

The Study of Multi-Tenant's Configurability

Dong Wang*

Dalian Maritime University, Dalian City, Liaoning, 116000, China

Abstract: SaaS (Software as a Service) the application mode occurs exactly along with the broadening and convenience of network platform, its core is the application of Multi-Tenant. The Multi-Tenant under SaaS not only makes Application Software more efficient and convenient, but also reduce the cost of software development, hardware purchase, training and upgrade maintenance, which can relieve the financial pressure of enterprises invisibly, so that enterprises can focus on the business development. Aiming at the personalized demand of Multi-Tenant's tenement, this thesis studies about personalized configuration of data, function, and operation interface, moreover, it shows the method of personalized configuration. It also indicates how to make the Multi-Tenant come true through personalized configuration, and realizes unified management of these applications.

Keywords: SaaS; Multi-Tenant

***Corresponding author:** Dong Wang. No.1, Linghai Road, Ganjingzi District, Dalian City, Liaoning Province, China. E-mail: ssmile@163.com

DOI: <http://dx.doi.org/10.26549/met.v1i1.326>

1. Introduction

To enterprise users, this kind of "directly applicable skills" of SaaS has huge allure. The users do not need to install internet applications in their computers or servers. The service supplier of SaaS undertakes the maintenance and upgrade of SaaS, the users will not spend energy. The "use as needed" mode enable the clients to only pay for the functions which are used, and do not need to waste money on the other functions. The operating environment of SaaS is based on the universal web browser, so the users can use the service to work on business on internet at any time.

2. Personalized Configuration of User Data

2.1 Database Extension

2.1.1 Application Mode of Database

1) Independent database mode

At present, the database project of customized develop-

ment is of independent database mode, that is to say, there is an application case in a single database, the user cannot enter into it without permission. The demo is as Figure 1. If each individual tenant has an independent database, that would need a huge cost of hardware purchase and independent maintain. This kind of project cannot entirely show the scale effect by SaaS.

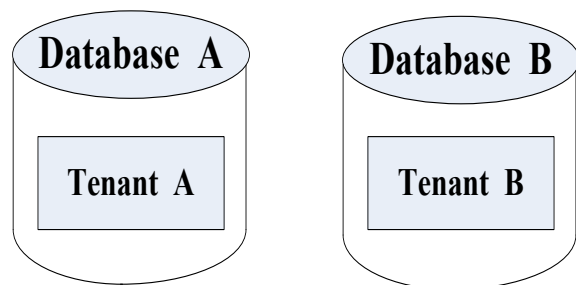


Figure 1. Standalone database model

2) Shared database-independent data

To reduce the cost, the tenants can share one or more databases. But each tenant has his different data structure, to some degree, there exists some Logical data isolation, in addition, every database will support more tenants.

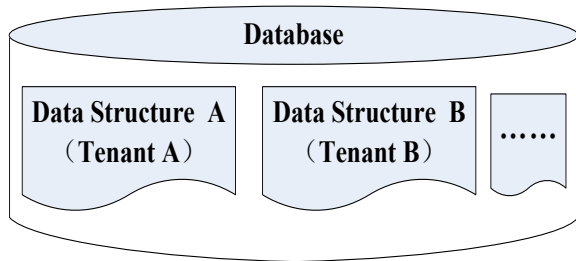


Figure 2. Share database-standalone database model

3) Shared database and data structure

SaaS mode is aimed at rational disposal of resources through Shared maximization. If the tenants not only share database but also share the same data structure, so this kind of project will maximize the sharing.

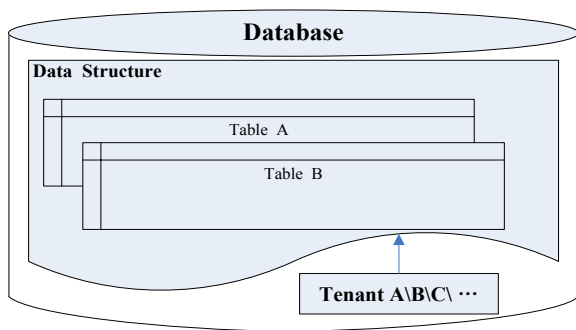


Figure 3. Share database and data structure

Any database mode has its own advantage, the financial enterprises such as bank, security demand of highly degree of data isolation, they can choose independent database mode, because their strong financial support enables them to afford the charge of purchasing and maintenance of independent database. As for medium and small-sized enterprises, choosing shard mode can reduce the cost. The users can choose the specific mode according to their economic strength and the demand of program efficiency and security.

