IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 4, 2020, accepted August 21, 2020, date of publication August 25, 2020, date of current version September 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019331

Big Data Energy Management, Analytics and Visualization for Residential Areas

RAGINI GUPTA[1], A. R. AL-ALI[2], (Life Senior Member, IEEE), IMRAN A. ZUALKERNAN[2], (Member, IEEE), AND SAJAL K. DAS[1], (Fellow, IEEE)

[1]Department of Computer Science, Missouri University of Science and Technology, Rolla, MO 65409, USA
[2]Department of Computer Science and Engineering, American University of Sharjah, Sharjah 26666, UAE

Corresponding author: A. R. Al-Ali (aali@aus.edu)

This work was supported in part by the Faculty Research, Department of Computer Science and Engineering, American University of Sharjah, Sharjah, UAE, under Grant FRG 17-R-22, and in part by the Open Access Program from the American University of Sharjah.

ABSTRACT With the rapid development of IoT based home appliances, it has become a possibility that home owners share with Utilities in the management of home appliances energy consumption. Thus, the proposed work empowers home owners to manage their home appliances energy consumption and allow them to compare their consumption with respect to their local community total consumption. This serves as a nudge in consumer's behavior to schedule their home appliances operation according to their local community consumption profile and trend. Utilizing the same common communication infrastructure, it also allows the utilities on different consumption levels (community, state, country) to monitor and visualize the energy consumption in their respective grid segments on daily, monthly, and yearly basis. A high-speed distributed computing cluster based on commodity hardware with efficient big data mathematical algorithm is employed in this work. To achieve this, two big data processing paradigms are evaluated with a set of qualitative and quantitative metrics with subsequent recommendations. One million smart meter data is simulated to access individual homes. With the utilization of distributed storage and computing cluster for handling energy big data, the utilities can perform consumer load analysis and visualization on a scale of one million consumers. This helps the utilities in providing consumers a more accurate representation of how much energy they are consuming with greater granularity and with respect to their local community. Consumer and Utility centric queries are developed to create a web-based real time energy consumption management system presented in terms of dashboard charts, graphs, and reports that can be accessed by the consumer and utility providers remotely.

INDEX TERMS Big data, IoT, smart meter, energy management system.

I. INTRODUCTION

The smart meters are playing a major role in the growing energy management system. IoT based smart meters read energy consumption from residential areas home appliances generating data that typically exhibits the 3V characteristics of big data; Volume, Variety, and Velocity [1]. The large volume, different formats, and staggering rate of smart energy data generated in short interval reads of smart meters tax the utilities' IT resources. It has been found that at 15-minute interval span, a million smart meters can produce 400 TB of data each year [1]. The staggering rate of growth in smart home devices enabled with IoT technology, and the need to perform data analytics on the captured datasets has challenged the use of traditional utility data centers using Relational Database Management Systems (RDMS). By utilizing the energy consumption data of the household, the utilities can reveal significant information about the energy consumption lifestyle and behavior in close relation to their energy efficiency programs. For household owners, the visualization on device-level energy consumption will empower the homeowners to better operate and manage the devices for lower energy bills. Significant research efforts are needed in order to implement such a vision. Substantial amount of work has been conducted in the domain of smart meter data analytics that incorporates load forecasting, anomaly detection, load shaping strategy and dynamic pricing, as described in [2]–[4]. However, consumers are only receiving energy consumption information through either their billing information or independent consumption

The associate editor coordinating the review of this manuscript and approving it for publication was Amjad Anvari-Moghaddam.

TABLE 1. Scalability results for RDBMS vs. time series big data tool [5].

Challenges	RDMS	Proprietary Big Data Tool
Time to run reports	2-7 hrs.	25s - 6min.
Storage to 1 million meters	1.3 TB	350 GB

programs where the data is stale, offline and inaccessible. Although the utilities provide basic analytic capabilities to individual customers, the data is often stale with obsolete methods and only accessible to those customers who are enrolled in utility endorsed high-priced programs. This accounts for the lack of a scalable, commodity hardware based distributed infrastructure for monitoring and managing smart meter data.

Traditional data warehouse techniques are challenged in supporting energy big data storage to monetize this data. Recent publication highlights disadvantages of using RDMS against proprietary big data tools that employ less storage and yield faster results than traditional RDMS as shown in Table 1 [5].

A significant amount of work in energy management systems incorporating smart home interfaces to collect data from smart meters and implementing control decisions in distributed systems of smart homes [6]–[8]. From the related work, it has been found that the consumers are cautious about sharing their smart meter data with third party vendors offering them remote services to better manage their energy use [9]. Consumers trust the utilities more than third party companies' sales pitch. Hence, the utilities play an instrumental role in raising awareness about consumer's understanding of their energy consumption on a large scale. But, due to the absence of a common communication infrastructure there is limited transparency between utility companies and consumers causing the utilities to operate as monopolies and taking advantage of consumers.

Additionally, the existing solutions do not scale on a range of more than few thousands consumers with limited appliances. Although some work has already been done in terms of scheduling home appliances using intelligent planning for controlling appliances for home automation and optimized power consumption little focus has been steered into managing large data volume in a real-time [10]–[13]. Most of the research on residential energy consumption is based on non-experimental or very small-scale studies with a dataset of a few GBs which is incapable of capturing the actual behavior of larger sample of population. Prior research shows that limited emphasis has been steered towards real time electricity consumption visualization and analysis for homeowners and utilities (community, state, and country level) on a nation-wide scale. Our previous work focused on real time energy consumption visualization platform for single smart home on a small size dataset of one smart meter only across single-node cluster [14]. Big data analytical models are yet to be implemented for managing distributed smart homes energy consumption utilizing open source tools and techniques. Such an analysis can help in bridging the gap between the utilities and consumers while maintaining the energy demand-response ratio. And in order to fulfill this, more rigorous experiments are needed to test the impact of large scale energy consumption visualization and its subsequent effect on energy conservation policies proposed by the utility providers. Additionally, huge fragmentation and diversity in limited sample of homes makes it difficult for the utility providers to target efficiency programs at scale. With these findings in the literature review, we aimed at scaling up our existing monetization platform to one million home owners so that the utility providers can best interpret the energy consumption profiling and provide better objective measures to assist the home owners on cutting down their consumption in real time.

Contribution of the Paper: By enabling high speed distributed computing platform for mass energy data storage and analytics, following outcomes can be achieved with a synergy of consumer-utilities: First, better engagement of utilities with consumers on a very large scale of 1 million community. In today's digital world, with the growing consumer electronics market, scalability is a top priority for consumer electronics products and distributed applications. The evolving customer expectations and lifestyle preferences have escalated the need to devise extensive customer driven solutions that support high scalability and availability. To the best of our knowledge, the literature review in the related work of energy management platforms have been focused predominantly on small scale users which is not subject to validation for scalability (nation-wide) and real time results. A scalable solution ensures that the proposed approach, architecture and results can handle increased number of consumers' energy data (with dynamic growth in number of appliances) with reasonable response time and computational resources, and this has been effectively achieved in the proposed work. Since the information managed by the utilities and visible by the consumers is the same through a unified platform, the utility companies can propose supplier programs and customer engagement facilities on a mass level such as different broadcasting announcements, energy consumption profiling, customer segmentation etc. Since the information managed by the utilities and visible by the consumers is the same through a unified platform, this will help the customers to receive direct help from the service providers that they need and to view utilities as their trusted partner in energy conservation. Thus, these utilities can work in close relation with consumers to provide a holistic load analysis based on the customer's demographic features and provide subsequent recommendation-alert feedback to a scale of a million consumers in real time. Second, with the help of smart meter data from a large population, the utilities can provide monetization results more accurately and target energy saving policies to help larger sect of the consumer population. Thus, the consumers will be empowered to have a better visibility on the amount of energy being used and when, with subsequent decision on energy-saving. Third, consumers can make informed decisions with near-real time

feedback and improve energy conservation with subsequent reduction in carbon footprint and community at large. Moreover, the proposed design and solution for residential areas energy management will also empower the Utility providers in different levels such as community, state and national level with monetization and energy management capabilities via their respective privileges. Sharing some of the electrical grid operational and monetization responsibility among different stakeholders' i.e. home owners, community owners, and state owners will reduce the heavy load on the main utility's control center without compromising on their the major role in the grid development and operation.

