Zy：各位评委老师大家好，我们是来自电子科技大学的imake小组，我们今天展示的项目是分布式爬虫系统，首先我们看到的是分布式爬虫的初始页面（记得开redis数据库），在这个页面我们将输入要爬取的网页url，我们目前做到能够爬取京东，淘宝，天猫。网易严选这几个电商网站，首先我们随便输入一个错误值作为异常测试（记得用回车），可以看到页面提示输入错误。接下来我们打开后端端口监听浏览器请求然后输入http://www.jd.com，接下来的演示都是基于京东网页的分布式爬取，输入网址之后点击确认按钮，前端页面通过websocket协议将目标网站url传输给后端进行爬取，现在数据展示页面还没有数据，因为我们的后端服务还未开启，接下来我们查看一下后端服务器。

Wq：现在我们打开后端界面，现在可以看到两个界面，一个是主机界面（左边还是右边），一个是从机界面（左边还是右边），主机在刚开启的阶段，主要负责三个任务，一个是负责采集主页面初始URL，并且将他们分发到从机，二是将代理网站的ip和port发送到从机，在从机进行可用性检测，与此同时开启监听端口，负责接收ip的请求，负责接收从 ，机发来的URL的去重，从机自己在采集URL后也会进行初次去重过滤，目的是为了减轻主机去重的负担。此时弹出的是从机操控的模拟浏览器，看到是京东界面。接下来看到启动的一批浏览器一共分成三类，一类是商品页面信息的提取，一类是商品列表页面的提取，一类是除此两类以外的其他页面的提取。我们对于商品页面的区分是依靠商品URL中的特征值，比如：item，商品页面中加入购物车字样来确定，列表页有list在url中。我们的提取信息的策略是使用正则表达式引擎。

在主机和从机的通讯上，发送采用基于tcp的socket连接，接收采用非阻塞的socketserver进行接收。

浏览器和后端python程序的通讯采用web socket协议，采用websocket有什么好处？首先他的通讯方式，第一次握手通过http，之后通过tcp连接传输数据，这么做以后，可以减少http报文报头的使用，这样就减少了程序的开销，同时Websocket是一个持久化的协议，相对于HTTP这种非持久的协议来说，websocket的主要是为了解决在web上应用长连接进行灵活的通讯应用而产生，但websocket本身只是一个基础协议，对于消息上还不算灵活，毕竟websocket只提供文本和二进制流这种基础数据格式．在实际应用则更偏向于对象消息的处理，而在这基础上更希望集成一系列的消息路由机制来解决消息处理上的问题．为了解决以上问题beetle针对websocket进行了一些高层次的封装，让服务端处理消息变得更简单灵活。

我们的分布式思想主要体现在，URL分批在从机爬取，也就是爬取任务分散在各个节点，爬取的数据存储在从机的数据库中，减少一个机器实现功能的压力。

Zy：现在看一下前端的数据展示页面，各个分节点的cpu利用率和内存占用的数据用动态柱状图显示，横坐标是节点编号，纵坐标是cpu利用率和内存占用，红色是内存占用，蓝色是cpu占用率，将鼠标移动到图上可以看到每个分节点的各个数据。

整个系统的I/O吞吐量通过动态折线图实时的采集展示，鼠标移动上去也可以看到具体的数据。

在下方是爬取的商品信息等内容，实时的滚动展示。整个页面整体简洁明了。

我们的项目采用瀑布模型，因为瀑布模型适用小型系统的开发，同时项目没有用户反馈，并且需求明确，所以采用瀑布模型合适，在开发中严格产生开发文档，以文档作为每个阶段的结束标志。