UNIVERSITÄT PADERBORN
Die Universität der Informationsgesellschaft

Master Thesis

Systematic Evaluation of a Decentralized Social Network

Presented by:

Dheeraj Balabhadruni (6644484)
Weifeng Peng (6628238)

Supervised by:

Jun.-Prof. Dr.-Ing. Kalman Graffi
Prof. Dr. Christian Scheideler

Overview

- Motivation
- Goals
- LifeSocial Architecture
- Testing
- Monitoring
- Evaluation
- Conclusion



Motivation

- Now a days, most of the social networks are centralized and based on client - server architecture.

- Centralized social networks have many disadvantages. For example:
  - Huge maintenance cost.
  - Users are worry about their private information.

- Peer-to-peer(P2P) technology provides the possibility to solve these problems.

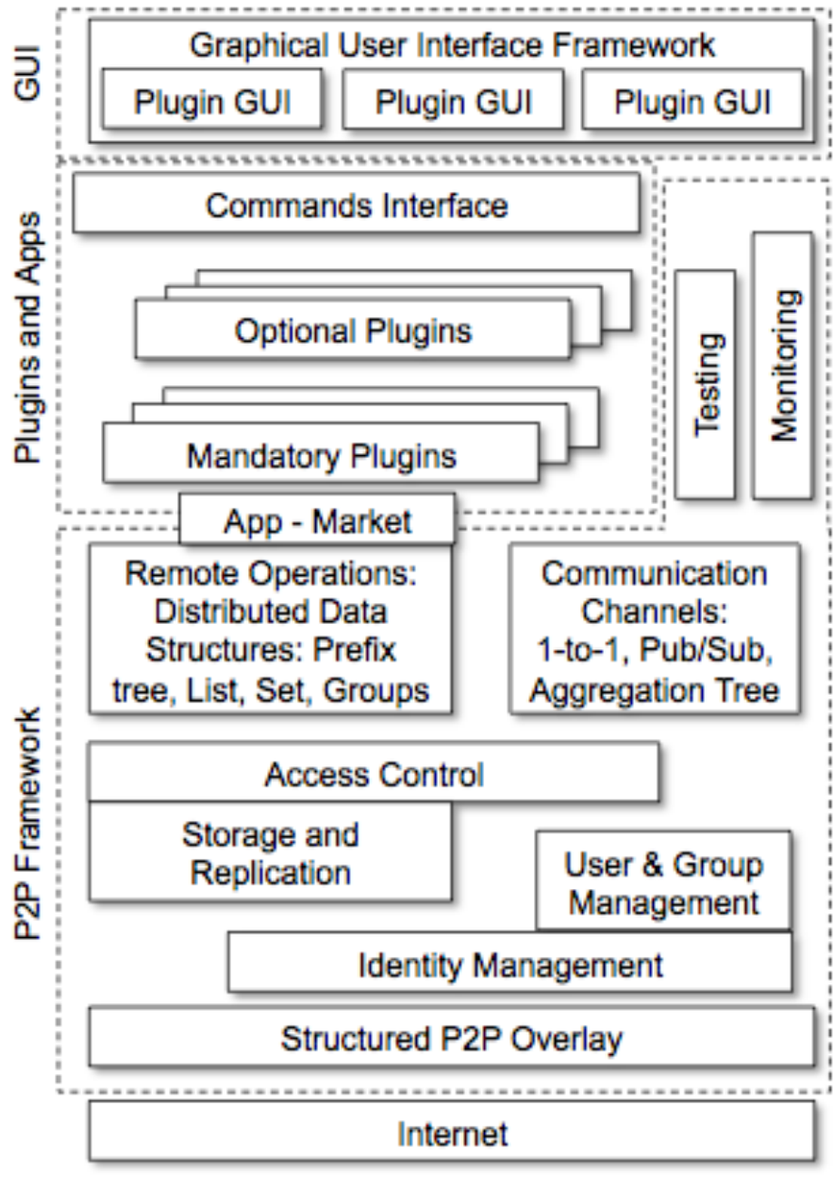
- LifeSocial is a new social network application based on P2P technology.

Goals

- Create Test Cases
- Synthesis Test
- Random Test
- Long Term Test
- Evaluation of Test Results



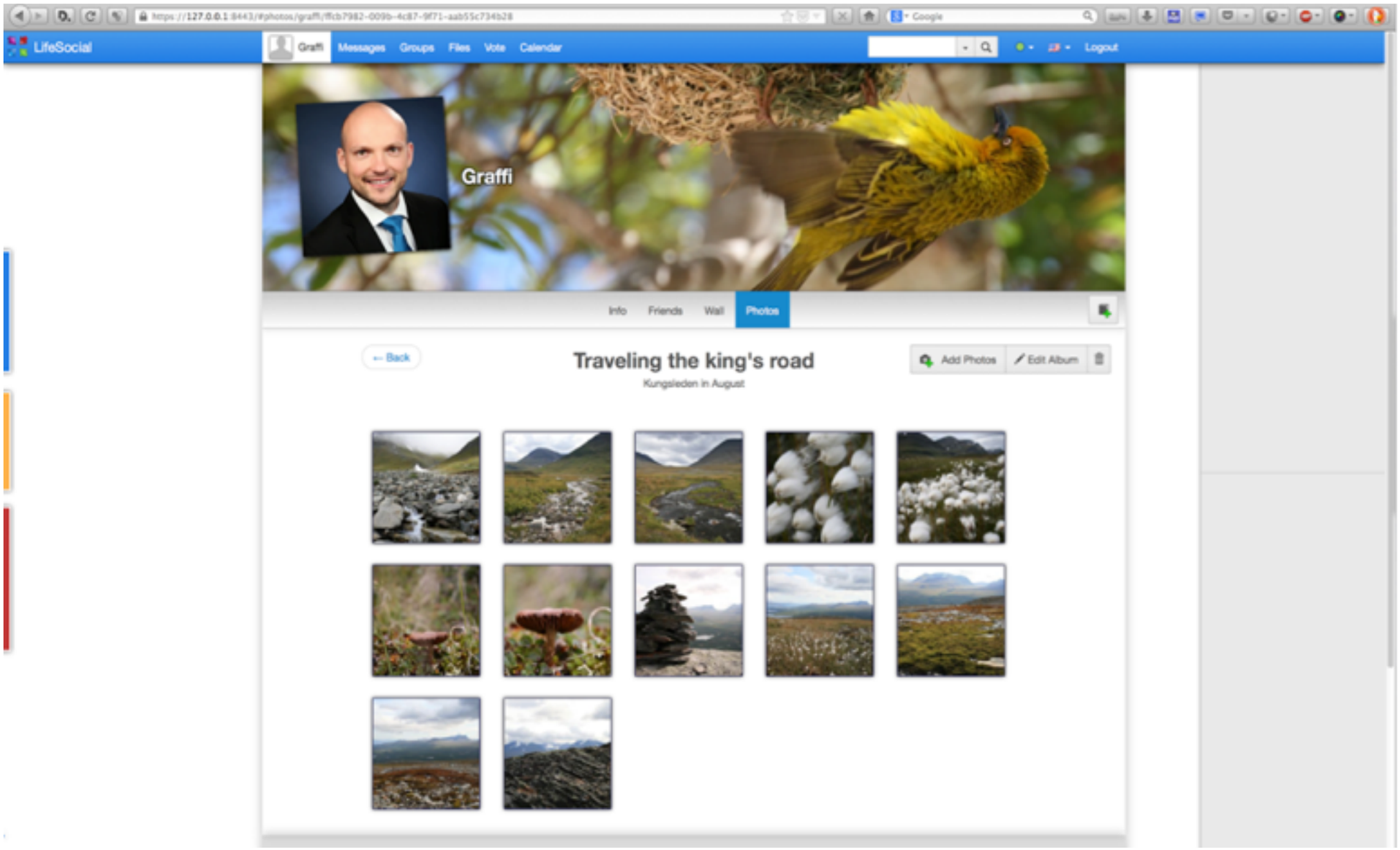
LifeSocial Architecture



Available Plugins

- Login Plugin
- Profile Plugin
- Friends Plugin
- Photos Plugin
- Notification Plugin
- Message Plugin
- Wall Plugin
- Chat Plugin
- Vote Plugin
- Groups Plugin
- Files Plugin
- Calendar Plugin
- Search Plugin
- Audio/Video Chat plugin

LifeSocial GUI



Testing

- Prerequisites
  - Resource Files
- Requirements
  - Master Node
  - Slave Node
  - Test Plan
- Test Case Structure
- Test Case Implementation
- Test Cases
- Metrics
- Social Parameters

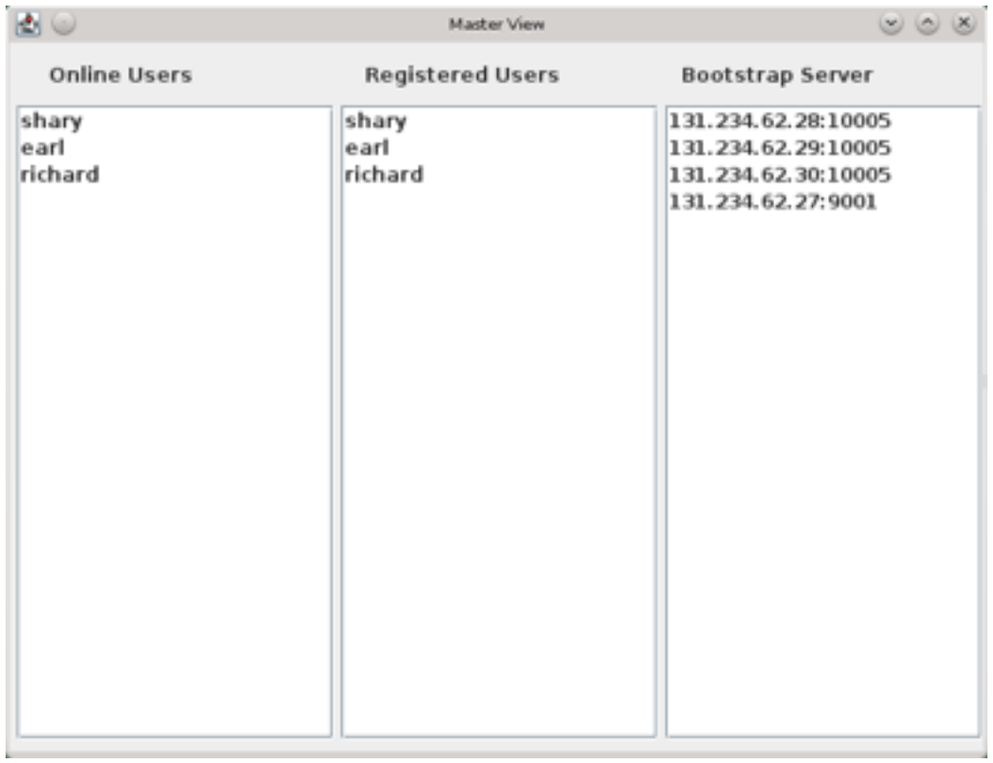
Prerequisites - Resource Files

File Name	Description
config.txt	Configure the IP address of the master node Enable or disable test plugin Test plan path configuration
cityNames.txt	Contains available city names.
countryNames.txt	Contains available country names.
groupNames.txt	Contains available group names.
interests.txt	Contains available user interests.
loginNames.txt	Contains available login names.
messages.txt	Contains available messages.
onlineDuration.txt	Contains online duration distribution.
startScript.txt	Defines a sequence of actions need to execute when joining the network in long term test.
subjects.txt	Consists of available message subjects.
pictures.txt	Contains available pictures.
testCaseRegistry.txt	Contains class names of all used and implemented test cases.

