IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 11, Issue 7, July 2022

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.118

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 7, July 2022||

| DOI:10.15680/IJIRSET.2022.1107024|

Database Sharding: Implementation by Oracle Database and Performance Analysis

Er. Purva Paroch[1], Dr. Simmi Dutta[2], Er. Vivek Mahajan[3]

P.G. Student, Department of Computer Engineering, Govt. College of Engg. & Technology, Jammu, India[1]

Head of Department, Department of Computer Engineering, Govt. College of Engg. & Technology, Jammu, India [2]

Principal, Govt. Polytechnic College, Reasi, India[3]

ABSTRACT: The amount of data being produced daily is increasing exponentially. Many organizations and companies track our personal details from applications since they generate a lot of data traffic. These applications are for media sharing, social networks, eCommerce etc. The database that stores the huge loads of data generated needed to scale out to handle the network traffic efficiently. Whenever an application or website is being set up, its scalability is to be maintained. After the growth of an application or website, in order to accommodate the increasing traffic, scalability comes to our aid. In case of data-driven applications and websites, integrity and security features are also maintained. Web application requirements have increased in terms of scalability. This includes the applications that implement Web 2.0 technologies. One cannot predict the rate at which data generated for a particular application will increase. This is why some organizations decide to implement their database using such an architecture that will allow them to scale their database dynamically. This is done by the sharded database approach.

KEYWORDS: Network traffic, database, scalability, data-driven, Web 2.0, sharding, sharded database.

I. INTRODUCTION

The word shard means ‘a piece’ or ‘a small part of a whole’. In terms of database, shard refers to a part of the database. More specifically, a database shard is a horizontal partition of data in a database or search engine. Each shard resides on a separate database server instance in order to spread load. This is done by breaking the big chunks of data into smaller pieces and allocating them to different servers. For this particular case, the servers used are database servers [1].

Data can be sharded over any particular attribute. We know that data in a database is stored in a tabular format. Scaling of a database can be done in two ways. This includes scaling up and scaling down. Scaling up, also known as vertical sharding, means enlarging the available resources to achieve the desired state of performance. We add additional resources to the existing system, which is expensive since it requires disk storage, bigger machines, memory and processors. Therefore, it is not always feasible to implement scaling up on a database. The other method is by implementing scaling out, or horizontal scaling, on the database. The tabulated data is separated along the rows to create multiple different tables. The partitioning is done according to a key. Each partition consists of the same schema and columns but the rows are different. Data in each shard is unique and independent of the data held in other shards [2]. Consider the given table in the database.

Table 1: Original Table

USER ID	NAME	AGE	SEX
001	ABC	24	F
002	PQR	30	M
003	XYZ	27	M

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 7, July 2022||

| DOI:10.15680/IJIRSET.2022.1107024|

As shown, the table has four attributes namely User ID, Name, Age and Sex. The records of ABC, PQR and XYZ are stored in the table. If the table is scaled up, or vertically sharded, then the following results will be obtained.

Table 2: Vertical Partitions

Table 2.1.: VP_1

USER ID	NAME	AGE
001	ABC	24
002	PQR	30
003	XYZ	27

Table 2.2.: VP_2

USER ID	SEX
001	F
002	M
003	M

Tables 2.1 and 2.2 namely VP_1 and VP_2 are obtained by scaling up the original table. There is a User ID column that is common in both partitions, and it will act as a key during database manipulation operations.
Another way to divide the Table 1 is by sharding, that is, by horizontal partitioning. Horizontal partitioning is as shown below.

Table 3: Horizontal Partitions

Table 3.1.: HP_1

USER ID	NAME	AGE	SEX
002	PQR	30	M
003	XYZ	27	M

Table 3.2.: HP_2

USER ID	NAME	AGE	SEX
001	ABC	24	F

The shards here are labeled as HP1 and HP2, with the original table being split along the rows. Here, the User ID attribute is used as a key to perform data manipulation operations on HP_1 and HP_2.

