

UCML 业务集群架构配置手册

概述

UCML 业务系统部署在 Web 服务器上，业务逻辑可以在 Web 服务器本地实现，也可以调用远端 ESB 服务器上注册的服务实现，远端业务逻辑可以分散 Web 的服务器的压力，但是随着客户端用户量的不断增加，对 Web 服务器的流量压力和 CPU 负载压力都同时增大，单台服务器无法承担再大的用户并发访问，必须采用多台服务器协同工作，来提高计算机的处理能力和计算速度，满足当前业务量的需求。

如何在多台业务服务器之间是合理的业务量分配，使之不会出现一台服务齐过忙，而其他设备没有充分使用的情况。为了解决这一问题，UCML 采用软件 7 层负载平衡方案，软件 7 层负载平衡方案基于 HTTP 方向代理方式，代表产品有 Nginx、L7SW、HAProxy 等。

Nginx 是一款非常优秀的软件负载平衡器，其最大的优点就是能承载高并发连接，目前国内知名站点：其中有新浪、网易、腾讯等门户网站都采用了 Nginx。

UCML 采用 Nginx 来做一台反向代理服务器，做为虚拟的 web 站点出口，后面的多台 Web 服务器是 IIS，运行在 Nginx 之后的内网内(见图 1)。Nginx 代理服务器服务器环境可以是 Linux 系统，也可以是 Windows 系统，服务器本身硬件性能越好，负载能力越大，相同服务配置前提之下，Linux 的下 Nginx 负载能力要比 Windows 的负载能力强。

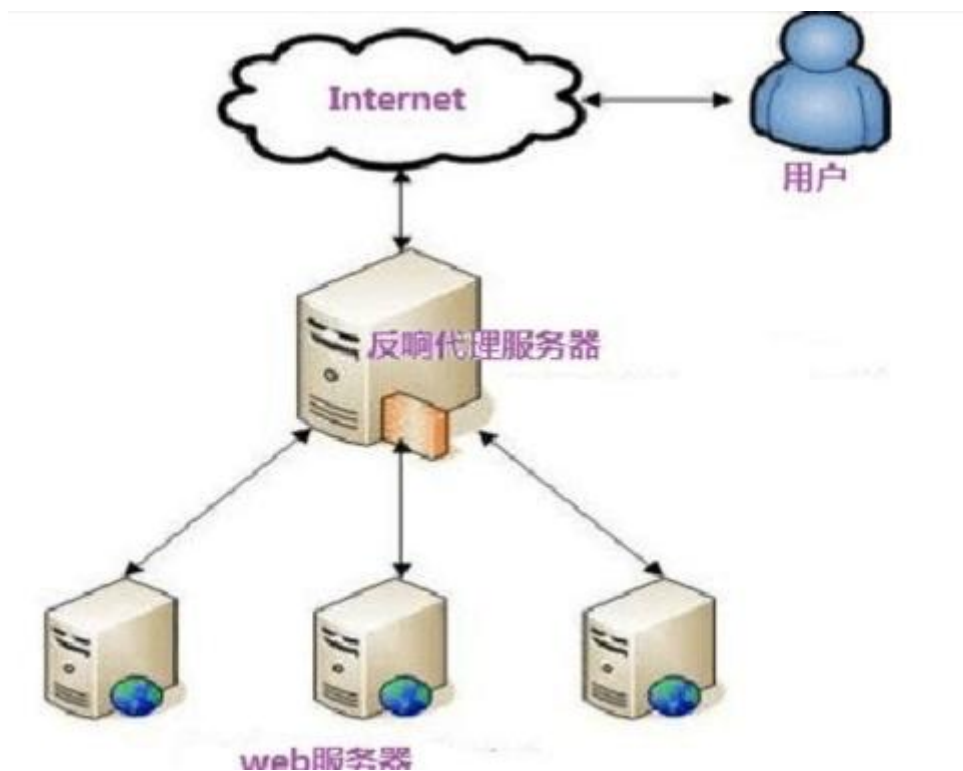


图 1

在 Nginx 之后配置多台 Web 服务器之后，带来一个新的问题，就是多台 Web 服务器会话 (Session) 一致性的问题和会话服务并发压力问题，UCML-Memcached 替换了 Asp.Net 原有的会话方案，高效器支持会话集群服务。下面具体介绍业务集群和会话集群配置步骤。

