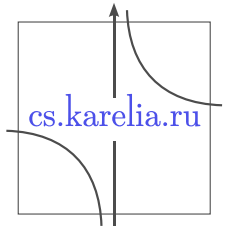
Петрозаводский государственный университет
Институт математики и информационных технологий
Кафедра информатики и математического обеспечения



Андрей Сергеевич Вдовенко

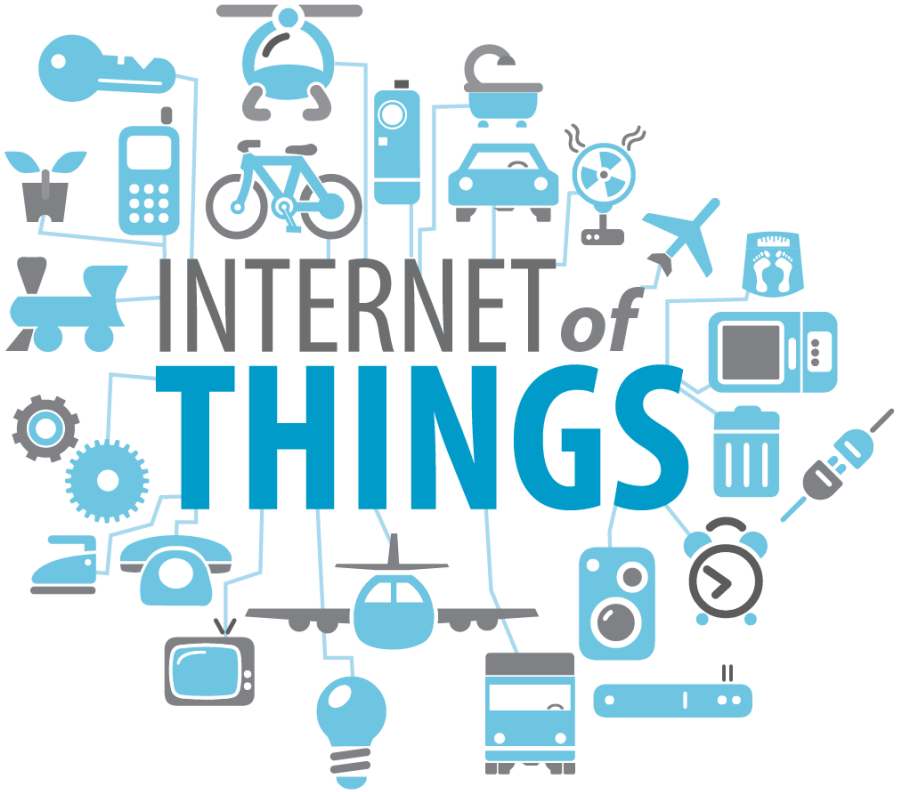
Изучение производительности при масштабировании для платформы Smart-M3 в реализации CuteSIB

Научный руководитель: к.ф.-м.н., доцент Д. Ж. Корзун

69-я Всероссийская (с международным участием) научная конференция
обучающихся и молодых ученых
секция «Информатика и математическое обеспечение»
12 апреля 2017 г., Петрозаводск

