

# 缓存雪崩和击穿

**缓存雪崩：** 缓存雪崩是指当大量缓存同时失效或过期后引起系统性能急剧下降的情况。

**发生场景：** 缓存失效或过期被清除后，系统需要再次访问数据库。由于处理这些请求的线程不知道其他线程正在生成缓存，所有请求都会去重新生成缓存，并访问数据库。

**表现：** 对数据库造成巨大的压力和不必要的性能损耗。拖慢整个系统，严重的会造成数据库宕机，形成一系列连锁反应，造成整个系统崩溃。

## 预防与补救：

**合理设置缓存失效时间：** 避免大量缓存同时失效，可以将缓存的失效时间分散开，比如加上一个随机的时间偏移或者使用带有范围的失效时间。

**使用多级缓存：** 引入多级缓存架构，如本地缓存、分布式缓存（如 Redis）、全局缓存（如 CDN），以降低缓存雪崩的风险。

**缓存预热：** 在系统启动或低峰期，提前加载热门数据到缓存中，避免在高峰期大量请求涌入时触发缓存失效。

**限流和熔断：** 实施限流和熔断机制，控制并发访问的数量，防止瞬时请求过多导致缓存雪崩。

**灾备机制：** 部署灾备机制，当主缓存出现问题时，可以切换到备用缓存，保障系统的可用性。

**分批更新缓存：** 当大量数据需要更新时，采用分批更新的方式，避免一次性大规模的缓存更新。

**缓存击穿：** 缓存击穿是指针对某个访问缓存非常频繁，但无法在缓存中处理（如缓存失效或不存在）的数据，其访问请求一下子都请求数据库，导致数据库压力倍增的情况。

**发生场景：** 某个热点数据在缓存中失效或不存在。大量针对该数据的请求直接访问数据库。

**表现：** 数据库承受巨大的访问压力，可能导致性能下降或宕机。

## 预防与补救：

**监控系统性能：** 实时监控缓存和数据库的性能指标，如命中率、响应时间等，及时发现并处理潜在问题。

**快速恢复机制：** 预先制定缓存雪崩和击穿的应急预案，如快速重启缓存服务、回滚到上一个稳定版本等。

**定期演练：** 定期进行缓存雪崩和击穿的模拟演练，提高团队应对此类问题的能力和效率。

## 在 3 台 linux 服务器上实现分布式缓存

首先,需要选择一个适合业务需求的分布式缓存技术。目前流行的分布式缓存技术包括 Redis、Memcached 等。这些技术都支持将数据分散存储在多个节点上,并提供了一致性哈希算法等机制来实现数据的分布式存储和访问。

### 安装和配置分布式缓存软件

步骤: 在每台 Linux 服务器上安装所选的分布式缓存软件(如 Redis)。

配置每台服务器的网络设置,确保它们可以相互通信。

根据分布式缓存技术的要求,配置相应的参数和策略,如缓存大小、过期时间、数据持久化等。

原因: 通过安装和配置分布式缓存软件,可以在多台服务器上搭建一个缓存集群,从而将数据分散存储在不同的节点上。

配置网络设置和参数策略可以确保缓存集群的稳定性和性能。

### 创建缓存集群

步骤: 根据所选的分布式缓存技术的要求,使用相应的命令或工具将三台 Linux 服务器加入到一个缓存集群中。配置集群的复制和分片策略,以确保数据的可靠性和性能。

原因: 通过创建缓存集群,可以实现数据的分布式存储和访问,从而提高缓存系统的性能和可扩展性。复制和分片策略可以确保数据的可靠性和容错性,当某个节点出现故障时,其他节点可以接管其数据,保持系统的正常运行。

### 测试和优化缓存集群

步骤: 使用测试工具或编写测试脚本对缓存集群进行性能测试,包括读写速度、并发处理能力等指标。

根据测试结果调整缓存集群的配置参数和优化策略,以提高其性能和稳定性。

原因: 测试是确保缓存集群能够满足业务需求的关键步骤。通过测试可以发现潜在的性能瓶颈和问题,并及时进行调整和优化。

优化缓存集群的配置参数和策略可以进一步提高其性能和稳定性,从而更好地支持业务的发展。

### 管理和维护缓存集群

步骤: 使用分布式缓存软件提供的监控和管理工具对缓存集群进行实时监控和管理。

定期对缓存集群进行备份和恢复测试,以确保数据的安全性和可靠性。

根据业务需求的变化,及时对缓存集群进行扩容或缩容操作。

原因: 实时监控和管理可以确保缓存集群的稳定性和性能,及时发现和处理潜在问题。

备份和恢复测试可以确保在数据丢失或节点故障等情况下能够迅速恢复业务。

根据业务需求的变化对缓存集群进行扩容或缩容操作可以确保资源的有效利用和业务的持续发展。