CHAPTER 5

Research Trends in Big Data Analysis

Big data analysis is a very active research area with significant impact on industrial and scientific domains, where is important to analyze very large and complex data repositories. In particular, in many cases, data to be analyzed are stored in cloud platforms and elastic computing clouds facilities are exploited to speedup the analysis. This chapter outlines and discusses main research trends in big data analytics and cloud systems for managing and mining large-scale data repositories. Topics and trends in the areas of exascale computing and social data analysis are reported. Section 5.1 discusses issues and challenges for implementing massively parallel and/or distributed applications in the area of big data analysis on exascale systems. Section 5.2 discusses recent trends in social data analysis, with a focus on mining mobility patterns from large volumes of trajectory data from online social network data. Finally, Section 5.3 discusses key research areas for the implementation of scalable data analytics dealing with huge, distributed data sources.

5.1 DATA-INTENSIVE EXASCALE COMPUTING

Computer system performance and storage capacity have increased very significantly in the past decades. This prodigious growth has powered many innovations across all sectors of our society. New advances in biology and medicine, physics and engineering, energy, goods design and manufacturing, transportation, environmental modeling, Internet services, financial analysis, and social media mainly depend on unceasing rises in computer performance and data storage.

In this scenario, the design of exascale computers is a very significant research challenge that is under investigation in the timeframe 2010–2020 with the goal of building computers composed of a large number of multicore processors (with more than 100 cores per chip) expected to deliver a performance of 10^{18} operations per second.

From a software point of view, these new computing platforms open big issues and challenges for software tools and environments and runtime systems that must be able to manage a high degree of parallelism and data locality. Additionally, to provide efficient methods for storing, accessing, and communicating data, intelligent techniques for data analysis and scalable software architectures enabling the scalable extraction of useful information and knowledge from data, are needed. Moreover, exascale systems and models are required to design and implement massively parallel and/or distributed algorithms and applications in the area of big data analysis. This trend needs new models and technologies for enabling cloud computing systems, HPC architectures and extreme scale platforms to support the implementation of clever data analysis algorithms that ought to be scalable and dynamic in resource usage.

Complex data analysis tasks involve data- and compute-intensive algorithms. These algorithms require large and efficient storage facilities together with high-performance processors. In this setting, exascale-computing infrastructures will play the role of an impressive platform for addressing both the computational and data storage needs of big data analysis applications. However, as aforementioned, in order to have a complete scenario, efforts must be performed for implementing big data analytics algorithms, architectures, programming tools, and applications in exascale systems.

We expect that data analytics systems on large-scale clouds and massively parallel systems will become common platforms for big data analytics within a few years. As new computing infrastructures will become common scalable platforms for big data analytics, programming tools, suites, and data mining strategies will be ported on such platforms for developing big data discovery solutions. In particular, the exploitation of the distributed workflow paradigm in the area of big data analytics will result in scalable solutions for big data analysis. Since offering data analysis as a service appears to be a viable approach to implement pervasive big data applications, the exploitation of exascale scalable computing platforms will provide the appropriate infrastructure for such service delivery.

5.1.1 Exascale Scalability in Data Analysis

As discussed in the previous chapters, data analysis gained importance because of the large and pervasive availability of data sources and the

continuous enhancements of techniques and algorithms to find insights in them. As a matter of fact, as data analysis technology advances, exploiting the power of data analytics is restructuring several scientific and industrial sectors. The amount of data that social networks daily generate is just one example of this trend. About 100 TB of data is uploaded daily to Facebook and Twitter. These notable amounts of data streams show that it is vital to design scalable solutions to process and analyze massive datasets. As a general forecast, some experts estimate data generated to reach about 45 ZB worldwide by 2020.