同时 UCML BPM 流程引擎服务器也支持集群方案，可以支撑不同流程实例在不同的服务器之内。

UCML 反向代理服务的配置说明

准备服务器环境：以 2 台服务器为例

第一台：

系统：Win2003

nginx：nginx-1.4.5

IP：192.168.0.51

环境：本地

第二台：

系统：Win2003

IP：192.168.0.52

环境：远程

说明：

软件 nginx 放在本地(192.168.0.51)，也就是说放在域名绑定的那台服务器，这台服务器的 IIS 不能使用 80 端口，因为等一下 nginx 软件要使用 80 这个端口。

为了方便，将本机的 hosts 文件添加了域名 192.168.0.51 www.g.cn)

把 UCML 安装盘自带的 nginx 程序包拷贝到 C:\，把目录名改成 nginx

一切准备就绪，开始下面步骤：

● 步骤一

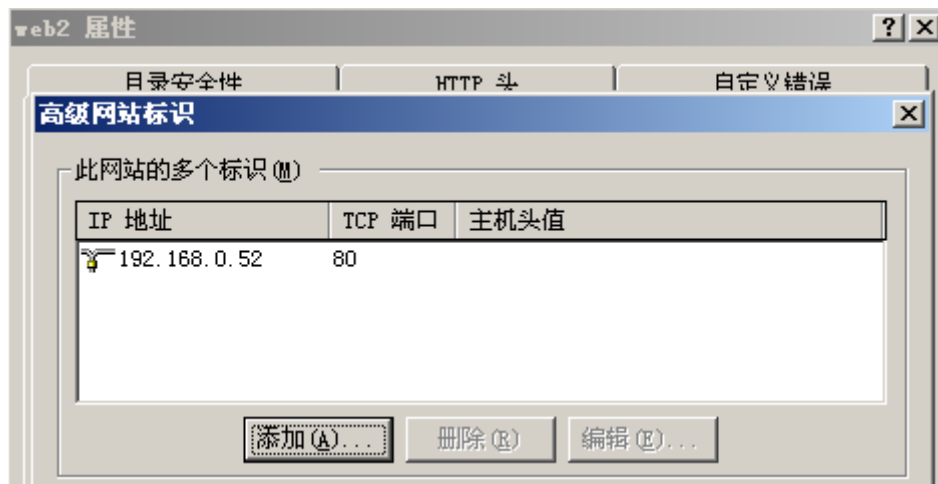
在本地(192.168.0.51)这台服务器 IIS 创建一个网站，使用端口为 808，如下图：



IIS 网站绑定设置图

● 步骤二

在远程 192.168.0.52 的 IIS 创建一个网站，使用端口为 80，如下图：



● 步骤三

以上已经设置好两台服务器的 IIS 了,下面配置 nginx 软件来实现网站负载均衡,打开如下文件:

C:\nginx\conf\nginx.conf

1、找到内容 server {

在它的上面加入如下内容:

```
upstream    www.g.cn {  
    server    192.168.0.51:808;  
    server    192.168.0.52:80;  
}
```

(这是负载均衡使用的服务器网站 IP)

```
2、找到 location / {  
    root      html;  
    index     index.html index.htm;  
}
```

把内容更改如下:

```
location / {  
    proxy_pass http://www.g.cn;  
    proxy_redirect default;  
}
```

3、找到 server {

```
listen 80;  
server_name localhost;
```

把内容改成如下：

```
server {  
    listen 80;  
    server_name 192.168.0.51;
```

(这是监听访问域名绑定那台服务器 80 端口的请求)

