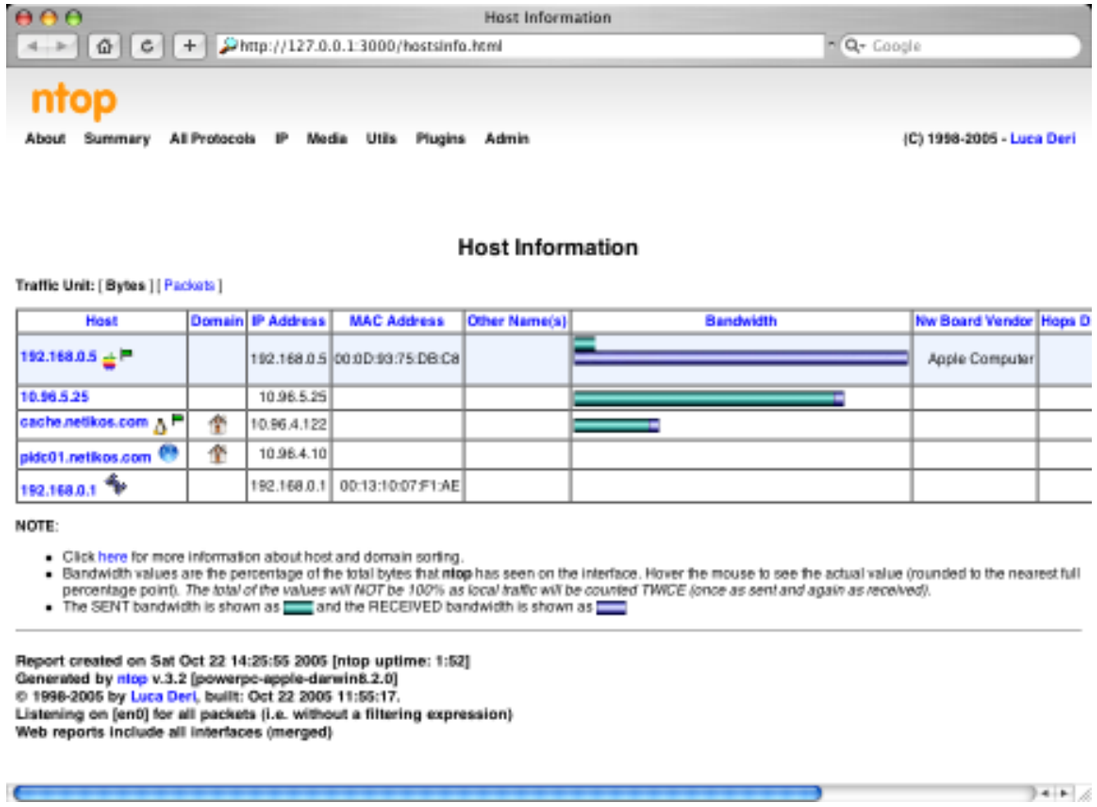
High-Speed Network Traffic Monitoring Using ntopng