There is a large consensus on the fact that scalability and performance requirements are challenging conventional data storage, file systems, and database management systems. Architectures of such systems have reached some limits in handling very large processing tasks involving petabytes of data because they have not been built for scaling. This scenario demands for new architectures and solutions for implementing analytics platforms that must process big data for extracting complex knowledge models. Exascale systems, both from the hardware and the software side, can play a key role to support solutions for these problems. Indeed, many applications require the use of scalable data analysis platforms. A well-known example is the ATLAS detector from the Large Hadron Collider at CERN in Geneva. The ATLAS infrastructure has a capacity of 200 PB of disk and 300,000 cores, with more than 100 computing centers connected via 10 Gbps links. The data collection rate is very high and ATLAS actually only records a fraction of the data produced by the collider. Different teams of scientists run complex applications to analyze portions of those huge volumes of data. This analysis would be impossible without a high-performance infrastructure that supports data storage, communication, and processing. Astronomy is another good example of the massive data amounts that today are collected for scientific analysis. Computational astronomers are collecting and producing larger and larger datasets each year that without scalable infrastructures cannot be managed and processed.

If we move from science to society, we can consider social data and e-health. Social networks, such as Facebook and Twitter, have become very popular and are receiving increasing attention from the research community since, through the huge amount of user-generated data, they provide precious information concerning human dynamics and

behaviors. When the volume of data to be analyzed is of the order of terabytes or petabytes (billions of Tweets or posts), scalable storage and computing solutions must be used. The same occurs in the e-health domain, where huge amounts of patient data are available and can be used for improving therapies, for forecasting and tracking of health data, for the management of hospitals and health centers, for improving patient understanding, and/or physician-patient communication with analytics.

5.1.2 Programming Issues for Exascale Data Analysis

Implementing scalable data analysis applications in exascale computing systems is a complex job, which requires high level fine-grained parallel constructs and skills in parallel and distributed programming. In particular, mechanisms and expertise are needed to express task dependencies and intertask parallelism, to use mechanisms of synchronization, load balancing, and to properly manage the memory and the communication among a very large number of tasks. Moreover, if the computing infrastructures are heterogeneous and require different libraries and tools to program applications on them, the problems are even more complex. To cope with all these issues, different scalable programming models have been proposed to write data-intensive applications (Diaz et al., 2012).

Scalable programming models may be categorized based on their level of abstraction (high- and low-level scalable models) and on how they allow programmers to create applications (visual or code-based formalisms). Using high-level scalable models, a programmer defines only the high-level logic of an application, while the low-level details that are not essential for application design are hidden, including infrastructure-dependent execution details. The programmer is helped in application definition, and application performance depends on the compiler that analyzes the application code and optimizes its execution on the underlying infrastructure.

Instead, low-level scalable models allow the programmers to interact directly with computing and storage elements of the underlying infrastructure and thus to define the applications parallelism directly. In this case, programming an application requires more skills, and the application performance strongly depends on the quality of the code written by the programmer.

Data analysis applications can be designed through a visual interface, which is a convenient design approach for high-level users, for example, domain-expert analysts having a limited understanding of programming. In addition, a visual representation of workflows intrinsically captures parallelism at the task level, without the need to make parallelism explicit through control structures (Maheshwari and Montagnat, 2010). Code-based formalism allows users to program complex applications more rapidly, in a more concise way, and with higher flexibility (Marozzo et al., 2015). The code-base applications can be designed in different ways:

- With a language or an extension of language that allows to express parallel applications;
- With annotations in the application code that allow the compiler to identify which instructions will be executed in parallel; and
- Using a library in the application code that adds parallelism to application.

Given the variety of data analysis applications and types of users (from end users to skilled programmers) that can be envisioned in future exascale systems, there will be a need for scalable programming models with different levels of abstractions (high- and low-level) and different design formalisms (visual- and code-based), according to the aforementioned classification. Thus, the programming models should adapt to user needs by ensuring a good trade-off between ease in defining applications and efficiency of executing them on exascale architectures composed by a massive number of processors.

