



Urban Risk Analytics in the Cloud

Rajiv Ranjan, Jedsada Phengsuwan, Philip James, Stuart Barr, and Aad van Moorsel, *Newcastle University, UK*

Rapid population growth in cities demands effective plans to protect people from vulnerabilities, for example, natural disasters. *Urban risk analytics* can play a significant role in enabling dynamic and timely decision-making for risk management in cities. Urban risk analytics is the process of analyzing huge amounts of urban data to understand and model city vulnerability in a holistic way. Due to the complexity of risk management for cities, this process requires sophisticated techniques such as data integration, pattern detection, and data mining to manage and process big city data from different sources using both real-time and batch-processing models.

Here, we propose a cloud-based general framework for facilitating effective urban risk analytics over big city data. We also discuss research challenges in developing cloud-based data integration and analytics algorithms for urban risk management.

Urban Risk Analytics Scenario

Urban data collection systems derived from pervasive sensors are being deployed in many cities across

the UK and beyond, including Newcastle. These systems—while currently delivering moderate data volumes (in Newcastle, for instance, we receive about 5 million observations per day)—will soon ramp up to produce significant data volumes for real-time computation and data-mining-type computation of historical data. Additional streams of potentially valuable data are already being created via social media (Twitter, Instagram, and so on) and mobile apps. A city is overflown many times a day by various orbital platforms that provide near real-time Earth observation (EO) data covering various metrics, from urban temperature to atmospheric conditions. Infrastructure systems underpin the movement of people, power, water, and waste around the city. Many of these systems also provide real-time information on the state or flows within them and are underpinned by static map-type data of features and their locations.

A public health official interested in the air quality risk to asthma patients might wish to

- find hotspots and examine historical real-time observations;
 - relate hotspots to particular flow indices in the traffic network;
 - analyze social media for relevant keywords or phrases to ascertain visibility issues; and
 - develop real-time warning systems for poor air quality.
- Individually, each of these tasks requires a huge amount of data collection, data preparation, and subsequent analysis.
- Cities exhibit multiple levels of complexity across a large number of interacting domains (transport, air quality, climate, traffic control, surface water management, and so on) that operate on many temporal and spatial scales. This diversity of data and associated temporal and spatial variability has direct impacts on our ability to reliably and objectively monitor and characterize the environmental condition of cities. For example, although it might be possible to monitor several streets in detail with regard to their temperature and air quality, it is unrealistic to undertake this for every street in a city. However, images acquired by EO satellites and airborne remote sensing devices could let us
- analyze an historical archive of EO data to look at changing patterns across the city;

In This Issue: IT for a Smarter World

As several emerging new applications demonstrate, IT plays a major role in creating a connected, smarter world and in improving operations in industry and business. Deployment of these applications is further fueled by advancements in cloud computing and software development and testing.

Highlighting these trends, in this issue, we feature six articles that focus on smart dairy farming, citizen well-being, energy harvesting for body area networks that find applications in healthcare and assisted living, smart building management, cloud computing, and software development beliefs that help identify process improvement actions for creating quality software and applications.

In the first article, “Opportunistic Wireless Networking for Smart Dairy Farming,” Chamil Kulatunga, Laurence Shaloo, William Donnelly, Eric Robson, and Stepan Ivanov highlight increasing the efficiency of milk production as a key to meeting the increasing worldwide demand for organic dairy products and succeeding in this niche market. By adopting smart dairy farming principles that harness the promise of the Internet of Things (IoT), cloud computing, and big data analytics, dairy farmers can increase milk production efficiency. A key challenge, however, is disseminating large volumes of data from the farms to the cloud for data analytics. The authors address this challenge by proposing an opportunistic networking paradigm and a delay-tolerant networking (DTN) approach to transmit dairy data to a fog computing node at an Internet gateway. This approach allows use of ultra-low-power short-range wireless technologies, such as ANT+, over a wider geographical terrain without an infrastructure network.