Luca Deri
@lucaderi

LISA14
November 9–14, 2014

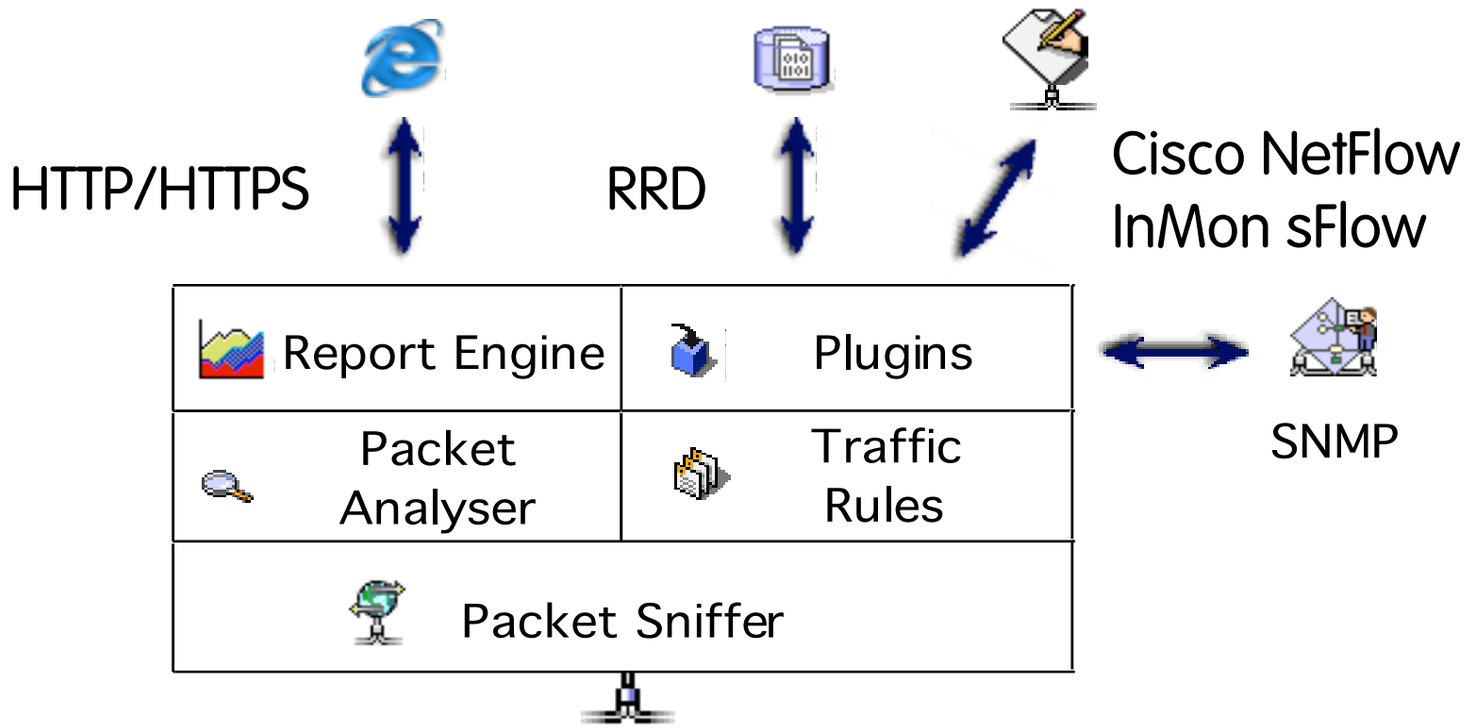
Some History

- In 1998, the original ntop has been created.
- It was a C-based app embedding a web server able to capture traffic and analyse it.



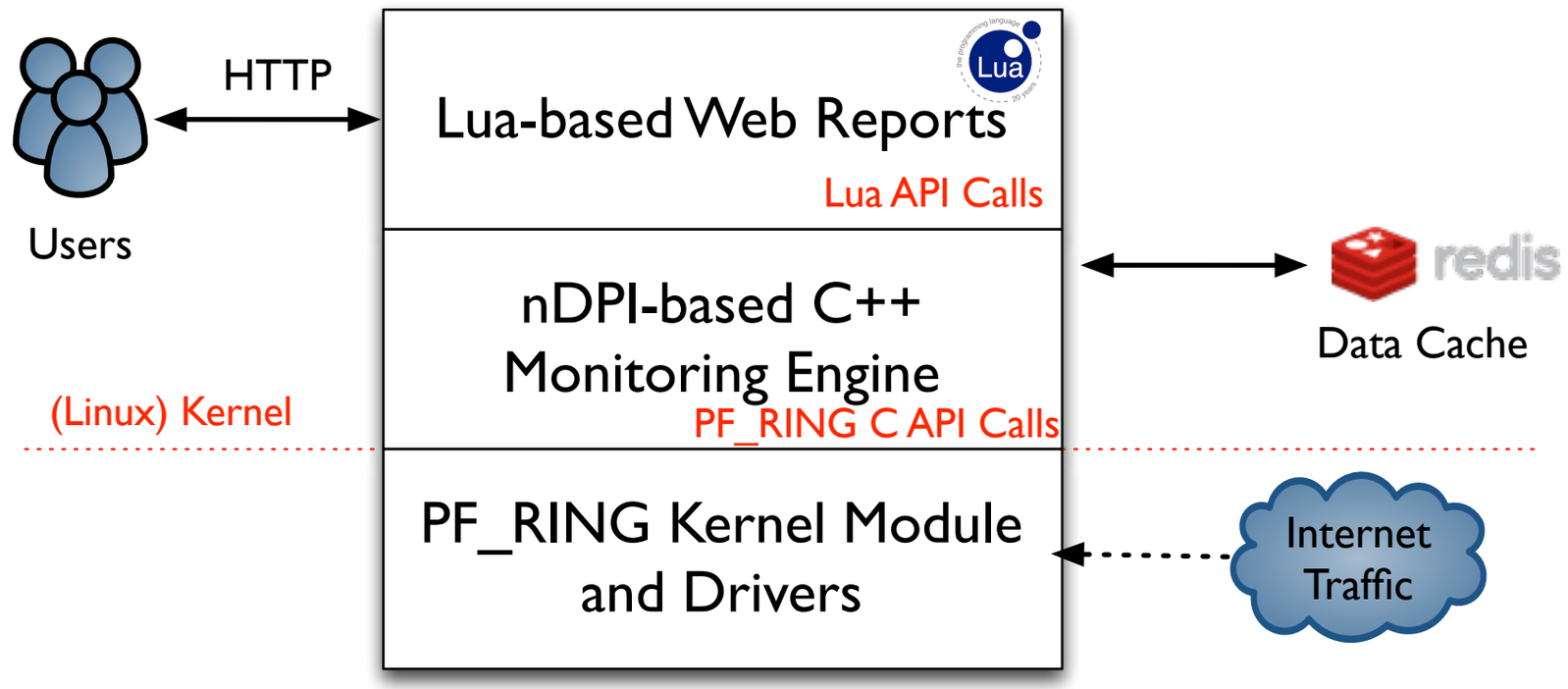
- Contrary to many tools available at that time, ntop used a web GUI to report traffic activities.
- It is available for Unix and Windows under GPL.

