

Made in Group: AI project

Database and Hosting Options

Introduction

The purpose of this report is to provide an overview of the database management systems (DBMS) available, the use cases that they are designed for, as well as a summary of their respective advantages and disadvantages relative to our project.

The report will be organised as follows. First a broad outline of the needs of the project will be given. Then an overview of a number of DBMS alternatives (so called, NoSQL databases) will be provided -- comparing and contrasting with traditional Relational DBMS (RDBMS), as well as highlighting their relative strengths compared with each other. Next, we will provide a recommendation for a (NoSQL) DBMS for the project, to use alongside the RDBMS system that Made In Group (MIG) is currently using. Finally, a number of hosting options will be provided, as well as an estimate for their associated costs.

AI Project Requirements

The AI project will entail building a module to run in parallel to the existing system serving MIG's current online platform. The purpose of the module will be to use Artificial Intelligence (AI) techniques -- specifically Machine Learning (ML) -- to provide recommendation services to the users of the platform, as well as team members at MIG. The overall scope covers the following key features:

- Member matching (recommending business to other businesses within a business based on characteristics such as sector, commerce, and location). This will also recommend users based on the member organisation that they work for.
- Member grouping / clustering / segmentation (grouping members based on similarity). This can also group users according to the member organisation that they work for and will be useful for grouping users into breakout rooms.
- Targeted advertising (displaying the most relevant adverts to `freemium` members -- customers who do not pay for membership -- based on characteristics of the member posting the advert, as well as the freemium member).
- Sales prospecting (ranking non-member organisations based on their relevance to existing members, and vice-versa). This will involve capturing some aspects about those non-member organisations, and likely developing an `incomplete` picture of them in order to recommend them.

After examining the current system, as well as the business, we have determined that it will be the `members` (the businesses representing the customers of MIG) that will be the primary entities used in the recommendation system. In order to build the most complete picture about them that we can, we will also need the users belonging to those businesses,

any articles / adverts that those businesses post, as well as visibility of which users attend events together / send messages to each other. The essence of the chosen ML approach will be to jointly use fixed characteristics about those member businesses (sectors / commerce etc.) as well as their connection to other member businesses (through events / chat / follows relationships) to form the basis of the recommendation.

Database Options

RDBMS

A relational database is a collection of data items with pre-defined relationships between them. These items are organized as a set of tables with columns and rows. Tables are used to hold information about the objects to be represented in the database. Each column in a table holds a certain kind of data and a field stores the actual value of an attribute. The rows in the table represent a collection of related values of one object or entity.

NoSQL

A NoSQL (Not only SQL) database includes simplicity of design, simpler horizontal scaling to clusters of machines and finer control over availability versus traditional RDBMS. The data structures used by NoSQL databases are different from those used by default in relational databases which makes some operations faster in NoSQL. The suitability of a given NoSQL database depends on the problem it should solve. Data structures used by NoSQL databases are sometimes also viewed as more flexible than relational database tables¹.

Advantages of NoSQL

1. **High scalability** – NoSQL database use sharding² for horizontal scaling³.
2. **High availability** – NoSQL databases feature auto-replication of data, which makes the data highly available because in case of any failure data replicates itself to the previous consistent state.

¹ <https://www.geeksforgeeks.org/introduction-to-nosql/>

² Sharding is partitioning of data and placing it on multiple machines in such a way that the order of the data is preserved.

³ Vertical scaling means adding more resources to the existing machine, whereas horizontal scaling means adding more machines to handle the data.

Disadvantages of NoSQL

1. **Narrow focus** – NoSQL databases have a very narrow focus as it is mainly designed for storage but it provides very little functionality.
2. **Management challenge** – The purpose of big data tools is to make management of a large amount of data as simple as possible. But it is not so easy. Data management in NoSQL is much more complex than a relational database. NoSQL, in particular, has a reputation for being challenging to install and even more hectic to manage on a daily basis.
3. **(Potentially) large document size** – Some database systems like MongoDB and CouchDB store data in JSON format. Which means that documents are quite large (BigData, network bandwidth, speed), and having descriptive key names actually hurts, since they increase the document size.

