ETM

ENGINEERING & TECHNOLOGY MANAGEMENT

Choosing the NoSQL Database Management System for Big Data Analysis: A Hierarchical Decision Model

Course Title:	Decision Making
Course Number:	ETM 530/630
Instructor:	Prof. Ramin Neshati
Term:	Spring 2017
Author:	Kwangwoong Yang

Table of Contents

I. Abstract

Because of dramatically increasing data called “Big Data.” As the Big Data market grows, new products and new technologies related to NoSQL have emerged dramatically in recent years. A number of enterprises relying on relational databases for supporting their businesses during decades have turned to utilize NoSQL database system. Since most NoSQL products are open source and low cost to get start, many small companies for preparing to introduce NoSQL system to utilize their data as their property. However, because there are too many NoSQL products on the market as open source these days, this makes small companies difficult for choosing a proper NoSQL product. At this point, the study of NoSQL selection has significant worth. For this project, a Hierarchical Decision Model (HDM) is used as a method to evaluate criteria, factors, and alternatives. For design HDM, I found several criteria and factors as the sub-criteria of the criteria from the research. However, this model has limitation for more accurate evaluation. In particular, the factors need to be redefined by the characteristic of each project, the type of data, or the business environment.

II. Introduction

Since 1970's Relational databases was developed, this database system is dominated so far in the DBMS market. Relational databases have supported critical applications and provided strong technical and professional work. However, NoSQL database system has gained much popularity in recent years and it has been questioned about limitation of relational databases as storage systems for the Big Data especially in Business Intelligence aspect. According to DB-ENGINES ("historical trend of the popularity ranking of database management systems," n.d.), the NoSQL market has been growing rapidly since 2013. Since most of NoSQL is open source in particular, many companies can easily consider introducing it. In addition, NoSQL databases offer many advantages over traditional relational technologies satisfying many companies' requirements, including an excellent performance, flexible data model, and powerful scalability. However, there are too many products in the market and there is a lack of indicators of professional and objective performance for NoSQL products. So, this paper focuses on what is the critical factors for choosing NoSQL database product through experts.

A. Problem Statement

There is a problem in choosing proper NoSQL database system. According to NOSQL Databases (NOSQL Databases, n.d.), as of May 2017 there are over 200 NoSQL products. Companies cannot easily select a NoSQL product by simply evaluating performance. Beyond the performance, there are a number of things to consider like economical, maintenance, and operation. The selection criteria will vary depending on the needs of the service provided, or the overall current system. This study will contribute to give companies, preparing to introduce a NoSQL system, the model to choose proper a NoSQL system.

B. Objective

The objective of this project is choosing NoSQL database product by developing an HDM decision model. This project focuses on identifying the most feasible database management system for big data analysis. We are living in Big Data era. NoSQL is a shorthand for Not only SQL. NoSQL is a complement to the difficulties that relational databases (RDBMSs) cannot solve as large-scale data processing becomes an issue in hierarchical, network, and relational databases. This project takes a look at NoSQL, which plays a key role in the era of Big Data, and choose the right product among NoSQL that can be an alternative.

III. Literature review (Big Data and NoSQL)

A considerable amount of literature has been published on NoSQL. These studies

A. Big Data

According to Ishwarappa et al. (2015), “Big data is a collection of massive and complex data sets and data volume that include the huge quantities of data, data management capabilities, social media analytics and real-time data.”

1. Characteristics of Big Data

To understand the big data, it is often described using five Vs: Volume, Velocity, Variety, Veracity and Value. These are also important characteristics for NoSQL database management system (DBMS).

a) Volume

Volume is related to the vast size of data generated every second. Ishwarappa el al. (2015) show that many companies have a lot of archived data from their service like the form of logs, but they do not have the capacity to process that data yet. The advantage of being able to process large amounts of information is a major attraction for Big Data analysis (Ishwarappa et al., 2015).

b) Velocity

Velocity means that the speed in increasing in generating data speed and the increasing speed in data should be processed, stored and analyzed (Ishwarappa et al., 2015). The speed represents the speed of the new generated data. Ishwarappa et al. (2015) show examples of the velocity that "About social media messages going to viral in seconds in 1999, Wal-Mart s data warehouse stored 1,000 terabytes (1,000,000 gigabytes) of data. In the year 2012, it had access to over 2.5 petabytes (2,500,000 gigabytes) of data. Every minute of every day, we upload hundreds hours of video on YouTube. We send over 200 million emails through Gmail's."

c) Variety

According to Ishwarappa et al. (2015), "Big Data is not always structured data and it is not always easy to put big data into a relational database. This means that the category to which Big Data belongs to is also a very essential fact that needs to be known by the data analysts Dealing with a variety of structured and unstructured data greatly increases the complexity of both storing and analyzing Big Data. 90% of data generated is data is in unstructured form."

d) Veracity

Ishwarappa et al. (2015) explains that system cannot guarantee 100% accuracy, when a system dealing with a big size, volume, velocity and variety of data. So, inevitably there will be inaccurate data. In addition, they mention that "The quality of the data being captured can vary greatly and the data accuracy of analysis depends on the veracity of the source data."

e) Value

Ishwarappa et al. (2015) insist that the most important of these five Vs of Big Data is Value. They also explain that "Though, the potential value of the Big Data is huge. It is all well and good having access to big data but unless we can turn it into value it is become useless. It becomes very costy to implement IT infrastructure systems to store big data, and businesses are going to require a return on investment."

B. DBMS

1. Brief History of DBMS

The database term was introduced in the early 1960's and this database simply support for representing the structure of a system (Pop et al., 2015). At that time, the database used a tape recorder and later introduced a system using disks (Pop et al., 2015). Pop et al. (2015) shows that in 1970, relational model using Structured Query Language (SQL) which is very influential model until now. By the early 2000s, data became increasingly complex and rapidly expanding. The social network era has also begun, which produces bigger data. At this time, the term of NoSQL came from the limit of RDBMS (Pop et al., 2015).

