

An Open, Flexible and Multilevel Data Storing and Processing Platform for Very Large Scale Sensor Network

Jing LIU*, Jing CHEN*, Li PENG**, Xixin CAO*, RenChun Lian***, Ping WANG*

* School of Software and Microelectronics, Peking University, China

** School of IoT Engineering, Jiangnan University, China

***Inspeed Communication Co.,Ltd, China

liujing@ss.pku.edu.cn , pwang@ss.pku.edu.cn

Abstract—All kinds of sensor network have been deployed all over the world. Combining some of these networks, we can get a larger sensor network. The data of very large scale sensor network are polymorphous, heterogeneous, large in quantity and time-limited. In its application, how to store and process these data has become a key technology.

The major contribution of this paper is proposing a data Processing model based on Cloud computing for very large scale sensor network. In this flexible model, the huge sensor data and node information are stored in Multilevel Data storage including local data bases of different devices and distributed data bases of cloud; Different kinds of computations are decomposed and distributed on different nodes mainly considering their differences on computation ability and power supply. The clouds make the development and deployment of applications base on these very large scale data very easy.

We have setup a platform to verify this model base on some open source software. This platform composes of sensor nodes, WIFI and Zigbee networks, embedded gateways, local servers, and clouds. Different data bases including SQLite, TinyDB, MySQL, MongoDB and Cassandra have been used for different type of devices to store data. We also have developed some cloud applications for it. The result shows that this platform base on the new computation model is flexible, easy to develop new application and energy balanced for sensor node.

Keywords—sensor network, cloud computing, large scale, model, flexible

I. INTRODUCTION

The sensor network is a network consisting of spatially distributed devices equipped with sensors which are used to collect physical data and monitor environmental condition at different locations (e.g. [1], [2]). In a typical sensor network, nodes cooperate to complete the task of collecting raw data and return to the application back-end. The application back-end, which is usually a local sever, finishes the task of data storing, analysing and drives actuator.

In the last few years, all kinds of sensor networks have been deployed all over the word. If we combine some of these networks, we can get a larger sensor network, which makes

the applications over a larger area become possible. Sensor networks are various in architecture and implementation. The data of very large scale sensor network are polymorphous, heterogeneous, large in quantity and time-limited. In a very large scale sensor network, How to manage the sensing resources and computational resources, and how to store and process these data has become a key technology.

Another trend in information and communication technology is cloud computing. Cloud is a virtualized platform which offers open and uniformed access to extensible computational resources, storages, and software services. Three cloud computing models are proposed. Computer infrastructure resources are offered in the cloud is IaaS (Infrastructure as a Service) (e.g. [3], [4]). Computational resources with completely supporting environment are offered in the cloud is PaaS (Platform as a Service) (e.g. [5], [6]). Online software accesses offered in the cloud is SaaS (Software as a Service) [7].

Cloud computing model can easily handle the massive data storing and processing works. It seems suitable as the back-end of the very large scale sensor network. But there are still many problems to be solved if we want to make these two technologies cooperate together and to produce new value. As (1) How to store the data? If all the data are send to the data centers of the cloud at the sampling time, the inflow of massive data to the wide-area networks may cause network congestion. (2) Where to processing the data? If all the data of the sensors are processing at the cloud, the communication latency may make the applications requiring for real-time demand intolerable.

To solve these problems, we propose a flexible and multilevel data processing model based on cloud computing. In this model the massive sensor data and node information are stored in multilevel data storage including local data bases of different devices and distributed data bases of cloud; Different kinds of computations are decomposed and distributed on different nodes mainly considering their differences on computation ability and power supply. The clouds make the development and deployment of applications base on these very large scale data very easy.

This paper is structured as follows. Section II provides three deployment scenarios of large scale sensor network. In section III, we proposed a multilevel storage model for the application demands of large scale sensor network. Section IV describes a uniform data access model. Section V describes the application model. Section VI presents our verify platform. Finally, Section VII presents some conclusions and future works.

