

TiDB TTL Table - T4

Distilling massive data, over time

T4 - Jiazhi Wen / Mengyu Hu / Xiaoguang Sun

BigData Scale Transactions

with Internet of Everything

TiDB Hackathon 2020



Value Price



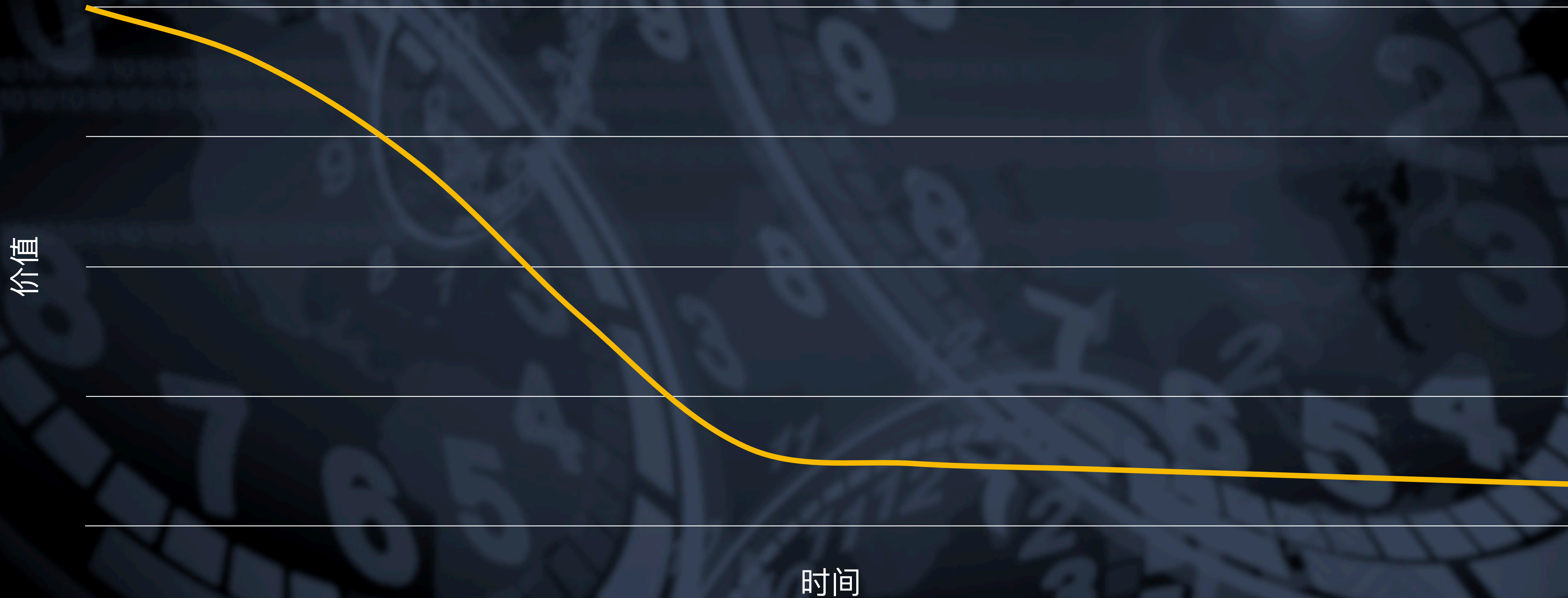
Value

Price



可选方案

- 降低成本
 - 分级存储
 - 空间优化
- 提升单位价值
 - 高价值数据
 - 价值蒸馏



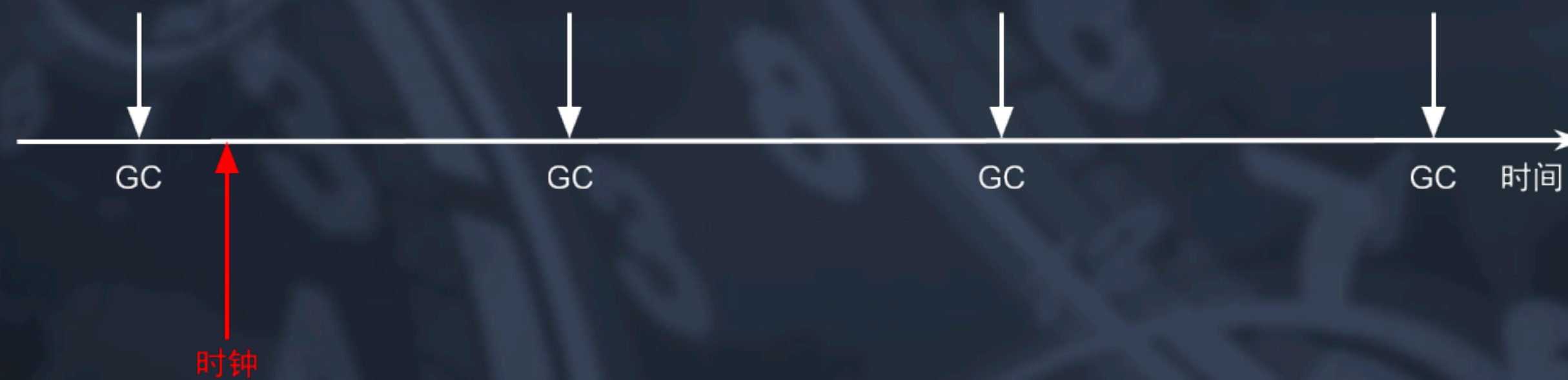
TTL Table

- 数据生命周期管理
- 回收策略
 - 按行
 - 按分区

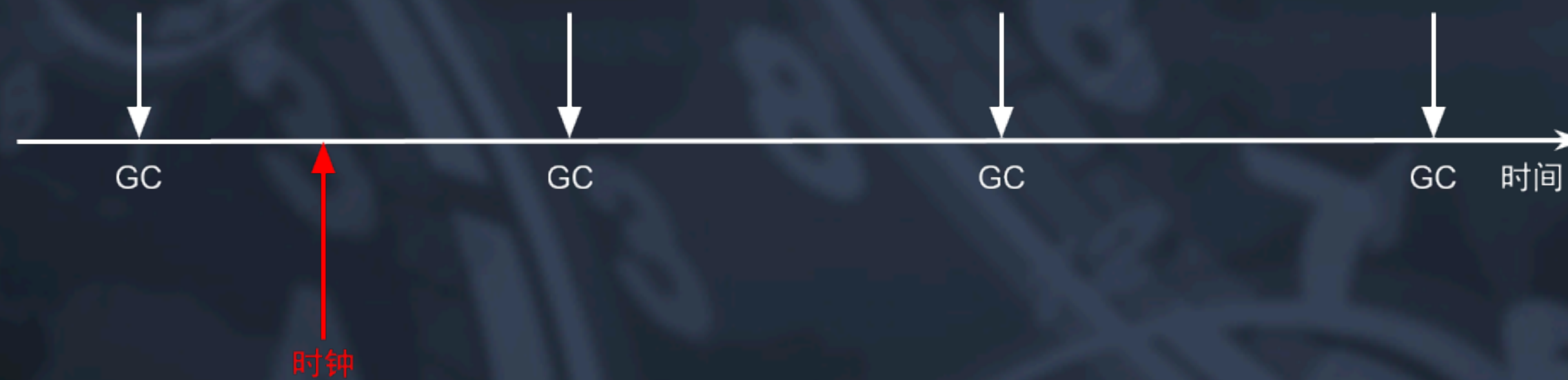
回收策略 - 按行

```
CREATE TABLE ttl_table {  
    id varchar(255),  
    author varchar(255),  
    content varchar(65535),  
    PRIMARY_KEY(id)  
} TTL='10m' TTL_GRANULARITY='ROW';
```

