COVER SHEET

NOTE:

- ***Please attach the signed copyright release form at the end of your paper and upload as a single 'pdf' file***
- ***This coversheet is intended for you to list your article title and author(s) name only***
- ***This page will not appear in the book or on the CD-ROM***

Title: *A big data management and analytics framework for bridge monitoring*

Authors (names are for example only): Seongwoon Jeong
Rui Hou
Jerome P. Lynch
Hoon Sohn
Kincho H. Law

IMPORTANT All authors' information will appear on the program according to the submission stub on the online submission system (not to the manuscript). The title and author list provided in the manuscript will be for future referencing and citation.

PAPER DEADLINE: ****June 1, 2017****

PAPER LENGTH: ****8 PAGES MAXIMUM ****

Please submit your paper in PDF format. We encourage you to read attached Guidelines prior to preparing your paper—this will ensure your paper is consistent with the format of the articles in the CD-ROM.

NOTE: Sample guidelines are shown with the correct margins. Follow the style from these guidelines for your page format.

Electronic file submission: When making your final PDF for submission make sure the box at "Printed Optimized PDF" is checked. Also—in Distiller—make certain all fonts are embedded in the document before making the final PDF.

--

(FIRST PAGE OF ARTICLE)

ABSTRACT

Bridge health monitoring involves massive volume of data with diverse and complex data types. While the data can potentially enhance the diagnostic and prognostic analyses of the structural condition of a bridge, the massive volume and the complexity of the data pose fundamental management and processing issues. Current practice of bridge health monitoring relies on proprietary servers and legacy data management tools, which are not well suited to meet today's big data processing and management requirements. This paper discusses a cyberinfrastructure framework that takes advantages of state-of-the-art computing technologies to handle the data issues in bridge monitoring applications. We explore cloud computing as a scalable and reliable computing infrastructure service offered by cloud vendors. The use of a distributed NoSQL database system with cloud computing infrastructure facilitates scalability of data storage. In addition, the distributed computing resources in the cloud environment can be dynamically scaled on demand. The proposed framework is implemented for the monitoring of bridges located along the I-275 corridor in Michigan. The framework can effectively cope with changing demands for data management and processing in bridge monitoring.

INTRODUCTION

The deployment of sensors for bridge monitoring has grown with advances in sensor and communication network technologies. Sensor measurement data enables diagnosis for anomaly detection as well as analysis for long-term structural behavior, thereby supporting decision-making for bridge management. Bridge monitoring

Seongwoon Jeong, Dept. of Civil & Environ. Eng., Stanford University, Stanford, CA, USA 94305
Rui Hou, Dept. of Civil & Environ. Eng., University of Michigan, Ann Arbor, MI, USA 48109-2125
Jerome P. Lynch, Dept. of Civil & Environ. Eng., University of Michigan, Ann Arbor, MI, USA 48109-2125
Hoon Sohn, Dept. of Civil & Environ. Eng., KAIST, Daejeon 305-701, Republic of Korea
Kincho H. Law, Dept. of Civil & Environ. Eng., Stanford University, Stanford, CA, USA 94305

systems collect a wide variety of data, such as video images, traffic information and weather data, that can help provide meaningful information about the bridge structure and its behavior. While the data can enhance the diagnostic and prognostic analyses for the structural condition of a bridge, the massive volume and the complexity of the data can pose fundamental data management and processing issues. To take advantage of the big data for bridge monitoring, it is necessary to properly design and develop a computational infrastructure framework that can support large-scale data management and processing [1]. This paper describes a scalable cyberinfrastructure framework for bridge monitoring based on the cloud computing paradigm and distributed computing technologies.

Current practice of civil and infrastructure engineering relies on traditional proprietary server, and file-based or relational database (RDB) systems for data management. Such approach, however, is not effective in meeting today's big data processing and management requirements [2]. Cloud computing, on the other hand, is a new computing paradigm wherein virtualized computing resources can be rapidly provisioned from the shared pool of computing resources over the Internet [3]. Recent advances in cloud computing technologies offer many benefits, such as low maintenance efforts, high scalability and high accessibility [4]. Cloud computing has been proposed in various engineering domains [5-7]. In infrastructure monitoring, there have also been efforts that adopt cloud computing for remote data processing and management [8, 9]. A cloud-based cyberinfrastructure framework for large-scale data management that can seamlessly support data analytics in infrastructure monitoring is desirable.

