GeoSpock

Spatial big data platform: everything you need to know before you buy

SPATIAL BIG DATA

CHOOSING THE RIGHT DATABASE IS KEY TO YOUR SUCCESS.

It is important to get the decision right the first time. Having to change once a product has been built will be costly, and at times an impossible process.

Big Data poses a number of challenges with the three biggest being:

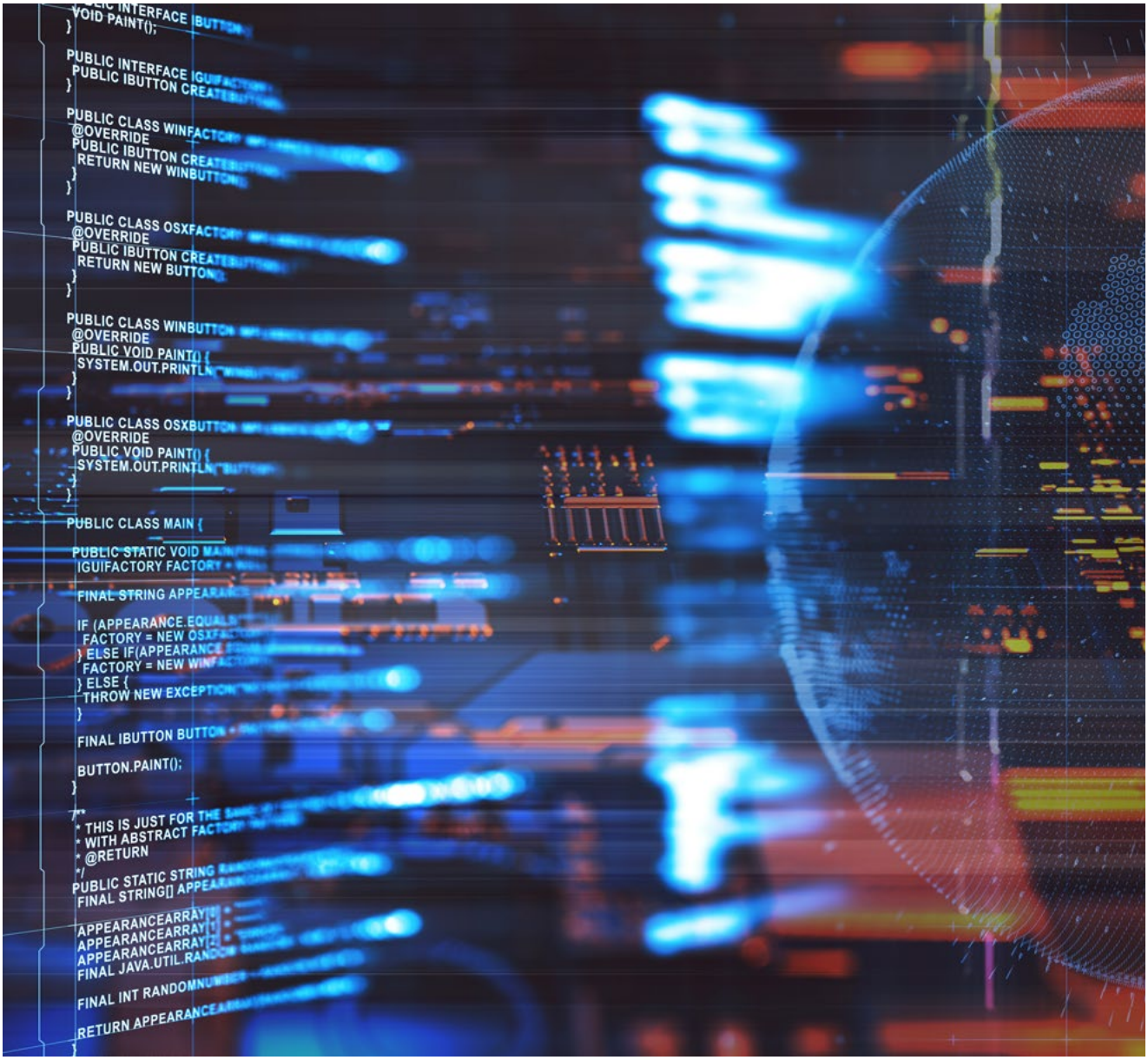
Scale: How to continually add more data to your system

Throughput: How to handle rapidly changing data

Responsiveness: How to keep your systems working in real-time

Current database systems were created to solve problems from the previous decades; they optimized performance to solve a number of these challenges, at the cost of not addressing other equally important challenges.

