Tedu.cn
达内教育

集群与存储

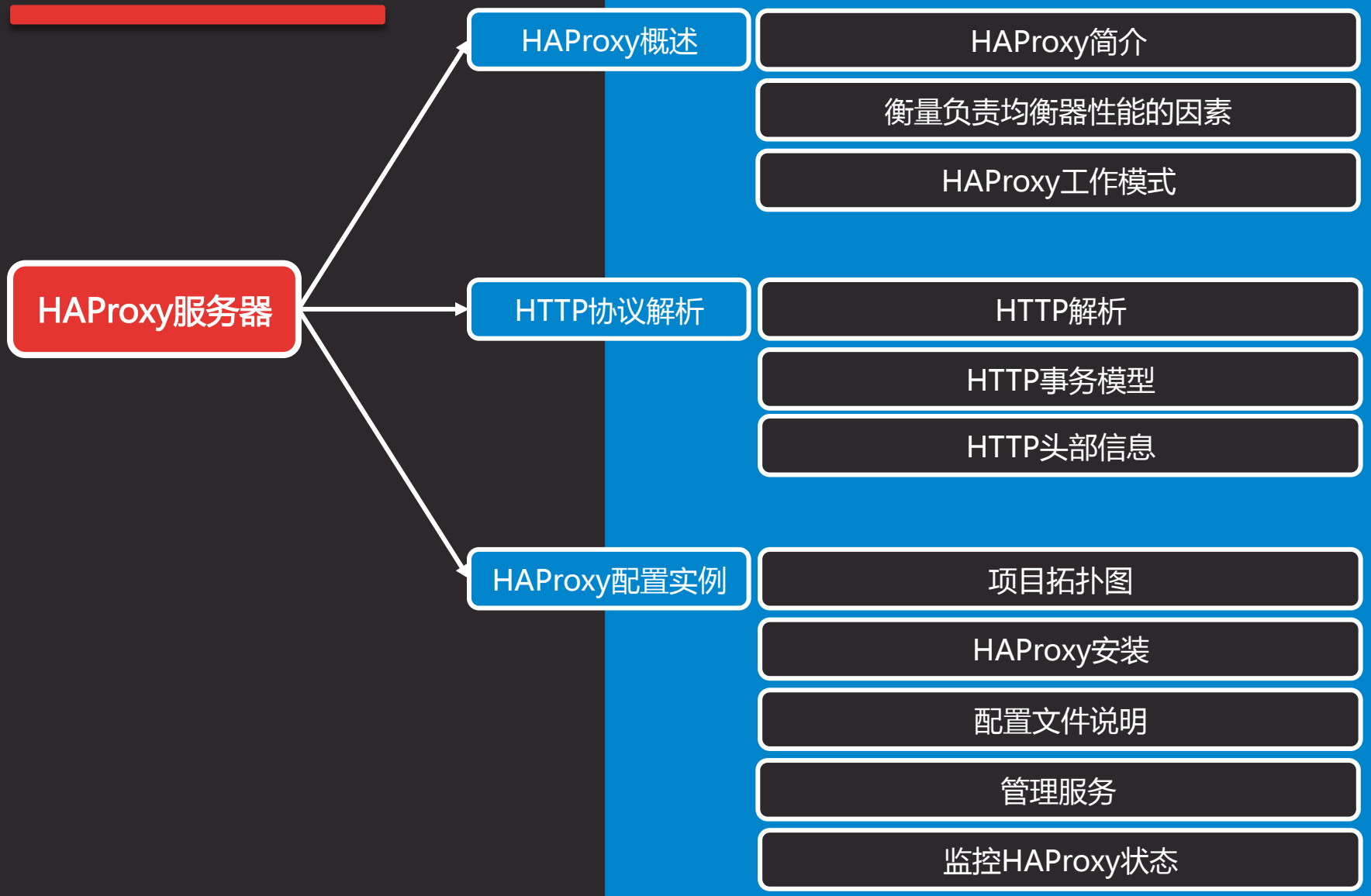
NSD CLUSTER

DAY03

内容

上午	09:00 ~ 09:30	作业讲解和回顾
	09:30 ~ 10:20	HAProxy服务器
	10:30 ~ 11:20	
	11:30 ~ 12:20	Keepalived热备
下午	14:00 ~ 14:50	
	15:00 ~ 15:50	Keepalived+LVS
	16:10 ~ 17:00	
	17:10 ~ 18:00	总结和答疑

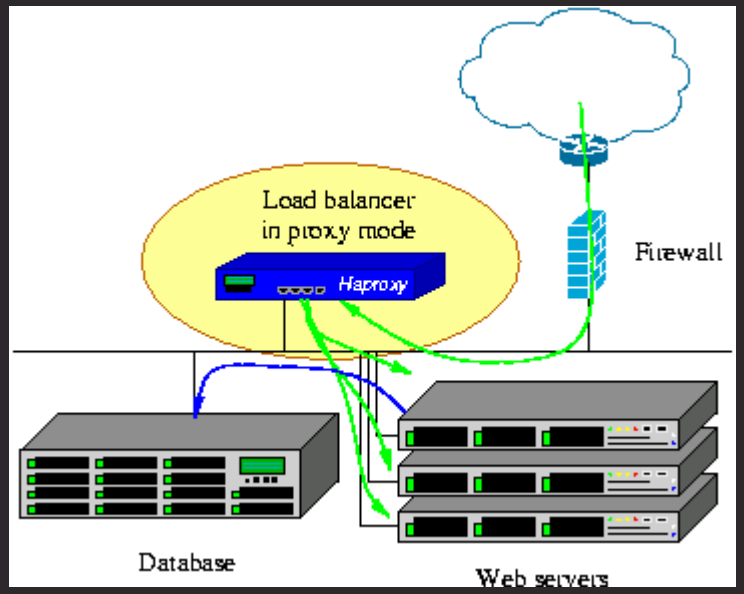
HAProxy服务器


HAProxy服务器
HAProxy概述
HAProxy简介
衡量负责均衡器性能的因素
HAProxy工作模式
HTTP协议解析
HTTP解析
HTTP事务模型
HTTP头部信息
HAProxy配置实例
项目拓扑图
HAProxy安装
配置文件说明
管理服务
监控HAProxy状态