To achieve these, the proposed work emphasizes on synthetically generating smart meter dataset for one million consumers in a distributed residential setting that contains periodic energy consumption data from multiple appliances followed by ad-hoc query analysis. Scaling up to a 4-nodes distributed file system storage and processing cluster [15] using the commodity hardware, a performance analysis experiment is conducted utilizing two open source big data processing engines for achieving an optimized querying, visualization platform for efficient monitoring and managing energy use.

Typically, energy big data can be deemed as the large volume of datasets beyond the technology's capacity to manage, store and process. With the help of distributed file storage and computing cluster, the utilities can conduct consumer load profiling and load analysis on a large scale of 1 million customers which helps in providing the end user/consumer a more accurate representation of how much energy they are consuming with greater granularity and with respect to their community neighborhoods as well. This facilitates real time monetization to all the one million home owners with minimal response time and high aggregated network throughput. Having achieved this, the responsiveness of the results for energy consumption analytic and monetization by homeowners will not be compromised irrespective of the increased number of consumers or increased number of home appliances. From previous studies [16], [17], it is observed that predictions on human consumption behavior varies extensively over socio-demographic situations due to which there are limits to generalization in research findings. Thus, the proposed model allows the home owners to also compare their individual energy consumption with their friends and community consumption. This serves as a nudge in facilitating the consumers to change their energy consumption behavior accordingly. The competition with the peers can persuade them into thinking how to reduce their consumption and energy bill more seriously [18], [19].

Checking the patterns of community electricity consumption, the consumers can proactively schedule their home appliances operation to avoid operating it during the peak hours which leads to reduction in their energy bill. Conventional querying continues to be the most popular query language for big data analysis. Considering this, two types of big data parallel processing engines characterized by disk caching [20] and in-memory [21] caching are utilized to determine an efficient platform that could be used to provide ad-hoc querying and visualization. In order to design an exhaustive set of queries for homeowners and utility providers, smart meter data was modeled as a cube to imitate cube-related operations into big data queries. This dimensional modeling of smart meter data allows instantiating queries that are optimized for high performance, i.e. more efficient real time results. Since we need analytics and monetization for one million home owners, it is imperative that we adopt an optimized schema to boost query performance to the home owners using dimensional modeling in contrast to the traditional ER modelling. Dimensional modelling allows writing good performance queries for customized reporting by joining different multi-dimensional structures to provide more intuitive queries on individual home energy consumption with respect to neighborhood [22]. The visualization results in the form of graphs, charts, and tables are rendered to a scale of one million homeowners and utility providers in real time through a web-based querying interface [23]. The querying engine for energy consumption visualization can be selected by the consumers and utility providers based on performance analysis results and recommendations.

The rest of the paper is organized as follows: In section 2, system architecture is presented. In section 3, a synthetic simulation is designed to generate smart meter data for one million homes. In section 4, a data modeling algorithm is proposed to build queries. A distributed file system storage cluster is implemented to host one million smart meters' big data in section 5. Section 6 elaborates on the visualization stakeholders for the proposed system. Section 7 represents the evaluation criteria and experimental objectives for performance analysis of two big data processing (or querying) engines. In section 8, results are analyzed and compared with existing big data processing techniques. Summary and future work is presented in the conclusion section 9.

II. SYSTEM REQUIREMENTS

Requirement is an important part of a system design. The proposed system deals with big data that incorporates one million energy smart meters in a residential area. The system requirements include a scalable distributed file system cluster with dynamic addition or deletion of nodes, automated load balancing, and high redundancy. The non-functional requirements include powerful commodity machines in terms of RAM and speed to perform CPU intensive querying, low latency, an intuitive query interface for end users. To satisfy the above-mentioned requirements, a system architecture has six building blocks described shown in Fig. 1.

1) The big data sources block: The data in this block is aggregated from multiple home appliances via smart meters. The data is read once every 30 minutes.
2) The big data-modeling block: The data-modeling algorithm used in this research is Online Analytic Processing (OLAP) [24] for optimized query design.

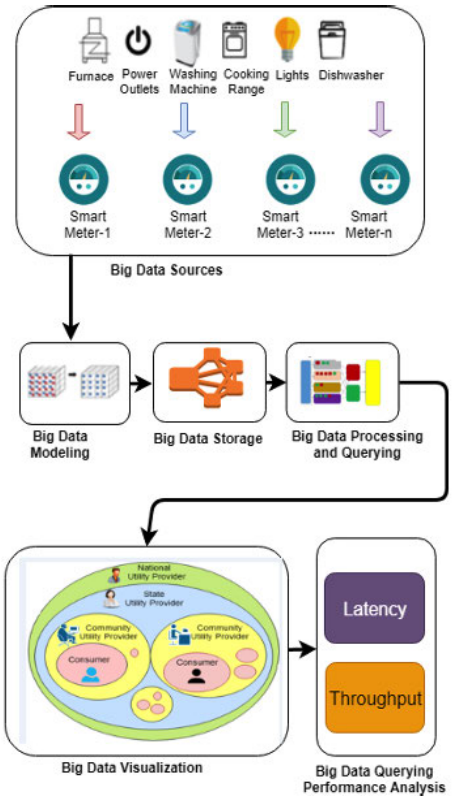

FIGURE 1. System architecture.

3) The big data storage block: The collected big data from smart meters will be stored in a distributed file system cluster of multiple nodes for parallel processing.
4) The big data processing and querying block: Two big data processing engines will be used to execute queries and their respective performance analysis. These processing engines share a common programming design of map-reduce [25] in parallel across the cluster nodes.
5) The big data visualization block: Visualization capabilities will be provided from smart meter data to consumers and utility provider stakeholders namely; community, state, and national level. A query interactive interface is provided on top of cluster to render query results in the form of graphs, charts, tabulated reports.
6) The big data querying performance analysis: Performance analysis experiments will be conducted to evaluate the two big data processing engines in terms of latency and throughput. These experiments will provide an insight into how well the two processing engines can scale up with the increase in data volume.