简单配置好了，下面看下以上 3 步配置的图：

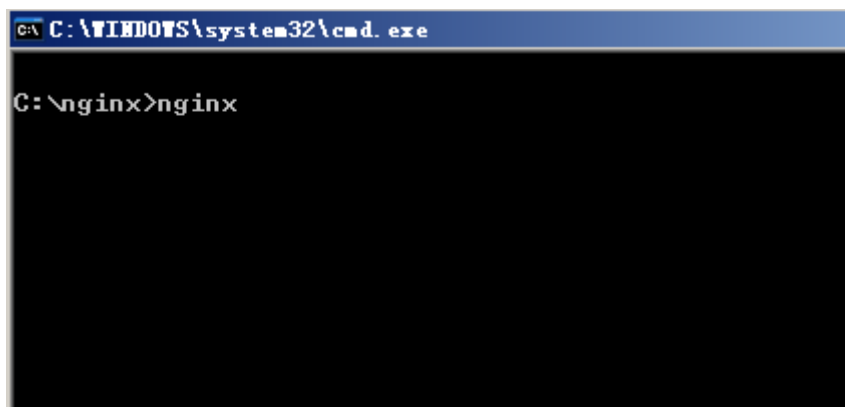


负载配置图

● 步骤四

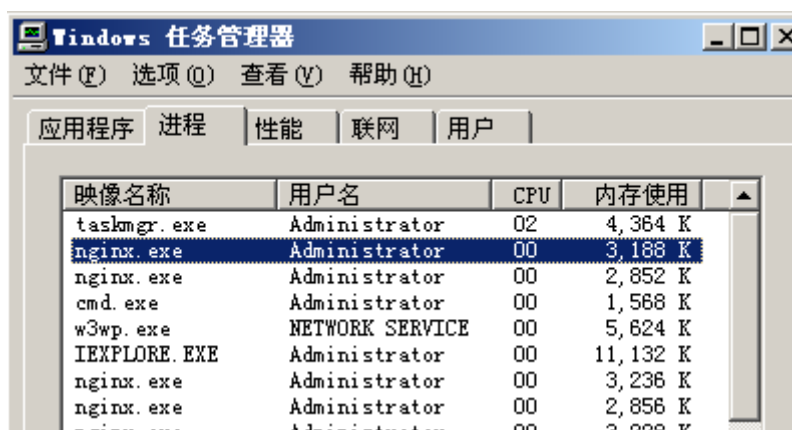
配置好了之后，启动 nginx 软件

进入命令提示符 CMD，进入 c:\nginx>，输入 nginx 命令，如下图：



启动 nginx

这时候，系统进程有两个 nginx.exe 进程，如下图：



系统 nginx 进程

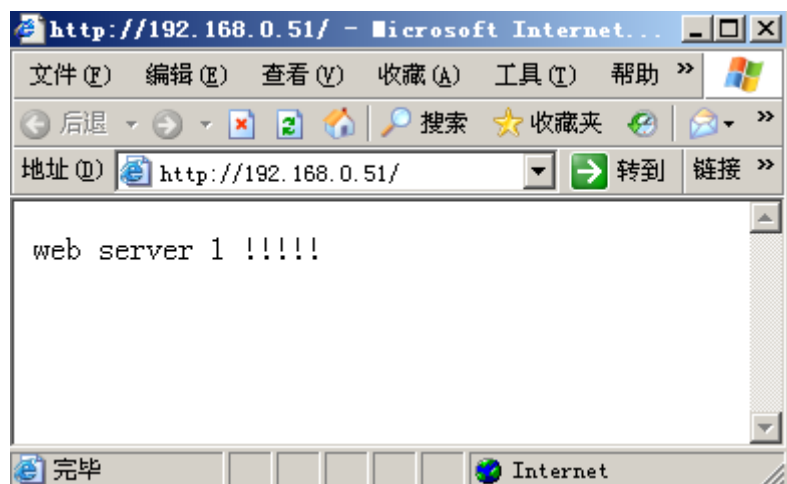
（停止 nginx 运行输入 `nginx -s stop` 即可）

● 步骤五

经过以上的配置，现在我们看下负载效果：

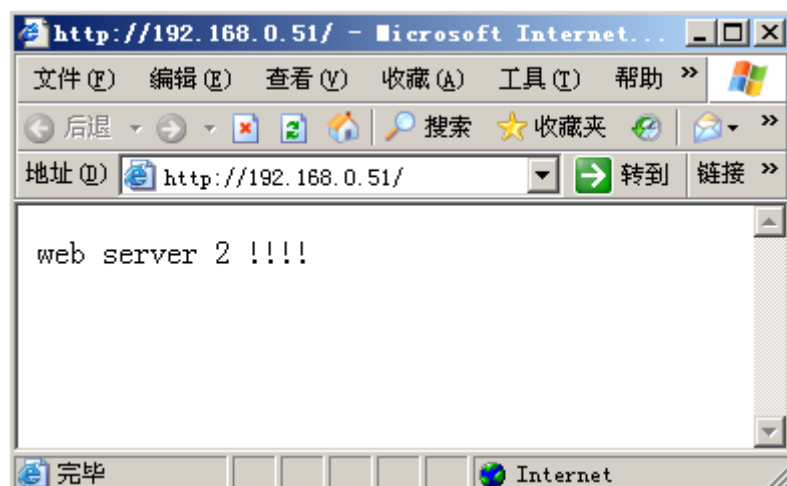
在本地(192.168.0.51)这服务器打开 IE，输入：`http://192.168.0.51`

第一次打开网站的结果：



第一次访问网站图

刷新一下网页，出现的结果：



再次访问网站图

补充一下，nginx 软件在 linux 上运行性能比在 windows 上要好，所以做负载可以用 linux 跑 nginx，.net 开发的网站放到 windows 服务器 IIS 上。如果网站访问量非常大，可以专门用一台服务器运行 nginx，其它服务器运行 Web IIS 服务器程序

nginx 可以根据客户端 IP 进行负载均衡，在 upstream 里设置 ip_hash，就可以针对同一个 C 类地址段中的客户端选择同一个后端服务器，除非那个后端服务器宕了才会换一个。

nginx 的 upstream 目前支持的5种方式的分配

1、轮询（默认）

每个请求按时间顺序逐一分配到不同的后端服务器，如果后端服务器 **down** 掉，能自动剔除。

```
upstream backserver {  
  
server 192.168.0.14;  
  
server 192.168.0.15;  
  
}
```

2、weight

指定轮询几率，**weight** 和访问比率成正比，用于后端服务器性能不均的情况。

```
upstream backserver {  
  
server 192.168.0.14 weight=10;  
  
server 192.168.0.15 weight=10;  
  
}
