Electric Power Systems Research 116 (2014) 174–181

Contents lists available at ScienceDirect

Electric Power Systems Research

journal homepage: www.elsevier.com/locate/epsr

A distributed data storage and processing framework for next-generation residential distribution systems

Ni Zhang, Yu Yan, Shengyao Xu, Wencong Su*

Department of Electrical and Computer Engineering, University of Michigan-Dearborn, Dearborn, MI 48128, United States

ARTICLE INFO

Article history:
Received 25 March 2014
Received in revised form 29 May 2014
Accepted 11 June 2014
Available online 28 June 2014

Keywords:
Smart grid
Distribution system
Big data
Cyber-physical systems
Power systems

ABSTRACT

As the number of smart meters/sensors increases to more than hundreds of thousands, it is rather intuitive that the state-of-the-art centralized information processing architecture will no longer be sustainable under such a big data explosion. Hence, an innovative data management system is urgently needed to facilitate the real-world deployment of a future residential distribution system. In this paper, we investigate a radically different approach through distributed software agents to translate the legacy centralized data storage and processing scheme to a completely distributed cyber-physical architecture. We further substantiate the proposed distributed data storage and processing framework on a proof-of-concept testbed using a cluster of low-cost and credit-card-sized single-board computers. Finally, we evaluate the proposed distributed framework and proof-of-concept testbed with a comprehensive set of performance measures.

1. Introduction

The U.S. power grid experienced several massive blackouts during the past 80 years, and the risk of power outages cannot be completely diminished [1,2]. As discussed in [3], under today's power system data storage framework, the power grid outage left investigators stumped since the data lacked the essential time stamp, also after the blackout, some of the historical data cannot be retrieved. These are the main reasons that made the investigators almost impossible to unravel the sequence of events. Thus it is very difficult to improve the power system without these small amount essential data. The emerging high penetration of intermittent renewable energy resources, distributed generators, and distributed energy storage devices may even complicate today's power grid operations, planning and monitoring [4]. A tremendous amount of energy devices will be interconnected to each other in an ultra-large-scale and highly dynamic network. And the amount of data generated by this gigantic amount of energy devices will come to skyrocket. Therefore, there is an urgent need to reconstruct the outdated U.S. power grid infrastructure at all levels (e.g., power generation, transmission, and distribution). In the last decade, the concept of cyber-physical-systems (CPS) has been widely adopted to next-generation power systems which depend upon the synergy of computational and physical components [5]. The ultimate goal is to enhance smart operations of complex power systems that promote economic efficiency, encourage customer participation, reduce operational cost, maintain grid reliability, and support grid integration of renewable energy resources. More recently, the transformative concept of future residential distribution system, which is referred to energy internet, has been proposed to enable the U.S. to take advantage of advances in renewable energy for a secure and sustainable future [6,7]. The vision of energy internet is inspired by the paradigm shift in the Information Technology (IT) industry from 30 years ago. The envisioned energy internet is indeed an internet-like CPS for plug-and-play of millions of power grid components. As illustrated in Fig. 1, with the development of the IT industry, distributed computing (personal computers) plays a major role in the modern society, taking place of the centralized mainframes in the past. Similarly, in the power system area, centralized power generation will give way to highly distributed energy sources; in this way, renewable energy resources will be regarded as important roles in the proposed energy internet. There is a significant amount of research and investigation to be done before the full vision of energy internet comes to fruition. In [7], we explored the electricity market framework for energy internet, which pertains to the CPS features. In our existing framework, we assume that the massive volume of real-time power grid data are collected by a number of meters/sensors and then stored and processed properly. Nevertheless, as the number of meters/sensors increases to more than hundreds of thousands, it is rather intuitive that the state-of-the-art centralized information

* Corresponding author. Tel.: +1 315 528 3344.
E-mail addresses: niz@umich.edu (N. Zhang), yuya@umich.edu (Y. Yan), shengyao@umich.edu (S. Xu), wencong@umich.edu (W. Su).

http://dx.doi.org/10.1016/j.epsr.2014.06.005
0378-7796/

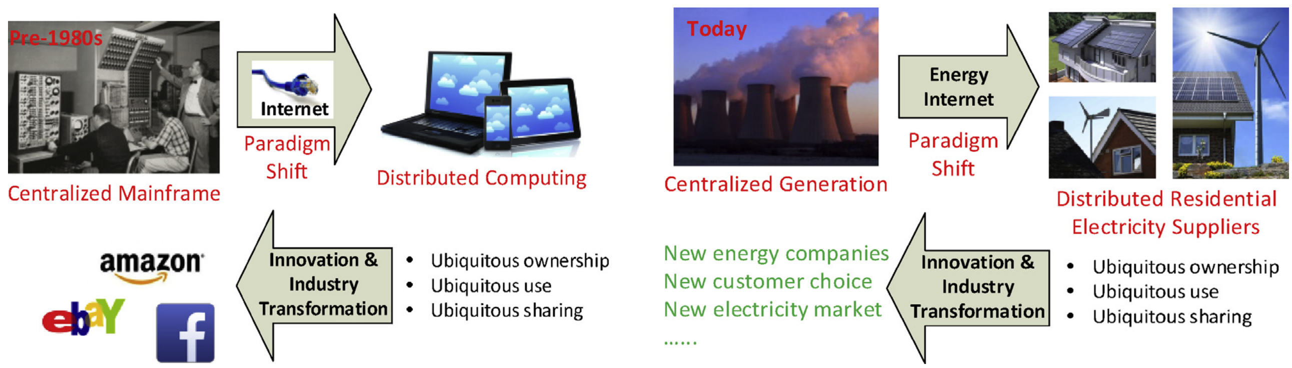


Fig. 1. A paradigm shift: information internet *vs.* energy internet [7].

processing architecture will no longer be sustainable under such a big data explosion. Hence, an innovative data management system is urgently needed to facilitate the real-world deployment of a future residential distribution system.

