

数据高可用架构设计与实现

大型企业如何实现MySQL到Redis的同步

缓存不命中

使用 Binlog 实时更新Redis缓存

Canal详解

商城项目的实现

安装运行Canal服务

MySQL打开Binlog日志

启动Canal服务端

tulingmall-canal的实现

基于Canal实现MySQL到Redis缓存数据同步

数据高可用架构设计与实现

大型企业如何实现MySQL到Redis的同步

前面曾提到过 Read/Write Through 和 Cache Aside这几种更新缓存的模式或者说策略，这几种策略都存在缓存不命中的可能性，如果缓存没有命中，就需要直接访问数据库以获取数据。

一般情况下，只要提前做好缓存预热，使缓存的命中率保持在一个相对比较高的水平上，那么直接访问数据库的请求比例就会非常低，这种情况下，一般没有什么问题。但是，如果是一个超大规模的系统或极高并发的情况下那就又不一样了。

缓存不命中

构建Redis集群后，由于集群可以水平扩容，因此只要集群足够大，理论上支持海量并发就不是问题。但是如果并发请求数量的基数过大，那么即便只有很小比率的请求直接访问数据库其绝对数量也仍然不小，再加上促销活动的流量峰值，还是会存在系统雪崩的风险。

那么，这个问题该如何解决呢？其实方法很简单，让所有请求都落在缓存上。硬件的价格一般总体是向下的，只要预算足够，Redis集群的容量理论上就是无限的。我们可以把全量数据都放在Redis集群中，处理读请求的时候，只需要读取Redis，而不用访问数据库，这样就完全没有“缓存不命中”的风险了。实际上，很多大型互联网公司都在使用这种方法。

不过在Redis中缓存全量数据，又会引发一个新的问题。那就是缓存中的数据应该如何更新呢？因为现在是从缓存中直接读到数据，则可以直接返回，如果没能读到数据，那就只能返回错误了。所以，当系统更新数据库的数据之后，必须及时更新缓存。

至此，我们又要面对一个老问题：如何保证Redis中的数据与数据库中的数据一致性？分布式事务当然可以来解决数据一致性的问题，但是不太适合用来更新缓存，至少大部分的分布式事务实现对数据更新服务有很强的侵入性，而且如果Redis本身出现了故障，写入数据失败则还会导致事务全部失败的问题，相当于是降低了服务的性能和可用性。

一个可行的方法是，启动一个更新订单缓存的服务接收数据变更的消息队列(Message Queue, MQ)中的消息，然后注意解决消息的可靠性问题即可，这种方式实现起来很简单，也没有什么侵入性。

使用 Binlog 实时更新Redis缓存

但是如果我们要缓存的数据原本就没有一份数据更新的消息队列可以订阅，又该怎么办呢？其实很多大型互联网企业所采用的更通用的解决方案是使用 Binlog 实时更新Redis缓存。

数据更新服务只负责处理业务逻辑，更新MySQL中的数据，完全不用考虑如何更新缓存。负责更新缓存的服务，把自己伪装成一个MySQL的从节点。从 MySQL 接收并解析Binlog之后，就可以得到实时的数据变更信息，然后该服务就会根据这个变更信息去更新Redis缓存。

订阅 Binlog更新缓存的方案，相较于上文中接收消息更新Redis缓存的方案，两者的实现思路其实是一样的，都是异步实时订阅数据变更信息以更新Redis缓存。只不过,直接读取Binlog这种方式通用性更强。

除此之外，由于在整个缓存更新链路上，减少了一个收发消息队列的环节，从 MySQL更新到Redis更新的时延变得更短，出现故障的可能性也更低，这也是为什么很多大型互联网企业更青睐于采用这种方案的原因。

