Configurazione di WPS per Hadoop

Guida all'installazione e alla configurazione di WPS per Hadoop

Versione: 4.0.3

www.worldprogramming.com

Indice

Introduzione

Definizione di Hadoop

Hadoop è un framework di software scalabile, a tolleranza d'errore e open source per l'archiviazione e l'elaborazione distribuite di set di dati molto grandi su cluster di computer. Viene distribuito con la licenza Apache.

Per coloro che hanno appena cominciato ad utilizzare Hadoop, si consiglia di far riferimento inizialmente a *Introduzione ad Hadoop* (pag. 7).

Vantaggi concessi dall'integrazione di Hadoop in WPS

- Con l'integrazione di Hadoop, WPS amplia le sue funzionalità di integrazione dei dati su dozzine di motori di database.
- La condivisione dei dati tra WPS e HDFS (Hadoop Distributed File System) offre l'interoperabilità a livello di dati tra i due ambienti. Sebbene non sia trasparente, è piuttosto chiara: È possibile importare i dati Hadoop in WPS per l'analisi (strutturata) e successivamente, se lo si desidera, inviarli di nuovo a HDFS.
- Gli utenti di WPS possono richiamare la funzionalità Hadoop dagli ambienti familiari dell'interfaccia utente WPS Workbench.
- Gli utenti possono creare e modificare le nuove operazioni di Hadoop tramite un linguaggio simile a SQL; non devono conoscere Java.

Ambito del documento

Il presente documento offre una panoramica sull'implementazione di WPS e Hadoop. Contempla anche la configurazione di Kerberos laddove è applicabile.

Riepilogo dell'integrazione di WPS/Hadoop

Le seguenti integrazioni attualmente implementate utilizzano le estensioni `filename`, `libname` e `PROC HADOOP` affinché WPS:

- Si connetta ad Hive tramite SQL standard
- Si connetta ad Impala tramite SQL standard
- Si connetta ad Hive tramite SQL passthrough
- Si connetta ad Impala tramite SQL passthrough
- Esegua i comandi HDFS e gli script Pig

L'integrazione di Hadoop di WPS ha ricevuto la certificazione per Cloudera 5 ed è stata provata con altre distribuzioni di Hadoop che si avvicinano allo standard Apache. Alla fine di questo documento, sono riportati diversi esempi di codice relativi all'integrazione (pag. 17).

Prerequisiti

Hadoop è un gruppo tecnologico complesso a più parti. Prima dell'integrazione con WPS, è necessario installarlo e configurarlo correttamente. È necessario eseguire e ricontrollare la seguente procedura preparatoria:

1. Ottenere il set corretto di file `.jar` che corrisponde all'installazione di Hadoop.

   Nota:

   Quando si usa Apache Hive come parte dell'installazione di Hadoop con WPS, è necessario utilizzare la versione 0.12 di Apache Hive o superiore.

2. Configurare i file di configurazione XML in base al proprio ambiente specifico di cluster (indirizzi IP, porte e così via).
3. Stabilire se la distribuzione di Hadoop include o impone il supporto per Kerberos. In tal caso, confermare che l'autenticazione Kerberos funzioni per il proprio server, che l'entità sia stata configurata correttamente e così via. Indipendentemente se o meno si utilizza Kerberos, completare la restante procedura dei Prerequisiti.
4. Verificare che il cluster funziona correttamente, forse consultando l'amministratore di cluster che dovrebbero avere l'accesso ai dashboard amministrativi.
5. Una volta verificato che il cluster funziona correttamente, stabilire che le attività relative ad Hadoop si possono inviare indipendentemente da WPS.

Kerberos

Stabilire l'identità con una autenticazione avanzata è la base per l'accesso protetto in Hadoop, con utenti che devono essere in grado di identificarsi per poter accedere alle risorse, e le risorse dei cluster Hadoop devono essere autenticate per evitare i sistemi dannosi che si fingono potenzialmente parte del cluster per avere l'accesso ai dati. Per creare questa comunicazione protetta tra i vari componenti, Hadoop può utilizzare Kerberos, che è un meccanismo di autenticazione di terzi, per cui gli utenti e i servizi che gli utenti che desiderano accedere si affidano al server Kerberos per gestire l'autenticazione.

Nota:

Alcune distribuzioni di Hadoop includono (o persino impongono) il supporto per Kerberos. Le specifiche della configurazione del server Kerberos spesso variano secondo il tipo e la versione della distribuzione, e vanno oltre l'ambito di questo documento. Consultare le informazioni sulla configurazione specifica della distribuzione in dotazione con il software Hadoop. Consultare *Configurazione di Kerberos e Hadoop sul client* (pag. 16) per le modalità di configurazione di Kerberos e Hadoop sul lato client.

Introduzione ad Hadoop

Negli ambienti analitici tradizionali, i dati vengono alimentati in un RDBMS tramite un processo ETL (Extract, Tranform, Load, ovvero Estrarre, Trasformare e Caricare) iniziale. I dati non strutturati vengono preparati e caricati nel database, acquisendo man mano uno schema. Una volta caricati, diventano disponibili per un'ampia gamma di tecniche analitiche ben consolidate.

Per grandi quantità di dati, tuttavia, questo flusso di lavoro presenta alcuni problemi:

1. Se il tempo che occorre per elaborare i dati di un giorno raggiunge un punto in cui non è possibile completarlo in termini economici prima del giorno successivo, è necessario adottare un altro approccio. L'ETL su larga scala impone una forte pressione sull'infrastruttura che è alla base.
2. Man mano che i dati invecchiano, spesso vengono archiviati. Tuttavia, è molto costoso recuperare i dati archiviati in volume (da nastri, blu-ray e così via). Inoltre, una volta archiviati, non è più conveniente ed economico accedervi.
3. Il processo ETL è un processo di astrazione. I dati si aggregano e normalizzano, e perdono l'elevato livello di fedeltà originale. Se l'azienda successivamente interroga i dati con un nuovo tipo di domanda, spesso non è possibile fornire una risposta senza un esercizio costoso che implichi la modifica della logica ETL, la correzione dello schema del database e il ricaricamento.

Hadoop è stato concepito in modo da offrire:

- Scalabilità sull'elaborazione e i dati, eliminando il collo di bottiglia di ETL.
- Migliore economia nel mantenere i dati attivi, e nella memoria principale, per una durata maggiore.
- La flessibilità di tornare e interrogare i dati ad alta fedeltà originale con nuovi tipi di domande.

Confronto tra RDBMS e Hadoop

Da un punto di vista analitico, le differenze principali tra un RDBMS e Hadoop vengono mostrate di seguito.