HAProxy概述

HAProxy简介

- 它是免费、快速并且可靠的一种解决方案
- 适用于那些负载特大的web站点，这些站点通常又需要会话保持或七层处理
- 提供高可用性、负载均衡以及基于TCP和HTTP应用的代理



知识讲解

衡量负责均衡器性能的因素

- Session rate 会话率
  - 每秒钟产生的会话数
- Session concurrency 并发会话数
  - 服务器处理会话的时间越长，并发会话数越多
- Data rate 数据速率
  - 以MB/s或Mbps衡量
  - 大的对象导致并发会话数增加
  - 高会话数、高数据速率要求更多的内存

知识讲解

HAProxy工作模式

- mode http
  - 客户端请求被深度分析后再发往服务器
- mode tcp
  - 客户端与服务器之间建立会话，不检查第七层信息
- mode health
  - 仅做健康状态检查，已经不建议使用

知识讲解

Tedu.cn
达内教育

HTTP协议解析

HTTP解析

- 当HAProxy运行在HTTP模式下，HTTP请求（Request）和响应（Response）均被完全分析和索引，这样便于创建恰当的匹配规则
- 理解HTTP请求和响应，对于更好的创建匹配规则至关重要

知识讲解

HTTP事务模型

- HTTP协议是事务驱动的
- 每个请求（Request）仅能对应一个响应（Response）
- 常见模型：
  - HTTP close
  - Keep-alive
  - Pipelining

知识讲解

HTTP事务模型（续1）

- HTTP close
  - 客户端向服务器建立一个TCP连接
  - 客户端发送请求给服务器
  - 服务器响应客户端请求后即断开连接
  - 如果客户端到服务器的请求不只一个，那么就要不断的去建立连接
  - TCP三次握手消耗相对较大的系统资源，同时延迟较大

知识讲解

HTTP事务模型（续2）

- Keep-alive
  - 一次连接可以传输多个请求
  - 客户端需要知道传输内容的长度，以避免无限期的等待传输结束
  - 降低两个HTTP事务间的延迟
  - 需要相对较少的服务器资源

知识讲解

HTTP事务模型（续3）

- Pipelining
  - 仍然使用Keep-alive
  - 在发送后续请求前，不用等前面的请求已经得到回应
  - 适用于有大量图片的页面
  - 降低了多次请求之间的网络延迟

知识讲解

HTTP头部信息

- 请求头部信息
  - 方法：GET
  - URI：/serv/login.php?lang=en&profile=2
  - 版本：HTTP/1.1

Line Number	Contents
1	GET /serv/login.php?lang=en&profile=2 HTTP/1.1
2	Host: www.mydomain.com
3	User-agent: my small browser
4	Accept: image/jpeg, image/gif
5	Accept: image/png

知识讲解

HTTP头部信息（续1）

- 请求头部信息
  - 请求头包含许多有关的客户端环境和请求正文的有用信息，如浏览器所使用的语言、请求正文的长度等

Line Number	Contents
1	GET /serv/login.php?lang=en&profile=2 HTTP/1.1
2	Host: www.mydomain.com
3	User-agent: my small browser
4	Accept: image/jpeg, image/gif
5	Accept: image/png

知识讲解

HTTP头部信息（续2）

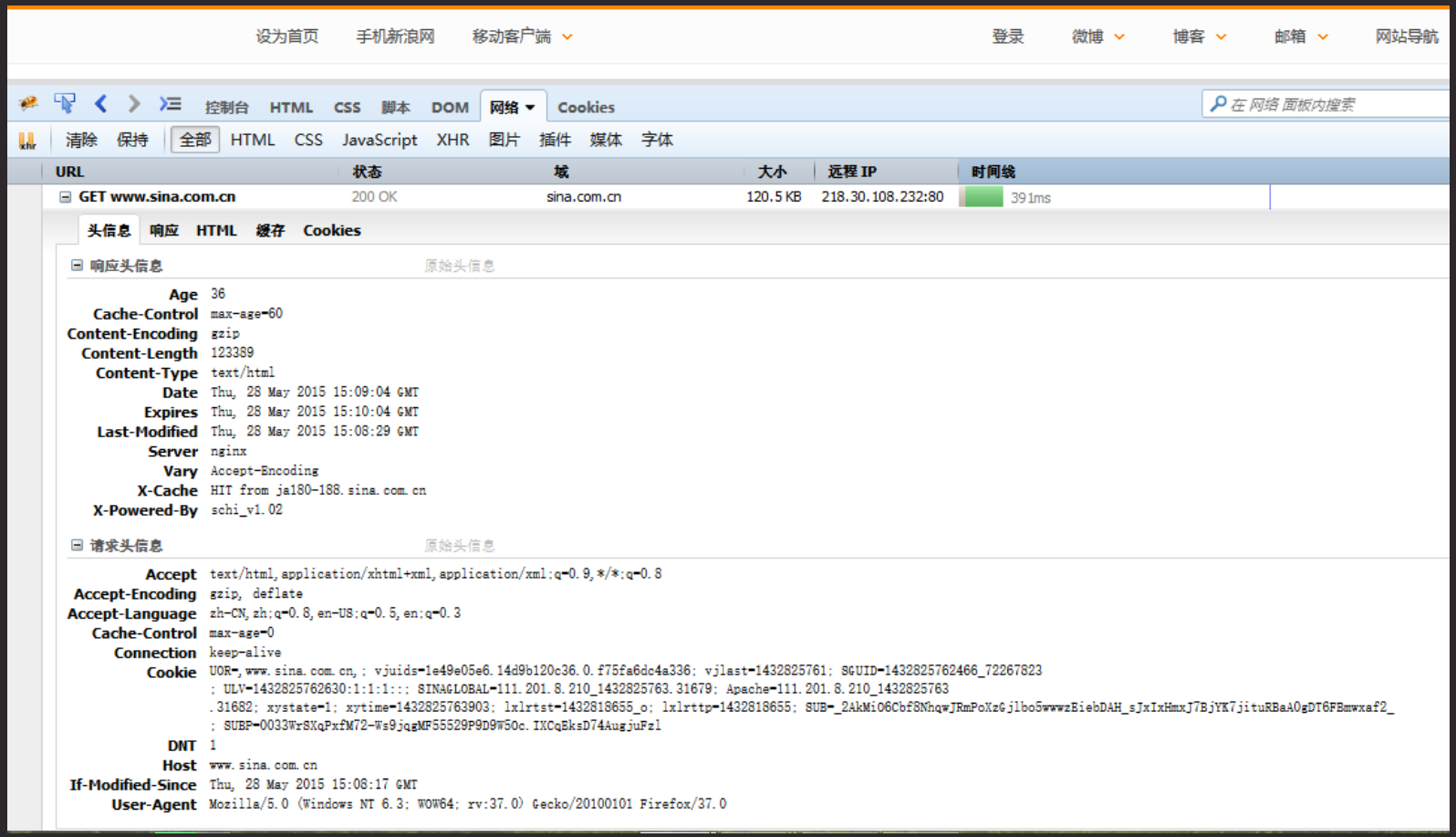
- 响应头部信息
  - 版本：HTTP/1.1
  - 状态码：200
  - 原因：OK

