Summer 2019

ADL Data Capturing System: A Big Data Approach

Junteng Li

ADL Data Capturing System: A Big Data Approach

By

Junteng Li

A technical report submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Major: Computer Science

Program of Study Committee:

Carl K. Chang, Major Professor

Ying Cai

Wensheng Zhang

The student author, whose presentation of the scholarship herein was approved by the program of study committee, is solely responsible for the content of this report. The Graduate College will ensure this report is globally accessible and will not permit alterations after a degree is conferred.

Iowa State University

Ames, Iowa

July 2019

TABLE OF CONTENT

LIST OF FIGURES

ACKNOWLEDGMENTS

I want to thank my major professor my committee chair Dr. Carl K. Chang, and my committee members Professor Ying Cai and Professor Wensheng Zhang for their guidance and support throughout the course of this research.

In addition, I would like to thank Dr. Yunfei Feng and Ph.D. Candidate Jiaxuan Wu for providing their expertise that greatly assisted this work. I want to thank everyone else in the Smart Home Lab and those from the computer science department who also were willing to help and provide suggestions.

Finally, I want to thank my parents for their consistent encouragement and support in my life.

ABSTRACT

Today big data is everywhere and is helping people to improve their day to day lives. It shows you when your uber driver will arrive and tells you the probability of precipitation on next Friday. In the field of improving health condition among older adults, big data helps to early detect and recognize diseases. Enormous amount of ADL data is needed for the practices of machine learning and data mining to create intelligent automatic systems. This work is aimed to implement a prototype of a high-throughput, high-availability distributed data collecting system that can store large amount of data. Chapters 1 & 2 both talk about the background and related work in this field. Chapters 3 & 4 will talk about the details of the data capturing system. In Chapter 5, we will summary the system configurations and show some examples of analysis of the collected data.

CHAPTER 1. INTRODUCTION

ADL is an abbreviation for "Activities of Daily Living," a term that refers to people's daily routine activities that can be completed by the persons themselves. The basic ADLs include bathing and showering, personal hygiene and grooming, dressing, toilet hygiene, functional mobility, and self-feeding.

Jefferson and colleagues' work [4] shows that there are no differences of ADL performance between individuals with mid-to-early stage of cognitive impairment and those have no cognition issue. Therefore, if a difference is found, the person might have severe impairment.

Thomas, Salazar, Rockwood and McDowell [8] ADL impairment can be predictive for future cognitive impairment and dementia. If the ADL data of the patient is available, researchers and experts can predict and make early decision to better improve their living conditions

Nowadays, older adults tend to live independently or with spouse. In order to provide support to elderly people, many smart assisted living systems have been developed. Sensors and automated devices are deployed in a smart home to capture residents' ADL data. These data are used to determine the resident's living status and can help to analyze the health condition of the resident. Health care experts can give suggestions and recommendations based on that data to improve the resident's living quality.

The motivation of this work is to implement a prototype of big data based architecture of activities of daily living (ADLs) capturing system, to collect ADL data for clinical and elderly care use. Usually, such ADL kind of data are collected in healthcare center manually via observation by the nurse or caregiver or from direct report of the resident. And then the data are digitized by office stuff from the notes or sheets. There are many kinds of information on the checklist and therefore human errors cannot be avoided during the process of manually transferring information. Also, the effort of monitoring residents all the time is a huge workload. And repeatedly asking similar generic sets of questions will burden the resident or the patient and affect their emotional and health status [10].

The proposed ADL capturing system will record the user's (the resident who needs to be monitored) ADL information as well as motion, environmental, geolocation etc. information through multiple modules on a smartphone carried by the user. The project reduces the cost of hardware setup and human labor forces, so that both care recipient and caregiver will benefit from the system eventually.

CHAPTER 2. RELATED WORK

This section briefly introduces some work done by other groups related to capturing ADL data. They have different procedures to collect ADL data, including web application, sensors installed in room and cloud service.

Longqi Yang, et al. [10] Introduce an ADL capturing approach called YADL (Your Activities of Daily Living) that uses images of ADLs and personalization to improve the efficiency of the collection of ADL data and the patient experience. They designed a web application which is accessible from a wide range of devices, it shows the user (patient/resident) different images of ADLs and asks them to answer how difficult it was for the user to perform the activity. Therefore, human workload of manually recording ADL have been reduced and patient experience have been improved.

