Cross-platform system to integrate IoT platforms for cross domain applications

Thanh Do Chi, Quan Dinh Huu Hai and Minh Nguyen Binh

Ha Noi University of Science and Technology, Viet Nam
{thanh.do-chi,quan.dinh-huu-hai,minh.nguyen-binh}@gmail.com
http://www.soict.hust.edu.vn

Abstract. The Internet of Things (IoT) is evolving very quickly. In this IoT world, there are massive of sensors and devices. To management these sensors and devices efficiently, we need IoT platforms, each often suit to a given scenario and use different kind of communication, device control protocols. Because IoT platforms are heterogeneous, which ones usually cannot communicate to each other, the problem of interoperability these IoT platforms is one of the most important and challenging part of IoT. In this paper, we propose a cross-platform layer, which allow integrate new IoT platforms easily, enable interoperability between platforms and also provide APIs for developers create innovative and cross-domain applications.

1 Introduction

The Internet of Thing (IoT) is an increasingly hot topic in the tech world. It is part of the theme of digitization. Different types of technology are combining to create digital ecosystems that have potential to drastically improve business processes. There are few interesting statistic on IoT, provided by CMO.com [2].

- In 2008, there were already more ẗhingsc̈onnected to the Internet than people. By 2020, the amount of internet-connected things will reach 50 billion, with $19 trillion in profits and cost savings coming from IoT over the next decade.
- A whopping 94% of all businesses have seen a return on their IoT investments.
- GE estimates that convergence of machines, data, and analytics will become a $200 billion global industry over the next three years.

That interesting IoT industry attract companies to build IoT platforms to manage their own devices and sale to customers. Therefore, there are heterogeneous IoT platforms. Those platforms usually domain-specific, use different of communication technology, represent the devices in different ways, ... According to German IoT market research firm IoT Analytics, the number of IoT platforms on the market has surged 25 percent in just 12 months and now stands at 450 (July 2017) [3].But large number of IoT platforms make it difficult to reuse IoT resources, data between related systems, make it harder to build a IoT ecosystem. An example, a temperature sensor in the house may be managed and seen differently by difference platforms, so it's make redundance of data, hard to control in cross-domain applications.

As the need of building cross-domain applications increasingly, it's raising the need of integrate IoT platforms to a unified cross-platform layer for saving resources, easier to devolope cross-domain applications. To address above mentioned limitations, in this paper, we propose a unified model and an experiment setup result.

This rest of this paper is structured as follows: In section 2, we consider some of the related work about integration of IoT platform. Section 3 introduces out cross-platform model including the architecture of the model, the ontology to presents the sematic of components and the resource graph of a testbed. Section 4 provides result of out experience. Finnaly, we summary the experience and the future work in section 5.

2 Related Work

Integration is not a new problem in IoT world, there are many research about this area. We will devide them into two aspect of the problem. The first one is architecture of the system, how they build the system, which components do the system have. The second ones see the integration in aspect of ontology, sematic meaning. In each aspect, a lot of studies were introduced to find the unified model for IoT world.

On architecture aspect, the Internet of Things Architecture project (IoT-A) [4] is proposing an architecture reference model for IoT interoperability together with components of future IoT to enable search, discovery and interact as one coherent network.

The IETF [5], [6] community has been involved in foundational IoT technologies such as IPv6 and the Constrained Application Protocol (CoAP), focusing on getting constrained devices and sensor networks connected to the Internet. Similarly, the IEEE [7] has several protocol standards that form the foundation of the IoT and provid connectivity between things and the Internet.

HYPERCAT use the concept hub equivalent with IoT platform. It's proposed a four stage path toward interoperability between IoT hubs: 1. **IoT Core.** Hubs expose things and associated metadata using the web architecture and RESTful web services - a web of things; 2. **IoT Model.** Agreement on basic approaches and models requiring a common understanding of what things and associated data a hub should contain. Achieving this stage will facilitate the development of adapters and other integration tools for hub interoperability; 3.**IoT Hub.** Agreement on certain implementation issues such as concrete representations, URLs and schema for describing and querying catalogs and data from hubs. This will include support for security mechanisms so that hubs can control access to things and offer some guarantees over who is providing things and their data and who is able to access and use these resources. 4.**IoT Profiles.** Agreement on the semantics of things and their associated data exposed on a hub. For example: a temperature sensor in one hub provides the same quality and value of temperature as one in another hub. Essentially the taxonomy of things and the ontological models that hubs support will need to be defined. By reaching agreement at this level, deep integration of application is possible, allowing hubs and things to link to and communicate directly with each other.

Although these work are making IoT interoperability closer, but some has it's limitation. Such as IETF and IEEE primarily focus on connect things to the internet. IoT-A project is a ambitious standardization effort but it's too big, maybe attempting to lock down aspect of IoT ecosystem while it is still evolve rapidly. Developers and vendors however, may favour less complex approaches that address requirements as they emerge. HYPERCAT according our opinion, follow the best practices and our work are independent but quite homologous

with it. But HYPERCAT has some problems : platforms/hubs in the project are closed platform, it's make developer harder to reimplement to paper. Our proposal are implemented on three open source platforms are Home Assitant [8], OpenHAB [9], ThingsBoard [10] which are the most popular opensource IoT platforms, large commudity and well documented.

Ontology ...
Ontology ...
Ontology ...

Ontology ...
Ontology ...
Ontology ...

Ontology ...

Ontology ...

3 Cross-platform model

3.1 Deployment Architecture

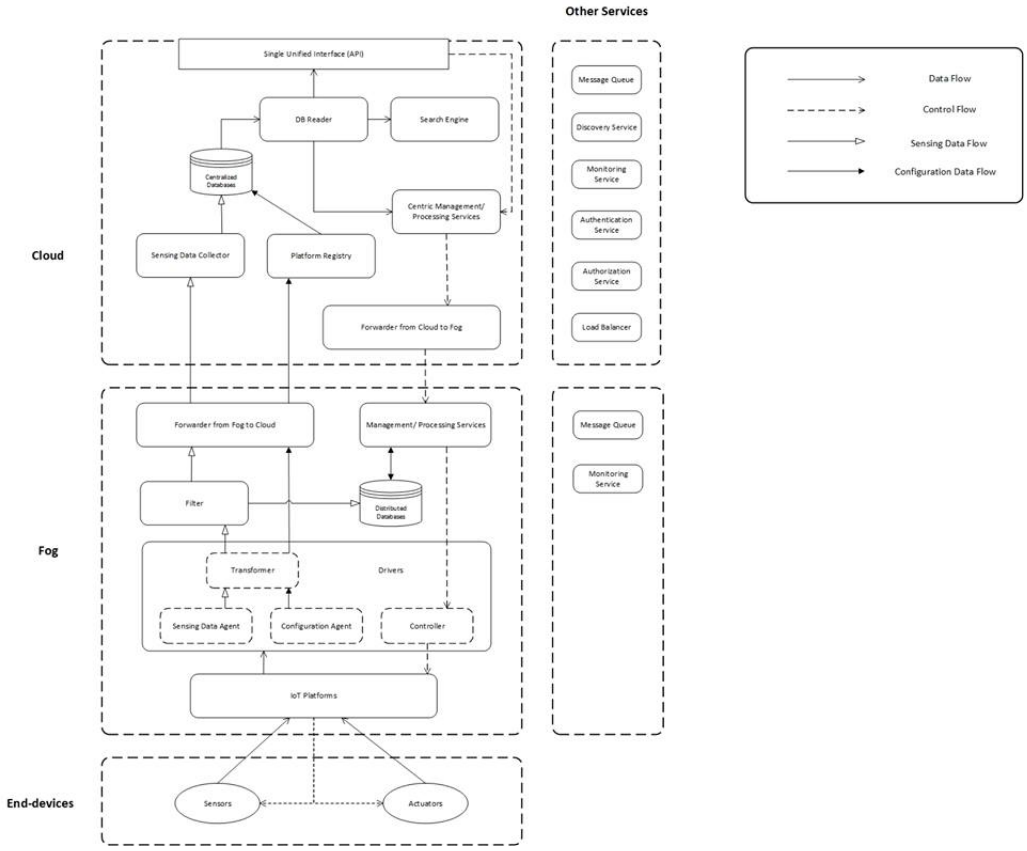

Fig. 1. Deploy architecture

The deployment architecture has three layers, including:

- `End-Devices`. This layer contains every IoT Things: sensors, intelligence devices, actuators.
- `Fog` IoT devices are sending data high frequently, so it is usually that data will not send directly to cloud but to fog layer cause of fog layer nearer to the IoT devices will help reduce the cost of bandwith, storage and reduce the delay between cloud and devices. IoT platforms installed on the Fog layer to manage the devices. To build a system integrate many platforms, we write a driver for each platform to tranforms the data models, generated

and APIs which platforms provided into unified data and API. For further purpose, some other modules also implemented such as Filter (to filter the data), Forwarder (to forward the data from Fog to Cloud), ...

– `Cloud` Fog layer often has low computational capacity, so we need to build main processing modules on the Cloud. Cloud layer use data received from `Fog` to manage information and data from platforms, devices in platforms. Once a platform join the system, it have to send meta data about it to Registry module. Registry will check if it had information of this platform, if not, Registry will provides unique ID for all devices in the platform. All devices data and platform information stored in a database. Also, Single Unified Interface converts APIs of different platforms into unified API.

System's modules had built following microservices architecture so it's flexible to remove, add and change a service.

