

SAE Python的发展历程

Alan Chan

Email: chenfeng2@staff.sina.com.cn

目录

- SAE Python介绍和现状
- SAE Python的技术变迁

Why SAE ?

- PaaS平台
- 帮助开发者更加快速的开发应用
- 从部署中解放出来，让开发者专注于应用开发
- 更高的性能，更高的可靠性
- ...



示例

第一步:创建一个python应用

第二步:打开终端

```
$ mkdir helloworld
$ cd helloworld
$ vim config.yaml

    name: helloworld
    version: 1

$ vim index.wsgi

def application( environ, start_response ):
    status = '200 OK'
    headers = [('Content-type', 'text/plain')]
    start_response( status, headers )

    return ['hello, world!']

$ saecloud deploy
```

访问 <http://helloworld.sinaapp.com>

支持WSGI

- 大部分的框架可以直接使用



• • •

支持tornado



- 直接支持tornado的异步模式

```
config.yaml
-----

worker: tornado
```

- 所有对tornado的修改通过monkey-patch的方式完成
- still at bleeding edge, 目前仅测试过tornado-2.1.1

基础服务

- Fetchurl, Socket, Mail
- Cron, Taskqueue
- MySQL, Storage, KVDB
- Memcache

直接import即可使用

```
from sae.taskqueue import Task, TaskQueue
```

```
queue = TaskQueue('queue_name')  
queue.add(Task("/tasks/foo"))
```

依赖关系管理

- 内置了丰富的第三方库

```
config.yaml
-----

libraries:
- name: django
  version: "1.4"
```

- 用户上传
 - virtualenv + pip + bundle_local.py
 - 支持zip包上传
 - 目前不支持c扩展

其它

- 本地开发环境 `dev_server.py`

```
$ git clone https://github.com/SAEPython/saepythondevguide.git
```

- 在线调试环境 ShellMiddleware

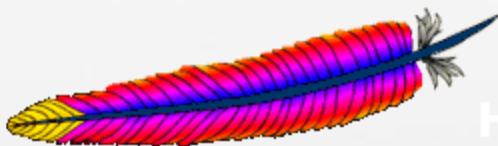
Now...

- 2000+ user
- 3000+ application
- 200+ active application
- 1000,000+ pv/day

Milestone

- 2011.08.09 第一个commit
- 2011.11.01 发布第一个内测版本
- 2012.04.29 开始重构server
- 2012.08.13 第二版server第一个稳定版本（lady）
上线
- 2012.08.17 tornado worker上线
- 2012.09.19 第二版server 1.0(greywind) 上线
- ...

The Dawn Age



Apache
HTTP SERVER PROJECT

+

MOD_WSGI

apache+mod_wsgi

优点:

- apache有比较成熟的技术经验
- 已开发的模块可以直接使用
- mod_wsgi相对比较成熟稳定
- mod_wsgi的代码比较规范，扩展修改容易

缺点:

- 从解释器隔离到进程隔离，httpd进程退化为完全的代理
- apache prefork的阻塞问题
- mod_wsgi, apache和sae的应用管理模型冲突，需要做非常多的修改。
- support wsgi only