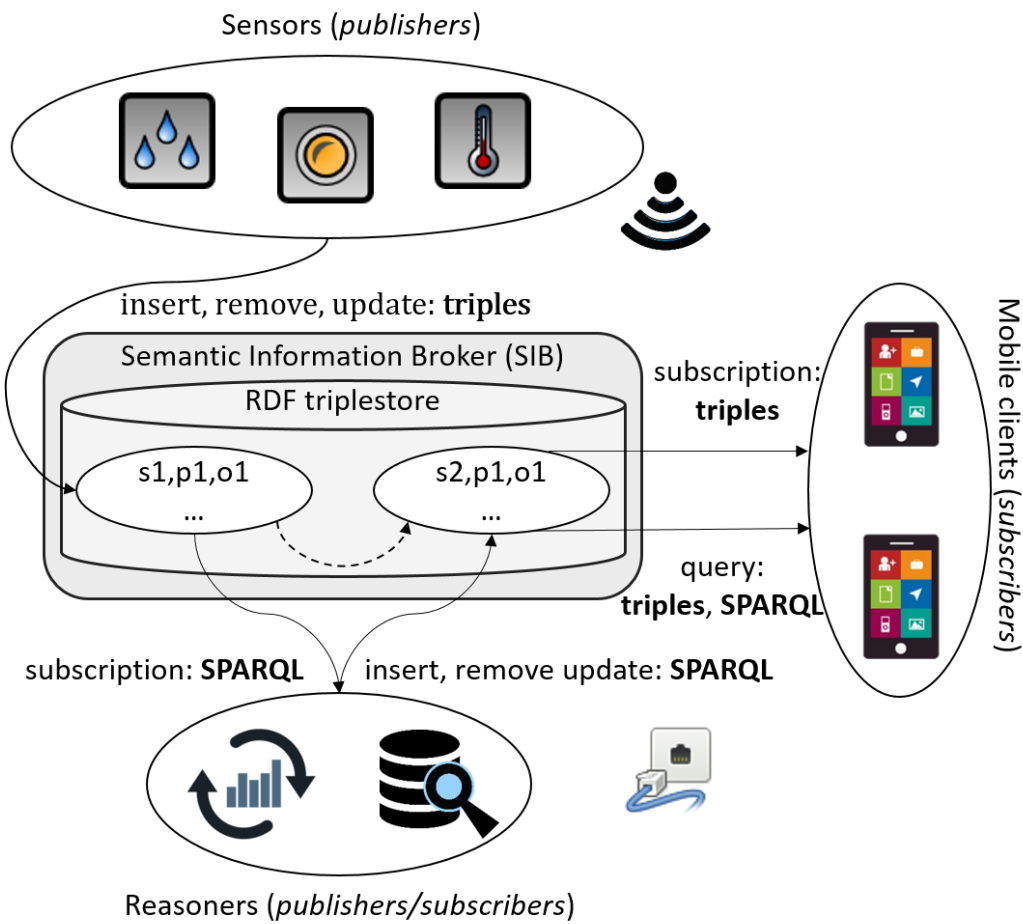
IoT Concept

- Wide range of application domains
- High process dynamicity
- Large amounts of capacity-heterogeneous devices
- No wide-spread effective approach for solving large multi-party interaction



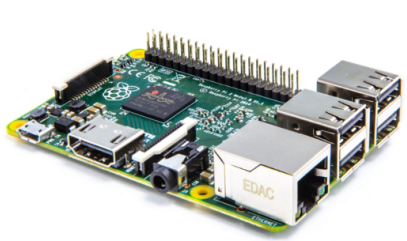
Smart-M3 Platform

- Platform for Smart Spaces deploying
- SIB: Semantic Information Broker for maintenance of shared content
- RDF data representation model: semantic interoperability and ontology-driven programming
- Subscription mechanism is expensive for broker



CuteSIB Implementation

- Qt framework:
  - Embedded devices
  - Wi-Fi routers
  - Smartphones
  - Laptops and desktops
  - TV and etc
- Dynamic libraries: configuration to specific tasks and devices
- Outperformance of CuteSIB compared with existing Smart-M3 SIB implementations
- Other Smart-M3 implementation: RedSIB, PySIB, The Piglet-based SIB, The OSGi SIB

