DEPARTMENT: Blue Skies

The Next Grand Challenges

Integrating the Internet of Things and Data Science

Rajiv Ranjan
Newcastle University

Omer Rana
Cardiff University

Surya Nepal
Data61, CSIRO

Mazin Yousif
T-Systems, International

Philip James, Zhenyu Wen, Stuart Barr, Paul Watson
Newcastle University

Prem Prakash Jayaraman, Dimitrios Georgakopoulos
Swinburne University of Technology

Massimo Villari, Maria Fazio
University of Messina

Saurabh Garg
University of Tasmania

Rajkumar Buyya
University of Melbourne

Lizhe Wang
Chinese Academy of Sciences

Albert Y. Zomaya
University of Sydney

Schahram Dustdar
TU Wien

Editor: Rajiv Ranjan:
raj.ranjan@ncl.ac.uk

This article discusses research challenges related to devising a new IoT programming paradigm for orchestrating IoT applications' composition and data processing across heterogeneous computing infrastructure (Cloud, Edge, and Things).

In the last decade, we have been transitioning from a data-poor to a data-rich world with the promise of unparalleled intelligence. Such transition will definitely require significant investments in every aspect in our societies including social, political, economic and cultural. Much of the (unprecedented) increase in data generation can be attributed to the abundance of mobile devices and wearables, the increase of instrumentation in every industry vertical, the mass adoption of social networks and the digitization of every aspect of our lives. Generically, the bulk of such data collection falls under the Internet of Things (IoT).[1–5] IoT data comes from a variety of sources that can be classified into (a) machine-based (e.g., environmental, weather, air quality, water quality, flows, traffic speeds, people flows and GPS location) or (b) people-based (e.g., social media, crowd sourced data collection, and simple text messaging) providing data and situational observations associated with events.

The emergence of computing paradigms such as Edge, Fog, and Osmotic Computing for supporting the analysis of data near the data sources are especially applicable for IoT use cases where insights need to be actioned on in the least amount of time possible.[1,6] Figure 1 depicts a typical IoT application infrastructure consisting of the *Things*, the *Edge*, and the *Cloud* layers. The layers are connected to each other in a plethora of ways. But the most interesting one is connecting the Things to the Edge of directly to the

Copublished by the IEEE CS and IEEE ComSoc
2325-6095/18/$33.00 USD ©2018 IEEE

cloud. Examples of networking protocols include (but not limited to) WiFi, Cellular (e.g., 4G & 5G), Bluetooth, Bluetooth Low Energy, LoRa-WAN [Lora], and Narrowband IoT (NB-IoT). On the other hand, the Edge layer consists of network gateways/middleboxes, Content Delivery Networks (CDNs), or micro datacenters, which provide limited computing and storage resources. The edge resources usually communicate with Cloud layer via wide Area Networks (WANs). The last layer is the Cloud, which is provided by different cloud providers such as Amazon, Microsoft, Tencent, Google and Alibaba. Cloud datacenters offer unlimited computational resources and their cloud services are usually offered on a pay-as-you-go fashion.

The increase in data collection, along with advances in infrastructure development and intelligence, have led to an opportunity for developing several new usage scenarios, ranging from smart cities, smart transportation, smart health care, to Industry 4.0 as depicted in Figure 2. However, the potential of these different paradigms/technologies requires coordination across several layers, leading to important research challenges to be addressed. Currently, existing IoT applications processing data run on remote Cloud infrastructure. To support new application scenarios, novel software/application abstractions are needed that can utilize distributed and dynamic infrastructure supported at Edge and Things layers (as shown in Figure 1). Moreover, IoT data is typified by the heterogeneity of data formats and types, which usually results in bespoke platforms and code that make subsequent integration and processing problematic and time-consuming. The provenance of data is another key aspect that IoT needs to address, not just to ensure the physical integrity of bytes produced, but to be able to trace decision making from model outputs to individual sensors or sensor platforms. This is significant to enable “trust” to be established in the analysis that is carried out on such data. IoT systems currently deployed are largely passive observers of the environment that transmit data to a remote location (with a varying and limited degree of on-board processing). Retasking this one-way behavior in a reliable fashion (e.g. changing sampling rates triggered by external stimuli) is a prerequisite for developing and deploying future IoT applications.

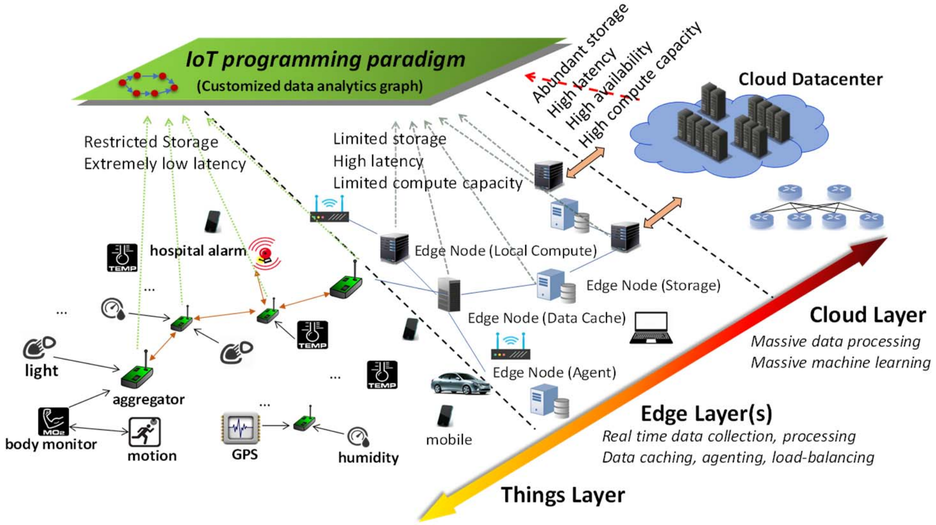

Figure 1. A typical IoT application infrastructure of a healthcare use case showing of Things, Edge, and Cloud layers.

