

INTRODUZIONE A

Solr



+

APACHE

ZooKeeperTM

Antonio Scaiella
8 Gennaio 2025

Corso di Information Retrieval A.A. 2024/2025



AGENDA

- Introduzione ad Apache Zookeeper
 - Cos'è Apache Zookeeper?
 - Caratteristiche
 - Architettura
- Installazione Zookeeper & SolR Cloud mode
 - Download
 - GUI
- Creazione Collection
- Indexing and searching

Antonio Scaiella
Gennaio 2025

INTRODUZIONE-ZOOKEEPER

Apache ZooKeeper è un software, di proprietà di The Apache Software Foundation, in grado di fornire supporto per la gestione e la sincronizzazione di sistemi distribuiti

È robusto poiché i dati persistenti sono distribuiti tra più nodi e perché un *client* (esempio SolR) si connette a uno di essi potendo migrare se un nodo fallisce;

fintanto che una rigida maggioranza ($n/2 + 1$) di nodi Zookeeper funziona, l'insieme di nodi ZooKeeper è attivo.

In particolare, un nodo master viene scelto dinamicamente per consenso all'interno dell'ensemble; se il nodo principale cade, il ruolo del master passa a un altro nodo.

INTRODUZIONE-SOLRCLOUD

Features of Solr Cloud mode

- Fault tollerant
 - Optimized for high traffic. (Affidabilità)
 - Easy sharding & replication (Scalabilità & Affidabilità)
 - Zookeeper (Load Balancing)
-
- In poche parole:

Solr Cloud è un sistema in cui i dati sono organizzati in più parti, o frammenti, che possono essere ospitati su più macchine, con repliche che forniscono ridondanza sia per la scalabilità che per la tolleranza agli errori.

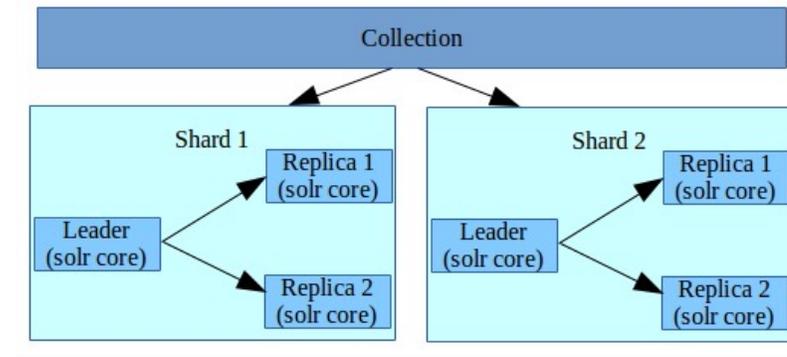
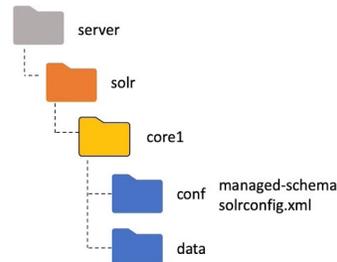
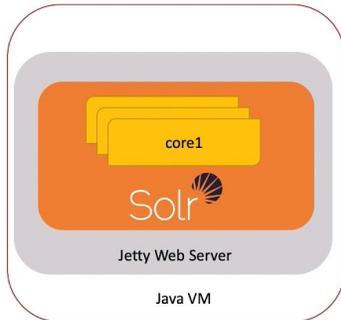
ZooKeeper aiuta a gestire la struttura generale in modo che sia l'indicizzazione che la ricerca possono essere indirizzate correttamente.

