Bull. Nov. Comp. Center, Comp. Science, 21 (2004), 21–32

The PABX monitoring

S.V. Bredikhin

Abstract. This paper describes the experience gained in the development of PABX monitoring system using the Internet-technologies. The basic attention is being given to the formal representation of data accounting, storage subsystems and access to data. Information about the system realization is given, and the user interface is described.

1. Introduction

In the modern world of telecommunications, there are information sources that generate data in automatic mode. Database management systems are the dominant tools for the storage and processing of such data. Information availability in many databases does not usually mean that it is easy for a non-professional user to access these data. In order that the data be accessed the user should have not only the physical access to the corresponding Database Management System (DMS), but also the knowledge about the data model to be used, database scheme, the skill of using the query language. That is why the problem of providing an easy access to existing databases is so urgent for many organizations today.

2. Problem formulation

The object of this research is the Private Automatic Branch Exchange (PABX), which is an automatic telephone switching system within a private enterprize. It is necessary to collect information about all phone calls going through this system and to prepare reports about the phone utilization on the basis of this information. Reports should be presented in the electronic format, using the Internet technology, upon the user's or administrator's requests. In addition, it is necessary to implement the monitoring of Dial-Up Service, which uses the PABX.

Thus, it is important to develop a PABX Monitoring System on the enterprize level in order that phone calls be controlled and information about access to Internet via switched phone lines be collected. This system should automatically accumulate information and on 24-hour basis.

3. A formal model of the monitoring system

Information about all phone calls, delivered by the PABX, is accumulated in the database. With the use of the standard network access tools the user connects to the known web server. Then he authorizes himself and requests the necessary information. The web server translates the user's request into the language, understandable by the DBMS, and sends it to the database server.

The database server checks this user access rights and, if the user has such rights, it processes this request. This way, the PABX monitoring system naturally includes three parts: Data Collection System, Information Storage System and User Interface System. These parts are logically independent of each other and can be executed on different workstations, interacting with each other.

3.1. The data collection system

The data collection process is the process of gathering information sent by the PABX and its storing into the database tables. Upon receiving an incoming call, the PABX locates the calling user's line and establishes the connection. The PABX sends information about this connection to the Data Collecting System after the connection is by some reason terminated. Information, provided by the PABX, depends on the type of a station. We are interested in general data, common to all the PABX:

- Source phone number;
- Destination phone number;
- Time and date of an event;
- Duration of a communication session.

The Data Collection System should be constantly ready to receive information from the PABX. Otherwise this information can be either lost or corrupted.

A state diagram of the Data Collection System is shown in Figure 1. It is represented as an oriented graph. Its nodes are labelled by the system states. Oriented arcs can interconnect two nodes or begin and finish in the same node. Each arc is labelled by the event, which caused the system state change. Each change of the system state can be accompanied by additional service actions, described in the Transition Diagram (Table 1).

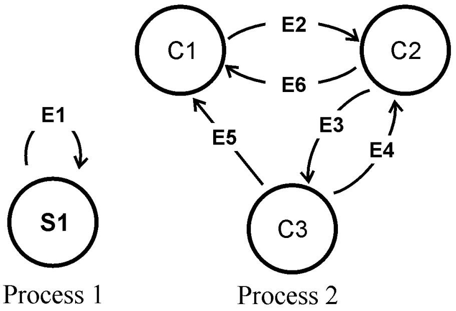

Figure 1. State Diagram of the Data Collection System

Table 1. Transition table of the Data Collection System

State "Before"	Event	Action	State "after"
S1. Waiting for the data coming from PABX	E1. Data receiving from PABX	Local storing of the new data	S1. Waiting for the data coming from PABX
C1. Waiting for the connection with DBMS	E2. Establishing the connection with DBMS	Transition to C2 state	C2. Waiting for the local data
C2. Waiting for the local data	E3. Appearance of the new local data	1. Request to DBMS on adding new data 2. Transition to C3 state	C3. Waiting for the answer from DBMS
C2. Waiting for the local data	E3. Appearance of the new local data	1. Request to DBMS on adding new data 2. Transition to C3 state	C3. Waiting for the answer from DBMS
C2. Waiting for the local data	E6. Termination of connection with DMS	Transition to C1 state	C1. Waiting for connection with DBMS
C3. Waiting for the answer from DBMS	E4. Data is successfully added to the database	1. Mark added information as "processed" in the local storage 2. Transition to C2 state	C2. Waiting for the local data
C3. Waiting for the answer from DMS	E5. Request was processed incorrectly or termination of the connection with DBMS	Transition to C1 state	C1. Waiting for the connection with DBMS

