nginx与node.js结合使用

Node.js是一个基于Chrome JavaScript运行时建立的平台， 用于方便地搭建响应速度快、易于扩展的网络应用。Node.js 使用事件驱动， 非阻塞I/O 模型而得以轻量和高效，非常适合在分布式设备上运行的数据密集型的实时应用，如实时聊天等等。然而对于gzip编码，静态文件，HTTP缓存，SSL处理，负载平衡和反向代理等，都可以通过nginx来完成，从而减小node.js的负载，并通过nginx强大的缓存来节省网站的流量从而提高网站的加载速度。

流程图

nginx配置如下：

```
http {
    proxy_cache_path  /var/cache/nginx levels=1:2 keys_zone=one:8m max_size=3000m inactive=600m;
    proxy_temp_path /var/tmp;
    include       mime.types;
    default_type  application/octet-stream;
    sendfile        on;
    keepalive_timeout  65;

    gzip on;
    gzip_comp_level 6;
    gzip_vary on;
    gzip_min_length  1000;
    gzip_proxied any;
    gzip_types text/plain text/html text/css application/json application/x-javascript text/xml application/xml application/xml+rss text/javascript;
    gzip_buffers 16 8k;

    ssl_certificate /some/location/sillyfacesociety.com.bundle.crt;
    ssl_certificate_key /some/location/sillyfacesociety.com.key;
    ssl_protocols        SSLv3 TLSv1;
    ssl_ciphers HIGH:!aNULL:!MD5;

    upstream silly_face_society_upstream {
      server 127.0.0.1:61337;
      server 127.0.0.1:61338;
      keepalive 64;
    }

    server {
      listen 80;
      listen 443 ssl;

      server_name sillyfacesociety.com;
      return 301 $scheme://www.sillyfacesociety.com$request_uri;
    }

    server {
        listen 80;
        listen 443 ssl;

        server_name www.sillyfacesociety.com;

        error_page 502  /errors/502.html;

        location ~ ^/(images/|img/|javascript/|js/|css/|stylesheets/|flash/|media/|static/|robots.txt|humans.txt|favicon.ico) {
          root /usr/local/silly_face_society/node/public;
          access_log off;
          expires max;
        }

        location /errors {
          internal;
          alias /usr/local/silly_face_society/node/public/errors;
        }

        location / {
          proxy_redirect off;
          proxy_set_header   X-Real-IP            $remote_addr;
          proxy_set_header   X-Forwarded-For  $proxy_add_x_forwarded_for;
          proxy_set_header   X-Forwarded-Proto $scheme;
          proxy_set_header   Host                   $http_host;
          proxy_set_header   X-NginX-Proxy    true;
          proxy_set_header   Connection "";
          proxy_http_version 1.1;
          proxy_cache one;
          proxy_cache_key sfs$request_uri$scheme;
          proxy_pass         http://silly_face_society_upstream;
        }
    }
}
```

配置段说明

```
http {
    ...
    upstream silly_face_society_upstream {
      server 127.0.0.1:61337;
      server 127.0.0.1:61338;
      keepalive 64;
    }
    ...
}
```

nginx负载均衡多个nodo.js实例。keepalive 64 指示nginx在任何时候保持最少64个HTTP/ 1.1连接到代理服务器。如果有更多的流量nginx将打开更多的连接。

```
http {
```

```
    ...
    server {
        ...
        location / {
          proxy_redirect off;
          proxy_set_header   X-Real-IP            $remote_addr;
          proxy_set_header   X-Forwarded-For  $proxy_add_x_forwarded_for;
          proxy_set_header   Host                   $http_host;
          proxy_set_header   X-NginX-Proxy    true;
          ...
          proxy_set_header   Connection "";
          proxy_http_version 1.1;
          proxy_pass         http://silly_face_society_upstream;
        }
        ...
    }
}
```

将符合哪些的请求发送到代理上。nginx的匹配规则可以取看看前面的文章。

nginx处理静态内容

```
http {
    ...
    server {
        ...
        location ~ ^/(images/|img/|javascript/|js/|css/|stylesheets/|flash/|media/|static/|robots.txt|humans.txt|favicon.ico) {
          root /usr/local/silly_face_society/node/public;
          access_log off;
          expires max;
        }
        ...
    }
}
```

设置缓存

```
http {
    ...
    proxy_cache_path  /var/cache/nginx levels=1:2 keys_zone=one:8m max_size=3000m inactive=600m;
    proxy_temp_path /var/tmp;
    ...
}
```

```
http {
  server {
     ...
     location / {
          ...
          proxy_cache one;
          proxy_cache_key sfs$request_uri$scheme;
          ...
     }
     ...
  }
}
```

缓存是通过HTTP头部来控制的。

参考：http://blog.argteam.com/coding/hardening-node-js-for-production-part-2-using-nginx-to-avoid-node-js-load/