III. SYSTEM DATASET GENERATION

An energy smart meter real-time dataset for one year was obtained online from research data center at University of Massachusetts [26]. The dataset contains energy consumption for ten home appliances recorded every thirty minutes for one year 2014-2015. From this data, an auto-regressive moving average (ARIMA) statistical technique was applied to generate time series energy consumption data for a million houses [27]. In ARIMA, the underlying process assumes that the predicted value of a variable is a linear function of some previous observations and randomized errors as shown in (1),

$$Y_t = c + \phi_1 y_d + \phi_p y_d + \cdots + \theta_1 e_{t-1} + \theta_q e_{t-q} + e_t \quad (1)$$

where $\phi_1, \phi_2, \theta_1, \theta_2$, are model parameters, p and q are parameters for number of autoregressive and number of moving average terms respectively, y_d is differenced d times between current and previous value, e_t is random error at time t, and c is a constant. The variable d represents the degree of differencing used for stationarizing the given time series dataset. Differencing a series involves simply subtracting its current and previous values d times. Differencing is used to stabilize the series when the stationarity assumption is not met. An Auto Arima R package [27] was used to implement best fitted ARIMA model. For building an optimal ARIMA model, a search is conducted using combinations by selecting the set of (p, d, q) values that optimizes the model fit criteria. Three fit criteria namely; Akaike Information Criterion (AIC), Akaike Information Criterion Corrected (AICc) and Bayesian Information Criterion (BIC) are used to choose the best fit ARIMA model for short term energy consumption prediction. In other words, AIC, AICc and BIC are measures of goodness-of-fit or estimators of prediction error for a given set of data. These measures are used for model selection to estimate the relative quality of given statistical models derived from the dataset. In our experiments as shown in Table-2, ARIMA (2, 1, 3) is considered the best model to make energy consumption forecasts for house appliances as it yields the lowest values for AIC, AICc and BIC. According to the UAE population statistics [28], the one million smart meters' data is divided into the seven UAE Emirates (States). Each state is divided into several communities wherein each community has many houses. For example, in the proposed study, house 1 (houseID-MH1), house 2 (houseID-MH2) are selected in Maliha community and house 3 (houseID-DH1) in Dasman community.

TABLE 2. Statistical measure results for different ARIMA fitted models.

ARIMA parameters (p,d,q)	Akaike Information Criterion (AIC)	Akaike Information Criterion Corrected (AICc)	Bayesian Information Criterion (BIC)
(0,1,0)	1596.29	1596.65	1598.23
(1,1,0)	1585.27	1585.39	1593.89
(1,1,1)	1569.34	1569.56	1578.25
(2,1,1)	1581.56	1581.99	1590.78
(2,1,3)	1536.51	1535.71	1548.42

IV. SMART METER BIG DATA MODELING

The smart meters' big data for houses is modeled in the form of a cube. OLAP operations were executed on top of this cube to render consumption query results to different levels of stakeholders. The cube was represented using

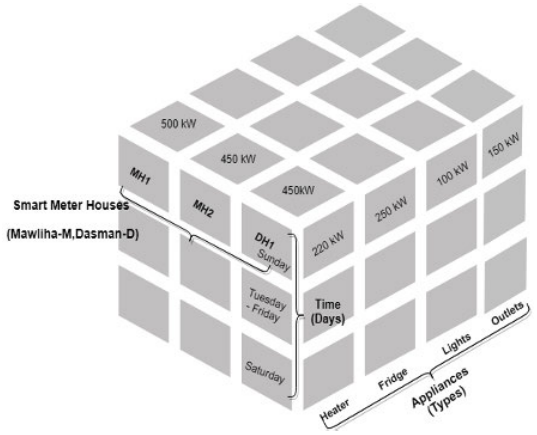

FIGURE 2. OLAP Cube for smart meter dataset.

three dimensions, namely, location (House IDs), time (Days), and appliances (Type). Each dimension was categorized into attributes. MH1, MH2, and DH1 represented house IDs, Time was represented in days, and the dimension appliance represented different types of appliances; Heater, Fridge, Outlets, Washing Machine, Cellar, Furnace, Lights. The cells of the cube were populated with power consumption values with respect to each dimension as shown in Fig. 2. Four operations are performed on OLAP cube; rollup, drill down, slicing, and dicing. For instance, rollup operation on the dimension location is performed to represent aggregated consumption from individual houses to total power consumption of all houses in a community.

V. DISTRIBUTED FILE SYSTEM CLUSTER DESIGN

A 4-nodes cluster was constructed and used to store the generated one million meters' data in a distributed fashion and perform parallel processing across nodes. The processing was conducted using one node, two nodes, three nodes, and four nodes respectively. Functions of nodes is described as follows.

1) *Master Node (Name node).* Distributed file system had one centerpiece machine (master) called a name node server that stored and managed the metadata for cluster. It is the directory for file blocks stored across data nodes. It ran a job tracker process to assign tasks to slave nodes.
2) *Slave nodes (Data nodes).* The slave nodes in a cluster are called data nodes. These data nodes stored the datasets and performed read/write operations on query execution from the client application.

Four conventional Intel-based computers were used setting up a cluster consisting of one master and four slave nodes as shown in Fig. 3. The following configuration parameters were set in designing the cluster.

1) The replication factor for each node was set to two for data duplication and each block size was 64MB.
2) For a million smart meters' dataset (1.5TB), the name node generates 23438 blocks (1.5TB/64MB) of data and distributed it amongst data nodes for storage and process.
3) Each data node stored 5860 blocks that added to 375GB.
4) The block distribution is such that block B1 to B5860, B5861 to B11720, and B11721 to B17580, and B17581 to B23440 were assigned across node 1, node 2, node 3, and node 4 respectively. It is worth mentioning that the cluster can be scaled up and more nodes can be added easily.

VI. BIG DATA QUERYING AND VISUALIZATION

The stored big data visualization and performance analysis were done using open-source commodity software on top of the distributed file system storage cluster. Fig. 4. demonstrates underlying map-reduce algorithm used to generate aggregated power consumption of two state communities, namely, Maliha and Dasman on a quarterly basis. This example illustrates how power consumption from each device of every house in the community was stored on the cluster data nodes followed by splitting and extraction of total power consumption data within each community using map-reduce. Four stakeholders are deemed important for real time visualization on smart homes energy consumption as follows:

1) Consumer: A homeowner is entitled to view the power consumption of all devices in the house with respect to time on a daily, weekly, monthly, and yearly basis and compare it with community's total consumption.
2) Community energy utility provider: A community utility provider monitors the aggregated power consumption of each household in the community. The analysis will help in identifying trends in energy consumption of each household that can help determine the peak load hours and plan accordingly.
3) State energy utility provider: A state utility provider can supervise the cumulative power consumption of all communities within its respective state.
4) Country energy utility provider: A country utility provider is the highest level of authority in the hierarchy. The cumulative power consumption from all states can be compared to the total power generation from the Central power station. This helps the utility providers to prioritize energy saving strategies and execute data driven energy actions accordingly.

VII. EVALUATION CRITERIA

Performance is critical in distributed file system cluster whether it is deployed on bare metal. The cluster was deployed on Intel-based machines. Three types of variables are used for evaluating cluster performance: input, output, and control variables as shown in Table 3.