II. LARGE SCALE SENSOR NETWORK DEPLOYMENT SCENARIOS

A large scale sensor network can provide much useful sensor data such as temperature, humidity, location, light, sound, image and so on. These data could be used to provide services to many useful applications. Different application may have different demands on data sources and data processing. In this section, we consider there typical applications that may be deployed on a large scale sensor network. We analyse their requirements for the data storing and processing.

We consider a very larger sensor network all over the national expressways. This network is consisting of many small sensor networks, which cover a small area as the entrances, the exits, the service places and so on. There are many potential applications of such network.

A. Vehicle tracking

Vehicle can be tracked by different ways, as image, RFID. Vehicle tracking is a hard real-time application. When the police office wants to track a fugitive car all over the country, the data must be processed and answer must be given quickly at the place such as the entrances and the exits. So the computation must be delivered to the local servers, and there is no need to storing the historical data for this scenario.

B. Traffic dispatching

Traffic dispatching need real-time data of traffic flow, weather data and so on. Traffic scheduling is a soft real-time application, as several minutes lag will not cause suffering result. If all the traffic related data are sent to the data centers of the cloud, the inflowing data may cause network congestion. So the statistics information must be computed on the local servers, but the global decisions can only be made on the cloud, as only the cloud has the information of wide area. If the scheduling algorithm only base on the real-data, there is no need to store the collected data in the cloud, but if it is a prediction algorithm base on historical data, the collected data must be stored in the cloud.

C. Expressway planning

Unlike in the scenario of traffic dispatching, expressway planning does not need real-time data, but need huge historical data. The cloud data centers can collect statistical data from local servers, and complete the computation using the cloud computing resources.

III. DATA STORING MODEL

Above scenarios show that different application has different data demands. In this section, we propose a multilevel model for storing sensor data of the very large scale sensor network.

In our model, heterogonous sensors are distributed over a very lager area. Space close sensors are grouped into a sensor network. There are coordinators and gateways in each group which are responsible for collecting raw data, storing them on the local storage, and act as accessing to the wide-area networks. One or more sensor networks are connected to a local server, which provides the local back-up storage and computing ability. Several clouds are work above these local servers to grasp the data for application needs.

Row sensor data are adjusted at sensor nodes and gateways. Metadata of the sensor network is stored at its gateway. These metadata usually include sensor type, node type (as a node may contain several different sensors), node Mac address, node ID, node location, sensor ID and so on. The local servers gather data from the gateways, and store them in local data base. As different sensor network has different inner data format. The local servers have to transform these data to the same format, and provide uniform accesses. Sensor ID is a good example. In order to save energy, short IDs are used inner the small sensor network, but in order to discriminate sensor nodes in the large network, short IDs must be transformed global unique IDs. Metadata of sensor networks, local servers are used in this transformation. Historical data are stored in the local servers. Clouds grasp interesting data from local servers for different applications and store them in the data centers. Clouds also provide platform and software resources for these applications. Data sent by different data storage nodes can be aggregated in order to reduce redundancy and minimize network traffic load.

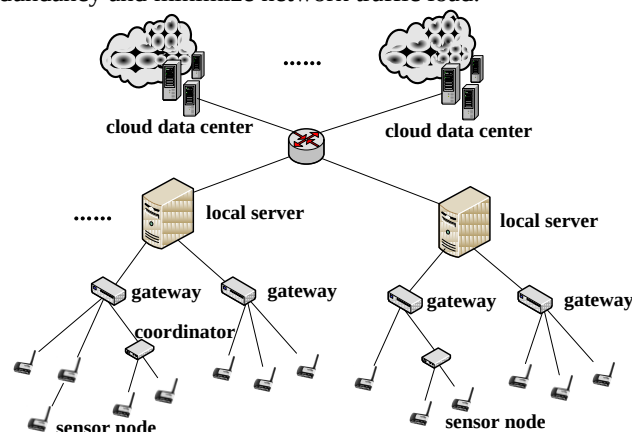


Figure 1. Multilevel data storing model

IV. DATA ACCESS MODEL

Different application requires different sensor data which stored at different places. We need a method to bind these data sets with applications. In this section we propose a uniform data access model. We first analyse four important properties of the sensor data.

