Performance Scalability Study of the Smart-M3 CuteSIB Implementation

Andrey S. Vdovenko, Ivan V. Galov, Dmitry G. Korzun
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{vdovenko, galov, dkorzun}@cs.karelia.ru

Abstract—**According to the M3 architecture (multidevice, multivendor, multidomain), a smart space is created by deploying a Semantic Information Broker (SIB) in a given Internet of Things (IoT) environment. The deployed SIB enables information sharing for all digital devices participating in the IoT environment. In this study, we consider CuteSIB—a new SIB implementation for the Smart-M3 platform, where the class of SIB host devices covers such Qt-based equipment as common computers (e.g., laptops and personal computers), embedded computers (e.g., single-board computers and wireless routers), and personal mobile computers (e.g., smartphones and tablets). Our study focuses on performance evaluation of the CuteSIB operation with a multitude of devices that simultaneously participate in data sharing, information processing, and service delivery. Our simulation experiments applies a scalability model for a smart space application with the three functional groups of participating devices: sensors, reasoners, and mobile clients. In the experiments, the number of devices is varied in each group as well as the parameters of interaction behavior with SIB. The evaluation shows the scalability level that CuteSIB can achieve being deployed on a host device of moderate capacity in an IoT environment.**

I. Introduction

Smart spaces form a programming paradigm, which is now augmented with the rapidly advancing suit of information and communication technologies, for creating a certain class of ubiquitous service-oriented environments—smart or intelligent environments [1], [2]. Such an environment is typically associated with a physical spatial-restricted place (office, room, home, city square, etc.) and equipped with a variety of devices (sensors, data processors, actuators, consumer electronics, personal mobile devices, multimodal systems, etc.).

Smart spaces are based on two innovative concepts [3]: the Semantic Web (SW) and the Internet of Things (IoT). The SW technology stack is primarily composed by technologies allowing the representation (RDF, RDFS, OWL) and retrieval (SPARQL) of semantically annotated data [4]. The IoT concept (e.g., see [5]) is a large-scale evolution of the innovative vision of Mark Weiser about ubiquitous computing [6]: the Internet, in addition to personal desktops and mobile computers, is also populated with billions of heterogeneous interconnected smart devices, which represent (and advance) physical things. Everyday life objects, alongside traditional computers, become data processors and service constructors to their users [7], [8].

Both SW and IoT form a vast research area characterized by a high interdisciplinary level, a high process dynamicity, and heterogeneity of the involved devices and applications. Even if we limit our consideration with the Smart-M3 platform [9] for creating smart spaces, a wide range of application domains is covered [10]: from collaborative work environments and electronic health to cybermedicine, from electronic tourism and cultural heritage education to smart cities, from transport logistics and Industrial Internet to socio-cyber-physical systems, and many more.

One of the IoT-enabled use cases is a system consisting of sensors, reasoners, and mobile clients. In particular, this use case is typical for collaborative work environments such as the SmartRoom system [10] and for mobile healthcare [11]. Our study focuses on performance scalability evaluation of CuteSIB—an implementation variant of Smart-M3 SIB with the focus on a wide spectrum of Qt-based IoT devices [12].

Previous study [3] showed outperformance of CuteSIB compared with existing Smart-M3 SIB implementations. That comparison was for typical scenarios without stress workload. In this work, we simulate concurrent participation of a multitude of sensors, reasoners, and mobile clients in the same smart space. The performance evaluation focuses on determining the capacity bounds when the amount of participants is growing and reaching the stress workload that exhausts the SIB host device capacity.

The rest of the work is organized as follows. Section II describes smart spaces application development using the Smart-M3 platform. Section III introduces our simulation model for evaluating CuteSIB. Section IV provides results of performance scalability evaluation. Section V summarizes our experimental conclusions.

II. Devices in a smart space application

Based on the Smart-M3 platform [9] a smart space allows its participating devices to communicate with each other using the Semantic Web methods and the subscription operation. An IoT environment can include extremely many devices (small or large) and there is still no wide-spread effective approach for solving large multi-party interaction [13], [14]. Participation of large amounts of capacity-heterogeneous devices leads to performance degradation and low scalability of many existing software development methods for IoT environments.

Many devices are sensors or sensing devices that collect user context. Such devices periodically provide new data and make updates in the smart space. Another class of devices that perform data processing to deduce new knowledge and facts that could be useful to the end-user. An important class of devices are for end-users clients (the typical case is personal mobile devices such as smartphones). Basically, they should receive the result of processing and make it available for the end-users (e.g., visualization). Additionally, a client can also participate in service construction together with other devices as well as interacting with some other mobile clients.

