GREENPLUM DATABASE®

Greenplum高可用理论与实践

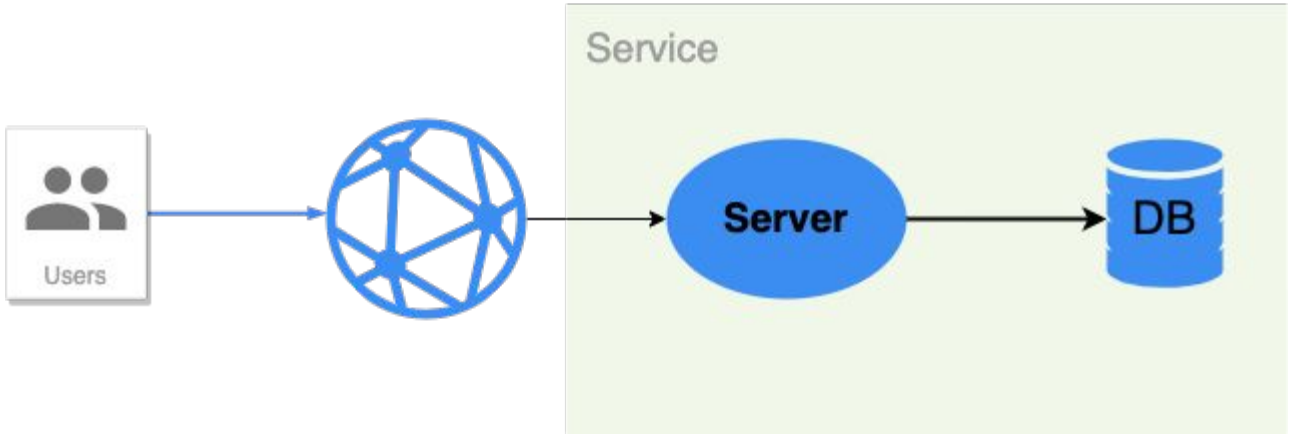
钉钉直播 | 12月17日 16：00 - 17：00

Greenplum高可用理论与实践

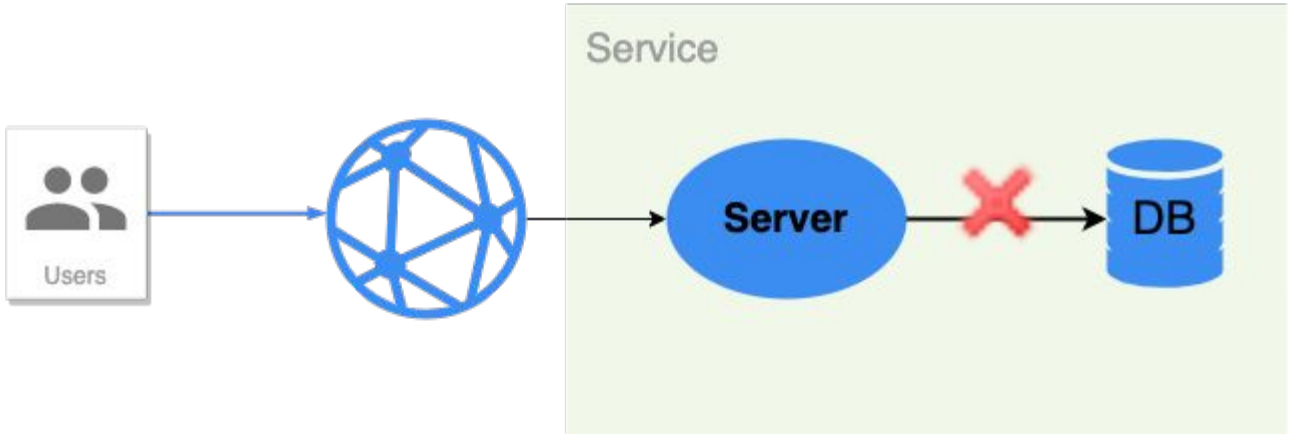
大纲

- 高可用简介
- 高可用的一般性原理
- 日志复制与数据一致性
- Greenplum的高可用实现FTS
- Greenplum Master节点的高可用

高可用简介



高可用简介



高可用的一般性原理

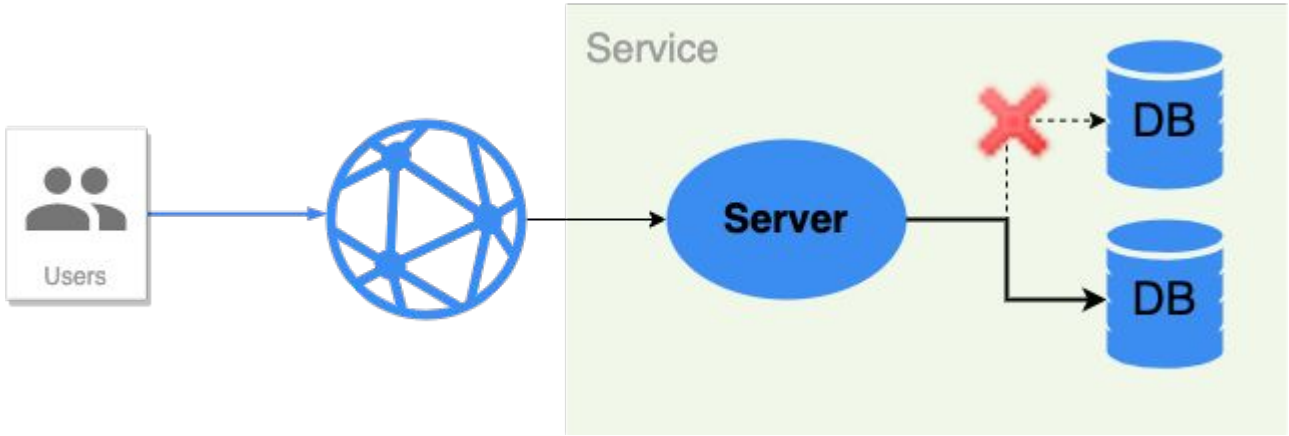
GREENPLUM DATABASE

- 单机做到无故障，unbreakable
  - 硬件故障，系统级软件缺陷，数据库软件自身BUG
  - 可用性的极限，成本随着可用性增加成倍增加
  - 地理因素

- 假定单机会出现软硬件故障
  - 其他节点来替代服务
  - 可靠性= $1-\prod_{i}(1-p_i)$
    - p: 单机可用性
    - n: 备用节点个数

高可用的一般性原理

- 多节点提高可用性满足条件(high level):
  - 切换时间足够小，可以接受
  - 新节点和老节点数据一致



高可用的一般性原理

GREENPLUM DATABASE

多节点提高可用性的做法：

- 数据复制
  - 数据即时同步到备用节点上（保证切换时间在可接受范围内）
- 一致性
  - 对数据同步施加限制来保证数据一致性
- 检测并切换到新的可用节点
  - 主从，主备，...