ntop Architecture



ntopng Architecture

- Three different and self-contained components, communicating with clean API calls.

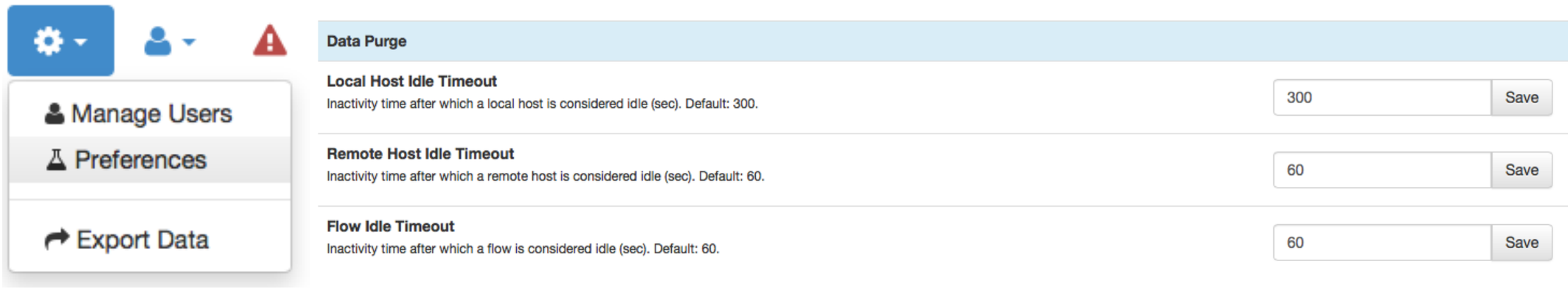


ntopng Monitoring Engine

- Coded in C++ and based the concept of flow (set of packets with the same 6-tuple).
- Flows are inspected with a home-grown DPI-library named nDPI aiming to discover the "real" application protocol (no ports are used).
- Information is clustered per:
  - (Capture) Network Device
  - Flow
  - Host
  - High-level Aggregations

Information Lifecycle

- ntopng keeps in memory live information such as flows and hosts statistics.
- As the memory cannot be infinite, periodically non-recent information is harvested.
- Users can specify preferences for data purge:



Packet Processing Journey

1. Packet capture: PF_RING (Linux) or libpcap.
2. Packet decoding: no IP traffic is accounted.
3. IPv4/v6 Traffic only:
   1. Map the packet to a 6-tuple flow and increment stats.
   2. Identify source/destination hosts and increment stats.
   3. Use nDPI to identify the flow application protocol
      1. UDP flows are identified in no more than 2 packets.
      2. TCP Flows can be identified in up to 15 packets in total, otherwise the flow is marked as "Unknown".
4. Move to the next packet.

The need for DPI in Monitoring

- Limit traffic analysis at packet header level it is no longer enough (nor cool).
- Network administrators want to know the real protocol without relying on the port being used.
- Selected protocols can be "precisely dissected" (e.g. HTTP) in order to extract information, but on the rest of the traffic it is necessary to tell network administrators what is the protocol flowing in their network.