In this column, we provide an IoT roadmap, moving from the (significant) existing research focus on handling streaming IoT data to developing application instances that have intelligence to adapt their behavior/operation based on several external (e.g. environment) and internal (e.g. application) QoS stimuli. We present our vision and associated challenges in the areas of IoT: (i) application composition, (ii) dynamic data management, and (iii) service orchestration across Things/Edge/Cloud infrastructure. Our discussion is based around the IoT application architecture shown in Figure 3 that illustrates how user requirements can be coupled with efficient utilization of computing resources and data provided by Cloud, Edge, and Things layers.

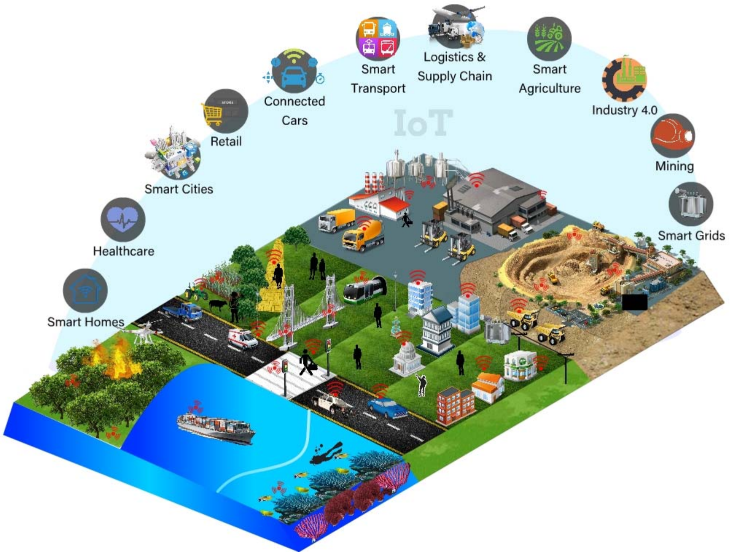

Figure 2. Examples of an IoT-driven smart world: from Smart Homes to Smart Retail, Industry 4.0 and Smart Grids.

In this environment, a user submits requirements to an IoT application orchestrator which identifies: types of services, data sources, QoS metrics that need to be monitored to meet user requirements. Following that, the orchestrator generates a graph showing services needed to realise application requirements. A data management component subsequently maintains and monitors IoT data sources including data governance, data analysis, and data warehousing. An IoT application orchestrator therefore considers IoT data sources as services with specific functions and Service Level Agreements (SLAs). The generated application graph will be deployed to computing resources according to the provisioned data resources and other SLA constraints. This deployment is not undertaken in a one-shot manner, requiring refinement and adaptation based on changes in the operating environment.

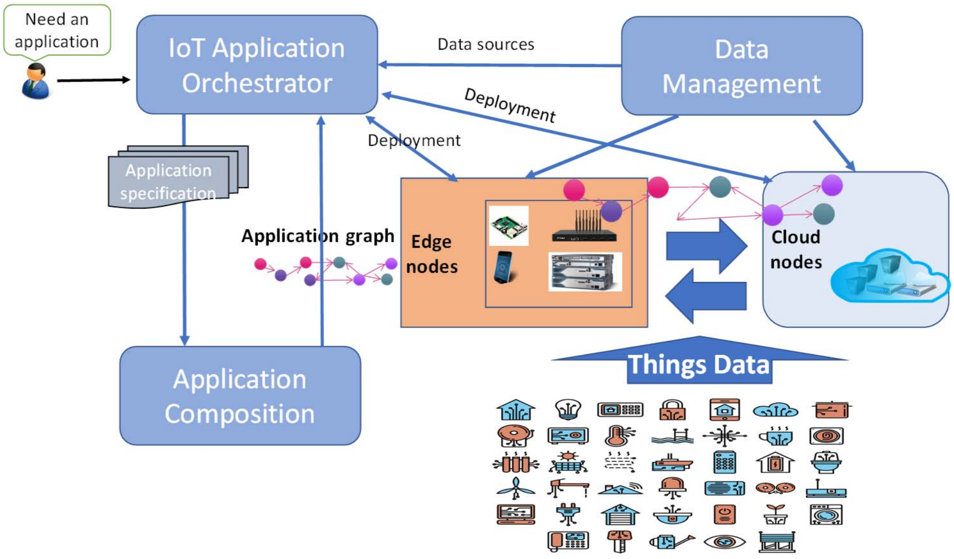

Figure 3. IoT application orchestration, showing interplay among different system components.

IOT APPLICATIONS COMPOSITION

IoT Data Sources

In general, IoT data can be of different types and can be collected at different rates and time scales. The data can be generated by two types of sources: Things (e.g. environment monitoring sensors, GPS, etc.) and People (e.g. social networking apps). Things, such as sensors, provide quantitative observational values; they provide measurement of physical phenomenon at different levels of precision. Moreover, these measurement data can be generated in different formats such as images from cameras, audio from satellites, and text from GPS. Conversely, social sensors provide a qualitative observation of a situation very quickly and succinctly. An IoT application, such as real-time flood forecasting and warning, requires the integration of machine and social sensors data to provide complementary and corroborative information. This aggregate data can be semantically tagged to generate and distribute events of interest (to particular subscribers). One of the key data science research question is how to identify IoT data sources that are most appropriate for a given IoT application context/use case. To answer this question, we need to overcome the following five challenges summarized by Baltrusaitis and colleagues:[7]

1. **Representation:** Structure and represent the data to facilitate multiple modalities, exploiting the complementarity and redundancy of different data sources.
2. **Translation:** Interpret data from one modality to another, i.e., provide a translator that allows the modalities to interact with each other for enabling data exchange.
3. **Alignment:** Identify the relation among modalities. This requires identifying links between different types of data.
4. **Fusion:** Fuse information from different modalities (e.g., to predict).
5. **Co-learning:** Transfer knowledge among modalities. This explores the field of how the knowledge of a modality can help or enhance a computational model trained on a different modality.