日志复制与数据一致性

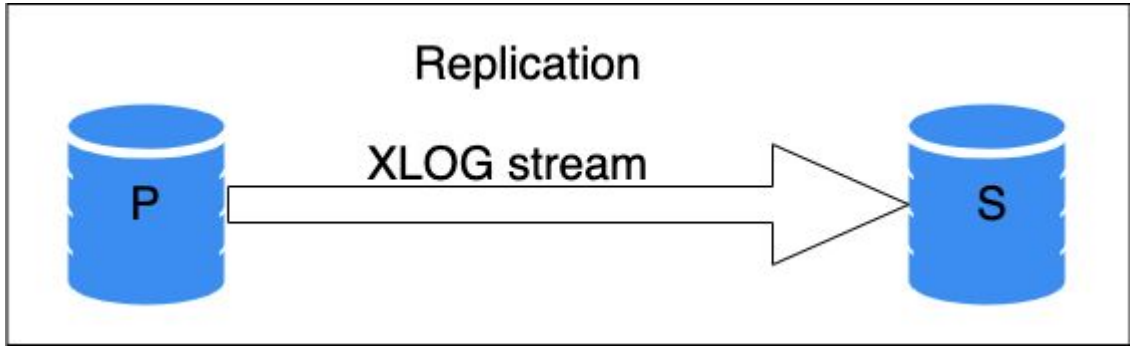
GREENPLUM DATABASE

数据复制

主流的数据同步方式：事务日志流(Replication)

PRIMARY: BEGIN1; W1(A); BEGIN2; W2(X), W1(B) ... COMMIT2; COMMIT1;

SECONDARY: BEGIN1; W1(A); BEGIN2; W2(X), W1(B) ... COMMIT2; COMMIT1;



日志复制与数据一致性

GREENPLUM DATABASE

一致性含义：

- 用户从主备节点看到相同的用户数据集合

数据一致性分类：

- 如果事务A在主节点上**完成**，那么它在备节点上也已经完成（强一致性）
- 事务A在切换到备节点后，它的**可见性**不变（弱一致性）

完成

事务已提交，且更改已经持久化

可见

隐含事务已提交，且当前快照对事务可见

日志复制与数据一致性

GREENPLUM DATABASE

同步事务日志(类型)

- Write: replication wait for remote write xlog(不可靠)
- Flush: replication wait for remote flush xlog(可靠)
- Apply: replication wait for remote apply xlog(可靠)

日志复制与数据一致性

GREENPLUM DATABASE

同步事务日志(逻辑)

日志同步方式以Flush为例

- primary写入(包含刷盘)本地commit日志记录(事务完成)
- primary传输commit日志记录给secondary，并等待对方确认
- secondary写入(包含刷盘)接收到的commit记录，并发送确认。
- primary收到commit确认
- primary返回结果给用户

日志复制与数据一致性

GREENPLUM DATABASE

Exception:

Primary	Secondary	Query
Flush commit		
Send commit record & wait		Read from primary(visible)
Crash		
	Promote to primary	
		Read from secondary(invisible)

日志复制与数据一致性

GREENPLUM DATABASE

修正同步逻辑

- primary写入本地commit日志记录(事务完成)
- primary传输commit日志记录给secondary，并等待对方确认
- secondary写入接收到的commit记录，并发送确认
- primary收到commit确认
- **primary将事务对其他查询可见**
- primary返回结果给用户

日志复制与数据一致性

GREENPLUM DATABASE

primary	secondary	query
Flush commit		
Send commit record & wait		Read from primary（invisible)
	Rcv-Flush-Ack commit	
Tx is visible to other T		
		Read from secondary(visible)

日志复制与数据一致性

GREENPLUM DATABASE

primary	secondary	query
Flush commit		
Send commit record & wait		Read from primary（invisible)
Crash		
	Promote to primary	
		Read from secondary(invisible)

日志复制与数据一致性

GREENPLUM DATABASE

两阶段提交（Two Phase Commit，2PC)

```
Begin;
W(x1);
W(x2);
...
Prepare Transaction;
Commit Prepared;
```

