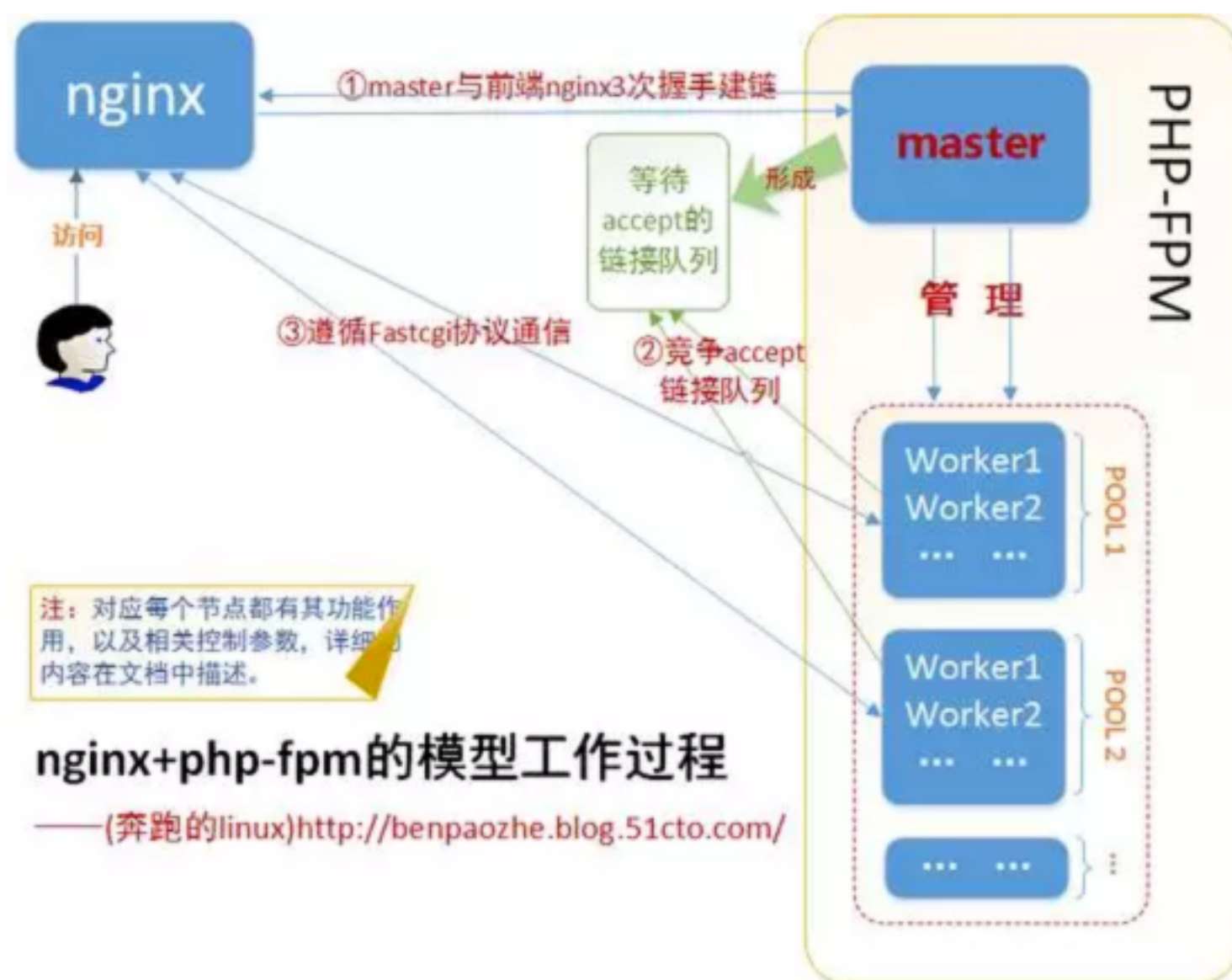


5.1.1 php-fpm

php-fpm

背景知识

nginx+php-fpm模式



优化建议

优化建议

`pm = static | dynamic | ondemand` 静态池、服务优先、内存优先

`pm.max_children = 256` 开启的最大 php 进程数

`pm.max_requests = 1024` 在执行了 1024 个请求后重启 worker 进程