A Standard way to describe an IoT Computation Unit

There is a requirement to define a basic IoT computation unit (a software abstraction) that can be ubiquitously deployed across different infrastructures (as proposed by the Osmotic Computing programming paradigm)[1] and can be migrated based on various potential “triggers (e.g., performance, security/privacy or cost)”. A software abstraction is used to describe a basic IoT computation unit called MicroELement (MEL).[2] A MEL encapsulates:

1. **MicroServices (MS),** which implement specific functionalities and can be deployed and migrated across different virtualized and/or containerized infrastructures (e.g., Docker) available across Cloud, Edge, and Things layers;
2. **MicroData (MD),** encodes the contextual information about (i) the sensors, actuators, edge devices, and cloud resources it needs to collect data from or send data to, (ii) the specific type of data (e.g., temperature, vibration, pollution, pH, humidity) it needs to process, and (iii) other data manipulation operations such as where to store data, where to forward data, and where to store results;
3. **MicroComputing (MC),** executing specific types of computational tasks (machine learning, aggregation, statistical analysis, error checking, and format translation) based on a mix of historic and real-time MD data in heterogeneous formats. These MCs could be realized using a variety of data storage and analytics programming models (SQL, NoSQL, stream processing, batch processing, etc.); and
4. **MicroActuator (MA),** implementing programming (e.g., for sending commands) interfaces with actuator devices for changing or controlling object states in the IoT environment.

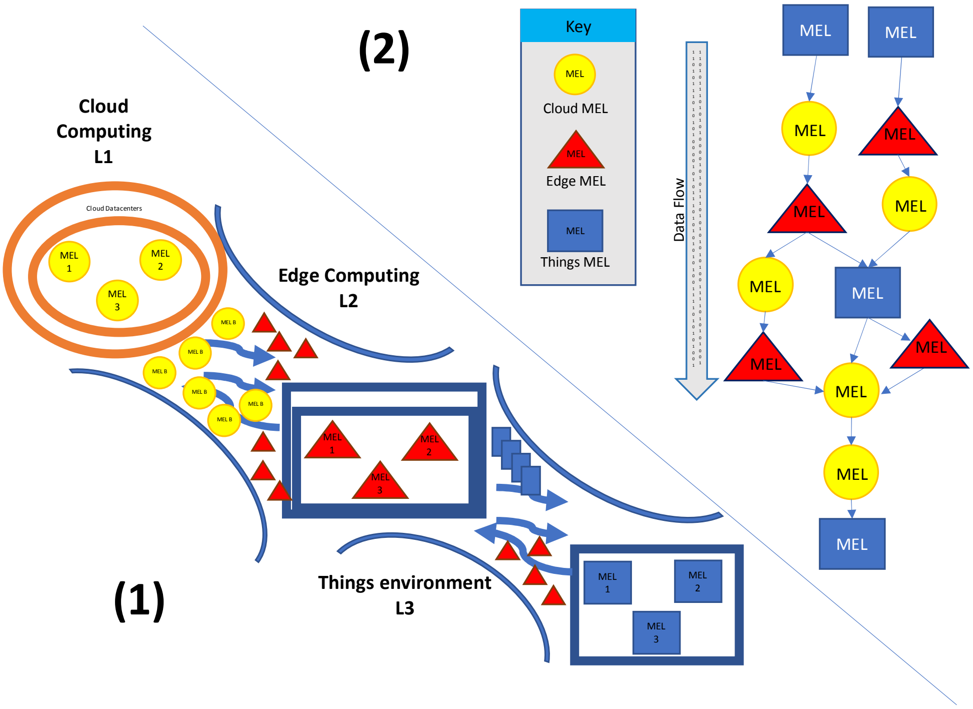

Figure 4. Osmotic "movement" of (1) MELs across Cloud, Edge, and Things; (2) MELs graph representation.

In summary, developments such as MEL,[2] provide a good starting point for describing the basic IoT computation unit. However, additional enhancements to the MEL abstraction are required so that it can be used to describe and program different types of IoT applications (e.g., Smart Homes, Smart Grids). Current IoT applications development is typically a vertical, proprietary application stack that is often difficult to generalise. Where there is heterogeneity of IoT sensors and platforms (Smart Cities) they are typified by large amount of bespoke code and data integration requirements. Adoption of standards and protocols across IoT deployments is piecemeal at best and chaotic at worst. A step-by-step development approach reducing the cost of deployment and configuration, putting flexibility of system design at the core, could be used to increase uptake of software abstractions such as MEL.

IoT Application Graph Choreography

We need fundamentally new IoT applications programming pattern (e.g., MEL graph as shown in Figure 4) for: (1) decomposing IoT data analysis activities into fine-grained activities (e.g., statistics, clustering, classification, anomaly detection, accumulation, filtering), each of which may impose different planning and run-time orchestration requirements; (2) identifying and integrating real-time data from IoT devices and historical IoT data distributed across Cloud and Edge resources; (3) identifying data and control flow dependencies between data analysis activities focusing on coordination and data flow variables, as well as the handling of dynamic system updates and re-configuration; and (4) defining and tagging each data analysis activity with run-time deployment constraints (QoS, security and privacy).

Existing composition and choreography standards (such as SOAP, TOSCA, and BPEL) are not suitable, as they *cannot sufficiently* capture the complexity of an IoT application graph (e.g., heterogeneous data sources, data and control flow dependencies across heterogeneous activities, and heterogeneous software and hardware configurations across Things, Edge, and Cloud layers). Additional research is required for implementing and deploying an IoT application graph (see Figure 4). Approaches such as Juju and Fabric8 provide promising developments in this area.

IoT Application Graph Performance Calibration

IoT applications developers need to systematically undertake performance characterization of data analysis activities (e.g., MELs) across different parts of the infrastructure (Cloud, Edge, and Things). They need to understand and reason about most important QoS metrics and/or security and privacy threats to each data analysis activity. For instance, QoS metrics required to characterize the performance of a data analysis activity mapped to a Cloud layer (see Figures 1 and 3) is quite different from a gateway and/or device in the Edge and Things layer Similarly, performance analysis[8] at the Things and Edge layers may require assessing network stability, throughput optimality, routing delays, fairness in resource sharing, available bandwidth, and sensor battery state. These QoS metrics might be very different from the ones relevant to a Cloud operator, who is interested in end-to-end response times, platform scalability and reliability, virtual server utilizations, and the costs of moving data to and from the Cloud.

