# Scalable Service Performance Monitoring

Idilio Drago, Aiko Pras

# Scalable service performance monitoring

Idilio Drago and Aiko Pras

University of Twente, The Netherlands
{i.drago,pras}@utwente.nl

**Abstract.** Dependable performance measurement is a common requirement for all on-line services. The ongoing tendency to outsource not only infrastructure, but also software parts to several suppliers, creates a new challenge to providers. Traditional solutions cannot easily monitor service performance as perceived by end users, since providers do not have control over all hardware/software involved. Our research focuses on evaluating how data passively collected from network devices can be used to verify end-to-end service performance. Our goal is to compare this approach with current methods used to monitor distributed services. Given the constant mutation of applications on the Internet, as well as the fast increase of the number of users, we are looking for flexible and scalable solutions.

**Key words:** Measurement, service performance, traffic analysis, IPFIX

## 1 Introduction

Services provides always depend on some measurement system to monitor the performance of their services. The common goal of all providers is to detect and to solve problems before their users are affected. From the perspective of end-users, the performance of an application is the combination of the performance of all software and infrastructure involved, including servers, network and client machines [1]. Providers usually monitor the health of their services both by collecting as much information as possible from existing activity on their infrastructure (*passive measurement*) and by regularly probing their services to inspect a set of performance metrics (*active measurement*) [2]. Protocols like SNMP [3], RMON-2 [4], Syslog [5], and NetFlow [6] are some of the most employed solutions in those situations.

The ongoing tendency to outsource not only infrastructure, but also software parts to several suppliers, creates a new challenge to providers. Although services like Amazon Elastic Compute Cloud (Amazon EC2) [7] (which offers a virtual computing environment) and Google Apps [8] (which offers on-line messaging and collaboration applications) create new possibilities to providers, they also make more difficult to measure and to control the performance of services, especially when only parts are deployed in such environments. In those situations, traditional monitoring solutions are no longer easy to implement for several reasons, as for example:

1. *Providers have neither control over all hardware/software delivering their service nor access to client machines*: Providers will not have the same monitoring tools that they have in their own data centre. Although a contract may stipulate quality limits, measurements are normally done by the supplier. Likewise, even in strict environments, like in an enterprise network, client devices may be uncontrollable or not flexible enough to receive parts of the monitoring solution.

2. *Server environments are decentralised, applications have multiple instances running on clusters, and capacity can be adjusted via virtualisation*: If a service is running on a virtual environment with dynamic capacity, agents collecting parameters in the virtual server may be useless. In the same way, few active probes will not give a correct overview of the performance if the service has several replicas running on clusters.

3. *Active approaches become excessively intrusive or application specific in those scenarios*: The main weaknesses of active approaches are amplified in highly distributed environments. For example, the amount of probes required to check performance metrics of services running on those environments can easy become prohibitive. Besides, some other solutions, like embedding performance metrics into application data, are application specific.

4. *Scalability of the measurement solution is even more critical*: Since services are distributed across several suppliers, measurement data must travel through the network. As the number of users increase, the impact of the measurement system on the normal operation becomes more critical.

Our research focuses on evaluating solutions for performance monitoring of services in those highly distributed environments. Given the weaknesses of traditional approaches on such environments, our research is based on the following three assumptions:

1. *Measurements must come from few sources, and a single collector device is desirable*: We are assuming that there are a few concentration points from where providers can watch the activity of their users. For example, edge routers in an enterprise/campus network. In that case, we avoid the complexity of having several instances of the monitoring solution distributed throughout the network, as well as the uncertainty of measurements in virtual environments.

2. *Measurement structure must be robust, flexible and scalable*: We are assuming that communication patterns on the network, represented by network flows, can provide enough information to extract end-to-end performance metrics. Network flows are already widely used to monitor high speed networks because of scalability concerns. In addition, we are assuming that the definition of a flow is adjustable, giving us flexibility to monitor heteroge-

neous applications.

3. *Standard solutions are preferable*: Instead of proposing a completely new architecture, we are interested in evaluating how standard solutions can be applied in a special situation. Since we are dealing with network flows, IPFIX [9] is the natural choice as a starting point. It is important to note that quality of service is a target application of IPFIX. However, end-to-end service monitoring, as defined for example in the RMON2 APP-MIB [1] is not addressed in IPFIX standard [10].