web 服务的机器是 12 核 cpu 、 16G 内存， nginx 开启 12 个 worker 进程， php 开启 256 个进程，

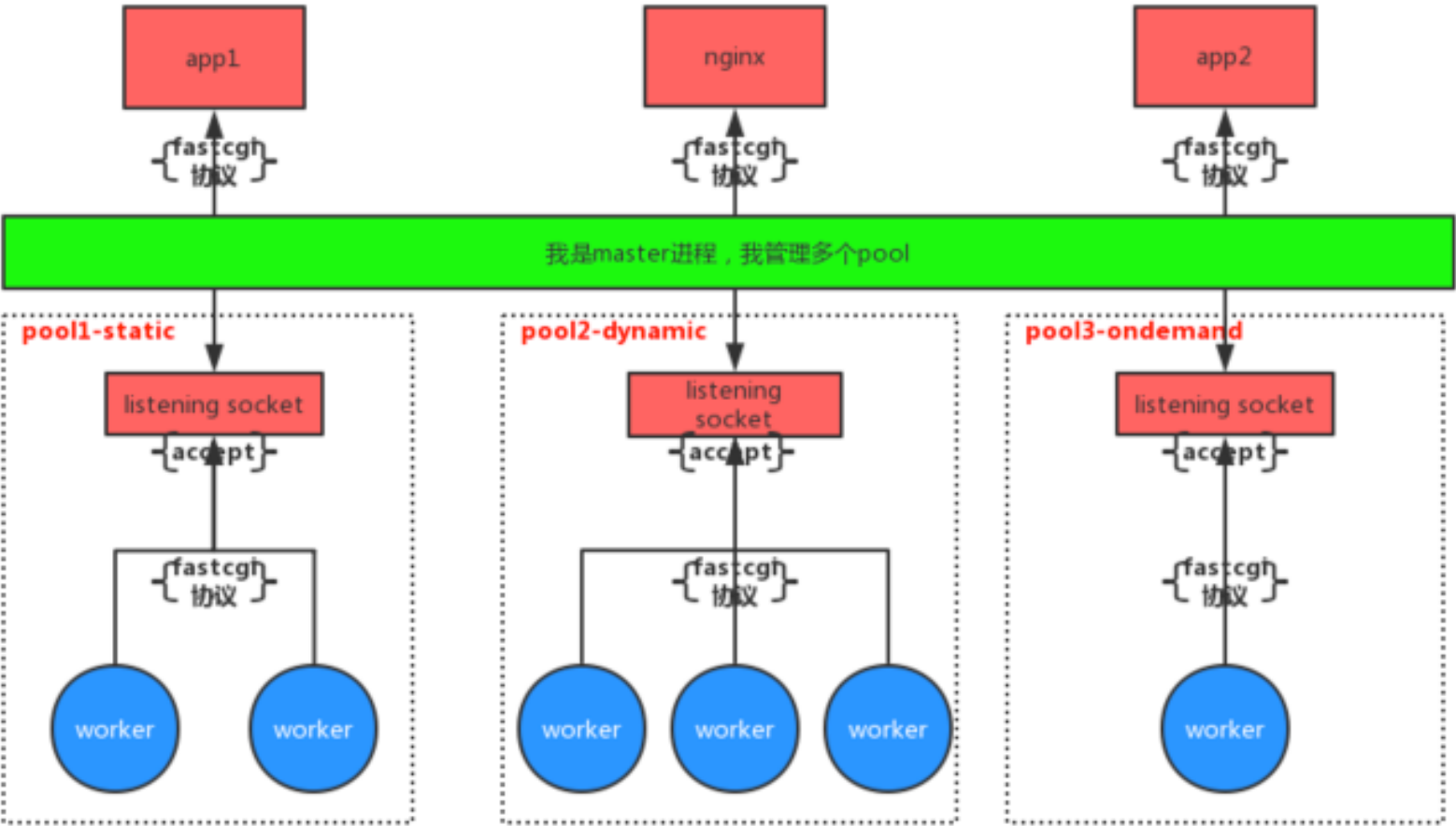
跑起来后每个进程大概占用 30M 内存，也就是 $(256+12) * 30=8G$ ，这种静态池的配置大大减少了prefork 进程带来的开销，RT时间100ms以内的占到85%（这个与程序写的如何有关）。

