

Using SimPy to Model AWS Autoscaling for Realtime Computations



<http://danielwilliams.org>

Wireless Video Surveillance



CheckVideo Video Camera



Interact using web...

Dashboard

Welcome [Daniel Williams](#) today is: 2012-02-03 03:36:15 PM (US/Eastern) [Log Out](#)

Dashboard

System Map


PLAY

- [quattro](#)
- [1] Camera 1
- [2] Camera 2
- [3] Camera 3
- [4] Camera 4
- [solo3454](#)

quattro
Camera1

2012-02-02
12:37:56 PM

Person




2 2

LIVE

System Activity

Events Since
Last Log On: **15**

Total Amount of Storage Used:
 **13%**

Show system events

Last Ten Events

<input type="checkbox"/>	Device	Camera	Date	Event
<input type="checkbox"/>	quattro	Camera1	2012-02-02 12:38:09 PM	Person
<input type="checkbox"/>	quattro	Camera1	2012-02-02 12:37:54 PM	Person
<input type="checkbox"/>	quattro	Camera1	2012-02-02 11:02:25 AM	Person
<input type="checkbox"/>	quattro	Camera1	2012-02-02 11:01:48 AM	Person
<input type="checkbox"/>	quattro	Camera1	2012-01-31 04:33:06 PM	Person
<input type="checkbox"/>	quattro	Camera1	2012-01-31 03:29:31 PM	Person
<input type="checkbox"/>	quattro	Camera1	2012-01-31 03:06:31 PM	Person
<input type="checkbox"/>	quattro	Camera1	2012-01-31 12:57:13 PM	Person
<input type="checkbox"/>	quattro	Camera1	2012-01-31 12:23:47 PM	Person
<input type="checkbox"/>	quattro	Camera1	2012-01-29 09:46:24 AM	Person

Delete

Save

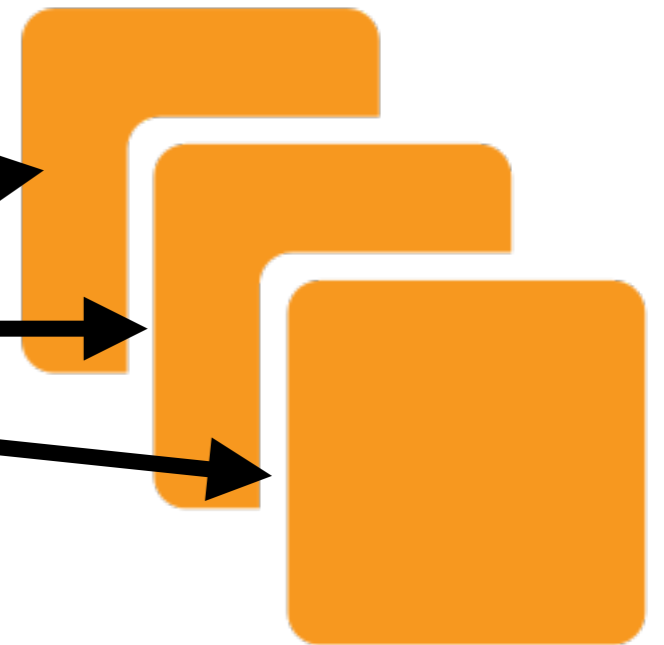
[View All Events](#)