The major contributions of this paper are summarized as follows:

1. Identify the key features and components of the proposed CPS-like next-generation residential distribution system.
2. Investigate a radically different approach through distributed software agents to translate the legacy centralized data storage and processing scheme to a completely distributed cyber-physical architecture.
3. Substantiate the proposed distributed data storage and processing framework on a proof-of-concept testbed using a cluster of low-cost and credit-card-sized single-board PCs.
4. Evaluate the proposed distributed framework and proof-of-concept testbed with a comprehensive set of performance measures.

The remainder of this paper is organized as follows. Section 2 discusses the limitation of the conventional centralized data storage and processing framework and the proposed CPS-like next generation residential distribution system with its key components, and compares the distributed data storage and processing architecture with the centralized one. Section 3 presents a mechanism of the distributed data storage and processing framework. Section 4 introduces the proof-of-concept testbed followed by a number of case studies. Section 5 summarizes this paper's main findings.

2. Materials and methods

2.1. Next-generation residential distribution system

Next generation residential distribution system is a CPS-like system. The cyber layer includes the computing, communication systems and information network, while the power system lies in the physical layer. Systems in the cyber layer can provide monitoring, management and control services for the physical world, moreover, cyber and physical layers are tightly coupled and they are related to one another [8]. In our proposed CPS-like system, we divided the system into three layers, namely, Operation Layer, Information Network Layer, and Power System Layer. Fig. 2 shows the system architectures. We define the order of these three layers by the direction of the information flow. Each layer will be discussed in the following sections. Due to the scope of this paper, we mainly focus on the Information Network Layer.

2.1.1. Operation Layer

The Operation Layer can help every participant in the next-generation residential distribution system by providing report monitoring and control of the electric network services to manage and monitor the smart grid [9,10]. It is the top layer of our proposed system. The data generated by this layer will be transmitted to the Information Network Layer.

And there are many applications which can be used in this layer, such as outage management system (OMS), geographic information system (GIS), consumer information system (CIS), and distribution management system (DMS) that provides power quality management and load forecasting based on the meter data and other sources [11,12].

2.1.2. Information Network Layer

Information Network Layer deals with the data storage and processing issues. In this layer, all the data will be stored and processed. The data can be measured by different kinds of devices in Power System Layer, such as the smart meter. Then all the data will be processed locally or be selected and uploaded to the Operation Layer upon requests.

Many big data applications have already been implemented in fields of power systems. For instance, as of 2009, there were currently more than 100 active PMU devices placed around the Eastern United States that actively send to the Tennessee Valley Authority (TVA). As a result, TVA had roughly 20 TB of archived data in 2009, and it grew to 120 TB per year by 2012. In early 2010, there were about 250 PMUs deployed across North America. The massive deployment of PMUs all around the world gave birth to wide-area measurement system (WAMS). The deployment of phasor measurement units (PMUs) can help the operators to measure the values of voltage and current accurately in very short time intervals (typically 30–60 phasors per second) [13]. WAMS can provide precise data not only to enable a better indication of grid stress, but also to analyze and reveal the root cause of a blackout and any catastrophic failure of power grids. Moreover, in today's power system infrastructure, many power system oscillations may not be detected by the traditional slow-response SCADA system [14].

Meanwhile, a number of existing work which concentrates on data storage and processing architectures for smart grid has been done. In [15,16], the authors conducted a comparative study by testing four kinds of data storage and processing architectures including single relational database, distributed relational database, key-value distributed database and hybrid database based on the combination of a file system and a relation database. According to the simulation results for each system architecture, distributed processing can accelerate the calculation of prices and readjustments of tariffs to satisfy the real-time goal of the smart grid [15] and reduce data center energy consumption. Also, in [17], the performance of the distributed data storage and processing

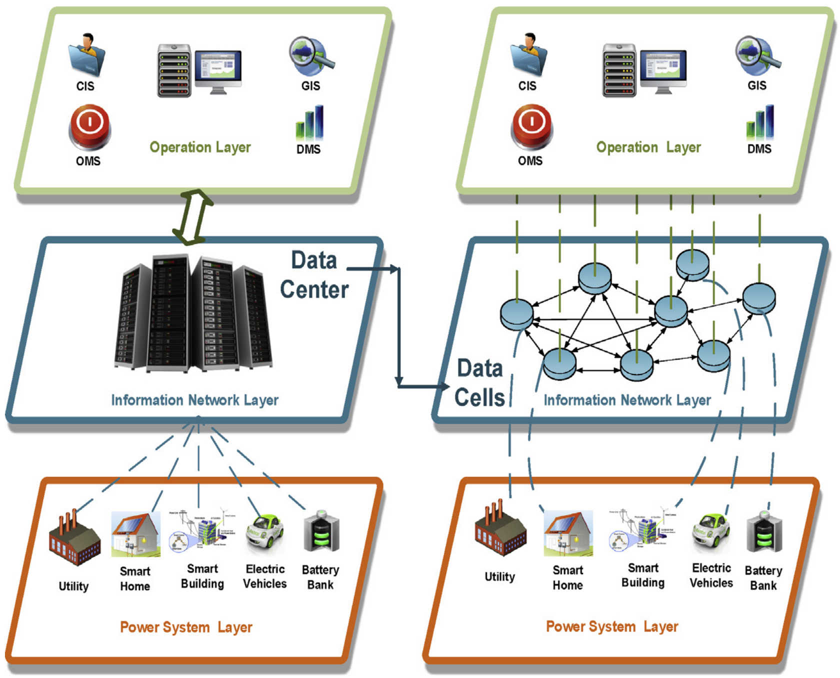


Fig. 2. System architecture.

