



# PRINT、PDB之外的调试思路

SYS.SETTRACE的妙用

XIADONG ZHU

2018.06.30

# 传统调试的姿势

PRINT

PDB

# 排障中遇到的问题

无法直接  
访问生产  
环境

生产环境  
不能安装  
新软件

不少问题  
偶发、需  
持续追踪

复杂的代  
码结构

# 我们需要什么样的工具

尽量少的  
依赖

确定排障  
思路，可  
持续跟踪

切入运行  
时环境

# SYS.SETTRACE介绍

```
import sys

def foo():
    bar()

def bar():
    print('Hello from bar()')
    print('Please notice the output from different tracers.')
    print('Say goodbye~')

def global_tracer(frame, event, args):
    print('# From the **GLOBAL** tracer:', {
        'event': event,
        'args': args,
        'context': frame.f_code.co_name
    })
    return local_tracer

def local_tracer(frame, event, args):
    print('# From the **LOCAL** tracer:', {
        'event': event,
        'args': args,
        'context': frame.f_code.co_name
    })

if __name__ == '__main__':
    sys.settrace(global_tracer)
    foo()
```



```
% python hello-settrace.py
('# From the **GLOBAL** tracer:', {'args': None, 'event': 'call', 'context': 'foo'})
('# From the **LOCAL** tracer:', {'args': None, 'event': 'line', 'context': 'foo'})
('# From the **GLOBAL** tracer:', {'args': None, 'event': 'call', 'context': 'bar'})
('# From the **LOCAL** tracer:', {'args': None, 'event': 'line', 'context': 'bar'})
Hello from bar()
('# From the **LOCAL** tracer:', {'args': None, 'event': 'line', 'context': 'bar'})
Please notice the output from different tracers.
('# From the **LOCAL** tracer:', {'args': None, 'event': 'line', 'context': 'bar'})
Say goodbye
('# From the **LOCAL** tracer:', {'args': None, 'event': 'return', 'context': 'bar'})
('# From the **LOCAL** tracer:', {'args': None, 'event': 'return', 'context': 'foo'})
```

# 事件枚举

事件	处理器	栈帧位置	参数	返回值
call	全局Tracer	函数定义所在行	None	局部Tracer   None
line	局部Tracer	执行当前行	None	局部Tracer   None
return	局部Tracer	返回语句所在行	返回值   None	Ignored
exception	局部Tracer	异常冒泡所在行	(exc, val, tb)	局部Tracer   None

## 案例：跟踪执行路径

# 快速实现

```
def tracer(frame, event, args):  
    print({  
        'event': event,  
        'filename': frame.f_code.co_filename,  
        'lineno': frame.f_lineno,  
        'context': frame.f_code.co_name  
    })  
    return tracer
```

```
if __name__ == '__main__':  
    sys.settrace(tracer)  
    requests.get('http://cn.bing.com')
```



```
{'lineno': 61, 'event': 'call', 'context': 'get', 'filename': 'site-packages/requests/api.py'}  
{'lineno': 71, 'event': 'line', 'context': 'get', 'filename': 'site-packages/requests/api.py'}  
{'lineno': 72, 'event': 'line', 'context': 'get', 'filename': 'site-packages/requests/api.py'}  
{'lineno': 16, 'event': 'call', 'context': 'request', 'filename': 'site-packages/requests/api.py'}  
{'lineno': 57, 'event': 'line', 'context': 'request', 'filename': 'site-packages/requests/api.py'}  
{'lineno': 344, 'event': 'call', 'context': '__init__', 'filename': 'site-packages/requests/sessions.py'}  
{'lineno': 349, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/requests/sessions.py'}  
{'lineno': 801, 'event': 'call', 'context': 'default_headers', 'filename': 'site-packages/requests/utils.py'}  
{'lineno': 805, 'event': 'line', 'context': 'default_headers', 'filename': 'site-packages/requests/utils.py'}  
{'lineno': 806, 'event': 'line', 'context': 'default_headers', 'filename': 'site-packages/requests/utils.py'}  
{'lineno': 792, 'event': 'call', 'context': 'default_user_agent', 'filename': 'site-packages/requests/utils.py'}  
{'lineno': 798, 'event': 'line', 'context': 'default_user_agent', 'filename': 'site-packages/requests/utils.py'}  
{'lineno': 798, 'event': 'return', 'context': 'default_user_agent', 'filename': 'site-packages/requests/utils.py'}  
{'lineno': 807, 'event': 'line', 'context': 'default_headers', 'filename': 'site-packages/requests/utils.py'}  
{'lineno': 808, 'event': 'line', 'context': 'default_headers', 'filename': 'site-packages/requests/utils.py'}  
{'lineno': 809, 'event': 'line', 'context': 'default_headers', 'filename': 'site-packages/requests/utils.py'}  
{'lineno': 40, 'event': 'call', 'context': '__init__', 'filename': 'site-packages/requests/structures.py'}  
{'lineno': 41, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/requests/structures.py'}  
{'lineno': 28, 'event': 'call', 'context': '__init__', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 34, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 36, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 37, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 37, 'event': 'exception', 'context': '__init__', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 38, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 39, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 40, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 41, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 42, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 142, 'event': 'call', 'context': 'update', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 151, 'event': 'line', 'context': 'update', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 154, 'event': 'line', 'context': 'update', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 156, 'event': 'line', 'context': 'update', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 158, 'event': 'line', 'context': 'update', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 159, 'event': 'line', 'context': 'update', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 161, 'event': 'line', 'context': 'update', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 164, 'event': 'line', 'context': 'update', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 168, 'event': 'line', 'context': 'update', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 170, 'event': 'line', 'context': 'update', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 170, 'event': 'return', 'context': 'update', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 42, 'event': 'return', 'context': '__init__', 'filename': 'site-packages/urllib3/packages/ordered_dict.py'}  
{'lineno': 42, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/requests/structures.py'}  
{'lineno': 44, 'event': 'line', 'context': '__init__', 'filename': 'site-packages/requests/structures.py'}
```