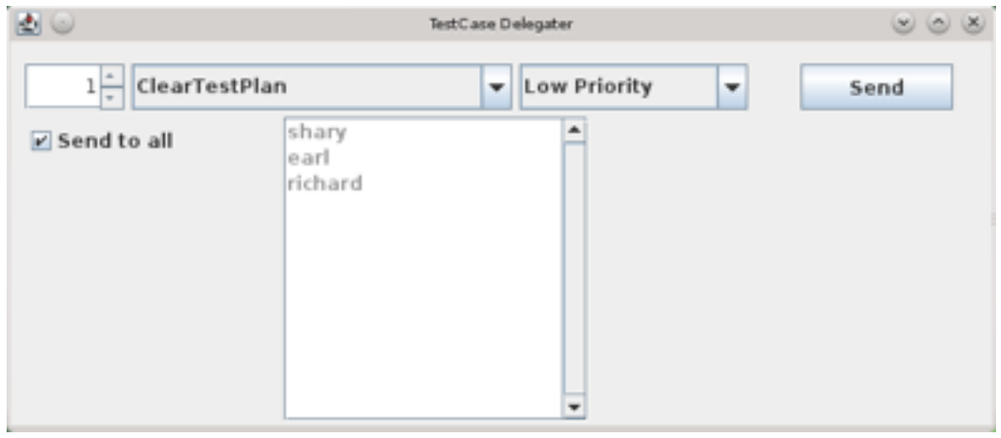
Requirements - Master Node

- Master node is unique in network.
- Started first in the test.
- Sends the command to all the slave instances in the network.
- Contains two types of views.
  - Master View
  - Delegater View

Requirements - Master Node Views



Master View



Delegater View

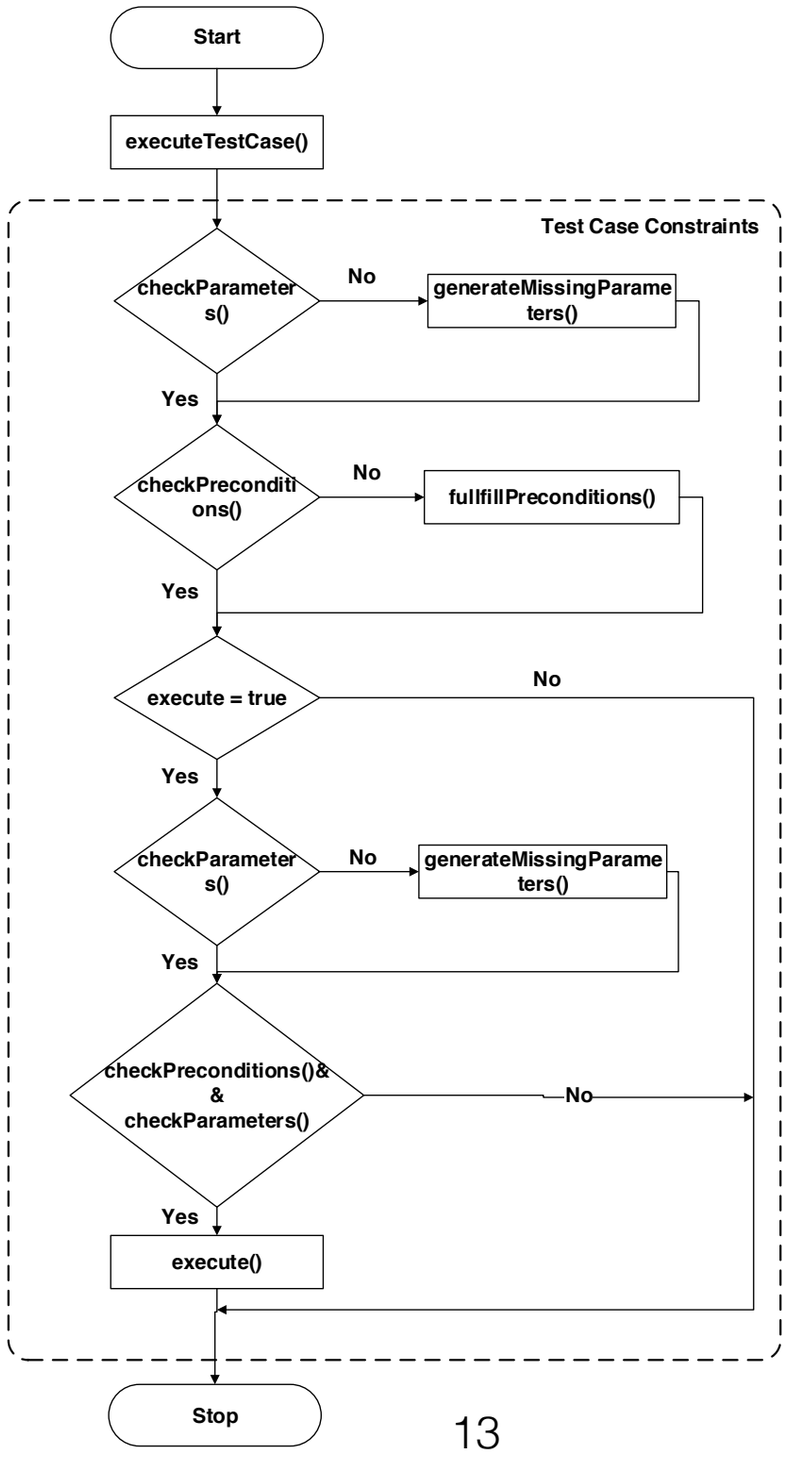
Requirements - Slave Node & TestPlan

- Slave node executes the command sent by the master node.

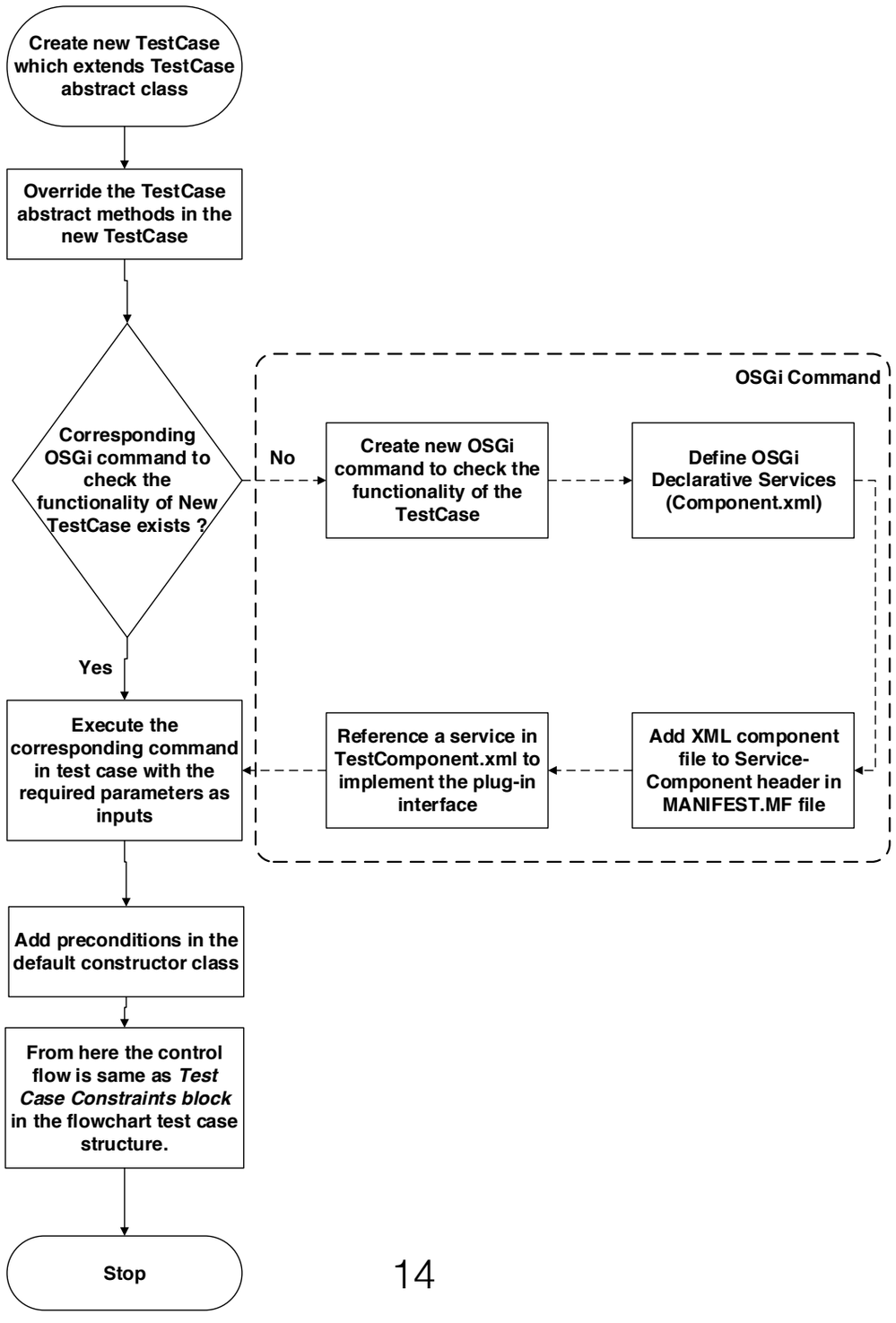
- Test Plan contains the the list of Test Cases to execute.

- Execution sequence is same as the order mentioned in the Test Plan.

Test Case Structure



Test Case Implementation



Test Cases

- A lot of plugins have been modified last year
  - Functions abandoned
  - Parameters changed
  - Plugins redesigned

- Status BEFORE:
  - Valid test cases: 7
  - Invalid test cases: 19

Test Case	Status
TestCaseChangeProfileInformation	valid
TestCaseChangeProfilePicture	valid
TestCaseViewOwnProfile	invalid
TestCaseViewFriendsProfile	invalid
TestCaseRegister	valid
TestCaseLogin	valid
TestCaseLogout	valid
TestCaseLogoutLogin	invalid
TestCaseSendWallPost	invalid
TestCaseCommentWallPost	invalid
TestCaseViewOwnWall	invalid
TestCaseViewFriendsWall	invalid
TestCaseCreatePhotoAlbum	valid
TestCaseLoadPhoto	invalid
TestCaseUploadPhoto	invalid
TestCaseViewFriendsPhoto	invalid
TestCaseSendFriendRequest	valid
TestCaseSearchCity	invalid
TestCaseSearchCountry	invalid
TestCaseSearchUser	invalid
TestCaseSendChatMessage	invalid
TestCaseViewChatHistory	invalid
TestCaseSendMessage	invalid
TestCaseForum	invalid
TestCaseStoreFile	invalid
TestCaseViewFriendsGroups	invalid

Test Cases

- During the thesis:
  - Deleted test cases of the abandoned functionalities
  - Modified other invalid test cases
  - Created test cases for new features

- Status AFTER:
  - More than 50 available test cases
  - Covered majority of functionalities

Metrics

- A metric is a measurement scale
  - Monitoring a certain attribute
  - Periodically collected

- The statistics of a metric can be plotted into a graph for better evaluation.

- Metric name consists of:
  - Statistic scope
  - Collection location
  - Measured object
  - Aggregation strategy (sum, max, mean, min)

- Example:
  GLOBAL_STORAGEDISPATCHER_NUM_STORED_REPLICATIONS_COUNT

Metrics

- Already lots of metrics existed covering many aspects of LifeSocial network.

- However, about half social metrics were missing or invalid.

- Social metrics monitor user actions
  - to evaluate test result
  - to analyze user behavior in social network

Current Social Metrics for LifeSocial Plugins

Plugin	Social Metrics
Calendar	created events; deleted events; modified events; view calendar
FileStorage	stored files
Forum	created posts; created threads; view forum; view forum thread
Friend	accepted friendship request; sent friendship request; friends list size
Group	added files; created folders; created private groups; created public groups; deleted items; invite user to group; join group; leave group; view folder; view group
LiveChat	chat channels; send chat messages; leave chat channel
Login	registration ; login; logout
Messaging	deleted inbox messages; deleted outbox messages; sent messages ; view inbox; view outbox
Photos	created albums ; deleted albums; uploaded photos; view friend album list; view own album list
Profile	created basic profiles; get own profile; request user profile
Search	search
Voting	created votings; deleted votings; get voting result; view my voting lists; vote action
Wall	view wall; wall comments; wall entries;

Setup Parameters

- Various parameters directly influence on test result.

- Different setup for different purpose

- 3 types of parameters:
  - Environment parameters
  - Social parameters
  - LifeSocial parameters

Setup Parameters

- Environment parameters:
  - –Number of peer
  - –Peer performance
  - –Peer bandwidth
  - –Operating system

- Social parameters:
  - –Churn
  - –Application load

Setup Parameters

- LifeSocial parameters:

```
# auto storage deletion before accessing into network (1 = enabled,
# 0 = disabled)
autoDeleteStorage = 0

# enables the heterogeneity awareness (1 = enabled, 0 = disabled)
enableHeterogeneityAwareness = 0

# size of the heterogeneitycache in MB
heterogeneityCacheSize = 1

# timeout for a canyouprovidemessage in ms
CanYouProvideTimeOut = 5000

# maximal number of providers for each data object
providerListLength = 1

# number of replications of a data object in the system
replicationFactor = 3

# enables the object size calculation for the monitoring
# (1 = enabled, 0 = disabled)
objectSizeCalculationEnabled = 0

# enables the random selection mode (1 = enabled, 0 = disabled)
randomSelectionModeEnabled = 0

# enables the measuring data function, the data measured located in Plugin,
# Framework levels (1 = enabled, 0 = disabled)
measuringData = 1
```