```

3、ip_hash

每个请求按访问 **ip** 的 **hash** 结果分配，这样每个访客固定访问一个后端服务器，可以解决 **session** 的问题。

```
upstream backserver {
```



```
ip_hash;
```

```
server 192.168.0.14:88;
```

```
server 192.168.0.15:80;
```

```
}
```

4、fair（第三方）

按后端服务器的响应时间来分配请求，响应时间短的优先分配。

```
upstream backserver {
```

```
server server1;
```

```
server server2;
```

```
fair;
```

```
}
```

5、url_hash（第三方）

按访问 url 的 hash 结果来分配请求，使每个 url 定向到同一个后端服务器，后端服务器为缓存时比较有效。

```
upstream backserver {
```

```
server squid1:3128;
```

```
server squid2:3128;
```

```
hash $request_uri;
```

```
hash_method crc32;

}
```

以上5种模式的负载均衡，除了 ip_hash 和 url_hash 模式外，都需要后端多台 web 服务器之间实现 session 共享，否则用户登录后访问页面会有问题。先介绍一下 ip_hash 模式的工作原理

nginx 可以根据客户端 IP 进行负载均衡，在 upstream 里设置 ip_hash，就可以针对同一个 C 类地址段中的客户端选择同一个后端服务器，除非那个后端服务器宕了才会换一个。

```
upstream www.u.com {

    ip_hash;

    server 202.96.1.184:9018;

    server 202.96.1.182:9018;

}
```

也就是说在两台服务器上的 Web 程序，还是可以用传统的 Asp.Net 会话 session 配置，前面启用 ip_hash，正常情况下，客户端上网获得 IP，客户的网页请求被转到固定的后端服务器上，这样就不会出先 Session 问题了。应该说来访 IP 分布越广，负载均衡就越平均。但是，如果都是同一个 IP 出来的用户，负载均衡效果就没有了。

