HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Master Thesis:
Systematic Benchmarking of a fully Distributed Communication System

Iakov Arkadjevic Dlikman

Institut für Informatik
Heinrich-Heine-Universität Düsseldorf

11.7.2018

Table Of Contents

Background

Problem Statement

Peer-to-Peer-Framework LibreSocial

Experimental Work and Results
- Availability
- Scalability
- Robustness
- Further Analysis

Conclusion and Recommendations

References

Background

Main challenge for Client-Server Systems

- Single Point of Failure

Main Goals for Distributed Systems according to Tanenbaum [1] are

- Openness, Scalability

...

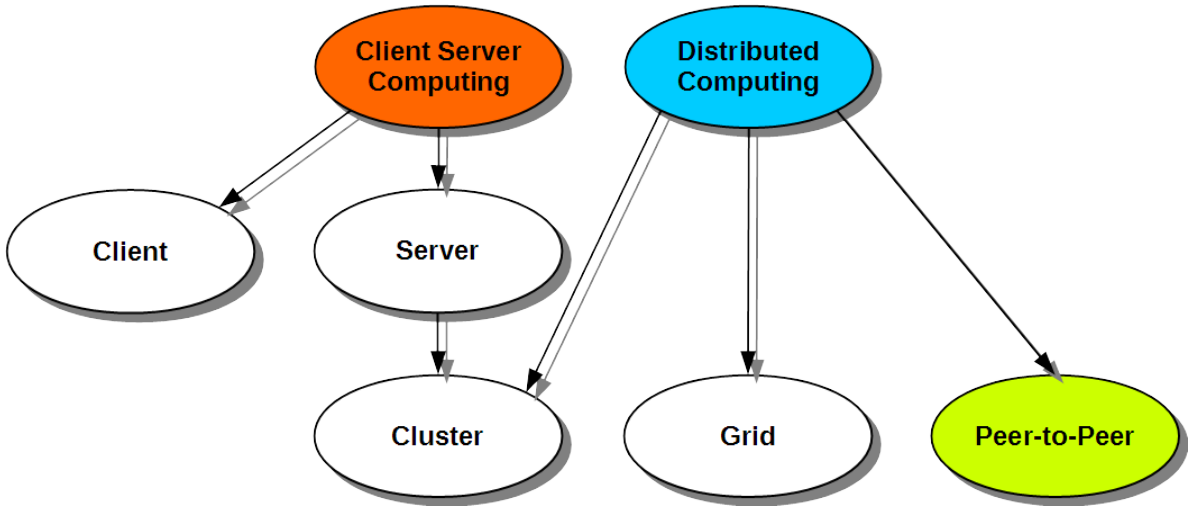


Figure: Distributed Systems Overview, based on [2]

P2P-Systems according to Leof [3] have specific characteristics such as

- Dynamic Population, Dynamic Topology

...

Problem Statement and Goals of this Work

Given
P2P-Framework, Distributed Application LibreSocial, Small-Scale Local Area Network.

Problem
Find capabilities and limitations of P2P-Framework in terms of quality of service.