3.2 Ontology

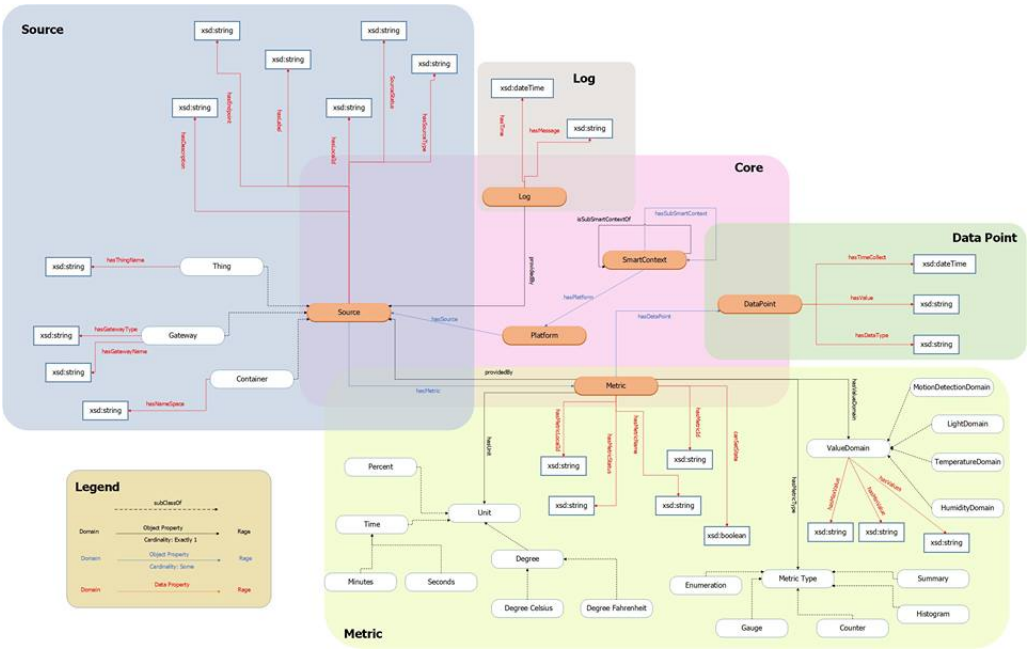

Fig. 2. Cross-platform ontology.

– **Smart Context** is the root of the graph, which is a testbed implemeted of the cross-platform system. It can be a smart home, smart building, smart city, ... One `Smart Context` maybe is a sub smart context of another smart-context; and a smart context maybe has one or many sub smart context.

- **Platform** is `a multi-layer technology that enables straightforward provisioning, management, and automation of connected devices within the Internet of Things universe. It basically connects your hardware, however diverse, to the cloud by using flexible connectivity options, enterprise-grade security mechanisms, and broad data processing powers` [1].
- **Source** is physical or vitrual devices. Source generate data about evironment or anything it monitor. A **Source** might be a **Thing**, a **Gateway** or a **Processing Unit**.
- **Thing** is a device which is a set of sensors.
- **Gateway** is a physical device or software program that servers as the connection point between the cloud and controllers, sensors and intelligence devices.
- **Processing Unit** is a virtual device to monitor devices's resource in the IOT system like CPU, memory, storage, ...
- **Log** is the data generated from **Sources**. Log file can use to extract historical information, monitor devices's status, use in alert services and rule engine, ...
- **Metric** is used to express diverse metric types of the collected data such as **Enumeration**, **Gauge**, **Counter**, **Histogram**, **Summary**. Each Metric has it's own **Unit** which might be **Percent**, **Time**, **Degree**. Metric connected to Source mean that each Source correspond to a Metric.
- **Data Point** is generated when a **Source** active. Based on `Metric` information of a **Source**, data generate will have suitable DataType and DataValue.

3.3 Resource Graph

Once we have the cross-platform system, we will need to integrate testbeds with it. One of the first step we need to care about is registration all its devices and services. The figure below show the resource graph from SmartBuilding testbed.

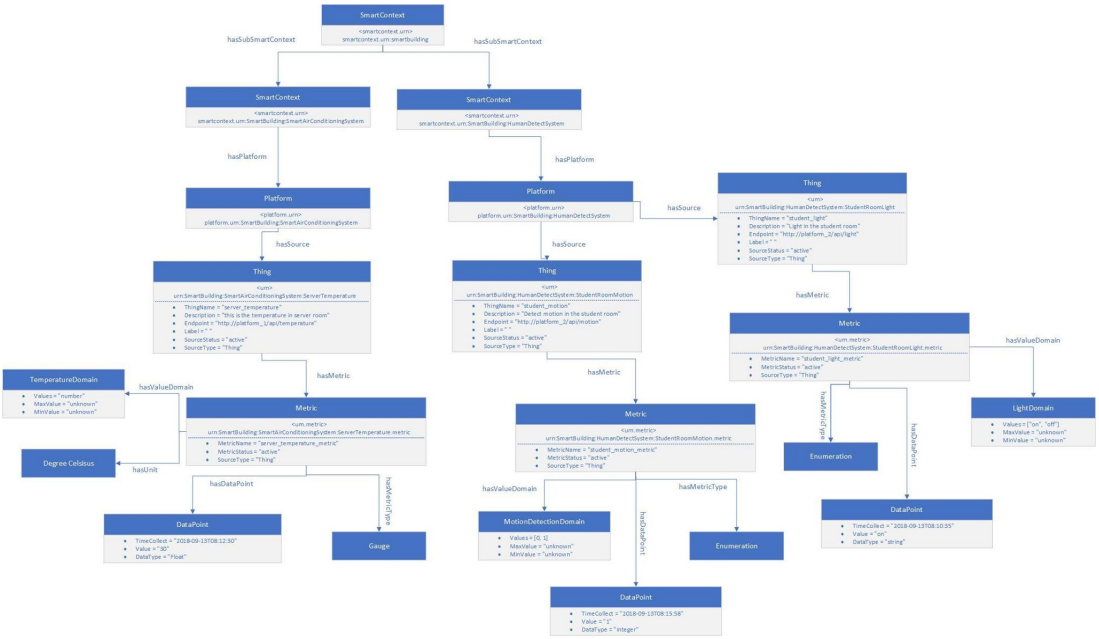

Fig. 3. Resource graph

Anotation for the resource graph:

- smartcontext.um:SmartBuilding is a `Smart Context`
  hasSubSmartContext smartcontext.um:SmartBuilding:SmartAirconditionSystem

- smartcontext.um:SmartBuilding:SmartAirconditionSystem is a `Smart Context`
  hasPlatform platform.um:SmartBuilding:SmartAirconditionSystem

- platform.um:SmartBuilding:SmartAirconditionSystem is a `Platform`
  hasSource urn:SmartBuilding:SmartAirconditionSystem:ServerTemperature

- urn:SmartBuilding:SmartAirconditionSystem:ServerTemperature is a `Thing`
  ThingName ”server_temperature”
  Description ”this is the temperature in server room”
  Endpoint ”http://platform_1/api/temperature”
  Label ””
  SourceStatus ”active”
  SourceType ”Thing”
  hasMetric urn:SmartBuilding:SmartAirconditionSystem:ServerTemperature.metric

- urn:SmartBuilding:SmartAirconditionSystem:ServerTemperature.metric is a `Metric`

MetricName ”server_temperature_metric”
MetricStatus ”active”
SourceType ”Thing”
hasValueDomain TemperatureDomain
hasUnit Degree Celsisus
hasDataPoint DataPoint
hasMetricType Gauge

4 Deploy IoT Platforms

4.1 Target

- Deploy IoT Platforms and related devices
- Testing with devices (collect data, turn on/off led, ...)

4.2 Implement

In this section, we will deploy 3 platforms: Home Assistant, OpenHAB and ThingsBoard. Each IoT Platform is deployed on different side, and combine:

- 1 temperature sensor
- 1 humidity sensor
- 1 photo sensor
- 1 motion sensor
- 3 LEDs

Deployed model on each IoT platform as figure:

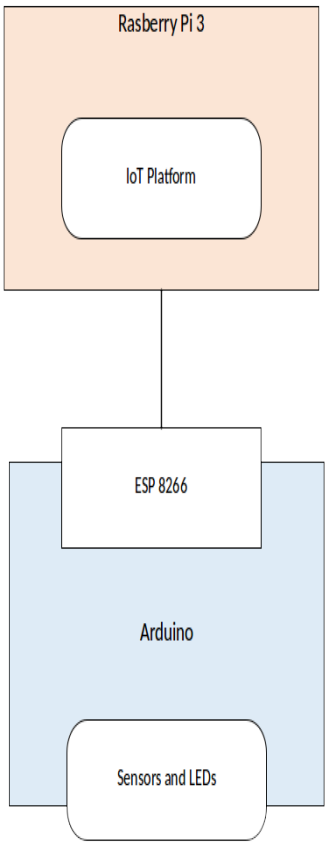

Fig. 4. Deploy model

To collect data, we connected sensors and LEDs to a Arduino. This Arduino responsibility for collect data and forward data to ESP8266. We have to use ESP8266 because Arduino is not able to connect to the network. ESP8266 forward data to one of three above IoT platforms deployed on Raspery Pi 3 in the

same LAN.