Instead of letting the patient do an online interactive survey, Kazunari Tamamizu, et al. ‘s work [7] reduced human labor work in another way. In their paper "Capturing activities of daily living for elderly at home based on environment change and speech dialog”, they propose a system that captures activities of daily living of the elderly based on speech dialogue triggered by environment changes. They deploy autonomous sensor boxes to collect and upload house environmental data to the cloud. When the system detects a change in the house environment, the virtual agent of the system will ask the resident what he/she is doing now. Therefore, the resident does not need to actively fill out the survey. The system will try to collect the data when ADL events are seemed to take place.

The ability to perform ADLs is dependent upon cognitive as well as motor abilities [5]. Motion data of the resident are therefore important regarding recognize and evaluate the resident health condition. Xiaoye Xia [9] in the paper "*Body Motion Capture Using Multiple Inertial Sensors*" shows that they proposed an inertial motion capturing system consisting of a central control host and ten sensor nodes, for capturing human body movements. The sensors record both angular and acceleration data to determine whether the resident is falling over or near falling. This work shows that the motion data of the resident is important since it can help diagnose fall-related diseases and help alert both caregiver and the care recipient.

This report is mainly motivated by Yunfei Feng and Carl K. Chang's work. "Engaging Mobile Data to Improve Human Well-being: the ADL Recognition Approach" (2017) [1]. They performed an ADL recognition experiment in a smart home environment, using their ADL recognition service deployed over the cloud. The data they collected are from the mobile device and they achieved overall 90% correct recognition rates. Their work includes ADL recognition and prediction which is beyond this report scope. This work is more focus on implementing a prototype that can collect and storage ADL data in a way that can handle massive data transactions.

CHAPTER 3. THE ADL CAPTURING SYSTEM

This section presents the overall structure of the ADL Capturing System. It contains mainly two parts. The first part is the Data Collection Device which is responsible for collecting ADL and other data. The second part is the Data Processing and Storage module. It is used to convert raw data into a more general form and then finalize persistently store the data. Researchers from all over the world will be able to access the collected data and make good use of it.

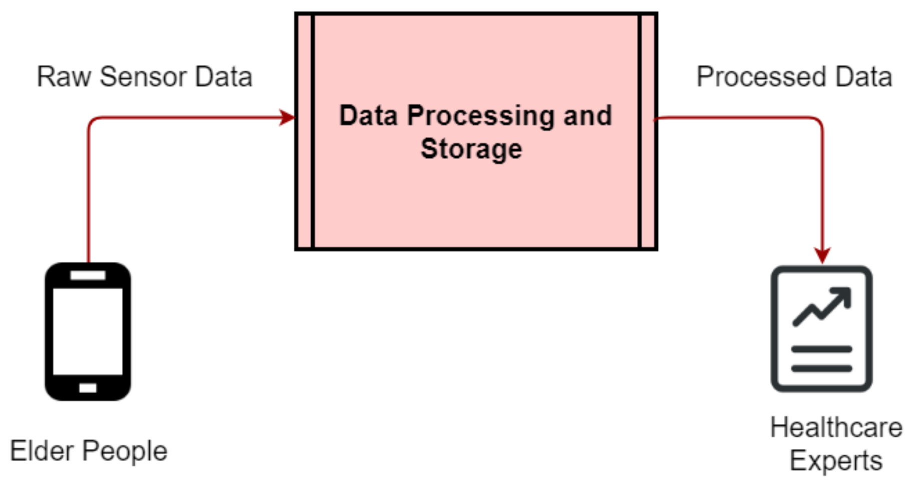

Figure 1. The overall structure of the ADL Capturing System

Figure 1 shows the basic structure of the data capturing system. The data recorded on the mobile phones are uploaded automatically to the processing and storage module. Then the healthcare experts or other study groups can pull the data and make use of it.

