International Journal of Control and Automation
Vol. 8, No. 7 (2015), pp. 125-132
http://dx.doi.org/10.14257/ijca.2015.8.7.13

The Research of Asynchronous Based Dispersion Handling Algorithm in order to Reduce Server Traffic

Seung-Min Baek, Hyun-Jun Shin and Chang-Heon Oh

Korea University of Technology and Education, Electrical, Electronics and Communication Engineering, Cheonan, Korea
gbtsoul@koreatech.ac.kr, champ5866@koreatech.ac.kr, choh@koreatech.ac.kr

Abstract

In this dissertation, we suggest an asynchronous-based dispersion-handling algorithm in order to improve on a common problem in previous synchronous-based data handling algorithms. The suggested type of algorithm assigns the diverse device's certification process into the created secondary server. Each of the secondary servers then carries out data processing by asynchronous style. In order to also evaluate the suggested algorithms capacity, we constructed mono-server handling environment and dispersion handling environment, then measured the TCP Network handling between server and terminal, while also measuring the CPU handling of the server. As a result, TCP Network handling reduced about 26.5% of handling compared to the mono handling style. The server's CPU handling was reduced to about 33.49% of handling compared to mono handling style.

Keywords: *Location Data, Distributed Processing Algorithm, Asynchronism, Server*

1. Introduction

Due recently to the propagation of smart devices, there have been a lot of Location Based Service (LBS) providing diverse services based on the users' location. In order to provide this LBS, the positioning technology of catching the users' location right away has become absolutely important [1, 2]. The location data that appears in the positioning process is an enormous atypical data, which means Big Data [3, 4]. In the past years, single servers collected location data from a lot of devices and carried out the work of assigning data processing into one server in a synchronized style in order to handle the location data. However, this type of work made it difficult to distribute the server's work efficiently because when the number of device connected to server increases, the data processing also increased. Also, when the data influx is too much at a time, there is a risk of server overload [5,6]. Based on this, we have suggested an asynchronous based dispersion-handling algorithm. This suggested algorithm improved on the previous type, which handled data by connecting a lot of devices into one server in a synchronized manner into connecting into a lot of secondary servers in a divided style, in an asynchronous manner.

2. Distributed Processing Algorithm

In this chapter, we will explain the process and the principle that make up an asynchronous based dispersion-handling algorithm. In order to handle location data in a virtual server, it needs to handle the data collected from smart devices in a divided manner. Since the amount of collected location data is really enormous, the data handling needs to consider versatile aspects such as memory, arithmetic operation, network overload and so on. In the previous data handling algorithm, like in the picture x, it incessantly repeats question and answer in order to handle the transmitted data. So the synchronization between processors is needed in the process of transmitting data to other processors.

ISSN: 2005-4297 IJCA

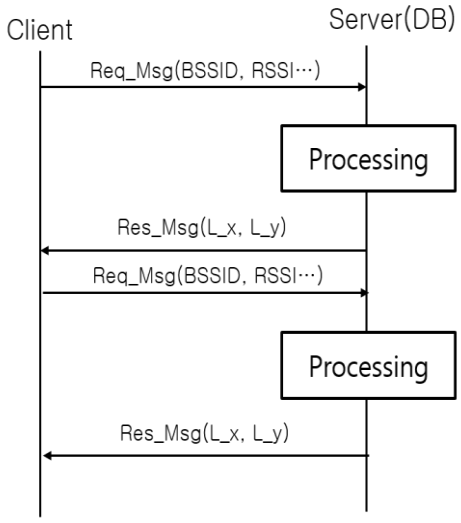

Figure 1. The Previous Style of Handling Data

For example, when two terminals send data to each other, two processors will use the server's sharing memory and receive the data. Then like in the picture x, the buffer of the server carries out a lot of processes in order. In this case, as the number of terminals increase the data handling process also increases, but since a lot of data continually gets handled at the same time, there is a risk of server overload or the loss of data. The server's efficient work distribution also becomes impossible, which also presents a problem.

First, we schematized the previous mono server handling the location data transmitted from a lot of terminals in the Figure 1. In this type, a lot of terminals get connected into a single server after certification, and because of this, the server's buffer repeats questioning in order as the location data arrives. In this case, as the number of device increases, the unnecessary question and answer process increases also.