Currently, many benchmarking kernels (e.g., TPCx-IoT, BigDataBench, TeraGen, TeraSort, TeraValidate, Google ROADDEF, Linear Road, DeepBench, and MLPack) exist for characterizing performance of IoT data analysis activities. Moreover, each benchmarking kernel type has its own benefits and can help us understand performance of specific type of IoT data processing activity under variable workload scenarios. For example, TPCx-IoT benchmark can be used to calibrate the performance at the Edge layer as it is representative of the data analysis activities (data aggregation, real-time analytics and persistent storage) that are typically hosted in IoT gateway systems. Similarly, Google ROADEF & Linear Road benchmarking kernels can be applied for calibrating performance of stream processing data analysis activity at the Edge layer. On the other hand, TeraGen, TeraSort, and TeraValidate benchmarking kernels can be applied to calibrate the performance of batch processing activity at the Cloud layer. As it can be inferred, none, by themselves, can reveal the true bottleneck of whole IoT application graph, which includes multiple data analysis activities unless multiple benchmarking kernels are properly combined together. Hence, one of the possible research directions will be to identify/build different suitable benchmarks from each type of the data analysis activities and hierarchically/logically combine them to draw accurate conclusions across an IoT graph in a holistic way.

IOT APPLICATION DATA MANAGEMENT

The ability to efficiently capture and manage multiple, diverse types of real-time and historical data streams lie at the heart of developing improved decision models and impact analytics. This includes traditional, structured data such as that acquired by environmental sensor networks, for instance. It also includes more challenging unstructured data streams including geospatial social media feeds (twitter, Instagram, news feeds, etc.), as well as data from the continuous monitoring of ambient environment, people, and machines. In addition to being harder to interpret and use, such data feeds have variable velocities and less structure and thus will require more opportunistic data management approaches.

Storage

It is now widely recognized that in the era of high velocity, volume and variety data no single data storage approach is optimal for IoT data management purposes. Thus, there is increasingly a move towards developing heterogeneous storage platforms where different data storage and analytics programming model (e.g., stream processing, batch processing, SQL, NoSQL) can be used depending on the subsequent analytics requirements. Hence, future research will need to address the challenges of selecting and tuning a suite of data storage and analytics programing models and related tools. To achieve this, it is necessary to take into account the specific characteristics of each IoT data stream, as well as the specific access requirements of the underlying IoT application. There is also growing interest in supporting storage at the network edge, eliminating the need to capture all data into a central repository, such as work proposed by Edge Analytics (www.edgeintelligence.com).

Access

The real-time and/or semi real-time data analysis requirements of IoT applications require seamless access to IoT data feeds. The research community will need to investigate techniques for efficient IoT data retrieval that will include indexing to meet the demands of real-time analytics and support for *sharding* of the data stream based on application and infrastructure requirements. Apache Kafka already provides some of this capability, i.e., the ability to shard a data stream based on the memory capacity of the computational nodes undertaking the analysis. However, understanding mechanisms to support such sharding at the network edge remains a challenge. One possible research direction will be to implement a federated approach; where data query and retrieval functionality from multiple individual platforms are mediated by an IoT programming abstraction (e.g., a MEL). As discussed earlier, MEL will need to expose a uniform programmatic interface (APIs) to models and analytics, hence, reducing the barriers (and associated latencies) to data ingestion into models and visualisations. The new federated API suite may utilize the Apache Spark SQL API to benefit from its existing interoperability features, in addition to other Apache libraries such as Samza and Kafka. The chosen API must be lightweight, and enable integration with services made available in libraries supported by other vendors.

Geo-Distributed Cross-Querying

As noted above, an IoT application may be described as a graph and stored across different parts of the infrastructure which are likely to be geo-distributed. The data sets that need to be processed through such applications are also distributed, requiring support for distributed search and for performing multiple analytical queries in a geo-distributed manner. Moreover, these analytical queries will differ depending on the type of storage and analytics programming model implemented by a given IoT data analysis activity. The analytical queries can be a SQL query, stream processing query, NoSQL query, or a MapReduce query. Hence the challenge is to design new types of *multi-query planning and provisioning algorithms* that can optimally distribute queries across data analysis activities mapped to different parts of an IoT infrastructure, while optimizing end-to-end QoS associated with the query graph (query plan), to improve resource utility and meet users' SLAs.

Existing geo-distributed querying systems[9–11] do not consider the heterogeneous computing infrastructure, neither do they execute the queries over different types data analysis activities programmed using heterogeneous models (e.g., stream processing, NoSQL, SQL, batch processing). While IoT applications need to process and query both static and real-time data, existing geo-distributed querying systems were designed for managing static data. Event Processing Language (EPL) has been used in majority of stream process platform such as Apache Spark, Kafka, Flink and Esper. Similar to SQL, EPL supports the following data querying operations: SELECT, FROM, WHERE, GROUP BY HAVING and ORDER BY. Unlike relational database systems, these EPL-based platforms can limit the query data size to guarantee real-time processing. However, one of the core limitations of EPL- and SQL-based querying approaches is that they cannot deal with heterogeneous data stored across multiple types of storage platforms and/or programmed using multiple types of storage and analytics programming models.

Integration

Future research efforts need to develop innovative techniques for supporting the integration of heterogeneous IoT data from thousands or millions of sources. However, establishing relationships between IoT data sources (e.g., CCTV), associated events (e.g., air pollution, traffic incidents, flooding, landslide), and stakeholders (e.g. decision makers, first responders) are generally difficult to detect as it is dependent on the context of the IoT application. For example, data integration techniques for air quality monitoring need to establish a relationship between road traffic patterns and the associated air and noise pollution. At the same time it should allow seamless integration of different types of air quality data including the pollution data extracted from the raw chemical sensors data, water quality sensors, traffic flow sensors (e.g., CCTV), and air quality sensors.

