Introduction to Axispot™, Real-time Spatio-temporal Data-management System, and Its High-speed Spatio-temporal Data-search Technology

Masayuki Hanadate, Tatsuro Kimura, Nobuhiro Oki, Naoko Shigematsu, Isoo Ueno, Ichibe Naito, Takashi Kubo, Kazuhiro Miyahara, and Atsushi Isomura

Abstract

We introduce the Axispot™ real-time spatio-temporal data-management system, which is a key component in responding to the demands for next-generation services such as communication between connected vehicles and augmented reality. We also describe a high-speed spatio-temporal data-search technology as a key function of Axispot, which can not only accumulate a large amount of data sent all at once from moving things (MTs), such as people or automobiles, but also search for MTs in a particular area and at a certain time from a large amount of data stored in a database in real time.

Keywords: spatio-temporal database, dynamic map, augmented reality, Digital Twin Computing

1. Introduction

The Internet of Things (IoT), a key technology for cloud-based, centralized management of various information sensed from people, things, and natural environments in real space, is becoming increasingly indispensable for next-generation services for moving things (MTs) such as people and automobiles. For example, with inter-vehicle communication services, large numbers of vehicles connected to the Internet (connected vehicles) continuously send information on their driving location and the time of data transmission to the cloud, which stores and analyzes this information, and notifies vehicles of traffic conditions (e.g., traffic accidents and congestion) in the relevant areas in real time. Another example is in augmented reality, in which various information transmitted from smartphones and wearable devices is stored together with the user's location and the time of data transmission in the cloud to enable quick retrieval and delivery of useful information to the user corresponding to his/her location and interests to help him/her decide what to do next (e.g., information about recommendations or shop congestion).

To respond to such future mobility demands, NTT Software Innovation Center (SIC) is developing Axispot™—a real-time spatio-temporal data-management system—to accumulate a massive amount of MT information and enable real-time searching of MTs in particular areas and at specific times from this large amount of MT information stored in the database.

In this article, we give an overview of Axispot. We first briefly describe a conventional spatio-temporal database (STDB) used with Axispot and the high-speed spatio-temporal data-search technology [1] we developed to solve the technical issues with a conventional STDB. We also give an overview of the Axispot architecture based on this high-speed spatio-temporal data-search technology.

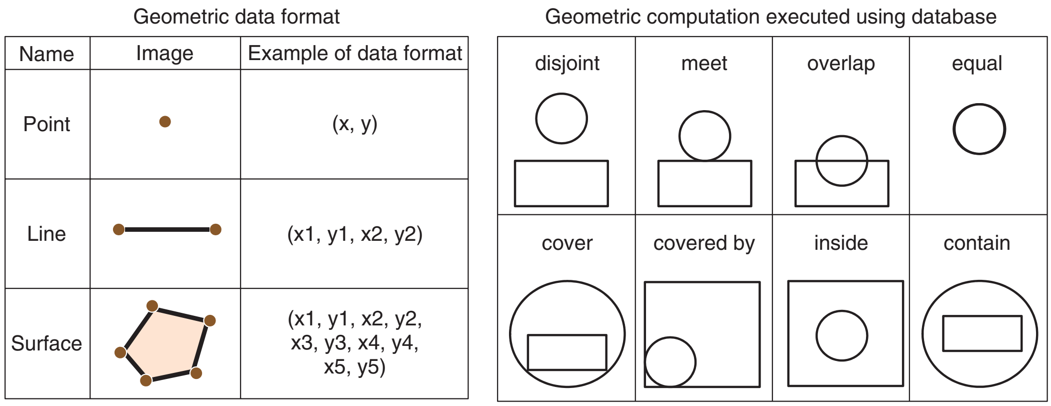


Fig. 1. Examples of data formats in spatial databases and geometric computation.

2. Current state of STDBs

An STDB is used for efficiently storing and retrieving data sets containing both spatial information, that is, information for position in space (e.g., longitude and latitude), and time information (e.g., time of day) [2, 3]. On the other hand, a database for only handling either temporal information or spatial information is known as a time-series database or spatial database, respectively. An STDB is generally implemented by expanding a spatial database to also manage temporal information [3]. Therefore, before we discuss an STDB, we briefly describe spatial-database technology.