Our previous works dealt with bridge information model [10] and cloud-based cyberinfrastructure [11]. This paper extends the discussion on large-scale data management and high-performing data analytics. The framework is built upon a cloud computing paradigm in which computing resources can be dynamically added, modified and removed on demands. Given the distributed nature of cloud computing, the framework employs a NoSQL database system that is highly scalable in a distributed computing environment. In addition, we employ a distributed computing engine to build a high-performing computing cluster whose resource and capacity can be easily scaled as demand requires. To facilitate platform-neutral access to the cyberinfrastructure, we build web services that comply with the *de facto* REST web service design style. The proposed cyberinfrastructure has been implemented for the monitoring of bridges along the I-275 corridor located in the state of Michigan.

CLOUD-BASED CYBERINFRASTRUCTURE FRAMEWORK

Figure 1 depicts the conceptual framework of the cloud-based cyberinfrastructure that provides a scalable data management service, as well as a high-performing computing service. The cyberinfrastructure consists of several subcomponents including cloud virtual machines, distributed database, distributed computing cluster and web server. Cloud virtual machines (VMs) are scalable computing infrastructure on which other subcomponents can be deployed. Distributed database is a permanent data store running on the VMs in a decentralized manner. Computing cluster is a system that organizes distributed computing resources to offer the computing power. Lastly, the web server is a system that hosts the web services that expose the database and applications residing in the cloud cyberinfrastructure via platform-neutral web interfaces. The data management and computing services can be accessed by authorized users and computing components (e.g., onsite computer, local desktop computer and end user devices) involved in bridge monitoring via standard web interfaces. In this section, we discuss the details of the cyberinfrastructure components.

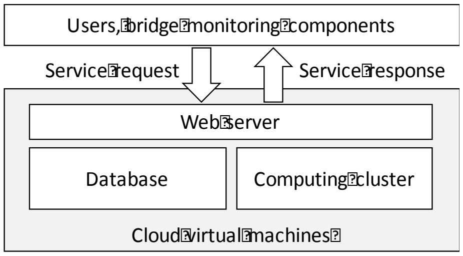
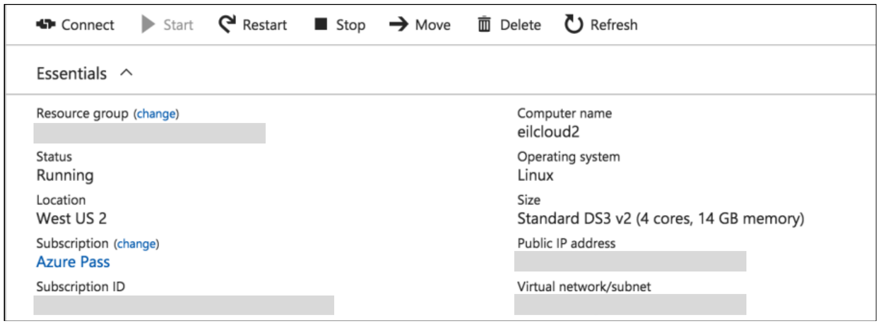


Figure 1. Conceptual framework of cloud-based cyberinfrastructure

CLOUD VIRTUAL MACHINE

A virtual machine (VM) is an emulated computer system that provides the similar functionalities of a physical computer [12]. In cloud computing environment, a VM is offered as an Infrastructure as a Service (IaaS) type computing service on which computing platforms (e.g., database and runtime environment) and applications can be deployed. In this study, we use the term “cloud virtual machine” or “cloud VM” to refer to VM deployed on cloud.

Cloud VMs can be rapidly provisioned from a shared pool of computing resources via cloud service interfaces. Figure 2, as an example, shows an instance of VM running on Microsoft Azure cloud platform (https://azure.microsoft.com/). A cloud service user can configure the type of OS, computing power (e.g., CPU and RAM), storage type (e.g., HDD and SSD) and storage size of VM as needed. It should be noted that the configuration of created VM can be easily modified as the computing requirement changes. Once the VM is up and running, the user can use the VM through the Secure Shell (SSH) protocol, similar to using a proprietary remote server.