Line Number	Contents
1	HTTP/1.1. 200 OK
2	Content-length: 350
3	Content-Type: text/html

知识讲解

HTTP头部信息（续3）

- 新浪页面实例

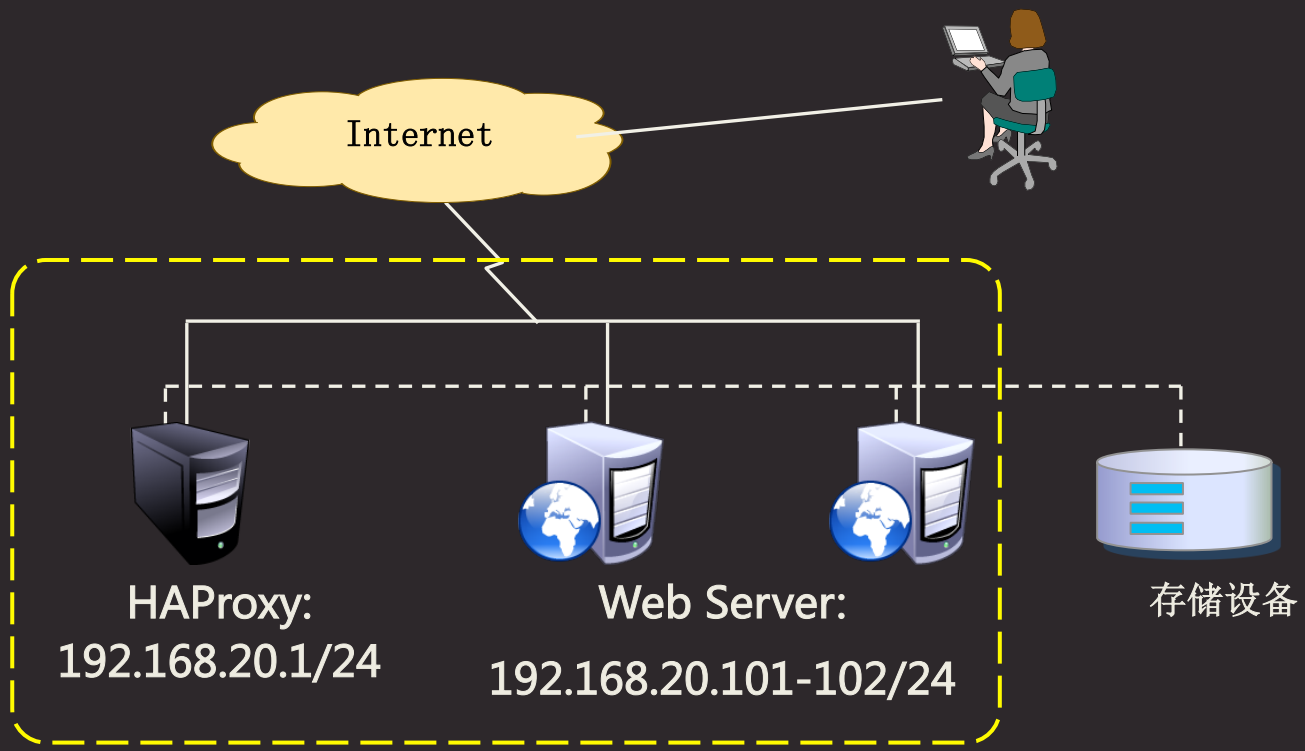


知识讲解

HAProxy配置实例

项目拓扑图

知识讲解



HAProxy安装

- RHEL7光盘中内置了HAProxy，只要配置好yum，可以直接安装

```
[root@svr1 ~]# yum install haproxy
```

知识讲解

配置文件说明

- HAProxy配置参数来源
  - 命令行：总是具有最高优先级
  - global部分：全局设置进程级别参数
  - 代理声明部分
    来自于default、listen、frontend和backend

知识讲解

配置文件说明（续1）

- 配置文件可由如下部分构成：
  - default
    - 为后续的其他部分设置缺省参数
    - 缺省参数可以被后续部分重置
  - frontend
    - 描述接收客户端侦听套接字（socket）集
  - backend
    - 描述转发链接的服务器集
  - listen
    - 把frontend和backend结合到一起的完整声明

知识讲解

配置文件说明（续2）

- /etc/haproxy/haproxy.cfg

```
global
 log 127.0.0.1 local2   ###[err warning info debug]
 chroot /usr/local/haproxy
 pidfile /var/run/haproxy.pid ###haproxy的pid存放路径
maxconn 4000     ###最大连接数，默认4000
 user haproxy
 group haproxy
 daemon      ###创建1个进程进入deamon模式运行
```

