PeerfactSim.KOM: A P2P System Simulator - Experiences and Lessons Learned

Kalman Graffi
University of Paderborn, Theory of Distributed Systems
Fürstenallee 11, 33102 Paderborn, Germany. Email: graffi@mail.upb.de

I. Introduction

Research on peer-to-peer (p2p) and distributed systems needs evaluation tools to predict and observe the behavior of protocols and mechanisms in large scale networks. PeerfactSim.KOM[1] [1] is a simulator for large scale distributed/p2p systems aiming at the evaluation of interdependencies in multi-layered p2p systems. The simulator is written in Java, is event-based and mainly used in p2p research projects[4]. The main development of PeerfactSim.KOM started in 2005 and is driven since 2006 by the project "QuaP2P"[2], which aims at the systematic improvement and benchmarking of p2p systems. Further users of the simulator are working in the project "On-the-fly Computing"[3] aiming at researching p2p-based service oriented architectures. Both projects[5] state severe requirements on the evaluation of multi-layered and large-scale distributed systems. We describe the architecture of PeerfactSim.KOM supporting these requirements in Section II, present the workflow, selected experiences and lessons learned in Section III and conclude the overview in Section IV.

II. PeerfactSim.KOM - A P2P System Simulator

PeerfactSim.KOM is a flexible and mature event-based simulator written in Java. The events follow a timeline which assures sequential processing. Events are part of so-called operations, which are generated either through the entities in the simulation or through an external actions file. Such an operation is for example a lookup in a DHT. Operations help to trace actions on every layer, allowing the protocols to store local state information easily and to react on operation timeouts. Operations on various layers are decoupled which allows for the combined simulation and evaluation of various protocols in parallel. The functional layers of PeerfactSim.KOM, as shown in Figure 1, are as follows. The **user layer** can be used to define strategies of various user types that are performed on the application layer. The **application layer** defines the application logic and its characteristics, such as file sharing with typical Zipf-distributed request patterns. Various advanced protocols are covered in the **service layer**. Management and control mechanisms, like [2], use system aggregation monitors, like [3], to constantly optimize the p2p system configuration. Such services are neither part of the application nor the p2p overlay. The service layer functions

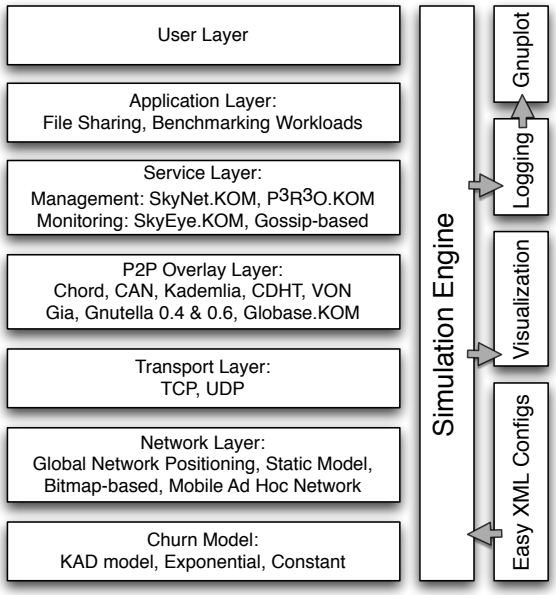


Fig. 1. Overview on PeerfactSim.KOM

use the p2p overlay interfaces in order to provide general functional offers that improve the quality of the p2p application or create a reusable functional building block for various p2p applications. The **p2p overlay layer** covers structured (Chord, Kademlia, Pastry, Globase), unstructured (GIA, Gnutella 0.4, Gnutella 0.6, Napster) and information dissemination (VON, pSense, Mercury) p2p overlays with corresponding interfaces (e.g. the Key-based Routing API [4]). The **transport layer** serializes messages and offers TCP and UDP as implemented protocols, which can be used in combination with the **network layer** to obtain realistic values for throughput, delay, jitter, loss and peer positioning. The network layer implements besides static and simple network models also advanced models, like Global Network Positioning (GNP) based on measurements from the PingER project. The **churn models** that can be activated for time intervals are either based on measurements (in KAD) or implement popular churn behavior (exponential). The simulations are conducted by the **simulation event queue**, which manages and schedules events in the simulation. Every event is processed at it scheduled time and logged for further analysis. The **logging** is twofold. First, a history of relevant events is stored for a later visualization. Second, a layer-wise protocol of the events is captured by **analyzers** creating simulation statistic files which can be directly fed into gnuplot. Thus, the simulator helps to easily create plotted results.