Data-intensive applications are software programs that have a significant need to process large volumes of data (Gorton et al., 2008). Such applications devote most of their processing time to run I/O operations, exchange, and move data among the processing elements of a parallel computing infrastructure. Parallel processing of data analysis applications typically involves accessing, preprocessing, partitioning, aggregating, querying, mining, and visualizing data that can be processed independently. These operations are executed using application programs running in parallel on a scalable computing platform that can be a large cloud system or an exascale machine composed of many thousands of processors. In particular, the main challenges for programming data analysis applications on exascale computing systems come from

the potential scalability and resilience of mechanisms and operations made available to developers for accessing and managing data. Indeed, processing very large data volumes requires operations and new algorithms, that are able to scale in loading, storing, and processing massive amounts of data that generally must be partitioned in very small data grains, on which analysis is done by thousands to millions of simple parallel operations.

Evolutionary models have been recently proposed that extend or adapt traditional parallel programming models, such as MPI, OpenMP, and MapReduce (e.g., Pig Latin) to limit the communication overhead in message passing paradigms or to limit the synchronization control, if shared-models languages are used (Gropp and Snir, 2013). On the other hand, new models, languages and APIs based on a revolutionary approach, such as X10, ECL, GA, SHMEM, UPC, and Chapel have been implemented. These novel parallel paradigms are devised to address the requirements of massive parallelism. Languages such as X10, UPC, GA, and Chapel are based on a partitioned global address space (PGAS) memory model that can be suited to implement data-intensive exascale applications because it uses private data structures and limits the amount of shared data among parallel threads. Together with different approaches, such as Pig Latin and ECL, those models must be further investigated and adapted to provide data-centric scalable programming models useful to support the efficient implementation of exascale data analysis applications composed of up to millions of computing units, which process small data elements and exchange them with a very limited set of processing elements.

A scalable programming model based on basic operations for data intensive/data-driven applications must include operations for parallel data access, data-driven local communication, data processing on limited groups of cores, near-data synchronization, in-memory querying, group-level data aggregation, and locality-based data selection and classification. An efficient model must be able to manage a very large amount of parallelism, implement-reduced communication, and synchronization.

Supporting efficient data-intensive applications on exascale systems will require an accurate modeling of basic operations and the

programming languages/APIs that will include them. At the same time, a significant programming effort of developers will be needed to implement complex algorithms and data-driven applications such as those used, for example, in big data analysis and distributed data mining. Programmers must be able to design and implement scalable algorithms by using the operations sketched above. To reach this goal, a coordinated effort between the operation/language designers and the application developers would be very fruitful.

At the exascale scale, the cost of accessing, moving, and processing data across a parallel system is enormous. This requires mechanisms, techniques and operations for efficient data access, placement and querying. In addition, scalable operations must be designed in such a way so as to avoid global synchronizations, centralized control, and global communications. Many data scientists want to be abstracted away from these tricky and lower level aspects of HPC until at least they have their code working and then potentially to tweak communication and distribution choices in a high level manner, in order to further tune their code. Interoperability and integration with the MapReduce model and MPI must be investigated with the main goal of achieving scalability on large-scale data processing.

Different data-centric abstractions can be integrated in order to provide a unified programming model and API that allow the efficient programming of large-scale heterogeneous and distributed memory systems. A software implementation can be developed as a library. Its application to reference data-intensive benchmarks allows gathering feedback that will lead to improvements in the prototype. In order to simplify the development of applications in heterogeneous distributed memory environments, large-scale data-parallelism can be exploited on top of the abstraction of n-dimensional arrays subdivided in tiles, so that different tiles are placed on different computing nodes that process in parallel the tiles they own. Such an approach allows one to easily process the tiles at each node in either regular CPUs or in any of the heterogeneous systems available (GPUs, Xeon Phi coprocessors, etc.) using a unified API and runtime that hides the complexity of the underlying process.

Abstract data types provided by libraries, so that they can be easily integrated in existing applications, should support this abstraction.

