START SLAVE;

Impicci e imbrogli con la replicazione di MySQL, per creare mostri a tre teste.

Salve!

Sono *Francesco Furiani*

A quanto pare mi tocca fare una presentazione per questo PiTech.

Quando non sono a fare presentazioni (o alla PS4) faccio il C.T.O. a ClickMeter.

Se volete contattarmi cercatemi su Twitter @ilfurio oppure lì.

1

MySQL & Replica

Manco le basi del mestiere... Arfio

MySQL

RDBMS più usato al mondo

- Open
- ACID
- Quasi SQL-99
- Facile da usare
- Difficile da capire

Replicazione

La replicazione, consente di avere gli stessi dati su più server.

Utile in vari scenari:

- Failover
- Scalabilità
- High Availability
- Backup
- Magia oscura ...

Replicazione - Le basi

Binary log

contiene gli eventi di cambiamento

Relay log

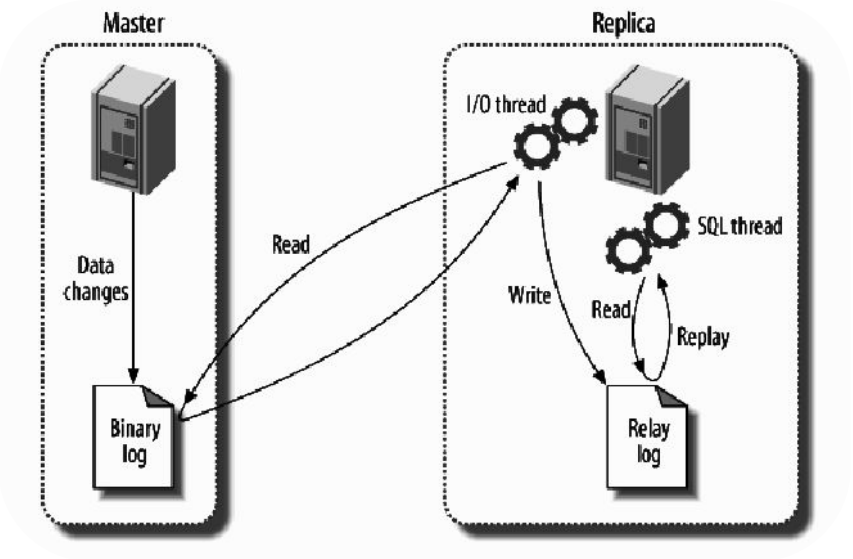
contiene alcuni eventi di cambiamento

I/O thread

copia gli eventi tra master e replica

SQL thread

applica eventi su replica



La replicazione è asincrona

Replicazione - Tipi

Statement

Esecuzione dello statement sulle repliche

Row

Esecuzione degli effetti degli statement sulle repliche

Mixed

Un po statement, un po row

Replicazione - Tipi

Statement

Pro:

- Data transfer

Contro:

- Divergenza con alcuni statement
- (*Pre 5.6*) Lentezza nell'esecuzione

Row

Pro:

- No divergenza
- Esecuzione immediata

Contro:

- (Pre 5.6) Data transfer
- Generazione più eventi
- DDL in statement

Mixed

Pro:

- Combinazione dei pro precedenti

Contro:

- Non sempre supportato dallo storage engine
- Binlog un po più caotico

2

Scenari

Fatto 30, famo 4294967296

Failover

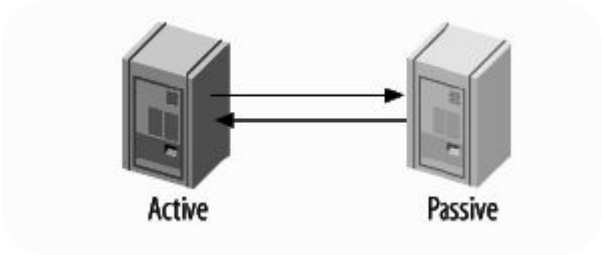
Tutto crasha ;(

So “problemi” per il database di produzione.

La replicazione aiuta con creazione di un hot spare “sempre” in sync.