Presently, machines are generating data at a rate and scale beyond current imagination and databases are starting to fail. Therefore a new type of database is needed.



The GeoSpock solution

Our solution is a NoSQL database designed for ease of use which is able to maintain consistent performance at extreme scales. Among other technologies, GeoSpock uses a Wide-Column database alongside Key-Value stores and has the ability to expose a Document storage system to the user, which provides a large amount of flexibility.

The system is an efficient multi-dimensional database, aimed to handle extreme scale datasets maintaining high performance. When accessing the information store it goes beyond 2D and 3D geospatial information, thanks to its unmatched design. Additionally it is able to handle otherwise computationally heavy multidimensional queries, such as K-nearest neighbour searches, extremely efficiently.

Our current solution, offers a Database-as-a-Service on the cloud. It is able to seamlessly handle the dynamic sharding of data as well as automatic load balancing, which keeps the system responsive even under heavy use and makes it exceptionally easy to get up and running quickly.

GeoSpock is perfectly suited for data intensive applications and also real-time systems, such as DataTech, logistics, IoT and telecommunication applications.

Optimised performance

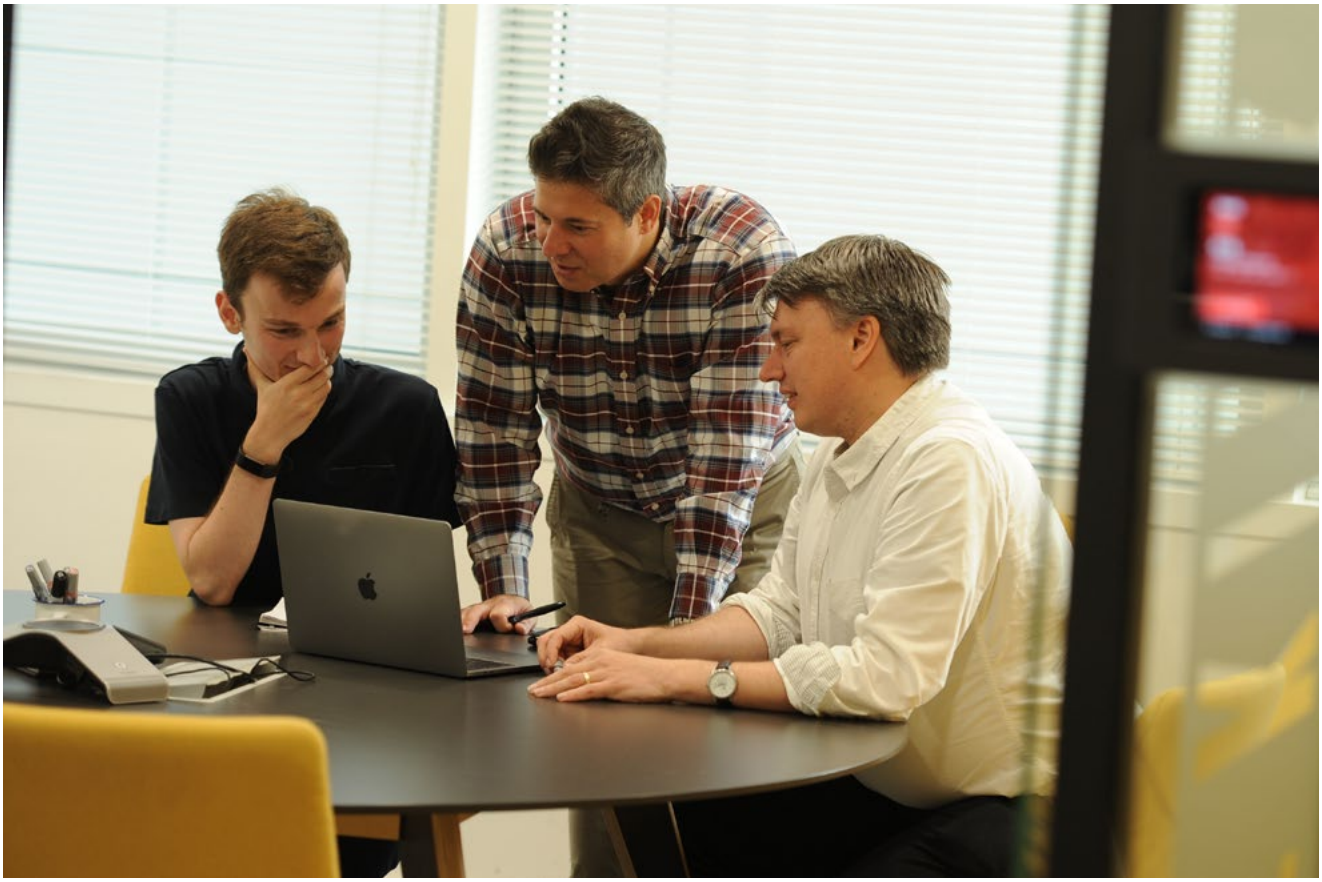
Typically as more data and more machines are added to existing NoSQL solutions the slower the queries become. However, we have developed new techniques for maintaining consistent real-time response times, even when the number of machines in a cluster reaches into the thousands and beyond.