A spatial database stores spatial information about geographic areas (e.g., land or buildings) that is represented in a geometric data format, such as point, line, or surface, and enables retrieval of some spatial information corresponding to a query also expressed as spatial information. To retrieve spatial information, the spatial database executes geometric computations on the stored spatial information and the query. These data formats and geometric computations are illustrated in **Fig. 1** [2]. For example, to search for buildings in a particular area, information on the area is represented as a surface, which is composed of data sets of points stored in an STDB that represent the longitude and latitude of the area's boundary. When the STDB receives a query, which is spatial information on a particular area that the user wants to search, and is represented as a surface, the STDB calculates an inclusion relation between the query and area including the buildings stored in the STDB. The STDB then returns the retrieved areas, which are the areas included in the query.

A spatial database can be implemented using a relational database (RDB) or distributed key-value store (KVS). An RDB contains a table with columns assigned to spatial information. However, if spatial information is multi-dimensional and the columns are independently prepared for each element of the spatial information (e.g., latitude and longitude), then it is necessary to search each column independently. Processing to search multiple individual columns degrades search efficiency. To address this technical issue, search trees for two-dimensional information (R-Tree [4], etc.) have been proposed.

With a distributed KVS, however, one column, *key*, is initially assigned in a key-value structure table, which plays an important role in high-performance searching. While this architecture is extremely simple and advantageous for search performance, there is only one data point assigned to the key. It is thus difficult to store multi-dimensional information, such as spatial information, without first converting it to single-dimensional data. Therefore, a space-filling curve was proposed as a conversion technique that can be used as the key in a distributed KVS. Notably, the geohash [5] is a data-conversion rule using a Z-curve, a type of space-filling curve, and is one of the most well-known techniques to convert multi-dimensional spatial information into single-dimensional data. This is because the geohash enables important spatial information operation—the zoom effect to expand or contract an area—done by changing the length of the converted single-dimensional spatial information to express the area (**Fig. 2**). As explained below, single-dimensional spatial information converted using the geohash is called a spatial code.

As described above, an STDB can be created by

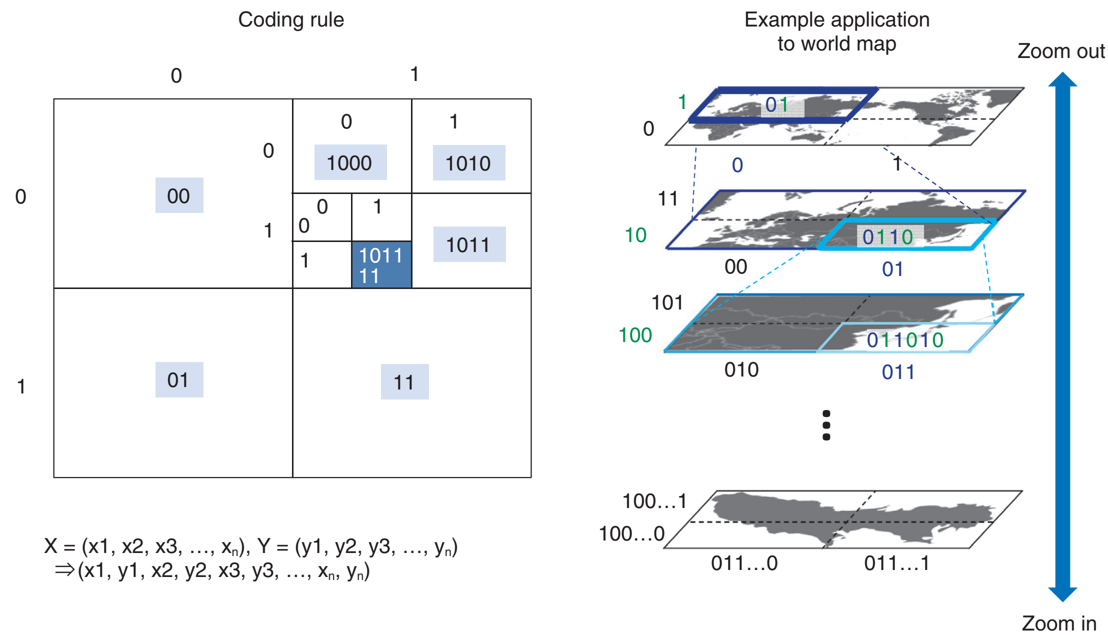


Fig. 2. The geohash.

