

运营商抓取完整性和增量抓取服务文档

陈雕

2016 年 12 月 11 日

目 录	2
-----	---

目录

1 概述	3
2 抓取完整性	3
3 增量抓取	4

1 概述

按照完整性的维度划分, 可以将抓取完整性划分为 **空间完整性**和 **时间完整性**.

时间复杂度 指的是诸如账单/通话详单/短信详单抓取的月份的完整性

空间复杂度 指的是诸如每个月的详单记录数/用户个人信息的各个字段的完整性

抓取完整性一方面可以用来衡量抓取服务的质量, 另一方面还可以用来做增量抓取. 所谓增量抓取, 即当同一用户多次抓取时, 后面只抓取尚未抓取过的数据, 而复用已经抓取过的数据. 可以设想, 当已抓取数据足够完整(包括时间和空间数据), 通过增量抓取几乎没有任何抓取代价.

2 抓取完整性

当前的运营商抓取完整性服务监测了运营商抓取账单/通话详单/短息详单的时间和空间完整性. 相关信息存储在 **MYSQL** 数据库中. 入库时只插入, 不更新, 因此该库可以记录每次抓取的完整性数据细节. 相关数据库建表语句为:

Field	Type	Null	Key	Default	Extra
id	bigint(20) unsigned	NO	PRI	NULL	auto_increment
mo_phonedata_id	bigint(20)	NO	MUL	0	
user_source	varchar(32)	NO	MUL		
crawler_channel	tinyint(4)	NO		0	
session_id	varchar(64)	NO	MUL		
month	varchar(10)	NO			
bill_type	tinyint(4)	NO		0	
total_records	int(11)	NO		0	
crawled_records	int(11)	NO		0	
create_time	timestamp	NO	MUL	CURRENT_TIMESTAMP	
update_time	timestamp	NO		CURRENT_TIMESTAMP	on update CURRENT_TIMESTAMP

该表需要说明两点:

1. session_id. 当前的抓取框架还没有发展出 session id, 因此目前使用 uid 代替 session id. 注意到, 使用 session id 可以记录每个 session 的抓取完整性数据. 但是每个抓取 session 的 uid 时相同的, 因此当前的 uid 并不能代替 session ID 的功能, uid 只是 session ID 的占位符

2. `total_records`. 该表使用了 `total_records` 和 `crawled_records` 两个字段来记录抓取的空间完整性. 其中 `crawled_records` 用于记录用户实际抓取到的记录数, `total_records` 用于记录运营商返回的该用户的总记录数. 然而并非所有运营商都会返回该数据, 此时默认 `total_records = crawled_records`. 当前各运营商并没有抓取运营商总记录数, 因此当前总是 `total_records = crawled_records`. 我们会在后面的抓取中对此进行改进.

3 增量抓取

增量抓取依赖于抓取数据的完整性.

当抓取服务接收到用户的抓取请求时, 会首先查询用户的抓取完整性数据. 以通话记录抓取为例, 设期望抓取的月份为 `Em` (`Expected Months`), 已抓取的月份为 `Cm` (`Crawled Months`), 则增量抓取会取两者的差集. 即抓取月份

$$M = Em - Cm$$

差集抓取的核心在于抓取完整性的判断. 当前的抓取服务尚难以精确判断一个月抓取数据是否完整. 因此当前采用如下规则进行判断:

1. 该月通话记录为 0, 则该月抓取不完整, 需重新抓取
2. 该月 `total_records != crawled_records`, 则该月抓取不完整, 需重新抓取
3. 当前月数据默认总是不完整, 需要重新抓取
4. 用户显式指定不强制更新当前月时, 可以不每次重新抓取当前要详单. 该规则适用于每个月详单均需要短信验证码的运营商
5. 该月 `total_records` 为整百, 则该数据较为可以, 可根据情况决定是否重新抓取

为了避免频繁读取完整性数据库, 在应用中使用 `redis` 缓存用户完整性数据. 使用缓存时, 数据一致性是不得不考虑的问题. 一旦抓取服务抓取了新的账单或详单数据时, 就需要对缓存进行更新. 当前的缓存更新策略是,

一旦有新的抓取发生时, 就将该数据从缓存中删除. 因此一次抓取可能会两次读库, 一次是抓取账单时读取完整性数据; 第二次是抓取详单时, 该用户缓存已被删除, 则会重新读取数据库中的完整性信息 (当抓取详单时, 该缓存尚未删除, 则不用再次读库).