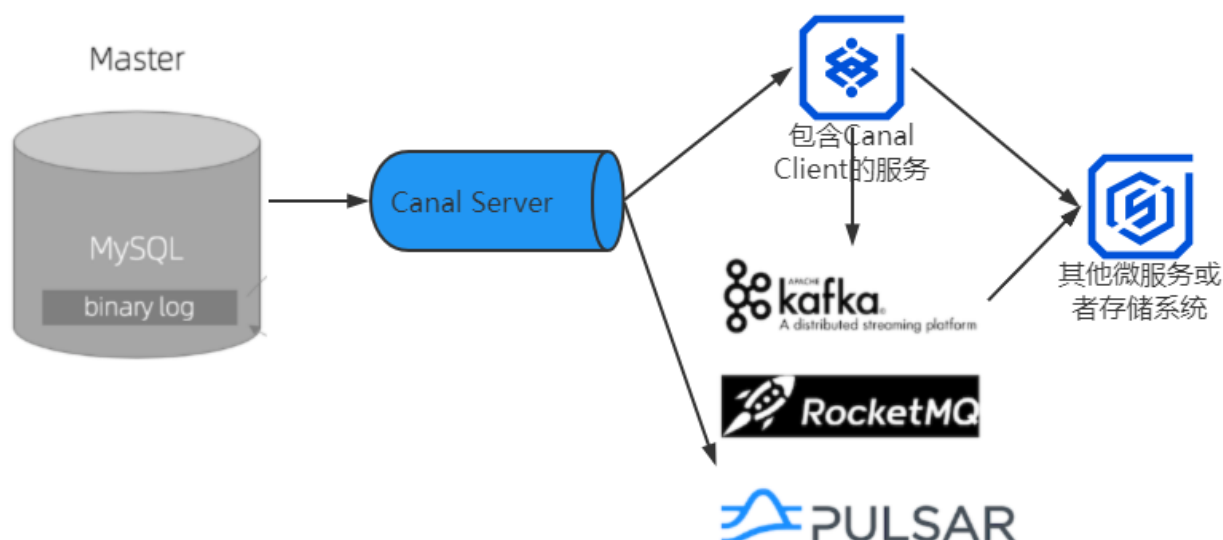
订阅Binlog更新缓存的方案唯一的缺点是：自行实现比较复杂，该方案毕竟不像接收消息那样,收到的直接就是订单数据，解析Binlog需要对MySQL的底层相当熟悉，还是挺麻烦的。

很多开源的项目都提供了订阅和解析MySQL Binlog的功能，在我们的商城项目中就使用了比较常用的开源项目Canal来实时接收Binlog更新Redis缓存。

Canal详解

Canal是阿里开源的一个项目，官方主页：<https://github.com/alibaba/canal>。

它通过模拟MySQL主从复制的交互协议，把自己伪装成一个MySQL的从节点，向 MySQL主节点发送dump请求。MySQL收到请求后，就会向 Canal开始推送Binlog，Canal解析Binlog字节流之后，将其转换为便于读取的结构化数据，供下游程序订阅使用。实际运用后的运行架构如图：



可以看到Canal有个服务端，在模拟MySQL从节点获得数据库服务器的数据后，我们可以使用一个包含 Canal Client的服务程序获得Canal服务端解析出的数据，也可以通过配置让Canal服务端直接将数据发送给MQ，当然我们的Canal Client程序经过数据处理后也可以发送给MQ。不管是经过Canal Client程序还是直接发给MQ，接下来还可由第三方的服务或者存储系统进行后续处理。

商城项目的实现

安装运行Canal服务

Canal服务端是用Java编写的，所以Windows和Linux下都可以运行，考虑到大多数同学会在本机进行测试，我们以Windows下的安装运行为例进行说明，目前环境为操作系统Windows11，MySQL 8.0.19。

MySQL打开Binlog日志

既然Canal模拟MySQL从节点获得数据库服务器的数据，很明显，对MySQL服务器的配置完全可以参考MySQL的主从复制中主节点的配置。怎么样让MySQL成为主节点？

先开启 Binlog 写入功能，配置 binlog-format 为 ROW 模式

```
2 show VARIABLES like "%log_bin%"
```

信息	结果 1	剖析	状态
Variable_name			Value
▶ log_bin			ON

```
2 show VARIABLES like "%binlog_format%"
```

信息	结果 1	剖析	状态
Variable_name		Value	
▶ binlog format	ROW		

```
2 show VARIABLES like "%server_id%"
```

信息	结果 1	剖析	状态
Variable_name		Value	
▶ server id		1	

如果本机的MySQL的配置与上述不符合，可以修改MySQL的配置文件并重启MySQL，一般来说是my.ini，大多情况下这个文件放在MySQL的安装目录或者C:\ProgramData的对应MySQL的目录下

此电脑 > SYSTEM (C:) > ProgramData > MySQL > MySQL Server 8.0 >

名称	修改日期	类型
Data	2022/11/2 22:42	文件夹
Uploads	2020/2/5 21:30	文件夹
installer_config.xml	2020/2/5 21:30	Microsoft
my.ini	2020/2/5 21:30	Configura

```
my.ini
9
10 # General and Slow logging.
11 log-output=FILE
12
13 general-log=0
14
15 general_log_file="MAOKE.log"
16
17 slow-query-log=1
18
19 slow_query_log_file="MAOKE-slow.log"
20
21 long_query_time=10
22
23 # Error Logging.
24 log-error="MAOKE.err"
25
26 # ***** Group Replication Related *****
27 # Specifies the base name to use for binary log files. With binary logging
28 # enabled, the server logs all statements that change data to the binary
29 # log, which is used for backup and replication.
30 log-bin="MAOKE-bin"
31
32 # ***** Group Replication Related *****
33 # Sets the binary logging format, and can be any one of STATEMENT, ROW,
34 # or MIXED. ROW is suggested for Group Replication.
35 # binlog_format
```

