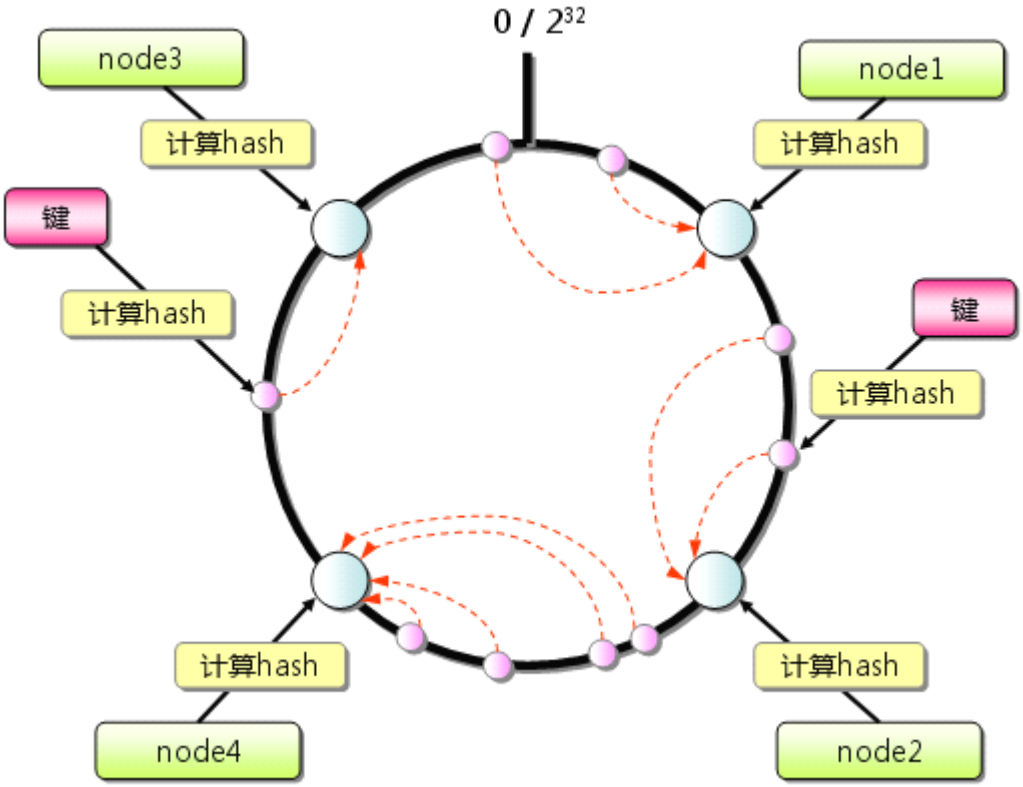
nginx_upstream_hash 增加一致性 hash

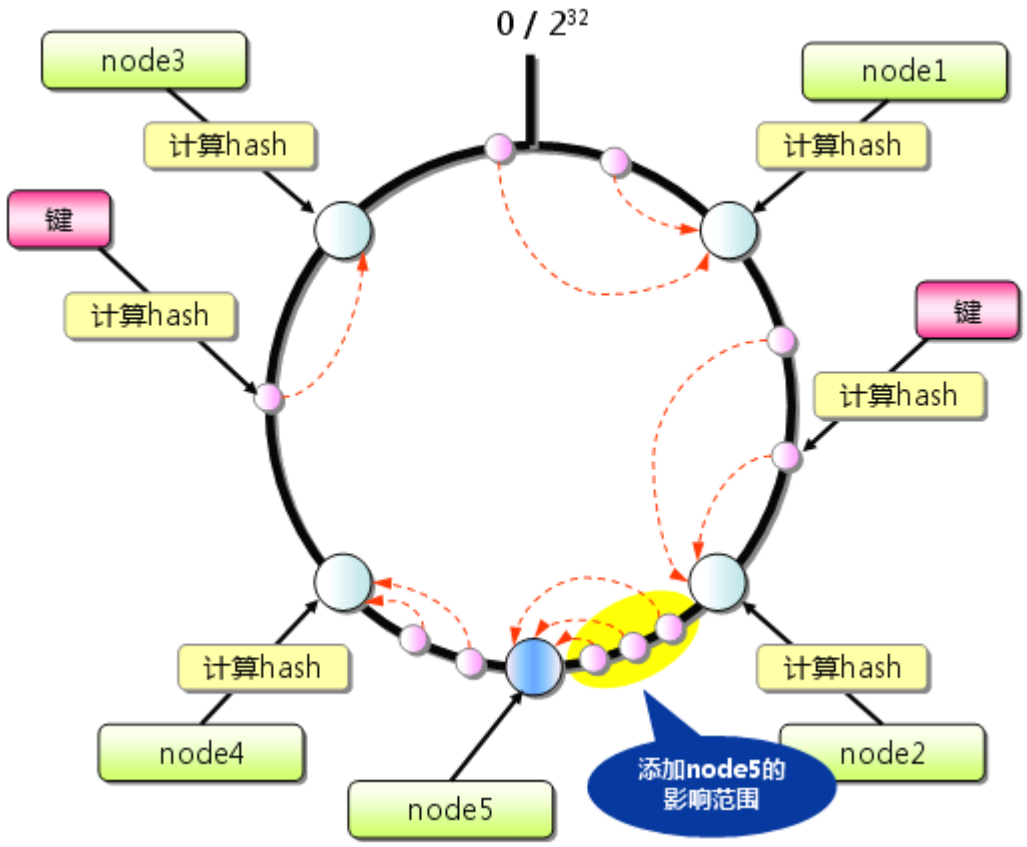
版本记录			
时间	版本	说明	修改人
2009.06.09	1.0	初稿	吴威