Summary

In summary, we can say that NoSQL databases are highly specialised and can be very advantageous to a business, should the needs of the business fall into a use case that the NoSQL database is designed for. On the other hand, if the specialised need is not there, then they are difficult to recommend.

NoSQL Options

NoSQL solutions fall into four main categories:

Key-value Databases

In contrast to relational databases, which define a data structure made up of tables of rows and columns with predefined data types, key-value databases store data as a single collection without any structure or relation. Key-value databases are often described as highly performant, efficient, and scalable. Common use cases for key-value databases are caching, message queuing, and session management.

Some popular open-source key-value data stores are Redis, Memcached and Riak.

Columnar Databases

Columnar databases, sometimes called *column-oriented databases*, are database systems that store data in columns. This may seem similar to traditional relational databases, but rather than grouping columns together into tables, each column is stored in a separate file or region in the system's storage. Because the data in each column is of the same type, it allows for various storage and read optimization strategies. In particular, many columnar

database administrators implement a compression strategy such as run-length encoding to minimize the amount of space taken up by a single column. This can have the benefit of speeding up reads since queries need to go over fewer rows. One drawback with columnar databases, though, is that load performance tends to be slow since each column must be written separately and data is often kept compressed. Incremental loads in particular, as well as reads of individual records, can be costly in terms of performance.

Some popular open-source columnar databases are Apache Cassandra, Apache HBase and Clickhouse.

Document-oriented Databases

Document-oriented databases, or *document stores*, are NoSQL databases that store data in the form of documents. Document stores are a type of key-value store: each document has a unique identifier — its key — and the document itself serves as the value.

The difference between these two models is that, in a key-value database, the data is treated as opaque and the database doesn't know or care about the data held within it; it's up to the application to understand what data is stored. In a document store, however, each document contains some kind of metadata that provides a degree of structure to the data.

Document-oriented databases have seen an enormous growth in popularity in recent years. Thanks to their flexible schema, they've found regular use in e-commerce, blogging, and analytics platforms, as well as content management systems. Document stores are considered highly scalable, with sharding being a common horizontal scaling strategy. They are also excellent for keeping large amounts of unrelated, complex information that varies in structure.

Some popular open-source document based data stores are MongoDB, Couchbase and Apache CouchDB.

Graph Databases

Graph databases can be thought of as a subcategory of the document store model, in that they store data in documents and don't insist that data adhere to a predefined shape or style. The difference, though, is that graph databases add an extra layer to the document model by highlighting the relationships between individual documents. Certain operations are much simpler to perform using graph databases because of how they link and group related pieces of information. These databases are commonly used in cases where it's important to be able to gain insights from the relationships between data points or in applications where the information available to end users is determined by their connections to others, as in a social network. They've found regular use in fraud detection, recommendation engines, and identity and access management applications.

Some popular open-source graph databases are Neo4J, ArangoDB, OrientDB, Amazon Neptune, and GraphDB.

Recommendation

Graph Database

The main issue with key-value, columnar, and document models is the lack of relationship representation power. Each respective class of database stores values / attributes / documents entirely separate from the others. Since we are interested in the inherent connection between many types of entities (members / users / events / messages / commerces etc.), these models offer no benefit over RDBMS other than potentially read speed. The lack of expression of relationships and overall small to medium size of the current database make them difficult to recommend for our project.

On the other hand, graph databases are very well suited to our needs. They allow us to model the complex interconnectedness of all the entities that we are interested in, leverage existing graph-based algorithms directly and are well studied in machine learning for the task of recommendation. Since our needs are highly specialised and fall into exactly what graph databases can provide, we recommend incorporating one into the system, alongside the existing RDBMS.

Which Graph Database?