As mentioned earlier, another issue is the gap between users with HPC needs and experts with the skills to make the most of these technologies. An appropriate directive-based approach can be to design, implement and evaluate a compiler framework that allows generic translations from high-level languages to exascale heterogeneous platforms. A programming model should be designed at a level that is higher than that of standards, such as OpenCL. The model should enable the rapid development with reduced effort for different heterogeneous platforms, including low energy architectures and mobile devices.

5.2 MASSIVE SOCIAL NETWORK ANALYSIS

A huge amount of user-generated data in social networks can be exploited to extract valuable information concerning human dynamics and behaviors. Social data analysis is emerging as a fast growing research area, which is aimed at extracting useful information from this mass of data. It can be used for the analysis of collective sentiments to understanding the behavior of groups of people or the dynamics of public opinion.

One of the most interesting features of social networks is its ability to associate spatial context to social posts. For example, Twitter, Facebook, Flickr, and Instagram, exploit the GPS readings of mobile phones to tag Tweets, posts, and photos with geographical coordinates. Therefore, social network users traveling through sets of locations, produce a huge amount of geo-location data that embed extensive knowledge about human dynamics and mobility behaviors. The potential to harness rich information provided by geo-tagged social data may impact many areas including urban planning, intelligent traffic management, route recommendations, security, and health monitoring.

In the past few years, many studies have been carried out regarding the extraction of trajectories from geo-tagged social data (Zheng, 2015). Compared to trajectory pattern mining from GPS data (Giannotti et al., 2007), extracting trajectories from social network data is a more challenging task because data from location-based social networks are often sparse and irregular, in contrast to GPS traces of mobile devices, which are highly available and sampled at regular time intervals.

In most cases, geo-tagged data from social networks provide positioning information of a huge number of users, but information about each user is limited to very few positions per day. Therefore, trajectories are usually generated at low or irregular frequency, thereby leaving the routes between two consecutive points of a single trajectory uncertain.

A research stream in this area focuses on identifying hot spots and tourist routine behaviors from global collection of geo-referenced photos. Photo-sharing social networks contain billions of publicly accessible images that are taken virtually everywhere on earth. These photos are annotated with various forms of information including geo-spatial and textual metadata. For example, Yin et al. (2011) exploited Flickr data to identify the most frequent travel routes and the top interesting locations in a given geo-spatial region. This was obtained by associating semantics to the locations based on the tags given to each photo by Flickr users.

Another approach for trajectory mining from social network data consists of exploiting only spatio-temporal information, without leveraging image features and tag-based data. Comito et al., (2015) analyzed the time- and geo-referenced information associated with Twitter data, in order to detect typical trajectories and discover common behavior, that is, patterns, rules, and regularities in moving trajectories. The basic assumption is that people often tend to follow common routes: for example, they go to work every day traveling the same roads. Thus, if we have enough data to model typical behaviors, such knowledge can be used to predict and manage future movements of people. In particular, the goal of this work was to provide top interesting spots and frequent travel sequences among locations in a given geo-spatial region. Interesting locations include popular tourist destinations and commonly frequented public areas, such as shopping malls/streets, restaurants, and cinemas.

Given the large amount of data to process, the methodology proposed by Comito et al. (2015) consists of various phases allowing the collection of Tweets, detect locations from them, identify travel routes between such locations, mine frequent travel routes using sequential pattern mining, and extract spatial-temporal information for each of those routes to capture the factors that might drive users' movements. As a case study, the

methodology was applied to a large set of Tweets posted within the city of London. The analysis distinguishes among routes involving multiple people and routes taken by a single person. Collective routes indicate crowd mobility and also how people behave in the city, whereas individual routes characterize a given person, highlighting her/his daily mobility patterns and providing insights about her/his daily routines.

