Fundamentals of Data Science

Data collection and management

Southampton Data Science Academy

Data storage and management

Introduction to data Management

- Having obtained data to analyse, we need to have some infrastructure in place to perform analysis on it
- The increased availability of processing power and the rise of cloud computing offers advantages to for performing data science
- This section will look at cloud computing as a concept, and the possible advantages of using MongoDB for large scale data

Cloud Architecture

Cloud architecture is a model which makes available devices, hardware, or software to other users on demand

This differs from the traditional model through the resources being leased based on the actual usage of the service

These come at different levels of service:

- **IaaS** (Infrastructure as a Service) provides networking capabilities, hardware, and storage
- **PaaS** (Platform as a Service) sits on top of IaaS where the provider offers components to run software, e.g., Java or Python
- **SaaS** (Software as a Service) provides software which the user can access through their Web browser

High Performance Computing in the Cloud

- There are various high performance computing (HPC) options available for cloud computing, depending on the choice of vendor
- Amazon AWS is a popular choice, and contains options for instances optimised for high RAM, high CPU, and high GPU
- Apache Hadoop is an application designed with parallel storage and processing in mind, and this works well with the cloud computing paradigm
    - Amazon also offers collections of Hadoop machines (clusters)
- Databases like MongoDB benefit from being able to distribute their storage and processing over multiple servers, and can be used alongside HPC in the cloud

Advantages of Cloud Computing

- It requires data to be held by another party. This can have potential privacy or other legal risks
- It may end up more expensive than traditional architectures, particularly if the project is done on a small scale
- Vendor lock in could restrict future development choices
- There may be a lack of flexibility in what the provider allows you to do within the terms of the EULA

Disadvantages of Cloud Computing

- It requires data to be held by another party. This can have potential privacy or other legal risks
- It may end up more expensive than traditional architectures, particularly if the project is done on a small scale
- Vendor lock in could restrict future development choices
- There may be a lack of flexibility in what the provider allows you to do within the terms of the EULA

Southampton Data Science Academy

NoSQL databases

NoSQL refers to databases which do not model their data according to a tabular, relational format

- No can also stand for “not only”, as some do support SQL-like syntax

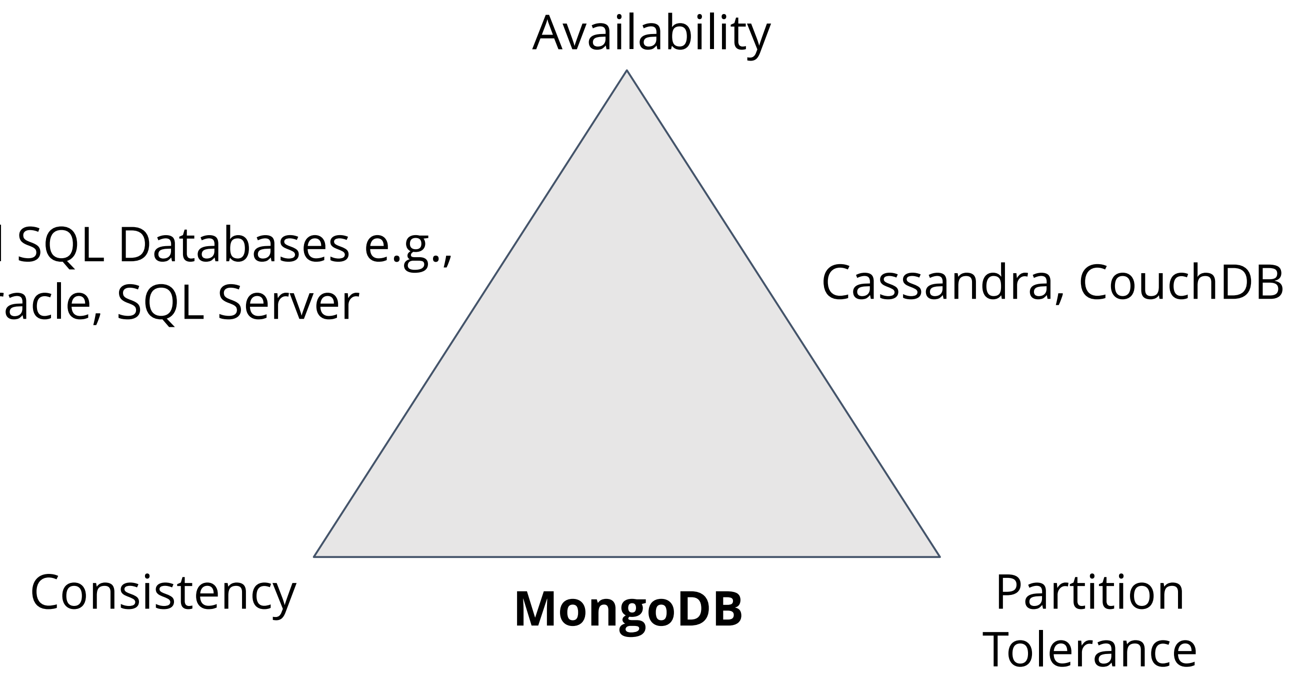
Without these constraints, databases for different purposes or types of data can focus on different priority