2. RDBMS

According to Pop el at. (2015), "In 1970, the Relational Database Management System proposed by Edgar Codd. This system has the relational model and it is better efficient data-storage compared to other systems in the past like IBM's Information Management System (IMS). This model utilizes Structure Query Language (SQL) to define tables or relationships and find stored data. Pop el at. (2015) insist that "this relational model was almost identical with what we call today the traditional relational model."

3. CAP theorem

The theory is related to data storage and distribution environment. In July 2000 this theorem was introduced in a lecture by Professor Eric Brewer of the University of Berkeley entitled “Towards Robust Distributed System.” According to Han el al. (2011), “CAP theorem's core idea is a distributed system cannot meet the three district needs simultaneously, but can only meet two.” Cap theorem is one of the major contributors of the NoSQL movement. (Chandra, 2015) CAP theorem consists of three part: Consistency, Availability, and Partition tolerance. According to Gilbert et al. (2012), the consistency requires proper response to each request for each server and appropriate response is important to the desired service specification. Second, availability is related to receiving responses. Each request should receive their response. (Gilbert et al., 2012) Third, Partition tolerance is related to system. Servers can be partitioned into multiple groups but it should be able to operate the system even if distributed to network nodes and work normally even if there is a communication problem between the nodes. (Gilbert et al., 2012)

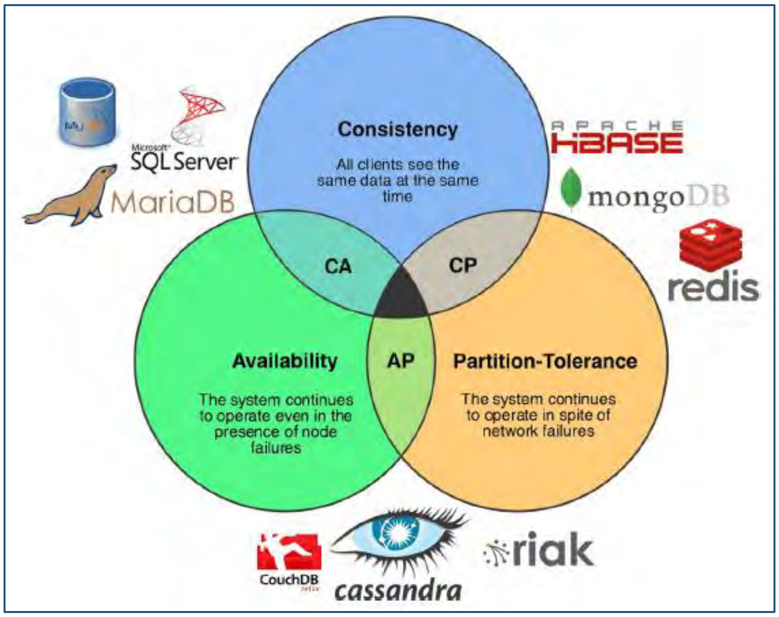

Consistency
All clients see the same data at the same time
Availability
The system continues to operate even in the presence of node failures
Partition-Tolerance
The system continues to operate in spite of network failures
CA
CP
AP
Microsoft SQL Server
MariaDB
APACHE HBASE
mongoDB
redis
CouchDB
cassandra
riak

Figure 1. CAP theorem with databases (Lourenço et al., 2015)

In the following sections NoSQL types of products are introduced. In that section the detail of NoSQL family will be introduced. Depending on the characteristics of the product family, each NoSQL database types emphasize the performance of either side of CAP. Most RDBMS products like Oracle, MySQL, and MSSQL have CA attributes. In this project we have three alternatives as choosing NoSQL database systems. Among them, MongoDB and Redis products correspond to CP and Cassandra product belongs to AP.

4. NoSQL

Yoon et al. (2015) insist that "RDBMS are too expensive, under-functioning and too complex to be applied to such fields as Big Data Analysis, Log Analysis, Social Network Services and Mobile Applications, which have recently been on the rise." According to Pop et al. (2015), NoSQL has a differences as an advantage from existing relational databases:

- No relational model, no joins to link tables
- Client's data is scalable and flexible
- Tie multiple database servers (clustering) and configure as one database
- Transactions are not guaranteed while Relational databases support it
- Diversify and dynamically define the schema and attributes of data
- Supports uninterrupted service and automatic recovery of database
- Most products are provided as Open Source

However, NoSQL also has a disadvantage. According to Pop et al. (2015),

- No reference standard
- Each database behaves differently
- No Structure Query Language (SQL) as common query language

- No guarantee for confirmation of writing of data

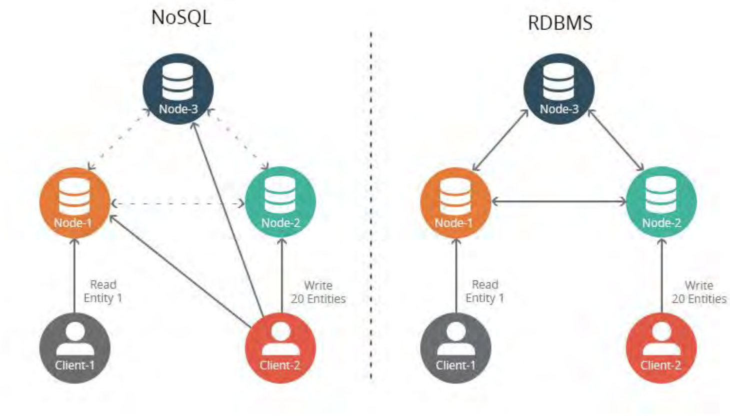


Figure 2. Comparison RDBMS vs. NoSQL (Fernando, 2016)

C. NoSQL Database

Currently, there are four different kinds of NoSQL DBMS are in the market: Key-Value database type, Column-oriented database type, Document database type, and Graph databases type. This section looks at the differences between these four types of databases.

