

ctrlNodes e ctrlClus

Nodes Control e Clusters Control

2025 04 04
ver.1.20

- **ORIGIN OF THE IDEA - CASE STUDY**
- **WHAT IT IS ABOUT**
- **PURPOSE**
- **DETAILED ARCHITECTURE**

ctrlNodes e ctrlClus

Nodes Control e Clusters Control

ORIGIN OF THE IDEA - CASE STUDY

Nel 2022, su 2 datacenter di un cliente in ambito bancario.

Dopo 1 mese di continui allarmi di "simultanei down di nodi cassandra".

ctrlNodes

Ho sviluppato un'architettura applicativa modulare per il monitoraggio dei nodi di cluster db. Messa in uso sui nodi di alcuni cluster cassandra di questo cliente.

La causa scatenante l'allarme è stata individuata al primo evento successivo alla messa in uso di questo sistema di controllo.

QUAL'ERA IL PROBLEMA?

Di sicuro non era un problema di:

- a) *Settaggio di parametri dei nodi o del cluster cassandra, quindi del DB*
- b) *Memoria*
- c) *Network*
- d) *Query*
- e) *Funzionamento del Cluster*

Dopo che trovi la causa capisci che la soluzione per trovarla è banale.

ctrlClus

- 1) Creare delle insert select che corrispondono alle indagini per periodo
- 2) Impaginare i risultati di una 10na di query sql e delle query al punto 1.
- 3) Notificare i risultati degli eventi elaborati al punto 1

Quanto descritto di colore grigio è da siluppare.

ctrlNods

Nodes Control

WHAT IT IS ABOUT

A Bash script application installed on each node of the Cassandra cluster

PURPOSE

1. Performs **CHECKS**
2. **LOGS** state changes
3. **SAVES events** to a local database (SQLite)
4. **SYNCHRONIZES** data to a central database for ctrlClus
5. **SENDS ALERTS** via Teams chat / email / SMS / email depending on the event

1. I MODULI DI CONTROLLO:

a) generic

1. S_DISK.sh
2. S_NMAP.sh
3. S_PING.sh

b) cassandra

1. S_Balancing.sh
2. S_CL_ClusterState.sh
3. S_CL_QueryLatency.sh
4. S_CPU.sh
5. S_MEM.sh
6. S_QueryQueue.sh

c) test

1. S_Partition.sh

2. LOGS WHAT

State changes:

- **ON/OFF**
- **AVAIABLE/UNAVAIABLE**
- **DOWN/UP**
- **UNDER_100/UPPER_100**
- **etc...**

WHAT IT DOES WITH THE LOGS:

- a) **Saves events to 2 files** (`data.log` and `data.sqlite`) located on a portion of disk in the RAM memory (ramdisk) of each client (mounted as tmpfs).
- b) Periodically synchronizes the **saved events to a central database** (MySQL, REDIS).

ctrlNods architecture

SETUP - LOG FILE HISTORY - SYNC	
00_POSTreboot.sh	ramdisk setup
01_updatefs.sh	Log file history
02_syncDB.sh	Sync Local DB vs Centralized DB

APPLICATION	
M_chk.sh	Application Launch
M_config.sh	Settings
M_control.sh	Application Logic

CONTROL MODULES	
/modules/generic S_*	Generic Modules
/modules/cassandra/S_*	Cassandra Modules
/modules/test S_*	Test Modules

LOG FILE	
/log/data.log	eventi in formato txt
/log/data.sqlite	eventi su db
UP_\$module_\$nod.ok (flag servizio UP)	flag file servizio up

BINARY FILE	
/bin/*	dd ping nmap sqlite3 sqldiff SchemaDB.sql <i>Script creazione DB locale (SQLite)</i>

ctrlNods

generic modules

checks to localhost

S_DISK.sh	
PURPOSE	I/O Disk
CONTROLS	/dev/zero of=/opt/ctrlNods/testfile_100MB bs=1024 count=402400
	Se la velocità disco diventa > xx

S_CPU.sh	
PURPOSE	stress java CPU
CONTROLS	top -b -n 1 grep java awk '{print \$9}'
	Se la CPU supera xx%, il nodo potrebbe essere sovraccarico

to checks against the other nodes of the cluster

S_PING.sh	
PURPOSE	Ping STATUS
CONTROLS	Tutti i nodi fanno un ping di tutti i nodi