Setup in 4 mosse:

1. i server a t0 hanno gli stessi dati
2. configurazione replicazione con server id e aggiornamenti dello “slave” (*log_slave_updates*)
3. server come slave uno dell’altro
4. HAProxy on top



Scalabilità - Lettura

Tante operazioni in lettura possono causare problemi ad un singolo nodo.

Una soluzione è, al solito, più CPU / RAM / RAID.

Un'altra è usare la replicazione di MySQL per creare delle copie del database principale da cui leggere.

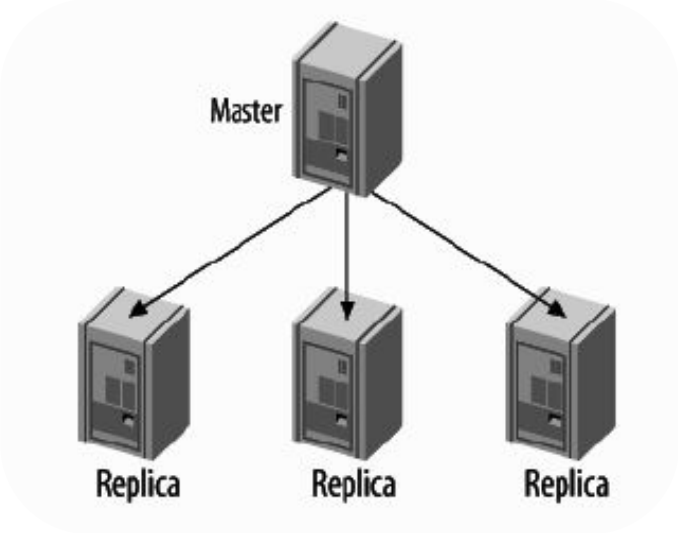
Scalabilità - Lettura

Setup simile al failover.

Le repliche non inviano eventi indietro al master.

Performance dipendenti dal carico di lavoro e dalla modalità di replicazione.

Possibile utilizzo della tecnologia “Blackhole”.



Scalabilità - Lettura - Proxy

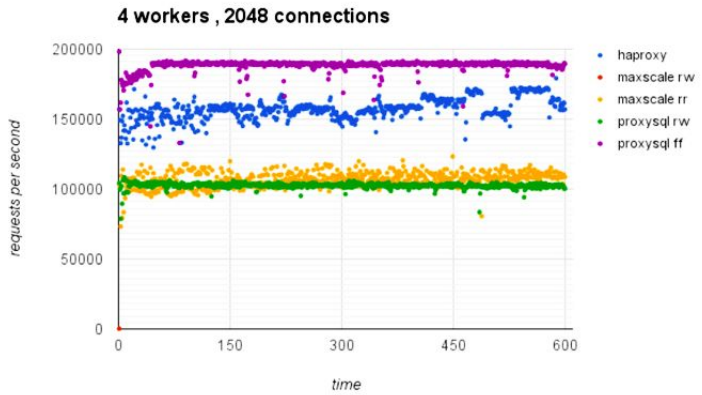
Questa soluzione richiede il cambiamento della logica applicativa.

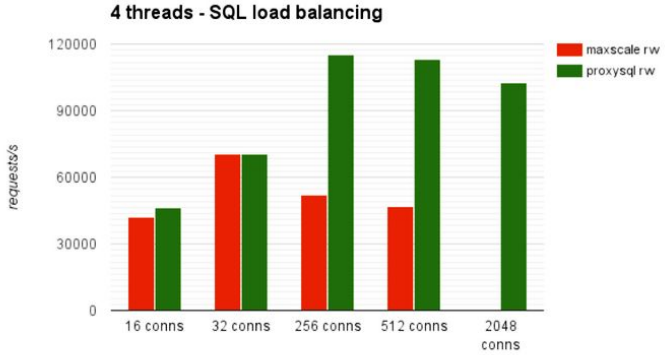
Ma anche no, viva i proxy ;)