Setup Parameters

```
# enables the collecting of social metrics, such as friend requests
# (1 = enabled, 0 = disabled)
socialMetricCollection = 1

# auto start monitoring (1 = enabled, 0 = disabled)
autoStartMonitoring = 1

# in monitoring, how many rounds you want to combine
combineParameter = 3

# enables the auto friend request accepted (1 = enabled, 0 = disabled)
autoFriendRequestAccept = 0

# enables the auto live chat invitation accepted (1 = enabled, 0 = disabled)
autoLiveChatInvitationAccept = 1

# enables the auto voting user invitation accepted (1 = enabled, 0 = disabled)
autoVotingInvitationAccept = 1

# only create GNUPlot monitoring data file (1 = enabled, 0 = disabled)
onlyGnuplot = 1

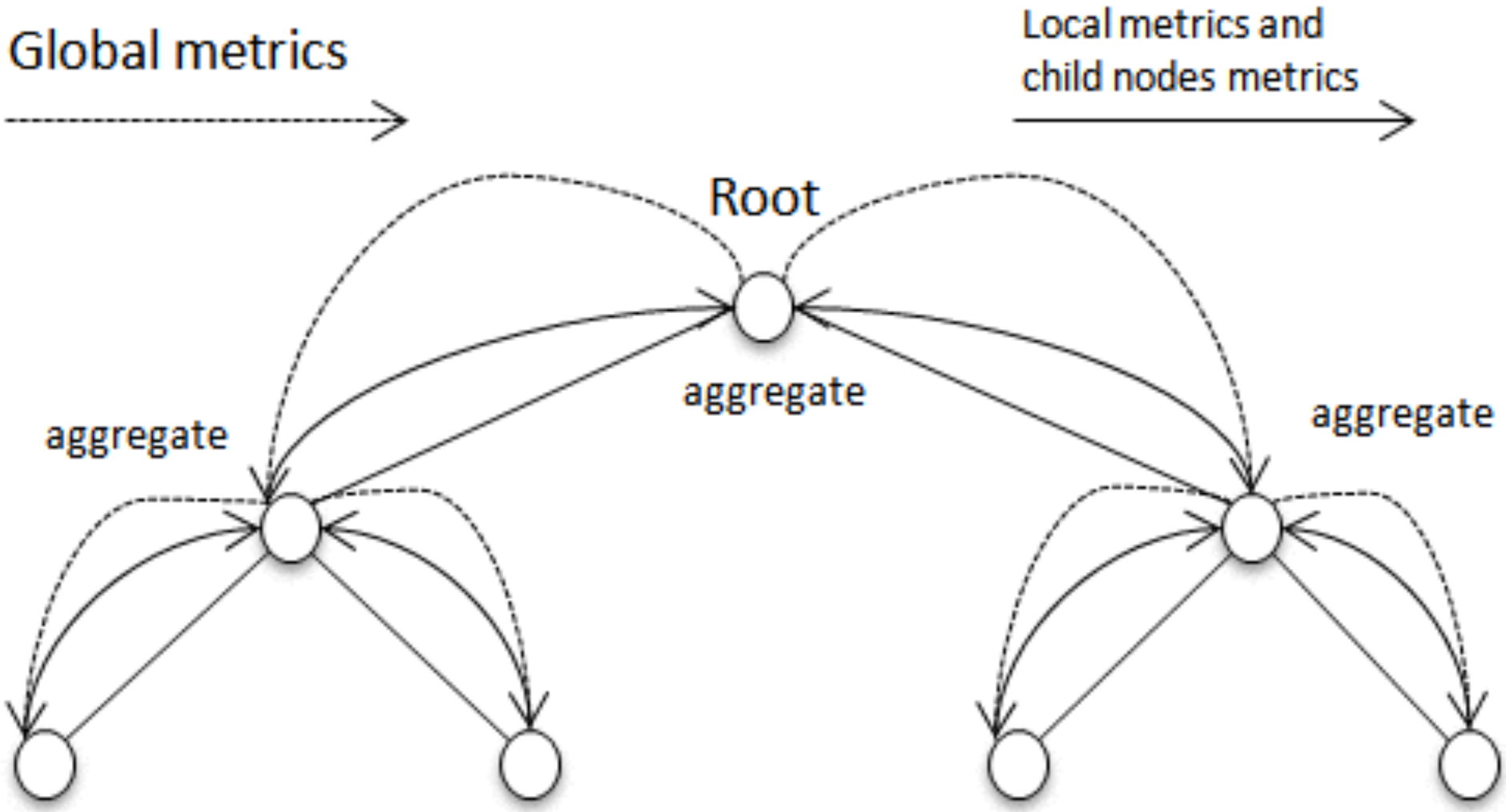
# visualization of monitored data in runtime, this function will keep the
# memory usage increasing! (1 = enabled, 0 = disabled)
monitorDataVisual = 0

# url to the bulletin board
bulletinURL = a url
```

Monitoring

- Goal of monitoring is to
  - Gather information from the peers.
  - Provide constant quality in P2P system.
- SkyEye.KOM mechanism is used to monitor the whole P2P network.

SkyEye.KOM


Global metrics
Local metrics and
child nodes metrics
Root
aggregate
aggregate
aggregate

Synthesis Test

- Goal:
  - to examine performance of LifeSocial under simple condition
    - With test plan
    - No logout

- Test machine (x20)
  - Processor : Intel i7-2600
  - RAM : 8 GB
  - Operating System : openSUSE 8.32.5-x86_64
  - Network Interface : Gigabit Ethernet
  - Java VM : 1.7.0_51 build 24.45-b08

Synthesis Test

- Deployment:
  - Eclipse Kepler (for RCP and RAP developers)
  - Smartgit

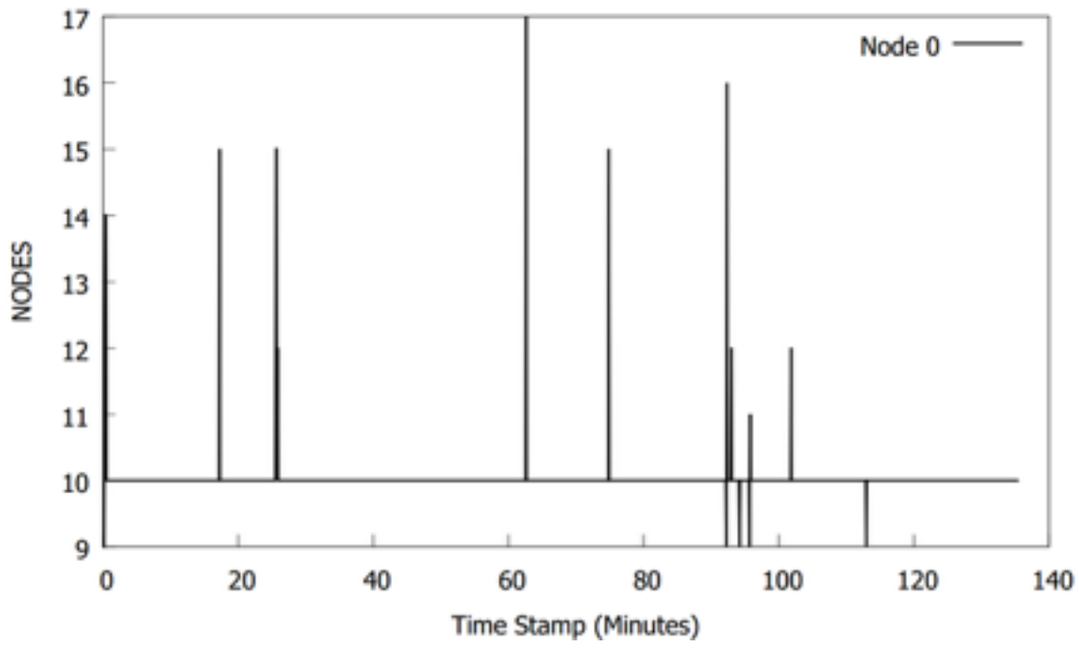
- Configurations:
  - 10 and 20 test machines
  - Synchronize test plan with Dropbox link
  - Set LifeSocial parameters (measuringData, socialMetricCollection, autoStartMonitoring and autoaccept etc.)

Synthesis Test - Test Plan

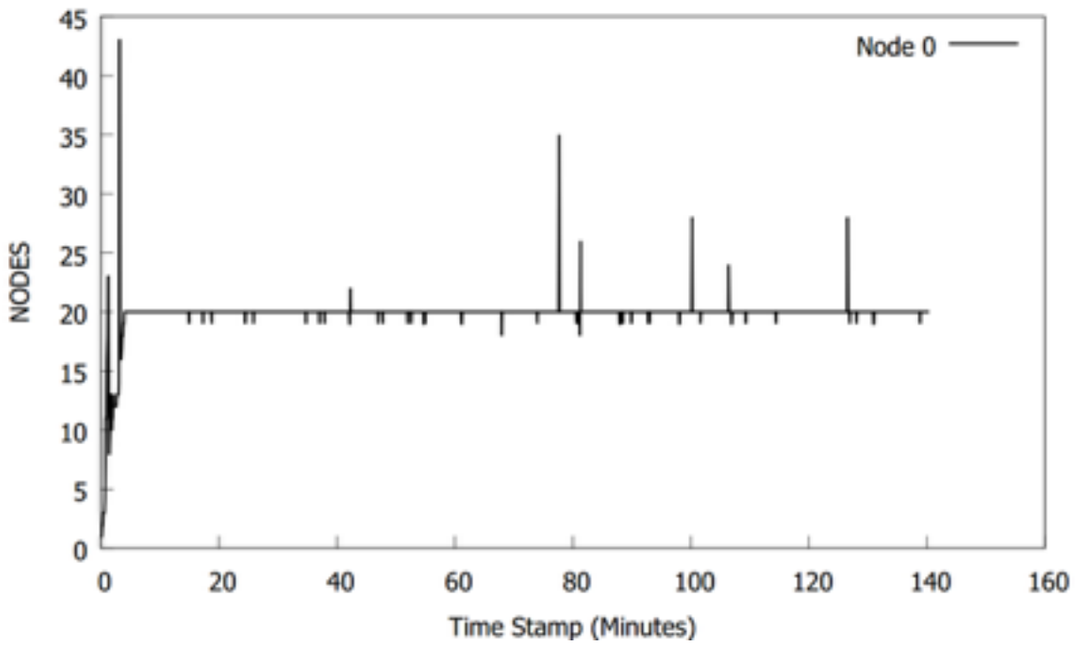
1x TestCaseChangeProfilePicture 1m
1x Wait 1m
1x TestCaseChangeProfileInformation 1m
1x Wait 1m
1x TestCaseViewOwnProfile 1m
1x Wait 1m
1x TestCaseViewFriendsProfile 1m
1x Wait 1m
5x TestCaseSendWallPost 2m
1x Wait 1m
5x TestCaseCommentWallPost 2m
1x Wait 1m
5x TestCaseViewOwnWall 2m
1x Wait 1m
5x TestCaseViewFriendsWall 2m
1x Wait 1m
10x TestCaseSendMessage 3m
1x Wait 1m
5x TestCaseViewOutboxMessage 2m
1x Wait 1m
5x TestCaseViewInboxMessage 2m
1x Wait 1m
2x TestCaseDeleteInboxMessage 1m
1x Wait 1m
1x TestCaseSendMultiChatInvitation 1m
1x Wait 1m
10x TestCaseSendMultiChatMessage 3m
1x Wait 1m
1x TestCaseLeaveMultiChatMessage 1m
1x Wait 1m
5x TestCaseSearch 2m
1x Wait 1m
1x TestCaseViewCalendar 1m
1x Wait 1m
5x TestCaseCreateCalendarEvent 2m
1x Wait 1m
2x TestCaseEditCalendarEvent 1m
1x Wait 1m
2x TestCaseDeleteCalendarEvent 1m
1x Wait 1m
1x TestCaseGetMyVotings 1m
1x Wait 1m
10x TestCaseCreateVote 3m
1x Wait 1m
10x TestCaseVote 3m
1x Wait 1m
5x TestCaseGetVotingResults 2m
1x Wait 1m
2x TestCaseDeleteVote 1m
1x Wait 1m
5x TestCaseCreatePhotoAlbum 2m
1x Wait 1m
50x TestCaseUploadPhoto 10m
1x Wait 1m
5x TestCaseViewOwnAlbums 2m
1x Wait 1m
5x TestCaseViewFriendsAlbums 2m
1x Wait 1m
10x TestCaseViewFriendsPhoto 3m
1x Wait 1m
5x TestCaseDeletePhoto 2m
1x Wait 1m
1x TestCaseDeleteAlbum 1m
1x Wait 1m
2x TestCaseCreateGroup 1m
1x Wait 1m
1x TestCaseViewForum 1m
1x Wait 1m
5x TestCaseCreateForumThread 2m
1x Wait 1m
5x TestCaseCommentForumThread 2m
1x Wait 1m
5x TestCaseViewForumThread 2m
1x Wait 1m
10x TestCaseCreateFolder 3m
1x Wait 1m
2x TestCaseViewGroup 1m
1x Wait 1m
2x TestCaseRenameFolder 1m
1x Wait 1m
50x TestCaseUploadFileinFolder 10m
1x Wait 1m
5x TestCaseViewFolder 2m
1x Wait 1m
1x TestCaseLeaveGroup 1m
1x Wait 1m