Tabella 1. Principali differenze tra RDBMS e Hadoop

RDBMS	Hadoop
È necessario creare lo schema prima che sia possibile caricare i dati.	I dati vengono semplicemente copiati nella risorsa di archiviazione dei file e non è necessaria alcuna trasformazione.
Il funzionamento di un ETL esplicito deve accadere trasformando i dati nella struttura interna del database.	Un serializzatore/deserializzatore viene applicato al tempo di lettura per estrarre le colonne necessarie.

RDBMS	Hadoop
È necessario aggiungere esplicitamente le nuove colonne prima che si possano caricare i nuovi dati per tali colonne.	I nuovi dati possono iniziare a fluire in qualunque momento e verranno visualizzati retrospettivamente, una volta aggiornato il serializzatore/deserializzatore per analizzarlo.

L'orientamento dello schema delle implementazioni di RDBMS tradizionali offre alcuni vantaggi fondamentali che hanno comportato la loro vasta adozione:

- Le ottimizzazioni, gli indici, il partizionamento e così via, diventa possibile, consentendo le letture molto veloci per alcune operazioni quali unioni, unioni multitabella e così via.
- Uno schema comune per tutte le organizzazioni indica che diversi gruppi in un'azienda possono comunicare utilizzando un vocabolario comune.

D'altra parte, le implementazioni di RDBMS perdono in termini di flessibilità, ovvero la capacità di crescita dei dati alla velocità in cui si evolvono. Con Hadoop, la struttura si impone solo sui dati al tempo di lettura, tramite un serializzatore/deserializzatore e, successivamente, non vi è alcuna fase ETL; i file vengono semplicemente copiati nel sistema. Fondamentalmente, Hadoop non è un database tradizionale nel normale senso del termine, date le sue proprietà ACID (Atomicità, Coerenza, Isolamento, Durabilità) e anche se fosse, sarebbe probabilmente troppo lento a gestire la maggior parte delle applicazioni interattive.

Entrambe le tecnologie possono integrarsi, entrambe possono avere un posto nell'organizzazione informatica, è sufficiente scegliere lo strumento giusto per il processo giusto.

Tabella 2. Confronto tra RDBMS e Hadoop: principali casi di utilizzo

Quando usare RDBMS	Quando usare Hadoop
OLAP interattiva: tempo di risposta sotto il secondo.	Quando è necessario gestire sia dati strutturati che dati non strutturati.
Quando è necessario supportare transazioni ACID a più passaggi su dati basati su record (ad es. ATM, ecc.)	Quando è necessaria la scalabilità della risorsa di archiviazione e/o dell'elaborazione.
Quando è richiesto il 100% della conformità a SQL.	Quando vi sono necessità di elaborazione dati complessi con grandissimi volumi di dati.

Architettura per Hadoop

Due concetti fondamentali alla base di Hadoop:

- L'HDFS (Hadoop Distributed File System): un file system basato su Java che fornisce una risorsa di archiviazione di dati scalabili e affidabili che si estendono su grandi cluster di commodity server.
- MapReduce – un modello di programmazione che semplifica il compito di scrivere programmi che funzionano nell'ambiente di elaborazione parallelo.

Un cluster Hadoop operativo ha molti altri sottosistemi, ma HDFS e MapReduce sono centrali rispetto al modello di elaborazione.

HDFS

HDFS è un file system distribuito, scalabile e portatile scritto in Java. HDFS archivia grandi file (tipicamente nell'intervallo da gigabyte a terabyte) su più computer. Raggiunge l'affidabilità replicando i dati su più host. Per impostazione predefinita, i blocchi di dati vengono archiviati (replicati) su tre nodi, due sullo stesso scaffale e uno su uno scaffale diverso (un sovraccarico di 3 volte superiore alla risorsa di archiviazione non replicata). I nodi di dati possono comunicare ai dati di ribilanciamento, spostare le copie e mantenere elevata la replica dei dati.

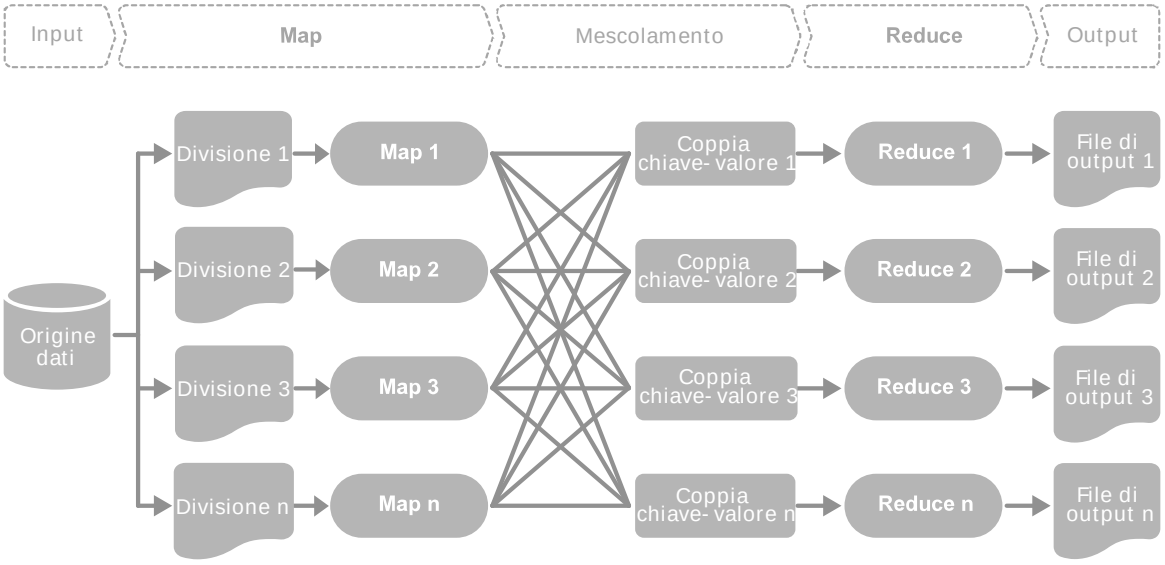
HDFS non è un file system completamente conforme a Posix ed è ottimizzato per la produttività. Alcune operazioni su file atomici sono proibiti o lenti. Non è possibile, per esempio, inserire i nuovi dati al centro di un file, sebbene sia possibile aggiungerli.

MapReduce

MapReduce è un framework di programmazione che, se seguito, rimuove la complessità dall'attività di programmazione negli ambienti a elevato parallelismo.