4.3 Result and Conclusion

After deployment, the result we have three IoT platforms with user interfaces below:

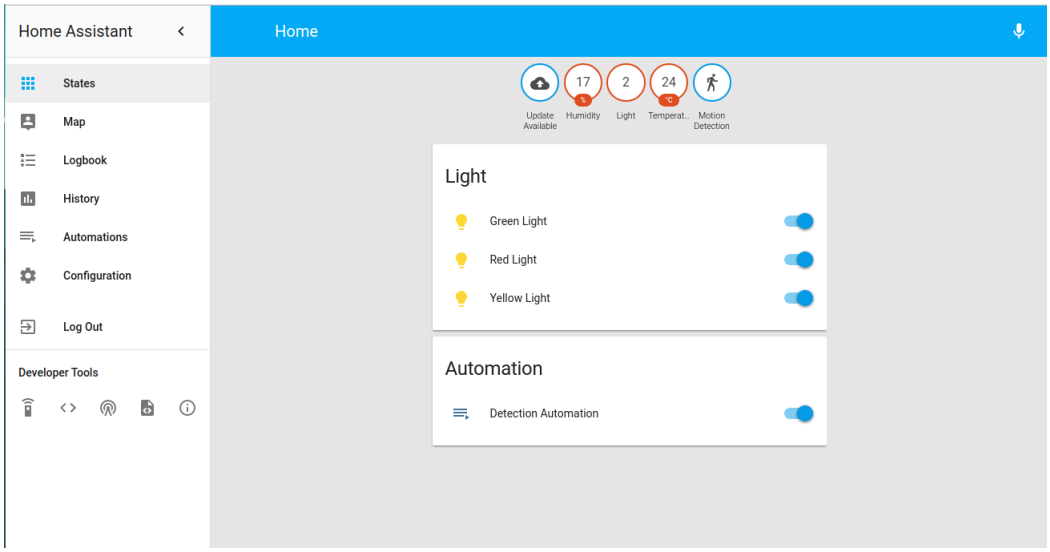

Fig. 5. User Interface of Home Assistant

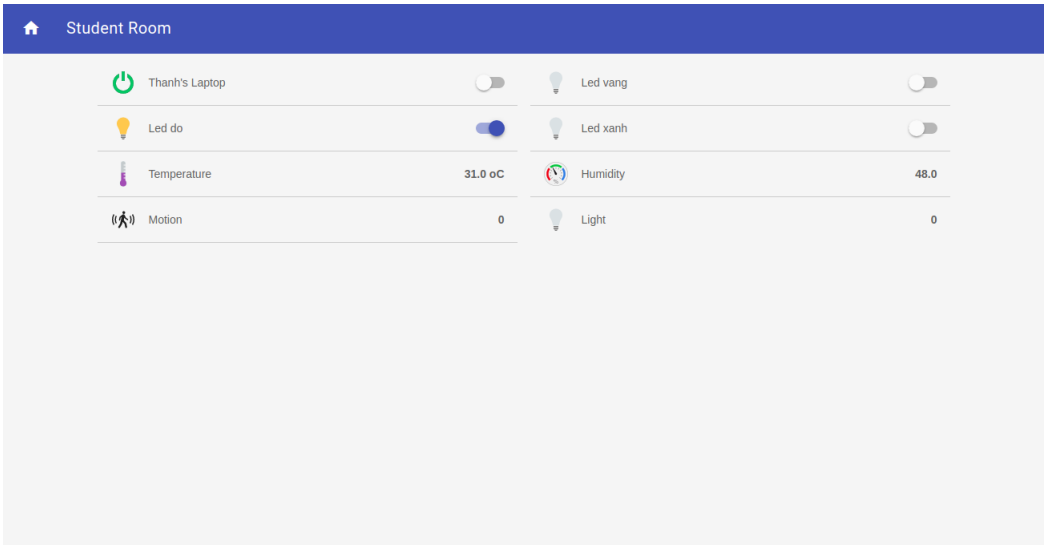

Fig. 6. User Interface of OpenHAB

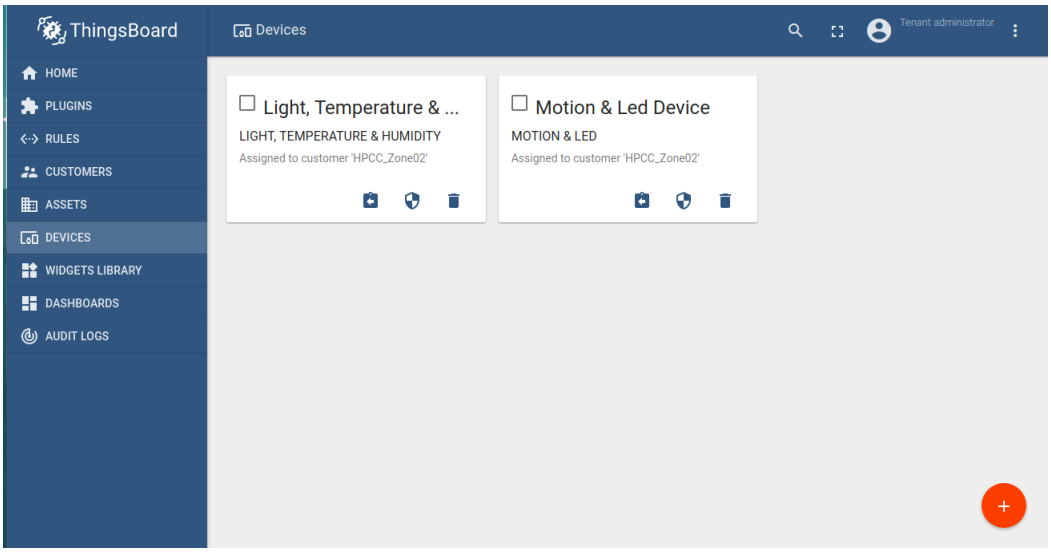

Fig. 7. User interface of Thingsboard

5 Implement a Alert application with data collected from 3 IoT platforms

5.1 Target

- Demonstrate that our cross-platform system can collect data from many IoT platforms.
- Demonstrate our released API can provide for developers to use them easily depend on demand.

5.2 Implementation

We will build a simple application. The model of this application as follow

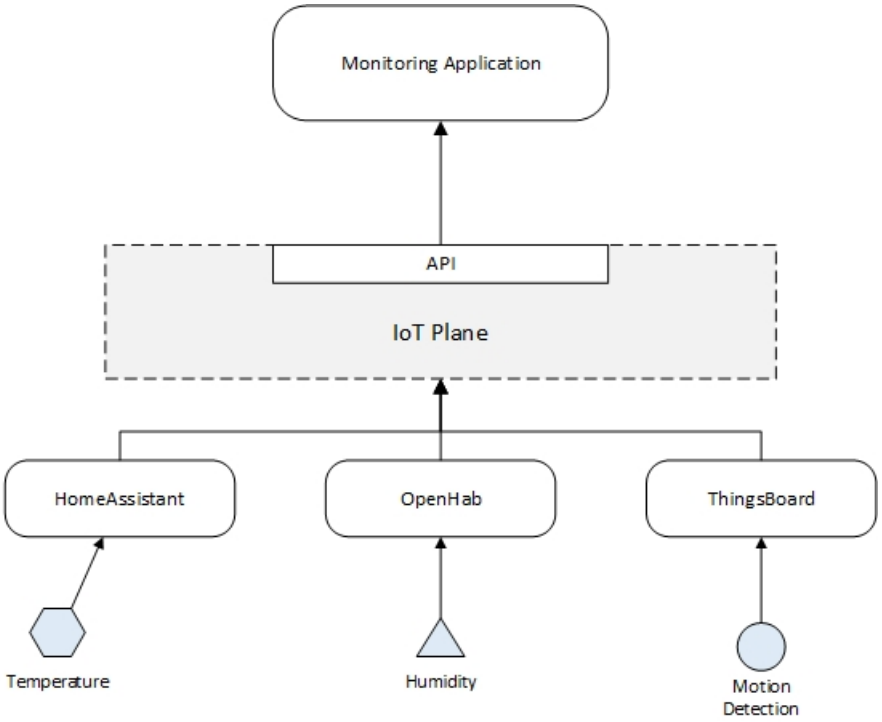

Fig. 8. Application model

The application monitor data collected from three platforms: Home Assistant, OpenHAB and ThingsBoard.

- Home Assistant: Data from temperature sensor

– OpenHAB: Data from humidity sensor
– ThingsBoard: Data from motion sensor

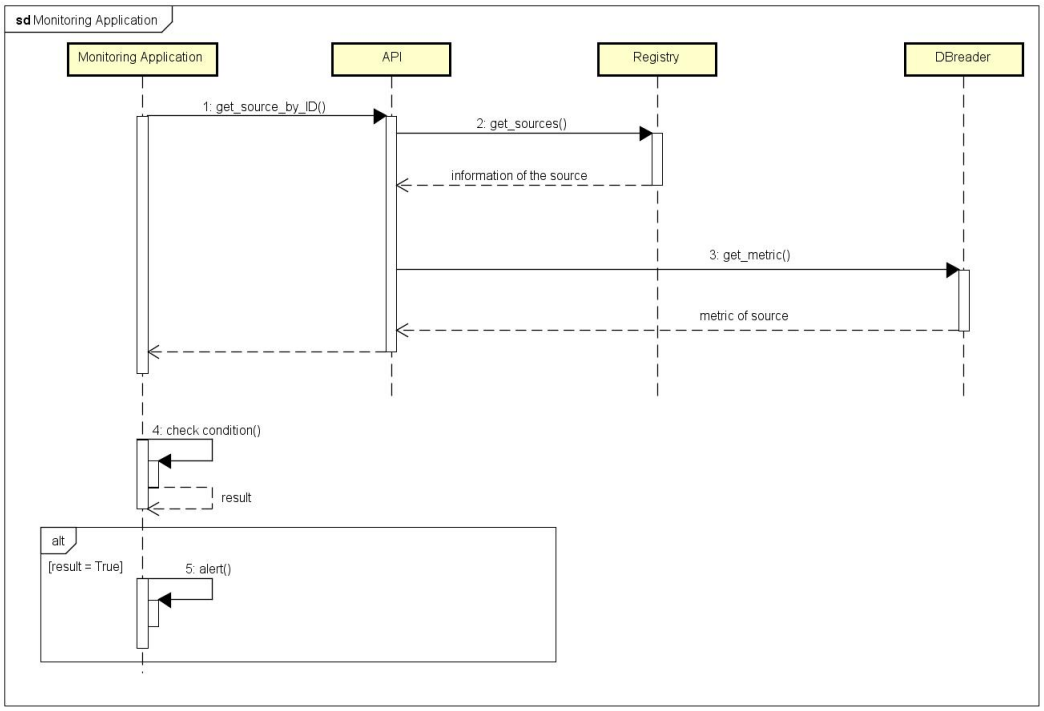

Fig. 9. Communication Diagram of monitor application

Application monitor the data and generate a alert on the interface when all following condition are satisfied:

