinas virtuales en entornos de trabajo como Microsoft Azure, en el contexto de sistemas distribuidos.

II. Objetivos Específicos

1. Conocer y usar servicios de computación en la nube como Microsoft Azure, Google Cloud, entre otros.
2. Comprender e implementar el procesamiento de datos mediante herramientas de software actuales que se desempeñan bajo escenarios de sistemas distribuidos
3. Implementar un estilo de arquitectura visto en clases
4. Relacionar e implementar conceptos vistos en cátedra de manera práctica.

III. Contexto

En la actualidad muchos sistemas de información se han adherido al procesamiento de grandes volúmenes de datos vía *streaming*, es decir, que a partir de una fuente de datos estos se envían constantemente hacia consumidores de interés, que utilizan dicho conjunto de datos para un fin en específico. Los datos viajan de un punto A a un punto B (pueden ser n puntos de destino) de manera secuencial, por lo cual deben ser procesados de manera gradual y consistente.

La idea principal de procesar datos de este modo, es darle un valor agregado a los datos, realizar análisis de estos en tiempo real y tomar decisiones con mayor seguridad. Sin dejar de lado un buen rendimiento.

En el presente documento se describen distintos problemas y escenarios a resolver a través del procesamiento vía *streaming*. El enfoque de solución debe ser en base a una *pipeline* que conecte diversas etapas como: almacenamiento, procesamiento, aplicar funciones de valor agregado, cálculo de características fundamentales, etc. A lo largo del documento se detalla en específico qué aspectos debe contener cada *pipeline* en relación a su temática.

Esta aplicación debe ser desarrollada bajo un enfoque relacionado con las arquitecturas de sistemas distribuidos, es decir, debe ser capaz de implementar un estilo de arquitectura, además de incluir aspectos asociados a conceptos como: balanceo de carga, replica de datos, disponibilidad, entre otros. Por lo tanto, debe ser capaz de identificar, diseñar e implementar un sistema distribuido aplicando los conceptos vistos en clases.

IV. Indicaciones iniciales

- Conformar grupos de máximo 3 personas (Puede ser individual).
- Escoger por grupo un tema de investigación y notificar al profesor mediante correo electrónico.
- Se informará gradualmente las fechas de entrega.

V. Indicaciones básicas

Dentro del laboratorio se utilizarán herramientas que se acomodan a la problemática, para esto todos los grupos deben alinearse a:

- Sistema de control de versiones para código fuente.
- Libres de utilizar el lenguaje de programación o *framework* que quieran, deben justificar su uso (*performance*, versatilidad, objetivo, compatibilidad con tecnologías, etc).
- Cualquier código, material visual, bibliografía utilizada debe ser citado en los entregables.
- Respetar el uso de las tecnologías indicadas para cada caso.
- Para conectar cada etapa de la *pipeline* utilizar Kafka.
- Utilizar el despliegue servicios *cloud*.

VI. Temas de investigación

VI.A. API Nasa

La NASA disponibiliza através de APIs sus diferentes datos, los cuales pueden incluir: imagen astronomica del día, datos climátologicos, ubicación de naves espaciales, entre otros. Todas estas APIs las puede encontrar en: `https://api.nasa.gov/`.

Por ejemplo, una API puede ser Sentry, en la cual se publica información sobre posibles impactos NEA conocidos, se debe tener en cuenta que una colisión con la Tierra por parte de un NEA considerable es un evento de muy baja probabilidad. Esta base de datos se actualiza cada día y pude encontrar documentación importante en: `https://ssd-api.jpl.nasa.gov/doc/sentry.html`

Sin embargo, puede utilizar la API que estime conveniente, queda a completa libertad. Independiente de su elección, debe cumplir con lo siguiente:

VI.A.1. Objetivos

- Ser capaz de procesar un gran volumen de datos en tiempo real.
- Almacenar datos en bruto
- Consultas de funciones agregadas como mínimo, máximo, media, mediana, desviación estándar, entre otros.
- Disponibilizar plataforma con un estilo de arquitectura tal como capas, por eventos, centrada en datos, etc. Se debe permitir interactuar con los datos, ejemplo permitir visualizar las funciones agregadas o cualquier atributo de la base de datos.
- Utilizar Docker para el ambiente de trabajo.