II. IMPLEMENTATION

Oracle BlueKai Data Management Platform:
Data Management Platform (DMP), which was previously known as BlueKai, is a company acquired by Oracle in 2014. Oracle DMP is the world's most extensively adopted big data platform and it is used for businesses that require actionable information and analytics on targeted audiences to create marketing campaigns. It ingests, organizes, and provides insights and analytics to users on a great amount of advertising platforms. Oracle BlueKai Data Management Platform is a part of Oracle Cloud Applications. The sharding feature of Oracle database is used

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 7, July 2022||

| DOI:10.15680/IJIRSET.2022.1107024|

for BlueKai platform, which is one of the biggest relational OLTP database deployments in the world. Key metrics of the primary database are as shown below:

Table 3.1.1

Transactions	1 Million/second
Events	30 Billion/day
API Calls	30 Billion/day
API Payload Size	125 Kilobytes/API call (Average)
Average read time	1.6 Milliseconds/API call
Average write time	2.5 Milliseconds/API call
Total database size	2.5 Petabytes
Rows in largest table	22 Billion
Redo generation rate	180 Terabytes/hour
Network traffic	1 Terabit/second
Total machines	52 Oracle Compute Instances
Total CPU	2,704 Cores
Total memory	38,740 Gigabytes

In order to process data efficiently on such a large scale, innovation in terms of the following areas is required:

- Automatic content classification algorithms
- Bespoke probabilistic data structures used to perform unique counting and overlap analysis at a massive scale
- Autonomic infrastructure services that automatically adapt to the dynamic nature of our data processing environments
- Robust audience taxonomies

Shards in Oracle Database meet 99.99% uptime SLAs due to high availability and disaster recovery.

Oracle Sharding helped DMP in the following areas:

1. Scalability: Sharding follows a linearly scalable architecture that allows performing scaling as the data volume grows, without impacting the latency. This feature also helps to support business growth.
2. Availability: Due to the fault isolation feature, any problems with one shard have no impact on availability of other shards. There are multiple availability domains where primary shards are located. The Oracle Data Guard is used to have a replica instance in an availability domain different than the primary instance. This whole setup provides protection from failure of an availability domain or even an entire region.
3. Performance: After the implementation of Oracle Sharding, an improvement in the performance was recorded, compared to other key-value stores used in the past.
4. Stability: Not a single database outage due to software issues has been experienced since the implementation of Oracle Sharding in Oracle Database.

The Oracle DMP environment has grown considerably over time. There are systems for data streaming, real-time key value databases, distributed batch data processing and workflow management among other functionalities. Increase in the number of systems gives rise to the complexity and cost. This can be in the form of workflow management and coordination between systems, data consistency issues, no support for ACID transactions which burdens the application or having to store multiple copies of data in different systems to support various use cases, which is expensive.

Previously, once the data volume and velocity reached a certain threshold, it was impractical and not possible to use traditional RDBMS approach. Sharding, which is vertical scaling, consolidates blast radius for system failure while also taking care of cost and practical limitations. Oracle Sharding supports an ACID compliant and horizontally scalable database that can support near-real-time key value use cases along with performing complex analytical operations. Due to the converged architecture of Oracle Database, a single data system can be used to drastically reduce complexity, cost and allow the simplification of data architecture by combining data and logic in a single data store that meets a diverse set of needs. As of January 2021, BlueKai is capable of making 1 million transactions per second, which makes this the largest OLTP deployment so far [3].

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 7, July 2022||

| DOI:10.15680/IJIRSET.2022.1107024|

III. PERFORMANCE ANALYSIS- SHARDING TRANSACTION THROUGHPUT

It is observed during experiments that independent of the number of shards, a steady state transaction rate decreases as data set increases. The same is observed in a non-sharded environment with a single database. Another observation is that transaction throughput follows a pattern around a medium range of warehouses, where sharding key store is kept in a multi-shard environment. This is observed in case of TPCC clients under sharding and is not determined outside sharding. These findings are independent of the shards but greatly depend on the warehouses.

