PeerfactSim.KOM: A Simulation Framework for Peer-to-Peer Systems


Dominik Stingl
Christian Gross
Julius Rückert
Aleksandra Kovacevic
Ralf Steinmetz

Dominik.Stingl@KOM.tu-darmstadt.de
Tel.+49 6151 165019

© 2011 author(s) of these slides including research results from the KOM research network and TU Darmstadt. Otherwise it is specified at the respective slide
Motivation

Situation in P2P research

- Design and implementation of new ideas states a common challenge
- Proving, evaluating, and comparing is a problem on its own

Three common evaluation techniques for P2P can be identified

- Analytical modeling
- Simulations
- Experiments

Simulations turned out to be the most popular tool to evaluate new ideas in the area of P2P
Agenda

Motivation

The Simulation Framework
- The Layered Architecture
- Using the Simulator

Evaluation

Conclusion
Overview

General characteristics
- Written in Java
- Event-based simulator
- Simulations with up to 100K peers
- Licensed under GPL v3

History
- Started in 2005 as evaluation tool for a Ph.D. thesis
- Further development during QuaP2P 1 (2006 - 2009)
- Currently improved and extended during QuaP2P 2 (2009 - 2012)
Layered Architecture and its Concepts

Layered Architecture
- Easy exchange of implementations for each layer
- Testing and evaluating new P2P mechanisms

Design concepts
- Each layer offers its functionality to layers above
- Functionality can be accessed by one or more interfaces

Implementation concept
- Default implementation
  - Functionality of a component can be completely implemented
- Skeletal implementation
  - Parts of a component can be implemented
Network Model

Underlying network model
- Hides topology of the Internet
- Considers only connections between peers

Components of the model
- **Subnet** models the connection between all peers
- **Network layer** connects each peer with the Subnet

Lessons learned from early implementations
- Separate implementations obstruct exchangeability
- Designing new models requires implementation of whole network model
→ Helped us to understand modeling the network

![Network Model Diagram]

http://4.bp.blogspot.com/_ADXt1UQuLe0/TOkDZnKQqII/AAAAAABF0c/HjM_ZspGc2Y/s1600/mesh_topology.gif
Modular Network Model

Identification of important aspects of our current network model

- Traffic control
- Packet sizing & fragmenting
- Packet loss
- Latency & jitter
- Positioning

Definition of different strategies for each considered aspect

- Static strategies
- Based on probability functions
- Based on Internet measurement studies

Implementation of a modular network model for the strategies
Transport Layer

Condensed model of a transport layer
- Simulating transmission of UDP

Provisioning of end-to-end communication to higher layers
- Addressing different overlays, applications by using ports

The send-and-wait concept
- Allows a receiver to acknowledge or reply to a request
- Initiator can react on timeouts
- No functionality of a transport layer
- … but relieves layers above from handling this
Overlay Layer

Unstructured overlays
- Gnutella 0.4 & Gnutella 0.6
- Gia

Hybrid overlay
- Globase.KOM

Distributed Hash Tables (DHT)
- CAN
- Centralized Hash Table
- Chord & Pastry
- Kademlia

Information Dissemination Overlays (IDO)
- VON
- pSense
- Centralized IDO
Service Layer

Idea of this layer
- Implementation of additional distributed services
- Offers these services to the whole system
- Can rely on underlying overlay to provide functionality

Existing services
- DHT replication
- Decentralized monitoring
  - Tree-based monitoring
  - Gossip-based monitoring
  - Centralized monitoring
- Decentralized overlay management
  - Maintains and improves an overlay based on data of tree-based monitoring

Interface for decentralized monitoring available
- Stores locally measured attributes
- Requests global view for monitored attributes
Agenda

Motivation

The Simulation Framework
- The Layered Architecture
- Using the Simulator

Evaluation

Conclusion
Configuring a Simulation

Configuration via xml-file
- General definitions for a simulation
- Defines how the peers are assembled
- Specifies the action-file

Define configuration variables from command line
- Possibility to modularize the configuration
  - Defining a single component once
  - Integrating this definition in different configurations

Description of scenario in action-file
- Offers a script-like language
- Specifies when a certain action should be executed by peers

```xml
<?xml version='1.0' encoding='utf-8'?>
<Configuration>
  <Default>
    <Variable name="seed" value="042"/>
    <Variable name="size" value="252"/>
    <Variable name="end" value="120m"/>
  </Default>
  <!-- Simulatoor Core -->
  <SimulatorCore>
    <!-- Components -->
    <NetLayer .../>
    <Overlay .../>
    <Monitor .../>
    <Monitor>
      <ChurnGenerator .../>
    </Monitor>
    <HostBuilder ...>
      <Group size="50" groupId="LatinAmerica">
        <NetLayer/>
        <TransLayer />
        <Chord />
        <Properties enableChurn="false"/>
      </Group>
    </HostBuilder>
  </SimulatorCore>
  <!-- Scenario actions -->
  <Scenario .../>
</Configuration>
```

