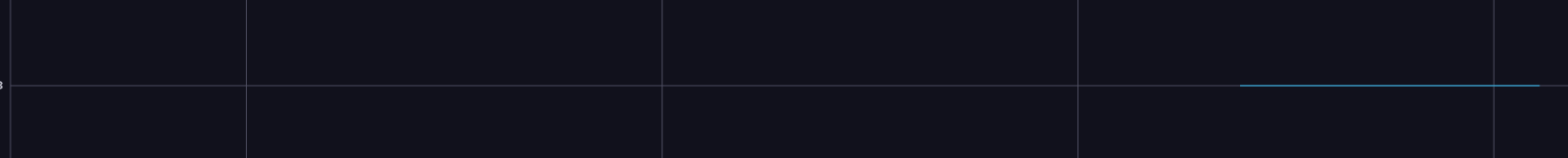
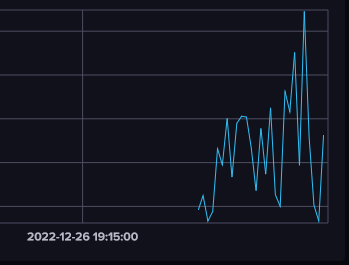
应用系统体系架构作业12

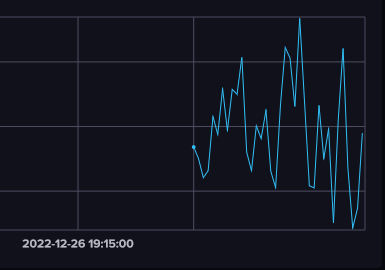
一．InfluxDB和Telgraf

1. 

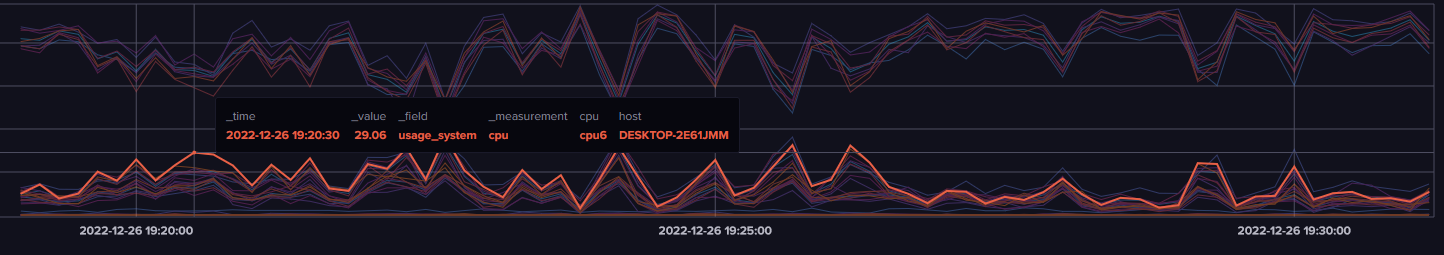
上图表示我的笔记本上面有8个CPU。

1. 

上图表示CPU的系统占用情况usage\_system。

1. 

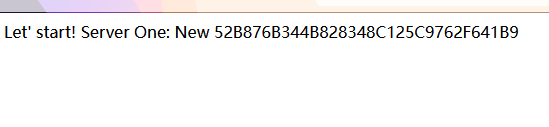
上图表示CPU的用户使用情况usage\_user。

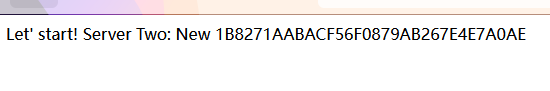
1. 

上图是对于CPU状态的完整监控的截图。从上图可以看出，在记录数据的15分钟内，各个不同的CPU运行相对平稳并且运行状态几乎相同。电脑运行正常。

二．Nginx（运行的程序即为老师提供的实例程序，nginx监听8000端口，两个server分别运行在8080和8090端口。）

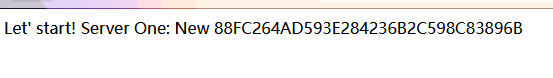
1. 配置1：使用默认的round-rabin负载均衡策略。不设置权重。





效果解释：由于使用默认的round-rabin的负载均衡策略，并且未设置权重的时候权重默认为1，所以连续运行两次的时候，就会有一次分配到server1，另外一次分配到server2.

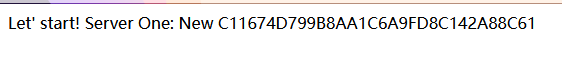
1. 配置2：使用ip\_hash的负载均衡策略，不设置权重。

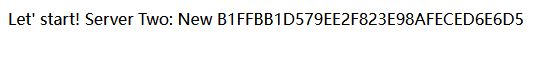




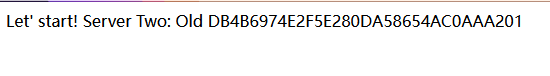
效果解释：由于使用的是ip\_hash的负载均衡策略，所以在第一次发送请求的时候会根据ip地址的hash值来确定请求会被那一个server处理。并且由于后续发送请求的IP都没有变，所以后续的请求还是在同一个server上被处理，并且session不会被删除。所以第一次会创建一个新的session，后续都是同一个server，并且使用旧的session。

1. 配置3：使用least\_conn的负载均衡策略，不额外配置权重。



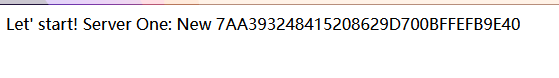


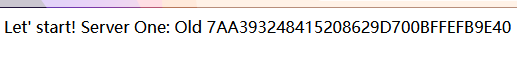


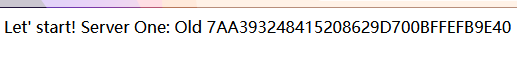


效果解释：由于使用的是least\_conn的负载均衡策略，所以这时候的运行结果就不再像前面两种那样有规律性。因为一个请求会被那个server处理取决于每个server当前的负载情况，会优先选择负载更小的server进行操作。所以效果如上图所示，一会会在server1上处理，一会会在server2上面处理。

1. 配置4：使用默认的round-rabin的负载均衡策略，并额外的配置权重，将server1的权重设置为3，server2不设置，仍然使用默认数值1.









效果解释：权重表示的是一个server在接受请求的时候能够分到的流量的比例。例如上述设置server1的权重为3，server2的权重为1，那么每四个请求中就会有三个分配到server1，有一个分配到server2。