1. Key-Value database type

Key-Value databases use a hash table. According to Sharma (2016), “this databases store the structured or unstructured data in key–value pairs, where the data is the value, which is retrieved by the key of the pair.” Amazon S3 (Dynamo), Voldemort, and Scalaris are using the Key-Value database system. The Key-Value model is very easy to implement and simple. According to Han et al. (2015), “Key-Value data model means that a value corresponds to a Key, although the structure is simpler, the query speed is higher than relational database, support mass storage and high concurrency, etc., query and modify operations for data through the primary key were supported well.”

2. Column-oriented database type

Cassandra and Hbase databases are Column-oriented type of database. This type of databases is using Table as the data model, but does not support table association. According to Chandra (2015), this type of database is a hybrid type between NoSQL and relational databases. "These databases provide some row-and-column structure, but do not have the strict rules of relational databases. Column-oriented databases store and process data by column instead of row. Having its origin in analytics and business intelligence, column-stores can be used to build high-performance applications." (Chandra, 2015)

3. Document database type

This database type stores the collections of the documents (Sharma, 2016). Document database and Key-value database are very similar in structure. However, Document database support JSON or XML format to store data and the system is semantic (Han et al., 2015). MongoDB, CouchDB, and Cloudant are using document database type.

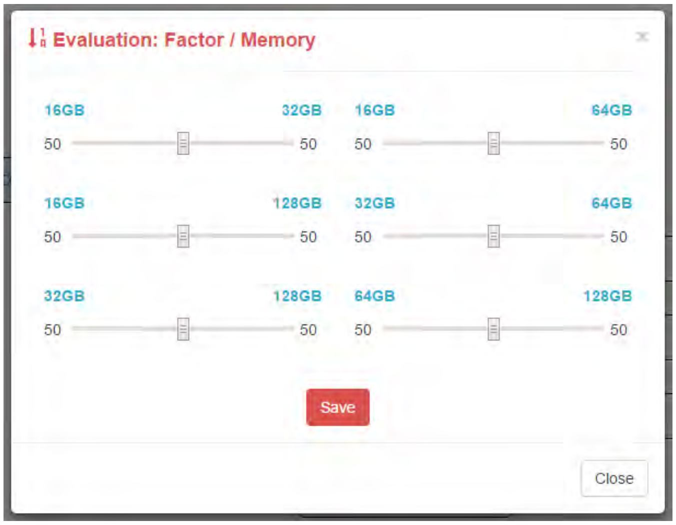
4. Graph Databases type

Graph Databases use graph (or edge or relationship) structures having multiple nodes as attributes. Node4j is a representative product of this database system, which store information in multi-attribute tuples that reflect relationships in a different way (Chandra, 2015). According to Chandra (2015), "At the top of these databases there may be a key/value store, columnar database, 'BigTable' database or combination of these and other architectures."

IV. Methodology – HDM

A. The Hierarchical Decision Model (HDM)

Hierarchical Decision Model (HDM) is a variant of AHP as a multi-attribute hierarchical decision making tool. There are three concepts form the basis of HDM/AHP modeling (Abbas, 2016). Abbas (2016) explains that these concepts are “1. Structuring the decision problem in a hierarchy consisting of goal, criteria and alternatives. 2. Conducting pairwise comparisons among all variables at every hierarchy of the decision model with respect to each criterion on the prior/higher level. 3. Synthesis of priorities at all levels of the hierarchy after obtaining the relative judgment weights, and checking the consistency.” At each level, experts evaluate the lower level elements by pairwise comparison. For example, if objective has three criteria (A, B, C), then an expert will evaluate for (A:B), (A:C), and (B,C).



Evaluation: Factor / Memory

16GB 32GB
50 50

16GB 64GB
50 50

16GB 128GB
50 50

32GB 64GB
50 50

32GB 128GB
50 50

64GB 128GB
50 50

Figure 3.Example of Pairwise Comparison

Abbas (2016) explains about the advantages of pairwise comparisons in the HDM that “HDM pairwise comparisons can be given in a single format of the above or a combination of them. This allows HDM users speed and flexibility without having to mentally or arithmetically convert data into a particular scale. The fine gradations, afforded by these input scales, allow better control and accuracy without the limitation of Saaty's nine-point scale. This also prevents the negative effects of discretization often associated with Saaty's scale"

V. The HDM Model

HDM model was created based on literature review. This model has four levels: objective, criteria, factors, and alternatives. Figure 2 shows the details of the model and nodes at each level. This figure was created by HDM Designer application.

