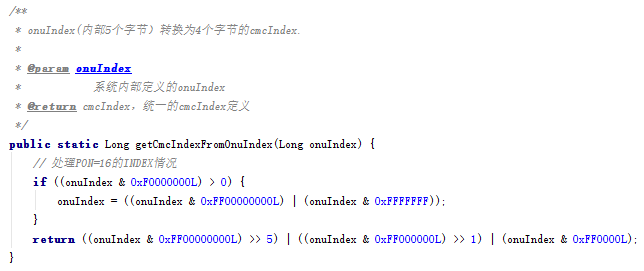
# CmcIndexUtils扩展研究

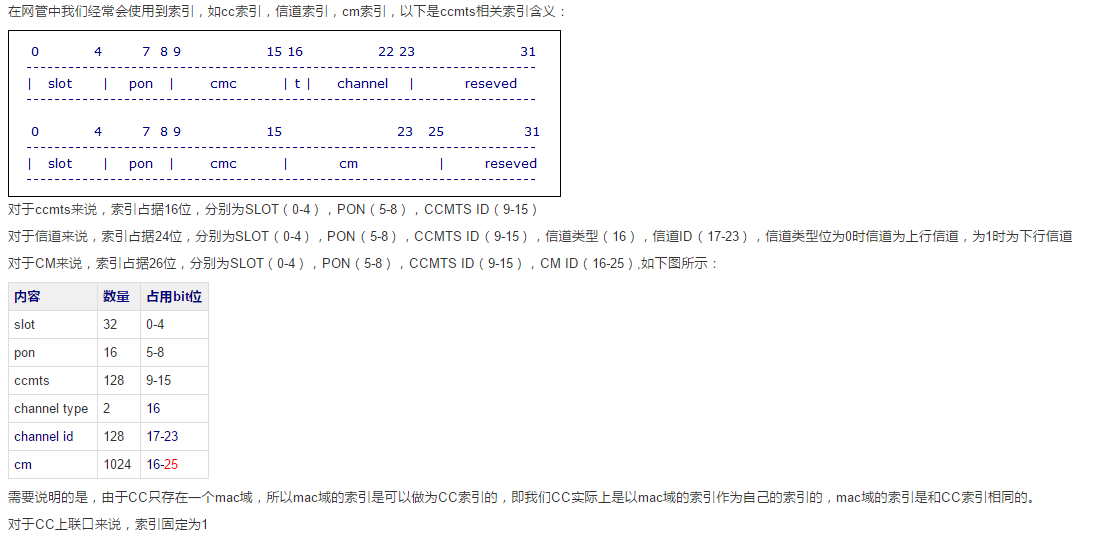
## getCmcIndexFromOnuIndex



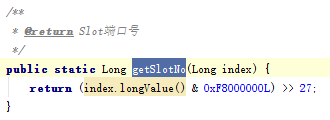
此方法用于将onuIndex（网管侧内部定义的Epon侧）转换为cmcIndex（CC侧定义）

