A Structured Overlay Network for Aggregating Sensor Data

Atsushi Takeda*, Takuma Oide† and Akiko Takahashi‡

* *Department of Information Science, Tohoku Gakuin University*
2-1-1 Tenjinzawa, Izumi-ku, Sendai 981-3193, Japan
Email: takeda@cs.tohoku-gakuin.ac.jp
† *Advanced Course of Information and Electronic System Engineering,*
Sendai National College of Technology
4-16-1 Ayashi-Chuoh, Aoba-ku, Sendai 989-3128, Japan
Email: a1102003@sendai-nct.jp
‡ *Department of Information Systems, Sendai National College of Technology*
4-16-1 Ayashi-Chuoh, Aoba-ku, Sendai 989-3128, Japan
Email: akiko@sendai-nct.ac.jp

***Abstract*—Symbiotic computing services require big data from a huge number of sensor devices in order to detect the real-world, so a distributed database system which can handle the big data is required for symbiotic computing system. In this paper, we propose a scalable structured p2p network which supports data aggregation. In the proposed p2p network, each node forwards partial statistical results to other nodes, and each node aggregates partial statistical results in order to calculate a complete statistical result. In the proposed p2p network, the data aggregation process does not need any specific protocols, so the communication cost of the extension mechanism is very low. We simulated the proposed p2p network by using a p2p network simulator. The simulation results indicate that the proposed data aggregation mechanism is effective enough.**

***Keywords*-peer-to-peer; structured overlay; structured p2p; data aggregation;**

I. Introduction

Symbiotic computing services are information services based on the real-world situation[1]. In order to provide symbiotic computing services, a database management system which can handle data from sensor devices is required for detecting the real-world situation. Especially, data aggregation mechanism is necessary for providing effective symbiotic computing services. However, the size of the sensor data is more than peta bytes, which is called Big Data. It is difficult to handle these data by using only one computer, so we need a distributed database management system which supports not only data management but also data aggregation.

In this paper, we propose a structured p2p network which supports data aggregation. The proposed p2p network achieves a scalable object management by using one-dimension ID space in the same way as Chord#[2]. In addition, our p2p network has a scalable data aggregation mechanism whose communication cost is $O(\log N)$, where N is the number of nodes. The proposed aggregation mechanism can calculate statistical results such as summation, average, maximal-value or minimal-value. Therefore, the proposed p2p network can analyze big data from sensor devices statistically, so this p2p network is usable as a database management system of symbiotic computing services for detecting the real-world. By using the proposed p2p network as a database management system, we can develop a symbiotic computing system which can handle big data from a huge number of sensor devices.

A basic concept of data aggregation is that each node forwards partial statistical results to other nodes, and each node aggregates partial statistical results in order to calculate a complete statistical result. The partial statistical results are forwarded as a part of maintenance messages which are sent by each node at a network maintenance process, and each node obtains the partial statistical results as a part of search messages. Thus, the proposed p2p network does not need any extra protocols for data aggregation. After a node obtains partial statistical results, the node calculates a complete statistical result by aggregating the partial statistical results.

The complete statistical result is calculated from partial statistical results which are forwarded by several nodes. So, there is a time lag between data updating and data aggregation. Therefore, the proposed data aggregation is not completely accurate. Especially, when the p2p network is unstable, the proposed data aggregation tends to be inaccurate. However, we believe that the complete statistical result calculated by proposed data aggregation is effective enough and we can use the proposed p2p network as database management system of symbiotic computing services.

In this paper, we describe a structure of the proposed p2p network. And, we explain an algorithm of data aggregation. This paper presents a description about forwarding partial statistical results as a part of maintenance messages and calculating a complete statistical result from forwarded partial statistical results. Through simulation results, we also show that the proposed p2p network achieves an effective data aggregation.

The remainder of this paper is organized as follows. Section 2 presents a description of related works and clarifies the issues. Section 3 proposes the structured p2p network supporting data aggregation. Section 4 describes evaluation through simulation results. Finally, section 5 presents a summary of our conclusions.

