ISSN (ONLINE): 2349-7084
GLOBAL IMPACT FACTOR 0.238
DIIF 0.876

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING IN RESEARCH TRENDS
VOLUME 2, ISSUE 6, JUNE 2015, PP 408-410

A review paper on comparision between Traditional and FPGA based storage engine

Abhinav Malviya, Brajendra Singh Rajput

Abstract— — As the number of users increasing, the size of database is also increasing day by day. On a large database processing of a query becomes complex. The traditional storage engine takes a lot of time to process a query and also consumes huge amount of energy. After the FPGA based storage engine introduced the query processing becomes easy, takes very less time to process a query and also consume very less amount of energy as compare to traditional storage engine. Handling of huge amount of data is one of the challenging task. In this paper we have presented comparision between the traditional and FPGA based storage engine..

Index Terms — DBMS, *FPGA, Hybrid, Ibex,* Storage engine, Xilinx.

1 INTRODUCTION

A Database storage engine is a part of the DBMS, which is use to write, read, delete, update from the database. The database storage engine is a software running on DBMS server. As increasing the size of the database and the complexity of query an opportunity for hybrid, FPGA-based smart storage engine to improve database performance. FPGA-based smart storage engine simply put between the disk and the database server. So work of FPGA is to filter the query and send the less amount of data to the database server. This improves the performance of the database server, because FPGA-based smart storage engine is filter the data that is transfer from disk to database server.

2 TRADITIONAL STORAGE ENGINE

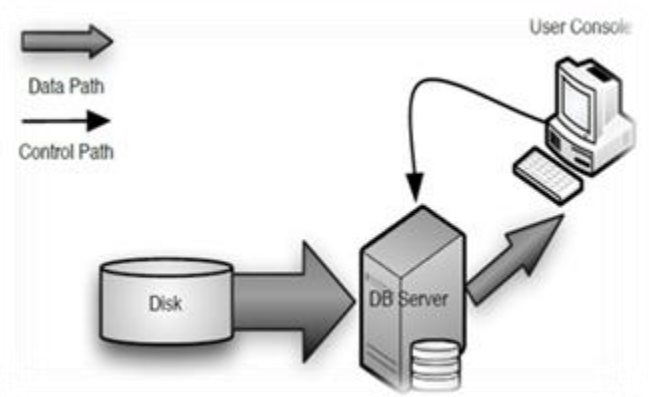

Figure 1 **:** Architecture over view of Traditional Database System[1]

In the fig. 1 shows the traditional architecture of the Database System [1], In which all the component of the database system including the storage engine are integrated into the database server(s).The storage engine work is to read from and write to data from the disk. So there are two bottle- necks on the performance of this architecture describe below:

1. In this architecture the storage engine is a part of database system, which shares the resources with the query optimizer and executer, So the time of query execution can be affected by the tasks perform by the database storage engine like writing and reading data [1].
2. The gray arrow width of the data path in figure 1 shows the volume of the data. The huge amount of data can slow down the certain volume intensive operations [1].

3 TYPES OF TRADITIONAL STORAGE ENGINE

InnoDB storage engine: This is the default storage engine of MySQL 5.5.5. InnoDB is ACID compliant and also safe transaction based storage engine and also support commit, rollback and crash recovery for protecting the data from user. It is based on row level locking and also support FOREIGN KEY referential integrity constraints to maintain data integrity[5].

MyISAM storage engine : This is also default storage engine of MySQL. It is based on the older (and no longer available) ISAM storage engine but having more extensions which are very useful. Each MyISAM table is stored on the hard disk with three files .frm[6], .MYD[6], MYI[6]. The .frm file stores the table format. .MYD is the extension data file and .MYI is the extension of the index file[6].

IIBMDB2I storage engine**:** IIBMDB2I is design as fully featured transaction- capable storage engine that enables MySQL to store its data in DB2 tables running on IBM [7]. Data can be share between the applications of MySQL and applications coded for native DB2 for interfaces by using IIMDB2I[7] storage engine. Also support FOREIGN KEY constraints and full crash recovery.

FEDERATED storage engine: By using FEDERATED storage engine we can access the data from a remote MySQL database without using the replication technology. Querying a local FEDERATED[8] table automatically pulls the data from the remote (federated) tables. No data is stored on the local tables.