architectures for wireless sensor networks (WSNs) was evaluated, considering energy consumption, processing, storage, scalability, bandwidth requirements as well as the capital costs. The results showed that distributed processing and storage architectures significantly improve the energy efficiency. For the distributed dataset, cloud computing is also one of the most up-to-date solutions. Some previous work has been done in this area. In [18], authors described their experiences about building a cloud-based software platform for data-driven analytics in the smart grid vision, and mainly focused on the dynamic demand response optimization (D^2R) application based on the cloud; however, they are limited by the finite resource of the distributed data centers but come with a high monthly bill. However, the current state-of-art cloud computing technologies cannot completely fulfill the needs of the smart grid [19]. The data storage and processing framework for the future smart grid CPSs is still not well defined. According to [20], we compare both merits and demerits of two architectures (Table 1). The distributed method is more suitable for large-scale, complex and heterogeneous systems. Thus, we proposed a distributed data storage and processing framework for the next-generation residential electricity system.

In this framework, as shown in Fig. 3, we propose a concept of DataCell to replace the traditional data center, which is an effective method to improve the performance.

Each DataCell can be regarded as a distributed storage and processing device. However, it is not just simply cutting the data stored in data center into several portions and then distributing them into different DataCells. Each DataCell not only stores its own part of data but also saves parts of data duplications of other cells. This difference is essential for next-generation power grid so that this working principle can greatly enhance the reliability of our proposed system.

2.1.3. Power System Layer

The Power System Layer is the physical layer that contains the distributed residential energy suppliers and consumers, regarded as the EnergyCell [7], such as utility companies, smart homes and electric vehicles, etc. In order to ensure the stability and reliability of the entire grid, the data produced by the EnergyCells needs to be monitored. In [21], the authors mentioned that the concept of information flow is an important concept to define the future new technologies in the smart grid, they described the requirements of the communication, potential applications and smart grid road maps for the future power systems as well. For monitoring services, this data will be converted into information flow and then transmitted to DataCells.

It should be highlighted that each EnergyCell is physically coupled to a DataCell; in other words they exchange information.

Table 1
Merits and demerits of centralized and distributed information processing architectures.

	Merits	Demerits
Centralized	•Simple to implement •Easy to maintain •Widely used and operated	•Computational burden •Requires high-bandwidth links •Single point of failure •Weak plug-and-play functionality
Distributed	•Easy to expand (plug-and-play) •Low computational cost •Avoid single point of failure •Suitable for large-scale, complex and heterogeneous systems	•Need synchronization •More time-consuming •Convergence rates partly depends on communication network topology •Upgrading cost on the existing control and communication facility

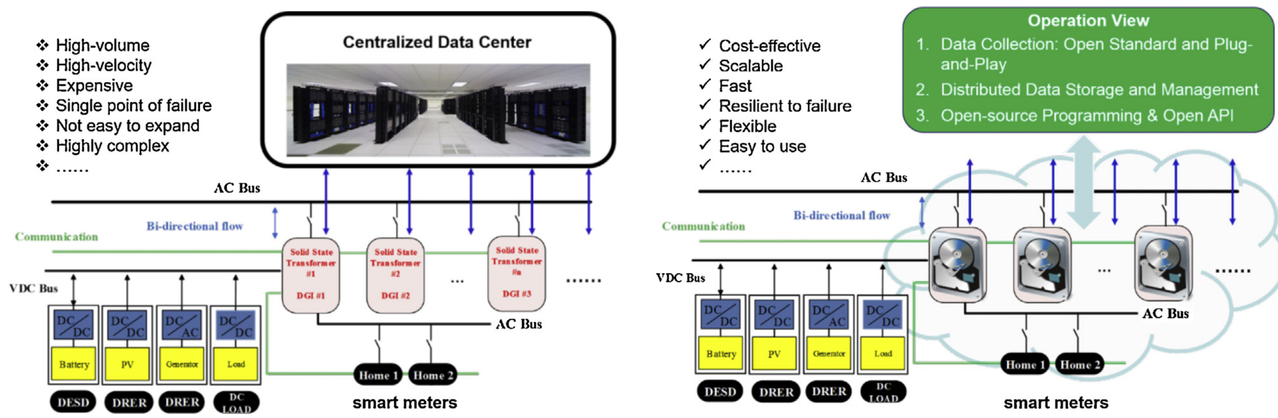


Fig. 3. (a) Centralized; and (b) envisioned distributed data storage and processing framework for a future residential distribution system.

Moreover, DataCells are connected to each other, leading to the real-time information exchange with each DataCell.