...and mobile devices

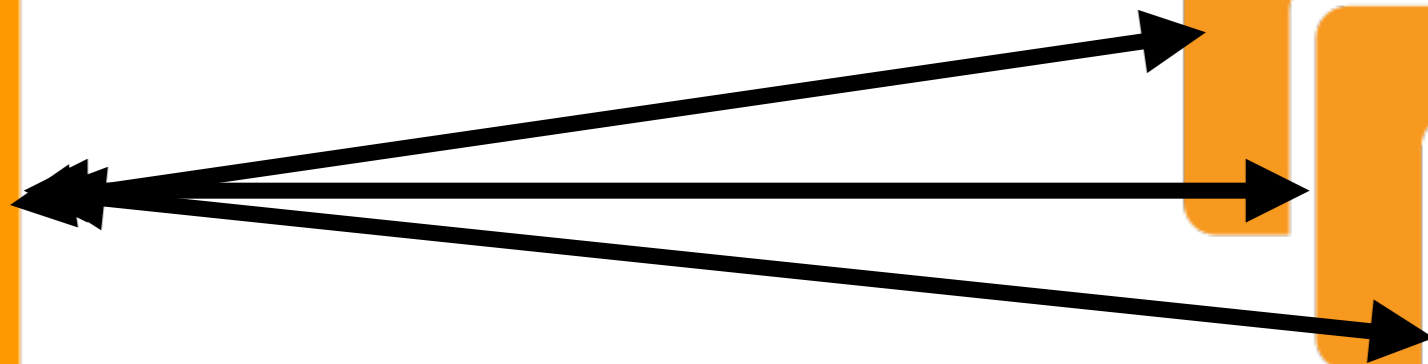




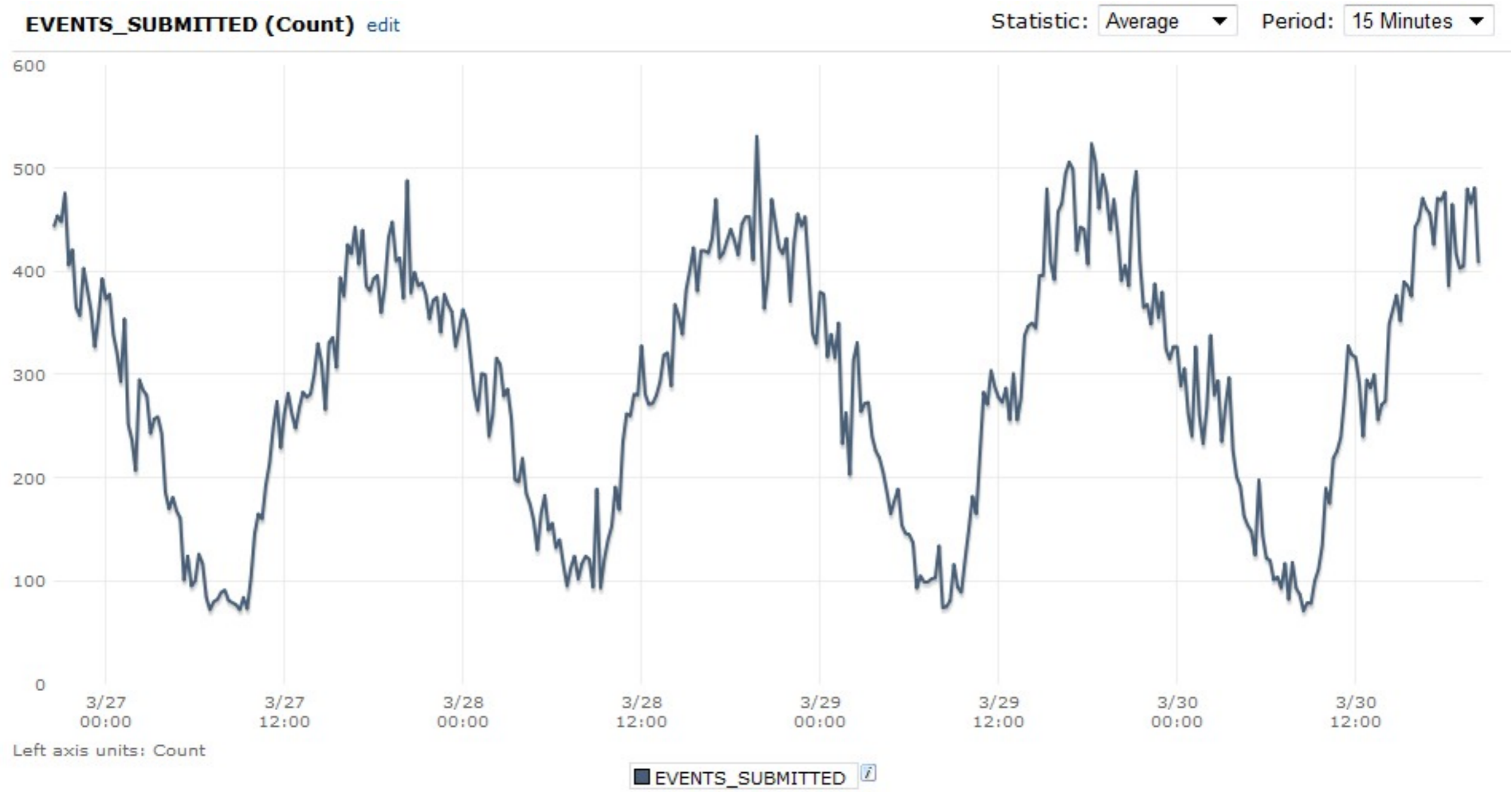
Requester

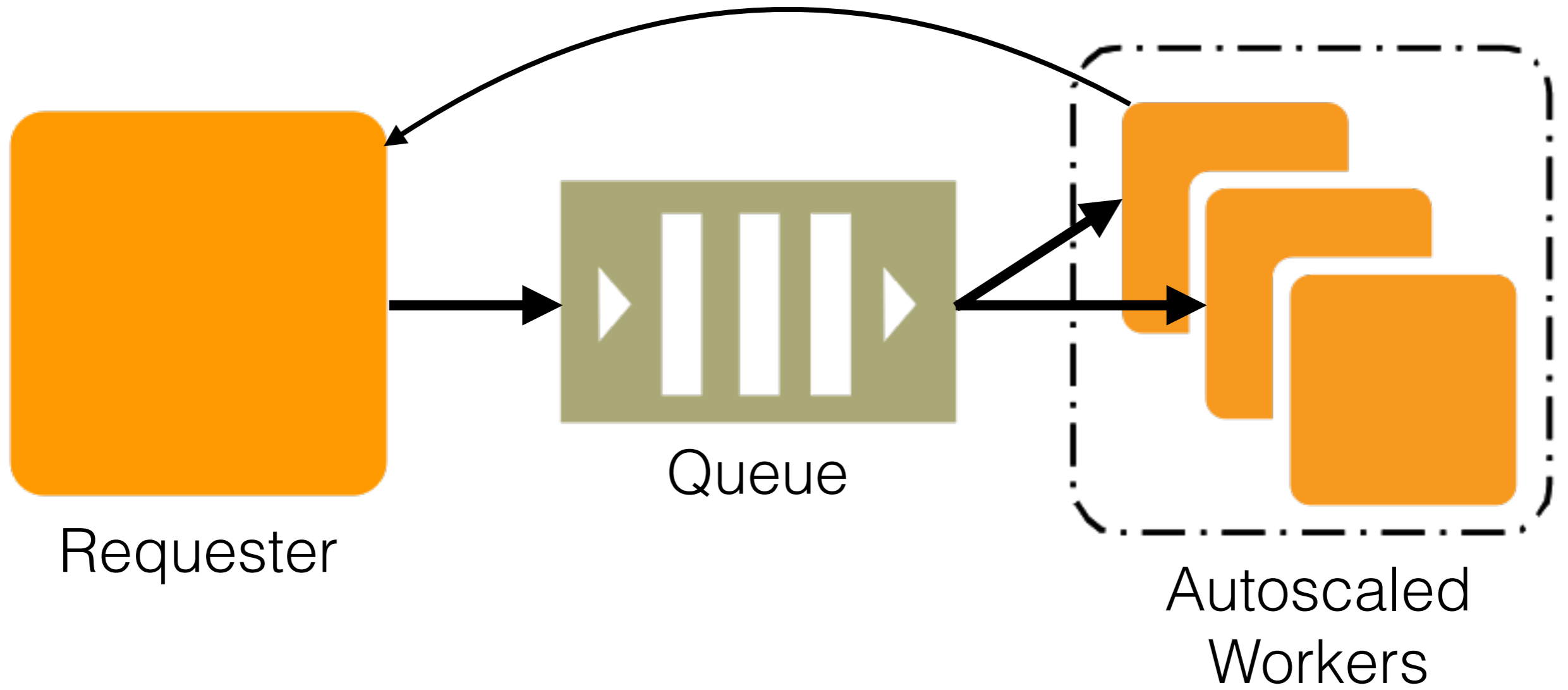


Workers



Our event load is cyclical







Use discrete event simulation
to model proposed solution



- uses 3 main object classes (Process, Resource, Monitor)
- uses Python generators as a sort of coroutine

