

# Apache Kafka

## Appunti ed Analisi

### **Cancellazione automatica dei messaggi vecchi (svecchiamento automatico)**

di Pietro Cornelio – rel. 0.1.5 del maggio 2021

La cancellazione automatica dei messaggi vecchi in Kafka, nota anche come "log retention", è un meccanismo cruciale per gestire lo spazio su disco e mantenere i dati rilevanti nel sistema. Ecco come funziona:

1. **log.retention.hours:** Questo è il parametro principale per configurare la retention dei messaggi. Definisce per quanto tempo i messaggi devono essere conservati prima di essere candidati alla cancellazione.

Per esempio, se impostiamo:

**log.retention.hours=168**

I messaggi verranno conservati per 168 ore (7 giorni) prima di poter essere eliminati.

Tuttavia, `log.retention.hours` non è l'unico parametro da considerare. Ecco altri parametri importanti correlati:

**log.retention.bytes:**

Questo parametro imposta la dimensione massima del log per partizione. Quando viene superata questa soglia, i messaggi più vecchi vengono eliminati, indipendentemente dalla loro età.

**log.segment.bytes:**

Definisce la dimensione massima di un singolo segmento di log. Quando un segmento raggiunge questa dimensione, ne viene creato uno nuovo.

**log.segment.ms:**

Specifica l'intervallo di tempo massimo dopo il quale un segmento di log deve essere chiuso e ne viene creato uno nuovo, anche se non ha raggiunto la dimensione massima.

**log.cleanup.policy:**

Questo parametro determina la politica di pulizia dei log. Le opzioni principali sono:

1. **"delete"**: elimina i messaggi vecchi (default)
2. **"compact"**: mantiene solo il messaggio più recente per ogni chiave

**log.retention.minutes** e **log.retention.ms:**

I 2 parametri sopra sono alternative a **log.retention.hours** per una granularità più fine nella configurazione del tempo di retention.

## Come funziona il processo

Kafka verifica periodicamente i segmenti di log – tramite un suo job definito job cleaner - per ogni partizione. Se un segmento è più vecchio del tempo di retention configurato (`log.retention.hours` o equivalenti) e **tutti i messaggi nel segmento sono candidati alla cancellazione**, **l'intero segmento viene eliminato**.

Se `log.retention.bytes` è configurato, Kafka eliminerà i segmenti più vecchi quando la dimensione totale dei log per una partizione supera questo valore.

È importante notare che:

**La cancellazione effettiva potrebbe non avvenire immediatamente allo scadere del tempo di retention, ma al prossimo ciclo di pulizia del job cleaner.**

**I messaggi vengono eliminati a livello di segmento, non singolarmente.**

Se si utilizzano sia `log.retention.hours` che `log.retention.bytes`, verrà applicata la condizione che si verifica per prima.

La corretta configurazione di questi parametri dipende dalle esigenze specifiche dell'applicazione, come la quantità di dati da conservare, lo spazio su disco disponibile e i requisiti di performance.

Approfondiamo il concetto “**I messaggi vengono eliminati a livello di segmento, non singolarmente**” :

Allora, questa è una caratteristica fondamentale del funzionamento di Kafka che ha importanti implicazioni pratiche. In Kafka, **i dati sono organizzati in segmenti di file su disco**. Ecco come funziona:

### Struttura dei log:

Ogni partizione di un topic è composta da uno o più segmenti.

Un segmento è un file su disco che contiene una serie di messaggi.

**Quando un segmento raggiunge una certa dimensione (configurata con `log.segment.bytes`) o età (configurata con `log.segment.ms`), viene chiuso e ne viene creato uno nuovo.**

### Eliminazione a livello di segmento:

Quando Kafka esegue la pulizia dei log, **non esamina o elimina singoli messaggi all'interno di un segmento**. Invece, valuta l'intero segmento basandosi sul timestamp del messaggio più recente in quel segmento. Quindi, se tutto il segmento è più vecchio del periodo di retention configurato, l'intero file del segmento viene eliminato dal disco.

### Implicazioni:

**Efficienza:** Eliminare interi file è molto più efficiente che rimuovere singoli messaggi e riscrivere i file.

**Granularità della retention:** La retention effettiva può essere leggermente più lunga del valore configurato per alcuni messaggi.

**Spazio su disco:** Lo spazio non viene liberato gradualmente, ma in "salti" corrispondenti alle dimensioni dei segmenti.

## ESEMPIO PRATICO

Supponiamo di avere i seguenti parametri :

**log.retention.hours=24** (1 giorno)

**log.segment.bytes=1GB**

E supponiamo che un segmento contenga messaggi da 23 ore fa a 25 ore fa

In questo caso, anche se alcuni messaggi nel segmento sono più vecchi di 24 ore, l'intero segmento non verrà eliminato finché tutti i messaggi al suo interno non avranno superato il periodo di retention.

### Considerazioni sulla configurazione:

Segmenti più piccoli (log.segment.bytes più basso) permettono una granularità più fine nella retention, ma possono aumentare l'overhead di gestione dei file.

Segmenti più grandi sono più efficienti da gestire, ma possono portare a retention leggermente più lunghe del necessario per alcuni messaggi.

### Questa caratteristica di Kafka è importante da comprendere per:

1. Stimare correttamente l'utilizzo dello spazio su disco.
2. Configurare in modo appropriato i parametri di retention e dimensione dei segmenti.
3. Capire perché alcuni messaggi potrebbero rimanere nel sistema leggermente più a lungo del periodo di retention configurato.

### Parametri di default di Kafka per la retention dei messaggi

Parametro	Valore di default	Descrizione
log.retention.hours	168	Tempo di retention dei messaggi in ore
log.retention.bytes	-1	Dimensione massima dei log per partizione
log.segment.bytes	1073741824	Dimensione massima di un singolo segmento di log
log.segment.ms	604800000	Tempo massimo prima della chiusura di un segmento
log.cleanup.policy	delete	Politica di pulizia dei log
log.retention.check.interval.ms	300000	Frequenza di controllo per la retention dei log (5 min.)
log.cleaner.min.cleanable.ratio	0.5	Rapporto minimo di dati sporchi per avviare la pulizia
log.cleaner.delete.retention.ms	86400000	Tempo di retention per i record tombstone
log.cleaner.min.compaction.lag.ms	0	Tempo minimo prima che un messaggio possa essere compattato
log.roll.hours	168	Tempo massimo prima di creare un nuovo segmento di log
log.roll.jitter.hours	0	Jitter casuale aggiunto a log.roll.hours

Pertanto come abbiamo visto, di default, Kafka mantiene i messaggi per 7 giorni (**log.retention.hours = 168**). Non c'è un limite di dimensione predefinito per i log (**log.retention.bytes = -1**), il che significa che i log possono crescere illimitatamente se non viene specificato diversamente.

La dimensione predefinita di un segmento è di 1 GB (**log.segment.bytes = 1073741824**).

La politica di pulizia predefinita è "**delete**", che elimina i messaggi vecchi.

Kafka controlla la retention dei log ogni 5 minuti per default.

È importante notare che questi valori di default sono generali e possono variare leggermente tra le diverse versioni di Kafka. Inoltre, in un ambiente di produzione, questi parametri vengono spesso personalizzati in base alle esigenze specifiche dell'applicazione e alle risorse disponibili.