1. Throughput regresses with increasing data size:

These results are obtained for a single shard configuration. This result is dependent on the number of warehouses and independent of the number of shards. The graphs shown below depict the result in a non-sharded as well as in a single-shard environment. Both these observations are consistent.

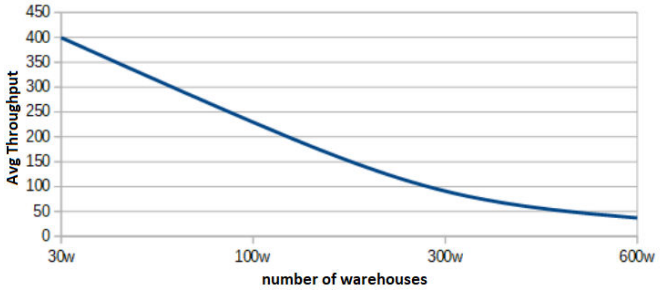

Fig.1.1 Average throughput in single database, non-shard environment

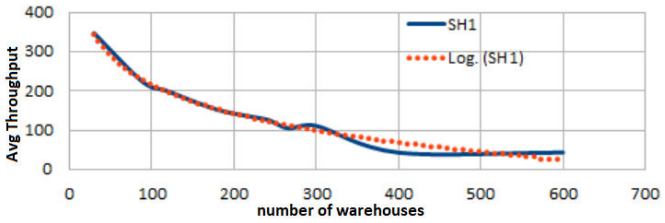

Fig.1.2. TPCC transaction, in single shard environment

2. Large fluctuations at a medium range of ware:

Throughput in case of transactions shows resonance oscillations with more than one shard, especially ranging from 260 to 400 warehouses. The figure below shows the result for 300 warehouses from a single shard (SH1) and for two shards (sh1 & sh2). These observations depend on the number of warehouses and are independent of the number of shards. The non-shard environment shows fluctuation in case of smaller data sizes, as shown in Fig.2.2.

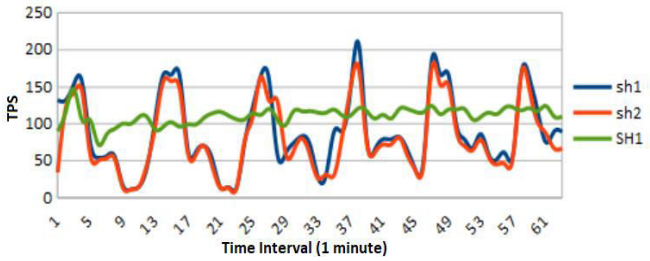

Fig.2.1 TPCC transaction throughput in 2 shard vs. single shard deployment

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 7, July 2022||

| DOI:10.15680/IJIRSET.2022.1107024|

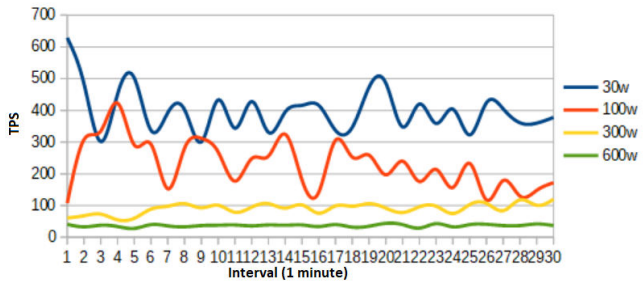

Fig.2.2. Transaction throughput in single database, non-shard environment

The above test results are further broken down into categories in order to show change in stability for various ranges of warehouses.