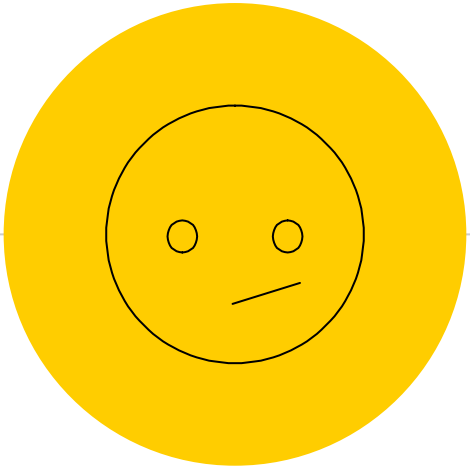
- Oracle MySQL Proxy
  - Scripting in LUA per decidere come gestire il traffico
- ProxySQL
  - Espressioni regolari per spostare il flusso delle query/statement
  - Server config aware
- MariaDB MaxScale
  - Regole e plugin per redigere il traffico anche su backend atipici

Scalabilità - Lettura - Proxy

Proxy benchmarks







E le scritture?

So’ bboni tutti senno’

Scalabilità - Scrittura

Un bel problema ...

Siamo limitati da tante problematiche:

- Consenso
- Conflitti
- Asincronia

Non si può fare in maniera generale, ma in alcuni casi ...

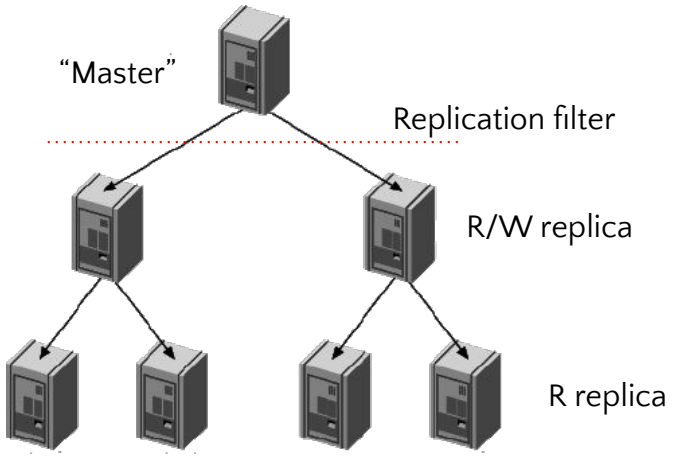
Scalabilità - Scrittura

Possiamo partizionare i dati?
Le partizioni sono indipendenti?

Si può creare una topologia dove gli eventi di replicazione vengono filtrati.

Possiamo quindi scrivere sulle repliche senza preoccuparci.

Oppure un proxy si preoccuperà per noi.



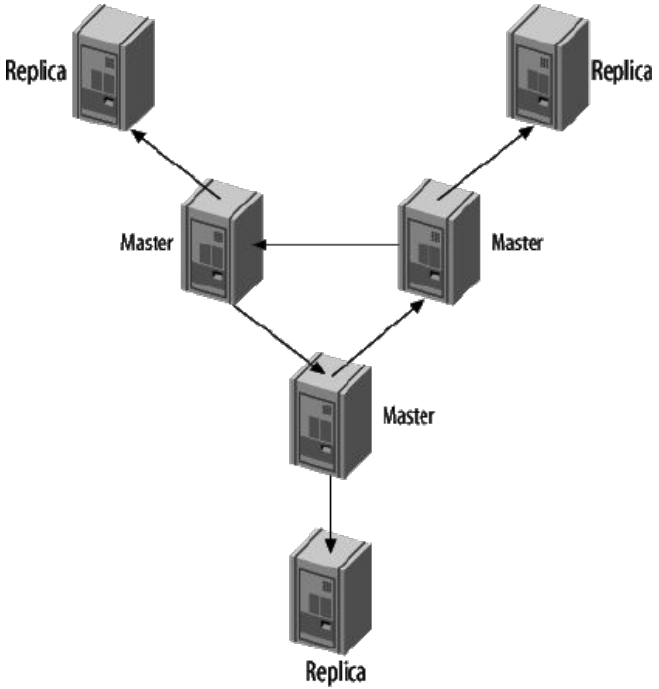
Scalabilità - Scrittura

Si possono ipotizzare altre topologie simili.