expanding a spatial database using an RDB or distributed KVS to manage temporal information. The future mobility requirement described in the first section involves the ability to handle continuous movements of a huge number of MTs, e.g., tens of millions of people or connected vehicles, that frequently change their positions and data transmission times. With an RDB, it would be necessary to update the structures of the search trees every time the RDB receives data, and such frequent updates would reduce the efficiency of writing the spatio-temporal information into the RDB. Therefore, a distributed KVS is a better choice for an STDB to manage a large number of MTs in real time because updating the RDB tree structure is not necessary. A distributed KVS is scalable, an even more advantageous feature for seamlessly storing a massive amount of data into multiple distributed data nodes. Consequently, we implemented an STDB with a distributed KVS.

In the following section, we describe high-speed spatio-temporal data-search technology developed by SIC using a distributed KVS and Z-curve and the fundamental STDB functions.

3. High-speed spatio-temporal data-search technology

High-speed spatio-temporal data-search technology is used to store sensor information received all at once from a large number of MTs in real space as well as spatial and temporal information associated with the sensor information, and simultaneously retrieve sensor information contained in a particular rectangular area and at a certain time given as a query. In particular, by applying the spatio-temporal code and limited node-distribution algorithm to a distributed KVS, the high-speed spatio-temporal data-search technology satisfies the following requirements:

(1) Efficient multi-dimensional information search: Using spatio-temporal code as the distributed KVS key makes it possible to simultaneously search multi-dimensional information—data sets consisting of time, latitude, longitude, and altitude.

(2) Adjustments to the search area: A spatio-temporal code prefix search enables changing the area and time to search—for example, one hour before the current time, longitude 10 to 20 degrees east and latitude 30 to 40 degrees north.

(3) Prevention of intensive access to particular nodes: The limited node-distribution algorithm can distribute information across all nodes that comprise the distributed KVS, which prevents intensive access to particular nodes that

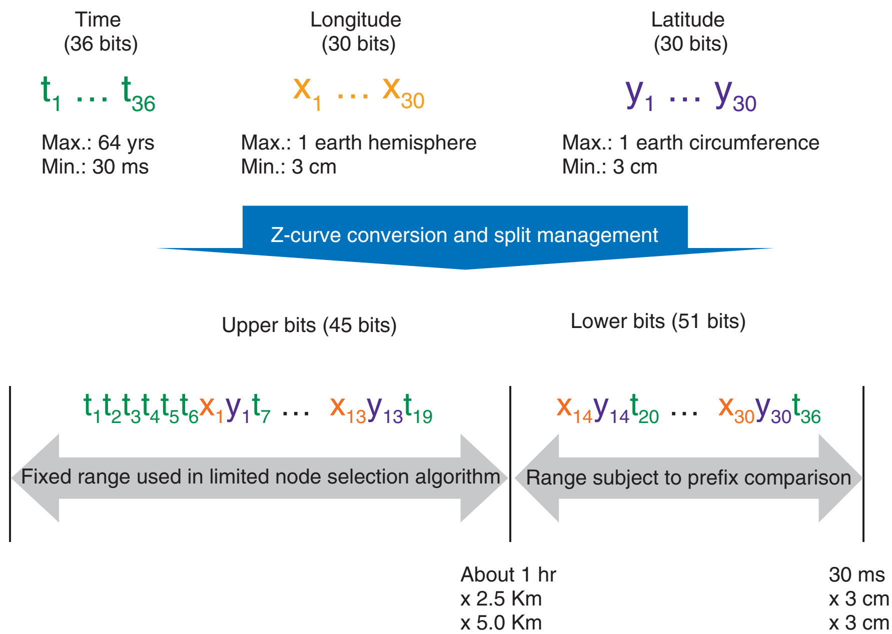


Fig. 3. Example of spatio-temporal code.

could occur with fluctuations in MT traffic in the real world (e.g., the urban area around a station is crowded in the morning, and the suburbs are crowded in the evening).