S_NMAP.sh	
PURPOSE	Test service
CONTROLS	7001 7199 9142
	7001 TCP Comunicazione tra nodi con cifratura SSL abilitata
	7199 TCP JMX (Java Management Extensions) per il monitoraggio e la gestione di Cassandra
	9142 TCP Porta CQL per connessioni SSL/TLS

ctrlNods cassandra modules

checks to localhost

S_HINTS.sh	
PURPOSE	Verifica la presenza di Hints files
CONTROLS	Data creazione di quale nodo per quale KEYSPACE per quale TABLES

to checks against the cluster nodes

S_QueryQueue.sh	
PURPOSE	Thread pool delle query
CONTROLS	<i>nodetool tpstats awk 'NR>1 && (\$3+0 > 0 \$5+0 > 0) {print "Attenzione:", \$1, "Pending:", \$3, "Blocked:", \$5}'</i>
	<i>Pending → Query in attesa di essere elaborate Blocked → Query bloccate indica un forte carico del nodo</i>

S_Balancing.sh	
PURPOSE	Streaming e bilanciamento del cluster
CONTROLS	<i>nodetool netstats grep -E 'Receiving from Sending to'</i>
	Controlla lo stato del trasferimento dati tra i nodi Se vedi ***Receiving from*** o ***Sending to*** , significa che un nodo sta ricevendo o inviando dati. Se lo streaming è troppo lungo, il nodo potrebbe essere sovraccarico.

S_CL_ClusterState.sh	
PURPOSE	Cluster State
CONTROLS	<i>nodetool status grep -E '^D ^UJ ^UM ^UL'</i>
	<i>D → Down (nodo non raggiungibile) UJ → Joining (nodo si sta unendo, ma se persiste è un problema) UM → Moving (nodo sta spostando i dati, verifica se dura troppo) UL → Leaving (nodo sta lasciando il cluster, ma deve completare il processo)</i>

S_CL_QueryLatency.sh	
PURPOSE	Query Latency
CONTROLS	<i>nodetool proxyhistograms</i>
	<p><i>Lo script definisce delle soglie massime per le latenze di lettura, scrittura e range. Se un valore supera questa soglia, viene segnalato come criticità.</i></p> <p><i>#Estrazione dei valori: Lo script esegue il comando <code>nodetool proxyhistograms</code> e analizza l'output utilizzando <code>grep</code> e <code>awk</code> per estrarre i valori di latenza dei percentili 50%, 75%, 95% e 99% per ciascuna metrica di latenza.</i></p> <p><i>#Funzione <code>check_criticality</code>: La funzione confronta il valore di latenza con la soglia definita. Se il valore supera la soglia, stampa un messaggio di avviso.</i></p> <p><i>#Controllo minimi e massimi: Lo script può anche verificare i valori minimi e massimi per ogni metrica di latenza, segnalando eventuali anomalie nei picchi di latenza.</i></p>

S_Partition.sh	
PURPOSE	Large Partitions
CONTROLS	<i>nodetool tablehistograms keyspace table</i>
	<p>Cicla per tutte le tabelle di tutti i keyspaces</p> <p>Controlla la dimensione delle partizioni</p> <p>Se ci sono partizioni molto grandi (> XXMB), possono rallentare le query.</p>

S_MEM.sh	
PURPOSE	Memoria e Garbage Collection
CONTROLS	<i>nodetool gcstats</i>
	<p>Controlla la frequenza del Garbage Collector</p> <p>Min` / `Max` / `Mean`** → Durata delle operazioni di GC</p> <p>Count`** → Numero totale di GC eseguiti</p> <p>Se il **tempo massimo del GC** è elevato (> 1 sec), Cassandra potrebbe avere problemi di memoria</p>

ctrlClus

Clusters Control

DI COSA SI TRATTA

Un applicativo in python script installato su un server visibile da tutti i nodi del cluster.

SCOPO:

Utilizza il DB CENTRALE

1. **ELABORA** i dati del database centrale che viene alimentato periodicamente da ciascun nodo di ciascun cluster.
2. **NOTIFICA** via chat teams/email/sms/email l'elaborazione degli allarmi ricevuti in un lasso di tempo.
3. **VISUALIZZA** i risultati delle domande che gli vengono chieste

- **Indisponibilità dei nodi:**

- E caduto un nodo?

- Quando?

- Quante volte è successa la stessa cosa a quel nodo e agli altri?

- Perché è successo? (rete, disco, ram, query, datacenter, computer, cpu)

- In quel momento altri nodi erano down?

- I settaggi sono corretti?