Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 6, June 2021: 1093-1101

A Study on the Environmental Status of Child Care Facilities and Child Care Teachers in Seoul Based on Big Data

Dong-Jin Shin[a], Jeong-Joon Kim[b]

[a]Department of Computer Engineering, Anyang University, South Korea,
djshin@ayum.anyang.ac.kr1
[b]Assistant Professor, Department of ICT Convergence Engineering, Anyang University, South Korea

***Corresponding author:** jjkim@anyang.ac.kr

Abstract

In 2018, the number of babies born fell to the lowest level in 40 years, and according to statistics from the Ministry of Education, the preschool enrollment rate of infants aged 3 to 5 who are subject to kindergarten education is half as of 2016. As of February 2016, Seoul, which has the fiercest competition for admission, had an average competition rate of 4.8 to 1, and the average waiting rate exceeded 62. The competition rate for private companies is 2 to 1 and the waiting list is 19 during the same period. Day care centers are more serious. According to the Ministry of Health and Welfare, the number of people waiting to enter national and public daycare centers as of June last year was about 282,000. Compared to 179,000 children attending daycare centers, the waiting rate is more than 157%. Therefore, in order to understand the relationship between childcare facilities and childcare teachers, this paper aims to analyze the ratio of daycare center quota and childcare teachers to the number of children born annually by dividing into 25 districts in Seoul using big data analysis solutions R and storage and pig.

Keywords: Childcare Centers, Childcare Teacher, Big Data Solution

1. Introduction

According to a study by Yang Mi-sun (2012), Korea is currently showing the lowest birth rate in the world. As one of the countermeasures, the government is strengthening its national responsibility for childbirth and childcare, especially emphasizing the country's role in childcare to reduce the burden of childcare and improve the quality of childcare services.

The government announced a five-year (2006-2010) mid- to long-term childcare plan to increase the number of children using national/public daycare centers to 2,700 by 2010 and to increase the number of children using daycare centers to 30%. Under the plan, the number of national and public daycare centers increased from 1,352 in 2005 to 1,826 in 2008.

Aisarang plan published alongside the Lee Myung-bak government took second long-term child care, however, private day care center rather than expanding public daycare centers and soup peullaenneun's quality and focus on improving the level of public daycare centers and soup. In addition, national/public daycare centers are not expanding as the policy direction shifts due to the expansion of vulnerable areas-oriented national/public daycare centers and the decrease in the number of waiting for national/public daycare centers. Therefore, despite the expansion of national/public daycare centers, 5.3 percent of all daycare centers and 10.8 percent of children use them.

On the other hand, parents' preference for national/public daycare centers still remains, with the total number of waiting at national/public daycare centers exceeding 110,000 as of March 2010, and the average number of waiting children per facility reaching 60 or so, in short supply. In particular, the congestion phenomenon in the Seoul metropolitan area is very serious as the waiting ratio in Seoul and Gyeonggi Province accounts for 92.9% of the total waiting population.

This problem remains an unresolved task today. Therefore, the number of daycare centers in 25 districts in Seoul is surveyed and data is processed and refined to meet the needs of children there, and analysis and visualization is conducted with two focus on whether the number of teachers and the ratio of children is correct.

Starting with the introduction, this paper explores the replication techniques and EC techniques used in HDFS, and the performance degradation factors that occur in EC-based HDFS through two related studies. Chapter 3 introduces solutions to disk write throughput problems that arise during encoding. Chapter 4 compares and analyzes existing EC-based HDFS and systems that apply solutions and concludes this paper with the conclusion of Chapter 5.

2. Related Works

Contrary to concerns that kindergartens will close down due to the continued decline in the number of births, Lee (29), who has 34-month-old children living in Seoul, is surprised by the government's announcement. Lee sent the child to a daycare center after fierce competition this year. At the end of 2016, she applied for a waiting list at a public daycare center she liked, but received a waiting number of 100. The home daycare center, which was put in the same year, received a waiting number of eight, but was contacted only in February this year.

The same is true of kindergartens. Lee applied for admission to kindergarten in advance last year due to the competition for admission to daycare centers. Of the three applications, only two received 200 single numbers and one double-digit waiting number. Lee said, "As soon as a child is born, he has to apply for admission to the state/public service."It was true," he said. "I'm worried that private kindergartens will also draw waiting numbers by lottery [2].

The election pledge of June 13, 2018 is also aimed at significantly expanding national and public daycare centers to solve this problem. This is not the only problem. According to a study by Yoon Ki-seol (2014), the working hours of Korean workers are 2090 hours per year as of 2011, the second-largest among OECD countries. As many parents work longer hours than 40 hours a week and 12 hours overtime, the increase in the number of hours their children spend in childcare facilities has been attributed to the difficulties of childcare teachers.

Byun Eun-ji's (2015) study[4] talked about the difficulty of longer child care hours because teachers at work daycare centers have to take care of their children earlier and later than parents' commuting hours.

3. Related Techniques

3.1. R-based statistics

As part of an open source project, R is a computer programming language that provides a variety of statistical analysis and very good programming capabilities, as well as excellent and diverse graphical methods that allow users to write, expand, and add new functions. In addition, it has an excellent help system, and a large number of statistical functions make it easy for users to use and apply. R can be used by downloading packages and others using the Internet. It provides new algorithms such as analysis, graphics, I/O and handling for data analysis, and a variety of libraries, and supports excellent graphical functions for data analysis and visualization and In-Memory computing technology for fast processing speed.[5][6].