Say hello to nDPI

- ntop has decided to develop its own GPL DPI toolkit in order to build an open DPI layer for ntop and third party applications.
- Supported protocols (> 180) include:
  - P2P (Skype, BitTorrent)
  - Messaging (Viber, Whatsapp, MSN, The Facebook)
  - Multimedia (YouTube, Last.gm, iTunes)
  - Conferencing (Webex, CitrixOnLine)
  - Streaming (Zattoo, Icecast, Shoutcast, Netflix)
  - Business (VNC, RDP, Citrix, *SQL)

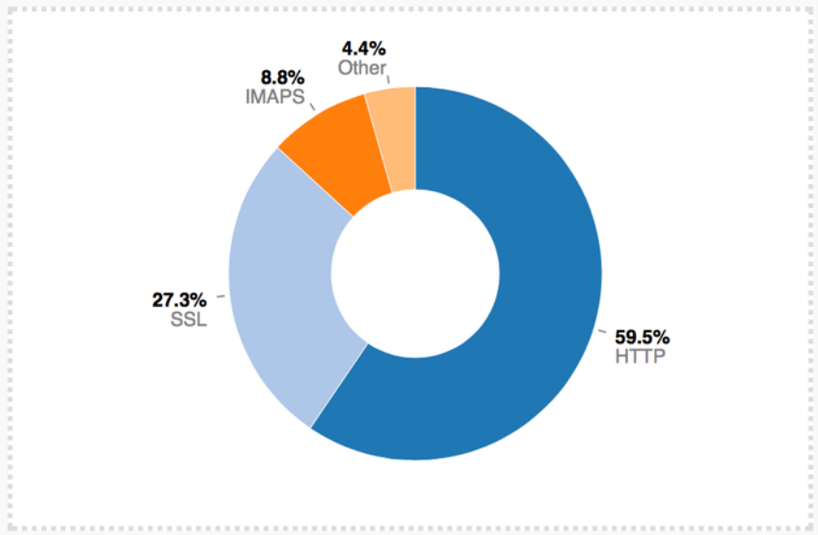
nDPI on ntopng

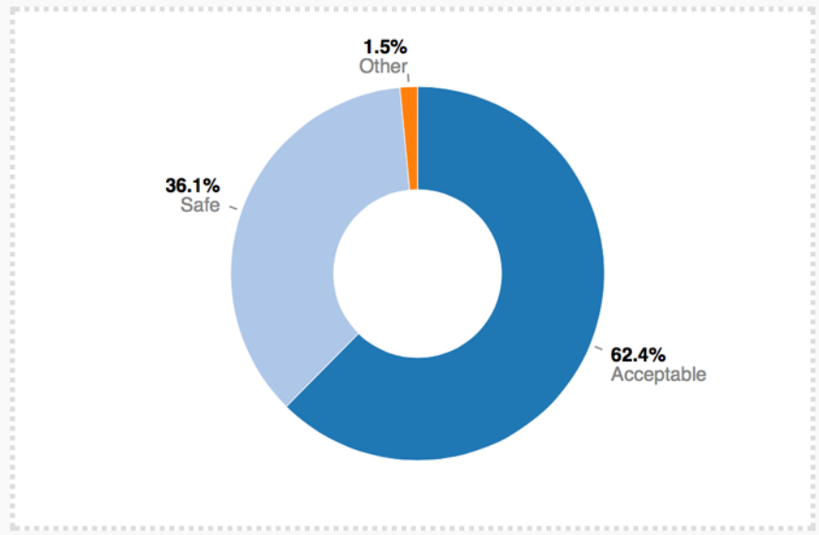
- In ntopng all flows are analysed through nDPI to associate an application protocol to them.
- L7 statistics are available per flow, host, and interface (from which monitoring data is received).
- For network interfaces and local hosts, nDPI statistics are saved persistently to disk (in RRD format).

