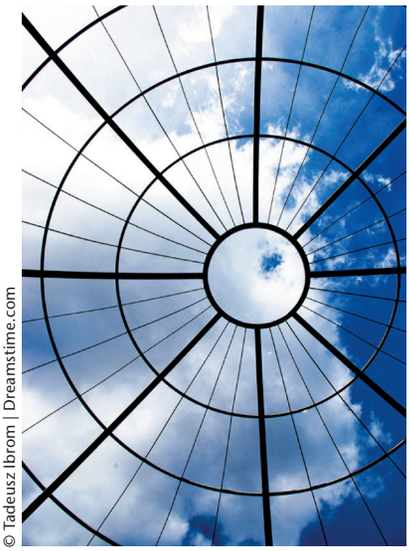
EDITOR: Irena Bojanova, US National Institute of Standards and Technology, irena.bojanova@computer.org



Monitoring Internet of Things Application Ecosystems for Failure

Ellis Solaiman and Rajiv Ranjan, *Newcastle University, UK*
Prem Jayaraman, *Swinburne University of Technology, Australia*
Karan Mitra, *Lulea University of Technology, Sweden*

The Internet of Things (IoT) paradigm promises to connect billions of devices to the Internet. This paradigm is well placed to benefit a wide range of application domains, including power and heating grids, home systems, agriculture, manufacturing, healthcare, and environmental monitoring. The IoT is expected to integrate heterogeneous data sources such as sensors and sensor networks (for instance, power grids), smart mobile devices, and social media (Twitter, Facebook, and so on) to make the aforementioned application domains "smarter." Cloud computing is already becoming the de facto platform for hosting and processing the big data these sources generate.

The IoT Application Ecosystem

Figure 1 provides an overview of the IoT application ecosystem, describing the relationships between *edge devices* (physical layer), the *network* (communication layer), and the *cloud* (cloud layer). As depicted in the figure, the physical layer is composed of embedded systems and sensors that could include a wide range of sensing and actuation devices (GPS, heart rate monitors, temperature sensors, and so on), smartphones, and smart vehicles (such as the Google car). In the communication layer, these devices are interwoven with various ubiquitous communication capabilities that let them be networked and connected to the Internet. The cloud layer hosts hardware and software resources that implement big data platforms such as Apache Spark and Apache Hadoop to process, analyze, and visualize actionable outcomes from IoT data. These actionable responses are then propagated back into the physical world via actuators. The IoT application ecosystem also encompasses a human aspect wherein humans provide data, act on analyzed data, and make informed decisions.

Consider a healthcare application that uses IoT wearable technologies such as a smart watch, heart rate monitor, and accelerometers (as shown in Figure 1). If there is a sudden drop in the user's heart rate, an ambulance could automatically be informed, find the patient location via GPS, and potentially save a life. But what if the device software crashes, the analytics engine running the cloud crashes or is unable to detect the event in time, or the communication network assigns these messages as low-priority and hence introduces delays? In short, the key challenge is how do we monitor the monitors? Ultimately, the success of the aforementioned IoT applications will depend on end-to-end monitoring and verification of the sensors, network communication resources, and the cloud systems (cloud layer) that form the integral parts of an end-to-end IoT ecosystem.[1]

Monitoring Techniques and Frameworks

Over the past 20 years, a large body of research has led to the development of several techniques and frameworks for monitoring the performance of hardware and

 Published by the IEEE Computer Society

application resources in distributed systems such as grids, clusters, and clouds. Monitoring tools such as R-GMA, Hawkeye, the Network Weather Service (NWS), and the Monitoring and Directory Service (MDS) were popular in the grid and the cluster computing era. However, these tools were only concerned with monitoring performance metrics at the hardware resource level (CPU percentage, TCP/IP performance, available non-paged memory, and so on), and not at the application level (event detection and decision-making delay in the context of particular IoT applications). On the other hand, cluster-wide monitoring frameworks (Nagios and Ganglia, adopted by big data orchestration platforms such as YARN, Hadoop, and Spark) provide information about hardware resource-level metrics (cluster, CPU, or memory utilization). In the public cloud computing space, monitoring frameworks (such as Amazon CloudWatch, used by Amazon Elastic MapReduce and Azure Fabric Controller)[2-4] typically monitor an entire CPU resource as a black box, and cannot monitor application-level performance metrics specific to IoT ecosystems. However, frameworks such as Monitis (http://portal.monitis.com) and Nimsoft (www.nimsoft.com/solutions/nimsoft-monitor/cloud) can monitor application-specific performance metrics, such as Web server response time. CASViD is an architecture that tackles service-level agreement violation at the application level,[3] but the application model described (for multilayered Web applications) is fundamentally different from IoT application ecosystems.

