

高并发

高并发指的是系统单位时间内请求量非常大，比如系统的 QPS（Query Per Second，服务器每秒可以执行的查询次数）大于 10 万。

高并发系统设计的目标有三个：

- **高性能**：系统的处理请求的速度很快，响应时间很短。
- **高可用**：系统几乎可以一直正常提供服务。也就是说系统具备较高的无故障运行的能力。
- **高扩展**：流量高峰时能否在短时间内完成扩容，更平稳地承接峰值流量，比如双 11 活动、明星离婚、明星恋爱等热点事件。

实现高性能的常用手段：

- 优化技术选型（比如使用 `Disruptor` 替代 `ArrayBlockingQueue`，再比如本地缓存使用 `Caffeine` 替换 `Guava` 内置缓存模块）
- 数据库
 - SQL 优化
 - 分库分表&读写分离 <<https://javaguide.cn/high-performance/read-and-write-separation-and-library-subtable.html>>
 - NoSQL
- 缓存
- 消息队列 <<https://javaguide.cn/high-performance/message-queue/message-queue.html>>
- 负载均衡
- 池化技术
- 零拷贝
-

实现高可用的常用手段：

- 限流 <<https://javaguide.cn/high-availability/limit-request.html>>
- 降级&熔断
- 超时和重试机制 <<https://javaguide.cn/high-availability/timeout-and-retry.html>>
- 冗余设计 <<https://javaguide.cn/high-availability/redundancy.html>>
- 灰度发布&回滚

实现可扩展架构的常用手段：

- 分层架构：面向流程拆分
- SOA、微服务：面向服务拆分
- 微内核架构：面向功能拆分