2.2. Problem statement

Energy management system (EMS) is the most important subsystem of the power system. In 1990s, the idea of moving the data storage and processing architecture of the EMS to a new one, which is based on the powerful centralized relational database, was proposed [22], and this architecture is still the main architecture of our power system today. This architecture contains three layers: user interface, database, and internet server application tiers, in this way, the system operation is performed using purely centralized optimization-based dispatching schemes that consider the problem at various time-scales. However, with the 15 years development of the power grid, the grid has already become a highly complex system that is only becoming more complex through the addition of the innovation [19]. Several problems (e.g., high cost, increasing complexity, vulnerability) will arise.

2.2.1. Data explosion

Smart meters are widely used in today's modern power system. For instance, Austin Energy in Texas has implemented 50,000 smart meters. These smart meters send data in a 15 min interval to the data center, which requires 200 TB of storage with disaster recovery redundancy factored in [23]. Because of log files of the execution and duplicate files, the data management takes up even more space than just storing the data. If Austin Energy wants to improve their service and move the 15-minute-interval to the 5-minute-interval, 600 TB more space has to be added to the data center in order to keep pace of the data growing [15,23]. The example of Austin Energy shows that how much data the smart meter can generate. In our proposed next-generation residential distribution system, more and more specified data storage units will be implemented in the power system, and this will cause the amount of data generated by them to skyrocket. Thus, it is better for the data storage part to provide only the appropriate data that the application needs to enhance the efficiency.

2.2.2. High cost

According to [19], on one hand, in the modern society, it becomes almost impossible to enhance the hardware performance by just improving the hardware itself without increasing the unit cost. Furthermore, the data privacy needs to be guaranteed, as this will affect the benefit of customers seriously if the data is unprotected, and to achieve this goal, some additional cost needs to be counted. On the other hand, when facing data exposition, under today' EMS architecture, the cost of the relational database included in the purchase of software with its license and the maintenance cost of the database center keeps rising all the time, and it is difficult to enlarge the volume of the existing data centers. Thus, power suppliers have to spend huge amounts of money in order to deal with the data explosion. In this situation, the proposed data storage and processing framework needs to be implemented on the low cost small-sized hardware. Furthermore, a cluster of such hardware can even match today's powerful central serves.

2.2.3. Increasing complexity

With the increasing complexity of the relational database, the tool to access the data will be limited. Moreover, the massive, heterogeneous and complex data that spans many hard disks is likely to cause the hardware failure more frequently. Sometimes, due to a single point of failure, the whole system may lose its operational file and lead to serious problems. To deal with such problems, a robust system which can be expanded easily as well as keeping the data in a secure condition is needed. In summary, a sophisticated and up-to-date data storage and processing framework is needed for managing large amount of data in a reliable, flexible and efficient way to support the CPSs in the future smart grid.

2.3. Summary

In this section, we proposed a CPS-like next-generation residential distribution system. It has three layers, and each layer has its own specific function, while three layers are interconnected with each other. When constructing such a complex and intelligent system, several problems will arise, such as the data exposition. To address this issue, there is an urgent need to investigate a new approach to deal with the big data generated by the next-generation residential distribution system.

3. Theory

3.1. A distributed data storage and processing framework

In this paper, the proposed framework is for massive distributed data storage and parallel processing based on the open source Apache Hadoop [24], which can be set up in extremely low-cost hardware (e.g., Raspberry Pi, Cubieboard). The key components of the framework are distributed file system (DFS) and corresponding distributed data processing model.

3.1.1. Distributed file system

Recently, various DFSs such as Hadoop distributed file system (HDFS), MooseFS, Parallel Virtual File System (PVFS) [25] have been developed for setting up the big data platform. In this paper, since

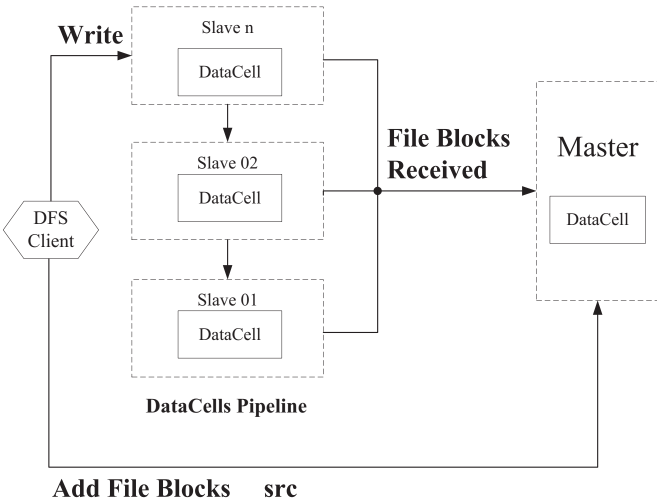


Fig. 4. Working flow of distributed file system.

HDFS is a highly fault-tolerant and can store information in a reliable fashion across multiple low-cost machines, we use the concept of HDFS to discuss the proposed DFS.

As mentioned above, DataCell is the basic physical unit of the proposed framework, which is capable of storing data and doing some processing work. In the proposed DFS, the hierarchical structure of DFS nodes is deployed to each DataCell including Master DataCell, Slave DataCells, Clients and Backup DataCells.