一、Consistent Hashing 的简单说明

Consistent Hashing 如下所示：首先求出所有备选服务器（节点）的哈希值，并将其配置到0～2^{32}的圆（continuum）上。然后用同样的方法求出存储数据的键的哈希值，并映射到圆上。然后从数据映射到的位置开始顺时针查找，将数据保存到找到的第一个服务器上。如果超过2^{32}仍然找不到服务器，就会保存到第一台服务器上。



从上图的状态中添加一台服务器的时候，余数分布式算法由于保存键的服务器会发生巨大变化而影响缓存的命中率，但 Consistent Hashing 中，只有在 continuum 上增加服务器的地点逆时针方向的第一台服务器上的键会受到影响。



因此，Consistent Hashing 最大限度地抑制了键的重新分布。而且，有的 Consistent Hashing 的实现方法还采用了虚拟节点的思想。使用一般的 hash 函数的话，服务器的映射地点的分布非常不均匀。因此，使用虚拟节点的思想，为每个物理节点（服务器）在 continuum 上分配100～200个点。这样就能抑制分布不均匀，最大限度地减小服务器增减时的缓存重新分布。

二、nginx_upstream_hash 介绍

Nginx_upstream_hash 是 nginx 的一个第三方模块，支持采用 nginx 内部的各种变量作 hash，然后针对生成的 hash 值，用求余的方式分布到后端（backend）服务器上，达到负载均衡的目的。就是说每个后端服务器只保存一份 cache，不会造成 cache 空间的浪费。

三、为什么需要在 nginx_upstream_hash 上增加一致性 hash 功能？

因为 nginx_upstream_hash 内部的算法采用 hash 求余的方式选择后端的服务器，当你要增加服务器的时候，整个服务器群的 cache 都会受到影响，产生瞬间的后端负载，对业务造成影响。通过增加一致性 hash 功能，只影响部分后端服务器。保证了业务的平稳运行。对系统的扩展带来了便利。

四、具体的操作方法

1、补丁下载地址:

https://bbs.be10.com/code/upstream_hash/nginx.path

https://bbs.be10.com/code/upstream_hash/upstream_hash.path

```
#为 nginx 打补丁
cd nginx-0.7.17
patch -p1 < ../nginx.path


#为 ngixn_upstream_hash 打补丁
cd nginx_upstream_hash-0.3
patch  -p1 < ../upstream_hash.path
```

2、为系统增加 ketama 的 md5 库
下载 ketama-0.1.1.tar.bz2

```
cd ketama/libketama
gcc -fPIC -O3 -c md5.c
gcc -shared -o libmd5.so md5.o
cp libmd5.so /usr/local/lib
cp md5.h /usr/local/include
```

编辑 /etc/ld.so.conf

添加一行 /usr/local/lib

运行 ldconfig

为系统增加用户自定义的动态库路径,要不 nginx 运行的时候可能报错

五、测试

1、apache 环境的建立

建立几个不同端口的虚拟主机，添加下面的配置

```
<IfModule mod_rewrite.c>
   RewriteEngine On
   RewriteRule ^(.*)$ /url_direct.php?$1
 </ifmodule>
```

2、Nginx 的配置

```
        location / {
          proxy_pass http://backend;
        }


        upstream backend {
                server 192.168.110.100:8088;
                server 192.168.1.98:8087;
                server 192.168.111.98:8086;
                #server 127.0.0.1:8085;
                hash    $request_uri;
        }
```