To help deliver better health and well-being services to citizens, smart city systems are now gaining greater interest. Besides using various types of sensors to collect required data, these systems can harness the concept of people as sensors, also known as human sensing, to gain insight or collect subjective information from humans. In “A Web-Based Portal for Assessing Citizen Well-Being,” Darren O’Neill and Cathryn Peoples present an interesting Web-based portal that lets citizens provide personal details and answer a set of questions that ascertain their mood and well-being. The authors harnessed a range of open source technologies to offer a cross-platform, interoperable, and rapidly deployable portal.

Wireless body area networks (WBANs) find applications in health monitoring, assisted living, telemedicine, and other areas. A WBAN can transmit data from distributed, low-power, wearable, and implantable sensor nodes to remote locations for further processing and decision making. Though powering them is a major challenge, the generation of electrical energy from ambient sources, known as energy harvesting, is a viable approach to address this challenge. In “Energy Harvesting for Self-Sustainable Wireless Body Area Networks,” Fayaz Akhtar and Mubashir Husain Rehmani examine energy harvesting techniques for WBANs. They review potential harvestable sources and their characteristics and usability in minimizing energy constraints, and discuss current challenges in exploiting these sources and possible future research directions.

The next three articles focus on IT systems—cloud computing and system testing beliefs. Providing dedicated cloud services that ensure users’ dynamic quality-of-service (QoS) requirements and avoid service-level agreement (SLA) violations is a key challenge facing cloud service providers. This calls for better resource allocation in the cloud, accounting for heterogeneity, dynamism, and failures. In the article “The Journey of QoS-Aware Autonomic Cloud Computing,” Sukhpal Singh, Inderveer Chana, and Maninder Singh present a broad literature analysis of resource management techniques in cloud computing, including autonomic resource provisioning and scheduling. They also discuss future directions of cloud resource management.

Next, in “Cloud Adoption in Brazil,” Jorge Pereira and coauthors provide an overview of the cloud adoption scenario in Brazil. They also outline the initiatives undertaken by governmental organizations, private companies, and universities for enhancing cloud adoption. Furthermore, they discuss the impact and challenges of cloud adoption in Brazil by identifying trends and the most widely used models and technologies.

Beliefs held by software developers and testers are a key factor in determining quality software that runs a huge number of applications. In the final article, “Examining Software Engineering Beliefs about System Testing Defects,” Akito Monden, Masateru Tsunoda, Mike Barker, and Kenichi Matsumoto explore four basic software engineering beliefs held by two midsize embedded software development organizations in Japan, and identify possible process improvement actions for each organization. They recommend that other organizations also use this approach to find possible directions to improve their process, which will result in better products.

Collectively, these six articles present a glimpse of emerging new applications and current advances in development and adoption of IT systems. They should inspire IT professionals and researchers to make further advances in their domains of interest to realize the vision of a better world for our own benefit and that of the generations that follow us.