INTRODUZIONE

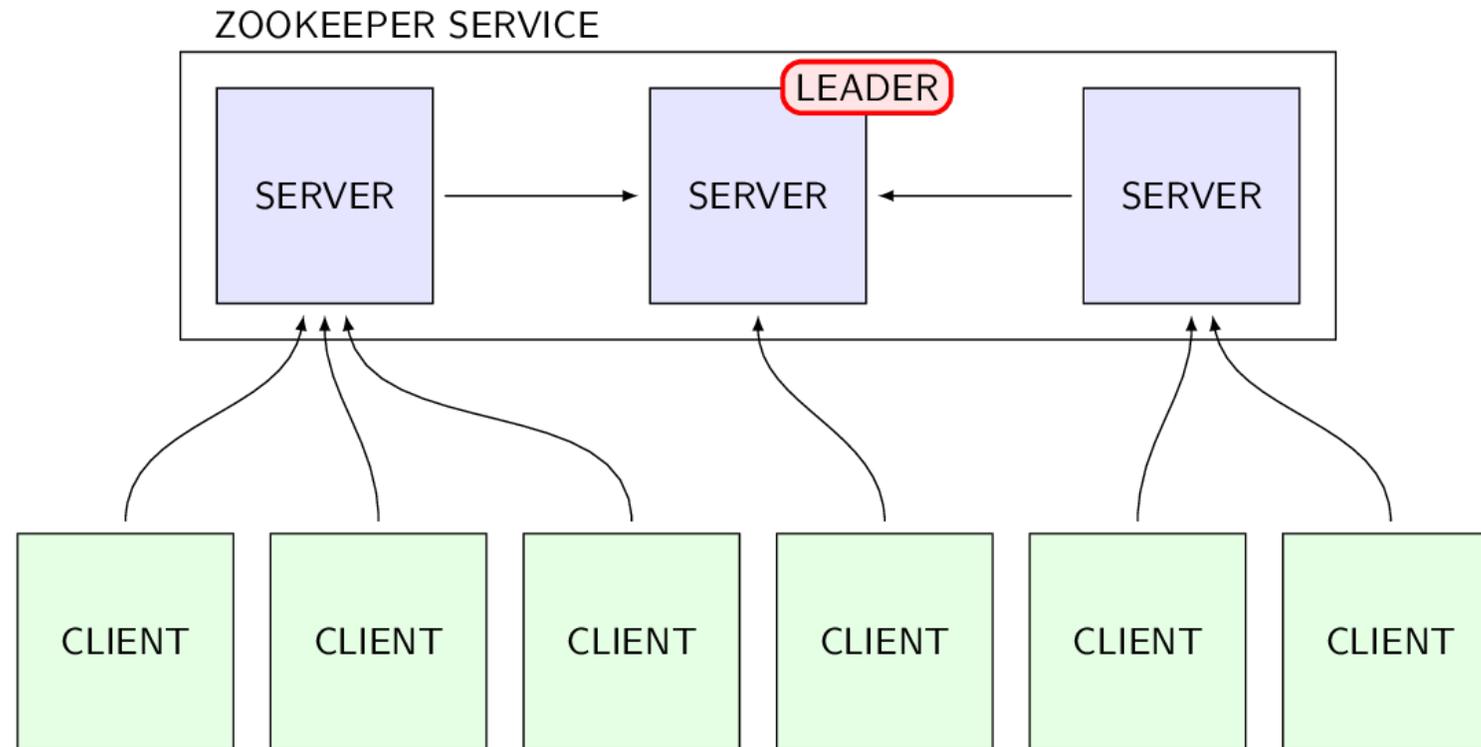
Core & collection

La **Collection** è un indice logico distribuito su più server. Il **Core** è quella parte del server che esegue una collection. Nella ricerca non distribuita, il server singolo che esegue Solr può avere più Collection e ciascuna di queste è anche un **Core**. Quindi la **Collection** e il **Core** sono gli stessi se la ricerca non è distribuita.