Le topologie stesse potrebbero essere a rischio a causa della loro stessa struttura.

La modalità asincrona di replicazione potrebbe ostacolarci.

Una soluzione è MySQL Cluster.



High Availability

La replicazione asincrona può soffrire di lag:
consistenza dei dati tra differente repliche non è garantita.

Per alcuni potrebbe essere OK, grazie a tool esterni (*mysql-master-ha o mysqlfailover*) si sceglie tra gli slave lo stato consistente più nel futuro (possibile data loss).

In HA è totalmente inaccettabile, ci sono quindi soluzioni:

- Galera (*patch set*)
- Semi-sync (post *MySQL 5.6*) + (*Fabric* o *MySQL Failover*)
- MySQL Cluster

HA - MySQL Cluster

MySQL Cluster è una versione NDB-based di MySQL.

NDB è uno “storage” multi nodo che di base supporta HA, replication, multi-master, auto partition (e tante altre feature).

Non è quindi lo standard MySQL, ma un sistema diverso su cui il layer SQL è on top.

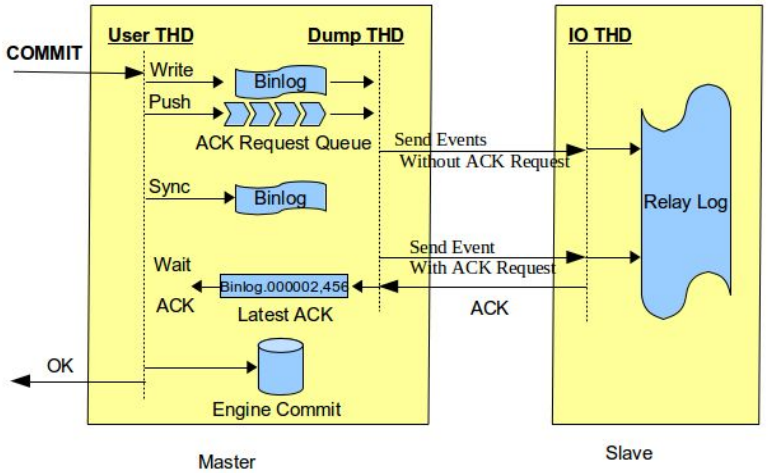
Per divertirvi ed imparare:
https://github.com/renecannao/mysql-cluster-tutorial

HA - Semi-sync

Questa feature ci garantisce che almeno uno slave stia in sync con il master.

Il binlog del master in caso di crash può ancora contenere transazioni non replicate ma non committed, quindi in caso di restore deve essere purificato :)

In genere questo viene fatto automaticamente da tool utilizzati per l'HA, come ad esempio mysqlfailover o MySQL Fabric (*).



HA - MySQL Fabric

MySQL Fabric è un processo che riceve connessioni XML-RPC e informa i client di quale dei gruppi HA che gestisce è ok e redirige la connessione a seconda del tipo di richiesta (WRITE / READ).

Configurabile tramite linea di comando e permette gestire vari cluster e anche permette di semi-automatizzare lo sharding delle tabelle con limitazioni (non si possono fare query cross-shard).

Ovviamente nella gestione dei cluster sono incluse tutte le procedure di failover automatico, solo se il Global Transaction ID è abilitato.

HA - Galera

Galera è una soluzione sviluppata per trasformare MySQL in un sistema nativo H.A. senza ricorrere a diverse soluzioni insieme:

- Multi-master
- Replicazione sincrona
- Nessuna divergenza tra nodi
- Multi threaded slave
- Provisioning dei nodi automatico
- Nessuno split tra letture e scritture