3.2. Big Data

Big data refers to structured and unstructured data that is so vast that it is difficult to collect, store, and analyze with existing methods or tools. It generates 2 million searches on Google, 72 hours of video on YouTube, and 270,000 tweets on Twitter. McKinsey, a global consulting firm, notes that big data is subjective and will continue to change in the future as it exceeds the capabilities of existing database management tools. In some groups, big data is defined as terabytes or more of data and as an architecture that processes large amounts of data [7].

3.3. Hive

Hive uses SQL-like query language as a Data Warehouse system for Hadoop that can perform data summarization, queries, and analysis using HiveQL (Hive Query Language). Hive can be used to create batch tasks that can review or reuse data in two directions, and Hive is designed for batch-based processing [8].

3.4. Hadoop

Hadoop is an open-source framework for processing large data analysis as a representative big data technology. This is an implementation of Google's Google Distributed File System (GFS) and MapReduce, which were announced in 2003 and 2004, similar to Google's in many ways. It consists of a Hadoop Distributed File System (HDFS) that distributes, stores, and manages large amounts of data, and MapReduce that performs analysis of large amounts of data [8].

The Hadoop Distributed File System (HDFS) is installed and operated on a number of Linux servers, and with excellent scalability, it can store more than petabytes of data. In addition, multiple servers simultaneously distribute data to ensure high speed in large-scale data processing. In particular, using Linux equipment is cheaper to build systems than relational database management systems (RDBMS) that use expensive equipment. Problems such as data loss and failure to store due to low-cost equipment use are solved by distributing the data to various nodes (servers).

The Hadoop Distributed File System (HDFS) is a distributed file system for distributing, storing, and managing large amounts of data that are stored on distributed servers divided into blocks. The block size is set to 64MB by default, and if the data does not fall apart by 64MB, the block is divided and the remaining parts are stored as the same size. Block size can also be changed by setting as many as the user wants.

The Hadoop distributed file system consists primarily of one NameNode and multiple DataNodes. The namespace manages all metadata (a tree-like namespace for directory names, file names, file blocks, etc.) in the Hadoop distributed file system. A datanode is a data server that stores data divided into blocks and manages data input and output requests from both the name node and the client.

3.5. Pig

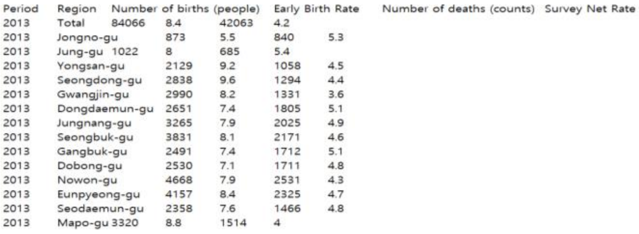
Pig is a textual language that originated in Yahoo, where the current version is 0.17.0, and the last version was released three years ago in 2020. Although there has been no update since the last version to date, Pig Latin makes it easy to process data with features that take considerable time to implement using Map-Reduce. For example, implementing general-purpose processing of Sort, Join, and Filter as Map-Reduce takes a considerable amount of time, but Pig Latin can only be written and processed simply with a script [9].

4. Processand Analysis

4.1. Store and Refine Data

The data was collected through the Seoul Open Data Plaza portal site. Data for the number of births by region is in CSV format, and data for child care teachers and daycare centers by region is in TXT format. Figure 1 shows an unrefined data source downloaded through an open data square portal site.

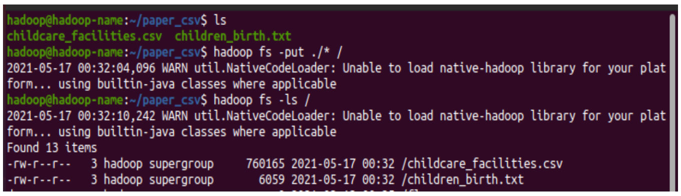
The data in Figure 1 consist of columns of duration, region, number of births (people), premature birth rate, death rate (people), and survey net rate, and the rest of the columns use Hive solutions to refine the data. Figure 2 illustrates a portion of an unrefined data source downloaded through an open data square portal site.



```
Period   Region    Number of births (people)   Early Birth Rate    Number of deaths (counts)  Survey Net Rate
2013     Total     84066    8.4      42063     4.2
2013     Jongno-gu          873      5.5       840       5.3
2013     Jung-gu   1022     8        685       5.4
2013     Yongsan-gu         2129     9.2       1058      4.5
2013     Seongdong-gu       2838     9.6       1294      4.4
2013     Gwangjin-gu        2990     8.2       1331      3.6
2013     Dongdaemun-gu      2651     7.4       1805      5.1
2013     Jungnang-gu        3265     7.9       2025      4.9
2013     Seongbuk-gu        3831     8.1       2171      4.6
2013     Gangbuk-gu         2491     7.4       1712      5.1
2013     Dobong-gu          2530     7.1       1711      4.8
2013     Nowon-gu           4668     7.9       2531      4.3
2013     Eunpyeong-gu       4157     8.4       2325      4.7
2013     Seodaemun-gu       2358     7.6       1466      4.8
2013     Mapo-gu 3320       8.8      1514      4
```

