Microcloud-based Network Traffic Monitoring

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Luca Deri
Ntop.org, Italy
deri@ntop.org

Francesco Fusco
ETH Zurich, Switzerland
fusco@ntop.org

1) Flow-based network monitoring is the de-facto monitoring architecture

- Flow meters analyze the traffic and produce *flow records*
- The collector receives , correlates and analyzes them
- Standardized protocols (sFlow, NetFlow, IPFIX) defines the record format
- Strict PUSH model: the collector does not communicate with the probe (s)
- Flow records are exported when a network communication ends

2) Software probes have enabled service-oriented network monitoring

- Extensible and flexible
- Application level traffic analysis: DNS, HTTP, MySQL, VoIP
- Support for many encapsulation protocols (GRE, LTE)
- Modern commodity hardware is powerful enough to enable flow monitoring in high-speed networks

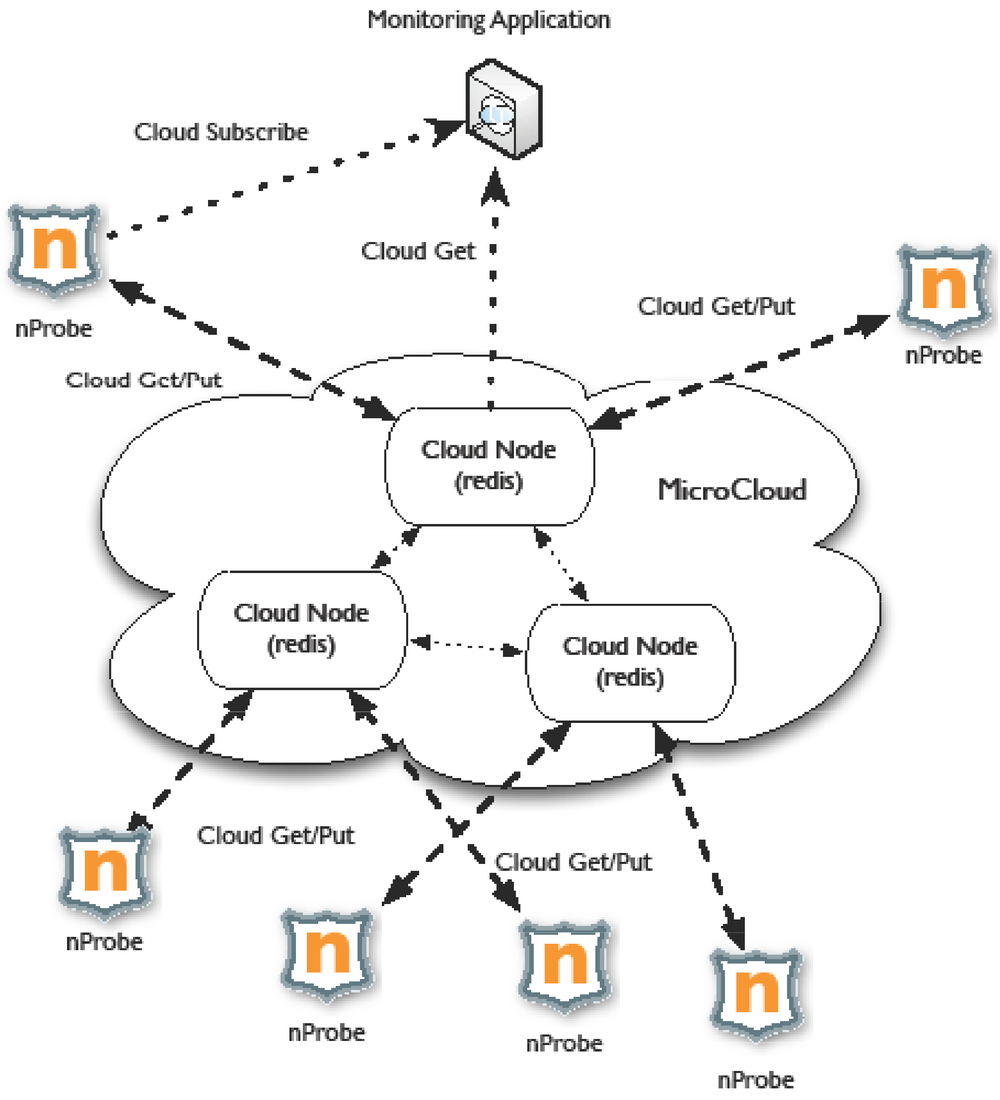
3) The PUSH model poses constraints

- The collector can only have a deferred view of the network
- Correlations can only be performed once the flows record are exported
- Software probes have made these limitations more pronounced
- How to correlate network flows belonging to the same L7 session?

Setting new goals

- Enable real-time aggregations
- Make application layer information available to 3rd party tools
- Enable information sharing between probes

The *MicroCloud* is a distributed knowledge database



- The Cloud Nodes are Redis[2] instances, a modern key-value store
  - keys are hierarchically organized: "ip.192.168.0.10"
  - values can be complex data types: {sent_pkt=2,rcvd_pkt=5}
  - keys can have a lifetime (e.g. traffic counters for a given host)
  - or live until removed: user to IP address association
- nProbe[1] is an IPFIX meter enabling application level analyses
  - plugins for DNS, HTTP, VoIP, databases (Oracle, MySQL)
  - support encapsulation and tunneling protocols (e.g. GTP)

The probes

- write time-sensitive information to the cloud databases
- emits flow records as in the push model
- can use information present in the cloud
  - e.g., what is the IP associated to this user?

The collector

- Receives flow-record as in the standard push model
- Can subscribe to specific events on any cloud node
  - e.g. send me an update if you see a new VoIP user
- Can poll any cloud database for information

External applications

- The monitoring data is available to 3rd party applications
- Monitoring applications can be implemented in any language supported by Redis (e.g. Python)
- Example: get all the active VoIP users

Traffic analysis use cases

- *3G/4G* : associate traffic with a specific user
- *Voice Over IP*: timely correlation of voice and signaling
- *DNS* : aggregate DNS queries in real-time

Main Benefits

- Modular monitoring architectures
  - the monitoring data is always available in the cloud
  - easily accessible by 3rd party applications
- The cloud stores time sensitive information
  - correlations can be done in real-time
  - collector can subscribe to time sensitive events

References

1. http://www.ntop.org/products/nprobe/
2. http://www.redis.io/