CGI

FAST-CGI

工作原理

三种工作模式



ondemand

```
[test]
listen = 127.0.0.1:9001
pm = ondemand
pm.process_idle_timeout = 60
pm.max_children = 10
```

当前pool的名字为test

- 优点
- 缺点

连接来了(注意是“连接”这个用语，所以用telnet连接来的也算)，这个连接来自nginx，或许其他程序



3、新建一个worker

2、检查test下是否有空闲的worker，如果有，退出

1、检查test下已经存在的worker数量，如果 \geq pm.max_children，打印WARNING日志，退出

ondemand原理图

dynamic

```
[test]
listen = 127.0.0.1:9001
pm = dynamic
pm.max_children = 10
pm.start_servers = 2
pm.min_spare_servers = 1
pm.max_spare_servers = 6
```

当前pool的名字为test

优点

缺点

我是master进程，我设置了1秒的定时器

每秒触发一次：
按照当前的idle worker数量，动态调整worker数量

初始启动 pm.start_servers 个worker



dynamic原理图

pm.min_spare_servers/pm.max_spare_servers有效范围(0,pm.max_children]

```
pm.max_children> 0
```

```
pm.min_spare_servers<=pm.max_spare_servers
```

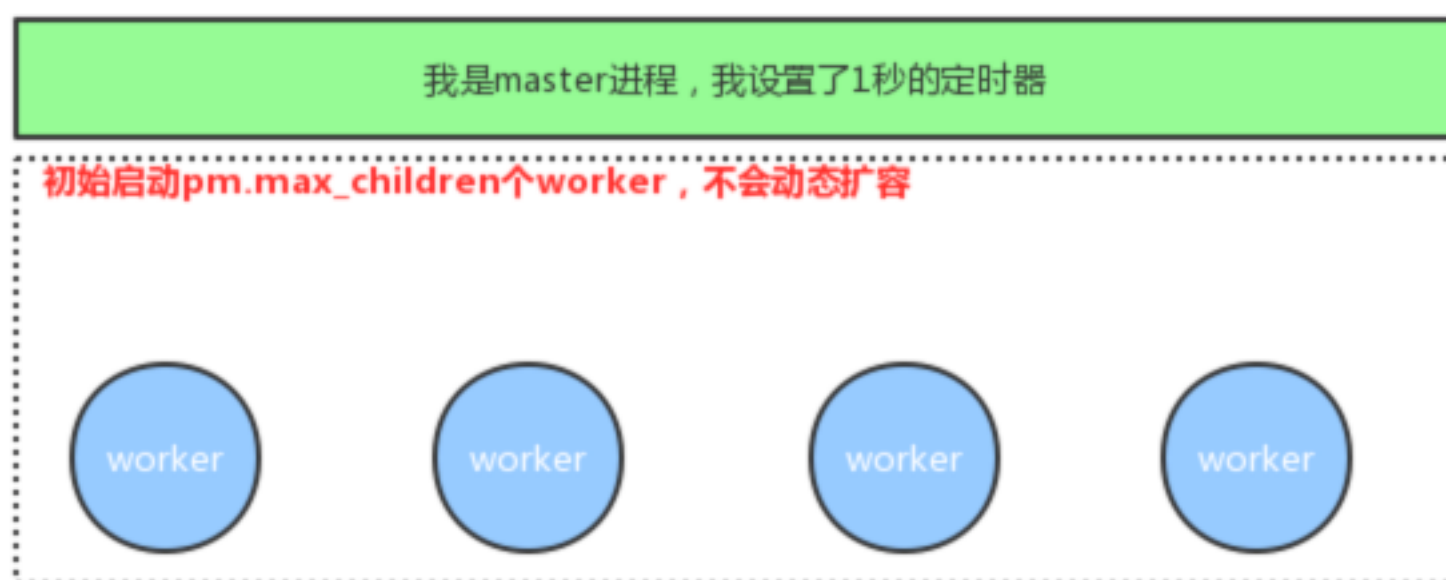
pm.start_servers有效范围[pm.min_spare_servers,pm.max_spare_servers]如果没有配置, 默认 $\text{pm.min_spare_servers} + (\text{pm.max_spare_servers} - \text{pm.min_spare_servers}) / 2$

static

```
[test]
listen = 127.0.0.1:9001
pm = static
pm.max_children = 40
```

当前pool的名字为test

优点缺点



配置优化

TCP切换为UNIX域套接字

```
upstream backend
{
    # UNIX domain sockets
    server unix:/var/run/fastcgi.sock;

    # TCP sockets
    # server 127.0.0.1:8080;
}
```