Un programmatore tipicamente deve scrivere due funzioni (una funzione Map e una funzione Reduce) e altri componenti nel framework Hadoop provvederanno alla tolleranza di errore, alla distribuzione, all'aggregazione, all'ordinamento e così via. L'esempio citato di solito è il problema di produrre un conteggio della frequenza di parole aggregate su un grande numero di documenti. Viene utilizzata la seguente procedura:

1. Il sistema suddivide i dati di input tra un numero di nodi denominati mapper. Il programmatore scrive una funzione che conta ogni parola in un file e quante volte si ripete. Questa è la funzione Mappa, l'output della quale è un insieme di coppie chiave-valore che includono una parola e un conteggio di parole. Ogni mapper lo fa al proprio set di documenti di input, in modo che nell'aggregato molti mapper producono molti set di coppie chiave-valore per la prossima fase.
2. Si verifica la fase di mescolamento, una funzione di hash coerente viene applicata alle coppie chiave-valore e i dati di output viene ridistribuita ai riduttori, in un modo tale che tutte le coppie chiave-valore con la stessa chiave vanno allo stesso riduttore.
3. Il programmatore ha scritto una funzione Reduce che, in questo caso, somma semplicemente le occorrenze della parola dai flussi in ingresso di coppie chiave-valore, scrivendo i totali in un file di output:



Questo processo isola il programmatore dai problemi di scalabilità man mano che cresce il cluster. Parte del sistema Hadoop gestisce il marshalling e l'esecuzione di risorse, questa parte è YARN se la versione di MapReduce è 2.0 o versioni successive.

Non vi è alcuna garanzia che questo processo intero sia più veloce di qualsiasi tipo di sistema alternativo (sebbene, in pratica, sia più veloce per alcuni tipi categorie di problemi e grandi volumi di dati). Il principale vantaggio di questo modello di programmazione è la capacità di sfruttare l'operazione di mescolamento spesso ottimizzata, dovendo al contempo scrivere le parti di map e reduce del programma.

L'ecosistema di Hadoop

Esistono diversi modi di interagire con un cluster Hadoop.

MapReduce Java

Questo è il metodo di accesso più flessibile e con migliori prestazioni, sebbene, dato che questo è il linguaggio di assemblaggio di Hadoop, possa comportare un ciclo di sviluppo.

Streaming di MapReduce

Consente lo sviluppo in Hadoop in qualsiasi linguaggio di programmazione scelto, al costo di piccole, fino a moderate, riduzioni delle prestazioni e della flessibilità. Dipende tuttavia dal modello di MapReduce, ma amplia il set di linguaggi di programmazione disponibili.

Crunch

Questa è una raccolta per pipeline di MapReduce a più stadi in Java, basata su FlumeJava di Google. Offre una API Java per attività come l'unione e l'aggregazione dei dati, che ha un'implementazione tediosa nel semplice MapReduce.

Pig Latin

Un linguaggio di alto livello (spesso denominato solo 'Pig') che è adatto per i carichi di lavoro del flusso di dati in batch. Con Pig non è necessario affatto pensare in termini di MapReduce. Apre il sistema a programmatori che non utilizzano Java e offre operazioni comuni come: unire, raggruppare, filtrare e ordinare.

Hive

Un interprete SQL (non conforme) che include un metastore che può eseguire il mapping dei file a loro schemi e serializzatori/deserializzatori associati. Poiché Hive è basato su SQL, i driver ODBC e JDBC consentono l'accesso a strumenti di business intelligence standard come Excel.

Oozie

Un motore per il flusso di lavoro di PDL XML che consente la creazione di un flusso di lavoro di processi composti da una delle modalità summenzionate.

HBase

Apache HBase è assegnato all'hosting di grandissimi tabelle, con miliardi di righe e milioni di colonne, su cluster di commodity hardware. Basata su Bigtable di Google, HBase offre funzionalità del tipo Bigtable su Hadoop e HDFS.

Zookeeper

Apache Zookeeper rappresenta l'impegno nello sviluppo e manutenzione di un server open-source che consenta un coordinamento distribuito molto affidabile. Zookeeper è un servizio centralizzato per mantenere le informazioni e la denominazione della configurazione, e per fornire la sincronizzazione distribuita e i servizi del gruppo.

Implementazione di WPS e Hadoop in Windows x64

Installazione di WPS in Windows x64

1. Prima di avviare l'installazione di WPS, assicurarsi che la copia di Windows abbia gli ultimi aggiornamenti e service pack applicati.
2. Le installazioni per workstation e server Windows utilizzano entrambe lo stesso software WPS, uso che viene controllato tramite un codice di licenza applicato con la procedura `setinit`.
3. È possibile scaricare il file di installazione di WPS per Windows dal sito Web di World Programming. Verrà richiesto un nome utente e una password per accedere alla sezione di download del sito.
4. Una volta scaricato il file di installazione (.msi), fare semplicemente doppio clic sul file, leggere e accettare l'EULA e seguire le istruzioni visualizzate.
5. Una volta installato il software WPS, sarà necessario applicare il codice di licenza. Il codice di licenza è stato inviato tramite posta elettronica al momento dell'acquisto del software WPS. Il modo più semplice di applicare il codice di licenza è quello di eseguire WPS Workbench come utente con accesso amministrativo al sistema, e seguire le istruzioni.
6. Questo conclude la configurazione di WPS.

Configurazione di Hadoop in Windows x64

Installazione di Hadoop

Se non è già stato fatto, installare Hadoop, facendo riferimento, se necessario, alla documentazione in dotazione con la propria distribuzione (ad esempio, Cloudera). Una volta installato Hadoop, è necessario procedere con i dettagli di configurazione delineati di seguito.

Nota:

A condizione che si disponga di una distribuzione che funziona in modalità Apache Hadoop standard, si applicano i dettagli di configurazione, anche se la distribuzione non è Cloudera. Non sono supportate le distribuzioni che disattivano o cambiano le funzionalità di Apache Hadoop standard.

File di configurazione

Tutte le chiamate a Hadoop di Cloudera 5 sono eseguite tramite Java e JNI. Sarà necessario ottenere e scaricare i file `.jar` del client Hadoop di Cloudera nel computer locale. I seguenti file contengono URL per vari servizi Hadoop ed è necessario configurarli in modo da corrispondere all'installazione attuale di Hadoop:

- core-site.xml
- hdfs-site.xml
- mapred-site.xml

Nota:

Se si utilizza un client Windows rispetto ad un cluster Linux, quest'ultimo file deve impostare il parametro di configurazione `mapreduce.app-submission.cross-platform` su `true`.

Consultare la documentazione di Hadoop per maggiori informazioni.

La variabile di ambiente CLASSPATH

È necessario configurare la variabile di ambiente `CLASSPATH` in modo che punti ai file del client Java di Cloudera. Questo varierà in base alla configurazione del client e del computer specifico, ma un possibile esempio potrebbe essere:

```
c:\Cloudera5\conf;c:\Cloudera5\*.jar
```

La variabile di ambiente HADOOP_HOME

In Windows, è necessario configurare la variabile di ambiente `HADOOP_HOME` in modo che punti ai file del client Cloudera Java. Prendendo l'esempio in alto, bisogna impostarla su: `C:\Cloudera5`.