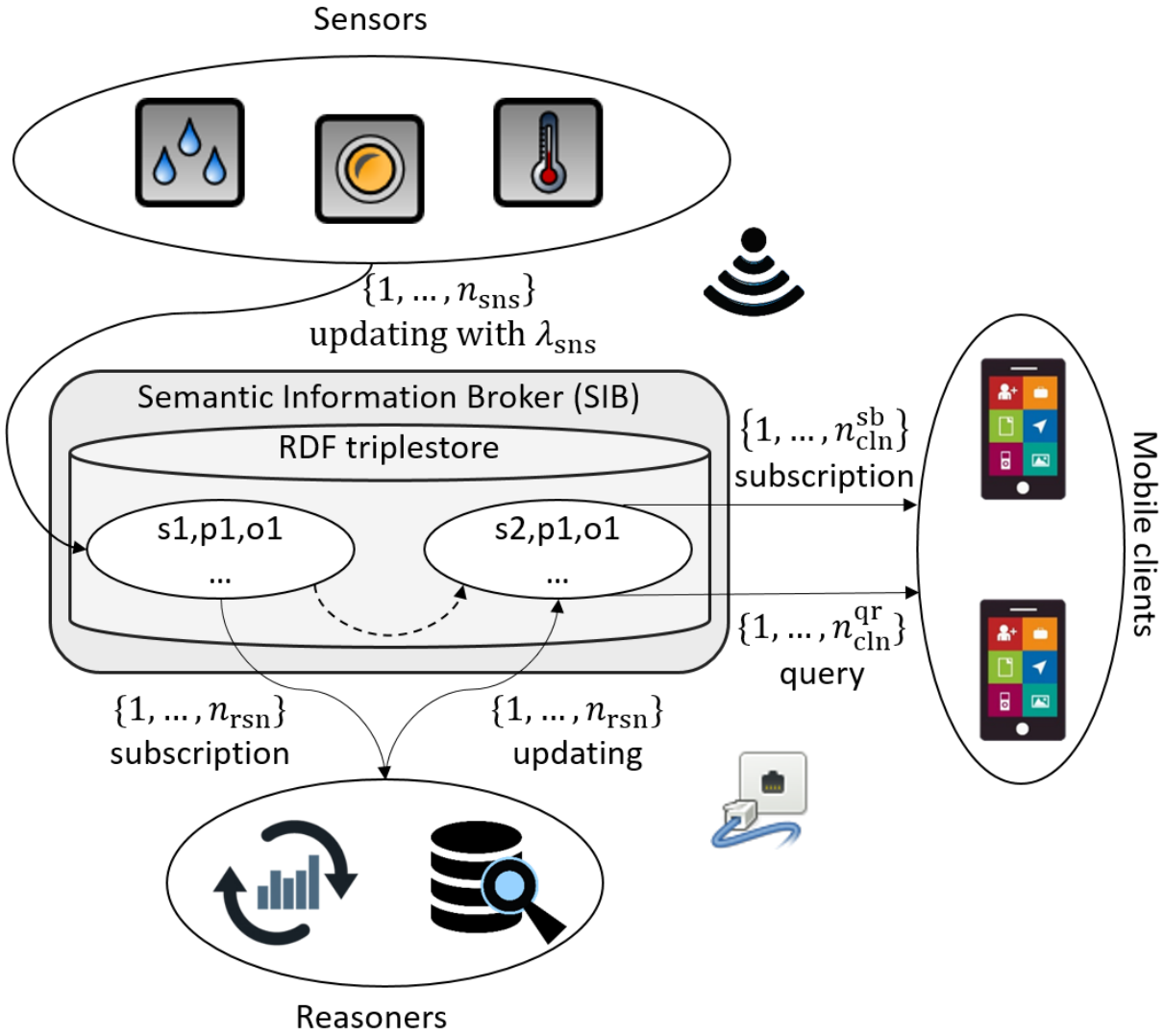


CuteSIB usage in IoT scenarios

Big amount of low-capacity and heterogeneous devices are interacting

- Typical agent roles:
  - Sensor — collects user context and periodically provides data
  - Reasoner — data processing to deduce new knowledge and facts
  - Client — receives the result of processing
- Real system examples:
  - SmartRoom: sensors collects data about Wi-Fi activity of participants, presence service processes this data and provide activity charts to clients
  - e-Tourism: POI equipped with sensors to provide information about it and current environment (e.g. visitor density)
  - m-Health: Users wear medical devices (e.g. cardio sensor), data from devices provides to processing service and to doctor on the other end

Simulation Model: I



- Every agent on its own thread
- Reasoners subscribed on data from subgroup of sensors
- Clients of different types aren't mixed
- Moderate-capacity computer to host SIB

Simulation Model: II

- Agents proportion:

$$n_{\text{rsn}} : n_{\text{cln}}^{\text{sb}} + n_{\text{cln}}^{\text{qr}} : n_{\text{sns}} = 1 : 10 : 100.$$

- Evaluated values:
  1. Sensor: processing time t_{sns} to update data
  2. Reasoner: processing time t_{rsn} from receiving data and update it
  3. Client (subscription): time $t_{\text{cln}}^{\text{sb}}$ for delivery notification from sensor to client
  4. Client (query): processing time $t_{\text{cln}}^{\text{qr}}$ to retrieve updates
- Delay between updates $delay_{\text{sns}}$ by sensors selected uniformly from $(0, 2n_{\text{sns}}/\lambda)$
- Delay between queries $delay_{\text{cln}}^{\text{qr}}$ by clients selected uniformly from $(0, 2n_{\text{cln}}^{\text{qr}}/\lambda)$

