Optimization of Distributed Services with UNISONO

Dirk Haage, Ralph Holz
Wilhelm-Schickard-Institut für Informatik
Universität Tübingen, Germany
{haage,holz}@informatik.uni-tuebingen.de

Abstract

Distributed services are a special case of P2P networks where nodes have several distinctive tasks. Based on previous work, we show how UNISONO provides a way to optimize these services to increase performance, efficiency and user experience. UNISONO is a generic framework for host-based distributed network measurements. In this talk, we present UNISONO as an Enabler for self-organizing Service Delivery Plattforms. We give a short overview of the UNISONO concept and show how distributed services benefit from its usage.

1 Introduction

Overlays seem to be one way of bringing more functionality to future networks. They can be used for services like media streaming, but are also discussed as a way to enhance grids or Service-Oriented Architectures [1]. It is a generally accepted fact that cross-layer information can be very useful for the optimization and re-structuring of such overlays. Furthermore, this type of information can be of special interest for services of all kind, e.g. in SOA environments or virtualized environments. The nature of these services, i.e. a complex service is cut into small parts, each provided as a service of its own and distributed over the network, offer an interesting possibility: fast relocation or reinstantiation of the service itself, depending on the needs of the user or the situation within the network.

An optimization strategy is likely to rely on accurate information about hosts and network. This opens new challenges. First, the task of information gathering is complex and needs to be adapted to the substrate as well as to available resource, used technology etc. Addressing this separately for each application, service or overlay is a waste of resources. Second, with more applications relying on information that has to be obtained by active measurements, the measurement noise within the network increases substantially. This wastes the limited resources of the network. Furthermore, each concurrent measurement within a distributed system adds bias to the measurement result, rendering them hardly usable [3].

In this talk, we present UNISONO, the cross-layer measurement component used by SpoVNet (Spontaneous Virtual Networks) [6] for optimization purposes. UNISONO is an implementation of the split architecture proposed in [4]. The service encapsulates measurement methods and logic.

2 UNISONO Framework

UNISONO is realized as a local service that only accepts queries from local applications. Queries are issued as *orders* in an asynchronous way via a simple interface. They can be one-shot, but also periodic with result delivery in fixed intervals or on exceeding a given threshold. The information is gathered with several measurement methods. These are encapsulated in modules, adhering to a plugin-concept. This enables easy extension of UNISONO with more methods and the quick exchange of algorithms. It is also possible to have more than one method for each purpose.

The main objective of UNISONO is to reduce the number of measurements within the network, especially for active measurements such as bandwidth estimations or topology detection. A first step towards this goal is already taken by cutting out the measurement functionality of applications. This allows aggregation of overlapping requests and efficient caching of results. If two queriers need the same measurement within a certain time frame, the measurement is only executed once.

The measurement logic is encapsulated in UNISONO, keeping applications light-weight and unaware of the underlying mechanisms. Furthermore, instances of the same service on different hosts are able to coordinate measurements and exchange results. It also allows distributed caching of measurement results.

In addition to network and link properties, UNISONO provides node properties via the same interface. Among them are CPU properties, available RAM and HDD, but also host uptime, host interfaces (like WLAN) and battery state.

Compared to SNMP [2], this permits additional processing like the computation of trends and change probabilities directly within UNISONO. Aside from that, the application uses a single source to get information and only needs to implement one interface.

3 Optimization of Distributed Services

While the main target of UNISONO was the organization and optimization of overlay networks, the collected information can be used in a more generic way. We give two examples how UNISONO can be instrumented for optimization purposes. The event notification[5] within SpoVNet [6] already does it while the placement of SOA services in the network is the generalization of this concept.

Distributed Event Service The event service provides SpoVNet-based applications a fully distributed publish/subscribe system. With this, it is possible to provide, aggregate and distribute any type of information. Furthermore, it features event correlation. To support the distributed nature of a SpoVNet-Overlay the service is splitted into three parts: a publisher, a correlator and an aggregator. Each of these parts can be placed on any node within the overlay. The placement of each component is crucial for the service to efficiently and within certain QoS restraints. Furthermore, dynamically changing groups of nodes interested in different parts of the provided information need individual (re-)placement. UNISONO provides the information about involved nodes and network links with little overhead, allowing the event service to achieve its goal without a deep knowledge about the underlying network. The service utilizes network delay and bandwidth estimations as well as available computational resources (mainly CPU and available memory) for its multi-dimensional optimization. Without UNISONO, this information must be collected by the service itself, resulting in more complexity and resource usage.

SOA service placement Taking the above described event service as a starting point, we can generalize the concept for service-oriented architectures. Similar to the given example, SOA services provide small tasks that can be plugged together to provide new, complex services. Placement of such services within the network is currently mostly static. Realizing a SOA service as a virtual network appliance (i.e. a small virtual machines) would not only allow fast re-placement of the service but also replication. The automation of these processes requires information about the underlying network as well as about the node a service is running on and the usage of the service itself. This cross-layer information is the same as required by overlays, hence the information already provided by UNISONO. Although we believe it is still a long way away, our vision would be a self-organizing SOA environment, where services are provided where needed, with more efficiency as can be provided by a centralized administration.

4 Ongoing Work

While we are continously adding more plugins for measurement procedures, correlation mechanisms and statistics to UNISONO, we are also considering security mechanisms to protect UNISONO from misuse. We also work on more dynamic caching and information aging strategies. The extension of UNISONO with an overlay of its own to optimize communication between instances as well as the distributed caching strategies is also considered

5 Conclusion

We have presented UNISONO, our framework for cross-layer measurements in distributed environments. UNISONO provides generic access to node and link information. With EONSON, the event service of SpoVNet, we have presented a distributed service already utilizing UNISONO for component placement. We have also sketched possible ways to use the acquired information for service placement and self-organizing SOA environments.

References

[1] M. Amoretti, F. Zanichelli, and G. Conte. Sp2a: a service-oriented framework for p2p-based grids. In *Proc. 3rd international workshop on Middleware for grid computing (MGC '05)*, pages 1–6, New York, NY, USA, 2005. ACM.

[2] J. Case, R. Mundy, D. Partain, and B. Stewart. Introduction and applicability statements for Internet-Standard management framework. RFC 3410, Internet Engineering Task Force, Dec. 2002.

[3] D. Croce, M. Mellia, and E. Leonardi. The quest for bandwidth estimation techniques for large-scale distributed systems. In *ACM Hotmetrics 2009*, Seattle, jun 2009.

[4] D. Haage, R. Holz, H. Niedermayer, and P. Laskov. CLIO – a cross-layer information service for overlay network optimization. In *Kommunikation in Verteilten Systemen (KiVS) 2009*, Kassel, Germany, March 2009.

[5] A. Tariq, B. Koldehofe, G. Koch, and K. Rothermel. Providing Probabilistic Latency Bounds for Dynamic Publish/Subscribe Systems. In *Proceedings of the 16th ITG/GI Conference on Kommunikation in Verteilten Systemen 2009 (KiVS 2009)*, Kassel, Germany, Januar 2009. Springer.

[6] O. Waldhorst, C. Blankenhorn, D. Haage, R. Holz, G. Koch, B. Koldehofe, F. Lampi, C. Mayer, and S. Mies. Spontaneous virtual networks: On the road towards the internet's next generation. *it – Information Technology Special Issue on Next Generation Internet*, 50(6):367–375, December 2008.