ISSN:0975-9646

K.Srikanth et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (3) , 2012, 4439 - 4442

Wireless Telecommunications Call Records Data warehouseETL Flows

K.Srikanth N.V.E.S.Murthy J.Anitha
Andhra University

Abstract:-The increased number of service offerings and available functionality result in an ever growing volume of call detail records (CDRs). For many services (e.g., pre-paid), CDRs need to be processed and analyzed in near real-time for several reasons, including charging, on-line subscriber access to their accounts, and analytics for predicting subscriber usage and preventing fraudulent activity. In this paper, we describe the challenges associated with near real-time extract, transform, and load (ETL) of CDR data warehouse flows for supporting both the operational and business intelligence needs of telecommunication services, and we present our approach to addressing these challenges.

Keywords: Real-time business intelligence, real-time ETL

1. INTRODUCTION

Gathering intelligence via real-time visibility into subscriber usage is a critical element in both business activity monitoring and real-time decision support solutions. Timeliness of the intelligence gained from real-time visibility into subscriber usage is important for revenue gain as

Well as protection from revenue loss. Consider, for example, the case where the absence of a rule in the logic associated with real-time charging of voice calls results in free international calls for subscribers who dial a specific prefix. Subscriber usage in WirelessTelecommunications is recorded in call detail records (CDRs). CDRs represent a wealth of information that can be mined in order to discover patterns related to both calling behavior and service feature usage (e.g., SMS, MMS, etc.).

Enabling near real-time CDR mining and analytics requires that the infrastructure responsible for extracting CDRs from the near-term storage component and inserting them into longer term storage (for billing applications as well as data warehouses for decision support and business intelligence) is able to operate in near real time. Such extractions are typically performed by customized extract, transform and load (ETL) flows. These ETL flows must address many challenges not typically found in traditional data warehousing projects. In particular:

1. Handle introduction of new service CDRs in a timely manner and with minimum changes in processing flows;

2. Handle changes to existing CDR attributes with minimal impact in processing flows;

3. Support multiple CDR versions for the same service (this occurs when different service functionality is enabled for sub-sets of subscribers);

4. Support multi-tenant solutions where either multiple services co-exist (e.g., CDMA and GSM versions of a pre-paid offering) or different versions of the same service are being offered to different subscriber groups.

In this paper, we present our work in the area of real-time processing of CDRs in the context of a hosted pre-paid service for mobile virtual network operators (MVNOs). In particular, we present a flexible and extensible CDR extraction, transformation, and load solution that handles dynamic changes to CDR attributes, introduction of new service CDRs, and versioning of CDR in a simple and efficient manner. The solution can be easily generalized to support services in other domains that exhibit the same characteristics with respect to the data that needs to be processed within strict timing constraints.

2. MVNO BACKGROUND

The mobile Wireless Telecommunications market is considered to be a very lucrative one. However, building a mobile network and purchasing spectrum licenses can be very expensive for companies that wish to enter this market. Fortunately, the costs associated with mobile networks and spectrum licenses have "driven" existing Mobile Network Operators (MNOs) to embrace a business model that supports leasing of their networks to third parties that wish to offer (mostly) non-competing mobile services. Such third parties are referred to as Mobile Virtual Network Operators (MVNOs). Examples of existing MVNOs include Kajeet (http://www.kajeet.com) and Boost Mobile (http://www.boostmobile.com).

What is important to mention is that MVNOs may require customization to the services they offer to their subscribers quite frequently. Such customization may result in the generation of new service CDRs or changes in the information captured in existing CDRs. While such changes may not create substantial challenges in a single-tenant environment, multi-tenant environments (the most common ones in the MVNO model) must be able to accommodate service CDRs that contain different attributes for different MVNOs.

3. PROBLEM STATEMENT

CDRs are generated by real-time call processing components whose top priority is to handle session signaling (e.g. connecting a caller to a callee). Because of this, CDRs are typically buffered in memory for a short period of time and then written to an operating system file in a file system

local to the telecom switch or IN call processing node responsible for session signaling it is responsible for providing access to these CDRs to all operational and business components that require such access, including revenue assurance, fraud detection, billing, and reconciliation components. Such access is typically made available by treating this storage component as the source used for populating a data warehouse via ETL flows.

Because of the above properties of the short-term CDR storage solution, CDRs are extracted, transformed, and loaded into proper data warehouses on a periodic basis. In a hosted MVNO solution, CDRs are typically pushed to MVNOs on a periodic basis so that they can be incorporated into MVNO-owned data warehouses and business intelligence applications. This process may be performed as part of the extraction, transformation, and load (ETL) flow or may be initiated after the CDRs are inserted into a data warehouse.

While the extraction and load part of the ETL flow do not present any challenges not already addressed by existing ETL solutions, the transformation component must be able to address, in real-time, the following challenges:

1. Identify the "type" of information present in CDRs in order to determine the transformation rules to be applied;

2.Identify new service CDRs for which no transformation rules are available and handle these CDRs in a way that enables re-processing of these CDRs once the appropriate rules are established;

3.Support multiple CDR versions and accommodate transformation rules that apply to all or a selective sub-set of these versions, including different transformation rules per CDR version.