Even though trajectory information extracted from geo-tagged data are indicative of a user's behavior, it lacks some semantics about the type of place where a user is (e.g., home, office, museum), which would allow a better understanding of users' patterns. Some location-based social network services (e.g., Foursquare) allow each user to explicitly indicate the place category she/he is at. Although the category information is very rich and can enable more refined applications, this is a manual process, in which the user is voluntarily "checking into" a place. In other social networking tools, a location is simply represented as latitude-longitude coordinates automatically associated to a post by the service. However, knowing the semantics of the type of place a user is at can be potentially very useful. It could allow inferring users' common interests, improving activity prediction, and enabling mobile user recommendation and advertisement. Falcone et al. (2014) defined a method to extract spatial-temporal patterns from geo-Tweets exploiting a number of features specific to the places, duration of the stay, time of day and of week of the typical stay, number of visitors and the regularity of their behavior in the place. Another work that identifies place category from social network data is the one by Ye et al. (2011), which derived eight place label categories from Whrrl, a location-based social network. They used a support vector machine on features such as check-in frequency and time of day to label over 53,000 places from almost 6,000 users.

The work by Cesario et al. (2015) is an example on how social networks can be exploited to analyze the behavior of large groups of people attending popular events. The goal was to monitor the attendance of Twitter users during the FIFA World Cup 2014 matches to discover the most frequent movements of fans during the competition. The data source is represented by all geo-tagged Tweets collected during the 64 matches of the World Cup from June 12, 2014 to July 13, 2014. For each match, only the geo-tagged Tweets whose coordinates fell within the area of stadiums, during the matches, were considered. Then, a

trajectory pattern mining analysis on the set of Tweets considered was carried out. The analysis is based on the search of frequent item sets that allow identifying the groups of matches attended most frequently by spectators. Original results were obtained in terms of number of matches attended by groups of fans, clusters of most attended matches, and most frequented stadiums. The number of Tweets posted from inside the stadiums during the soccer matches was pretty high (about half a million), allowing their analysis with well-known data mining techniques, such as the Apriori algorithm, for frequent itemsets computing and association rules discovery. The methodology adopted to carry out this data analysis task could be adopted in similar scenarios, where groups of people attend social events to understand collective behaviors that are very hard to discover with traditional social analysis techniques.

Cloud systems can be effectively exploited to support trajectory-mining applications from social network data, such as those described previously, or from other data sources (e.g., open data). The application by Altomare et al. (2014), introduced in Section 4.4.1, demonstrates the benefits derived from the use of big data solutions for trajectory mining in urban computing and smart city applications. Urban computing is the process of acquisition, integration, and analysis of big and heterogeneous urban data to tackle major issues that cities face today, including air pollution, energy consumption, traffic flows, human mobility, environmental preservation, commercial activities, and savings in public spending. In urban computing scenarios, clouds can play an essential role by helping city administrators to quickly acquire new capabilities and reduce initial capital costs by means of a comprehensive pay-as-you-go solution. In fact, by providing applications, infrastructure, networking, systems software, middleware, and maintenance, cloud computing lowers the barrier of entry and enables city managers to deliver high quality services to their citizens. In addition, managing heterogeneous data volumes while allowing interoperability among different tools, also needs compliance to standards. In this regard, cloud computing systems are suitable platforms to fulfill most of the above requirements, due to their features such as scalable computing, on-demand processing, facilitating data accessibility, and storage across platforms.

Several cloud-enabled tools for urban planning and management, proposed so far, demonstrate the important role of cloud computing in

this area. Environmental Software and Services (ESS) exploits the cloud paradigm to offer a range of services for environmental planning and management,[1] policy, and decision making world wide. Analogously, the Environmental Virtual Observatory pilot (EVOp) uses clouds to achieve similar objectives in the soil and water domains.[2] The European Platform for Intelligent Cities (EPIC) combines a cloud computing infrastructure with the knowledge and expertise of the Living Lab approach to deliver sustainable,[3] user-driven Web services for citizens and businesses. The Life 2.0 project offers a set of services ranging from basic geographical positioning systems to socially networked services and to local market-based services.[4] The project aims to provide solutions that increase opportunities for social contacts among elderly people in their local area, by providing new services based on the use of tracking systems and social network applications. Finally, IBM introduced Smarter City Solutions on the IBM SmartCloud Enterprise,[5] a public cloud platform that includes hardware, network, and storage. The platform provides pay-as-you-go services for urban management within cities. Those services include application software, infrastructure, networking, systems software, middleware, and maintenance.