知识讲解

配置文件说明（续3）

- /etc/haproxy/haproxy.cfg

```
defaults
 mode http    ###默认的模式mode { tcp|http|health } log global   ###采用全局定义的日志
option dontlognull  ###不记录健康检查的日志信息
option httpclose  ###每次请求完毕后主动关闭http通道
option httplog   ###日志类别http日志格式
option forwardfor  ###后端服务器可以从Http Header中获得客户端ip
 option redispatch  ###serverid服务器挂掉后强制定向到其他健康服务器
timeout connect 10000 #如果backend没有指定，默认为10s
 timeout client 300000 ###客户端连接超时
timeout server 300000 ###服务器连接超时
maxconn  60000  ###最大连接数
retries  3   ###3次连接失败就认为服务不可用，也可以通过后面设置
```

知识讲解

配置文件说明（续4）

- /etc/haproxy/haproxy.cfg

```
listen stats
    bind 0.0.0.0:1080   #监听端口
    stats refresh 30s   #统计页面自动刷新时间
    stats uri /stats    #统计页面url
    stats realm Haproxy Manager #统计页面密码框上提示文本
    stats auth admin:admin  #统计页面用户名和密码设置
    #stats hide-version   #隐藏统计页面上HAProxy的版本信息
```

知识讲解

配置文件说明（续5）

- /etc/haproxy/haproxy.cfg

```
listen  websrv-rewrite 0.0.0.0:80
        cookie  SERVERID rewrite
        balance roundrobin
        server  web1 192.168.20.101:80 cookie \
        app1inst1 check inter 2000 rise 2 fall 5
        server  web2 192.168.20.102:80 cookie \
        app1inst2 check inter 2000 rise 2 fall 5
```

知识讲解

管理服务

- 启动服务

  [root@svr1 ~]# systemctl start haproxy

- 停止服务

  [root@svr1 ~]# systemctl stop haproxy

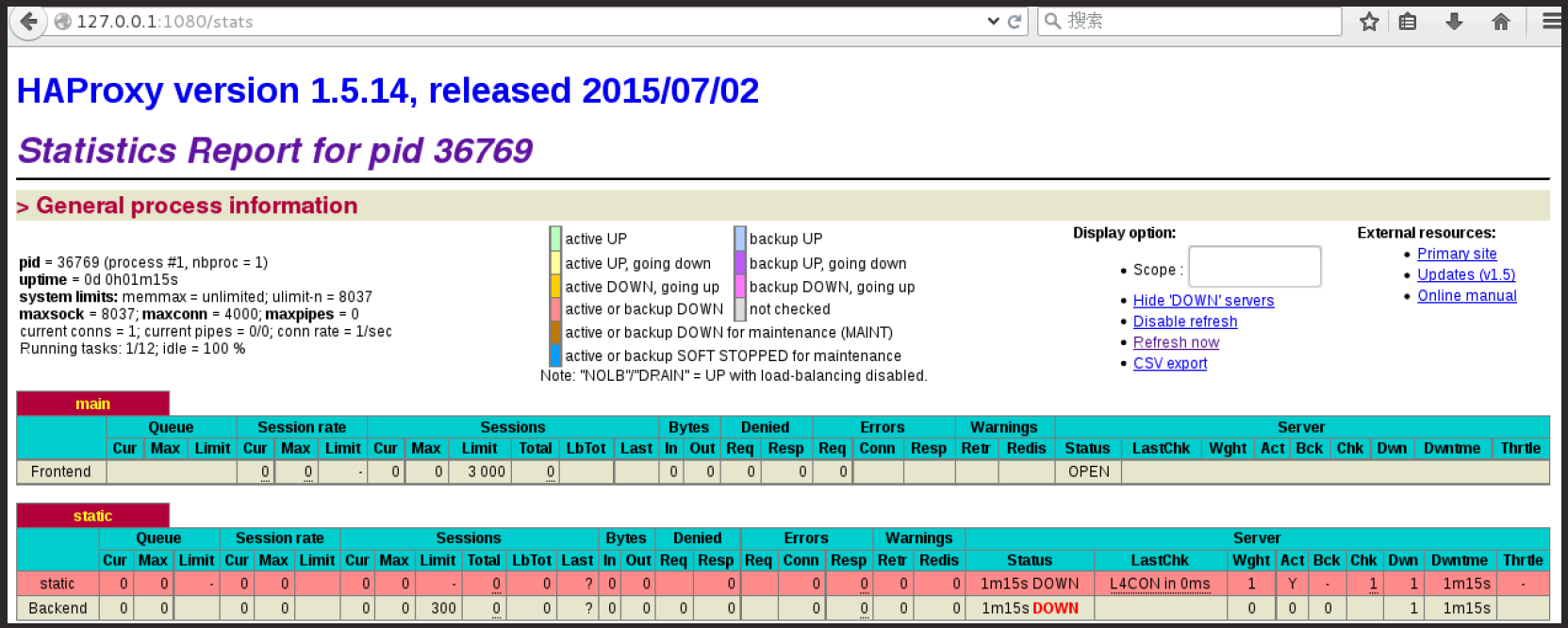
- 查看状态

  [root@svr1 ~]# systemctl status haproxy

知识讲解

监控HAProxy状态

知识讲解



127.0.0.1:1080/stats

HAProxy version 1.5.14, released 2015/07/02

Statistics Report for pid 36769

> General process information

pid = 36769 (process #1, nbproc = 1)
uptime = 0d 0h01m15s
system limits: memmax = unlimited; ulimit-n = 8037
maxsock = 8037; **maxconn** = 4000; **maxpipes** = 0
current conns = 1; current pipes = 0/0; conn rate = 1/sec
Running tasks: 1/12; idle = 100 %

- active UP
- active UP, going down
- active DOWN, going up
- active or backup DOWN
- active or backup DOWN for maintenance (MAINT)
- active or backup SOFT STOPPED for maintenance
- backup UP
- backup UP, going down
- backup DOWN, going up
- not checked

Note: "NOLB"/"DRAIN" = UP with load-balancing disabled.

