I.J. Education and Management Engineering **2012, 4, 45-53**
Published Online April 2012 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2012.04.08

Available online at http://www.mecs-press.net/ijeme

Study on the Application of the Distributed Report Platform Based on the OSS System

Wang Chunzhi[a], Hu Chang[b], Wang Ruoxi[c]

[a,b]*School of Computer Science, Hubei University of Technology, WuHan, China, 430068*
[c]*Department of Computer Science, HuaZhong Normal University, Wuhan, China, 430079*

Abstract

With the higher and higher qualities which are required by the operators due to launching abundant services all the time, there is an urgent need that the distributed report platform which is based on the OSS system acts as an important application support. This paper aims at making a study of the application of the distributed report platform which is based on the OSS system, designing the three-tier architecture of the whole system of the distributed report platform, analyzing the modules of the whole platform system, elaborating the business process and the communication style, and deploying the application platform.

Index Terms: operation support system (OSS), distributed report, desktop service, application service, database, common object requesting broker architecture (CORBA)

1. Introduction

OSS (Operation Support System) is referred to the telecom operators' integrated support system which shares the telecom resources. It consists of network management, system management, billing, accounts and customer service. The systems are integrated organically by a unified information bus. Each subsystem distributes business reports, accounts reports, settlement reports, audit reports, customer service reports and the reports which are needed by SOX internal audit, etc. Due to the independence of each system, the reports are separated physically and logically so that the reports of the same type can't be shown centrally in only one report.

At present, in each operator's OSS system, reports are just regarded as a part of the production system. The consumption of the host performance when the reporting system is processing will affect the stability of the production system. Especially when the reports are generated at the beginning of the month that is the expenditure presentation date of the operator, the disposition of the reports is intensive. The impacts on the whole system is huge, which will make the whole system running slowly during this period. With the higher and higher qualities which are required by the operators due to launching abundant services all the time[1], there is

* Corresponding author.
E-mail address: [a]chunzhiwang@126.com; [b]wisdomron@163.com; [c]ruoxiwang@163.com

an urgent need of the distributed report platform which is based on the OSS system as an important application support. This report management system which is independent of the production system merges all the subsystems' reporting functions. This system can manage and maintain the reports centrally and diverge the generation of the reports in the production system by data extraction, management of report generating, maintenance management, version management, audit management, presentation management, output management and a series of function modules.

This paper aims at making a study of the application of the distributed report platform which is based on the OSS system, designing the three-tier architecture of the whole system of the distributed report platform, analyzing the modules of the whole platform system, elaborating the business process and the communication style, and accomplishing the disposition of the application platform.

2. System architecture

2.1 Three-tier architecture

At present, operating system generally uses C/S two-tier visiting architecture. As shown in Figure 1.

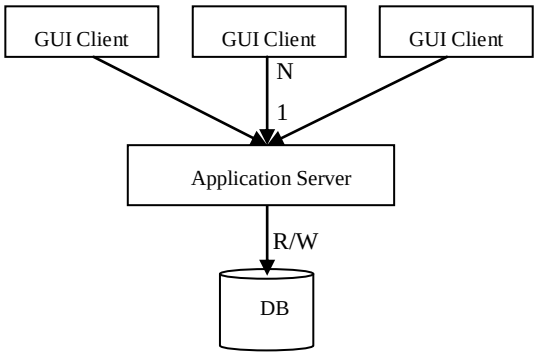

Fig 1. C/S two-tier architecture

In this two-tier architecture mode, every client can visit the R/W (read/write) database directly through the server.

When there are too many clients operating concurrently, the sever will be crowded and the whole system will become slower. Besides, if a client is cracked, the database will be easier to be destroyed.

In view of the problems which exist in the two-tier architecture mode, the distributed report platform which is based on the OSS system will use three-tier architecture[2], as shown in Figure 2.