## 2 Hypothesis and Research Questions

Our hypothesis is that most of the metrics used today to indicate end-to-end performance of services could be calculated or satisfactorily estimated from network flows. In order to verify that hypothesis, we formulated our main research question as follows:

– *How to monitor service performance based on network flow information?*

Given the limitations of current approaches described in the previous section, we split our main research question in the following four embedded questions:

– *How to identify services by their network flow patterns?*

The traffic generated by Internet applications is constantly mutating. Invariant properties that summarise all communication in a network are desirable, but difficult to define [11]. In order to extract performance metrics from network flow data, we have to connect user actions with a set of flows [10]. Additionally, our solution should not be limited to few applications or specific versions. As an example, in [12] network flows are used to monitor applications, but this solution is only valid for one type of application (SIP protocol). We are interested in methods that automatically identify Internet traffic, as for example some of the techniques presented in [2].

– *How to extract service performance metrics from flow data?*

Once we have identified the flows generated by a user action, we have to extract performance metrics from them. In order to answer this question, we have to define the performance metrics of interest. The RMON2 APP-MIB [1] will be our starting point for that. The ongoing work presented in [13], which shows methods to extract basic performance metrics from flow data, will also be considered in our research.

– *How would the proposed solution perform in real environments?*

Several aspects of real networks can unfavourably affect our solution. For example, background traffic, data lost, tunnelling and data encryption are major hurdles for our proposal. In order to overcome those hurdles, we intend to use data from real networks to develop and to evaluate our solution.

– *How would the proposed solution perform when compared to other approaches?*

In Section 1, we have identified some of the main weaknesses of current solutions in the studied scenario. We want to compare our solution to current approaches and verify if those weaknesses are satisfactory overcome by our proposal.

## 3 Summary

We are researching the use of network flows for scalable service performance monitoring in highly distribute environments. Our goal is to extract end-to-end performance metrics from network flow data, and to compare this solution with current approaches. We are planning to check our results through experimental validation both in controlled environments, like small laboratory experiments, and in large scale environments, like our campus network.

**Acknowledgments.** This work has been carried out in the context of the IOP GenCom project Service Optimisation and Quality (SeQual), which is supported by the Dutch Ministry of Economic Affairs via its agency SenterNovem.

## References

1. Waldbusser, S.: Application Performance Measurement MIB (2004) http://www.ietf.org/rfc/rfc3729.txt.
2. Callado, A., Kamienski, C., Szabó, G., Gero, B.P., Kelner, J., Fernandes, S., Sadok, D.: A Survey on Internet Traffic Identification. IEEE Communications Surveys & Tutorials **11**(3) (2009) 37–52
3. Case, J., Fedor, M., Schoffstall, M., Davin, J.: A Simple Network Management Protocol (SNMP) (1990) http://www.ietf.org/rfc/rfc1157.txt.
4. Waldbusser, S.: Remote Network Monitoring Management Information Base Version 2 (1997) http://www.ietf.org/rfc/rfc2021.txt.
5. Gerhards, R.: The Syslog Protocol (2009) http://www.ietf.org/rfc/rfc5424.txt.
6. Claise, B.: Cisco Systems NetFlow Services Export Version 9 (2004)
7. Amazon.com Inc.: Amazon Elastic Compute Cloud (Amazon EC2) (2010) http://aws.amazon.com/ec2/.
8. Google Inc.: Google Apps (2010) http://www.google.com/apps/.
9. Quittek, J., Zseby, T., Claise, B., Zander, S.: Requirements for IP Flow Information Export (IPFIX) (2004) http://www.ietf.org/rfc/rfc3917.txt.
10. Zseby, T., Boschi, E., Brownlee, N., Claise, B.: IP Flow Information Export (IPFIX) Applicability (2009) http://www.ietf.org/rfc/rfc5472.txt.
11. Williamson, C.: Internet Traffic Measurement. IEEE Internet Computing **5** (2001) 70–74
12. Anderson, S., Niccolini, S., Hogrefe, D.: SIPFIX: A Scheme For Distributed SIP Monitoring. In: IM'09: Proceedings of the 11th IFIP/IEEE International Symposium on Integrated Network Management, Piscataway, NJ, USA, IEEE Press (2009) 382–389
13. Kögel, J.: Including the Network View in Application Response Time Diagnostics using NetFlow. 2009 USENIX Annual Technical Conference (2009)