Proceedings of the 2020 Winter Simulation Conference
K.-H. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, eds.

ADVANCING SELF-HEALING CAPABILITIES IN INTERCONNECTED MICROGRIDS VIA DYNAMIC DATA DRIVEN APPLICATIONS SYSTEM WITH RELATIONAL DATABASE MANAGEMENT

Abdurrahman Yavuz
Joshua Darville
Nurcin Celik

Department of Industrial Engineering
University of Miami
1320 S Dixie Hwy
Coral Gables, FL 33146, USA

Jie Xu
Chun-Hung Chen

Department of Systems Engr & Operations Research
George Mason University
4400 University Drive
Fairfax, VA 22030, USA

Brent Langhals
Ryan Engle

Department of Systems Engineering & Management
Air Force Institute of Technology
2950 Hobson Way
Wright-Patterson Air Force Base, OH 45433, USA

ABSTRACT

A microgrid is an interdependent electrical distribution system containing renewable energy sources, local demand and a coupled connection to the main grid. A very appealing feature of a microgrid is its capability to self-heal from disruptions, which is made even more viable with the emergence of interconnected collaborative microgrids. In this study, we present a dynamic data driven application system framework that integrates a relational database management system (RDBMS) to advance self-healing capabilities among interconnected microgrids. A RDBMS facilitates access to various sensors in the microgrid for fast abnormality detection and for determining the optimal self-healing action to implement. We build an agent-based simulation model (ABM) for three self-healing interconnected microgrids. Using the ABM, we compare self-healing operations of microgrids with and without an RDBMS. Simulation results show that an RDBMS may lead to faster response time and thus advance self-healing capabilities of interconnected microgrids.

1 INTRODUCTION

A microgrid can be defined as a combination of interrelated demand loads and distributed energy resources that may operate as a particular embodiment in connection or disconnection with the main power grid. Microgrids bring along a multitude of benefits including the minimization of energy consumption costs by facilitating the integration of renewable energy sources, which also mitigates the environmental impact of energy consumption by reducing greenhouse gas emissions. Microgrids also contribute to energy surety by using controllable load and congestion relief to improve network flow. In the past decade, both private and public sectors have been increasingly embracing microgrids as a new paradigm of energy production and distribution. Specifically, a microgrid presents a very promising energy infrastructure to reduce operations cost and improve energy independence from the utility power grid in case of severe weather events.

A newly emerging mode of microgrid operation is collaboration among neighboring microgrids. This collaborative mode allows microgrid operators to coordinate energy production, and achieve better operational outcomes. Specifically, collaborative operations of microgrids may lead to autonomous and robust self-healing microgrids, providing a resilient energy infrastructure.

The resilience of collaborative self-healing microgrids depends on the microgrids operators' access to situational awareness, which requires pervasive sensing with an efficient means to transmit and process a large amount of data in heterogeneous formats, to detect any disturbance in real time. Hence, a seamless integration of dynamic sensor and instrumentation data providing information on power consumption/generation and flow, into an executing application model is paramount. All these dynamic events take place at various time and spatial scales. Consequently, the executing collaborative microgrids application needs to control and steer the sensors and instruments to acquire the most relevant data at the required resolution to support on timely decision making. Only doing so can one turn the Big (Dynamic) Data deluge into smart data regimes that provide the foundation to identify optimal actions in response to system state changes or disturbances, and thus achieve self-healing resilient microgrids operations.

The aforementioned capability transcends beyond the results and tools developed within the framework of classical feedback control theories. Recently, a new powerful paradigm, Dynamic Data Driven Application System (DDDAS) (Darema 2004, 2011; Blasch 2018; Blasch et al. 2019) offers a promising path to achieve such capabilities in collaborative microgrids operations. DDDAS advocates a novel paradigm for dynamic data driven application simulations that establishes a bi-directional flow of data and control/sensing decisions. Under DDDAS, an executing application dynamically detects state changes via timely data feed from sensors and instrumentation. The application simulation evaluates alternative control decisions and simultaneously steers sensors and instruments to acquire additional data to support accurate predictions by the application simulation of the outcome of a control decision. Furthermore, DDDAS has already shown vast potential through successful applications across a broad and diverse array of fields. Examples abound in aerospace engineering (Bazilevs et al. 2015; Lecerf et al. 2015), homeland security (Khaleghi et al. 2013), sensing and tracking (Lagor and Paley 2014), materials modeling (Li et al. 2017), cybersecurity (Badr et al. 2015), smart cities (Fujimoto et al. 2016), and computing systems and software (Jin and Nicol 2015). In the context of microgrids, Damgacioglu et al. (2018), Shi et al. (2015), and Thanos et al. (2015) have shown that DDDAS provided a powerful framework for microgrid control by enabling dynamical data updates for a running application simulation.

Whereas many studies assume data is readily available upon the application's execution, this study potentially influences the perspective of future DDDAS studies and its derivatives by focusing on the data retrieval response time to the application or initialization period. The database management system (DBMS) used by the application has a major role in determining data retrieval time. Traditionally, many applications use a relational database as the primary means to store and retrieve data ever since E.F. Codd articulated the relational database in 1970 (Meijer and Bierman 2011). Relational databases use Structured Query Language (SQL) to join related tables based on shared key values. The tables are pre-defined and can accept a variety of heterogeneous data types. In this study, we analyze the performance of a DDDAS executing application integrated with a relational database management system (RDBMS) in the context of microgrid operations.