We focus on three aspects of quality of service

- Availability
- Scalability
- Robustness

Peer-to-Peer-Framework and LibreSocial - Architecture

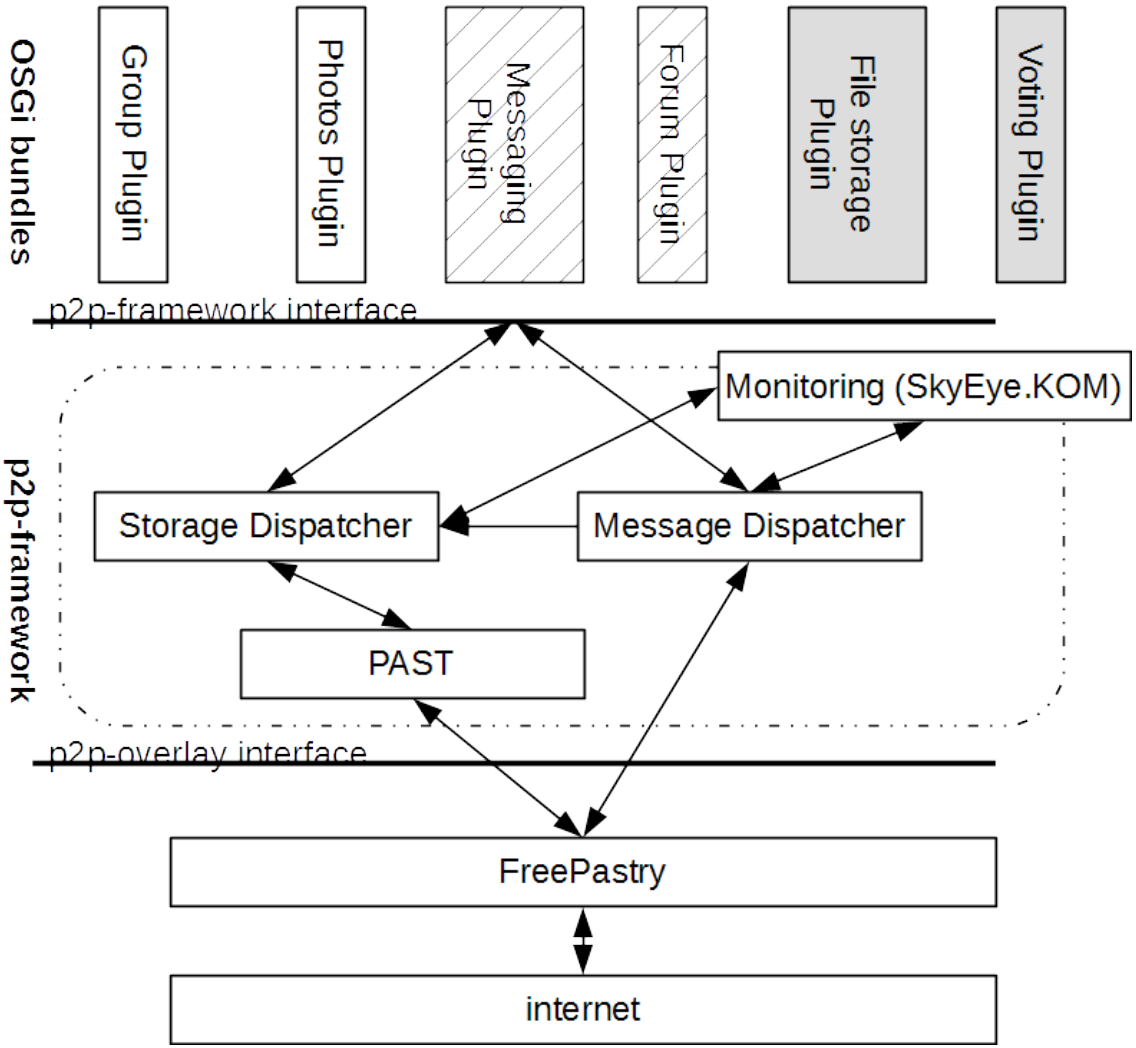


Figure: LibreSocial and P2P-Framework Architecture, based on [4]

P2P-Framework and LibreSocial - Metrics

We define the metrics as a measure for quality of service.

Metric Name	Summary	Calculation
Availability	probability of successful file transfer	$p_A = \frac{\#successfullTransfers}{\#requestedTransfers}$
Robustness	probability of stable availability or download rate given peer dynamics	$\mathcal{P}(p_A = 99\%)_{\|peer_dynamics=model_A}$
Scalability	probability of stable availability or download rate given population growth	$\mathcal{P}(p_A = 99\%)_{\|population_growth=16\%}$

Table: Metric Summary and Calculation, according to Kwok [5]

P2P-Framework and LibreSocial - Monitoring

Tree-based Monitoring Protocol

- based on ID peers join Monitoring Tree
- send measurements to parent node
- received aggregated results from parent node

