

# 网站集群架构

利用开源软件构建 高可用、高性能、  
可扩展 的集群系统



兰锋

[bluedata@gmail.com](mailto:bluedata@gmail.com)



## 摘要

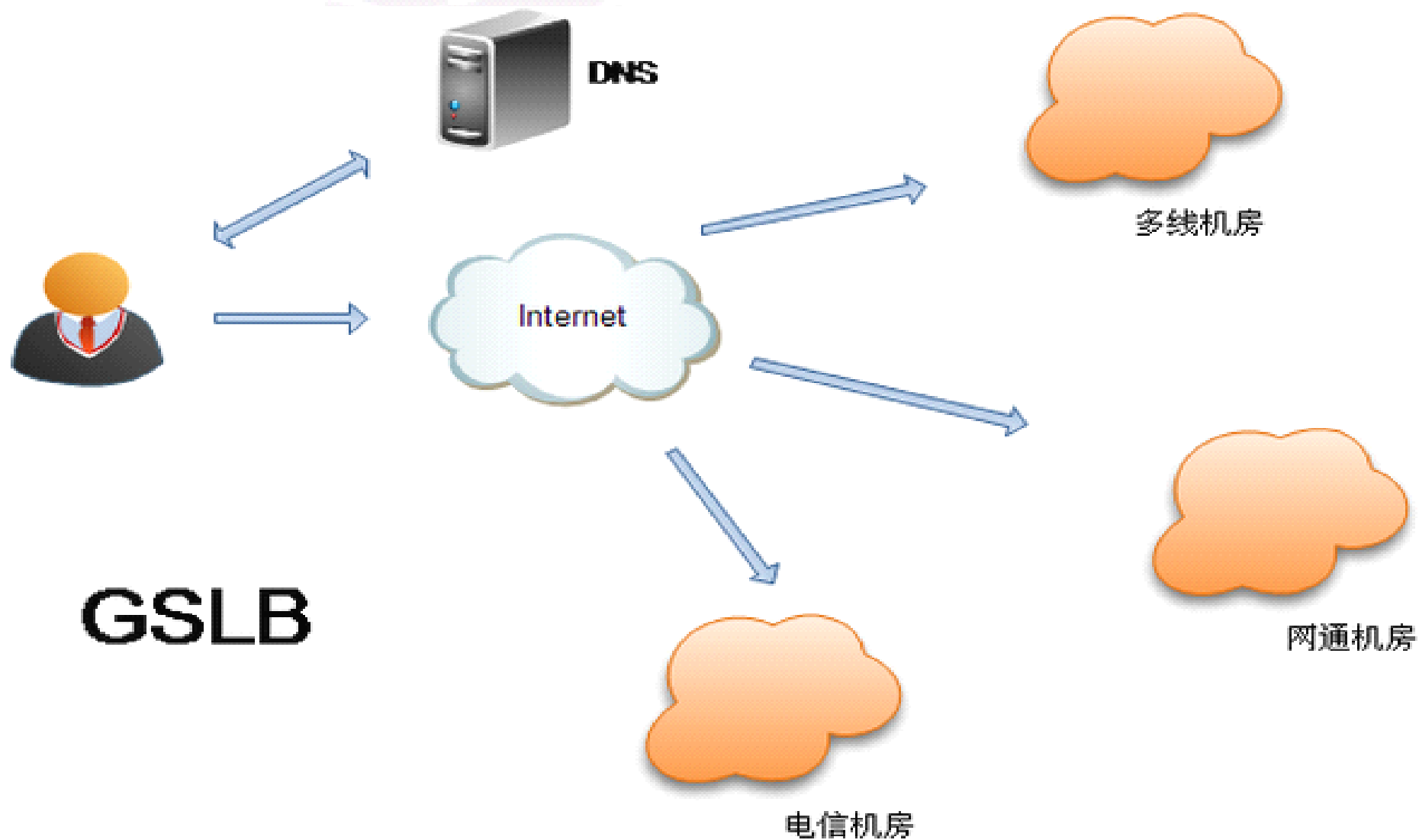
- 1. GSLB: 全局负载均衡
- 2. SLB: 服务器负载均衡
  - 四层交换 LVS
  - 七层交换 Nginx
- 3. Heartbeat 实现 HA
- 4. MySQL 数据库集群
- 5. 集群环境下的存储备份
- 6. 集群的监控及管理

