**Summary1: Application of NoSQL Database in Web Crawling**

Web crawling is one of the most important applications of the internet. The main purpose of web crawling is to filter and gather wide variety of data and resources from internet grouped by a certain theme and to construct search engine for users. . Huge capacity of storage space and low hardware costs are required by web-crawling. The basic concept of web crawling resolves around a spider , controller and original page library. The URLs distributed to the spiders for the crawling purpose by the controller. The controller accesses the URL database to control the spider crawling. We can say that web crawling is a process that automatically obtains information from the web pages through the link relationships between them and expands to the entire web.

In order to have a better understanding about web-crawling, we should know about Meteorological BBS information. It’s a collection information system filters and collects the posts of the representative meteorological BBS in internet. Handling the data for a meteorological BBS system using a relational database is not so effective as there will be more than ten thousand posts and more that hundred thousand floors and the data increases at an enormous rate every day. The NO SQL database provided a simple and very effective solution to handle the data of such a system. The design philosophy of NOSQL is by reducing the data consistency and integrity constraints in exchange for high availability and partition tolerance. MongoDB is one of the most popular NOSQL database whose main objective is to bridge the gap between key-value stores with high performance and scalability.

Relational database has multiple tables’ storage with foreign key, sharp decline in query performance with huge amount of data and vertical scalability with high cost. Compared to relational database, MongoDB supports schema-free, has great query performance with huge amount of data and provides easy horizontal scalability with low cost of hardware. It is more suitable for data storage in web crawling.

**Summary Report2: Comparing NoSQL MongoDB to an SQL DB**

The database which is having a non-relational nature is basically a NoSQL database. The main necessity of a NoSQL database came into the picture when the data in applications around the world increased on a tremendous rate and became so huge that it became very difficult for a relational database to maintain the ACID properties on the data as well as show highly efficient behavior as far as the basic database operations like storing and access data from the tables. The NoSQL database is not having anything other than drastically increasing the performance of any application from the database side. The need of using a NoSQL database came from the problem that a relational database faces when there is a wide network of tables connected together using the primary and foreign key concept and the join feature present in any relational database. Combining information based on a matching value for a primary key and foreign key across multiple tables in the relational model requires using a join operation. The larger the schema and the more tables that need to be joined, the longer it takes for the relational database to fetch the data.

There are some problems with MongoDB and other NoSQL databases. They are lack of many features which relational database has. For instance, MongoDB only provides atomic operations within a single document. Another important feature which is missing in MongoDB is most aggregate functions. It does not have many of the simple aggregate functions which are like standard in relational database. But still MongoDB deals it with other way with the introduction of map reducing concept. It is basically a programming abstraction which is widely used to process big data. It basically consists of two stages map stage and reduce stage. During map stage user specifies the required computational elements which is emitted in the form a key values. These emitted values will be given as input to the reduce function where the aggregation happens depending on the logic provided by the user.

Eventually, when SQL is compared to MongoDB, MongoDB has better runtime performance for inserts, updates and simple queries. SQL did well when updating and querying non-key attributes as well as for aggregate queries. MongoDB is known to work best as a distributed database.

**Summary Report3: Data Aggregation System**

Data Aggregation System is designed as an additional layer on top of a heterogeneous ecosystem of existing CMS data services. The architecture of DAS was designed as a series of independent components, which can sale from all running on a single node to multiple nodes and duplicate elements as necessary. The web server handles user sessions, whether they originate from a browser or automated scripts. Queries made here are parsed onto the cache server for processing, during which time the web server periodically polls for the current status and displays it to the user by AJAX until the request is completed. The cache server consists of a pool of worker threads which handle the DAS queries received from the web front-end. Each query is handled entirely by single thread. The data in DAS is stored in two separate collections i.e. the raw and merger caches. The raw cache contains two types of data i.e. the documents returned by the respective APIs and the documents describing the current status of queries.

The cache consists of one or more MongoDB shards. MongoDB is a document store which natively stores the JSON documents DAS uses as its internal representation. This is used both for the primary record stores and for other databases required by DAS, such as storing the server logs, analytics data and mapping between keys. The analytics server provides a facility for scheduling regular tasks. The rest of DAS operates only when triggered. The DAS implements the cache aspect in order to reduce the latency that the users experience when performing the complex, cross-service queries, but since the data in question usually has TTL of less than an hour users will relatively often find that the data they request is not in the cache and must be fetched on demand.