Fig 1. Original data for number of births by region

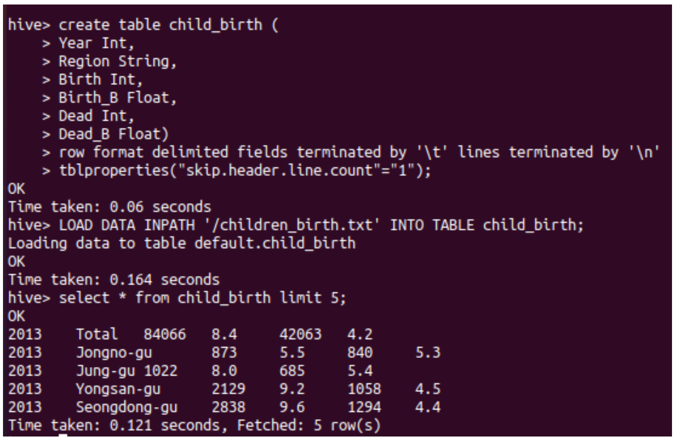
The data in Figure 2 consists of 26 columns, including city and county life, daycare center code, daycare center type, operation status, detailed address, phone number, fax number, homepage address, and the rest of the columns use Hive solution to refine the data. Figure 3 shows that the data in Figure 1 and Figure 2 above are stored and verified locally as an HDFS file system.



```
hadoop@hadoop-name:~/paper_csv$ ls
childcare_facilities.csv  children_birth.txt
hadoop@hadoop-name:~/paper_csv$ hadoop fs -put ./* /
2021-05-17 00:32:04,096 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your plat
form... using builtin-java classes where applicable
hadoop@hadoop-name:~/paper_csv$ hadoop fs -ls /
2021-05-17 00:32:10,242 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your plat
form... using builtin-java classes where applicable
Found 13 items
-rw-r--r--   3 hadoop supergroup     760165 2021-05-17 00:32 /childcare_facilities.csv
-rw-r--r--   3 hadoop supergroup       6059 2021-05-17 00:32 /children_birth.txt
```

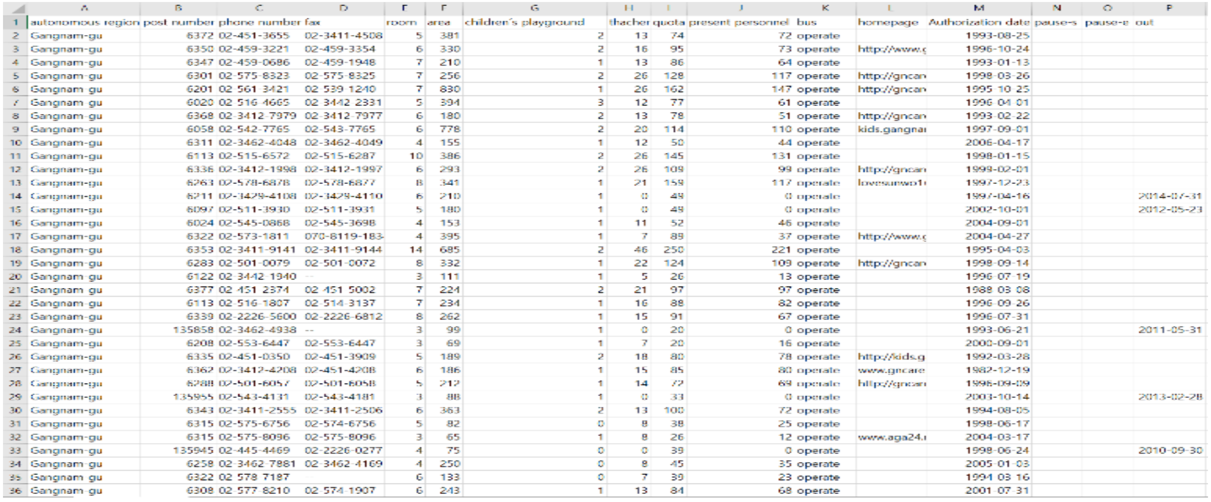
Fig 3. Save local files to Hadoop

If you check the list of files to save using the ls command, you can check childrencare_facilities.csv and children_birth.txt, which you can save using the hadoopfs –put command. Figure 4 shows the creation of a table in Hive, loading files stored in HDFS, and entering data.



```
hive> create table child_birth (
    > Year Int,
    > Region String,
    > Birth Int,
    > Birth_B Float,
    > Dead Int,
    > Dead_B Float)
    > row format delimited fields terminated by '\t' lines terminated by '\n'
    > tblproperties("skip.header.line.count"="1");
OK
Time taken: 0.06 seconds
hive> LOAD DATA INPATH '/children_birth.txt' INTO TABLE child_birth;
Loading data to table default.child_birth
OK
Time taken: 0.164 seconds
hive> select * from child_birth limit 5;
OK
2013    Total   84066   8.4     42063   4.2
2013    Jongno-gu       873     5.5     840     5.3
2013    Jung-gu 1022    8.0     685     5.4
2013    Yongsan-gu      2129    9.2     1058    4.5
2013    Seongdong-gu    2838    9.6     1294    4.4
Time taken: 0.121 seconds, Fetched: 5 row(s)
```

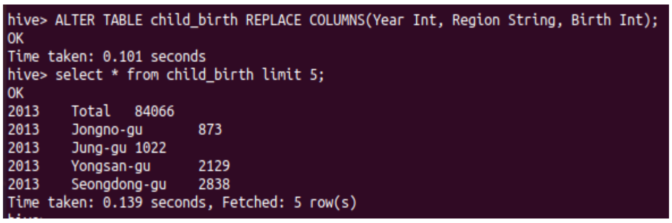
Fig 4. Confirm through data input after creating a table