According to Brewer’s CAP theorem, it is impossible to guarantee all three of the properties: Consistency, Availability, Partitioning

- It is possible to have any two of these, but only by sacrificing guarantees on the third

Southampton Data Science Academy

Brewer's CAP Theorem

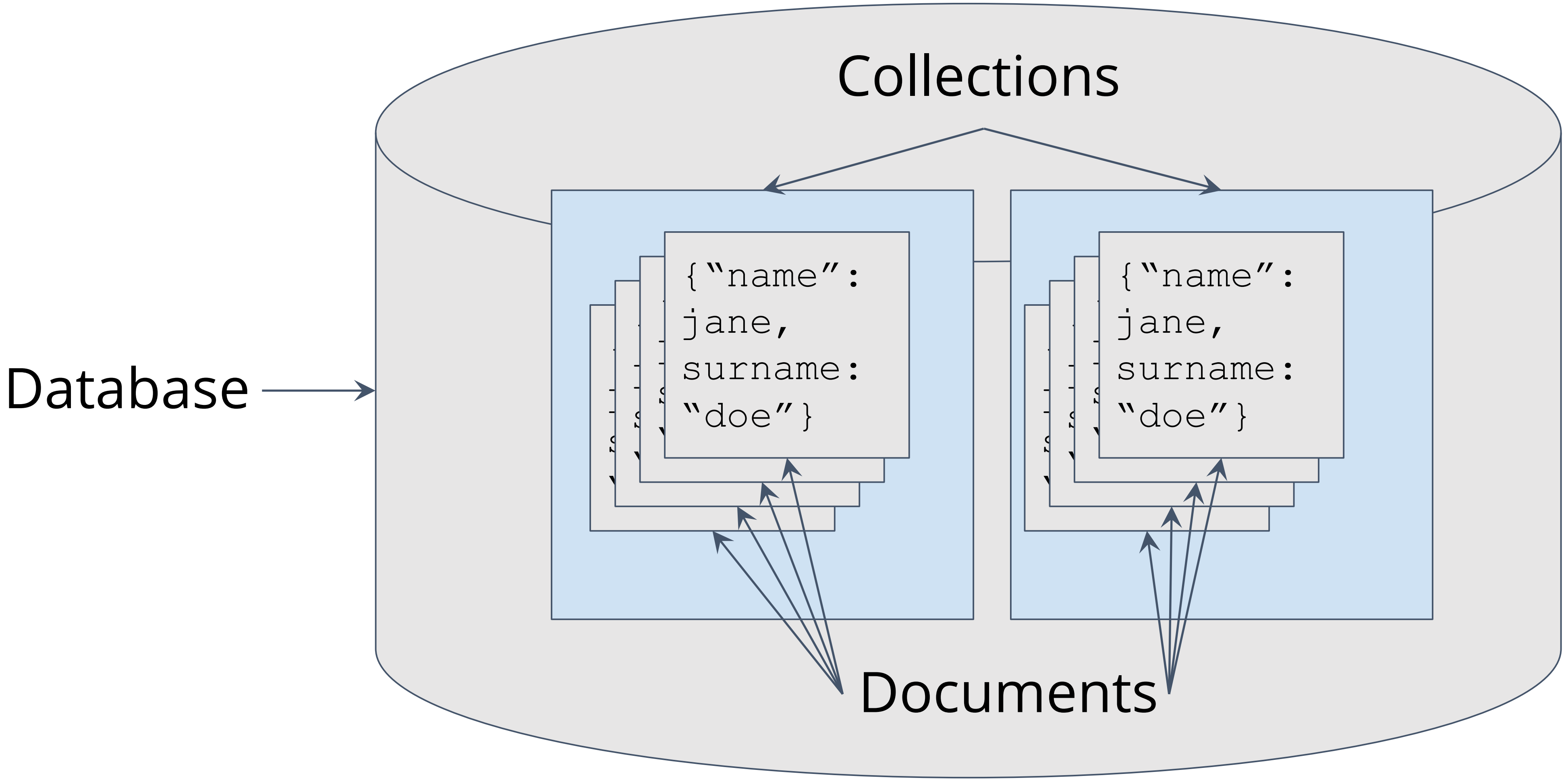


Adapted from http://blog.nahurst.com/visual-guide-to-nosql-systems

What is MongoDB?

- MongoDB is an example of a NoSQL database
- Stores data as a series of documents in a collection
  - Documents are stored in a JSON-like syntax BSON
  - Do not require a consistent schema
- Uses JavaScript as a query language
- Powerful, flexible indexing - including geospatial data
- Works on many platforms, with a selection of drivers for popular programming languages

MongoDB storage structure



MongoDB queries

MongoDB uses dot notation to perform functions on collection objects.

Assuming a collection called users, to get all users we would do:

```
db.users.find()
```

We can add more criteria to the search by adding an object as a parameter in the format `{ field_name: condition }`

```
db.users.find({ “surname”:
“Tables”})
```

The output of a find query is a cursor, which also has methods, e.g.,

`db.users.find().count()` gives the amount of records returned by the query

MongoDB and SQL Trade-offs: Schemas

Enforced Schema vs Schemaless

- Using SQL systems requires that the schema be set out in advance
  - If the data model is going to change regularly, this could complicate development
  - The document oriented model is more similar to the object oriented paradigm commonly used today
- The specific schema enables normalisation, the process of separating the storage of data to minimise duplication