A. Space

The sensor data is sampled at certain location and applications need to access sensor data of certain areas or locations. Space information of sensor data can be specified by longitude, latitude, and height. Some relevant space information is also attached to the data if this data related to a special and meaningful object. As a RFID sensor which used to monitor the bypass vehicles at an entrance of expressway, we can attach the entrance ID to this space data.

B. Time

The sensor data is sampled at a certain time and different application need sensor data at different time. Real-time applications use real time data, and other applications need historical data.

C. real sensor data and virtual sensor data

We divide the data into two categories: real sensor data and virtual sensor data. When we use a RFID sensor to track the bypass vehicles, we need real data sampled from the sensor. But we want to know the temperature at some point, there mayn't be a temperature sensor just at this point, but we can deduce it from the nearby sensors, it is virtual data. So we can see that real sensor data is the data which can directly get from a sensor. Virtual sensor data is the data that deduced from related data.

D. Data precision

The precision of sensor data is mainly affected by two factors: the precision of the sensor for real sensor data, and the method used to deduce the virtual sensor data.

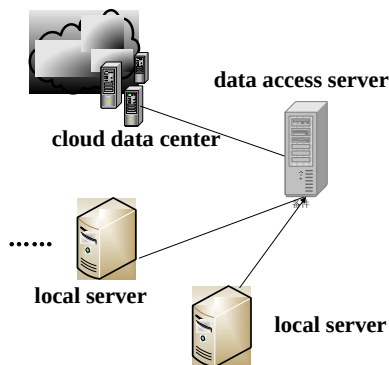


Figure 2. Uniform data access model

In order to providing a uniform access to the sensor data, we must consider these properties. In our model the local server is responsible to describe the data access interface. The specifics of data access interfaces are stored in a global server. Through this global server, the clouds know where to get their interesting data and how to deploy applications. Figure 2 shows this architecture.

In our model, we use a nest rectangular structure to describe the space property of the sensor data set. Figure 3 shows an example of a local server area. This server

connected to two networks: sensor network1 and sensor network2. The outside larger rectangular presents the local server area, and it contain two small rectangular which presents sensor network 1 area and sensor network 2 area respectively.

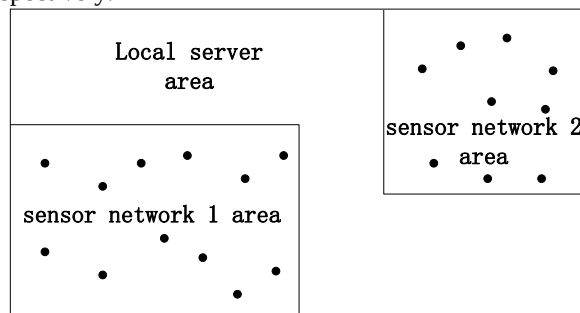


Figure 3. Uniform data access model

We also use a uniform method to describe the time attribute of every sensor data set. If a sensor set is described by [start time, end time], it means that the set contains historical sensor data which sampled between the start time and the end time. [start time, now) means the set contains sensor data from start time to now, and the real-time data. [now, now) means the set only contains real-time sensor data.