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	autonomous region	post number	phone number	fax	room	area	children's playground	thacher	quota	present personnel	bus	homepage	Authorization date	pause-s	pause-e	out
2	Gangnam-gu	6372	02-451-3655	02-3411-4508	5	381	2	13	74	72	operate		1993-08-25			
3	Gangnam-gu	6350	02-459-3221	02-459-3354	6	330	2	16	95	73	operate	http://www.	1996-10-24			
4	Gangnam-gu	6347	02-459-0686	02-459-1948	7	210	1	13	86	64	operate		1993-01-13			
5	Gangnam-gu	6301	02-575-8323	02-575-8325	7	256	2	26	128	117	operate	http://gncan	1998-03-26			
6	Gangnam-gu	6201	02-561-3421	02-539-1240	7	830	1	26	162	147	operate	http://gncan	1995-10-25			
7	Gangnam-gu	6020	02-516-4665	02-3442-2331	5	394	3	12	77	61	operate		1996-04-01			
8	Gangnam-gu	6368	02-3412-7979	02-3412-7977	6	180	2	13	78	51	operate	http://gncan	1993-02-22			
9	Gangnam-gu	6058	02-542-7765	02-543-7765	6	778	2	20	114	110	operate	kids.gangnai	1997-09-01			
10	Gangnam-gu	6311	02-3462-4048	02-3462-4049	4	155	1	12	50	44	operate		2006-04-17			
11	Gangnam-gu	6113	02-515-6572	02-515-6287	10	386	2	26	145	131	operate		1998-01-15			
12	Gangnam-gu	6336	02-3412-1998	02-3412-1997	6	293	2	26	109	99	operate	http://gncan	1999-02-01			
13	Gangnam-gu	6263	02-578-6878	02-578-6877	8	341	1	21	159	117	operate	lovesunwo1	1997-12-23			
14	Gangnam-gu	6211	02-3429-4108	02-3429-4110	6	210	1	0	49	0	operate		1997-04-16			2014-07-31
15	Gangnam-gu	6097	02-511-3930	02-511-3931	5	180	1	0	49	0	operate		2002-10-01			2012-05-23
16	Gangnam-gu	6024	02-545-0868	02-545-3698	4	153	1	11	52	46	operate		2004-09-01			
17	Gangnam-gu	6322	02-573-1811	070-8119-183	4	395	1	7	89	37	operate	http://www.	2004-04-27			
18	Gangnam-gu	6353	02-3411-9141	02-3411-9144	14	685	2	46	250	221	operate		1995-04-03			
19	Gangnam-gu	6283	02-501-0079	02-501-0072	8	332	1	22	124	109	operate	http://gncan	1998-09-14			
20	Gangnam-gu	6122	02-3442-1940	--	3	111	1	5	26	13	operate		1996-07-19			
21	Gangnam-gu	6377	02-451-2374	02-451-5002	7	224	2	21	97	97	operate		1988-03-08			
22	Gangnam-gu	6113	02-516-1807	02-514-3137	7	234	1	16	88	82	operate		1996-09-26			
23	Gangnam-gu	6339	02-2226-5600	02-2226-6812	8	262	1	15	91	67	operate		1996-07-31			
24	Gangnam-gu	135858	02-3462-4938	--	3	99	1	0	20	0	operate		1993-06-21			2011-05-31
25	Gangnam-gu	6208	02-553-6447	02-553-6447	3	69	1	7	20	16	operate		2000-09-01			
26	Gangnam-gu	6335	02-451-0350	02-451-3909	5	189	2	18	80	78	operate	http://kids.g	1992-03-28			
27	Gangnam-gu	6362	02-3412-4208	02-451-4208	6	186	1	15	85	80	operate	www.gncare	1982-12-19			
28	Gangnam-gu	6288	02-501-6057	02-501-6058	5	212	1	14	72	69	operate	http://gncan	1996-09-09			
29	Gangnam-gu	135955	02-543-4131	02-543-4181	3	88	1	0	33	0	operate		2003-10-14			2013-02-28
30	Gangnam-gu	6343	02-3411-2555	02-3411-2506	6	363	2	13	100	72	operate		1994-08-05			
31	Gangnam-gu	6315	02-575-6756	02-574-6756	5	82	0	8	38	25	operate		1998-06-17			
32	Gangnam-gu	6315	02-575-8096	02-575-8096	3	65	1	8	26	12	operate	www.aga24.i	2004-03-17			
33	Gangnam-gu	135945	02-445-4469	02-2226-0277	4	75	0	0	39	0	operate		1998-06-24			2010-09-30
34	Gangnam-gu	6258	02-3462-7881	02-3462-4169	4	250	0	8	45	35	operate		2005-01-03			
35	Gangnam-gu	6322	02-578-7187		6	133	0	7	39	23	operate		1994-03-16			
36	Gangnam-gu	6308	02-577-8210	02-574-1907	6	243	1	13	84	68	operate		2001-07-31			

