



Sistemas Distribuidos I (75.74)

Metrics && Alert Server

TP1: Concurrencia y Comunicaciones

Docentes

- Pablo D. Roca
- Ezequiel Torres Feyuk
- Ana Czarnitzki
- Cristian Raña



Requerimientos Funcionales

- Se solicita un sistema distribuido que brinde servicios para el reporte de eventos de aplicación, consulta de métricas y disparo de alertas.
- El sistema distribuido será instalado dentro de una empresa que posee un gran ecosistema de aplicaciones y usuarios que monitorean los servicios.
- Las aplicaciones deberán enviar eventos al sistema indicando: **MetricID** (string), **Value** (float)
 - Respuestas a contemplar:
 - Métrica recibida correctamente
 - Formato de métrica incorrecto
 - Servidor no disponible. Intente más tarde



Requerimientos Funcionales

- Los usuarios de monitoreo podrán agregar los valores de métricas mediante consultas con los siguientes parámetros:
 - **MetricId** (String, Obligatorio): Id de la métrica a procesar
 - **From, To** (Dates, Opcionales): Intervalo de tiempo desde-hasta.
 - **Aggregation** (Enum, Obligatorio): Operación de agregación para los valores de la métrica. [AVG|MIN|MAX|COUNT].
 - **AggregationWindowSecs** (float, Obligatorio): Intervalo de segundos dentro del cual se agregan los valores. 0 secs obtiene datos individuales.
 - Respuestas a contemplar:
 - *Query* procesada correctamente. Retorna arreglo con agregación
 - MetricId recibido no existe
 - Formato de *Query* incorrecto
 - Servidor no disponible. Intente más tarde



Requerimientos Funcionales

- El sistema emitirá alertas cada 1 minuto cuando detecte que cierta métrica agregada excede un valor definido por configuración.
- La alerta se configuración indicando:
 - **MetricId** (String, Obligatorio): Id de la métrica a verificar
 - **Aggregation** (Enum, Obligatorio): idem anterior.
[AVG|MIN|MAX|COUNT].
 - **AggregationWindowSecs** (float, Obligatorio): idem anterior.
 - **Limit** (float, Obligatorio): valor límite que dispara la alerta si es excedido



Requerimientos No Funcionales

- Se esperan una gran cantidad de aplicaciones. Las mismas enviarán cantidades masivas de eventos por segundo.
- Las aplicaciones y los usuarios de monitoreo deberán ser simuladas y se conectarán al sistema mediante TCP.
- Se espera que el sistema se encuentre **disponible** en todo momento, permitiendo que descarte eventos en caso de no poder procesarlos
- El sistema debe generar su propia marca de tiempo (timestamp) para cada evento a almacenar.
- No se pueden utilizar bases de datos (e.g. MongoDB) para almacenar los eventos
- No está permitido almacenar todos los eventos recibidos en un único archivo. Se deben particionar los datos.
- Está permitido el uso de Python, goLang, Rust, Java y C++.
- No está permitido el uso de librerías de comunicación.



Se espera del alumno:

- Empleo del tiempo de consultas en clase para resolver dudas y clarificar el negocio del sistema a construir previo a su diseño
- Exposición y verificación en clase de la arquitectura propuesta antes de iniciar su implementación
- Empleo del grupo de correos para realizar consultas que no pudieran ser resueltas en clase
- Consideración de prácticas distribuidas según lo estudiado en clase para elaborar una arquitectura flexible, escalable y robusta
- Aprobación del cuerpo docente para el uso de cualquier librería.
- Demo del sistema en funcionamiento previamente ensayada



- Fecha de entrega:
 - 26/4/2022
- Fecha de re-entrega:
 - 10/5/2022
- Formato de entrega:
 - Demostración del sistema utilizando Docker.
 - Entrega digital mediante correo personal.
 - Informe incluyendo:
 - Carátula
 - Vista Física con 1 diagrama
 - Vista de Procesos con 1 diagrama
 - Al menos una sección que indique cómo se resuelve la concurrencia.