(4) Prevention of searching all nodes: If information is stored at random across the storage nodes of the distributed KVS, then it is difficult to identify the nodes in which the information related to the query is stored, so an STDB has to access all storage nodes including those with no related data when the STDB is searched for data. This results in inefficient data search. To solve this problem, the limited node-distribution algorithm identifies only the nodes that store information related to the query, so that the STDB only has to retrieve information from them, avoiding unnecessary node access.

3.1 Spatio-temporal code

Spatio-temporal code is information that expands on the aforementioned spatial code to include a time range and single-dimensional information with spatial and temporal information bits rearranged according to conversion rules using a Z-curve. **Figure 3** illustrates an example of spatio-temporal code, which consists of 36 bits for time, 30 bits for longitude, and 30 bits for latitude. In this design, the minimum rectangle size of this code is 30 ms × 3 cm × 3 cm.

We now describe the procedure for storing and searching data using the spatio-temporal code with the limited node-distribution algorithm (**Fig. 4**).

First, a spatio-temporal code is generated using the temporal (time) and spatial (longitude and latitude) information received from the client. Second, a hash computation on the fixed upper bits of the spatio-temporal code generates a hash-value that corresponds to a unique combination of nodes comprising the distributed KVS as candidate nodes to store data. Finally, one node is randomly selected from among these candidates.

A spatio-temporal code is first generated from the search query that includes the temporal and spatial information received from the client; then a hash computation on the fixed upper bits of the spatio-temporal code generates a hash-value to identify candidate storage nodes. The candidate storage nodes (not all the storage nodes) then execute the prefix match of the spatio-temporal code of the search query with their stored spatio-temporal codes. Finally, each candidate node returns its matching data to the client.

3.2 Limited node-distribution algorithm

With a spatio-temporal code on a particular area in the limited node-distribution algorithm, the fixed upper bits of the spatio-temporal code change so that