The control variables are based on the physical environment and machine specifications as mentioned in the previous section. The input variables were the optimization parameters that can be controlled by the user during queries execution to optimize the performance. Two output variables were

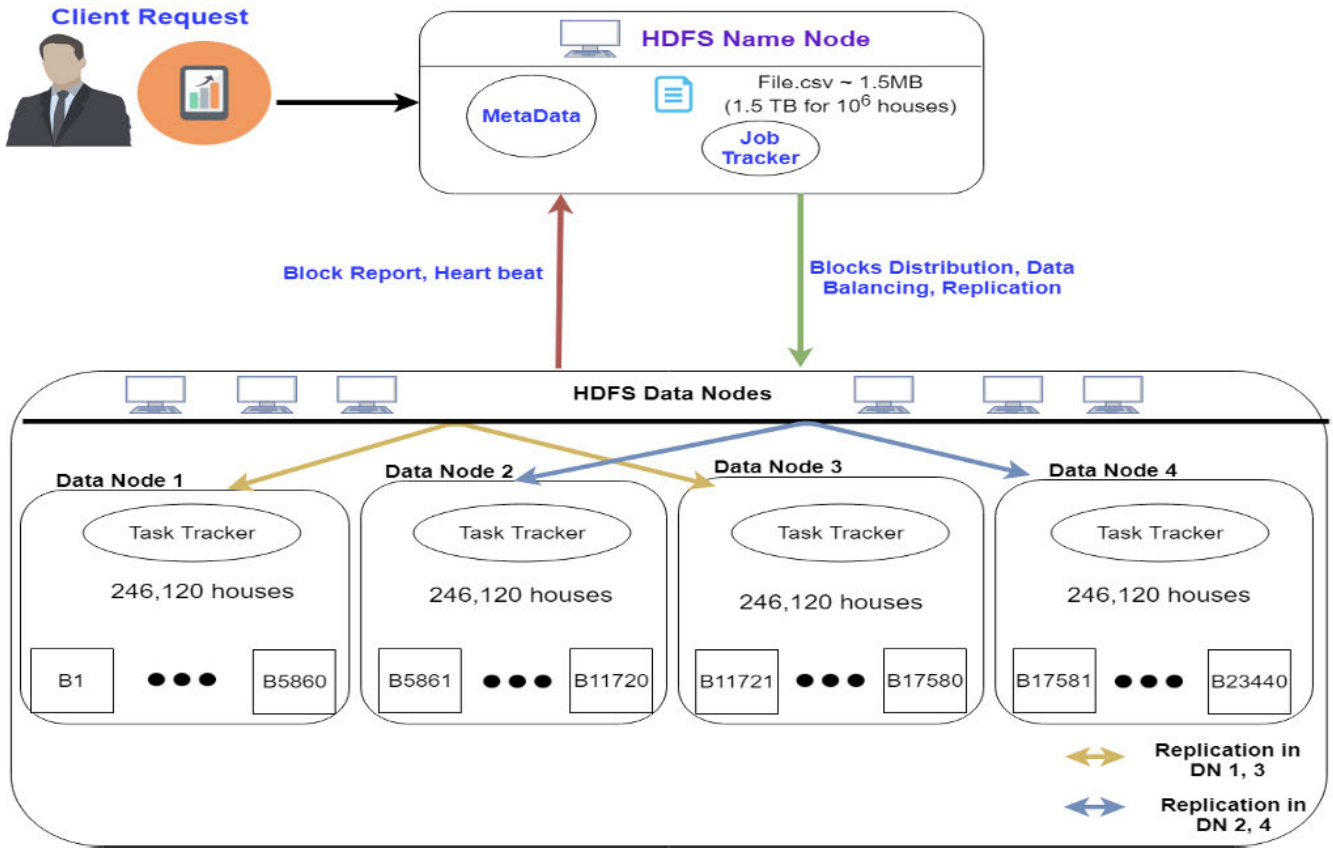

FIGURE 3. A conceptual model for four-nodes cluster.

TABLE 3. Experimental variables under study.

Input Variables	Control variables	Output variables
# Nodes	Cores	Latency
# Data files	Memory size	Throughput
# Queries	Network bandwidth	

evaluated by varying input variables; latency and throughput. Latency is the completion time of rendering query results across the cluster data nodes. Similarly, 'throughput' denotes the number of reads/writes completed per unit time. For input variables, the smart meter data files were imported in log scale of 10, 100, 1000, 10000, 100000, and 1,000,000 in CSV format. Two experiments are designed to study the effect on output variables of two big data querying engines characterized by in-memory and disk caching.

(A) *Experimental Objective I:* To determine the total latency for data querying, each query was scheduled to run 100 times using a scheduled workflow tool [25] on top of the distributed file system of the cluster.

(B) *Experimental Objective II:* To determine the processor throughput, data querying throughput was based on the file size and the elapsed time to do so. Two test cases are generated to determine the elapsed time (latency) and throughput for executing queries as follows.

- Number of smart meter files- The impact of data size during querying was determined by measuring the elapsed time (latency), throughput for each batch of data.
- Number of data nodes- The impact of the cluster size on querying was determined by measuring the latency, throughput, for submitting query jobs across one, two, three, and four data nodes. Each query workflow was scheduled to run every fifteen minutes and for each query execution, hundred points of latency were logged for statistical significance.
- Latency- The Distributed file system cluster's mean execution (querying) time was evaluated by varying input variables, such as the number of active slave nodes and dataset size. A query job is executed in multiple stages wherein each stage contained several map reduce tasks as represented in (2).

$$J = \{St_i : 0 < i < M\}, \quad St = \{Tsk_j, j : 0 < j < N\} \quad (2)$$

Here, M is the number of stages in a job and N is the number of map-reduce tasks in a stage. The resource manager of master node distributed these stages across the cluster nodes. The map-reduce task processed assigned to each stage are executed in parallel across the assigned data notes of the cluster. Latency (L) corresponded to the total execution time taken by all the