Solr Core



INTRODUZIONE

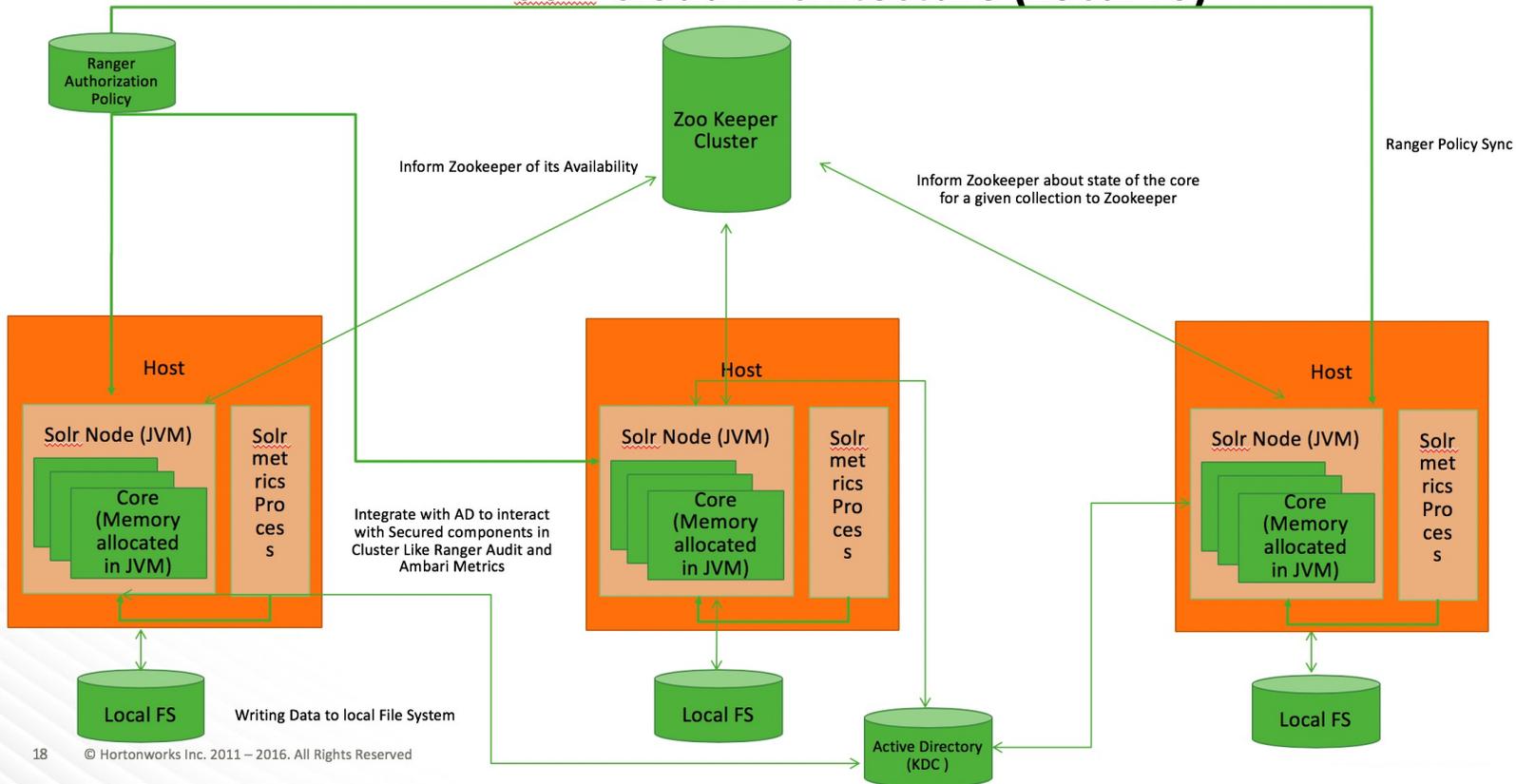


Based on figure from ZooKeeper web ... <https://zookeeper.apache.org/doc/current/images/zkservice.jpg>

Antonio Scaiella
Gennaio 2025

INTRODUZIONE

Solr Cloud Architecture (Local FS)



INSTALLAZIONE

Download Solr

- Windows

<https://solr.apache.org/downloads.html>

Scarichiamo

Binary releases: solr-8.11.4.tgz

- Linux

```
wget https://dlcdn.apache.org/lucene/solr/8.11.4/solr-8.11.4.tgz
```

```
tar -xvzf solr- solr-8.11.4.tgz
```