—San Murugesan, Editor in Chief

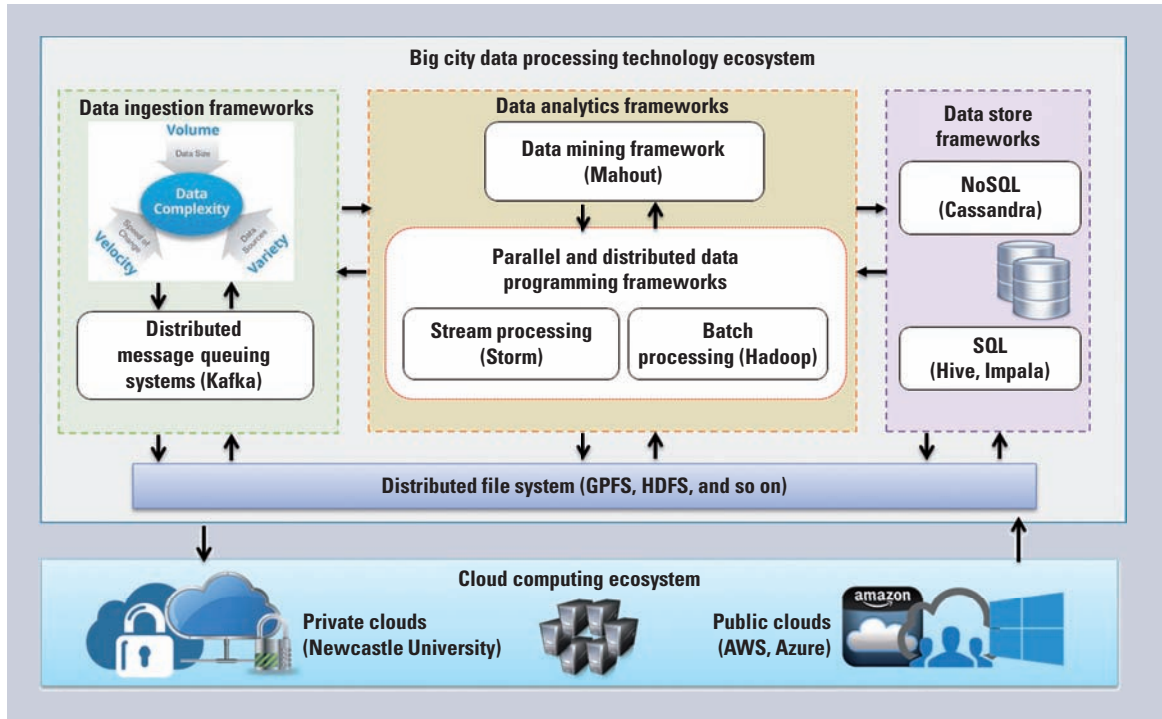


Figure 1. An urban risk analytics framework for processing heterogeneous city data. The framework comprises comprehensive components that satisfy the requirements of general urban risk analytics.

extrapolate such measurements to the entire city. However, to gain maximum utility from such a diverse range of data, we require new integration approaches and associated analytics. This has been identified as a grand challenge problem in the computing science domain.^{1,2}

State of the Art

Figure 1 shows the conceptual architecture of our cloud-based urban risk analytics framework. This proposed framework comprises comprehensive components that satisfy the requirements of general urban risk analytics. The framework has several main components.

Big City Data Processing Technology Ecosystem

This layer includes big data processing frameworks (BDPFs) that enable the creation of a big data application architecture. These frameworks can be classified as follows.

Distributed message queuing frameworks. Such frameworks provide a reliable, high-throughput, and low-latency system of queuing real-time datastreams from social media and other streaming sources. Examples include Amazon Kinesis and Apache Kafka.

Data mining frameworks. These frameworks implement a wide range of data analysis algorithms for analyzing massive datasets, from natural language processing (NLP, including latent Dirichlet allocation, regression, or naïve Bayes) to computational statistics (Bayesian networks or state vector machines). Examples include FlexGP, Apache Mahout, MLBase, and Apache SAMOA.

Parallel and distributed data programming frameworks. These frameworks, such as Apache Hadoop, Apache Spark, and Apache Storm, provide a distributed system implementation of

big data programming models that includes stream processing and batch processing. Distributed system resource management complexities such as task scheduling, data staging, fault management, inter-process communication, and result collection are automatically taken care of in Apache Hadoop and Apache Storm. The large-scale data mining frameworks mentioned previously are generally implemented on top of Hadoop, Spark, or Storm (see Figure 1).

Data store frameworks. These include SQL and NoSQL database frameworks in which message queuing, data mining, and parallel or distributed data programming frameworks persist the intermediate and final data. NoSQL frameworks (such as MongoDB, HyperTable, Cassandra, or Amazon Dynamo) support data manipulation based on nonrelational primitives. Such nonrelational data manipulation patterns lead to better

scalability and performance for unstructured data (for instance, social media postings or mobile app data). On the other hand, SQL data stores (MySQL, SQL Server, or PostgreSQL) are based on relational data manipulation primitives in which SQL can be used to manipulate data (insert, delete, or update). Urban risk analytics frameworks will use both NoSQL and SQL data stores (see Figure 1), driven by data variety and querying needs.

