

Research on the Comparison Test of Massive Data Storage in Electric Power Acquisition

Qi Qian^{1, a}, Jianhua Yang^{2, b}, Tao Liu^{1, c}, Shuiping Ding^{1, d}

¹Beijing Kedong Electric Power Control System co., LTD, Beijing, 100192, China

²Central China Power Grid Company Limited, Wuhan, 430077, China

^aqianqi@sgepri.sgcc.com.cn, ^b227262318@qq.com, ^cliutao7@sgepri.sgcc.com.cn, ^d18610271296@163.com

Keywords: mass data storage, distributed database, comparison test

Abstract: In this paper, we study the problem of mass data storage and encountered in its process. We choose D3Base, one of distributed database products, to compare with the Oracle, and do relevant testing work. The deployment of distributed database products and the construction of network environment are also discussed.

1. Introduction

Power information acquisition system is a system for collecting, processing and real-time monitoring of power consumption information, realizing automatic acquisition, anomaly measurement, power quality monitoring, power quality analysis and management of power expenditure information. Simultaneously, this system has the function of power grid information publishing, monitoring of distributed energy, information exchange among the intelligent power devices. Most of the existing power information acquisition systems use data extraction methods (such as Oracle database). But the data acquisition points of China's power grid are large and intensive. In the advanced application development process, if the data extraction method is still used, it is bound to affect the efficiency of the use of power information acquisition system.

The most common query requirement of time series data acquired by electric information acquisition system is retrieving data (precisely, conditionally or fuzzily) in time section. Its search feature is more inclined to the data scan, rather than the best data seeking of the traditional database. For limited archival data, the traditional database just needs to establish indexes in the time column and key in order to realize this kind of retrieval.

When the amount of time series data is too large, indexes are difficult to continue to reside entirely in memory. What's worse is the constant update operations (such as the disorderly insertion) which bring the index fragment trouble. These two points make the traditional database can't scan the target data area in the time of the query (the head needs to jump to address). Therefore, with the increasing and continuous updates of time series data, the query performance of the traditional database will become worse and worse, and it is an exponential decline. Apart from the impact on traditional database indexing mechanism of the two above points, the disorderly data loader meaning that record order is not strictly in chronological order, is a disaster for the traditional indexing mechanism. For example, multi sensor data come from remote transmission, and then concentrate into storage. After this long process, it is almost certain that the order of data storage is not equal to the time sequence of the data. In the case of the disorderly writing of time series data, the traditional database can be caused that the data in the physical storage are also distributed out of order. So even if the time series are built on the index, the query according to the time section is unable to scan disk sequentially and needs to constantly seek in order to locate the data. A large number of tests show that the only sequential read/write performance in the disk is the best, and random read/write performance differs by 1-2 level or more. This paper studies the use of the distributed database system and the solution of massive data storage and use.

2. Distributed Database

2.1 Characteristics of Distributed Database

Distributed database system is a distributed and centralized database system. Its dispersion is that it is usually in the specific application of the environment across the region, and its focus is on its management and centralized unity. Distributed database system is developing rapidly by virtue of its unique characteristics. Its characteristics can be summarized as follows:

Dispersion of physical distribution: Data in the distributed database system are distributed on multiple sites. This is the main difference between the centralized database system and distributed database system.

Logical unity of management: Although the distributed database system is dispersed in the physical distribution, it is uniformly managed by the distributed database management system. This is the biggest difference between decentralized database system and distributed database system.

Autonomy management of the site: Autonomy management of the site is reflected by the ability to manage a scattered site on its own.

Independent existence of data: Although the data are a logic whole, each independent site can have independent data to achieve the transparency of the data.

Associativity of centralization and autonomy: The distributed database system can control the global data, and each site can be autonomous.

Certain data redundancy: Proper data redundancy enhances the ability of the system to recover from failure, and strengthens the performance of the whole system.

2.2 Brief Introduction of Distributed Database Named D3Base

D3Base database system (D3Base-RDB) is a relational distributed database and achieves large-scale data storage through distributing data to multiple nodes. Its main features are high query speed, data loading speed and batch DML processing speed. And the performance can be added linearly with the increasing hardware. It has a very good scalability and is a cost-effective, easy-to-use system supporting high concurrency, linear expansion, fast response and high availability.