Display option:

- Scope :
- Hide 'DOWN' servers
- Disable refresh
- Refresh now
- CSV export

External resources:

- Primary site
- Updates (v1.5)
- Online manual

main

	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
Frontend				0	0	-	0	0	3 000	0			0	0	0	0	0					OPEN								

static

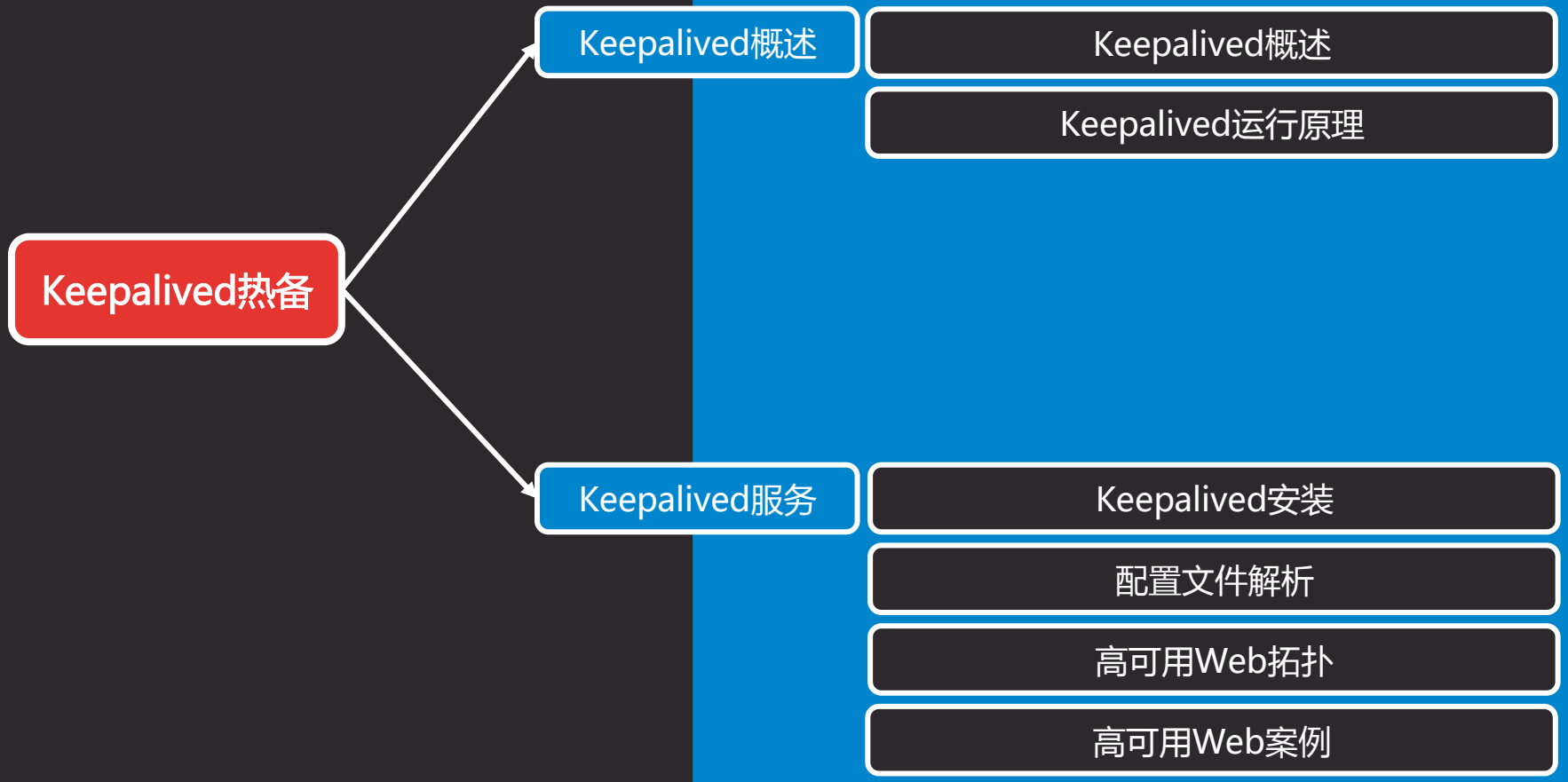
	Queue			Session rate			Sessions						Bytes		Denied		Errors			Warnings		Server								
	Cur	Max	Limit	Cur	Max	Limit	Cur	Max	Limit	Total	LbTot	Last	In	Out	Req	Resp	Req	Conn	Resp	Retr	Redis	Status	LastChk	Wght	Act	Bck	Chk	Dwn	Dwntme	Thrtle
static	0	0	-	0	0		0	0	-	0	0	?	0	0		0		0	0	0	0	1m15s DOWN	L4CON in 0ms	1	Y	-	1	1	1m15s	-
Backend	0	0		0	0		0	0	300	0	0	?	0	0	0	0		0	0	0	0	1m15s DOWN		0	0	0		1	1m15s	

案例1：配置HAProxy负载平衡集群

- 准备三台虚拟机
  - 两台做Web服务器，一台安装HAProxy
- 安装并配置HAProxy
  - 发往HAProxy的连接请求，分发到真正的Web服务器
  - 把HAProxy设置为开机自动启动
- 设置HAProxy以实现监控，并查看监控信息

课堂练习

Keepalived热备



Keepalived概述

Keepalived概述

- 调度器出现单点故障，如何解决？
- Keepalived实现了高可用集群
- Keepalived最初是为LVS设计的，专门监控各服务器节点的状态
- Keepalived后来加入了VRRP功能，防止单点故障