调整工作进程数

```
// 机器处理器个数
cat /proc/cpuinfo | grep processor
```

```
# We have 16 cores
worker_processes 16;

# connections per worker
events
{
    use epoll;
    worker_connections 4096;
    multi_accept on;
}
```

设置upstream负载均衡

```
upstream backend {
    server unix:/var/run/php5-fpm.sock1 weight=100 max_fails=5
fail_timeout=5;
    server unix:/var/run/php5-fpm.sock2 weight=100 max_fails=5
fail_timeout=5;
}
```

禁用访问日志文件

```
access_log off;
log_not_found off;
error_log /var/log/nginx-error.log warn;

// 如果不能关闭, 新增日志缓冲区, 减少频繁IO操作
access_log /var/log/nginx/access.log main buffer=16k;
```

启用GZip

```
gzip on;
gzip_disable "msie6";
gzip_vary on;
gzip_proxied any;
```

```
gzip_comp_level 6;  
gzip_min_length 1100;  
gzip_buffers 16 8k;  
gzip_http_version 1.1;  
gzip_types text/plain text/css application/json application/x-javascript  
text/xml application/xml application/xml+rss text/javascript;
```

缓存被频繁访问的文件相关的信息

```
open_file_cache max=200000 inactive=20s;  
open_file_cache_valid 30s;  
open_file_cache_min_uses 2;  
open_file_cache_errors on;
```

调整客户端超时时间

```
client_max_body_size 500M;  
client_body_buffer_size 1m;  
client_body_timeout 15;  
client_header_timeout 15;  
keepalive_timeout 2 2;  
send_timeout 15;  
sendfile on;  
tcp_nopush on;  
tcp_nodelay on;
```

调整输出缓冲区大小

```
fastcgi_buffers 256 16k;  
fastcgi_buffer_size 128k;  
fastcgi_connect_timeout 3s;  
fastcgi_send_timeout 120s;  
fastcgi_read_timeout 120s;  
reset_timedout_connection on;  
server_names_hash_bucket_size 100;
```

/etc/sysctl.conf调优

```
# Recycle Zombie connections  
net.inet.tcp.fast_finwait2_recycle=1
```

```
net.inet.tcp.maxtcptw=200000

# Increase number of files
kern.maxfiles=65535
kern.maxfilesperproc=16384

# Increase page share factor per process
vm.pmap.pv_entry_max=54272521
vm.pmap.shpgperproc=20000

# Increase number of connections
vfs.vmiodirenable=1
kern.ipc.somaxconn=3240000
net.inet.tcp.rfc1323=1
net.inet.tcp.delayed_ack=0
net.inet.tcp.restrict_rst=1
kern.ipc.maxsockbuf=2097152
kern.ipc.shmmax=268435456

# Host cache
net.inet.tcp.hostcache.hashsize=4096
net.inet.tcp.hostcache.cachelimit=131072
net.inet.tcp.hostcache.bucketlimit=120

# Increase number of ports
net.inet.ip.portrange.first=2000
net.inet.ip.portrange.last=100000
net.inet.ip.portrange.hifirst=2000
net.inet.ip.portrange.hilast=100000
kern.ipc.semvmx=131068

# Disable Ping-flood attacks
net.inet.tcp.msl=2000
net.inet.icmp.bmcastecho=1
net.inet.icmp.icmplim=1
net.inet.tcp.blackhole=2
net.inet.udp.blackhole=1
```

监控

```
./configure --with-http_ssl_module --with-http_stub_status_module --  
without-mail_pop3_module  
--without-mail_imap_module --without-mail_smtp_module  
make install BATCH=yes
```