[1] www.peerfact.org [2] www.quap2p.de [3] sfb901.upb.de

[4] This work has been partially funded by the DFG SFB 901, DFG FOR 733
[5] Both projects are funded by "Deutsche Forschungsgemeinschaft" (DFG)

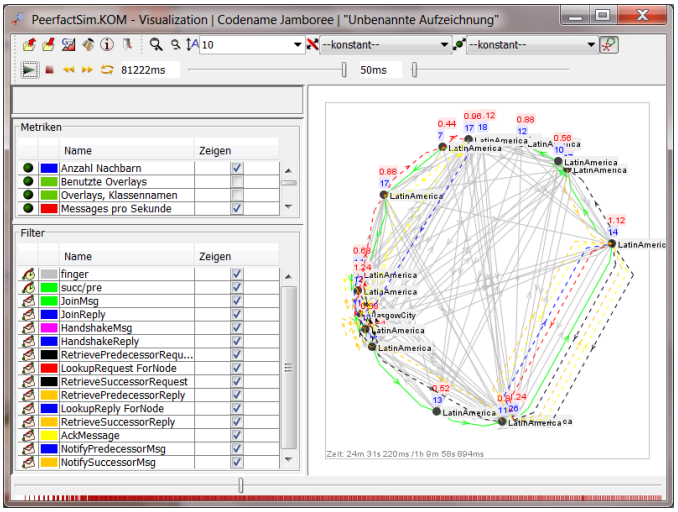


Fig. 2. Visualization of a Chord Simulation

III. Experiences and Lessons Learned

The workflow with the simulator is very simple. An XML-based configuration file is used to specify which implementations and configurations on which layers are to be used. The configuration file further specifies an action file, which contains the operations that are to be started by specific peers at specific time intervals. Once, the configuration is completed, the user may start a GUI to choose the configuration file and to observe the simulation status. If a visualization module was chosen, the traffic and recorded statistics over the simulation time may be observed, once the simulation finished. An example of this visualization is given in Figure 2. Independent of running simulations visualized or headless, the plotable simulation statistics are automatically generated.

PeerfactSim.KOM aims at the evaluation of interdependencies in multi-layered p2p systems. A typical paper presenting PeerfactSim.KOM's capabilities is [2]. In this paper, a distributed management and control mechanism for p2p systems is presented on top of a system statistics monitor [3] on top of Chord using a GNP network. Another example is [5], in which a SLA-oriented resource reservations approach is presented using a peer capacity monitoring and search mechanism [3] on top of a KBR-overlay and a GNP network. Various underlay modeling approaches were presented in [6]. Effects of proximity-awareness in Kademlia have been investigated in [7]. A proximity- and bandwidth-aware streaming approach is presented in [8]. An example of combining a p2p overlay for location-based search, Globase.KOM [9], with advanced packet priorization [10] and a (bit)map-based network resulted in a p2p-based emergency call handling approach [11].

Through the focus of PeerfactSim.KOM on the interaction of (p2p) protocols on various layers, complex systems can be built and evaluated. The aim of the researchers in the project "QuaP2P" is to create a benchmarking platform for p2p systems [12], which allows the systematic and objective comparison of p2p protocols. In order to support this, PeerfactSim.KOM is built to be compliant with a common interface for overlay network simulators, as shown in [13], allowing the reuse of code in various simulators. The project "On-the-Fly Computing" will use PeerfactSim.KOM to evaluate service-oriented architectures on top of p2p overlays. This requires various advanced mechanisms in the protocol stack, a requirement that is matched by PeerfactSim.KOM. Demonstrations of PeerfactSim.KOM can be found on `www.peerfact.org` and will be shown along with the presentation of this paper.