Simulation Testbed

- Agents implemented in Python 2.7 with Smart-M3 PythonKPI library
- All agents work in local network
- One agent is equal to one thread

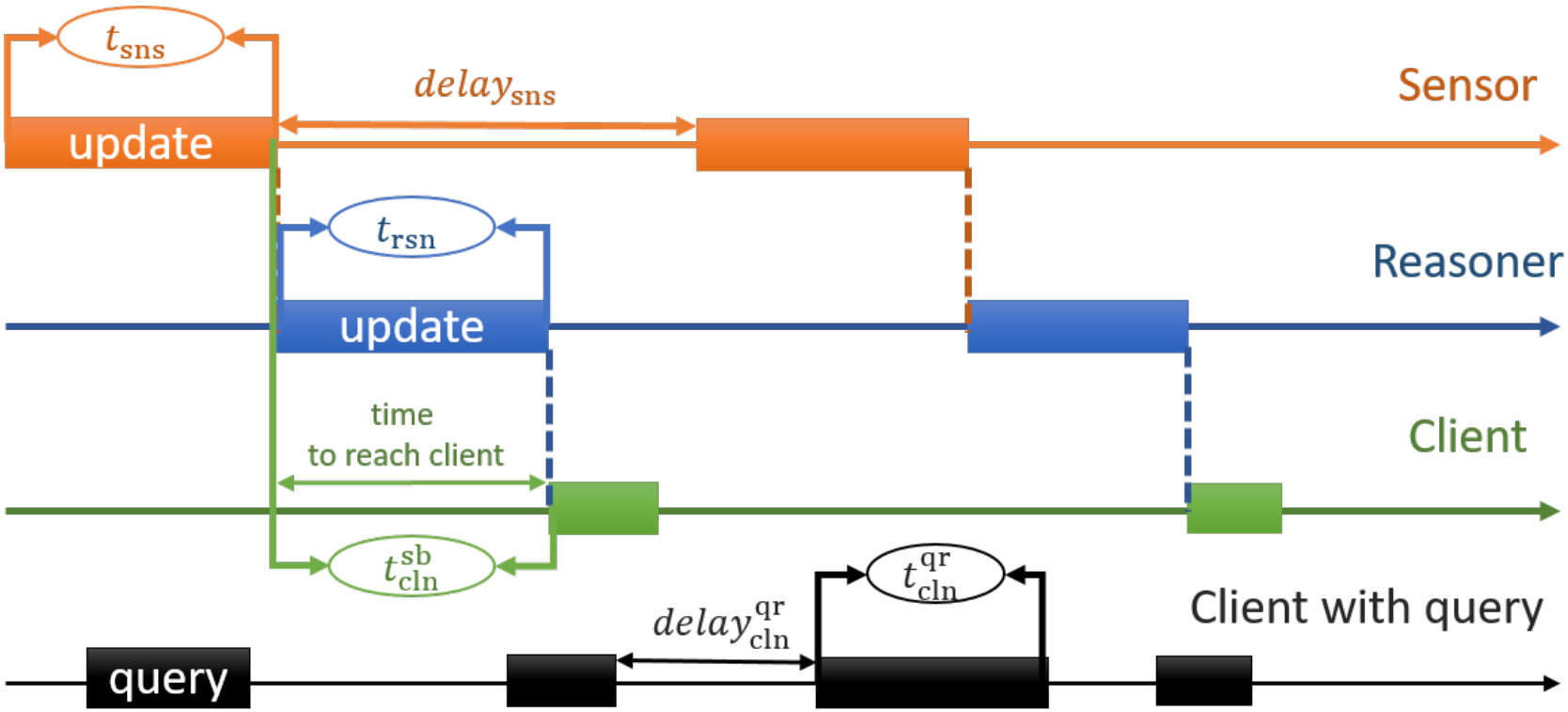
Functional role	Capacity	Device specification
SIB host machine	3 000 subscription	CPU Intel Core i3, CPU 1.90 GHz, RAM 4Gb, wired connection with 100 Mbps, Ubuntu 15.10
Sensors KP	2 machine 10 000 agents	CPU Intel Dual Core, CPU 2.60 GHz, RAM 2Gb, wired connection with 100 Mbps, XUbuntu 16.04
Reasoner KP	1 machine 100 agents	CPU Intel Core i5, CPU 1.70 GHz, RAM 6Gb, wireless connection with 21 Mbps, Ubuntu 15.10
Mobile Client KP	1 machine 10 000 agents	CPU Intel Core i5, CPU 2.50 GHz, RAM 3Gb, wireless connection with 21 Mbps, XUbuntu 16.04
Test agent	1 machine 4 agents	CPU Intel Core i5, CPU 2.50 GHz, RAM 3Gb, wireless connection with 21 Mbps, XUbuntu 16.04