In summary, none of these approaches can monitor and detect root causes of failures and performance degradation for entire end-to-end IoT ecosystems across edge devices (physical layer), the network (communications layer), and the big data platforms (cloud layer). Developing formal approaches for monitoring end-to-end IoT ecosystems is what we term the "grand challenge." Our recent work in developing performance models for big data applications that use machine learning techniques running on batch processing frameworks such as Hadoop highlights the dependency between the various components of the big data platforms and hardware resources, and the effects this dependency has on the performance of application-level metrics such as event-detection delay.[5] As we have discussed, current platforms and techniques for monitoring the IoT and cloud computing fall short of this grand challenge.

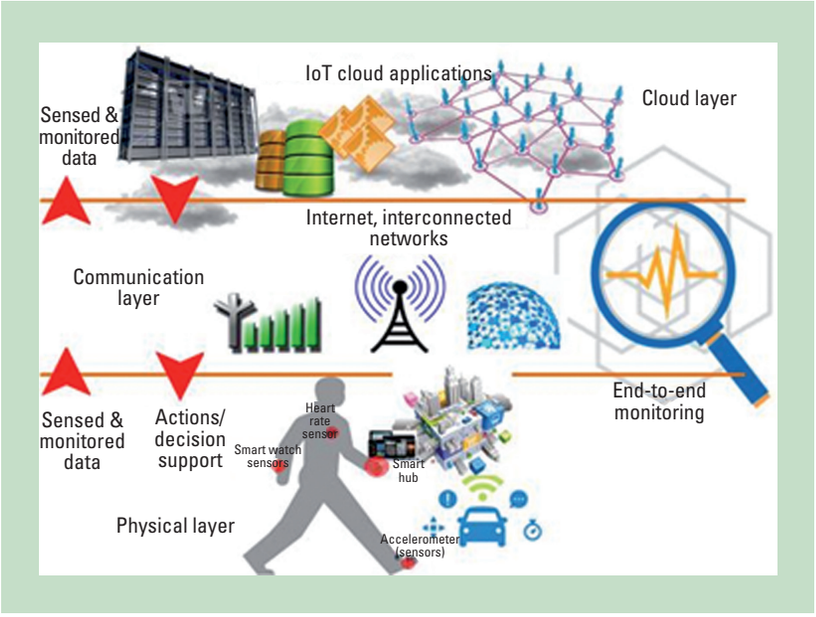


Figure 1. The Internet of Things application ecosystem. This ecosystem hinges on the relationships among edge devices, the network, and the cloud.

Research Directions

The monitoring challenge is multilayered. For example, continuous sensor monitoring is essential for preventing data loss and enabling the dynamic adjustment of sensor performance during critical events (for instance, increasing the frequency of the heart rate sensor during a heart attack, or fine tuning the measurement of a moisture sensor during or after flooding to detect landslides). In many cases, sensor data is not recoverable if the loss occurs due to inefficient caching and inappropriate communication protocols. And if the lost data is critical to determining the probability that an important event (such as a heart attack or flood) will occur, then there might not be enough time to respond to or recover from a potential disaster or life-threatening situation. Additionally the ability to observe accumulated results in real time is necessary to ensure data integrity when it is collected.[6] Multiple factors could lead to sensor malfunctions, such as calibration errors, environmental conditions, attacks, decay of sensor energy, and so on. Different monitoring techniques must thus be investigated, combined, and tested, ranging from simple techniques such as profiling sensor baseline behavior to complex ones such as fine-grained diagnosis techniques for sensors. Also, different topologies for monitoring the IoT ecosystem need to be considered—for

example, whether the monitoring service should be distributed to the nodes or performed centrally. Furthermore, before monitoring services can be deployed, what needs to be monitored and how must be specified in the form of state-aware policies that have been coded correctly, and that do not conflict with each other.[7]

The monitoring problem at the big data platform level (the cloud layer; see Figure 1) is also complicated.[8] This is because the performance metrics related to the big data platform (software implementing programming models) and cloud resources are not necessarily the same—these can include key performance metrics such as throughput and latency in distributed messaging queuing systems (Apache Kafka); response time for batch processing systems (Hadoop); response time for processing top-*k* queries in transactional systems (Apache Hive); read/write latency and throughput for distributed file systems; and utilization and throughput for CPU resources. Therefore, future research needs to focus on both how these performance metrics could be defined and formulated coherently across IoT application ecosystems, and how various performance metrics should be combined to give a holistic view of IoT data flowing from sensors to multiple software frameworks and cloud resources. Specifically, there is an important need to investigate end-to-end and scalable algorithms (for example, by significantly extending distributed data structures such as self-balanced trees and distributed hash tables) for monitoring performance across IoT sensors, big data programming models, and hardware resources. Monitoring algorithms could also encompass the intelligence to cater to the energy constraints of IoT sensors. Finally, novel IoT application ecosystem monitoring middleware that realizes such algorithms needs to be developed.