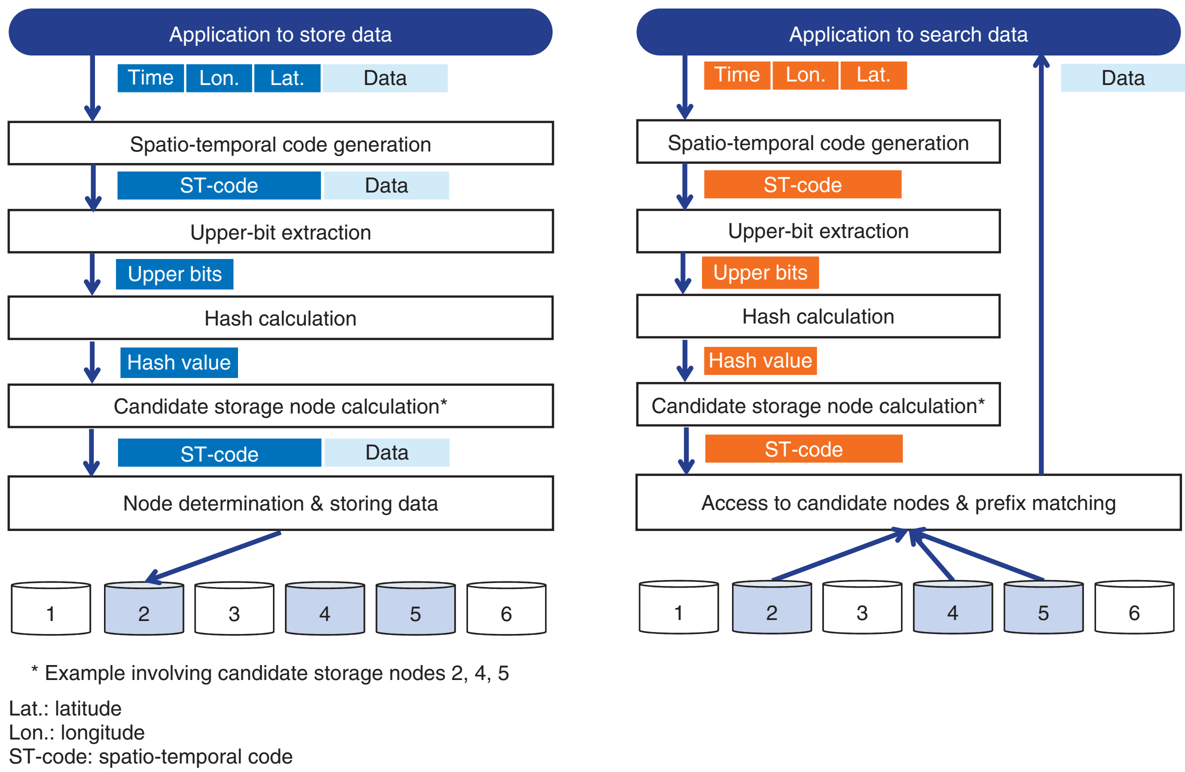


Fig. 4. Procedure of data storing and data search.

the combinations of candidate storage nodes also continuously change. For example, even when traffic congestion occurs in a certain area, the combinations of candidate storage nodes also switch over time. As a result, continuous intensive access to particular nodes during traffic congestion is avoided. An illustration of the transition of candidate-node combinations is shown in **Fig. 5**. With data search, searching is only executed on the candidate storage nodes; hence, the limited node-distribution algorithm reduces the overall workload to search only the desired spatio-temporal data from the large amount of stored data.

Particular information related to a certain area and time can also be searched instantly by comparing the prefixes of the single-dimensional spatio-temporal code stored in the database with those of the spatio-temporal code given with the search query.

Moreover, changing the length of the spatio-temporal code in the search query enables applications to adjust the width and length of the rectangle area and the search time. For example, to search a wider area or a longer period of time, a shorter spatio-temporal code in the query can be used. Conversely, to search a narrower area or a shorter period of time, a longer spatio-temporal code can be used in the query.

Through our implementation and evaluation of the limited node-distribution algorithm, we confirmed that its throughput for storing data is 13 times better than conventional algorithms, and its throughput for searching data is 5 times better than conventional algorithms [1, 6].

4. Overview of Axispot architecture

With the high-speed spatio-temporal data-search technology, we aim to further advance spatio-temporal data-management functions, such as searching complicated, non-rectangular areas (e.g., roads and building areas), that will contribute to next-generation services such as inter-vehicle communication and augmented reality.

Figure 6 shows the overall Axispot architecture. Axispot consists of the following five layers: database, database management, geomesh, geometric search, and geometric analysis. The database layer consists of a distributed KVS consisting of multiple nodes to manage data. In the database-management layer, the

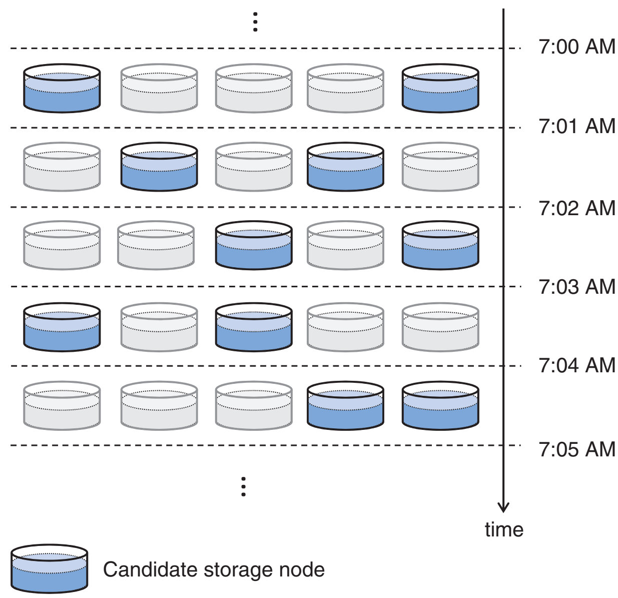


Fig. 5.   Transition of candidate storage-node combinations.