II. Related Works

Cooperation between information provider and sensor devices is important for providing symbiotic computing services. Thus, a database management system is required for managing and aggregating data from sensor devices. However, the size of the sensor data is more than peta bytes. Therefore, in order to provide effective symbiotic computing service, a distributed database management system which supports data aggregation is required.

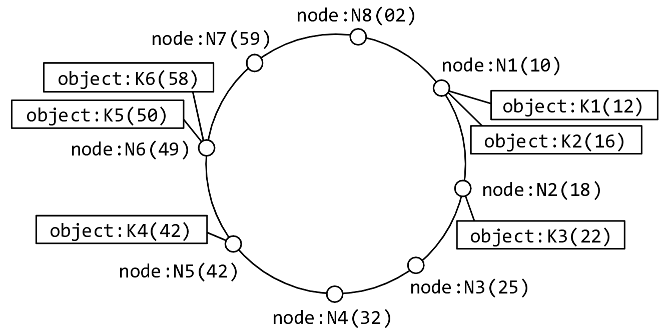
Distributed database management systems have been developed for a long time. Recently, BigTable which is a distributed database management system was proposed[3]. In addition, MapReduce which is a data aggregation mechanism was also proposed[4]. These systems can handle big data and achieve data aggregation. However, these proposed systems assume that computers in the systems are connected by specific high-speed network. Thus, when we start these systems, we must configure complex settings such as server settings or network settings.

On the other hand, structured p2p networks do not require complex settings usually. Recent years, a lot of structured p2p networks have been proposed. CAN[5], Chord[6], Pastry[7] and Tapestry[8] are structured p2p networks which are called Distributed Hash Table (DHT). DHT is a scalable p2p network, and can handle big data. However, DHT supports only exact search. Therefore, it is hard to use DHT as database management system of symbiotic computing services.

SkipGraph[9] and Chord#[2] are structured p2p networks which supports range queries. These p2p networks are also scalable p2p networks which can handle big data. In addition, these p2p networks enable range search and prefix search. We think that these p2p networks are usable as a database management system of a web service. However, these networks do not support data aggregation, so these p2p networks are not usable as a database management system of a symbiotic computing service.

Estimating the summation of the number of nodes is required for efficient dynamic load balancing. So, some data aggregation mechanisms for estimating the size of the p2p networks were proposed[10], [11]. In these structured p2p networks, each node obtains information from other nodes which are chosen at random, and calculates the size of the p2p network by aggregating the information. However these data aggregations calculate the network size by using partial data from several nodes which are chosen at random. Therefore, these mechanisms can not calculate a complete statistical result by aggregating information from all nodes in the p2p network.

In our proposed p2p network, each node can obtain partial statistical results which are enough to calculate a complete statistical result. This means that the proposed p2p network can calculate a statistical result of big data from sensor devices. Therefore, our proposed p2p network is more suitable for a database management system for symbiotic computing service than existing proposals.



The numbers in parenthesis are locations on the ID space

Figure 1. ID Space of the Proposed P2P Network

III. Proposal: Structured P2P Network with Aggregation Mechasnism

A. Overview of the Proposed P2P Network

We propose a structured p2p network which supports data aggregation. Basically, the structure of the proposed p2p network is as same as Chord#[2]. Fig.1 shows the ID space of the proposed p2p network. The proposed p2p network uses one-dimension ID space, where all nodes and all objects are located on. ID space is a circle that is indexed [0, ID_{max}) linearly, where ID_{max} is a huge number such as 2^{256}. Connections among nodes are decided by the node locations on the ID space. And, object assignments to nodes are decided by the object locations on the ID space.