The realization of useful and dependable services created from the interconnection of sensors, actuators, social media, networks, and clouds will require the IoT research community to tackle a number of important challenges. Such services will require potentially billions of devices to accurately sense the environments in which they are placed, and then to securely transmit the generated data to multiple possible destinations using energy-efficient technologies that are yet to be developed and refined. The sheer complexity of IoT applications—in which data is transmitted from multiple real-time as well as historical data sources to the cloud for storage and processing—means that we will require new protocols and big data processing frameworks that can deal with the volume, variety, and variability of the data transmission rates.

Additionally, a major challenge whenever data is accessed and transmitted across complex networks is security. There are many ways in which IoT systems can be targeted and attacked. Any vulnerable component in the end-to-end IoT ecosystem could mean unauthorized access to the entire system, disabling key components, or sending misleading data to users. We believe that our research into the development of sophisticated capabilities for monitoring IoT ecosystems is essential for tackling such important IoT research challenges. The challenge of our work is to provide tools and methods that can accurately provide fine-grained monitoring of specific components and layers, not only to ensure that services operate correctly without failure, but also to provide researchers and developers with the ability to collect the data necessary for ensuring that IoT applications are dependable, secure, and efficient. IT

References

1. E. Bertina, S. Nepal, and R. Ranjan, "Building Sensor-Based Big Data Cyberinfrastructures," *IEEE Cloud Computing*, vol. 2, no. 5, 2015, pp. 64–69.
2. K. Alhamazani et al., "Cross-Layer Multi-Cloud Real-Time Application QoS Monitoring and Benchmarking as-a-Service Framework," *IEEE Trans. Cloud Computing*, to appear; doi: 10.1109/TCC.2015.2441715.
3. V.C. Emeakaroha et al., "CASViD: Application Level Monitoring for SLA Violation Detection in Clouds," *Proc. IEEE 36th Ann. Computing Software and Applications Conf.* (COMPSAC), 2012, pp. 499–508.
4. L. Romano et al., "A Novel Approach to QoS Monitoring in the Cloud," *Proc. 1st Int'l Conf. Data Compression, Comm. and Processing* (CCP), 2011, pp. 45–51.
5. M. Wang et al., "A Case for Understanding End-to-End Performance of Topic Detection and Tracking Based Big Data Applications in the Cloud," *Proc. EAI Int'l Conf. Cloud, Networking for IoT Systems*, 2015.
6. B. Shebaro, D. Midi, and E. Bertino, "Fine-Grained Analysis of Packet Losses in Wireless Sensor Networks," *Proc. 11th Ann. IEEE Int'l Conf. Sensing, Comm., and Networking* (SECON), 2014, article no. 38.
7. E Solaiman, I. Sfyrakis, and C. Molina-Jimenez, "High Level Model Checker-Based Testing of Electronic Contracts," *Comm. Computer and Information Science*, vol. 581, 2016, pp. 193–215.
8. R. Ranjan, "Streaming Big Data Processing in Datacenter Clouds," *IEEE Cloud Computing*, vol. 1, no. 1, 2014, pp. 78–83.

Ellis Solaiman *is a teaching fellow in computing science at Newcastle University, UK. He works on projects related to trust management, quality-of-service*

monitoring, and service-level agreements (SLAs). Solaiman's recent work is in the area of electronic contract specification, monitoring, and verification (model checking) within distributed systems (business, cloud, and Internet of Things). Contact him at ellis.solaiman@ncl.ac.uk.

Prem Prakash Jayaraman *is a research fellow at Swinburne University of Technology, Melbourne, Australia. He is working on projects related to the Internet of Things, cloud and edge computing, and mobile computing. Contact him at prem.jayaraman@gmail.com.*

Karan Mitra *is an associate senior lecturer in the School of Computing Science at the Lulea University of Technology, Sweden. His research interests are in quality-of-service monitoring and benchmarking in clouds, and mobile and pervasive computing. Contact him at karan.mitra@ltu.se.*

Rajiv Ranjan *is a reader (associate professor) in computing science at Newcastle University, UK. His research is in emerging areas in parallel and distributed systems, including cloud computing, the Internet of Things, and big data. Ranjan serves on the editorial boards of international journals, including* IEEE Transactions on Computers, IEEE Transactions on Cloud Computing, IEEE Cloud Computing, *and* Future Generation Computer Systems. *Contact him at raj.ranjan@ncl.ac.uk.*