nDPI on ntopng: Interface Report

Interface: en0 | Overview | Packets | Protocols | Historical Activity

Protocol Overview

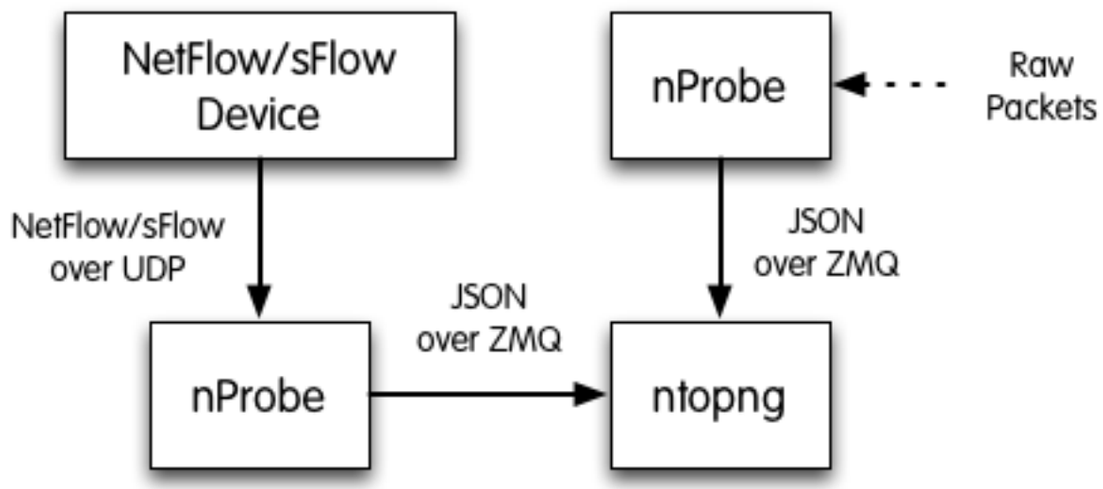




Application Protocol	Total (Since Startup)	Percentage
AppleiCloud	8.43 KB	0.33 %
DNS	4.19 KB	0.17 %
DropBox	6.15 KB	0.24 %
Google	9.04 KB	0.36 %
HTTP	1.43 MB	57.8 %
ICMP	280 Bytes	0.01 %
IMAPS	216.79 KB	8.56 %

ntopng as a NetFlow/sFlow Collector [1/2]

- The "old" ntop included a NetFlow/sFlow collector. Considered the effort required to support all the various NetFlow dialects (e.g. Cisco ASA flows are not "really" flows), in ntopng we have made a different design choice.



ntopng as a NetFlow/sFlow Collector [2/2]

Flows are sent in the following format

- {“8”:"192.12.193.11","12":"192.168.1.92","15":"0.0.0.0","10":0,"14":0,"2":5,"1":406,"22":1412183096,"21":1412183096,"7":3000,"11":55174,"6":27,"4":6,"5":0,"16":2597,"17":0,"9":0,"13":0,"42":4}
- Where:
  - “<Element ID>”: <value> (example 8 = IPV4_SRC_ADDR)

- nProbe has been integrated with sysdig.org to report network+system information, so we can have visibility of network activities carried on by system processes.

Flow/Process Drill-down [1/2]

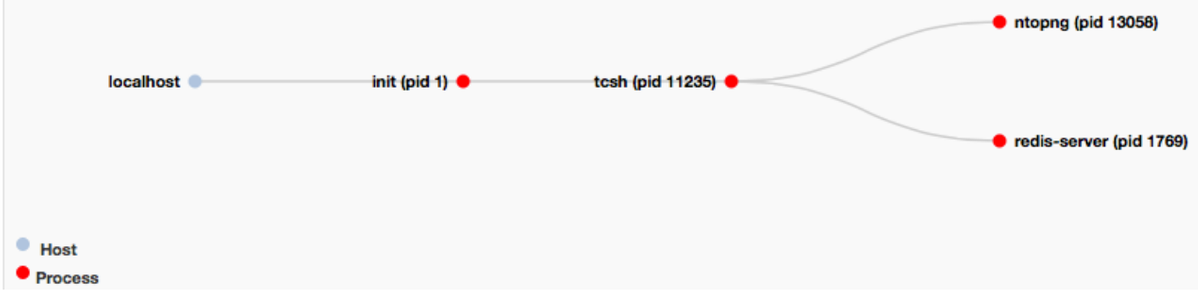
Active Flows

10 ▾ Applications▾

Info	Application	L4 Proto	Client Process	Client Peer	Server Process	Server Peer	Duration	Breakdown	Total Bytes
Info	SSH	TCP		dnsmon.nic.it :22		pc-deri.nic.it :46861	1 day, 6 h, 12 min, 6 sec	Client S	5.41 GB
Info	Redis	TCP	ntopng	localhost.localdomai... :53452	redis-server	localhost.localdomai... :6379	1 day, 6 h, 12 min, 5 sec	Client Server	3.8 GB

Flow: localhost.localdomain:53452 ⇄ localhost.localdomain:6379 | Overview

Flow Peers	localhost :53452 ⇄ localhost :6379	
Protocol	TCP / Redis	
First / Last Seen	30/09/2014 15:01:34 [1 day, 6 h, 13 min, 5 sec ago]	01/10/2014 21:14:33 [6 sec ago]
Total Traffic Volume	3.81 GB	
Client vs Server Traffic Breakdown	localhost.localdomain:53452 / localhost.localdomain:6379	
Client to Server Traffic	7,677,287 Pkts / 1.85 GB	
Server to Client Traffic	6,754,744 Pkts / 1.95 GB	
TCP Flags	SYN PUSH ACK This flow is active.	