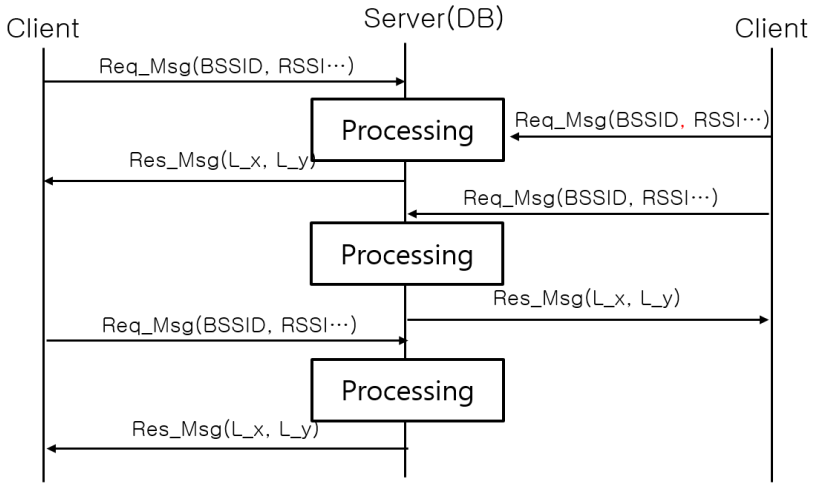

Figure 2. The Previous Style of Mono Server Data Handling Algorithm

In this dissertation, in order to improve on this type, we suggested asynchronous based data handling algorithm with the Figure 2. The suggested algorithm divides the server into a main server and many secondary servers, while each of the secondary servers carry out the certification process independently with one or more of the devices. The certificated device connects into an assigned secondary server and carries out the question and answer process, after which the location data collected in this process is transmitted into the main server and gets constructed as a database.

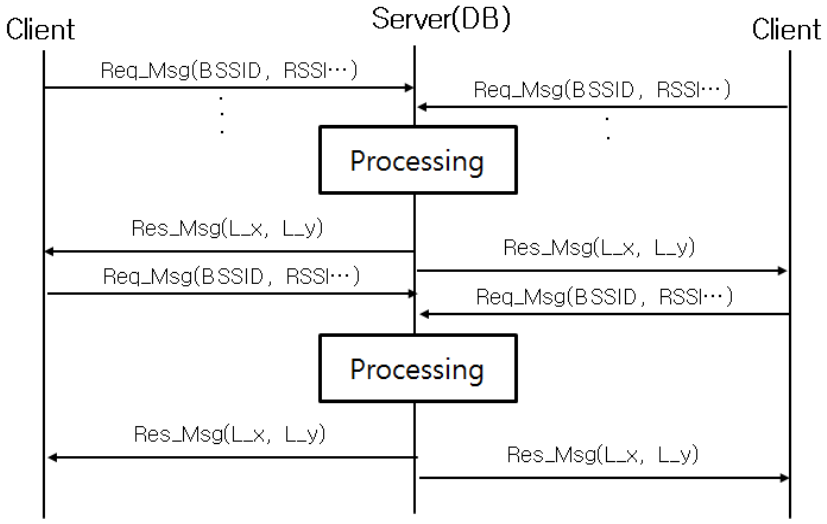

Figure 3. Data Dispersion Handling Algorithm

3. Performance Evaluation

In this chapter, we constructed a test bed like the chart X and progressed an experiment in order to evaluate the capacity of the developed data dispersion handling algorithm.

Table 1. Data Dispersion Handling Experiment Environment

Categories		Contents
Server specification	Main server(1 only)	CPU : Quad-core, 3.4GHz Memory : 8GB
	Secondary servers(2)	CPU : I7-3770, 3.4GHz Memory : 4GB
Operating system	Main server	Windows Server 2008
	Secondary server	Windows 8
Terminal specification		Samsung galaxy note2 (2 notebooks) LG G2 (1 only)
Capacity Analysis Tool		Jmeter (ver. 2.12)
Ramp-Up period (in seconds)		1

In this chapter we constructed each of the mono server handling environment and dispersion handling environment and then measured the TCP Network handling and server's CPU handling in order to evaluate the capacity of the algorithm. As a tool for capacity analysis, we used Jmeter, which is an Apache foundation-made web application that analyzes diverse services' capacity. The unit used here is Jmeter's random figure. In the case of TCP Network handling experiment, we settled an environment wherein a lot of terminals connected into one server, then collected Network handling amount when the data is transmitted in a TCP style in order to compare with the previous handling style. Then, based on the asynchronous dispersion-handling algorithm suggested in this dissertation, we collected the Network handling amount, which appears when the main server and the 2 secondary servers get certificated in the terminal. In the case of the Server's CPU handling, it is hard to notice the difference in capacity although we apply stress on the three terminals, which then becomes a problem. Based on this, we used Jmeter in this dissertation and connected 100 fixed virtual nods into one server in a virtual environment, then applied data dispersion handling algorithm, created two secondary

servers, and then constructed a virtual environment which assigned 100 virtual nods in order to catch a noticeable difference.

