Integrazione fra eXtraWay ed Elasticsearch

Versione Elasticsearch supportata: **5.2.2** (*o superiore se mantenuta la major version*) con **JAVA 8**

Caratteristiche integrazione

- L'integrazione fra eXtraWay ed Elasticsearch è gestita a livello applicativo (strato broker)
- Per ogni archivio eXtraWay (che deve essere indicizzato su Elasticsearch) deve essere creato uno specifico indice su Elasticsearch (relazione 1-1 fra archivi eXtraWay ed indici di Elasticsearch, ad ogni archivio su eXtraWay corrisponde un indice su Elasticearch)
- Su eXtraWay vengono mantenuti tutti gli indici primari e i seriali, mentre ogni altro dato da indicizzare viene lasciato ad Elasticsearch
- Tutte le scritture su eXtraWay vengono replicate sull'indice Elasticsearch (tramite specifico componente sul broker)
    - In caso di errore in scrittura sull'indice di Elasticsearch viene registrato un file di errore sul file system
    - Ogni minuto un processo verifica la presenza di eventuali file di errore e ritenta l'operazione indicata (in modo da mantenere l'allineamento fra i dati memorizzati su archivio eXtraWay e quelli presenti sull'indice di Elasticsearch)
- Le ricerche vengono realizzate:
    - Su eXtraWay se i filtri di ricerca riguardano solo campi seriali indicizzati su eXtraWay
    - Su Elasticsearch in ogni altro caso
- Ogni ricerca in formato eXtraWay ricevuta dal broker viene opportunamente parsata e, se deve essere rediretta su Elasticsearch, convertita in formato JSON
- **N.B.**: Non esiste più il concetto di selezione conosciuto in eXtraWay. La paginazione dei risultati di una ricerca su Elasticsearch comporta una nuova esecuzione della stessa (possono quindi variare anche il numero totale di risultati) → Questo porta a differenze di comportamento a livello di interfaccia dell'applicativo
- L'integrazione è stata inclusa sulle seguenti applicazioni:
    - DocWay4 (DocWay4-service)
    - 3diWS
    - MailArchiver (MSA)
- La gestione applicativa di tutte le selezioni derivanti da query ha comportato un maggiore utilizzo di memoria da parte delle webapp (o altre applicazioni) Java. Per questo motivo è stata integrata una libreria specifica per gestire la cache applicativa (mantenimento in memoria fino ad una determinata soglia e successiva serializzazione su file system): **Apache JCS**

Creazione dell'indice su Elasticsearch

Per creare un nuovo indice su Elasticsearch e popolarlo con i dati contenuti in un archivio eXtraWay occorre procedere nel modo seguente:

1. Creare l'indice su Elasticsearch (indicando il json di mapping per l'archivio). Si consiglia di chiamare l'indice con un nome del tipo '*[nome_archivio]_ddMMyyyy*', in modo da poter gestire il reindex senza dover stoppare Elasticsearch.
2. Assegnare all'indice creato '*[nome_archivio]_ddMMyyyy*' l'alias '*[nome_archivio]*'. In questo modo sarà possibile accedere all'indice tramite '*[nome_archivio]*'
3. Lanciare la procedura di importazione dati da eXtraWay

Mapping su Elasticsearch

L'operazione di mapping consiste nel definire tutte le risorse e i campi (con relativa modalità di indicizzazione) da gestire su Elasticsearch per uno specifico archivio.

Alla pagina corrente sono stati allegati due esempi di file di mapping (relativi agli archivi DocWay e ACL):

- File Mapping DocWay
- File Mapping ACL

Esempio di comando di creazione indice su Elasticsearch:

```
curl -XPUT "http://localhost:9200/xdocwaydoc_27062017?pretty=true" -d @xdocwaydoc-mapping.json
```

Assegnazione Alias

```
curl -XPOST 'localhost:9200/_reindex?pretty' -H 'Content-Type: application/json' -d'
{
  "source": {
    "index": "xdocwaydoc_27062017"
  },
```

```
  "dest": {
    "index": "xdocwaydoc"
  }
}
'
```

Per maggiori info consultare la pagina: https://www.elastic.co/guide/en/elasticsearch/reference/current/docs-reindex.html

Importazione dati da eXtraWay

L'importazione dei dati su un indice Elasticsearch dato un archivio eXtraWay già popolato, può essere realizzata tramita una nostra specifica app.