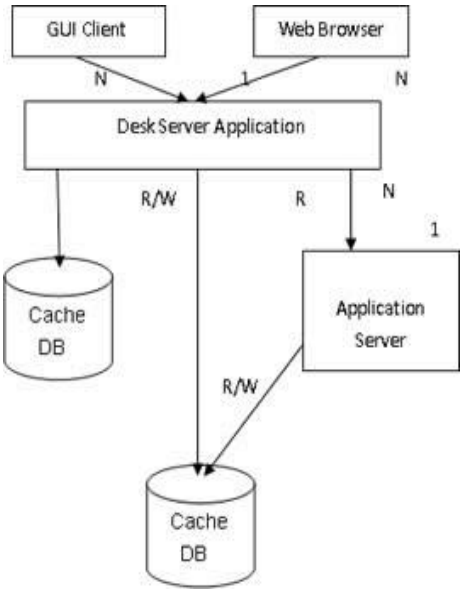

Fig 2. The three-tier architecture

In this two-tier architecture mode, every client can visit the R/W (read/write) database directly through the server. When there are too many clients operating concurrently, the sever will be crowded and the whole system will become slower. Besides, if a client is cracked, the database will be easier to be destroyed.

In view of the problems which exist in the two-tier architecture mode, the distributed report platform which is based on the OSS system will use three-tier architecture, as shown in Figure 2.

In this three-tier architecture mode, each client won't access the physical database directly in order to avoid the risks that database will be destroyed because of the client being cracked. When there are too many clients operating concurrently, they can just operate the cache database. Thereby it can avoid the risks of performance degradation of the whole system due to the client storm. At the same time, it can also enhance the security of the system data. Since DS disposition can support different kinds of accesses, the whole system can satisfy the scalability of client access.

As for process management, under normal circumstances, it can use the least service process to handle the most requests. Thus the start/stop counts will be less. When the amounts of service requests exceed the processing speed of the server, the middleware will let request queue buffer. As for the usage of database connection, it can maintain and reuse database connection. If the service process wants to visit the database, it must link to the database. For instance, open and close the database. The middleware's connection with the database can be maintained and reused by resident service process. Thereby the time and times of connection with the database will be reduced greatly.

2.2 System design

As for architecture, there are access layer, business logic layer, and data logic layer in the distributed report platform which is based on the OSS system. As shown in Figure 3.

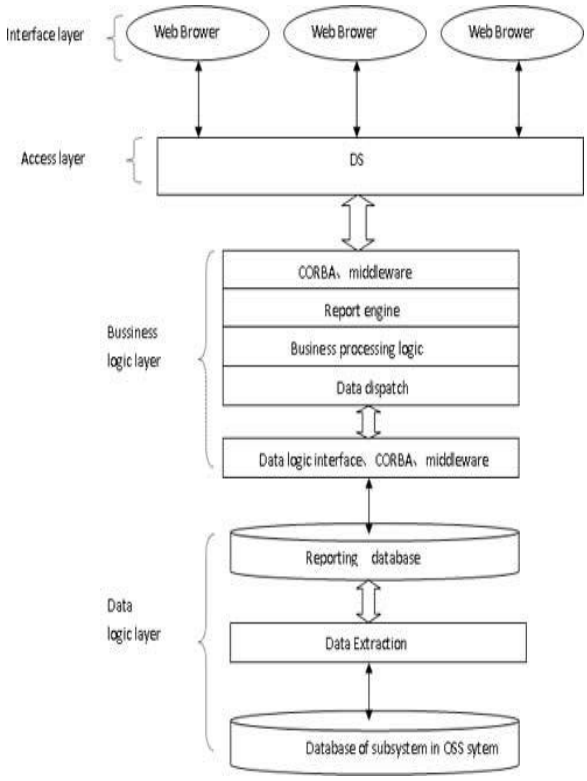

Fig 3. System design

The functions of the distributed report platform which is based on the OSS system's layers are[3][4] as follows:

2.2.1 Access layer

The access layer offers services of clients included. Service fulfillment is shown in the way of GUI in client. Access layer displays various report management, versions of report management, configuration report data, base data management and so on.

2.2.2 Business layer

The report business is operated and report instructions are audited in business layer, covering all the application service and business processing, which is accomplished by component technologies. It packages all the business logic in components, and offers business service by components.

2.2.3 Data layer

The core data is stored in data layer, including summary data extracted from each subsystem to report system and the information of auxiliary tables which are needed by statistics.