A Clash of Kings

- 阻塞 => 非阻塞
- 选择什么python server
 - gunicorn ?
 - uwsgi ?
 - **OR ...**

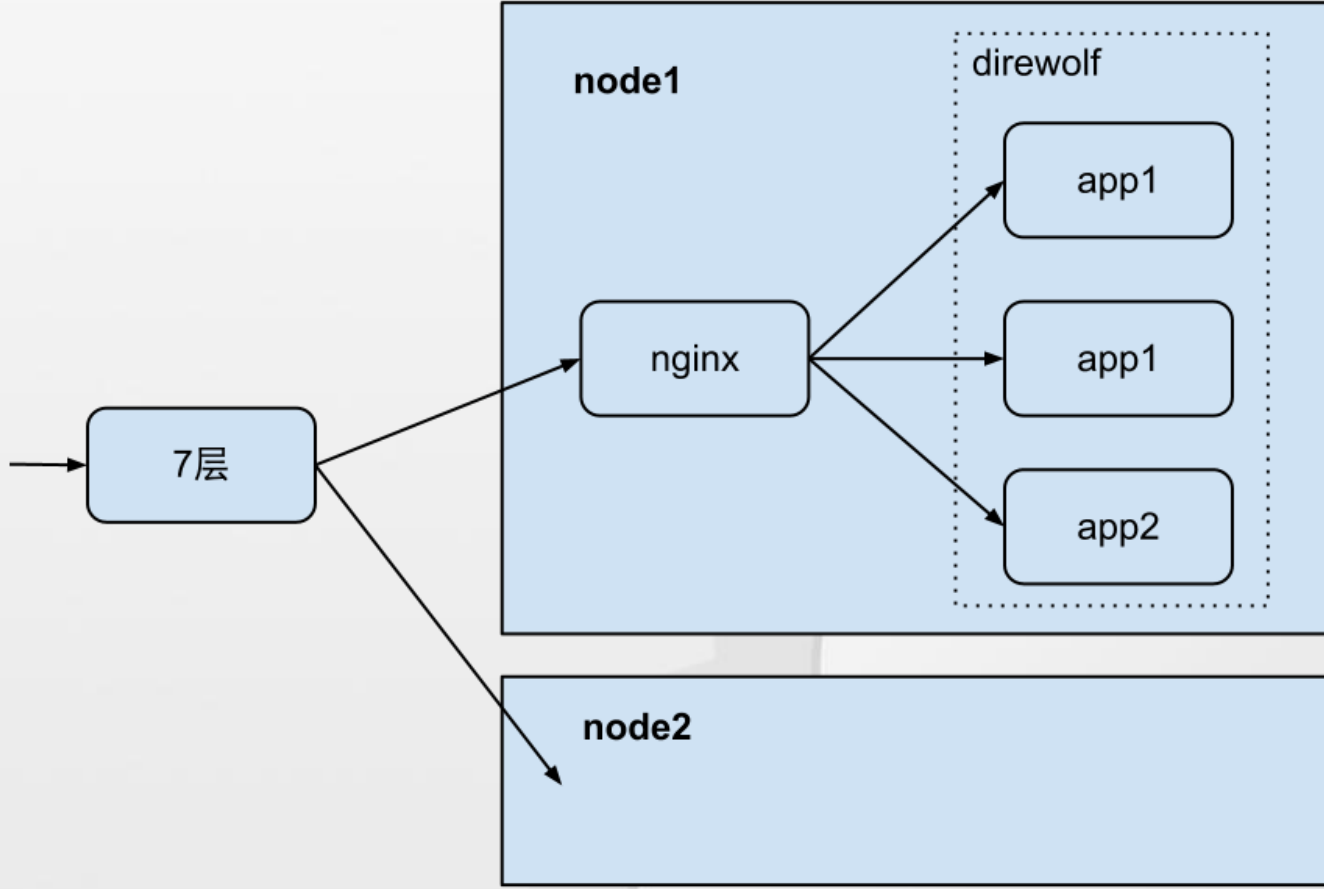
Age of The North

NGINX™

+



架构



静态文件处理

```
config.yaml
-----

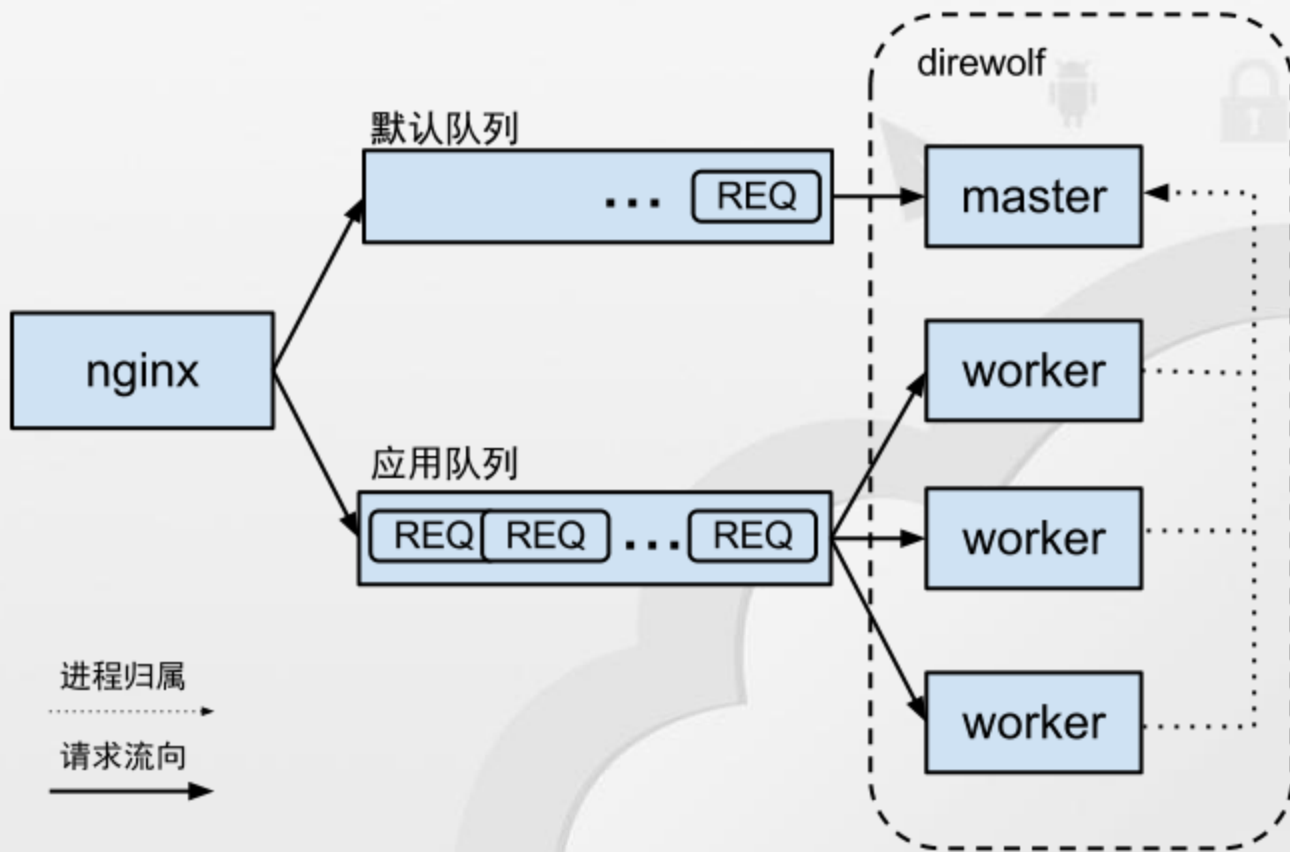
url: /static
static_dir: static
```

- nginx直接处理静态文件
- 同一个server上所有应用共享同一个nginx

动态内容处理

- nginx proxy_pass给python server
- 入口 index.wsgi:application
- 目前使用的python版本为2.7.3
- 应用和应用之间进程隔离，每个应用单独的instance
- 请求必须在300s内完成

动态内容处理



Auto-Scaling

- 伸：请求等待处理时间 & 负载

每个应用维护一个请求队列，如果队列中的请求在指定的时间内还没开始处理，则为该应用增加 instance。

- 缩：超过指定时间没有任何访问

direwolf

- 纯C, 使用Python C API
- lightweight, 3000+行代码
- KISS, 只负责解释python脚本
- 目前支持wsgi/tornado worker
- 为SAE Python的需求量身定制
- inspired by gunicorn & bjoern

资源限制

目标：

- 防止用户代码执行越权的操作
- 保护内部服务接口，保证所有内部请求合法

资源限制

- 文件系统，system-call：LD_PRELOAD
- cpu/memory: cgroups
- 网络
 - iptables
 - socket模块强制使用sockproxy服务
 - 内部REST服务接口，fetchurl模块
 - 底层socket可以直接访问内网



资源限制

- 禁用模块（ctypes, fcntl等），禁用函数（主要是posix模块的）。
- 禁用进程，线程（dummy thread）
- Python自带的c模块全部builtin
- 限制第三方的c模块的路径，该路径用户不可读（方便管理）

package management

- using standard python to install packages for the restricted sae python
- patch & extend distutils
- no setuptools, force to use distutils, if failed, fallback to patch setuptools
- add SAE.txt configuration file to preform sae customized action

```
$ export PYTHONHOME=/path/to/standard/python  
$ xpython setup.py install
```

Roadmap



The Next Big Issue

- alpha => beta

And

- gevent worker?
- resident worker?

...

About

- 文档 <http://appstack.sinaapp.com>
- github: <https://github.com/SAEPython/saepythondevguide>
- 邮件列表: <http://groups.google.com/group/sae-python>

Thank You!

<http://sae.sina.com.cn>