

Monitor Your App's Behaviour

Milad Rastian (@slashmili)

- **IT team lead @ Sam Media.**
- **Python/Perl/PHP Software Developer**
- **System administrator**

Monitoring Your App

- When your app grows and you have many components in your system, you need to make sure it's working as what you expect.
- There is lot of elements in your app that can be fed to your business peeps.

Monitoring with SNMP

- Internet-standard protocol for managing and monitoring devices on IP networks
- Don't let the word "Simple" in the name fool you!
- Sends data over UDP protocol
- Open source applications like below items could read data and show you nice graphs
 - Zenoss
 - Nagios
 - Zabbix

How to use SNMP

- `apt-get install snmpd`

- `snmpwalk -v1 -cpublic localhost`

SNMPv2-MIB::sysDescr.0 = STRING: Linux vagrant-ubuntu-63 2.6.32-279.el6.
x86_64 #1 SMP Fri Jun 22 12:19:21 UTC 2012 x86_64

...

Behind the scenes

- In HTTP, you map the request based on request URL:
 - <http://localhost/home/user/profile>
- In SNMP, OID are the essential part of request mapping:
 - udp://localhost/1.3.6.1.4.868.2.4.1.2.1.1.1.3.3562.3

Behind the scenes

What hell is 1.3.6.1.4.868.2.4.1.2.1.1.1.3.3562.3 !?

- Iso(1)
 - org(3)
 - dod(6)
 - internet(1)
 - private(4)
 - transition(868)
 - products(2)
 - chassis(4)
 - ...

How to request specific OID

```
snmpget -v1 -cp public localhost .1.3.6.1.4.1.2022.1.1.4.1.84
```

Client -> GET\n

Client -> .1.3.6.1.4.1.2022.1.1.4.1.84\n

Server -> .1.3.6.1.4.1.2022.1.1.4.1.84\n

Server -> INTEGER\n

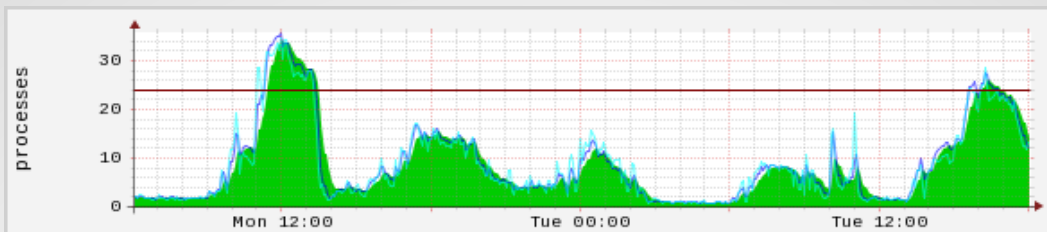
Server -> 205\n

How to write custom OID

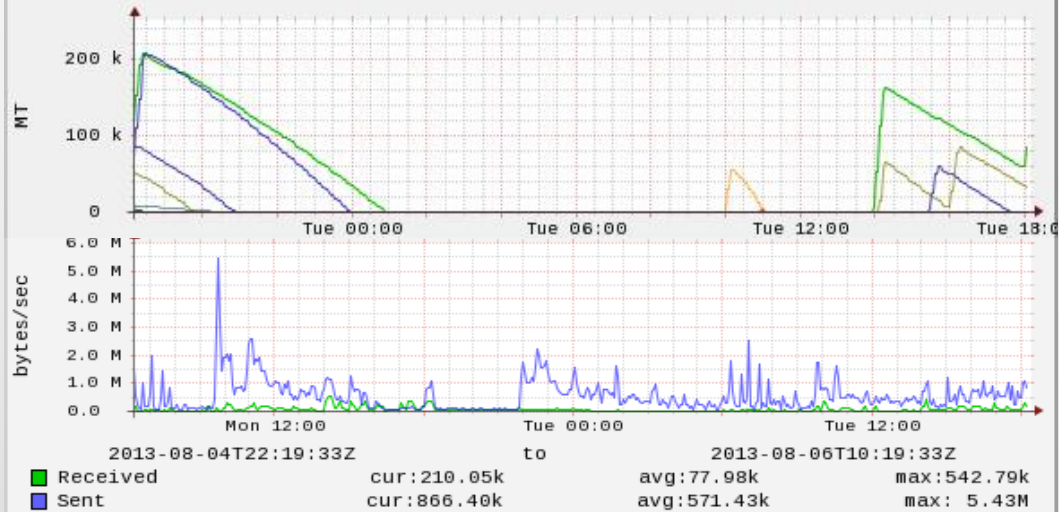
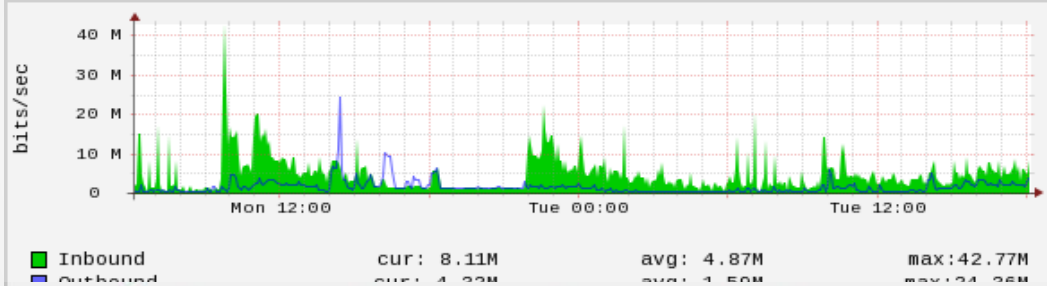
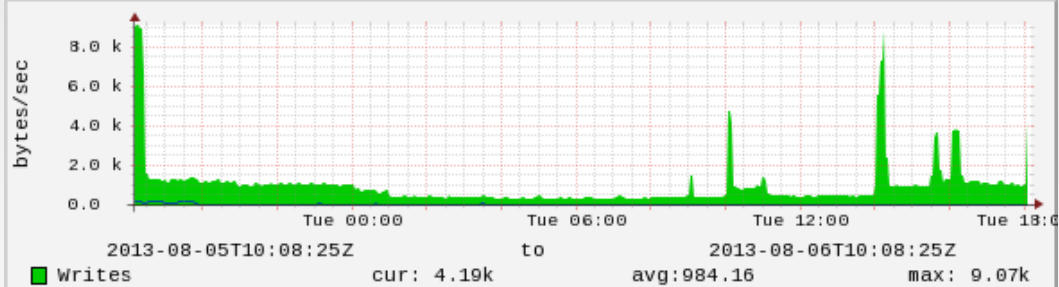
- From OID `.1.3.6.1.4.1.2022.*` is for private usage.
- Add this line to your snmp config(
`/etc/snmp/snmpd.conf`)
`pass_persist .1.3.6.1.4.1.2022.1 /usr/local/bin/my-app-snmp.rb`
- Handle SNMP commands that are passed to your app through standard Input and Answer them in STDOUT