# 过滤无关信息

```
filter = re.compile(r'site-packages/requests/.*')
```

```
def tracer(frame, event, args):
    if filter.search(frame.f_code.co_filename):
        print({
            'event': event,
            'filename': frame.f_code.co_filename,
            'lineno': frame.f_lineno,
            'context': frame.f_code.co_name
        })
    return tracer
```

```
if __name__ == '__main__':
    sys.settrace(tracer)
    requests.get('http://cn.bing.com')
```

[illegible]

# 友善提醒

- 本操作会增加CPU开销，有观察的数据会增加5%~12%
- 本操作会降低Python解释器的执行效率
- 本操作会给磁盘IO增加压力
- 请一定记得使用RotatingFileHandler
- 请在运维爸爸的指导下进行操作

## 案例：找到沉默异常

The background is a dark blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines connecting to small circles.

“有什么比抛出异常更可怕的事情？”

“把异常吃掉，什么日志输出都没有，告诉你I'M FINE。”

# SALTSTACK的例子

```
def connect_pull(self, timeout=1):
    """
    Establish a connection with the event pull socket
    Default timeout is 1 s
    """

    if self.cpush:
        return True

    if self._run_io_loop_sync:
        with salt.utils.async.current_ioloop(self.io_loop):
            if self.pusher is None:
                self.pusher = salt.transport.ipc.IPCMessageClient(
                    self.pulluri,
                    io_loop=self.io_loop
                )
            try:
                self.io_loop.run_sync(
                    lambda: self.pusher.connect(timeout=timeout))
                self.cpush = True
            except Exception:
                pass
    else:
        if self.pusher is None:
            self.pusher = salt.transport.ipc.IPCMessageClient(
                self.pulluri,
                io_loop=self.io_loop
            )
        # For the async case, the connect will be deferred to when
        # fire_event() is invoked.
        self.cpush = True
    return self.cpush
```

- 心跳数据通过IPC经由MASTER进程上报
- WINDOWS下通过本地TCP套接字通信
- 默认端口被占用
- CONNECT报连接异常
- 异常被捕获后直接pass，没有信息输出
- 该函数返回False，表示连接失败了.....