既然是将Canal模拟MySQL从节点，自然还要给Canal设置一个用来复制数据的MySQL账号，我们这里设定这个用户名为Canal，密码Canal

```
CREATE USER canal IDENTIFIED BY 'canal';

GRANT SELECT, REPLICATION SLAVE, REPLICATION CLIENT ON *.* TO 'canal'@'%';

FLUSH PRIVILEGES;
```

此时在MySQL的表中就应该存在canal用户，拥有select、repl权限

```
1 select * from mysql.`user`;
2 -- show VARIABLES like "%server_id%"
```

信息	结果 1	剖析	状态	
Host	User	Select_priv	Insert_priv	Update_priv
%	canal	Y	N	N

当然要完成主从复制，还有一步，找到当前binlog的进度让从MySQL使用

```
1 show master STATUS
```

信息	结果 1	剖析	状态
File	Position	Bir	
▶ MAOKE-bin.000049	155		

启动Canal服务端

把得到的Canal服务端程序解压缩

此电脑 > WORK (D:) > TuLing > VIPL > Mall > env > canal.deployer-1.1.6

名称	修改日期	类型
bin	2022/10/19 13:23	文件夹
conf	2022/10/19 14:09	文件夹
lib	2022/10/19 13:23	文件夹
logs	2022/10/19 14:25	文件夹
plugin	2022/10/19 13:23	文件夹

进入conf目录，修改canal.properties文件，比较关键的是canal.destinations

```
#####
##### destinations #####
#####
canal.destinations = promotion,seckill
# conf root dir
canal.conf.dir = ../conf
# auto scan instance dir add/remove and start/stop
```

这里表示，我们需要监控与促销promotion、秒杀seckill相关的数据变动，而相关的数据库、表配置会分别放在promotion、seckill目录下，如果没有这两个目录需要新建这两个目录

> 此电脑 > WORK (D:) > TuLing > VIPL > Mall > env > canal.deployer-1.1.6 > conf

名称	修改日期	类型	大小
example	2022/10/19 13:23	文件夹	
metrics	2022/10/19 13:23	文件夹	
promotion	2022/10/19 14:10	文件夹	
seckill	2022/10/19 14:10	文件夹	
spring	2022/10/19 13:23	文件夹	
canal.properties	2022/10/19 14:09	PROPERTIES 文件	7 KB
canal_local.properties	2021/6/22 16:48	PROPERTIES 文件	1 KB
logback.xml	2022/5/20 15:26	Microsoft Edge ...	5 KB

进入promotion目录，修改instance.properties，如果目录中没有这个文件，可以从example目录拷贝一个过来

> 此电脑 > WORK (D:) > TuLing > VIPL > Mall > env > canal.deployer-1.1.6 > conf > example

名称	修改日期	类型	大小
h2.mv.db	2022/10/30 21:09	Data Base File	904 KB
instance.properties	2022/8/1 16:35	PROPERTIES 文件	3 KB
meta.dat	2022/10/30 21:09	媒体文件 (.dat)	1 KB

对于instance.properties的修改比较关键的就是几处，一是MySQL主服务的连接配置


```
# position info
canal.instance.master.address=127.0.0.1:3306
canal.instance.master.journal.name=MAOKE-bin.000049
canal.instance.master.position=155
canal.instance.master.timestamp=
canal.instance.master.gtid=
```

```
# username/password
canal.instance.dbUsername=canal
canal.instance.dbPassword=canal
```

二则是要对哪些相关的业务表进行监视，比如我们这里是promotion促销信息，数据放在tl_mall_promotion库中：

```
# table regex
canal.instance.filter.regex=tl_mall_promotion.sms_home_advertise,tl_ma
```

配置完成后，进入bin目录，执行startup.bat即可

此电脑 > WORK (D:) > TuLing > VIPL > Mall > env > canal.deployer-1.1.6 > bin

名称	修改日期	类型	大小
restart.sh	2021/6/22 16:48	Shell Script	1 KB
startup.bat	2021/10/9 17:39	Windows 批处理...	2 KB
startup.sh	2022/5/23 17:36	Shell Script	4 KB
stop.sh	2021/6/22 16:48	Shell Script	2 KB