– Temperature greater than 35 Cencius degree
– Humidity less than 50 percent
– There is a motion on the monitering region

5.3 Result

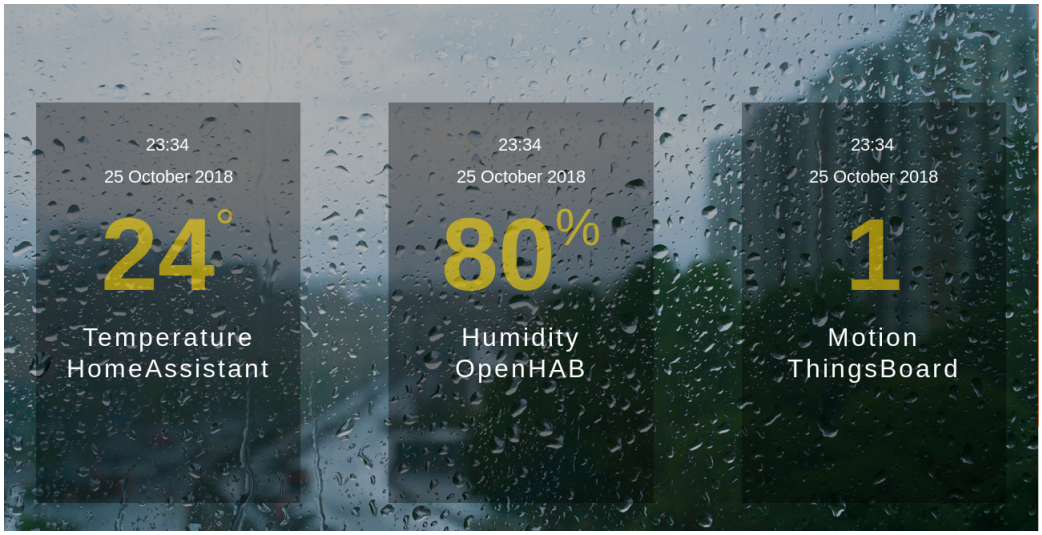

Fig. 10. Monitoring Application interface

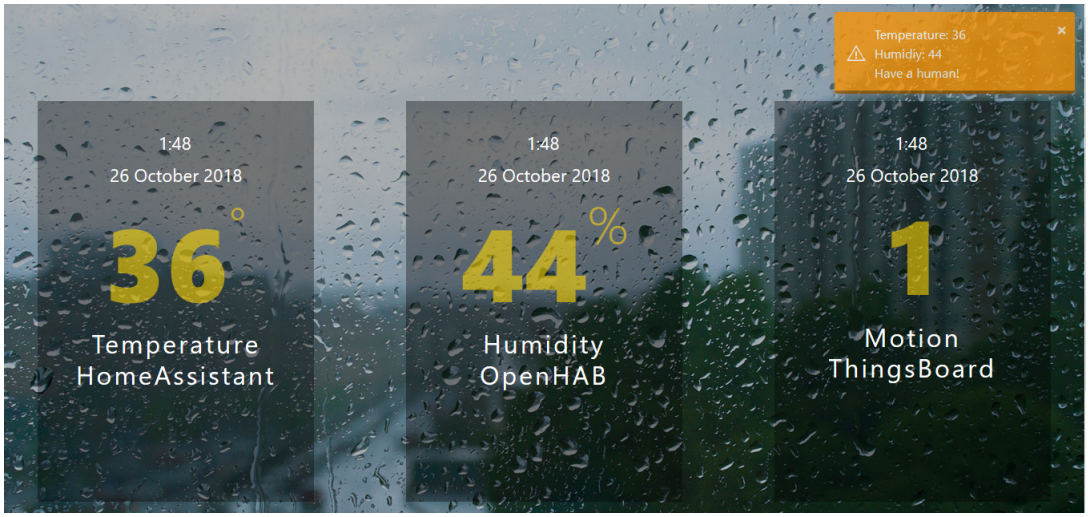

Fig. 11. Monitoring Application interface with alert

6 Experiment

6.1 Target

– Measure some operation runtime of the system to see some characteristic and conclusion

6.2 Implementation

In this section, we build a simple program to collect continous data and tuning devices status. Implementation process is describe in the below diagram.