Moreover, such data integration efforts can build on existing standards such as the Semantic Sensor Networks (SSN) to allow consistent representation of IoT sensors and their data streams. Another very interesting research direction will be to apply graph-based approaches and machine learning techniques for discovering relationships between IoT data sources, associated events, and stakeholders.

Data Provenance

An IoT data provenance technique is responsible for logging the origins (IoT data sources) and historical derivations of data by means of recording data analysis transformation operations (those data analysis activities that are in charge of manipulating data). IoT-based sensing equipment is deployment to improve decision making. Automated decision making at the source (e.g., traffic control signal) requires metadata to be able to validate decisions post-event and check the reliability/efficiency of the decision-making processes. Once one steps away from real-time automated decision making, the process chain of data manipulation becomes more complex, and likewise, the associated decision making process moves further from the data source. Thus, the provenance and metadata of IoT systems are critical to the implementation, trust, and social use necessary for the deployment and use of IoT based sensing.

Dealing with provenance in the context of large IoT application graphs is challenging often due to the size and volume of data involved, the heterogeneous configurations of the underlying infrastructure (Cloud vs. Edge vs Things), and the heterogeneous data analysis activities (e.g., type of storage and analytics programming model). Developing contextual metadata is essential for reasoning about heterogeneous IoT data and related lifecycle activities (e.g., produce, store, process, and query). Traditional data provenance techniques require collection and transmission of large data volumes, which is impractical for IoT applications that warrant sub-second decision making and data processing latency. Hence, new techniques are required which can reduce and enhance the efficiency of provenance and metadata collection, recording, and transmission. Understanding how provenance relationships can be derived from IoT data processing activities therefore remains a challenge, as precedence relationships identifying which output was a consequence of a particular set of inputs may be difficult to establish.

One possible research direction to develop IoT data provenance technique based on Blockchain's Distributed Ledger Technology (DLT) to record lifecycle activities on data as it travels through the IoT ecosystem. Another hard challenge to solve will be to develop provenance techniques than can verify complying with data privacy regulation such as GDPR. Here GDPR-based IoT data access policies need to be verified, to ensure that the user has provided consent on how their data can be analysed and fused with other data sources. Undertaking GDPR compliance for statically held data (e.g. user information) can be easier to manage, however extending this to a dynamic data stream (which may be context dependent) remains a challenge. Another research topic is the use of smart contracts in IoT deployments that involve more than one vendor and where data exchange needs to take place. This will be needed in complex use cases such as smart cities.

IOT APPLICATION ORCHESTRATION

When an IoT application is expressed as a collection of multiple self-contained data analysis activities (e.g., MEL), future research will need to consider the following: (i) choosing storage and analytics programming models (e.g., stream processing, batch processing, NoSQL) and computational (e.g., data analysis algorithms) models that can seamlessly execute in highly distributed and heterogeneous IoT infrastructure (see Figure 3 and Figure 4); (ii) dynamically detecting faults across multiple parts of the IoT infrastructure; (iii) dynamically managing *resources, data, and software* available in Things, Edge and Cloud layers driven by IoT-specific applications requirements (data volume, data velocity, QoS, security, and privacy).

Optimal Configuration selection

Mapping of IoT application (graph of data analysis activities) demands selecting bespoke configurations[1] of resources at Things, Edge, and/or Cloud layers from abundance of possibilities. For example, in context of: (i) Cloud: we need to consider configurations such as datacentre location, pricing policies, compute/storage configurations, virtualization features, upstream/downstream network latency, etc. (ii) Edge: we need to consider configurations such as Edge device (Raspberry Pi 3, UDOO board, ESP8266) hardware features (e.g., CPU power, main memory size, storage size), upstream/downstream network latency, supported virtualization features, etc.; and (iii) Things: we need to consider data source location, battery, upstream/downstream network latency, network type, life, sensor type etc.. The diverse configuration space coupled with conflicting (trade-off) QoS, security, and privacy requirements leads to exponential growth of potential search space. Hence, computing a near-optimal solution for mapping IoT application graph to Things, Edge and/or Cloud layers in a reasonable time is NP-hard in much stronger sense when compared against task mapping and scheduling problems in Cloud computing, Services computing, and Grid computing systems.

Given the complexity of multilayered configuration search space and mix of conflicting requirements, future research efforts need to focus on developing computationally tractable optimisation techniques that can accommodate cross-layer resource configurations and conflicting QoS, security and privacy requirements. Moreover, these techniques will need to cater for diverse requirements of heterogeneous IoT applications.

Holistic Monitoring

To automatically predict and detect anomalies and their root causes, it is critical to monitor[12] and profile the following contextual information in real-time: QoS parameters (whole IoT application graph, activity-specific, edge-specific, Things-specific and cloud-specific) and activity-specific data flow. Much of the difficulty in monitoring IoT application graphs is due to the massive scale and heterogeneity of underlying computing infrastructure and multi-modal data sources.

Although QoS monitoring topic has attracted a lot of attention from the distributed computing (Grid, Cloud, Web Service) community, none of the existing monitoring tools and techniques are able to monitor performance of IoT application graphs in a holistic way. Data streams themselves need monitoring and here context and location can be important. Understanding a fault or change in a sensor may require profiling of the normal operation of that sensor which cannot be assumed to operate in a generic fashion. Hence, novel monitoring techniques providing detailed data flow and QoS information related to IoT application graph are required. These techniques will need to give deep insights into how data analysis tasks and underlying resources are performing, where possible QoS bottlenecks should be monitored along with all security or privacy threats. At the same time, these techniques should be able to give holistic view of QoS and data flow in an end-to-end fashion.