Configurazione di Kerberos in Windows x64

Se la distribuzione di Hadoop include o impone il supporto per Kerberos, procedere con la *Configurazione di Kerberos e Hadoop sul client* (pag. 16).

Implementazione di WPS e Hadoop in Linux x64

Installazione di WPS in Linux x64

1. WPS è supportato in qualsiasi distribuzione di Linux, compatibile con LSB (Linux Standard Base) 3.0 o versioni successive. WPS è supportato in Linux utilizzato in x86, x86_64 e IBM System z che include IFL (Integrated Facility for Linux).
2. Se è installata una distribuzione di Linux a 64 bit, si ha l'opzione di utilizzare WPS a 32 o 64 bit. È necessario notare che alcune distribuzioni di Linux a 64 bit installano solo raccolte di sistema a 64 bit per impostazione predefinita. WPS a 64 bit, utilizzato in queste distribuzioni, sarà pronto per l'uso. Se, tuttavia, si sceglie di usare WPS a 32 bit, sarà necessario prima installare le raccolte di sistema a 32 bit. Consultare la documentazione sulla distribuzione di Linux per indicazioni sulle modalità di realizzazione.
3. WPS per Linux è disponibile attualmente solo come file di archivio tar compresso. Un programma di installazione della piattaforma basata su RPM nativo sarà messo a disposizione in futuro.
4. Il file compresso di WPS per Linux viene fornito in formato gzipped tar (`.tar.gz`) ed è possibile scaricarlo dal sito Web di World Programming. Verrà richiesto un nome utente e una password per accedere alla sezione di download del sito.
5. Per installare WPS, estrarre i file dal file compresso utilizzando gunzip e tar come segue. Scegliere un percorso di installazione idoneo in cui si dispone dell'accesso in scrittura e passare (`cd`) a quella directory. Il file compresso è totalmente autosufficiente e si può decomprimere dovunque. Il percorso dell'installazione può essere situato in una posizione che richieda l'accesso alla radice, come `/usr/local`, se s'installa per tutti gli utenti, o può essere situato nella home directory.
6. Decomprimere il file d'installazione digitando: `tar -xzof <file-installazione-wps>.tar.gz` o: `gzip -cd <file-installazione-wps>.tar.gz | tar xvf -`
7. Sarà necessario un codice di licenza per eseguire WPS. È possibile applicarlo dall'interfaccia utente grafica o dalla riga di comando avviando una delle due applicazioni come segue.
   a. Per avviare l'interfaccia utente grafica WPS Workbench, eseguire il seguente comando: `<directory-installazione-@breve-versione-completa-prodotto@-wps>/eclipse/workbench`. Il sistema aprirà una finestra di dialogo dove si può importare il codice di licenza.
   b. Per avviare WPS dalla riga di comando, eseguire il seguente comando: `<directory-installazione-@breve-versione-completa-prodotto@-wps>/bin/wps -stdio -setinit < <file-codice-wps>`. Un messaggio confermerà il completamento dell'applicazione della licenza.
8. Questo conclude la configurazione di WPS.

Configurazione di Hadoop in Linux x64

Installazione di Hadoop

Se non è già stato fatto, installare Hadoop, facendo riferimento, se necessario, alla documentazione in dotazione con la propria distribuzione (ad esempio, Cloudera). Una volta installato Hadoop, è necessario procedere con i dettagli di configurazione delineati di seguito.

Nota:

A condizione che si disponga di una distribuzione che funziona in modalità Apache Hadoop standard, si applicano i dettagli di configurazione, anche se la distribuzione non è Cloudera. Non sono supportate le distribuzioni che disattivano o cambiano le funzionalità di Apache Hadoop standard.

File di configurazione

Tutte le chiamate a Hadoop di Cloudera 5 sono eseguite tramite Java e JNI. Sarà necessario ottenere e scaricare i file `.jar` del client Hadoop di Cloudera nel computer locale. I seguenti file contengono URL per vari servizi Hadoop ed è necessario configurarli in modo da corrispondere all'installazione attuale di Hadoop:

- core-site.xml
- hdfs-site.xml
- mapred-site.xml

Consultare la documentazione di Hadoop per maggiori informazioni.

La variabile di ambiente CLASSPATH

È necessario configurare la variabile di ambiente `CLASSPATH` in modo che punti ai file del client Java di Cloudera. Per un esempio fittizio, si potrebbero aggiungere le seguenti righe al profilo utente (quale `.bash_profile`):

```
CLASSPATH=/opt/cloudera5/conf:/opt/cloudera5/*.jar
```

```
EXPORT CLASSPATH
```

Configurazione di Kerberos in Linux x64

Se la distribuzione di Hadoop include o impone il supporto per Kerberos, procedere con la *Configurazione di Kerberos e Hadoop sul client* (pag. 16).

Configurazione di Kerberos e Hadoop sul client

In Windows e Linux, potrebbe essere necessario eseguire prima il comando `kinit` e immettere la password al prompt. Questo può essere costituito dall'implementazione del SO di `kinit` (in Linux) o il binario `kinit` nella directory JRE all'interno di WPS.

In Windows:

- È necessario registrarsi come utente di Active Directory, non come utente del computer locale
- L'utente non può essere un amministratore locale del computer
- È necessario impostare la chiave del Registro di sistema per consentire a Windows di autorizzare l'accesso da parte di Java alla chiave della sessione di TGT:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Lsa\Kerberos\Parameters
Value Name: allowtgtsessionkey
Value Type: REG_DWORD
Value: 0x01
```

- È necessario installare JCE (Java Cryptography Extension) Unlimited Strength Jurisdicton Policy Files in JRE (vale a dire, la JRE presente all'interno della directory di installazione di WPS).

Quindi, sarà necessario configurare le varie entità Kerberos nei file di configurazione XML di Hadoop. Con Cloudera, sono disponibili tramite Cloudera Manager. L'elenco dei file di configurazione comprende:

- `dfs.namenode.kerberos.principal`
- `dfs.namenode.kerberos.internal.spnego.principal`
- `dfs.datanode.kerberos.principal`
- `yarn.resourcemanager.principal`
- `yarn.resourcemanager.principal`

Nota:

L'elenco riportato in alto non è completo e spesso può essere specifico del sito: le dichiarazioni di `libname` richiedono ulteriormente che il parametro `hive_principal` sia impostato su `hive_principal` del cluster Kerberos.

Esempi di codice relativi all'integrazione

Connessione ad Hive tramite SQL standard

```
libname lib hadoop schema=default server="clouderademo" user=demo
password=demo;

proc sql;
 drop table lib.people;
run;

data people1;
 infile 'd:\testdata.csv' dlm=',' dsd;
 input id $ hair $ eyes $ sex $ age dob :date9. tob :time8.;
run;

proc print data=people1;
 format dob mmddyy8. tob time8.;
run;

data lib.people;
  set people1;
run;

data people2;
  set lib.people;
run;

proc contents data=people2;
run;

proc print data=people2;
 format dob mmddyy8. tob time8.;
run;

proc means data=lib.people;
 by hair;
  where hair = 'Black';
run;
```

Connessione ad Impala tramite SQL standard

