Big Sequence Management: Scaling up and Out

Karima Echihabi
Mohammed VI Polytechnic University
karima.echihabi@um6p.ma

Kostas Zoumpatianos*
LIPADE, Université de Paris
konstantinos.zoumpatianos@u-paris.fr

Themis Palpanas
LIPADE, Université de Paris
French University Institute (IUF)
themis@mi.parisdescartes.fr

ABSTRACT

Data series are a prevalent data type that has attracted lots of interest in recent years. Specifically, there has been an explosive interest towards the analysis of large volumes of data series in many different domains. This is both in businesses (e.g., in mobile applications) and in sciences (e.g., in biology). In this tutorial, we focus on applications that produce massive collections of data series, and we provide the necessary background on data series storage, retrieval and analytics. We look at systems historically used to handle and mine data in the form of data series, as well as at the state of the art data series management systems that were recently proposed. Moreover, we discuss the need for fast similarity search for supporting data mining applications, and describe efficient similarity search techniques, indexes and query processing algorithms. Finally, we look at the gap of modern data series management systems in regards to support for efficient complex analytics, and we argue in favor of the integration of summarizations and indexes in modern data series management systems. We conclude with the challenges and open research problems in this domain.

1 INTRODUCTION

In various scientific and industrial domains analysts are required to measure quantities as they fluctuate over a dimension; these values are commonly called *data series* or *sequences*. The dimension over which data series are ordered depends on the application domain and can have various diverse physical meanings. By far, the most common dimension over which data are ordered is time. In this case, we specifically talk about *time series*. Other applications though, produce series ordered over position (DNA sequences), mass (mass spectrometry) or angle (shapes). In all cases, data have to be captured, stored and analyzed as series rather than individual values.

Applications range from forecasting methods to correlation analysis, summarization, representation methods, sampling, outlier detection and more [6–8, 35, 41]. Moreover, it is not unusual for applications to involve numbers of sequences in the order of hundreds of millions to billions [1, 3]. As a result, analysts are more frequently than ever deluged by the vast amounts of data series that they have to filter, process and understand. Consider for instance, that for several of their analysis tasks, neuroscientists are currently reducing each of their 3,000 point long sequences to a single number (the global average) in order to be able to analyze their huge datasets [1]. In astronomy, there are currently available more than 70TB of spectroscopic sequence data from 200 million sky objects, collected by the Sloan Digital Sky Survey [3], allowing scientists to study the universe. These data have to be processed and analyzed, in order to identify patterns, gain insights, and detect abnormalities.

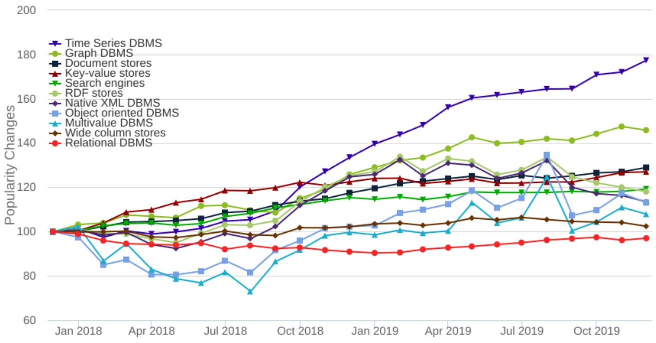

Figure 1: DBMS category popularity change trend [2]

Recent advances in domains such as cloud computing and data centers, IoT and smart cities, self-driving cars and communications, generated a tremendous interest in developing specialized systems able to manage and mine data series. This is evident both by industrial [42] [17] and academic interest [5, 28], as well as through popularity studies [2], where time series management systems gather the most intense interest change over the last two years, as shown in Figure 1.

Our goal is to describe the current state in data series management, including applications, query types and data types, complex analytic algorithms, their components and their implementation in modern data systems. Further on we will explore how modern techniques can be leveraged to speed up complex analytical pipelines, and take a glimpse on how these techniques can be improved by applying machine learning.

2 SEQUENCE MANAGEMENT OVERVIEW

We take a holistic look at the problem of managing and analyzing very large collections of data series, discuss the state-of-the-art and pinpoint the opportunities for optimizing complex query execution.