# 实现思路

```
def suppress_exception():  
    try:  
        1 / 0  
    except Exception:  
        pass
```

- “沉默异常”定义：异常被try...except捕获后，直接pass，结束except块
- 异常发生后，局部Tracer会被“exception”事件激活，此时栈帧处于raise异常的位置
- 下一次，局部Tracer会被一个“line”事件激活，此时栈帧处于except关键字的位置
- 如果是沉默异常的情形，那么接下去，局部Tracer会再被一个“line”事件激活，并且这个“line”事件对应栈帧的当前行，是“pass”
- 那么，只要识别出这是个空语句行，就可以识别出沉默异常

# 如何识别这个空语句行

```
def suppress_exception():  
    try:  
        1 / 0  
    except Exception:  
        pass ←
```

- 我们注意到，一旦执行完空语句行，`except`块随即关闭
- 通过`frame.f_lasti`可以拿到最近一次试图执行的指令的偏移量
- 通过`frame.f_code[frame.f_lasti]`可以拿到最近一次试图执行的指令
- 如果该指令表示结束当前语句块，即锁定当前行为空语句行（`JUMP_FORWARD`）



# 检验样例

```
def suppress_exception():
    try:
        1 / 0
    except Exception:
        pass

def raise_exception():
    try:
        1 / 0
    except Exception as exc:
        print('Oops!', exc)
        raise

def global_tracer(frame, event, args):
    if event == 'exception':
        return silent_exception_capturer
    return global_tracer

def silent_exception_capturer(*_):
    def capturer(frame, event, args):
        if ord(frame.f_code.co_code[frame.f_lasti]) == dis.opmap['JUMP_FORWARD']:
            print('Silent Exception!', frame.f_code.co_filename, frame.f_lineno)
            return global_tracer
    return capturer

if __name__ == '__main__':
    sys.settrace(global_tracer)
    suppress_exception()
    raise_exception()
```



```
% python capture-silent-exception.py
('Silent Exception!', 'capture-silent-exception.py', 9)
('Oops!', ZeroDivisionError('integer division or modulo by zero',))
Traceback (most recent call last):
  File "capture-silent-exception.py", line 37, in <module>
    raise_exception()
  File "capture-silent-exception.py", line 14, in raise_exception
    1 / 0
ZeroDivisionError: integer division or modulo by zero
```



# 通过三方库验证

```
filter = re.compile(r'site-packages/requests/.*')

def global_tracer(frame, event, args):
    if event == 'exception':
        return silent_exception_capturer
    return global_tracer


def silent_exception_capturer(*_):
    def capturer(frame, event, args):
        if (ord(frame.f_code.co_code[frame.f_lasti]) == dis.opmap['JUMP_FORWARD']
            and filter.search(frame.f_code.co_filename)):
            print('Silent Exception!', frame.f_code.co_filename, frame.f_lineno)
        return global_tracer
    return capturer

if __name__ == '__main__':
    sys.settrace(global_tracer)
    requests.post('http://cn.bing.com')
```



```
('Silent Exception!',
'/home/indexmotion/.
local/share/virtualenvs/settra
ce-tutorial-LaKmh1TI/local/lib
/python2.
7/site-packages/requests/sessi
ons.py', 182)
```

# 确认现场

 sessions.py ✕

```
175         prepared_request.headers.pop(header, None)
176         prepared_request.body = None
177
178     headers = prepared_request.headers
179     try:
180         del headers['Cookie']
181     except KeyError:
182         pass
183
184     # Extract any cookies sent on the response to the cookiejar
185     # in the new request. Because we've mutated our copied prepared
186     # request, use the old one that we haven't yet touched.
187     extract_cookies_to_jar(prepared_request._cookies, req, resp.raw)
188     merge_cookies(prepared_request._cookies, self.cookies)
189     prepared_request.prepare_cookies(prepared_request._cookies)
```

The background is a dark blue gradient. In the corners, there are decorative white line art elements resembling circuit boards or neural network connections. These include straight lines, right-angle turns, and small circles at the ends of the lines.

挑战：记录变量变化路径

# 感知上下文中名字赋值的变化

```
1: def fn():  
2:     a = {}  
3:     a['hello'] = 5  
4:     a['world'] = 6  
5:     a = [1, 2]  
6:     a.append(3)
```



```
a@<filename>:2 {}  
a@<filename>:3 {} -> {'hello': 5}  
a@<filename>:4 {} -> {'hello': 5,  
    'world': 6}  
a@<filename>:5 {'hello': 5,  
    'world': 6} -> [1, 2]  
a@<filename>:6 [1, 2] -> [1, 2, 3]
```

# 拓展

- 通过[sys.settracer](#)章节了解settrace的细节
- 通过[inspect](#)章节了解frame对象的细节
- 通过[sys.setprofile](#)了解如何自定义profiler

谢谢聆听  
Q&A