Fig 2. Original data for number of births by region

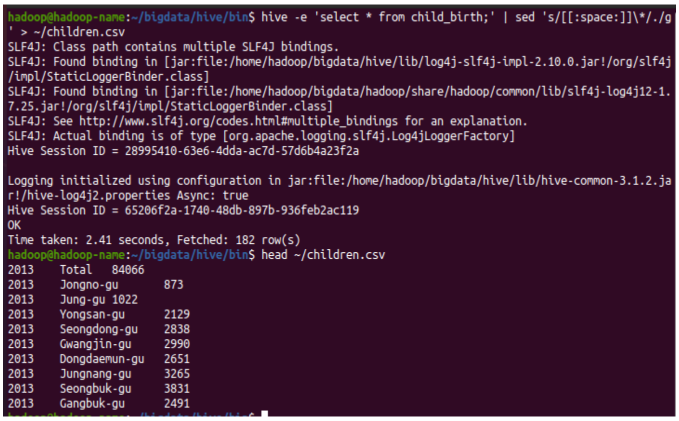
child_birth table Each column name and data consist of an Integer column, Region column is a string, Birth column is a numeric column, Birth_B column is an numeric column, Dead column is an integer, Dead_B column is a space for convenient viewing of data. After application, the data stored in HDFS is loaded through the LOAD DATA INPATH syntax and entered into the child_birth table. Finally, we show the results of checking only the top five data using select query statements.Figure 5 illustrates the re-creation of the table by extracting only the required columns from the child_birth table, period, region, and number of births separately and overwriting the child_birth table.



```
hive> ALTER TABLE child_birth REPLACE COLUMNS(Year Int, Region String, Birth Int);
OK
Time taken: 0.101 seconds
hive> select * from child_birth limit 5;
OK
2013    Total   84066
2013    Jongno-gu       873
2013    Jung-gu 1022
2013    Yongsan-gu      2129
2013    Seongdong-gu    2838
Time taken: 0.139 seconds, Fetched: 5 row(s)
```

Fig 5. Refine and check the required columns of the table

The replace columns option can change the table schema by extracting only the Year, Region, and Birth columns, as shown in Figure 5.

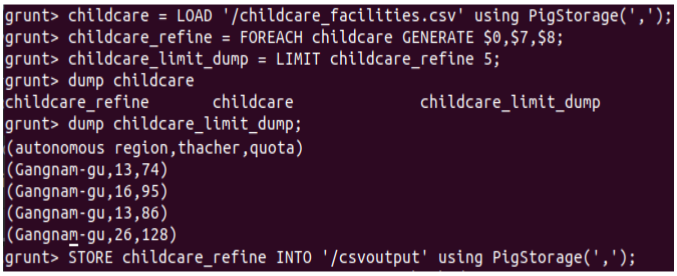


```
hadoop@hadoop-name:~/bigdata/hive/bin$ hive -e 'select * from child_birth;' | sed 's/[[:space:]]\*/./g
' > ~/children.csv
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/bigdata/hive/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j
/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/bigdata/hadoop/share/hadoop/common/lib/slf4j-log4j12-1.
7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = 28995410-63e6-4dda-ac7d-57d6b4a23f2a

Logging initialized using configuration in jar:file:/home/hadoop/bigdata/hive/lib/hive-common-3.1.2.ja
r!/hive-log4j2.properties Async: true
Hive Session ID = 65206f2a-1740-48db-897b-936feb2ac119
OK
Time taken: 2.41 seconds, Fetched: 182 row(s)
hadoop@hadoop-name:~/bigdata/hive/bin$ head ~/children.csv
2013    Total   84066
2013    Jongno-gu       873
2013    Jung-gu 1022
2013    Yongsan-gu      2129
2013    Seongdong-gu    2838
2013    Gwangjin-gu     2990
2013    Dongdaemun-gu   2651
2013    Jungnang-gu     3265
2013    Seongbuk-gu     3831
2013    Gangbuk-gu      2491
```

Fig 6. Converting refined table to CSV format

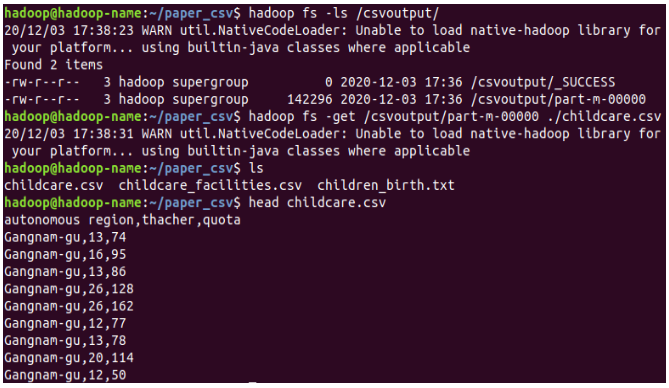
Figure 6 shows a column extraction to convert and store the output of the final refined table into the CSV format of the child.csv file name using the addition of the sed option in the Hive select query statement locally on Linux. Figure 7 shows loading local nursery teacher and garden data using Pig, refining loaded data and storing purified data in CSV format.



```
grunt> childcare = LOAD '/childcare_facilities.csv' using PigStorage(',');
grunt> childcare_refine = FOREACH childcare GENERATE $0,$7,$8;
grunt> childcare_limit_dump = LIMIT childcare_refine 5;
grunt> dump childcare
childcare_refine       childcare               childcare_limit_dump
grunt> dump childcare_limit_dump;
(autonomous region,thacher,quota)
(Gangnam-gu,13,74)
(Gangnam-gu,16,95)
(Gangnam-gu,13,86)
(Gangnam-gu,26,128)
grunt> STORE childcare_refine INTO '/csvoutput' using PigStorage(',');
```