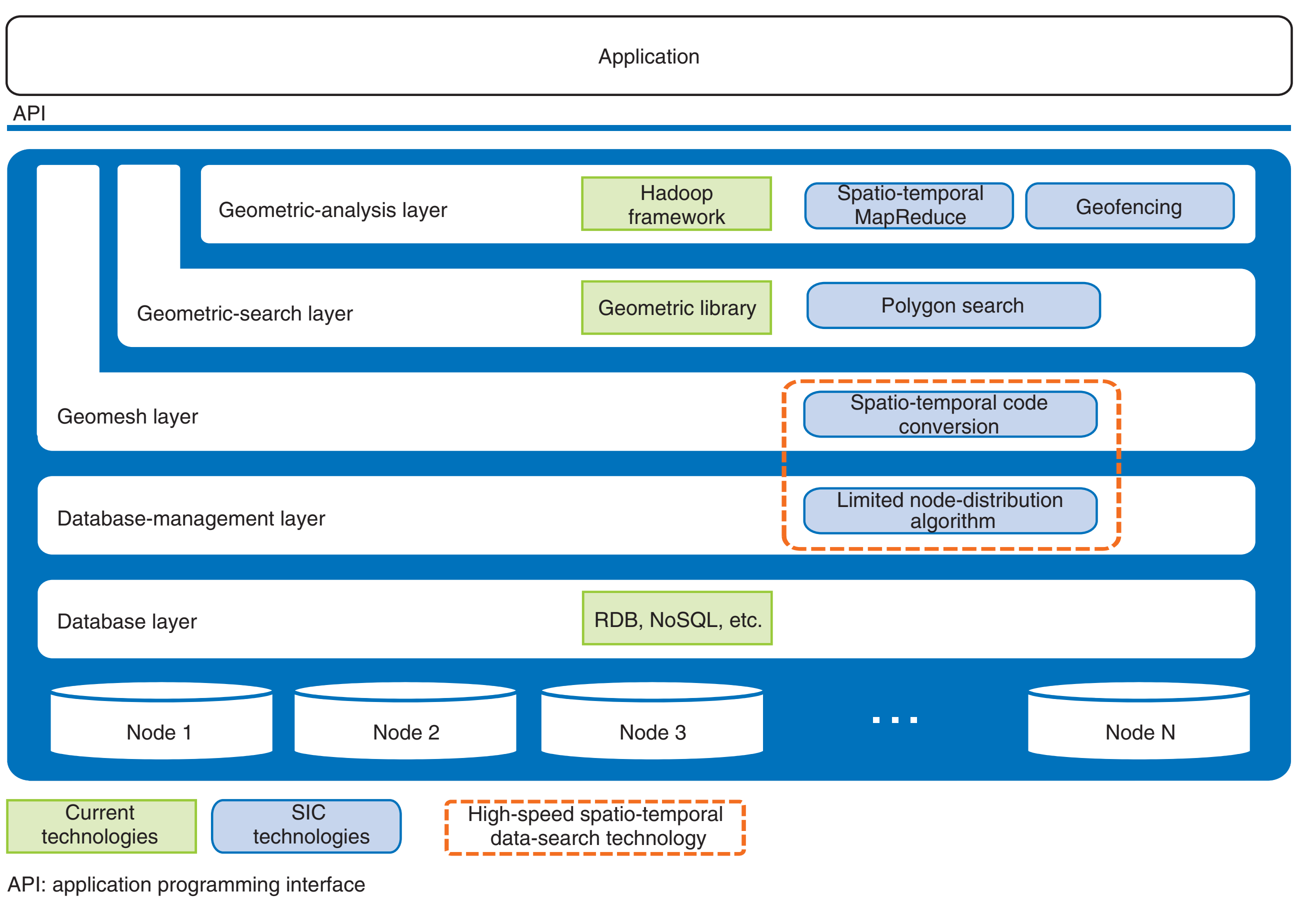


Fig. 6.   Axispot architecture.

nodes to store and search data are determined with the high-speed spatio-temporal data-search technology. In the geomesh layer, a spatio-temporal code is generated using a Z-curve. The shape of the search area in this layer is always a fixed-length rectangle defined by the spatio-temporal code (e.g., a 100-km square). Then, the geometric-search layer extracts MT data sets for still more complicated non-rectangular areas from the search result (the rectangular area) in the geomesh layer. For example, this layer only identifies MTs in complicated polygonal, non-rectangle areas such as roads, parks, or school districts from all the MT data sets for the particular rectangular area. Finally, the geometric-analysis layer enables spatio-temporal analysis using the output information from the geomesh and geometric-search layers. For example, this layer combines search results in the geomesh layer with MapReduce technology [7] to efficiently calculate geometric statistic information for distribution of a massive amount of MTs in a particular area and at a certain time. This also enables geofencing, detecting whether an MT enters or exits a specific area, called a fence.