The Simulation Classes

```
class Job(object):
```

```
    "A request for work to be performed by a  
    Server"
```

```
class Sources(Process):
```

```
    "Sources generate Jobs with Poisson arrivals"
```

```
class Server(Process):
```

```
    "Consume Jobs from the global queue"
```

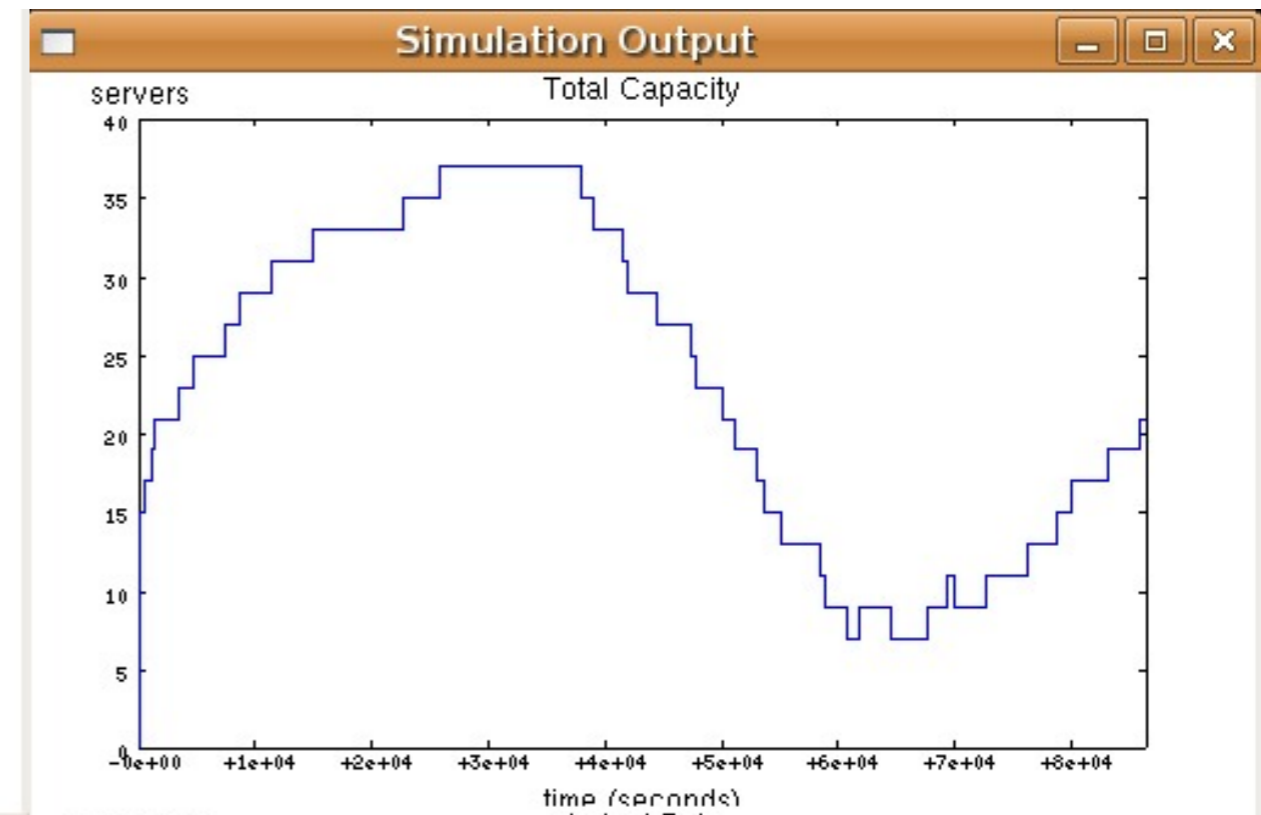
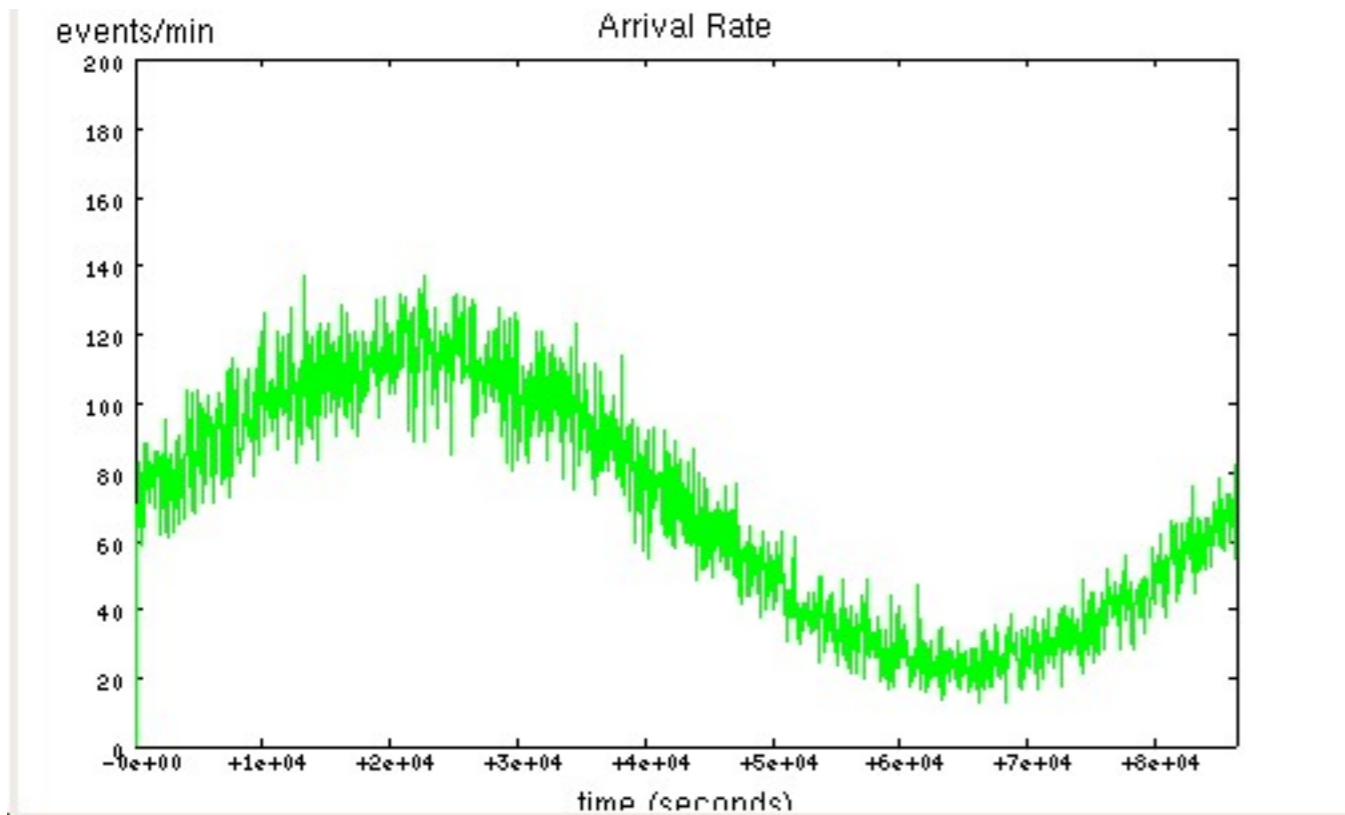
```
class Watcher(Process):
```

```
    "Implement Autoscaling, launching and  
    terminating Servers"
```