D3Base-RDB adopts Massively Parallel Processing and Shared Nothing architecture (MPP + Shared Nothing). Its designing idea is no single-point dependence. The storage capacity and processing capacity of the system can be improved by adding nodes in the MPP architecture. D3Base-RDB is easy to extend nodes and will be able to complete the redistribution of data in a very short time. The host, operating system, memory and storage are self-controlled and have no sharing. D3Base-RDB is mainly composed of master host and data node. D3Base-RDB distributes the data to the data nodes to make full use of IO capability of the data node in order to allow the system to achieve the maximum IO capacity (network bandwidth). Each table in the D3Base-RDB is distributed on all nodes. Master host makes one or more columns of table do hash operation firstly and distributes data to data nodes according to the hash results. During the whole process, the master host does not store any user data, just makes the access control to client and stores metadata.

D3Base-RDB also supports linear extensions, high availability architecture, data and host fault tolerance mechanisms, and data partitioning and compression. D3Base-RDB products can be installed on the x86 architecture of the hardware platform and support good lateral extension. Because D3Base-RDB products have functions of automatic data distribution and load balancing, they are suitable for relational structured massive data storage, analysis and mining. D3Base-RDB provides SQL statement access interface, and provide JDBC, C/C++ API, ODBC and other program interfaces.

2.3 D3Base System Components

D3Base basic system architecture is shown below. D3Base includes the master host and data nodes. The master host consist of global transaction manager and task coordinator. The function of master host is managing the data nodes in the cluster and the transaction and transaction response in the database operation. Data node is used as the data physical storage of the database (Figure 1).

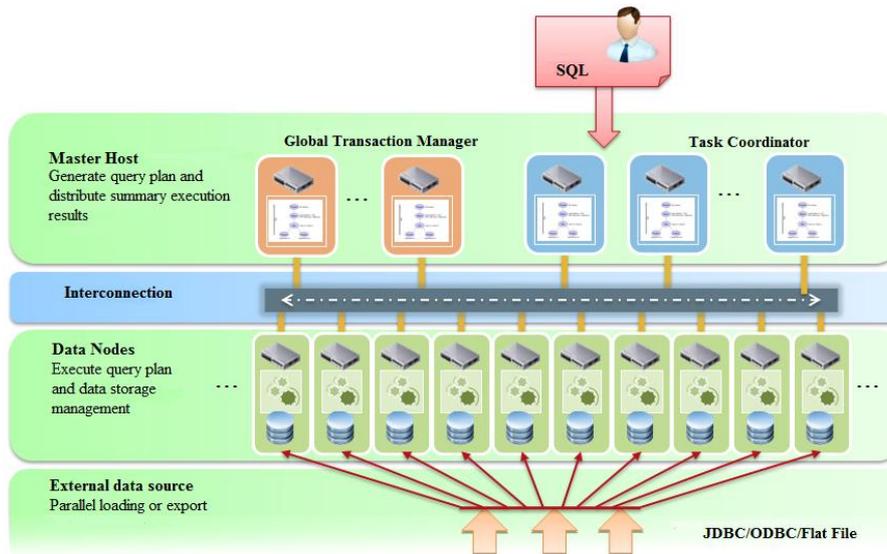


Fig. 1 Composition of D3Base system

D3Base has the MPP mechanism, which allows direct communication between data nodes, the exchange of complex data, the cross-node correlation and querying related data. It can reduce the load of the task coordinator.

D3Base supports multiple coordinators, implementing multiple connection points to the database. The task collaborator executes and analyzes query statement, then generates SQL execution plan. Management is divided into two phases to submit and stores the global catalog information.

D3Base supports multiple data nodes, table data storage and data index. It can complete the actual data storage of the database on data nodes. Data can be scattered and distributed to the data nodes in the cluster. Slice data query can be implemented locally in data nodes and the query can be done in parallel in all related nodes.

Global transaction manager provides a consistent view of transactions. The system performance can be enhanced by adding agent services of global transaction manager.

3. Data Storage and Load Comparison Test

This research project is based on the preliminary performance test of D3Base database server and Oracle11g. The server which is the latest development of the domestic company has less vulnerability and high stability. The client is composed of the user-end and the management. Through long-term comparative study, D3Base is selected as a distributed database system, which can solve the practical problems of storage, load and query of large data.

3.1 Testing Principles and Environment

The comparison test follows the principles like these:

1. Research on query performance and response time, using the same software and hardware environment, without optimization
2. Table structure and data are from the power information acquisition system.
3. Based on the test data of Oracle

The test environment is shown in table 1.