Furthermore, we assume that Axispot can also be applied not only to the real world but also to cyberspace; Axispot makes it possible to put MTs, such as digital humans and virtual automobiles corresponding to a specific time and location managed in Axispot, into virtual cities and virtual natural environments in cyberspace. We will develop this technology as a key component for synthesizing digital twins dynamically, hence, contributing to Digital Twin Computing [8].

References

[1] A. Isomura, "Real-time Spatiotemporal Data Utilization for Future Mobility Services," RedisConf19, San Francisco, USA, June 2019.

[2] T. Abraham and J. F. Roddick, "Survey of Spatio-temporal Databases," GeoInformatica, Vol. 3, No. 1, pp. 61–99, 1999.

[3] N. Pant, M. Fouladgar, R. Elmasri, and K. Jitkajornwanich, "A Survey of Spatio-temporal Database Research," Intelligent Information and Database Systems, LNCS, Vol. 10752, 2018.

[4] M. Hadjieleftheriou, Y. Manolopoulos, Y. Theodoridis, and V. J. Tsotras, "R-Trees: A Dynamic Index Structure for Spatial Searching," Encyclopedia of GIS, pp. 47–57, 2017.

[5] Labix Blog by G. Niemeyer, https://blog.labix.org/2008/02/26/geohashorg-is-public

[6] D. Hochman, "Geospatial Indexing at Scale: The 15 Million QPS Redis Architecture Powering Lyft," June 2017.

[7] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Proc. of the 6th Conference on Symposium on Operating Systems Design & Implementation (OSDI 2004), pp. 137–150, San Francisco, USA, Dec. 2004.

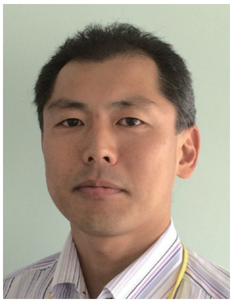
[8] Press release issued by NTT, "NTT proposes the 'Digital Twin Computing Initiative' – a platform to combine high-precision digital information reflecting the real world to synthesize diverse virtual worlds, generate novel services and bring about society of the future," June 10, 2019.
https://www.ntt.co.jp/news2019/1906e/190610a.html

Masayuki Hanadate
Senior Research Engineer, Supervisor, IoT Framework SE Project, NTT Software Innovation Center.

He received a B.E. in communications engineering from Tohoku University, Miyagi, in 1997. Since joining NTT in 1997, he has been mainly engaged in software development of digital money systems, distributed storage systems, and distributed database systems at NTT laboratories. He also worked at NTT DATA as manager for development of open source distributed storage systems from 2010 to 2014.

Tatsuro Kimura
Senior Researcher, IoT Framework SE Project, NTT Software Innovation Center.

He received an M.S. in agricultural and life sciences from the University of Tokyo in 2001. He joined NTT Information Platform Laboratories in 2001. He engaged in research and development (R&D) of Internet live streaming, content delivery networks, a network address translation box for SIP-ALG (Session Initiation Protocol - application layer gateway), and line authentication in NTT's Next Generation Network (NGN). In 2008, He moved to NTT Communications, where he designed Internet protocol version 4 (IPv4) to IPv6 migration for enterprise networks. In 2014, he moved to NTT Software Innovation Center and engaged in R&D of bare metal server provisioning in OpenStack and the CI (continuous integration) framework for network devices. He is currently working on Axispot.



Nobuhiro Oki

Senior Research Engineer, IoT Framework SE Project, NTT Software Innovation Center.