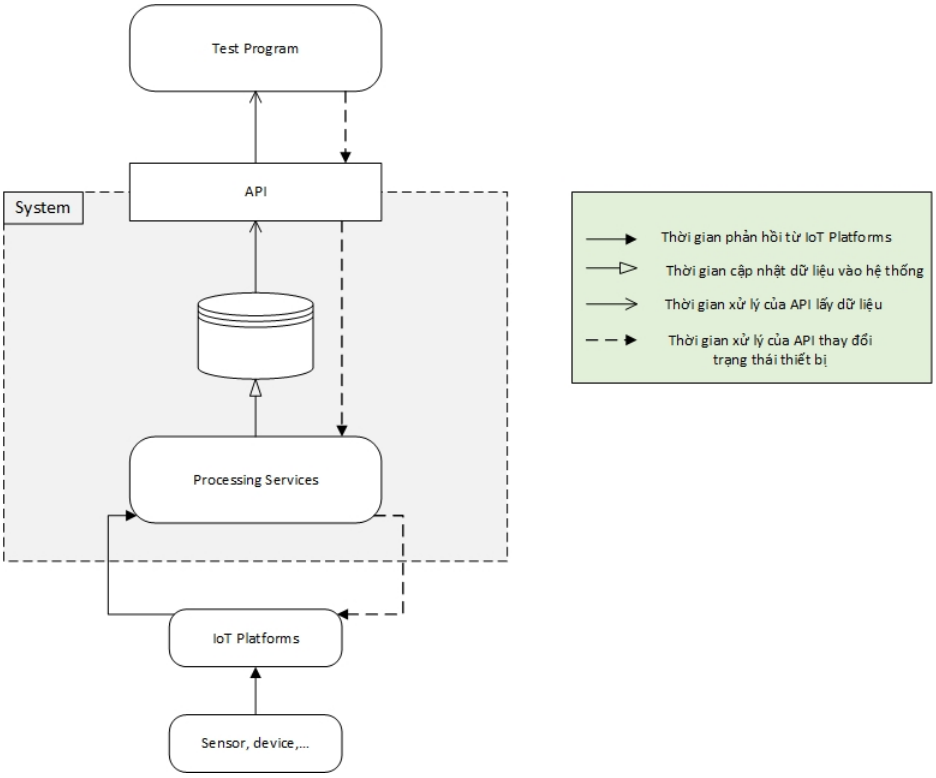

Fig. 12. Implementation process

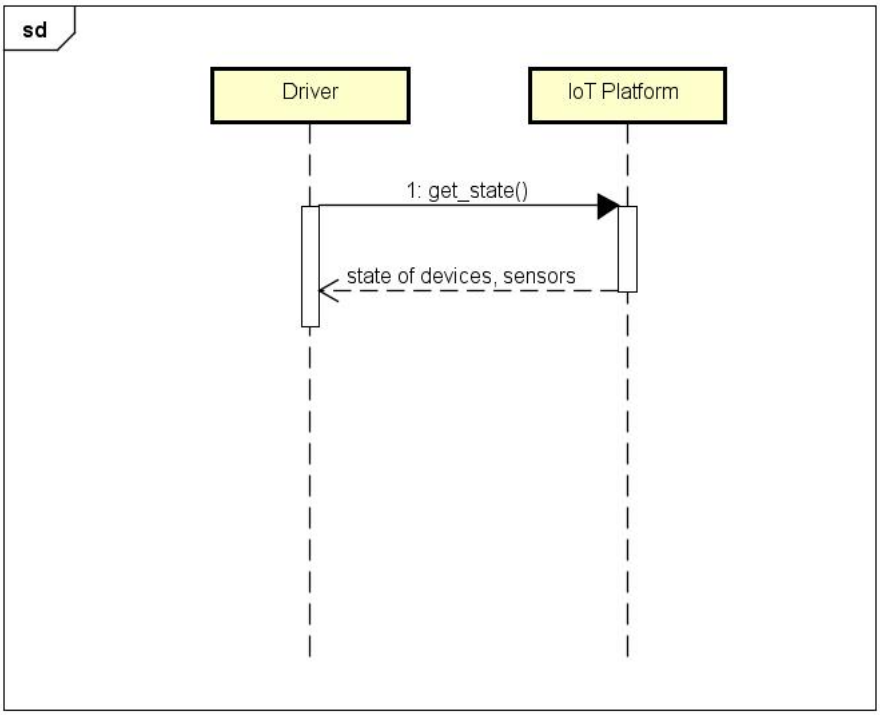

Fig. 13. Communication Diagram Collecting data from IoT platform

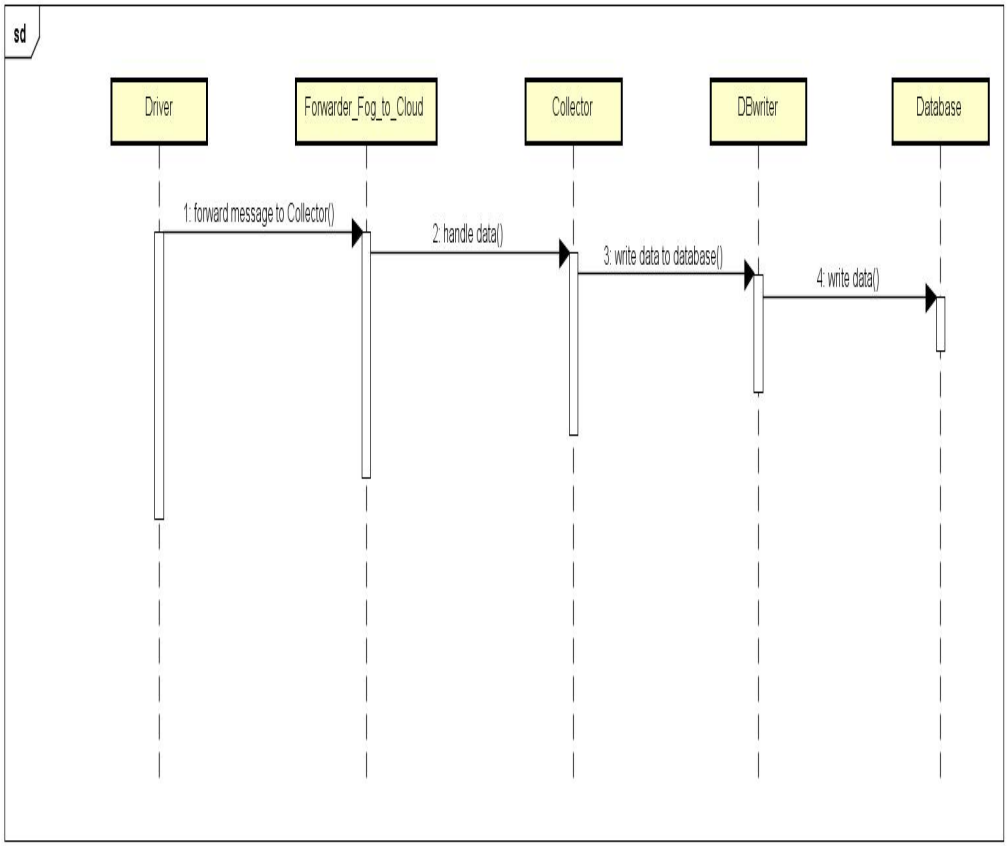

Fig. 14. Communication Diagram Updating data into our system

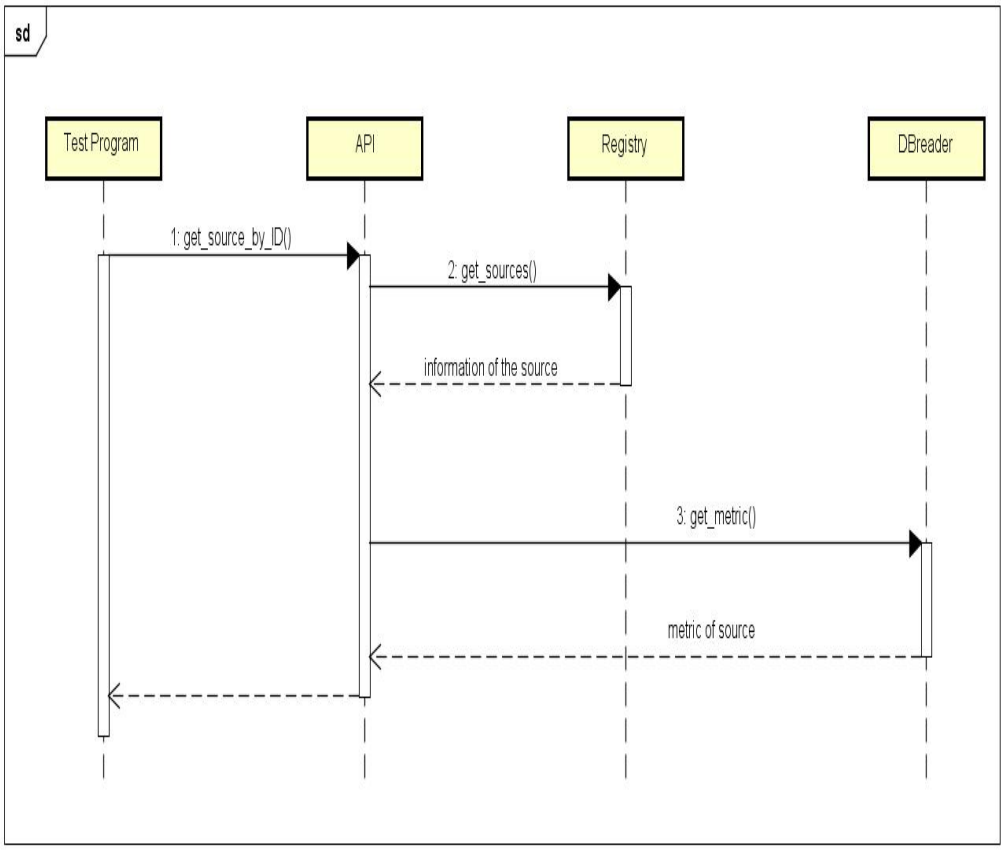

Fig. 15. Communication Diagram Getting data from a device

With above process, we tuning config variables of the system, make test cases and measure carefully to get the best configuration on different cases. We consider some cases:

- Number of sensors: 25 sensors, 50 sensors, 100 sensors
- Data frequency: 1 second, 1.5 second, 2 second (per data point)

6.3 Result and Conclusion

Runtime of basic APIs We measure runtime of three API that are most used:

- API to get list all IoT platforms in the system
- API to get list all devices on a specific platform
- API to get a specific device on the system

No	25 devices				50 devices				100 devices			
	Respond time of IoT platforms	Update time when update new data	API runtime to get new data	API runtime to set new state	Respond time of IoT platforms	Update time when update new data	API runtime to get new data	API runtime to set new state	Respond time of IoT platforms	Update time when update new data	API runtime to get new data	API runtime to set new state
1	12.9495	272.5785	68.0995	45.8252	14.0071	322.5467	71.6698	54.971	12.445	494.9176	52.8324	43.2138
2	15.7521	303.5607	68.7394	46.7145	11.0774	338.7997	77.8759	60.6768	163.5165	222.5361	195.8714	43.1836
3	8.0864	232.9221	50.8857	43.0288	11.6837	297.4751	86.998	61.6577	18.9004	256.8254	46.7565	49.1891
4	13.1102	249.2838	51.2941	59.1631	12.1987	285.1675	59.912	41.378	12.5372	301.4727	57.6973	51.6169
5	8.8987	242.8868	52.9799	61.3871	10.4926	276.6895	68.0461	64.2605	21.7309	305.9766	62.5582	53.5176
6	8.7597	288.2922	54.2297	47.5435	13.0553	267.7965	53.4663	74.924	21.5933	412.2558	76.0853	60.5211
7	13.8164	226.5344	64.9066	47.3137	19.3434	285.9678	83.8547	64.7924	20.767	369.0312	51.733	73.0541
8	16.4149	504.8232	61.9001	58.512	13.3443	261.8711	94.1601	68.356	11.6711	262.6195	49.5274	63.0205
9	17.4429	266.4938	72.1195	55.0523	17.9834	229.672	57.2648	57.941	21.5843	383.4798	62.7344	64.558
10	17.6046	220.0284	63.0901	39.1109	17.4994	291.7683	59.2535	59.7236	23.3366	306.5414	60.971	74.6543
11	8.316	274.6668	58.0735	58.4781	20.3319	253.6783	59.5737	53.9844	16.0923	503.2454	67.5855	64.5404
12	15.909	267.3852	64.5027	56.1447	17.8709	261.9653	81.8787	55.4159	19.8762	487.4432	70.5223	56.0782
13	16.0992	271.3265	68.2704	49.1221	14.838	306.55	56.0749	166.4906	21.8019	331.5058	73.9641	621.659
14	16.083	282.4435	78.4075	59.9267	9.0382	229.408	61.1126	62.3109	23.5212	435.1094	87.4031	81.4793

15	13.5362	265.5084	71.173	58.2013	21.8852	243.5658	52.1383	134.0892	17.1323	367.4781	68.9464	549.673
16	13.8974	266.0253	88.2692	62.6056	11.3292	262.7025	82.9682	49.5572	22.506	432.9209	89.0498	436.575
17	14.6551	260.0389	44.1403	60.2601	9.8445	279.9857	119.2253	57.5616	16.4766	284.6966	86.297	335.156
18	12.0203	246.7301	82.9332	58.7316	12.64	313.6675	54.6753	67.3866	20.8447	401.1455	62.685	238.695
19	14.555	234.5345	75.5293	56.3803	11.0505	267.446	96.8657	65.5103	16.0408	301.7404	66.4399	230.689
20	13.9	276.5288	76.1163	95.8352	14.5385	314.0585	102.8583	107.5902	19.4318	280.4732	69.7489	104.223
21	15.9519	314.1134	71.4946	52.8581	11.0416	260.5901	65.0327	50.7357	13.4606	441.3731	74.9295	57.7166
22	13.7691	226.0189	105.074	89.1144	16.6001	452.0776	75.9416	53.0083	20.1092	627.6498	47.3378	67.4469
23	14.1046	212.0988	106.415	55.9175	98.0523	278.7008	63.0219	55.8996	285.6236	215.867	57.7343	44.349
24	8.7383	261.1823	67.6055	58.9375	22.1016	249.5222	58.3479	55.9022	71.0304	186.1372	56.6509	64.26
25	18.7459	343.0917	63.545	53.8311	17.7228	284.8499	47.0467	39.4063	13.8273	307.2903	78.1984	54.395
26	8.4848	220.7448	70.9445	64.4619	9.5081	287.3755	50.2198	47.7703	16.3989	342.7775	70.3826	50.9217
27	15.0998	236.9571	62.8271	64.2118	9.2559	280.0164	60.0636	51.7306	27.9143	423.3761	69.1764	52.2447
28	15.7197	232.5397	84.4314	55.7618	17.5219	255.9087	75.3176	53.5691	21.071	394.2726	59.6025	55.4242
29	13.6027	254.1993	65.3136	72.4423	15.3568	278.9919	66.7934	55.4354	18.5654	287.6101	70.9782	49.6492
30	42.1433	173.0444	79.3202	63.3566	14.9865	316.2787	63.4091	40.8487	21.3771	303.9484	67.5666	56.3352
31	10.0501	232.5082	57.5581	52.0244	12.0327	425.7097	64.6138	43.3774	17.3907	413.8861	69.211	52.3815
32	18.9238	245.9404	70.8489	64.5747	12.3281	299.3531	73.3306	45.0616	17.2646	268.8637	56.6821	51.3749
33	8.4302	199.3306	61.3751	62.4714	14.8773	243.0813	64.368	54.4858	21.8642	331.984	70.2963	37.4646

