Availability Simulation of Peer-to-Peer Architectural Styles

Simon Giesecke, Timo Warns, Wilhelm Hasselbring

Referee: Timo Warns
Motivation

- Evaluation of availability of P2P services
- Specifics of P2P context impacting availability
  - Failure distribution of peers
  - Means of handling failures
  - Dynamic architecture / topology
- How to integrate these aspects?
  - Focus: Architectural Style
Conceptual framework

- P2P styles
- P2P architectures
- P2P systems

Evaluation by simulation

- “most real-world systems are too complex to allow realistic models to be evaluated analytically”
  Law and Kelton, 2000
- Flexible
Peer-to-Peer Styles

Classification scheme
- Type of decentralization
  - Decentralized, hybrid, super-peer
- Type of communication
  - Direct, Indirect, Mediated
- Structural Characteristics
  - Ring, Tree, Small-World Network

Rules for evolution
- Joining / leaving of peers
- No formalisation yet
Graph-based formalism $A = (N, C, \nu, \lambda, \tau)$

- $N, C$ – Sets of nodes and connections
- $\nu: C \rightarrow \{\{n_1, n_2\} | n_1 \neq n_2 \text{ and } n_1, n_2 \text{ in } N\}$ – Node function
- $\lambda: N \rightarrow L$ – Labelling function
  - $L$ is a set of node labels (e.g., “Peer”, “Server”, ...)
- $\tau: T \rightarrow NC_T$ – Time mapping

$\tau$ describes evolution over time

- E.g., peer $p$ participates at system from $t_n$ to $t_m$
  $\Rightarrow p$ is in image of $\tau$ for $t$ in $[t_n, t_m]$
Example Description Model

- \( N = \{p_1, ..., p_4\} \)
- \( C = \{c_1, ..., c_5\} \)
- \( \lambda(n) = \text{Peer} \) for all \( n \) in \( N \)
- \( V: \)
<table>
<thead>
<tr>
<th>( c )</th>
<th>( v(c) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( c_1 )</td>
<td>( {p_1, p_3} )</td>
</tr>
<tr>
<td>( c_2 )</td>
<td>( {p_1, p_2} )</td>
</tr>
<tr>
<td>( c_3 )</td>
<td>( {p_2, p_3} )</td>
</tr>
<tr>
<td>( c_4 )</td>
<td>( {p_3, p_4} )</td>
</tr>
<tr>
<td>( c_5 )</td>
<td>( {p_1, p_4} )</td>
</tr>
</tbody>
</table>
- \( T: \)
  | \( T \) | \( NC_\tau \) |
  | [\( t_0 \), \( t_1 \)] | \( p_1, ..., p_3, c_1, ..., c_3 \) |
  | [\( t_1 \), \( t_2 \)] | \( p_1, ..., p_4, c_1, ..., c_4 \) |
  | [\( t_2 \), \( t_3 \)] | \( p_1, ..., p_4, c_1, ..., c_5 \) |
Prototype of simulator
- Based on graph formalism
Peer model
- Derived from real-world system
- Enhanced by classic replication strategies
Evaluation of availability of replicated resources
Conclusions

- Conceptual framework
  - Evaluation of availability of P2P services
  - Architectural styles, architectures, systems
- Classification scheme for architectural styles
- Description model for P2P architectures
- Simulator prototype
Future Work

- Formalisation of architectural styles
  - Graph grammars?
  - Benefit: Automated creation of architectures
- Formalisation of peer model
  - Add peer model to input for simulation
  - UML?
- Development of improved simulator
  - Prototype used manually created architectures and one fixed peer model