

高并发与多线程的关系、区别、高并发的技术方案

作者：有故事的人 来源：范文网 www.wtabcd.cn/fanwen/

本文原地址：<https://www.wtabcd.cn/fanwen/zuowen/453247f787a195f1589a5e20caf87a6b.html>

范文网，为你加油喝彩！

什么是高并发？

高并发（high concurrency）是一种系统运行过程中遇到的一种“短时间内遇到大量操作请求”的情况，主要发生在web系统集中大量访问收到大量请求（例如：12306的抢票情况；天猫双十一活动）。该情况的发生会导致系统在这段时间内执行大量操作，例如对资源的请求，数据库的操作等。

高并发的处理指标？

高并发相关常用的一些指标有：

1.响应时间（response time）

响应时间：系统对请求做出响应的的时间。例如系统处理一个http请求需要200ms我和春天有个约会歌曲，这个200ms就是系统的响应时间

2.吞吐量（throughput）

吞吐量：单位时间内处理的请求数香蕉面膜量。

3.每秒查询率qps（query per second）

qps：每秒响应请求数。在互联网领域，这个指标和吞吐量区分的没有这么明显。

4.并发用户数

并发用户数：同时承载正常使用系统功能的用户数量。例如一个即时通讯系统，同时在线量一定程度上代表了系统的并发用户数。

高并发和多线程的关系和区别

“高并发和多线程”总是被一起提起，给人感觉两者好像相等，实则 高并发 多线程

1.多线程

多线程是java的特性，因为现在cpu都是多核多线程的，可以同时执行几个任务，为了提高jvm的执行效率，java提供了这种多线程的机制，以增强数据处理效率。多线程对应的是cpu，高并发对应的是访问请求，可以用单线程处理所有访问请求，也可以用多线程同时处理访问请求。

在过去单cpu时代，单任务在一个时间点只能执行单一程序。之后发展到多任务阶段，计算机能在同一时间点并行执行多任务或多进程。虽然并不是真正意义上的“同一时间点”，而是多个任务或进程共享一个cpu，并交由操作系统来完成多任务间对cpu的运行切换，以使得每个任务都有机会获得一定的时间片运行。

再后来发展到多线程技术，使得在一个程序内部能拥有多个线程并行执行。一个线程的执行可以被认为是一个cpu在执行该程序。当一个程序运行在多线程下，就好像有多个cpu在同时执行该程序。

总之，多线程即可以这么理解：多线程是处理高并发的一种编程方法，即并发需要用多线程实现。

2.高并发

高并发不是java的专有的东西，是语言无关的广义的，为提供更好互联网服务而提出的概念。

典型的场景，例如：12306抢火车票，天猫双十一秒杀活动等。该情况的发生会导致系统在这段时间内执行大量操作，例如对资源的请求，数据库的操作等。如果高并发处理不好，不仅仅降低了用户的体验度（请求响应时间过长），同时可能导致系统宕机，严重的甚至导致oom异常，系统停止工作等。

如果要想系统能够适应高并发状态，则需要从各个方面进行系统优化，包括，硬件、网络、系统架构、开发语言的选取、数据结构的运用、算法优化、数据库优化等.....而多线程只是其中解决方法之一。

多线程并发技术

php多线程编程将会涉及到如下技术点：

1.并发编程三要素

原子性原子，即一个不可再被分割的颗粒。在java中原子性指的是一个或多个操作要么全部执行成功要么全部执行失败。有序性程序执行的顺序按照代码的先后顺序执行。（处理器可能会对指令进行重排序）可见性当多个线程访问同一个变量时，如果其中一个线程对其作了修改，其他线程能立即获取到最新的值。

2. 线程的五大状态

创建状态当用 new 操作符创建一个线程的时候就绪状态调用 start

方法，处于就绪状态的线程并不一定马上就会执行 run 方法，还需要等待cpu的调度运行状态cpu

开始调度线程，并开始执行 run

方法阻塞状态线程的执行过程中由于一些原因进入阻塞状态比如：调用 sleep

方法、尝试去得到一个锁等等死亡状态run 方法执行完 或者 执行过程中遇到了一个异常

3.悲观锁与乐观锁

悲观锁：每次操作都会加锁，会造成线程阻塞。乐观锁：每次操作不加锁而是假设没有冲突而去完成某项操作，如果因为冲突失败就重试，直到成功为止，不会造成线程阻塞。

4.线程之间的协作：wait/notify/notifyall等

5.synchronized 关键字

6.cas

cas全称是compare and swap，即比较替换，是实现并发应用到的一种技术。操作包含三个操作数——内存位置（v）、预期原值（a）和新值(b)。

如果内存位置的值与预期原值相匹配，那么处理器会自动将该位置值更新为新值。否则，处理器不做任何操作。

7.线程池

如果我们使用线程的时候就去创建一个线程，虽然简单，但是存在很大的问题。如果并发的线程数量很多，并且每个线程都是执行一个时间很短的任务就结束了，这样频繁创建线程就会大大降低系统的效率，美丽的秋天因为频繁创建线程和销毁线程需要时间。线程池通过复用可以大大减少线程频繁创建与销毁带来的性能上的损耗。

高并发技术方案：提高高并发能力

1.分布式缓存：redis、memcached等，结合cdn来解决图片文件等访问。

2.消息队列中间件：activemq等，解决大量消息的异步处理能力。

3.应用拆分:一个工程被人不在其位打一字拆分为多个工程部署，利用dubbo解决多工程之间的通信。

4.数据库垂直拆分和水平拆分(分库分表)等。

5.数据库读写分离，解决大数据的查询问题。

6.还可以利用nosql，例如mongodb配合mysql组合使用。

7.还需要建立大数据访问情况下的服务降级以及限流机制等。

以上内容希望帮助到大家，很多phper在进阶的时候总会遇到一些问题和瓶颈，业务代码写多了没有方向感，不知道该从哪里入手去提升，对此我整理了一些资料，包括但不限于：分布式架构

、高可扩展、高性能、高并发、服务器性能调优、tp6 , laravel , yii2 , redis , swoole、swift、kafka、mysql优化、shell脚本、docker、微服务、nginx等多个知识点高级进阶干货需要的可以免费分享给大家，需要的可以加入我的官方群[点击此处](#)。

更多 作文 请访问 https://www.wtabcd.cn/fanwen/list/92_0.html

文章生成doc功能，由[范文网](#)开发