A Straightforward Introduction to Twisted

Peter Stephan
peter.stephan@teamaol.com

Agenda

- What is Twisted and why use it?
- Twisted abstractions
- Simple server
- Simple client
- Deferreds
- Threads and Processes
- Database access
- InlineCallbacks
- Logging
- Testing
- Debugging

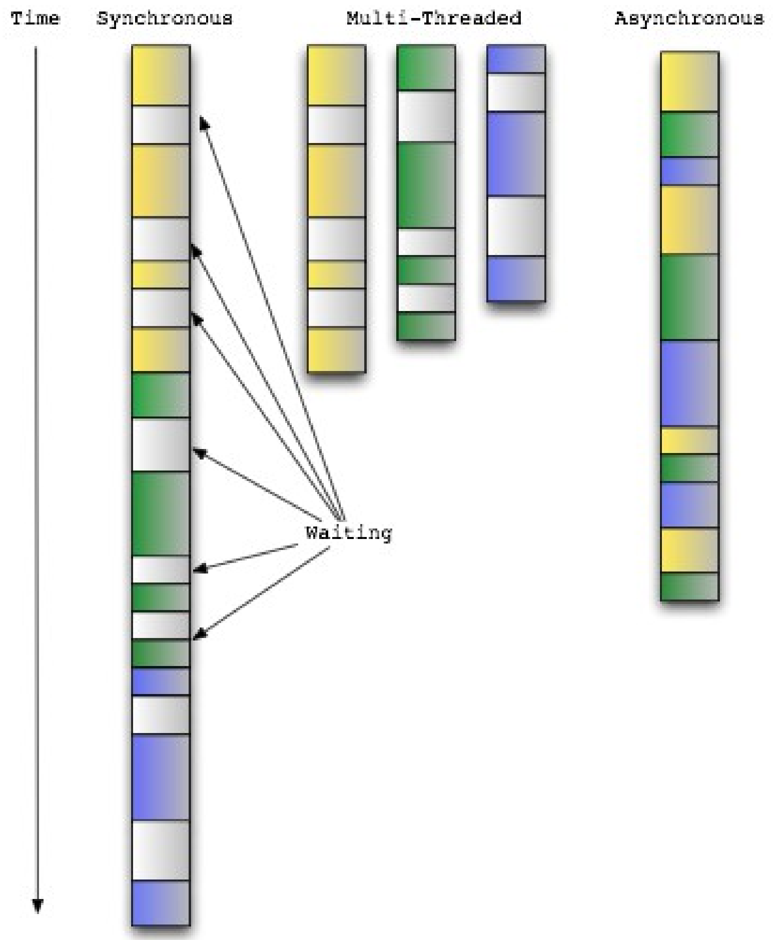
What is Twisted?

- Asynchronous, event-driven programming framework
- Single threaded*
- Non-blocking
- Program flow based on events
- But it isn't magic
- Similar frameworks
  - Perl POE
  - Ruby EventMachine
  - Node.js

Why Use Twisted?

- Faster than synchronous application
- Simpler than threaded or multiprocess application
- Lots of protocols and features
- Actively developed, maintained, supported
- But requires change in programming paradigm

Synchronous vs. Threaded vs. Asynchronous Processing



Reactor

- Main event loop of a Twisted application
- Controls the execution of code associated with events
- Relies on programmer to write non-blocking code (don't sleep!)
- Application runs in a single thread (usually)
- Only one reactor per application (Singleton)

Reactor

- Reactor is started explicitly
- Reactor can be stopped
- Cannot restart reactor once stopped
- Types of reactors
  - select()
  - poll()
  - epoll() - linux default
  - kqueue() - freebsd

First Twisted Program

- Minimal Twisted Application
  - Import libraries
  - Start the reactor

- Hello, World
  - First callback
  - More graceful termination
  - Traceback

Transport

- Object that sends/receives data
  - TCP, UDP, Pipe, SSL, etc
- Implements
  - write(), loseConnection(), getPeer(), getHost(), etc.
- Rarely have to create a new Transport class

Protocol

- One instance per connection
- Implements the communication protocol
  - HTTP, FTP, AMP, DNS, telnet, NNTP, IMAP, etc.
- Implements
  - dataReceived(), connectionLost,() makeConnection(), connectionMade(), etc.

ProtocolFactory

- Creates protocol objects
- Only need one ProtocolFactory per protocol
- Implements
  - buildProtocol(), doStart(), doStop(), etc.