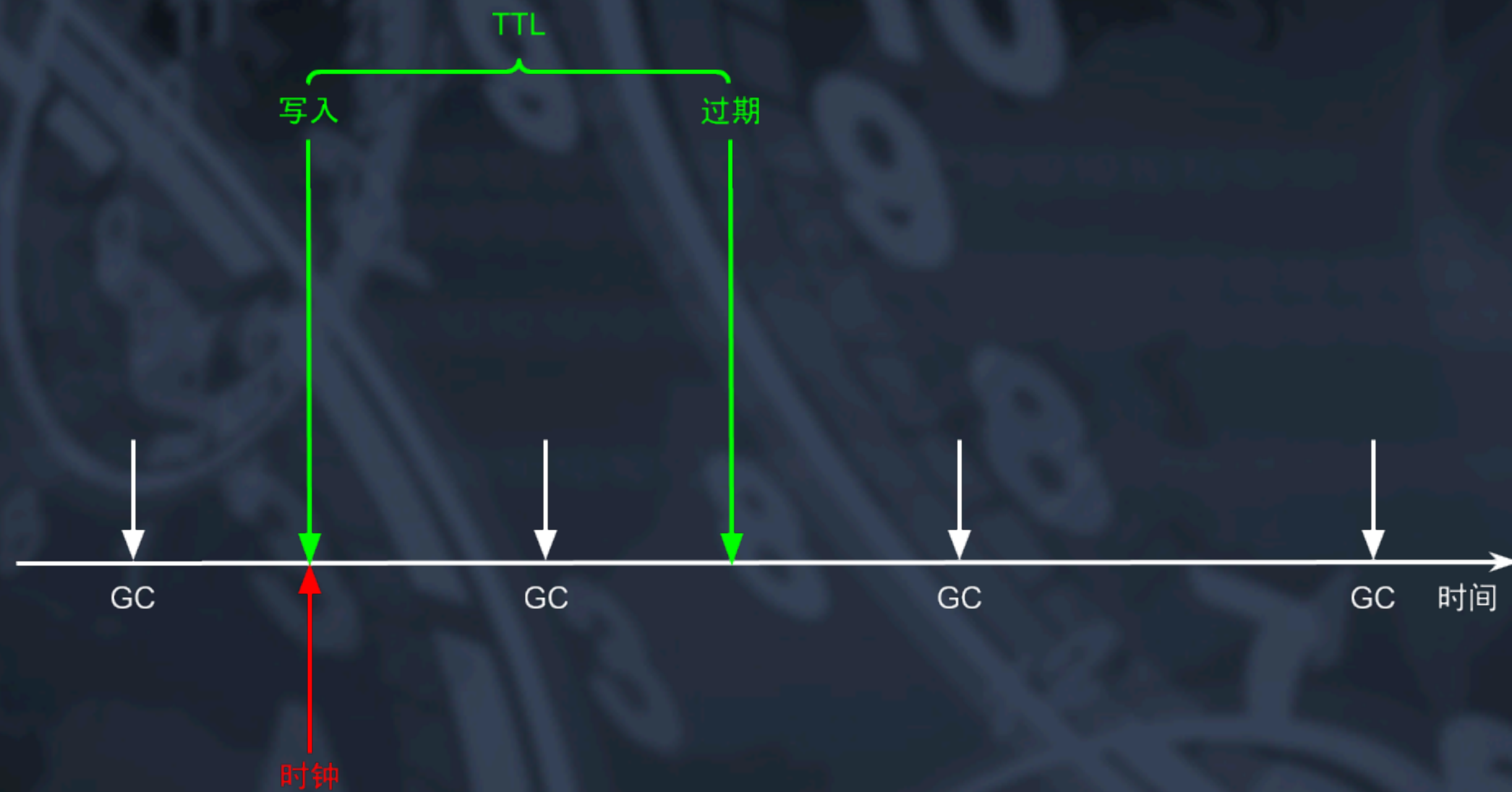

回收策略 - 按行



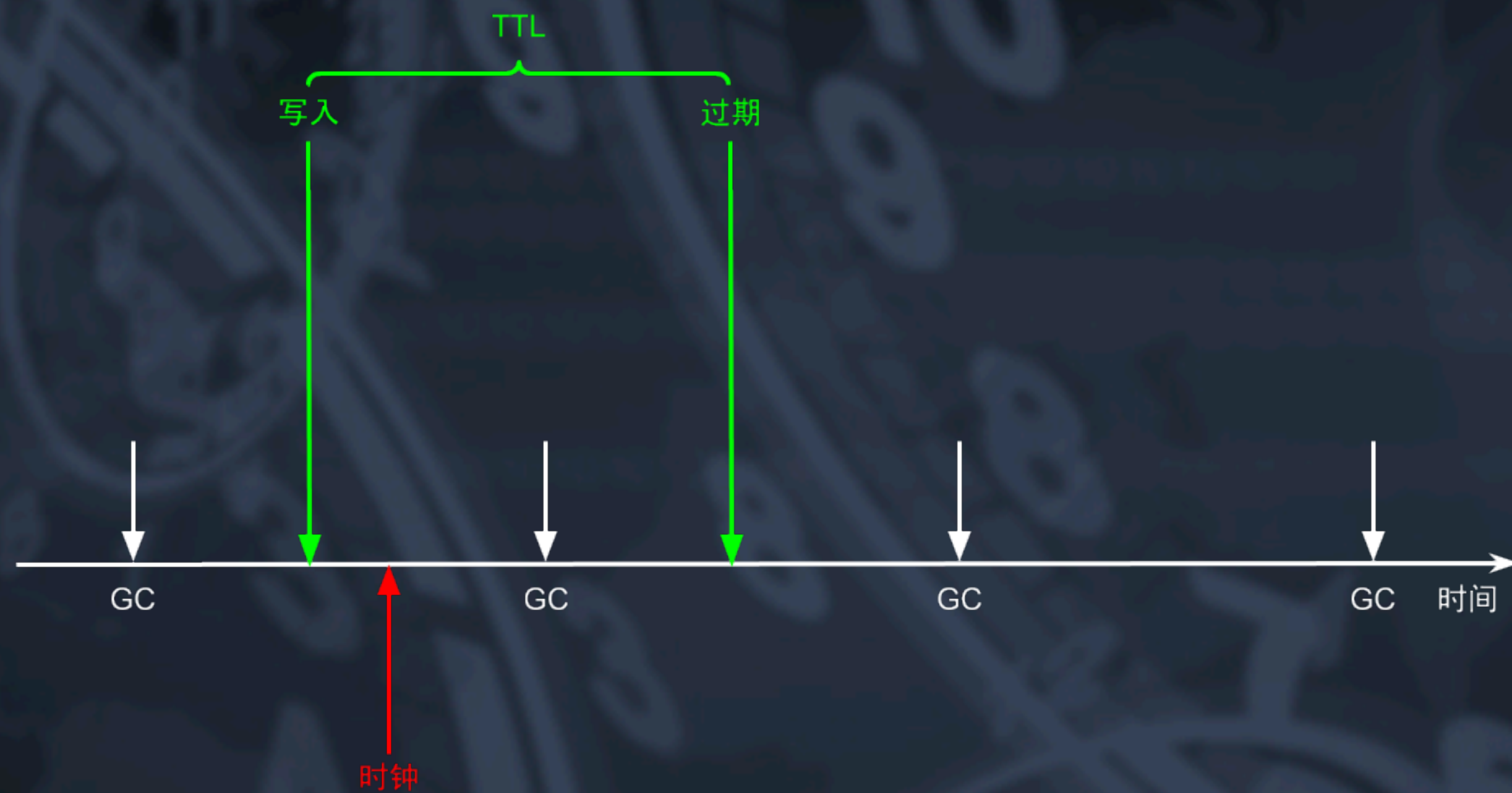
回收策略 - 按行



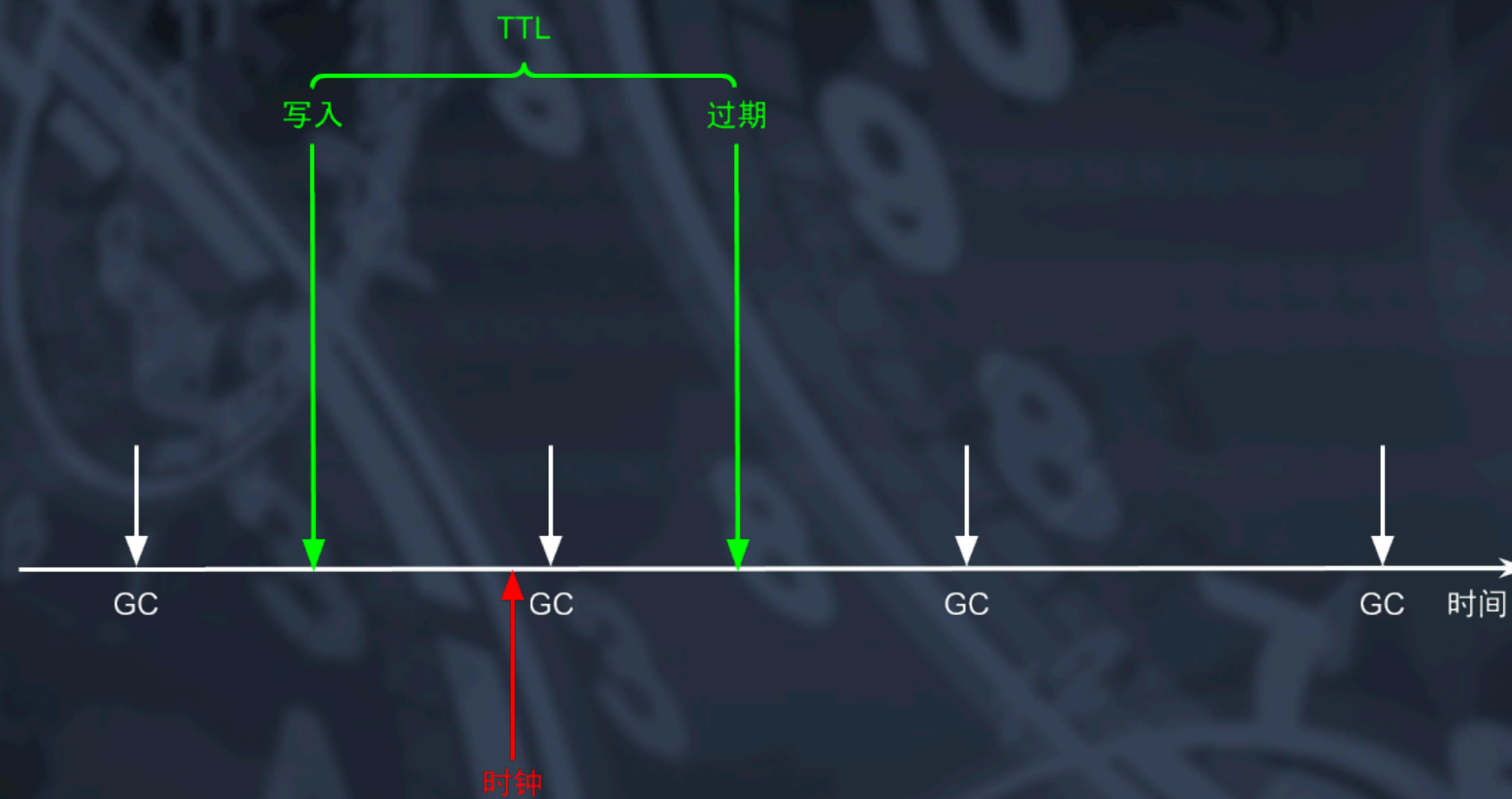
回收策略 - 按行



