



/home/español/java/¿Qué determina la compensación del consumidor de Kafka?_

¿Qué determina la compensación del consumidor de Kafka?

Soy relativamente nuevo en Kafka. He hecho un poco de experimentación con él, pero algunas cosas no me resultan claras con respecto a la compensación del consumidor. Por lo que he entendido hasta ahora, cuando un consumidor comienza, la compensación desde la cual comenzará a leer está determinada por la configuración `auto.offset.reset` (corríjame si me equivoco).

Ahora digamos, por ejemplo, que hay 10 mensajes (compensaciones de 0 a 9) en el tema, y que un consumidor consumió 5 de ellos antes de que bajara (o antes de que yo matara al consumidor). Entonces digamos que reinicio ese proceso de consumo. Mis preguntas son:

1. Si el `auto.offset.reset` se establece en `smallest`, ¿siempre va a comenzar a consumir desde el offset 0?
2. Si el `auto.offset.reset` se establece en `largest`, ¿empezará a consumir desde el offset 5?
3. ¿Es el comportamiento con respecto a este tipo de escenario siempre determinista?

Por favor, no dude en comentar si algo en mi pregunta no está claro. Gracias por adelantado.

java distributed-computing apache-kafka

133



Asif Iqbal



4 sept. 2015



NEOLAND School - Java Development Bootcamp

Anuncio Aprende a programar en ur lenguajes más demandados en las g corporaciones.

neoland.es

Saber más

Es un poco más complejo de lo que has descrito. La configuración `auto.offset.reset` se activa SOLAMENTE si su grupo de consumidores no tiene un desplazamiento válido confirmado en algún lugar (2 almacenamientos de desplazamiento compatibles ahora son Kafka y Zookeeper). Y también depende del tipo de consumidor que uses.



Si utiliza un consumidor de Java de alto nivel, imagine los siguientes escenarios:

1. Tiene un consumidor en un grupo de consumidores `group1` que ha consumido 5 mensajes y ha fallecido. La próxima vez que inicie este consumidor, ni siquiera usará esa configuración de `auto.offset.reset` y continuará desde el lugar donde murió porque solo recuperará el desplazamiento almacenado del almacenamiento de compensación (Kafka o ZK, como he mencionado).
2. Tiene mensajes en un tema (como lo describió) e inicia un consumidor en un nuevo grupo de consumidores `group2`. No hay desplazamiento almacenado en ninguna parte y esta vez la configuración de `auto.offset.reset` decidirá si comenzar desde el principio del tema (`smallest`) o desde el final del tema (`largest`)

Una cosa más que afecta a qué valor de compensación corresponderá con las configuraciones `smallest` y `largest` es la política de retención de registros. Imagina que tienes un tema con retención configurada a 1 hora. Produces 5 mensajes y, una hora después, publicas 5 mensajes más. El desplazamiento `largest` seguirá siendo el mismo que en el ejemplo anterior, pero el `smallest` no podrá ser 0 porque Kafka ya eliminará estos mensajes y, por lo tanto, el desplazamiento más pequeño disponible será 5.

Todo lo mencionado anteriormente no está relacionado con `SimpleConsumer` y cada vez que lo ejecute, decidirá por dónde empezar a usar la configuración `auto.offset.reset`.



209



serejja



4 sept. 2015

NEOLAND School - Java Development Bootcamp

Anuncio Aprende a programar en ur lenguajes más demandados en las g corporaciones.

neoland.es

Saber más

Solo una actualización: desde Kafka 0.9 y siguientes, Kafka está utilizando una nueva versión de Java del consumidor y los nombres de los parámetros `auto.offset.reset` han cambiado; Del manual:



Qué hacer cuando no hay un desplazamiento inicial en Kafka o si el desplazamiento actual ya no existe en el servidor (por ejemplo, porque esos datos se han eliminado):

firstiest : restablece automáticamente el offset al primer offset

latest : restablece automáticamente el offset al último offset

none : lanza una excepción al consumidor si no se encuentra un desplazamiento anterior para el grupo de consumidores

cualquier otra cosa: lanzar excepción al consumidor.

Pasé un tiempo para encontrar esto después de verificar la respuesta aceptada, así que pensé que podría ser útil para la comunidad publicarla.



65



israel.zinc



9 mar. 2017

Además hay más `offsets.retention.minutes`. Si el tiempo desde la última confirmación es `> offsets.retention.minutes`, entonces `auto.offset.reset` también se activa



7



Sasa Ninkovic



2 nov. 2017





¿Por qué es más rápido procesar una matriz sin clasificar?

¿Por qué es $2 * (i * i)$ más rápido que $2 * i * i$ en Java?

How do I convert a String to an int in Java?

How do I generate random integers within a specific range in Java?

¿Cómo declarar e inicializar una matriz en Java?

Cómo dividir una cadena en Java

Why is subtracting these two times (in 1927) giving a strange result?

How do I read / convert an InputStream into a String in Java?

Cómo arreglar `Java.lang.UnsupportedClassVersionError: Versión mayor.minor no soportada`

¿Por qué `array [idx++] += "a"` aumenta `idx` una vez en Java 8 pero dos veces en Java 9 y 10?

¿Cómo creo un archivo y lo escribo en Java?

¿Cómo puedo determinar si una matriz contiene un valor particular en Java?

Evitando! = Declaraciones nulas

How do I fix `android.os.NetworkOnMainThreadException`?

Versión 52.0 mayor no soportada.

Cómo resolver `Java.lang.NoClassDefFoundError: javax/xml/bind/JAXBException` en Java 9

What's the simplest way to print a Java array?

¿Cómo comparo cadenas en Java?

When to use `LinkedList` over `ArrayList` in Java?

¿Cómo puedo crear un JAR ejecutable con dependencias usando Maven?