All servers are connected by Gigabit Ethernet switches and IP addresses are in the same network. Test terminal configuration: CPU: T8100, 2G RAM, Windows 7.

The name of the sample data file: YZ_E_MP_CUR_CURVE.csv. Its file size (bytes): 2402675588, the number of records: 7152924 (7 million 150 thousand). Note: The data file is selected by the E_MP_CUR_CURVE table in the online environment of the power information acquisition system. Its export format is CSV format.

Table 1 Test environment

	Oracle server	D3BASE server
Physical configuration	IBM 3650M (2 CPU, 64G RAM, 20T Hardware Disk)	1 IBM 3650M (2 CPU, 64G RAM, 20T Hardware Disk) 3 PowerLeader servers (PR2000R) (2 CPU, 32G RAM, 5*2T Hardware Disk)
Operating system		Combination method is 4*(database + storage)+ 1*Lock controller (concurrent) Red Hat Enterprise Linux 6.5
DBMS	Oracle 11g2R	D3Base-DFM(Management framework)+ D3base-Node(Database node)
IP address	192.168.1.20	Master node: 192.168.1.14; Port number: 9000; Branch node: 11/12/19/21;

3.2 Data Loading and Export Test

Through the respective data loading tool, the sample data files (YZ_E_MP_CUR_CURVE.csv) are repeatedly inserted into the destination table, to form 7 million 150 thousand record tables which are indexed and not indexed respectively. Data loading comparison test results with Oracle and D3BASE are shown in Figure 2-4.

```

Password:
[root@d3base-Node1 oracle]# date;
Tue Aug 18 14:42:51 CST 2015
[root@d3base-Node1 oracle]# sqlldr testuser/testuser@epodb control=/coldata/epodb/sql/load2m.ctl direct=y parallel=y

SQL*Loader: Release 11.2.0.1.0 - Production on Tue Aug 18 14:42:51 2015

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

Load completed - logical record count: 7152924.
[root@d3base-Node1 oracle]# date;
Tue Aug 18 14:45:26 CST 2015
[root@d3base-Node1 oracle]#
    
```

Fig. 2 Loading data file: 7 million 150 thousand records, Oracle takes 155 seconds

```

Input file(s) will be read from Bulkload root directory : /hotd
ata/d3base/data/bulk
Job description file : /hotdata/d3base/k/job/Job_500.xml
Log file for this job: /hotdata/d3base/data/bulk/log/Job_500.lo
g
2015-08-18 13:42:52 (309) INFO : successfully loaded job file /
hotdata/d3base/data/bulk/job/Job_500.xml
2015-08-18 13:42:52 (309) INFO : Job file loaded, run time for
this step : 0.372424 seconds
2015-08-18 13:42:52 (309) INFO : PreProcessing check completed
2015-08-18 13:42:52 (309) INFO : input data file /q
c/cur_curve.csv
2015-08-18 13:42:52 (309) INFO : PreProcessing check completed
2015-08-18 13:42:52 (309) INFO : PreProcess completed, run time
for this step : 0.773548 seconds
2015-08-18 13:42:52 (309) INFO : No of Read Threads Spawned = 1
2015-08-18 13:42:52 (309) INFO : No of Parse Threads Spawned =
3
2015-08-18 13:43:35 (309) INFO : For table epodb.ye2m.cur_cu
rve_16k: 7152924 rows processed and 7152924 rows inserted
2015-08-18 13:43:35 (309) INFO : Bulk load completed, total r
un time : 44.8014 seconds
    
```

Fig. 3 Loading data file: 7 million 150 thousand records, D3Base takes 45 seconds

Environment: Oracle stand-alone IP20-1400 million data export			
SQL statements	Execution times	Time (seconds)	Notes
<pre> set echo off; set feedback off; set heading on; set verify off; set trimspool off; set pagesize 0; set term off; set linesize 150; spool; /coldata/epodb/sql/OraDump.csv; select * from CUR_TEST; spool off; </pre>	1	504	1. The related SQL statements are placed in SQL scripts and are executed in the SQL-PLUS environment. 2. Export 14 million 300 thousand records.
Environment: Oracle stand-alone IP19-1400 million data export			
SQL statements	Execution times	Time (seconds)	Notes
<pre> select * from test_15k into outfile '/coldata/d3baseDump.csv'; </pre>	1	552	1. Execute data export commands in the D3Base SQL interface. 2. Export 14 million 300 thousand records

Fig. 4 Data export performance comparison

In aspect of data export, D3Base is slightly slower than the Oracle. For the export of about 14 million records, both methods spend the same time and D3Base falls behind about 10%.