Evaluation Approaches

I Determining of the upper bound for the sum operation rate with variation of agents number

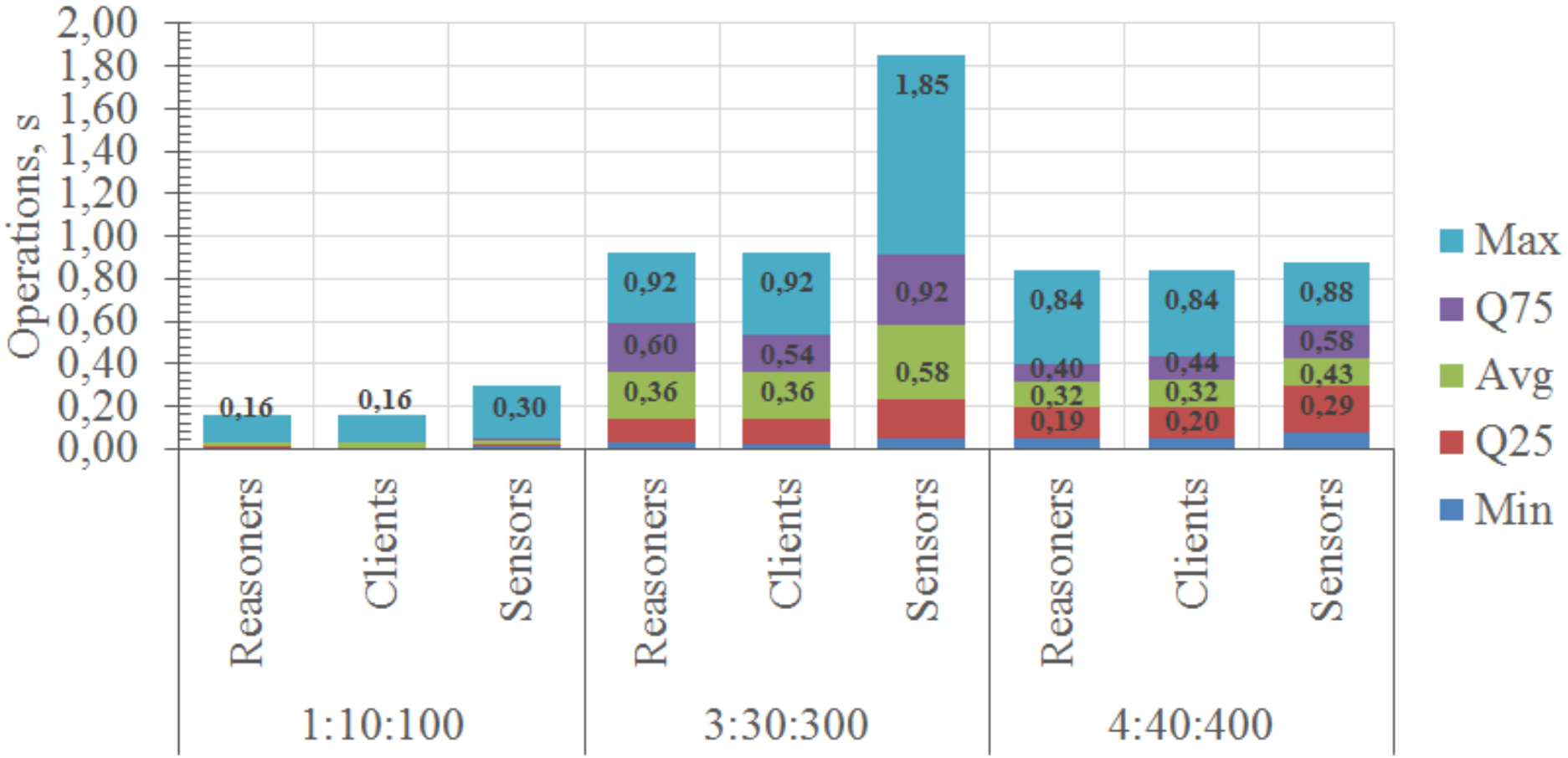
II Analyzing the influence of device group size in conjunction with increasing the sum operation rate to the SIB