Cloud Computing Ecosystem

This layer comprises hardware resources (CPU, storage, and networking) provided by private (the Natural Environment Research Council datacenters, for example) and public (Amazon Web Services) cloud datacenters. The hardware resources at this layer provide computational and storage capabilities to the big data processing frameworks (mentioned previously). The end-to-end lifecycle operations³ (including selection, deployment, monitoring, and runtime control) of big data programming frameworks on cloud resources can be dynamically controlled via research orchestration frameworks.⁴

Current big data analysis frameworks (such as Apache YARN or Mesos) do not need to meet the requirement raised by new classes of applications—that is, no workflows, no dynamic indexing of existing and new data sources, no cloud-based implementation, and no dynamic tuning of the performance of big data processing frameworks to meet users' decision-making requirements. Applications such as urban risk analytics, however, require support for holistically processing data emitted by multiple sources.

Data Analytics Challenges

A variety of urban risk management applications can lead to new

research opportunities in urban risk analytics. The following research challenges—including data classification, data indexing, trajectory data, and edge analytics—arise when developing cloud-based data integration and analytics algorithms for urban risk management.

Data Classification

Datasets from multiple sources (social media, mobile apps, Instagram, and sensor networks, for example) flow at different speeds and volume, and in heterogeneous formats (text streams from social media or mobile apps and numeric streams from landslide sensors, for instance). This leads to heterogeneous requirements in terms of developing computer algorithms for data classification (NLP for text streams, and continuous numeric computation, including finding the max, min, average, and standard deviation, over streams from landslide sensors) and event detection (detecting the occurrence of keywords from social media streams and detecting flooding, landslide, or tsunami signals from real-time sensor streams).

Furthermore, based on data characteristics (static versus real-time), these computer algorithms will need to be implemented in multiple BDPFs that support heterogeneous programming abstractions. For example, static or historical datasets are in general handled by frameworks such as Apache Hadoop and Apache Mahout (a machine learning library for Hadoop), which offer map and reduce functions. On the other hand, computer algorithms for classification and event detection (also known as sliding window analytics) over real-time data will need to be implemented in stream processing frameworks such as Apache Storm and Yahoo S4. It is well understood that pro-

gramming computer algorithms in these BDPFs that can handle multisource and multiformat data simultaneously—while ensuring data processing efficiency (that is, minimizing query response time, maximizing event detection precision and accuracy, and so on)—is a hard research problem.^{1,2}

Data Indexing

Developing an indexing algorithm that can seamlessly integrate and establish relationships among static and real-time data across multiple sources in a multidimensional querying context (spatial, temporal, semantics, source types, event types, and so on) remains a very challenging problem.⁵ Although it is relatively straightforward to design relational or nonrelational schema to store the raw or classified data for a single source type (such as social media or sensor feeds), establishing a relationship and dependencies among the sources in a multidimensional querying context remains an unsolved problem.

Trajectory Data

Dealing with the trajectories of dynamic data produced by multiple sources is also a challenge (for example, the trajectories of taxis and buses are sequences of GPS samples, whereas the trajectories of smartcard ticketing devices are sequences of bus or subway stations). Notably, these trajectories differ in terms of data velocity, volume, and location accuracy.

Edge Analytics

Latency-sensitive data analytics tasks (such as analyzing streaming data from sensors) can benefit from “edge analytics” techniques,⁶ which have benefits including

- reduced network congestion achieved by filtering non-relevant events at the edge; and

- reduced event-detection latency (such as detecting dangerous water flow levels by analyzing real-time images in on-board processors available in sensor gateways such as Raspberry Pi 3), as sensors and gateways no longer need to send data to far-off cloud datacenters.

However, it remains an open challenge how to enact and provision data analytics tasks across edge and cloud datacenters so that decision-making latency is minimized while event-detection precision and accuracy is maximized.