Synthesis Test - Results Evaluation

- Only the second round

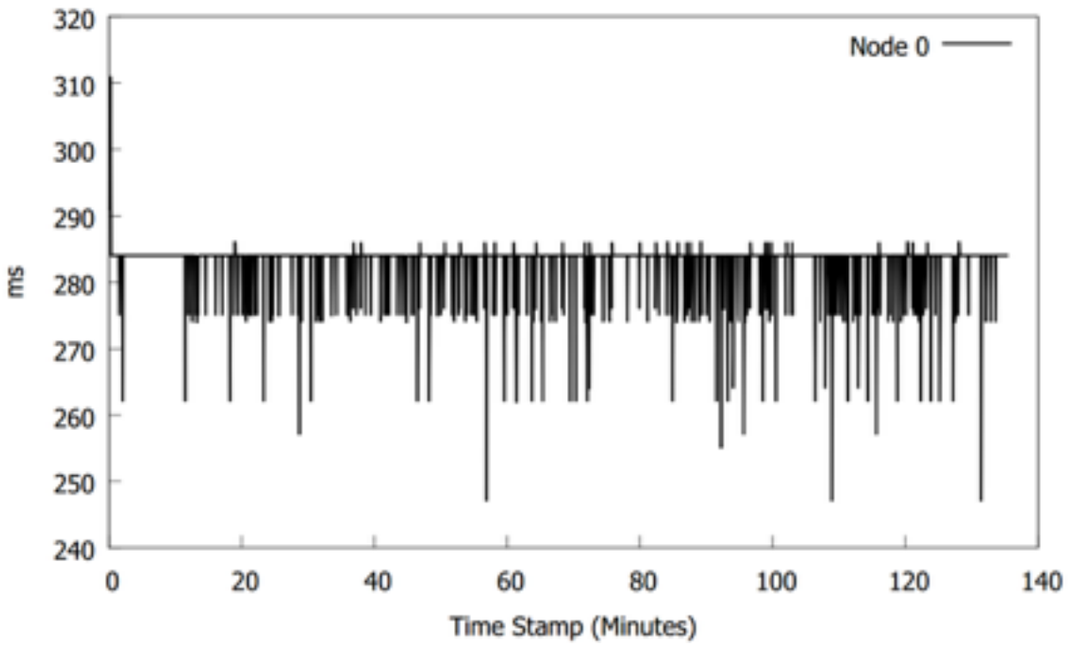


Global Nodes – 10 instances

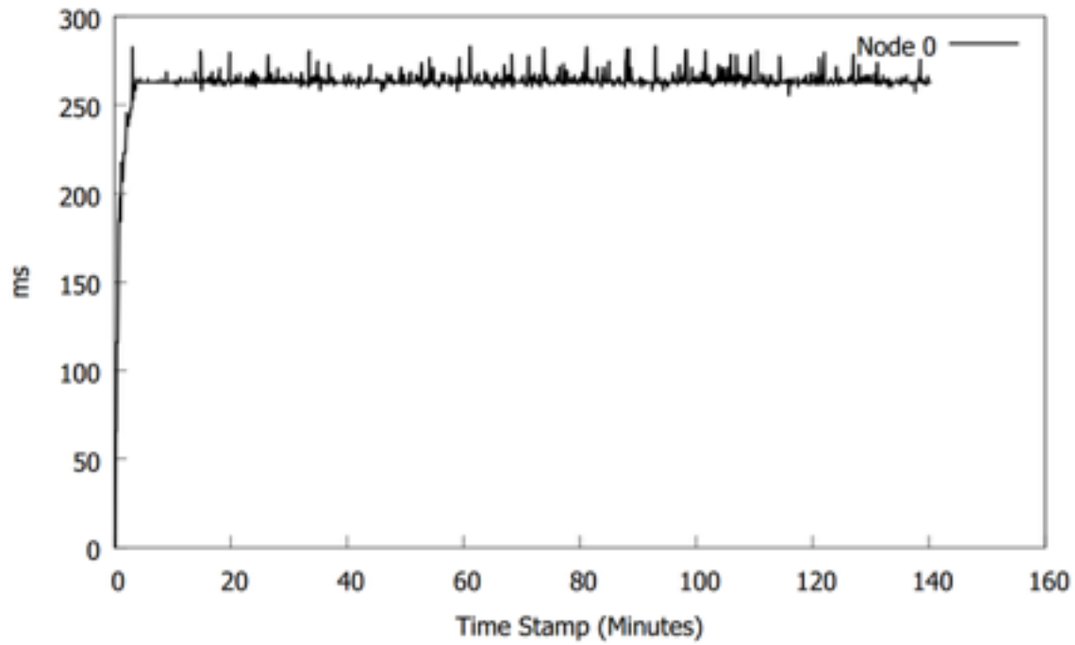


Global Nodes – 20 instances

Synthesis Test - Results Evaluation

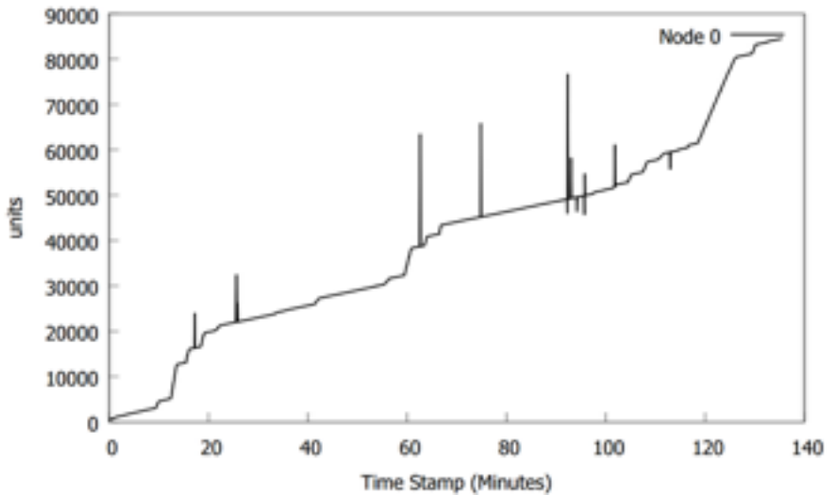


Join Network Time – 10 instances

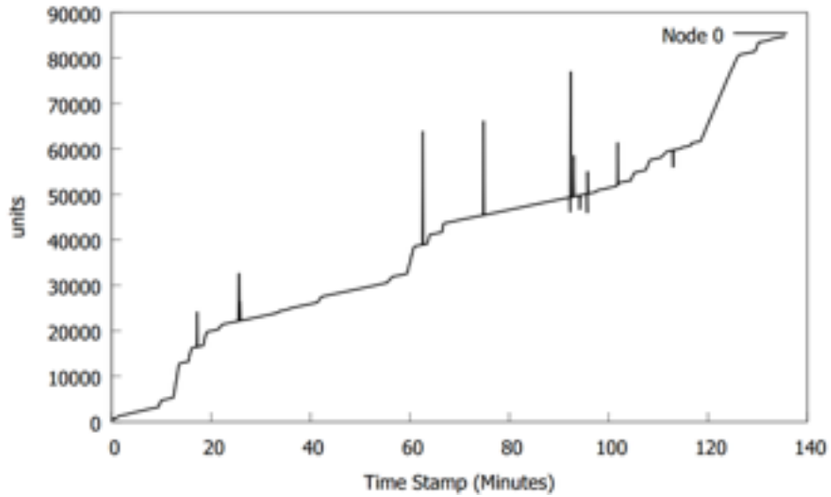


Join Network Time – 20 instances

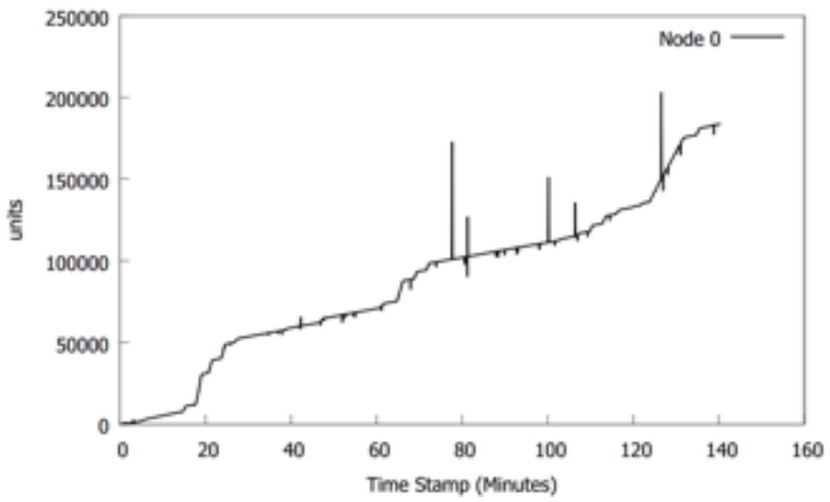
Synthesis Test - Results Evaluation



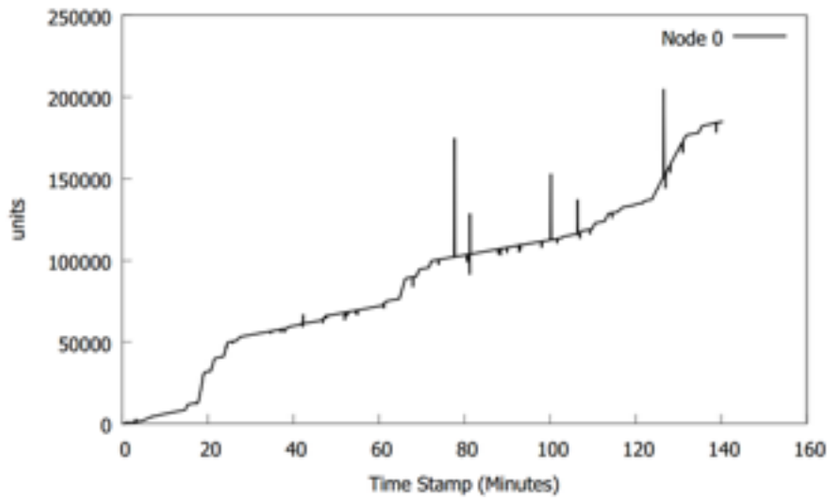
Global MessageDispatcher Sent Messages - 10 instances



Global MessageDispatcher Received Messages - 10 instances

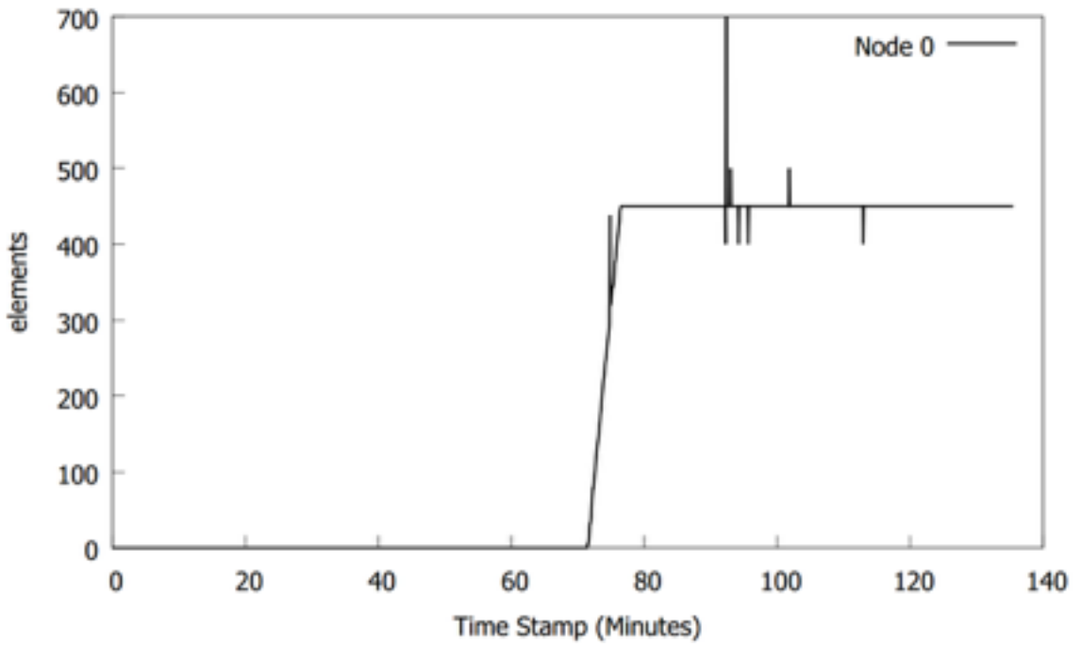


Global MessageDispatcher Sent Messages - 20 instances

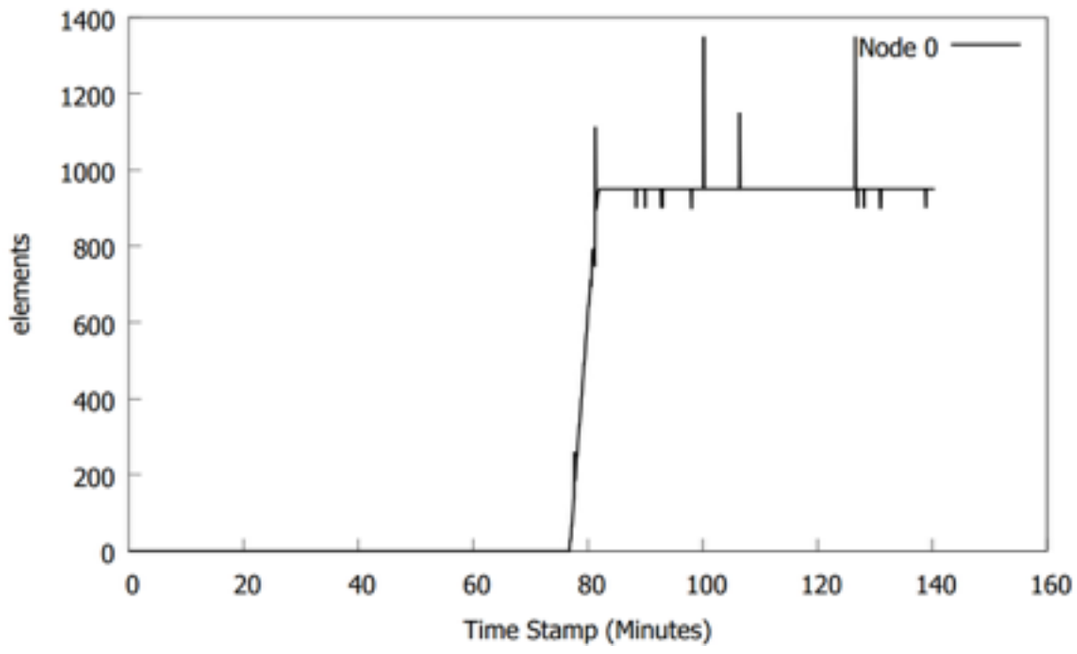


Global MessageDispatcher Received Messages - 20 instances

Synthesis Test - Results Evaluation



Global Uploaded Photos – 10 instances



Global Uploaded Photos – 20 instances

Synthesis Test

- In general, the performance of LifeSocial in synthesis test is GOOD.