Fault-detection and Debugging

In future IoT environments where multiple decentralized and distributed devices and resources from the Things (e.g., sensors), Edge (e.g., compute and storage) and Cloud layers function together, the probability of failures will be high—simply due to the plethora of connected things. Moreover, the complexity of multiple data analysis activities simultaneously happening across these devices further adds to the chances of failure particularly in the cases when they are interdependent of each other. Such failures can be in several forms for example hardware, software or wrong user inputs or interaction with the ecosystem including connectivity, mobility and battery power of different devices. For example, some devices may be connected via wireless connections that may also vary in speed and reliability, this will impact the data transfer rate required by the applications. Most of the edge devices may have wireless connectivity and many cases they may be mobile devices, which are battery powered. In other words, their capacity and capability may vary quite frequently and some may fail if overloaded. In summary, the complexity of failure management within such heterogeneous and changing environments is not trivial. This gets complicated when various data analysis activities have some SLA defining bounds on QoS parameters with IoT providers. Therefore, the challenge for IoT providers is how to ensure SLA /QoS in such failure-prone environments. Or question may be asked how to define QoS parameters in SLAs for such environments.

Run-time Reconfiguration

One of the most important aims of IoT applications orchestration processes is to design run-time reconfiguration algorithms to dynamically allocate and reallocate data analysis activities (within an application graph) to different parts of the infrastructure depending upon many unpredictable events including sudden unavailability of sensing devices (e.g., due to battery drain, a power failure, or IoT devices being made unavailable by their owners) and degradation of either an edge node or the communication network (e.g., due to overloading, edge failure or changes in the IoT data flow rate).To determine how each data analysis activity consistently achieves its QoS objectives while dynamically handling the run-time uncertainties of data flow behaviour and "Cloud-Edge-Things" performance is a unsolved research challenge. Moreover, data analysis activities are interdependent; changes in the execution and data flow of one activity will influence others. At run time, the reconfiguration technique must therefore be aware of these interdependencies between activities – passing aggregate data from the edge to the cloud, for example. In other words, all of the above uncertainties in IoT applications demand bespoke run-time reconfigurations.

To achieve multilayer ("Cloud-Edge-Things") reconfiguration, IoT research community needs to investigate a comprehensive set of QoS prediction models for heterogeneous data analysis activities mapped across Cloud, Edge, and Things layers. The QoS prediction models should be dynamically tuneable based on real-time monitoring information available from holistic monitoring approaches. These QoS prediction models will also need to undertake trade-off analysis between limited compute capability and low latencies at the Edge (i.e., close to the IoT devices), versus large compute capability and high latency (i.e., at the Cloud). In order to achieve this, new research efforts are required focussed on designing network, compute and storage aware optimisation algorithms to identify the best topology for IoT graph dataflow that may arise from the same underlying physical configuration of the edge and IoT networks.

CROSS CUTTING CONCERNS

Security, Privacy and Compliance

With increasing up take of IoT application services (smart city, smart traffic, smart home, smart healthcare), often hosted over (distributed cloud, edge, and IoT) infrastructure, there is a realization that IoT services can involve an interlinked set of providers (data, service, network and infrastructure). Stakeholders in IoT environments implicitly expect and demand their data and services to be secure, trusted as well as to preserve their privacy. Users of IoT applications may only interact with other applications via simple web interface without actually being aware of the large, distributed service, data, and network ecosystem. They often entrust their data and identity without realising that IoT applications providers may share their data with several back-end services (Cloud hosted analytics, mobile edge network provider, government stakeholders).

Security in IoT applications involves satisfying mainly two key properties: Authentication and Integrity. Achieving successful authentication in IoT ecosystem requires a device identity management and a suitable authentication scheme. The ubiquitous (and heterogeneous) nature of the variety of IoT devices from different vendors and their presence in an untrusted environment with no central authority makes the traditional enterprise-based identity management system incapable of working for IoT applications. The challenge for a further research is how to manage the identity of IoT devices in fully decentralised and distributed systems. A comprehensive identity management framework that works seamlessly with existing enterprise-based identity management systems is needed.

Traditional authentication schemes are no longer applicable to IoT environments due to limited resources (e.g., memory, battery life, etc.). Different lightweight authentication protocols have been explored in the literature to address this challenge. A practical lightweight authentication scheme still remains as a problem to be solved. In addition, many schemes under development have largely ignored the possible realisation of Quantum Computing. Developing lightweight post-quantum authentication schemes is the next grand challenge in IoT authentication. It is important to note that such authentication schemes should not only support devices to gateways/edges authentication, but also mutual authentication between devices and all other components in the system (e.g., other devices, servers, users, etc.).

Integrity is the most important security property when you consider the integration of IoT with the emerging data science paradigm. First, a device needs to be trusted so that the data generated by the device can be reliably used in making the (right) decision. Because of the diversity of devices and manufactures, many different firmwares could be present at any one point in time; all could be vulnerable to attacks by adversaries. Performing static and dynamic analysis might help to identify potential vulnerabilities, but it is almost impossible to do so for every firmware in the market. Furthermore, even as vulnerabilities are detected and a patch is developed to fix them, the distribution of such patches to all IoT devices is very difficult since the devices may or may not support automatic discovery. Further research is needed in this area. Second, we need to ensure that data integrity is maintained, not only when the data is at rest and in motion, but also while performing the data analytics. Providing integrity to data analytics is a challenging area of research, and a good amount of attention has been paid to it in recent time. Adversarial machine learning (e.g., GAN-based approaches) is an active research area and it is important to understand the impact of it on the integration of IoT and data science.

Privacy has been widely studied under different research disciplines: Privacy Preserving Technologies (PPT), Privacy Enhancing Technologies (PET) and Privacy Engineering (PE). While this has been a well-defined problem in the community, it is greatly exacerbated by the expansion of Internet-connected devices. A large number of techniques have been developed; most notably, ones in recent time include differential privacy and privacy-by-design. Some of these privacy technologies have been extended and used in IoT applications. This area needs further research in terms of developing privacy guidelines for IoT devices and data. General Data Protection Regulation (GDPR) introduced by the European Union (EU), which ensures that non-expert users can make informed decisions about their privacy and thereby give 'informed consent' to the use, sharing and repurposing of their personal data, is a right direction towards addressing some of the problems. Other world geographies are likely contemplating regulations similar to GDPR. Privacy preserving data trading platform is needed to ensure that IoT data can be made available for data science without worrying too much about privacy breaches.