III Scalability evaluation with replacing subscription on clients by query



Experiment I

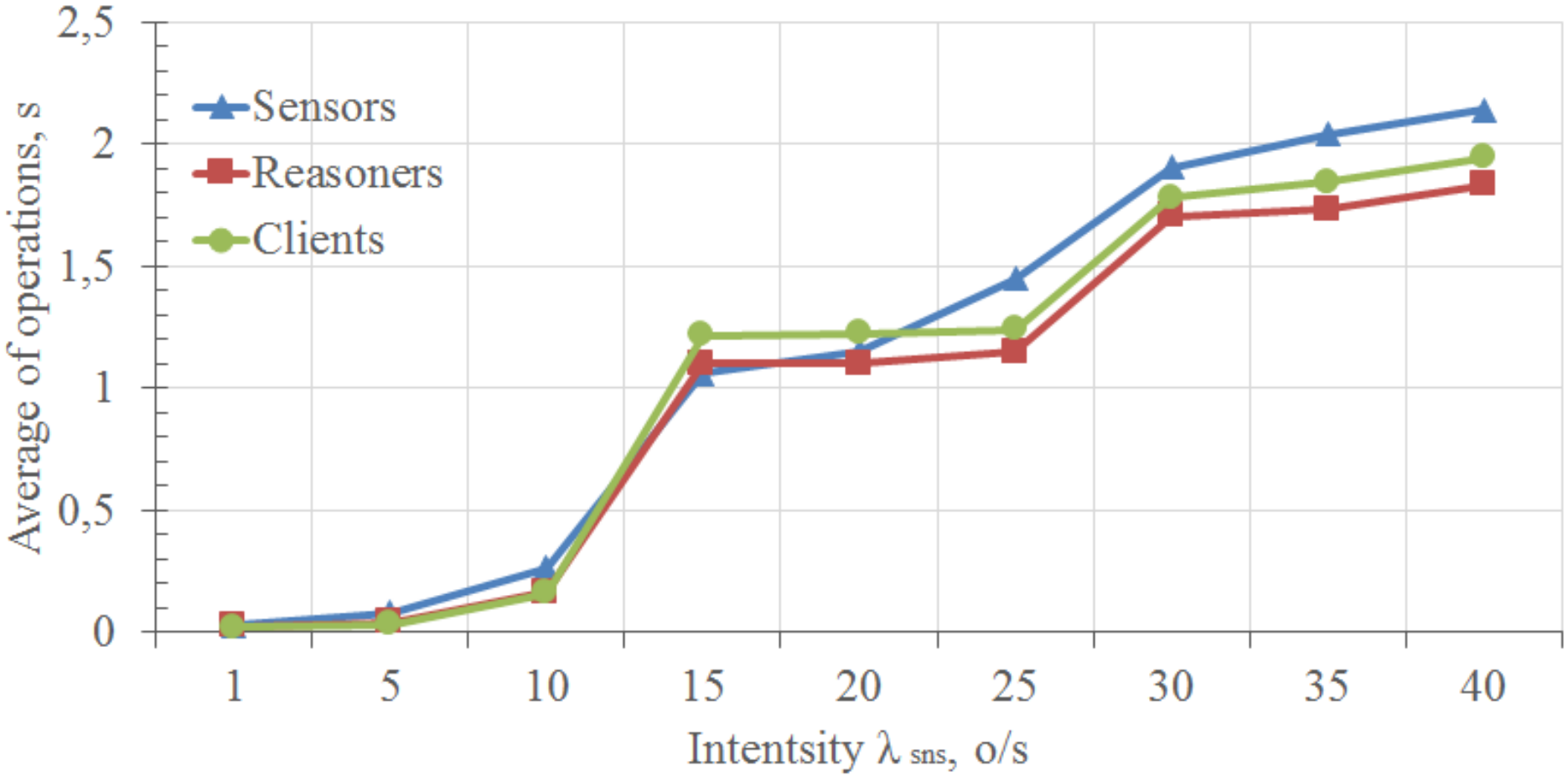
Variation of devices group size with fixed delay