  – Most functionalities (test cases) are working properly.
    - TestCaseCommentWallPost and TestCaseUploadFileinFolder lead to exceptions.

  – Network quality is stable.
    - no lost message
    - no crashed node
    - short reaction time

  – Monitoring function is working well.

Random Test

- Goal:
  - Examine performance of LifeSocial under more realistic condition
    - No test plan
    - With(out) logout

- Test machine (x20)
  - Processor : Intel i7-2600
  - RAM : 8 GB
  - Operating System : openSUSE 8.32.5-x86_64
  - Network Interface : Gigabit Ethernet
  - Java VM : 1.7.0_51 build 24.45-b08

Random Test

- Deployment:
  - Eclipse Kepler (for RCP and RAP developers)
  - Smartgit
- Configurations:
  - 10 and 20 test machines
  - Synchronize start script with Dropbox link
  - With logout and without logout
  - Modify parameter
    - enableAutomaticLogout = true (with logout)
    - enableAutomaticLogout = false (without logout)

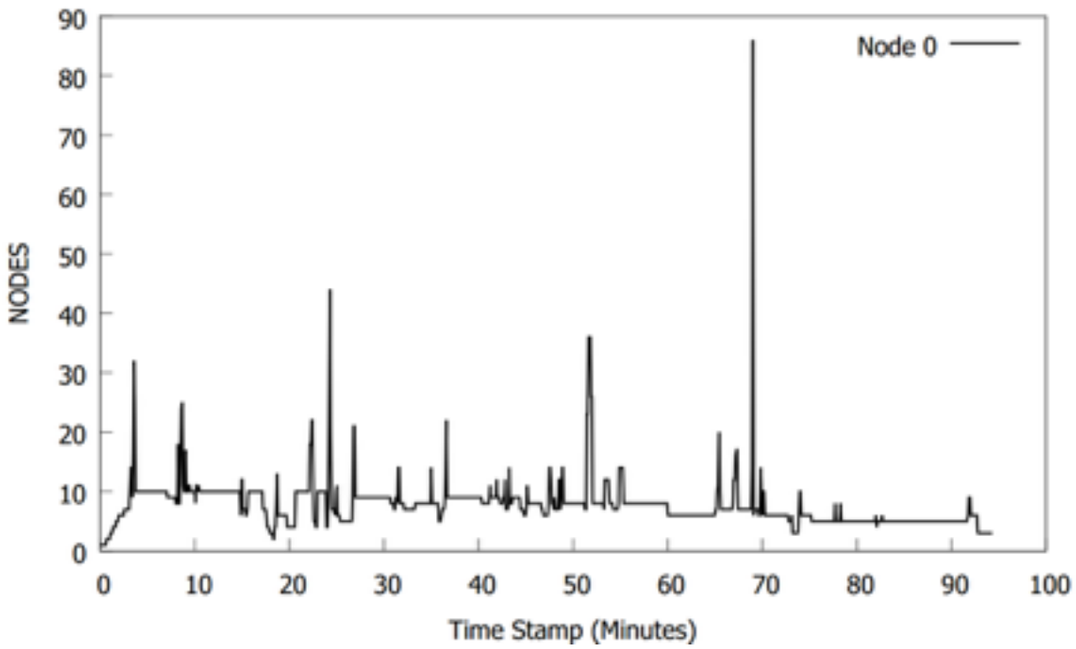
Random Test

- Start script is executed when a new user logs in the network
- After start script, user executes test cases randomly until
  - assigned duration is expired (with logout)
  - end of test (without logout)

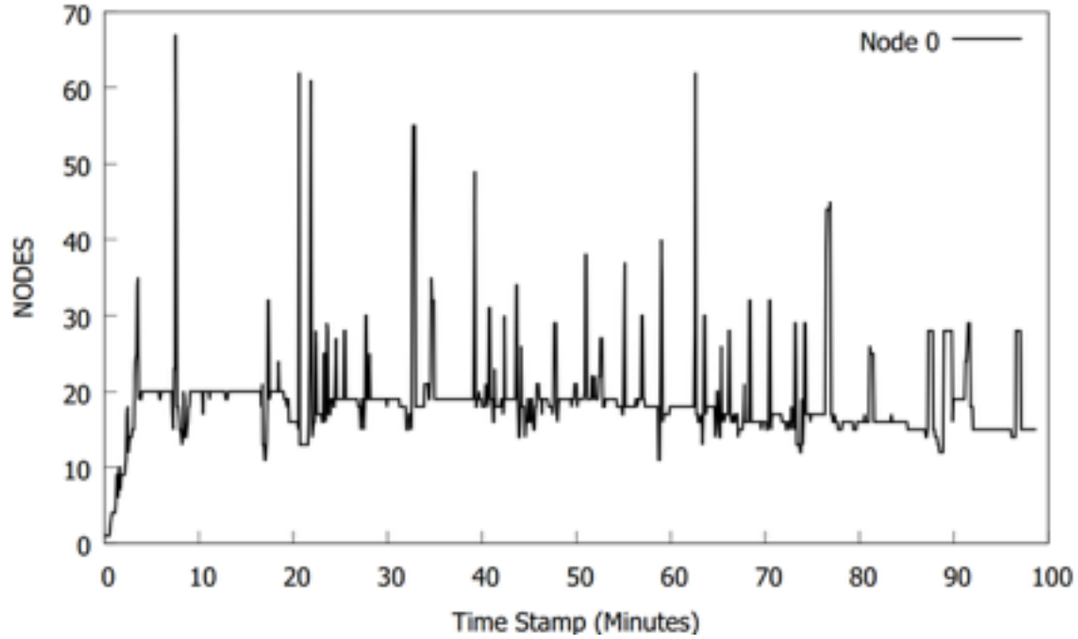
1x TestCaseChangeProfilePicture
1x TestCaseChangeProfileInformation
1x TestCaseSendFriendRequest
1x TestCaseCreatePhotoAlbum
5x TestCaseUploadPhoto
2x TestCaseSendWallPost
1x TestCaseCommentWallPost
1x TestCaseSendMultiChatInvitation
1x TestCaseCreateCalendarEvent
2x TestCaseCreateVote
1x TestCaseCreateGroup
1x TestCaseCreateFolder
2x TestCaseUploadFileinFolder
1x TestCaseViewForum
1x TestCaseCreateForumThread
1x TestCaseCommentForumThread

Random Test - Results Evaluation

- Random test with logout (Only the second round)