网管内部侧定义为5级索引来定位slot，pon，onu，onu slot，uni

CC侧定义

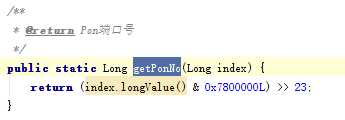


## getSlotNo



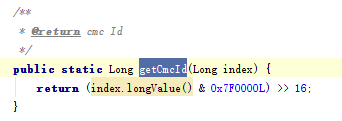
Index是cc侧定义

## getPonNo



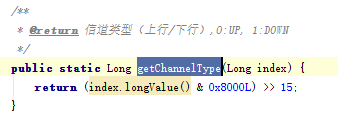
Index是cc侧定义

## getCmcId

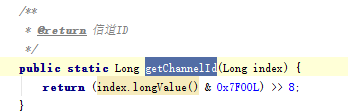


Index是cc侧定义

## getChannelType

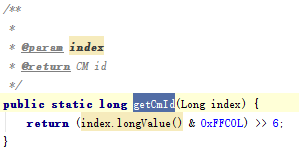


## getChannelId



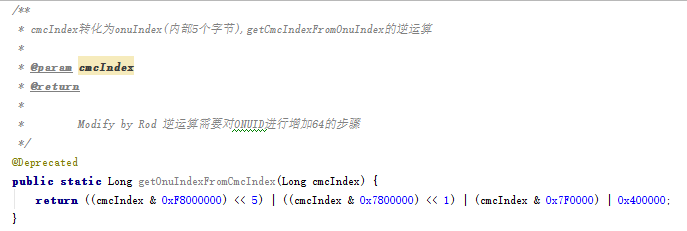
Index是cc侧定义

## getCmId



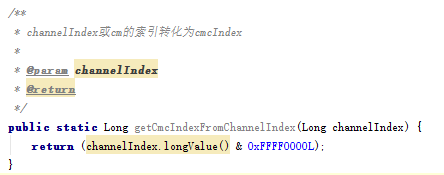
Index是cc侧定义

## getOnuIndexFromCmcIndex



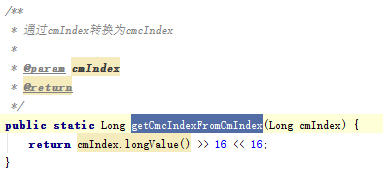
废弃方法 没有地方用到

## getCmcIndexFromChannelIndex



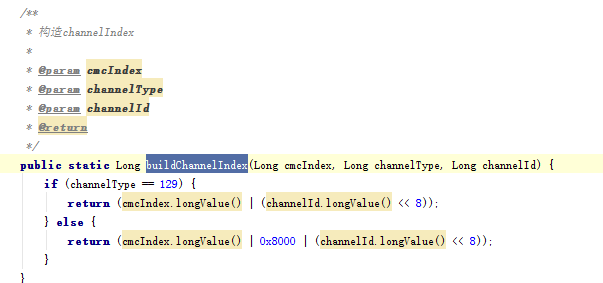
Index是cc侧定义

## getCmcIndexFromCmIndex



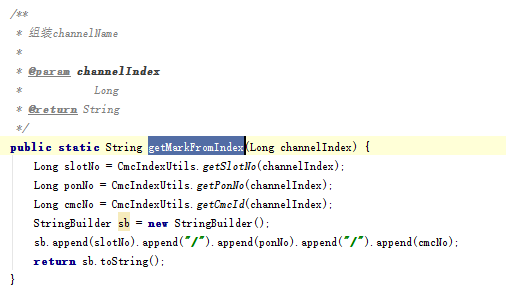
Index是cc侧定义

## buildChannelIndex



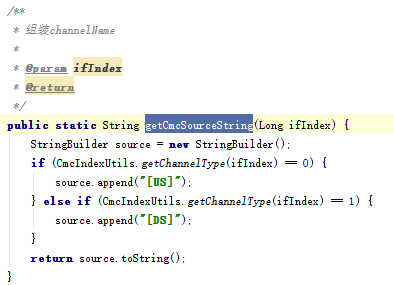
Index是cc侧定义

## getMarkFromIndex



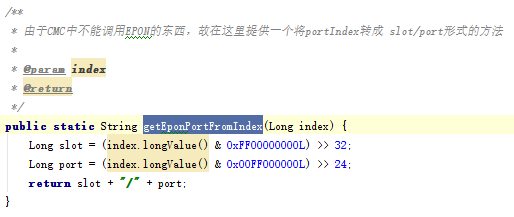
Index是cc侧定义

## getCmcSourceString



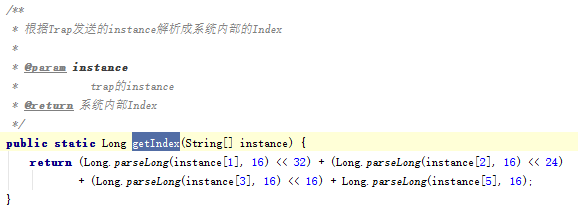
Index是cc侧定义

## getEponPortFromIndex



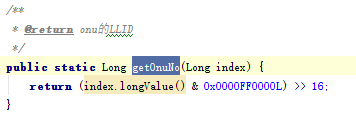