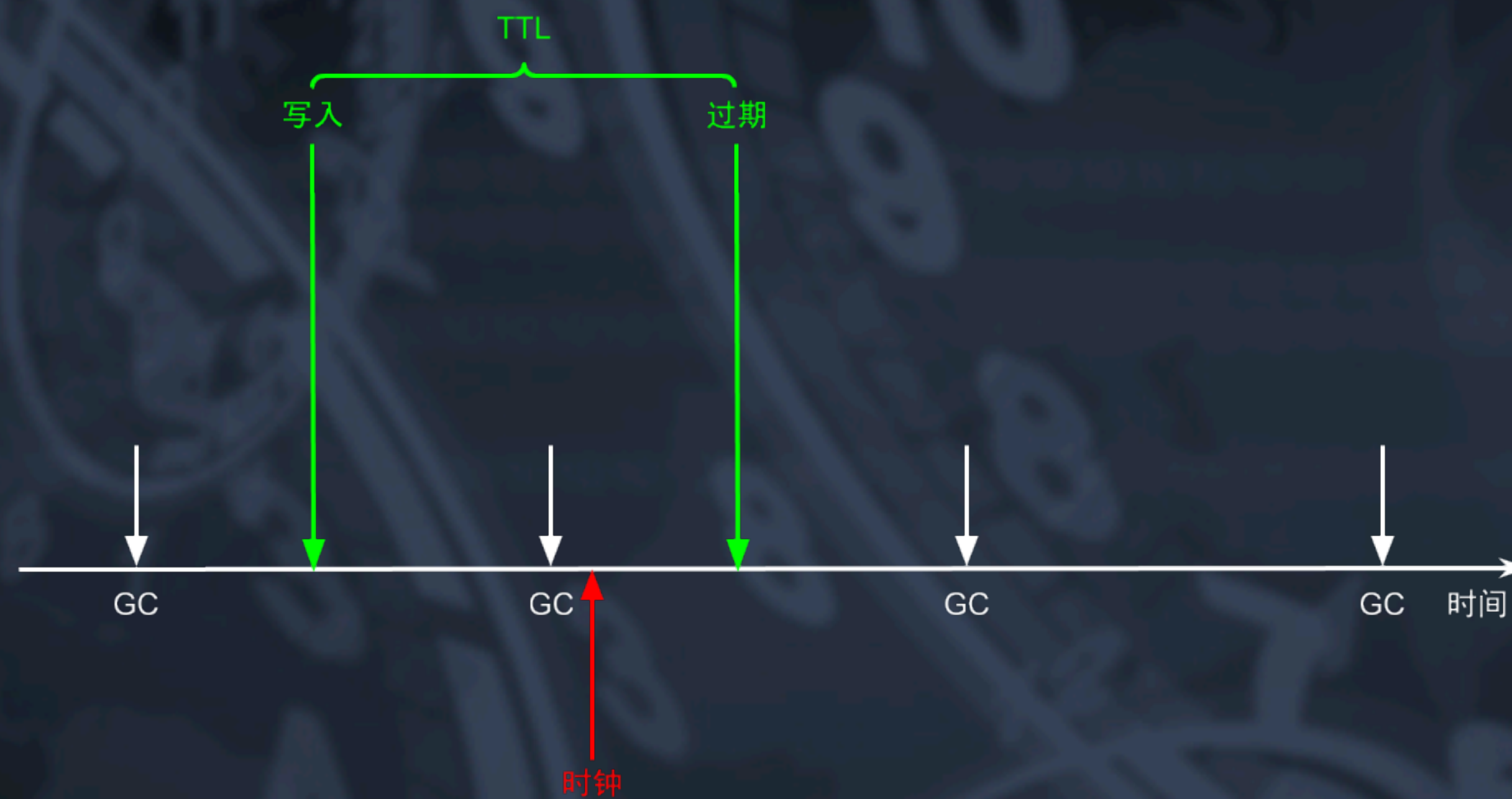
回收策略 - 按行



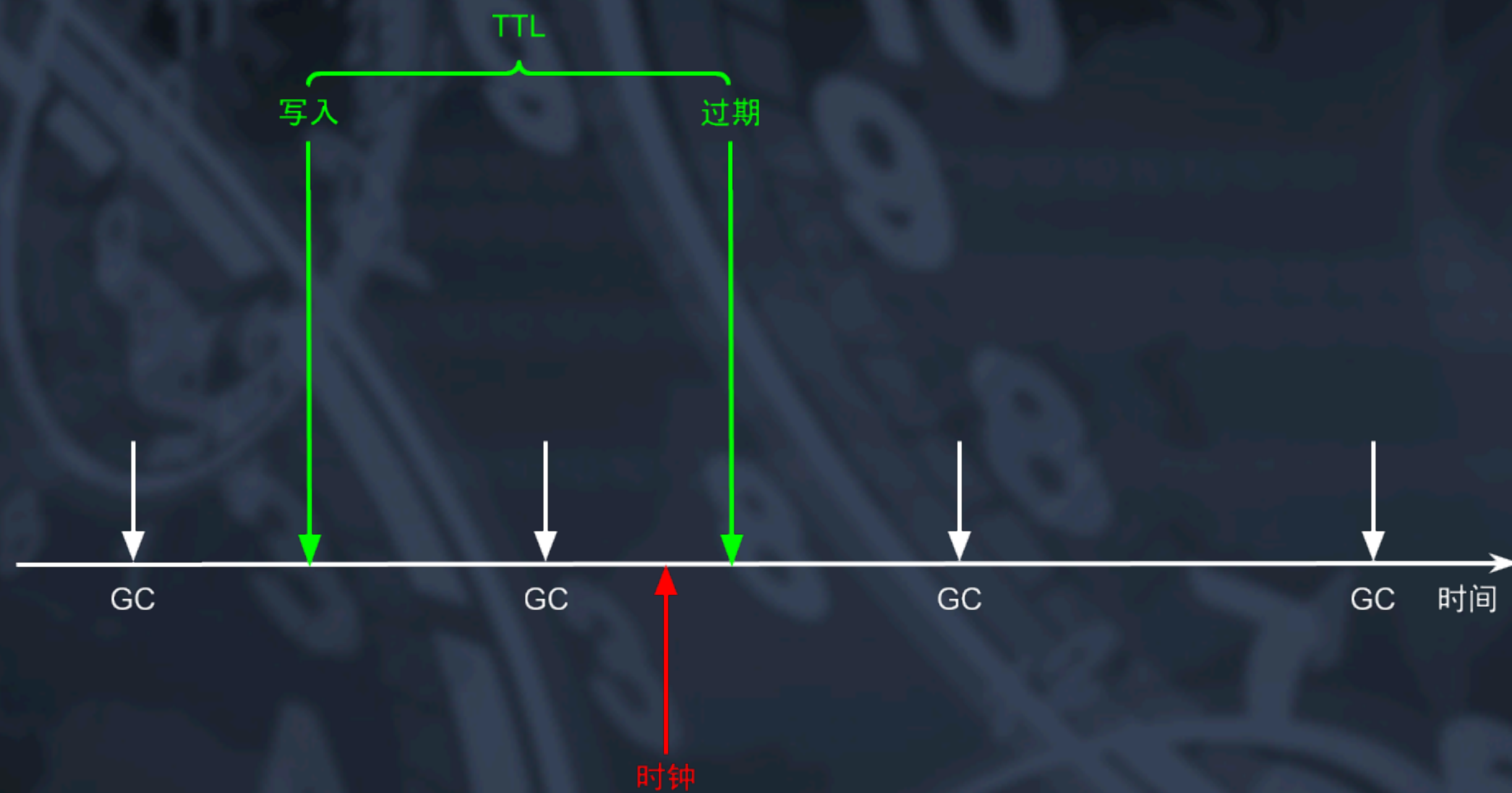
回收策略 - 按行



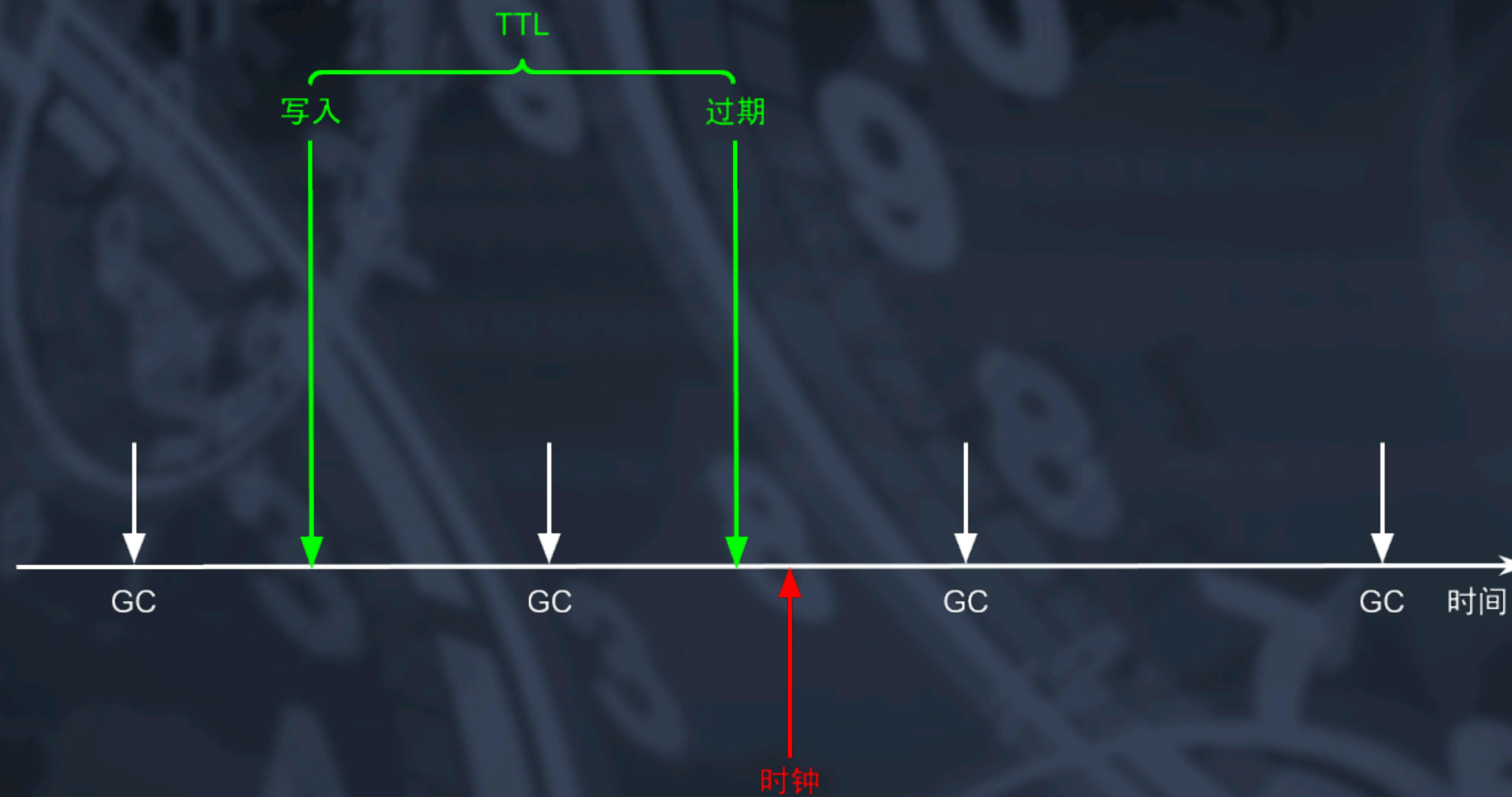
回收策略 - 按行