Progetto GIT: http://gitlab.bo.priv/docway5/it.tredi.xway2elastic-import

Repository Nexus: **it.tredi.xway2elastic-import**

- **Il processo di importazione dati da eXtraWay ad Elasticsearch si aspetta che il nome dell'archivio eXtraWay e dell'indice su Elasticsearch siano identici**
- E' richiesto *Java8* per l'esecuzione del comando

Configurazione

I file di configurazione dell'applicazione sono presenti nella directory [XWAY2ELASTIC-HOMEDIR]/classes. Il principale è '*application.properties*':

```
# Connessione ad elasticsearch
elasticsearch.host=127.0.0.1
elasticsearch.port=9200
elasticsearch.threads.count=
elasticsearch.client.sniff=false
#elasticsearch.username=
#elasticsearch.password=
elasticsearch.connectionRequestTimeout=30000
elasticsearch.maxRetryTimeout=30000
elasticsearch.socketTimeout=30000

# Connessione ad eXtraWay
xway.host=127.0.0.1
xway.port=4859
# Il nome del db eXtraWay e l'eventuale query per filtrare i record devono essere specificati
# come argomento del comando shell
xway.dbname=

# Classe di implementazione della conversione di documenti da XML a JSON per salvataggio su
Elasticsearch. Questo parametro e' obbligatorio, puo' essere definito anche come
# parametro di avvio dell'applicazione da shell (parametro '-doc2json.impl=')
doc2json.impl=
#doc2json.impl=it.tredi.xway2elastic.parser.DocWayDoc2Json

# Elenco di query (e relativi ordinamenti) in base alle quali recuperare i documenti da eXtraWay
per la sincronizzazione
# su Elasticsearch
#xway.query.1=([UD,/xw/@UdType/]="doc") AND ([/doc/@tipo/]="interno" OR "varie" OR "partenza")
#xway.sort.1=
#xway.query.2=([UD,/xw/@UdType/]="doc") AND ([/doc/@tipo/]="arrivo")
#xway.sort.2=
#xway.query.3=([UD,/xw/@UdType/]="fascicolo" OR "raccoglitore" OR "seduta")
#xway.sort.3=
#...
#xway.query.n=
#xway.sort.n=
```

Per poter avviare l'importazione dei dati è necessari specificare la query tramite la quale recuperare i documenti da eXtraWay. Da quanto si può notare dal file di properties precedente, è possibile specificare più istanze di query (eventualmente con relativo sort) in modo da avviare più thread di inserimento e quindi diminuire la durata dell'importazione.

Maggiore è la distribuzione dei risultati delle query, minore sarà il tempo totale di importazione.

Utilizzo

WINDOWS:

```
[XWAY2ELASTIC-HOMEDIR]/bin/xway2elastic.bat -dbname=xdocwaydoc ...
```

LINUX:

```
sh [XWAY2ELASTIC-HOMEDIR]/bin/xway2elastic.sh -dbname=xdocwaydoc ...
```

Parametri supportati:

- **-dbname=**, nome del db eXtraWay da sincronizzare su Elasticsearch (l'indice su Elasticsearch deve già essere stato creato)
- **-doc2json.impl=**, classe di implementazione della conversione di documenti da XML a JSON per salvataggio su Elasticsearch (*parametro obbligatorio*, es.: it.tredi.xway2elastic.parser.DocWayDoc2Json)
- **-titlePageSize=**, numero di salvataggi da includere in una singola richiesta su Elasticsearch (*BulkRequest*) (default = *1000*)
- **-enableXwFiles=**, abilita o meno l'indicizzazione del contenuto degli allegati xw:file (*true* / *false*, default = *true*)
- **-xwFileIndexByBulkRequest=**, true per indicizzare gli xw:file tramite BulkRequest, false tramite richiesta singola (*true* / *false*, default = *true*)
- **-disableRefresh=**, disabilita il refresh automatico dell'indice durante l'inserimento bulk (*true* / *false*, default = *true*)

Per maggiori informazioni sui comandi di sincronizzazione dati di xway2elastic-import si rimanda al file **README.md** presente nella directory [XWAY2ELASTIC-HOMEDIR]/bin.

