The Design and Implementation of Railway Freight Applications on China's Railway Freight Data Grid Platform

Honghui Li, Yan Zhang, Feng Liu, Gang Dai

Abstract

Railway Freight Data Grid (RFDGrid) platform applies Grid technology in railway freight information systems in order to manage and share freight data which are distributed at different railway administrations. How the railway freight applications can efficiently use distributed and heterogeneous data and computational resources managed by data Grid platform is the key point to provide high quality freight information services to users. The work presented in this paper elaborates on the design and implementation of railway freight applications on RFDGrid platform. First, the background knowledge of RFDGrid is introduced; and then all railway freight applications are discussed in detail, the three-layer architecture of freight applications' implementation are described and the granularity of components are defined, all the applications are divided into two groups - statistical analysis applications and decision support applications according to the characteristics of processing flows; Finally, an application scenario about statistical analysis task is given to illustrate the concrete implementation of railway freight applications, the execution times on different nodes scale are analyzed.

Keywords: Grid Computing, Railway Freight Data Grid (RFDGrid), Railway Freight Applications

1. Introduction

Railway transportation is the lifeline of China's economic development. Railway informationization is not only an effective way to connotative expanded reproduction and efficiency potential, but also a necessary choice to realize the modernization of China's railway system [1]. The necessary condition for improving railway informationization is all-directional sharing of railway freight information and collaboration among different levels or different departments in the railway system [2]. Due to the original organization of China's railway freight information system, a large amount of information is generated in geographically distributed subsystems. The dispersion inside subsystems and among subsystems makes it impossible for subsystems to connect and interact with each other. So it is difficult to share and collaborate on these data and information via conventional technologies.

In China’s railway transportation information system, three-level management model is adopted, which includes railway freight stations, eighteen railway administrations and the ministry of railway. Related freight information is generated at distributed freight stations and then is uploaded to the railway administrations, and then is uploaded to the central data warehouse in the ministry of railway. At last, all freight data and information are centralized and stored in the ministry of railway. The data is transferred via railway internal network and the bandwidth between railway administrations and the ministry of railway is either 2 Mbit/s or 4 Mbit/s. Because of the limitation of bandwidth, the freight information distributed at different railway administrations are not uploaded to the ministry of railway in real time, so the model cannot provide the real-time statistic and analysis services [3]. Moreover, the centralized processing model brings some limitations, such as the distributed data and computational resources cannot be shared transparently, computing capability distributed at railway administrations cannot be used in balance, which further limit the extraction of knowledge from freight data.

Railway Freight Data Grid (RFDGrid) platform applies Grid technology in railway freight information systems in order to provide an effective and integrated approach to share and coordinate geographically distributed and heterogeneous resources. Compared with conventional technologies,
Grid is flexible, autonomous, dynamic and service-oriented [4], which attract more and more research on different fields [5, 6]. How the railway freight applications can efficiently use distributed and heterogeneous data and computational resources on data Grid platform is the key point to provide high quality freight information services to users.

In this paper we discuss all railway freight applications that can be deployed on RFDGrid platform in detail, and the design and implementation solution are presented. The rest of the paper is organized as follows. Section 2 describes the background about RFDGrid platform; Section 3 analyzes the railway freight applications and discusses about the implementation outline; In Section 4, an application scenario illustrates how freight applications works on RFDGrid platform, and the execution times on different nodes are analyzed; finally, we draw the conclusions in Section 5.

2. Railway Freight Data Grid

RFDGrid platform aims to process huge amount of freight data which are distributed at different railway administrations to provide freight applications with unified global virtual data views which can shield the distribution and heterogeneity of resources [3]. RFDGrid platform consists of data resources and data Grid middleware, where data resources are freight waybill data and freight acknowledgement data located at 18 railway administrations and data Grid middleware can provide data management functions, including access control, data view management, metadata management, data sources agency, etc.