Firstly, master DataCell stores metadata such as the namespace and the mapping of the file blocks to Slave DataCells (the physical location of file data) [26]. Master DataCell is able to create, open, remove and rename a file or a directory as well as managing the namespace; while Slave DataCells manage the storage attached to which they run on, and do the reading and writing actions requested from Client (the file system's users). For the reliability of the system, Slave DataCells report Heartbeat and Block-report messages to the Master DataCell at regular intervals to confirm that slave DataCells are running well and the block replicas are available [26]. Besides, Backup DataCells are able to store all metadata information except for file block locations and are always synchronized with the state of the Master DataCell. If the Master DataCell was down, the Backup ones could restore the latest state of the original Master DataCell in a short time. The working flow of DFS is shown in Fig. 4. When an application reads a file, the Client will ask Master DataCell for the list of Slave DataCells which own the replicas of the file blocks and obtain the desired block from the given Slave DataCell. In a similar way, when the application writes, the file will be broken into several small blocks firstly and the Client lets Master DataCell choose Slave DataCells nearby to store replicas of the blocks. Once the list was created, the Client pipes the blocks to the given Slave DataCells.

3.1.2. Distributed data processing model

The data in our proposed DFS will be distributed to all over DataCells. As a result, there is a need for us to use MapReduce [27], a distributed data-parallel processing model, to process those data from multiple DataCells simultaneously. Actually, numerous specific MapReduce [24] models such as Google MapReduce, Amazon Elastic MapReduce (EMR) have emerged as widely used processing models on large datasets. The whole programming model can be divided into two basic functions, namely, Map and Reduce.

The working principle of MapReduce is illustrated in Fig. 5. When the Client calls the MapReduce function, the model in the Client can split the input file into some small processing chunks

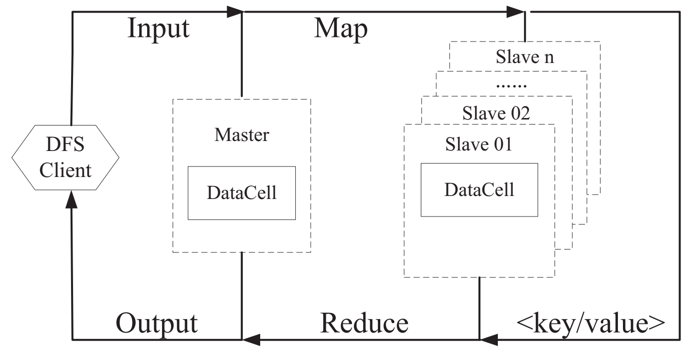


Fig. 5. Working principle of MapReduce.

according to a predefined Split function [28]. Afterwards, the Map function will deal with these chunks and generate a set of new intermediate key/value pairs and then passes these pairs to the Reduce function. The Reduce function accepts the key/value pairs and merges these values together to generate a smaller set of values as the output, then, sends back the output to the Client (Fig. 6).

4. Results and discussion

4.1. Hadoop architecture

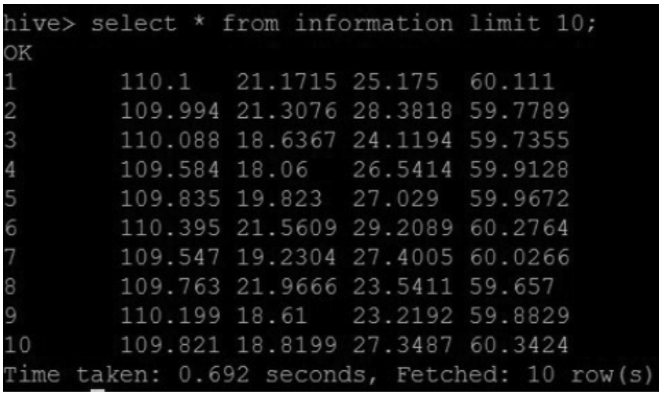
In this paper, the proposed system is based on Hadoop, which is an open-source platform for large scale storage and processing of data-sets on clusters of commodity hardware. It is architected as a Master–Slave structured system. The Master is regarded as NameNode which stores metadata such as file names and locations in a cluster, and the Slave is regarded as DataNodes to store replicated blocks. NameNode is available to create, open, remove and rename a file or a directory as well as managing the file system namespace. DataNodes manage the storage attached to the nodes they run on, and serve the reading and writing requests from the file system's users. And we set up a proof-of-concept cluster which contains 15 nodes, each node is based on the low cost credit-card sized single board PC (Cubieboard), as shown in Figure 6.

Fig. 6. A Proof-of-concept demonstration.

4.2. A proof-of-concept implementation and results

In the implementation, every Cubieboard is emulated as a DataCell in the future distributed power system, while the server is regarded as Master. In the cluster, all the DataCells contain the data in a specific format which can be recognized by Apache Hive based on the Hadoop.

Additionally, we can control the Master DataCell and monitor the other DataCells on another PC via SSH (putty), and it is



```
hive> select * from information limit 10;
OK
1       110.1   21.1715 25.175  60.111
2       109.994 21.3076 28.3818 59.7789
3       110.088 18.6367 24.1194 59.7355
4       109.584 18.06   26.5414 59.9128
5       109.835 19.823  27.029  59.9672
6       110.395 21.5609 29.2089 60.2764
7       109.547 19.2304 27.4005 60.0266
8       109.763 21.9666 23.5411 59.657
9       110.199 18.61   23.2192 59.8829
10      109.821 18.8199 27.3487 60.3424
Time taken: 0.692 seconds, Fetched: 10 row(s)
```

Fig. 7. Data format.

convenient to monitor all cells on the same screen. Fig. 7 shows that the data is appended in a table which contains the information about the Place_ID, voltage, current, temperature and frequency. At each row, different categories of the information are separated by a comma. We use Apache Hive to create a table at the Master side and load the measured data into the table to make it ready for further processing using MapReduce.