The recent study of Smart-M3 platform implementations has shown the advantages of the CuteSIB platform over others [3]. In this work, we evaluate the CuteSIB performance for operation in IoT environments with many concurrently participating devices. CuteSIB targets even low-capacity devices with limited computing, storage, and network capabilities. The problem of this experimental evaluation is to quantitatively determine the scalability bounds of CuteSIB when the number of devices is growing.

III. Simulation model

Organization of a real-life experiment testbed with large amount of participating devices is expensive. Instead, we apply the simulation modeling A small set of desktop computers are used to allocate many parallel processes, where each process corresponds to a single agent acting as a Knowledge Processor (KP). Each KP simulates the activity for device from one of the following three groups as Fig. 1 shows.

- Sensors that are regular publishers of specified small data fragments.
- Reasoners that observe sensed data as a whole and extract information as a service from this fragmented data corpus.
- Mobile clients that detect the activity of reasoners to react on new information appearance.

Each sensor makes (regularly and randomly) update operation of its data values, which match the interest of some reasoners. The update rate of a sensor is λ_{sns} (operations per second, s^{-1}). The time between two consecutive updates is selected uniformly at random. There are n_{sns} sensors in total,

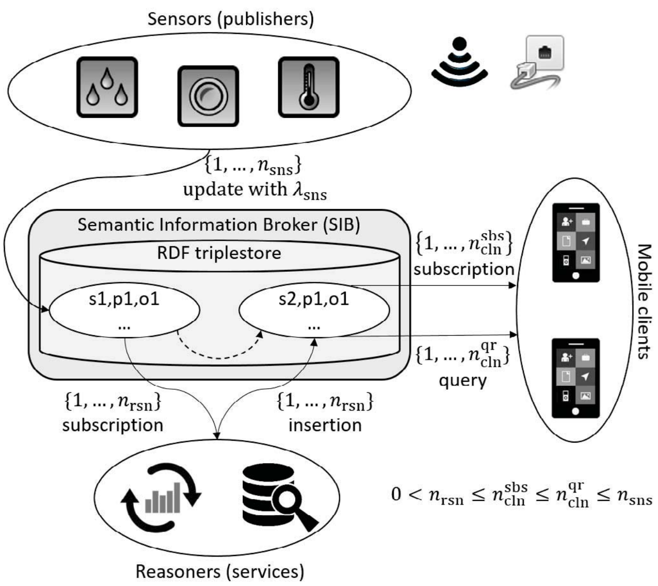

Fig. 1. Simulation model for a smart space application

where n_{sns} can be large. The assumption reflects that sensors are responsible for the data workload to the smart space.

Each reasoner is subscribed on its own part of data concurrently updated by a large subgroup of sensors. Whenever some sensor is making an update in this part then the reasoner is notified to construct its service, i.e., the reasoner reads the updated data from the smart space, makes their local processing, and publishes the new derived information in the smart space. There are n_{rsn} reasoners in total, where $n_{\mathrm{rsn}} \ll n_{\mathrm{sns}}$. The assumption reflects that reasoners are responsible for the workload of tracking the smart space.

Each mobile client is interested in new information provided by any reasoner (as a service). In dependence on the detection way of such an event, mobile clients are categorized on two types: 1) with explicit use of the subscription operation and 2) active query to periodically check the service availability. first subscribes on all reasoners and second uses periodical query. There are $n_{\mathrm{cln}}^{\mathrm{sb}}$ and $n_{\mathrm{cln}}^{\mathrm{qr}}$ clients in total, where

$$n_{\mathrm{rsn}} \ll n_{\mathrm{cln}}^{\mathrm{sb}} + n_{\mathrm{cln}}^{\mathrm{qr}} \ll n_{\mathrm{sns}}.$$

In the experiments, we do not mix clients of the different types, i.e., we assume that either $n_{\mathrm{cln}}^{\mathrm{sb}} > 0$ and $n_{\mathrm{cln}}^{\mathrm{qr}} = 0$ or $n_{\mathrm{cln}}^{\mathrm{sb}} = 0$ and $n_{\mathrm{cln}}^{\mathrm{qr}} > 0$.

To reflect the settings of a smart space application, our simulation model assumes the following proportion:

$$n_{\mathrm{rsn}} : n_{\mathrm{cln}}^{\mathrm{sb}} + n_{\mathrm{cln}}^{\mathrm{qr}} : n_{\mathrm{sns}} = 1 : 10 : 100. \quad (1)$$

That is, a large number of sensors feeds the smart space with raw data. The latter are source for a small number of reasoners to construct services by tracking updates and processing these raw data (one service—many sensed data items). The number of mobile clients is in the middle such that each service is provided to several end-users.

We consider the case of a low- or moderate-capacity computer to host a SIB. (The typical case is a desktop or a laptop.) This assumption leads to a limited number of simultaneous network connections from KPs to the SIB. In particular, the Smart-M3 platform primarily support TCP connections. Depending on a programming technique, a given KP can either establish one connection for many operations with SIB or the KP establishes (and then closes) a separate connection for every operation with SIB. In both cases, the parameter $\lambda = n_{\mathrm{sns}}\lambda_{\mathrm{sns}}$ cannot be made high for the considered class of SIB host computers.