=> Every peer in the network gains sight of entire network

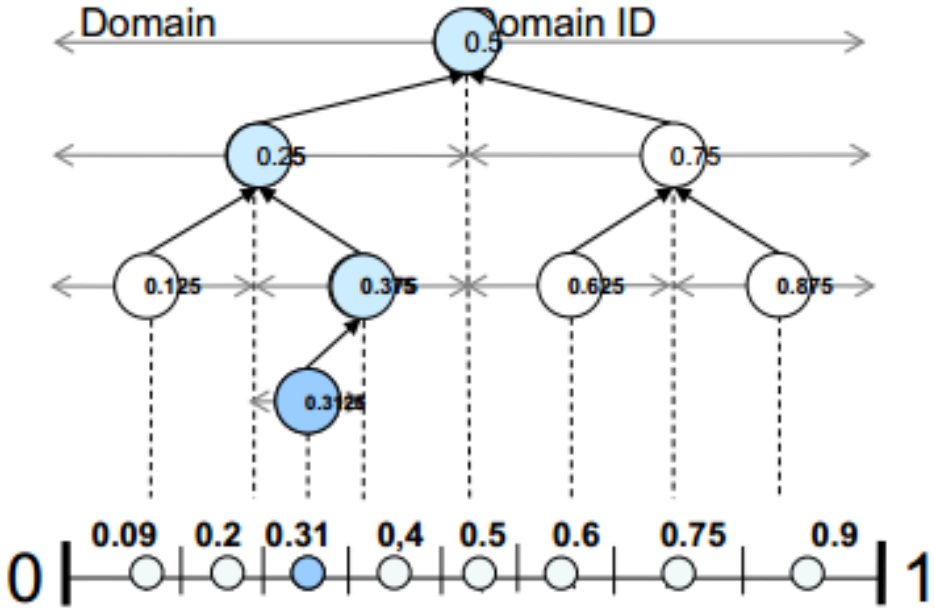


Figure: Tree-Based Monitoring

Experimental Work and Results



For the discussion of the results we choose the following pattern

- **Scenario** of peer behavior in the network
- description of **Peer-Dynamics** and **Population-Growth** for scalability and robustness
- **Metric** for the specific aspect
- **Evaluation Results** in form of graphs and tables

At the end we add further results about the P2P-Framework.

Availability - Scenario

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Listing 1: availability-scenario

```
# ...
# STORAGE impact
# ------------------------
# PHOTO plugin
#
4x TestCaseCreatePhotoAlbum 2m
1x Wait 2m
32x TestCaseUploadPhoto 8m
# ...
# ---------------------------------------- 32 min.
#
# GROUPS plugin
#
1x TestCaseCreateGroup 2m
1x Wait 2m
8x TestCaseCreateFolder 2m
# ...
# ---------------------------------------- 58 min.
#
# ...
```

Availability - Metrics

We observe

- rate of successful actions
- compare them with the total number of initiated actions

$$p_A = \frac{\#successfullTransfers}{\#requestedTransfers} \tag{1}$$

Storage Example

For storage we observe

- Photos plugin
- Groups plugin

$$availability_{photos} = \frac{\#uploadedphotos}{\#initiateduploads} \tag{2}$$

$$availability_{groups} = \frac{\#uploadedfiles}{\#initiateduploads} \tag{3}$$

Availability - Evaluation Results I

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Storage Availability

Storage availability drops with rising number of action repetitions.

photos								
action	time	metric	A	B	C	avail.A	avail.B	avail.C
create-photo-album	40-42m	created-albums	4	8	16	100%	100%	81%
upload-photo	44-52m	created-albums	32	64	128	100%	70%	34%
view-own-albums	54-56m	view-own-albums	4	8	16	100%	100%	88%
delete-photo	66-68m	created-albums	4	8	16	100%	100%	63%
						Σ 100%	Σ 92.5%	Σ 66.5%
groups								
create-group	70-72m	create-public-group	1	2	4	100%	100%	100%
create-folder	74-76m	create-folder	8	16	32	88%	63%	31%
upload-file	78-86m	add-files	32	64	128	94%	41%	28%
delete-file	88-90m	deleted-item	4	8	16	75%	NA	50%
view-group	92-94m	view-group	1	2	4	200%	0%	250%
						Σ 89.25%	Σ 68%	Σ 52.25%

Table: Storage Availability Results

Availability - Evaluation Results II

Messaging Availability

Messaging availability drops as well.
Messaging is more stable than storage, in our setup.

messaging action	time	metric	A	B	C	avail.A	avail.B	avail.C
send-message	20-22m	send-msg	8	16	32	100%	94%	50%
view-outbox	23-25m	view-outbox	4	8	16	100%	100%	56%
view-inbox	26-28m	view-inbox	4	8	16	100%	100%	54%
						Σ 100%	Σ 98%	Σ 54%
forum								
create-thread	30-32m	create-thread	4	8	16	25%	0%	0%
comment-thread	33-35m	created-posts	4	8	16	100%	100%	56%
view-thread	36-38m	view-thread	4	8	16	100%	100%	56%
						Σ 100%	Σ 100%	Σ 56%

Table: Messaging Availability Results

Scalability - Scenario

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Listing 2: internal-scenario