4.3. System evaluation

When predicting the grid information (e.g., frequency and voltage level) over the next time interval, the grid operators have to rely on numerous recorded data of the previous time periods. The data collection becomes a critical part of power system operations. In our case study, we implement the data collection application into the proposed proof-of-concept testbed. In this proof-of-concept testbed, each Cubieboard represents a DataCell, and we assume that the associated information (e.g., voltage, current, frequency, and power) has been already installed in each DataCell.

4.3.1. Processing efficiency

In order to compare the performance between distributed data storage and processing framework and conventional database, we use MySql relational database in the Master DataCell, and then, we processed with identical groups of data for a number of trials through both distributed manner (blue) and conventional database (red). We collected ten groups of data with the size of different groups of data ranging from 39.4 MB to 10322.8 MB and compared the total processing time consumed at each data size step. The total processing time includes the data loading time and the data processing time.

For the distributed processing method, the total processing time may vary with different trials. There existed the chance of costing different time at each trial. Thus, at different data size, we process the exactly dataset for four trials. The processing times are shown in Fig. 8.

When processing a certain amount of data, due to the internet connection speed variation, the total processing time will fluctuate within a small range at each measurement regards to the distributed data storage and processing method. In order to get a more general result, at each data size step, as shown in Fig. 8, we processed the data four times using the distributed storage and processing method and recorded the processing time at each time. Then, we use the average total processing time to compare the distributed manner and the conventional centralized one.

We evaluated the processing time using Mann–Kendall Trend test to test the processing time trend for both methods in Fig. 9.

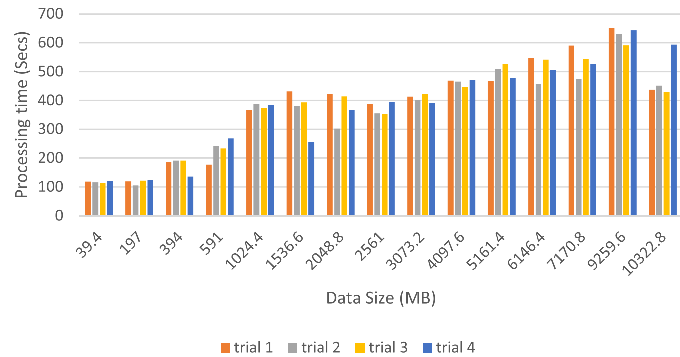


Fig. 8. Total processing times for the distributed data storage and processing method.

Mann–Kendall Trend test is the method which is used to see if the data is following a trend and if the trend is increasing or decreasing. z is the Mann–Rendall Statistic, it is an indicator variable, and it can be calculated by the parameters extracted from the data.

Suppose we have the following data trend conditions:

H0: data has the significantly increasing trend; H1: data has the increasing trend but not significantly; H2: data has the significantly decreasing trend; H3: the data has the decreasing trend but not significantly.

We can indicate the trend of the data by the value of z:

$z > 1.96$ means the data has the feature H0; $0 < z < 1.96$ means the data has the feature H1; $z < -1.96$ means data has the feature H2; $-1.96 < z < 0$ means the data has the feature H3.

z can be calculated through the following steps:

Define "n" as the number of the total data points.

Variance (assuming no field groups) can be calculated by the following formula:

$$V = \frac{n*(n-1)*(2n+5)}{18}$$

Calculated Mann–Rendall Statistic:

When $s = 0$, $z = 0$; when $s > 0$, $z = (s-1)/\mathrm{sqrt}(V)$; when $z < 0$, $z = (s+1)/\mathrm{sqrt}(V)$; where s is the slope of each two points.

Let' us define:

$$Y(n) = D(n) - C(n)$$

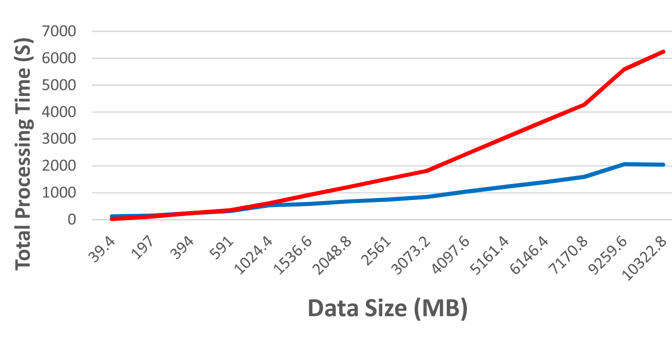


Fig. 9. Comparison on average data processing time.