Each row of the table contains a system state before the event has occurred, necessary action for processing of the incoming event, and the next system state.

The data collection consists of two processes (see Figure 1). The first process is always ready for the interaction with a phone station, it awaits the data coming from the station and stores it in the "raw" form, unprocessed on the local computer hard disk. The second process starts working and provides the transfer of local data to the database. If, by any reason, the second process is not ready for the interaction with the DBMS (it processes the previous information, DBMS failure, etc.) the information will not be lost, since the first process stored it into the local file. The collected data will be automatically transferred to the database when the second process is ready. Such an approach provides constant readiness of the Data Collection System to interact with the PABX and allows one to avoid losses during the Information Storage System failures.

3.2. The Data Storage System

The Data Storage System is used for storing information about all phone calls, passing through the PABX. In general, the Data Storage System is a Database Management System, whose main goal is to store data and to process users queries.

Let us look at the data structure. The database contains the three main tables: Table of Users, Table of Extensions (the phone directory), and Table of Phone Calls.

Table of Users contains information about all users, who have the access to the PABX information system. It contains the following fields:

- Login name;
- User's password, which is used for the user's authentication;
- User's full name;
- User's access list.

In this table, a standard set of operations includes the following: add user, remove user, change user's information, change user access rights.

Table of Extensions stores information about all phones and communication links, connected to the PABX. It contains the following:

- Extension or line;
- Type of phone (digital, pulse, tone, voice mail, etc.);
- Full name of the phone owner;
- Information about the phone location;

A standard set of operations in this table includes the following: add/remove phone, change phone information.

Only the system administrator can add to and change data in all the above mentioned tables.

Table of Phone Calls contains information about all phone calls, passing through the PABX. The main fields of this table are:

- Extension;
- Dial number;
- Time and date of an event;
- Duration of a session;

- Number of the line to which the phone was connected.

The Data Collection System automatically adds information to this table.

3.3. The Data Access System

The Data Access System, i.e., the interaction between the user and the monitoring system, assumes that for each user there will be generated the process-respondent, which will act according to the scheme presented in Figure 2 and Table 2. This method of the user's request processing allows to distribute data to several users simultaneously.

It is unknown in advance which particular reports should be generated. That is why it is necessary to develop and provide users with a user-friendly toolkit for the database queries construction. It should provide the possibility to create correct requests according to the call parameters and should contain tools for displaying results and their interpretation.

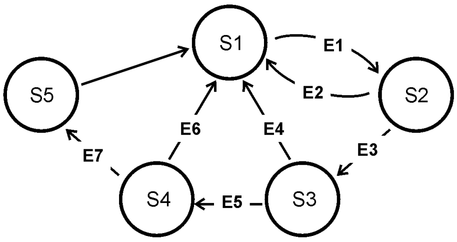

Figure 2. Data Access State Diagram

Parameters of the Request Compiler are:

1. *Phone extension* – an arbitrary set of extension values with an option to group them. Existing phones are grouped according to various characteristics (pulse or tone, external or internal, etc.). There is a feature to define these groups.
2. *Dialed number* – an arbitrary set of values from the unique number (searching for particular call) up to all possible numbers. There is a feature to distinguish calls: long-distance and inter-city, incoming and outgoing.
3. *Time and date* – an arbitrary time interval or a set of time intervals with the feature of sorting values of other fields by date of the call.
4. *Duration of the call* – a particular time interval can be introduced. The following functions are implemented: summarization, search for a minimum, a maximum and an average value.
5. *City line number* – direct indication to required extensions.