Simple SNMP handler

```
STDIN.each do |str|
  str.chomp!.upcase!
  if str == 'PING'
    puts 'PONG'
  elsif str == 'GET'
    oid = STDIN.gets.chomp!
    puts oid
    puts 'STRING'
    puts 'foo'
  else
    puts 'NONE';
  end
end
```



■ high load



SNMP Data Types

- INTEGER
- IPADDRESS
- COUNTER
- GAUGE
- TIMETICKS
- ...

Monitoring with Statsd

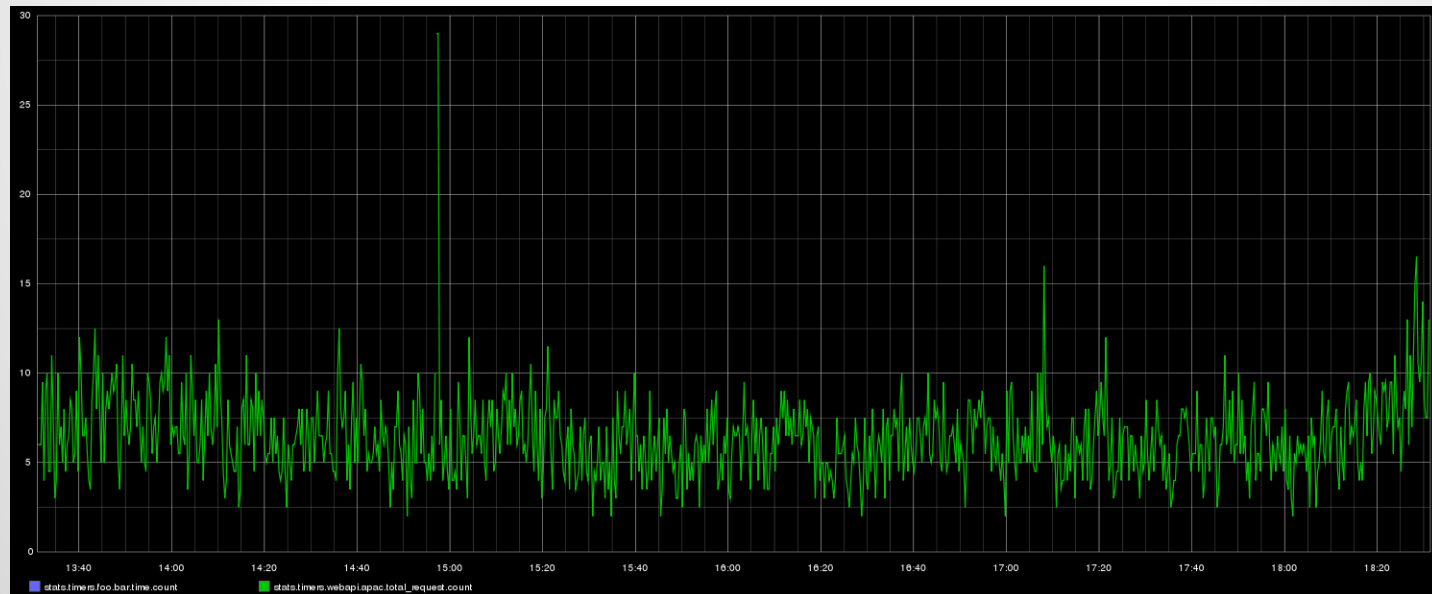
- Statsd is a daemon running on your server.
- listens for statistics, like counters and timers (sent through UDP)
- Sends aggregates to one or more pluggable backend services
- StatsD was inspired (heavily) by the project (of the same name) at Flickr.

Why should choose statsd over SNMP monitoring?

- It's much simpler!
- You want have more flexibility over your stats.

Ruby client

```
statsd = Statsd.new 'localhost', 1234  
statsd.timing 'webapi.apac', request_time
```



Statsd Data Types

- **Counters**
 - `statsd.increment 'http.request.visitor'`
 - `statsd.increment 'foo.bar', -1`
- **Timers**
 - `statsd.timing 'process.heavy.foo', 1000.3840923309`
- **Gauges**
 - `statsd.gauge 'process.waiting.foo', count_waiting_p`
- **Sets**
 - `statsd.sets 'users.register', userid`