Prepare Transaction需要同步？

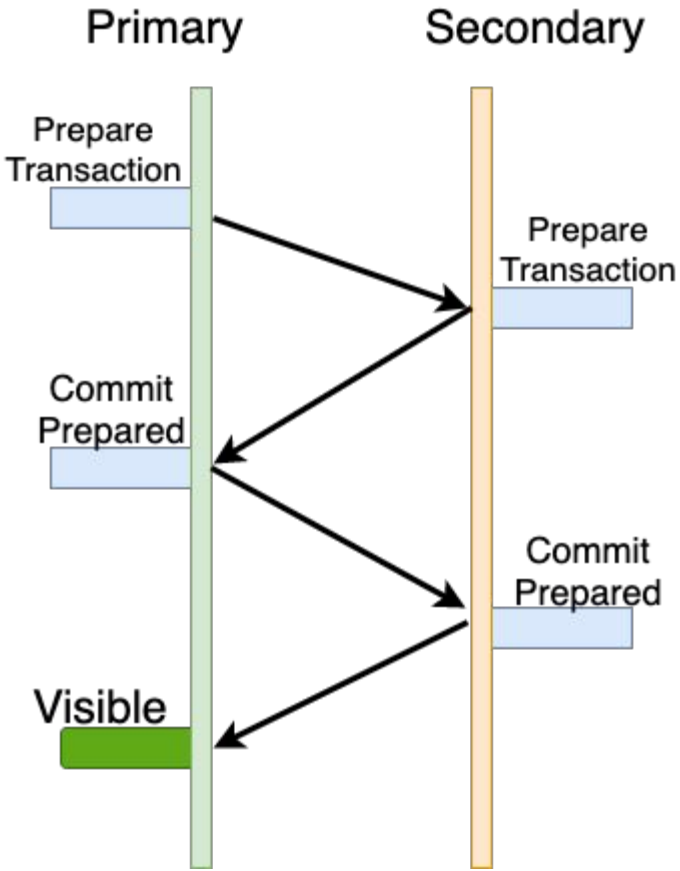
日志复制与数据一致性

GREENPLUM DATABASE

两阶段提交同步事务日志逻辑

- **primary写入本地prepare日志记录**
- **primary传输prepare日志记录给secondary，并等待对方确认**
- **primary接收prepare记录的确认**
- primary写入本地commit prepared日志记录（事务完成）
- primary传输commit prepared日志记录给secondary，并等待对方确认
- primary接收commit prepared记录的确认
- primary将事务对其他查询可见
- primary返回结果给用户

日志复制与数据一致性


Primary
Secondary
Prepare Transaction
Prepare Transaction
Commit Prepared
Commit Prepared
Visible

日志复制与数据一致性

GREENPLUM DATABASE

参考代码

```
CommitTransaction():
    latestXid = RecordTransactionCommit();
        XLogFlush(XactLastRecEnd);
        TransactionIdCommitTree()
        SyncRepWaitForLSN(XactLastRecEnd, true);
    ProcArrayEndTransaction(MyProc, latestXid);
```

Greenplum的高可用实现FTS

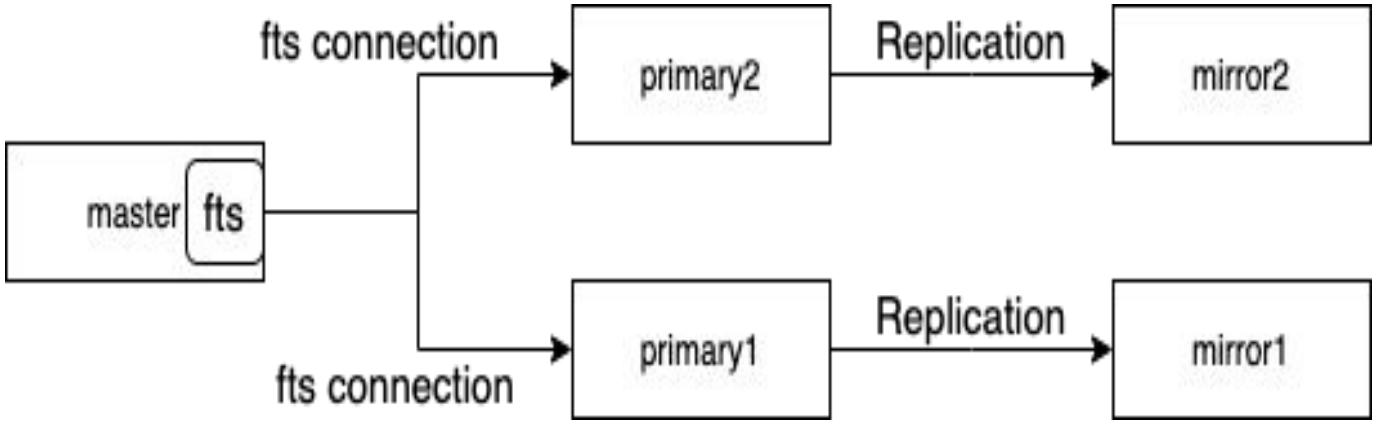
FTS（Fault Tolerance Service)