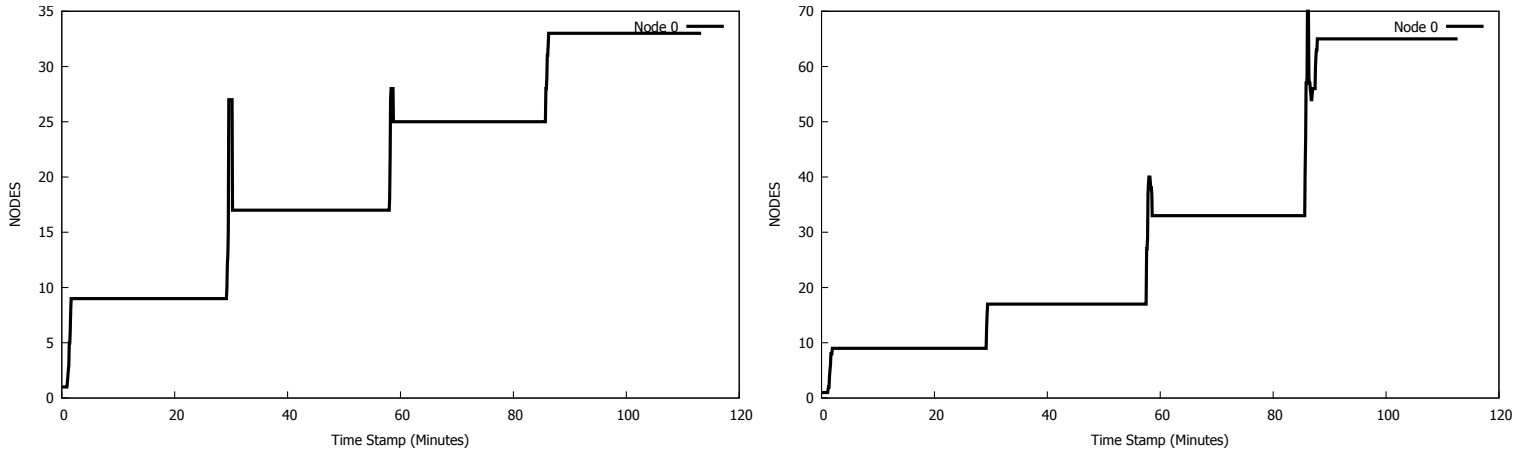
```
# MESSAGING impact
# ----------------------
8x TestCaseSendMessage 2m
1x Wait 1m
# ...
#
# STORAGE impact
# ----------------------
32x TestCaseUploadPhoto 8m
1x Wait 1m
# ...
```

In addition to the peer scenario

- Population Growth

New application instances start on physical machines.

Scalability - Population Growth



((a)) Linear ((b)) Exponential

Figure: Population Growth

Linear Growth Model

- each time frame - constant number of peers join

Exponential Growth Model

- each time frame - number of peers grows exponentially

The maximum number of peers is limited by the physical setup (64).

Scalability - Metrics

Therefore, to measure scalability we observe if the expected availability is met given a certain model of population growth.
We observe

- does system meet expected availability?
- effect of population growth on availability

Scalability, as defined by Kwok [5].

$$\mathcal{P}(p_A = 99\%)_{|population_growth=16\%} \tag{4}$$

Scalability - Evaluation Results I

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

LA			
#peers	minute	storage.avail	message.avail
8	0-30m	100%	100%
16	30-60m	94%	38%
24	60-90m	63%	63%
32	90-120m	35%	100%
		Σ 73%	Σ 75%
ref. to exp. avail.		Π 73%	Π 84%

LC			
#peers	minute	storage.avail	message.avail
8	0-30m	34%	34%
16	30-60m	35%	47%
24	60-90m	20%	41%
32	90-120m	6%	44%
		Σ 24%	Σ 42%
ref. to exp. avail.		Π 71%	Π 84%

Table: Linear Population Growth

Scalability - Evaluation Results II

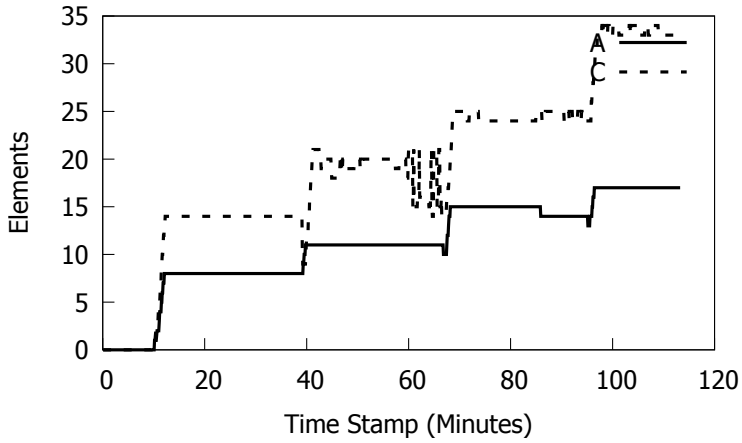
HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