Information, received due to the request can be presented in the two forms: as a table or as a graph. A table allows one to get the detailed information about calls. And a graph presentation (a bar graph, a diagram, a graph) allows one to see a general qualitative picture (for example, frequency

Table 2. Transition table of the Data Collection System

State "Before"	Event	Action	State "after"
S1. Waiting for user requests	E1. User request	Transition to S2 state	S2. User authentication
S2. User authentication	E2. User doesn't have access rights	Transition to S1 state	S1. Waiting for user requests
S2. User authentication	E3. User has access rights Transition to S3 state	Transition to S3 state	S3. Generation of the request to DBMS
S3. Generation of the request to DBMS	E4. The request can't be compiled	1. Inform the user about the error 2. Transition to S1 state	S1. Waiting for user requests
S3. Generation of the request to DBMS	E5. The request is compiled	1. Establish the connection with DBMS 2. Send the request for its execution 3. Transition to S4 state	S4. Waiting for DBMS answer.
S4. Waiting for the DBMS answer	E6. DBMS either sent incorrect answer or error message	1. Inform the user about the error 2. Transition to S1 state	S1. Waiting for user requests
S4. Waiting for the DBMS answer	E7. DBMS sent the correct answer, containing necessary data	Transition to S5 state	S5. Interpretation of the data received
S5. Interpretation of the data received		1. Conversion of the received data to HTML format 2. Forwarding data to the user 3. Transition to S1 state	S1. Waiting for user requests

of long-distance calls depending on time of the day, or the utilization of city lines, connected to a phone station).

4. Implementation of the Monitoring System

Let us list the main parts of the system, which were developed during implementation of the Monitoring System:

- The PABX Data Collecting Process and allocation of these data in the database;

- Information Storage System;
- User Interface.

4.1. The Data Collecting Process

All communication lines, connected to the PABX, are divided into the two categories: local lines (connected to phones) and external lines (connected to the city automatic telephone exchange). The coral PABX [1, 2] enables connections between two local lines or between local and external lines. Only sessions which use external lines are registered.

This PABX has a special interface (RS-232) for connecting to the computer and transferring information about phone calls. On the entry call, the PABX finds the line of the called abonent, establishes connection between lines and fixes the time of establishing the connection. Right after the connection is terminated the station accounts the duration of the connection, forms a data packet about the event and sends it to the computer.

Each packet contains the following information:

- Symbol-marker of the beginning of the packet (defined during the PABX configuration);
- Date of the call, in mm/dd format;
- Time of the call, in hr:minute format;
- Duration of a session, in minute:second format;
- Trunk group;
- Trunk number;
- Local line number;
- Type of the call – incoming or outgoing;
- A telephone number dialed;
- Symbol-marker of the end of the packet (defined during the PABX configuration);

The Data Collecting Process operates on the computer, directly connected to the PABX, in the FreeBSD environment. A special process, launched during the system loading, waits for data packets from the PABX and stores them in the phone calls data file. The information, received from the PABX, should be converted into SQL-queries before adding it to the database. The Perl language [3] is used for this purpose. It has special modules to access databases [4]. The Data Collecting Process establishes connection with the DBMS using these modules. It sends out SQL-queries to add a new information on receiving data.

Below we describe the main execution algorithm of the Data Collecting Process (see Listing 1). During the initialization phase (lines 10–40), the program sets the last_date variable to the value saved in the status file (date and time of the last event, fixed in the database) and establishes connection with the database server. During the main cycle (lines 50–210) the program looks through records of the phone calls data file and compares the date of the last event, stored in the database, with that of the current event (line 90). If the value of the last_date variable is older than the date of the current event, then the information is converted into SQL-query and added to the database. The time and date of a newly processed record is written into the status file. If the date of the current event is older than the value of the last_date variable, the program switches to the next record in the file. After examining the file, the program will be waiting for the new data. If connection with the database server is terminated while the program is running, then the program will be turned to the waiting mode until the connection is reestablished (a cycle in lines 110–200).