- java -version
- Almeno la "1.8"

Antonio Scaiella
Gennaio 2025

INSTALLAZIONE

Download Zookeeper

- Windows

- <https://dlcdn.apache.org/zookeeper/zookeeper-3.9.3/>

Scarichiamo: `apache-zookeeper-3.9.3-bin.tar.gz` ed estraiamo

- Linux

`wget https://dlcdn.apache.org/zookeeper/zookeeper-3.9.3/apache-zookeeper-3.9.3-bin.tar.gz`

`tar -xvzf apache-zookeeper-3.9.3-bin.tar.gz`

CONFIGURAZIONE

- E' opportuno avere sempre un numero dispari ≥ 3 di istanze zookeeper su macchine diverse.
- Il numero di nodi Solr, invece, dipende da quanta replicabilità/scalabilità/affidabilità si vuole garantire.
- Nel nostro caso, per questioni pratiche, andremo a creare sulla stessa macchina 3 istanze di Zookeeper e 3 di Solr in ascolto su porte diverse.
- In uno scenario reale sarebbe opportuno avere, per il nostro caso, almeno 3 macchine e su ognuna di esse configurare una coppia zookeeper/solr.
 - L'ideale sarebbe, sempre per il nostro caso, avere 6 macchine, 3 dedicate a zookeeper e 3 a Solr per garantire il massimo dell'affidabilità

CONFIGURAZIONE

- Creiamo una directory "zookeeper" e copiamo al suo interno apache-zookeeper-3.6.4-bin
- All'interno della directory "zookeeper" digitiamo:
 - `cp apache-zookeeper-3.9.3-bin/conf/zoo_sample.cfg apache-zookeeper-3.9.3-bin/conf/zoo1.cfg`
 - `cp apache-zookeeper-3.9.3-bin/conf/zoo_sample.cfg apache-zookeeper-3.9.3-bin/conf/zoo2.cfg`
 - `cp apache-zookeeper-3.9.3-bin/conf/zoo_sample.cfg apache-zookeeper-3.9.3-bin/conf/zoo3.cfg`

 - `mkdir data1`
 - `mkdir data2`
 - `mkdir data3`
 - `mkdir logs`

CONFIGURAZIONE FILE ZOO1.CFG

- Apriamo zoo1.cfg e cancelliamo tutto il contenuto di default e settiamo:

```
tickTime=2000
initTime=10
initLimit=5
syncLimit=5
clientPort=2181
dataDir=«path di data1»
server.1=ipdellamacchina:2888:3888
server.2=ipdellamacchina:4888:5888
server.3=ipdellamacchina:6888:7888
autopurge.snapRetainCount=3
autopurge.purgeInterval=1

4lw.commands.whitelist=mntr,conf,ruok
```

CONFIGURAZIONE FILE ZOO2.CFG

- Apriamo zoo2.cfg e cancelliamo tutto il contenuto di default e settiamo:

```
tickTime=2000
initTime=10
initLimit=5
syncLimit=5
clientPort=2182
dataDir=«path di data2»
server.1=ipdellamacchina:2888:3888
server.2=ipdellamacchina:4888:5888
server.3=ipdellamacchina:6888:7888
autopurge.snapRetainCount=3
autopurge.purgeInterval=1

4lw.commands.whitelist=mntr,conf,ruok
```

CONFIGURAZIONE FILE ZOO3.CFG

- Apriamo zoo3.cfg e cancelliamo tutto il contenuto di default e settiamo:

```
tickTime=2000
initTime=10
initLimit=5
syncLimit=5
clientPort=2183
dataDir=«path di data3»
server.1=ipdellamacchina:2888:3888
server.2=ipdellamacchina:4888:5888
server.3=ipdellamacchina:6888:7888
autopurge.snapRetainCount=3
autopurge.purgeInterval=1

4lw.commands.whitelist=mntr,conf,ruok
```

CONFIGURAZIONE LOGS

- creare "apache-zookeeper-3.6.4-bin/conf/zookeepr-env.sh"