34	15.81	292.6273	47.5736	60.7238	18.7402	283.7758	65.7454	46.8864	15.8522	263.2179	59.5245	53.8671
35	13.8314	219.0478	84.4245	52.6583	10.7822	276.1941	65.0604	62.094	18.5318	406.6176	64.6231	52.3217
36	17.9501	415.1833	75.1889	53.4217	16.9165	241.261	75.7754	74.7666	26.7601	285.615	59.7532	54.024
37	17.087	166.9023	70.3771	48.5766	9.2309	286.1555	58.7749	52.7966	14.3373	322.6082	69.2356	445.017
38	12.4516	162.5783	76.1738	56.3219	10.8216	272.7263	71.2397	52.9208	20.7632	407.5267	57.9412	319.631
39	16.3698	152.7624	76.565	56.4313	12.1531	258.7459	74.5304	71.8992	17.4251	330.8523	53.9334	322.006
40	10.0605	114.9087	70.8723	51.3005	11.9779	253.1707	75.3582	56.767	11.2898	382.7834	67.7922	213.323
41	8.1878	238.9717	137.439	52.47	10.9911	250.8144	76.8855	61.6806	22.2073	318.0137	74.9748	78.4786
42	14.8716	153.7399	62.9771	56.4311	15.2419	282.9285	62.6035	62.429	24.8764	286.9506	70.199	67.8496
43	12.4223	146.9612	57.8294	57.411	13.8028	267.7481	70.8876	54.4441	18.4221	339.5951	76.1557	53.9219
44	17.0629	171.0501	70.2879	54.4329	15.2216	288.4653	69.1628	140.1377	20.7748	324.2829	60.5009	47.7154
45	13.2139	186.8713	56.0079	54.8248	16.4127	245.1336	96.2398	70.6551	22.1896	568.1555	197.0704	63.6618
46	14.5938	186.224	59.51	51.4507	16.7303	325.7911	77.8358	47.9445	265.3534	75.654	70.0743	52.6888
47	19.7179	118.8312	64.744	39.7282	10.8204	308.0156	110.2839	55.0528	16.8598	303.8709	51.4016	44.7917
48	11.2453	168.7772	129.018	54.2288	19.2561	237.654	76.9577	76.7555	11.7121	334.0702	61.7111	61.2922
49	9.5487	133.7755	72.4587	128.423	15.9111	333.9493	68.2747	51.2486	11.8725	326.7062	61.7223	40.6773
50	8.0225	210.3832	67.8544	38.2812	17.7479	244.3202	90.4992	56.3803	25.9342	386.1597	47.5833	66.874

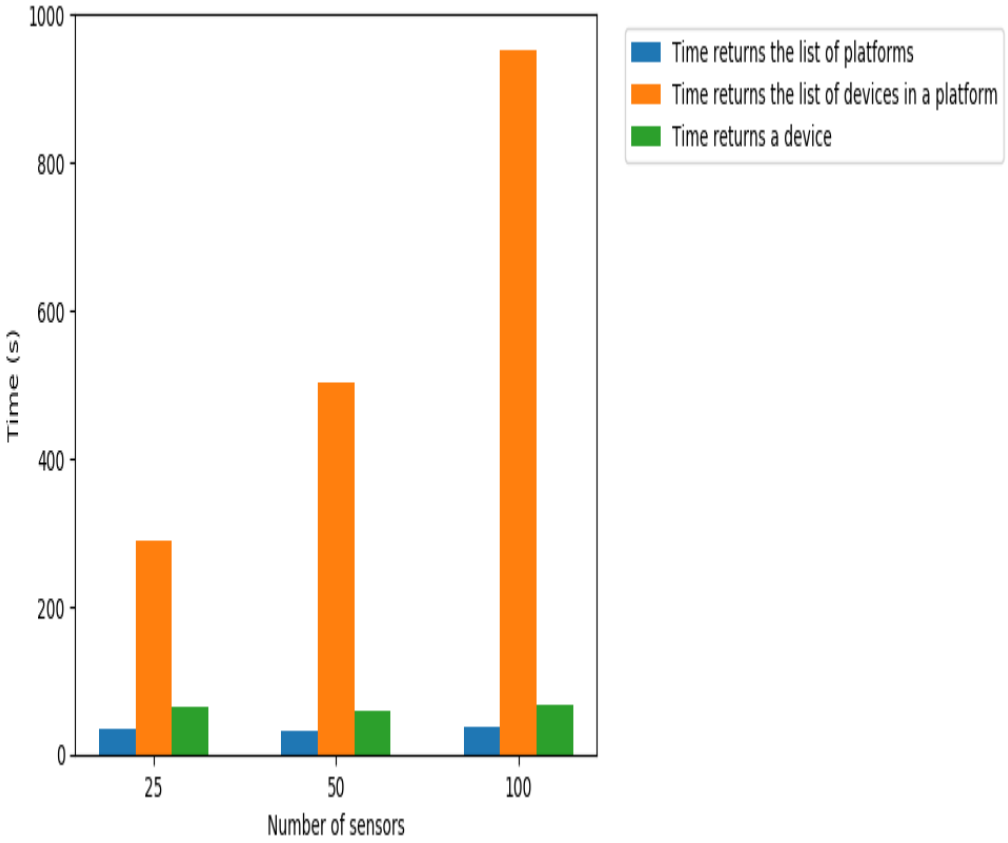

Fig. 19. Average result when change number of device

We can conclude:

- Time to get list all IoT platforms is unchange when changing number of device.
- Time to get list all devices of a specific platform will increase when number of devices increase and vice versa. This is understanable because of:
  - Work with database: Increase number of device will increase respond time and query time in database.
  - Tranfer time: Larger amount of data will cost more time to tranfer
- Even time to get list all devices increase when number of device increase but with 100 device, the API respond time is less than 1 second. We think this respond time is acceptable.
- Time to get a specific device on the system is unchange when changing number of device

Runtime of some tasks In this experiment, we test with different number of device, data frequency to measure runtime of some tasks:

- Respond time of IoT platforms when system query data
- Update time when update new data into system
- API runtime to get new data of a device
- API runtime to set new state for a device

No	25 devices				50 devices				100 devices			
	Respond time of IoT platforms	Update time when update new data	API runtime to get new data	API runtime to set new state	Respond time of IoT platforms	Update time when update new data	API runtime to get new data	API runtime to set new state	Respond time of IoT platforms	Update time when update new data	API runtime to get new data	API runtime to set new state
1	12.9495	272.5785	68.0995	45.8252	14.0071	322.5467	71.6698	54.971	12.445	494.9176	52.8324	43.2138
2	15.7521	303.5607	68.7394	46.7145	11.0774	338.7997	77.8759	60.6768	163.5165	222.5361	195.8714	43.1836
3	8.0864	232.9221	50.8857	43.0288	11.6837	297.4751	86.998	61.6577	18.9004	256.8254	46.7565	49.1891
4	13.1102	249.2838	51.2941	59.1631	12.1987	285.1675	59.912	41.378	12.5372	301.4727	57.6973	51.6169
5	8.8987	242.8868	52.9799	61.3871	10.4926	276.6895	68.0461	64.2605	21.7309	305.9766	62.5582	53.5176
6	8.7597	288.2922	54.2297	47.5435	13.0553	267.7965	53.4663	74.924	21.5933	412.2558	76.0853	60.5211
7	13.8164	226.5344	64.9066	47.3137	19.3434	285.9678	83.8547	64.7924	20.767	369.0312	51.733	73.0541
8	16.4149	504.8232	61.9001	58.512	13.3443	261.8711	94.1601	68.356	11.6711	262.6195	49.5274	63.0205
9	17.4429	266.4938	72.1195	55.0523	17.9834	229.672	57.2648	57.941	21.5843	383.4798	62.7344	64.558
10	17.6046	220.0284	63.0901	39.1109	17.4994	291.7683	59.2535	59.7236	23.3366	306.5414	60.971	74.6543
11	8.316	274.6668	58.0735	58.4781	20.3319	253.6783	59.5737	53.9844	16.0923	503.2454	67.5855	64.5404
12	15.909	267.3852	64.5027	56.1447	17.8709	261.9653	81.8787	55.4159	19.8762	487.4432	70.5223	56.0782
13	16.0992	271.3265	68.2704	49.1221	14.838	306.55	56.0749	166.4906	21.8019	331.5058	73.9641	621.659
14	16.083	282.4435	78.4075	59.9267	9.0382	229.408	61.1126	62.3109	23.5212	435.1094	87.4031	81.4793