Resource group (change)
[redacted]

Status
Running

Location
West US 2

Subscription (change)
Azure Pass

Subscription ID
[redacted]

Computer name
eilcloud2

Operating system
Linux

Size
Standard DS3 v2 (4 cores, 14 GB memory)

Public IP address
[redacted]

Virtual network/subnet
[redacted]

Figure 2. Virtual machine running on a public cloud

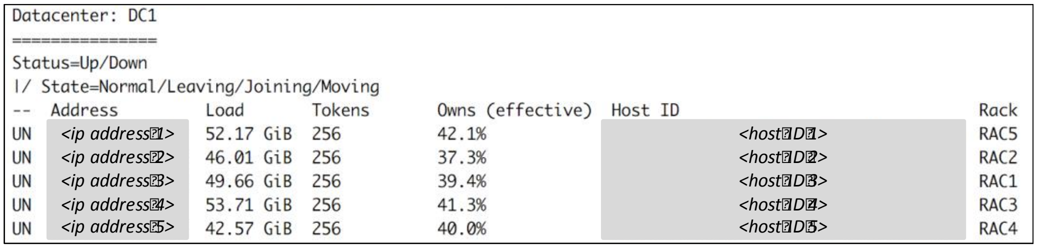
One of the advantages of cloud VM over the proprietary physical server is its scalability, in particular horizontal scalability that increases computing capacity of a system by adding more VMs. Since the computing capacity of a single VM has limited resource, the use of distributed computing system is inevitable to meet any demanding computing requirements. In the cloud computing environment, users can create as many VMs as needed. With the use of appropriate middleware, the VMs can be glued together to provide a high computing throughput and storage capacity.

DISTRIBUTED DATABASE

To handle large amount of data involved in SHM, the selection of an appropriate database management tool is important. Relational database (RDB) or file-based system has been a typical choice for data management in SHM applications. However, RDB is not effective to manage large and complex data. NoSQL database systems have been proposed as alternatives to RDB to satisfy the data processing demand requirement [13]. Many NoSQL database systems are designed to handle large-scale distributed data management. The use of NoSQL database can help obtain high scalability and performance required for bridge monitoring applications.

The cyberinfrastructure framework employs Apache Cassandra database (http://cassandra.apache.org/), which is one of the most widely used distributed NoSQL database. Cassandra database adopts the peer-to-peer (P2P) architecture, in which every database node (i.e., a database instance running on a computer) is self-sufficient and has an identical role, thereby preventing single point of failure. Furthermore, the P2P architecture enables dynamic scaling where the database capacity (i.e., storage size and throughput) can be linearly scaled as new nodes are added. Therefore, the combination of scalable cloud computing infrastructure and the P2P-based Cassandra database can guarantee very high scalability.

To construct a P2P-based distributed database, each database node has to have network information (e.g., IP address) of other nodes. Cassandra designates a few database nodes as “seed” nodes that database nodes can access to share their network information when starting up. Once the database nodes “handshake” each other, the database nodes are glued together and start to serve as a distributed database. Figure 3, for example, shows the status of a distributed Cassandra database running on five cloud VMs. The status indicates that the distributed database includes five database nodes and every node is in “UN” (i.e., up and normal) state.



```
Datacenter: DC1
===============
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address           Load       Tokens       Owns (effective)  Host ID                                  Rack
UN   <ip address 1>   52.17 GiB  256          42.1%                          <host ID 1>                 RAC5
UN   <ip address 2>   46.01 GiB  256          37.3%                          <host ID 2>                 RAC2
UN   <ip address 3>   49.66 GiB  256          39.4%                          <host ID 3>                 RAC1
UN   <ip address 4>   53.71 GiB  256          41.3%                          <host ID 4>                 RAC3
UN   <ip address 5>   42.57 GiB  256          40.0%                          <host ID 5>                 RAC4
```

Figure 3. Cassandra data schema for managing bridge monitoring data

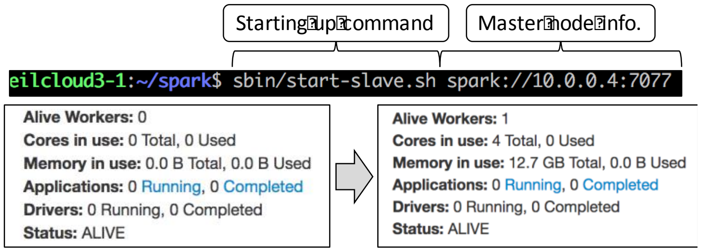
Another important consideration in choosing a database system is the flexible data structure, because bridge monitoring involves a wide variety of data types, including but not limited to time-series sensor data, object-oriented bridge information and image data. Cassandra database offers flexible data structure that can elegantly handle bridge monitoring data and relevant information. Based on the flexible data structure, data schema for managing bridge monitoring data and relevant bridge information can be defined as described in [10].