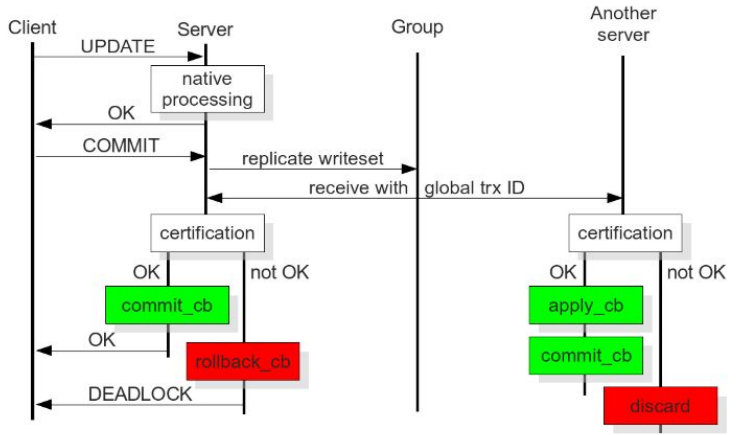
HA - Galera

Consistenza garantita grazie ad un sistema di certificazione e di GTID.

Post commit su un nodo vengono replicate le certificazioni del writeset interessato.

Se la certificazione passa, le scritture vengono replicate e la modifica è resa visibile (*), altrimenti rollback.

ORM: Al commit si potrebbe ricevere errore di deadlock.

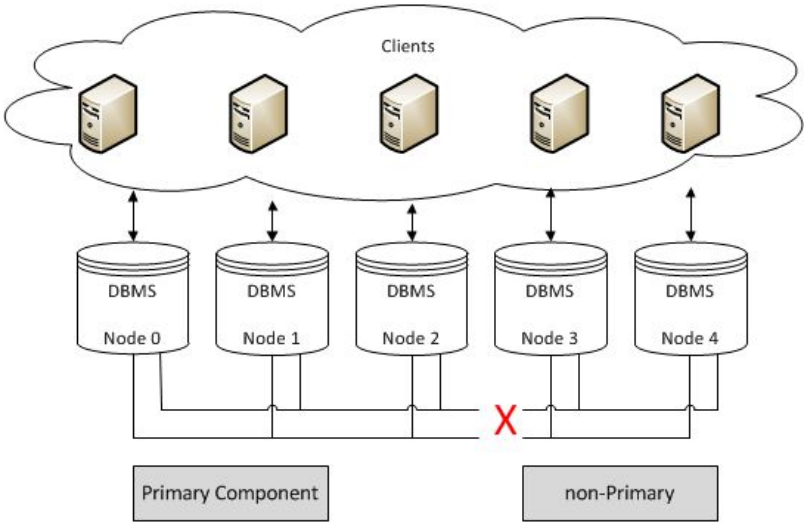


HA - Galera

Il cluster usa un sistema di quorum pesato per mantenere consistenza.

In caso di partition, solo un componente risponderà gli altri si rifuteranno di rispondere ai client.

Alla riconnessione i nodi (grazie al GTID) verificano lo stato e tramite IST o SST si ripristina la consistenza.

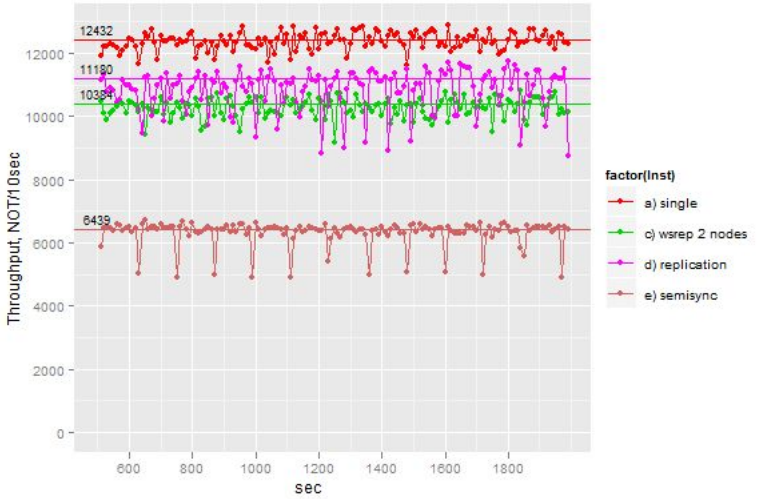


HA - Galera

Le performance di questo tipo di replicazione sono comparabili alla replicazione asincrona (e nel test l'asincrona è andata in slave lag).