Overall, the distributed report platform which is based on the OSS system consists of reporting engine mode, report designer, audit report instructions, full service data extraction tool, report print view tool, report rights management and etc.

2.3 Feature Summary

a) It offers the configuration of distributed platform, the support of character terminal application under UNIX, an open standard interface and C++ interface.
b) It uses C/D/S three-tier architecture mode, offering multi-modal visiting. It can be visited not only by java client but also by web browser as well as supporting large-scale terminal access service.
c) It uses a unique access layer, logic layer and data layer architecture. And the layers are separated, which makes the maintenance easier, performance more stable and data securer.
d) It offers an automated audit reporting tool, which can further assure the quality of report output. Independent centralized deployment can meet the need of stability of production system. Data extraction mode can offer scalability extracting metadata by reports.
e) It uses cluster server deployment, attaining the host configuration easy maintenance and scalability.

3. Analysis on application points

3.1 Main Modules

3.1.1 Report Engine

It uses the principle of workflow engine running. On the basis of raw data, it defines report formats and algorithms. The report engine is operated artificially or timingly based on the defined report formats and algorithms. Users can also adjust the defined languages and report formats of report theme and algorithm according to their specific needs. It can meet the changes of the needs of users. The operation is flexible and simple. It can help telecom operators develop reports quickly at a low costs.

3.1.2 Report Design

It can complete system configuration of report query scripts, output formats and algorithm definitions. It consists of four working windows, including report data window, which is used to set up the size, position, order, style, group summary of the elements shown in the report; data source window, in which users can chose corresponding data sources, data source alias, whether to establish a DBLINK to database or not; window of dynamic conditions, in which users can define dynamic queue, condition queue, whether to convert to Chinese; and preview window, by the interface of which users can test whether the definitions in the design are correct or not.

3.1.3 Report Instructions Auditing

In order to ensure the accuracy of the data, the management system of distributed report offers a function of auditing data index automatically, which will check the reports outputted by the system one by one. It makes the report data consistent by auditing comprehensively. Vertical mode can audit the report data of the prior period or the same period in previous years. Horizontal mode can cover the data related to other reports (such as book data, certificate data). The automation of report auditing increases the efficiency of report auditing and reduces the workload of report auditing.

3.1.4 Full Service Data Extraction Tool

In order not to affect the production system in which the data source is, the system makes the data extinction tool extract the data in the subsystems of the production system into independent report statistics database and then summarize the data. In the process of data extraction, it uses multi-process technology, which will make the data extraction safe and fast.

3.1.5 Report Print View Tool

It links to the report engine objects and uses open sourced IREPORT control. When users want to view the report, they can print the output page and preview the print. After report browsing users can print and export the data.

3.1.6 Rights Management of Report

As for daily management of report contents, the system starts using roles to manage users' rights, ensuring the data safety strictly when all levels of users are inquiring and designing reports. It uses the access control system the same as the OSS system, which can allow the operators of the OSS system to use their original accounts, which report projects operators can view, depends on their rights. The rights can be configured dynamically. Therefore, it can conveniently make operators of different levels view different report items, protecting the security of data.

3.2 Business Process

Reports collect raw data from OSS system to process. The data contains daily processing data and data after accounting treatment. The OSS system uses database propagation software by the third party to replicate data. It can support replicating data according to tables or on demand. Business process is shown in Figure 4.