Node	Last Contact	Admin State	Configured Capacity (GB)	Used (GB)	Non DFS Used (GB)	Remaining (GB)	Used (%)	Used (%)	Remaining (%)	Blocks
datacell1	0	In Service	91.7	0	6.04	85.66	0		93.42	0
datacell10	0	In Service	91.7	0	6.06	85.63	0		93.39	0
datacell11	2	In Service	91.7	0	6.04	85.66	0		93.42	0
datacell12	0	In Service	91.7	0	6.04	85.66	0		93.42	0
datacell13	2	In Service	91.7	0	6.06	85.64	0		93.39	0
datacell15	1	In Service	91.7	0	6.04	85.66	0		93.42	0
datacell16	2	In Service	91.7	0	6.04	85.66	0		93.42	0
datacell3	0	In Service	91.7	0	6.04	85.66	0		93.42	0
datacell4	1	In Service	91.7	0	6.04	85.66	0		93.42	1
datacell5	0	In Service	91.7	0	6.04	85.66	0		93.42	0
datacell6	0	In Service	91.7	0	6.04	85.66	0		93.41	0
datacell7	0	In Service	91.7	0	6.04	85.66	0		93.42	0
datacell8	1	In Service	91.7	0	6.04	85.66	0		93.42	0
datacell9	0	In Service	91.7	0	6.04	85.66	0		93.42	0
master	2	In Service	450.73	0	26.39	424.34	0		94.14	2

Fig. 10. DataCell working condition.

Cluster Summary

263 files and directories, 243 blocks = 506 total. Heap Size is 115.31 MB / 888.94 MB (12%)

Configured Capacity	:	1.69 TB
DFS Used	:	17.6 GB
Non DFS Used	:	126.77 GB
DFS Remaining	:	1.55 TB
DFS Used%	:	1.01 %
DFS Remaining%	:	91.68 %
Live Nodes	:	15
Dead Nodes	:	0
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	0

Cluster Summary

263 files and directories, 243 blocks = 506 total. Heap Size is 115.31 MB / 888.94 MB (12%)

Configured Capacity	:	1.43 TB
DFS Used	:	16 GB
Non DFS Used	:	108.62 GB
DFS Remaining	:	1.3 TB
DFS Used%	:	1.1 %
DFS Remaining%	:	91.46 %
Live Nodes	:	12
Dead Nodes	:	3
Decommissioning Nodes	:	0
Number of Under-Replicated Blocks	:	40

Fig. 11. Plug and play features.

where $C(n)$ is the processing time using the conventional centralized method, and $D(n)$ is the processing time using the distributed method. After the calculation, we have:

$z = -5.1467$

Slope Estimate $= -279.2709$

which indicates that the trend of $Y(n)$ satisfied the condition H2, and the distributed processing ability is superior to the conventional method when facing the increasing size of the data.

As shown in Fig. 9, using the conventional database, the processing time grows exponentially with respect to the increasing data size. In contrast, the distributed processing time does not rise dramatically as the size of the data increases. That is because MapReduce technique splits the data into smaller pieces (64 MB) for parallel data processing. However, the time of Hadoop's initialization is also noticeable even for small data sizes.

4.3.2. System robustness

In this test bed, we can monitor the test bed's working situation easily through the web page report; from the webpage report, we can know that how much data size capacity we have and how much storage we have used. Fig. 10 shows the working condition for each DataCell, we can know whether each DataCell is in service, and how much storage used in each DataCell.

In addition, during the simulation, we stopped two DataCells manually to mimic the switch failure and server power outage. And the entire cluster operational files were still readable, which demonstrates the reliability of the proposed framework. After the fault, DataCells communicated to each other to rebalance data in order to keep the replication times of the data.

4.3.3. Plug and play

When we want to add a DataCell in the cluster, we can easily duplicate the existing DataCell and scale up the cluster without reconfiguring the entire data storage and processing system. In this way, the test bed can be easily scaled up and down without doing too much reconfiguring work regards to the system. As shown in Fig. 11, when we plug or unplug DataCells, the working condition can be monitored thorough the webpage reports.

4.3.4. Cost analysis

As [22] indicates that expanding the existing data center will cost large amounts of money.

However, in our proposed framework, each DataCell can be installed on the low cost small-sized Arm Board, moreover, as energy devices are becoming more and more intelligent and the chip inside the energy device is power enough to integrate the distributed working mechanism as stated in the previous sections. This will be a huge saving for building the future smart grid CPSs.

4.4. Discussion

The U.S. power grid has been serving for decades and experienced several widespread blackouts. After a blackout, electricity shortage will lead to loss of service for numerous customers. Therefore, both the risk analysis of power outages and the subsequent power restoration are necessary. In order to achieve these tough tasks, huge amount of data needs to be recorded and stored.

After the evaluation, we can see that the proposed data storage and processing frame work has the ability to cope with the future power grid data explosion. It is important to mention that the improved performance is only noticeable when dealing with a relatively large amount of data.

5. Conclusion

In this paper, we first introduced a CPS-like next-generation residential distribution system. Secondly, we developed technical specifications for key functionalities and features of the proposed framework against yet-to-be defined system requirements. Thirdly, we investigated a radically different approach through distributed data cells to process a massive amount of power grid data in a timely and reliable manner. Finally, we substantiated the distributed data storage and processing framework on a proof-of-concept demo testbed using low-cost single-board PCs and further analyzed its performance with conventional centralized database with a comprehensive set of performance measures.

The proposed research work illustrated how advanced ideas from IT industry and power industry can be combined in a unique way. The proposed high-availability and fault-tolerant distributed file system and data processing framework can be easily tailored to support other data-intensive applications in a large-scale and complex power grid. If implemented successfully, we can translate Smart Grid with high-volume, high-velocity, and high-variety data to a complete distributed cyber-physical architecture. In addition, the proposed work can be easily extended to support other CPS applications.

Acknowledgment

This work is supported in part by the National Science Foundation under Award Number EEC-0812121 and the New Faculty Startup Fund at University of Michigan-Dearborn.