Research Gap Analysis

Despite the recent emergence of clouds (Microsoft Azure, Google App Engine, and Amazon Web Services, for instance) that provide virtualized hardware resources and BDPFs, the state of the art in efficiently undertaking multisource and multidimensional big data analytics for urban risk management domains is still fairly primitive. For example, BDPFs such as Apache Mahout and Apache SAMOA provide a platform for developing and executing classification and event-detection algorithms (based on machine learning algorithms for NLP) over Apache Hadoop and Apache Storm, respectively. However, they provide no guidance on how to define and model “events” relevant to a particular data source type or how to train the existing NLP algorithms to automatically detect and query⁷ these events from the real-time and historical data.

Moreover, BDPFs have no knowledge of the underlying machine learning algorithm and the overarching data analytics application. Hence, they are unable to adapt the algorithm’s performance based on application requirements and cloud resource availability. Furthermore, there is still a gap in the development of unsupervised

machine learning approaches that can help match a given data source to the best and most accurate machine learning algorithm based on application-level goals,⁸ such as maximizing event detection accuracy and precision, minimizing querying latency across multiple data sources, and so on.

Furthermore, the Spark project at the University of California, Berkeley, released a new heterogeneous data querying engine called Spark SQL.⁹ Spark SQL’s DataFrame API is able to manage a distributed collection of data organized into named columns,¹⁰ which is similar to a traditional database. Multiple data sources from both external databases (JavaScript Object Notation, relational database management systems, or Apache Hive) and internal Spark data collections can be manipulated and processed through this API. In addition, the mechanisms of multidimensional querying and ad hoc analysis are important to urban risk analysis frameworks. Integrating online analytical processing—a business intelligence technique—with DataFrame is one potential challenge for big data integration. Although Spark SQL can query multiple structured data sources, it cannot automatically integrate and resolve dependencies across those data sources in a multidimensional querying context, as noted.

Integrating and analyzing heterogeneous sensor data from multiple sources in an urban risk analytics framework is very hard due to the variety of data formats and sources. An effective urban risk analytics framework is driven by enabling technologies, which can range from in situ sensor technology to remote sensing technology. Moreover, with the high volume and extremely high rate of the datastreams generated by heterogeneous sensors, ontol-

ogy and Semantic Web technologies have emerged as one possible solution for integrating heterogeneous data. In other words, to develop an effective mechanism for urban risk data integration, there is a strong requirement to provide a formal description of the relationships among the variety of data sources.

Ontology engineering is a widely used technique in data integration, in which a knowledge base is captured from multiple sources (articles, domain experts, processes, and so on), and knowledge is modeled using some standardized ontology language (for instance, the Web Ontology Language). Recently, a number of methodologies have been proposed for developing multisource data integration ontologies.¹¹ METHONTOLOGY is one method that is widely used to develop ontologies in several domains.¹² This method provides completed processes that cover the whole lifecycle of ontology development. Based on this, ontology engineering has become important in establishing a common understanding among experts from different areas that are working toward urban risk data analytics frameworks.


The Semantic Sensor Network Ontology (SSN) is a W3C standard for describing the concepts of sensors and observations. These concepts include sensor and sensor network modeling, measuring capabilities, sensor data, constraints, processes, deployments, and so on. SSN is widely used in sensor-based applications, including satellite imagery, scientific monitoring, and industrial infrastructure (www.w3.org/TR/vocab-ssn/). SSN is a key ontology used for integrating varieties of sensor data and analyzing disastrous events. However, these comprehensive concepts do not cover descriptions related to specialized

urban risks such as flooding, tsunamis, landslides, and so on.

Future Research Directions

We envision the following research activities within the context of cloud-based urban risk analytics frameworks:

- algorithmic techniques for urban risk analytics that support storage, classification, and event detection over data obtained from multiple sources, both in real time (such as data emitted by wireless sensor networks) and via historical repositories (for example, Twitter Firehose);
- scalable data integration (meta-data management) techniques that can enable multidimensional querying over heterogeneous, real-time, and historical data in multiple contexts (spatial, temporal, semantics, source types, event types, and so on); and
- cloud resource management methodologies that can seamlessly deal with heterogeneity in data analytic tasks, computational models, big data programming models, and cloud resource types (datacenter versus network edge, for example).