Each node's location on the ID space is decided when the node joins to a p2p network. After the node joins to the p2p network, the node makes connections to both the clockwise closest node and the counterclockwise closest node. The clockwise closest node is called "successor" node, and the counterclockwise closest node is called "predecessor" node. For example, in fig.1, a successor node of node N1 is node N2, and a predecessor node of node N1 is node N8. In addition, each node makes connections to the 2^ith clockwise nodes. Because of these connections, the proposed p2p network achieves not only scalable object search but also scalable data aggregation.

Each object's location is decided by the object's key such as name, type or meta-information. For example, if objects are sensor data, the keys would be decided by the sensor device IDs or the sensor device locations. Each node manages objects which are located between the node and the successor node. For example, in fig.1, node N1 manages object K1 and K2 because object K1 and K2 are located between node N1 and a successor node of node N1.

When a node joins to a p2p network, the node makes connections to both its successor node and its predecessor node, and the node receives objects which are located between the node and the node's successor. When a node leaves from a p2p network, the node sends message to both its successor and its predecessor, and the node sends objects to its predecessor node.

In the proposed p2p network, object locations are decided by the object's key, so the object locations are not uniform. Therefore, in order to distribute objects and messages uniformly, dynamic load balancing mechanism is required. In our proposed p2p network, we can decide node's locations freely, so we believe that we can add a load balancing mechanism to the proposed p2p network by using existing dynamic load balancing algorithms which are used in an existing structured p2p network such as Mercury[12] or Waon[13].

In our proposed p2p network, basic procedures, which are a joining process, a leaving process and a search process, are same as Chord#. However, data aggregation procedure is novel. There is no structured p2p network which supports data aggregation. The proposed data aggregation procedure contains 2 steps: each node forwards partial statistical results to other nodes at a network maintenance process, and a node obtains the partial statistical results in order to calculate a complete statistical result. We explain about data aggregation procedure in the following sections.

B. Forwarding Partial Statistical Results

In the proposed p2p network, each node has following attributes for data aggregation.

$$
\begin{aligned}
\text{node} &:= \; < \text{objects}, \text{successor}, \text{routes} > \\
\text{successor} &:= \; \text{node} \\
\text{objects} &:= \; \{\text{object}_0, \text{object}_1, \text{object}_2, \cdots\} \\
\text{routes} &:= \; \{\text{route}_0, \text{route}_1, \text{route}_2, \cdots\} \\
\text{object}_i &:= \; < \text{key}, \text{value} > \\
\text{route}_i &:= \; < \text{node}, \text{statistic} > \\
\text{statistic} &:= \; < \textit{a list of target statistical data} >
\end{aligned}
$$

In this definition, *successor* represents the node's successor, *objects* represents objects assigned to the node, *routes* represents communicating nodes and *statistic* denotes a partial statistical data.

Fig.2 shows a procedure of forwarding partial statistical results. At first, each node creates a list of target statistical data which is based on objects assigned to the node (at line: 03). These statistical data means a statistical result of objects which are located between the node and the successor node. Then, the node starts network maintenance process. In the network maintenance process, each node sends maintenance request messages to nodes which the node communicates with, and receives maintenance response messages from the communicating nodes. In the proposed p2p network, a maintenance response message contains a partial statistical data of sender node, so each node gets new statistical data during maintenance process. After the node gets new statistical data from its successor node, the node aggregates the node's own statistical data and the successor's statistical data (at line: 11). The aggregated statistical data means a statistical result of objects which are located between the node and the 2nd clockwise node. In the same way, each node gets new statistical data from 2^ith clockwise nodes. And the node aggregates these statistical data. As this result, each node has partial statistical results of objects which are located between the node and 2^ith clockwise nodes. Each node remembers the partial statistical results as route node's attribute.

```
01: // update routing table and forward statistical results
02: n.update ( ) {
03:   statistic = < a list of target statistical data >;
04:   node = n.successor;
05:   c_distance = distance(node);
06:   p_distance = 0;
07:
08:   for (i = 0; p_distance < c_distance; i = i + 1) {
09:     n.routes[i].statistic = statistic;
10:     n.routes[i].node = node;
11:     statistic =
          aggregate(statistic, node.routes[i].statistic);
12:     node = node.routes[i].node;
13:     p_distance = c_distance;
14:     c_distance = distance(node);
15:   }
16: }
```