15	13.5362	265.5084	71.173	58.2013	21.8852	243.5658	52.1383	134.0892	17.1323	367.4781	68.9464	549.673
16	13.8974	266.0253	88.2692	62.6056	11.3292	262.7025	82.9682	49.5572	22.506	432.9209	89.0498	436.575
17	14.6551	260.0389	44.1403	60.2601	9.8445	279.9857	119.2253	57.5616	16.4766	284.6966	86.297	335.156
18	12.0203	246.7301	82.9332	58.7316	12.64	313.6675	54.6753	67.3866	20.8447	401.1455	62.685	238.695
19	14.555	234.5345	75.5293	56.3803	11.0505	267.446	96.8657	65.5103	16.0408	301.7404	66.4399	230.689
20	13.9	276.5288	76.1163	95.8352	14.5385	314.0585	102.8583	107.5902	19.4318	280.4732	69.7489	104.223
21	15.9519	314.1134	71.4946	52.8581	11.0416	260.5901	65.0327	50.7357	13.4606	441.3731	74.9295	57.7166
22	13.7691	226.0189	105.074	89.1144	16.6001	452.0776	75.9416	53.0083	20.1092	627.6498	47.3378	67.4469
23	14.1046	212.0988	106.415	55.9175	98.0523	278.7008	63.0219	55.8996	285.6236	215.867	57.7343	44.349
24	8.7383	261.1823	67.6055	58.9375	22.1016	249.5222	58.3479	55.9022	71.0304	186.1372	56.6509	64.26
25	18.7459	343.0917	63.545	53.8311	17.7228	284.8499	47.0467	39.4063	13.8273	307.2903	78.1984	54.395
26	8.4848	220.7448	70.9445	64.4619	9.5081	287.3755	50.2198	47.7703	16.3989	342.7775	70.3826	50.9217
27	15.0998	236.9571	62.8271	64.2118	9.2559	280.0164	60.0636	51.7306	27.9143	423.3761	69.1764	52.2447
28	15.7197	232.5397	84.4314	55.7618	17.5219	255.9087	75.3176	53.5691	21.071	394.2726	59.6025	55.4242
29	13.6027	254.1993	65.3136	72.4423	15.3568	278.9919	66.7934	55.4354	18.5654	287.6101	70.9782	49.6492
30	42.1433	173.0444	79.3202	63.3566	14.9865	316.2787	63.4091	40.8487	21.3771	303.9484	67.5666	56.3352
31	10.0501	232.5082	57.5581	52.0244	12.0327	425.7097	64.6138	43.3774	17.3907	413.8861	69.211	52.3815
32	18.9238	245.9404	70.8489	64.5747	12.3281	299.3531	73.3306	45.0616	17.2646	268.8637	56.6821	51.3749
33	8.4302	199.3306	61.3751	62.4714	14.8773	243.0813	64.368	54.4858	21.8642	331.984	70.2963	37.4646

34	15.81	292.6273	47.5736	60.7238	18.7402	283.7758	65.7454	46.8864	15.8522	263.2179	59.5245	53.8671
35	13.8314	219.0478	84.4245	52.6583	10.7822	276.1941	65.0604	62.094	18.5318	406.6176	64.6231	52.3217
36	17.9501	415.1833	75.1889	53.4217	16.9165	241.261	75.7754	74.7666	26.7601	285.615	59.7532	54.024
37	17.087	166.9023	70.3771	48.5766	9.2309	286.1555	58.7749	52.7966	14.3373	322.6082	69.2356	445.017
38	12.4516	162.5783	76.1738	56.3219	10.8216	272.7263	71.2397	52.9208	20.7632	407.5267	57.9412	319.631
39	16.3698	152.7624	76.565	56.4313	12.1531	258.7459	74.5304	71.8992	17.4251	330.8523	53.9334	322.006
40	10.0605	114.9087	70.8723	51.3005	11.9779	253.1707	75.3582	56.767	11.2898	382.7834	67.7922	213.323
41	8.1878	238.9717	137.439	52.47	10.9911	250.8144	76.8855	61.6806	22.2073	318.0137	74.9748	78.4786
42	14.8716	153.7399	62.9771	56.4311	15.2419	282.9285	62.6035	62.429	24.8764	286.9506	70.199	67.8496
43	12.4223	146.9612	57.8294	57.411	13.8028	267.7481	70.8876	54.4441	18.4221	339.5951	76.1557	53.9219
44	17.0629	171.0501	70.2879	54.4329	15.2216	288.4653	69.1628	140.1377	20.7748	324.2829	60.5009	47.7154
45	13.2139	186.8713	56.0079	54.8248	16.4127	245.1336	96.2398	70.6551	22.1896	568.1555	197.0704	63.6618
46	14.5938	186.224	59.51	51.4507	16.7303	325.7911	77.8358	47.9445	265.3534	75.654	70.0743	52.6888
47	19.7179	118.8312	64.744	39.7282	10.8204	308.0156	110.2839	55.0528	16.8598	303.8709	51.4016	44.7917
48	11.2453	168.7772	129.018	54.2288	19.2561	237.654	76.9577	76.7555	11.7121	334.0702	61.7111	61.2922
49	9.5487	133.7755	72.4587	128.423	15.9111	333.9493	68.2747	51.2486	11.8725	326.7062	61.7223	40.6773
50	8.0225	210.3832	67.8544	38.2812	17.7479	244.3202	90.4992	56.3803	25.9342	386.1597	47.5833	66.874

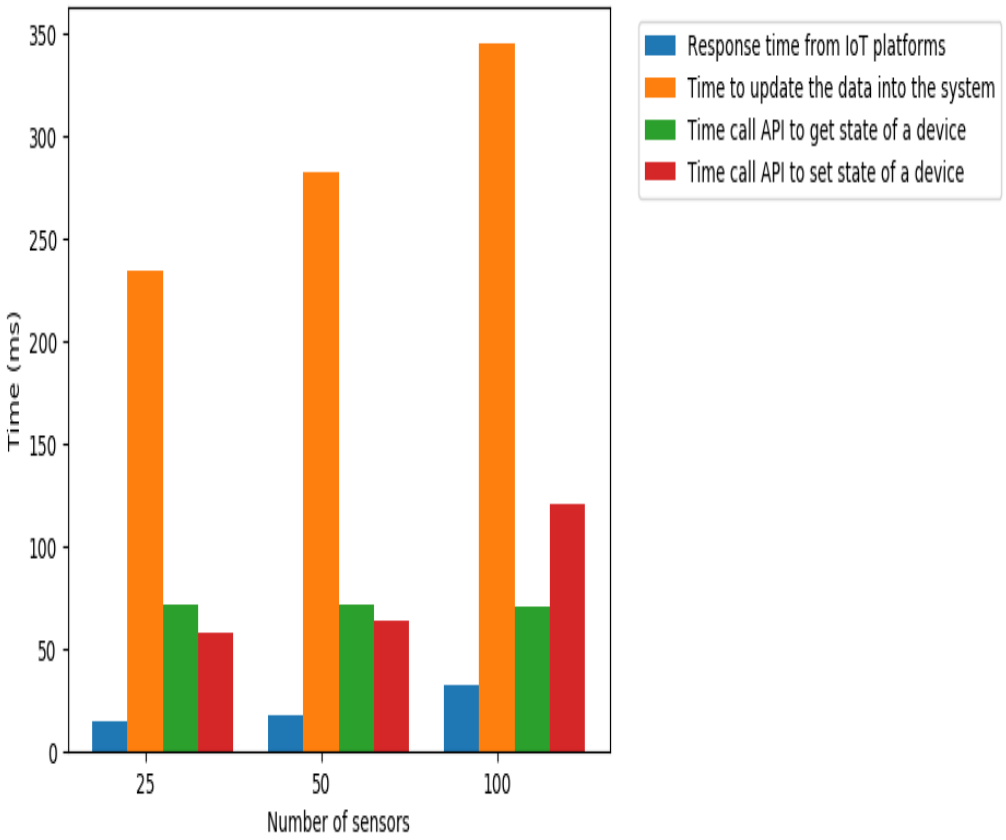

Fig. 23. Average result when change number of device

We have conclusion:

- Respond time of IoT platforms change when change number of device. This is understandable because managing large amount of device need more time to process.
- Update time when update new data into system increase linearly with number of device. The reason is more data also mean need more time to tranfer and more read/write operation on database.
- API runtime to get new data of a device is stable when number of device changing. This mean the system is stable with end user when the system get larger
- API runtime to set new state for a device increase when increase the number of device. It's not mean the system process inefficiently but because of above reason: IoT platforms respond and process slowly when number of device get large.