Windows PowerShell

Windows Terminal can be set as the default terminal application in your settings. [Open Settings](#)

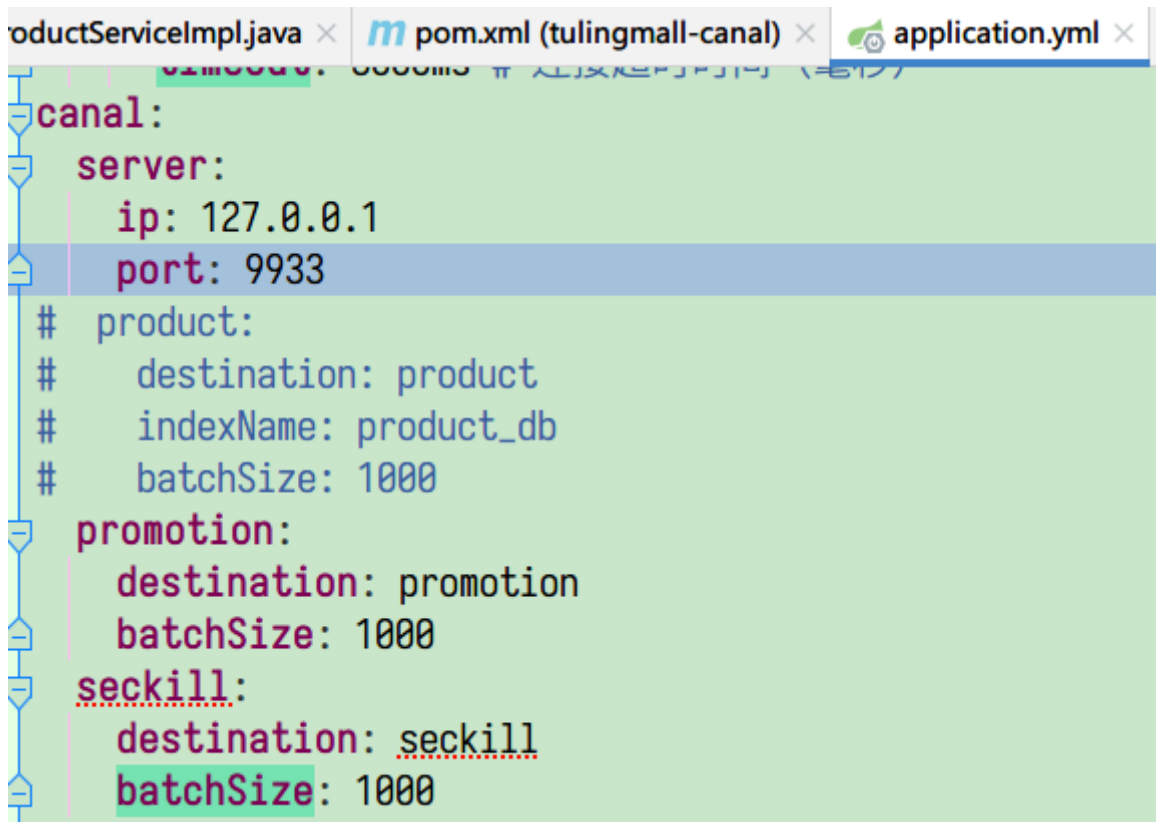
```
Windows PowerShell
版权所有 (C) Microsoft Corporation。保留所有权利。

安装最新的 PowerShell，了解新功能和改进！ https://aka.ms/PSWindows

PS D:\TuLing\VIPL\Mall\env\canal.deployer-1.1.6\bin> .\startup.bat
start cmd : java -Xms128m -Xmx512m -XX:PermSize=128m -Djava.awt.headless=true -Djava.net.preferIPv4St
ication.codeset=UTF-8 -Dfile.encoding=UTF-8 -server -Xdebug -Xnoagent -Djava.compiler=NONE -Xrunjdwp:tra
t,address=9099,server=y,suspend=n -DappName=otter-canal -Dlogback.configurationFile="D:\TuLing\VIPL\Mall
oyer-1.1.6\bin\...\conf\logback.xml" -Dcanal.conf="D:\TuLing\VIPL\Mall\env\canal.deployer-1.1.6\bin\...\c
rties" -classpath "D:\TuLing\VIPL\Mall\env\canal.deployer-1.1.6\bin\...\conf\...\lib\*;D:\TuLing\VIPL\Mall
oyer-1.1.6\bin\...\conf" java -Xms128m -Xmx512m -XX:PermSize=128m -Djava.awt.headless=true -Djava.net.
=true -Dapplication.codeset=UTF-8 -Dfile.encoding=UTF-8 -server -Xdebug -Xnoagent -Djava.compiler=NONE -
ort=dt_socket,address=9099,server=y,suspend=n -DappName=otter-canal -Dlogback.configurationFile="D:\TuLi
v\canal.deployer-1.1.6\bin\...\conf\logback.xml" -Dcanal.conf="D:\TuLing\VIPL\Mall\env\canal.deployer-1.1
\canal.properties" -classpath "D:\TuLing\VIPL\Mall\env\canal.deployer-1.1.6\bin\...\conf\...\lib\*;D:\TuLi
v\canal.deployer-1.1.6\bin\...\conf" com.alibaba.otter.canal.deployer.CanalLauncher
Java HotSpot(TM) 64-Bit Server VM warning: ignoring option PermSize=128m; support was removed in 8.0
Listening for transport dt_socket at address: 9099
```