4 fpga based hybrid storage engine

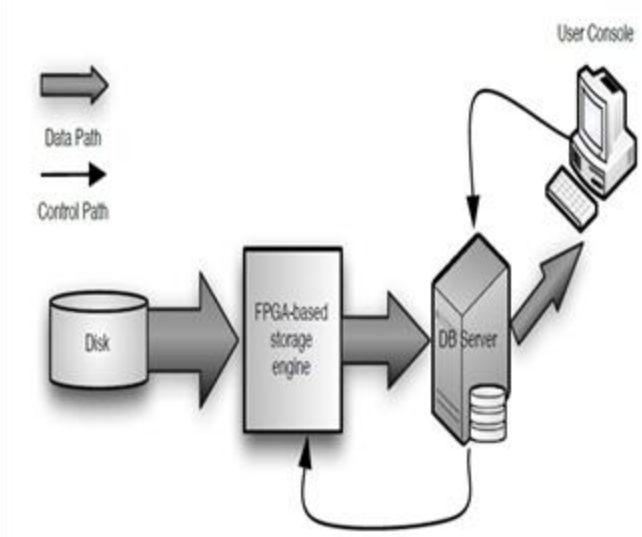

Figure 2 : Overview Of Hybrid Database System [1]

In the fig. 2 shows the architecture of hybrid database system [1], In which the FPGA-based storage engine is inserted into the system's data path, the FPGA is working as a filter which is putted nearby the data source. The tasks of the FPGA-based storage engine are describe below:

1. the beginning of the query step FPGA is filter the data as much as possible.
2. Send only essential data back to the database server.

5 Comparison between traditional and hybrid storage engine on the basis of CPU utilization and power consumption

In Ibex, an FPGA is inserted between disk and CPU into the data path as a query off loading engine, operating on the stream of data towards the query processor[2]. The amount of data going to the CPU is reduced, thereby increase performance and decrease the power consumption at the same time[2]. The performance impact of query off-loading, which measure directly in MySQL and comparing Ibex with other storage engine such as MyISAM or INNODB represent new approaches for evaluating of complex WHERE clause, as well as GROUP BY aggregation queries in hardware. The power saving due to query off-loading, by using the electric meter.

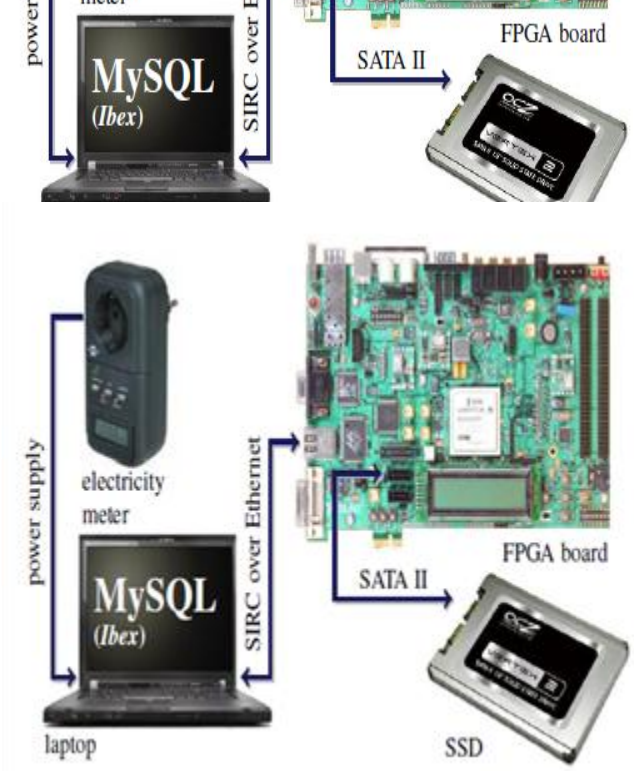

Figure 3: Hybrid MySQL Server Setup[2]

Fig. 3 shows the demo of the setup. The MySQL database is installed on the system. Tables are stored on the system harddisk, whereas Ibex tables are stored on a same model of harddrive, that is directly connected to the XUPV5 board via SATA II [2]. FPGA and system are connected by the Gigabit Ethernet. The below query is run on a 1 GB table.

SELECT id, sum (Val) FROM table GROUP BY id [2].