No	25 devices			50 devices			100 devices		
	API get list all IoT platforms	API get list all devices	API get a specific device	API get list all IoT platforms	API get list all devices	API get a specific device	API get list all IoT platforms	API get list all devices	API get a specific device
1	30.4329	285.007	60.7548	31.9331	467.8745	46.14	79.5984	944.2301	52.4304
2	35.9294	288.1842	62.5358	32.855	425.3552	62.439	64.3063	850.1198	79.7865
3	31.6467	273.7958	63.041	30.0241	447.763	61.3894	35.1458	975.4298	70.8101
4	31.8868	292.9807	65.0647	31.3218	482.0771	65.0477	30.5572	1019.274	66.9453
5	29.953	261.6179	71.3398	38.7688	521.0857	65.388	32.4776	905.3512	73.1235
6	62.1009	274.4586	68.4295	39.8948	584.379	75.1829	38.1222	940.5143	71.7719
7	40.0343	263.2308	62.5448	35.8	489.0468	65.7997	39.3395	964.6208	64.5516
8	33.0546	269.7675	72.2289	34.8659	484.6849	49.3979	44.404	944.2365	63.9215
9	32.9096	300.3702	66.2851	36.3169	498.3678	52.8541	30.3841	965.5132	58.9776
10	36.2263	313.225	58.7776	25.1801	508.6937	48.1708	37.7173	906.9228	72.9184
11	34.1029	303.2818	56.8769	37.8938	515.6567	62.8889	37.7195	993.7363	53.838
12	31.5173	292.8934	55.2692	24.7169	492.5022	52.3109	48.1737	935.1437	68.1281
13	39.1314	291.3651	70.54	43.0031	476.8527	67.3609	41.3733	979.9695	76.0398
14	31.5828	279.7878	47.2462	36.6759	532.9258	61.9261	37.0708	839.6192	80.6878
15	37.0398	286.5522	73.4239	43.2396	466.747	51.6195	47.6365	923.2075	66.1232
16	38.8887	290.627	49.5746	40.2095	541.8277	45.5637	27.4582	983.5496	76.1974
17	35.3167	290.4041	61.949	36.7558	504.5815	53.5569	35.2383	895.9703	65.2013
18	35.2235	274.0767	59.2499	34.3845	549.686	63.3066	30.3695	971.1912	60.837

19	31.4379	297.59	66.5996	32.6648	513.6013	55.4628	41.2164	925.8404	67.9698
20	32.4068	283.5143	70.0927	28.1558	523.689	72.427	27.1354	1052.776	49.7637
21	44.3547	272.1288	67.1737	29.8197	491.5791	41.234	31.9779	1038.3	60.5271
22	22.5158	302.6214	66.0734	22.6967	433.4695	43.2804	32.9468	1002.034	59.5379
23	26.3021	295.7757	58.5251	31.0757	521.8658	71.3019	37.1592	969.1172	59.1371
24	28.6162	312.1769	63.9193	32.5429	436.6393	44.3509	39.1929	987.4387	50.2853
25	32.6796	294.5564	81.9092	25.3806	524.7893	43.855	29.3803	988.5714	54.9476
26	34.8272	303.1006	60.3936	29.9067	488.091	51.5616	38.8091	956.1133	65.1414
27	40.3066	268.491	58.4538	21.5878	491.3595	61.8865	22.8837	905.2989	89.0155
28	36.2651	343.5347	65.2273	33.5164	510.0837	54.6529	32.1574	939.5525	51.1107
29	37.4265	301.6253	52.9301	33.6447	493.7456	56.1774	23.9234	914.9692	51.5661
30	27.1084	308.9199	68.1233	30.4952	484.4244	77.3966	56.4992	986.2766	60.9617
31	40.2801	264.0271	64.0726	35.1365	460.1097	64.3289	37.7381	963.5048	77.585
32	32.2015	264.7214	68.9311	23.0634	449.0371	44.4186	45.0041	1023.683	71.923
33	42.2151	308.9919	70.5438	29.8598	495.6357	51.0521	23.7489	929.3551	75.8474
34	23.5636	291.749	62.1047	43.2682	480.8345	50.7083	31.0578	880.2531	57.1918
35	31.5282	338.5448	92.8698	33.4921	484.5557	50.2093	33.4074	947.567	74.3351
36	52.1812	284.5058	51.2776	32.0432	509.3586	64.1744	48.4202	976.1102	64.023
37	26.9554	285.3544	52.3255	31.0369	451.4725	84.9607	38.2047	926.4517	74.2016
38	32.2385	259.3217	68.929	48.6219	529.2118	61.6372	29.1839	968.3564	67.2815
39	40.2577	293.2248	69.0415	23.0587	524.5631	52.5482	35.0869	964.7219	61.7347
40	23.6092	255.7461	64.2602	31.9285	528.4398	71.9683	41.1322	978.4544	82.2153

41	29.9258	381.4759	66.293	31.4405	483.114	53.6091	35.9795	943.969	71.7511
42	40.5993	272.0168	69.1702	33.8802	489.9414	72.0022	41.4555	950.7575	58.4052
43	36.2172	263.7877	70.5473	25.229	496.8894	65.5382	40.4942	915.6718	63.8072
44	34.5802	291.9841	62.3362	31.8291	544.8158	47.6918	36.3891	1004.147	67.0021
45	33.7398	280.8561	48.1634	31.3797	465.4479	45.989	42.2125	941.3402	66.9897
46	44.2917	263.5348	46.5562	24.2805	869.6301	64.3053	46.9902	947.1531	62.3999
47	32.6974	305.3205	63.5095	34.348	504.7445	54.6026	38.8896	1007.003	61.6007
48	42.047	259.7213	71.5947	39.5241	490.5767	65.5127	29.2275	991.8244	65.2568
49	35.9592	294.5716	54.7664	37.3948	476.1569	56.5276	22.2809	936.492	63.5195
50	31.743	277.7882	60.3676	28.6138	500.0997	49.6941	37.2574	843.7121	60.4234

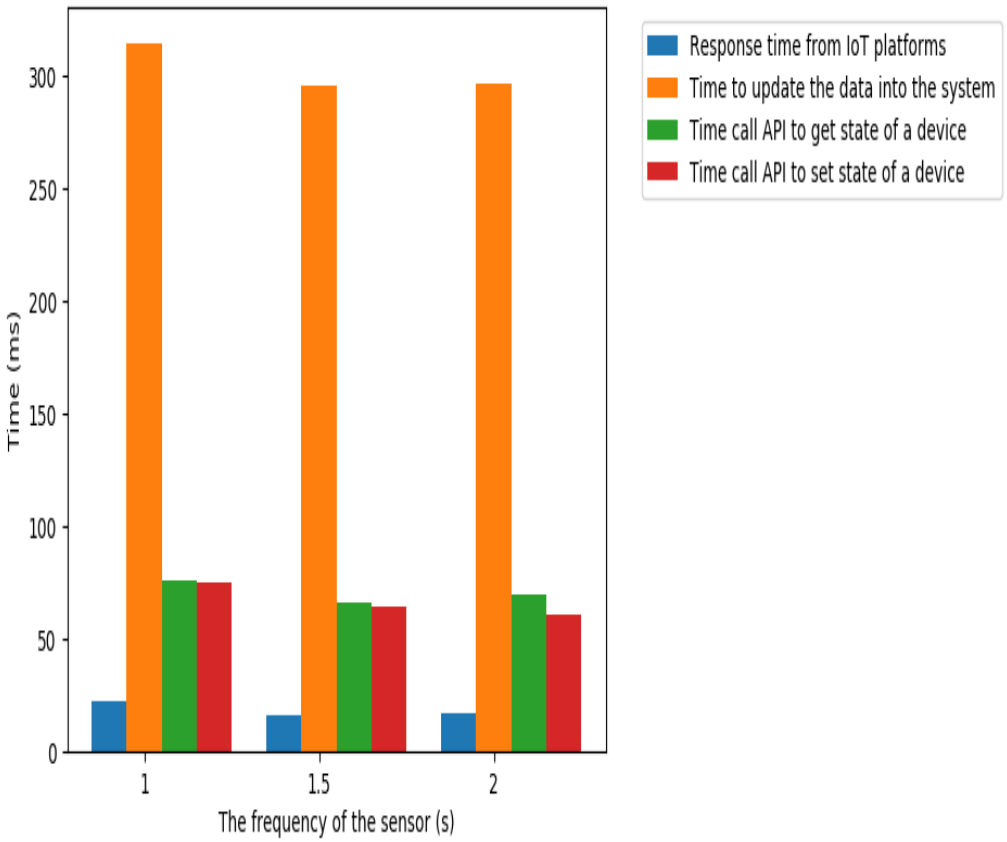

Fig. 27. Average result when change frequency send data

We have conclusion:

- Respond time of IoT platforms when system query data is not changing too much if change frequent sending data and the number of device unchange.
- Update time when update new data into system is not changing too much if change frequent sending data and the number of device unchange.
- API runtime to get new data of a device is stable even increase number of device and frequent sending data.
- API runtime to set new state for a device will increase if increase frequent sending data. The reason is platforms need to update a lot of data when increase frequent sending data and respond slowly.

7 Conclusion

With the results introduced above, we haved demonstrated that:

- Our Cross-platform can collect, monitoring data from different IoT platforms. We just present a simple monitoring application here but in the large scale, we can scale up the system and do complex tasks on different IoT platforms, which is impossible if IoT platforms work independently.
- Our APIs provide for developer to build applications on many domain and cross-domain easily.

8 The References Section

References

1. https://www.kaaproject.org/what-is-iot/

2. https://www.cmo.com/features/articles/2015/4/13/mind-blowing-stats-internet-of-things-iot.html#gs.NxjVM9k1

3. https://iot-analytics.com/iot-platforms-company-list-2017-update/
4. "Internet of Things - Architecture IOT-A: Internet of ThingsArchitecture." http://www.iot-a.eu/public.[Accessed: 28-May-2013]
5. "IEEE-SA - Internet of Things Related Standards." http://standards.ieee.org/innovate/iot/stds.html.
6. A. P. Castellani, M. Gheda, N. Bui, M. Rossi, and M. Zorzi, WebServices for the Internet of Things through CoAP and EXI, in 2011IEEE International Conference on Communications Workshops (ICC),2011, pp. 16
7. IEEE-SA - Internet of Things Related Standards. [Online]. http://standards.ieee.org/innovate/iot/stds.
8. https://www.home-assistant.io/
9. https://www.openhab.org/
10. https://thingsboard.io/