Railway freight waybill data provides original information for transportation statistics, financial management and wagons flowing analysis. It contains the following information: detailed freight information, wagons information, departure station, destination station, cost information, transportation route, etc [7]. Acknowledgement information is an abbreviation for train marshalling schedule. Usually, a wagon of freight departs the load station and it must be organized many times before it arrives at the destination. The acknowledgement information records how the train is organized every time and it contains the train's information (including train number, the length of train, total weight, marshalling station, etc.) and all wagons' information (including wagon number, wagon type, departure station, destination station, freight name, weight, remark, etc.) [8]. One piece of waybill is saved as one record in the waybill table, and it occupies about 4 KB. Every day an average of 150,000 wagons freight are loaded - so 150,000 records are generated every day. We can conclude that 220 GB (150,000 × 4KB × 365 ≈ 220 GB) waybill data is generated every year. The data quantities of acknowledgement information are in the same level [9]. All freight applications need to execute computational tasks on these data.

In order to provide the freight applications with data related core services, RFDGrid platform uses two key technologies [10]:

- Virtual global data view. Virtual database system is built and all distributed data resources can register on data Grid middleware. The unified, virtual, integrated data views are formed without the movement of physical data locations. The upper freight applications can access virtual database by standard query language and then get the aggregated result.
- The decomposition and execution of queries on large amount of data. In order to response the request of upper freight applications, the data Grid middleware can obtain the efficiency of query processing using many algorithms and techniques, such as decomposing global query into many subqueries which can be executed on local physical databases and assigning computational tasks to nodes where data is located. The decomposition and scheduling are transparent to the freight applications.

3. Railway Freight Applications

Railway freight applications aim to provide statistical analysis functions and decision support functions to meet the demands of users at different departments. All routine functions, algorithms as well as some existing specialized tools are encapsulated as services and are deployed on RFDGrid. All freight applications can be classified to six groups according to the important fields of applications planned by the Ministry of Railway [1], as shown in Figure 1.
Transportation organization aims to improve the efficiency of transportation production and it covers all steps of transportation; Income statistics aims to guarantee the optimal allocation of railway transportation resources and decrease the transportation costs; Transportation marketing aims to improve the market competition ability of railway transportation and increase the income of transportation; Tracking services aim to grasp the position information of all the wagons and freight during transportation and it is the foundational function; Public information services aim to provide better services to the owners of freight and lay a good foundation for the construction of railway electronic business platform; Logistics information sharing aims to build the data sharing platform to realize the data exchange between railway system and harbors, customs and relevant government monitoring departments.

There are 2-6 subgroups under every group. The final functions are classified into these subgroups, such as the shadow box in the transportation statistic subgroup and there are 141 functions altogether.

![Figure 1. The Freight Applications on RFDGrid](image)

### 3.1 The Implementation Solution

The three-layer architecture is adopted in the design and implementation of railway freight applications, as shown in Figure 2, which includes presentation layer, services layer and component layer.
Component layer consists of basic components and business components. Basic components are finer-grained operations, including data extraction, data transformation, data aggregation, data transfer, file operations and data mining operations; business components are coarse-grained and they are business flows which are composed of a sequence of basic components and can provide complete business functions, for example, when analyzing the flow direction of loaded cargo, four types of basic components - data extraction, data aggregation, data transformation and report display are needed. Many business functions can use one business component when they have the same business flow. Data access related basic components can use unified interfaces provided by RFDGrid platform to access data resources, and the whole process is transparent to component layer. The loose-couple relationship between data Grid layer and applications layer can make components layer independent on specific data Grid middlewares.

Service layer wraps all business components as services and provides unified web services interfaces to presentation layer and other applications. The service-oriented access can separate the change of implementation of components from the users access interfaces, on the other hand, users in railway system can access all business functions by B/S structure.

Presentation layer provides user-friendly GUI to users and freight applications services are invoked by standard web services interfaces, and the result of request can be displayed as reports or charts. Other applications can invoke the services and plug the result in their systems. During the implementation of freight applications, basic components are the key modules. The present basic components can be divided into six groups. With the increase of freight applications, new basic components can be added easily. The present basic components groups are the following:

- **Data extraction** can extract freight data from RFDGrid by the interfaces and provide data access functions to other basic components. The execution process is invisible to users, for example, *GlobalVirtualQuery* component can execute query on the target virtual view, and the input is a query statement and output is data record set (a list of tuples); *ExtractViewSchema* component can get the schema of the view, and the input is the resource name and the output is the schema information of view.

- **Data aggregation** can aggregate the input record set by columns or rows, for example, *ColumnAggregation* component can aggregate values by columns, and the input is a record set and the output is a new record. Because the statistical data often need to aggregate freight data based on 18 railway administrations, all statistical analysis tasks need the aggregation components.