COMPUTING CLUSTER

As the amount of data in bridge monitoring applications increases, the computing capacity needed to process and analyze the data will also increase. Since bridge monitoring and management involves real-time anomaly checking as well as long-term trend analysis on a regular basis, the required computing capacity varies over time. Given the varying computing requirement, the use of cloud computing has advantages over the physical computers because users can easily change the available computing capacity as needed, thereby using the optimal amount of resources.

The cyberinfrastructure framework employs Apache Spark (https://spark.apache.org/) to build a scalable computing cluster in a distributed cloud computing environment. Spark organizes resources distributed over multiple nodes (i.e., physical or virtual machines) and makes the distributed system viewed as a single high-performing computing system. Specifically, Spark adopts the master-slave architecture in which the master node has control over slave nodes to distribute workloads and process jobs in parallel.

To construct a Spark cluster, a master node needs to be deployed in a cloud VM. Then, slave nodes, each of which is deployed in a cloud VM, can be connected to the cluster by starting up slave node with the input argument specifying the address of the master node. Figure 4, for example, shows a Spark cluster instance where a new slave node is attached to an empty cluster that has no slave node. The slave nodes attached to a cluster can also be detached by stopping the slave node. The simple process of attaching and detaching slave nodes enables Spark to be highly scalable in that the computing capacity can be easily modified according to computing demands.



```
eilcloud3-1:~/spark$ sbin/start-slave.sh spark://10.0.0.4:7077
```

```
Alive Workers: 0
Cores in use: 0 Total, 0 Used
Memory in use: 0.0 B Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE
```

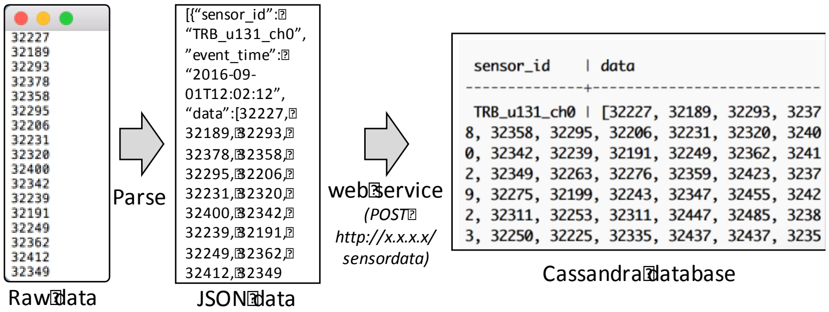
```
Alive Workers: 1
Cores in use: 4 Total, 0 Used
Memory in use: 12.7 GB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE
```

Figure 4. Constructing a Spark cluster

Spark cluster provides diverse functionalities, including Spark SQL for performing queries over data sets and Spark Machine Learning library (MLlib) for supporting data-driven analyses. Spark can be connected to the Cassandra database using Spark-Cassandra Connector (DataStax, Inc. 2014) so that the data sets in the Cassandra can be imported to Spark cluster for data analyses. Furthermore, many software packages, such as Scikit-learn integration package for Spark (https://github.com/databricks/spark-sklearn), have been developed to enrich the functionality of Spark. In the cyberinfrastructure, we use Spark to enable distributed data analytics for bridge monitoring.

WEB SERVER

A web server is a computer system that receives HTTP requests from client-side systems, processes the requests and sends HTTP responses back to the client-side systems. The cyberinfrastructure framework deploys web servers to host web services that deliver access to the distributed data storage and applications residing in the cloud server. The web services are designed based on the *de facto* REST design style to provide lightweight and platform-neutral web services. The current design of web server includes web services for storing and retrieving bridge monitoring data (e.g., time-series sensor data, bridge information model and video image data). For example, Figure 5 depicts the use of web service for sensor data storage to transmit sensor data from the onsite computer to the database. The raw sensor data is first transformed to a JSON format that the web service can read. Once the raw data is transformed, the onsite computer invokes the web service by sending a HTTP request with the JSON data as an attachment. Receiving the request, the web service parses and stores the JSON data to the Cassandra database. In the proposed cyberinfrastructure framework, Node.js (https://nodejs.org/) is employed to construct the web server.



```
32227
32189
32293
32378
32358
32295
32206
32231
32320
32400
32342
32239
32191
32249
32362
32412
32349
```

```
[{“sensor_id”: 
“TRB_u131_ch0”,
”event_time”: 
“2016-09-
01T12:02:12”,
“data”:[32227, 
32189, 32293, 
32378, 32358, 
32295, 32206, 
32231, 32320, 
32400, 32342, 
32239, 32191, 
32249, 32362, 
32412, 32349
```

```
 sensor_id    | data
--------------+--------------------------
 TRB_u131_ch0 | [32227, 32189, 32293, 3237
8, 32358, 32295, 32206, 32231, 32320, 3240
0, 32342, 32239, 32191, 32249, 32362, 3241
2, 32349, 32263, 32276, 32359, 32423, 3237
9, 32275, 32199, 32243, 32347, 32455, 3242
2, 32311, 32253, 32311, 32447, 32485, 3238
3, 32250, 32225, 32335, 32437, 32437, 3235
```