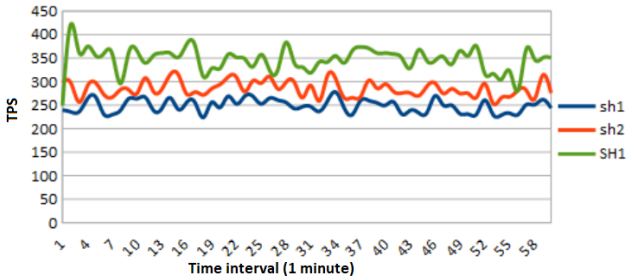

Fig. 2.a. TCPP transaction throughput for two deployments with 30 warehouses

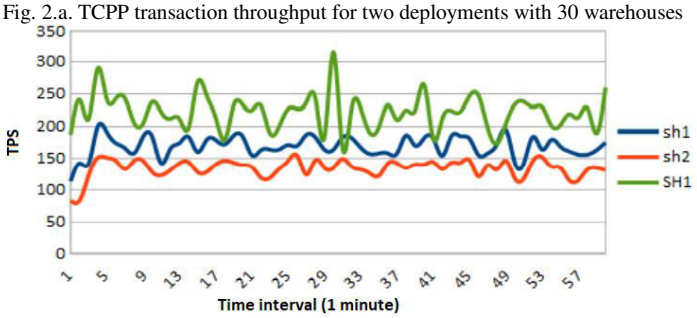

Fig.2.b. TPCC transaction throughput for two deployments with 90 warehouses

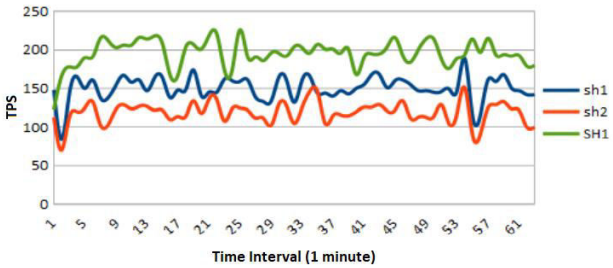

Fig. 2.c. TPCC transaction throughput for two deployments with 120 warehouses

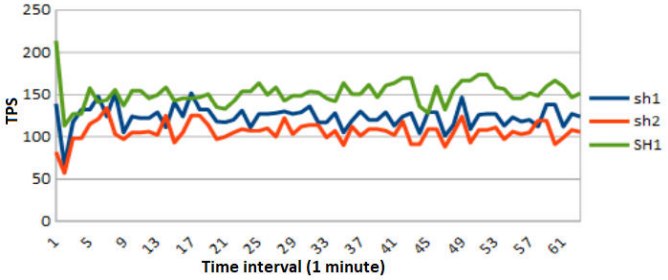

Fig. 2.d. TPCC transaction throughput for two deployments with 180 warehouses

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 7, July 2022||

| DOI:10.15680/IJIRSET.2022.1107024|

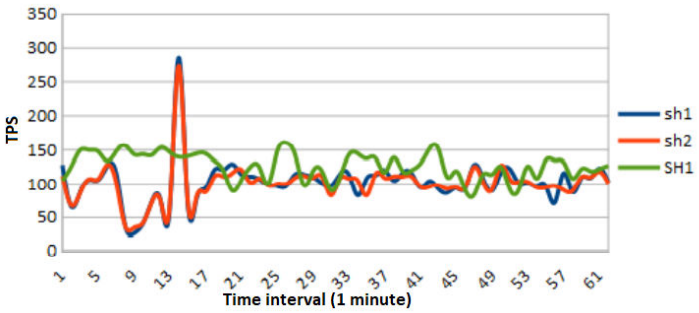

Fig. 2.e. TPCC transaction throughput for two deployments with 240 warehouses

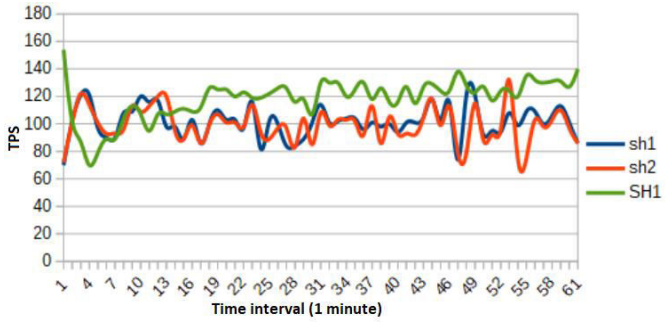

Fig. 2.f. TPCC transaction throughput for two deployments with 250 warehouses

In case of steady state transactions, due to increased data size there is decrease in transaction rate. Another observation is that TPS is unstable for certain ranges of warehouses. Both these observations are intertwined and exclusive to data shards while also being dependent on the warehouses. The cause of this observation can be determined by further experimentation.

