

RDF GRAPH SYNCHRONIZATION FOR COLLABORATIVE ROBOTICS

ADAM FREJ

PATRYK TOMASZEWSKI

PROBLEM DEFINITION

Synchronise knowledge graphs between multiple agents in real-time, while each agent continuously appends new data, and while also handling issues typically present in wireless communication.

By:

- Adding knowledge graph versioning

- Implementing merging mechanism

- Designing and implementing communication protocols for graph synchronisation

IMPLEMENTED SOLUTION

3 main modules:

- Agents – behaviors, messaging
- RDF document – revision history, sync algorithms
- GUI / simulation – user interaction, app control

IMPLEMENTED SOLUTION

Agents

- 6 behaviors: register, status send, status receive, local revision create, remote revision receive, revision request receive
- 3 messages: status, revision, revision request
- Internal agent knowledge: RDF document, known agents (ttl, latest revision), merge master JID
- Additional algorithms: send/publish revision, send revision request, elect merge master

IMPLEMENTED SOLUTION

RDF document

- Storage: revision history, cached graph state
- Main algorithms: append revision, merge, rebase
- Helper algorithms: common ancestor, revisions between, combine revisions, recreate cached state, check if new revision is a tip

Revision:

- Storage: added/removed triples, parent(s), hash
- Algorithms: combine

IMPLEMENTED SOLUTION

GUI / simulation

- Server: register/deregister agents, restart
- Graph generator: generate fake data, uncover triples for agents, store ground truth
- Simulation: control agent number, command over the server, log graph sync state
- Endpoints: API for front-end
- GUI: html front-end, JS scripts for visualization

IMPLEMENTED SOLUTION

Deployment

- Python 3.9 – strict requirement
- Spade 3.2.3 – strict requirement
- Open source – MIT license, <https://github.com/adFrej/RDFAgents>
- Docker – provided Dockerfiles,
out of the box solution for self-hosted local XMPP server connected with our app

RESEARCH QUESTIONS

- Is the system feasible to implement?
- Will the agents synchronise in a timely manner?
- Can the system handle addition/removal of agents?
- How the system scales with the number of agents?

