1.数据库完全自增：所有服务器连同一个db，在一张表中定义一个auto_increment自增的方法获得序号，这样保证不重复，这种方法每次获取id都需要连接一次db，效率比较低，可以用在低频交易系统。

2.数据库不完全自增：所有服务器连同一个db，db中用自增的方法获取一个序号，并设置步长，每台服务器每次从db中获取到一次序号后，保留在内存中，每次需要生产序号则从内存中序号递增直到步长使用完毕再从db中获取，这样以减少与db的交互。这种方案每次重启会丢弃部分序号。该方案不适合序号必须连续的场景。

3.时间+应用+数据库不完全自增（个人觉得也可以变为数据库完全自增）：如果对于交易记录这样交易id可能再实际应用中数据量比较大，建议增加时间的控制，而对于分布式多DB还可以将应用服务器再流水中记录，即给每台服务器分配一个id。最后该id的组成为：年月日+机器id+数据库不完全自增id。在此之上还可以再延伸修改数据库id为1，然后以1为基础增加。该方案的id记录长度较大，适合高频大数据量的业务，且不会发生数据库id到上限的风险。