[Introduction and Foundations] We will start by looking at some foundational aspects of data series management. Those include the *data characteristics*, the query workloads, and the *specialized data structures* used to index sequential data. Data series can be categorized under many dimensions: i.e., the way that data arrive: streaming vs static, the lengths of data series: fixed vs variable length per series, the way that points are sampled: fixed intervals vs variable sampling intervals, and the presence of uncertainty in their values.

In terms of workloads, we will then look at various applications and query patterns that recur in each one of those. Specifically, we will discussing both simple Selection-Projection-Transformation (SPT) queries, where analysts filter based on data properties (e.g.,

*The author is currently at Snowflake Computing thresholds) or meta-data values, as well as complex data mining (DM) analytics, like clustering, outlier detection and more [39]. We will look at the core component of advanced analytics, which is similarity search, and look at the different flavors of this problem. Those include whole matching vs sub-sequence matching, exact vs approximate similarity search, as well as various distance measures that are commonly used in practice. Finally, we will briefly talk about the different data structure categories that exist, and how they are used to organize and retrieve data in each one of the aforementioned query patterns.

[Complex Analytics] We will dive in analytics like outlier detection [12, 16], frequent pattern mining [51], clustering [29, 52, 54, 63], and classification [13]. Such analytics involve a series of operations that are performed in a pre-processing step (e.g., sliding windows, normalization, interpolation, etc.), as well as operations that are repeated in the context of an iterative algorithm (e.g., similarity search). We will discuss these operations, and pinpoint the ones that can be optimized at the database kernel level. Such operations include sliding windows, normalization, interpolation, and various transformations such as dft that are specific to each algorithm. During the iterative part of these analytics, multiple similarity search operations need to be performed. This is useful for finding series within a given radius from a centroid in clustering, or for identifying distances from a given model in anomaly detection and classification, but also for retrieving patterns in frequent pattern mining. All of these operations can be implemented externally, in the application side. However, since some of them are data-intensive, pruning or incremental computation can significantly improve their performance. For this reason, performing them at the database level can provide large improvements in terms of execution time. We will focus on similarity search as such an example, being a crucial and expensive component of most mining algorithms, and motivate a deep-dive at its characteristics and scalable implementations.

[Systems for Data Series Management] We will then look at current state-of-the-art systems, describing their storage layers and data structures, as well as how they implement the aforementioned data manipulation operations. In particular, we will both look at systems that have been specifically designed to support sequential data, as well as systems that have been adapted to support them.

Specialized systems either utilize custom storage layers, or existing solutions. Common off-the-shelf storage systems are log-structure merge tree (LSM) based engines like RocksDB and LevelDB, and distributed systems such as HBase. Custom engines utilize domain-specific compression, indexing and data partitioning to increase efficiency. They support both simple and complex analytical queries and some of the systems offer encryption and distributed query processing.

Beringei [42] is developed by Facebook, it has a custom in-memory storage engine. It compresses and organizes data in a series per series scheme. CrateDB [15] partitions data in chunks, stores them in a distributed file system, and indexes them using Apache Lucene. InfluxDB [27] uses Time-Structured Merge Trees (LSM tree variant), logging data on disk as they arrive, and periodically merge-sorting overlapping time-stamps. Prometheus [48] is based on the Beringei ideas. QuasarDB [49] utilizes either RocksDB or Hellium [26]. Riak TS [53] supports both LevelDB or Bitcask, which is a custom log structured hash table. Timescale [59] is a Postgres extension. It partitions time series both in groups of series as well as in distinct time segments. It then provides an abstraction of a single table. Finally, various systems such as OpenTSDB [38], Timely [58] (concentrated on security) and Warp10 [62] are developed on top of HBase.

All the aforementioned systems support range scans in the positions, aggregation functions and filtering. Beringei additionally supports correlation queries through a brute force implementation. Crate supports geospatial queries. InfluxDB supports queries like moving averages, prediction, transformations, etc, and Timescale supports gap filling.

[Advanced Techniques for Optimizing Analytics] We will present techniques for speeding up similarity search, which plays a central role in several algorithms related to complex data series analytics, and discuss opportunities for integrating such techniques in modern data series management systems. Previous work on similarity search has proposed the use of spatial indexes such as R-Trees with DFT [4, 50] and DHWT [11]. Specialized indexes are based on domain specific summarizations. Examples include DS-Tree [61], *i*SAX [40, 56], *i*SAX 2.0 [9], *i*SAX2+ [10], ADS+ [68, 68], SFA [55], Coconut [31, 32], and ULISSE [33, 34].