DDDAS has proved to be a promising control framework for microgrid operations. Damgacioglu et al. (2018) introduced a dynamic data driven multi-objective optimization model (DDD-MOM) for microgrid. DDD-MOM dynamically obtains system and environmental data as input into a data driven simulation with bi-objective optimization to determine operational plan of a single microgrid. Likewise, Thanos et al. (2015) investigated a DDDAS framework for load dispatching by utilizing an online learning algorithm to feed the database for faster future computations. The first study describes the application's effect on the physical system while the second study describes the physical system's effect on the database, but we investigate the database's effect on the application. Although the previously mentioned studies utilize the DDDAS foundation to enhance their individual applications, they assume the data is readily available and neglect the data's response time to their unique applications. Similar to the downstream consequences of adjusting an upstream process, this study highlights the upstream effect of response time on the downstream agent-based simulation of three interconnected microgrids. In this way, we not only enhance the accuracy of our application using DDDAS similar to the previous studies, but we also enhance the overall performance of DDDAS by improving an upstream process within the framework itself.

An RDBMS provides a benchmark for improving the data storage design and thus the coupled application performance. The executing application is an agent-based simulation model that replicates the collaborative self-healing operation of three interconnected microgrids. Therefore, in this study we developed an approach for measuring and comparing a RDBMS performance versus an external application's performance within the DDDAS framework. The paper is organized as follows. In section 2, we describe the experimental setup and various components associated with the overall architecture. In section 3, we conduct numerical analysis to determine a metric for assessing the performance of the coupled application system and database system. We present conclusive remarks and future work in Section 4.

These two capabilities are especially crucial to self-healing operations of microgrids supplied by renewables because distributed generation (DG) units such as micro turbines create two-way power flows in microgrids. Renewable generations are highly intermittent and lead to frequent changes in power supplies, which for collaborating microgrids further create complex dynamic interactions. This task is further complicated when microgrids owned by different entities are interconnected and thus data confidentiality may need to be preserved when collaborative microgrids share sensor and demand loads data. Second, microgrids operators must have the capability to transform real time situational awareness into (near) optimal control decisions to respond to any disturbances and assure reliable power supply to some much needed loads, while minimizing the impact on other loads.

2 EXPERIMENTAL SETUP

In this paper, we model the operation of three interconnected hypothetical microgrids. The first is a Microgrid 1 (MG-1) which requires a high degree of energy surety for continual operation. The MG-1 has four types of distributed generation (DG) units: diesel-powered microturbines, micro wind turbines, biomass generators, and solar panels. The MG-1 is located such that a nearby area contains two other microgrids. The second is operated by Microgrid 2 (MG-2) and the third is third is governed by Microgrid (MG-3). All three microgrids contain three major load categories: essential, priority, and non-essential. The highly intermittent nature of renewable generation poses a challenge to reliably stream supply to essential and priority loads given the high costs and environmental impact of diesel microturbines. Therefore, MG-1, MG-2, and MG-3 are also connected to the main utility power grid for grid stability while being interconnected and able to collaborate as described by Table 1 and shown in Figure 1.

Table 1: Overall system descriptions of each considered notional microgrid.

Microgrid	# of Solar Arrays – Average Power Output (kWh)	# of Wind Turbines – Average Power Output (kWh)	# of Diesel Turbines – Average Power Output (kWh)
MG-1	1 – 779.68	1 – 101.03	–
MG-2	1 – 550.41	–	–
MG-3	1 – 625.12	1 – 82.65	1 – 114.94

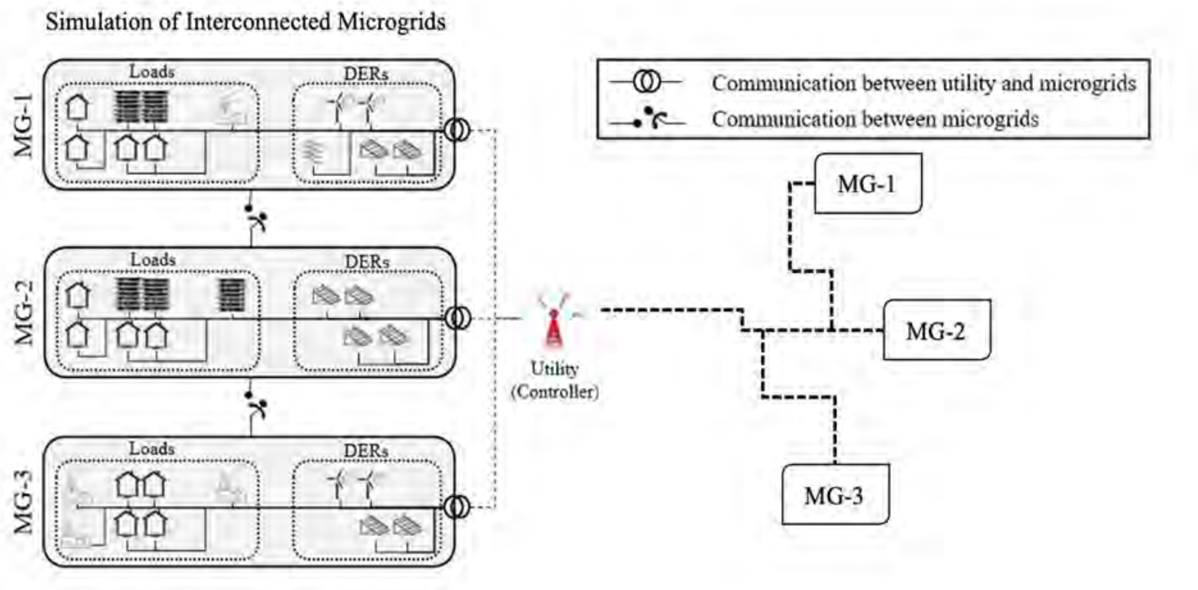


Figure 1: Overall depiction of interconnected notional self-healing microgrid coalition model.