网管侧内部定义的EponIndex,不用处理

## getIndex



返回网管内部定义的EponIndex，不用处理

## getOnuNo



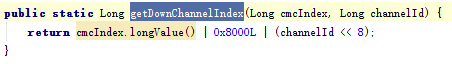
网管侧内部定义的EponIndex,不用处理

## getUpChannelIndex



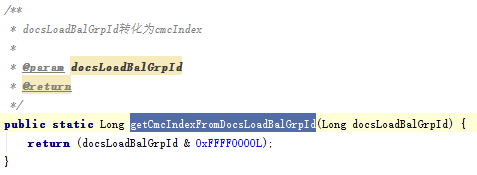
Index是cc侧定义

## getDownChannelIndex



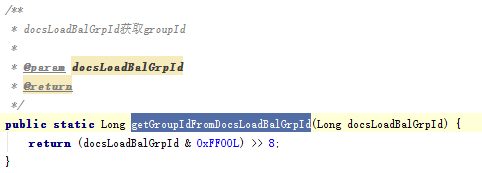
Index是cc侧定义

## getCmcIndexFromDocsLoadBalGrpId



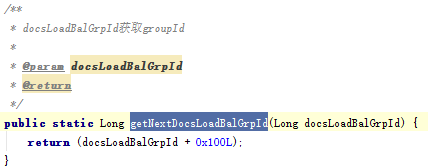
Index是cc侧定义

## getGroupIdFromDocsLoadBalGrpId



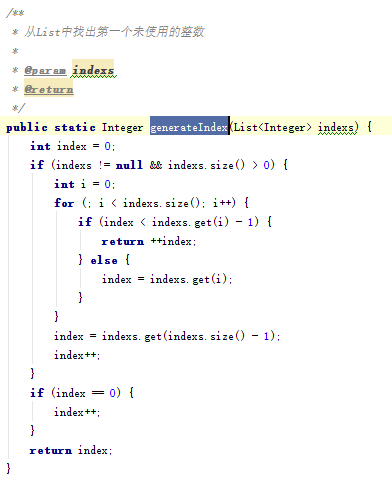
Index是cc侧定义

## getNextDocsLoadBalGrpId



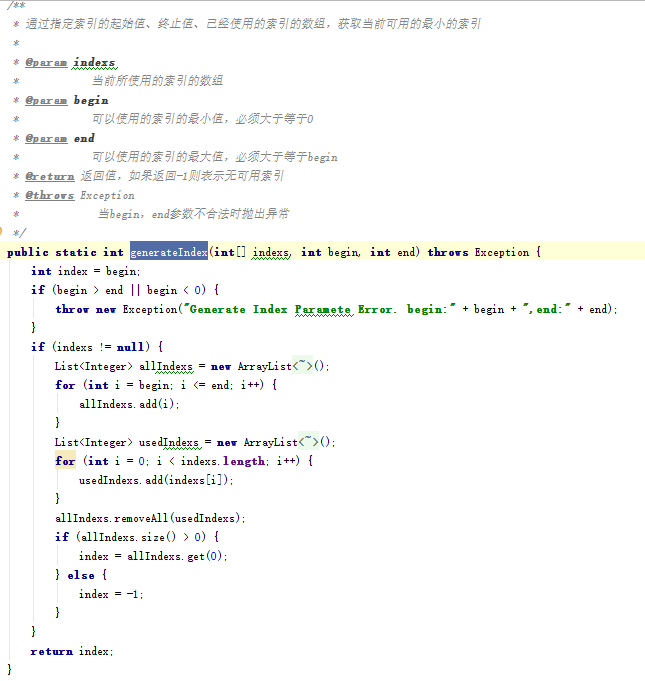
Index是cc侧定义

## generateIndex



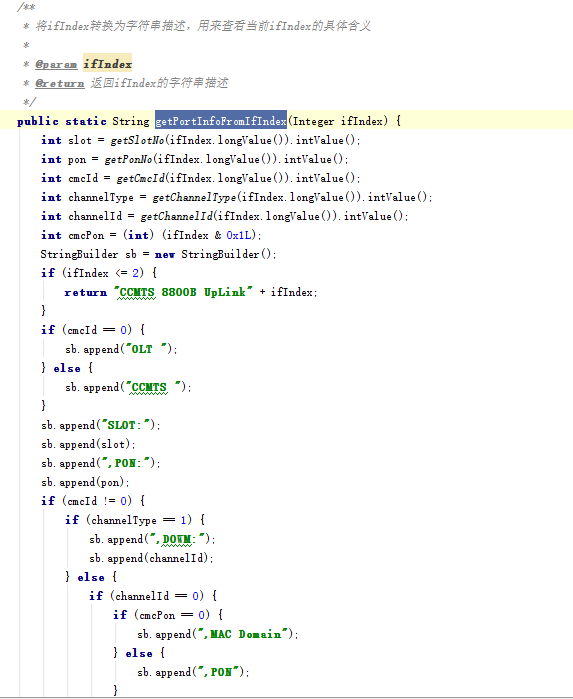
无需修改

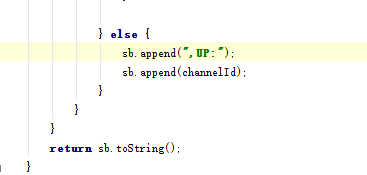
## generateIndex 2



无需修改

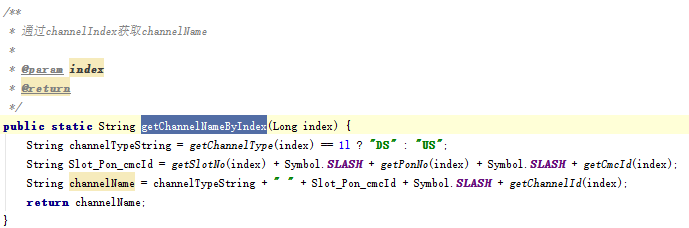
## getPortInfoFromIfIndex





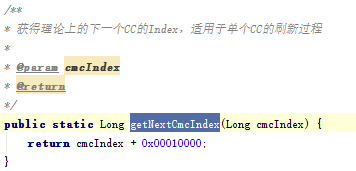
Index是cc侧定义

## getChannelNameByIndex



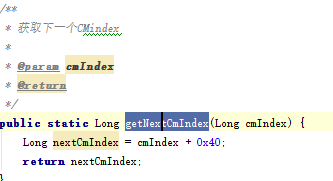
Index是cc侧定义

## getNextCmcIndex



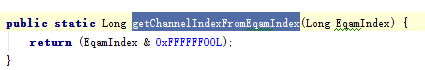
Index是cc侧定义