3.1 DATA COLLECTION

In Dr. Yunfei Feng and Dr. Carl Chang's work [2], they developed an ADL Recorder App which can collects a variety of data through the embedded hardware on the smartphone, includes the microphone, Wi-Fi scan module, motion and geolocation sensors. The data recorded from different sensors on the mobile phone including light strength, geolocation of the phone, nearby weather, battery and signal information of the device, accelerator, etc. (See Figure 2)

Figure 2. ADL Recorder App collects a variety of data through the embedded hardware

Global Positioning System (GPS) is a very popular tool to retrieve location. However, when the device is indoor, it's very hard to get the precise position because the signal is too weak. With the widely distributed indoor Wi-Fi access points, and Wi-Fi Positioning System (WPS), the app combines Wi-Fi-based Received Signal Strength Indicator (RSSI) and GPS information to compute the indoor location where ADL events take place.

Motion and Environment information are collected through the embedded sensors on the mobile phone. Potentially, it can collect acceleration, gravity, rate of rotation, step counter, air and device temperature, light, pressure, humidity etc. However, due to the availability of the installed hardware sensors as well as software-based sensors, some data may not be able to be collected from a certain model of phone.

Audio information is recorded through the microphone and could be further used to ADL recognition.

The ADL Recorder App collects the data periodically when environment changes are detected. For example, when the user moves from one place to another, the geolocation will detect this change and trigger all sensors and other data collection units to record. Then a new set of data will be sent to the Data Processing and Storage module.

-0.34	9.83	2		238	0	0	42.02815	-93.6497	261.3	EatingByPC	20190617_
-0.34	9.79	0	0	0	-0.02	0.01	-0.02	9			
IASTATE	1.51E+09	0	622								
lat:42.03}	weather:[	main:"Cle	descriptio	icon:"01d'	base:"stat	main:{"ter	pressure:1	humidity:5	temp_min	temp_max:3	visibility:1
D2:5F:E1											
1	0	1	0	365	4313	2	0.99	1	0	1	

Figure 3. snippet of a single raw data record collected from ADL Record App

From Figure 3 we can see that different types of data including ADL data are collected. “EatingByPC” here is the ADL.

3.2 DATA STREAMING AND PROCESSING

In order to achieve the goal and handle massive data transactions, Apache Kafka was introduced into the system. It is a software responsible for the real-time streams of the ADL data and it converts and puts the data into Hadoop. Compared to the traditional message systems like RabbitMQ, Kafka has better performance (i.e. larger throughput, quicker response time), and is also native to Hadoop and therefore works very well in the distributed system.

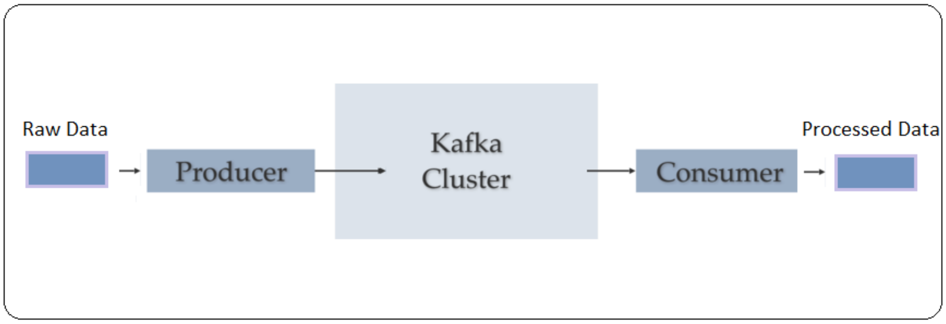

Figure 4. Kafka Architecture

Figure 4 shows the architecture of Kafka. The Kafka Producer receives the data and forwards it to the Kafka Cluster. The Kafka Consumer pulls and analyzes the data from the Cluster. Kafka decouples the input/output of massive data transactions therefore achieved high performance.

In this ADL capturing system, a micro service was created using Kafka to handle data streaming and processing. The Kafka Producer continues listening to the network port where mobile devices will upload the ADL data.

latitude	longitude	altitude	action	soundFileS	currentDat	azimuth	pitch	roll	tmTemper
42.02815	-93.6497	261.3	EatingByP(	20190617_153555		1.9	1.9	-1	0

Figure 5. snippet of a single processed record after file converting

After uploading raw data to the Data Processing and Storage module, raw data will be first converted to a more general form like in Figure 5.

then the data will be stored into the non-relational distributed database, Hadoop Database (HBase) on top of Hadoop file system and will talked about in the next chapter.