Figure 2. Pseudo-code of Forwarding Partial Statistical Results

Our proposal does not need any specific protocols, so a requirement for forwarding partial statistical results is only extension of maintenance response messages. Because our proposal does not require new messages to forward partial statistical results, the communication cost of the proposed procedure is very low. In addition, it is easy to apply the proposed forwarding procedure to existing structured p2p networks.

```
01: // get a statistical result between node.id and id
02: n.calculate ( id ) {
03:   if ( n.isResponsible(id) ) {
04:     ret.node = n;
05:     ret.statistic =
            < a statistical result of assigned objects >;
06:     return ret;
07:   }
08:
09:   forward = n.routes[0];
10:
11:   for (i = 1; i < count(n.routes); i = i + 1) {
12:     if ( distance(n.routes[i].id) < distance(id) ) {
13:       route = n.routes[i];
14:     }
15:   }
16:
17:   result = route.node.calculate(id);
18:   ret.node = result.node;
19:   ret.statistic =
          aggregate(route.statistic, ret.statistic);
20:   return ret;
21: }
```

Figure 3. Pseudo-code of Calculating a Complete Statistical Result

C. Calculating a Complete Statistical Result

In the proposed p2p network, each node has partial statistical results of objects which are located between the node and the 2^ith clockwise nodes. In order to calculate a complete statistical result, a node obtains partial statistical results which are required for the calculation. This procedure is similar to a search procedure of existing structured p2p networks.

Fig.3 shows a procedure of calculating a complete statistical result. A function *calculate* obtains partial statistical results from other nodes, and calculates a complete statistical result of objects which are located between the node's ID and the specified *id*. At first, a node checks if the specified *id* is in a responsible range of the node (at line: 03). If the *id* is in a responsible range of the node, the node returns a complete statistical result immediately (at line: 04-06). If the *id* is not in the responsible range, the node sends a search request message to a forward node which is decided by a routing table of the node (at line: 11-15). The forward node which has received a search request message returns a search response message which contains a partial statistical result of objects which are located between the forward nodes' ID and the specified *id* (at line: 17). The node, which has received a search response message, aggregates a partial statistical result owned by the node and a partial statistical result received from the forward node (at line: 19). As this result, the node gets a complete statistical result of objects which are located between the node's ID and the specified *id*.

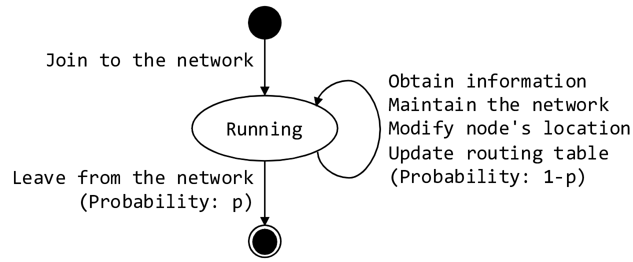


Figure 4. Node Activity in P2P Network Simulator

```
var/lib/pgsql/base/17229/16693         => 192.168.1.16
usr/share/man/man3/Net.Ping.3pm        => 192.168.1.5
usr/include/ncurses/term.h             => 192.168.1.9
usr/lib/python2.5/test/test_capi.py    => 192.168.1.12
usr/share/man/man3/clearok.3x.gz       => 192.168.1.17
bin/c++filt.exe                        => 192.168.1.3
lib/ruby/1.8/soap/streamHandler.rb     => 192.168.1.6
lib/ruby/1.8/tk/listbox.rb             => 192.168.1.6
lib/python2.5/hmac.pyo                 => 192.168.1.4
```

Figure 5. Objects used for the Simulation

The procedure of calculating a complete statistical result is logically same as a procedure of searching an object in structured p2p networks. Thus, the communication cost of calculating a complete statistical result is $O(\log N)$, where N is the number of nodes. This means that the proposed data aggregation is scalable. In addition, it is easy to add the proposed calculating procedure to existing structured p2p networks because the requirement is only extension of a search message.