Collaborations among these three interconnected microgrids enable self-healing operations of microgrids. In order to test self-healing capabilities of collaborative microgrids, two hypothetical cases of disruption to the daily microgrids' operation were simulated where in the first one, a part of the considered network area is affected by a major storm and the MG-3's power generation remains shut down for the next 48 hours after the storm. In the second case, there is an earthquake affecting MG-2. As a precautionary measure, MG-1 disconnects itself from the outside, e.g., the main utility grid and MG-2/MG-3, for two hours such that any potential cascading effect to MG-1 can be eliminated. Meanwhile, it would take six hours to repair the damage on MG-2 caused by the earthquake.

In both scenarios, operators of MG-1, MG-2, and MG-3 all have to make some important control decisions that would determine the success of their self-healing operations. Specifically, the DDDAS framework executes a data driven application simulation and uses a new method known as ordinal transformation (Xu et al., 2014, 2016) to make real-time decisions on the sharing of electricity generation capacity during a power disruption among collaborating microgrids, isolation of part of all of a microgrid from the rest of the network, and the discharge from energy storage devices within microgrids.

Making (near) optimal operational control decisions have direct impact on the operational performance of microgrids and determine the viability of self-healing operations. However, we face unprecedented challenges posed by the two-way power flow from DG including both highly intermittent, uncertain renewable generation and unpredictable load changes, which may lead to extreme conditions (Thanos et al., 2015). These interconnected microgrids are accompanied by an increased number of state parameters, fast-evolving dynamics and higher dimensions of control decisions with their inter-dependencies all exacerbating the data retrieval/processing and computational burdens.

Despite these daunting data and computation challenges, collaborative microgrids have a tremendous potential to provide an autonomous, energy efficient and a resilient energy infrastructure. This work aims at addressing the data and computation challenges of such advancement.

2.1 Self-healing Collaborative Microgrids Simulation

We developed an agent-based simulation model for operations of self-healing collaborative microgrids. This application simulation model incorporates sensory data generated from wind, solar and weather sensors for our hypothetical microgrid setting. Based on the sensory data feed, the simulation model runs and makes decisions about operational planning of each microgrid while steering sensors and instrumentation for data collection to support this decision making as an executing simulation application under the DDDAS paradigm. Figure 2 depicts this DDDAS-based framework for microgrid control.

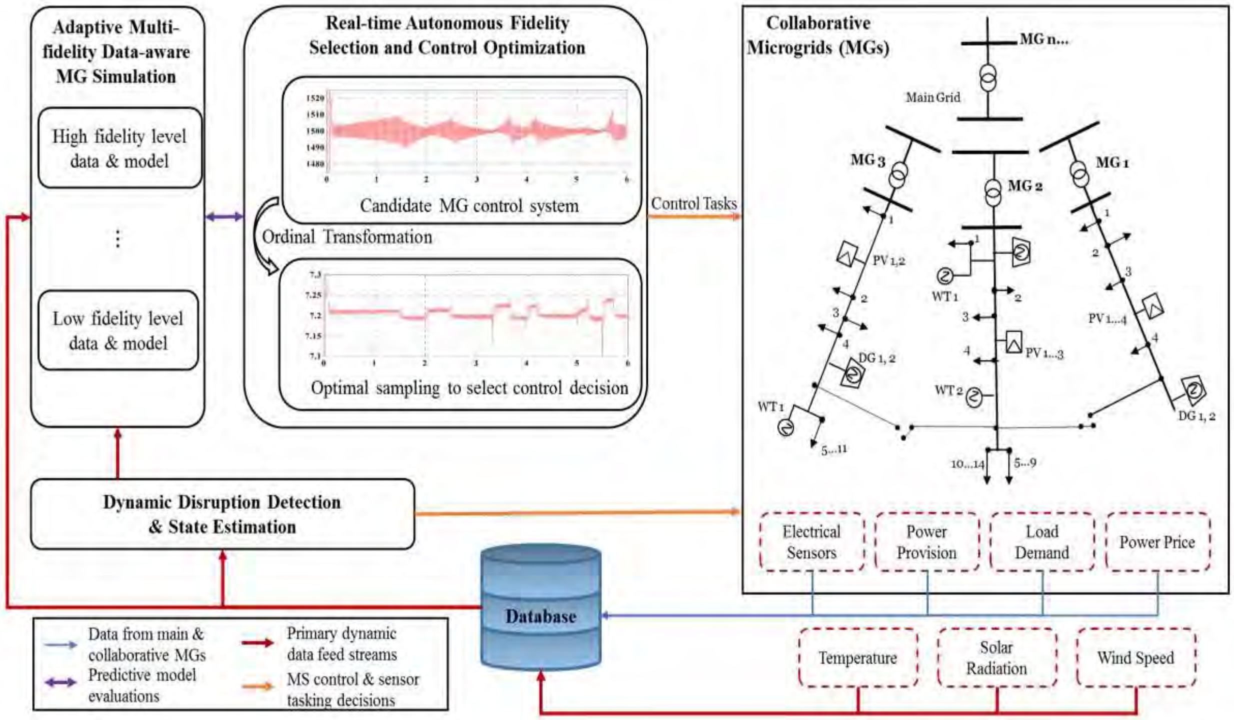


Figure 2: A DDDAS-based framework for microgrid control.