5.3 KEY RESEARCH AREAS

As discussed in the previous sections, scalable data analytics requires high-level and easy-to-use design tools for programming large applications dealing with huge, distributed data sources. This necessitates further research and development in several key areas such as:

- *Programming models for big data analytics*: big data analytics programming tools require novel complex abstract structures. The MapReduce model is often used on clusters and clouds, but more research is needed to develop scalable high-level models and tools. State-of-the-art solutions generated major success stories, however

[1] www.ess.co.at
[2] www.evo-uk.org
[3] www.epic-cities.eu
[4] www.life2project.eu
[5] www-01.ibm.com/software/industry/smartercities-on-cloud

they are not mature and suffer several problems from data transfer bottlenecks to performance unpredictability.

- *Data and tool interoperability and openness*: interoperability is a main issue in large-scale applications that use resources such as data and computing nodes. Standard formats and models are needed to support interoperability and ease cooperation among teams using different data formats and tools.
- *Integration of big data analytics frameworks*: the service-oriented paradigm allows running large-scale distributed workflows on heterogeneous platforms along with software components developed using different programming languages or tools.
- *Scalable software architectures for fine grain in-memory data access and analysis*: exascale processors and storage devices must be exploited with fine-grain runtime models. Software solutions to handle many cores on processors and scalable processor-to-processor communications have to be designed to exploit exascale hardware.
- *Tools for massive social network analysis*: the effective analysis of social network data on a large scale requires new software tools for real-time data extraction and mining, using cloud services and high-performance computing approaches. Social data streaming analysis tools represent very useful technologies to understand collective behaviors from social media data.
- *Tools for data exploration and model visualization*: new approaches for data exploration and model visualization are necessary, taking into account the size of data and complexity of the knowledge extracted. As data grow bigger, visualization tools will become more useful to summarize and show them in a compact and easy-to-see way.
- *Local mining and distributed model combination*: as big data applications often involve several local sources and distributed coordination, collecting distributed data sources into a centralized server for analysis, is neither practical nor possible. Scalable data analysis systems have to enable local mining of data sources, model exchange, and fusion mechanisms to compose the results produced in the distributed nodes (Wu et al., 2014). According to this approach, the global analysis can be performed by distributing local mining and supporting the global combination of every local knowledge to generate the complete model.

- *In-memory analysis:* most of the data analysis tools query data sources on disks while, differently from those, in-memory analytics query data in main memory (RAM). This approach brings many benefits in terms of query speed up and faster decisions. In-memory databases are, for example, very effective in real-time data analysis, but they require high-performance hardware support and fine-grain parallel algorithms. New 64-bit operating systems allow addressing memory, up to 1 TB, making it realistic to cache a very large amount of data in RAM. Hence, this research area is very promising.

5.4 SUMMARY

The field of big data analysis is a very active research discipline that has significant impact on industrial and scientific processes and applications. In many cases, big data are stored and analyzed in cloud platforms. In this chapter two research areas, exascale computing and social data analysis, have been discussed and a few key research topics that need to be deeply investigated in the future to find solutions for the implementation of scalable data analytics on huge and distributed data sources, have been outlined.

In particular, the development of exascale computers is a very significant research challenge that is under investigation with the goal of building computers composed of up to hundreds of thousands cores for delivering a performance of 10^{18} operations per second. From a software point of view, these new computing platforms open big issues and challenges for software tools, environments, and runtime systems that must be able to manage a high degree of parallelism and data locality. Moreover, efficient methods to store, access, communicate, and mine data are needed. When those methods will be designed and implemented, exascale systems will be used to implement massively parallel and distributed applications in the area of big data analysis.