Therefore, we can use the two key scalability variations. First, n_{sns} is increased while preserving the sum update rate λ for sensors in reasonable bounds. In particular, the delay between updates for a sensor is selected uniformly from $(0, 2n_{\mathrm{sns}}/\lambda)$. For an example, if $n_{\mathrm{sns}} = 500$ sensors and $\lambda = 10\ \mathrm{s}^{-1}$ then the delay is in $(0, 100)$ s. Second, λ_{sns} is varied for fixed n_{sns}. The latter cannot be large in this case.

IV. Experiments

Our simulation experiments use four modest-capacity computers, see their specification in Table I. We experimented with CuteSIB version 0.5.0. Local wireless network is primarily used except the computer for simulating the reasoners (it

TABLE I. COMPUTERS TO ALLOCATE SIMULATED KPS

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

TABLE II. AVERAGE OPERATION PROCESSING TIME FOR DIFFERENT SIZES OF DEVICE GROUPS

Group	Size proportion – #sensors : #reasoners : #clients		
	1 : 10 : 100	3 : 30 : 300	4 : 40 : 400
Sensors	0,016	0,620	1,796
Reasoners	0,018	0,761	1,914
Clients	0,020	0,641	1,914

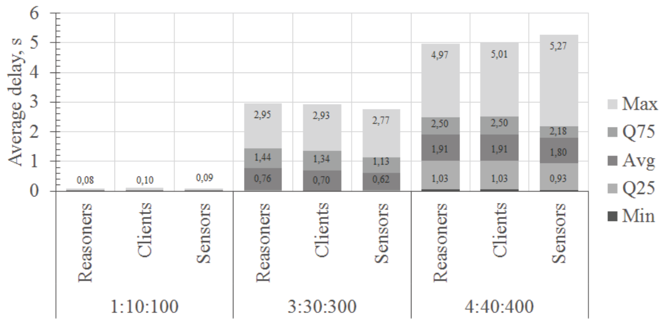

Fig. 2. Operation performance for different sizes of device groups

uses wired connection). All simulated KPs (sensors, reasoners, and clients) are implemented using Python 2.7 language and PythonKPI used for SIB access primitives.

For each KP group one KP is selected (a typical agent) that calculate the time needed to perform an operation with the SIB.

Experiment I: The number of KPs is varied keeping proportion (1). We evaluate the upper bound for the sum rate (operations per second) to the SIB influenced by all three groups of KPs. Mobile clients use subscription in this experiment. Table II shows the results for the fixed delay time $\tau = 10$ s between consequent updates for every sensor. For an example in the largest case, the sum SIB workload is 80 s^{-1}, which is made by 444 KPs (400 sensors, 40 clients, and 4 reasoners). The average processing time is 1.8 s for an operation. Additional statistical measures are shown in Fig. 2.

Experiment II: We analyze the role of device group size in conjunction with increasing the sum operation rate to the SIB. Mobile clients do not use subscription in this experiment. Table III shows the average operation processing time (in seconds). For example the rate 20 s^{-1} of 555 agents leads to the average processing time 0,03 s. While 1110 agents with the same sum rate lead to the average time 0,08 s. The dependency on the rate is shown in Fig 3. We also found that for relatively high load (e.g., 50 s^{-1}) subscription notifications to reasoners are typically received earlier than response to sensors for updates.

The performance degrades when the sum rate exceeds 30 operations per second. This observation is the result of the

TABLE III. AVERAGE OPERATION PROCESSING TIME FOR DIFFERENT SUM RATES TO SIB

Agent	Size proportion – #sensors : #reasoners : #clients / rate			
	5 : 50 : 500 / 20 s^{-1}	10 : 100 : 1000 / 20 s^{-1}	5 : 50 : 500 / 50 s^{-1}	50 : 500 : 5000 / 5 s^{-1}
Sensors	0,037	0,089	2,703	16,548
Reasoners	0,037	0,085	3,068	2,390
Clients	0,014	0,063	2,620	8,875

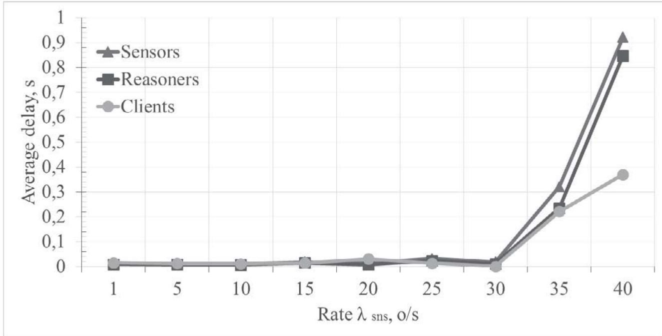

Fig. 3. Operation performance in dependence on the sum rate to SIB

capacity of the SIB host machine. The latter cannot maintain many network connections simultaneously. Another reason is ineffective implementation of subscription operation in SIB. The recent version of CuteSIB straightforwardly inherits the code from RedSIB, and further development is needed to optimize subscription maintenance in SIB.