Like all other computer systems, the weakest link in security and privacy is always the end users. Hence, the development and deployment of technological solutions should put users at the core. Human-centric security and privacy for IoT is the next big research challenge. Comprehensive security and privacy guidelines need to be developed for users, whether they are employees or citizens. Different levels of governments have a role to play in this space. Guidelines and regulations can only work if there is a way to check the compliance against such regulations and guidelines. A number of governments around the world have started to look at this seriously. The research challenge is how to automate the compliance checking process.

Ultimately, the IoT research community will need to investigate new security-, privacy- and compliance-aware applications graph provisioning mechanism to enable: (i) greater trust among users, stakeholders and IoT applications' providers; (ii) emergence of new actors that can offer services; and (iii) an IoT data marketplace that enables greater control of personal data by users.

As IoT is directly involved in the physical world we are living in and the data we generate, security, privacy and compliance will always remain sensitive topics and must be managed carefully. That said we feel the aforementioned points are just snippets of what needs to be covered. So, we urge the research community and industry at large to considerably elaborate on these topics going forward.

Scalable and Unified messaging

An IoT application may use a hybrid approach for message communication depending on context (i.e., centralized and decentralized). The message communication includes sensor to sensor (S2S), sensor to edge (S2E), edge to cloud (E2C) and sensor to cloud (S2C) interaction. The use of centralized modes such as S2C cannot meet the requirements of soft and hard real-time applications. For example, a neighbouring smart vehicle system in New York city uses local WiFi to support the real-time interaction of vehicle to vehicle (V2V) and vehicle to traffic infrastructure (V2I).[13] Existing cloud-based communication solutions, such as AWS IoT and Azure IoT Hub, are unable to meet the strict QoS, security, and compliance requirements of diverse IoT applications. As we note in a previous Blue Skies instalment,[14] future efforts needs to focus on developing distributed IoT messaging middleware which can leverage the ever- increasing amount of resources at the edge of the network to provide reliable, ultra-low-latency, and privacy-aware message routing and communication. Having said that, the protocol heterogeneity inherent to Edge (e.g. WiFi, 4G/5G, wired) and Things (e.g. Bluetooth Low energy, COAP, MQTT) resources, and the unpredictability and resource constrained nature of Edge and Things resources, make it extremely challenging to provide resilient coordination mechanisms and guaranteed message delivery. In order to fit these heterogeneous protocols on the existing architecture such as OSI Model, we need to *unify and abstract* the protocols present across different layers into a *new IoT communication API stack*. The new IoT communication API will need to include communication adapters for multiple, heterogeneous communication protocols relevant to Things, Edge and Cloud layers.

Programmable networks for supporting IoT

The ability to independently manage the control and data plane, as proposed by software-defined networking (SDN) is an approach that allows network administrators to program and initialise, control, change and manage networking components of the OSI model. SDN is designed to address programmability shortcoming of static architecture of traditional networks such as those are used in current datacenters.[15] SDN has already shown great performance improvements in other fields such as flow optimisation or bandwidth allocation in cloud-based datacentres, and as yet has never been realistically utilised for IoT application infrastructure (see Figures 1–4). This is due to the fact that current SDN platforms are built on two assumptions: (1) having a centralized controller and (2) the requirement to contact devices to pull usage statistics or to push commands to devices. None of the above assumptions are applicable to IoT applications infrastructure because (1) connecting millions of IoT devices to a centralised controller is not scalable; and (2) IoT sensors/actuators may have intermittent connections. Thus, one of the important research direction will be to further study and consequently modify current SDN controllers (such as OpenDayLight) to first subdivide the controlling layer and secondly to tolerate lossy connections.

CONCLUSION

There is significant potential for IoT applications to improve our well-being and be drivers for social good. To go beyond the hype and the use of bespoke solutions, we need to address the many architectural challenges that will enable demonstrable and ongoing value. While many challenges exist firmly in the ICT sphere, these also accompany external factors and immature technologies elsewhere, so development must go hand-in-hand with e.g., more reliable and accurate sensing systems (and acknowledgement of the uncertainty in many systems we measure), better and more reliable communication stacks and improved power management and battery life. IoT applications also do not exist in a technology vacuum. They are scaffolded by existing regulatory systems, processes, social, economic and legal systems, which means holistic change may be required to achieve the IoT vision we have been promising the world. IoT infrastructure needs to be developed within a symbiotic feedback loop with these existing systems to engender public trust and the social and political license to benefit fully from technological advances.

It now also appears that Artificial Intelligence (AI) based technologies are the new operating system, as many hardware & software vendors attempt to integrate these in their IoT systems and software libraries. Understanding how these AI algorithms process and interpret such data remains a challenge, as opening up the "box" to understand the actual operations of these algorithms remains unclear, leading to issues around trust in how such algorithms make their decisions (but this is a topic for another column).

