

Scalable Python Server with Coroutines

Damien Hou

<http://damienh.org>

PyCon 2011, Shanghai, China

Agenda

- Background
- Python Multi-threading
- Coroutine and Greenlet
- Gogreen

About Me

- I'm a long time Python coder
- Built indie social games using Python
- Worked at Slide, Inc as dev lead
- Currently a dev lead at Google

View this online

<http://dhou.github.com/pyconcn-coro>

Background

The Slide Chronicles

- 2005 - Founded by Max Levchin
 - Started providing slide show widgets

The Slide Chronicles

- 2006 - First SNS viral widget on MySpace
- 2007 - First developers on Facebook platform
 - Top Friends
 - Fun Space
 - SuperPoke!
 - SuperPoke! Pets
 - SPP Ranch

The Slide Chronicles

- 2010 - Acquired by Google
 - Disco Messenger
 - Prizes.org
 - Photovine

Python at Slide

We breathe Python

- Python is the language of choice
- Started with `mod_python` on Apache
- Moved on to in-house `coro-httpd` stack

Challenges of SNS apps and games

- Highly viral
 - hundreds of thousands of new users a day
- High concurrency
 - more than 117 million unique monthly viewers
 - millions of daily active users

State of Python servers ca. 2005

- Not many choices
- Apache + mod_python
- Twisted

Evolution of Slide's Python server

- Started with `mod_python` on Apache
- Migrated to `corohttplib` based pure Python server stack

Python Multi-threading

Multi-threading in Python

- Python threads are real OS threads
- Managed by the OS
- the GIL

Stackless Python

- Forked interpreter to remove restrictions
- Comes with thread scheduler
- Custom Executable

Coroutine and Greenlet

Coroutine

- Micro-thread
- Suspend and resume execution
- Lightweight, runs on single OS thread

Greenlet

- Spin-off of Stackless
- C extension module on the default Python interpreter
- A greenlet is a coroutine
- No scheduler

Greenlet Example

```
from greenlet import greenlet
```

```
def test1():  
    print 12  
    gr2.switch()  
    print 34
```

```
def test2():  
    print 56  
    gr1.switch()  
    print 78
```

```
gr1 = greenlet(test1)  
gr2 = greenlet(test2)  
gr1.switch()
```

gogreen

gogreen

- Coroutine utilities package based on Greenlet
- Coroutine MySQL module
- Thread scheduler and others
- Includes pure Python web server corohttpd

corohttpd Example

```
server = HttpServer(  
    args = (('0.0.0.0', 7001), 'access.log'))  
file_handler = HttpFileHandler('/home/htdocs/')  
server.push_handler(file_handler)  
server.start()  
coro.event_loop(30.0)
```

How it works

- Everything is a `coro.Thread`
- `HttpServer` spawns `HttpProtocol` to handle requests
- `HttpProtocol` generates `HttpRequest` and sends to handler code
- Build higher level web framework on top

Why Coro

- Concurrency from a single thread
- Clean code

Caveats

- Custom coro based modules
- The code is the document

Get gogreen

<https://github.com/slideinc/gogreen>

Questions?

Find me at

- github.com/dhou
- twitter.com/houyr
- weibo.com/dhouyr
- damienh.org