IV. Simulation Evaluation

A. Implementation of Simulator

We implemented a p2p network simulator in order to evaluate our proposal. The simulator is implemented by using Java, and the number of the code line is about 20,000. In this simulator, each node is implemented as objects which perform a joining process, a leaving process or a network maintenance process independently and mutually communicate at every step. Packet loss and latency are not simulated in this simulation.

Fig.4 shows a activity of a node in the p2p network simulator. At first, all nodes which are out of the p2p network perform joining process. Then, each node performs a leaving process with probability p which means a churn parameter, otherwise the node performs a network maintenance process.

Fig.5 portrays some objects used in the simulation. This simulation assumes that shared objects in the p2p network are files in a UNIX operating system. Consequently, in this simulation, nodes share some files of FreeBSD 8.0, which

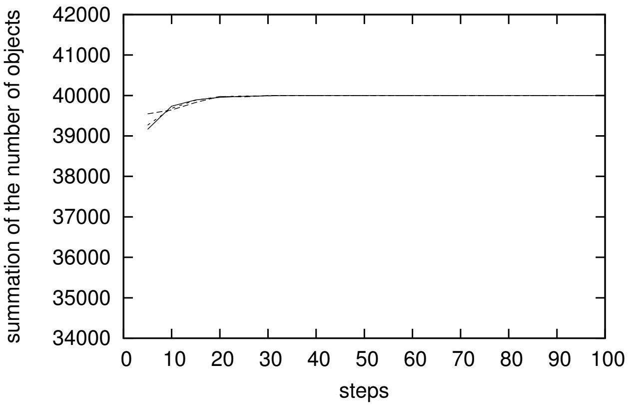


Figure 6. Summation of the Number of Objects in a Stable Condition

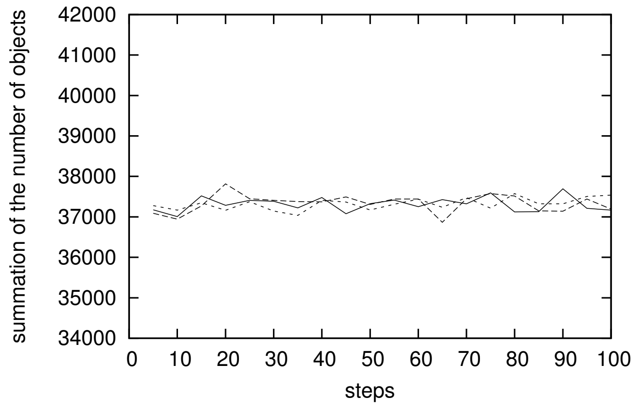


Figure 7. Summation of the Number of Objects in an Unstable Condition

is a UNIX operating system. This simulation assumes that shared objects are pointers such as URLs, so a node requires only a message when the node transfers an object to another node. The shared objects are put on ID space based on the file paths.

We simulated the proposed structured p2p network by using a simulator described above. In this simulation, the number of nodes is 4,000, and the number of objects is 40,000. We simulated the p2p network at 2-type conditions: one is a stable condition where parameter p is 0.0, another is an unstable condition where parameter p is 0.1. Through the simulation results, we evaluate accuracy of the proposed data aggregation mechanism.

B. Evaluation of Accuracy

Fig.6 shows the summation of the number of objects which is calculated with the proposed data aggregation mechanism by three nodes which are chosen at random. In this simulation, the number of objects is 40,000, and the p2p network condition is stable. Fig.6 indicates that the chosen nodes calculated that the summation of the number of objects is 40,000. This means that the proposed data aggregation mechanism is accurate when the p2p network is stable.

