1.使用内存数据库

内存数据库，其实就是将数据放在内存中直接操作的数据库。相对于磁盘，内存的数据读写速度要高出几个数量级，将数据保存在内存中相比从磁盘上访问能够极大地提高应用的性能。内存数据库抛弃了磁盘数据管理的传统方式，基于全部数据都在内存中重新设计了体系结构，并且在数据缓存、快速算法、并行操作方面也进行了相应的改进，所以数据处理速度比传统数据库的数据处理速度要快很多。

但是安全性的问题可以说是内存数据库最大的硬伤。因为内存本身有掉电丢失的天然缺陷，因此我们在使用内存数据库的时候，通常需要，提前对内存上的数据采取一些保护机制，比如备份，记录日志，热备或集群，与磁盘数据库同步等方式。对于一些重要性不高但是又想要快速响应用户请求的部分数据可以考虑内存数据库来存储，同时可以定期把数据固化到磁盘。

2.使用RDD

在大数据云计算相关领域的一些应用中，Spark可以用来加快数据处理速度。Spark的核心是RDD，RDD最早来源与Berkeley实验室的一篇论文《Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing》。现有的数据流系统对两种应用的处理并不高效：一是迭代式算法，这在图应用和机器学习领域很常见；二是交互式数据挖掘工具。这两种情况下，将数据保存在内存中能够极大地提高性能。

3.增加缓存

很多web应用是有大量的静态内容，这些静态内容主要都是一些小文件，并且会被频繁的读，采用Apache以及nginx作为web服务器。在web访问量不大的时候，这两个http服务器可以说是非常的迅速和高效，如果负载量很大的时候，我们可以采用在前端搭建cache服务器，将服务器中的静态资源文件缓存到操作系统内存中直接进行读操作，因为直接从内存读取数据的速度要远大于从硬盘读取。这个其实也是增加内存的成本来降低访问磁盘带来的时间消耗。

4.使用SSD

除了对内存方面的优化，还可以对磁盘这边进行优化。跟传统机械硬盘相比，固态硬盘具有快速读写、质量轻、能耗低以及体积小等特点。但是ssd的价格相比传统机械硬盘要贵，有条件的可以使用ssd来代替机械硬盘。

5.优化数据库

大部分的服务器请求最终都是要落到数据库中，随着数据量的增加，数据库的访问速度也会越来越慢。想要提升请求处理速度，必须要对原来的单表进行动刀了。目前主流的Linux服务器使用的数据库要属mysql了，如果我们使用mysql存储的数据单个表的记录达到千万级别的话，查询速度会很慢的。根据业务上合适的规则对数据库进行分区分表，可以有效提高数据库的访问速度，提升服务器的整体性能。另外对于业务上查询请求，在建表的时候可以根据相关需求设置索引等，以提高查询速度。

6.选择合适的IO模型

IO模型又分为：

(1).阻塞I/O模型：数据没到达之前，I/O一直阻塞，如果数据到达，则会返回。典型的是recvfrom，一般的默认都是阻塞的。

(2).非阻塞的I/O模型：和阻塞相反，只要不能得到结果的时候，I/O立刻返回。不会阻塞当前线程。

IO复用模型：也就是自己要学习的部分。多路复用的意思是，将多路信号合并到一路上进行处理，类似多个管道汇集到一个管道，与之相反的是多路分解。

IO复用模型主要是select，poll，epoll；对一个IO端口，两次调用，两次返回，比阻塞IO并没有什么优越性；关键是能实现同时对多个IO端口进行监听；函数也会使进程阻塞，但是和阻塞I/O所不同的的，这两个函数可以同时阻塞多个I/O操作。而且可以同时对多个读操作，多个写操作的I/O函数进行检测，直到有数据可读或可写时，才真正调用I/O操作函数。

信号驱动：首先开启套接口信号驱动I/O功能,并通过系统调用sigaction安装一个信号处理函数。当数据报准备好被读时，就为该进程生成一个SIGIO信号。随即可以在信号处理程序中调用recvfrom来读数据报，并通知主循环数据已准备好被处理中。也可以通知主循环，让它来读数据报。

异步的IO模型：告知内核启动某个操作，并让内核在整个操作完成后(包括将数据从内核拷贝到用户自己的缓冲区)通知我们。这里并不是说一定要用某个模型，epoll也并不是在所有情况下都比select性能要好的，在选择的时候还是要结合业务需求来。

7.使用多核处理策略

现在运行服务器的主流机器配置都是多核CPU的，我们在设计服务器的时候可以利用多核心的特点，采用多进程或者多线程的框架。关于选择多线程还是多进程可以根据实际的需求，结合各自的优缺点进行选择。对于多线程的使用，特别是使用线程池的时候可以通过测试不同线程池服务器的性能来设置合适的线程池。

8.分布式部署程序

当单机服务器已经找不到合适的优化点时，我们可以通过分布式部署来提高服务器的响应能力。优秀的服务器开发都会为自己的服务器的扩容，容灾提出一些解决方案。个人觉得服务器设计的时候简单点比较好，这样后期扩容的时候会很方便。