Specifically, operational planning consists of decisions related to how much electricity is needed to satisfy the load in an economic and reliable manner. Within the self-healing framework, if a microgrid is not able to meet its demand in a certain time period, it communicates its need to another microgrid within the coalition. Subsequently, the other microgrids follow predetermined protocols for assisting the microgrid suffering from energy shortage. These protocols involve two conflicting objectives: maximizing the energy surety and minimizing cost. However, the focus remains on the self-healing microgrids being a complicated network, which requires a substantial amount of data to simulate the behavior of the system. During the execution of the simulation, all data transaction should be network related because the generation and distribution of electricity is realized through this network. For instance, if a blackout occurs in a certain region of a microgrid, we need to specify which elements of the microgrid are affected by the blackout. This requires considerable computational time.

Table 2: Agents and corresponding quantities.

Agent	Number of agents
Demand Nodes	346
Wind Turbines	2
Diesel Generator	6
Battery	3
Diesel Generator	1

The simulation model in this work is developed in a Java based simulation software environment. Table 2 shows agent types and their quantities. This table shows the high-dimensional decision space and the number of parameters that need to be dynamically estimated. Further attributes of each agent type can be found in Figure 3.

2.2 Database and Software

Naturally, each data driven application may require different types of data storage options based on its data formats, types of data query operations, and the timescale the system operates on. In our experiment, we build a relational database considering the needs of an executing application simulation for three interconnected microgrids. To simulate self-healing operations of interconnected microgrids, we need sensing data retrieved from different sensors dispersed throughout a relatively large geographical area. Therefore, we need to define geographical position related attributes for each agent in our simulation. In order to keep each agent informed of the network information, we also need to store parameters related to the interconnections among agents. Similarly, time stamps and observation data are also important as they provide information needed to detect changes in generation and loads. These two features make our database a spatial-temporal one.

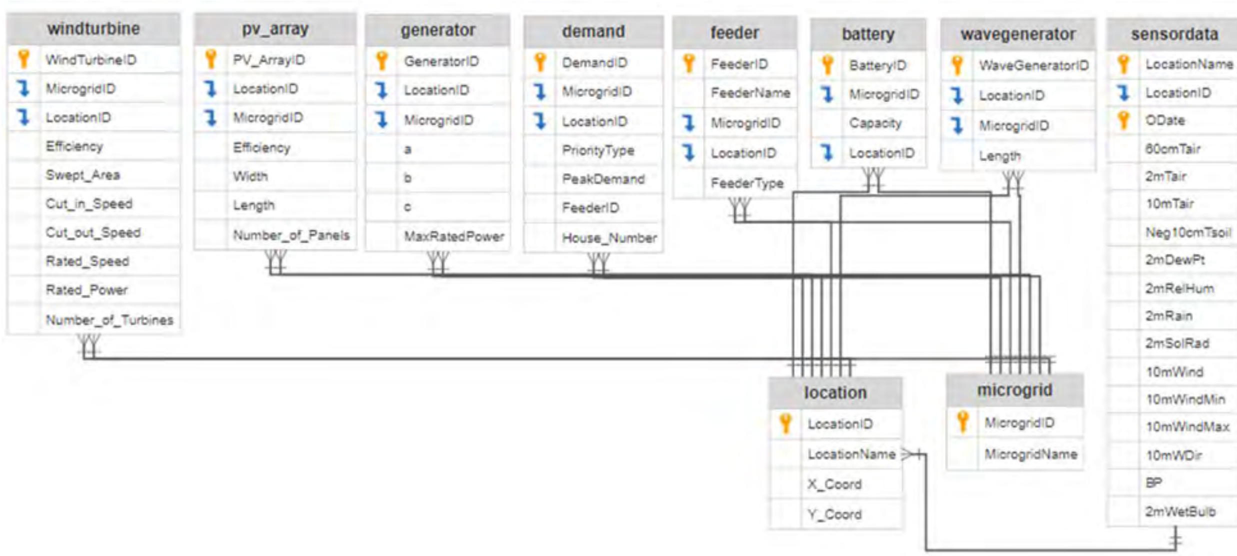


Figure 3: Entity relationship diagram of the considered database.

We chose MySQL 8.0.19 to construct the database, due to the availability of MySQL application programming interfaces (APIs) and its popularity in many real-life applications. APIs make it easy to retrieve data from the MySQL server using programs written in Java and Python, which consequently makes it easy to replicate the results of our study for testing purposes. Figure 3 gives the entity relationship diagram for the database we integrate with our executing application simulation under the DDDAS framework.