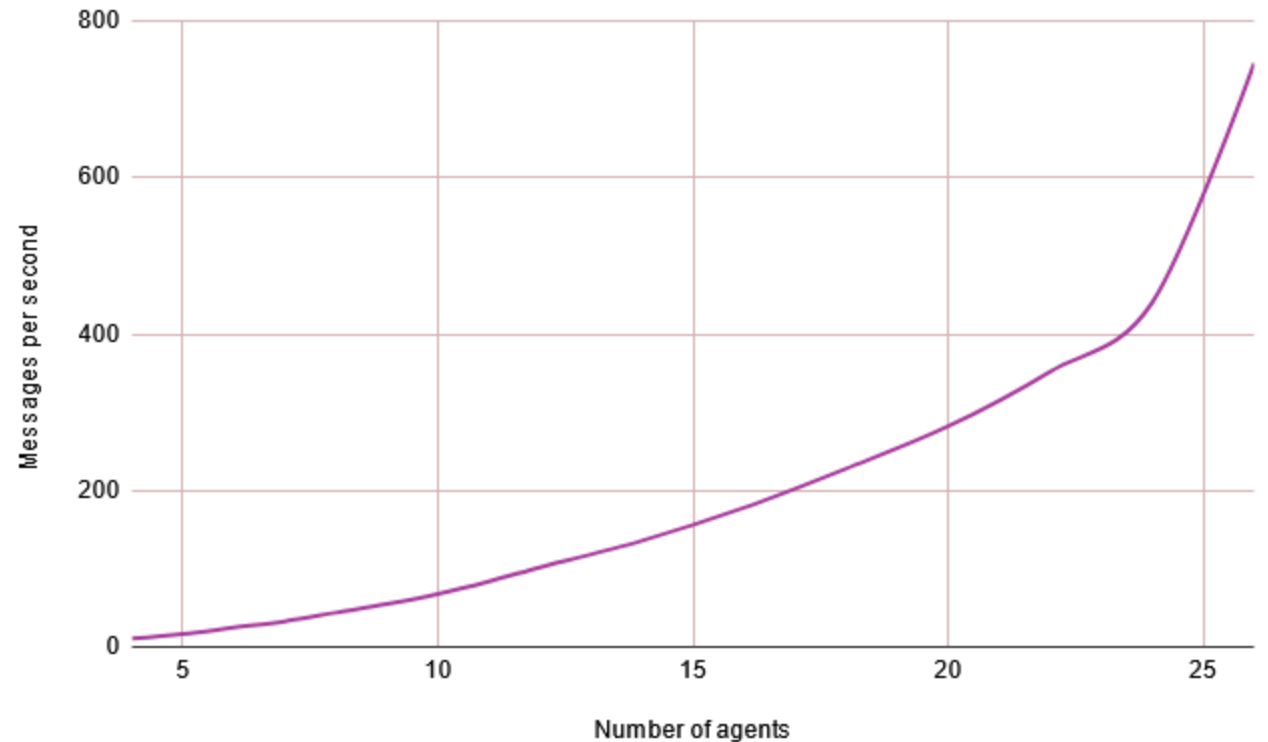
SYSTEM SHOWCASE



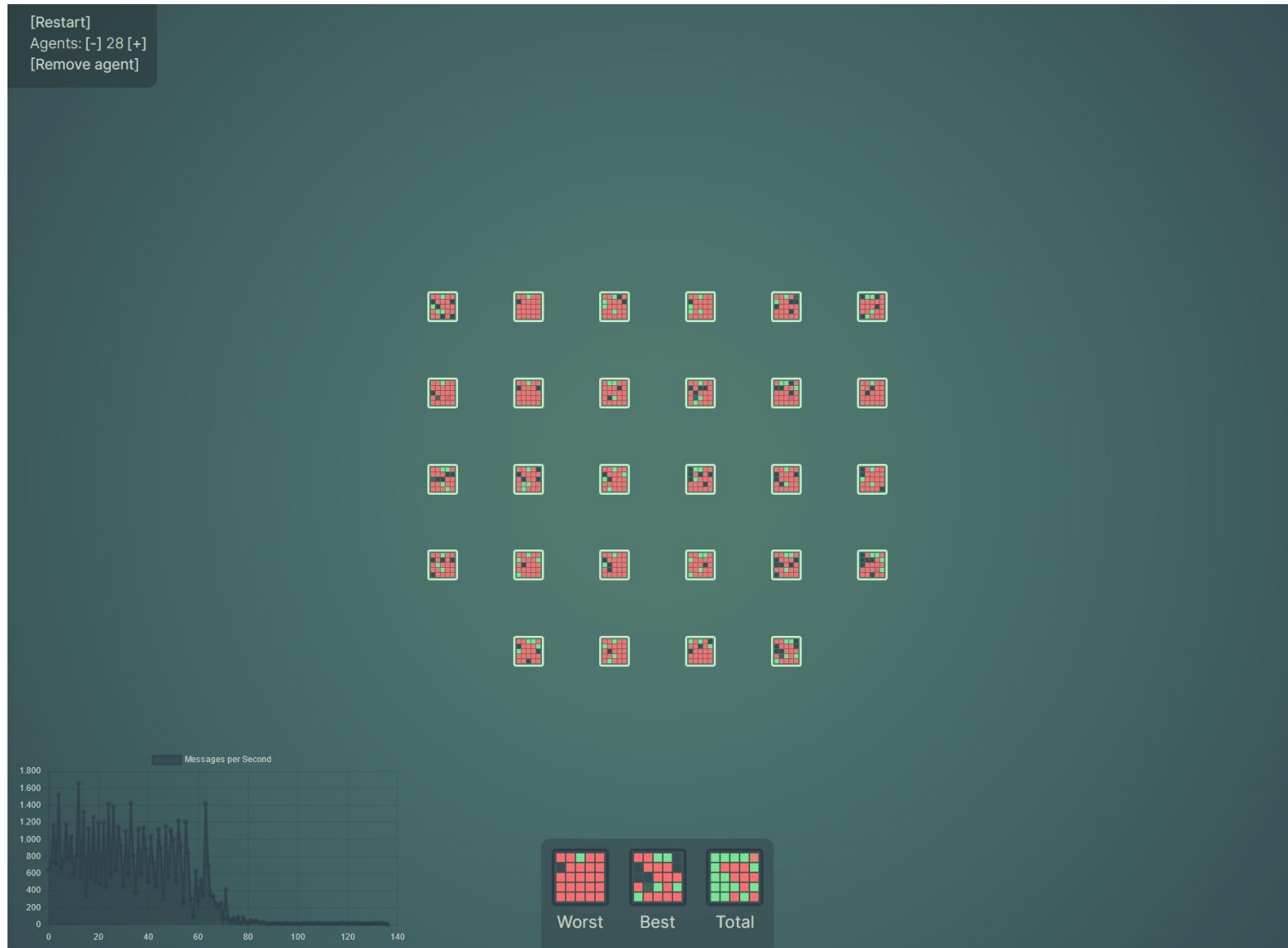
<https://www.youtube.com/watch?v=c55aLkx5GNM>

