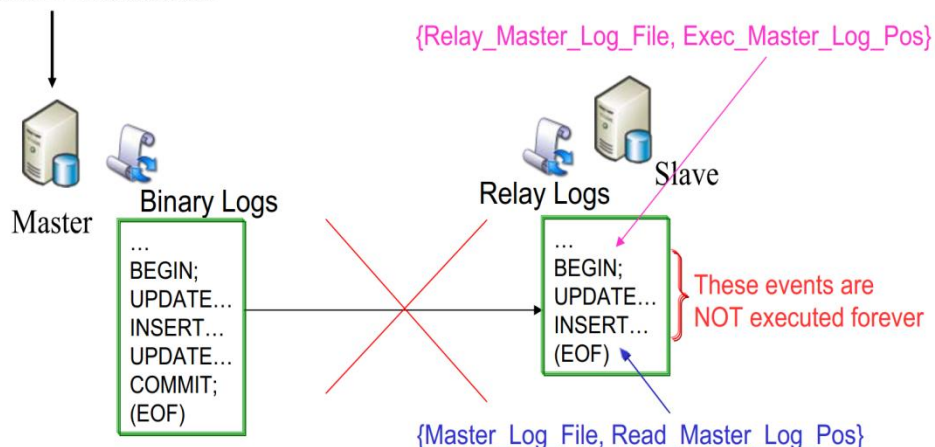


在 MySQL 一主多从的架构中，如果 Master 宕机，我们需要选择一个最新的 Slave 作为新 Master，并且尝试从原 Master 补齐数据。但是如果遇到“Partial Transaction”，我们该如何处理呢？

首先什么是“Partial Transaction”？（以下内容来自 MHA 相关文档）

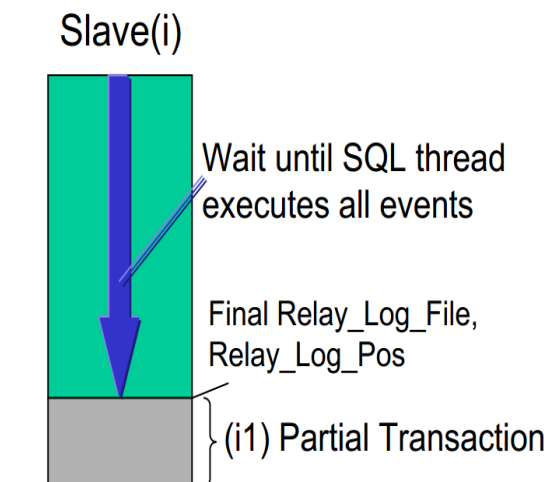
Massive transactions



就像这个图中描述的，Master 有一个非常大的事务，并且成功提交。Slave 的 IO thread 去拉 binlog 时，Master 宕机。这时 Slave 拉取到的是部分事务，同时 Slave 又会自动在 relay-log 的结尾加上 ROLLBACK。这时在 Slave 上 show slave status 时会发现：

Read_Master_Log_Pos != Exec_Master_Log_Pos，这就是“Partial Transaction”。

如果我们以常规的 Read_Master_Log_Pos 最大的 Slave 为新 Master，然后以 Read_Master_Log_Pos 的点，去原 Master 拉数据的话，那么下图中的 Partial Transaction 会丢掉：



还是用实际的测试来验证这个吧。

环境准备：

1、首先搭建好 Master-Slave 环境。

2、创建测试表：

```
CREATE TABLE `ml` (
```

```
`id` int(11) NOT NULL AUTO_INCREMENT,  
`name` varchar(100) DEFAULT NULL,  
PRIMARY KEY (`id`)
```

```
) ENGINE=InnoDB AUTO_INCREMENT=1 DEFAULT CHARSET=utf8 ;
```

- 3、写一个程序，做一个大事务【对一个表做 10w 次 update】，提交之后马上 kill Master 进程。这样我们就可以模拟，事务在 Master 提交，Slave 只能拉取到部分事务的场景【如果您机器的性能很好，这个事务可以再大点】。程序代码见 test.py

测试步骤：

- 1、在 Master 上以 root 帐号运行程序
运行方式：python test.py <mysql 进程号>
- 2、事务提交后，原 Master 被程序自动 kill
- 3、去验证 Slave 是否只拉取到部分事务

```
Slave_IO_State: Reconnecting after a failed Master event read
```

```
.....  
Master_Log_File: mysql-bin.000005  
Read_Master_Log_Pos: 7947670
```

```
.....  
Relay_Master_Log_File: mysql-bin.000005
```

```
Slave_IO_Running: Connecting  
Slave_SQL_Running: Yes
```

```
.....  
Exec_Master_Log_Pos: 10353
```

```
.....  
Last_IO_Errno: 2003
```

```
Last_IO_Error: error reconnecting to Master 'repl@127.0.0.1:3306' -  
retry-time: 60 retries: 86400
```

注：从输出的信息看，Slave 已经不能连接到 Master，同时 Read_Master_Log_Pos 已经读取到 7947670，但是 Exec_Master_Log_Pos 只到 10353，这时因为 Slave 拉取的事务是不完整的，并且 Slave 不会再继续应用 relay log。

可以看到如果 Slave 以 7947670 应用日志，那么 10353 到 7947670 之间的事务就丢掉了。

- 4、补齐数据【假设 Master 还能访问】

通常的步骤是：

- 1) 在 Slave 先应用日志到 7947670
- 2) Trim relay log 后面自动加的 ROLLBACK（这个是 mysqlbinlog 自动加的，因为应用自己也会有 ROLLBACK，所以 HA 软件需要自动识别出来）
- 3) 然后在原 Master 的 7947670 出开始 应用 binlog

注：这个过程稍显繁琐，QGuard For Mysql 中对这个过程做了优化，步骤更少，效率更高。