- The “Von Neumann Bottleneck”: It is common to observe very powerful CPUs starving because the memory bus cannot cope with the volume of data needed by the CPUs, and so a limit in performance is reached.

  We have solved this by improving how information is accessed with distributed indexing mechanisms that makes accessing data more efficient.

- Other NoSQL – and SQL – solutions encounter a performance bottleneck because their indexing mechanisms require either read/write locks, or repeatable-read locks; that is, keeping enough data in a state until it can be reliably read and reported on.

  Our solution solves this issue, by using distributed indexing mechanisms that avoid those readers-writer locks. This allows us to achieve extremely high write-throughput and **is able to operate both reads and writes concurrently**, avoiding hidden costs while



keeping flexibility in the queries. Essential for IoT environments.

- In searching for performance and the ability to provide quick query response times, many solutions leverage the technique of pre-aggregating results of queries and reports they expect their clients will ask of the data. The pre-processing and aggregation of data is required to overcome shortcomings of the solution from both a scale and query response prospective hence reducing flexibility of the system and speed to insight. However, questions and queries performed in a system of this kind are unique due to changing business needs, therefore the aggregation of answers and pre-calculation of queries need to be re-done constantly, impacting the associated cost and downservice.

  We store all the data in its original state, allowing access to all ingested data and keeping response times to industry leading standards regardless of scale. **Our solution answers do not need to be pre-calculated** or stored in the system, they are answered as requested in the database and can be changed as required.

- Another important factor to consider is the cost of the platform hosting the solution. Traditional databases, as well as some newer approaches such as the CPU-based solutions, require expensive and typically specially designed hardware to be able to cope with performance requirements. New NoSQL systems try to address this by using farms of inexpensive systems, but although they can be inexpensive low level hardware systems – or their cloud incarnations – to keep up with the required performance expectations it needs an enormous number of servers that finally will increase the overall operational cost well beyond what was initially estimated.

  We have found a clever solution to this issue by being able to reduce – or increase – the size of the cluster as is needed. In this sense, the most CPU consuming activity is tied to the initial ingestion of information, where huge amounts of historical information is acquired from customers, indexed and stored in our internal database system. Once this initial ingestion is finished, the computing power needed for subsequent ingestion of smaller quantities of information is much lower than the initial ingest, therefore **most of the cluster can be decommissioned and the cost of running the hardware supporting the system is much lower** than in other comparable platforms.