References

1. M. Villari et al., "Osmotic Computing: A New Paradigm for Edge/Cloud Integration," *IEEE Cloud Computing*, vol. 3, no. 6, 2016, pp. 76–83.
2. M. Villari et al., "Software Defined Membrane: Policy-Driven Edge and Internet of Things Security," *IEEE Cloud Computing*, vol. 4, no. 4, 2017, pp. 92–99.
3. L. Wang and R. Ranjan, "Processing Distributed Internet of Things Data in Clouds," *IEEE Cloud Computing*, vol. 2, no. 1, 2015, pp. 76–80.
4. H.L. Truong and S. Dustdar, "Principles for Engineering IoT Cloud Systems," *IEEE Cloud Computing*, vol. 2, no. 2, 2015, pp. 68–76.
5. A. Taivalsaari and T. Mikkonen, "A Roadmap to the Programmable World: Software Challenges in the IoT Era," *IEEE Software*, vol. 34, no. 1, 2017, pp. 72–80.
6. A. Dastjerdi and R. Buyya, "Fog Computing: Helping the Internet of Things Realize its Potential,," *Computer*, vol. 49, no. 8, 2016, pp. 40–4.
7. T. Baltrusaitis, C. Ahuja, and L.P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. Early access, 2018; doi.org/10.1109/TPAMI.2018.2798607.
8. G. Kecskemeti et al., "Modelling and Simulation Challenges in Internet of Things," *IEEE Cloud Computing*, vol. 4, no. 1, 2017, pp. 62–69.
9. R. Viswanathan, G. Ananthanarayanan, and A. Akella, "Clarinet: WAN-Aware Optimization for Analytics Queries," *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation* (OSDI 16), 2016.
10. Q. Pu et al., "Low Latency Geo-distributed Data Analytics," *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication* (SIGCOMM 15), 2015, pp. 421–434.
11. K. Kloudas et al., "Pixida: optimizing data parallel jobs in wide-area data analytics," *Proceedings of the VLDB Endowment*, vol. 9, no. 2, 2015, pp. 72–83.
12. A. Souza et al., "Osmotic Monitoring of Microservices between the Edge and Cloud," *20th IEEE International Conference on High Performance Computing and Communications* (HPCC 18), 2018.
13. k.m.d. Dikaiakos et al., "Location-aware services over vehicular ad-hoc networks using car-to-car communication," *EEE Journal on Selected Areas in Communications*, vol. 25, no. 8, 2007, pp. 1590–1602.
14. T. Rausch, S. Dustdar, and R. Ranjan, "Osmotic Message-Oriented Middleware for the Internet of Things," *IEEE Cloud Computing*, vol. 5, no. 2, 2018, pp. 17–25.
15. D. Cho et al., "Real-Time Virtual Network Function (VNF) Migration toward Low Network Latency in Cloud Environments," *Proceedings of the IEEE 10th International Conference on Cloud Computing* (CLOUD 17), 2017, pp. 798–801.

ABOUT THE AUTHORS

Rajiv Ranjan is a Chair Professor of Computing Science and IoT at Newcastle University, UK. He received a PhD in Computer Science and Software Engineering from the University of Melbourne. His research interests include Internet of Things, Big Data Analytics. Contact him at raj.ranjan@ncl.ac.uk.

Omer Rana is a full professor of performance engineering in the School of Computer Science and Informatics at Cardiff University. He received a PhD from the Imperial College London. His research interests include performance modelling, simulation, IoT, and edge analytics. Contact him at ranaof@cardiff.ac.uk.

Surya Nepal is a principal research scientist at CSIRO Data 61, Australia. His research interests include cloud computing, Big Data, and cybersecurity. Nepal has a PhD in computer science from Royal Melbourne Institute of Technology, Australia. Contact him at surya.nepal@csiro.au.

Mazin Yousif is the editor in chief of *IEEE Cloud Computing*. He's the chief technology officer and vice president of architecture for the Royal Dutch Shell Global account at T-Systems International. He has a PhD in computer engineering from Pennsylvania State University. Contact him at mazin@computer.org.

Philip James is a senior lecturer in the School of Civil Engineering and Geosciences at Newcastle University, UK. His research interests include the Internet of Things, next-generation analytics, and spatial data management. Contact him at philip.james@ncl.ac.uk.

Zhenyu Wen is research fellow at Newcastle University, UK. He received a PhD (2016) in Cloud Computing from Newcastle University. His research interests includes IoT, Distributed systems, Big Data Analytics and Computer network. Contact him at z.wen@ncl.ac.uk.

Stuart Barr is a professor in the School of Civil Engineering and Geosciences at Newcastle University, UK. He works on a range of cross-disciplinary problems in the field of Earth systems engineering. Contact him at stuart.barr@ncl.ac.uk.

Paul Watson is a professor of computer science at Newcastle University. His research interests include scalable computing systems, data-intensive problems, and cloud computing. Watson has a PhD in computer science from Manchester University. Contact him at paul.watson@newcastle.ac.uk.

Prem Prakash Jayaraman is a research fellow at the Swinburne University of Technology. His research interests include Internet of Things, cloud computing, big data analytics. Jayaraman has a PhD in computer science from Monash University. Contact him at prem.jayaraman@gmail.com.

Dimitrios Georgakopoulos is a professor at Swinburne University of Technology. His research interests include the Internet of Things, data management. Georgakopoulos has a PhD in computer science from the University of Houston, Texas. Contact him at dgeorgakopoulos@swin.edu.au.

Massimo Villari an associate professor of computer science at the University of Messina. His research interests include cloud computing, Internet of Things, big data analytics, and security systems. Villari has a PhD in computer engineering from the University of Messina. Contact him at mvillari@unime.it.

Maria Fazio is an assistant researcher in computer science at the University of Messina (Italy). Her main research interests include distributed systems and wireless communications. She has a PhD in advanced technologies for information engineering from the University of Messina. Contact her at mfazio@unime.it.

Saurabh Garg is currently working as a lecturer in the Department of Computing and Information Systems at the University of Tasmania, Hobart, Tasmania. He received a Ph.D. from the University of Melbourne in 2010. Contact him at Saurabh.Garg@utas.edu.au.

Rajkumar Buyya is a professor of computer science and software engineering, at the University of Melbourne, Australia. His research interests include cloud, grid, distributed, and parallel computing. Buyya has a PhD in computer science from Monash University. Contact him at rbuyya@unimelb.edu.au.

Lizhe Wang is a professor at the Institute for Remote Sensing and Digital Earth, Chinese Academy of Sciences, and at the School of Computer at the China University of Geosciences. Wang has a PhD in computer science from Karlsruhe University. Contact him at lizhewang@icloud.com.

Albert Y. Zomaya is the Chair Professor of High Performance Computing & Networking at the University of Sydney. His research interests include parallel and distributed computing, and data intensive computing. Zomaya has a PhD from Sheffield University. Contact him at albert.zomaya @sydney.edu.au.

Schahram Dustdar is a full professor of computer science heading the Distributed Systems Group at TU Wien, Austria. His work focuses on Internet technologies. He's an IEEE Fellow and a member of the Academy European. Contact him at dustdar@dsg.tuwien.ac.at.