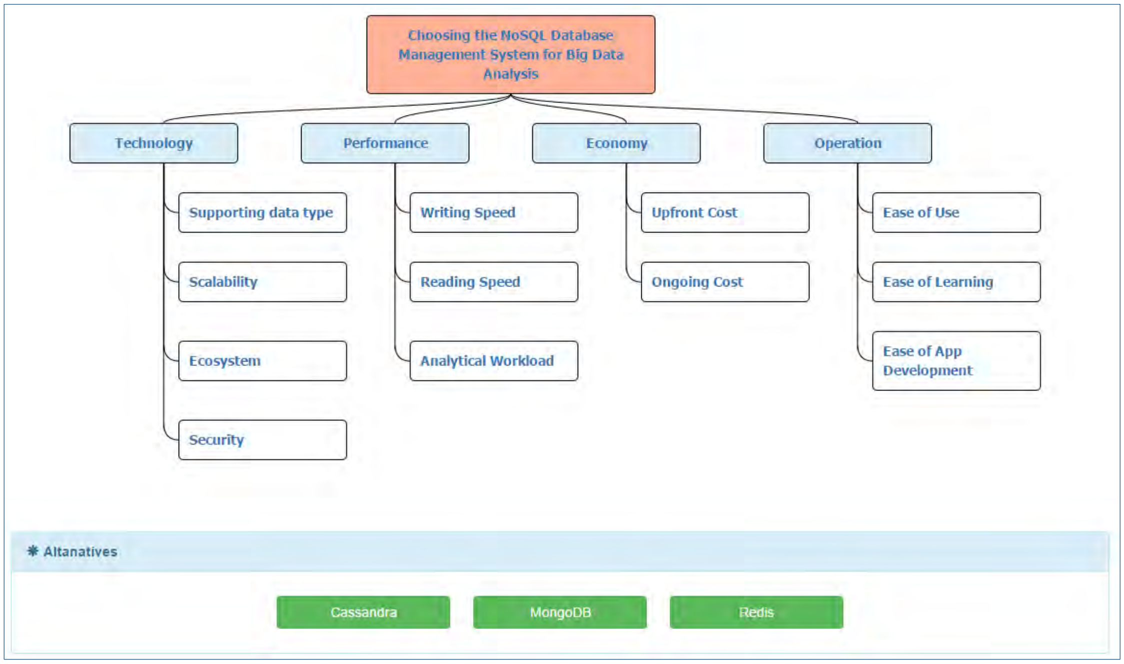


Figure 4. Final HDM Design

1st Level: This level represents the objective of the model as a goal of the project, "Choosing the NoSQL Database Management System for Big Data Analysis"

2nd Level: This level has four criteria: Technology, Performance, Economy, and Operation.

3rd Level: There are 12 factors as the sub-criteria for the four factors. These criteria will be discussed in following subject, Criteria Building.

4rd Level: This level comprises of three alternatives: MongoDB, Cassandra, and Redis. According to DB-Engines Ranking (2017), these products are the most widely used products on the market in top 10 of DBMS in 2017.

A. Criteria Building

1. Technological

a) Supporting data type:

As we have seen before, there are four kinds of NoSQL DBMS products. This criterion evaluates the proper kind of system.

b) Scalability:

large unstructured distributed data requires high scalability to expand the system. According to Pokorny (2013), “Database partitioning across multiple cheap machines added dynamically, so-called horizontal scaling (also scale-out), can apparently ensure scalability in a more effective and cheaper way.”

c) Ecosystem:

NoSQL cannot work by itself but, it cooperates with other related systems like Hadoop system. A NoSQL system having a good ecosystem will have better powerful performance.

2. Performance

For the performance, NoSQL databases are divided mostly into two categories: read and write optimized, which is related to the mechanisms utilizing storage, organization and retrieval of data (Lourenço et al. 2015).

a) Reading Speed

b) Inserting Speed

c) Analytical Workload

3. Economical

According to Henschen (2013), because NoSQL is free and open source, to many IT teams attracted to introduce NoSQL system.

a) Upfront Cost

Upfront Cost includes initial development costs (Application coding and Data store), initial administrator costs (Installation, configuration, Shard/Replication, etc.), software licenses, server hardware, and server storage costs.

b) Ongoing Cost

Ongoing Cost includes ongoing developer costs (Change according to user's demand), ongoing administrator costs (Data Maintenance, Health Check, Backup & Recovery, etc.), Software Maintenance and Support, Server Maintenance and Support, and Storage Maintenance and Support.

4. Operation

a) Ease of Use

This factor is about the ease of installation, management, and administration. For management and administration parts are related to support client program.

b) Ease of Learning

To use RDBMS, people have to learn Structured Query Language (SQL). This language has been standardized for most RDBMSs. However, because almost NoSQL product use different instructions each other, it should be easy to learn.

c) Ease of App Development

Ultimately, programs will be developed with NoSQL to represent the data stored in the DBMS. So, it should support convenient tools and enough APIs for program development.

B. Experts

To evaluate the model has two experts in the Business Intelligence field related to NoSQL. This table shows that experts are currently student, but they have experience in Big Data and NoSQL.

Expert		**Experience**		
	Position	**Big Data**	**Software Development**	**NoSQL**
Expert 1	Business Intelligence Major	🟦		🟦
Expert 2	Engineering & Tech. Management, Business Intelligence Major	🟦	🟦	🟦

VI. Data Analysis and Result

A. HDM model results

From the evaluation of the experts, the results for the alternatives are as follows.

	Cassandra	**MongoDB**	**Redis**	**Inconsistency**
Expert 1	0.32	0.36	0.33	0
Expert 2	0.32	0.4	0.29	0.01
Mean	0.32	0.38	0.31	

Minimum	0.32	0.36	0.29	
Maximun	0.32	0.4	0.33	
Std. Deviation	0	0.02	0.02	
Disagreement				0.016

Table 1. The result of experts' evaluation

The statistical F-test for evaluating the null hypothesis (Ho: ric = 0) is obtained by dividing between-subjects variability with residual variability:

<table>
<tr><td>Source of Variation</td><td>Sum of Square</td><td>Deg. Of Freedom</td><td>Mean Square</td><td>F-test Value</td></tr>
<tr><td>Between Subjects:</td><td>0.01</td><td>2</td><td>0.003</td><td>3.58</td></tr>
<tr><td>Between Conditions</td><td>0.00</td><td>1</td><td>0.000</td><td></td></tr>
<tr><td>Residual</td><td>0.00</td><td>2</td><td>0.001</td><td></td></tr>
<tr><td>Totl:</td><td>0.01</td><td>5</td><td></td><td></td></tr>
<tr><td colspan="4">Critical F-value with degrees of freedom 2 & 2 at 0.01 level:</td><td>99</td></tr>
<tr><td colspan="4">Critical F-value with degrees of freedom 2 & 2 at 0.025 level:</td><td>39</td></tr>
<tr><td colspan="4">Critical F-value with degrees of freedom 2 & 2 at 0.5 level:</td><td>19</td></tr>
<tr><td colspan="4">Critical F-value with degrees of freedom 2 & 2 at 0.1 level:</td><td>9</td></tr>
</table>

Table 2. The statistical F-test for evaluating the null hypothesis

B. Discussion

The results of evaluating the model shows that the experts chose MongoDB as the most important alternative and next, Redis is followed. Cassandra received the lowest score. Cassandra shows high performance on an objective basis. Currently, MongoDB is ranked top in the NoSQL market. Cassandra database shows objectively high performance but lower score here and lower market share. This is because it does not have the advantages of the RDBMS. On the other hand, MongoDB has a driver concept. MongoDB supports most of the drivers for mainstream libraries that can interface with and interact with MongoDB. Cauchy DBI uses web standards-based interactions, so any programming language that supports web communication can access it in any language ("Driver Compatibility," n.d.). RDBMS has evolved for 40 years, but NoSQL is now in its infancy. An RDBMS provides a consistent way to model data. The data model is under the relationship. This theory is well established and the implementation is standardized. Therefore, a consistent way of modeling and normalizing data is well understood and well documented. The NoSQL world does not have these standards or well-defined data models. This is because all NoSQL products are not designed to address the same problem, nor do they have the same architecture. As Lourenço et al. (2015) mention that NoSQL has not yet reached a mature stage, but it is still in-developing stage.

