**系统高并发、高负荷下的应对**

首先声明，我们没有对软件进行任何功能性的限制，例如在线人数、并发请求、运算能力等，只限制了学员等注册数量。

当学员使用量较大，学习系统运行变慢甚至可能出现宕机风险时，可以从数据库、服务器配置等方面进行优化，以下给出一些指导意见与方法。

### 数据库

1. 采用企业版数据库，而不是Express版

如果在win7或win10等个人版操作系统中安装sqlserver，一般是默认的Express版，它是免费的，无须激活，但是它是单线程的。只适用于本地开发或测试，在实际应用中缺陷明显。

建议安装sqlserver enterprice（企业版），会要激活码。

可以在/help页面的“检测数据库”菜单项，查看数据库版本。

以下是其它客户的数据库版本信息，供参考

Microsoft SQL Server 2008 R2 (RTM) - 10.50.1600.1 (X64) Apr 2 2010 15:48:46 Copyright (c) Microsoft CorporationEnterprise Edition (64-bit) on Windows NT 6.2 <X64> (Build 9200: ) (Hypervisor)

1. 创建数据库索引

初始数据库脚本在生成数据库后，只有主键索引，能满足基本业务需求。更多索引需要执行额外的sql脚本。

https://gitee.com/weishakeji/LearningSystem/blob/master/DbScripts/other/%E5%88%9B%E5%BB%BA%E4%B8%80%E4%BA%9B%E5%B8%B8%E7%94%A8%E7%B4%A2%E5%BC%95.sql

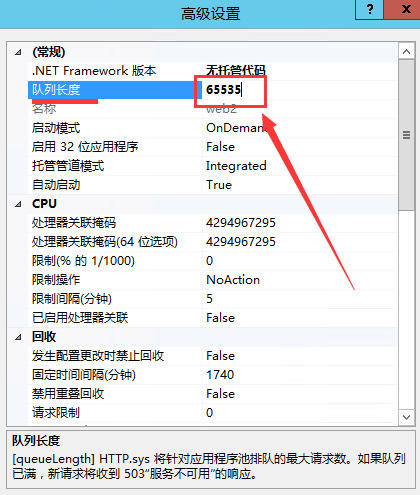
### **服务器**

1. 调整IIS应用程序池配置

找到部署的程序对应的应用程序池，右键选择“高级设置”。

1. 队列长度：65535（根据服务器配置可以加长）

队列长度值设置为预期最多用户数的 1.5 倍



1. 固定时间间隔（分钟）：1440分钟（24小时）
2. 闲置超时（分钟）：0分钟（不超时）
3. 调整IIS的appConcurrentRequestLimit设置

cmd命令行直接运行：

c:\windows\system32\inetsrv\appcmd.exe set config /section:serverRuntime /appConcurrentRequestLimit:100000

可以直接在

%systemroot%\System32\inetsrv\config\applicationHost.config 路径下配置文件查看到该设置是否被修改（默认为5000）：

<serverRuntime appConcurrentRequestLimit="100000" />

1. 调整machine.config中的processModel > requestQueueLimit的设置

cmd命令行直接运行：

notepad %systemroot%\Microsoft.NET\Framework\v4.0.30319\Config\machine.config

notepad %systemroot%\Microsoft.NET\Framework64\v4.0.30319\Config\machine.config

打开对应.net版本配置文件，编辑requestQueueLimit（默认为5000）：

<configuration>

<system.web>

<processModel enable="true" requestQueueLimit="100000"/>

参考链接：

https://learn.microsoft.com/zh-cn/previous-versions/office/communications-server/dd425294(v=office.13)

1. 修改注册表，调整IIS支持的同时TCP/IP连接数

cmd命令行直接运行：

reg add

HKLM\System\CurrentControlSet\Services\HTTP\Parameters /v MaxConnections /t REG\_DWORD /d 100000

运行命令重启使设置生效

net stop http & net start http & iisreset

1. 增加服务器配置

完成上述设置，一般情况就可以支持10万个并发请求，具体并发量还跟服务器配置有关。学习系统默认推荐的基础服务器配置是2核CPU，4Gb内存，可以在此基础上增加硬件配置。

### 分布式**部署**

最基本的方式可以将数据库与web服务分开部署。当需要部署多个web服务器时，可以通过Nginx反向代理实现。不过当前系统对分布式部署仍有不足之处，正在优化之中。