Le performance rimangono sono paragonabili ad un singolo nodo.

In caso di collegamenti tra nodi molto lenti le cose potrebbero peggiorare (ma risolvibile in parte grazie ai component).



Backup

Le configurazioni in replica, possono essere abusate per alcuni task.

- Repliche in modalità delay per analizzare il comportamento in ritardo delle transazioni.
- Repliche semi-sync (o anche async) possono essere usate per eseguire backup:
  - in linea (innobackupex)
  - bloccando la replica (mysqldump)
  - a livello di filesystem utilizzando snapshot (zfs / lvm)

Si può migrare più facilmente da una versione di MySQL ad un'altra, senza downtime (usando una config HA) o minimamente.

Magia oscura

Klaatu verata... nike... netta, nettare, nichel...

3

ClickMeter

Ma te sta robba la usi ???

ClickMeter

Analytics per marketing

Stranamente la gente clicca tanto

Tanti dati e l’infrastruttura deve stargli dietro

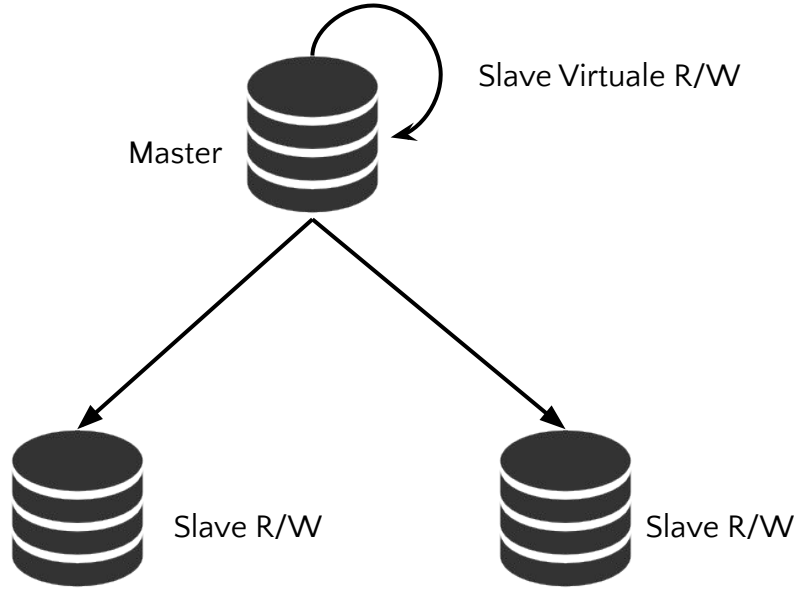
Infrastruttura ibrida (lambda) con MySQL

Clickmeter - Come era

Pre NoSQL.

Aumento capacità scrittura con scalabilità lettura (unione di due tipologie di scenari gia' visti).

Niente proxy, tutta logica applicativa.



