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\(^1\) 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\(^2\). The main development of PeerfactSim.KOM started in 2005 and is driven since 2006 by the project “QuaP2P”\(^3\), 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”\(^4\) 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 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 its 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. sb901.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)
for overlay network simulators, as shown in [13], allowing the
Sim.KOM is built to be compliant with a common interface
comparison of p2p protocols. In order to support this, Peerfact-
p2p systems [12], which allows the systematic and objective
project “QuaP2P” is to create a benchmarking platform for
be built and evaluated. The aim of the researchers in the
of (p2p) protocols on various layers, complex systems can
in a p2p-based emergency call handling approach [11].
example of combining a p2p overlay
delay modeling approaches were presented in [6]. Effects
on top of a KBR-overlay and a GNP network. Various un-
dispatch and a reasonable time. With various parallel protocols
in place, PeerfactSim.KOM enforces strengths. Peerfact-
Sim.KOM uses the GPLv3 license and is recommended for
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 Pro-
toPeer, 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 enforces strengths. Peerfact-
Sim.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

Management Over-Overlay for Getting the Oracle View on Structured
“Towards a P2P Cloud: Reliable Resource Reservations in Unre-
metz, “Modelling the Internet Delay Space Based on Geographical
Locations,” in Proc. of PDP ’09, 2009.
metz, “Overlay Bandwidth Management: Scheduling and Active Queue
Benchmarking of Structured Peer-to-Peer Overlays for Network Virtual