Executes in 54.18 sec with MyISAM, 179.28 sec with INNODB and in only 3.71[2] sec with Ibex. For 1GB, 3.71sec[2] corresponds to a throughput of 270MB/s, nearby the maximum speed of the SATA II SSD[2]. During query execution the system consumes 45 watt[2] of power with MyISAM, whereas only 31.5 watt[2] with the Ibex. The FPGA running the *Ibex* hardware is estimated to consume 2.8 watts[2] (Xilinx Power Analyzer tool)

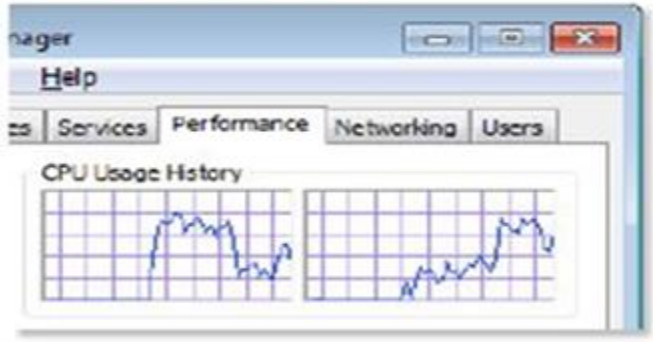

Figure 4: CPU usage during query execution: When using MyISAM as a storage engine[4]

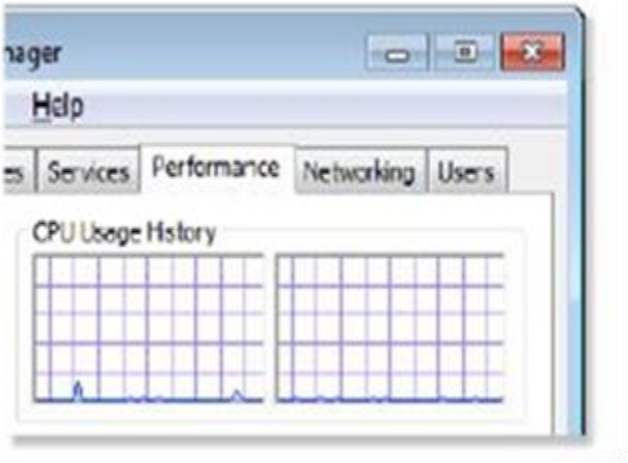

Figure 5: CPU usage during query execution: Ibex used as Storage engine[4]

The Fig. 4 show when we use the storage engine MyISAM than the CPU utilization is very high but in Fig.5 if we use Ibex[4] than the usage of CPU is very less so that cycles are can be used in other works[4].
Energy consumption is also less by Ibex as compared to other storage engine. For GROUP BY query the system needs E=2438 Joules[4] where as E=117 Joules[4] needed if we perform same query on Ibex[4].

6 CONCLUSION

From this paper we have conclude that traditional storage engine fetch the data from the disk and process it on database server. As the size of the data is large it consumes lot of time to process the data. But in case of FPGA based storage engine filters the data before processing over the database server hence it improves the performance of overall system by saving time as well as energy.

.

7 REFERENCES

[1] Chongling Nie, An *FPGA-based Smart Database Storage Engine*, Master Thesis, Department of Computer Science ETH Zurich, Switzerland,2012.

[2] Louis Woods, Zsolt Istv´an, and Gustavo Alonso, *Hybrid FPGA-Accelerated SQL Query Processing*, IEEE 2013, *978-1-4799-0004-6/13*

[3] Louis Woods, Zsolt Istv´an and Gustavo Alonso, *Ibex – An Intelligent Storage Engine with Support for Advanced SQL Offloading*, in proceeding of the 40th International Conference on Very Large Data Bases, Hangzhou, China, *21508097/14/07.*

[4] Louis Woods, Gustavo Alonso and Jens Teubner, *Less Watts, More Performance: An Intelligent Storage Engine for Data Appliances*, in proceeding of the SIGMOD'13,ACM, New York, USA, *978-1-4503-2037-5/13/06*

[5] http://dev.mysql.com/doc/refman/5.0/en/innodb-storage- engine.html

[6] https://dev.mysql.com/doc/refman/5.0/en/myisam-storage-engine.html

[7] https://dev.mysql.com/doc/refman/5.1/en/se-db2.html

[8] https://dev.mysql.com/doc/refman/5.1/en/federated-storage-engine.html