In addition, we will pay particular attention to parallel and distributed solutions for similarity search. These include methods that support both exact and approximate similarity search query answering, and make use of modern hardware (e.g., SIMD, multi-core, multi-socket, GPU) such as ParIS+ [43, 45], Delta-Top-Index [47], MESSI [44], and SING [46], as well as distributed computation (e.g., Spark) such as DPiSAX [65, 66], TARDIS [67], KV-match [64], MVS-match [23], and L-match [22]. These methods are in a much better position than traditional single-node techniques to address the scalability challenges of modern data series analytics applications that have to deal with very large data collections.

Apart from exact indexes, there are also various approximate index structures proposed in the literature. Those include methods based on hashing [30, 57], sketches and grid indexes [14], and kNN-Graphs [36, 37]. Recent studies [20, 21] have compared several data series and high-dimensional similarity search methods under a common framework, revealing multiple promising future research directions, which we will analyze.

[Challenges and Conclusions] Massive data series collections are becoming a reality for virtually every scientific and social domain. This leads to the need of designing and developing general-purpose Data Series Management Systems, able to cope with big data series, that is, very large and fast-changing collections of data series, which can be heterogeneous (i.e., originate from disparate domains and thus exhibit very different characteristics), and which can have uncertainty in their values (e.g., due to inherent errors in the measurements). These systems should have data series indexes and summarizations integrated into their engines, so as to speedup the time-intensive operations of complex analytics pipelines, and support interactive exploration of big data series. To this end, progressive analytics operators would also be very useful [24, 25, 60]. At the same time, the role that deep learning techniques can play should be studied in more detail, especially with regards to similarity search [18, 19] and query optimization. Finally, there is a pressing need for developing data series specific benchmarks [69, 70] able to stress test index structures in a principled way.

REFERENCES

[1] [n.d.]. ADHD-200. http://fcon_1000.projects.nitrc.org/indi/adhd200/.

[2] [n.d.]. DB-Engines. https://db-engines.com/en/ranking_categories.

[3] [n.d.]. Sloan Digital Sky Survey. https://www.sdss3.org/dr10/data_access/ \rvolume.php.

[4] Rakesh Agrawal, Christos Faloutsos, and Arun N. Swami. 1993. Efficient Similarity Search In Sequence Databases. In *FODO*.

[5] Andreas Bader, Oliver Kopp, and Michael Falkenthal. 2017. Survey and Comparison of Open Source Time Series Databases. In *BTW*.

[6] Anthony J. Bagnall, Richard L. Cole, Themis Palpanas, and Konstantinos Zoumpatianos. 9(7), 2019. Data Series Management (Dagstuhl Seminar 19282). *Dagstuhl Reports* (9(7), 2019).

[7] Paul Boniol, Michele Linardi, Federico Roncallo, and Themis Palpanas. 2020. Automated Anomaly Detection in Large Sequences. In *ICDE*.

[8] Paul Boniol and Themis Palpanas. 2020. Series2Graph: Graph-based Subsequence Anomaly Detection for Time Series. *PVLDB* (2020).

[9] Alessandro Camerra, Themis Palpanas, Jin Shieh, and Eamonn J. Keogh. 2010. iSAX 2.0: Indexing and Mining One Billion Time Series. In *ICDM*.

[10] Alessandro Camerra, Jin Shieh, Themis Palpanas, Thanawin Rakthanmanon, and Eamonn J. Keogh. 2014. Beyond one billion time series: indexing and mining very large time series collections with iSAX2+. *KAIS* 39, 1 (2014), 123–151.

[11] Kin-pong Chan and Ada Wai-Chee Fu. 1999. Efficient Time Series Matching by Wavelets. In *ICDE*.

[12] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)* 41, 3 (2009), 15. http://scholar.google.de/scholar.bib?q=info:jAfBmk-9uAcJ:scholar.google.com/&output=citation&hl=de&ct=citation&cd=0

[13] Yihua Chen, Eric K. Garcia, Maya R. Gupta, Ali Rahimi, and Luca Cazzanti. 2009. Similarity-based Classification: Concepts and Algorithms. *J. Mach. Learn. Res.* 10 (June 2009), 747–776. http://dl.acm.org/citation.cfm?id=1577069.1577096

[14] Richard Cole, Dennis E. Shasha, and Xiaojian Zhao. 2005. Fast window correlations over uncooperative time series. In *KDD*.