Figure 4 shows a simple XML format description for data access interface of the above example.

```
<server ID="1">
  <data_set ID="1" type="temperature"
    start_time=2011:1:11:12:54, end_time = now >
    <area right_top="(1000,1000)" left_bottom="(1500,
      1500)" >
      <sensor_network ID="1">
        <area right_toright_top="(1300,1000)"
          left_bottom="(1500,1200)">
          <node ID ="1"> </node>
          .....
          <node ID = "10"></node>
        </area>
      </sensor_network>
      <sensor_network ID="2">
        .....
      </sensor_network >
    </data_set>
  </server>
```

Figure 4. XML format description for data access interface

V. APPLICATION MODEL

Many sensor network applications have large amount of data to be deal with. Google's map-reduce [8] and Hadoop [9] are effective tools to support massive data computations. Figure5 shows the principle of map-reduce.

Briefly, map-reduce uses a map function to map the data stored in files into key-value pairs. All the produced pairs are routed by a master controller to one of several Reduce processes and all the pairs with the same key wind up at the same reduce process. The reduce processes use a reduce function to combine the values associated with one key to

produce a single result for that key. The master is a controller who monitors the map and reduce processes and is able to redo them if a process fails.

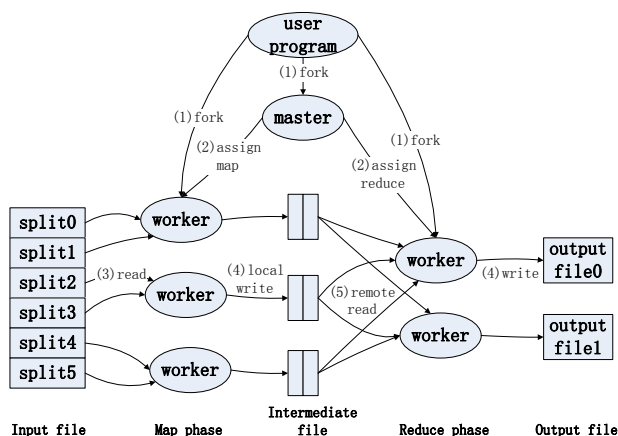


Figure 5. Map-reduce principle

Map-reduce model is a natural way to implement data-intensive applications in parallel. But using it in the very large scale sensor networks, there are still some drawbacks. (1) In map-reduce mode the input data are stored in files where the master can easily get them. But in the large sensor networks, the data are stored in different place with different format. (2) The application of sensor network usually needs to deal with real-time data. The time delay of map-reduce may make the real-time application intolerable. (3) Key-value map is not enough to describe the sensor data sets. The space and time attributes of sensor data give me natural hits of how to map the data.

In order to solve these problems, we propose a new parallel computation model for sensor network. Figure 6 shows this model.

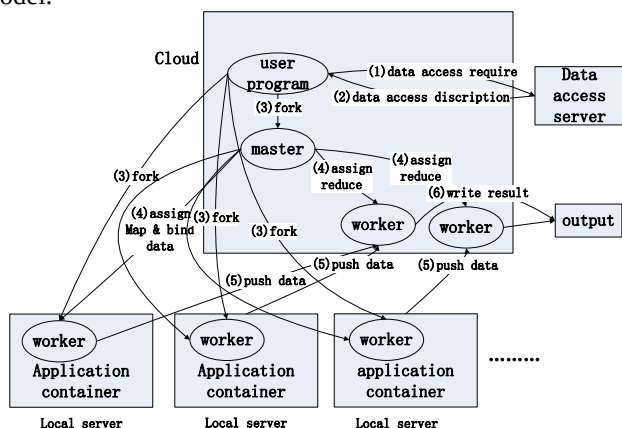


Figure 6. New computation model for sensor network

The steps of this new model are as follows. (1) The application submits the data access requires to data access server. (2) Data access server return the data access description to the application. (3) According to the data access description, the application program is deployed to the

workers on the application containers of local servers. (4) The workers of the local servers bind data to the application, and run the map function. (5) The workers of local servers push data to the workers on the cloud. (6) The workers on the cloud store the data on the cloud data centers, run reduce function and write the result to the output. The master monitors the map and reduce processes and tries to redo them if a process fails.