3.3 Simple SQL Query Test

The operation performances of the simple SQL statements for 200 million, 800 million, 1 billion 600 million and 2 billion 400 million data are compared. The query properties of the two methods are compared, and the results are shown in Figure 5- Figure 9:

Number	Test Table	Oracle Execution Time						D3Base Execution Time					
		No Index			Index			Stand-alone(big table)			Distributed(4 nodes)		
		First	Second	Third	First	Second	Third	First	Second	Third	First	Second	Third
1. select count(*) from test table													
1	YZ_E_MP_CUR_CURVE_2C (200 million records)	271.674	272.09	271.67	28.02	27.81	28.38	1.063	0.875	0.984	0.36	0.328	0.328
2	YZ_E_MP_CUR_CURVE_8C (800 million records)	482.719	480.59	478.56	76.59	63.08	37.44	4.046	3.968	4.047	1.078	1.031	1.031
3	YZ_E_MP_CUR_CURVE_16C (1 billion 600 million records)	1450.534	1447.74	1461.83	188.42	187.53	191.63	9.547	9.641	9.562	2.531	3.484	3.531
4	YZ_E_MP_CUR_CURVE_24C (2 billion 400 million records)	2072.196	2057.55	2054.27	270.55	272.22	274.55	10.407	10.44	10.52	3.937	5.484	5.562
2. select count(id) from test table													
1	YZ_E_MP_CUR_CURVE_2C (200 million records)	270.347	271.63	293.84	27.61	27.47	27.77	1.875	1.688	1.922	1.594	0.5	0.5
2	YZ_E_MP_CUR_CURVE_8C (800 million records)	478.146	482.08	573.53	63.29	67.89	36.73	7.8	7.797	7.891	3.656	1.656	1.687
3	YZ_E_MP_CUR_CURVE_16C (1 billion 600 million records)	1452.589	1452.09	1480.61	187.71	189.13	189.58	17.703	17.91	18.13	4	9.532	9.79
4	YZ_E_MP_CUR_CURVE_24C (2 billion 400 million records)	2060.878	2053.89	2397.69	273.89	266.91	274.16	19.7	20	20	6.313	14.81	14.86
3. select count(id) from test table													
1	YZ_E_MP_CUR_CURVE_2C (200 million records)	274.933	276.42	325.84	0.03	0.03	0.03	1.265	1.266	1.328	0.735	0.704	0.703
2	YZ_E_MP_CUR_CURVE_8C (800 million records)	501.244	499.67	589.91	0.00	0.00	0.00	4.781	4.578	4.781	2.406	2.438	2.397
3	YZ_E_MP_CUR_CURVE_16C (1 billion 600 million records)	1480.793	1475.31	1480.92	0.05	0.03	0.05	11.39	11.36	11.44	5.766	5.766	5.781
4	YZ_E_MP_CUR_CURVE_24C (2 billion 400 million records)	2069.322	2054.83	2411.5	0.05	0.02	0.02	11.563	11.47	11.49	9.234	9.188	9.219
4. select count(id) from test table													
1	YZ_E_MP_CUR_CURVE_2C (200 million records)	275.259	273.28	323.30	281.30	281.19	328.39	1.312	1.265	1.281	1.265	0.515	0.516
2	YZ_E_MP_CUR_CURVE_8C (800 million records)	505.436	492.38	567.70	729.57	733.25	835.58	5.516	5.484	5.469	2.688	1.687	1.719
3	YZ_E_MP_CUR_CURVE_16C (1 billion 600 million records)	1458.766	1462.41	1701.72	1167.94	1145.53	1366.34	12.422	13.17	13.13	6.125	6.156	5.937
4	YZ_E_MP_CUR_CURVE_24C (2 billion 400 million records)	2070.551	2072.27	2403.81	5231.58	5207.14	5204.08	13.969	14.02	14.08	9.422	9.032	9.203
5. select count(1) from test table													
1	YZ_E_MP_CUR_CURVE_2C (200 million records)	277.654	277.8	329.66	281.79	285.11	334.84	1.296	1.328	1.297	0.813	0.766	0.781
2	YZ_E_MP_CUR_CURVE_8C (800 million records)	481.114	488.67	572.93	747.82	731.66	846.77	4.563	4.61	4.565	2.641	2.593	2.594
3	YZ_E_MP_CUR_CURVE_16C (1 billion 600 million records)	1498.992	1468.88	1687.73	1172.1	1164.63	1370.11	11.265	11.27	11.28	6.359	6.344	6.359
4	YZ_E_MP_CUR_CURVE_24C (2 billion 400 million records)	2099.379	2075.83	2425.02	5233.49	5190.11	5175.66	11.593	11.53	11.56	10.13	10.13	10.14