Listing 1. The Data Collecting Algorithm

```
10   Initialization
20   If the status file exists then
30      Assign to the last_date variable
                 the value from the status file
40   Establish the connection with DBMS.
50   Do
60      Read the phone calls data file
70      If the end of the file is reached then
80         Wait for the new records in file
90      If a record exists and the time and date of the call is
           more than last_date then
100        Convert current data to the SQL query.
110        Do {
120            Execute the query.
130            If the query is executed successfully then
140               Save time and date of the current call
                        in the status file.
150            Else
160               If there is no connection with DBMS then
170                  Restore the connection with DBMS.
180               Else
190                  Log error message. }
200        while there is no connection with DBMS.
210  Enddo.
```

4.2. The information storage system

Relational Database Management System PostgreSQL [5] is used for storing and processing phone calls information. The architecture of this DBMS is based on the client-server model. DBMS session consists of the following interactive processes:

- Postmaster – managing process-daemon, which manages interactions between external and internal processes. It assigns the shared dynamic memory buffer and executes other initialization work during the launch.
- Postgres – internal database process, executing user queries. Postmaster always forks a new postgres-process for each user application. This process is executed on the server part.
- An external application can be located on another computer (for example, on a workstation). It communicates with Postgres via Postmaster.

Once started, the postmaster daemon manages databases, installed on the server. For each external application needing to access to one of these databases, the postmaster generates postgres-process and connects an external client program with the server. From this moment on the client and the server processes interact without postmaster. This way, the postmaster runs permanently, waiting for requests, while connections with external applications are being established and terminated.

An extended version of the SQL query language is used for the data management.

The LIBPQ library, containing a set of functions for interaction with the database server, is used together with the DBMS. The scheme of database access from external programs is rather simple. The application establishes the TCP connection on a particular port (usually 5432) with the postmaster process using a special function – PQsetdb. The function parameters are: the database name, the IP address of the server, and the port number. If the connection is established successfully, a PQexec function is executed. Its task is to begin database transaction, execute the query and finish the transaction. After this, the connection with a database is terminated. After the query for retrieving data from a database is executed, the PostgreSQL creates a temporary table of results. Elements of this table can be easily accessed using the SQL operators, related to cursors, and a special LIBPQ function, related to the corteges and the relational fields. This allows one to generate arbitrary reports according to the user's requests.

4.3. The User Interface

Interaction between the user and the monitoring system is realized via Internet using the HTTP protocol [6]. The user connects to the web server using a standard browser and receives the page, containing the application form. He should indicate the database query parameters. The form is sent to the web server that the DBMS query results be received. To interpret and to format the data, the web server uses special HTML documents (templates) with built-in it php-functions [7]. These functions check whether input parameters are correct, convert them to the SQL query, establish a connection with the DBMS, transfer the query to the DBMS and convert a table of results into the HTML format. The result of the processing is sent to the user.

Let us view the algorithm of accessing the phone calls database on an example of the problem of viewing the phone calls registry. The user fills in the form the list of fields in the report, their format, type of report representation (a table or a graph) and other parameters. It should be mentioned that the monitoring system administrator has additional options: to add, to delete and to modify records in the phone book; to add, delete and modify records in the user table; to assign the user access rights, to submit the SQL queries and to execute them.

Listing 2 shows the algorithm of the program that processes the data upon the user or the monitoring system administrator request. The program processes input data (line 20) and converts it into the SQL query. Then the program establishes a connection with the database server and transfers the generated request for its execution (line 60). If the request can not be executed, the program displays on erroneous message to the user and terminates its work. If the request is executed successfully, then the program receives data from the DBMS and converts it into the HTML format. To provide an easy view of information, to avoid the transfer of the large volumes of data and to avoid delays, due to such a transfer, the program divides data into pages.

The algorithm, shown below, is a general algorithm for processing of all requests. In each particular case, processing of parameters and their conversion to the SQL query is done differently, but its further processing is done according to one scheme: to connect with the DBMS, to send out the query for its execution, to receive results, to convert them to the HTML format and to send them out to the user. Two main services were realized according to this algorithm: providing the access to the phone calls database and generation of the corresponding reports; providing the access to the Dial-Up service database.