In summary, urban risk analytics has exhibited great potential in cloud computing and big city data research to realize urban risk management. Our proposed framework provides a conceptual architecture along with comprehensive guidance that supports data integration and analytics for urban risk management. 

References

1. R. Ranjan, "Streaming Big Data Processing in Datacenter Clouds," *IEEE Cloud Computing*, vol. 1, no. 1, 2014, pp. 78–83.
2. L. Wang and R. Ranjan, "Processing Distributed Internet of Things Data in Clouds," *IEEE Cloud Computing*, vol. 2, no. 1, 2015, pp. 76–80.
3. R. Ranjan et al., "Cloud Resource Orchestration Programming: Overview, Issues, and Directions," *IEEE Internet Computing*, vol. 19, no. 5, 2015, pp. 46–56; doi: 10.1109/MIC.2015.20.
4. R. Ranjan, K. Mitra, and D. Georgakopoulos, "MediaWise Cloud Content Orchestrator," *J. Internet Services and Applications*, vol. 4, no. 2, 2013; doi:10.1186/1869-0238-4-2.
5. E. Bertina, S. Nepal, and R. Ranjan, "Building Sensor-Based Big Data Cyberinfrastructures," *IEEE Cloud Computing*, vol. 2, no. 5, 2015, pp. 64–69.
6. M. Villari et al., "Osmotic Computing: A New Paradigm for Edge/Cloud Integration," *IEEE Cloud Computing*, vol. 3, no. 6, 2016, pp. 76–83; doi: 10.1109/MCC.2016.124.
7. A. Khoshkbarforousha and R. Ranjan, "Resource and Performance Distribution Prediction for Large Scale Analytics Queries," *Proc. 7th ACM/SPEC Int'l Conf. Performance Eng.*, 2016, pp. 49–54.
8. M. Wang et al., "A Case for Understanding End-to-End Performance of Topic Detection and Tracking-Based Big Data Applications in the Cloud," *Lecture Notes of the Inst. for Computer Sciences, Social Informatics, and Telecommunications Eng.*, vol. 169, 2015, pp. 315–325.
9. M. Armbrust et al., "Spark SQL: Relational Data Processing in Spark," *Proc. 2015 ACM SIGMOD Int'l Conf. Management of Data*, 2015, pp. 1383–1394.
10. M. Armbrust et al., "Spark SQL: Relational Data Processing in Spark," *Proc. 2015 ACM SIGMOD Int'l Conf. Management of Data*, 2015, pp. 1383–1394.
11. M. Fernández-López, "Overview of Methodologies for Building Ontologies," *Proc. IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5)*, 1999, pp. 4:1–4:13.
12. M. Fernández-López, A. Gómez-Pérez, and N. y Juristo, "METH-

ONTOLOGY: From Ontological Art Towards Ontological Engineering," *Proc. Ontological Eng. AAAI-97 Spring Symp. Series*, 1997.

Rajiv Ranjan is a reader (associate professor) in the School of Computing Science at Newcastle University, UK. He works on projects related to emerging areas in parallel and distributed systems, including cloud computing, the Internet of Things, and big data. Contact him at raj.ranjan@ncl.ac.uk.

Jedsada Phengsuwan is a PhD student in the School of Computing Science at Newcastle University, UK. His research interests are in big data integration, real-time stream processing, and ontology. Contact him at j.phengsuwan2@newcastle.ac.uk.

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.

Stuart Barr is a senior lecturer 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, collaborating with civil engineers on developing robust solutions and adaptation options for a range of climate change hazards. Contact him at stuart.barr@ncl.ac.uk.

Aad van Moorsel is a professor in distributed systems and Head of School at the School of Computing Science, Newcastle University, UK. His research is in security, privacy, and trust, with elements of quantification through system measurement, predictive modeling, or online adaptation. Contact him at aad.vanmoorsel@ncl.ac.uk.

 Read your subscriptions through the myCS publications portal at <http://mycs.computer.org>.