Lessons Learned Building a Scalable Distributed Storage System in Python

SciPy 2011

Chuck Thier
What is Swift?

- Distributed object storage
- Rackspace CloudFiles
- OpenStack Object Storage
- NASA
- Cloudscaling
- Others?
The Challenge

- 100 petabytes of storage
- 100 billion objects
- 100 gigabit/sec throughput
- 100 thousand requests per second
Simple Successes

- Python!
- Development and Operations together
- Small group that owns the project
- KISS
  - < 10K lines of Python code
  - > 18K lines of tests
  - Build on the shoulders of giants
- Leverage the economies of scale
Network I/O

- Saturating GigE is easy, 10 GigE needs a lot of CPU
- SSL on 10GigE (throughput not transactions)
- Load balancing
- Python threading is problematic
- Twisted doesn't fit my brain
- Eventlet to the rescue
Disk I/O

- RAID 5/6 performance
  - Random I/O is worst case scenario
  - Several week rebuild times on failure
  - 48TB file systems are problematic
- Filesystems
  - fsync changes the game a bit
  - Performance degradation over time
  - Directory listings
- Async file I/O is a pipe dream
- Chunking file operations is "good enough"
"At Scale, Everything Breaks"

- 100 PB = ~180K 2TB Hard drives = ~7500 Storage Nodes
- Multiple failures daily even at much smaller scales
- Handling failure is priority #1
Handling Failure

- Eliminate single points of failure
- Fail fast
  - Timeouts
  - 100-continue
- Fail gracefully
  - Handoff
  - Error limiting
- Recover from failure
  - Replication
Eventlet Timeout Example

```python
try:
    with ConnectionTimeout(self.app.conn_timeout):
        # Try to make a connection

    with Timeout(self.app.node_timeout):
        # Do stuff

eexcept (Exception, Timeout):
    # Log error
```
Questions?

- Project: [https://launchpad.net/swift](https://launchpad.net/swift)
- Docs: [http://swift.openstack.org/](http://swift.openstack.org/)
- Code: bzr branch lp:swift
- IRC: #openstack on freenode

- Me: Chuck Thier (@creiht) <chuck.thier@rackspace.com>