CHAPTER 4. THE BIG-DATA-BASED DATA STORAGE SOLUTION

This section presents our big-data-based structure of the system storage module. This work is aimed mainly to provide a solution to conquer two problems:

1) How to avoid break down when massive transactions happen at the same time.

2) How to analysis the huge data when no single PC alone can store it.

These big data issues happen in almost every field of research; therefore, this system is designed to confront the big data challenge and adopt the reality that big data is important in ADL recognition. It helps researchers discover and understand the patient's behavior patterns, predict for future diseases, and alert caregivers when emergencies happen.

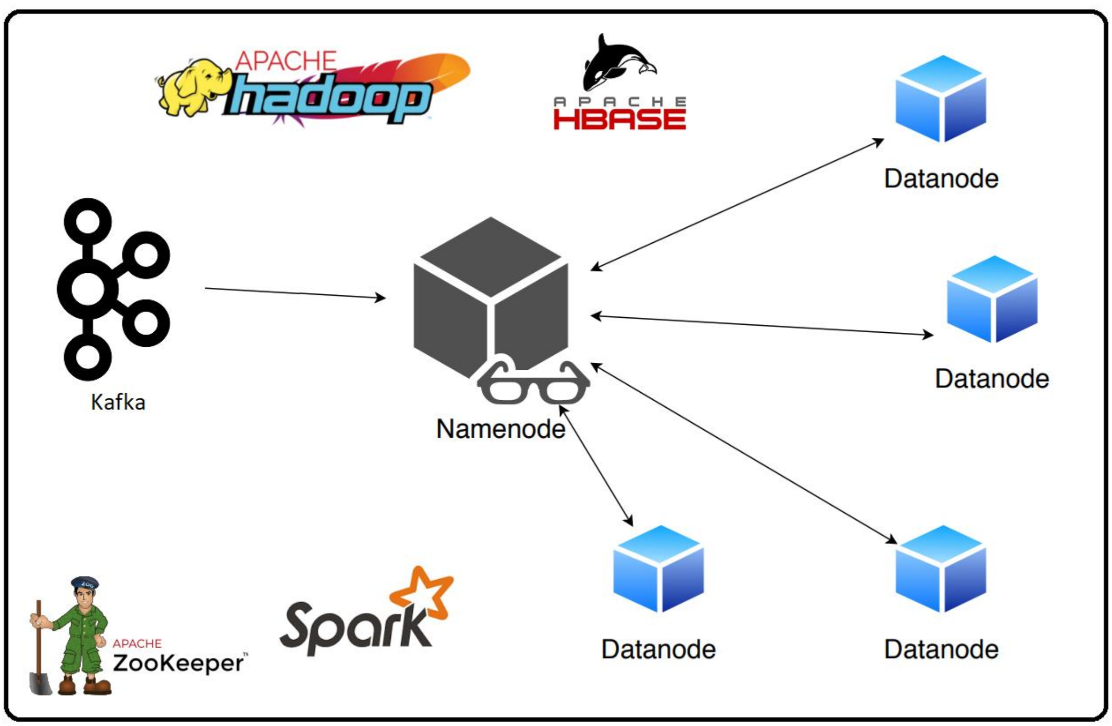

Figure 6. Structure of Data Storage Module

Figure 6 shows the structure of the data storage module of the system. The collected data will be first uploaded through Apache Kafka, then the data will be stored in Apache HBase deployed on the Hadoop distributed file system (HDFS). Apache ZooKeeper is responsible for service synchronization and configuration management in the distributed file system. Apache Spark is the computation engine providing the system the ability to analyze the potential huge among of data.

4.1 HADOOP AND HADOOP DISTRIBUTED FILE SYSTEM (HDFS)

In order to support massive ADL data generation (uploaded from mobile devices) and processing, the Apache Hadoop framework was introduced into the capturing system for his study. The version 2.0 of Hadoop has been deployed therefore other computation engines can be uses such as Apache Spark.