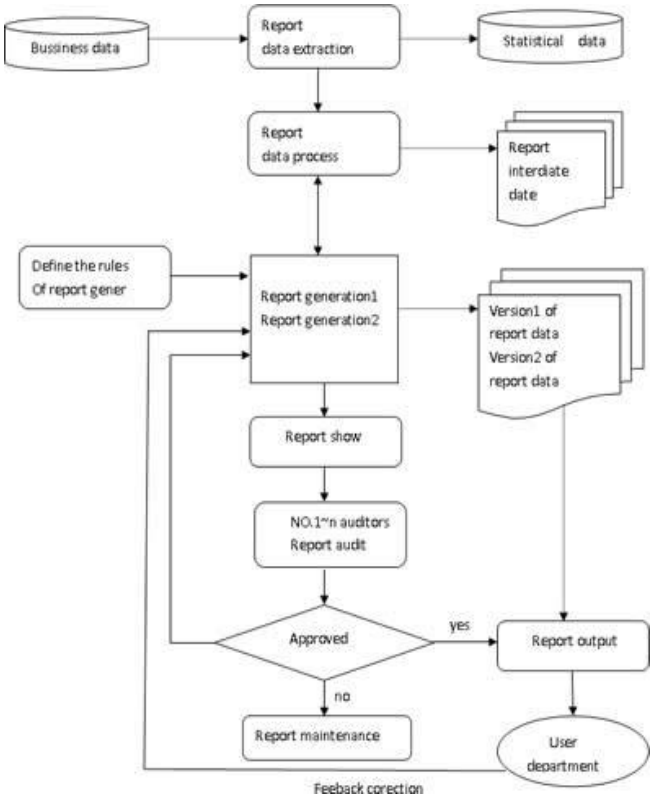

Fig 4. Business process

3.3 Communication Style

It adopts TCP between DS and client protocol. It uses CORBA communication between DS and AS. DS and AS use middleware to visit the database, as shown in Figure 5. CORBA (Common Object Request Broker Architecture) is the framework of OMG organization for processing and integration of distributed objects, which is used in distributed heterogeneous environment. It is transplanted, reused, interlinked and object-faced platform of software development and application. It aims at developing the rules of distributed object technology and object management, establishing common integration framework of application system, and making it reused, transplanted and interoperable on the basis of object software. Due to the features that simple object model, language independence and distributed transparency of CORBA, the whole system is independent of any operating system platform. The use of database visiting middleware makes the database operating easier.

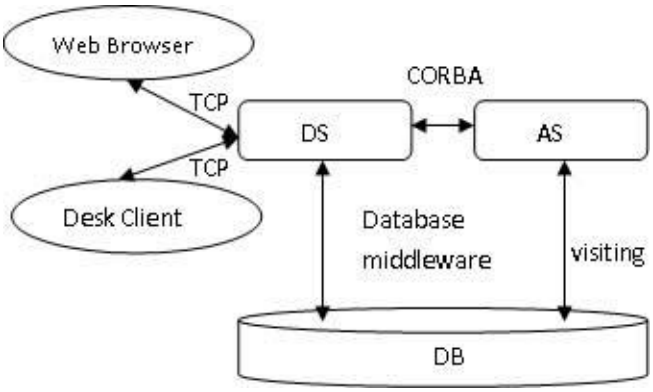

Fig 5. Communication style

3.4 Application Deployment Environment

The distributed report platform which is based on the OSS system is deployed by host affinity: desktop application services host (1~n), application server host (1~n), middleware host, database host and different kinds of client host, as shown in Figure 6.

Web browser client and desk client may be deployed in different places. DS servers are deployed by multi-machine backup, which serve all the clients with the same and unified interfaces and access to host application by the same middleware in order to meet the demand that large number of customers wanting to access to it and protect the application host. Even if one of the DS 1 servers are physically damaged or breakdown, it can be substituted for other DS 2, DS 3……DS n immediately so that the server can run normally. Besides, maintainers can add or reduce the numbers of DS host servers in order to lower the costs of DS host and secure the maintenance of host performance, and enhance the DS server's maintainability and scalability.

As host is deployed by parallel server cluster, serving the DS with unified interfaces, backing up each other, implementing the same service, and reducing the applicable response time[6]. Besides, every as server also has fault-tolerant tasks. When one or more than one servers are at fault, the system can separate the server from it at the back of the system software. And it completes the new load distribution by every server's load transfer mechanism. Other servers immediately undertake the service tasks. Thus without manual intervention, the system can serve automatically.