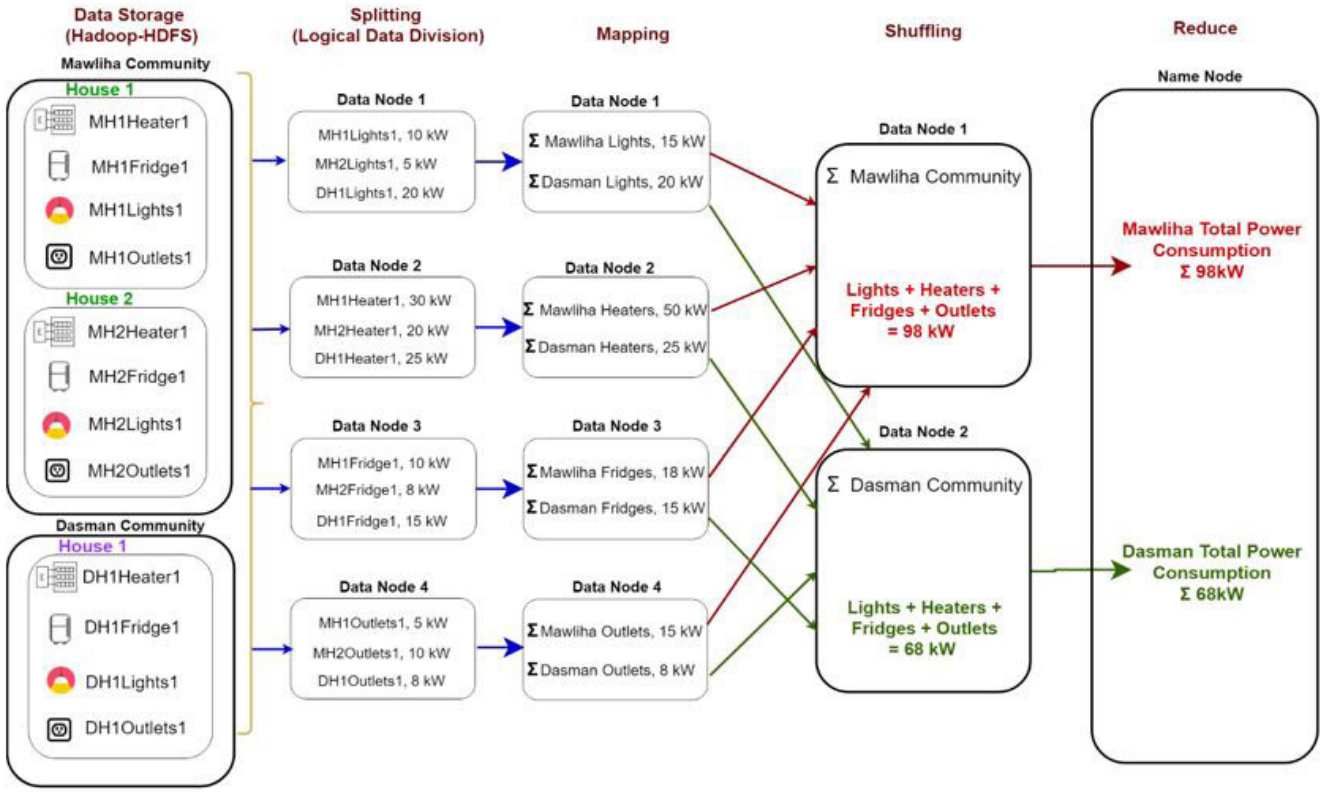

FIGURE 4. A map reduce programming model.

mapper-reducers tasks within each stage when run in parallel to process a set of queries for each stakeholder.

$$L = \sum_{i=0}^{M} St_i \sum_{j=0}^{N} Tsk_j \tag{3}$$

- *Throughput.* Throughput refers to the amount of data executed, per second, for each query execution. The expression used to calculate throughput was based on input file size and latency measurements is shown in (4).

$$R = \frac{NF}{L} \times S \tag{4}$$

R: Throughput (MB/S), NF: Number of Smart Meter Files
L: Latency (Secs), S: Size of 1-smart meter file (MB)

VIII. IMPLEMENTATION AND RESULTS

The proposed hardware cluster was built and software algorithm was developed for validation and testing. The results are divided into three subsections; visualization, quantitative evaluation based on latency and throughput, and comparative analysis with existing data solutions.

A. VISUALIZATION

For visualization purpose, we sampled 10 smart meters' data from the generated dataset. An open-source visualization tool [23] on top of distributed file system was used for visualization. Eighteen queries are constructed to enable the user with graphical visualization on consumption per device, per day, per month, and per year, (queries 1-5 are consumer queries, queries 6-10 are for community utility provider, queries 11-14 are for state utility provider, and queries 15-18 are for county utility provider).

Queries have four privileges according to the stakeholders; homeowner, community, state, and country operator. These results are useful to allow consumers to monitor and operate home appliances efficiently while comparing it with their respective neighborhood. Such graphs are helpful for utilities to understand the status of the grid. The results for each query is illustrated in the following figures (Fig. 5 - Fig. 12) categorized according to four stakeholder levels. Different stakeholder queries are executed on big data distributed file system visualization tool to generate graphs and tabular data corresponding to each stakeholder. Home consumers and utility providers can gain useful insights into the periodic consumption trend of different home appliances, houses, communities, states, and country at large monitoring these visualization graphs and charts from ad-hoc querying. Fig. 5 - Fig. 7 are energy consumption graphical representation for consumers. Fig. 8 - Fig. 9 demonstrate energy consumption graphs for all houses within a community vicinity for the Community utility provider view.

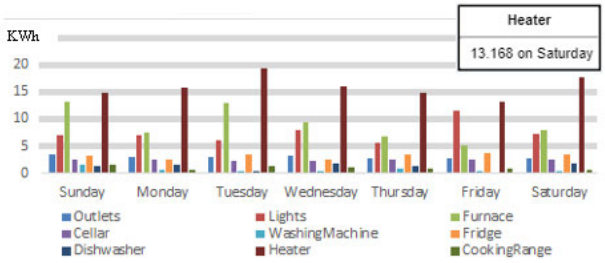

FIGURE 5. Total consumption of each appliance for consumer's house every day in a week.

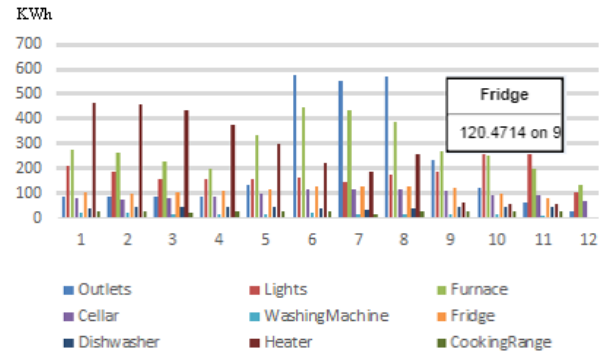

FIGURE 6. Total consumption of each appliance for consumer's house on a monthly basis.

myhouseConsumption.percent	b.totalofmyHouse [KWh]	b.totalofmyCommunity [KWh]
23.46	14440.95	60585.28

FIGURE 7. Annual consumption in percentage with respect to the community's total consumption (Consumer).

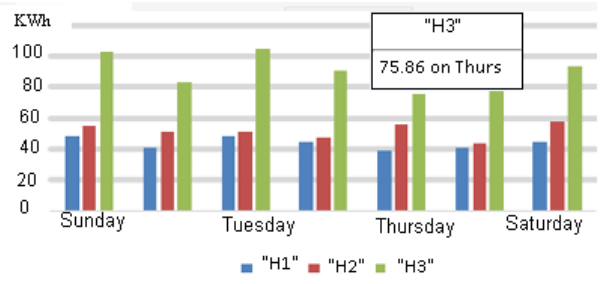

FIGURE 8. Total consumption of each house in a community for everyday/week.

Fig. 10 - Fig. 11 illustrate State utility provider view for energy consumption of all communities within a state. Fig. 12 represents the Country utility view for energy consumption graphs of each state (Emirate).

B. QUANTITATIVE EVALUATION

In this subsection, a quantitative evaluation for the cluster latency and throughput is tested between two big data map-reduce processing tools; disk caching (main memory) processing tool and in-memory processing (secondary memory) tool. The result is obtained as an average of executing each query hundred times in a scheduled workflow.

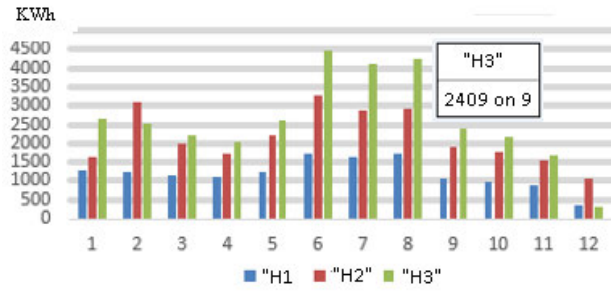

FIGURE 9. Total consumption of each house in a community every month/year.