Fig.7 shows the summation of the number of objects which is calculated with the proposed data aggregation mechanism by three nodes which are chosen at random. In this simulation, the number of objects is 40,000, and the p2p network condition is unstable. Fig.7 indicates that the chosen nodes calculated that the summation of the number of objects is about 37,300. In this simulation, the calculated summation is a little different from the actual summation of the number of objects. This means that a process of forwarding partial statistical results is unstable because some nodes perform a joining process or a leaving process while partial statistical results are forwarded. This simulation result indicates that the proposed data aggregation mechanism is not so accurate when the p2p network is unstable. But the statistical result is close to actual value, so it is possible to use the proposed p2p network as a database management system of symbiotic computing service.

V. Conclusion

Symbiotic computing services require big data from sensor devices in order to detect the real-world, so a distributed database system is required for symbiotic computing system to manage the big data. Existing distributed database systems assume that computers are connected by a dedicated high-speed network, so existing systems need complex settings.

In this paper, we proposed a structured p2p network which supports data aggregation. In the proposed p2p network, each node forwards partial statistical results to other nodes, and each node aggregates partial statistical results in order to calculate a complete statistical result. The data aggregation process does not need any specific protocols, so it is easy to apply the proposed data aggregation to existing structured p2p network. We simulated the proposed p2p network by using a p2p network simulator, and the simulation results indicate that the proposed data aggregation mechanism is accurate when the p2p network is stable. In addition, simulation results show that the proposed mechanism calculates a value which is close to actual value even if the p2p network is unstable.

We think that the proposed data aggregation mechanism gives many advantages to structured p2p networks. In the future, we will make a symbiotic computing service by using the proposed p2p network. In addition, we are going to develop more efficient load balancing algorithm with the proposed data aggregation mechanism.

Acknowledgment

This work was supported by a Grant-in-Aid for Challenging Exploratory Research (No.23650032) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

References

[1] T. Suganuma, K. Sugawara, and N. Shiratori, "Symbiotic computing: Concept, architecture and its applications," *Lecture Notes in Computer Science*, vol. 4611, pp. 1034–1045, 2007.

[2] T. Schütt, F. Schintke, and A. Reinefeld, "Range queries on structured overlay networks," *Computer Communications*, vol. 31, no. 2, pp. 280–291, 2008.

[3] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Transactions on Computer Systems*, vol. 26, no. 2, 2008.

[4] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Communications of the ACM*, vol. 51, no. 1, pp. 107–113, 2008.

[5] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," *Proceedings of ACM SIGCOMM*, pp. 161–172, 2001.

[6] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, 2003.

[7] A. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," *Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms*, pp. 329–350, 2001.

[8] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, and J. D. Kubiatowicz, "Tapestry: A resilient global-scale overlay for service deployment," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 41–53, 2004.

[9] J. Aspnes and G. Shah, "Skip graphs," *ACM Transactions on Algorithms*, vol. 3, no. 4, 2007.

[10] T. M. Shafaat, A. Ghodsi, and S. Haridi, "A practical approach to network size estimation for structured overlays," *Proceedings of the 3rd International Workshop on Self-Organizing Systems*, pp. 71–83, 2008.

[11] D. Warneke and C. Dannewitz, "Load balancing in p2p networks: Using statistics to fight data and execution skew," *Journal of Advances in Information Technology*, vol. 2, no. 1, pp. 40–49, 2011.

[12] A. R. Bharambe, M. Agrawal, and S. Seshan, "Mercury: Supporting scalable multi-attribute range queries," *Proceedings of ACM SIGCOMM*, vol. 34, pp. 353–366, 2004.

[13] A. Takeda, T. Oide, and A. Takahashi, "Simple dynamic load balancing mechanism for structured p2p network and its evaluation," *International Journal of Grid and Utility Computing*, vol. 3, no. 2, pp. 126–135, 2012.