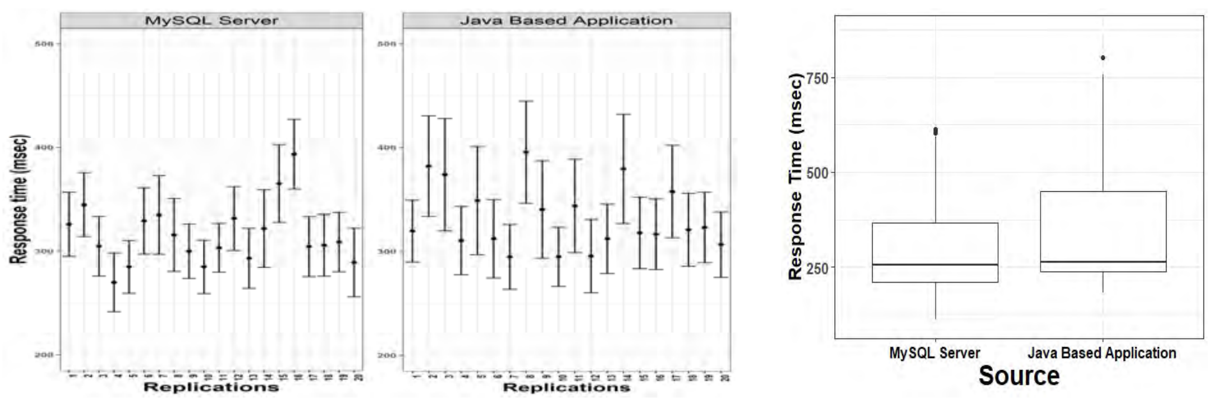
2.3 Experimental Design

We compare the performance of the executing application simulation with an integrated relational database and without it when sensory data are stored as external files such as .txt and .xlsx files. The relational database serves as a tool for initialization of the simulation and supports space-time related queries. When external data files are used, the simulation is initialized using one set of external files, while sensor data are retrieved from another set of external files.

Because data retrieval are common operations in both cases, and are also the most frequently performed operations, we measure the performance using the completion time of data retrieval operations milliseconds. For space-time related queries, only RDBMS supports such queries, and to the absence of such practice in the first case. In contrast, when external data files are used, instead of using queries, we wrote Java codes to collect any required space-time information, which brings additional computational burden to the simulation module.

3 EXPERIMENTAL RESULTS ANALYSIS

Because of the stochastic nature of the problem, we perform multiple replications to obtain statistically valid results. Each replication required its own initialization, and thus a reliable and fast initialization phase is of great importance to the simulation. The initialization times are recorded from 500 independent and identically distributed (i.i.d.) simulation replications for each case (100 runs each with five replications) where the confidence intervals are computed on the resultant initialization times with at least 95% level of confidence. The number of replications is determined as the sample size required to obtain 0.8 power of the test value. The replication numbers are determined based on Java heap size where any further increase in these numbers triggers a Java heap size error in the considered simulation.

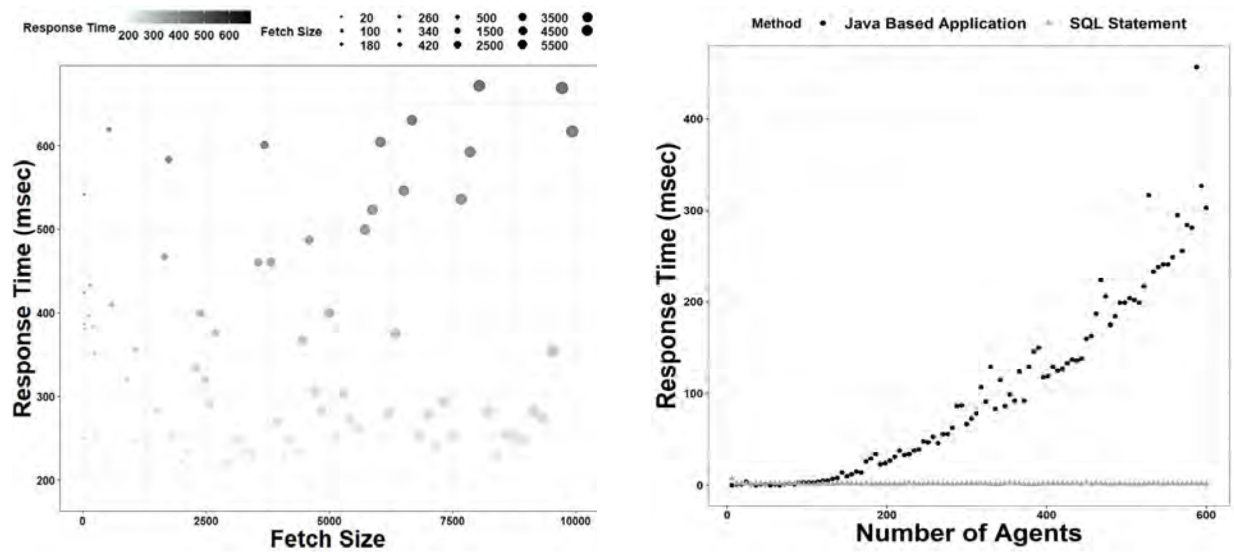


(a): Confidence intervals for randomly selected 20 replications. (b): Box plot of initialization times.

Figure 4: Comparison of initialization times.

To better illustrate the initialization performance, we present the box plot for the initialization times measured from 200 replications in Figure 4(a). It is worth noting that we have considerable variations in initialization times across replications. Furthermore, MySQL server has much more consistent performance than using external data files. This can be seen by the fact that the median initialization time using MySQL server is only about 250 milliseconds, versus the large spread of initialization times when external data files are used. To further show the dispersion of response times, Figure 4(b) plots confidence intervals on response times from 20 randomly selected replications. Designed hypothesis test with the null hypothesis claiming that average response times obtained from MySQL and Java based application are equal results in a p-value of 0.004 and hence verifies that the response time of the MySQL server is shorter than that of the Java based application at 95% significance level.