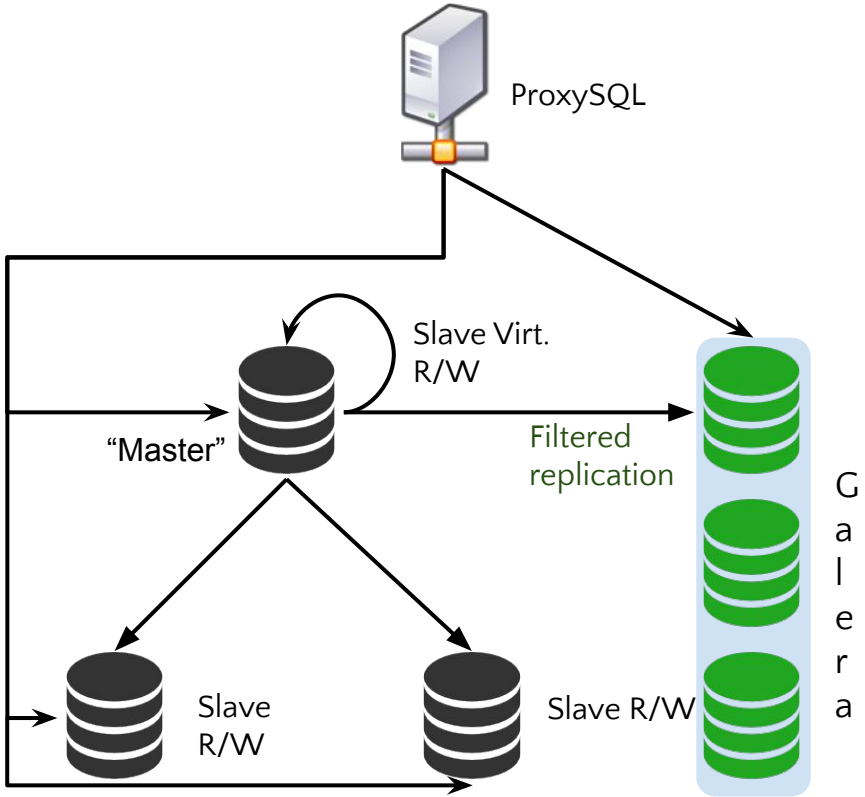
Clickmeter - Migrazione

Per alcuni dati ad altissima frequenza di scrittura si è scelto un NoSQL.

Per il resto si è scelto Galera (accounting e aggregati a bassa granularità).

Ovviamente và tutto migrato e nel frattempo per l'applicazione tutto deve sembrare uguale.

ProxySQL to the the rescue.

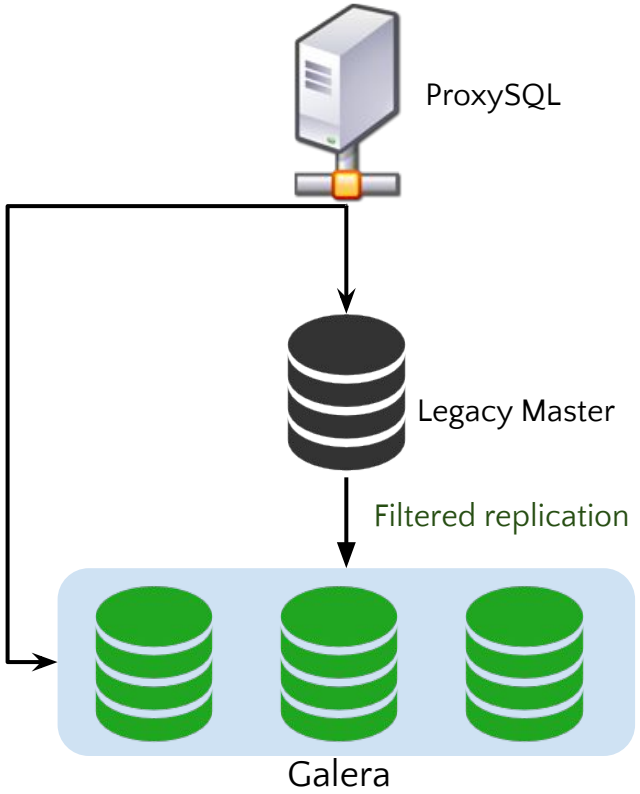


Clickmeter - Come è

Legacy Master per applicazioni legacy.

Distribuzione carico letture/scritture su ProxySQL per applicazioni standard (redirect applicazioni legacy).

Accesso diretto tramite connettore MySQL (con elenco DSN) a Galera per applicazioni speciali (HA needed).