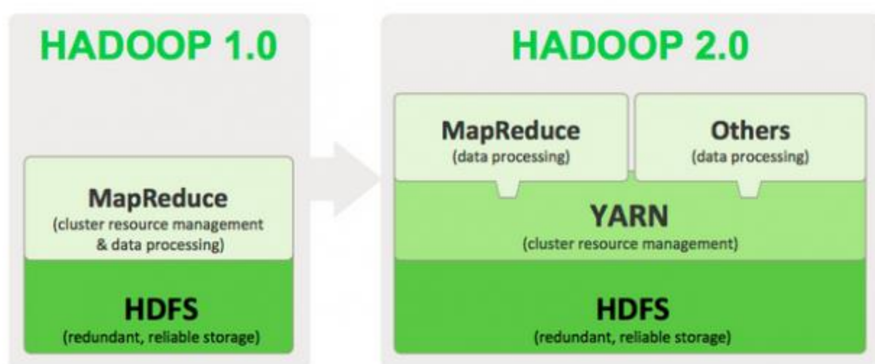

Figure 7. Difference between Hadoop 1.0 and Hadoop 2.0 In Hadoop 2.0. In Hadoop 2.0 YARN takes the responsibility of resource management from MapReduce.

Apache Hadoop is an open source software framework for storage and processing of large-scale data [6]. The framework includes the components: Hadoop Distributed File System (HDFS), Yet Another Resource Negotiator (YARN), MapReduce and Common.

The HDFS stores all the data collected from the patient through the ADL Recorder App across multiple machines or servers in a distributed manner.

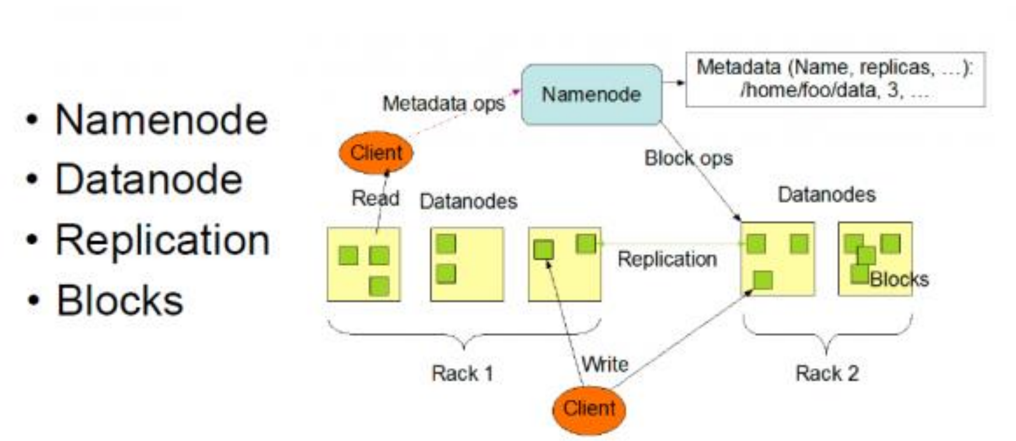

Figure 8. Structure of HDFS

As shown in Figure 8. It was inspired by Google File System [3] that HDFS is designed to have a master-slave architecture [6]. An HDFS cluster includes a single master server called "Namenode" and multiple slave servers called "Datanode". The Namenode manages the file system namespace and controls the user's access to the file. The Datanode is responsible for storing the data blocks split from the file data. By default, for each file, there are 3 copies across the whole file system, two of them are within the same rack, and one is in a different rack to preserve the file's high availability and prevent data loss when the system breaks down.

The Namenode and Datanode are pieces of software designed to run on commodity machines. In this prototype, the Namenode is deployed on machine "H01".The Datanodes are deployed on machine "H02" through "H06".