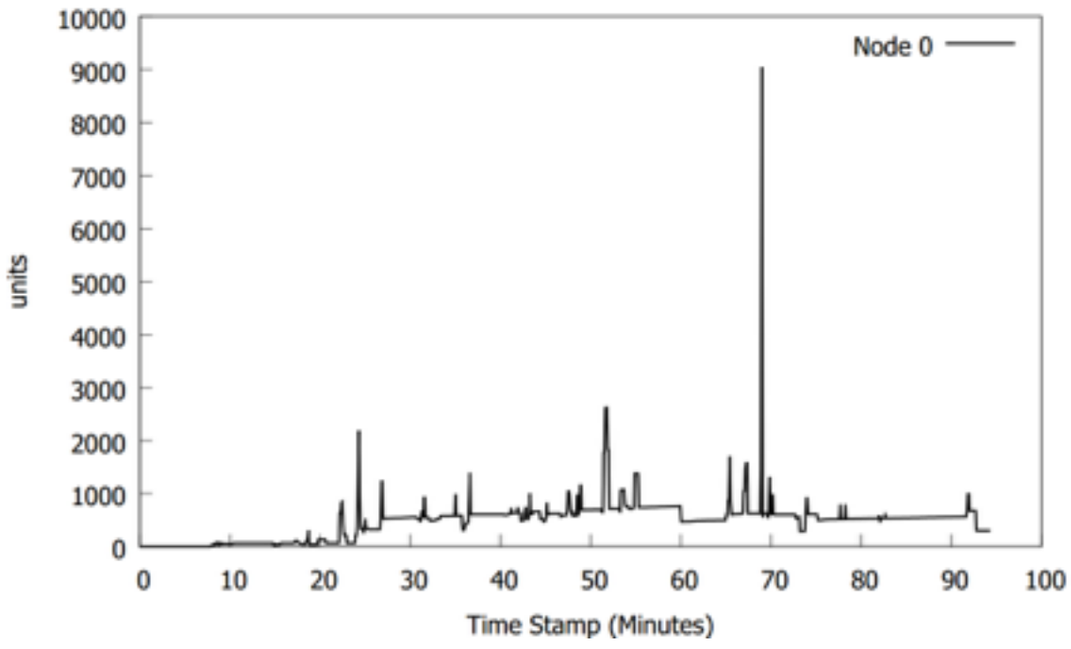


Global Nodes – 10 instances

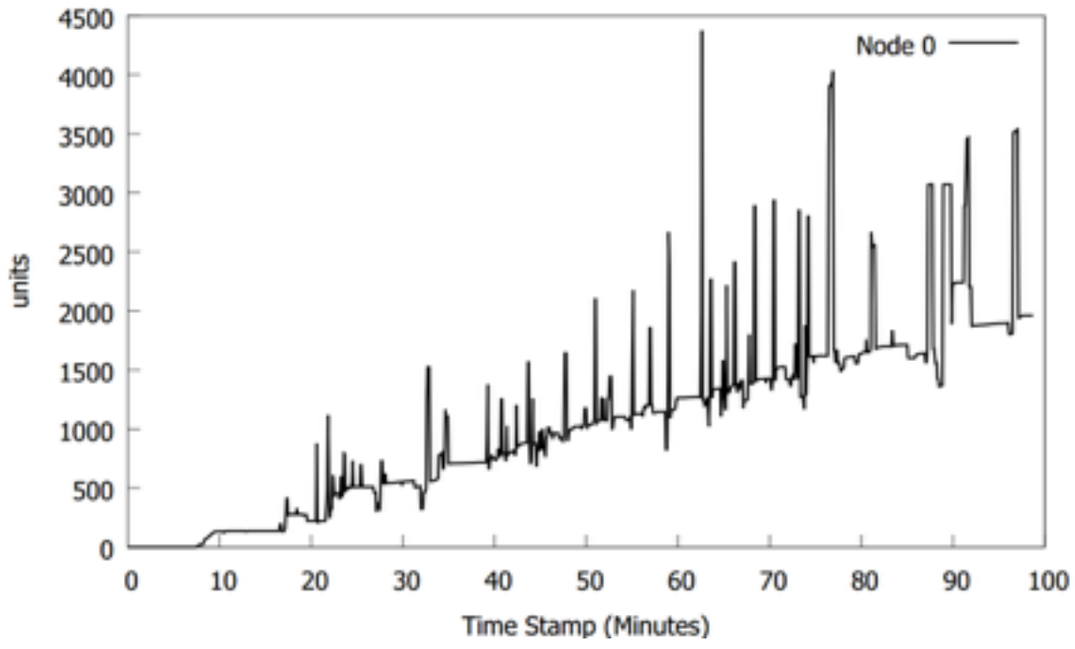


Global Nodes – 20 instances

Random Test - Results Evaluation

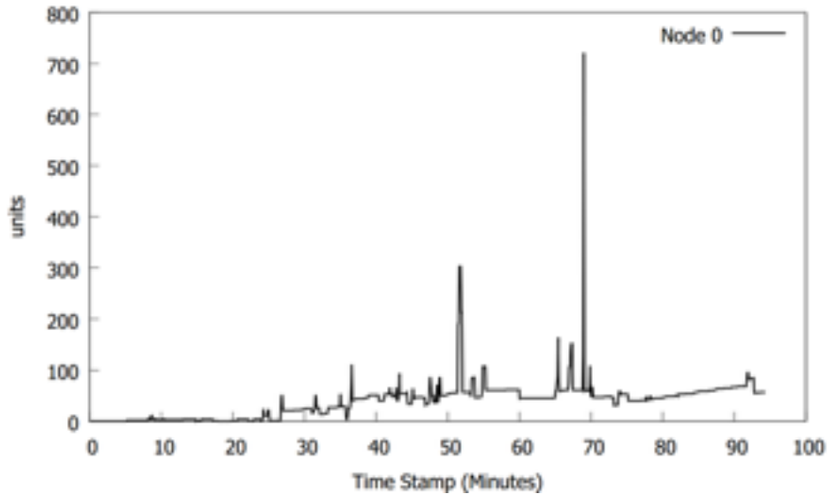


Global Lost Messages – 10 instances

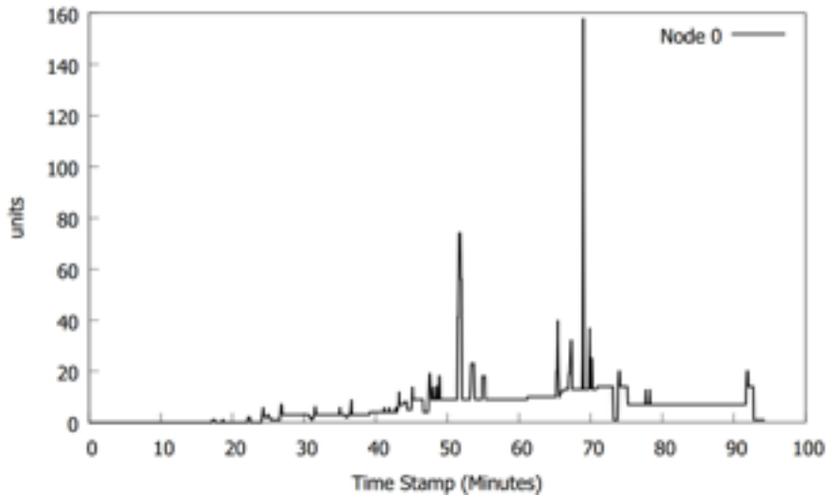


Global Lost Messages – 20 instances

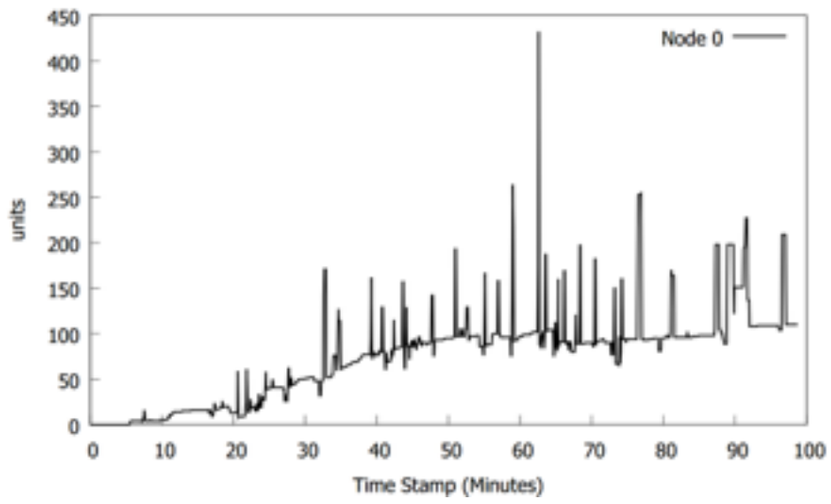
Random Test - Results Evaluation



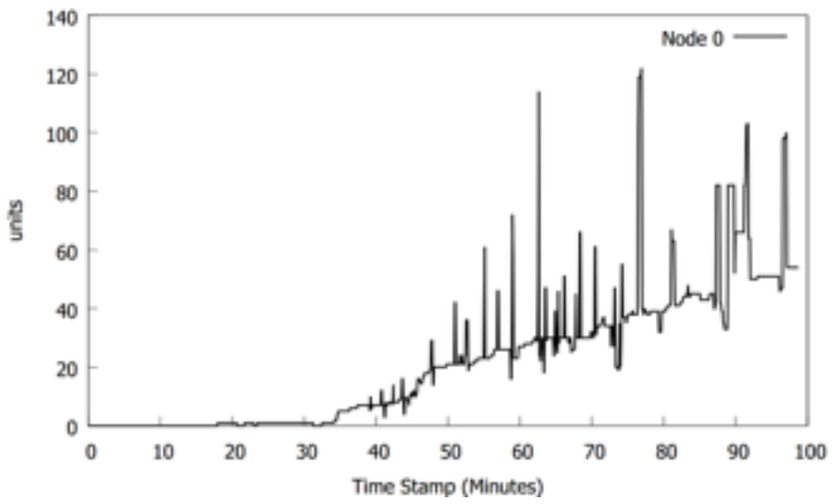
Global Null Pointer Exceptions - 10 instances



Global Other Exceptions - 10 instances



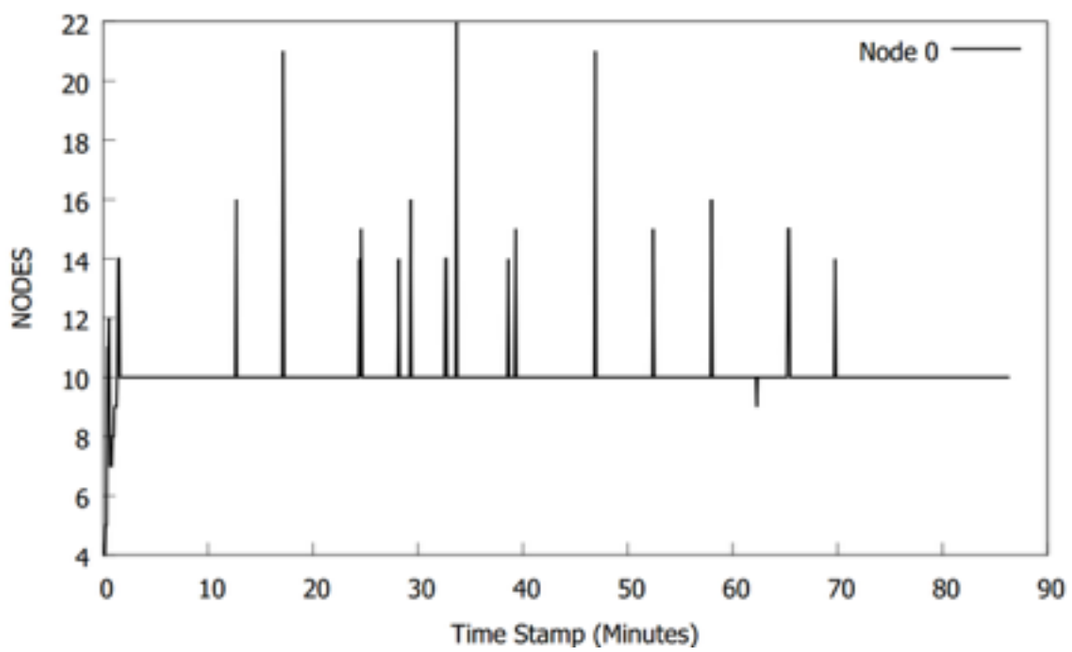
Global Null Pointer Exceptions - 20 instances



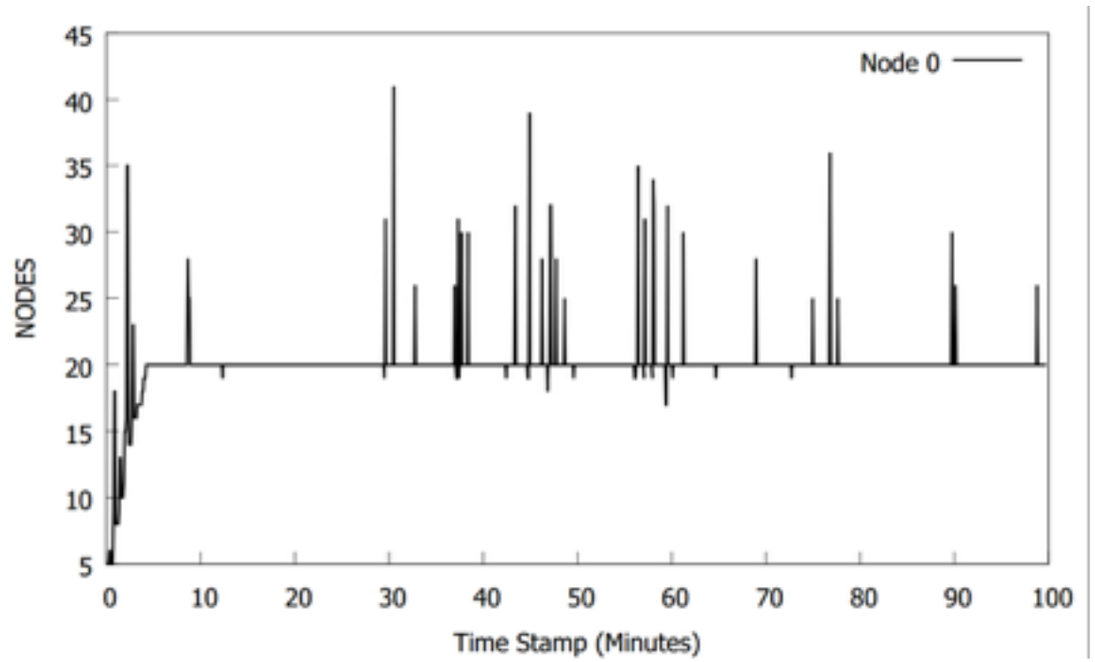
Global Other Exceptions - 20 instances

Random Test - Results Evaluation

- Random test without logout (Only the second round)