Table 2. TCP Network Handling Experiment Results

Experiment number	handled amount when mono handled	handled amount when handled dividedly			
		Main server	Secondary server 1	Secondary server 2	Total handled amounts
1	6.0	1.2	3.0	2.7	10.2
2	5.4	0.88	3.2	2.8	9.48
3	5.2	1.3	2.9	2.7	9.4
4	5.4	1.5	2.9	2.7	9.8
5	6.0	1.6	2.9	2.7	10.5
6	6.0	1.8	2.8	2.7	10.6
7	5.7	2.0	2.8	2.7	10.5
8	7.0	2.2	2.8	2.7	12
9	5.4	1.3	3.0	2.7	9.7
10	5.2	1.4	2.9	2.7	9.5
Average	5.73	1.52	2.92	2.71	10.17

Table 3. The Result of Experiment Regarding CPU Handling Amounts

Experiment nubmer	handled amount when mono handled	handled amount when handled dividedly
1	87.3	29.7
2	100	34.2
3	104.2	34.9
4	103.4	34.1
5	102.8	35.7
6	97.2	33.9
7	106.1	35.3
8	101	35
9	100.9	34.3
10	103	29.8
Average	100.59	33.69

As a result of the experiment, by applying data dispersion handling algorithm, the average handled amount of main server was 1.52/min, which was reduced from 4.21/min from the mono handling type. And if we compare this with the mono handling style, we were able to notice 26.5% of handling amounts decrease. Also, when we collected server's CPU handling amounts by applying data dispersion algorithm, we were able to notice that the main server's average handling amount was 33.69/sec, which had decreased 66.9/sec from the mono handling style. As we put together the results of the experiment, when we use the asynchronous data dispersion algorithm suggested here in the dissertation, the handled amount per hour was lowered compared to the mono server handling type. This means that since we can lower the traffic of the server, when we handle big data, we can increase the efficiency of sever work and lower the risk of server overload.