[15] Crate. 2018. CrateDB: Real-time SQL Database for Machine Data & IoT. http://crate.io/

[16] Michele Dallachiesa, Themis Palpanas, and Ihab F. Ilyas. 2014. Top-k Nearest Neighbor Search in Uncertain Data Series. *PVLDB* 8, 1 (2014).

[17] Lars Dannecker, Gordon Gaumnitz, Boyi Ni, and Yu Cheng. 2015. Multi-representation Storage of Time Series Data. US Patent 20170161340A1.

[18] Karima Echihabi. 2020. High-Dimensional Vector Similarity Search: From Time Series to Deep Network Embeddings. In *SIGMOD*.

[19] Karima Echihabi, Kostas Zoumpatianos, and Themis Palpanas. 2020. Scalable Machine Learning on High-Dimensional Vectors: From Data Series to Deep Network Embeddings. In *WIMS*.

[20] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2018. The Lernaean Hydra of Data Series Similarity Search: An Experimental Evaluation of the State of the Art. *PVLDB* 12, 2 (2018).

[21] Karima Echihabi, Kostas Zoumpatianos, Themis Palpanas, and Houda Benbrahim. 2019. Return of the Lernaean Hydra: Experimental Evaluation of Data Series Approximate Similarity Search. *PVLDB* (2019).

[22] Kefeng Feng, Peng Wang, Jiaye Wu, and Wei Wang. 2020. L-Match: A Lightweight and Effective Subsequence Matching Approach. *IEEE Access* 8 (2020), 71572–71583.

[23] Kefeng Feng, Jiaye Wu, Peng Wang, Ningting Pan, and Wei Wang. 2019. MVS-match: An Efficient Subsequence Matching Approach Based on the Series Synopsis. In *DASFAA*, Vol. 11448. Springer, 368–372.

[24] Anna Gogolou, Theophanis Tsandilas, Karima Echihabi, Themis Palpanas, and Anastasia Bezerianos. 2020. Data Series Progressive Similarity Search with Probabilistic Quality Guarantees. In *SIGMOD*.

[25] Anna Gogolou, Theophanis Tsandilas, Themis Palpanas, and Anastasia Bezerianos. 2019. Progressive Similarity Search on Time Series Data. In *Proceedings of the Workshops of the EDBT/ICDT 2019 Joint Conference, EDBT/ICDT*.

[26] Hellium. 2018. Hellium: Ultra high performance key/value storage. https://www.levyx.com/helium

[27] InfluxDB. 2018. InfluxDB - Open Source Time Series, Metrics, and Analytics Database (http://influxdb.com/). http://influxdb.com/

[28] Søren Kejser Jensen, Torben Bach Pedersen, and Christian Thomsen. 2017. Time Series Management Systems: A Survey. *TKDE* 29, 11 (2017).

[29] Eamonn Keogh and M. Pazzani. 1998. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *KDD*, R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro (Eds.). ACM Press, New York City, NY, 239–241.

[30] Yongwook Bryce Kim. 2017. *Physiological time series retrieval and prediction with locality-sensitive hashing.* Ph.D. Dissertation. Massachusetts Institute of Technology, Cambridge, USA.

[31] Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, and Themis Palpanas. 2018. Coconut: A Scalable Bottom-Up Approach for Building Data Series Indexes. *PVLDB* 11, 6 (2018), 677–690. https://doi.org/10.14778/3184470.3184472

[32] Haridimos Kondylakis, Niv Dayan, Kostas Zoumpatianos, and Themis Palpanas. 2019. Coconut: sortable summarizations for scalable indexes over static and streaming data series. *VLDBJ* 28, 6 (2019).

[33] Michele Linardi and Themis Palpanas. 2018. Scalable, Variable-Length Similarity Search in Data Series: The ULISSE Approach. *PVLDB* 11, 13 (2018), 2236–2248.

[34] Michele Linardi and Themis Palpanas. 2018. ULISSE: ULtra compact Index for Variable-Length Similarity SEarch in Data Series. In *ICDE*.