The performance of MySQL server can be further improved by adjusting parameters such as fetch size and indexes in the database. These parameters should be dynamically adjusted according to the sensory data during the simulation execution. Fetch size defines the number of rows returned for each query. Currently, there is no known rule or optimization tool that could determine the optimal fetch size for all queries of a user application. Thus, we experimented with different fetch sizes. Figure 5(a) shows the response time values in milliseconds for different fetch sizes. We initialized the simulation ten times for each fetch size and calculated the average value of the response times measured. A decision on fetch size needs to consider the volume of data retrieval. It is possible to adaptively set different fetch sizes in different queries. In this study, we determined the fetch size by sorting the response times of the experiments in ascending order and averaging the top ten fetch sizes with shorter response times compared to other experiments.



(a): Response time for different fetch sizes. (b): Response times for different number of agents.

Figure 5: Comparisons of different fetch sizes and different number of agents.

DDDAS requires fast and reliable interaction channels among modules. In our study, we need to maintain synchronous and efficient interactions among different optimization and simulation modules. We focus on response times of key operations of self-healing microgrids when assessing the effect of the incorporation of a RDBMS. Specifically, we compare the operations to modify variables, such as the number of connected demand points in a specific network (defined as feeders in microgrids), with and without RDBMS. In comparison with RDBMS, Table 3 gives the pseudo code to update connected demand points using external data files. In our experiment, we wrote Java programs to perform these operations.

In contrast, with an RDBMS, we used the query described in Table 4 to perform the same operations. Figure 5(b) shows the operations' response time for both approaches as the number of agents increased. Increasing the number of agents allows us to examine the scalability of different approaches.

Table 3: Pseudo code to update connected demand points.

```
Function: Updating Variables
for feeders do
    for demands do
        if demand node is connected then
            Find demand's type and its corresponding microgrid
            Update related variables
        end if
    end for
end for
```

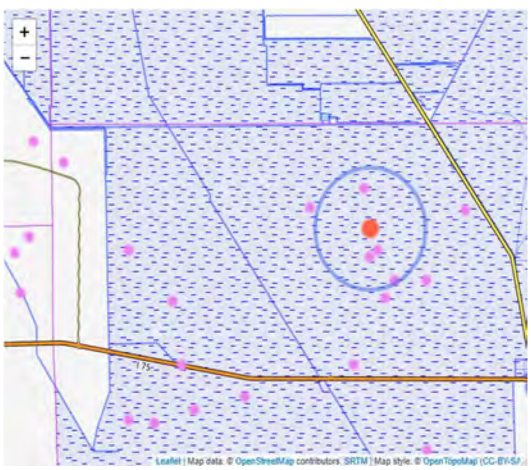
Fetch size is set to a constant as previously explained above by the help of Figure 5(a). In Figure 5(b), we clearly see response time increasing at an approximately linear rate directly proportional to the number of agents. However, when using MySQL server, the response time remains stable for experiments including up to 600 agents. This is an important result for dynamic data driven simulation of interconnected self-healing microgrids, pointing to a scalable DDDAS approach to collaborative microgrid operational planning.

Table 4: SQL code of the function to update connected demand points.

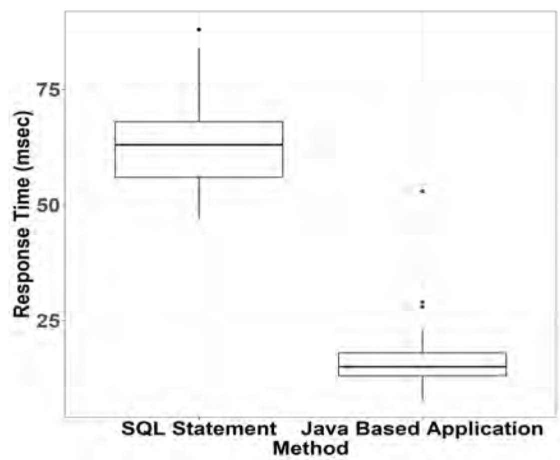
```
SELECT Feeder.FeederID, Subquery.*
FROM Feeder, (SELECT FeederID, PriorityType,COUNT(*)
          FROM demand GROUP BY demand.FeederID,demand.PriorityType) AS Subquery
WHERE Feeder.FeederName = Subquery.FeederID
SELECT MicrogridID, COUNT(*)
FROM demand
GROUP BY demand.MicrogridID
```

3.1 Utilization of Spatial Queries

For an interconnected self-healing microgrids application simulation to perform well, it often requires sampling of the same type of information but from sensors placed in different geographical locations. Figure 6(a) shows an example of sensor locations, where the blue circle defines a five kilometers radius of a given sensor. If there is an anomaly such as the application misses the data transmitted by the sensor located at the center of the blue circle and indicated by a large red dot, the application simulation may consider retrieving readings from other sensors close to the faulty sensor, as indicated by smaller pink dots on the map.



(a): A faulty sensor location and nearby sensors that may provide back-up data.



(b): Comparison of response times in determining locations of sensors.

Figure 6: Performance in case of spatial queries.