- 备节点复制主节点数据
  - 事务日志流
- 备节点拥有主节点断开前的所有可见事务，满足数据一致性约束
  - 对prepare/commit记录进行同步，且同步完成前对正在运行的事务不可见
- **数据库系统进行故障检测和切换**
  - Greenplum FTS检测

Greenplum的高可用实现FTS

FTS（Fault Tolerance Service)

- 检测节点连通性(PROBE)
- 提升mirror节点为新的primary节点(PROMOTE)
- 断开segment之间的主备同步(SYNCREP_OFF)



Greenplum的高可用实现FTS

GREENPLUM DATABASE

FTS检测连通性

- fts进程周期性地向所有的primary节点发起连接进行探测
- primary检查replication状态，更新自身和mirror的状态，并发送给master 节点
- fts进程更新节点状态和fts状态机

Greenplum的高可用实现FTS

GREENPLUM DATABASE

FTS提升mirror节点：

- fts超时未收到primary的回复
- fts交换primary和mirror的角色
- fts更新gp_segment_configuration
- fts向原mirror节点发送FTS_MSG_PROMOTE
- mirror节点执行提升逻辑，成为primary

一致性约束:

- status = 's'
- mirror is up && walsender->state == WALSNDSTATE_STREAMING

Replication Slot:

- 防止XLog被删除进而不得不执行全量恢复
- gprecoverseg vs gprecoverseg -F

Greenplum的高可用实现FTS

FTS断开segment之间的主从同步

必要性

日志同步会block写操作事务

类型

- 提升mirror节点
- mirror节点断开replication

Greenplum Master节点的高可用

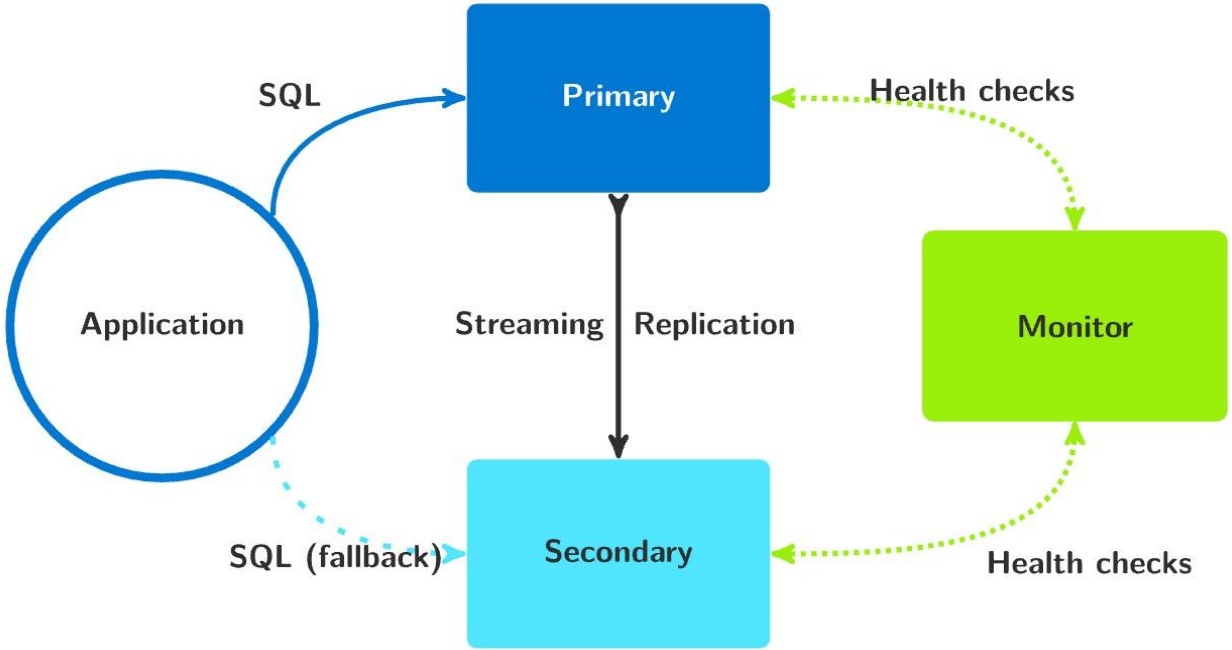
Greenplum FTS的不足

FTS只对segment进行了探测，无法在master节点故障时发生切换

Greenplum Master/Coordinator Auto Failover

- Greenplum 7
- 基于pg_auto_failover
- 研发进行中

Greenplum Master节点的高可用



(from https://pg-auto-failover.readthedocs.io/en/latest/architecture.html)

Greenplum Master节点的高可用

GREENPLUM DATABASE

可用性

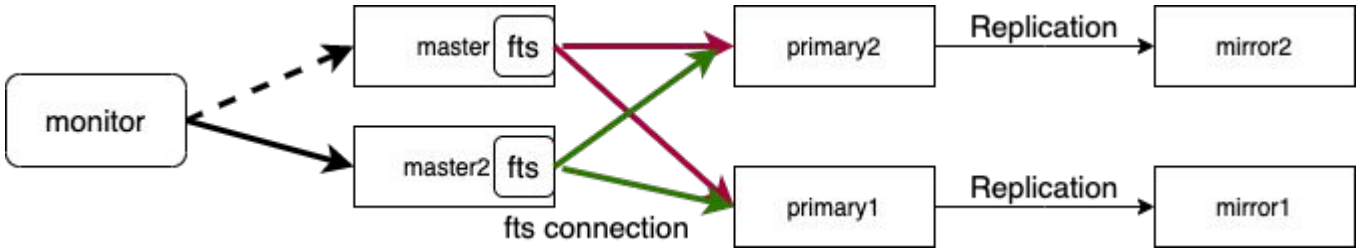
Master, Standby, Monitor三者至多只能有一个节点不可用

失效节点	操作	DB可用性
Master	提升standby	提升后的Standby对外服务
Standby	Monitor标记日志不同步	Master提供服务
Monitor		Master提供服务

Greenplum Master节点的高可用

Greenplum面临的主要问题：

- 两个master存活节点可能对segment进行写操作
  master写数据表
  FTS/Global Deadlock Detect/Dtx Recovery