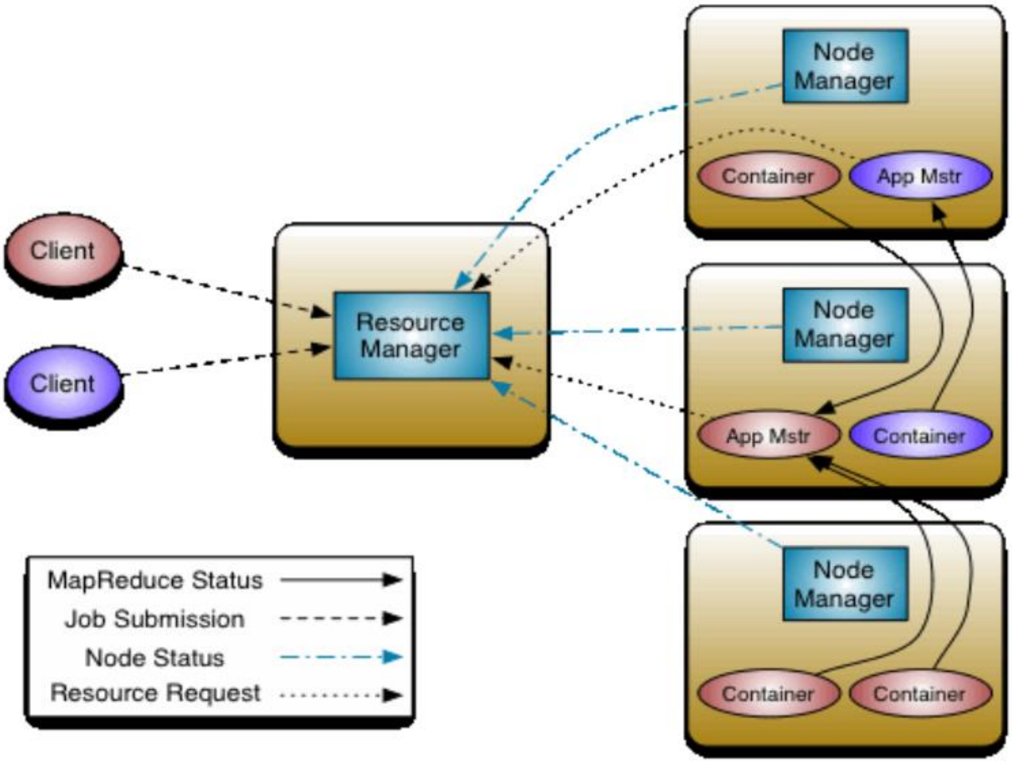

Figure 9. YARN structure

4.2 YARN

YARN (Yet Another Resource Negotiator) was introduced into Hadoop 2.0 in order to separate the functionalities of resource management where YARN is responsible for, and job scheduling/monitoring where MapReduce is responsible for. See Figure 9, YARN has kind of "master and slave structure", is to have global resource manager called "ResourceManager (RM)" and per-application resource managers called "Application-Master(AM)". The central resource manager is the ResourceManger (RM). For each application using the Hadoop, additional second-level resource manager ApplicationMaster(AM) will be created by RM. The YARN is deployed on machine

H02. In addition, a secondary namenode (standby namenode) is also deployed on machine H02 to achieve high availability and prevent single-failure breakdown.

4.3 MapReduce, Hadoop MapReduce and Common

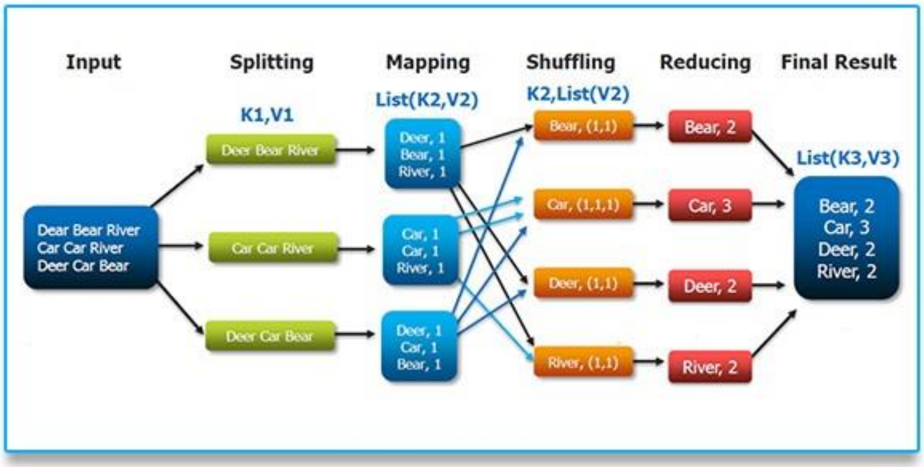

Figure 10. the process of MapReduce.