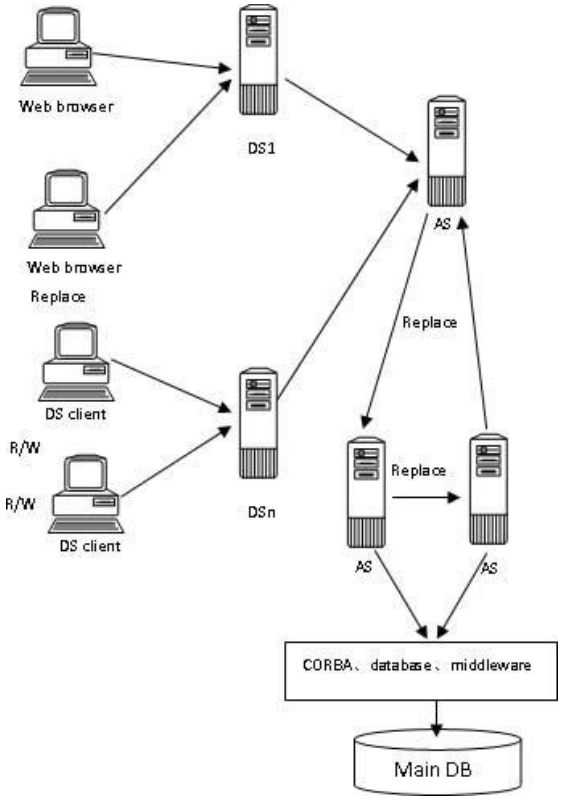

Fig 6. Application deployment environment

3.5 Platform Advantages

The advantages of the distributed report platform which is based on the OSS system are summarized as follows:

3.5.1 High system maintainability

The open system is of high maintainability, and the report query and display can be configured. Adding and modifying reports can be completed in the configuration interface with no additional development process; supporting simple SQL queries and complex logic of the spin-off set Package; in order to settle the efficiency problem involving a large amount of data, we can collect all the data to the middle table and then allocate the SQL query and display mode programs; the acquisition of source of the data has already been achieved, all you need to do is to extract data from SQL and the program will become automatic; it supports different kinds of script configuration.

3.5.2 Low coupling and easy to deploy

Report management system uses the independent host and database in its physical deployment, thus having no impact on the existing running system; improvement against a subsystem reporting can quickly be applied to other subsystems, and the loosely coupled relationship with OSS system can meet the heterogeneous OSS systems and be promoted to a large extent.

3.5.3 Full opening service management

Each subsystem can be extracted from extraction program to independent database[7], coupled with a low degree to the operational subsystems. Reporting system can share integrated operator management with other OSS subsystems; Administrators can enter the report system, checking the reports of its all subsystems; it also supports the seamless integration of J2EE and SOA.

3.5.4 Intelligent data auditing

After the original extraction, the system automatically audits each data; manual and automatic real-time scheduling audits guarantee the correctness of statements and save the enormous amounts of human and material resources; audit results can be informed by various ways such as Email, SMS, page news alerts, etc. any auditing projects are fully equipped, and increasing or modifying of the audit program can be completed within the configuration interface without the need to modify the program.

4. Conclusion

This paper researches on the application of the distributed report platform based on the OSS system, which includes the three-tier architecture of the whole system, the analysis of every module sectors in the platform system, the research of its business process, the communication style and the deployment of the platform. Finally the distributed report platform based on the OSS system will act as the support of the OSS system is reached.

References

[1] Yi ling, “Telecommunications enterprise information construction in the development of BOSS”,BEIJING, GUANGDONG COMMUNICATION TECHNOLOGY, 2004 24(10)Lu
[2] Shao Honggang, “New Data Exhibition Solution for Telecom Carriers' OSS”,Beijing, 2006 (12)
[3] Cui Lei , “IP-Based OSS Management Solution”,Beijing, MODERN SCIENCE & TECHNOLOGY OF TELECOMMUNICATIONS, 2002
[4] Hanhua ,Wang Yashi,and Min Lijuan, “OSS/BSS Framework Based on NGOSS” ,TELECOMMUNICATIONS SCIENCE.Beijing, 2009 25(10)
[5] Caobing,”Plan for Constructing Telecom Operation Support System”,Guanzhou, GUANGDONG COMMUNICATION TECHNOLOGY, 2003 23(6)
[6] Wang chenwei,Shen chenwie,“EPON network service control and management of technology”,Beijing, ELECTRONIC TECHNOLOGY, 2007 (9)
[7] Frost&Sullivan,”Discussion on the application of OSS in communication enterprises”,Haerbing, HEILONGJIANG COMMUNICATIONS TECHNOLOGY, 2002 (3)