VI.A.2. Etapas

Dentro de las etapas de la *pipeline*, considerar al menos:

- Procesar un gran volumen de datos
- Almacenar los datos en bruto
- Aplicar las funciones agregadas mencionadas previamente
- Disponibilizar la aplicación para la interacción del usuario.

VI.A.3. Tecnologías que podría a utilizar

- MongoDB como BD.
- Apache Kafka para conectar etapas del pipeline.
- Spark o Hadoop, para operar los datos.
- Docker.
- Microsoft Azure

VI.B. Ably Hub

La API de Ably Hub disponibiliza en tiempo real una serie de bases de datos con diferentes características. Entre las opciones se tiene:

- Información de transporte de Londres de Transport for London
- Predicciones de PredictIt
- Artículos de noticias de HackerNews y Reddit
- Precio de Bitcoin e información de Coindesk y BitFlyer
- Información financiera de IEX Trading
- Datos meteorológicos de Open Weather Map
- Actualizaciones de noticias de la BBC

Documentación: `https://ably.com/documentation`
Tutoriales: `https://ably.com/tutorials`
Algunas de las bases de datos: `https://ably.com/api-streamer/examples`

VI.B.1. Objetivos

- Ser capaz de procesar un gran volumen de datos en tiempo real.
- Almacenar datos en bruto
- Consultas de funciones agregadas como mínimo, máximo, media, mediana, desviación estándar, entre otros.
- Disponibilizar plataforma con un estilo de arquitectura tal como capas, por eventos, centrada en datos, etc. Se debe permitir interactuar con los datos, ejemplo permitir visualizar las funciones agregadas o cualquier atributo de la base de datos.
- Utilizar Docker para el ambiente de trabajo.

VI.B.2. Etapas

Dentro de las etapas de la *pipeline*, considerar al menos:

- Procesar un gran volumen de datos
- Almacenar los datos en bruto
- Aplicar las funciones agregadas mencionadas previamente
- Disponibilizar la aplicación para la interacción del usuario.

VI.B.3. Tecnologías que podría a utilizar

- Cassandra como BD.
- Apache Kafka para conectar etapas del pipeline.
- Spark para operar los datos.
- Docker.
- Microsoft Azure

VI.C. NRE Darwin Web Service

Darwin es el motor oficial de información sobre la circulación de trenes de la industria ferroviaria británica, que proporciona predicciones de llegadas y salidas en tiempo real, números de plataforma, estimaciones de retrasos, cambios de horarios y cancelaciones.

Documentación: `https://www.nationalrail.co.uk/static/images/structure/css/Developer_Guidelines_v\%2005-01.pdf`

Tutorial: `https://www.nationalrail.co.uk/static/documents/WA063A02411_Staff_Web_Service_User_Guide_Issue_1a_(2).pdf`

- Ser capaz de procesar texto en tiempo real.
- Almacenar datos representativos en bruto.
- Implementar funciones agregadas como mínimo, máximo, media, mediana, etc para profundidad y magnitud.
- Disponibilizar plataforma sencilla con un estilo de arquitectura tal como capas, por eventos, centrada en datos, etc. Se debe permitir interactuar con los datos, ejemplo permitir visualizar las funciones agregadas o cualquier atributo de la base de datos.