I. Track the ETL flow persistently to avoid duplicate CDR processing or missed CDRs;

II. Enforce a run-time deadline for each ETL iteration in order to accommodate periodic ETL invocations without requiring synchronization between such invocations;

III.dentify when a speed-up (e.g., catch-up) in CDR processing is required in order to maintain an upper bound on the time it takes between CDR creation and CDR processing by an ETL flow.

4. OUR SOLUTION

In this section we describe the architecture of our approach to near real-time ETLflows for CDRs. Figure 1 shows the overall system architecture. The mainfunctionality of the data warehouse component is to store CDRs. A custom ETLapplication was developed for fetching call detail information from the short-termCDR storage tables and storing this information into the data warehouse on a periodicbasis (every 5 minutes).

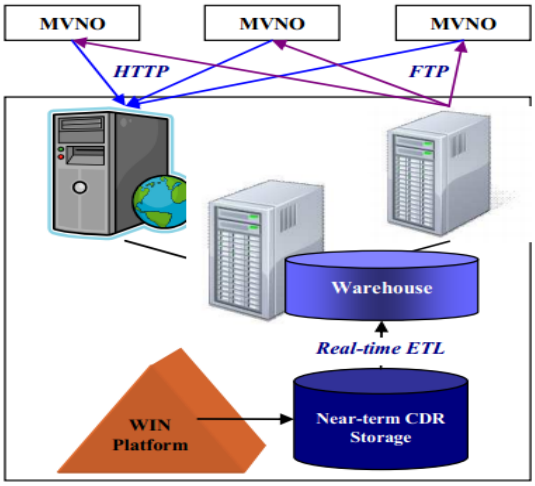

Figure.1: Overall CDR processing system architecture.

4.1 Transformation Rules

The transformation component of the ETL flows is driven by an XML configuration file. This file contains the rules that must be applied to source CDR attributes before being inserted into the data warehouse. These rules are grouped together using the CDR type as the grouping criterion. Typically, the CDR type is identified by either a particular CDR attribute value, a combination of several CDR attribute values, or a join operation between the main table containing CDR rows and auxiliary tables containing meta-data about these CDRs.The following XML fragment shows an example mapping.

In the above mapping, the input CDR record, whose type is identified by the "PPN_sample" value, contains several attributes. These attributes are mapped to table columns in table "voice_sms_samples". When the CDR is inserted into the data warehouse the call duration is recorded in seconds. This is done by instructing the CDR transformation logic to multiply the number of minutes by 60 and then add the seconds to the resulting number. As another example, the input stream may contain two attributes for capturing date and time information. However, we want to combine these two attributes into one before inserting the record into the data warehouseWe should note that the XML-driven approach used for describing mappings from input attributes to table columns is quite extensible and can accommodate additional mapping logic.

4.2 ETL Flows

The following high-level algorithmic steps describe the ETL flow between the short-term CDR storage component and the data warehouse. The following assumptions are made with respect to these steps. Firstly, each CDR in the short-term storage contains an Insert Timestamp attribute. This attribute records when the CDR was inserted into the short-term database. Secondly, ETL progress status is maintained in a persistent manner The status attributes include, among others, ETL process start and end

timestamps, timestamp of the most recently processed CDR record (CDRTime), and the status of the ETL flow (e.g., Active, Done, Error). Thirdly, the ETL process is processing CDRs whose Insert Timestamp attribute is within a specific time window, say 5 minutes, referred to as ETL_window.

Some of the important implementation properties of the steps outlined above include: concurrent CDR fetch from the short-term CDR storage database and application of CDR transformation rules , direct or bulk CDR load into the data warehouse depending on volume of fetched CDRs during each ETL execution, handling of CDRs for which no transformation rules exist, and handling of CDR versioning using a CDR type hierarchy that depends on values present in one or more CDR attributes. We will elaborate on the properties in the following paragraphs.

The use of a generic table that mirrors the table used in the near-term CDR storage database allows us to run the ETL flow against this table once the appropriate CDR transformation rules are put in place without having to rely upon different post-processing application logic. The only difference between running the normal ETL flow between the short-term CDR storage database and the data warehouse and the ETL flow between the exception CDR table in the data warehouse and the data warehouse is that in the latter case the ETL flow will not attempt to re-insert exception CDRs into the exception table (a configuration property or command line argument can be used for this purpose).

4.3 MVNO CDR Flows

Call detail records in the data warehouse are pushed to MVNO backend systems on a periodic basis. This push can take place as part of the ETL flows covered in the previous section or after the CDRs are inserted into the data warehouse. In either case, custom formatting may be required and, in addition, only a subset of the available database table columns may be required. In order to achieve this, an XML configuration file is used for specifying the following:

1. Attributes to be included in the push;
2. Order of attributes included in the push;
3. Format of file containing the pushed records.