# GSLB - Global Server Load Balance



- GSLB 是 Global Server Load Balance 的缩写，意思是全局负载均衡。
- 实现在广域网（包括互联网）上不同地域的服务器间的流量调配。
- 使用最佳的服务器（组）为最近的客户提供服务，从而确保访问质量。

# GSLB 架构图





## 多 IDC 与单 IDC 对比的好处

- 更高的可用性：不会因某一 IDC 机房由于“不可抗力”造成的网络中断而影响访问。
- 容灾：避免灾难性事件（比如地震）造成无可挽回的数据丢失。



## 多 IDC 与单 IDC 对比的坏处

- 实现复杂：流量分配问题，数据同步问题。
- 管理复杂：跨地区、距离远，维护麻烦。
- 成本较高：要在多个 IDC 租用机柜或机位。

# 利用 Bind9 的视图功能实现 GSLB 1



- cat named.conf
- ...
- include "acl\_chinanet.conf";
- view "chinanet" {
- match-clients { "chinanet"; };
- ...
- include "acl\_cnc.conf";
- view "cnc" {
- match-clients { "cnc"; };
- ...
- view "other" {
- match-clients { "any"; };
- ...

# 利用 Bind9 的视图功能实现 GSLB 2



- `cat acl_chinanet.conf`
- `acl "chinanet" {`
- `58.32.0.0/13;`
- `58.40.0.0/15;`
- `...`
- `222.222.0.0/15;`
- `222.240.0.0/13;`
- `};`
- `acl_cnc.conf` 则为网通的 IP 段。
- 然后分别定义各视图 zone 文件中域名所对应的 IP 地址。
- 详细原理可参考 GSLB using xBayDNS: <http://you.video.sina.com.cn/b/9144571-1435882772.html>





# SLB - Server Load Balancing

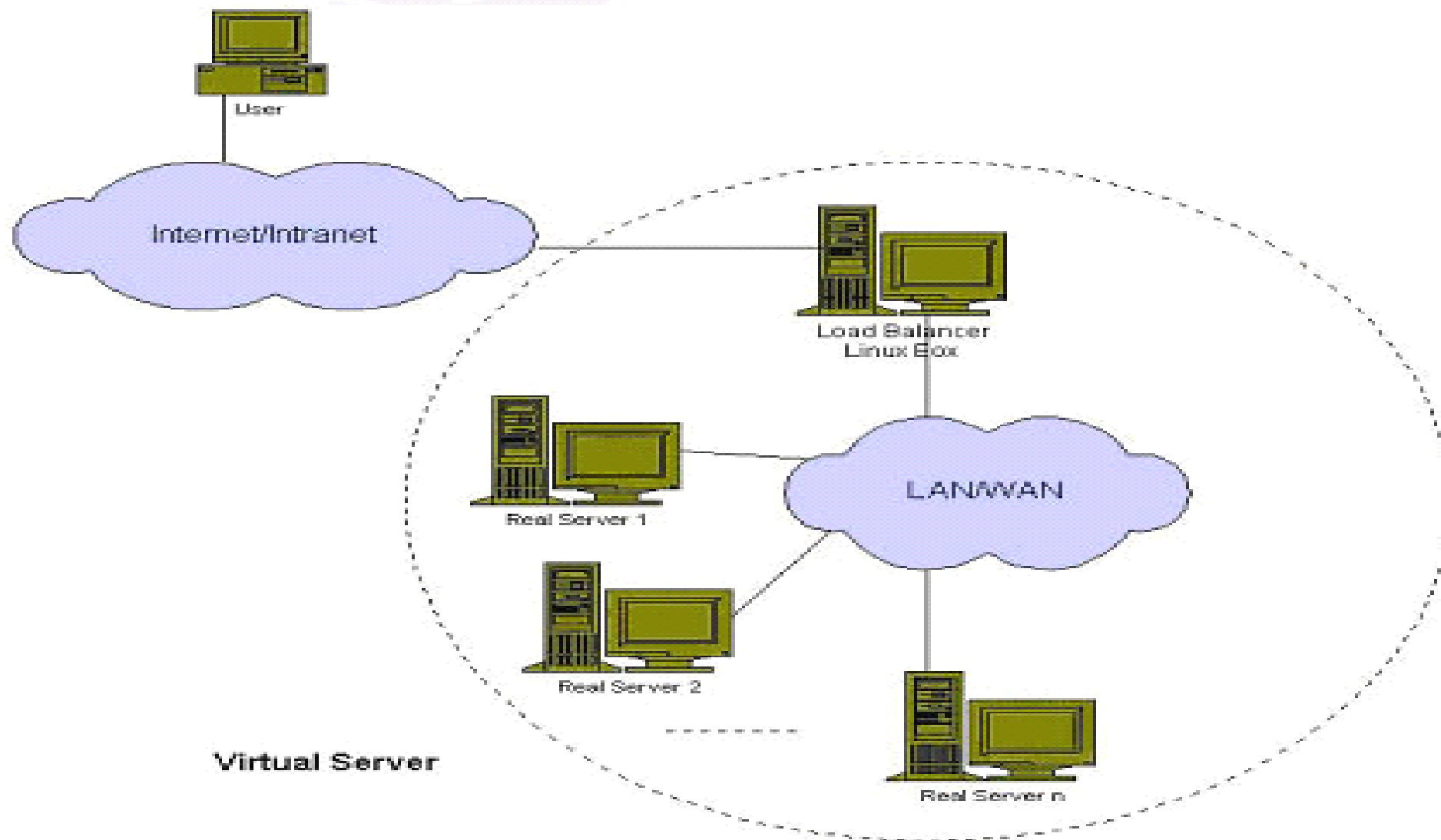
- SLB 是 Server Load Balancing 的缩写，意思是服务器负载均衡。可实现多个服务器之间的负载均衡。
- 当客户端向虚拟服务器（虚拟服务器是多个真实服务器的群集）发起连接时，通过均衡算法转发到真实服务器。