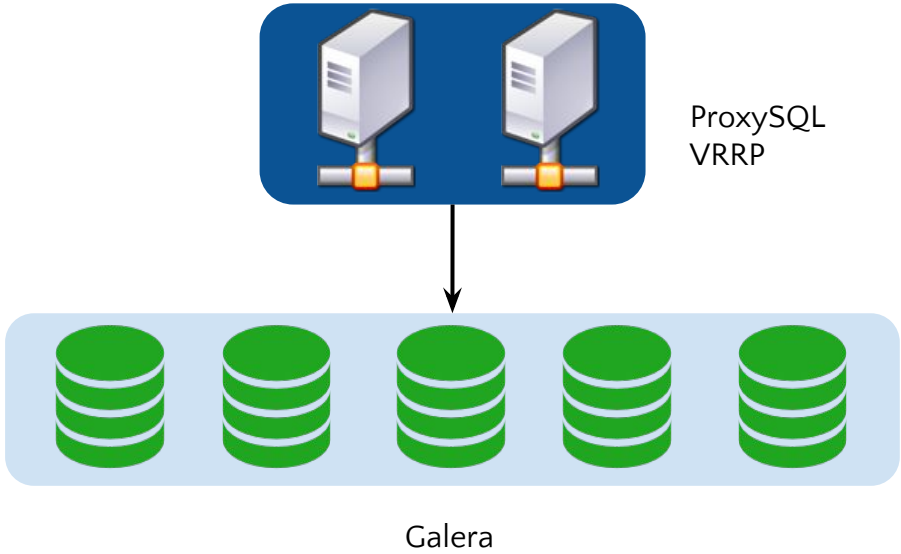


Clickmeter - Come sarà

No more legacy.

Distribuzione carico letture/scritture su ProxySQL per tutti (in modalità FF).

VRRP su ProxySQL per HA.



4

Altro??

Speriamo di no ...

Cloud

Tutte le configurazioni presentate sono valide sul cloud come sono valide sul ferro.

Alcune differenze sono da tenere a mente:

- Località dei dati (gli hard disk potrebbero essere di rete)
- Condivisione del ferro
- Condivisione del mezzo di rete
- Cold performance
- Configurazioni esoteriche della rete

Letture interessanti:

https://aphyr.com/posts/288-the-network-is-reliable
https://forums.aws.amazon.com/thread.jspa?messageID=454155
http://dbadojo.com/2014/07/28/prewarm-your-ebs-backed-ec2-mysql-slaves/

Libri

MySQL High Availability
by Charles Bell, Mats Kindahl, and Lars Thalmann

High Performance MySQL, Third Edition
by Baron Schwartz, Peter Zaitsev, and Vadim Tkachenko

Understanding MySQL Internals
by Sasha Pachev

Links

http://dev.mysql.com/doc/refman/5.6/en/replication.html
http://www.xenialab.it/meo/web/white/oracle/myrepl.htm
http://plusbryan.com/mysql-replication-without-downtime
https://www.percona.com/blog/2013/01/09/how-does-mysql-replication-really-work/
https://www.percona.com/blog/2008/09/22/fighting-mysql-replication-lag/
https://dev.mysql.com/tech-resources/articles/mysql-5.6-replication.html
https://dev.mysql.com/doc/refman/5.6/en/ha-zfs-config.html
http://en.wikipedia.org/wiki/WebScaleSQL
http://galeracluster.com/products/technology/
http://www.severalnines.com/blog/9-tips-going-production-galera-cluster-mysql
https://www.percona.com/blog/2011/10/13/benchmarking-galera-replication-overhead/
https://www.percona.com/blog/2014/11/17/typical-misconceptions-on-galera-for-mysql/
https://www.percona.com/blog/2012/06/14/comparing-percona-xtradb-cluster-with-semi-sync-replication-cross-wan/
https://code.google.com/p/mysql-master-ha/wiki/Other_HA_Solutions
http://en.wikipedia.org/wiki/MySQL_Cluster
http://dev.mysql.com/doc/mysql-utilities/1.3/en/mysqlfailover.html
http://yoshinorimatsunobu.blogspot.it/2014/04/semi-synchronous-replication-at-facebook.html
https://github.com/renecannao/mysql-cluster-tutorial
https://dev.mysql.com/doc/refman/5.6/en/replication-gtids-concepts.html
http://my-replication-life.blogspot.it/2013/09/loss-less-semi-synchronous-replication.html
https://dev.mysql.com/doc/refman/5.6/en/replication-gtids-concepts.html
https://aphyr.com/posts/288-the-network-is-reliable
https://forums.aws.amazon.com/thread.jspa?messageID=454155
http://dbadojo.com/2014/07/28/prewarm-your-ebs-backed-ec2-mysql-slaves/
https://www.percona.com/live/mysql-conference-2013/sessions/galera-cluster-best-practices

Grazie!