He received a B.S. and M.S. in chemistry from Tokyo Institute of Technology in 1994 and 1996. He joined NTT in 1996 and launched the world's first Internet telegram system (D-Mail). He enrolled in NTT EAST in 1999 and worked with NTT Access Network Service Systems Laboratories to develop digital television transmission systems using gigabit Ethernet passive optical networks (GE-PONs) and verified the application of power line communication to home gateway equipment. In 2008, he joined NTT Communications and launched various cloud services (Biz Hosting Basic, Biz Simple Disk, and Enterprise Cloud) where he worked with NTT Open Source Software Center and NTT Software Innovation Center. In 2018, he joined NTT Software Innovation Center, where he developed a container application distribution management platform (IoT-MANO) and is currently working on Axispot.

Naoko Shigematsu

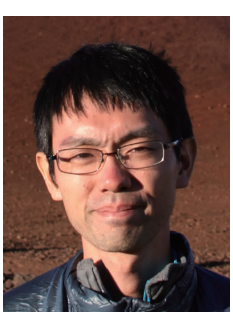
Research Engineer, IoT Framework SE Project, NTT Software Innovation Center.

She received a B.S. and M.S. in geophysics from Tohoku University, Miyagi, in 1993 and 1995. She joined NTT Telecommunication Networks Laboratories in 1995 and moved to NTT EAST R&D Center in 1999, where she engaged in research on asynchronous transfer mode network operation systems. In 2000, she moved to NTT Information Sharing Platform Laboratories, where she engaged in research on storage area networks. She joined NTT Software Innovation Center in 2012 and is working on Axispot.

Isoo Ueno

Research Engineer, IoT Framework SE Project, NTT Software Innovation Center.

He received a B.S. and M.S. in earth science from Kobe University, Hyogo, in 1990 and 1992. He joined NTT in 1992 and engaged in basic research on artificial life and complex systems. In 1997, he moved to the Global Business department, where he launched several global services, e.g., local Internet service providers in Hong Kong and UK, and OCN mail & web services. From 1999 to 2014, he worked for NTT Communications and was involved in several development projects regarding Windows hosting, Biz authentication, and global billing, where he also was engaged in operations and customer services. In 2014, he joined NTT Software Innovation Center and is currently working on Axispot.



Ichibe Naito

Senior Research Engineer, IoT Framework SE Project, NTT Software Innovation Center.

He received a B.S. and M.S. in information science from Waseda University, Tokyo, in 2006. Since joining NTT in 2006, he has been engaged in R&D of a distributed data stream management system and personal information management system. In 2010, he moved to NTT Communications, where he designed the database of the operation support system and developed an infrastructure as a service (Iaas) service. In 2013, he joined NTT Software Innovation Center and developed an operation support system for distributed object storage. His current research includes efficient management of huge amounts of geospatial data.

Takashi Kubo

Research Engineer, IoT Framework SE Project, NTT Software Innovation Center.

He received an M.S. in information science and technology from Osaka University in 2009. He joined NTT WEST in 2009 and developed call control servers for NGN. In 2016, he moved to NTT Software Innovation Center and engaged in support for using distributed object storage. His current research is on distributed spatio-temporal database-management technology.

Kazuhiro Miyahara

Researcher, IoT Framework SE Project, NTT Software Innovation Center.

He received a B.S. in mathematics and M.S. in information science and technology from Waseda University, Tokyo, in 2012 and 2014. He joined NTT Software Innovation Center in 2014. He has been engaged in R&D of distributed object storage software (OpenStack Swift). He is now working on Axispot and LASOLV™, a computer that uses a pulse laser beam as a tool to solve challenging mathematical problems. He also developed functions that enable geometrical search and flexible search in research related to Axispot. For his LASOLV research, he constructs mathematical models to solve practical problems.

Atsushi Isomura

Researcher, IoT Framework SE Project, NTT Software Innovation Center.

He received a B.S. and M.S. in information science and technology from Aichi Prefectural University in 2014 and 2016. He joined NTT Software Innovation Center in 2016 and is working on Axispot. In 2019, he participated in RedisConf19 in San Francisco as a presenter and introduced the core technologies of Axispot. He is a member of the Institute of Electronics, Information and Communication Engineers (IEICE). He received the IEICE Tokai Section Student Award in 2015 and the best poster award at the 12th International Conference on Ubiquitous Healthcare.