Listing 2. General algorithm of access to the DBMS

```
10   Establish connection with DBMS.
20   Convert input parameters to SQL query.
30   If the page number is not specified then
40      Assign page_number=1.
```

```
50   Assign the number_of_records_on_page=25.
60   Execute request.
70   If the request was executed unsuccessfully then
80      Display error message.
90   Else
100     Start displaying the table.
110  Assign Record_number=(page_number-1)*number_of_records_on_the_page.
120  Assign i=0.
130  Do until i <= number_of_records_on_the_page
140     Retrieve a record with number =(Record_number + i).
150     Convert record into HTML format.
160     Display an HTML line.
170     Assign i=i+1.
180  Enddo.
190  Quit displaying the table.
200  Terminate the connection with the DBMS.
```

5. Dial-Up Monitoring

The Dial-Up Service was organized for providing the access to Internet via the city phone lines. It is based on the PABX Coral II, the modem pool and the access server (Figure 3).

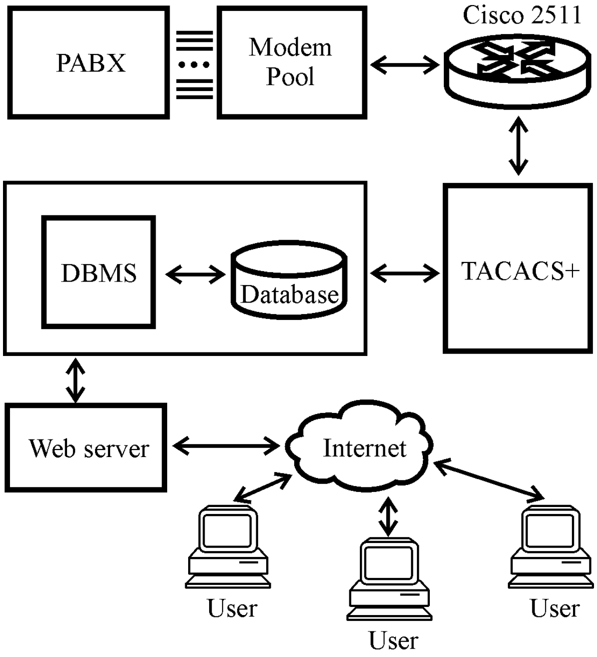

Figure 3. Dial-Up Service Monitoring System

A part of the internal analogue PABX phone lines are switched to the modem pool, consisting of Motorola 3267 modems, supporting V.34 protocol and V.22 error correcting protocol. Cisco 2511 router, configured to use TACACS+ access control protocol for the dial-up users plays the role of the login server. The TACACS+ server software works on the SPARC 5 workstation and executes the functions of authentication, authorization and accounting.

The Dial-Up Service Monitoring System is organized in the same way as the PABX Monitoring System. A special data analyzing program gets an accounting information from the TACACS+ server and puts it into the database, making it available to the users via Internet. In addition to the information, registered by the PABX, the database stores the following:

- Accounting name of the user, who obtains access to the NSCnet;
- Fact of login to the Net;
- Time and date of accessing the Net;
- Number of modem from the modem pool;
- An identifier of the process, that serves a particular user.
- Connection duration;
- Fact of logout from the Net;
- Number of bytes, sent to the user;
- Number of bytes, received from the user.

This information allows the user to receive a report about resources he used. This is done by defining the request parameters in the corresponding form. Additionally, a system administrator can receive reports about all users, change the user accounting information, assign the access rights to the web resources.

References

[1] Coral Installation Procedure & Hardware Reference Manual. – Tadiran Telecommunications.

[2] Coral. Program Interface. – Tadiran Telecommunications.

[3] Perl. – http://www.perl.org/.

[4] Database Modules for Perl. – http://reference.perl.com/query.cgi?database.

[5] PostgreSQL Home. – http://www.postgresql.org/.

[6] HTTP/1.1 – Proposed Standard RFC 2068. – http://www.w3c.org/Protocols/rfc2068/rfc2068.

[7] PHP: Hypertext Preprocessor. – http://www.php.net/.