知识讲解

Keepalived运行原理

- Keepalived检测每个服务器节点状态
- 服务器节点异常或工作出现故障， Keepalived将故障节点从集群系统中剔除
- 故障节点恢复后，Keepalived再将其加入到集群系统中
- 所有工作自动完成，无需人工干预

知识讲解

Keepalived服务

Keepalived安装

- RHEL7的光盘中已经包含Keepalived软件包，只要配置好yum，指向光盘源即可安装

```
[root@svr1 ~]# yum install -y keepalived
```

知识讲解

配置文件解析

- /etc/keepalived/keepalived.conf

知识讲解

```
global_defs {
  notification_email {
    admin@tarena.com.cn                    //设置报警收件人邮箱
  }
  notification_email_from ka@localhost     //设置发件人
  smtp_server 192.168.20.1                 //定义邮件服务器
  smtp_connect_timeout 30
  router_id  LVS_devel                     //设置路由ID号
}
```

配置文件解析（续1）

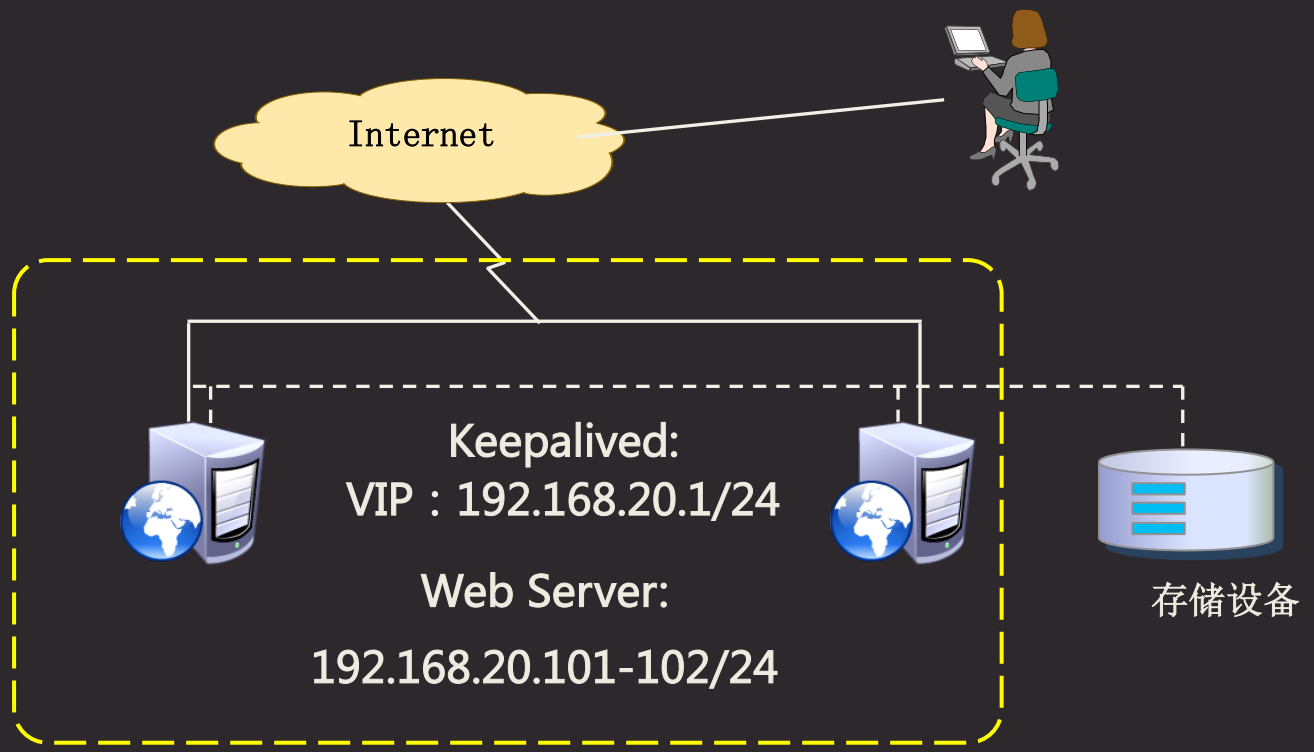
- /etc/keepalived/keepalived.conf

```
vrrp_instance VI_1 {
  state MASTER                  //主服务器为MASTER，辅助为SLAVE
  interface eth0                //定义网络接口
  virtual_router_id 51          //主辅VRID号必须一致
  priority 100                  //服务器优先级
  advert_int 1
  authentication {
    auth_type pass
    auth_pass forlvs            //主辅服务器密码必须一致
  }
  virtual_ipaddress { 192.168.20.100 }
}
```

知识讲解

高可用Web拓扑

- 使用Keepalived为主从设备提供VIP地址漂移



知识讲解

高可用Web案例

- 配置Web服务器

知识讲解

```
[root@web1 ~]# ifconfig eth0 192.168.20.101
[root@web1 ~]# yum -y install httpd
[root@web1 ~]# systemctl start httpd; systemctl enable httpd
[root@web2 ~]# ifconfig eth0 192.168.20.102
[root@web2 ~]# yum -y install httpd
[root@web2 ~]# systemctl start httpd; systemctl enable httpd
```

高可用Web案例（续1）

- 使用Keepalived为服务器提供VIP

```
vrrp_instance VI_1 {
  state MASTER                  //主服务器为MASTER，辅助为SLAVE
  interface eth0                //定义网络接口
  virtual_router_id 51          //主辅VRID号必须一致
  priority 100                  //服务器优先级
  advert_int 1
  authentication {
    auth_type pass
    auth_pass forlvs            //主辅服务器密码必须一致
  }
  virtual_ipaddress { 192.168.20.1 }
}
```