Fig 7. Refine and format conversion after loading data

Using the LOADUSING option in the childcare variable, the childcare_facilities.csv file stored under the HDFS root is imported separately by comma (","). In childcare, the city and county names are $0 because it is the 0th column, the 7th column because it is the 7th column, and the garden is $8 because it is the 8th column, and the FOREACH GENERATE option is used to store them in the childcare_refine variable. After saving, only the top five data is outputted using the LIMIT option to verify that the data stored in the childcare_refine variable is extracted, and the STORE USING option is used to store the data separately in a comma-like directory under the HDFS root.



```
hadoop@hadoop-name:~/paper_csv$ hadoop fs -ls /csvoutput/
20/12/03 17:38:23 WARN util.NativeCodeLoader: Unable to load native-hadoop library for
 your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--   3 hadoop supergroup          0 2020-12-03 17:36 /csvoutput/_SUCCESS
-rw-r--r--   3 hadoop supergroup     142296 2020-12-03 17:36 /csvoutput/part-m-00000
hadoop@hadoop-name:~/paper_csv$ hadoop fs -get /csvoutput/part-m-00000 ./childcare.csv
20/12/03 17:38:31 WARN util.NativeCodeLoader: Unable to load native-hadoop library for
 your platform... using builtin-java classes where applicable
hadoop@hadoop-name:~/paper_csv$ ls
childcare.csv  childcare_facilities.csv  children_birth.txt
hadoop@hadoop-name:~/paper_csv$ head childcare.csv
autonomous region,thacher,quota
Gangnam-gu,13,74
Gangnam-gu,16,95
Gangnam-gu,13,86
Gangnam-gu,26,128
Gangnam-gu,26,162
Gangnam-gu,12,77
Gangnam-gu,13,78
Gangnam-gu,20,114
Gangnam-gu,12,50
```

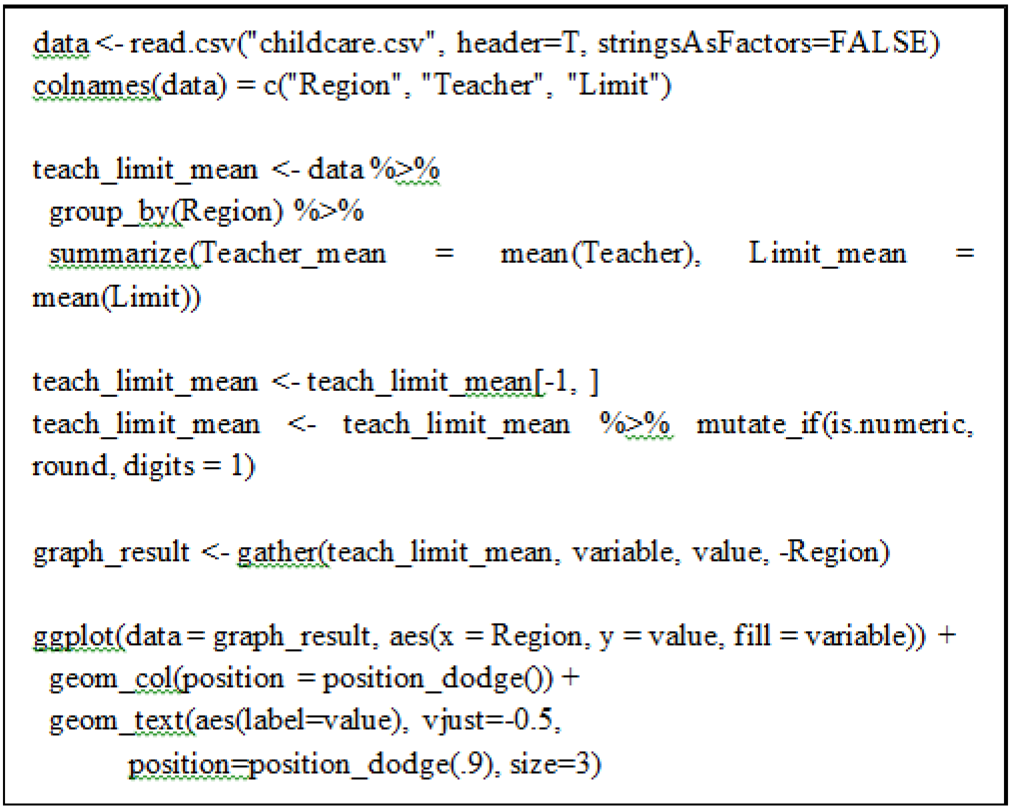
Fig 8. Converting refined data to CSV format

Figure 8 shows the final refined data stored in the /csvoutput directory of HDFS being re-imported locally on Linux, transforming the form into CSV format. When importing, the Hadoop-get command is used, and the top 10 data is checked locally through the head command to confirm that the data has been refined normally and that the conversion has been completed in CSV format.

4.2. Data analytics and visualization.

The directory where the final refined data is located is designated by R Studio to select the working directory, and an average analysis of nursery teachers and daycare center gardens is performed.

Figure 9 shows a source code that uses childcare.csv among the final refined data to calculate and visualize the average number of nursery teachers and daycare centers by city and county. The data variable is loaded with a childcare.csv file using the read.csv function, which then adds the StringAsFactors option to prevent the data from being categorized automatically by R by giving the header option True.



```
data <- read.csv("childcare.csv", header=T, stringsAsFactors=FALSE)
colnames(data) = c("Region", "Teacher", "Limit")

teach_limit_mean <- data %>%
  group_by(Region) %>%
  summarize(Teacher_mean     =     mean(Teacher),     Limit_mean     =
mean(Limit))

teach_limit_mean <- teach_limit_mean[-1, ]
teach_limit_mean   <-   teach_limit_mean   %>%   mutate_if(is.numeric,
round, digits = 1)

graph_result <- gather(teach_limit_mean, variable, value, -Region)

ggplot(data = graph_result, aes(x = Region, y = value, fill = variable)) +
  geom_col(position = position_dodge()) +
  geom_text(aes(label=value), vjust=-0.5,
          position=position_dodge(.9), size=3)
```

Fig 9. Source code for average analysis of the number of childcare teachers and daycare centers by city, county and district