Database comparison table

DATABASE	GEOSPOCK	RDBMS SYSTEMS (traditional databases)	KEY-VALUE	WIDE-COLUMN	DOCUMENT ORIENTED
HARDWARE NEEDED	• Commodity • Inexpensive • Elastic growth	• Non-commodity • Expensive • Difficult growth	Commodity / inexpensive	Commodity / inexpensive	Commodity / inexpensive
OLAP	Yes	No (limited capacity)	Yes	Yes	No (not well suited by design for this task)
OLTP	No	Yes	No (only batch processing)	No	Yes (with limitations on functionality)
DATA VOLUME	• Huge volumes, in the order of 100s of TB. • Not limit in theory	• Limited to several TB maximum • Needs the use of additional external repositories to guarantee growth	Easy to scale-out, in the order of hundreds of TBs	Huge volumes, in the order of 100s of TB	High, hundreds of TBs (but not recommended for drop in performance)
RESPONSE TIMES	• Constant • Faster than BigQuery in many cases	• Degrade as the size of the system grows • Typically several minutes to several hours, depending on the complexity of the query and the volume to analyse	Very high when restricted to one dimension (unique index). Otherwise a full scan may be required and lead to a very poor performance.	Can increase exponentially as the size or complexity of the query increases (aimes to minutes, hours, days…)	Fast, if indices can be kept in memory (reducing thus effective volume of DB). Otherwise, can be very slow.

DATABASE	GEOSPOCK	RDBMS SYSTEMS (traditional databases)	KEY-VALUE	WIDE-COLUMN	DOCUMENT ORIENTED
QUERY RICHNESS / FUNCTIONALITY	• SQL supported • APIs for data ingestion and results in the roadmap	• Very complete and flexible • APIs usually in SQL and other languages to increase flexibility	• Very poor functionality • Only simple queries	• Rich functionality • Quite flexible in terms of queries you can perform	• Relative simple queries only • Flexible in other storage capabilities.
EASE OF MANAGEMENT	• Fully managed service • SaaS in the roadmap	Usually complex management to keep up performance and maintain regular installations (even in cloud deployments)	• Medium, as keeping index uniqueness can lead to complex management • Other queries require additional indexes and increased complex management	Quite complex management, especially as the volume of information increases and the complexity of the queries increases	Can be challenging if additional query complexity is required by maintaining additional indices and infrastructure
CLOUD DEPLOYMENT	Native	Not native (in most cases possible, as a regular software installation)	Native in many products	Yes, easily suited for cloud developments	Native in most cases
ON-PREMISE DEPLOYMENT	In roadmap	Yes (in most cases)	Possible/Native in some products	Medium Suitable for on-premises, but may increase cost/complexity significantly	Native/easy in most cases
GEOGRAPHICAL SUPPORT	Yes (native support)	• Yes (non-native, limited support in many cases). • Even GIS systems have been developed taking an RDBMS as a starting point, meaning geo support is added at a later stage.	Not explicit	Non native / complex to add.	Not native
TEMPORAL SUPPORT (Time as a dimension)	Yes (native support)	• Yes (non-native, limited support) • Although most systems can store time as any other data, it is not treated in a multidimensional way to speed up queries based on this field or show evolution of information along time or temporal series comparison, etc	Not explicit	Not native (just another field of information)	Not native, just another column / dimension

About GeoSpock

GeoSpock brings sensor data to life – translating complex connections into meaningful visualisations that reveal the bigger picture. Its state-of-the-art spatial big data platform has the power to transform lives and businesses – whether it's cutting harmful emissions by reducing traffic congestion or maximising profitability by optimising commercial operations. In just seconds, GeoSpock harnesses trillions of data points to discover hidden patterns and create a valuable new perspective in markets ranging from maritime and logistics to smart cities and data technology.

www.geospock.com

GeoSpock

CAMBRIDGE | LONDON | SINGAPORE | TOKYO