VI.C.1. Etapas

Dentro de las etapas de la *pipeline*, considerar al menos:

- Almacenamiento de datos en bruto
- Calcular estadísticas básicas mencionadas anteriormente.
- Disponibilizar aplicación para la interacción de datos

VI.C.2. Tecnologías que podría utilizar

- Apache Kafka.
- Docker.
- Azure Blob Storage o Azure Data Lake Storage como almacenamiento predeterminado
- Spark

VII. Entregables

VII.A. Entrega 1: Investigación de tecnologías y tema

La primera experiencia de este laboratorio esta basada en comprender el estado del arte de las tecnologías a utilizar en cada caso y comprender el tema a abordar. Es primordial que para esta entrega comprendan de manera conceptual cómo funciona Kafka (clúster, tópicos, particiones, consumidor, productor, *broker*, *offset*, etc).

VII.A.1. Entregable

El entregable para esta experiencia consiste en un informe y presentación que debe durar a lo más 15 minutos y debe explicar los distintos aspectos básicos de las tecnologías que se utilizarán. Para el informe, se espera que siga los siguientes lineamientos:

1. Presentación e integrantes.
2. Tabla de contenidos.
3. Introducción.
4. Presentación de tema y problema que se quiere resolver.
5. Descripción de tecnologías a utilizar.
6. Estado del arte de las tecnologías a utilizar.
7. Ventajas y desventajas de las tecnologías.
8. Tecnologías similares.
9. Quiénes ocupan estas tecnología.
10. Diseño de *pipeline* a desarrollar.
11. Extrapolación, otros usos u objetivos de la *pipeline*.
12. Conclusiones.
13. Referencias/bibliografía.

VII.B. Entrega 2: Prueba de concepto

La segunda entrega consiste en realizar una prueba de concepto de las tecnologías a utilizar en cada caso. La idea principal de la entrega consiste en un acercamiento de implementación del diseño de la *pipeline*.

VII.B.1. Entregable

El entregable 2 para esta experiencia consiste en una presentación que debe durar a lo más 20 minutos, explicar aspectos importantes del código fuente y exposición de los resultados obtenidos. La presentación debe abordar:

- Presentación e integrantes.
- Tabla de contenidos.
- Resumen presentación anterior.
- Inconvenientes de la experiencia.
- Resultados obtenidos.
- Reporte de si ha modificado el diseño de la *pipeline* inicial.
- Contestar las siguientes preguntas:
  - ¿Cómo se puede escalar horizontalmente cada etapa de la *pipeline*?
  - ¿Cómo se puede asegurar la integridad de los datos a lo largo de la *pipeline*? ¿Existe pérdida de datos?
  - ¿Cómo puedo monitorear el funcionamiento de la *pipeline*?
  - ¿Cómo puedo monitorear consumo de recursos por la *pipeline* (CPU, RAM, *network*, etc)?
- Conclusiones de la experiencia.
- Referencias/bibliografía.

VII.C. Entrega 3: *Pipeline* en funcionamiento

La última entrega consiste en tener la *pipeline* en funcionamiento. Se espera que para esta entrega sea posible monitorear el funcionamiento mediante herramientas como: Kibana, Grafana, Prometheus, etc.

VII.C.1. Entregable

El último entregable del laboratorio consiste en una presentación de la *pipeline* realizada. Puede apoyarse en diapositivas que expliquen aspecto importantes del producto, caso de éxito/fracaso, flujo de datos, etc. También se requiere de un PaaS (Plataforma como Servicio por su sigla en inglés), en la cual se pueda navegar por los datos (visualización o realizar consultas interesantes para una persona natural). La idea es visibilizar el resultado de la *pipeline* mediante una plataforma.

VIII. Consejos para el laboratorio

- Investiguen las características esenciales de las tecnologías y sus versiones (traten de ser consistentes con las versiones que utilice el grupo).
- Orienten su solución a micro servicios, simplificará sus vidas.
- Documenten casos de éxito y fracaso, sirve para el aprendizaje y para generar discusión en el curso.
- Vea casos de estudio como Netflix, Spotify, Facebook, etc. Para contrastar el uso de tecnologías e implementación en gran escala.
- Empiece a trabajar lo antes posible, el diseño, estudio e implementación requieren de mucho tiempo.