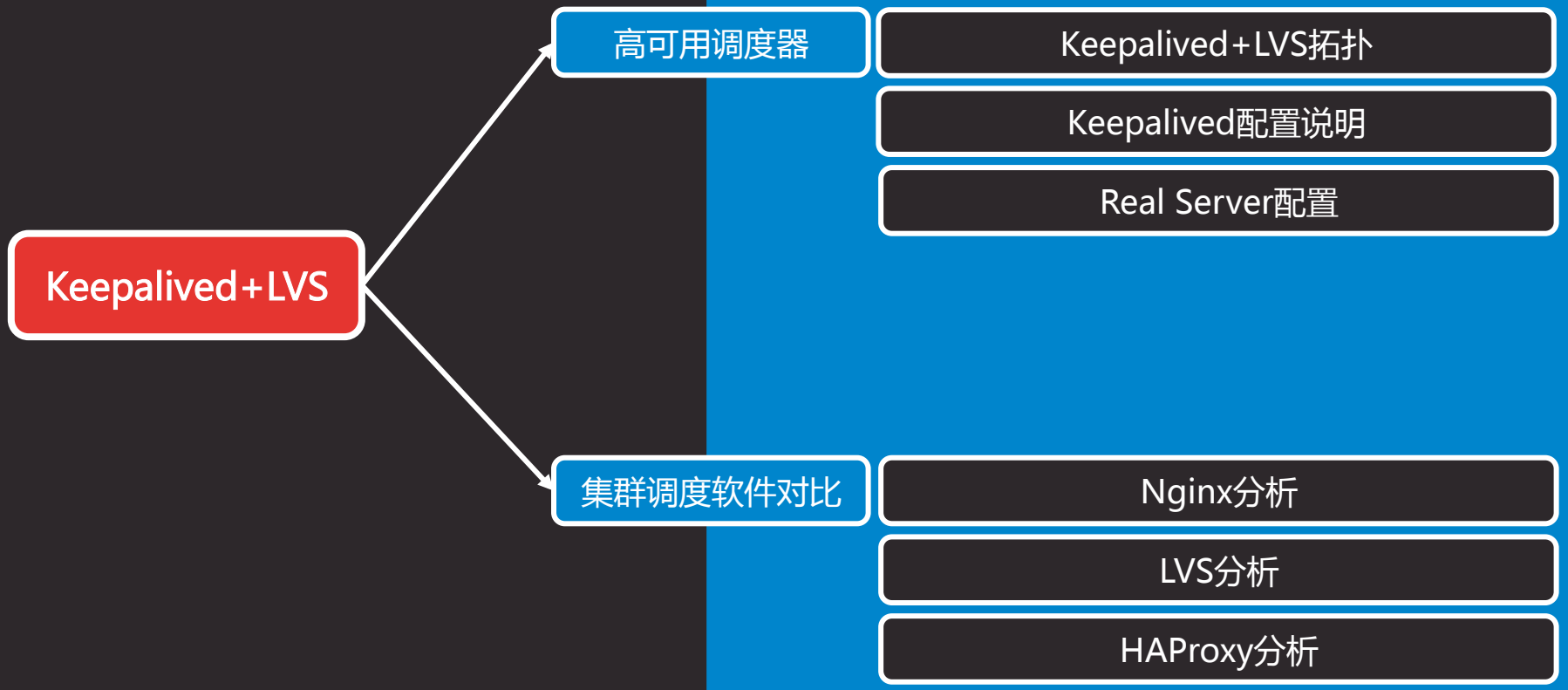
知识讲解

案例2：Keepalived高可用服务器

- 使用Keepalived实现web服务器的高可用
  - Web服务器IP地址分别为172.16.0.10和172.16.0.20
  - Web服务器的VIP地址为172.16.0.1
  - 客户端通过访问VIP地址访问Web页面

课堂练习

Keepalived+LVS

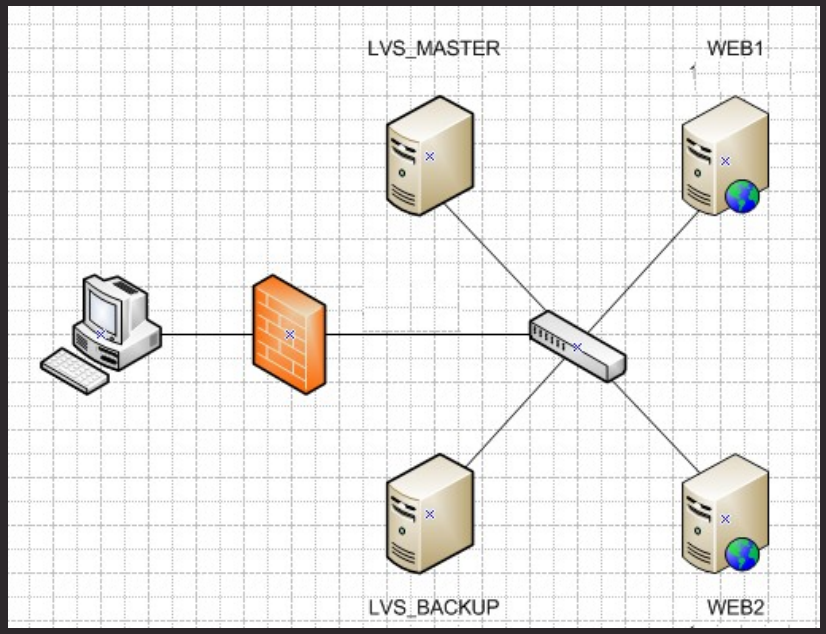


Tedu.cn
达内教育

高可用调度器

Keepalived+LVS拓扑

- 使用Keepalived高可用解决调度器单点失败问题
- 主、备调度器上配置LVS
- 主调度器异常时，Keepalived启用备用调度器调度器



知识讲解

Keepalived配置说明

- LVS相关信息通过Keepalived配置即可
- 主要配置文件说明如下：

```
global_defs {
  notification_email {
    admin@tarena.com.cn
  }
  notification_email_from ka@localhost
  smtp_server 192.168.20.1
  smtp_connect_timeout 30
  router_id  LVS_devel
}
```