EA			
#peers	minute	storage.avail	message.avail
8	0-30m	91%	100%
16	30-60m	91%	100%
32	60-90m	78%	50%
64	90-120m	84%	75%
		Σ 86%	Σ 81%
ref. to exp. avail.		Π 86%	Π 81%

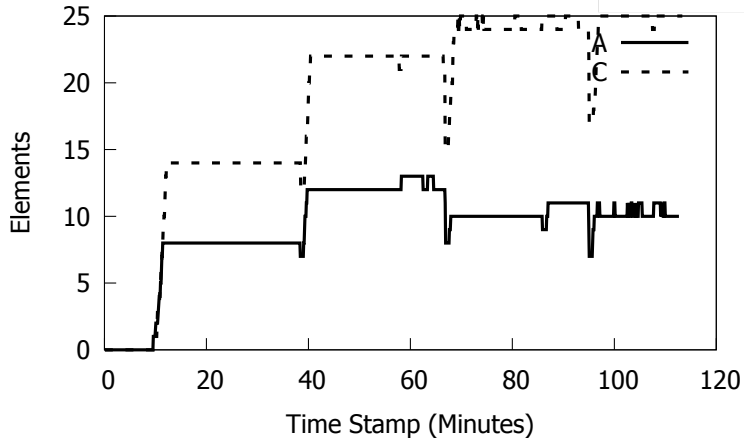
EC			
#peers	minute	storage.avail	message.avail
8	0-30m	35%	47%
16	30-60m	26%	47%
32	60-90m	30%	34%
64	90-120m	28%	34%
		Σ 30%	Σ 40%
ref. to exp. avail.		Π 88%	Π 80%

Table: Exponential Population Growth

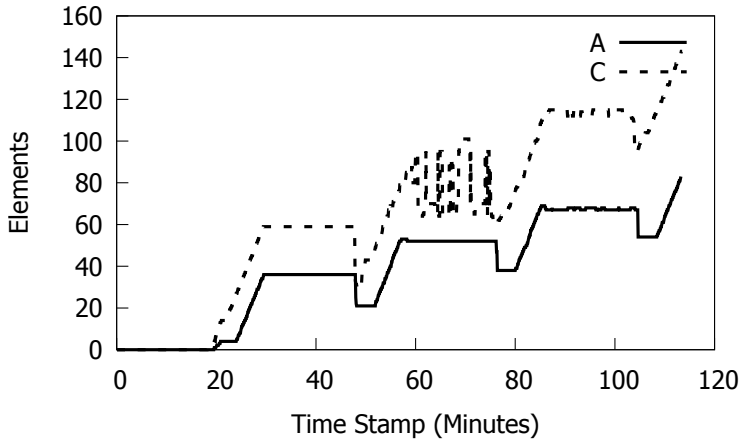
Scalability - Evaluation Results III



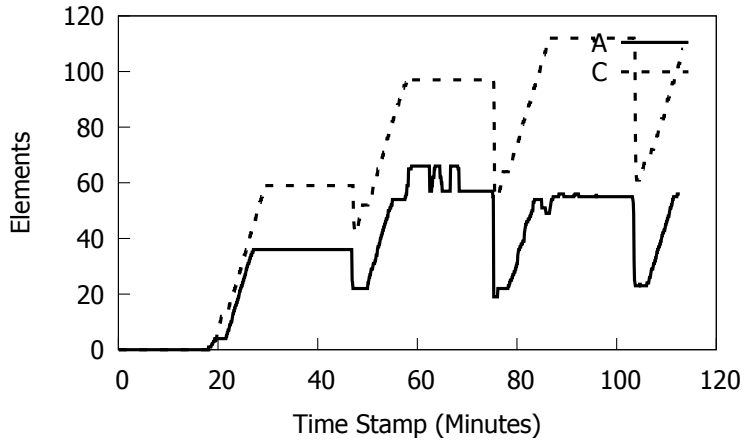
((a)) Linear Growth



((b)) Exponential Growth



((c)) Linear Growth



((d)) Exponential Growth

Figure: Messaging and Storage Scalability

Scalability - Evaluation Results IV

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Scalability

Messaging and Storage **scale up** for exponential growth.

Linear growth leads to **unexpected behavior**.

Scaling experiments is **limited by physical setup**. High Performance Computing allows up to 128 physical machines. Total number of applications can rise to 1024 or above.