MapReduce is a programming model and Hadoop MapReduce is the associated implementation. Both are for the purpose of processing and generating large data sets. The map function is to process and generate a set of intermediate key-value pairs. The reduce function is responsible for merging values associated with the same intermediate key. In the ADL capturing system, we deployed Spark as our computation engine. It is very similar to MapReduce as use the mapping-reducing model. The difference is Spark takes advantage of its host machine's memory while Hadoop MapReduce does not.

Hadoop Common includes the native code and required jar files. The purpose of native code is for performance reasons and for non-availability of Java implementations. The user of Hadoop would not need to worry about the native code.

Figure 10 gives an example of how MapReduce executes a word count job. It first creates the key-value pairs by splitting the words and mapping each individual word (the key) with a integer value (the value). Then the same key pairs will be sent to the same nodes for reducing (aggregation). At last, the result will be combined and calculated.

CHAPTER 5. SUMMARY

The system stores 1068 files and directories, including 963 blocks which is equals to 2030 total filesystem objects. The heap memory usage is 56.36 MB out of 413MB. The max heap memory is 889 MB. The non-heap memory usage is 49 MB out of 50.02 MB.

Configured Capacity:	749.84 GB
DFS Used:	22.82 MB (0%)
Non DFS Used:	207.26 GB
DFS Remaining:	542.56 GB (72.36%)
Block Pool Used:	22.82 MB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.01% / 0.00%
Live Nodes	6 (Decommissioned: 0)
Dead Nodes	0 (Decommissioned: 0)
Decommissioning Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0

Figure 11. The system summary

As we can see from Figure 11, there are 6 living nodes in the system. The total configured capacity of the system is 749.84 GB. Hadoop DFS used 22.82 MB of storage. The host OS and Hadoop software used 207.26 GB of storage. And the system has 542.56 GB storage space left.

The root directory contains the ADL_Data folder. All the uploaded data (both raw and processed) are backed up on the HDFS.

/ Go!

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
drwxr-xr-x	wjx	supergroup	0 B	7/11/2019, 3:40:35 PM	0	0 B	ADL_Data

Figure 12. HDFS root directory

Node(machine) information is shown in the Figure 12. Namenode is deployed on node h01. Secondary Namenode and YARN are deployed on node h02. Totally 6 Datanodes are deployed across h01 – h06. As we can see from Figure 13, that each single machine has less 500 GB storage. However, together the HDFS can have a more 700 GB storage capacity.

Node	Last contact	Admin State	Capacity	Used	Non DFS Used	Remaining	Blocks	Block pool used	Failed Volumes	Version
h03.vm.com:50010 (192.168.1.203:50010)	1	In Service	37.24 GB	655.38 KB	13.74 GB	23.5 GB	31	655.38 KB (0%)	0	2.7.2
h02.vm.com:50010 (192.168.1.202:50010)	2	In Service	37.24 GB	731.47 KB	14.02 GB	23.22 GB	40	731.47 KB (0%)	0	2.7.2
h04.vm.com:50010 (192.168.1.204:50010)	0	In Service	37.24 GB	2.58 MB	14.44 GB	22.8 GB	285	2.58 MB (0.01%)	0	2.7.2
h01.vm.com:50010 (192.168.1.201:50010)	1	In Service	456.33 GB	10.36 MB	72.41 GB	383.91 GB	955	10.36 MB (0%)	0	2.7.2
h06.vm.com:50010 (192.168.1.206:50010)	1	In Service	144.54 GB	5.47 MB	78.24 GB	66.3 GB	373	5.47 MB (0%)	0	2.7.2
h05.vm.com:50010 (192.168.1.205:50010)	0	In Service	37.24 GB	3.06 MB	14.41 GB	22.83 GB	333	3.06 MB (0.01%)	0	2.7.2

Figure 13. The node information.

5 out of 6 machines of the system has less than 150 GB storage. This system combines less capability machines to a much powerful system. This work shows that the idea of using big data technology to solve the big data issues is feasible.