Simple Server

- echo_server
- Transport is TCP
- EchoFactory subclasses ServerFactory
- EchoProtocol subclasses Protocol

Simple Client

- New protocol – SimpleDataProtocol
  - connect to server
  - collect data until connection is closed
- New factory - SimpleDataClientFactory
  - subclasses ClientFactory

Simple Client – Take 2

- Simple Client not very twistedy
- Factory should not stop the reactor
- Transport error looks like success
- Should return data to caller
- Should use callbacks and errbacks

Deferreds

- Callbacks/Errbacks are key to asynchronous processing in Twisted
- Encapsulated in an object called a Deferred (deferred result)
- Can add a chain of callbacks and errbacks to a deferred

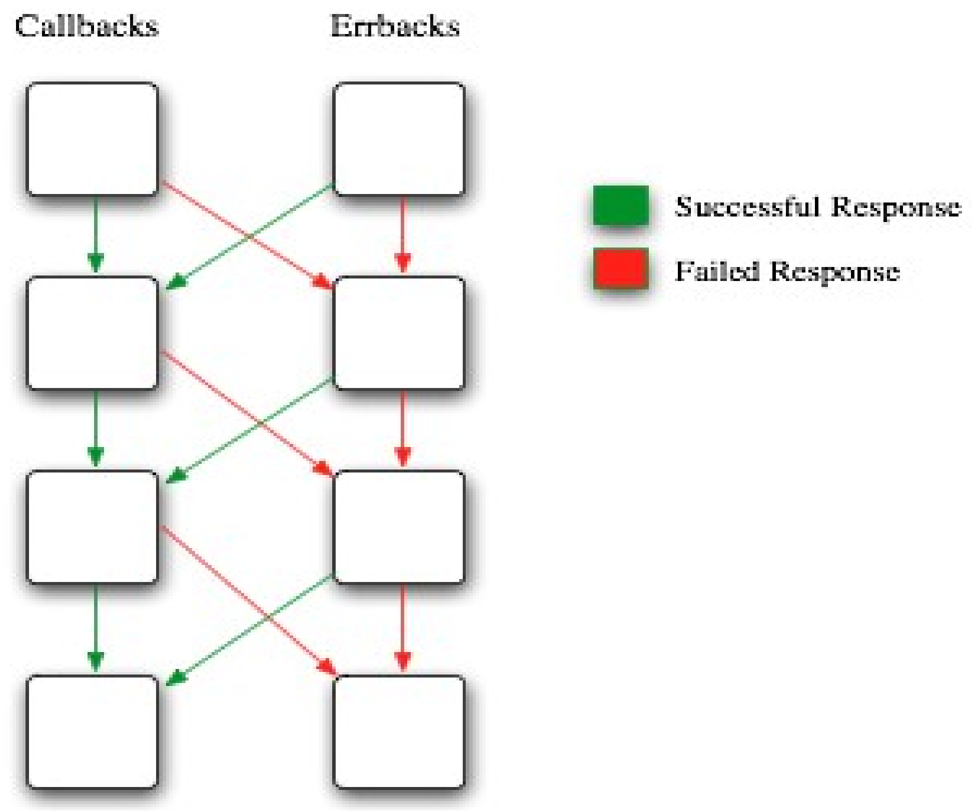
Simple Client – Take 3

- Factory gets a deferred
- Deferred has two levels of callbacks

More Deferreds

- addCallback()
- addErrback()
- addCallbacks()
- addBoth()

Callback/Errback Chain



Deferred Callback/Errback Chain

DeferredList

- Bundle several deferreds into one group
- Get called back when all complete
- Last version of simple client

Threads in Twisted

- reactor.callInThread()
- threads.callMultipleInThread()
- threads.deferToThread()
- threads.callFromThread()

Spawning Processes

- reactor.spawnProcess()
- util.getProcessOutput()

Databases and Twisted

- Most database calls are blocking
- adbapi module
- Runs blocking DB calls in separate threads

inLineCallbacks

- Write deferreds that look like sequential code
- Uses python generators
- 'yield' when a deferred is returned
- Method is called back when deferred completes with result of deferred

Logging in Twisted

- from twisted.python import log
- log.startLogging()
  - Stdout
  - File – rotation, include stdout
- log.msg
- log.err

Twisted.trial

- Testing framework for twisted apps
- Based on unittest
- setUp(), tearDown()
- trial runs reactor
- Each asynchronous test returns a deferred
- trial twisted – runs all tests of twisted

Debugging

- twistd -b – runs app in python debugger
- manhole – telnet/ssh into application, read/change variables, call methods
- pdb
- logging

Resources

- Developer Guides
  - http://twistedmatrix.com/documents/current/core/howto/
- Examples
  - http://twistedmatrix.com/documents/current/core/examples/
- API Documentation
  - http://twistedmatrix.com/documents/current/api/
- O'Reilly Book
  - http://www.amazon.com/Twisted-Network-Programming-Essentials-McKellar/dp/1449326110

More Resources

- Documentation in the source code
- Twisted-Python mailing list and IRC
  - https://twistedmatrix.com/trac/wiki/TwistedCommunity
- Online tutorials
  - http://www.wallix.org/2011/08/30/getting-started-with-twisted/
  - http://krondo.com/?page_id=1327