VII. Conclusion

A. Limitations and Future Research

For this paper there are some limitations as follows:

- Difficult to find a perfect expert: there was a difficulty in finding the person responsible for working in the analysis using NoSQL.
- Professional experience for various NoSQL products: most students who requested an evaluation had only one experience with NoSQL product. This is expected to have had a significant impact on their assessment.
- From the point of looking for a job related to Data Analysis: this is related to the previous point. Because the experts are preparing for the job related Big Data Analysis using NoSQL, experts have used the product which is needed in the job market.

VIII. References

McAfee, A., Brynjolfsson, E., & Dearstyne, B. W. (2012). Big Data's Management Revolution: Interaction. *Harvard Business Review*, *90*(12), 16–17.

Lourenço, J. R., Cabral, B., Carreiro, P., Vieira, M., & Bernardino, J. (2015). Choosing the right NoSQL database for the job: a quality attribute evaluation. *Journal of Big Data*, *2*(1), 18.

Fernando, S. F. (2016, October 25). NoSQL ? Retrieved May 31, 2017, from https://www.linkedin.com/pulse/nosql-suzanne-fiona-fernando

Driver Compatibility. (n.d.). Retrieved May 31, 2017, from https://docs.mongodb.com/ecosystem/drivers/driver-compatibility-reference/

Ishwarappa, & Anuradha, J. (2015). A Brief Introduction on Big Data 5Vs Characteristics and Hadoop Technology. *Procedia Computer Science*, *48*, 319–324.

Yoon, J., Jeong, D., Kang, C., & Lee, S. (2016). Forensic investigation framework for the document store NoSQL DBMS: MongoDB as a case study. *Digital Investigation*, *17*, 53–65.

Ganesh Chandra, D. (2015). BASE analysis of NoSQL database. *Future Generation Computer Systems*, *52*, 13–21. https://doi.org/10.1016/j.future.2015.05.003

Gilbert, S., & Lynch, N. (2012). Perspectives on the CAP Theorem. *Computer*, *45*(2), 30–36. https://doi.org/10.1109/MC.2011.389

Grad, B., & Bergin, T. J. (2009). Guest Editors' Introduction: History of Database Management Systems. *IEEE Annals of the History of Computing*, *31*(4), 3–5. https://doi.org/10.1109/MAHC.2009.99

DB-Engines Ranking - popularity ranking of database management systems. (n.d.). Retrieved May 30, 2017, from https://db-engines.com/en/ranking

Jaroslav Pokorny. (2013). NoSQL databases: a step to database scalability in web environment. *International Journal of Web Information Systems*, *9*(1), 69–82. https://doi.org/10.1108/17440081311316398

Pop, D.-P., & Butoiu, E. C. (2015). An analisys of RDBMS, OOP and NoSQL conccepts. *Journal of Information Systems & Operations Management*, *9*(1), P1.

Han, J., E, H., Le, G., & Du, J. (2011). Survey on NoSQL database. In *2011 6th International Conference on Pervasive Computing and Applications* (pp. 363–366).

NOSQL Databases. (n.d.). Retrieved May 29, 2017, from http://nosql-database.org/

historical trend of the popularity ranking of database management systems. (n.d.). Retrieved May 29, 2017, from https://db-engines.com/en/ranking_trend

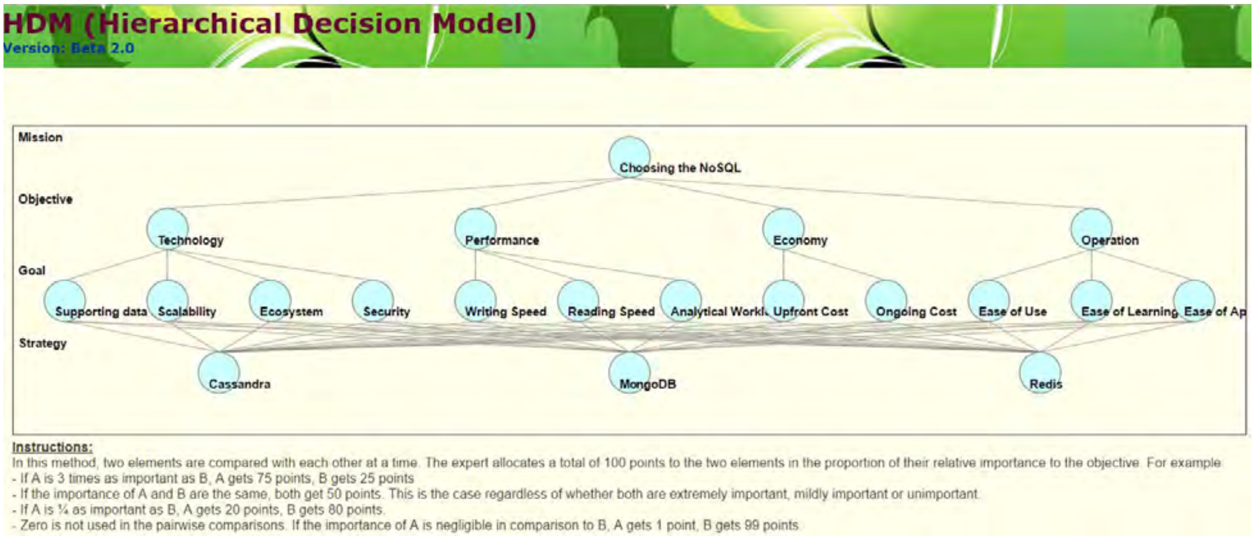
Henschen, D. (2013). When NoSQL makes sense: IT leaders must know the trade-offs they face to get NoSQL's scalability, flexibility and cost savings.(Cover story). *InformationWeek*, (1376), 8.