In this model we can see that several changes have been introduced. Application containers on local server provide executing environment for the map functions. They also bind data to the local workers. Real-time demands can be satisfied on local computation. Local workers push data to the works on cloud so the cloud could allocate enough computing resources to deal with the data as required.

As in the scenario of vehicle tracking, for example, the application first looks up all the useful sensor data sources from the data access server. The data sources may include RFID signals, images, or videos. The program is deployed to the application containers on the local server according to the look up result. The data sources are bind to the application by the application containers. Different methods are used to deal with different data type. Simple ID comparing is used for RFID signals, image analyze technique is used for images and videos. The results are pushed to different workers on cloud. These workers on the cloud are distinguished by different tracking objects, so the objects move track could be synthesized on the cloud.

In the other scenario of expressway planning, the workers on the local servers are responsible to collect statistical data from local servers. These historical data are stored in the cloud data centers. The workers on the cloud analyse these data to get the planning result.

VI. PLATFORM TO VERIFY THE MODEL

In order to verify the effectiveness of this model, we have setup a platform using some open source software.

We have used five types of sensor. They are temperature sensor, light sensor, humidity sensor, location sensor and carbon dioxide density sensor. We set up four WSNs (wireless sensor networks) base on WIFI and Zigbee protocol. Four Mini 6410 ARM11 embedded gateways are used to collecting data from sensor network. Two local servers each connect to two gateways. Data are formatted on the local servers and the uniform access descriptions are sent to a global data access server. We also set up two cloud computation environment base on and Cassandra, MongoDB distribution data bases.

Different data bases including SQLite, TinyDB, MySQL, MongoDB and Cassandra have been used for different type of devices to store data. SQLite and TinyDB are used on gateways to store meta data of the sensor networks. MySQL is used on local servers to store sensor data and historical data. MongoDB and Cassandra are used to store application related data.

We have developed some cloud applications for this platform. In one application, we track the highest temperature. Works on local servers report the highest temperature of

associated WSNs to the work on the cloud. The worker on the cloud tracks the highest temperature node and shows it information. In another application, temperature, humidity, light and carbon dioxide density data are collected by the four workers on the local servers. Each worker is for a WSN. Four workers on the cloud analyses these data and give out the environmental indexes of four areas.

The result shows that this platform base on the new computation model is flexible, easy to develop new application and energy balanced for sensor node.

VII. CONCLUSIONS

In this paper, we argue that tradition Map-Reduce computation model is not enough for large sensor network as the data of very large scale sensor network are polymorphous, heterogeneous, large in quantity and time-limited, and there are many real-time application demands. So we propose a new computation model for very large sensor network. Multilevel storage model, uniform data access, and local application container are introduced.

But there are still many problems that are worth to be exploited. We list some of them as follows: (1) How to give a uniform description the sensor data set precisely and effectively. (2) The data structure used to describe the data access information on the data access server. (3) How to describe the data precision property. (4) How to bind the data sources to the workers on the local application container.

REFERENCES

- [1] Akyildiz I.F, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor network: A survey [J]. Computer Networks, 2002, 38(4): 393422.
- [2] Ren Fengyuan, Huang Haining, Lin Chuang. Wireless sensor networks [J]. Journal of software, 2003, 14(2):1148-1157.
- [3] Amazon Elastic Compute Cloud (EC2), available online: <http://aws.amazon.com/ec2/>, accessed July 2010.
- [4] SliceHost Cloud Services, available online: <http://www.slicehost.com/>
- [5] Google App Engine, available online at: <http://code.google.com/appengine/>
- [6] Sales Force, available online at: <http://www.salesforce.com/platform/>
- [7] A. Dubey, and D. Wagle, "Delivering software as a service," The McKinsey Quarterly, May 2007.
- [8] J. Dean and S. Ghemawat, "Mapreduce: Simplified Data Processing on Large Clusters," Comm. ACM, vol. 51, no. 1, pp. 107-113, 2008.
- [9] Apache, "Hadoop," <http://hadoop.apache.org/>, 2006.