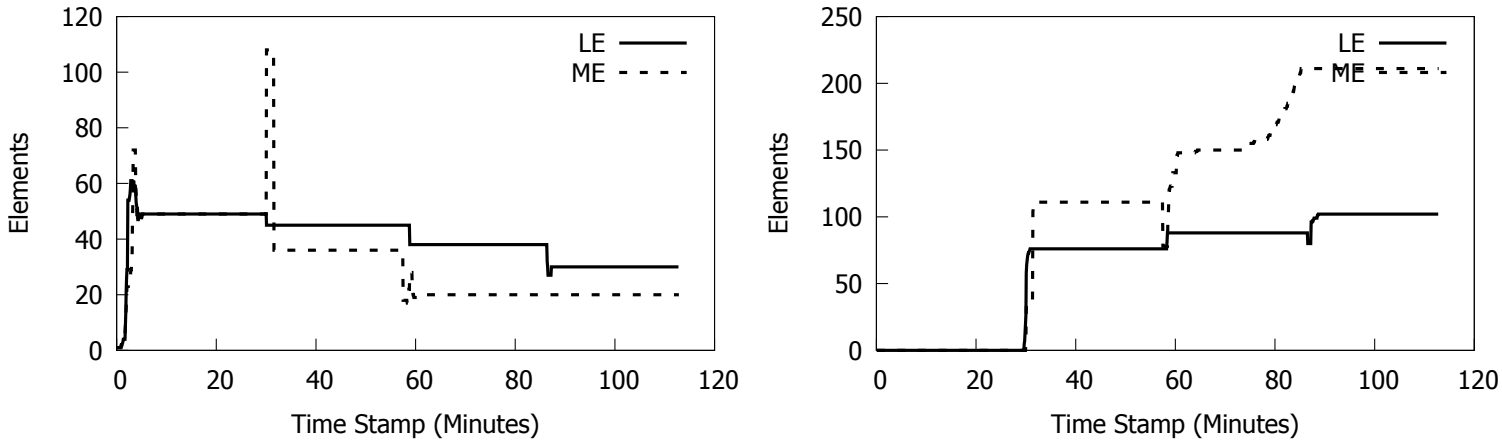
Robustness - Scenario

Listing 3: global-scenario-robustness

```
0m      48 nodes join
5m      nodes send friend requests
10m     nodes start local scenario
15m
20m
25m
30m     8 (16) nodes leave; 8 (16) nodes join
35m     nodes send friend requests
40m     nodes start local scenario
45m
50m
#       ...
```

Robustness - Peer Dynamics

In addition to peer behavior we add peer dynamics.



((a)) Nodes Count ((b)) Lost Messages

Figure: Light (LE) and Medium (ME) Churn

Given 48 online peers

- percentage of churn in the light model **(LE)**is 16.5%
- this means 8 peers leave and 8 new peers join

Second model varies the percentage

- medium model **(ME)** raises percentage to 33%
- number of peers leaving and joining rises to 16

Robustness - Metrics

We refer to Kwok's [5] definition of robustness.
We observe

- availability
- compare it to availability we expect
- subject to varying churn model

$$\mathcal{P}(p_A = 99\%)_{|peer_dynamics=light_churn} \tag{5}$$

Robustness - Evaluation Results I

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

LE

#peers	minute	storage.avail	message.avail	#lost.msg
48/48	0-30m	91%	100%	0
45/48	30-60m	56%	100%	75
38/48	60-90m	69%	100%	10
29/48	90-120m	84%	75%	15
		Σ 75%	Σ 94%	
ref. to exp. avail.		Π 75%	Π 94%	

ME

#peers	minute	storage.avail	message.avail	#lost.msg
48/48	0-30m	94%	100%	0
38/48	30-60m	60%	100%	112
20/48	60-90m	63%	100%	38
20/48	90-120m	0%	0%	60
		Σ 54%	Σ 75%	
ref. to exp. avail.		Π 54%	Π 75%	