Abbas, M. (2016). Consistency Analysis for Judgment Quantification in Hierarchical Decision Model. *Dissertations and Theses*. https://doi.org/10.15760/etd.2695

Sharma, S. (2016). Expanded cloud plumes hiding Big Data ecosystem. *Future Generation Computer Systems*, *59*, 63–92. https://doi.org/10.1016/j.future.2016.01.003

IX. Appendix

A. Appendix A: Current evaluation tool developed by ETM

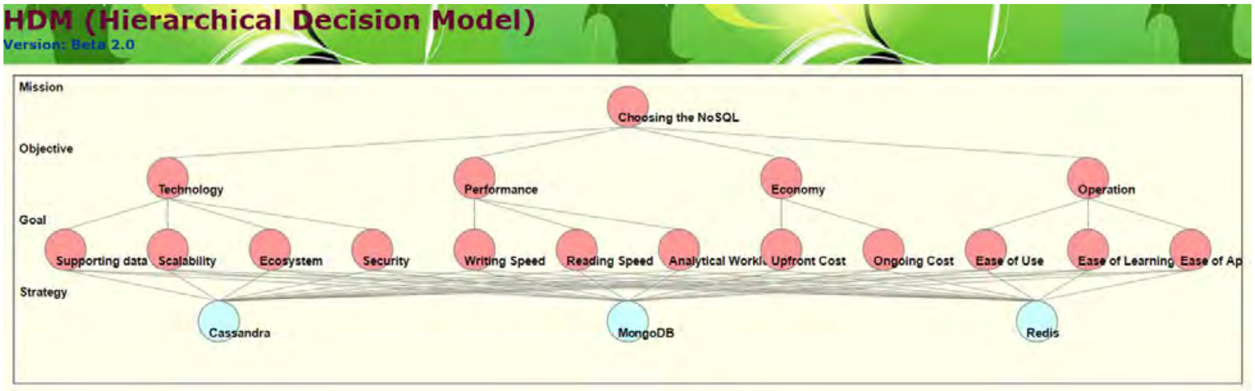

HDM (Hierarchical Decision Model)

Version: Beta 2.0

Instructions:

In this method, two elements are compared with each other at a time. The expert allocates a total of 100 points to the two elements in the proportion of their relative importance to the objective. For example

- If A is 3 times as important as B, A gets 75 points, B gets 25 points
- If the importance of A and B are the same, both get 50 points. This is the case regardless of whether both are extremely important, mildly important or unimportant.
- If A is ¼ as important as B, A gets 20 points, B gets 80 points.
- Zero is not used in the pairwise comparisons. If the importance of A is negligible in comparison to B, A gets 1 point, B gets 99 points.

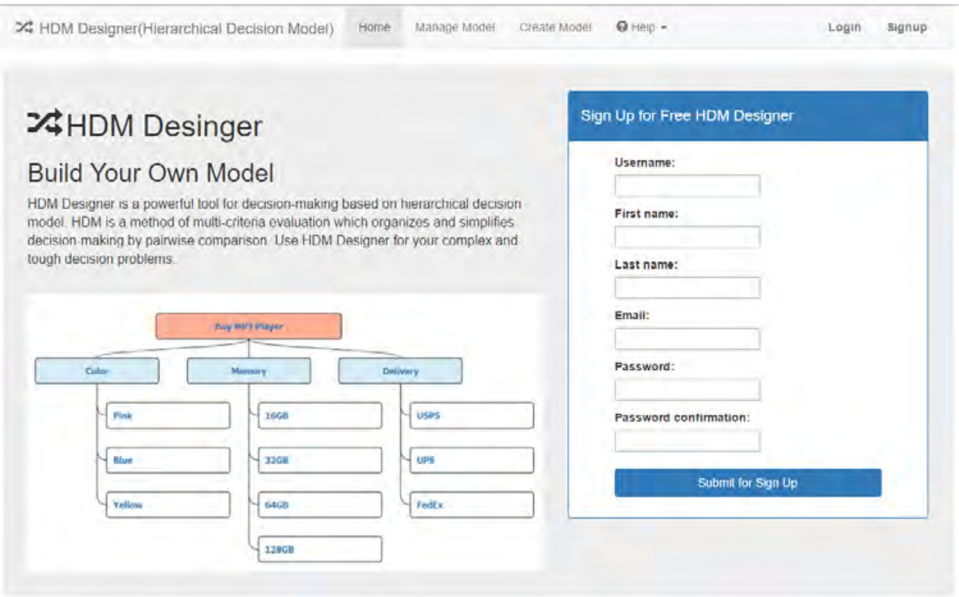

HDM (Hierarchical Decision Model)

Version: Beta 2.0

B. Appendix B: New evaluation tool under development

- Main page

People can create accounts through sign up in the main page.



HDM Desinger

Build Your Own Model

HDM Designer is a powerful tool for decision-making based on hierarchical decision model. HDM is a method of multi-criteria evaluation which organizes and simplifies decision-making by pairwise comparison. Use HDM Designer for your complex and tough decision problems.

