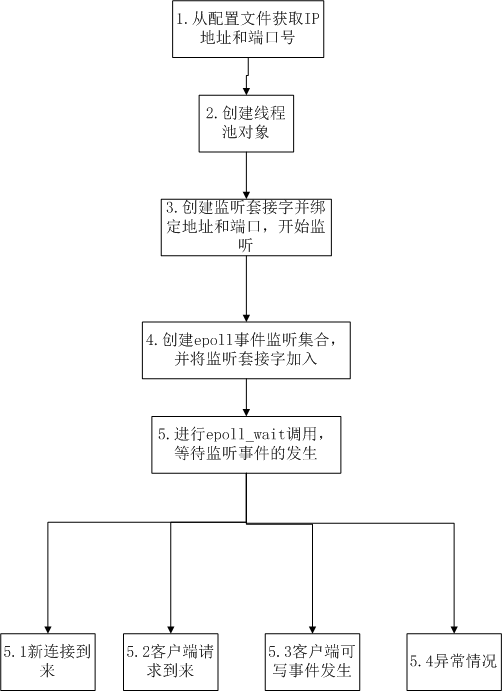
**服务器监听模块流程图**



**注解:**

1. 调用配置文件解析模块解析IP地址和端口号。
2. 创建线程池对象，线程池类采用的是单例模式，而且创建即运行。所以创建好线程池对象后，工作线程就开始等待任务了。
3. Socket、bind、listen一系列操作（服务器的标准流程）
4. 第一个添加进epoll事件监听集合的就是服务器监听套接字。

5. epoll\_wait侦听事件发生后的四种处理

**5.1新连接到来**

初始化新连接对应的http\_conn(http连接对象)，将套接字设置为非阻塞模式，并将其添加到epoll事件监听集合中，事件包括EPOLLIN | EPOLLRDHUP | EPOLLET（边沿触发模式）

**5.2客户端请求到来**

客户端的请求数据，即get或post请求，交给请求分析模块去读取客户端请求，并分析。如果是一个正确的请求，则将该客户端连接对象加入线程池任务队列

**5.3客户端可写**

设计上，是线程池那边解析好请求头之后，确认如果是POST请求，则将给客户端发送响应的任务交给CGI程序；如果是GET请求，则将请求的文件映射到内存上，然后由线程池线程发送响应头，然后设置描述符的EPOLLOUT事件，即将响应消息体的发送任务交给监听线程来处理。

当EPOLLOUT这个事件发生后，就去向客户端发送数据，也有可能消息体过大，数据没有一次性发完，造成TCP发送缓存满了，然后又得再次注册EPOLLOUT事件，此时EPOLLOUT被触发说明缓存有空闲空间了，然后继续去发送数据。等到所有数据都发完了，则又重新开始监听EPOLLIN，即检测客户端请求。

**5.4异常情况**

检测的异常情况包括EPOLLRDHUP | EPOLLHUP | EPOLLERR，所做的处理很简单，关闭出错、或断开的套接字描述符。