We compare response times of this operation using the MySQL server versus reading external data files. Figure 6(b) shows the response times in milliseconds. As can be observed in the figure, to determine sensors within a five kilometers radius of the faulty sensor, Java codes outperformed the spatial SQL queries. The reported results were obtained with optimized SQL statements running on 100 randomly generated geographic locations. To check the dependency of performance on the scale of the problem, we also ran experiments with more geographical locations. The results indicate that using spatial SQL statements may negatively affect the computational time of the simulation. However, SQL statements produced slightly more accurate results than Java codes. The means of response times of the SQL statement and Java-based application are recorded as 63 msec and 18 msec, respectively, with the lower bound for the difference between the two being 43 msec at 95% confidence level. The standard deviation of response times for the SQL statement and Java-based application is 9.69 and 15.93, respectively.

Therefore, we conclude that there is a tradeoff between the accuracy and response time that can be explored to further enhance the performance of a DDDAS executing application and the choice between the two methods should be based on how often this kind of spatial information is needed.

4 CONCLUSION

In this work, we presented a DDDAS framework to address data operations and retrieval speed of interconnected microgrids with self-healing capabilities. We introduce an RDBMS as a tool to speed up simulation executions and build a bridge between processes within DDDAS. We conducted a synthetic case study utilizing an agent-based simulation model for three interconnected collaborative self-healing microgrid. We tested the efficiency of an RDBMS as a tool to perform simulation initialization and update variables during simulation. The initialization time of a simulation model is an integral part of the total computational time. When a large number of replications are needed, initialization creates a significant burden on computational resources. We show that an RDBMS may significantly lower this computational burden compared to using external data files.

We then tested the response time for updating variables, as an example of a typical data operation that a DDDAS executing application simulation may need to execute frequently throughout the simulation. Our experimental results show that fetch size has an impact on response time and could be optimized based on the characteristics of data queries. We then show that an RDBMS may offer a scalable approach to perform important data operations to support fast and synchronous interactions among different modules under the DDDAS paradigm. The results for space-time related queries of the simulation application indicate a decrease in response time when using a RDBMS. However, RDBMS also achieved better accuracy for spatially oriented queries than the Java based application.

In summary, our results demonstrate the potential of the incorporation of an RDBMS into the DDDAS framework and provide initial evidence to support further investigation of the use of RDBMS. Because RDBMS is only one possible form of DBMS, in the future, we will also investigate other DBMS, such as NoSQL, to identify the best DBMS and optimize its design parameters for a particular DDDAS application. One promising approach is to use ordinal transformation (Xu et al. 2014, 2016) to efficiently use expensive simulation data to improve accuracy of DBMS selection using a new Bayesian procedure (Peng et al. 2018).

ACKNOWLEDGEMENTS

This work is supported in parts by the Air Force Office of Scientific Research under award number FA9550-19-1-0383.

REFERENCES

Badr, Y., Hariri, S., Youssif, A.N. and Blasch, E. 2015. “Resilient and trustworthy dynamic data-driven application systems (DDDAS) services for crisis management environments”. *Procedia Computer Science* 51:2623–2637.

Bazilevs, Y., Deng, X., Korobenko, A., Lanza di Scalea, F., Todd, M.D. and Taylor, S.G. 2015. “Isogeometric fatigue damage prediction in large-scale composite structures driven by dynamic sensor data”. *Journal of Applied Mechanics* 82(9).

Blasch, E., Xu, R., Nikouei, S. Y., & Chen, Y. 2019. “A Study of Lightweight DDDAS Architecture for Real-Time Public Safety Applications Through Hybrid Simulation”. In *Proceedings of the 2019 Winter Simulation Conference*, edited by N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, 762–773. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Blasch, E. 2018. “DDDAS Advantages from High-dimensional Simulation”. In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 1418–1429. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Damgacioglu, H., Bastani, M., & Celik, N. 2018. “A Dynamic Data-Driven Optimization Framework for Demand Side Management in Microgrids”. *Handbook of Dynamic Data Driven Applications Systems*, 489–504.

Darema, F. 2004. "Dynamic Data Driven Applications Systems: A New Paradigm for Application Simulations and Measurements. In *Computational Science-ICCS 2004*, 662–669, Springer.

Darema, F. 2011. "DDDDAS Computational Model and Environments". *Journal of Algorithms & Computational Technology* 5(4):545–560.

Engle, R.D., Langhals, B.T., Grimaila, M.R. and Hodson, D.D. 2018. "The Case for NoSQL on a Single Desktop". In *International Conference on Information and Knowledge Engineering*, Las Vegas, NV.

Fujimoto, R. M., Celik, N., Damgacioglu, H., Hunter, M., Jin, D., Son, Y. J., & Xu, J. 2016. "Dynamic Data Driven Application Systems for Smart Cities and Urban Infrastructures". In *Proceedings of the 2016 Winter Simulation Conference*, edited by T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, 1143–1157. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Jin, D. and Nicol, D.M. 2015. "Parallel Simulation and Virtual-machine-based Emulation of Software-defined Networks". *ACM Transactions on Modeling and Computer Simulation (TOMACS)* 26(1):1-27.

Khaleghi, A.M., Xu, D., Wang, Z., Li, M., Lobos, A., Liu, J. and Son, Y.J., 2013. "A DDDAMS-based Planning and Control Framework for Surveillance and Crowd Control via UAVs and UGVs". *Expert Systems with Applications* 40(18):7168–7183.

Lagor, F.D. and Paley, D.A. 2014. "Active Singularities for Multivehicle Motion Planning in an N-Vortex System". In *International Conference on Dynamic Data-Driven Environmental Systems Science,* edited by S. Ravela and A. Sandu, 334–346. Springer, Cham.