## getNextCmIndex



Index是cc侧定义

## getChannelIndexFromEqamIndex



Index是cc侧定义

## getSessionIdFromSessionIndex



Index是cc侧定义

## 修改方案

1. 数据库中存储index的是bigint（20），可以存储一个64位的数
2. 设备侧定义的index都是32位的，新老版本都是32位，不确定会不会扩展成64位
3. 我们可以在拓扑阶段将获取到的cmcIndex、channelIndex、cmIndex转换成一个64位的数，最高位必须为1
4. 编码版本（version）、设备分类（category）（RMD RMDC）、设备类型（deviceType）（E F）、索引类型（indexType）(cmcIndex、cmindex、channelIndex)
5. Flag（1）（1）version（3）（2-4）reseved（2）（5-6）category（2）（7-8）reseved（3）（9-11）deviceType（5）（12-16）reseved（2）（17-18）indexType（2）（19-20）reseved（12）（21-32）realIndex（32）（33-64）
6. Version取值只能是001-111，先约定老的编码version取值001，新版本version取值010
7. Category取值00=RMD，01=RMDC
8. deviceType取值

00000=30022(CC8800F)

00001=(CC8810F)

00010=30024(CC8820F)

1. indexType取值 00=cmcIndex，01=channelIndex，10=cmIndex
2. 保持兼容，在每个cc侧定义的方法处，加入一个判断64位最高位是不是1的判断，如果为1表示用新的网管侧CC编码规范，不为1表示为纯CC侧编码规范，是由老版本拓扑发现升级为新版本网管的情况

long index = -9223372036568449024L;

long tmp = index & 0x8000000000000000L;

if (tmp == 0x8000000000000000L) {

System.out.println("TestFFT.main");