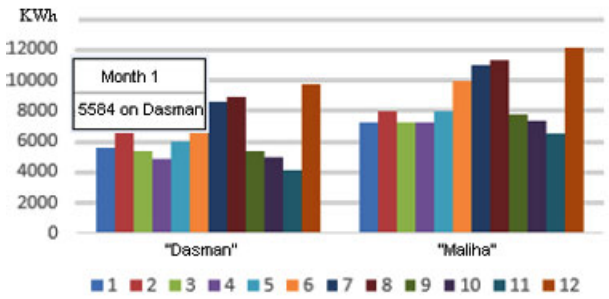

FIGURE 10. Total consumption of each community of a state on monthly basis.

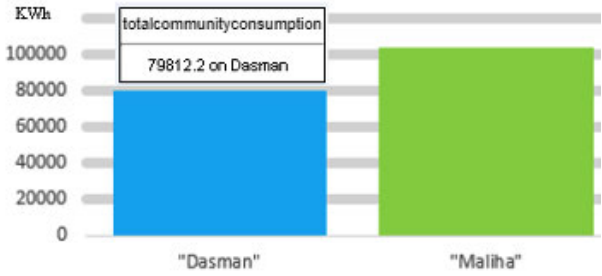

FIGURE 11. Total annual consumption of each community of a state.

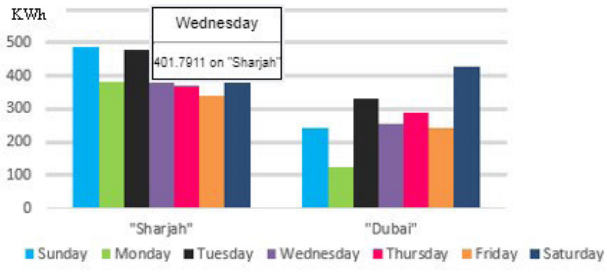

FIGURE 12. Total consumption of each state of a country each day/week.

1) LATENCY

Fig. 13 and Fig. 14 depict Query-wise mean execution time for disk caching and in-memory caching across one node and four nodes respectively for one million smart meter dataset representing energy consumption of 10 home

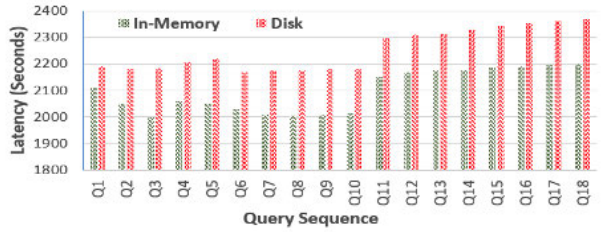

FIGURE 13. Mean latency per query for 1 million smart meters across 1-Node.

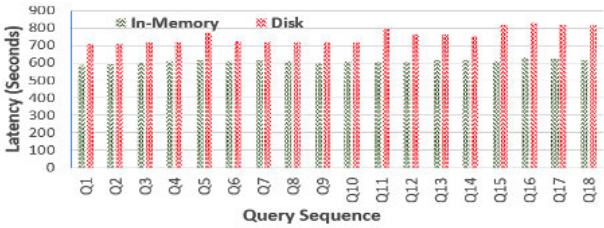

FIGURE 14. Mean latency per query for 1 million smart meters across 4-Nodes.

appliances every 30 minutes spanned through one year from 2014-2015. It is observed that the query execution time largely depended on the query selectivity characteristic. For queries formulated with large selectivity parameters took longer time to process than other queries for the same processing volume across each node. For example, Query 11 took a longer execution time because of multiple JOIN clauses used in this query to find the total house consumption and its respective neighborhood community consumption. Due to multiple JOINs, there is a strong interdependency between the files. With larger interdependency between smart meters' dataset, more numbers of reducers strongly dependent on each other. With increase in the number of reducers, the execution time increased sharply depending on varying computing needs. The maximum time is taken by Query 15 to Query 18 as these queries had a wider selectivity requirement to aggregate records for national-utility provider. It is clear that a 4-Node based solution can run even the most complex queries on a million meters in less than 15 minutes. Adding more nodes will certainly reduce this time proportionally. Fig. 13 and Fig. 14 show that the best processing performance for a million meters is achieved with a cluster size of four nodes. Thus, for processing a larger batch of files, the addition of nodes to the cluster has a significant impact on reducing the execution time. The times to execute is reduced by a factor of one-third when scaling up from one to four nodes. Fig. 15 shows that regardless of the number of nodes, the latency increases log-linearly. For example, for 2-Nodes, increasing the number of meters from 100,000 to 1000,000 increases average latency from 1400 to 1800 seconds only. Slopes of log-linear line is sharper for 1-node as opposed to higher number of nodes. Additionally in each cluster size of Fig. 15, it is observed that the slope is clearly higher with disk caching in contrast to the in-memory caching. This implies that the rate of change of latency with

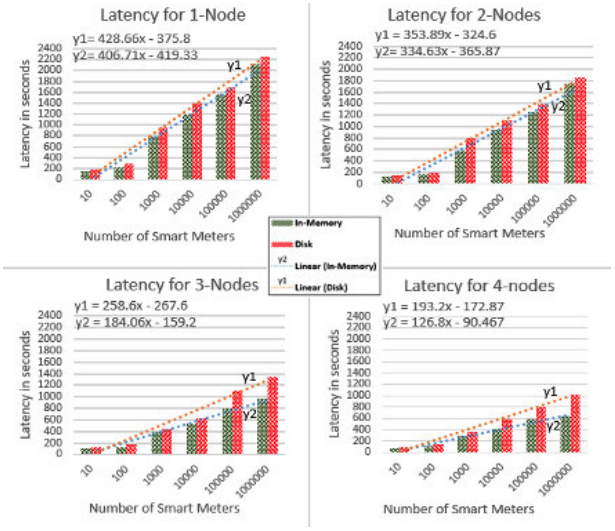

FIGURE 15. Mean latency across 1-Node, 2-Nodes, 3-Nodes, and 4-Nodes.

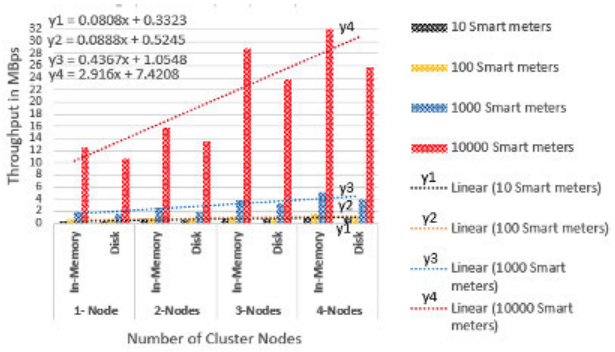

FIGURE 16. Mean Throughput for 10, 100, 1000, 10,000 smart meters data across 1, 2, 3, and 4 nodes.

increasing data volume from 10 to 1000,000 smart meters file is faster for querying on the disk-caching engine than the in-memory caching engine.

2) THROUGHPUT

Processor throughput is calculated from latency points wherein it is inferred that queries with larger latency had smaller throughput. Mean throughput across each cluster size for a batch of 10, 100, 1000, and 10,000 smart meters dataset is shown in Fig. 16.