  - Data is connected through the use of joins and foreign keys
- MongoDB does not support joins
  - This can require additional round trips to the server to perform queries
  - These can be mitigated by appropriate duplication of data

The flexibility of MongoDB requires developers to be careful in dealing with issues like duplication rather than having the RDBMS do it

MongoDB and SQL Trade-offs: Transactions

A **transaction** is a feature of RDBM systems which isolates a change from other changes in the database

MongoDB does not support transactions, although it does allow for some limited **atomicity –** I.e., that either all changes will occur or none will occur

- And it is possible to get transaction like features with MongoDB, although they are complicated to implement

Where it is important to ensure that business critical data remains consistent, it is a better choice to use an RDBM system

- However, the use of transactions limits horizontal scaling

Southampton Data Science Academy

MongoDB and SQL Trade-offs: Scalability

MongoDB supports sharding out of the box, which allows horizontal scaling

- A database scales horizontally where it is possible to use more machines on the database, as opposed to vertical scaling which increases the power (CPU/RAM Etc.) to an individual machine
- A MongoDB shard splits a single logical database up into several physical databases

It is possible to have similar features in RDBMS, e.g., MySQL Cluster, but these are complex to set up

- The schema has to be planned right in order to get good performance

MongoDB supports MapReduce, which allows parallel processing of large datasets across many instances of MongoDB data stores

- See https://en.wikipedia.org/wiki/MapReduce for an explanation of the MapReduce paradigm