References

[1] A. Castillo, Risk analysis and management in power outage and restoration: A literature survey, Electr. Pow. Syst. Res. 107 (2014) 9–15.
[2] A. Mazur, T. Metcalfe, America's three electric grids: are efficiency and reliability functions of grid size, Electr. Pow. Syst. Res. 89 (2012) 191–195.
[3] J.J. Stroker, Power grid reliability, Cem. Ind. Tech. Conf. (2004) 51–59.
[4] H.E. Brown, S. Suryanarayanan, S.A. Natarajan, S. Rajopadhye, Improving reliability of islanded distribution systems with distributed renewable energy resources, IEEE Trans. Smart Grid 3 (4) (2012) 2028–2038.
[5] H. Li, L. Lai, H.V. Poor, Multicast routing for decentralized control of cyber physical systems with an application in smart grid, IEEE J. Sel. Areas Commun. (2012) 1097–1107.
[6] A.Q. Huang, M.L. Crow, G.T. Heydt, J.P. Zheng, S.J. Dale, The Future Renewable Electric Energy Delivery and Management (FREEDM) System: The Energy Internet, Proc. IEEE 99 (1) (2011) 133–148.
[7] W. Su, A.Q. Huang, A game theoretic framework for a next-generation retail electricity market with high penetration of distributed residential electricity suppliers, Appl. Energy 119 (2014) 341–350.
[8] C.K. Tham, T. Luo, Sensing-driven energy purchasing in smart grid cyber-physical system, IEEE Trans. Syst. Man Cybern. Syst. 43 (2010) 773–784.
[9] S.J. Huang, C.C. Lin, Application of ATM-based network for an integrated distribution SCADA-GIS system, IEEE Trans. Power Syst. 17 (2002) 80–86.
[10] G.N. Korres, A. Tzavellasb, E. Galinasb, A distributed implementation of multi-area power system state estimation on a cluster of computers, Electr. Pow. Syst. Res. 102 (2013) 20–32.
[11] J. Zhou, R.Q. Hu, Y. Qian, Scalable distributed communication architectures to support advanced metering infrastructure in smart grid, IEEE Trans. Parallel Distributed Syst. 23 (2012).
[12] A.A. El-Keib, X. Ma, H. Ma, Advancement of statistical based modeling techniques for short-term load forecasting, Electr. Pow. Syst. Res. 35 (1995) 51–58.
[13] A. Bose, Smart transmission grid applications and their supporting infrastructure, IEEE Trans. Smart Grid 1 (2010) 11–19.
[14] M. Kezunovic, X. Le, S. Grijalva, The role of big data in improving power system operation and protection, IREP Symp. Bulk Pow. Syst. Dyn. Control – IX Optimization Secur. Control Emerg. Pow. Grid (IREP) (2013) 1–9.
[15] N. Tsiftes, A. Dunkels, A database in every sensor, Proc. 9th ACM Conf. Embedded Netw. Sens. Syst. (2011) 316–332.
[16] M.A. Martinez, S.H. Lopez, A. Sanchez, J.R. Williams, P. Roth, P. Hofmann, A. Zeier, A comparative study of data storage and processing architectures for the smart grid, First IEEE Int. Conf. Smart Grid Commun. (2010) 285–390.
[17] R.M. Oviedo, F. Ramos, S. Gormus, P. Kulkarni, M. Sooriyabandara, A comparison of centralized and distributed monitoring architectures in the smart grid, IEEE Syst. 7 (2013) 832–844.
[18] Y. Simmhan, S. Aman, A. Kumbhare, L. Rongyang, Cloud-based software platform for big data analytics in smart grids, Comp. Sci. Eng. 15 (4) (2013) 38–47.
[19] R.C. Green, L.F. Wang, M. Alam, Applications and trends of high performance computing for electric power systems: focusing on smart grid, IEEE Trans Smart Grid 4 (2) (2013) 922–931.
[20] W. Su, J. Wang, Energy management systems in microgrid operations, Electr. J. 25 (2012) 45–60.
[21] V.V. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, G.P. Hancke, A survey on smart grid potential applications and communication requirements, IEEE Trans. Ind. Inform. 9 (1) (2013) 28–42.
[22] J.T. Ma, Y.M. Liu, L.F. Wu, New energy management system architectural design and intranet/internet applications to power systems, Int. Conf. Energy Manage. Pow. Deliv. (1998) 207–212.
[23] S.E. McHann, Grid analytics: how much data do you really need, IEEE Rural Electr. Power Conf. (2013), C3-1–C3-4.
[24] Hadoop, Available from http://http://hadoop.apache.org/, 2014 (online).
[25] Y. Wu, F. Ye, K. Chen, W. Zheng, Modeling of distributed file systems for practical performance analysis, IEEE Trans. Parallel Distributed Syst. 25 (2014) 156–166.
[26] K. Shvachko, H. Kuang, S. Radia, R. Chansler, The Hadoop distributed file system, IEEE 26th Symp. Mass Storage Syst. Technol. (2010) 1–10.
[27] J. Ekanayake, S. Pallickara, G. Fox, MapReduce for Data Intensive Scientific Analyses, IEEE Fourth Int. Conf. eSci. (2008) 277–284.
[28] S. Loughran, J.M.A. Calero, A. Farrell, J. Kirschnick, J. Guijarro, Dynamic cloud deployment of a MapReduce architecture, IEEE Internet Comp. 16 (2012) 40–50.