

分布式系统全局唯一 ID 生成方案对比

51164500220

黎谢鹏

在业务系统中，各种资源都有唯一标识符的需求，比如数据库中的主键，操作系统中的进程 ID，消息队列系统中的消息 ID。通常，这个唯一标识符要求连续递增增长。数据库中的 auto_increment 是最常见的做法，利用数据库本身提供的功能就能满足连续递增唯一的需要。但随着业务系统的发展，分布式高并发高可靠的需求也日益增长，这时仅仅满足于单库单表的 auto_increment 就显得捉襟见肘了。我们知道，数据库自增长的方案有两个劣势：一是高并发性能不好，主键生成的性能上限限制了单机的性能，二是扩展性差，比如一主多从的模式下无法保证唯一性。

于是产生了分布式环境下的全局唯一 ID 的各种方案。通常全局唯一 ID 需要满足以下要求：

- 保证 ID 全局唯一性
- 数据在分片间的相互迁移不会受到 ID 的限制
- ID 能够附带时间信息，有利于利用时间排序
- ID 存储空间限制在 64bits 以下
- ID 服务要有较高性能
- ID 服务能够避免单点故障

方案	简介	优点	缺点	容量	性能
UUID	<p>全称 universally unique identifier, 128bit, 别称 GUID。</p> <p>UUID 通过 MAC 地址, 时间戳, 名字空间, 随机或伪随机数, 时序等来保证唯一性。</p> <p>UUID 具有 5 个版本, 每个版本的算法不同, 应用范围也不同。</p> <p>格式: 123e4567-e89b-12d3-a456-426655440000</p>	<p>1.本地生成, 不依赖远程调用</p> <p>2.java 自带, 版本 4</p> <p>3.水平扩展能力强</p>	<p>1.128bits, 占用空间大</p> <p>2.类型为字符串, 索引效率低</p> <p>3.无法保证趋势递增。</p> <p>4.用户不可读</p>	2^{128}	1billion/s 大约 85 年产生一次冲突, 可忽略
Flickr	<p>flickr 巧妙地使用了 MySQL 的自增 ID, 及 replace into 语法, 实现了分片 ID 生成功能。</p> <p>一个 Flickr 票服务器是一个专用数据库服务器, 不同长度的 id 由不同的表生成 Tickets32 生成 32-bit IDs, Tickets64 生成 64-bit IDs。</p>	<p>1.开发成本低</p>	<p>1.ID 没有包含时间戳和分片信息等</p>	2^{64}	相对于 mysql 的 auto_increment 线性增长
Twitter Snowflake	<p>twitter 利用 zookeeper 实现了一个全局 ID 生成的服务 Snowflake。</p> <p>格式: 63bits = 41bits Timestamp (毫秒级)+10bits: node ID (datacenter ID 5 bits + worker ID 5 bits)+12bits:sequence number。</p> <p>整个过程中只是在 Worker 启动的时候会对外部有依赖(需要从 Zookeeper 获取 Worker 号) 之后就可以独立工作了, 做到了去中心化。</p>	<p>1.去中心化</p> <p>2.可用性强, 速度快</p>	<p>1.依赖 zookeeper 独立的 snowflake 专用服务器</p>	2^{63}	由于每毫秒最大生成 2^{12} 个, 因此 2^{15} 每秒
Instagram	<p>instagram 参考了 flickr 的方案, 再结合 twitter 的经验, 利用 Postgre 数据库的特性, 实现了一个更简单可靠的 ID 生成服务。</p> <p>格式: 41bits + 13bits shard id + 10bits increment sequence</p>	<p>1.开发成本低</p>	<p>1.基于 postgresQL 的存储过程, 通用性差</p>	2^{63}	2^{13} 每秒
MongoDB ObjectId	<p>MongoDB 的 ObjectId, 采用 12 个字节的长度, 并且将时间戳进行编码。其生成方式如下:</p> <p> 0 1 2 3 4 5 6 7 8 9 10 11 </p> <p> 时间戳 机器 ID PID 计数器 </p> <p>前四个字节时间戳是从标准纪元开始的时间戳, 单位为秒。</p>	<p>1.秒级唯一性</p> <p>2.蕴含文档创建时间</p> <p>3._id 可在客户端生成, 有效降低服务器压力</p>	<p>1. 存储空间大</p>	2^{96}	2^{24} 每秒