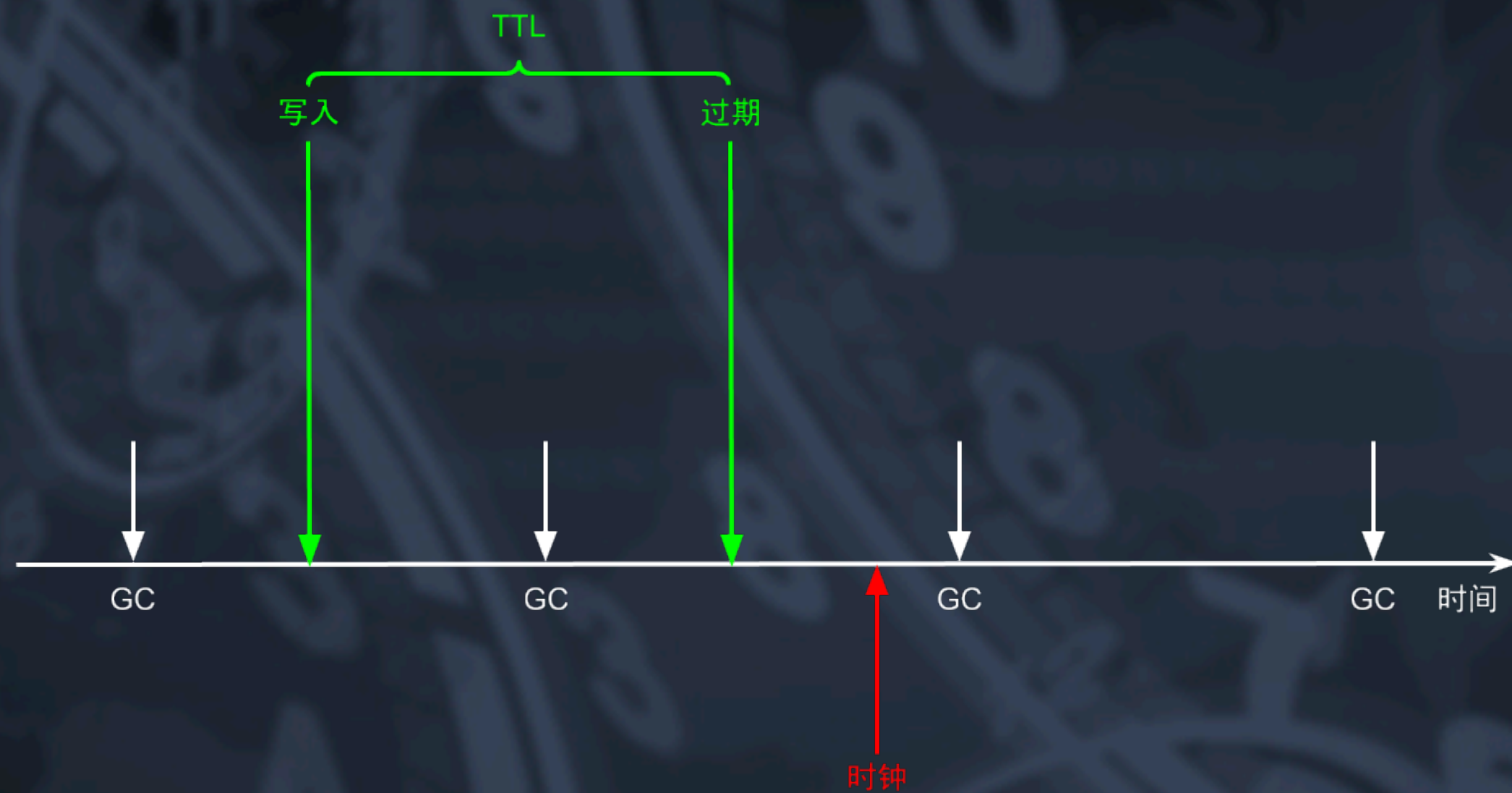
回收策略 - 按行



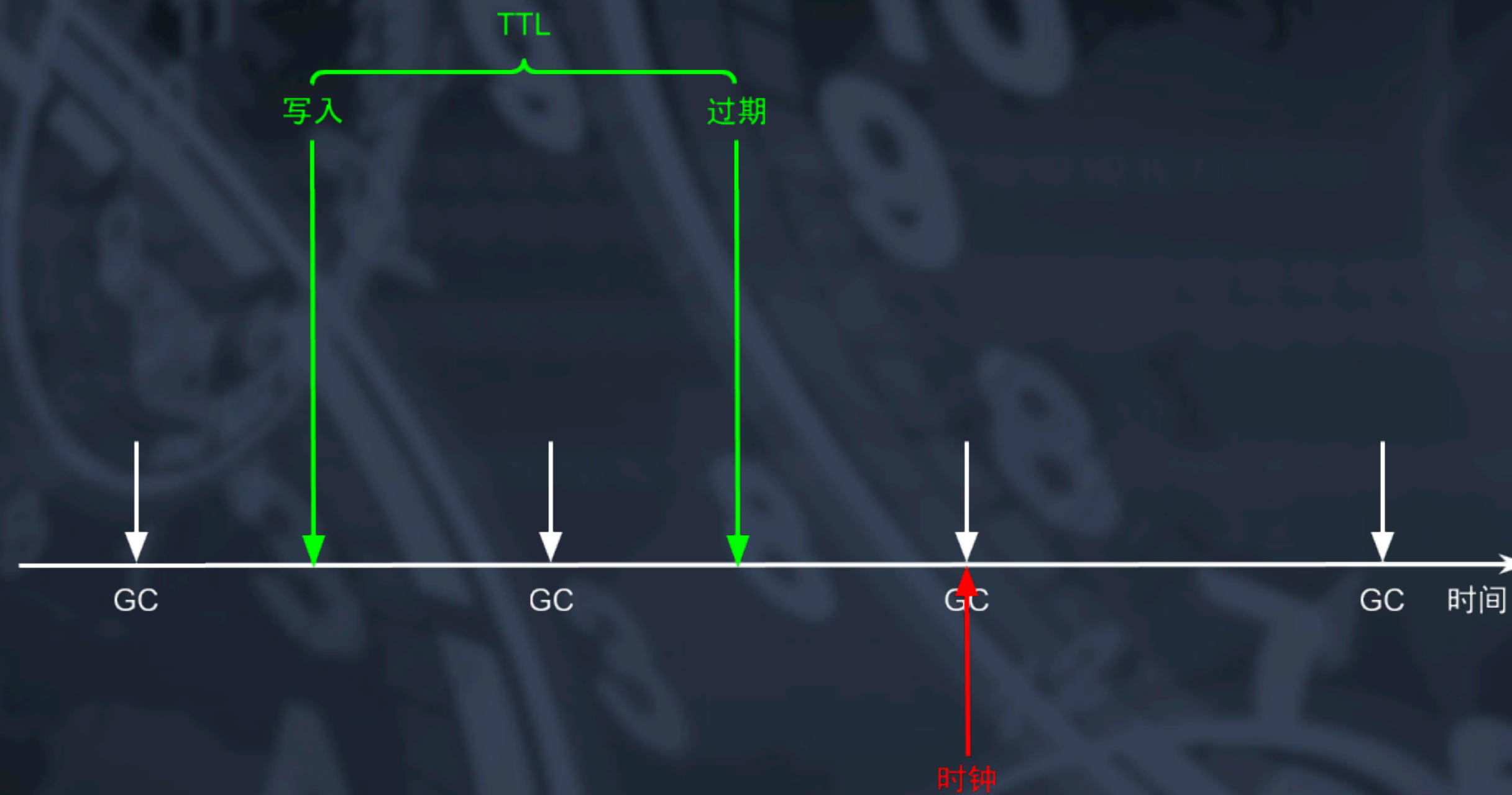
回收策略 - 按行



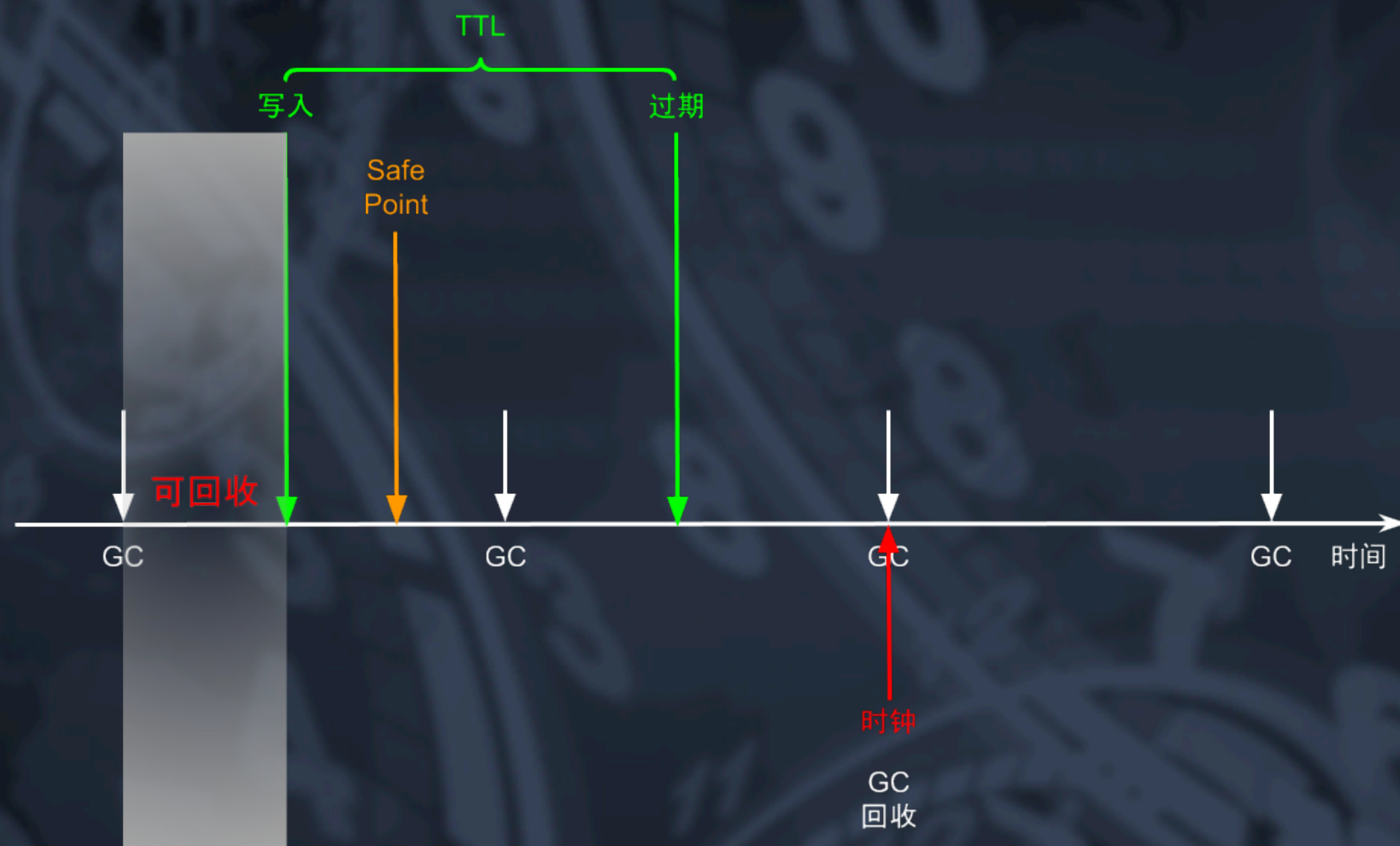