80 operation per second in average for 444 agents

Experiment II

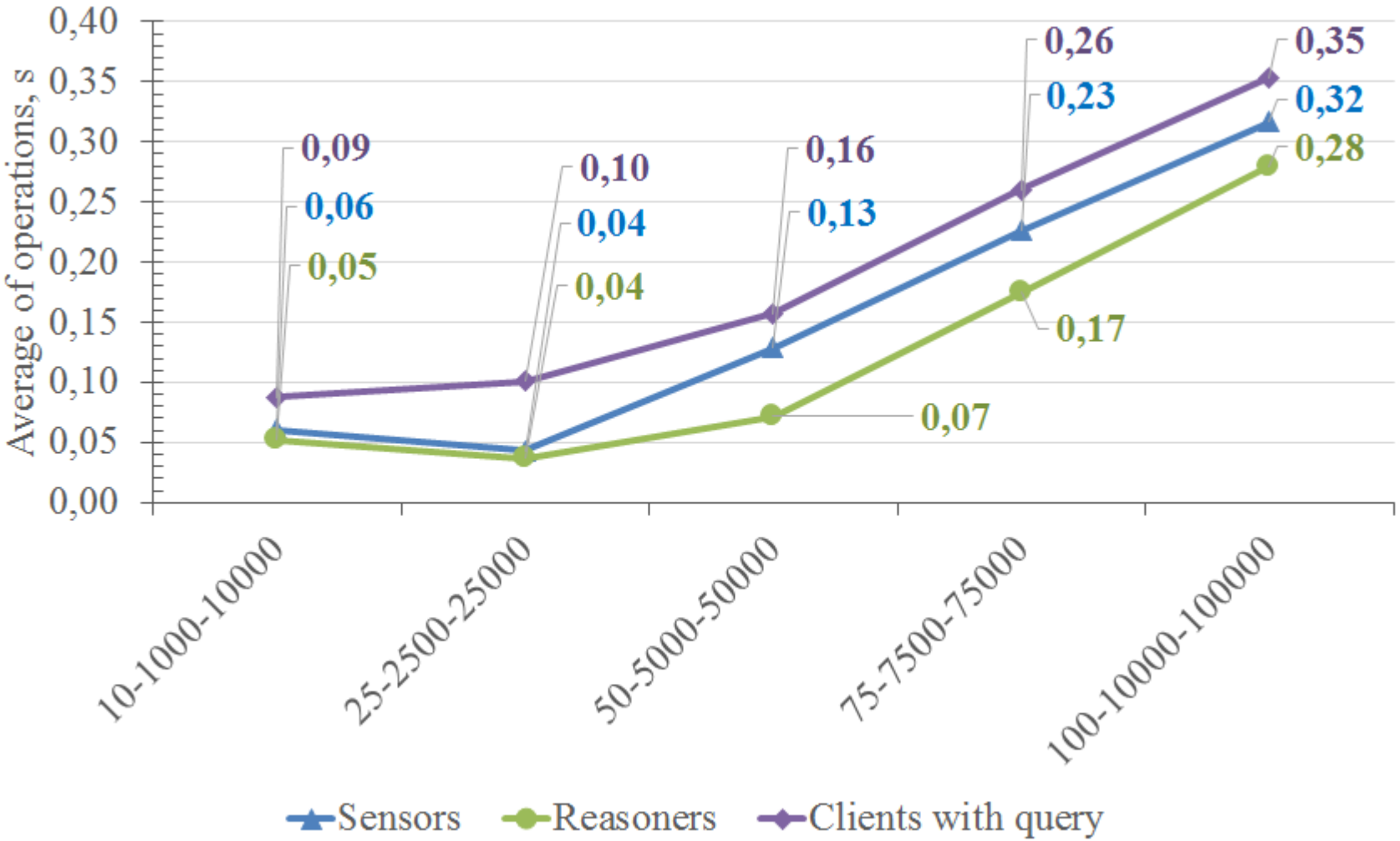
Influence of device group size (10:100:1000) in conjunction with increasing the sum operation rate to the SIB



Performance degrade with rate higher than 15 operation per second

Experiment III

Sensors simulated as 5 agents on one thread with low rate (5 operation per second)



Conclusion

- CuteSIB implementation is basically suitable for localized IoT environments
- Most expensive operations are persistent queries such as subscription
- Optimization of subscription processing on SIB is needed for large agents numbers
- Clients with queries make smaller load on SIB

Thank you for attention

E-mail: vdovenko@cs.karelia.ru