- **Data transformation** can transform the data type, data format and data unit, for example, when counting the number of loaded freight wagons, the unit of result is wagon at railway administrations, but at Ministry of Railway, the unit of result is ten thousand wagons, at this time, transformation is necessary in order to get the correct result.

- **Data transfer** can transfer data among the nodes by FTP, HTTP, or bulk load data into data resource located at different nodes, which include files transfer, data record set transfer, etc.

- **Data mining** can provide components used in data mining algorithms, this type of basic components is achieved by specific middlewares.
components are mainly used in decision support tasks, for example, \textit{AllNonEmptySubsets} component aims to generate all non-empty subsets of a set and it can be used in association rules mining and sequential patterns mining.

- **Reports load** can load input record set into specified report format, for example, \textit{ReportZcqxtj} component can generate the report for the flow direction of loaded wagons, and the input is record set and the output is the report object which can be transformed as XML file and every value can be accessed directly.

The implementation of every freight function is a process model which is composed of a sequence of basic components, and the execution engine can schedule the execution of process models. Figure 3 shows the execution process of a freight application service, from service to basic components. Here, users or other applications send request by invoking freight services, and in the implementation of services, the requests are converted to business components which are business flow models, and then the business flows are scheduled by execution engine, and during the execution, basic components are invoked.

![Figure 3. The Execution Process](image)

There are 141 freight function modules, and they can be divided into two groups according to the characteristics of the flows, statistical analysis applications and decision support applications. The following two subsections describe them respectively.

### 3.2 Statistical Analysis Applications

In statistical analysis applications, the statistics based on all freight data are composed of statistics on 18 railway administrations, which means that every railway administration makes statistics based on local data and all operations and related data attributes are same, and these local statistics result are aggregated to generate the final result at Ministry of Railway. The outstanding feature of this kind applications is that the processing flows are static, as shown in Figure 4, where data extraction component access the virtual data views to get local statistic results from 18 railway administrations; theses intermediate result are aggregated; and then the aggregated result are transformed and processed; at last, the transformed data is loaded in report object and the report is displayed in a specific format. In railway freight information system, more than four fifths applications modules are classified as statistical analysis tasks.

![Figure 4. Statistical Analysis Applications](image)
3.3 Decision Support Applications

Decision support applications aim to extract useful knowledge from large amount of accumulated freight data, and many data mining algorithms are involved, so it is inevitable for these tasks to occupy more computational resources and take longer execution times. This type of tasks can provide data mining functions, such as predication, rules discovery and classification, so they are very important to decision-makers. Compared with statistical analysis tasks, the business flows of decision support tasks are dynamic, because different data mining algorithms are composed of different data mining basic components. Moreover, the business components of decision support tasks are the data mining execution process models, and during the schedule, the execution process should be optimized according to status of nodes.

4. Application Scenario and Experiments

The section describes a typical freight application on top of RFDGrid platform, namely statistic of the flow direction of all loaded wagons. It belongs to the transportation statistics subgroup of transportation organization. It is the gray box in Figure 1. In this example, we do not set arguments values and it means to make statistic about all types of wagons and all types of goods. This means we will calculate:

for every railway administrations
among the total wagons loaded on its stations
from 1-1-2010 to 12-31-2010,
how many wagons and tonnage are transferred to
the other railway administrations respectively

The finally targeted report is shown as Figure 5. The freight departments can grasp the wagons flow direction by the statistical result. In the traditional method, the computation will be executed just on the central database server in the Ministry of Railway, but based on RFDGrid platform; we can distribute the fragment queries to the 18 railway administrations nodes.

The implementation of freight applications is independent to the concrete data Grid middlewares, so we can use open source data Grid middleware OGSA-DAI \cite{11, 12} to build the data Grid platform. During the implementation, basic components are wrapped as OGSA-DAI activities and business components are implemented as OGSA-DAI workflows. It is necessary to add more user-defined activities to satisfy the demand of freight applications.