Graph Database	Pros	Cons	Used by
ArangoDB	<ul style="list-style-type: none">• Graph edges can also store data values• Open-source with available commercial support (Apache 2.0 licensed) community version free for all uses• Multi-model store (not only graphs)• Allows visualisation• Sharding	<ul style="list-style-type: none">• Proprietary query language	<ul style="list-style-type: none">• UBS,• Walmart,• eBay,• Adobe,• Volvo Cars,• Orange and• Airbus
Neo4j	<ul style="list-style-type: none">• Most established• Allows visualisation• Standard Graph query language (Cypher)	<ul style="list-style-type: none">• Can ONLY store graphs	<ul style="list-style-type: none">• TigerGraph• Torch AI• Cisco• General Dynamics Mission Systems• SOS International LLC• Credit One Bank
OrientDB	<ul style="list-style-type: none">• Multi-model• Apache 2.0 license	<ul style="list-style-type: none">• Minimal documentation• Some extra programming complexity	<ul style="list-style-type: none">• GittiGidiyor.• Securly.• Vagas.com.• Covve.• JimmyCode.• Bright Power.• 7bridges.• LevelUP IVS.

AWS Neptune	<ul style="list-style-type: none"> Fully managed by Amazon Integrates with other Amazon technologies 	<ul style="list-style-type: none"> Single model AWS can be expensive 	<ul style="list-style-type: none"> Herren. Geniusee. FetchyFox. extractBot. juncture. CloudDevelopment. SEQL Tech Stack. Industrial Inference.
--------------------	--	--	--

Supported Programming Languages

Graph Database	Server-side Languages	Supported Languages
Neo4j		.Net Clojure Elixir Go Groovy Haskell Java JavaScript Perl PHP Python Ruby Scala
ArangoDB	javascript	C# C++ Clojure Elixir Go Java JavaScript (Node.js) PHP Python R Rust
OrientDB	javascript java	.Net C C# C++ Clojure Java JavaScript JavaScript (Node.js) PHP Python Ruby Scala

GraphDB	Javascript	.Net C# Clojure Java JavaScript (Node.js) PHP Python Ruby Scala
Amazon Neptune	No	C# Go Java JavaScript PHP Python Ruby Scala

Our recommendation is ArangoDB for its multi-model functionality.

Costs of Hosting a Graph Database

The next section outlines the costs associated with hosting a graph database on the cloud. We will first outline a summary of the advantages of the three main cloud providers. Then, following our recommendation of ArangoDB in the previous section, we will compare the costs of ArangoDB to Neo4j, the most popular graph database on the market.

Cloud Hosting Options

The following subsection paraphrases a comparison between the three main cloud hosting services Amazon Web Services (AWS), Microsoft Azure and Google Cloud Platform (GCP) provided by Scott Carey, editor for Computerworld⁴.

Provider	Pros	Cons
AWS	<ul style="list-style-type: none">• Extensive options• Head start on the competition (began in 2006)• Large number of third-party software services on AWS marketplace	<ul style="list-style-type: none">• Breadth of options can be complex• Dismissive of the benefits of on-premise private clouds• Ethical implications of supporting Amazon
Microsoft Azure	<ul style="list-style-type: none">• Easy transition for businesses that make extensive use of Microsoft products• Increasingly open to open-source technologies	<ul style="list-style-type: none">• Many outages (last AWS outage in 2017, GCP in 2019)• Poor customer support
GCP	<ul style="list-style-type: none">• Strong standing in open-source community• Strengths in "big data and other analytics applications, machine learning projects, cloud-native applications, or other applications optimised for cloud-native operations."	<ul style="list-style-type: none">• Struggled to break into the enterprise market ("focused on proving itself on smaller, innovative projects at large organisations, rather than becoming a strategic cloud partner") but keen to change this

⁴

<https://www.computerworld.com/article/3429365/aws-vs-azure-vs-google-whats-the-best-cloud-platform-for-enterprise.html>

Summary

In very broad terms, AWS continues to lead the way in terms of offering the widest range of functionality and maturity. It continues to be the clear market leader, but the gap is closing.

Its expansive list of tools and services, along with its enterprise-friendly features make it a strong proposition for large organisations. Meanwhile its huge and continuously growing infrastructure provides economies of scale that enable aggressive price cuts.