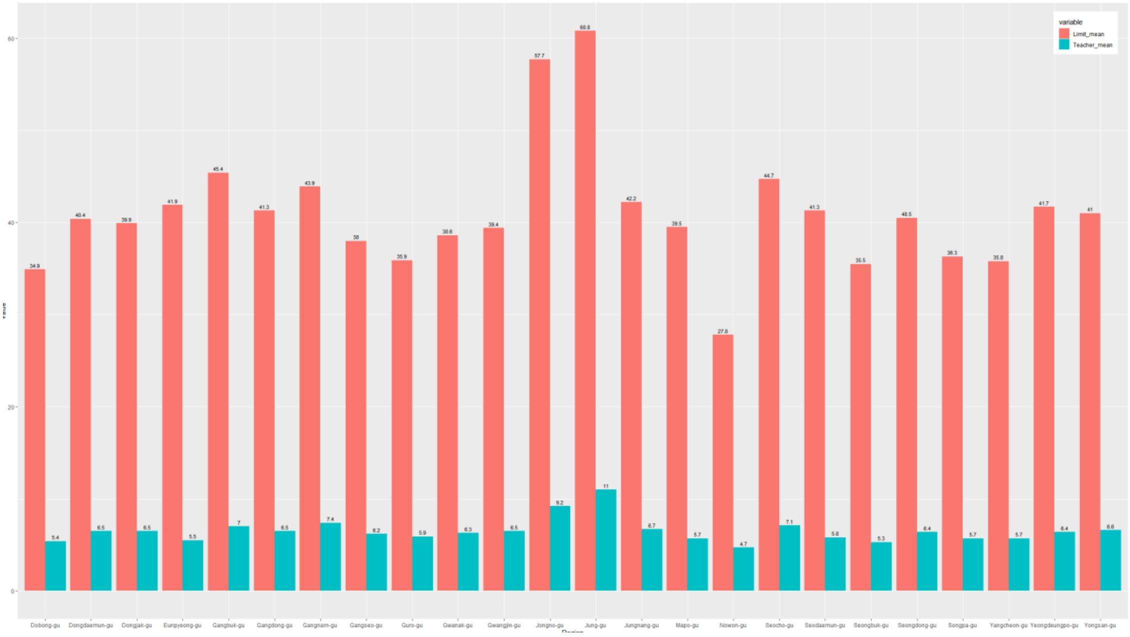
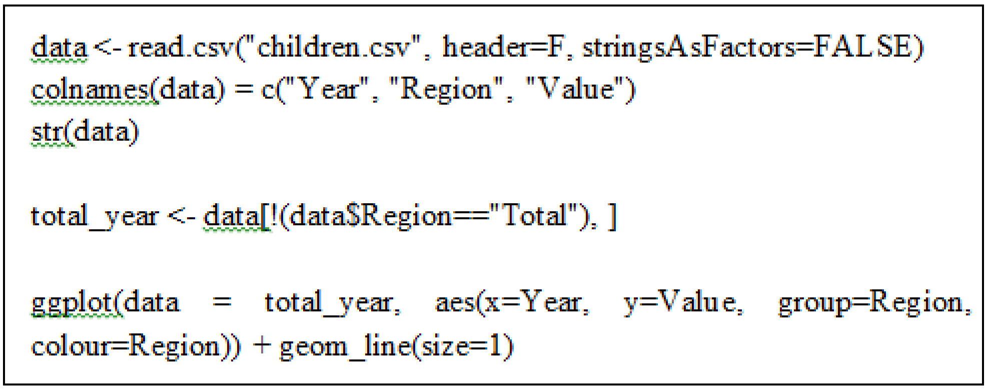


Fig 10. Visualization for average analysis of the number of childcare teachers and daycare centers by city, county and distric

Using the magrittr package in the teach_limit_mean variable, group Regoin columns to calculate the average of the Teacher and Limit columns. After the change, the ggplot function of the ggplot2 package is used to express the average number of nursery teachers and daycare centers calculated above on the x-axis and the y-axis by city and district. The visualization results expressed are shown in Figure 10.

If you check by city and county, the average number of daycare centers is higher than the average of daycare teachers because childcare teachers are not 1:1 when they are in charge of children, but if you check the ratio, you can see that about one childcare teacher manages six to seven children. It can be confirmed that the number of childcare teachers is slightly short due to the number of children recommended through existing policies.



```
data <- read.csv("children.csv", header=F, stringsAsFactors=FALSE)
colnames(data) = c("Year", "Region", "Value")
str(data)

total_year <- data[!(data$Region=="Total"), ]

ggplot(data = total_year, aes(x=Year, y=Value, group=Region,
colour=Region)) + geom_line(size=1)
```