Flow/Process Drill-down [2/2]

Client Process Information	
User Name	deri
Process PID/Name	13058/ntopng [son of 11235/tcsh]
Average CPU Load	0.71 %
I/O Wait Time Percentage	0 %
Memory Actual / Peak	1.4 MB / 1.46 MB [95.7%]
VM Page Faults	0

Flow-to-Process binding

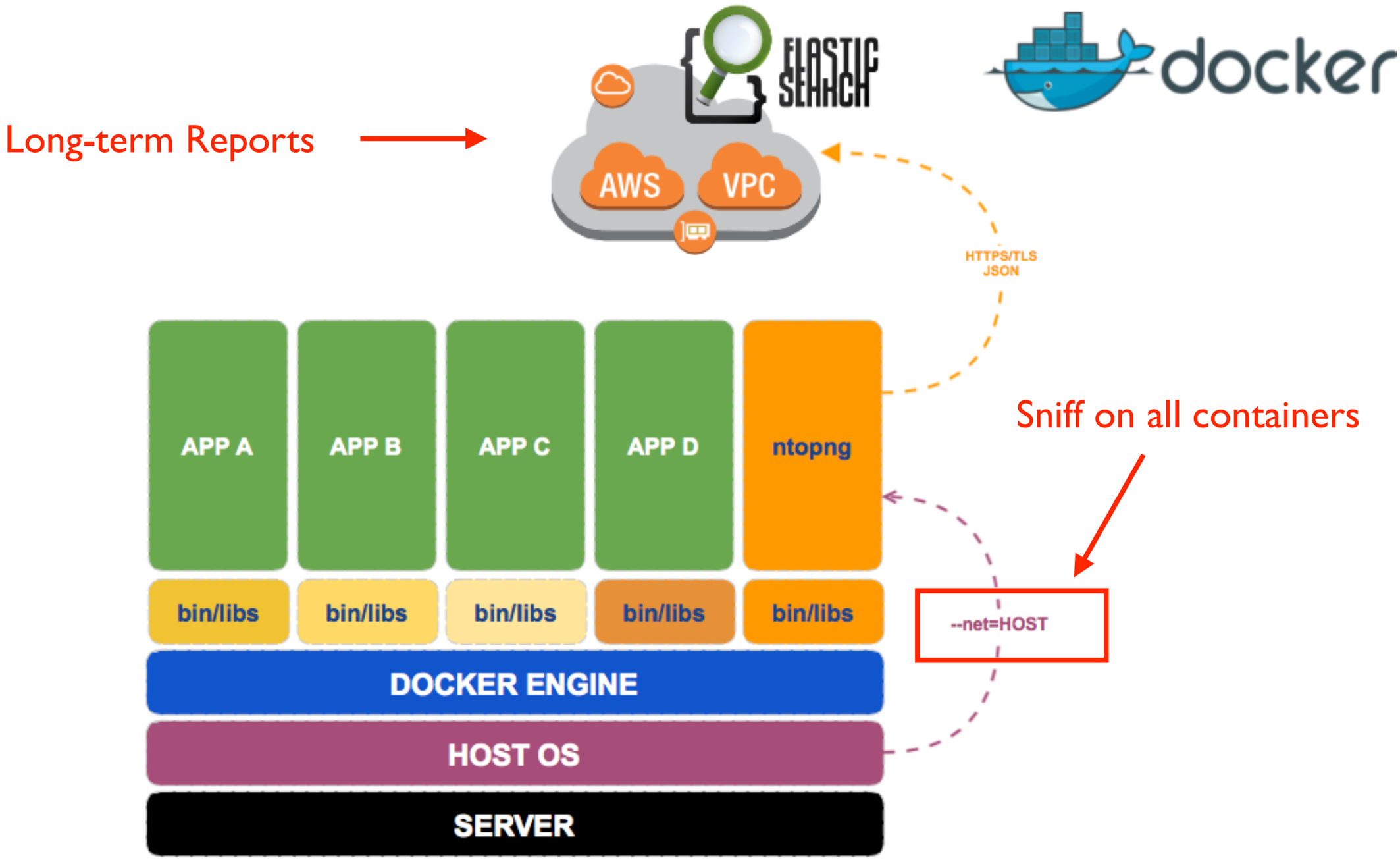
} Dynamically Updated

Server Process Information	
User Name	redis
Process PID/Name	1769/redis-server [son of 1/init]
Average CPU Load	0.12 %
I/O Wait Time Percentage	0 %
Memory Actual / Peak	344.13 KB / 344.13 KB [100%]
VM Page Faults	0