Sign Up for Free HDM Designer

Username:

First name:

Last name:

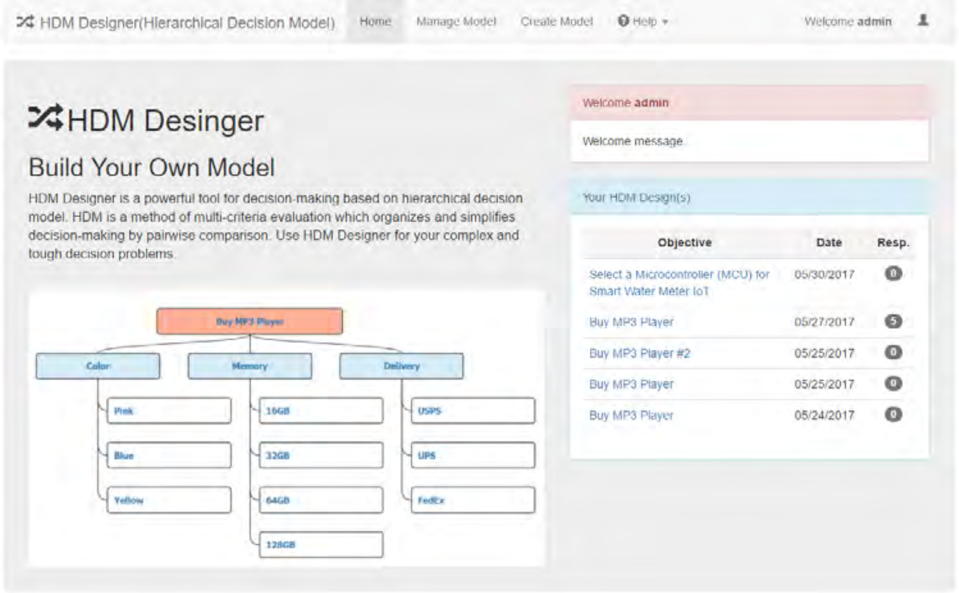
Email:

Password:

Password confirmation:

- Main page for staff accounts

After logging in, each user can see a list of models they have designed and can check the response of an expert.



HDM Desinger

Build Your Own Model

HDM Designer is a powerful tool for decision-making based on hierarchical decision model. HDM is a method of multi-criteria evaluation which organizes and simplifies decision-making by pairwise comparison. Use HDM Designer for your complex and tough decision problems.

Welcome admin

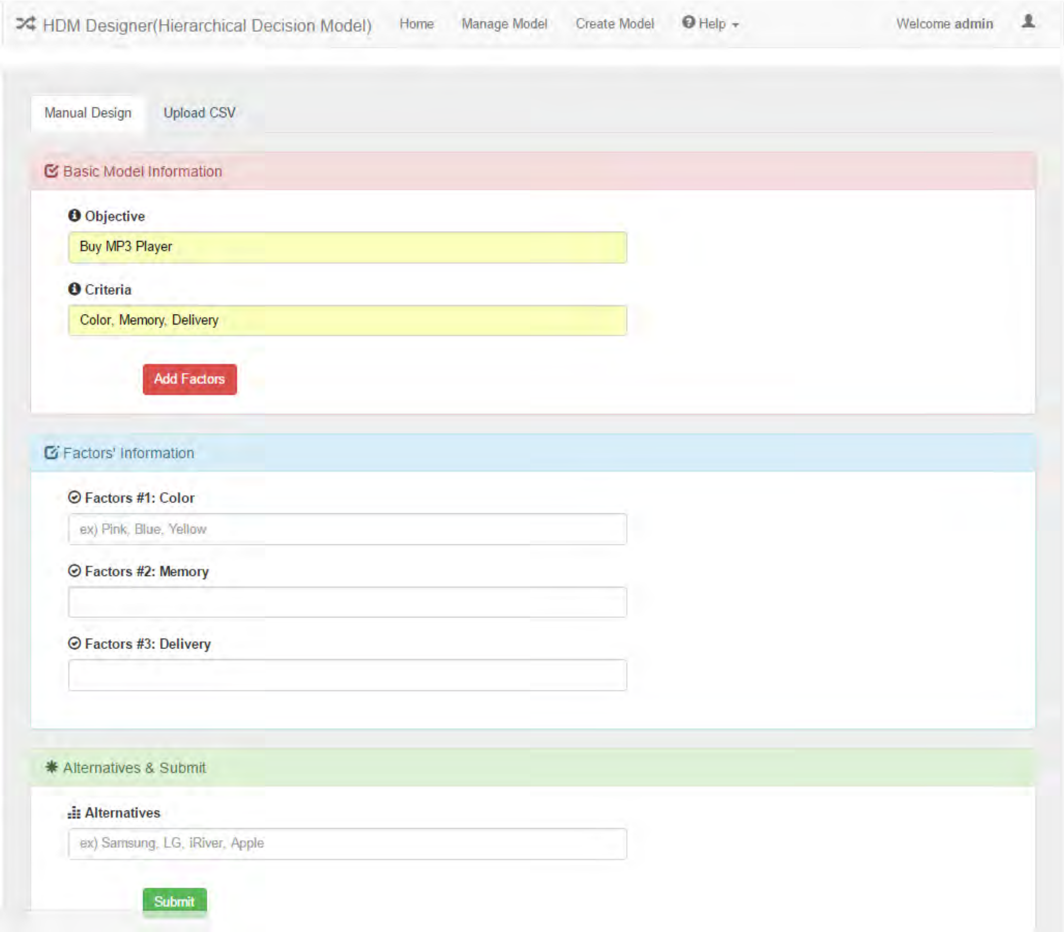
Welcome message.

Your HDM Design(s)

Objective	Date	Resp.
Select a Microcontroller (MCU) for Smart Water Meter IoT	05/30/2017	0
Buy MP3 Player	05/27/2017	5
Buy MP3 Player #2	05/25/2017	0
Buy MP3 Player	05/25/2017	0
Buy MP3 Player	05/24/2017	0

- Creating Design

A user can create a new model, which is also possible by uploading a CSV file.



