

Decentralized Unstructured Flat P2P Network with Streaming Content Delivery Method and User Collaboration

Masterstudium:
Software Engineering & Internet Computing

Daniel Kanev

Technische Universität Wien
Institut für Computersprachen
Arbeitsbereich: Programmiersprachen und Übersetzerbau
BetreuerIn: A.o. Univ. Prof. Dr. Dipl.-Ing. eva Kühn
Co-Betreuerin: Dr.techn.Mag. Dipl.Math Vesna Šešum-Čavić

Context

In fully distributed P2P networks there is no mapping between data and peers. Search algorithms in such overlays can be designed to support complex search queries. In video on-demand (VoD) streaming schemes users can change the playback position of the streamed video. User collaboration is the process of collaboratively modifying shared content.

Motivation and Requirements

- ▶ **Develop a P2P application** with support for VoD streaming, fully distributed video search, and user collaboration by sharing video comments. It should also provide a security mechanism, which will be centralized and implemented using the MozartSpaces network middleware.
- ▶ **Adapt the Slime Mold algorithm from [1]**, and use it in the context of P2P. The resulting search algorithm should allow more complex queries to be executed, and not only “exact matches”, as in [2].

Slime Mold

The slime molds are organisms which use spores to reproduce. This work is focused on the lifecycle of the cellular slime mold: *Dictyostelium discoideum* (Dd). The Dd's lifecycle goes through five stages: vegetative movement, aggregation, mound, slug movement, and fruiting body [1].

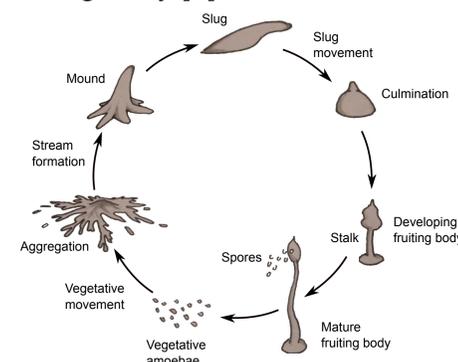


Figure: Dictyostelium Discoideum Lifecycle

Monismith [1] uses the lifecycle of *Dictyostelium discoideum* to solve a numerical single objective function optimization problem.

Design and Implementation

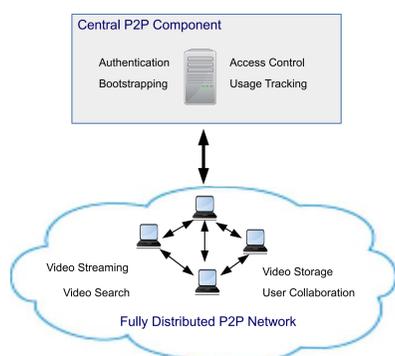


Figure: High Level Architecture

P2P Application

- ▶ Video streaming, search, storage, and user collaboration are performed in a fully distributed mesh, no central component involved.
- ▶ Authentication, bootstrapping, access control, and usage tracking are performed by a Central P2P Component.

Slime Mold for P2P

- ▶ Each peer is a slime mold node.
- ▶ Amoebae are created when a search query is issued.
- ▶ Amoebae move along overlay links, aggregate, and collaborate.
- ▶ Dispersal “brings” the result back to the query originator.

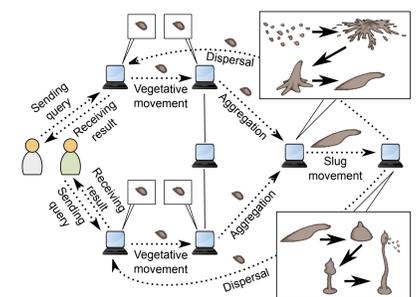


Figure: Slime Mold for P2P

Slime Mold Results

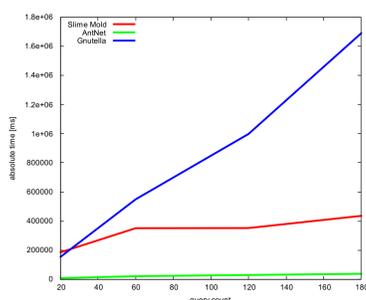


Figure: 200 nodes network

Scalability: positive scalability, better than AntNet for higher load. Slime Mold scales better than Gnutella in all cases. Slime Mold scales slightly worse than AntNet for lower load.

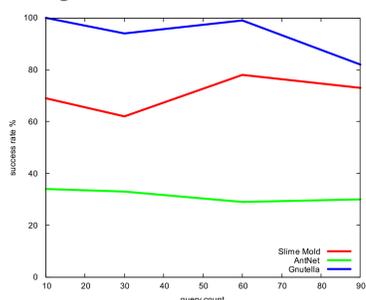


Figure: 100 nodes network

Success rate: lower than Gnutella, better than AntNet for small replication.

Absolute time: higher absolute time than AntNet, lower absolute time than Gnutella for larger network sizes.

Conclusion

Requirements fulfilled

- ▶ Slime Mold adapted to the context of P2P
- ▶ P2P application with fully distributed search
- ▶ Mesh-based VoD with fully distributed search
- ▶ User collaboration while streaming
- ▶ Role-based access control

Future work

- ▶ Performance tests and optimizations with large-scale networks
- ▶ User collaboration with intelligent conflict resolution
- ▶ Performance test framework

References

- [1] D. R. Monismith Jr., 2010, The Uses of the Slime Mold Lifecycle As a Model for Numerical Optimization. *Oklahoma State University*.
- [2] V. Šešum-Čavić, E. Kühn, 2013, Peer-to-Peer Lookup Based on Slime Mold Intelligence. *TU-Vienna, E185/1, SBC-Group, Technical Report*.