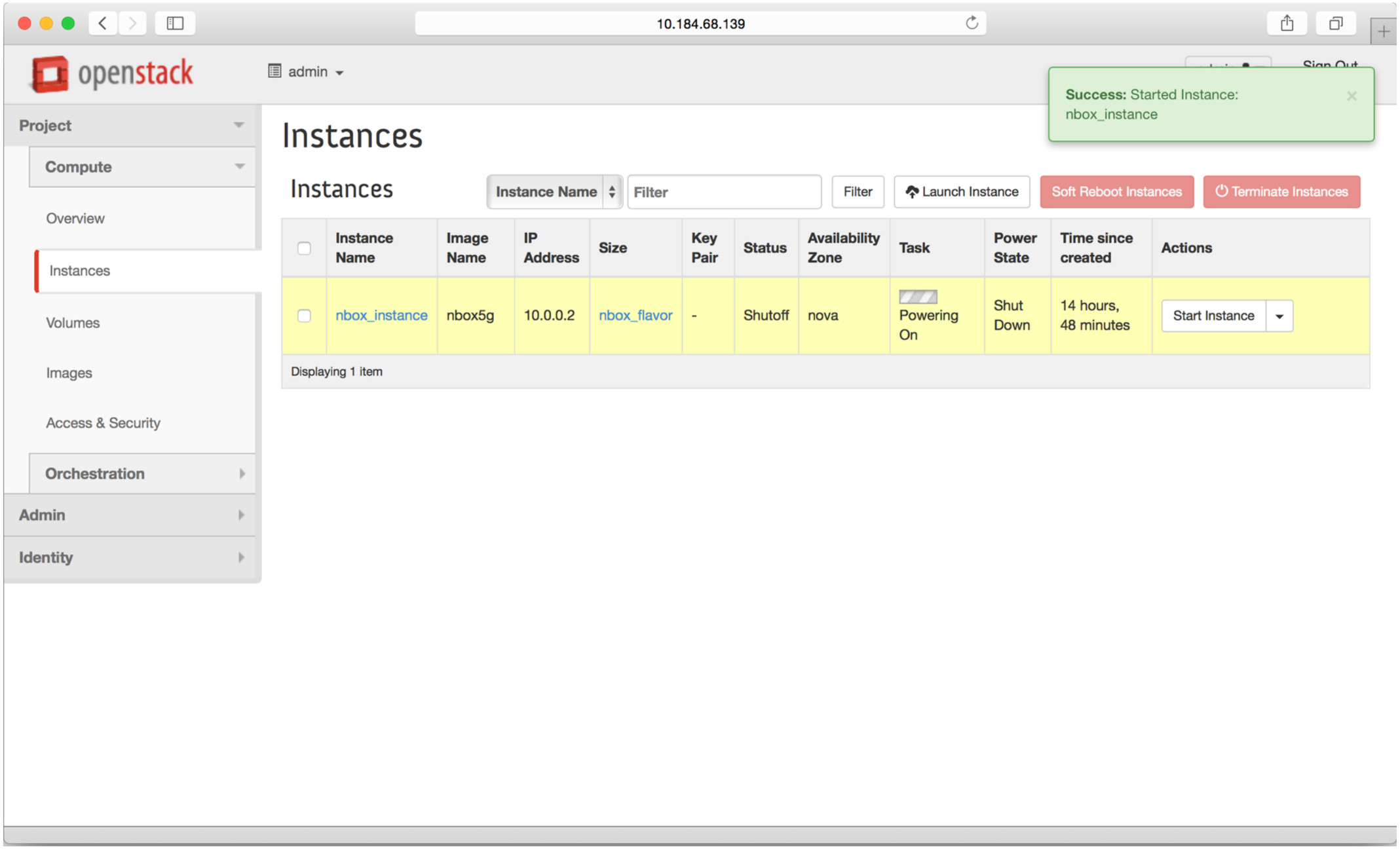
Flow-to-Process binding

} Dynamically Updated

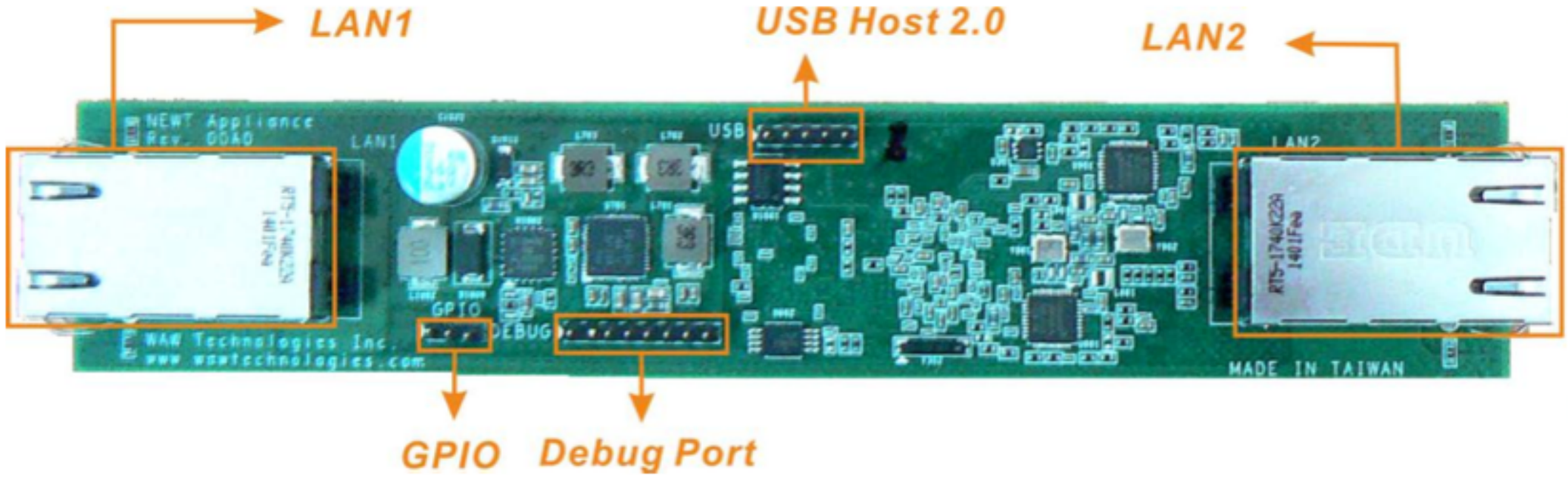
ntopng on Docker

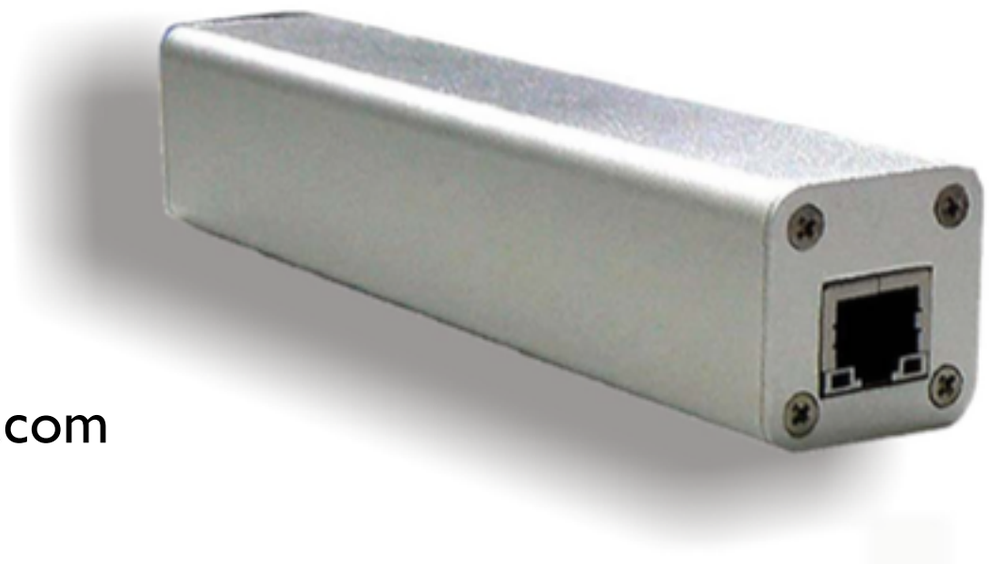


ntopng on OpenStack



Building a Cheap ntopng Probe





www.wawtechnologies.com

Soon a Kickstarter campaign will be launched targeting the creation of cheap monitoring devices

Final Remarks

- We believe that open-source traffic network monitoring should be simple and cheap.
- Commodity hardware, with adequate software, can now match the performance and flexibility that markets require. With the freedom of open source.
- ntopng is available under GNU GPLv3 from http://www.ntop.org/.