3、 url_direct.php 内容

```
<?php
echo "$_SERVER[QUERY_STRING]";
echo ":$_SERVER[SERVER_PORT]";
?>
```

4. Curl 测试脚本

```
#添加服务器前测试一次
for i in `cat uri.txt`;do curl -s localhost:8089/$i;echo "";done > a.txt

#添加服务器后再测试一次
for i in `cat uri.txt`;do curl -s localhost:8089/$i;echo "";done > b.txt
```

```
#对比两次的结果
diff a.txt b.txt
```

5、uri.txt 内容 （uri.txt 是用 keepass 的密码生成功能手动生成的 50 个 30 长的字串）

```
csy89j91tl64iaaf0xgcqallbjk4fk
of20vsq9z7riy3kjba51tvch8vxg8x
hq3qb1b7zfhz5snhlggioy1ygqxo04
om808tm0f7p5a2jk5p9cf7d80ghp1o
gcb5bpw6fga9isu6ca9q1njmad42lo
bn88btizergm83m79929h4yyqdakq
bezc0nllb8fxedgjnm7wlm297zh5f5
91xf6i0vks35u9581yiz8tv1apyljq
v1b98rg3jj6u3vw8sodv7p5ydogcww
tss8pwzcy1tvdx567obuegth0lrbgs
9k3wg7ie2zhzidgdqifhd7doye418a
b3l75hirt2zp1jftz5h8zxtwz9i8nj
0pgete6y6cye0g2z659vy9savrxved
n85bmj7lfps1brld3u7oha0zrkv323
ln5e74r0yn5x2ccb6h6vgx9e8eyb5s
fzb686ip79qj8r2jwupuflhx07l5vd
hfzd4zabv81rytd2mcze20mlgexq30
5zlvhl7jwtrnwsww2z9dikb1c9fhgs
rwduji0g4oiqzitibrc4u7fhzx8o5h
gychjboafreb7v3fghwun2fswopeic
gwkkx97kwnhsavjrzb3anxp74nxd2w
e6j0acz5jw7ukhtuwhsww5jbjuysk0
m3o32vpeop3uz7pnqwpbn2kfhvfb5v
19tg4albat3b78bi3okziakj7hc77v
5c5vrmi2gnvj3cqaf7lshhxsvbf6z2
zwsvbre0zvzjcznsryw0k1bdw3nfhw
ar6195ml5lirfdf2fbpg94jpubbjrn
ectbkshrgdce7vkpyc02fsjisdo9hz
ms0u99wxa1aysgp0lj2f5isw8jkn20
o9l0jzzjucqv9xqspdeqet8jecf2ab
eqra4wmral6swxgatt24fshmfvuh2z
```

```
1k7m8hxfk140mpsxtkzbdb8ojs59hc
hisx2sug7ilk417xp8ljya41mjz42v
mcd4zstvl8k74e0v0fx2j43vwpwpk1
fyu0fvq7y6b9zz41dwkj6udxqdrgit
i0fo6gf3k93ve9lkio1ayob68ivv6d
yl902qtf9s51hpuas6gmreht51wn1h
122hvzb5hplbz2zyznux6lkbme6vtg
om0sdgzruea56n2peyrc8xeywb4o3a
7u6fekl57hc6eu4rx0ewcj9bp7grkt
jvgok3yo2guv4kh1ei8opk7ma4cwsw
x9qxvlpgbfh4rzby4ibu77nd1hfkvb
groggwofeo0g0mx7kstbuxm7c7k6ya
3ttsdb87xuqqpivcygxcrxcsgkr184
cvcf1hd5hnt2chlnb72xavc0kumyuo
yu44jsb7tay5lh8b8npv4ebtuu5lrv
ggtlcxu6xocechqzywl51o36f019i0
8flm74ct3p5mvvvrzc63ejcejw1ebr
nqxs76z1k2q214uffcrea5sb0tabc6
baankomwgqijmdr3kv5dnqkoxdrufk
```

6、测试结果

经过测试发现增加一个服务器，50 个 uri 中会改动 13 个 uri 的映射服务器关系。实现了一致性 hash 的功能。即只影响部分服务器上的业务。

六、参考资料

1. http://tech.idv2.com/2008/07/24/memcached-004/
2. http://www.evanmiller.org/nginx-modules-guide.html
3. http://code.google.com/p/memagent/wiki/HowMagentWorks