Fig. 17. illustrates the mean throughput for 100,000 and 1,000,000 smart meters dataset. From the two figures, it is observed that the throughput is higher across the cluster size of 4-nodes in contrast to 1-node cluster. It can be deduced from the figures that the slopes for a large dataset volume of 1,000,000 smart meters is sharper than a small dataset size of 10 smart meters. Due to this, the throughput increases more rapidly with increasing number of cluster nodes for a larger dataset than a smaller dataset. Additionally, for a set of one million smart meters data queried in a cluster size of 4-nodes, a maximum throughput of roughly 2300MBps is

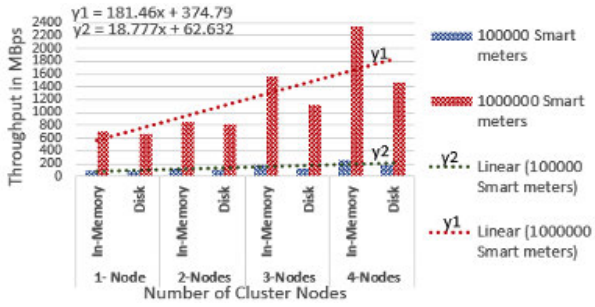

FIGURE 17. Mean Throughput for 100,000 and 1,000,000 smart meters data across 1, 2, 3, and 4 nodes.

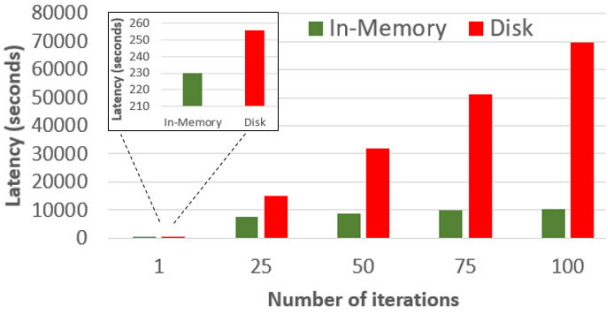

FIGURE 18. Iterative querying performance of in-memory and disk caching engines for one-million smart meters in 4-nodes cluster.

TABLE 4. Comparison of proposed model with existing tools.

Criteria	Big Data Tool	RDMS	In-memory caching engine	Data caching engine
Runtime	25s to 6min.	2-7hrs.	12-15min.	28-34min.
Storage	350 GB	1.3 TB	1.5 TB (w/o replication)	1.5 TB (w/o replication)

achieved for in-memory parallel processing in contrast to a throughput of 1400 MBps with disk based processing. This is accounted to the fact that with the addition of four nodes the processing time decreases substantially for executing queries on the same volume of smart meters dataset. Additionally, as the dataset size increases for the same set of nodes and the same processing engine type, the throughput grows in a monotonically increasing fashion. This result matches the theoretical expectation as the latency across four nodes is smaller in comparison to one node, and hence higher throughput.

Overall, in-memory caching outperforms disk caching as the former offers a performance boost in querying one million smart meters dataset. This is largely applicable in an iterative querying framework when storing the input data in-memory benefits the in-memory caching latency in contrast to disk caching. However, the performance of both in-memory and disk caching processing engines is comparable to each other in non-iterative querying.

For example, as shown in Fig. 18, it is observed that for in-memory and disk based processing engines the execution time for the first query run is comparable to each other. However, in iterative querying the in-memory engine is observed to have performed almost 10 times better than disk caching engine iteratively. This is attributed to the fact that in the first iteration the in-memory engine fetches the data on to the memory and utilizes the cached memory data in subsequent iterations which reduces the response time. On the other hand, the disk processing engine spills the data over to the disk in each iteration consecutively without any significant reduction in the response time of subsequent query iterations.

Selection of an appropriate big data processing engine in reality is subjective. Even though in-memory caching tool provides in-memory computation in iterative querying that fosters low latency, the memory constraint and limited network bandwidth should be taken into consideration. Disk caching tool is recommended for processing and analysis if the available hardware resources are limited. For processing small datasets on a small cluster, disk caching is a good choice to obtain stable query response without any out-of-memory exceptions. Moreover, the execution times for writing a few MBs on disk or in-memory do not have much difference so disk caching is recommended for small datasets in a cluster. Alternatively, in a large size cluster for processing medium to large datasets, in-memory processing engine is advised.

C. COMPARATIVE ANALYSIS WITH EXISTING BIG DATA SOLUTIONS

An IT based multinational company's proprietary big data tool and a RDMS from [5] were used as a benchmark to compare the processing time for running queries on one million smart meters' data. Table 4 provides a synopsis of performance comparison between big data in-memory caching engine and disk caching engine with respect to proprietary tool and RDMS. From Table 4, it is observed that for a million smart meters processing, the proposed system outperformed the traditional RDMS system by 20 times. On the other hand, the proprietary processing tool is better than the proposed model. However, the one million smart meter data used for querying on the proprietary tool is only 350 GB in size whereas the data queried utilizing the proposed processing paradigms is 1.5TB in size. Thus, it can be claimed that for a volume of 1.5 TB smart meters dataset, the proprietary tool could potentially take longer processing time than the proposed model that relies on open source big data processing engines. All in all, it can be inferred that a large data processing procedure which takes hours of processing time on a centralized relational database might take roughly 15 minutes when the same data is distributed across distributed file system cluster nodes with parallel processing querying.

IX. CONCLUSION

One million residential area energy smart meter data is synthesized for one million house smart meters using a one-year real dataset. The meters data were geographically distributed

into the seven UAE Emirates residential areas based on the latest population percentage in each Emirate.

Our proposed system is designed and developed to empower the stakeholders to visualize individual homes, community, state, and country energy consumption at each level. Using off-the-shelf commodity hardware we were able to achieve a maximum of less than 15 minutes of querying time while generating reports for stakeholders on a variety of levels. It is worth mentioning that the proprietary system is not an open-source platform and used specialized cluster resources that cost more money compared to our open source commodity hardware cluster and open source software tools. It is recommended that further study be conducted to increase the cluster nodes and perform data mining and monetization.

ACKNOWLEDGEMENT

This paper represents the opinions of the author(s) and does not mean to represent the position or opinions of the American University of Sharjah.

REFERENCES

[1] H. Bhosale and D. Gadekar, "A review paper on big data and Hadoop," *Int. J. Sci. Res.*, vol. 4, no. 10, pp. 1–7, 2014.

[2] E. Pan, D. Wang, and Z. Han, "Analyzing big smart metering data towards differentiated user services: A sublinear approach," *IEEE Trans. Big Data*, vol. 2, no. 3, pp. 249–261, Sep. 2016.

[3] T. Jiang, Y. Cao, L. Yu, and Z. Wang, "Load shaping strategy based on energy storage and dynamic pricing in smart grid," *IEEE Trans. Smart Grid*, vol. 5, no. 6, pp. 2868–2876, Nov. 2014.

[4] J. Han, C.-S. Choi, W.-K. Park, I. Lee, and S.-H. Kim, "Smart home energy management system including renewable energy based on ZigBee and PLC," *IEEE Trans. Consum. Electron.*, vol. 60, no. 2, pp. 198–202, May 2014.

[5] *Managing Big Data for Smart Grids and Smart Meters*, IBM Corp., Endicott, NY, USA, May 2012.

[6] D.-M. Han and J.-H. Lim, "Smart home energy management system using IEEE 802.15.4 and zigbee," *IEEE Trans. Consum. Electron.*, vol. 56, no. 3, pp. 1403–1410, Aug. 2010.

[7] D.-M. Han and J.-H. Lim, "Design and implementation of smart home energy management systems based on zigbee," *IEEE Trans. Consum. Electron.*, vol. 56, no. 3, pp. 1417–1425, Aug. 2010.