# Column meanings (format):
# Scenario with the name chord-actions
GlasgowCity 1m join callback
LatinAmerica 2m-50m join callback
# Some comments for the action-file
GlasgowCity 102m store datalocal callback
LatinAmerica 105m-250m valueLookup datalocal callback
The Statistics Architecture

Gathering data is an important aspect of a simulation
→ Capabilities to measure data are integrated into the simulator

Components of the Statistics Architecture

- **Monitor** measures data at different interception points during a simulation
- **Analyzers** register at the monitor to process measured data
  - Monitor forwards data to responsible analyzers
  - Different types of analyzers for different types of data

The statistics architecture allows to write data in files or in databases
Watching a Simulation

Simulator enables live monitoring of a simulation
- Special analyzer measures data of interest

Facilitates the identification of bugs
- Increased amount of dropped messages
- Long traffic control queue size

Visualizing executed simulations
- Displays the topology of peers
- Highlights frequently used connections
- Highlights the utilization of peers
Agenda

Motivation

The Simulation Framework
- The Layered Architecture
- Using the Simulator

Evaluation

Conclusion
Evaluation Setup

Simulation platform
- Server with two Dual-Core AMD Opteron with 2200MHz
- 20GiB per simulation
- Ubuntu Server 10.04 (64bit)
- JDK 6

Simulated components
<table>
<thead>
<tr>
<th>Modular Network Model</th>
<th>Easy-, GNP-preset</th>
</tr>
</thead>
<tbody>
<tr>
<td>P2P overlay</td>
<td>Gnutella 0.6, Kademlia</td>
</tr>
<tr>
<td>Application</td>
<td>File-Sharing</td>
</tr>
</tbody>
</table>

Setup

<table>
<thead>
<tr>
<th>Churn Generator</th>
<th>Exponential Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulation duration</td>
<td>180min</td>
</tr>
<tr>
<td># Peers</td>
<td>500, 1,000, 5,000, 10,000, 50,000</td>
</tr>
</tbody>
</table>

Scenario
- Peers are equally divided into four groups
- Peers join the overlay during the first 80min
- 50% of peers publish and lookup their files
- Churn starts after 120min
Evaluation Results

<table>
<thead>
<tr>
<th></th>
<th>500</th>
<th>1,000</th>
<th>5,000</th>
<th>10,000</th>
<th>50,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kademlia, GNP</td>
<td>1.4</td>
<td>3.2</td>
<td>20.3</td>
<td>43.5</td>
<td>246.4</td>
</tr>
<tr>
<td>Kademlia, Easy</td>
<td>1.2</td>
<td>2.9</td>
<td>19.2</td>
<td>41.5</td>
<td>238</td>
</tr>
<tr>
<td>Gnutella, GNP</td>
<td>6.7</td>
<td>13.5</td>
<td>69.8</td>
<td>142.4</td>
<td>732.4</td>
</tr>
<tr>
<td>Gnutella, Easy</td>
<td>6.6</td>
<td>13.1</td>
<td>67.2</td>
<td>137.6</td>
<td>723.4</td>
</tr>
</tbody>
</table>

Avg. number of msg *10^6
Agenda

Motivation

The Simulation Framework
- The Layered Architecture
- Using the Simulator

Evaluation

Conclusion
Conclusion

Overview of the P2P simulation framework PeerfactSim.KOM

- Layered architecture
- Concepts and existing implementations
- Tools to handle simulations
- Performance evaluation of the simulator

Future work

- Integrating a model of TCP at the transport layer
- Improving the existing connections for databases
- Tackling the simulation of wireless communication models for mobile devices
Thank you for your attention.
Questions?

Find us at www.peerfact.org

Dipl.-Inform. Dominik Stingl
Dipl.-Inform. Dominik Stingl@KOM.tu-darmstadt.de
Rundeturmstr. 10
64283 Darmstadt
Germany

Phone +49 (0) 6151/165019
Fax +49 (0) 6151/166152
www.kom.tu-darmstadt.de
## Backup Slides

### Simulation Duration

<table>
<thead>
<tr>
<th># Peers</th>
<th>Kademlia, GNP</th>
<th>Kademlia, Easy</th>
<th>Gnutella, GNP</th>
<th>Gnutella, Easy</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.1</td>
<td>0.1</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>1,000</td>
<td>3.2</td>
<td>2.9</td>
<td>13.5</td>
<td>13.1</td>
</tr>
<tr>
<td>10,000</td>
<td>43.5</td>
<td>41.5</td>
<td>142.4</td>
<td>137.6</td>
</tr>
<tr>
<td>100,000</td>
<td>523.4</td>
<td>506.6</td>
<td>1488.7</td>
<td>1476.1</td>
</tr>
</tbody>
</table>

### Memory Utilization

<table>
<thead>
<tr>
<th># Peers</th>
<th>Kademlia with GNP</th>
<th>Kademlia with Easy</th>
<th>Gnutella with GNP</th>
<th>Gnutella with Easy</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>1,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
<td>10,000</td>
</tr>
<tr>
<td>10,000</td>
<td>100,000</td>
<td>100,000</td>
<td>100,000</td>
<td>100,000</td>
</tr>
</tbody>
</table>

**Avg. number of msg *10^6**