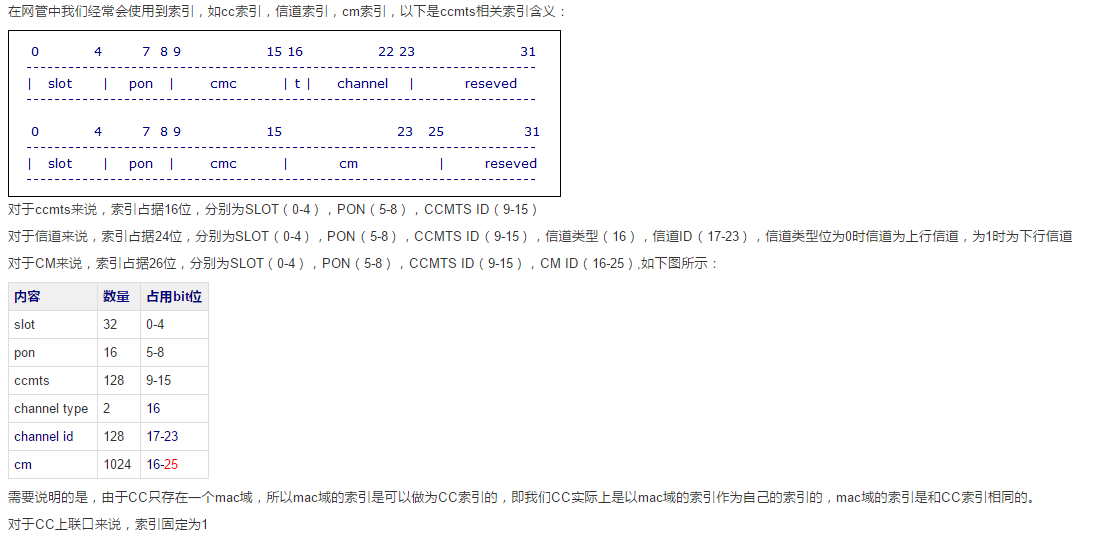
} else {

System.out.println("TestFFT.main1");

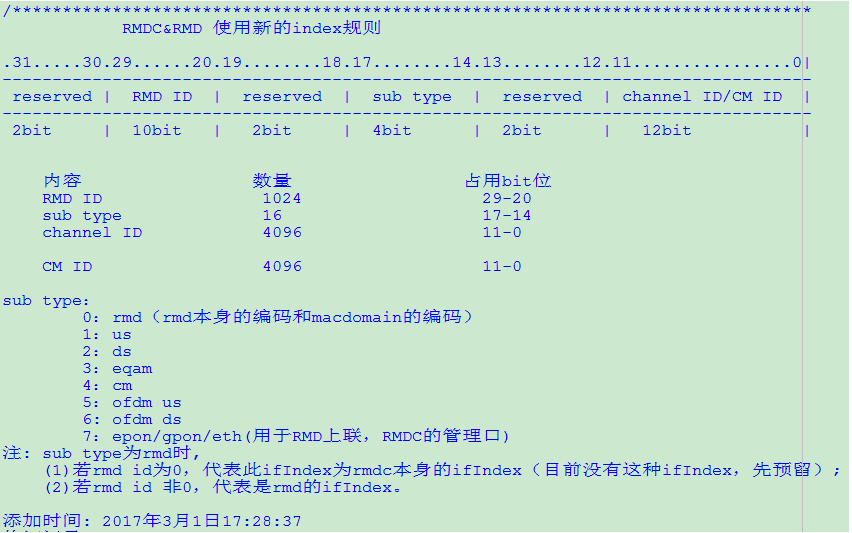
}

1. 新版网管侧CC编码又分为001版编码和010版编码格式，分别对应V3.0.0之前的编码格式和V3.0.0之后的编码格式

001版，此格式只说明realIndex（33-64）字段部分的编码方式



010版，此格式只说明realIndex（33-64）字段部分的编码方式



1. 提供一个内存缓存的cmcId获取设备类型的方法
2. 获取cmcIndex、channelIndex、cmIndex的地方可以通过cmcId获取一个网管侧ccindex生成器，生成器可以将一个domain或一组domain进行转换，将其中需要被转换的index字段转换成一个网管侧cc index，转换可以放在engine端get、gettable、gettableline、gettablelines、getnext被调用的地方来实现
3. 要转换的index做一个列表，发现需要转换的就直接转了，保证采集完之后不管是入库还是直接前端使用都是转换过的网管侧cc index
4. 修改cmcIndexUtils，每一个传入是cc index的方法都需要加入一个判断逻辑
   1. 第64为高位是不是1，不是1使用000编码版本的解析类（即纯cc侧32位老编码规则）处理，是1则走b步
   2. 判断编码版本，通过编码版本找到对应版本的解析类进行编码处理