Fig 11. Source code for analyzing the number of births by city, county, district

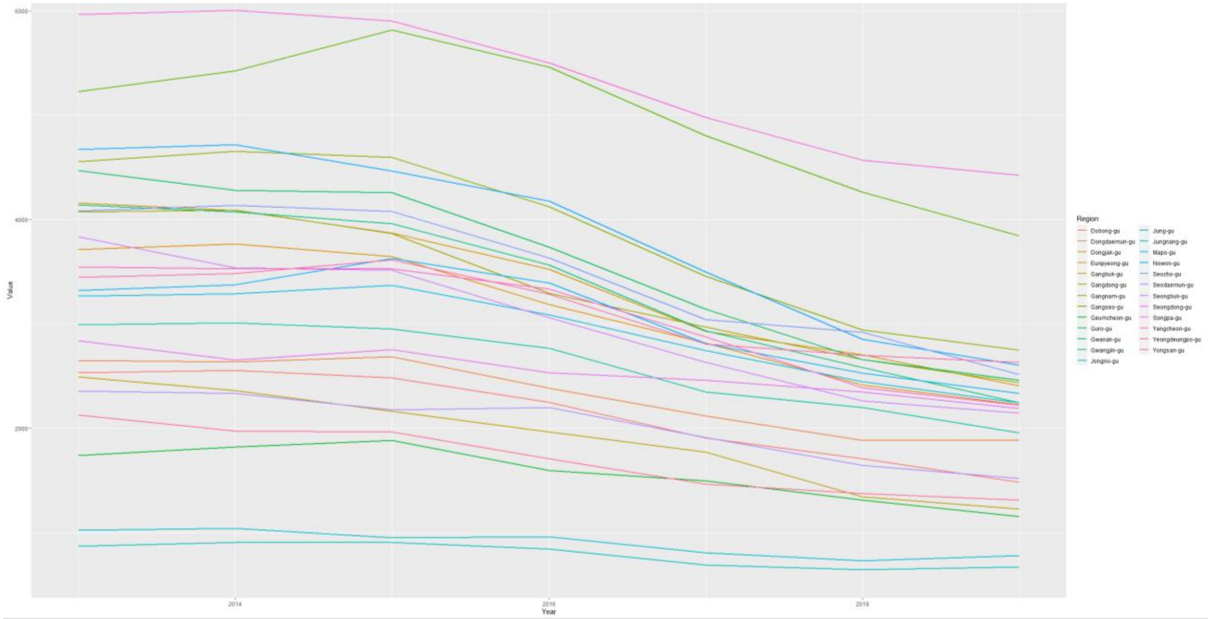


Fig 12. Visualization of analysis of the number of births by city, county and district

Figure 11 shows the source code for comparing the number of births in each city and county by year using children.csv among the final refined data. The data variable uses the read.csv function to load the children.csv file, and when the header option is given to FALSE to avoid maintaining the header of the column, the string adds the stringAsFactors option because R automatically categorizes the type of Factor data. In order to check the total_year variable for data by city and county, the sum of the Region column is excluded and stored, and the ggplot function of the ggplot2 package is used to express Year on the x-axis and Value on the y-axis by city and county. The visualization results expressed are shown in Figure 12.

If you check the number of babies born according to the year by city and district, you can see that the overall number of babies decreases definitely over the year. As of 2013, Songpa-gu recorded about 5,900 births in 2013, but it is decreasing every year, and Gangseo-gu recorded about 5,200 births in 2013, but it can be seen that it has decreased sharply since 2015. As of 2013, Jongno-gu has the smallest number of babies born, with about 800 people decreasing and increasing little by little every year, but it is confirmed that there is little change. In addition, the second is Jung-gu District, which had about 1,000 people in 2013, has been decreasing very little over the years, and has increased slightly in 2018.

5. Conclusion

This work stores, processes and purifies data using big data storage and processing solutions, Hadoop and Hive and Pig, and simultaneously analyzes and visualizations through analysis solutions, R programming. On average, about five children are taken care of per nursery teacher, but the results of the study show that about seven children are taken care of. There is no difference between five and seven, but considering the suddenness of a child, the difference between two people can act as a very large difference. Therefore, the birth rate is gradually decreasing, but the proportion of children in charge of each childcare teacher is high, so treatment improvements such as wage improvement and hiring more childcare teachers should be made so that childcare teachers can be satisfied.

References

[1] B.G. Chun, F. Dabek, A. Haeberlen, E. Sit, H. Weatherspoon, M.F. Kaashoek, J. Kubiatowicz, R. Morris (2006) Efficient Replica Maintenance for Distributed Storage Systems. NSDI '06 Proceedings of the 3rd conference on Networked Systems Design & Implementation 6:45-58

[2] J. Li, B. Li (2013) Erasure coding for cloud storage systems: A survey. Tsinghua Science and Technology 18(3):259-272

[3] J.D Cook, R. Primmer, A. de Kwant (2014) Compare Cost and Performance of Replication and Erasure Coding. Hitachi Review 63:304-310

[4] D.O. Kim, H.Y. Kim, Y.K. Kim, J.J Kim (2019) Cost analysis of erasure coding for exa-scale storage. The Journal of Super Computing 75(8):4638-4656

[5] D. Sun, Y. Xu, Y. Li, S. Wu, C. Tian (2016) Efficient Parity Update for Scaling RAID-like Storage Systems Networking. Architecture and Storage (NAS), 2016's IEEE International Conference 1-10

[6] L.J. Mohan, R.L. Harold, P.I.S. Caneleo, U. Parampalli, A. Harwood (2015) Benchmarking the Performance of Hadoop Triple Replication and Erasure Coding on A Nation-Wide Distributed Cloud. Network Coding (NetCod), 2015's International Symposium 61-65

[7] Dong-Jin Shin, Kwang-Jin Kwak, Seung-Yeon Hwang, Jeong-Min Park, Jeong-Joon Kim (2018) Efficient Storage Structure Research in Hadoop Distributed Storage System. 2018's IPACT Conference 56-57

[8] Dong-Jin Shin, Seung-Yeon Hwang, Kwang-Jin Kwak, Kyoung-Won Park, Jeong-Min Park, Jeong-Joon Kim (2019) A Study on the Recovery Techniques of Distributed File System in a Big Data Environment. 2019's IIBC Conference 83-86

[9] K. Nansai, X. Chen, S. Chen, J. Zang (2019) HDFS Erasure Coding in Production. CLOUDERA Blog