[35] Michele Linardi, Yan Zhu, Themis Palpanas, and Eamonn J. Keogh. 2020. Matrix Profile Goes MAD: Variable-Length Motif And Discord Discovery in Data Series. *DAMI*.

[36] Yury Malkov, Alexander Ponomarenko, Andrey Logvinov, and Vladimir Krylov. 2014. Approximate nearest neighbor algorithm based on navigable small world graphs. *Inf. Syst.* 45 (2014), 61–68.

[37] Yury A. Malkov and D. A. Yashunin. 2016. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. *CoRR* abs/1603.09320 (2016).

[38] OpenTSDB. 2015. OpenTSDB - A Distributed, Scalable Monitoring System (http://opentsdb.net/). http://opentsdb.net/

[39] Themis Palpanas. 2015. Data Series Management: The Road to Big Sequence Analytics. *SIGMOD Rec.* 44, 2 (2015), 47–52.

[40] Themis Palpanas. 2020. Evolution of a Data Series Index. *CCIS* 1197 (2020).

[41] Themis Palpanas and Volker Beckmann. 48(3), 2019. Report on the First and Second Interdisciplinary Time Series Analysis Workshop (ITISA). *SIGREC* (48(3), 2019).

[42] Tuomas Pelkonen, Scott Franklin, Paul Cavallaro, Qi Huang, Justin Meza, Justin Teller, and Kaushik Veeraraghavan. 2015. Gorilla: A Fast, Scalable, In-Memory Time Series Database. *PVLDB* 8, 12 (2015), 1816–1827.

[43] Botao Peng, Panagiota Fatourou, and Themis Palpanas. 2018. ParIS: The Next Destination for Fast Data Series Indexing and Query Answering.

[44] Botao Peng, Panagiota Fatourou, and Themis Palpanas. 2020. MESSI: In-Memory Data Series Indexing. In *ICDE*.

[45] Botao Peng, Panagiota Fatourou, and Themis Palpanas. 2020. ParIS+: Data Series Indexing on Multi-core Architectures. *TKDE* (2020).

[46] Botao Peng, Panagiota Fatourou, and Themis Palpanas. 2021. SING: Sequence Indexing Using GPUs. In *ICDE*.

[47] Danila Piatov, Sven Helmer, Anton Dignös, and Johann Gamper. 2019. Interactive and space-efficient multi-dimensional time series subsequence matching. *Inf. Syst.* 82 (2019), 121–135.

[48] Prometheus. 2018. Prometheus – Monitoring system & time series database. http://prometheus.io/

[49] QuasarDB. 2018. QuasarDB: high-performance, distributed, time series database. https://www.quasardb.net/

[50] Davood Rafiei and Alberto O. Mendelzon. 1997. Similarity-Based Queries for Time Series Data. In *SIGMOD*.

[51] Thanawin Rakthanmanon, Bilson J. L. Campana, Abdullah Mueen, Gustavo Batista, M. Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn J. Keogh. 2012. Searching and mining trillions of time series subsequences under dynamic time warping.. In *KDD*.

[52] Thanawin Rakthanmanon, Eamonn J Keogh, Stefano Lonardi, and Scott Evans. 2011. Time series epenthesis: Clustering time series streams requires ignoring some data. In *ICDM*.

[53] RiakTS. 2018. Riak TS – Basho Technologies. http://basho.com/products/riak-ts/

[54] Pedro Pereira Rodrigues, João Gama, and João Pedro Pedroso. 2006. ODAC: Hierarchical Clustering of Time Series Data Streams.. In *SDM*, Joydeep Ghosh, Diane Lambert, David B. Skillicorn, and Jaideep Srivastava (Eds.). SIAM, 499–503. http://dblp.uni-trier.de/db/conf/sdm/sdm2006.html#RodriguesGP06

[55] Patrick Schäfer and Mikael Högqvist. 2012. SFA: a symbolic fourier approximation and index for similarity search in high dimensional datasets. In *EDBT*.

[56] Jin Shieh and Eamonn J. Keogh. 2008. *i*SAX: indexing and mining terabyte sized time series. In *KDD*.

[57] Yifang Sun, Wei Wang, Jianbin Qin, Ying Zhang, and Xuemin Lin. 2014. SRS: Solving c-Approximate Nearest Neighbor Queries in High Dimensional Euclidean Space with a Tiny Index. *PVLDB* 8, 1 (2014), 1–12.