Greenplum Master节点的高可用

GREENPLUM DATABASE

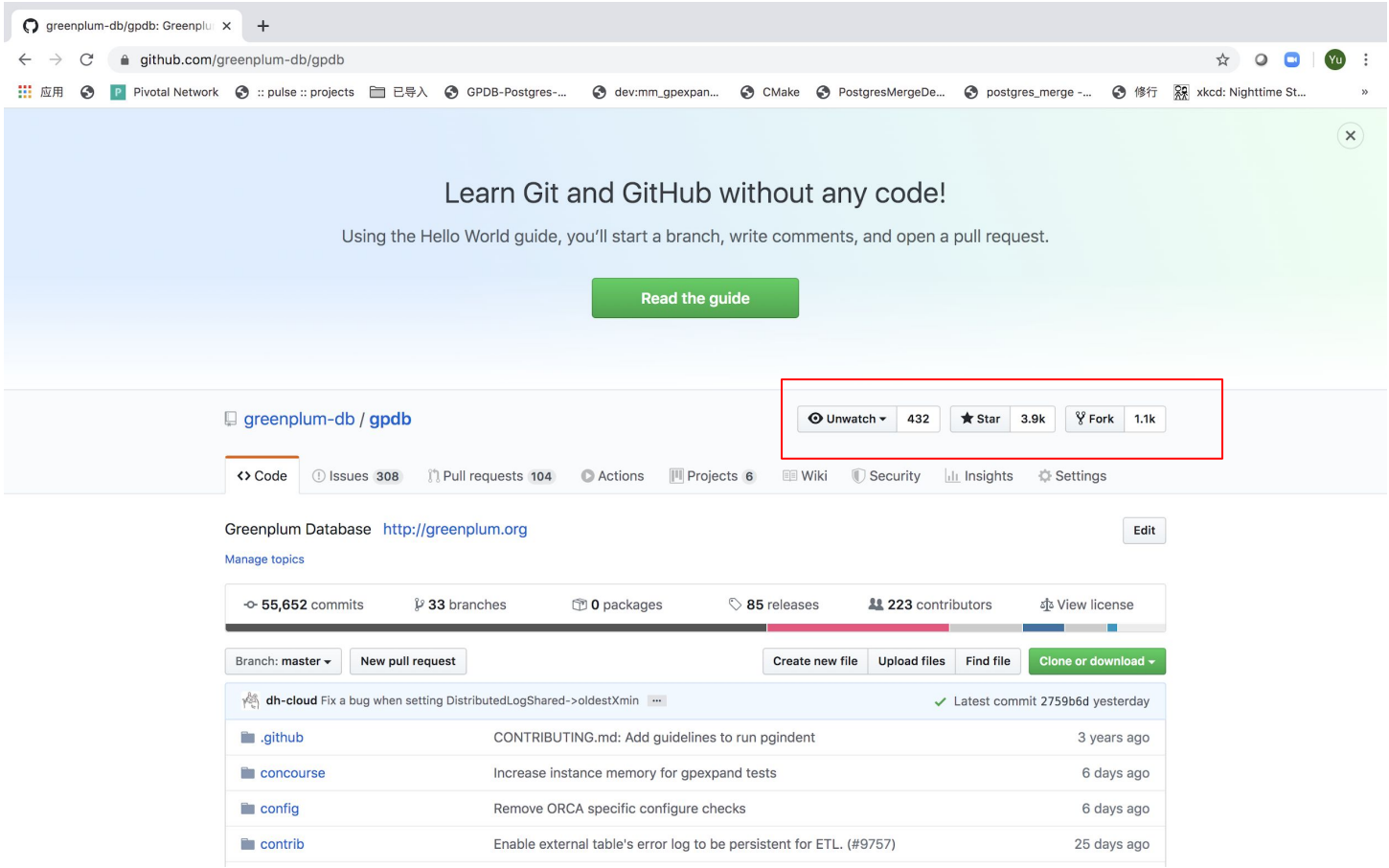
解决思路

1. master主动失活
   master节点主动退出/不再发起新的连接

2. segment确认master

- 完成standby提升前，通知所有segment重置活跃的master节点
- segment收到master重置消息后，杀死所有backend进程
- segment记录活跃的master节点
- segment拒绝所有来自其他节点的dispatch（重置master消息除外）

从源代码开始：下载编译Greenplum源代码



GREENPLUM DATABASE®

微信技术讨论群
添加入群小助手：gp_assistant

微信公众号
技术干货、行业热点、活动预告

欢迎访问Greenplum中文社区：cn.greenplum.org

全新的问答论坛

https://cn.greenplum.org/askgp


GREENPLUM
DATABASE®
Thank You
Q&A