During the deployment, \texttt{StatisticFlowDirectionsService} is a statistical analysis service and is deployed on the front end computer of the Ministry of Railway node; \texttt{DataService} is used to encapsulate the data resources and is deployed on all nodes. In this case, the data resources are the main tables of the railway waybill databases distributed in 18 railway administrations nodes. We configure all the data resources in the configure file which will be used by the DAI factory data resource when a DAI data resource is created. Figure 6 depicts the deployment and implementation of this statistical analysis service.
Each step is described in detail:

1) Users submit the request via the portal.
2) The request invokes the factory service - StatisticFlowDirectionsServiceFactory and on the Ministry of Railway node via the URI of the services specifying details.
3) Then the service instance is created with specified initial conditions on the Ministry of Railway node. The handle of the service is returned by which clients can send messages to it during the process of query execution.
4) In StatisticFlowDirectionsService, makeDQPFactory activity first creates a DAI data resource. The configure file specifies exactly how the deployed DAI data resource should be configured, then the resourceID of the created data resource is returned to StatisticFlowDirectionsService.
5) In the makeDQPQuery activity of StatisticFlowDirectionsService, the following SQL query request is submitted to DAI Data Resource.

```
SELECT DepartureRailwayAdministrationID, ArrivalRailwayAdministrationID,
TheNumberOfWagons, TheNumberOfTonnage
FROM hpwzzk  WHERE zkzprq BETWEEN '20100101' AND '20101230'
GROUP BY ArrivalRailwayAdministrationID.
```
6) Then the query partitions are assigned to Query Evaluation Services (QES). In fact, this step is invisible to the client, and the requestor does not know anything about the data service resource in other nodes.
7) The query results are returned from QESs directly to StatisticFlowDirectionsService. Then the following two activities are executed to deal with the query result. MakeDataRecordTransformation will transform the output of makeDQPQuery activity into a two-dimensional array; makeReport will make the further calculation - add the numbers in all rows and all columns and get the numbers of the first row and the first column;
8) Display the flow directions of all loaded freight wagons report in the portal, as shown in figure 5.

In this description we just listed the important activities. In general it is convenient to organize activities in workflows to complete certain tasks. In this scenario, via OGSA-DAI optimization, the big task is decomposed into many small tasks which can be executed on distributed nodes so the computational capabilities of the railway administrations are used efficiently.

In order to show the strength of the proposed implementation, we design two groups of experiments in which StatisticFlowDirectionsService is executed on different amount of freight datasets. We performed the experiments on PCs with CPU clock rate of 1.86GHz, 1.0GB of main memory, OS is Fedora9 and the data is stored in Oracle 10 database. OGSA-DAI 3.2 is used as data Grid middleware.

- In the first group of experiments, the statistic function is executed on data distributed at k (k=1, 2… 18) nodes, and each node contains 1 or 2 months freight data. Average amount of one month freight data is 300,000 records. The results are shown in Figure 7; we can see that although the amount of dataset increases quickly with the involvement of more nodes, the execution time increases slowly, the main reason is the parallel execution of all nodes.
In the second group of experiments, the statistic function is executed on the same amount of freight dataset and these data is averagely distributed at k (k=1, 2… 18) nodes. The amount of dataset is 5.4 million records and 10.8 million records respectively. The results are shown in figure 8, we can see when the total amount of data is fixed, the execution time decreases quickly with the involvement of more nodes, because the more nodes are involved, the less data each node contains, and the shorter are the execution times on individual nodes.

In real freight applications, real-time statistics and analysis freight services are not supported because of the limitation of bandwidth of railway internal network. When considering statistics and analysis services are based on the centralized freight data stored in data warehouse, the execution time of applications is same with that executed on the one node. From the above experiments, we can see that the execution time with k (k>1) nodes involvement is much better than that on only one node.

5. Conclusion

The railway freight information system is heterogeneous and geographically distributed; it generates large volume of data but cannot share them transparently and computational capability cannot be used collaboratively. RFDGrid applies Grid technology to integrate all types of resources. Based on the data Grid platform, it is necessary to form an applications’ platform to provide efficient, high quality applications services to users in a transparent manner.

This paper analyzes all railway freight applications in detail, and then presents the relevant design concepts and provides an implementation outline towards railway freight applications based on RFDGrid platform. By applying it onto a typical application scenario we show the feasibility and effectiveness of our proposed implementation solution.

Acknowledgement

This work has been supported by Project NO.2006AA01A121 of the National High-Technology Research & Development Program of China and Project NO.2012AA040912 of the National High-Technology Research & Development Program of China.

References