回收策略 - 按行



回收策略 - 按行



回收策略 - 按行



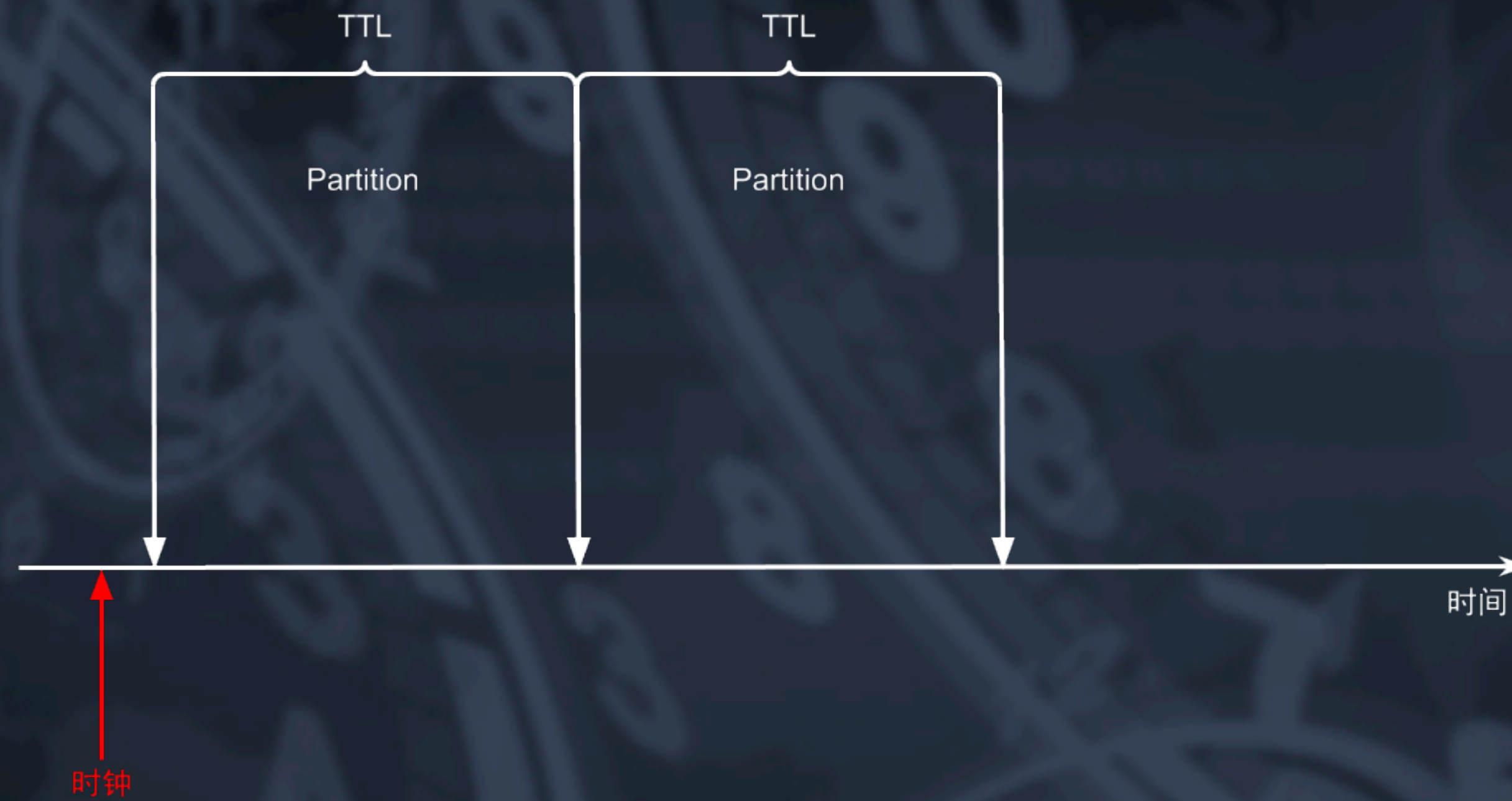
回收策略 - 按行



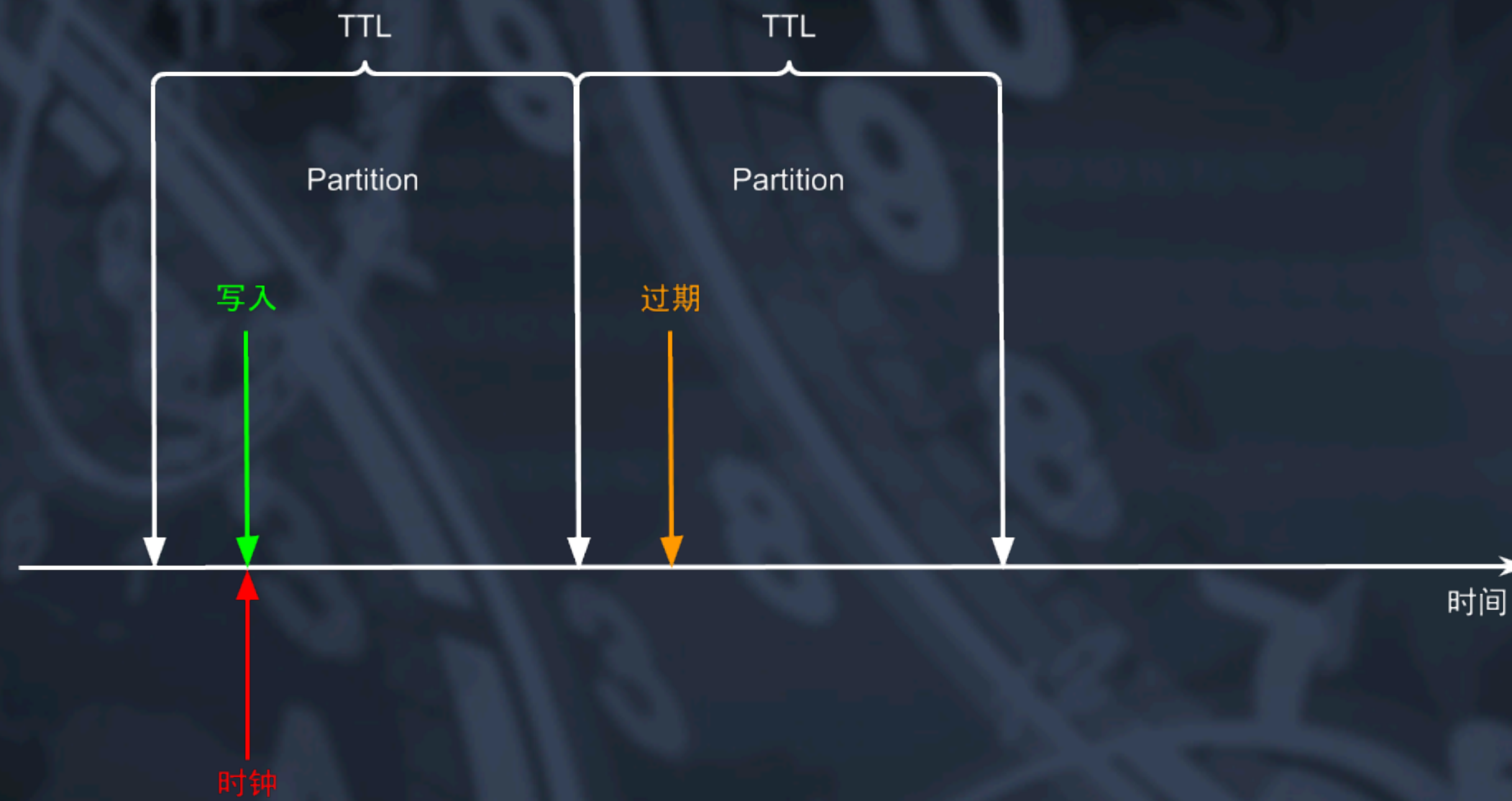
回收策略 - 按分区

```
CREATE TABLE ttl_table {  
    id varchar(255),  
    author varchar(255),  
    content varchar(65535),  
    PRIMARY_KEY(id)  
} TTL='10m' TTL_GRANULARITY='PARTITION';
```