On the other side, social data analysis is emerging as a fast growing research area aimed at extracting useful information from data directly provided by people. It can be used for the analysis of collective sentiments, for understanding the behavior of groups of people, or the

dynamics of public opinion. In this chapter, we introduced the main issues that are addressed in this area for developing scalable data analysis by exploiting cloud computing features and discussed some recent research contributions that show how social data analysis can be carried out in smart city and urban computing applications.

The eight key research areas listed in the previous section represent important research topics, which are very promising for researchers and professionals working in the area of cloud-based data analysis. Future solutions that will be created in these areas will have significant impact on the development of big data analysis frameworks and applications in the next decade.

REFERENCES

Altomare, A., Cesario, E., Comito, C., Marozzo, F., Talia, D., 2014. Trajectory pattern mining over a Cloud-based framework for urban computing. In: Proceedings of the 16th International Conference on High Performance Computing and Communications (HPCC 2014), IEEE, Paris, France, pp. 367–374.

Cesario, E., Congedo, C., Marozzo, F., Riotta, G., Spada, A., Talia, D., Trunfio, P., Turri, C., 2015. Following soccer fans from geotagged tweets at FIFA world cup 2014. In: Proceedings of the 2nd IEEE Conference on Spatial Data Mining and Geographical Knowledge Services (ICSDM 2015), Fuzhou, China, pp. 33–38.

Comito, C., Falcone, D., Talia, D., 2015. Mining popular travel routes from social network geotagged data. In: Damiani, E., Howlett, R.J., Jain, L.C., Gallo, L., De Pietro, G. (Eds.), Intelligent interactive multimedia systems and services, pp. 81–95.

Diaz, J., Munoz-Caro, C., Nino, A., 2012. A survey of parallel programming models and tools in the multi and many-core era. IEEE Trans. Parallel Distr. Syst. 23 (8), 1369–1386.

Falcone, D., Mascolo, C., Comito, C., Talia, D., Crowcroft, J., 2014. What is this place? Inferring place categories through user patterns identification in geo-tagged tweets. In: Proceedings of the International Conference on Mobile Computing, Applications, and Services (MobiCASE 2014), Austin, TX, USA.

Giannotti, F., Nanni, M., Pinelli, F., Pedreschi, D., 2007. Trajectory pattern mining. In: Proceedings of the Thirteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07). ACM, New York, NY, USA, pp. 330–339.

Gorton, I., Greenfield, P., Szalay, A.S., Williams, R., 2008. Data-intensive computing in the 21st century. IEEE Comput. 41 (4), 30–32.

Gropp, W., Snir, M., 2013. Programming for exascale computers. Comput. Sci. Eng. 15 (6), 27–35.

Maheshwari, K., Montagnat, J., 2010. Scientific work flow development using both visual and script-based representation. In: Proceedings of the 6th World Congress on Services, SERVICES '10, Washington, DC, USA, pp. 328–335.

Marozzo, F., Talia, D., Trunfio, P., 2015. JS4Cloud: Script-Based Workflow Programming for Scalable Data Analysis on Cloud Platforms. Concurrency and Computation: Practice and Experience. Wiley InterScience, Chichester.

Wu, X., Zhu, X., Wu, G.-Q., Ding, W., 2014. Data mining with big data. IEEE Trans. Knowledge Data Eng. 26 (1), 97–107.

Ye, M., Shou, D., Lee, W.-C., Yin, P., Janowicz, K., 2011. On the semantic annotation of places in location-based social networks, In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2011), San Diego, CA, pp. 520–528.

Yin, Z., Cao, L., Han, J., Luo, J., Huang, T.S., 2011. Diversified trajectory pattern ranking in geotagged social media, In: Proceedings of the 11th SIAM International Conference on Data Mining (SDM 2011), Mesa, AZ, pp. 980–991.

Zheng, Y., 2015. Trajectory data mining: an overview. ACM Trans. Intel. Syst. Technol. 6 (3), 1–41.