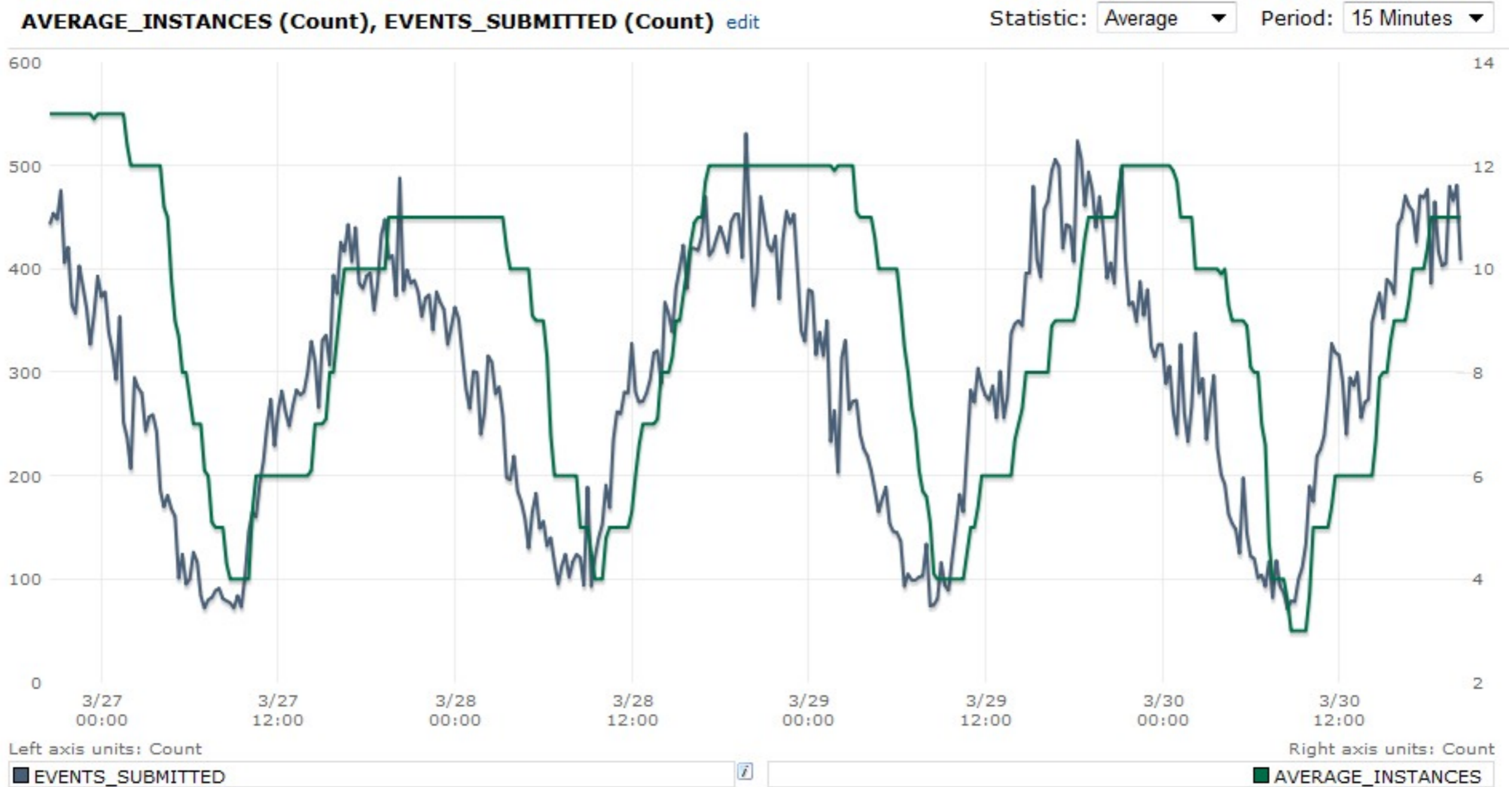
Simulation Run Results

```
$ ./verifyd-simulation.py \  
  --plot \  
  --encoders 5000 \  
  --capacity 15 \  
  --si_amplitude 0.65 \  
  --as_lower_threshold 40 \  
  --as_upper_threshold 60 \  
  --as_breach_duration 300 \  
  --as_upper_breach_scale_increment=2 \  
  --as_lower_breach_scale_increment=-2  
  
utilization                : 43.7%   (goal: > 50%)  
requests served in < 30s  : 95.7%   (goal: > 95%)  
requests timed out (180s):  0.11%  (goal: < 0.1%)
```

Simulated Autoscaling



Autoscaling works as simulated



Thanks for listening!