But it appears that Microsoft has started to bridge the gap between the two, and will continue to do so with its ongoing investment in building out the Azure cloud platform and further plans to strengthen ties with its on-premise software.

For organisations already heavily invested in Microsoft in terms of technology and developer skills – of which there are undoubtedly many – Microsoft Azure will continue to be a strong proposition.

Then there is Google, which could prove a more serious enterprise competitor under its new leadership. It was already making good progress with certain customers, especially with its Kubernetes and machine learning expertise, but has much more work to do to prove itself a viable enterprise option.

Categories of Hosting Services

For cloud-based hosting of a graph database, there are three main categories of service:

- Fully managed
- Partially managed
- Self managed

Fully Managed Options

Managed cloud services are the partial or complete management and control of a client's cloud platform, including migration, maintenance and optimization. By using a managed cloud service provider, a business can ensure its cloud resources run efficiently. Outsourcing cloud management also allows businesses to avoid new hiring and training costs⁵. Ideal for developers building a cloud-based application who need an easy-to-use, fully-managed and cost-effective graph database service.

Service	Memory Per Node	Storage Per Node	Hourly	Month (30 days)
Neo4j Aura Professional	8GB	16GB	~£0.52/h	~£374.40
ArangoDB Oasis (on AWS/Google Cloud/Azure)	8GB	20GB	~£0.41/h	~£295.20

5

https://www.insight.com/en_US/glossary/m/managed-cloud-services.html#:~:text=Managed%20cloud%20services%20are%20the,its%20cloud%20resources%20run%20efficiently.

Cloud Managed Services

A paid service **in addition** to hosting costs.

From the Neo4j page: ``ideal for enterprises who wish to run a Neo4j database in their own cloud infrastructure but need a qualified team of experts to manage it for them so they stay focused on their core business operations”.'

Neo4j

Service Level	Cost (30 days)	What you get
Standard	\$49 (~£35)	<ul style="list-style-type: none">• Managed Security• Basic Monitoring• 24 x 7 Support
Pro	\$99 (~£71)	<ul style="list-style-type: none">• All Standard features• Managed Backup Full and Daily Snapshots• Managed Operating System Patches and Updates, Hardening, Configuration and Tuning• Instance Monitoring and Response CPU, RAM, DISK IO, and URL• 24 x 7 Monitoring Software
Enterprise	\$125 (~£90)	<ul style="list-style-type: none">• All Standard and Pro features• Application Monitoring and Response CPU, RAM, Disk IO, URL, and Application metrics• Advanced Enterprise Analytics and Dashboard

ArangoDB with Additional Packages and Scripts by Intuz

A software package on AWS marketplace for a cloud managed service for ArangoDB on AWS⁶.

Service	Software/hr	EC2/hr	Total/hr	Total/month (30 days)
T2.medium <ul style="list-style-type: none">• 4GB memory• 1 core• EBS storage only• Low / moderate network	\$0.01	\$0.052	\$0.062 (~£0.045)	~£32.40
T2.large <ul style="list-style-type: none">• 8GB memory• 2 (virtual) cores• EBS storage only• Low / moderate network	\$0.01	\$0.106	\$0.116 (~£0.083)	~£59.76

Self-Managed Cloud Options: Neo4J

Google Cloud Platform (GCP)

Running on Google Cloud provides a few options, depending on what you want to do.

- Test Drive - GCP and Neo4j offer a free test drive with hands-on walkthroughs and introduction materials on graphs and Neo4j.
- Single instance - our Google cloud image documentation gives you the steps to launch a single instance from an image with a few commands and interact with the Neo4j instance.
- Causal Cluster - the Enterprise edition of Neo4j is also available on GCP Marketplace, and users can launch a causal cluster from there. The launch and interaction steps to deploy Neo4j causal clusters on GCP are shown in the Neo4j documentation.

⁶

https://aws.amazon.com/marketplace/pp/B07957T8G2?qid=1615303519269&sr=0-1&ref=srh_res_product_title

-
- Google Kubernetes Marketplace (Docker container-based) - Neo4j Enterprise is also available on Kubernetes Marketplace, so users can launch Neo4j clusters into Google Kubernetes Engine (GKE) clusters.

Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.

There are three options for running on EC2 detailed below, with each option depending on the needs of the user and environment.

- Single instance (VM-based) - instructions for launching VMs with Amazon's command line tool are provided in the developer guide to deploy Neo4j on EC2 with a custom Amazon Machine Image (AMI). Using this method, both Community and Enterprise options are available.
- Neo4j Community edition - available directly from the AWS marketplace.
- Causal Cluster - launch directly from the AWS Marketplace, as well. This option creates a multi-VM clustered configuration with the choice to configure a number of aspects of the cluster, including number of core nodes, read replicas, hardware sizing, encrypted EBS volumes, and other options.

Microsoft Azure

Neo4j can be deployed directly from Azure Marketplace.

- Single Instance.
- Causal Cluster.

Self-Managed Cloud Options: ArangoDB

Deploying ArangoDB on AWS

Up to and including ArangoDB 3.2, official ArangoDB AMI were available in the AWS marketplace. Such AMIs are not being maintained anymore, though. However, deploying on AWS is still possible, and again, a quite common scenario.

After having initialized your preferred AWS instance with one of the ArangoDB supported operating systems, using the ArangoDB Starter, performing a Manual Deployment or using Kubernetes are all valid options to deploy on AWS.

Deploying ArangoDB on Microsoft Azure

No Azure-specific scripts or tools are needed to deploy on Azure. Deploying on Azure is still possible, and again, a quite common scenario. Like AWS, ArangoDB can be initialised with ArangoDB Starter, a manual deployment or using Kubernetes.


Self Managed Cloud Option Summary

Graph Database	Azure	AWS	Google Cloud
Neo4j	Y	Y	Y
ArangoDB	Y	Y	N

Cloud Option Price Comparison


A summary of four comparable instances from AWS, Azure and Google Cloud:⁷

Types of Instances						
Instance Type	AWS Instances	AWS RAM (GiB)	Azure VMs	Azure RAM (GiB)	Google VMs	Google RAM (GB)
General purpose	m6g.xlarge	16	B4MS	16	e2-standard-4	16
Compute optimized	c6g.xlarge	8	F4s v2	8	c2-standard-4	16
Memory optimized	r6g.xlarge	32	E4a v4	32	m1-ultramem-40	961
Accelerated computing	p2.xlarge	61	NC4as T4 v3	28	a2-highcpu-1g	85



Each instance has 4 virtual CPUs (except memory-optimized and accelerated-computing instances for Google cloud, which start from a minimum of 40 and 12, respectively). The general purpose instance type seems the most applicable to our needs. The costs are summarised below:

On-Demand Pricing						
Instance Type	AWS	Azure	Google	AWS pricing (per hour)	Azure Pricing (per hour)	Google pricing (per hour)
General purpose	m6g.xlarge	B4MS	e2-standard-4	\$0.154	\$0.166	\$0.156
Compute optimized	c6g.xlarge	F4s v2	c2-standard-4	\$0.136	\$0.169	\$0.235
Memory optimized	r6g.xlarge	E4a v4	m1-ultramem-40	\$0.202	\$0.252	\$6.303
Accelerated computing	p2.xlarge	NC4as T4 v3	a2-highcpu-1g	\$0.90	\$0.526	\$3.839



AWS edges the competition for general purpose at \$0.154/h. It is worth noting that AWS offers a one-year commitment plan at a discounted rate of \$0.0924.

⁷ Images taken from

<https://www.simform.com/compute-pricing-comparison-aws-azure-googlecloud/>

Executive Summary

Overall, we recommend ArangoDB running on AWS. Some key features that differentiated ArangoDB from the others are:

- Support for multi-models on the same instance
- An Apache 2.0 open source license option with commercial support options
- Support for storing full json on edges
- Amount of available documentation and examples
- Ease of integration with a number of programming languages
- Low cost on AWS for the compute power and resources that we need.