Configurazione dell'integrazione sul broker

```
#### it.highwaytech.broker 2.0.0 ####
maxConn=10
cacheNumDoc=0
port=4859
conTimeout\ (\#\ of\ seconds\ before\ acquireConnection()\ gives\ up)=60
maxQuery=0
maxTitle=0
maxDoc=0
logStream=0
logCommand=0
notifyUser=1
host=localhost
userName=lettore
password=reader

# Parametri di connessione ad Elasticsearch
elastic.enabled=false
# Elenco di host elasticsearch: host1:port1,host2:port2,...,hostN:portN
elastic.hosts=127.0.0.1:9200
elastic.threads.count=20
elastic.sniff=false
#elastic.username=
#elastic.password=
elastic.connectionRequestTimeout=30000
elastic.maxRetryTimeout=30000
elastic.socketTimeout=30000

# Definisce se occorre forzare il refresh dell'indice dopo ogni attivita' di
salvataggio/cancellazione (true / false, default = false). False equivale al refresh dopo 1s.
elastic.forceRefresh=false

# Directory all'interno della quale registrare eventuali errori di indicizzazione riscontrati su
elasticsearch. Se non viene specificata alcuna directory, verra'
# creata ed utilizzata una directory 'elasticerrors' all'interno della directory dei temporanei.
```

```
#elastic.indexError.tempDir=
# Tempo (in minuti) di sleep del thread di che si occupa di ritentare il salvataggio degli indici
su Elasticsearch per i quali era stato
# riscontrato errore (default = 1 min)
elastic.indexErrorsJob.sleep=1

# Elenco di nomi di archivi eXtraWay (separati da virgola) che devono essere indicizzati su
Elasticsearch
elastic.dbNamesToElasticsearch=xdocwaydoc,acl

# Configurazione della relazione fra un archivio eXtraWay e il relativo indice su Elasticsearch.
Viene definito per ogni nome di archivio eXtraWay il nome (o alias) dell'indice
# su Elasticsearch e l'implementazione da adottare per la conversione da XML a JSON dei documenti
# xway2elastic.[XWAY_DBNAME].indexName=[ELASTIC_INDEXNAME]
# xway2elastic.[XWAY_DBNAME].mapping=[CLASSE_IMPLEMENTAZIONE_DOC2JSON]
xway2elastic.xdocwaydoc.indexName=xdocwaydoc
xway2elastic.xdocwaydoc.mapping=it.tredi.xway2elastic.parser.DocWayDoc2Json
xway2elastic.acl.indexName=acl
xway2elastic.acl.mapping=it.tredi.xway2elastic.parser.DocWayDoc2Json
```

Cache applicativa con Apache JCS

Il file di configurazione di Apache JSC è il seguente:

- [APPLICATION-HOME]/WEB-INF/classes/cache.ccf (in caso di webapp)
- [APPLICATION-HOME]/classes/cache.ccf (in caso di app lanciata come processo)

```
#
# http://commons.apache.org/proper/commons-jcs/LocalCacheConfig.html
#

# DEFAULT CACHE REGION
jcs.default=DC

jcs.default.cacheattributes=org.apache.commons.jcs.engine.CompositeCacheAttributes

# Numero massimo di oggetti che vengono mantenuti in memoria. Superato il limite vengono
serializzati su disco (se attiva la serializzazione) o
# rimossi (inaccessibili all'applicazione)
jcs.default.cacheattributes.MaxObjects=5000

jcs.default.cacheattributes.MemoryCacheName=org.apache.commons.jcs.engine.memory.lru.LRUMemoryCach
e

# Se disattivo non viene mai rimosso nessun oggetto dalla memoria (NO serializzazione su disco)
jcs.default.cacheattributes.UseMemoryShrinker=true

# Tempo di permanenza in memoria (espresso in secondi) dall'ultimo accesso all'oggetto. Superato
il limite vengono serializzati su disco (se attiva la serializzazione) o
# rimossi (inaccessibili all'applicazione)
jcs.default.cacheattributes.MaxMemoryIdleTimeSeconds=600

# Intervallo di attivazione dello shrink (controllo oggetti da serializzare)
jcs.default.cacheattributes.ShrinkerIntervalSeconds=30

#jcs.default.cacheattributes.MaxSpoolPerRun=500

jcs.default.elementattributes=org.apache.commons.jcs.engine.ElementAttributes

# Definisce se l'oggetto serializzato deve essere mantenuto per sempre o eliminato in determinate
condizioni (si vedano le properties successive)
jcs.default.elementattributes.IsEternal=false

# Tempo di permanenza dell'oggetto serializzato (espresso in secondi) dall'ultimo accesso.
Superato il limite l'oggetto viene rimosso.
```

```
jcs.default.elementattributes.IdleTime=1800

# Tempo di permanenza dell'oggetto serializzato (espresso in secondi) dalla creazione dello
stesso. Superato il limite l'oggetto viene rimosso.
#jcs.default.elementattributes.MaxLife=21600


# CONFIGURAZIONE PER SERIALIZZAZIONE OGGETTI SU DISCO

jcs.auxiliary.DC=org.apache.commons.jcs.auxiliary.disk.indexed.IndexedDiskCacheFactory
jcs.auxiliary.DC.attributes=org.apache.commons.jcs.auxiliary.disk.indexed.IndexedDiskCacheAttribut
es

# Percorso assoluto su disco nel quale salvare gli oggetti serializzati (dovrebbe riferirsi ad una
partizione su disco SSD per garantire adeguate performance)
jcs.auxiliary.DC.attributes.DiskPath=/tmp/jcsdw4

# Dimensione dell'area di memoria (espressa in numero di oggetti) sulla quale vengono spostati gli
oggetti da accodare per la serializzazione su disco (default = 5000). Nel caso venga richiesto
# il caricamento di un oggetto in coda nell'area di purgatorio, questo viene ripristinato in
memoria e non viene effettuata alcuna serializzazione su disco.
jcs.auxiliary.DC.attributes.MaxPurgatorySize=10000

# Numero massimo di chiavi che possono essere serializzate su disco (default = 5000). Superato il
limite gli oggetti piu' vecchi vengono rimossi.
jcs.auxiliary.DC.attributes.MaxKeySize=10000

# Cancellazione dei dati serializzati all'avvio (visto che serializziamo delle selezioni e' bene
mantenere attiva questa property visto che all'avvio di Tomcat selezioni prodotte
# in precedenza sono comunque inutilizzabili dagli utenti).
jcs.auxiliary.DC.attributes.ClearDiskOnStartup=true

# Definisce il numero di eliminazioni dopo le quali ottimizzare il file di cache su disco
(cancellazione dal file di vecchi oggetti). Di default questa property e' disattivata (-1)
# e comporta il continuo aumento della dimensione del file (che sara' poi ottimizzato solamente in
fase di shutdown).
jcs.auxiliary.DC.attributes.OptimizeAtRemoveCount=5000
```

N.B.: In caso di serializzazione su disco della cache occorre utilizzare una partizione disco ad accesso veloce per la memorizzazione degli oggetti in modo da garantire performance elevate. Property: **jcs.auxiliary.DC.attributes.DiskPath**

Per maggiori informazioni relative alla configurazione di Apache JCS si rimanda alla documentazione ufficiale: http://commons.apache.org/proper/commons-jcs/index.html

Indicizzazione di Files

TODO