PERFORMANCE ANALYSIS

- It takes the system a couple of seconds (4s - 12s) to synchronise after new agent is added.
- Number of messages per second grows non-linearly with number of agents!



CRITICAL FAILURE AT ~28 AGENTS



```
(Agent-admin@localhost/2): Lost connection with admin@localhost/11
(Agent-admin@localhost/25): Lost connection with admin@localhost/11
(Agent-admin@localhost/16): Lost connection with admin@localhost/11
(Agent-admin@localhost/3): Lost connection with admin@localhost/11
(Agent-admin@localhost/6): Lost connection with admin@localhost/11
(Agent-admin@localhost/23): Lost connection with admin@localhost/11
(Agent-admin@localhost/22): Lost connection with admin@localhost/11
(Agent-admin@localhost/17): Lost connection with admin@localhost/11
(Agent-admin@localhost/7): Lost connection with admin@localhost/11
(Agent-admin@localhost/18): Lost connection with admin@localhost/11
(Agent-admin@localhost/26): Lost connection with admin@localhost/11
(Agent-admin@localhost/24): Lost connection with admin@localhost/12
(Agent-admin@localhost/8): Lost connection with admin@localhost/12
(Agent-admin@localhost/27): Lost connection with admin@localhost/12
(Agent-admin@localhost/9): Lost connection with admin@localhost/12
(Agent-admin@localhost/28): Lost connection with admin@localhost/10
(Agent-admin@localhost/28): Lost connection with admin@localhost/20
(Agent-admin@localhost/10): Lost connection with admin@localhost/13
(Agent-admin@localhost/10): Lost connection with admin@localhost/14
(Agent-admin@localhost/10): Lost connection with admin@localhost/1
(Agent-admin@localhost/20): Lost connection with admin@localhost/13
(Agent-admin@localhost/20): Lost connection with admin@localhost/14
(Agent-admin@localhost/20): Lost connection with admin@localhost/1
(Agent-admin@localhost/11): Lost connection with admin@localhost/15
(Agent-admin@localhost/11): Lost connection with admin@localhost/5
(Agent-admin@localhost/11): Lost connection with admin@localhost/2
(Agent-admin@localhost/11): Lost connection with admin@localhost/21
(Agent-admin@localhost/12): Lost connection with admin@localhost/15
(Agent-admin@localhost/12): Lost connection with admin@localhost/5
(Agent-admin@localhost/12): Lost connection with admin@localhost/21
(Agent-admin@localhost/12): Lost connection with admin@localhost/25
(Agent-admin@localhost/13): Lost connection with admin@localhost/5
(Agent-admin@localhost/13): Lost connection with admin@localhost/2
(Agent-admin@localhost/13): Lost connection with admin@localhost/16
(Agent-admin@localhost/13): Lost connection with admin@localhost/25
(Agent-admin@localhost/14): Lost connection with admin@localhost/2
(Agent-admin@localhost/14): Lost connection with admin@localhost/16
(Agent-admin@localhost/14): Lost connection with admin@localhost/25
```

SUMMARY

- System could have been implemented, all agents synchronize, even with removal and addition of new agents.
- Analyzed design is ill-suited for large scale applications, handling maximum of around 27 agents.
- On github we provide the full system implementation in dockerized image, along with simulation and visualization environment and comprehensive usage readme.