Figure 5. Web service example: Sensor data store service

IMPLEMENTATION

The proposed cyberinfrastructure framework has been implemented and tested for the bridge monitoring system on the I-275 corridor in Michigan [11]. Sensor measurement data has been collected from wireless sensor networks installed on two bridges namely the Telegraph Road Bridge (TRB) and the Newburg Road Bridge (NRB) as shown in Figure 6. Furthermore, the bridge monitoring system involves other relevant data, such as traffic video images and bridge finite element model.

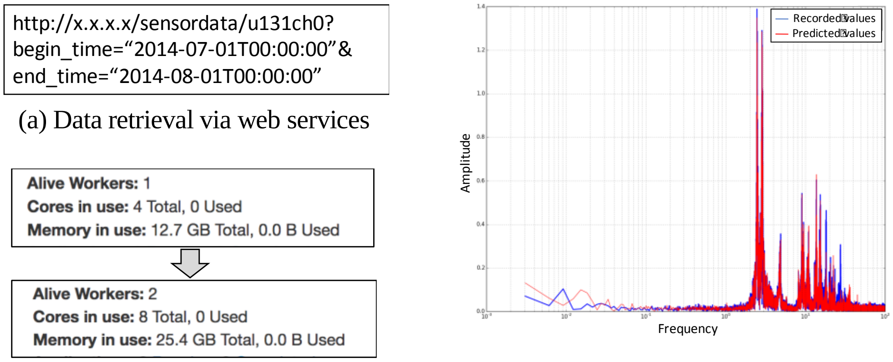
(a) Telegraph Road Bridge

(b) Newburg Road Bridge

Figure 6. Bridges installed with bridge monitoring system on the I-275 corridor

To implement the cyberinfrastructure, we first create seven Linux-based VMs on the Microsoft Azure cloud platform. Cassandra database is installed on five VMs. A single VM is used to construct a single-node Spark computing cluster where the VM serves both the master and slave nodes. In addition, the last VM is used to implement the Node.js web server and to deploy web services written in JavaScript. Once the components of cyberinfrastructure are implemented, sensor data and other bridge information are uploaded to the Cassandra database in the cloud via standard web services. The data stored in the database can be retrieved using web services. To analyze retrieved data, Spark computing cluster can be utilized.

As a demonstrative example, we perform a data-driven analysis for the reconstruction of sensor data [14]. To perform the analysis, sensor information and sensor data are retrieved by invoking web services as shown in Figure 7(a). The retrieved data is analyzed using Scikit-learn package on a Spark cluster. To perform an intensive analysis effectively, the number of Spark computing nodes can be temporarily increased as shown in Figure 7(b) by creating a new VM. Finally, Figure 7(c) shows the data analysis results where the data sets are retrieved from the Cassandra database and the analysis is performed on the Spark cluster in a distributed manner.



```
http://x.x.x.x/sensordata/u131ch0?
begin_time="2014-07-01T00:00:00"&
end_time="2014-08-01T00:00:00"
```

(a) Data retrieval via web services

Alive Workers: 1
Cores in use: 4 Total, 0 Used
Memory in use: 12.7 GB Total, 0.0 B Used

Alive Workers: 2
Cores in use: 8 Total, 0 Used
Memory in use: 25.4 GB Total, 0.0 B Used

(b) Scaling Spark computing cluster

(c) Analysis results

Figure 7. Cassandra database implemented on Microsoft Azure cloud

SUMMARY

In this study, a cloud-based cyberinfrastructure framework for scalable data management and high-performing computing is described. The framework is composed of cloud virtual machines, distributed database system, cluster computing engine and web server. The cloud virtual machines serve as a scalable computing infrastructure for deploying computing platforms and applications. For the scalable data management on distributed cloud computing environment, we employ Cassandra database, a P2P-based NoSQL database system, which not only prevents single point of failure, but also improves scalability and processing performance. For the high-performing computing cluster, we employ Spark cluster that can be scaled easily according to computing demand. The web server is employed to provide standard web interfaces through which users and client-side systems can access and utilize the cyberinfrastructure. The developed framework has been implemented for the bridge monitoring system on the I-275 corridor in Michigan. The results show that the proposed framework not only manages large and complex data collected from bridge monitoring, but also copes with the changing needs for data processing.