# LVS - Linux Virtual Server

- LVS 是在 Linux 内核中做四层交换。
- LVS 只用 128 个字节记录一个连接信息，不涉及到文件句柄操作，性能很高，可以支持上百万的并发连接。
- 老的 Redhat9 甚至 Redhat7.2 下的安装文档已经过时，Linux 2.6 内核已经包含了 ipvs 模块。
- 真实服务器需要解决 ARP 问题（ARP 缓存造成的 ARP “欺骗”）
- `arp_ignore = 1`      ## 只要这台机器上面任何一个设备有这个 IP，就响应 ARP 请求，并发送 MAC 地址应答。
- `arp_announce = 2`      ## 发送 ARP 请求时只使用本设备的 IP 地址和 MAC 地址，避免造成 ARP 欺骗。
- LVS 的有关中文文档：  
<http://www.linuxvirtualserver.org/zh/index.html>

# LVS 架构图





# Nginx

- Nginx ("engine x") 是一个高性能的 HTTP 和 反向代理 服务器，也是一个 IMAP/POP3/SMTP 代理服务器。
- 七层交换，大约能支持五万个并发连接。
- 可以通过定义 upstream 实现后端应用服务器（如运行 php-cgi 的机器）的负载均衡。
- ```
upstream php-fcgi {
```
- ```
    server 192.168.0.101:9000 weight=5;
```
- ```
    server 192.168.0.102:9000 weight=4;
```
- ```
    server 192.168.0.103:9000 weight=3;
```
- ```
}
```
- ```
location ~ .*\.php?$ {
```
- ```
    fastcgi_pass php-fcgi;
```
- ```
    include fastcgi_params;
```
- ```
}
```

# ngx\_http\_upstream\_hash\_module



- url hash 是用于提高 squid 群集命中率的一种算法。
- 通过 DNS 或 LVS 可以把流量分配到多个 squid 节点，但每台 squid 的缓存数据是重复的。
- upstream squid {
  - server squid1:3128;
  - server squid2:3128;
  - hash \$request\_uri;
  - hash\_again 1;
- }
- 定义 hash\_again ， 在有 squid 节点失效时，会重新 hash 。
- Nginx 的中文维基: <http://wiki.nginx.org/NginxChs>



# Heartbeat

- Heartbeat 最核心两个部分：心跳监测部分和资源接管部分。
- 心跳监测可以通过网络或串口进行，而且支持冗余链路，节点之间相互发送报文来告诉对方自己当前的状态。如果一个节点在指定的时间内未收到对方发送的报文，那么就认为对方失效。
- 这时需要启动资源接管模块来接管运行在对方主机上的资源或服务。
- 使用网络监测要注意同一网络中存在多个 HA 应用时的广播 (bcast) 造成的干扰，应该使用单播 (ucast)。
- 危险的脑裂 (split-brain)。
- Heartbeat 应用在操作存储设备的情况下是有一定的风险的。

# Heartbeat + LVS + Nginx + Squid + PHP