Basic Model Information

Objective: Buy MP3 Player

Criteria: Color, Memory, Delivery

Factors' Information

Factors #1: Color: ex) Pink, Blue, Yellow

Factors #2: Memory:

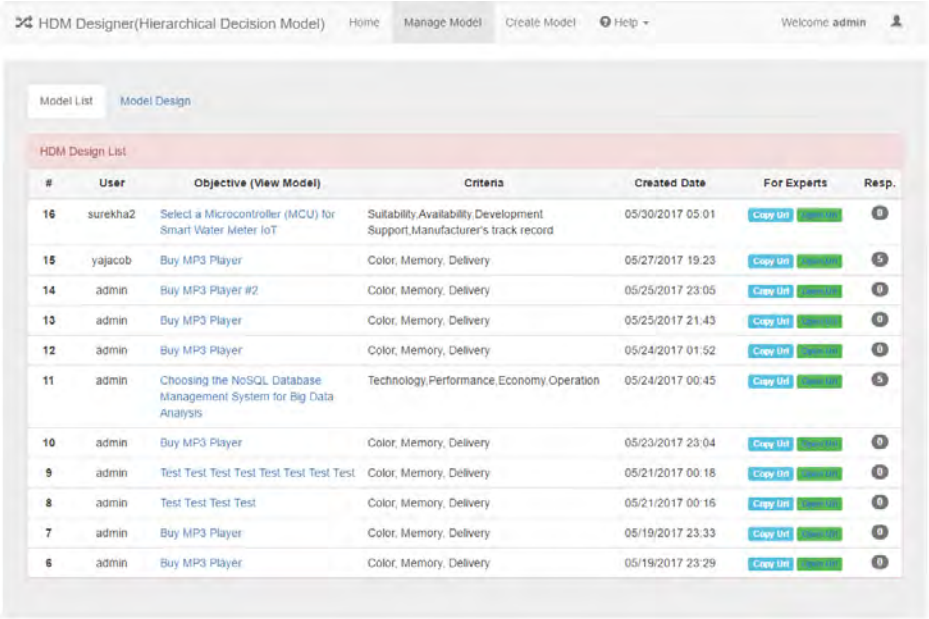
Factors #3: Delivery:

Alternatives & Submit

Alternatives: ex) Samsung, LG, iRiver, Apple

- Page for manage models

  On the ‘Model List’ page, users can see all the models they have created and copy the URL to the clipboard to send to the Expert for requesting an evaluation.

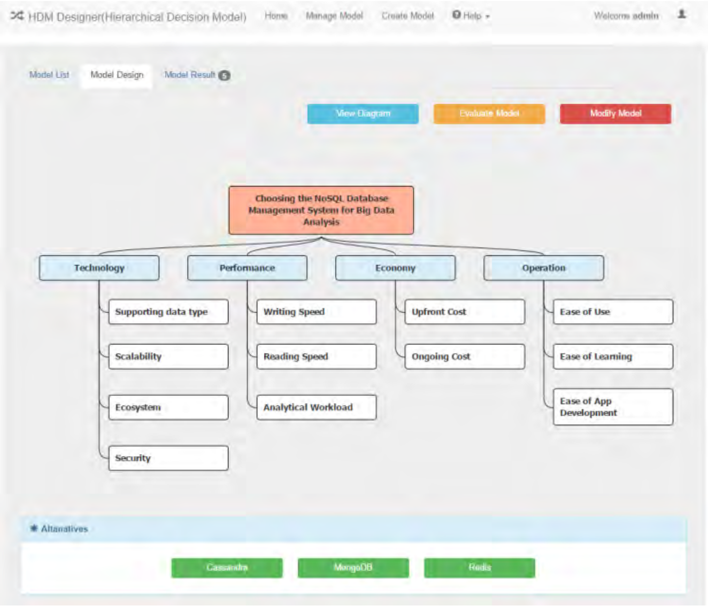


HDM Design List

#	User	Objective (View Model)	Criteria	Created Date	For Experts	Resp.
16	surekha2	Select a Microcontroller (MCU) for Smart Water Meter IoT	Suitability,Availability,Development Support,Manufacturer's track record	05/30/2017 05:01	Copy Url Open Url	0
15	yajacob	Buy MP3 Player	Color, Memory, Delivery	05/27/2017 19:23	Copy Url Open Url	5
14	admin	Buy MP3 Player #2	Color, Memory, Delivery	05/25/2017 23:05	Copy Url Open Url	0
13	admin	Buy MP3 Player	Color, Memory, Delivery	05/25/2017 21:43	Copy Url Open Url	0
12	admin	Buy MP3 Player	Color, Memory, Delivery	05/24/2017 01:52	Copy Url Open Url	0
11	admin	Choosing the NoSQL Database Management System for Big Data Analysis	Technology,Performance,Economy,Operation	05/24/2017 00:45	Copy Url Open Url	5
10	admin	Buy MP3 Player	Color, Memory, Delivery	05/23/2017 23:04	Copy Url Open Url	0
9	admin	Test Test Test Test Test Test Test Test	Color, Memory, Delivery	05/21/2017 00:18	Copy Url Open Url	0
8	admin	Test Test Test Test	Color, Memory, Delivery	05/21/2017 00:16	Copy Url Open Url	0
7	admin	Buy MP3 Player	Color, Memory, Delivery	05/19/2017 23:33	Copy Url Open Url	0
6	admin	Buy MP3 Player	Color, Memory, Delivery	05/19/2017 23:29	Copy Url Open Url	0

- Model diagram view

  On this screen, users can modify their model or open a page for the expert to check their input.


HDM Designer(Hierarchical Decision Model)
Home
Manage Model
Create Model
Help
Welcome admin
Model List
Model Design
Model Result 5
View Diagram
Evaluate Model
Modify Model
Choosing the NoSQL Database Management System for Big Data Analysis
Technology
Performance
Economy
Operation
Supporting data type
Scalability
Ecosystem
Security
Writing Speed
Reading Speed
Analytical Workload
Upfront Cost
Ongoing Cost
Ease of Use
Ease of Learning
Ease of App Development
Altanatives
Cassandra
MongoDB
Redis