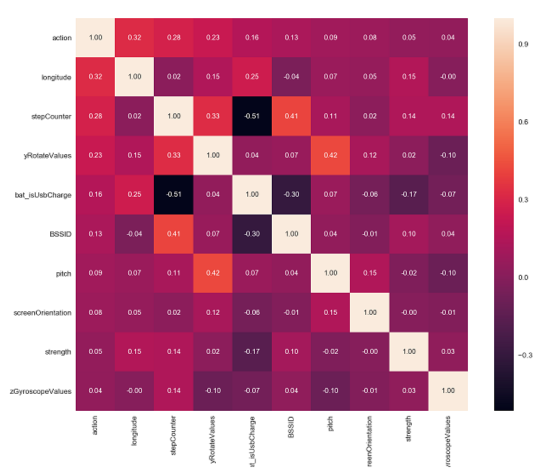

Figure 14. Feature Correlations

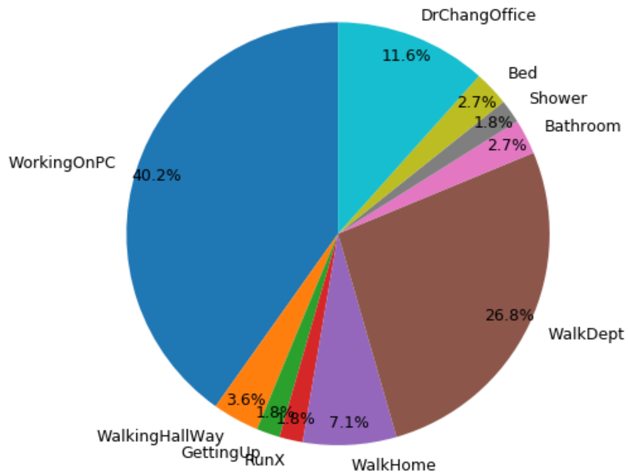

Figure 15. ADL Predictions

Figure 14 shows the correlations between different type of sensor collected data. Figure 15 shows the probabilities of different actions a care recipient could be doing between 2 PM and 4 PM during a day. These data used in creating these figures are collected during the process of the implementation of the system. The purpose is that to show that this system and the data it collects could be used for researcher and experts to help improve the living quality and health condition of not only elderly people but people in need for caring and monitoring.

REFERENCES

[1] Feng, Yunfei, Carl Chang, and Hua Ming. "Engaging Mobile Data to Improve Human Well-being: the ADL Recognition Approach." *IT Professional* (2017).

[2] Feng, Yunfei, Carl K. Chang, and Hanshu Chang. "An adl recognition system on smart phone." *International Conference on Smart Homes and Health Telematics*. Springer, Cham, 2016.

[3] Ghemawat, Sanjay, Howard Gobioff, and Shun-Tak Leung. "The Google file system." (2003).

[4] Jefferson, Angela L., et al. "Characterization of activities of daily living in individuals with mild cognitive impairment." *The American Journal of Geriatric Psychiatry* 16.5 (2008): 375-383.

[5] Mlinac, Michelle E., and Michelle C. Feng. "Assessment of activities of daily living, self-care, and independence." *Archives of Clinical Neuropsychology* 31.6 (2016): 506-516.

[6] Shvachko, Konstantin, et al. "The hadoop distributed file system." *MSST*. Vol. 10. 2010.

[7] Tamamizu, Kazunari, et al. "Capturing activities of daily living for elderly at home based on environment change and speech dialog." *International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management*. Springer, Cham, 2017.

[8] Thomas, Vince Salazar, Kenneth Rockwood, and Ian McDowell. "Multidimensionality in instrumental and basic activities of daily living." *Journal of clinical epidemiology* 51.4 (1998): 315-321.

[9] Xia, Xiaoye. *Body Motion Capture Using Multiple Inertial Sensors*. Diss. University of Saskatchewan, 2012.

[10] Yang, Longqi, et al. "Your activities of daily living (YADL): An image-based survey technique for patients with arthritis." *Proceedings of the 10th EAI International Conference on Pervasive Computing Technologies for Healthcare*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2016.