2.1.2 Distributed Database

Along with the more and more users, the storage data should be huger and huger, meanwhile, the demand of handling ability of database becomes more and more strict. When the number of users of a set of online software reaches to a certain degree, it is hard for the database

system on a single PC server to conduct business data, which causes low efficiency. To solve this problem, we can use distributed database instead of the present centralized model.

2.2 Data Structure Extension

Data structure is the most important section in system design, and the configurable emphasis of Multi-Tenant is exactly how to extend data structure.

2.2.1 Customization of Additional Fields

This is a kind of traditional implementation scheme of data structure. When the tenant's demand of data changes, corresponding fields would be added on data table.

2.2.2 Preassignment of Fields

Preassignment of fields is to assign and extend some fields in table in advance, when the tenants need to extend fields, then they can use these preassigned fields.

2.2.3 Extend Association Table

It is to make an additional table for storing the fields which need to extend, meanwhile make a data table for storing extended data. In this way, these related three tables will meet the needs of extension of different tenants.

2.2.4 HDFS (Hadoop Distributed File System)

At present, when the database is storing data file which more than 1TB (1024GB), it uses DAS/NAS/SAN framework generally. The current system needs to record the user's operation log and store them for more than one year. Otherwise, some users' special demand such as oversized video file\ search engine file\ log package, which will pose a big pressure on data storage space and even make it break down. To relieve the pressure, we can distribute the large file into small parts. HDFS (Hadoop Distributed File System), the core distributed file system of Hadoop of Apache Funds includes two sorts of technics: HDFS and MapReduce.

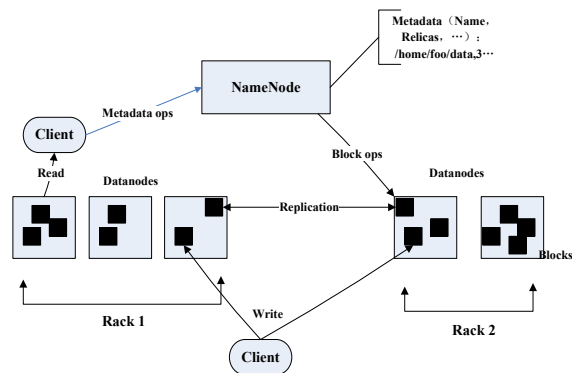


Figure 4. HDFS framework

HDFS file consists of sequence block (the regular size

of HDFS is 64M, a HDFS file can be cut into 128 blocks in maximum, that makes it sure for HDFS to store huge file), each block has many copies, each copy is stored at different data node at different computers in the whole colony. MapReduce cut the adhibition into many small blocks as a working unit, each working unit can coupling treat the copy and collect the data processing results in the meantime.

3. Personalized Configuration of System Function

For the tenants under SaaS mode, after the extension of business, they may need a deeper level of system function or other new functions. If the business is integrated, some current system functions may not be used any more. So how a tenant changes the system function according to business requirement at different times? How could it be possible for different tenants to use different system functions at the same time? To solve these problems, SaaS should be configurable on system functions.

For the users, even the more huge or complicated system consists of all kinds of system functions. If all the system functions are divided into basic function in minimum unit, then these "micro-adjustment" basic function is easier to conduct, and it is easier to collocate the tenants' functions.

3.1 Division of System Function

Complicated system should be divided according to its business. There are some principles of functional division as below:

1) The most important about divided function is that it should be of value for the tenants. No one would buy the function without use value. For example: in a ERP system, the checkout ID function in the order feedback is the minimum function unit. But to the user, the minimum unit

is order feedback, they do not care about the ID checkout. If we subdivide the checkout function, the user would be puzzled.

2) The subdivided function could not be subdivided, which fails to realize minimum;

3) The subdivided functions cannot cross and overlap with each other;

4) The subdivided functions are not allowed for circular reference.

3.2 The Relationships of System Function

The definition of system function is shown as Figure 5, there are two sorts of relationships among functions: depend and include.

4. Personalized Configuration of Operation Interface

4.1 The Meta Model of Operation Interface

Whatever it is users' data or system function, to users, the most directly configurable feelings are reflected in operation interface. The nice User Experience will increase the satisfaction about system. The personalized configuration of operation interface is mainly reflected in two sides: the content of operation web and individuation of menu.

Because SaaS online software allows the tenants have user-defined extensional data, the page element in operation interface may differ in layout or quantity. It will be pretty complicated to configure separately, here, we use the idea of MDA (Model Driven Architecture) which is based on metadata management, manage the configuration parameter included system function menu and page display element.