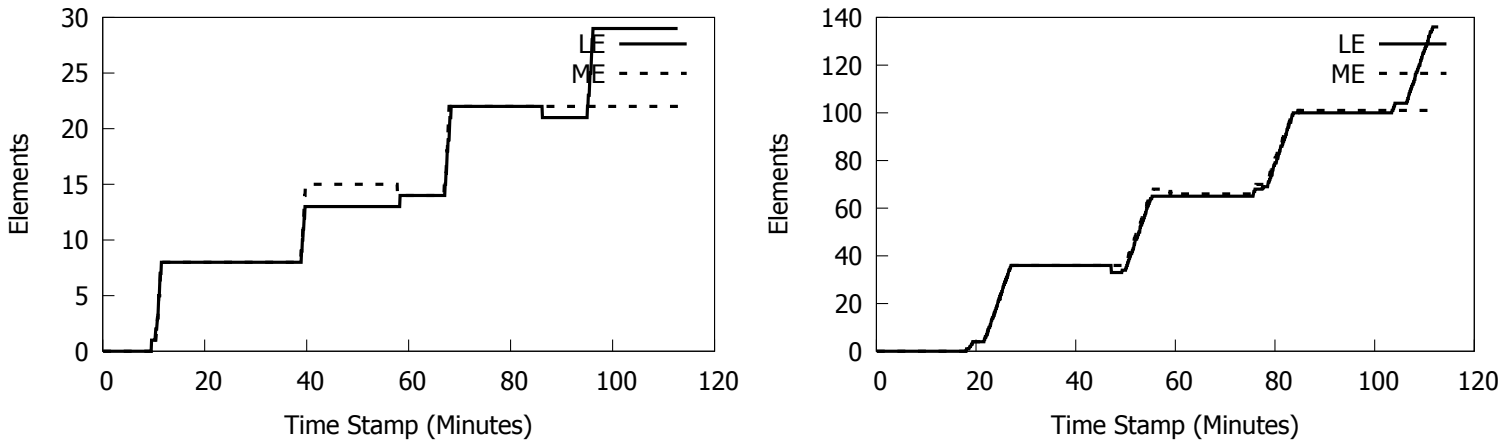
Table: Light and Medium Churn

Robustness - Evaluation Results II

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Robustness

Availability of messaging and storage is **robust** against light churn.
For **medium churn**, system **looses availability** after fourth round.



((a)) Messaging Robustness

((b)) Storage Robustness

Figure: Messaging and Storage during Light (LE) and Medium (ME) Churn

Further Analysis - P2P-Framework Performance

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

	Exp. Outcome			Storage-Dispatcher (KBps)				Message-Dispatcher (KBps)			
Exp. Type	Strg.	Msg.	#Peers	Store-Time	Min	Avg	Max	#Lost-Msg.	Min	Avg	Max
Availability											
A	100%	100%	64	0.05s	0	3	27	0	2	28	1044
B	70%	84%	64	0.06s	0	4	51	0	2	36	877
C	34%	50%	64	0.06s	0	6	86	0	2	38	1035
Σ				0.056s	0	4.3	55	0	2	34	985
Scalability											
LA-start	100%	100%	8	0.09s	1	4	71	0	3	20	86
LA-end	35%	100%	32	0.06s	0	9	55	0	2	35	1773
LC-start	34%	34%	8	0.09s	0	5	21	0	4	31	330
LC-end	6%	44%	32	0.06s	0	15	102	0	2	24	230
EA-start	91%	100%	8	0.09s	0	3	27	0	2	21	267
EA-end	84%	75%	64	0.06s	0	0	16	0	2	34	1142
EC-start	35%	47%	8	0.08s	2	5	24	0	4	27	164
EC-end	28%	34%	64	0.06s	0	10	100	0	2	25	732
Σ				0.073s	0.4	6.4	52	0	2.6	27.1	590.5
Robustness											
LE-start	91%	100%	48	0.4s	0	3	29	0	2	32	709
LE-end	84%	75%	29	0.06s	0	2	19	102	2	26	771
ME-start	94%	100%	48	0.06s	1	3	25	0	2	49	1058
ME-end	0%	0%	20	0.06s	0	1	26	211	2	23	359
Σ				0.145s	0.25	2.3	24.8	78.3	2	32.5	724.3

Table: Performance of Messaging and Storage on P2P-Framework Layer



Further Analysis

Problem Statement

What are the capabilities and limitations of the P2P-Framework?

Conclusion

Quantitative Results for Storage

- Storage of p2p-framework allows data-rates up to **100KBps**
- P2P-Framework **scales up** for exponential peer growth
- It is **robust** against churn for up to **16%**
- With rising number of queries, **storage service limits** overall system performance

Quantitative Results for Messaging

- Messaging of p2p-framework allows data-rates up to **1.7MBps**
- It **scales up** during exponential population growth
- **Robust** for a churn rate up to **16%**
- Rising number of queries shows, **application layer limits** overall system performance

Review

So far we

- introduced P2P-Framework and LibreSocial
- defined problem statement
- introduced means to measure quality of service
- defined aspects of quality of service
  - Availability
  - Scalability
  - Robustness
- presented results from application point of view
- summarized capabilities and limitations of P2P-Framework

How can we proceed with benchmarking?