Lecerf, M., Allaire, D. and Willcox, K., 2015. "Methodology for Dynamic Data-driven Online Flight Capability Estimation". *AIAA Journal* 53(10):3073–3087.

Li, X., Tran, P.H., Liu, T. and Park, C. 2017. "Simulation-guided Regression Approach for Estimating the Size Distribution of Nanoparticles with Dynamic Light Scattering Data". *IISE Transactions* 49(1):70–83.

Meijer, E. and Bierman, G., 2011. "A Co-relational Model of Data for Large Shared Data Banks". *Communications of the ACM* 54(4):49–58.

Peng, Y., Xu, J., Lee, L.H., Hu, J. and Chen, C.H., 2018. "Efficient Simulation Sampling Allocation using Multi-fidelity Models". *IEEE Transactions on Automatic Control* 64(8):3156–3169.

Pereira, D. A., Morais, W. O. De, & De, E. P. 2018. "NoSQL Real-time Database Performance Comparison ". *International Journal of Parallel, Emergent and Distributed Systems* 33(2):144–156.

Shi, X., Damgacioglu, H., & Celik, N. 2015. "A Dynamic Data-Driven Approach for Operation Planning of Microgrids". *Procedia Computer Science* 51(1):2543–2552.

Thanos, A. E., Bastani, M., Celik, N., & Chen, C. 2015. "Framework for Automated Control in Microgrids". *IEEE Transactions on Smart Grid* 8(1):209–218.

Xu, J., Zhang, S., Huang, E., Chen, C. H., Lee, L. H., and Celik, N. 2014. "Efficient Multi-fidelity Simulation Optimization". In *Proceedings of the 2014 Winter Simulation Conference*, edited by A. Tolk, S.Y. Diallo, I.O. Ryzhov, L. Yilmaz, S Buckley, and J. A. Miller, 3940–3951. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Xu, J., Zhang, S., Huang, E., Chen, C.H., Lee, L.H., and Celik, N. 2016. "MO2TOS: Multi-fidelity optimization with ordinal transformation and optimal sampling". *Asia-Pacific Journal of Operational Research* 33(3):1650017.

AUTHOR BIOGRAPHIES

ABDURRAHMAN YAVUZ is a graduate student in the Department of Industrial Engineering at the University of Miami. His research interests are applications of data analytics, optimization and simulation techniques to the efficient planning of smart power grids His email adress is axy286@miami.edu.

JOSHUA DARVILLE is a Ph.D. student in the Department of Industrial Engineering at the University of Miami. His research interests lie broadly in application of dynamic data driven adaptive (DDDAS) simulation and optimization of microgrid dispatch. He was appointed captain of the Fisk university rocket team which competed in the NASA student launch competition in 2018. His email address is jmd437@miami.edu.

NURCIN CELIK is an associate professor at the Department of Industrial Engineering at the University of Miami. She received her M.S. and Ph.D. degrees in Systems and Industrial Engineering from the University of Arizona. Her research interests include architectural design and application of dynamic data-driven adaptive simulations for distributed systems. She received the 2017 Presidential Early Career Award for Scientists and Engineers. She received several other awards including the 2020 Provost's Award for Scholarly Activity, ICCS2015 Best Workshop Paper Award, ISERC2015 Best Paper Award, 2014 Eliahu I. Jury Early Career Research Award, and WSC2011 Best Paper Award. Her email address is celik@miami.edu.

JIE XU is an associate professor in the Department of Systems Engineering and Operations Research at George Mason University. He received the M.S. degree in computer science from the State University of New York, Buffalo, in 2004, and the Ph.D. degree in industrial engineering and management sciences from Northwestern University, Evanston, IL, in 2009. His research interests are data analytics, dynamic data driven application systems, stochastic simulation and optimization, with applications in cloud

computing, energy systems, health care, manufacturing, and transportation. His email address is jxu13@gmu.edu.

CHUN-HUNG CHEN is is a professor of Systems Engineering and Operations Research at George Mason University. He has led research projects in stochastic simulation and optimization. He served as Co-Editor of the Proceedings of the 2002 Winter Simulation Conference and Program Co-Chair for 2007 Informs Simulation Society Workshop. He has served on the editorial boards of *IEEE Transactions on Automatic Control*, *IEEE Transactions on Automation Science and Engineering*, *IIE Transactions, Journal of Simulation Modeling Practice and Theory*, and *International Journal of Simulation and Process Modeling*. He received his Ph.D. degree from Harvard University in 1994. His email address is cchen9@gmu.edu.

BRENT T. LANGHALS is an assistant professor of Information Resource Management in the Systems Engineering and Management Department at the Air Force Institute of Technology (AFIT), Wright-Patterson AFB, Ohio USA. He received his Ph.D. in Management Information Systems at the University of Arizona in 2011. His research interests include database design and optimization, data analytics, dynamic data driven application systems, human systems integration and vigilance.

RYAN D. L. ENGLE is an assistant professor of Systems Engineering in the Systems Engineering and Management Department at the Air Force Institute of Technology. He earned his Ph.D. from AFIT in 2019. He is commissioned as a Major in the United States Air Force and served as a Computer Developmental Engineer at Wright-Patterson Air Force Base (AFB) in Dayton, OH, a Systems Engineer at Lackland AFB in San Antonio, TX, and a Test Engineer at Eglin AFB, FL. He holds Department of Defense certifications in Systems Engineering, Test & Evaluation, and Program Management.