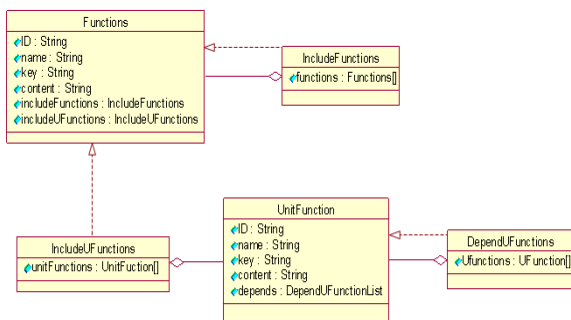


Figure 5. The definition of system function

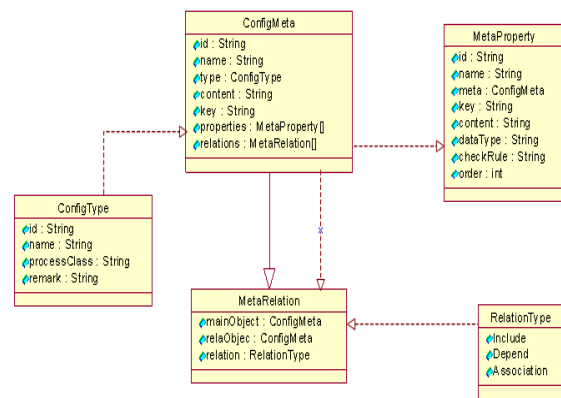


Figure 6. Configured metadata class diagram

In Figure 6, ConfigMeta class is for expression of some general attributes which need collocation point; ConfigType class included all kinds of collocation points; MetaProperty class is for description of extended attributes of collocation point.

4.2 Operating Mechanism

The operating mechanism of operation interface based on meta model is as Figure 7.

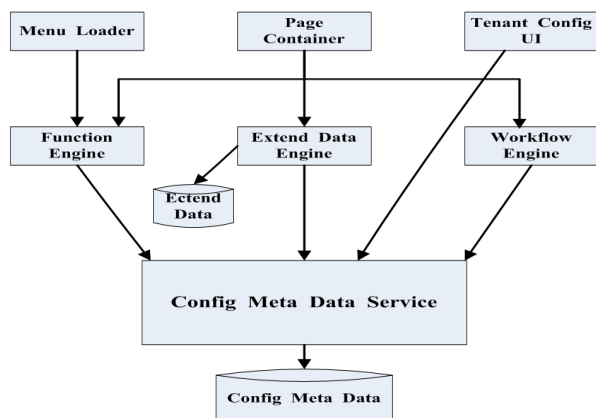


Figure 7. The operating mechanism of operation interface that is based on meta model

Menu Loader is used for loading tenants' personalized configuration menu, it needs Function Engine to load menu and function unit; Page Container uses Extend Data Engine to load users' personalized configuration page elements; Tenant Config UI supply unified page layout to the users that need personalized page layout. Function Engine control menu, function and the loading and display of page elements through searching tenants' pre-configured function. In the end, it maintains the system operation through unified Service Interface from metadata.

5. Conclusion

Today as the age of Internet information develops rapidly, because of all kinds of development factors, SaaS mode is superior to traditional software customization mode. And the Multi-Tenant under SaaS is a core problem, this thesis is just based on such background, analyzing and studying the configurable attribute under Multi-Tenant. This thesis mainly studies three sides of personalized configuration under Multi-Tenant as below:

- 1) Personalized configuration of user data.
- 2) Personalized configuration of system function.
- 3) Personalized configuration of operation interface.

In section 1, we introduce how to make separate and personalized configuration of user data through database and data structure, and show how to solve the pressure of huge data through HBase distributed database. In section 2, we use the subdivided function to combine needed function and meet the demand of customized function through checkout the user ID. In section 3, we import the concept of meta model, manage the configuration parameter generally in the form of metadata, and make personalized display and conduct through united metadata flat interface.

References

- [1] StormPlayer. Baidu Baike[EB/OL]. <http://baike.baidu.com/view/36>. 2017 [2017-3-15].
- [2] SaaS. Baidu Baike[EB/OL]. <http://baike.baidu.com/view/369107.htm>. 2016 [2017-3-15].
- [3] Shichang Jiang, Shengwen Zhang. Research on the Page Element Configuration of Multi-Tenant Applications[J]. Computers and Modernization, 2010, 1(8):105-108.(in Chinese)