V. CONCLUSION

This work showed that CuteSIB implementation is suitable for creating smart spaces in resource-restricted and localized IoT environments. When compared with other Smart-M3 SIB implementations, the performance and capacity bounds are higher. Furthermore, the higher dependability level is achieved when SIB resists stress workload (although with exhausted capacity) and continues the operation after the workload reduction. The most expensive is persistent queries as the subscription operation, and further development is needed to optimize the SIB performance for large amount of simultaneous subscriptions. Nevertheless, when the number of subscription is moderate the SIB performance is reasonable for typical workload generated in a localized IoT environment.

ACKNOWLEDGMENT

This research is supported by the Ministry of Education and Science of Russia within project # 2.5124.2017 of the basic part of state research assignment for 2017–2019.

REFERENCES

[1] J. Augusto, V. Callaghan, D. Cook, A. Kameas, and I. Satoh, “Intelligent environments: a manifesto,” *Human-centric Computing and Information Sciences*, vol. 3, no. 1, pp. 1–18, 2013.

[2] D. Korzun, “On the smart spaces approach to semantic-driven design of service-oriented information systems,” in *Proc. 12th Int’l Baltic Conf. on Databases and Information Systems (DB&IS 2016)*, G. A. et al., Ed. Springer International Publishing, Jul. 2016, pp. 181–195.

[3] F. Viola, A. D'Elia, D. Korzun, I. Galov, A. Kashevnik, and S. Balandin, "The M3 architecture for smart spaces: Overview of semantic information broker implementations," in *Proc. of the 19th Conference of Open Innovations Association FRUCT*, S. Balandin and T. Tyutina, Eds. FRUCT Oy, Helsinki, Finland, Nov. 2016, pp. 264–272.

[4] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, pp. 34–43, May 2001.

[5] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[6] M. Weiser, "The computer for the twenty-first century," *Scientific American*, vol. 265, no. 3, pp. 94–104, 1991.

[7] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the Internet of Things," *IEEE Internet Computing*, vol. 14, no. 1, pp. 44–51, Jan. 2010.

[8] D. Korzun, S. Balandin, and A. Gurtov, "Deployment of Smart Spaces in Internet of Things: Overview of the design challenges," in *Proc. 13th Int'l Conf. Next Generation Wired/Wireless Networking and 6th Conf. on Internet of Things and Smart Spaces (NEW2AN/ruSMART 2013), LNCS 8121*, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer, Aug. 2013, pp. 48–59.

[9] J. Honkola, H. Laine, R. Brown, and O. Tyrkkö, "Smart-M3 information sharing platform," in *Proc. IEEE Symp. Computers and Communications (ISCC'10)*. IEEE Computer Society, Jun. 2010, pp. 1041–1046.

[10] D. Korzun, A. Kashevnik, S. Balandin, and A. Smirnov, "The Smart-M3 platform: Experience of smart space application development for Internet of Things," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems. Proc. 15th Int'l Conf. Next Generation Wired/Wireless Networking and 8th Conf. on Internet of Things and Smart Spaces (NEW2AN/ruSMART 2015), LNCS 9247*, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer, Aug. 2015, pp. 56–67.

[11] D. G. Korzun, "Internet of things meets mobile health systems in smart spaces: An overview," in *Internet of Things and Big Data Technologies for Next Generation Healthcare*, ser. Studies in Big Data, C. Bhatt, N. Dey, and A. S. Ashour, Eds. Springer International Publishing, 2017, vol. 23, pp. 111–129.

[12] I. Galov, A. Lomov, and D. Korzun, "Design of semantic information broker for localized computing environments in the Internet of Things," in *Proc. 17th Conf. of Open Innovations Association FRUCT*. ITMO University, IEEE, Apr. 2015, pp. 36–43.

[13] A. Whitmore, A. Agarwal, and L. Da Xu, "The internet of thingsa survey of topics and trends," *Information Systems Frontiers*, vol. 17, no. 2, pp. 261–274, 2015.

[14] D. Korzun, A. Borodin, I. Paramonov, A. Vasiliev, and S. Balandin, "Smart spaces enabled mobile healthcare services in internet of things environments," *International Journal of Embedded and Real-Time Communication Systems (IJERTCS)*, vol. 6, no. 1, pp. 1–27, 2015.