The following XML fragment shows an example of such a mapping for a particularMVNO.

The above configuration file specifies that the CDR files pushed to an MVNO include fixed-length attributes. For each attribute included in the CDR file, theelement specifies the source CDR attribute to be used, the order of this attribute, and its maximum size. The two attribute is used for including a header in the CDR file, if required.

Typically, MVNO CDR files are pushed to MVNOs using FTP. Since network conditions, scheduled maintenance, and various other scenarios (e.g., crash of remote FTP server) may interrupt the FTP flow, a persistent process is required in order to ensure that generated MVNO CDR files will reach the MVNO. In our solution, this process uses a database table for storing status information for each CDR file. The status information includes, among other attributes, a unique ID associated with a CDR file, the minimum and maximum Insert Timestamp values associated with the CDRs in the file, and the status of the FTP transfer (not started, in progress, completed, failed).

The minimum and maximumInsert Timestamp values serve two important purposes. Firstly, should we need to recreate a specific CDR file for a given MVNO, we can do so by just knowing the unique ID associated with this file (included in the CDR file name). Secondly, we can easily detect CDR files that may contain overlappingThe minimum and maximum Insert Timestamp values serve two important purposes. Firstly, should we need to recreate a specific CDR file for a given MVNO, we can do so by just knowing the unique ID associated with this file (included in the CDR file name). Secondly, we can easily detect CDR files that may contain overlapping.

4.4 Arroyo

Arroyo [7], our home-grown ETL tool, is a graphical data transformation development environment supporting a wide range of data filtering capabilities. These capabilities are broadly broken into three basic stages: data source selection, data transformation (including matching, selection, classification) and output staging. Specific filters or functions in each of the stages are chained together to produce an overall data flow. Within each stage, processing block customization and user defined transformation functions can be defined and added to the block. At each stage, external sources can be used for refinement and validation.

The Java-based tool is able to process complex data analysis flows across a wide variety of data sources. The tool is readily extensible, where new pre-processing or matching functions can be encapsulated into Java classes and dynamically added to the selection palette. Custom components can be written as Java plug-ins and used as first class components.

5. RELATED WORK

Progress in ETL and associated tools have occurred mostly in the commercial arena. Ad-hoc and self-built ETL tools were prevalent in the 80's and the 90's. These tools were scripts using metadata and applied mostly to databases. Pentaho Data [1] Integration (previously kettle), Talent [2] and others are examples of open source ETL tools.

ETL tools have started to migrate into Enterprise Application Integration systemsthat now cover much more than just the extraction, transformation, and loading of data. Data transformation and integration are becoming critical in business intelligence projects often requiring a separated storage area (sometimes call staging or work area) for intermediate results, similar to our work. Ralph Kimball [6] states that ETL design and development may amount for up to 70% of the IT side of a BI project

which makes ETL tools a crucial component in the BI architecture.

According to Gartner, "the stand-alone data integration technology markets — such as extraction, transformation and loading (ETL), replication and federation, and enterprise information integration — will rapidly implode into a single market for multi-mode, multipurpose data integration platforms." Indeed if one looks at the top vendors in the market, it is clear that this is happening or has happened. InformaticaPowerCenter has added a real-time module to their software, allowing Informatica tobrand PowerCenter as an EAI tool; while IBM has added DataStage, acquired from Ascential, currently under the InfoSphere family.

6. CONCLUSIONS

In this paper, we have presented our approach to near real-time processing of call details records in the context of Wireless Telecommunications services. Our approach is based on configurable transformation rules captured in XML configuration files. We have implemented the approach described in this paper in the Telcordia MVNO hosted pre-paid service and have been able to handle introduction of new CDRs, versioning of CDRs for different MVNOs, and different MVNO formatting requests with only changes in the configuration files that drive the ETL flows. Our solution can be easily generalized to support services in other domains that exhibit the same characteristics with respect to the data that needs to be processed within strict timing constraints.

REFERENCES :

1. Pentaho Data Integration, http://kettle.pentaho.org.
2. Talend ETL, http://www.talend.com.
3. Microsoft SQL Server Integration Services, http:// msdn.microsoft.com/en-us/library/ms169917.aspx.
4. Oracle Warehouse Builder (OWB), http://www. oracle. com/technology/products/warehouse/index.html.

5 IBM Info SphereData Stage, http://www. ibm.com/software/data/infosphere/datastage/features.html

6. Kimball, R. "Data Warehouse Training," http://www. ralphkimball.com/html/articlesbydate/articles2007.html

7 Caruso, F., Cochinwala, M., Ganapathy, U., Lalk, G., Missier, P.: Demonstration of Telcordia'sDatabase Reconciliation and Data Quality Analysis Tool. Proceedings of 26TheInternational Conference on Very Large Data Bases (VLDB 2000), September 10-14, 2000, Cairo, Egypt.