其他三种模式，需要会话共享的问题，UCML 平台自身提供了会话替代方案，利用 UCML-Memcached 缓存集群方案来解决会话共享和会话服务访问压力的问题。

UCML 缓存集群服务器配置说明

UCML-Memcached 介绍

Memcached 是一个高性能的分布式内存对象缓存系统，用于动态 Web 应用以减轻数据库负载。它通过在内存中缓存数据和对象来减少读取数据库的次数，从而提供动态、数据库驱动网站的速度。Memcached 基于一个存储键/值对的 hashmap。其守护进程（daemon）是用 C 写的，但是客户端可以用任何语言来编写，并通过 memcached 协议与守护进程通信。但是它并不提供冗余（例如，复制其 hashmap 条目）；当某个服务器 S 停止运行或崩溃了，所有存放在 S 上的键/值对都将丢失。

UCML-Memched 配置过程

安装 Memcache Server(也可以不安装直接启动)

1. 拷贝 UCML 自带的 memcached 文件到 c:\memcached
2. 在 CMD 下输入 "c:\memcached\memcached.exe -d install" 安装.
3. 再输入: "c:\memcached\memcached.exe -d start" 启动。NOTE: 以后 memcached 将作为 windows 的一个服务每次开机时自动启动。这样服务器端已经安装完毕了。

运行可以加上参数来加以设置。

常用设置:

- | | |
|------------------|-------------------------------------|
| -p <num> | 监听的端口 |
| -l <ip_addr> | 连接的 IP 地址, 默认是本机 |
| -d start | 启动 memcached 服务 |
| -d restart | 重起 memcached 服务 |
| -d stop shutdown | 关闭正在运行的 memcached 服务 |
| -d install | 安装 memcached 服务 |
| -d uninstall | 卸载 memcached 服务 |
| -u <username> | 以<username>的身份运行 (仅在以 root 运行的时候有效) |

| | |
|-------------|-------------------------------|
| -m <num> | 最大内存使用，单位 MB。默认 64MB |
| -M | 内存耗尽时返回错误，而不是删除项 |
| -c <num> | 最大同时连接数，默认是 1024 |
| -f <factor> | 块大小增长因子，默认是 1.25 |
| -n <bytes> | 最小分配空间，key+value+flags 默认是 48 |
| -h | 显示帮助 |

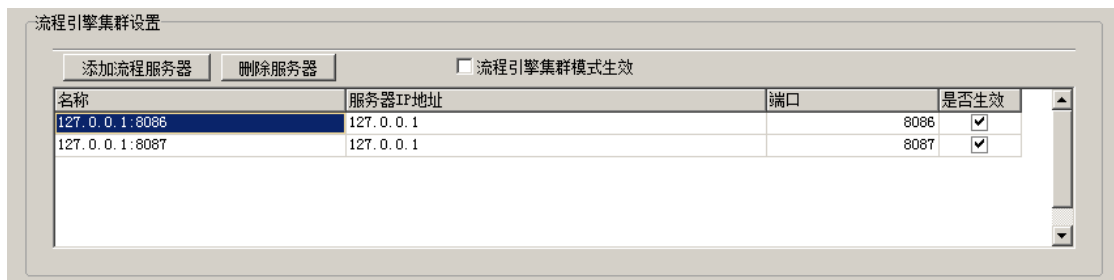
接下在 UCML 应用的 web.config 配置如下节：

```
<enyim.com>
  <memcached protocol="Binary">
    <servers>
      <!-- make sure you use the same ordering of nodes in every configuration
you have
      <add address="192.168.1.104" port="11211" />
      <add address="192.168.1.106" port="11211" />-->
      <add address="127.0.0.1" port="11211"/>
    </servers>
    <socketPool          minPoolSize="10"          maxPoolSize="100"
connectionTimeout="00:10:00" deadTimeout="00:00:10"/>
  </memcached>
</enyim.com>
```

其中<add addres/>列表代表缓存集群的服务里列表
可以在 UCML 平台里自动配置

UCML 流程引擎集群配置说明

1. 在 UCML 平台之内配置流程引擎服务器地址：



2. UCML 配置文件生成生效
3. 用如下代理函数访问流程接口

```
public WorkFlow.WorkFlowSession FlowSessionEx(Guid InstanceID)
```