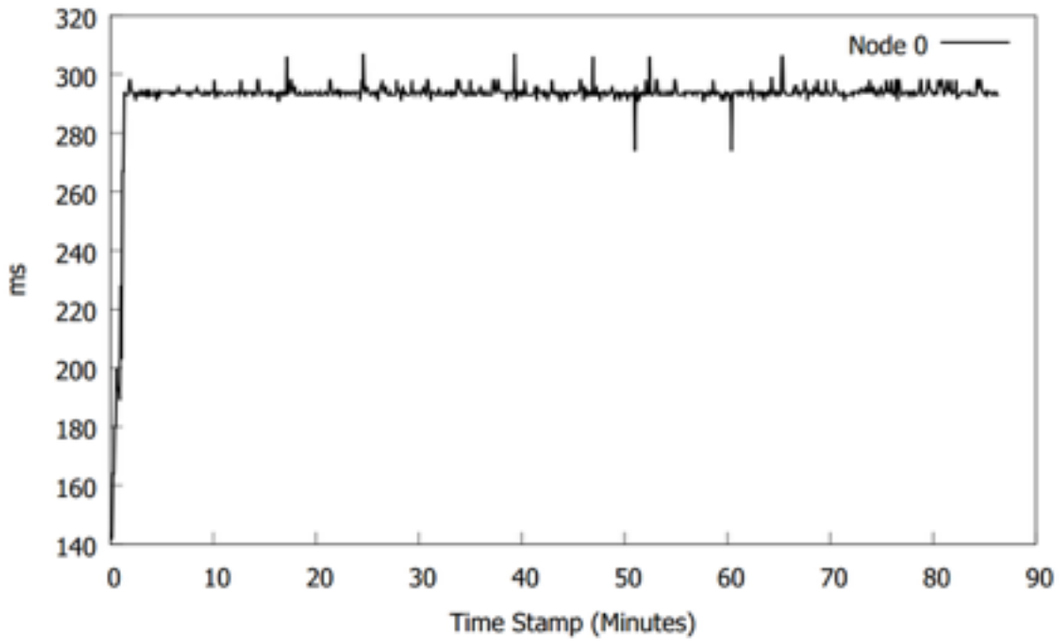


Global Nodes – 10 instances

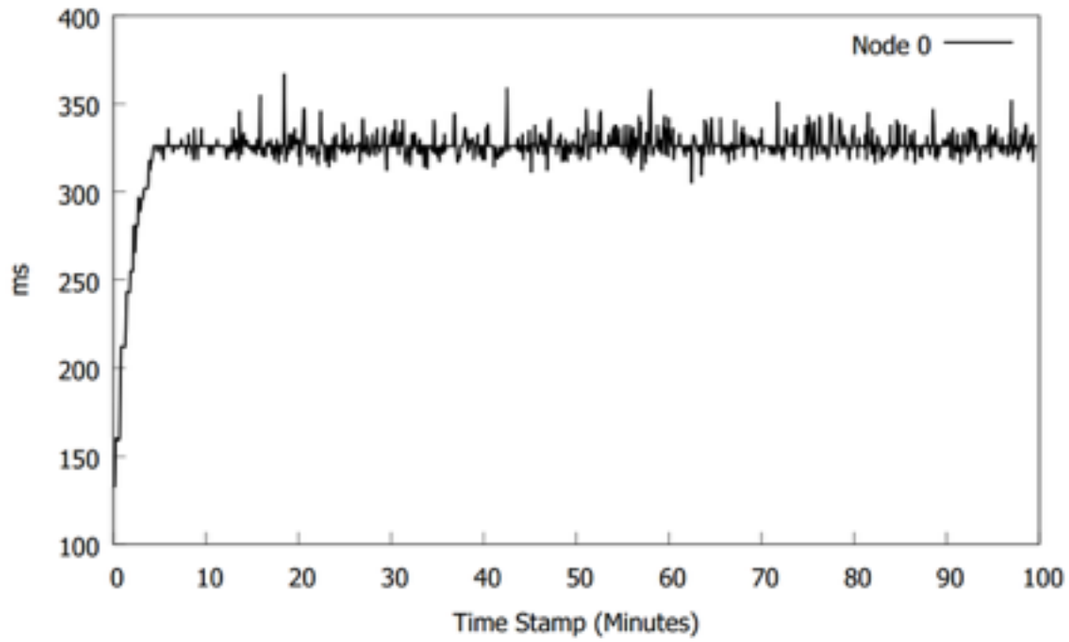


Global Nodes – 20 instances

Random Test - Results Evaluation

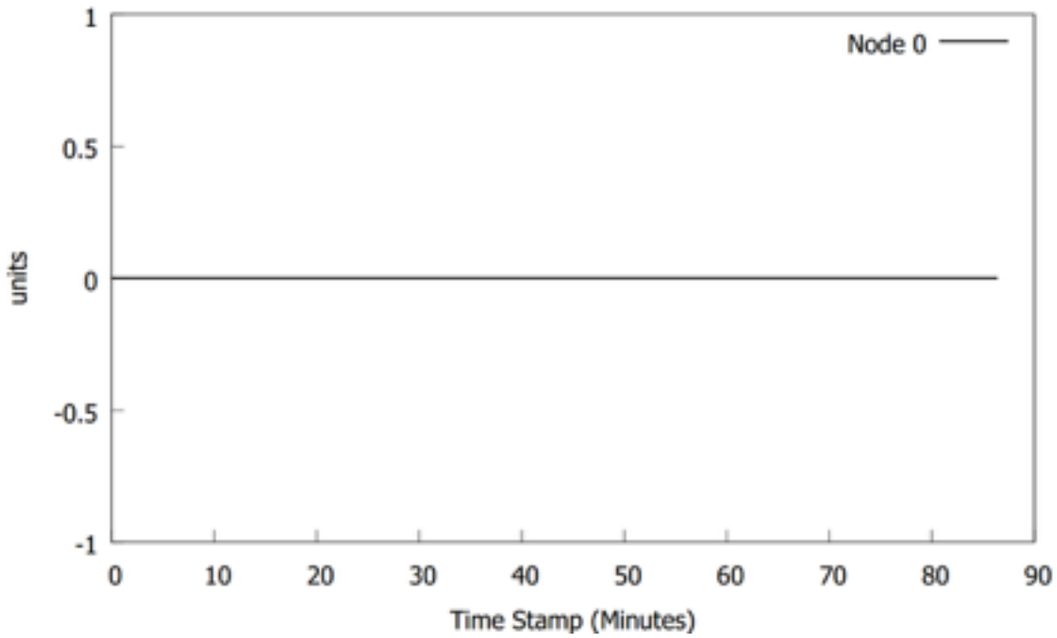


Global Join Network Time – 10 instances

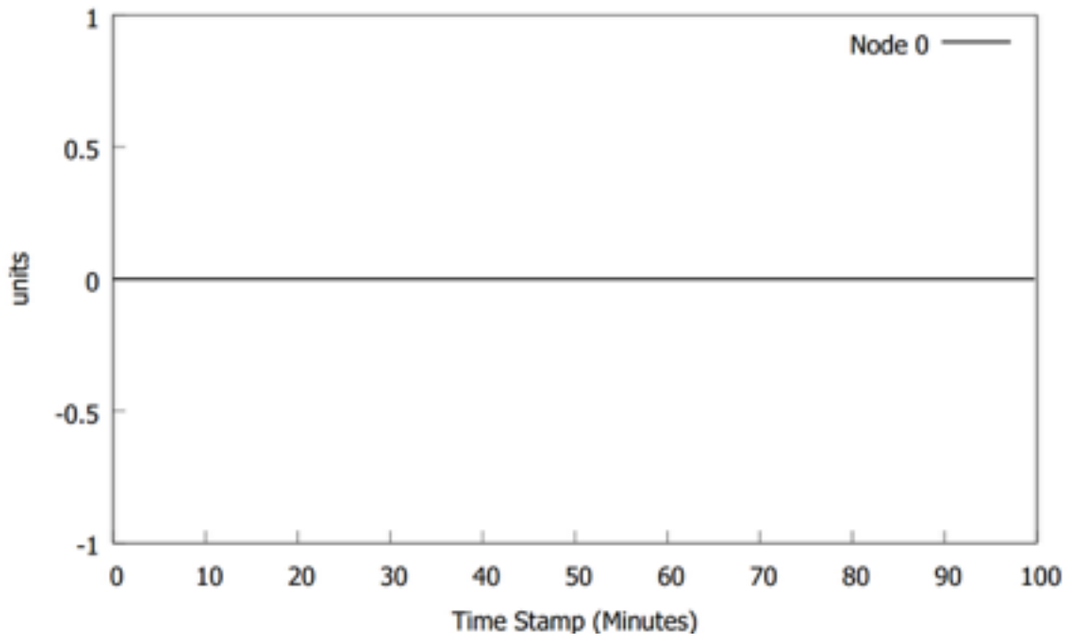


Global Join Network Time – 20 instances

Random Test - Results Evaluation

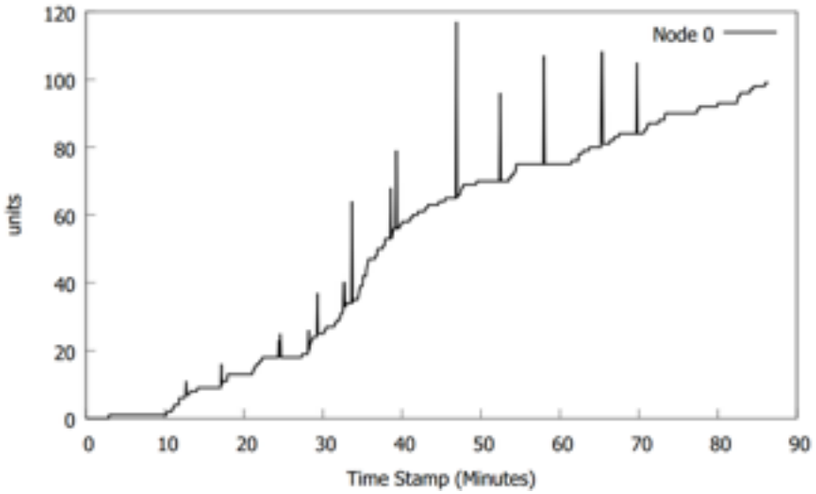


Global Lost Messages – 10 instances

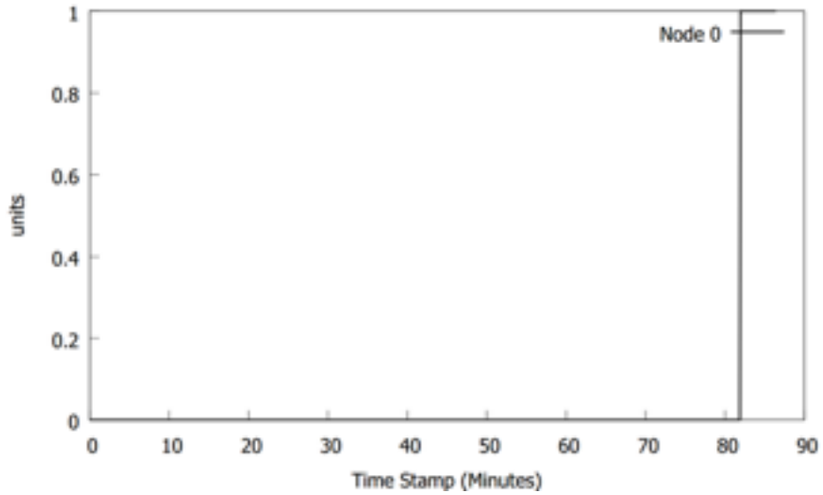


Global Lost Messages – 20 instances

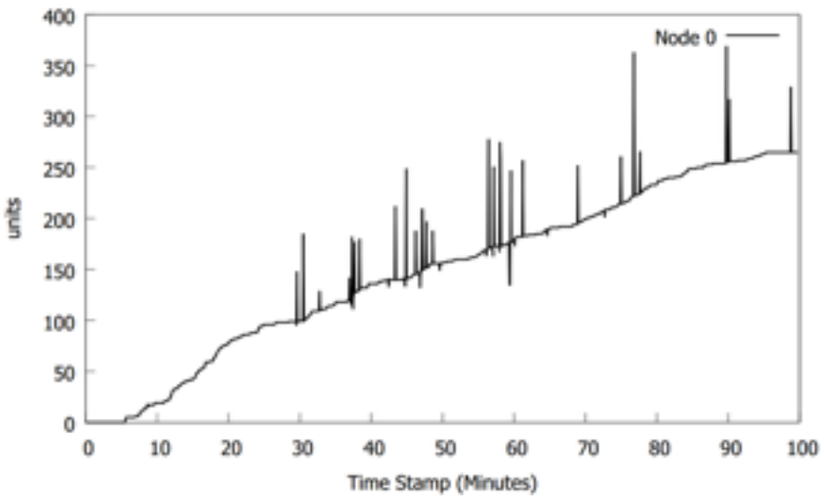
Random Test - Results Evaluation



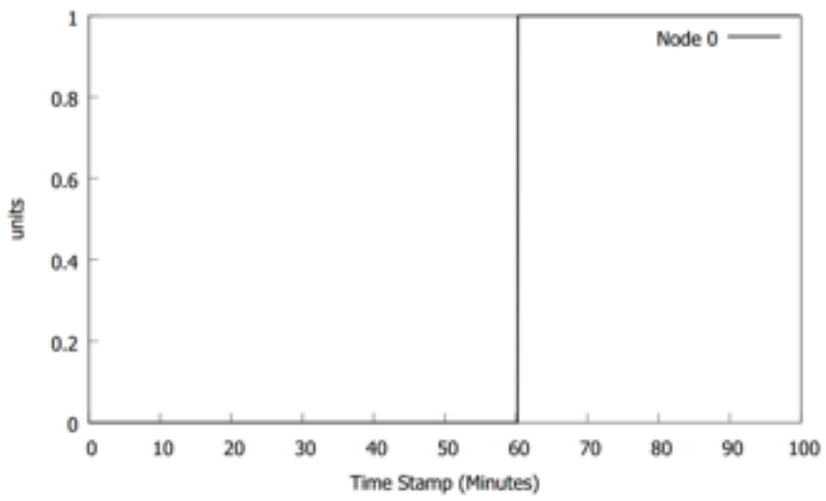
Global Null Pointer Exceptions - 10 instances



Global Other Exceptions - 10 instances



Global Null Pointer Exceptions - 20 instances



Global Other Exceptions - 20 instances

Random Test

- Observations:
  - Random test with logout has obvious problems (lost messages, crashed nodes).
  - Random test without logout is as stable as synthesis test.
  - There are still number of exceptions in both the tests.

- In general, LifeSocial is NOT robust enough under complicated conditions.

Long Term Test

- Goal is to maintain the network for long time, check the performance and reliability of LifeSocial application.

- 10 servers are used in this test.

- Hardware and Software specifications are same as the systems used in synthesis test.

- One machine is used to initiate and control the tests remotely on all servers.

- Deployment:

  – Eclipse Kepler (for RCP and RAP developers)

  – Smartgit

Long Term Test - Test Setup

- Secure Connection
  - Local port forwarding via SSH tunneling is used.
  - X11Forwarding must be enabled in both server and client.
- Graphical User Interface
  - Virtual Network Connection (VNC) is used.
  - VNC Server Component is used to start and stop VNC session.
  - VNC Viewer is used to view the display of the server.