ACKNOWLEDGMENTS

This research is supported by a Grant No. 13SCIPA01 from Smart Civil Infrastructure Research Program funded by Ministry of Land, Infrastructure and Transport (MOLIT) of Korea government and Korea Agency for Infrastructure Technology Advancement (KAIA). The research is also partially supported by a collaborative project funded by the US National Science Foundation (Grant No. ECCS-1446330 to Stanford University and Grant No. ECCS-1446521 to the University of Michigan). The authors thank the Michigan Department of Transportation (MDOT) for access to the Telegraph Road Bridge and the Newburg Road Bridge and for offering support during installation of the wireless monitoring system. Any opinions, findings, conclusions or recommendations expressed in this paper are solely those of the authors and do not necessarily reflect the views of NSF, MOLIT, MDOT, KAIA or any other organizations and collaborators.

REFERENCES

1. Law, K. H., Smarsly, K. and Wang, Y. 2014. "Sensor data management technologies for infrastructure asset management," in *Sensor Technologies for Civil Infrastructures: Applications in Structural Health Monitoring*, (Eds., Wang, M.L., Lynch, J.P. and Sohn, H.), Woodhead Publishing, Cambridge, UK, 2(1), 3-32.
2. Stonebraker, M., Madden, S., Abadi, D. J., Harizopoulos, S., Hachem, N. and Helland, P. 2007. "The end of an architectural era (it's time for a complete rewrite)," in *33rd International Conference on Very Large Data Bases*, pp. 1150–1160.
3. Mell, P. and Grance, T. 2011. "The NIST definition of cloud computing - Recommendations of the National Institute of Standards and Technology," NIST Special Publication 800-145, Computer Science Division, Information Technology Laboratory, National Institute of Standards.
4. Zhang, Q., Cheng, L. and Boutaba, R. 2010. "Cloud computing: state-of-the-art and research challenges," *Journal of Internet Services and Applications*, 1(1), 7–18.
5. Lea, R. and Blackstock, M. (2014). "City Hub: A Cloud-Based IoT Platform for Smart Cities," in *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, pp. 799-804.
6. Ye, X. and Huang, J. 2011. "A framework for cloud-based smart home," in *2011 International Conference on Computer Science and Network Technology*, pp. 894-897.
7. Das, M., Cheng, J. C. P. and Kumar, S. S. 2015. "Social BIMCloud: a distributed cloud-based BIM platform for object-based lifecycle information exchange," *Visualization in Engineering*, 3(8), pp. 1-20.
8. Liao, Y., Mollineaux, M., Hsu, R., Bartlett, R., Singla, A., Raja, A., Bajwa, R. and Rajagopal, R. 2014. "Snowfort: An open source wireless sensor network for data analytics in infrastructure and environmental monitoring," in *IEEE Sensors Journal*, 14(12), pp. 4253-4263.
9. Alampalli, S., Alampalli, S. and Ettouney, M. 2016. "Big data and high-performance analytics in structural health monitoring for bridge management," in *2016 SPIE Smart Structures/NDE Conference*, art no. 980315.
10. Jeong, S., Hou, R., Lynch, J. P., Sohn, H. and Law, K. H. 2017. "An information modelling framework for bridge monitoring," *Advances in engineering software* (accepted).
11. Jeong, S., Hou, R., Lynch, J. P., Sohn, H. and Law, K. H. 2017. "A distributed cloud-based cyberinfrastructure framework for integrated bridge monitoring," in *2017 SPIE Smart Structures/NDE Conference*, art no. 101682W.
12. Smith, J.E. and Nair, R. 2005. "The architecture of virtual machines," *Computer*, 38(5), pp.32-38.
13. Hecht, R. and Jablonski, S. 2011. "NoSQL evaluation: A use case oriented survey," in *2011 International Conference on Cloud and Service Computing*, pp. 336-341.
14. Law K. H., Jeong, S. and Ferguson M. "A data-driven approach for sensor data reconstruction for bridge monitoring," in *2017 World Congress on Advances in Structural Engineering and Mechanics* (submitted).