[8] S. S. Refaat, H. Abu-Rub, and A. Mohamed, "Big data, better energy management and control decisions for distribution systems in smart grid," in *Proc. IEEE Int. Conf. Big Data*, Dec. 2016, pp. 3115–3120.

[9] (2012). *A Regulator's Privacy Guide to Third-Party Data Access for Energy Efficiency*. [Online]. Available: https://www4.eere.energy.gov/seeaction/system/files/documents/

[10] Y.-S. Son, T. Pulkkinen, K.-D. Moon, and C. Kim, "Home energy management system based on power line communication," *IEEE Trans. Consum. Electron.*, vol. 56, no. 3, pp. 1380–1386, Aug. 2010.

[11] H.-C. Jo, S. Kim, and S.-K. Joo, "Smart heating and air conditioning scheduling method incorporating customer convenience for home energy management system," *IEEE Trans. Consum. Electron.*, vol. 59, no. 2, pp. 316–322, May 2013.

[12] G. Mokhtari, A. Anvari-Moghaddam, and Q. Zhang, "A new layered architecture for future big data-driven smart homes," *IEEE Access*, vol. 7, pp. 19002–19012, 2019.

[13] (2018). *Google Powermeter*. [Online]. Available: https://sites.google.com/site/powermeterpartners/google-meter-api

[14] A. R. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta, and M. Alikarar, "A smart home energy management system using IoT and big data analytics approach," *IEEE Trans. Consum. Electron.*, vol. 63, no. 4, pp. 426–434, Nov. 2017.

[15] S. G. Manikandan and S. Ravi, "Big data analysis using apache Hadoop," in *Proc. Int. Conf. IT Converg. Secur. (ICITCS)*, Oct. 2014, pp. 700–703.

[16] E. R. Frederiks, K. Stenner, and E. V. Hobman, "Household energy use: Applying behavioural economics to understand consumer decision-making and behaviour," *Renew. Sustain. Energy Rev.*, vol. 41, pp. 1385–1394, Jan. 2015.

[17] W. Abrahamse and L. Steg, "Factors related to household energy use and intention to reduce it: The role of psychological and socio-demographic variables," *Human ecology Rev.*, vol. 12, pp. 30–40, Oct. 2011.

[18] I. Ayres, S. Raseman, and A. Shih, "Evidence from two large field experiments that peer comparison feedback can reduce residential energy usage," *J. Law, Econ., Org.*, vol. 29, no. 5, pp. 992–1022, Oct. 2013.

[19] T. G. Papaioannou, D. Kotsopoulos, C. Bardaki, S. Lounis, N. Dimitriou, G. Boultadakis, A. Garbi, and A. Schoofs, "IoT-enabled gamification for energy conservation in public buildings," in *Proc. Global Internet Things Summit (GIoTS)*, Jun. 2017, pp. 1–6.

[20] V. Garg, "Optimization of multiple queries for big data with apache Hadoop/Hive," in *Proc. Int. Conf. Comput. Intell. Commun. Netw. (CICN)*, Dec. 2015, pp. 938–941.

[21] I. Stocia, "Conquering big data with spark," in *Proc. IEEE Int. Conf. Big Data*, Oct. 2015. p. 3

[22] N. H. Liyanage, "Advanced Query model design concept to support multi-dimensional data analytics for relational database management systems," in *Proc. Int. Conf. Big Data Analytics Comput. Intell. (ICBDAC)*, Mar. 2017, pp. 432–435.

[23] N. Sirisha and K. V. D. Kiran, "Stock exchange analysis using Hadoop user experience (Hue)," in *Proc. Int. Conf. Intell. Sustain. Syst. (ICISS)*, Dec. 2017, pp. 1141–1144.

[24] K. Dhanasree and C. Shobabindu, "A survey on OLAP," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Dec. 2016, pp. 1–9.

[25] M. Bhandarkar, "MapReduce programming with apache Hadoop," in *Proc. IEEE Int. Symp. Parallel Distrib. Process. (IPDPS)*, 2010, pp. 1–4.

[26] T. Weibel. *Umasstracerepository*. Accessed: Mar. 30, 2020. [Online]. Available: http://traces.cs.umass.edu/index.php/Smart/Smart/

[27] R. Hyndman and Y. Khandakar, "Automatic time series forecasting: The forecast package for R," *J. Stat. Softw.*, vol. 27, no. 3, 2008. [Online]. Available: https://www.jstatsoft.org/article/view/v027i03

[28] G. Blogger. (2018). *UAE Population Statistics in 2018 (Infographics)|GMI*. [Online]. Available: https://www.globalmediainsight.com/blog/uae-population-statistics/

RAGINI GUPTA received the B.S. and M.S. degrees in computer engineering from the American University of Sharjah, UAE, in 2016 and 2018, respectively. She is currently pursuing the Ph.D. degree in computer science with the Missouri University of Science and Technology, MO, USA. Her research interests the span across the IoT, big data, consumer electronics, and wireless sensor networks.

A. R. AL-ALI (Life Senior Member, IEEE) received the B.Sc. (EE) degree from Aleppo University, Syria, in 1979, the M.S. degree from the Polytechnic Institute of New York, USA, in 1986, and the Ph.D. degree in electrical engineering and a minor in computer science from Vanderbilt University, Nashville, TN, USA, in 1990. From 1991 to 2000, he was a faculty with the EE Department, KFUPM, Saudi Arabia. Since 2000, he has been working as a Professor of computer engineering with the American University of Sharjah, UAE. His research, teaching interests include embedded systems, cyber physical systems, and IoT and IIoT0 applications in smart cities.

IMRAN A. ZUALKERNAN (Member, IEEE) received the B.S. (Hons.) and Ph.D. degrees from the University of Minnesota, Minneapolis, USA. He has been a faculty member with Pennsylvania State University and University of Minnesota. He has also practiced as a Principal Design Engineer for high-end robotic applications. He has been involved in technology startup companies as a Chief Technology Officer and as a Chief Executive Officer. He is currently a Professor of computer engineering with the American University of Sharjah. He is on the executive board of the IEEE technical committee on learning technologies. His primary areas of research are embodied and pervasive computation, wearable computing, the IoT, and learning technologies.

SAJAL K. DAS (Fellow, IEEE) received the B.S. degree in computer science and engineering from Calcutta University, India, in 1983, the M.S. degree in computer science from the Indian Institute of Science, Bengaluru, India, in 1984, and the Ph.D. degree in computer science from the University of Central Florida, Orlando, USA, in 1988. He is currently a Professor of computer science and a Daniel St. Clair Endowed Chair with the Missouri University of S&T, MO, USA. His research interests include wireless sensor networks, cyber physical systems, smart environments, distributed cloud computing, big data analytics, and the IoT. He has published extensively in these areas with more than 700 research articles in high quality journals and refereed conference proceedings. He holds five U.S. patents and coauthored four books. His H-index is 82 with more than 28,500 citations according to Google Scholar. He was a recipient of ten Best Paper Awards and awards for teaching, mentoring research including IEEE Computer Society's Technical Achievement Award for pioneering contributions to sensor networks and mobile computing. He serves as the Founding Editor-in-Chief of *Pervasive and Mobile Computing* Journal, and as an Associate Editor of journals including the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS OF MOBILE COMPUTING, and *ACM Transactions on Sensor Networks*.