Fig. 5 Detailed comparison records of the query performance

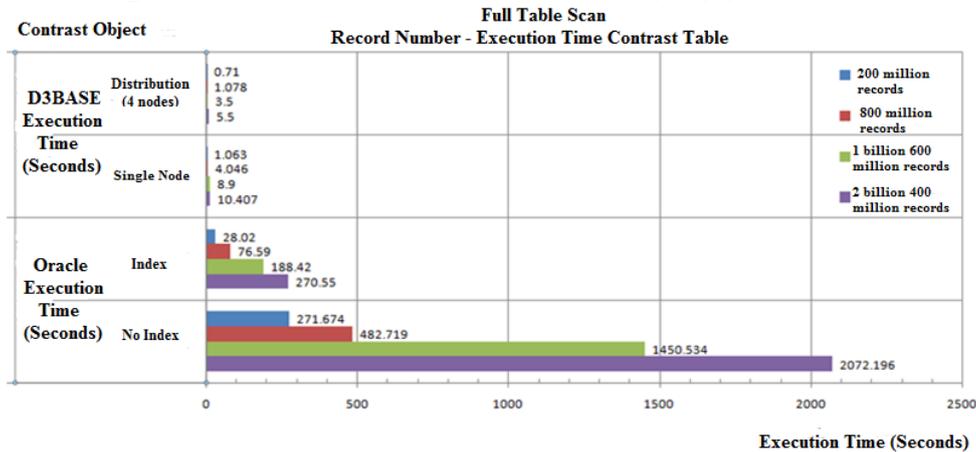


Fig. 6 Comparison of query performance

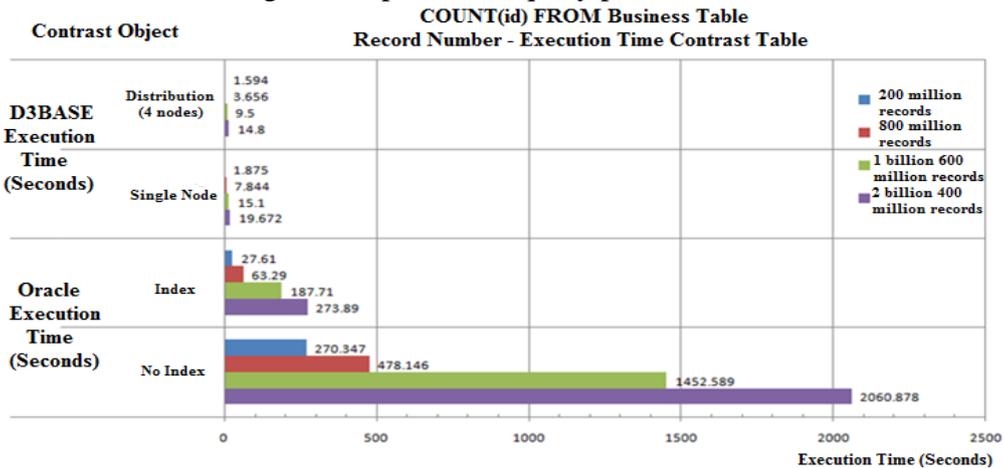


Fig. 7 Comparison of query performance

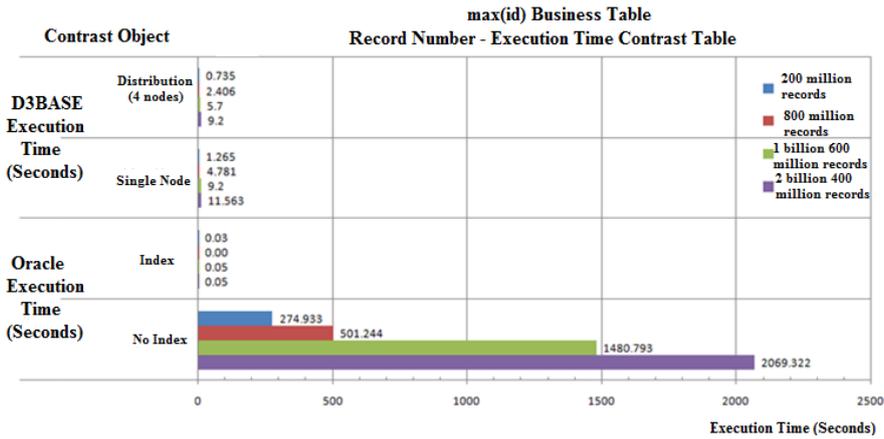


Fig. 8 Comparison of query performance

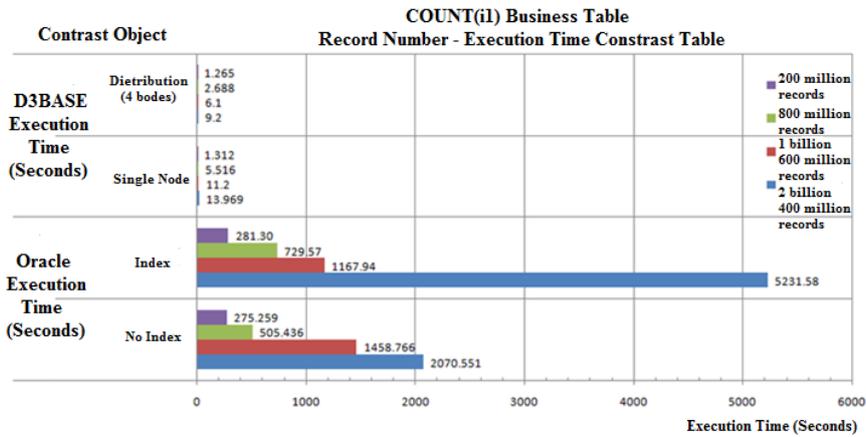


Fig. 9 Comparison of query performance

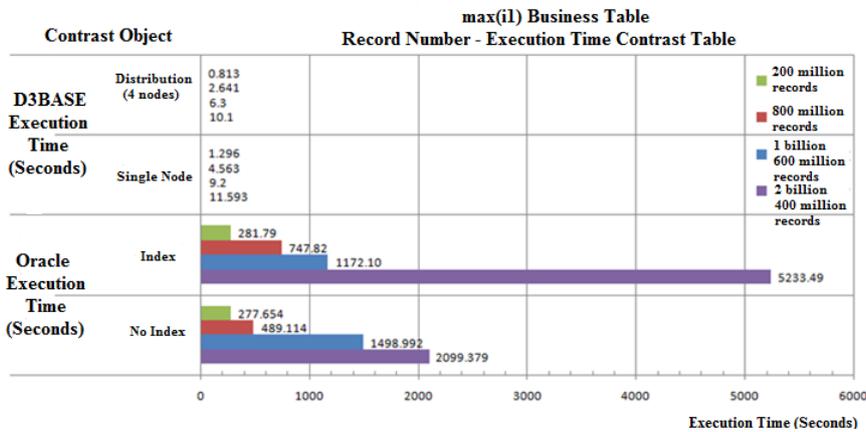


Fig. 10 Comparison of query performance

Comparison test results are summarized as shown in Table 2.

Table 2 Comparison test results

Project	Test Result
Simple query	D3BASE is about 5-10 times faster than Oracle on average.
Count statistics in specified id feild	D3BASE is about 20-30 times faster than Oracle under the index.
Retrieval of maximum value in index id feild	D3BASE is basically equivalent to Oracle in speed, Oracle is dominant.
Count statistics In non-indexed field	D3BASE is about 100-1000 times faster than Oracle under the index.
Retrieval of maximum value in non-indexed feild	D3BASE is about 10-200 times faster than Oracle on average.