- Al suo interno scrivere:

```
ZOO_LOG_DIR="path della cartella logs"
```

```
ZOO_LOG4J_PROP="INFO,ROLLINGFILE"
```

```
SERVER_JVMFLAGS="-Xms2048m -Xmx2048m -verbose:gc -XX:+PrintHeapAtGC -  
XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -  
XX:+PrintTenuringDistribution -XX:+PrintGCApplicationStoppedTime -  
Xloggc:$ZOO_LOG_DIR/zookeeper_gc.log -XX:+UseGCLogFileRotation -  
XX:NumberOfGCLogFiles=9 -XX:GCLogFileSize=20M"
```

- In genere `in ZOO_LOG_DIR` va impostato il path alla cartella "logs" creata in precedenza

CONFIGURAZIONE MYID

- Creare in "data1" un file con nome `myid` senza estensione e al suo interno scrivere 1
- Creare in "data2" un file con nome `myid` senza estensione e al suo interno scrivere 2
- Creare in "data3" un file con nome `myid` senza estensione e al suo interno scrivere 3

AVVIO ZOOKEEPER

- Possiamo ora avviare le tre istanze di zookeeper:

- `bash apache-zookeeper-3.9.3-bin/bin/zkServer.sh start zoo1.cfg`
- `bash apache-zookeeper-3.9.3-bin/bin/zkServer.sh start zoo2.cfg`
- `bash apache-zookeeper-3.9.3-bin/bin/zkServer.sh start zoo3.cfg`

RECAP

- A questo punto abbiamo 3 istanze di zookeeper che girano sulla stessa macchina.
- In uno scenario reale in cui abbiamo almeno 3 macchine da dedicare a zookeeper possiamo evitare di duplicare i file di configurazione zoo*.cfg e di creare varie cartelle "data". Inoltre è possibile mettere in ascolto tutte le istanze sulle stesse porte.
- Esempio:

Supponiamo di avere a disposizione tre macchine: 10.5.0.30, 10.5.0.31, 10.5.0.32

Su ogni macchina :

- Creiamo una directory "zookeeper" e copiamo al suo interno apache-zookeeper-3.6.3-bin
- **Digitiamo:** `cp apache-zookeeper-3.9.3-bin/conf/zoo_sample.cfg apache-zookeeper-3.9.3-bin/conf/zoo.cfg`
- `mkdir data`
- `mkdir logs`

RECAP

- Su ogni macchina :

Apriamo zoo.cfg e cancelliamo tutto il contenuto di default e settiamo:

```
tickTime=2000
initTime=10
initLimit=5
syncLimit=5
clientPort=2181
dataDir=/zookeeper/data
server.1=10.5.0.30:2888:3888
server.2=10.5.0.31:2888:3888
server.3=10.5.0.32:2888:3888
autopurge.snapRetainCount=3
autopurge.purgeInterval=1
4lw.commands.whitelist=mntr,conf,ruok
```

RECAP

Soffermandoci su questa parte di configurazione:

```
server.1=10.5.0.30:2888:3888
```

```
server.2=10.5.0.31:2888:3888
```

```
server.3=10.5.0.32:2888:3888
```

Il numero dopo "server" è l'id della macchina ed è lo stesso numero che inseriremo poi nel file myid.

RECAP

- Su ogni macchina in base all'id assegnato :
- Creare in "data" un file con nome `myid` senza estensione e al suo interno scrivere l'id assegnato alla macchina

AVVIO SOLR CLOUD MODE

Copiamo 3 directory "solr-8.11.4" _1 _2 _3

Innanzitutto impostare in ognuna solr-8.11.4_*/bin/solr.in.sh :

```
SOLR_HOST="ipdellamacchina"
```

Nota: non scrivere "localhost"