```
libname lib hadoop schema=default server="clouderademo" user=demo
password=demo port=21050 hive_principal=nosasl;

proc sql;
 drop table lib.peopleimpala;
run;
```

```
data people1;
 infile 'd:\testdata.csv' dlm=',' dsd;
 input id $ hair $ eyes $ sex $ age dob :date9. tob :time8.;
run;

proc print data=people1;
 format dob mmddyy8. tob time8.;
run;

data lib.peopleimpala;
  set people1;
run;

data people2;
  set lib.peopleimpala;
run;

proc contents data=people2;
run;

proc print data=people2;
 format dob mmddyy8. tob time8.;
run;

proc means data=lib.peopleimpala;
 by hair;
  where hair = 'Black';
run;
```

Connessione ad Hive tramite SQL passthrough

```
proc sql;
connect to hadoop as lib (schema=default server="clouderademo" user=demo
 password=demo);
    execute (create database if not exists mydb) by lib;
    execute (drop table if exists mydb.peopledata) by lib;
    execute (CREATE EXTERNAL TABLE mydb.peopledata(id STRING, hair STRING, eye
 STRING, sex STRING, age INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES
 TERMINATED BY '\n' STORED AS TEXTFILE LOCATION '/user/demo/test') by lib;
 select * from connection to lib (select * from mydb.peopledata);
  disconnect from lib;
quit;

/* options sastrace=,,,d; */
libname lib2 hadoop schema=mydb server="clouderademo" user=demo password=demo;
data mypeopledata;
  set lib2.peopledata;
run;

proc print data=mypeopledata;
run;
```

Connessione ad Impala tramite SQL passthrough

```
proc sql;
  connect to hadoop as lib (schema=default server="clouderademo" user=demo
 password=demo port=21050 hive_principal=nosasl);
    execute (create database if not exists mydb) by lib;
```

```
    execute (drop table if exists mydb.peopledataimpala) by lib;
    execute (CREATE EXTERNAL TABLE mydb.peopledataimpala(id STRING, hair STRING, eye
 STRING, sex STRING, age INT) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' LINES
 TERMINATED BY '\n' STORED AS TEXTFILE LOCATION '/user/demo/test') by lib;
 select * from connection to lib (select * from mydb.peopledataimpala);
  disconnect from lib;
quit;

libname lib2 hadoop schema=mydb server="clouderademo" user=demo password=demo
 port=21050 hive_principal=nosasl;
data mypeopledata;
  set lib2.peopledataimpala;
run;

proc print data=mypeopledata;
run;
```

Esecuzione di comandi HDFS e script Pig tramite WPS

Esempio di codice WPS

```
filename script 'd:\pig.txt';
proc hadoop options='d:\hadoop.xml' username = 'hdfs' verbose;
       hdfs delete='/user/demo/testdataout' recursive;
run;

proc hadoop options='d:\hadoop.xml' username = 'demo' verbose;
       pig code = script;
run;

proc hadoop options='d:\hadoop.xml' username = 'demo';
       hdfs copytolocal='/user/demo/testdataout/part-r-00000' out='d:\output.txt'
 overwrite;
run;

data output;
 infile "d:\output.txt" delimiter='09'x;
 input field1 field2 $;
run;

proc print data=output;
run;
```

Esempio di codice Pig

```
input_lines = LOAD '/user/demo/test/testdata.csv' AS (line:chararray);
-- Extract words from each line and put them into a pig bag
-- datatype, then flatten the bag to get one word on each row
words = FOREACH input_lines GENERATE FLATTEN(TOKENIZE(line)) AS word;

-- filter out any words that are just white spaces
filtered_words = FILTER words BY word MATCHES '\\w+';

-- create a group for each word
word_groups = GROUP filtered_words BY word;

-- count the entries in each group
word_count = FOREACH word_groups GENERATE COUNT(filtered_words) AS count, group AS
 word;
```

```
-- order the records by count
ordered_word_count = ORDER word_count BY count DESC;
STORE ordered_word_count INTO '/user/demo/testdataout';
```

Uso di WPS con Hadoop Streaming

Hadoop Streaming è una utilità in dotazione con la distribuzione di Hadoop. L'utilità consente la creazione e l'esecuzione di processi di MapReduce con qualsiasi file eseguibile o script come un mapper e/o un riduttore.

La sintassi della struttura è la seguente:

```
$HADOOP_HOME/bin/hadoop jar $HADOOP_HOME/hadoop-streaming.jar \
    -input myInputDirs \
    -output myOutputDir \
    -mapper /bin/cat \
    -reducer /bin/wc
```

I mapper e riduttori ricevono il loro input e output in `stdin` e `stdout`. La vista dei dati è orientata dalle righe e ogni riga viene elaborata come coppia chiave-valore separata dal carattere di tabulazione.

È possibile utilizzare Hadoop Streaming per sfruttare la potenza di WPS in modo da distribuire programmi scritti in linguaggio di SAS su molti computer in un cluster Hadoop, come nell'esempio del processo MapReduce riportato di seguito.

Nota:

Data la vasta distribuzione di programmi, qualsiasi esempio necessariamente utilizza un approccio non tradizionale per il linguaggio di SAS, in quanto ogni mapper e riduttore solo rileva un subset limitato di dati.

Prima di procedere, assicurarsi di avere familiarità con i concetti HDFS e MapReduce in *Architettura per Hadoop* (pag. 8).

Il seguente esempio mostra la creazione ed esecuzione di un processo MapReduce per produrre i conteggi di parole che vengono visualizzate nei file di testo, nella directory fornita come input al processo MapReduce. Ogni singolo conteggio di una parola viene elaborato come la coppia chiave-valore `<parola><tabulazione>1`.

1. Assicurarsi di aver configurato la directory `input` in HDFS, per esempio utilizzando `hadoop fs -mkdir /user/rw/input` e che i file di testo contenenti le parole da contare siano stati aggiunti alla directory. Ogni cluster può rilevare questa directory.
2. Assicurarsi che WPS sia stato installato nello stesso percorso su ciascun nodo del cluster, in modo che qualsiasi mapper e riduttore possa richiamarlo.

3. Creare un programma mapper denominato **map.sas**:

```
options nonotes;

data map;
 infile stdin firstobs=1 lrecl=32767 encoding='utf8' missover dsd;
 informat line $32767.;
 input line;
 do i=1 by 1 while(scan(line, i, ' ') ^= '');
  key = scan(line, i,' ');
  value = 1;
  drop i line;
  output;
 end;
run;

proc export data=map outfile=stdout dbms=tab replace;
 putnames=no;
run;
```

4. Creare uno script denominato **map.sh** per chiamare **map.sas**:

```
#!/bin/bash
/opt/wps/bin/wps /home/rw/map.sas
```

5. Creare un programma riduttore denominato **reduce.sas**:

```
options nonotes;

data reduce;
 infile stdin delimiter='09'x firstobs=1 lrecl=32767 missover dsd;
 informat key $45.;
 informat value best12.;
 input key value;
run;

proc sql;
 create table result as select key as word, sum(value) as total from reduce
 group by key order by total desc;
quit;

proc export data=result outfile=stdout dbms=tab replace;
 putnames=no;
run;
```

6. Creare uno script denominato **reduce.sh** per chiamare **reduce.sas**:

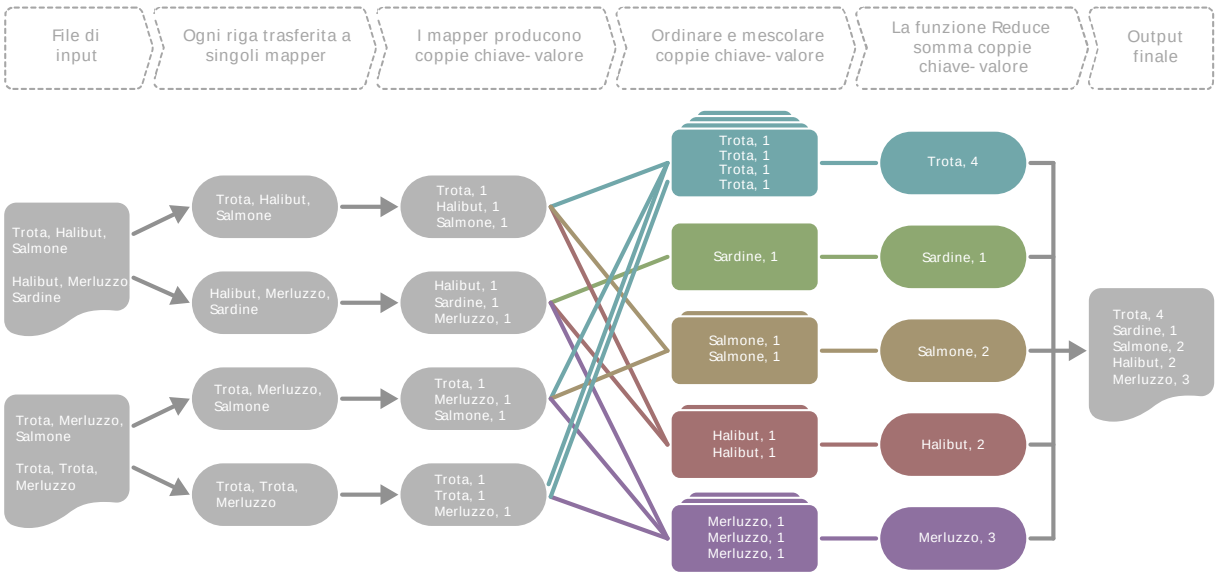
```
#!/bin/bash
/opt/wps/bin/wps /home/rw/reduce.sas
```

7. Assicurarsi che **map.sh**, **map.sas**, **reduce.sh** e **reduce.sas** siano copiati nello stesso percorso di ogni nodo del cluster, in modo che i mapper e i riduttori si possano eseguire quando è necessario.
8. Assicurarsi che la variabile di ambiente `CLASSPATH` sia stata configurata nel computer client per il proprio sistema operativo, secondo *Configurazione di Hadoop in Windows x64* (pag. 12) o *Configurazione di Hadoop in Linux x64* (pag. 15).

9. Eseguire la seguente riga di comando da un computer client con un client Hadoop installato, regolando i numeri di versione se appropriato:

```
hadoop jar hadoop-streaming-2.5.0-cdh5.3.2.jar -input input -output output -
mapper "/home/rw/map.sh" -reducer "/home/rw/reduce.sh"
```

L'esecuzione del comando ha l'effetto di avviare il processo MapReduce nel cluster specifico. Ogni istanza di un mapper (laddove è lo script **map.sh** in un nodo particolare che richiama **map.sas**) produce un set di coppie chiave-valore che comprendono ciascuno una parola e un conteggio di 1. La fase di mescolamento quindi si verifica con le coppie chiave-valore con la stessa chiave che vanno allo stesso riduttore. Ogni istanza di un riduttore (laddove è lo script **reduce.sh** in un nodo particolare che richiama **reduce.sas**) somma le occorrenze della parola per la chiave specifica in un file di output. L'output risultante è una serie di parole e conteggi ad esse associati. Un esempio di questo processo è riportato di seguito:



Nota:

L'output finale può causare la suddivisione in più di un file all'interno della directory di output, in base alla configurazione del cluster.

Riferimento

Le definizioni dei diagrammi sintattici sono notazioni che aiutano a spiegare la sintassi dei linguaggi di programmazione, e sono utilizzate nella presente guida per descrivere la sintassi del linguaggio.

Come leggere gli diagrammi sintattici

I diagrammi sintattici sono una notazione di sintassi grafica che accompagna le strutture del linguaggio significative, quali procedure, istruzioni e così via.

La descrizione di ciascun concetto linguistico comincia con la diagramma sintattico.

Immissione di testo

Il testo che è necessario immettere esattamente nel modo in cui è visualizzato, viene mostrato con un carattere macchina da scrivere:

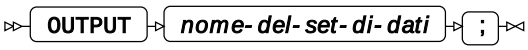
Questo esempio descrive un frammento di sintassi in cui la parola chiave `OUTPUT` viene seguita da un carattere punto e virgola: `;` . La forma del diagramma sintattico è: 

.

Generalmente le maiuscole/minuscole del testo non sono importanti, ma in questo contesto, è consuetudine utilizzare le maiuscole per le parole chiave.

Elementi segnaposto

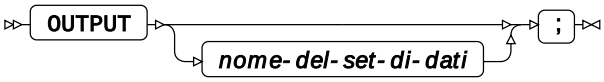
I segnaposto, che bisogna sostituire con testo pertinente e dipendente dal contesto, sono rappresentati in un carattere minuscolo e corsivo:



Qui, è necessario immettere la parola chiave `OUTPUT` letteralmente, ma bisogna sostituire il *nome-del-set-di-dati* con qualcosa di appropriato per il programma, in questo caso, il nome del set di dati a cui aggiungere una osservazione.

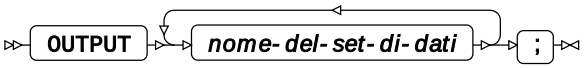
Opzionalità

Quando gli elementi sono opzionali, vengono visualizzati su un ramo sotto la linea principale nei diagrammi sintattici. L'opzionalità è rappresentata da un percorso alternativo senza ostacoli attraverso il diagramma:



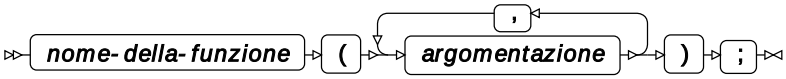
Ripetizione

Nei diagrammi sintattici, la ripetizione viene raffigurata con un cappio di ritorno che specifica opzionalmente il separatore che bisogna collocare tra istanze multiple.



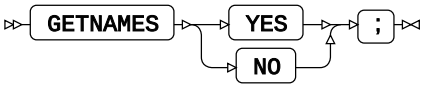
In alto, è necessario immettere la parola chiave `OUTPUT` letteralmente e farla seguire da una o più ripetizioni del *nome-del-set-di-dati*; in questo caso, non è stato necessario nessun separatore, tranne uno spazio.