3.4 Test Results Analysis

Based on the analysis of the test results, the following conclusions can be drawn:

- (1). In the case of the same hardware and software conditions, D3BASE is 5-10 times faster than Oracle on the performance of integrated query.
- (2). D3Base is not sensitive to the index, which can save a lot of index time and index space.
- (3). D3Base can be deployed dynamically in a number of PC servers and become the clustered distributed database, which is conducive to the full use of hardware resources, achieving flexible deployment, on-demand expansion, and enhancing the purpose of overall performance.
- (4). The four-node D3Base database is about twice as fast as the single-node database (The main reason is that the hardware performances of nodes have a wide gap and the performance of the lowest node responds slowly so as to decrease the overall performance).
- (5). Loading data from the sample file under the same conditions, D3Base 2-3 times faster than Oracle. Faster data loading performance is conducive to the migration of production data.

4. Conclusion and Expectation

D3Base supports the SQL92 standard. NOSQL massive data access storage technology through SQL interface package can greatly reduce the workload and difficulty of application system transplantation.

D3Base can integrate different grades of PC servers into distributed nodes, and make full use of existing resources.

D3Base uses the distributed architecture. The performance and the number of servers increased linearly. System deployment is fast, highly flexible, well scalable. When traversing and retrieving conditionally in a single table with 2 billion records, it can ensure that the response time is less than 3 seconds. In this way, various business capabilities of power information acquisition system in the aspect of massive data processing and analysis can be greatly improved and supported, such as: summary query, etc..

With the expansion of the nodes in the cluster, the performance of the system grows linearly. D3BASE new node implements the true linear extension of performance. This is determined by the D3Base massively parallel processing and the node sharing (MPP + Share Nothing) architecture. Because Share-storage have restrictions on I/O throughput, the performance of other shared-memory databases with new nodes has a ceiling limit and will no longer be promoted. Because of no data synchronization between nodes in the architecture mode where D3Base nodes share nothing, the performance and storage capacity of the system can achieve linear growth with the growth of server nodes in the cluster (Figure 11).

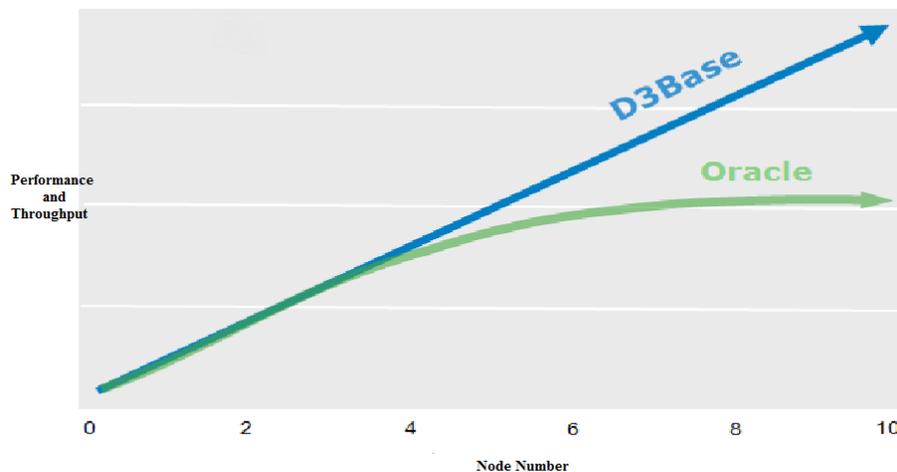


Fig. 11 Performance and Throughput

Using D3Base database, the database platform can be run on multiple PC X86 domestic servers. It can save a lot of maintenance costs, and reduce operation and maintenance, service, procurement costs of the hardware and software.

The solutions of D3Base distributed database platform for massive data have achieved a complete localization from the database to the operating system. It will play a good demonstration effect for our bureau in the aspect of implementation of national information security policy and the application innovation.

References

- [1] Wang Shan, Wang Qiuyue, Liu Fang, Luo Li, Zhang Xiao. Database Research: Achievement and Opportunities Into the 21st Century[J]. Computer Science. Vol. 23(1996) No. 4, p. 1-7.
- [2] Dong Xuanming, Xu Guangyou. The Architecture Reference Model for Distributed Multimedia Systems[J]. MINI-MICRO SYSTEMS. Vol. 17(1996) No. 1, p. 1-6.
- [3] Ouyang Jingwu. Introduction to distributed database systems[M]. Aerospace Industry Press, 1989.
- [4] Shao Peiying. Distributed database system and application[M]. Science Press, 2005.
- [5] Zhu Yangyong. Multimedia database: modeling, storage, and query[J]. Computer Science. Vol. 23(1996) No. 2, p. 70-71,75.