Avviamo 3 istanze di solr una per ogni directory

- `./bin/solr start -c -z ipdellamacchina1:clientPort1, ipdellamacchina2:clientPort2, ipdellamacchina3:clientPort3 -p 8983 -force`
- `./bin/solr start -c -z ipdellamacchina1:clientPort1, ipdellamacchina2:clientPort2, ipdellamacchina3:clientPort3 -p 8984 -force`
- `./bin/solr start -c -z ipdellamacchina1:clientPort1, ipdellamacchina2:clientPort2, ipdellamacchina3:clientPort3 -p 8985 -force`

Dove:

- -z indica la lista delle macchine e rispettiva porta che abbiamo inizializzato in precedenza.
- Ipdellamacchina1, ipdellamacchina2, ipdellamacchina3 nel nostro caso saranno uguali, mentre le porte saranno clientPort1:2181, clientPort2:2182, clientPort3:2183

AVVIO SOLR CLOUD MODE

Nel caso di 3 macchine diverse per solr noi avvieremo **su ogni macchina** :

- `./bin/solr start -c -z ipdellamacchina1:2181, ipdellamacchina2: ipdellamacchina1:2181, ipdellamacchina3:2181 -p 8983 -force`
- Quindi potremo avviare solr sulla porta 8983 per tutte le macchine, i zookeeper saranno in ascolto tutti sulla porta 2181 come discusso nel recap precedente e ciò che cambia è solamente gli ip delle macchine

Creazione Collection

- Da qualsiasi macchina che ospita solr possiamo digitare:
- `./bin/solr create -c mioesempio -s 1 -rf 3`
- Dove
 - -s indica il numero di shards desiderati
 - -rf indica il numero di repliche desiderate

Import Dataset di esempio

- Spostarsi in uno dei solr-8.11.4_*\example\exampledocs
 - `java -jar -Dc=mioesempio post.jar *.xml`

Modifiche dello schema

Nel caso di solr in modalità Non cloud le modifiche allo schema o la lista di stopwords, sinonimi e in generale tutto ciò che riguarda la configurazione vanno applicate nei file in :

```
solr-8.11.4\server\solr\mioesempio\conf
```

Se avviamo, invece, solr in modalità cloud e creiamo la nostra Collection, non troveremo più la configurazione al path sopracitato. Questo perché le configurazioni sono gestite da Zookeeper, quindi le modifiche vanno inviate a Zookeeper.

Download e aggiornamento dello schema

Possiamo verificare la configurazione esistente scaricandola da zookeeper con il comando:

```
./bin/solr zk downconfig -n mioesempio -d pathdidestinazione -z  
ipdellamacchina1:clientPort1,  
ipdellamacchina2:clientPort2,ipdellamacchina3:clientPort3
```

Ad esempio:

```
./bin/solr zk downconfig -n mioesempio -d ../ -z  
192.168.56.61:2181,192.168.56.61:2182,192.168.56.61:2183
```

Download e aggiornamento dello schema

Dopo aver modificato la configurazione, secondo le nostre necessità, possiamo fare upload nel seguente modo:

- Da una delle istanze Solr spostiamoci nella directory:

```
/server/scripts/cloud-scripts
```

e digitiamo il comando:

```
./zkcli.sh -zkhost [host_macchina_zookeeper] -cmd upconfig -confdir  
[pathDellaConfigurazione] -confname mioesempio
```

dove [host_macchina_zookeeper] è l'indirizzo di una delle macchine che ospita Zookeeper e [pathDellaConfigurazione] è la cartella conf. Ad esempio:

```
./zkcli.sh -zkhost 192.168.56.61 -cmd upconfig -confdir  
/mnt/c/zookeeper/conf -confname mioesempio
```

Aggiungere Documenti con la Solr Web Interface



Dashboard

Logging

Security

Core Admin

Java Properties

Thread Dump

mioesempio

Overview

Analysis

Dataimport

Documents

Files

Ping

Request-Handler (qt)

/update

Document Type

JSON

Document(s)

```
{
  "id" : "44",
  "name" : "Antonio",
  "age" : 27,
  "Designation" : "Manager",
  "Location" : "Hyderabad",
},
{
  "id" : "45",
  "name" : "Robert",
```

Commit Within

1000

Overwrite

true

Submit Document