PyCloudPath

Cory Cotter
Mentors: Frank Lind, Ryan Volz, Victor Pankratius
Background

- Multiprocessing
- Cloud environment

http://images.anandtech.com/reviews/cpu/intel/SNBE/Core_I7_LGA_2011_Die.jpg
Pipeline Concepts

- **Worker**
- **Stage**
- **Pipeline**

Diagram:
- Task → Task → Result
- Head → Stage → Result

Functions:
- \( f() \)
- \( g() \)
PyCloudPath

• Applications
  • Create cloud multiprocessing pipelines

• Features
  • Highly parallelized
  • Allocates virtual machines per pipeline
  • Multiple scaling options
  • Transports arbitrary python objects
  • Per-Stage event-driven control
  • Built-In logging
User Pipeline Creation

- Heads & Workers
  - doInit
  - doTask
  - doClose
  - handler

- INI Files
  - Stages
  - Pipeline Layout
  - GUI/Events
  - Buttons
  - String input
Worker Example

```python
import pyccloudpath

class AddWorker(pyccloudpath.OrderedWorker):
    '''Adds 10 to the data'''

    def doInit(self):
        '''Set on as default'''
        self.on = True

    def handler(self, message):
        '''Switch add worker if correct message'''
        if message['type'] == 'command':
            if message['name'] == 'add':
                if message['data'] == 'switch':
                    self.on = not self.on

    def doTask(self, self, data):
        '''Take input from previous stage and add 10'''
        if self.on:
            self.log('Adding 10 to data')
            data += 10
        return data
```
Spectral Monitoring

Comparisons
- 1 process on 8 core machine
  - 69s
- 8 processes using former method on 8 core machine
  - 46s
- Pipeline using four 8 core machines
  - 24s

Not optimized and still performs much better
Passive Radar Correlation

Not yet benchmarked
Future Development

- Dynamic process/machine allocation
- Additional fanout and visualization
- Update internal data transport
- More interface implementations
Summary

• PyCloudPath multiprocessing framework
• Enables cloud scale signal processing and workflows
• Event driven control and remote configuration
• Demonstrations of spectral monitoring and passive radar
Acknowledgments

- Frank Lind, Ryan Volz, and Victor Pankratius
- Bob Schaefer
- Program Organizers and Support Staff
- Chester Ruszczyk, Juha Vierinen, Chris Eckert
- All of Haystack