知识讲解

Keepalived配置说明（续1）

- VRRP实例设置

```
vrrp_instance VI_1 {
  state MASTER                  //主服务器为MASTER，辅助为SLAVE
  interface eth0
  virtual_router_id 51
  priority 100                  //优先级
  advert_int 1
  authentication {
    auth_type pass
    auth_pass forlvs            //主辅服务器密码必须一致
  }
  virtual_ipaddress { 192.168.20.100 }
}
```

知识讲解

Keepalived配置说明（续2）

```
virtual_server 192.168.20.100 80 {          //设置VIP为192.168.20.100
  delay_loop 6
  lb_algo rr                                //设置LVS调度算法为RR
  lb_kind DR                                //设置LVS的模式为DR
  persistence_timeout 50
  protocol TCP
  real_server 192.168.20.150 80 {
    weight 3                                //设置权重为3
    TCP_CHECK {
            connect_timeout 3
            nb_get_retry 3
            delay_before_retry 3
    }
  }
 real_server 192.168.20.151 80 { 同real1}
}
```

知识讲解

Real Server配置

- 真实服务器运行在DR模式下
- 修改内核参数，并附加VIP
- 详细配置参见LVS相关章节

知识讲解

Tedu.cn
达内教育

集群调度软件对比

Nginx分析

- 优点
  - 工作在7层，可以针对http做分流策略
  - 正则表达式比HAProxy强大
  - 安装、配置、测试简单，通过日志可以解决多数问题
  - 并发量可以达到几万次
  - Nginx还可以作为Web服务器使用
- 缺点
  - 仅支持http、https、mail协议，应用面小
  - 监控检查仅通过端口，无法使用url检查

知识讲解

LVS分析

- 优点
  - 负载能力强，工作在4层，对内存、CPU消耗低
  - 配置性低，没有太多可配置性，减少人为错误
  - 应用面广，几乎可以为所有应用提供负载均衡
- 缺点
  - 不支持正则表达式，不能实现动静分离
  - 如果网站架构庞大，LVS-DR配置比较繁琐

知识讲解

HAProxy分析

- 优点
  - 支持session、cookie功能
  - 可以通过url进行健康检查
  - 效率、负载均衡速度，高于Nginx，低于LVS
  - HAProxy支持TCP，可以对MySQL进行负载均衡
  - 调度算法丰富
- 缺点
  - 正则弱于Nginx
  - 日志依赖于syslogd，不支持apache日志

知识讲解

案例3：Keepalived+LVS服务器

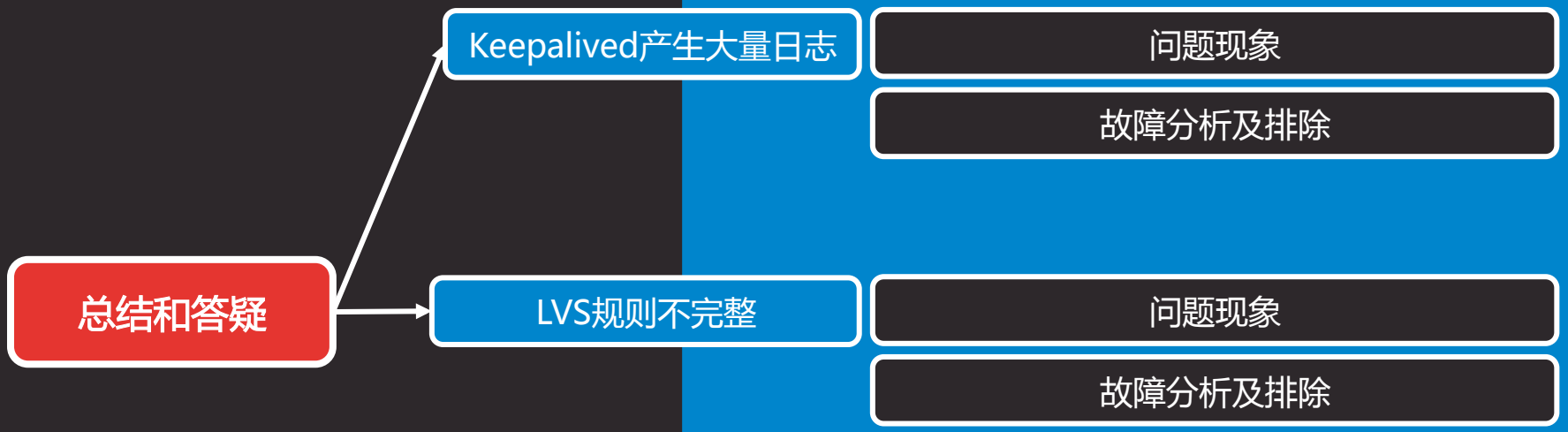
- 准备5台服务器
  - 两台用于Real Server
  - 两台用于搭建高可用、负载平衡集群
  - 一台作为路由器
- 在Real Server上配置VIP并调整内核参数
- 两台调度器节点均安装Keepalived和LVS
- 通过Keepalived配置DR模式的LVS

课堂练习

Tedu.cn
达内教育

总结和答疑

总结和答疑


总结和答疑
Keepalived产生大量日志
问题现象
故障分析及排除
LVS规则不完整
问题现象
故障分析及排除

Keepalived产生大量日志

问题现象

- 当观察/var/log/messages日志时，发现该文件每秒钟都产生了很多条日志记录
- 如果不及时解决，该文件会迅速增长

知识讲解

故障分析及排除

- 原因分析
  - Keepalived的工作原理与VRRP相同
  - VRRP相同组要求有相同的密码、VIP和组号，如果不一致就会产生日志通知
- 解决办法
  - 检查两台Keepalived配置，将虚拟IP、虚拟路由器ID和密码修改成一样的

知识讲解

LVS规则不完整

问题现象

- 通过Keepalived配置LVS规则，查看LVS规则时，只有一台real server
- 经检查real server工作都未出现异常

知识讲解

故障分析及排除

- 原因分析
  - 直接访问real server没有异常
  - 问题应该出现在Keepalived配置文件
- 解决办法
  - 经检查，发现配置文件中，TCP_CHECK与后面的花括号少了一个空格

知识讲解