IV. BENEFITS

a. Facilitates scaling out

In horizontal scaling, more machines are added to an existing stack to spread out the load which allows for more traffic and faster processing. In vertical scaling, there is up-gradation of hardware on an existing server, by adding more RAM or CPU.

b. Flexible setup

It is usually easy to scale up a relational database by upgrading its computing resources, but a non-distributed database will still be limited in terms of storage and compute power. Therefore, using sharding will make the setup more flexible.

c. Fast query response time

Whenever a query is submitted on the database that is not sharded, every row in the table has to be searched. In case of an application with a large database, querying process can be quite time consuming. In a sharded database where one table is sharded into multiple shards, fewer rows will be searched and resulting sets are returned more quickly.

d. Ensures high availability

If the application or website relies on an unsharded database, any outage is most likely to render the entire application or website unavailable. However, with a sharded database an outage will affect only a single shard. This might make some parts of the application or website unavailable to some users, but the overall impact of the outage will be less than that in the case of an unsharded database.

e. Cost effective

A network of smaller, cheaper servers is more cost effective in the long term than maintaining one big server [4].

| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 8.118|

||Volume 11, Issue 7, July 2022||

| DOI:10.15680/IJIRSET.2022.1107024|

V. DRAWBACKS

a. Query overhead

Each sharded database must have a separate machine or service which understands how to route a querying operation to the appropriate shard. This leads to additional latency on every operation.

Furthermore, if the data required for the query is horizontally partitioned across multiple shards, the router must query each shard and merge the results together. This can make an otherwise simple operation quite expensive and slow down response times.

b. Complexity of Administration

With a single unsharded database, only the database server itself requires upkeep and maintenance. With every sharded database, on top of managing the shards themselves, there are additional service nodes to maintain. In cases where replication is being used, any data updates must be mirrored across each replicated node. Overall, a sharded database is a more complex system which requires more administration.

c. Increased Infrastructure Costs

Sharding by its nature requires additional machines and compute power over a single database server. While this allows your database to grow beyond the limits of a single machine, each additional shard comes with higher costs. The cost of a distributed database system, especially if it is missing the proper optimization, can be significant.

d. No Native Support

Sharding is not natively supported by every database engine. For example, PostgreSQL does not include automatic sharding features, so user has to do manual sharding [4].

VI. CONCLUSION

Sharding can be a great solution for users that need to scale their database horizontally, for applications with large data requirements, when single database of our application is not capable to handle or store a large amount of growing data and high-volume read/write workloads. However, it also adds a great deal of complexity and creates more potential failure points for our applications. Sharding may be necessary for some, but the time and resources needed to create and maintain a sharded architecture could outweigh the benefits for others. It is debatable whether or not one should implement sharded database architecture. One needs to consider whether the benefits outweigh the costs or whether there is a simpler solution before they begin implementation.

REFERENCES

[1] How Sharding Works, https://medium.com/@jeeyoungk/how-sharding-works-b4dec46b3f6

[2] Understanding Database Sharding, https://www.digitalocean.com/community/tutorials/understanding-database-sharding

[3] Oracle BlueKai Data Management Platform scales to 1 Million transactions per second with Oracle Database Sharding deployed in Oracle Cloud Infrastructure, https://blogs.oracle.com/database/post/oracle-bluekai-data-management-platform-scales-to-1-million-transactions-per-second-with-oracle-database-sharding-deployed-in-oracle-cloud-infrastructure

[4] Database Sharding: Concepts and Examples, https://www.mongodb.com/features/database-sharding-explained

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.118

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

Scan to save the contact details