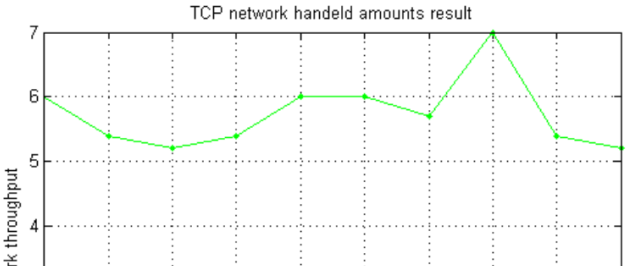

Figure 4. TCP Network Handled Amounts Result

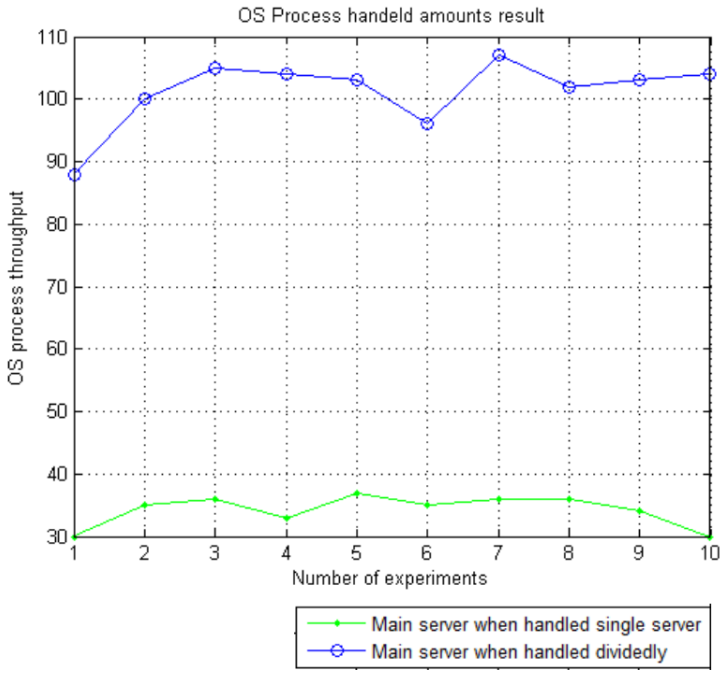

Figure 5. Server CPU Handled Amounts Result

4. Discussion and Conclusions

Although the LBS is being activated due to the propagation of smart devices, in previous times, a single server collected location data from a lot of devices then assigned data processing into one server in order to handle location data. This type of handling had a risk of server overload since data processing also increases as the number of devices connected to the server increases. In this dissertation, we suggested asynchronous based data dispersion handling algorithm in order to improve the problems that could appear in the previous synchronized data-handling algorithm. The suggested style of algorithm assigned the certification of a lot of devices into the created secondary server, then each of

the secondary servers carried out data processing in an asynchronous manner. In order to also evaluate the suggested algorithms capacity, we constructed a mono-server handling environment and dispersion-handling environment, then measured the TCP Network handling between server and terminal, while also measuring the CPU handling of the server. As a result, TCP Network handled amounts showed 26.5% decrease in the amounts handled compared to the mono type, and the Server's CPU handled amount showed 33.49% decrease in the amounts handled compared to the mono type. Therefore if we use the algorithm suggested here in this dissertation, compared to the previous type of mono server handling, we can be able to confirm the decrease in the handling amount per hour. By using this advantage, when we handle big data, it can increase the efficiency of server work and lower the risk of server overload.

Acknowledgement

The study was performed with support of the graduate school affairs team, Korea University of Technology and Education in 2014.

References

[1] M. Gruteser, D. Grunwald, "Anonymous Usage of Location-based Services through Spatial and Temporal Cloaking", Proceedings of the 1st International Conference on Mobile Systems **(2003)**.
[2] C. Y. Chow, M. F. Mokbel, X. Liu, "A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-based Service", Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems , **(2006)**, pp. 171-178.
[3] S. Madden, "From Databases to Big Data", **(2012)**, IEEE Internet Computing, vol. 16, no. 3, pp. 4-6.
[4] B. Brown, M. Chui, J. Manyika, "Are You Ready for the Era of 'Big Data'", **(2011)**, McKinsey Quarterly, **(2011)**, no. 4, pp. 24-35.
[5] G.-C. Park, K. Sung, S.-S. Kim, "Traffic Distributed Processing System Implementation on the Web Server Networking", **(2004)**, Journal of information and communication convergence engineering, vol. 8, no. 4, pp. 846-853.
[6] J.-H. Jung, K.-H. Eom, "Efficient Method of Processing Long-term Transactions for Distributed Environment", **(2003)**, Journal of information and communication convergence engineering, vol. 7, no. 7, pp. 1498-1508.

Authors

Seung-Min Back, he received a B.S degree in the Department of electrical engineering from Korea University of Technology and Education, Cheonan, Korea, in 2013. He is currently pursuing a M.S. degree in Electrical, Electronics and Communication Engineering at the Korea University of Technology and Education, Cheonan, Korea. His research interests are in the Big data.

Hyun-Jun Shin, he received a B.S degree in the Department of electrical engineering from Namseoull University, Cheonan, Korea, in 2011, and an M.S. degree in Electrical, Electronics and Communication Engineering at the Korea University of Technology and Education, Cheonan, Korea, in 2011. He is currently pursuing a Ph.D. degree in Electrical, Electronics and Communication Engineering at the Korea University of Technology and Education, Cheonan, Korea. His research interests are in the wireless sensor network, wireless localization, channel coding.

Chang-Heon Oh, he received the B. S. and M.S.E. degrees in telecommunication and information engineering from Korea Aerospace Univ. in 1988 and 1990, respectively. He received the Ph.D. degree in avionics engineering from Korea Aerospace Univ., in 1996. From Feb. 1990 to Aug. 1993, he was with Hanjin Electronics Co. From Oct. 1993 to Feb. 1999, he was with the CDMA R&D center of Samsung Electronics Co. Since Mar. 1999, he has been with the School of Electrical, Electronics and Communication Engineering, Korea University of Technology and Education, where he is currently a professor. His research interests are in the areas of wireless communications, mobile communication, and wireless sensor networks with particular emphasis on wireless localization.