L'esempio in basso dimostra l'uso di un separatore.



Opzioni

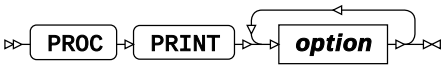
Nei diagrammi sintattici, l'opzione è illustrata mediante diversi rami paralleli.



Nell'esempio illustrato in alto, è necessario immettere la parola chiave `GETNAMES` letteralmente e poi la parola chiave `YES` o la parola chiave `NO`.

Frammenti

Quando la sintassi è troppo complicata per rientrare in una definizione, si potrebbe suddividere in frammenti:



option

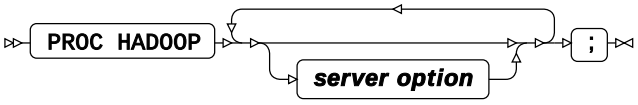
In alto, l'intera sintassi è suddivisa in frammenti di diagrammi sintattici. Il primo indica che `PROC PRINT` deve essere seguito da una o più istanze di un'*opzione*, ciascuna delle quali deve aderire alla sintassi fornita nel secondo diagramma.

Procedura HADOOP

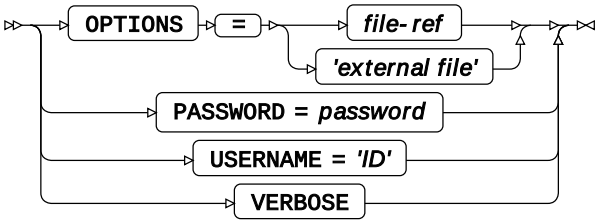
Istruzioni supportate

PROC HADOOP

Accede ad Hadoop tramite WPS.

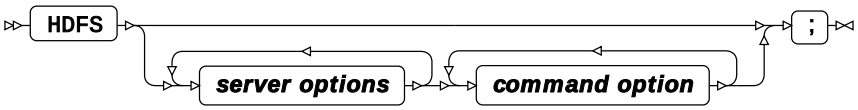


server option

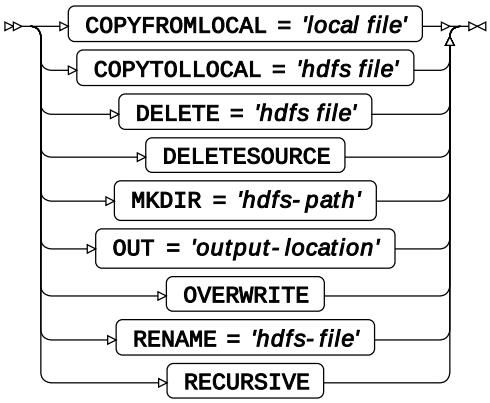


HDFS

Specifica il file system Hadoop distribuito da usare.

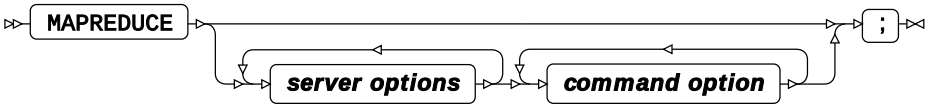


command option

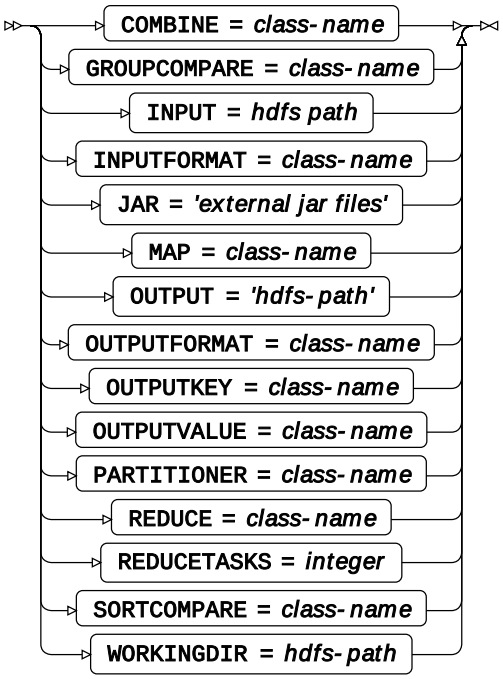


MAPREDUCE

Avvia processi MapReduce.

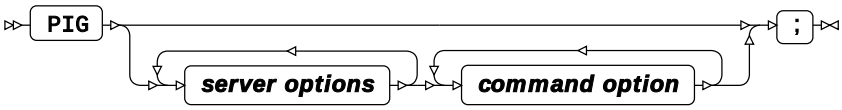


command option

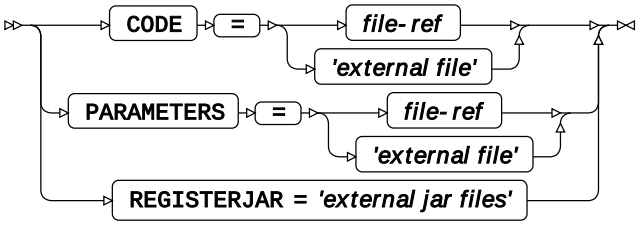


PIG

Consente l'invio di file esterni ad un cluster.



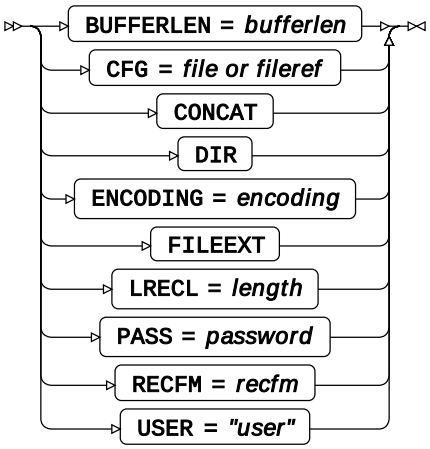
command option



Istruzioni globali

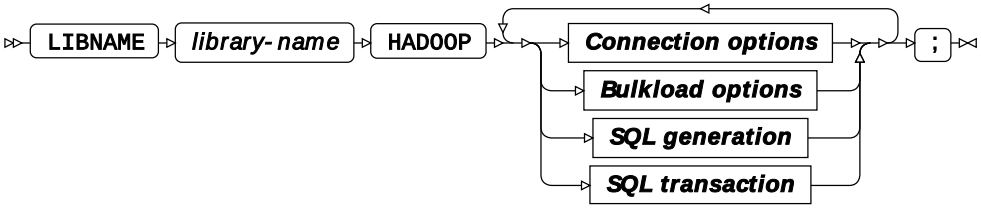
Metodo di accesso di Hadoop, FILENAME

hadoop-option



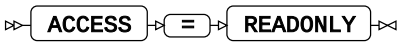
Motore WPS per Hadoop

HADOOP

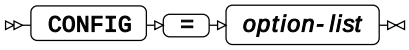


Connection options

ACCESS

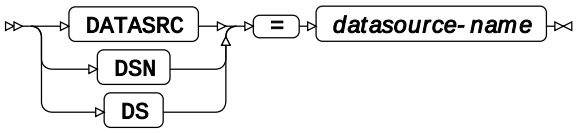


CONFIG



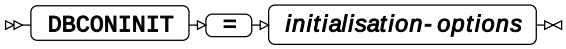
Type: String

DATASRC



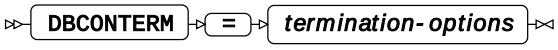
Type: String

DBCONINIT



Type: String

DBCONTERM

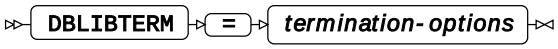


Type: String

DBLIBINIT

Type: String

DBLIBTERM



Type: String

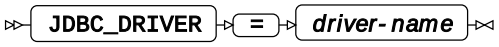
HIVE_PRINCIPAL

Type: String

JDBC_CONNECTION_STRING

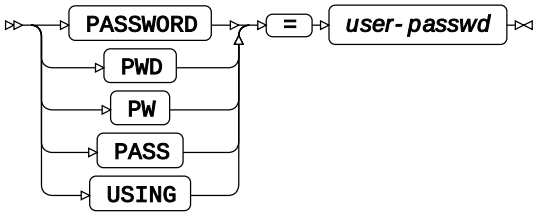
Type: String

JDBC_DRIVER



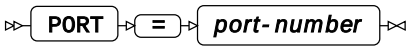
Type: String

PASSWORD



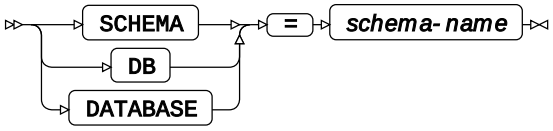
Type: String

PORT



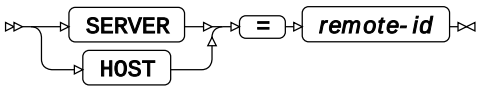
Type: Numeric

SCHEMA



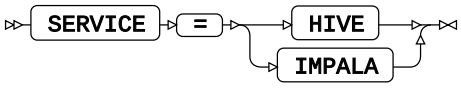
Type: String

SERVER

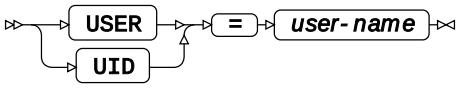


Type: String

SERVICE



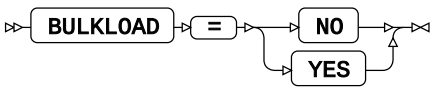
USER



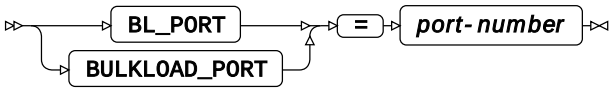
Type: String

Bulkload options

BULKLOAD



BL_PORT



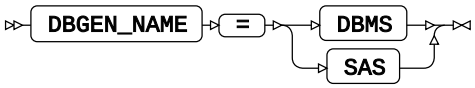
Type: Numeric

SQL generation

DBCREATE_TABLE_OPTS

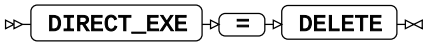
Type: String

DBGEN_NAME

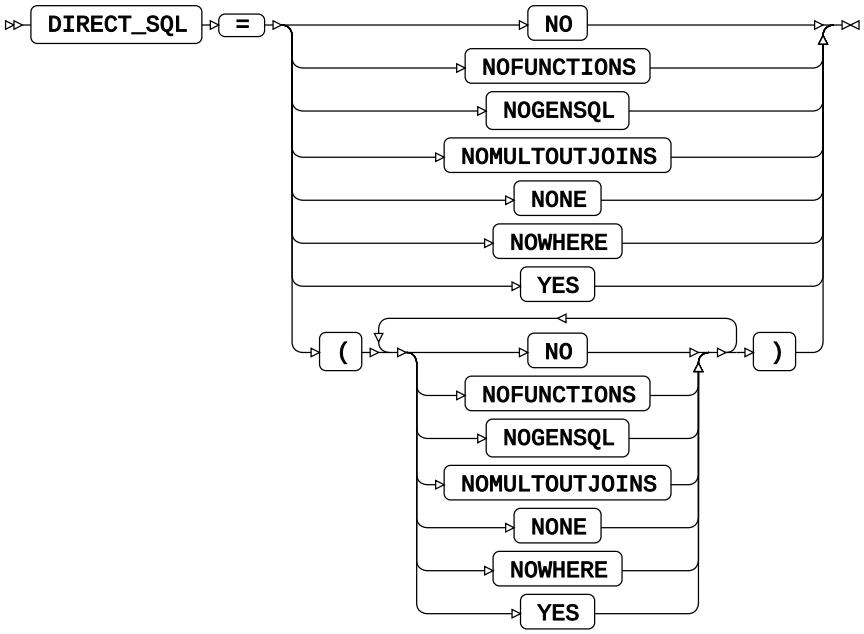


Default value: `SAS`

DIRECT_EXE

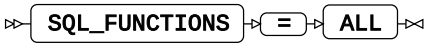


DIRECT_SQL

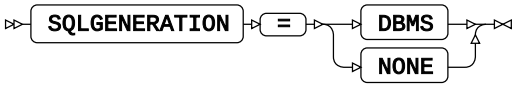


Default value: `YES`

SQL_FUNCTIONS

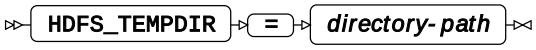


SQLGENERATION



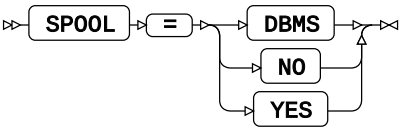
SQL transaction

HDFS_TEMPDIR



Type: String

SPOOL



Avvisi legali

Marchi commerciali

Avvisi generali