IV. Conclusions

PeerfactSim.KOM has been used and developed since 2005, aiming at the evaluation of multi-layered p2p systems. The simulator is user-friendly both for users, which are supported by an automatized workflow, and for developers, which are supported by tutorials, docs and well documented java code. Like the p2p simulators P2PSim, OverSim, PeerSim or ProtoPeer, the presented simulator is also able to simulate over 100,000 peers as it was documented in corresponding papers. For multi-layered p2p systems a network size of 10,000 peers is recommended to handle the occurring events in a sufficient detail and a reasonable time. With various parallel protocols in place, PeerfactSim.KOM enfolds is strengths. PeerfactSim.KOM uses the GPLv3 license and is recommended for any researcher aiming at the simulation of multi-layered p2p systems in combination with realistic network models.

Acknowledgement

The author thanks Sebastian Kaune, Aleksandra Kovacevic, Konstantin Pussep, Dominik Stingl, Christian Groß, Julius Rückert, Leo Nobach, Alexander Nigl and Ralf Steinmetz for their precious work invested in PeerfactSim.KOM.

References

[1] A. Kovacevic, S. Kaune *et al.*, "Benchmarking Platform for Peer-to-Peer Systems," *it - Information Technology*, vol. 49, no. 5, 2007.
[2] K. Graffi, D. Stingl, J. Rückert *et al.*, "Monitoring and Management of Structured Peer-to-Peer Systems," in *Proc. of IEEE P2P '09*, 2009.
[3] K. Graffi, A. Kovacevic, S. Xiao, and R. Steinmetz, "SkyEye.KOM: An Information Management Over-Overlay for Getting the Oracle View on Structured P2P Systems," in *Proc. of IEEE ICPADS '08*, 2008.
[4] F. Dabek, B. Y. Zhao, P. Druschel *et al.*, "Towards a Common API for Structured Peer-to-Peer Overlays," in *Proc. of IPTPS '03*, 2003.
[5] K. Graffi, C. Groß, D. Stingl, H. Nguyen, A. Kovacevic, and R. Steinmetz, "Towards a P2P Cloud: Reliable Resource Reservations in Unreliable P2P Systems," in *Proc. of IEEE ICPADS '10*, 2010.
[6] S. Kaune, K. Pussep, C. Leng, A. Kovacevic, G. Tyson, and R. Steinmetz, "Modelling the Internet Delay Space Based on Geographical Locations," in *Proc. of PDP '09*, 2009.
[7] S. Kaune, T. Lauinger, K. Pussep *et al.*, "Embracing the Peer Next Door: Proximity in Kademlia," in *Proc. of IEEE P2P '08*, 2008.
[8] K. Graffi, S. Kaune, K. Pussep, A. Kovacevic, and R. Steinmetz, "Load Balancing for Multimedia Streaming in Heterogeneous Peer-to-Peer Systems," in *Proc. of ACM NOSSDAV '08*, 2008.
[9] A. Kovacevic *et al.*, "Globase.KOM - A P2P Overlay for Fully Retrievable Location-based Search," in *Proc. of IEEE P2P '07*, 2007.
[10] K. Graffi, K. Pussep, S. Kaune, A. Kovacevic, N. Liebau, and R. Steinmetz, "Overlay Bandwidth Management: Scheduling and Active Queue Management of Overlay Flows," in *Proc. of IEEE LCN '07*, 2007.
[11] K. Graffi, A. Kovacevic *et al.*, "ECHoP2P: Emergency Call Handling over Peer-to-Peer Overlays," in *Proc. of IEEE P2PNVE'07*, 2007.
[12] A. Kovacevic, K. Graffi, S. Kaune, C. Leng, and R. Steinmetz, "Towards Benchmarking of Structured Peer-to-Peer Overlays for Network Virtual Environments," in *Proc. of IEEE ICPADS '08*, 2008.
[13] C. Groß, M. Lehn, D. Stingl *et al.*, "Towards a Common Interface for Overlay Network Simulators," in *Proc. of IEEE ICPADS '10*, 2010.