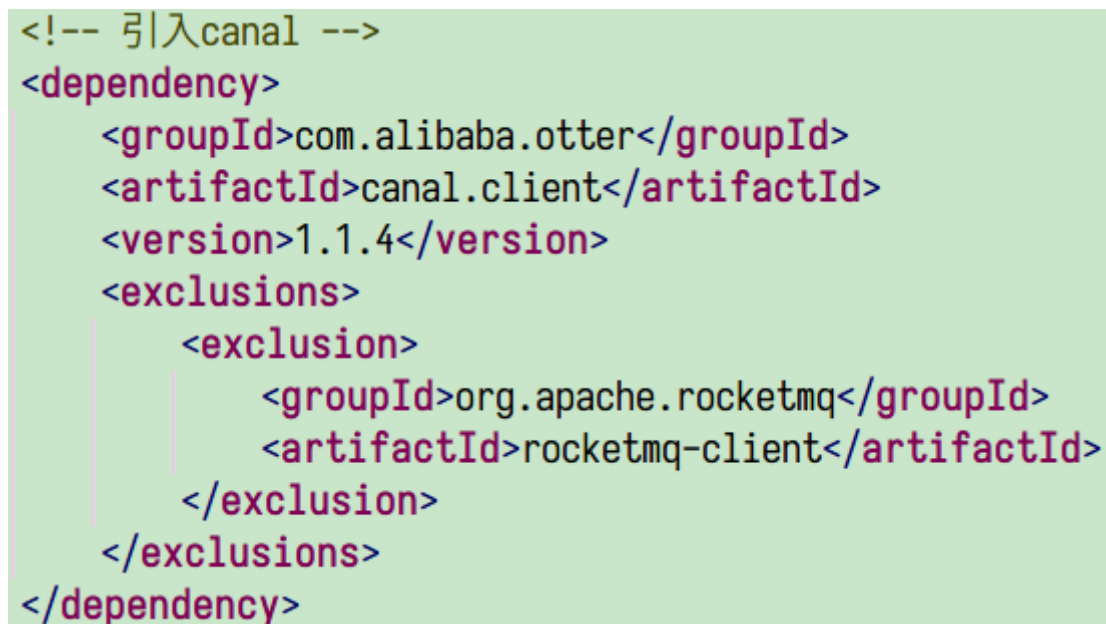
tulingmall-canal的实现

完成了Canal服务端配置后，接下来我们就可以实现包含Canal Client的服务了。当然，首先要把Canal服务端的相关信息配置到我们的代码中



```
productServiceImpl.java x pom.xml (tulingmall-canal) x application.yml x
canal:
  server:
    ip: 127.0.0.1
    port: 9933
  # product:
  #   destination: product
  #   indexName: product_db
  #   batchSize: 1000
  promotion:
    destination: promotion
    batchSize: 1000
  seckill:
    destination: seckill
    batchSize: 1000
```

和Maven配置



```
<!-- 引入canal -->
<dependency>
  <groupId>com.alibaba.otter</groupId>
  <artifactId>canal.client</artifactId>
  <version>1.1.4</version>
  <exclusions>
    <exclusion>
      <groupId>org.apache.rocketmq</groupId>
      <artifactId>rocketmq-client</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

就可以进入实际的开发了。

在我们的PromotionData类中，主要的业务就是发现促销信息相关的表发生变动后删除促销信息在Redis中保存的键值对，以方便缓存的更新。具体解释请参考代码注释或者视频的讲解。

有道云笔记链接: <https://note.youdao.com/s/5kl5foDv>