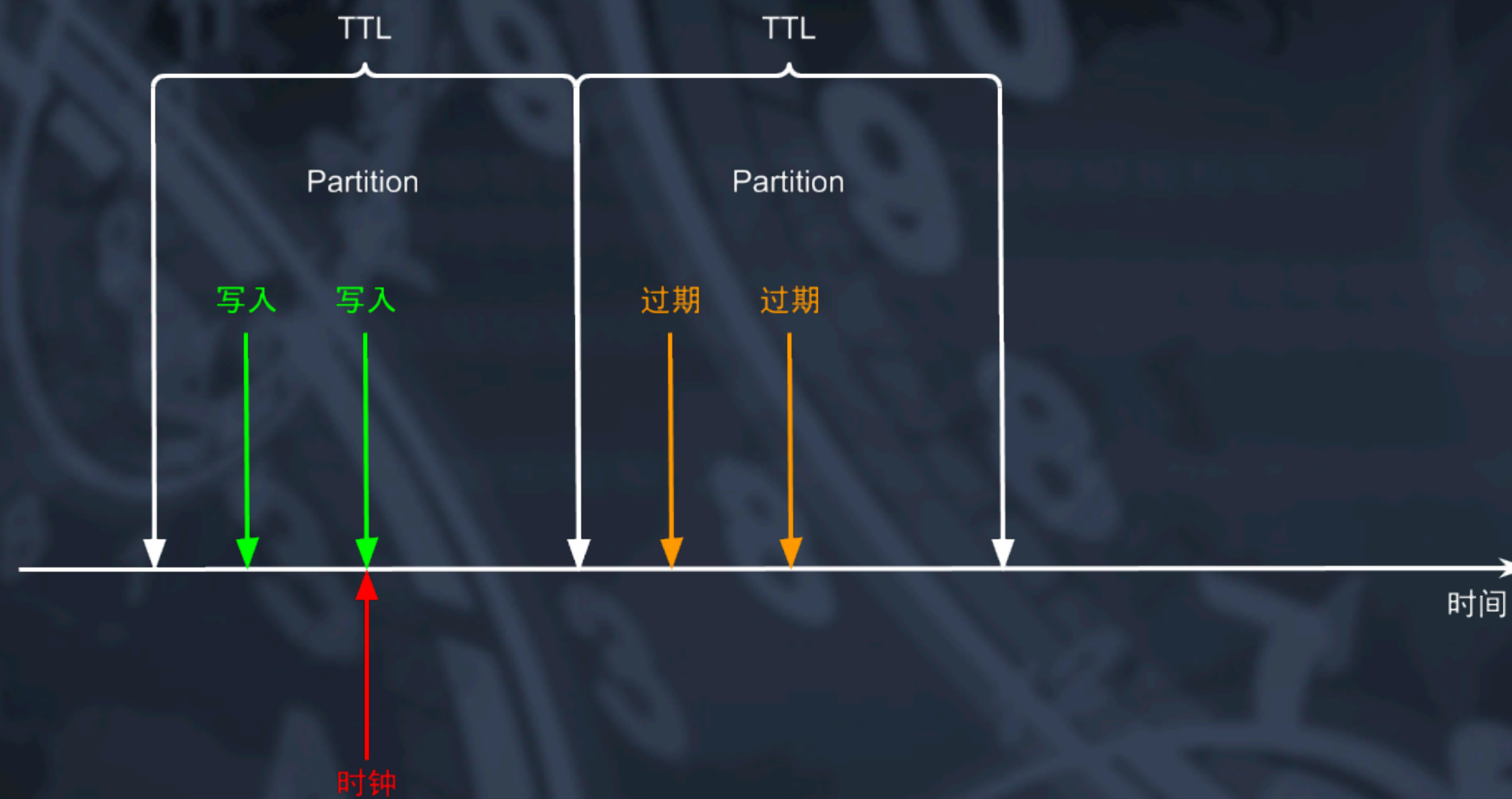

回收策略 - 按分区



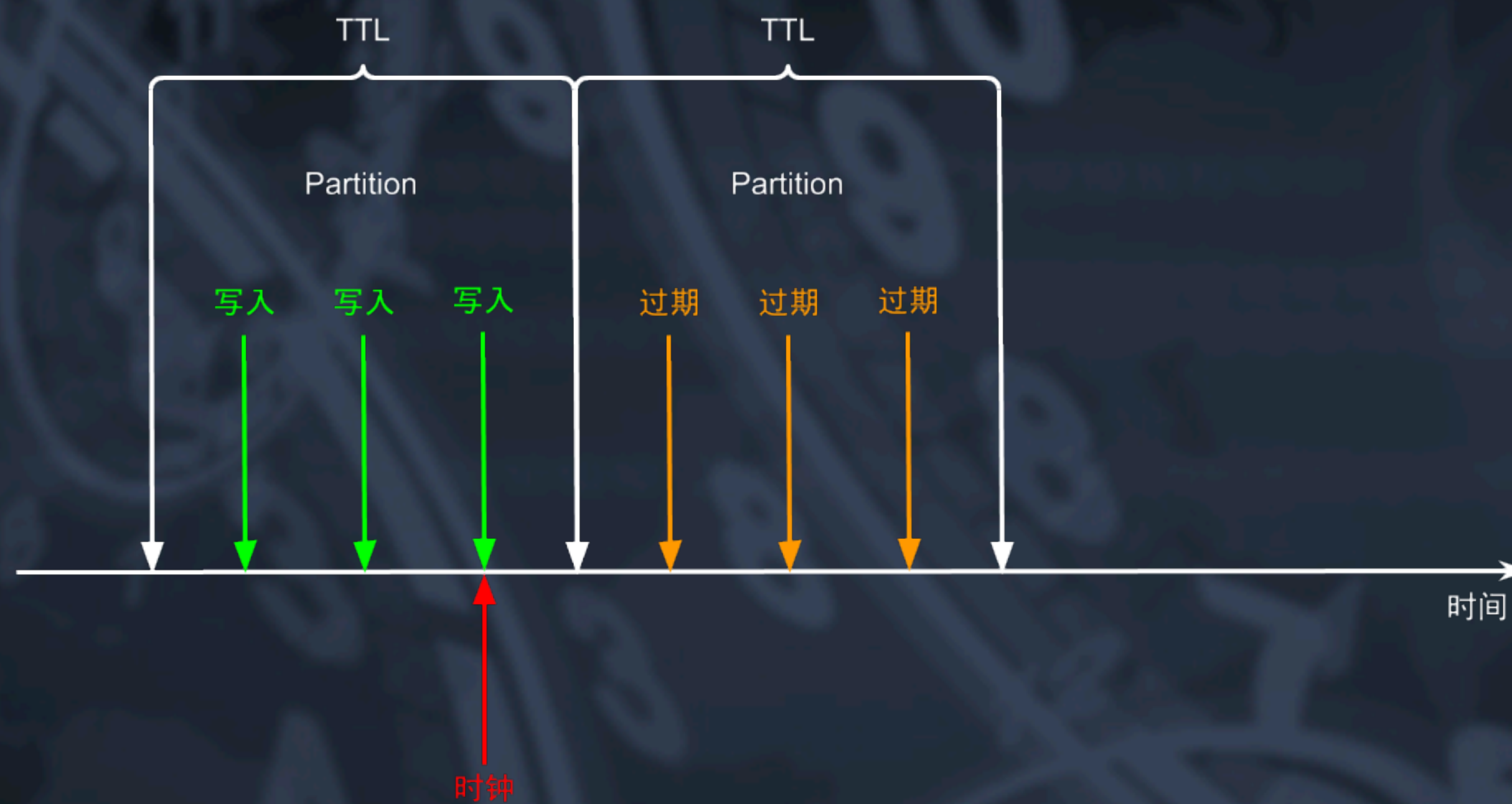
回收策略 - 按分区



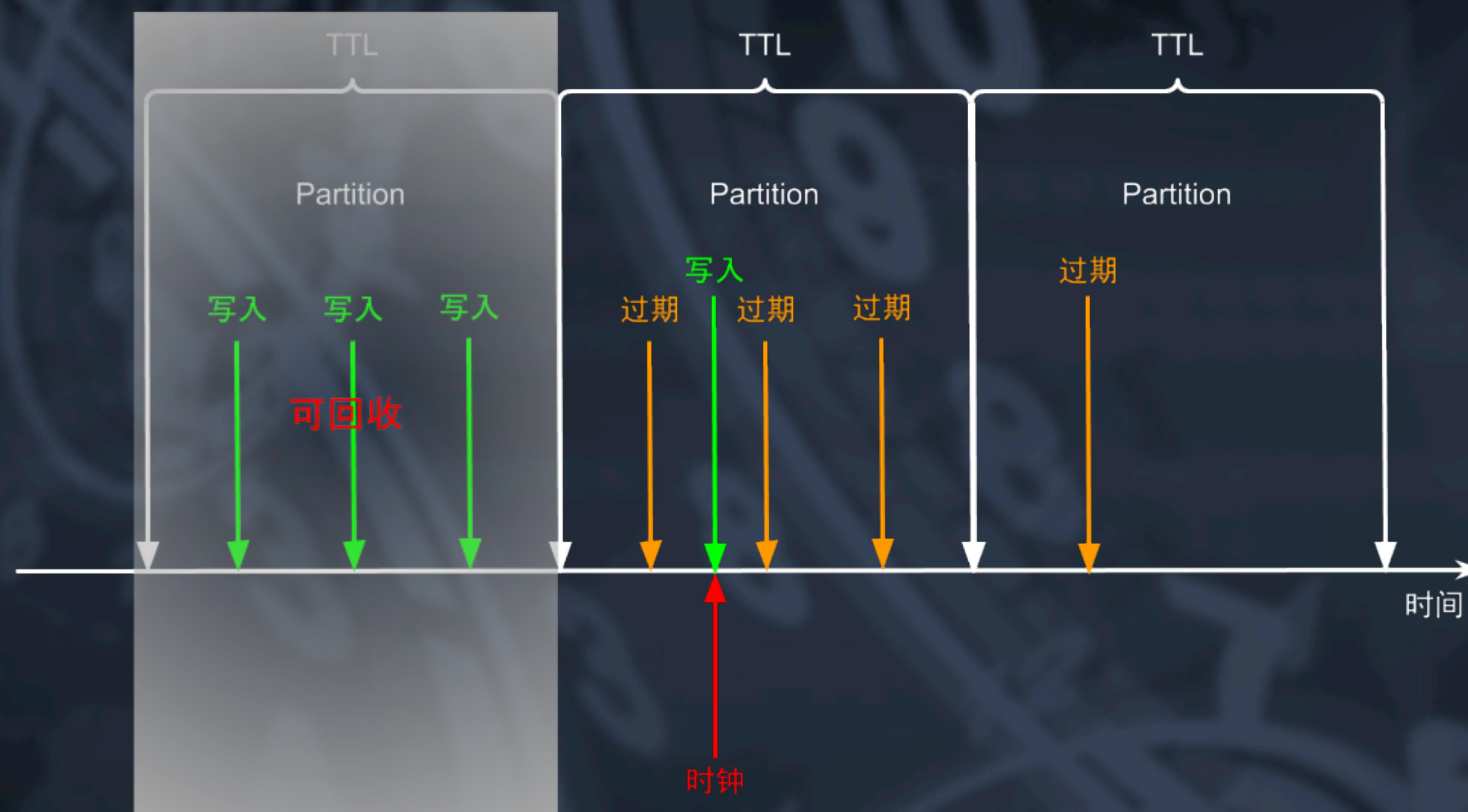
回收策略 - 按分区



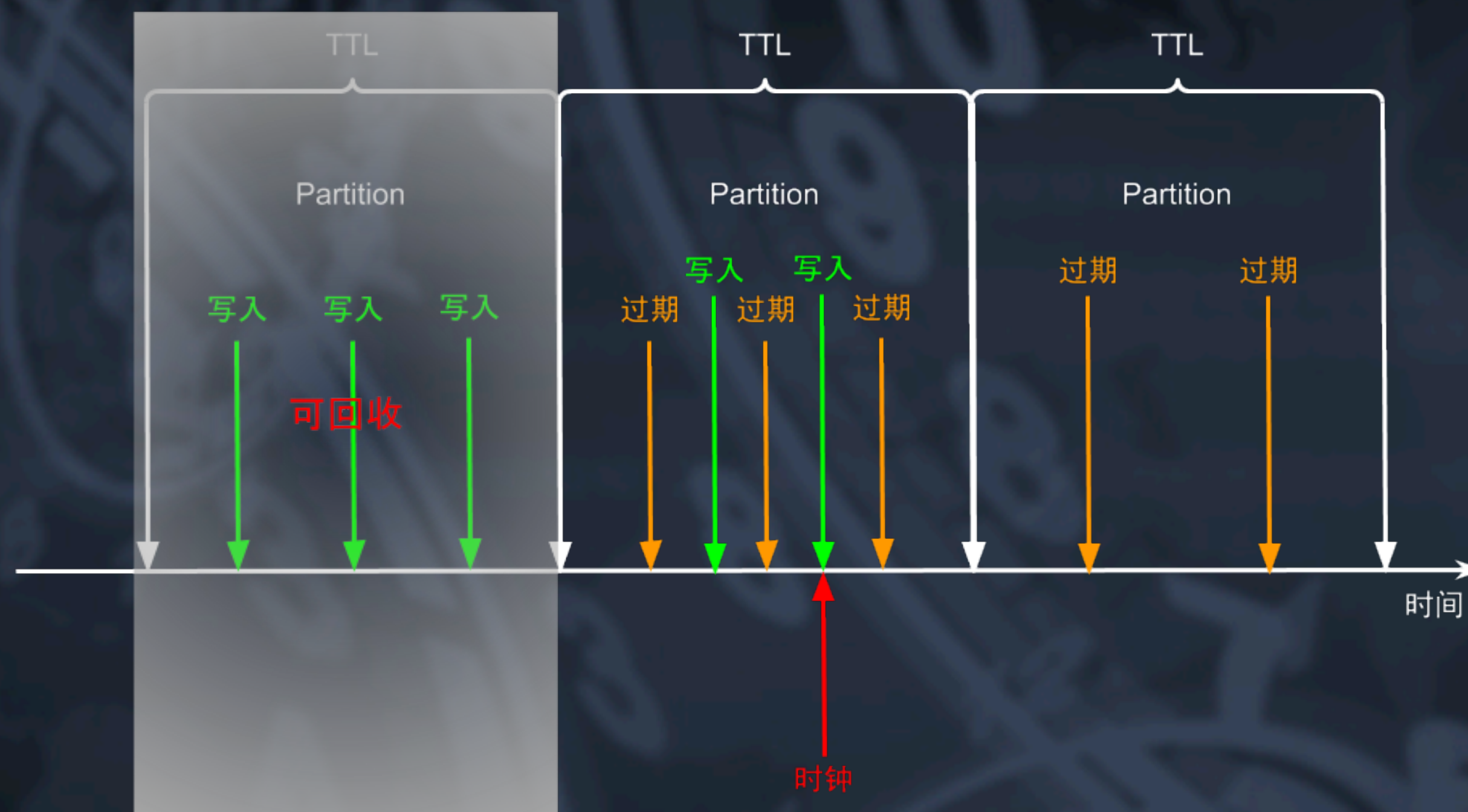
回收策略 - 按分区



回收策略 - 按分区



回收策略 - 按分区



应用场景 - 维度报表



- 决策支撑
- 横向扩展
- 字典编码
- 列存储
- 历史报表自动过期

<https://github.com/tidb-hackathon/kylin>



应用场景 - Distributed Tracing



- 高吞吐写入
- 易维护
- 历史跟踪自动过期

<https://drive.google.com/file/d/1qT8zoq-irvop7OQmpEAfiQgG8bbI-UYX/view>



应用场景 - Kubernetes long term event store



- No more - 'mvcc: database space exceeded'
- 跟踪历史，追溯问题
- 高性能
- 结构化查询能力
- 历史事件自动过期

<https://github.com/tidb-hackathon/kubernetes-event-exporter>

TiDB Hackathon 2020



应用场景 - 时效推送



- QoS 1 & 2 会话持久存储
- 新闻热点
- AMBER Alert
- 天气预报
- 更多 XXX 场景过期数据自动消失

<https://github.com/tidb-hackathon/emitter>



应用场景 - IoT



- 万物互联
- 数据爆炸
- 实时分析
- 短期分析
- 数据价值高速贬值
- Bring everything into TiDB, Affordably

TiDB - Unleash unlimited possibilities

Thanks

TiDB Hackathon 2020