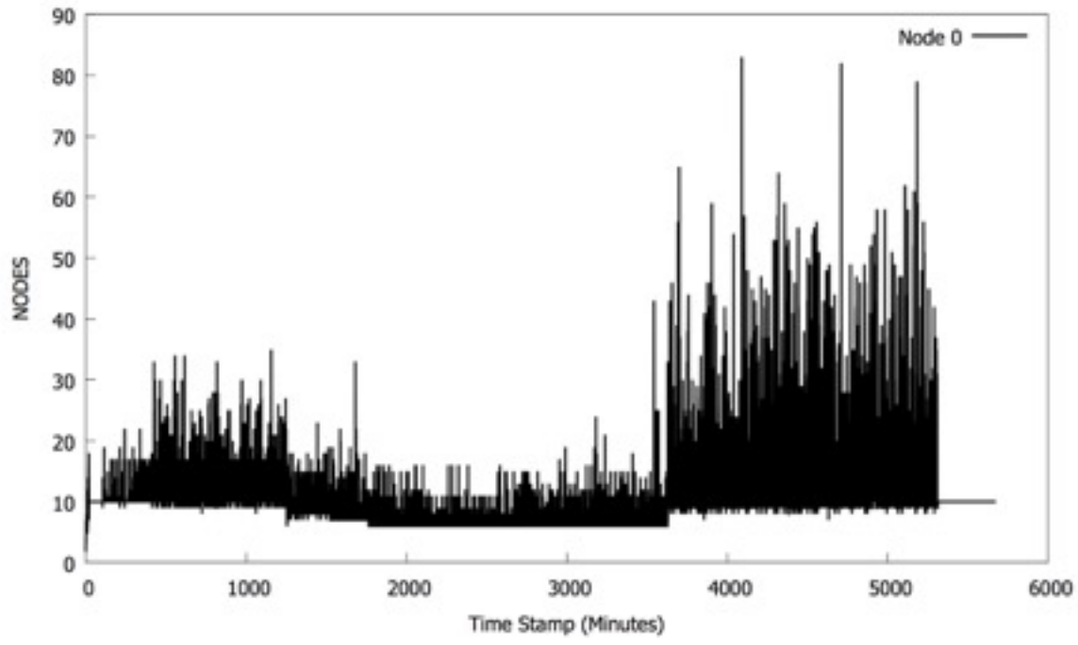
Long Term Test - Test Plan

1x Wait 20m
1x TestCaseChangeProfilePicture 10m
1x Wait 20m
1x TestCaseChangeProfileInformation 10m
1x Wait 20m
1x TestCaseViewOwnProfile 10m
1x Wait 20m
1x TestCaseViewFriendsProfile 10m
1x Wait 20m
5x TestCaseSendWallPost 10m
1x Wait 20m
5x TestCaseCommentWallPost 10m
1x Wait 20m
5x TestCaseViewOwnWall 10m
1x Wait 20m
5x TestCaseViewFriendsWall 10m
1x Wait 20m
5x TestCaseSendMessage 10m
1x Wait 20m
5x TestCaseViewOutboxMessage 10m

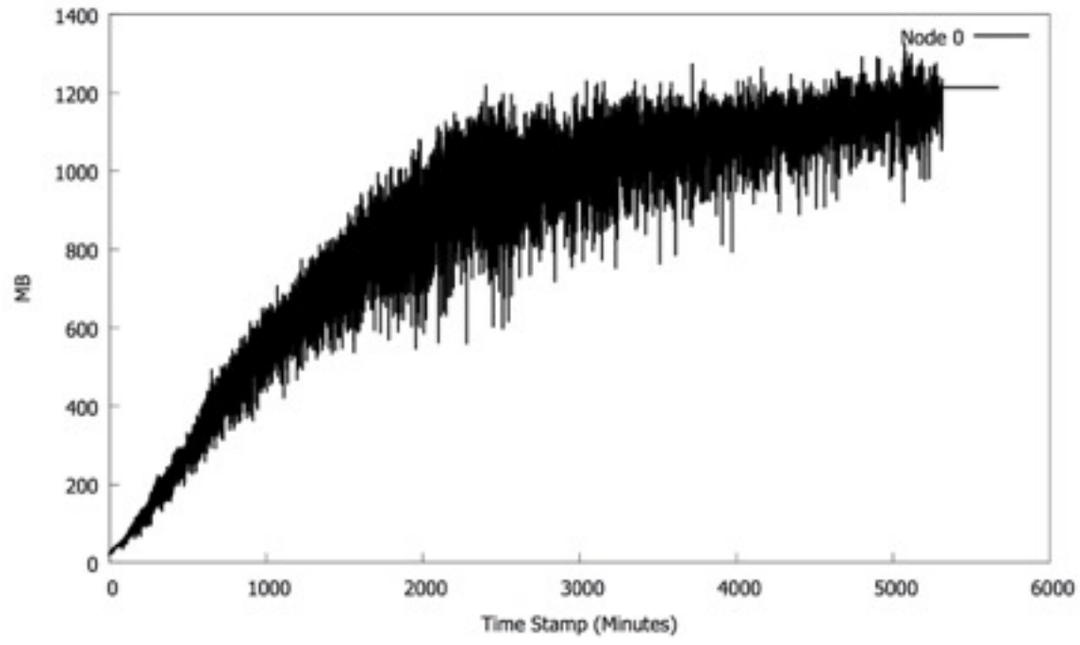
1x Wait 20m
5x TestCaseViewInboxMessage 10m
1x Wait 20m
2x TestCaseDeleteInboxMessage 10m
1x Wait 20m
2x TestCaseCreateGroup 10m
1x Wait 20m
1x TestCaseViewForum 10m
1x Wait 20m
5x TestCaseCreateForumThread 10m
1x Wait 20m
5x TestCaseCommentForumThread 10m
1x Wait 20m
5x TestCaseViewForumThread 10m
1x Wait 20m
5x TestCaseCreateFolder 10m
1x Wait 20m
2x TestCaseViewGroup 10m
1x Wait 20m
2x TestCaseRenameFolder 10m
1x Wait 20m
5x TestCaseUploadFileinFolder 10m

1x Wait 20m
1x TestCaseLeaveGroup 10m
1x Wait 20m
1x TestCaseSendMultiChatInvitation 10m
1x Wait 20m
5x TestCaseSendMultiChatMessage 10m
1x Wait 20m
1x TestCaseLeaveMultiChatMessage 10m
1x Wait 20m
5x TestCaseSearch 10m
1x Wait 20m
1x TestCaseViewCalendar 10m
1x Wait 20m
5x TestCaseCreateCalendarEvent 10m
1x Wait 20m
2x TestCaseEditCalendarEvent 10m
1x Wait 20m
2x TestCaseDeleteCalendarEvent 10m
1x Wait 60m

Long Term Test - Results Evaluation

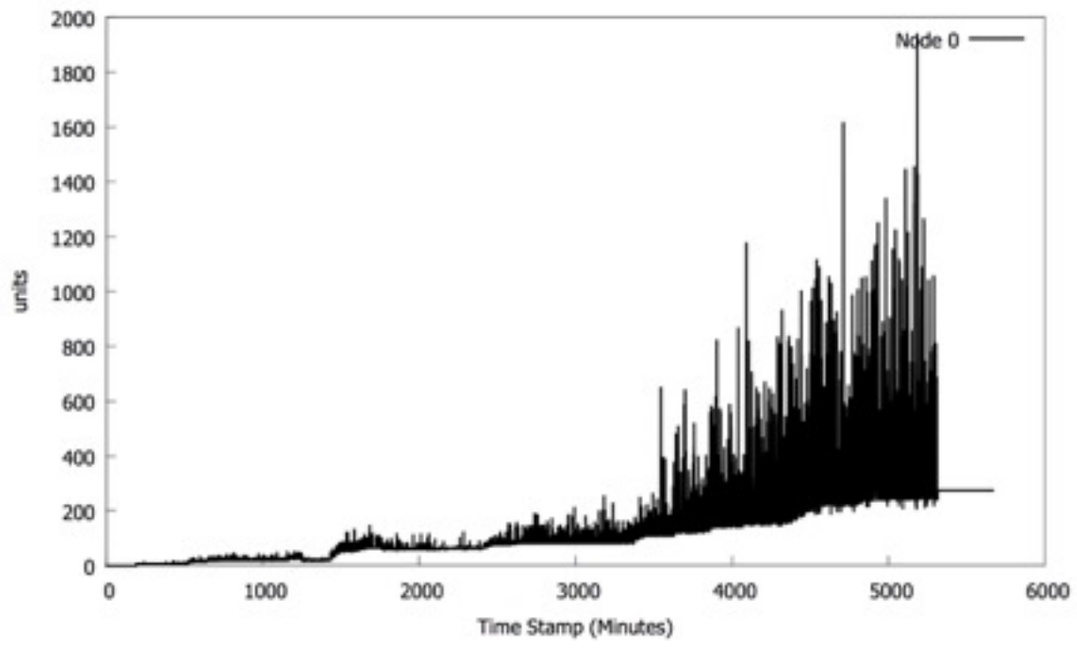


Global Nodes – 10 instances

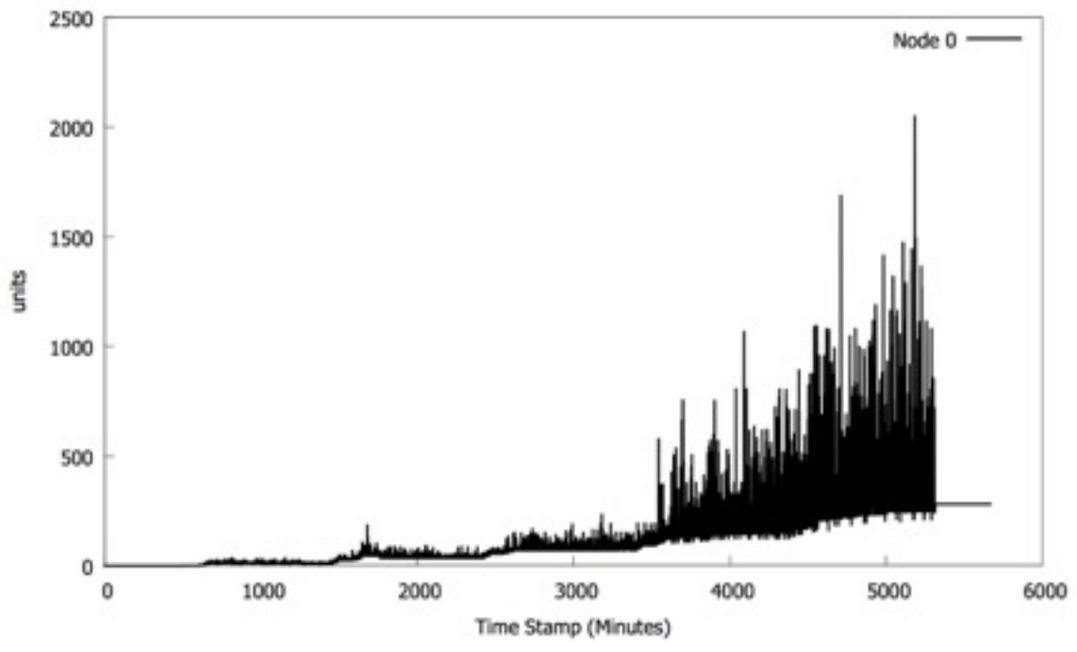


Global Memory Usage Size – 10 instances

Long Term Test - Results Evaluation

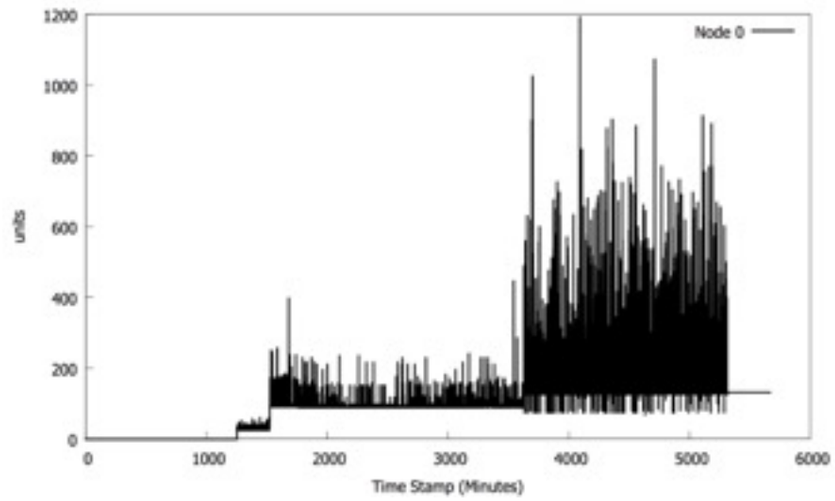


Global Null Pointer Exceptions - 10 instances



Global Other Exceptions - 10 instances

Long Term Test - Results Evaluation



Global Messages Lost – 10 instances

Long Term Test - Issues

- Out Of Memory Exception
  - JProfiler memory analyzer tool is used to detect memory leaks.
  - Memory Leak is detected in MonitoringPluginHandler.
- Queue Overflow Exception
  - Caused by PriorityTransportLayer in FreePastry.
  - Messages are lost because of overflow in the message queue.

Long Term Test

- Observations:
  - Message queue is overflowed.
  - Memory consumption is slowing increasing all the time.
  - Both these problems are caused by FreePastry.
- Overall Performance of long term test is satisfactory as LifeSocial is able to run for 5300 mins = 88.3 hours = 3.67 days.

Conclusion

- Achievements:
  - Fixed old test cases and created new test cases
  - Synthesis test
  - Random test
  - Long term test

- Future work:
  - Solve exception issue (WallPlugin, GroupsPlugin).
  - Improve reliability under complicated condition.
  - Improve stability in long term use.

Thanks!