[58] Timely. 2018. Timely – A secure time series database based on Accumulo and Grafana. https://code.nsa.gov/timely/

[59] Timescale. 2018. Timescale - an open source time series management system. http://timescale.com/

[60] Cagatay Turkay, Nicola Pezzotti, Carsten Binnig, Hendrik Strobelt, Barbara Hammer, Daniel A. Keim, Jean-Daniel Fekete, Themis Palpanas, Yunhai Wang, and Florin Rusu. 2018. Progressive Data Science: Potential and Challenges. *CoRR* abs/1812.08032 (2018). arXiv:1812.08032 http://arxiv.org/abs/1812.08032

[61] Yang Wang, Peng Wang, Jian Pei, Wei Wang, and Sheng Huang. 2013. A Data-adaptive and Dynamic Segmentation Index for Whole Matching on Time Series. *PVLDB* 6, 10 (2013), 793–804.

[62] Warp10. 2018. Warp 10 – The Most Advanced Time Series Platform. https://www.warp10.io/

[63] T. Warren Liao. 2005. Clustering of time series data—a survey. *Pattern Recognition* 38, 11 (2005), 1857–1874.

[64] Jiaye Wu, Peng Wang, Ningting Pan, Chen Wang, Wei Wang, and Jianmin Wang. 2019. KV-Match: A Subsequence Matching Approach Supporting Normalization and Time Warping. In *ICDE*.

[65] D. E. Yagoubi, R. Akbarinia, F. Masseglia, and T. Palpanas. 2017. DPiSAX: Massively Distributed Partitioned iSAX. In *ICDM*. 1135–1140.

[66] Djamel-Edine Yagoubi, Reza Akbarinia, Florent Masseglia, and Themis Palpanas. 2020. Massively Distributed Time Series Indexing and Querying. *TKDE* 32, 1 (2020).

[67] Liang Zhang, Noura Alghamdi, Mohamed Y Eltabakh, and Elke A Rundensteiner. 2019. TARDIS: Distributed indexing framework for big time series data. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 1202–1213.

[68] Kostas Zoumpatianos, Stratos Idreos, and Themis Palpanas. 2016. ADS: the adaptive data series index. *VLDB J.* 25, 6 (2016), 843–866.

[69] Kostas Zoumpatianos, Yin Lou, Ioana Ileana, Themis Palpanas, and Johannes Gehrke. 2018. Generating data series query workloads. *VLDB J.* 27, 6 (2018).

[70] Kostas Zoumpatianos, Yin Lou, Themis Palpanas, and Johannes Gehrke. 2015. Query Workloads for Data Series Indexes. In *KDD*.

Karima Echihabi is an Assistant Professor at Mohammed VI Polytechnic University (UM6P) in Morocco. She is interested in scalable data analytics and data series management and has performed an extensive analysis of data series indexes. She holds a PhD degree from Mohammed V University (Morocco) and the University of Paris (France) and a Masters Degree in Computer Science from the University of Toronto. She has worked as a software engineer in the Windows team at Microsoft, Redmond (USA), and the Query Optimizer team at the IBM Toronto Lab (Canada).

Kostas Zoumpatianos is a Software Engineer at Snowflake Computing. He has been a Marie Curie Fellow at the University of Paris and a postdoctoral researcher at Harvard University. He got his PhD from the University of Trento in topics related to indexing and managing large collections of data series. He also holds a M.Sc. in Information Management and a Dipl.Eng. in Information and Communication Systems Engineering from the University of the Aegean in Greece.

Themis Palpanas is Senior Member of the French University Institute (IUF), a distinction that recognizes excellence across all academic disciplines, and professor of computer science at the University of Paris (France), where he is director of the Data Intelligence Institute of Paris (diiP), and director of the data management group, diNo. He received the BS degree from the National Technical University of Athens, Greece, and the MSc and PhD degrees from the University of Toronto, Canada. His interests include problems related to data science (big data analytics and machine learning applications). He is the author of 9 US patents and 2 French patents. He is the recipient of 3 Best Paper awards, and the IBM Shared University Research (SUR) Award. He is currently serving on the VLDB Endowment Board of Trustees, and as an Editor in Chief for the BDR Journal. He has served as General Chair for VLDB 2013, and in the program committees of all major conferences in the areas of data management and analysis.