Conclusion and Recommendations - Outlook

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

High Performance Computing
Scalability measurement is heavily limited by number of physical nodes.
High Performance Computing (HPC)[6] allows up to 128 physical nodes.
Each physical node can carry several LibreSocial instances. Therefore peer number up to 1024 become feasible.

Large-Scale Internet Setup
HPC still offer homogenous local area setup.
A realistic scenario involves Network Address Translation (NAT) trespassing, heterogeneous interconnections with varying latencies, variety of operating systems, and node capacities.
Planetlab.org [7] offers open source, large-scale test bed for realistic internet scenarios.

Main Requirement for Future Benchmarking

HPC and planetlab.org requirer a fully automated conduction of experiments.
Communication with OSGi console of P2P-Framework using Bash is crucial!

References

- A. S. Tanenbaum and M. Van Steen, *Distributed systems: principles and paradigms*. Prentice-Hall, 2007.
- C. Baun and G. Bengel, *Masterkurs Parallele und Verteilte Systeme*, 2015.
- B. Leuf, *Peer to peer: Collaboration and sharing over the internet*. Addison-Wesley Longman Publishing Co., Inc., 2002.
- K. Graffi, “Monitoring and management of peer-to-peer systems,” Ph.D. dissertation, Technische Universität, 2010.
- R. Kwok, *Peer-to-peer computing: Applications, architecture, protocols, and challenges*. Crc Press, 2011.
- HILBERT, “High Performance Computing,” 2018. [Online]. Available: https://wiki.hhu.de/display/HPC/
- J. Turner and P. Crowley, “Supercharging planetlab: a high performance, multi-application, overlay network platform,” *ACM SIGCOMM Computer Communication Review*, 2007.
- P. Weifeng and B. Dheeraj, “Systematic Evaluation of a Decentralized Social Network,” 2015.

Appendix

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

- Regression Test
- Discussion on Qualitative Interpretation of Results
- Tree-Based Monitoring
- Pastry-Routing

Appendix - Regression Test

Regression Test

- total number of peers 7, 54, 56
- scenario as used in previous work by Weifeng [8]

Regression Test Conclusion

- system performance as discussed in previous work [8] **confirmed**
- varying number of **peers per machine** affect overall system performance
- extract a map of plugins, test cases and metrics
- impact of available plugins on storage and messaging service

Choice of Plugins for further testing bases on their impact.

Appendix - Regression Test I - Plugin Map (Excerpt)

Plugin Name	Available Metrics	Test Cases
Filestorage	stored_files_count	deleteFileFromFolder uploadFileInFolder
Forum	created_posts_count created_threads_count thread_views_count views_count	commentForumThread createForumThread viewForum viewForumThread
Groups	added_files_count created_folders_count created_public_group_count deleted_items_count folder_views_count group_joins_count group_leaves_count group_views_count	createGroup leaveGroup viewGroup
Messaging	deleted_messages_count inbox_views_count sent_messages_count outbox_views_count	deleteInboxMessage sendMessage viewInbox viewOutbox
Photos	created_albums_count view_albums_count	createAlbum deleteAlbum uploadPhoto viewFriendsAlbum
Voting	created_votings_count deleted_votings_count get_voting_results_count my_voting_list_views_count votes_count	createVote deleteVote vote

Appendix - Regression Test II - Plugin Impact

	plugins	sys.data.stored [KB]	sys.data.retrieved [KB]	sys.data.rate.down [KBps]	sys.data.rate.up [KBps]
1	messaging	5419	916	77061	11438
2	filestorage	763401	7314	2117369	27920
3	groups	36471	104544	105856	276480
4	photos	13389	143008	33853	730270
5	voting	631	6384	7200	36925
6	forum	703	15803	8750	66052

Table: Storage-Dispatcher Table

	plugins	#sys.msg.total	sys.msg.rate [msg/s]	#peer.msg.total	peer.msg.rate [msg/s]
1	filestorage	86571	332	1406	4
2	forum	15462	254	488	3
3	groups	33856	332	1129	4
4	messaging	13489	233	213	3
5	photos	12968	103	144	1
6	voting	15963	398	207	6

Table: Message-Dispatcher Table

Appendix - Qualitative Interpretation

HEINRICH HEINE
UNIVERSITÄT DÜSSELDORF

Interpretation of quantitative outcome

How to interpret the quantitative outcome

- 50% or 70% for quality of service requirement seems very low
- keep in mind - the testing performed is synthetic
- in a realistic scenario 64 users do not perform exactly the same action at exactly the same time
- previous work discusses probabilistic choice of actions