White Paper: Firebird Replication

PDF Download

Holger Klemt, August 2016

In 1989 I implemented my first major commercial database application as a freelancer, for a children's clothing retailer. They had a head office with catalogue-based mail order and three further retail outlets in the Oldenburg area in North Germany. The client was convinced that a central computer-supported data administration was a vital component of his business management. Success in this industry depended upon viewing current stock levels at all locations, in order to serve both the retail customers as well as the mail order customers quickly and reliably.

At that time Internet was still a long way away; it was possible to transmit data by modem, but this was costly and extremely slow. Therefore a direct data connection did not initially come into question. We therefore implemented a data exchange mechanism using the usual media of the day (the more mature readers may still remember): floppy disks.

Due to the limited hardware budget, the company had relatively old computers in use, so that we had to make do with 5.25" disks with 360 kilobyte, which even then were no longer so widespread in use. Hard drives were available, but their capacity was still specified in megabytes. I seem to remember that 40 MB was an unlimited luxury at the time. I know that my first PC, purchased in 1987, was equipped with such a hard drive.

The complete software was developed with dBase III for MS DOS. Looking back today, the possibilities were very modest back in those days, but nevertheless sufficient for the chosen purpose, and the jointly developed software was a huge help for the customer. The comparatively small amount of product master data with approximately 200-300 items was maintained at the headquarters, and alterations were regularly copied onto disk by the owner and added at each store location. As no alterations were being made at the branches, the tables could simply be overwritten. Each store maintained their own separate table for products sold. As these were only written in each branch, the owner could overwrite the tables intended for the central administration. The tables for each store had different names, and could therefore be copied directly onto disk together with the data from the other stores, without the risk of overwriting foreign data. The copy operations were automated via batch files.

The various tables on the disk were then combined and transferred to central tables. Now the entire inventory of all items at all locations could be viewed at any time. The amount of data was within reasonable limits, because the 80×25 character text screen could not display any images. And although there were already scanners available, they were very expensive and made little sense in MS DOS text mode.

After the owner had deployed the well-developed software for a few years, we can say with a clear conscience that this custom software was a success.

A number of years later, I was asked about a possible modification of the software. Unfortunately I had to reject the project due to the lack of a 5.25" drive.

From an enterprise perspective replication requirements today have hardly changed. Central and decentralized recorded data should be capable of being displayed and edited if necessary at all

locations, and to be synchronized as soon as possible at all sites. Of course even today the above model could be implemented, by sending data packets back and forth, but to commute with floppy disks between all sites is rather inappropriate in this age of globalisation. Certainly, the data packets can also be sent nowadays via the Internet, but the requirements have increased considerably. Whereas the article text used to consist of two text fields each of 40 characters in length, nowadays images and PDF files are important components of any software for any company and their clients.

Over the last 27 years, I have therefore constantly had to react to new customer requirements and continuously improved the system of distributed data management. In this document I will cite a number of important aspects, which should be taken into consideration.

Basic concepts

Based on the implementation of a large project for an enterprise in the catering industry, I will illustrate our concept using examples from the field of gastronomy.

We will assume that our fictional enterprise has just been established and further requirements need to be fulfilled as the company steadily grows and expands.

The business sets up a restaurant at a highly promising location and starts off with a simple POS system.

All orders and product modifications are performed directly on the single POS. As is customary, the orders taken at the table are written down on notepads and then typed into the till to produce the order slips for the bar and the kitchen. The chef does a good job, the service is attentive and the business grows.

Due to the success of this restaurant, the outdoor seating area in the summer becomes increasingly important; due to the good weather many guests want to sit outside. The routes from the outdoor terrace to the POS are significantly longer, so that even more time elapses between taking the order and the order slips reaching the bar and the kitchen, so that they can be carried out. By now the number of employees has increased significantly, which leads more and more frequently to delays. A second POS, as near as possible to the outdoor seating area, is the solution. It is vital that important master data on both POS is always in sync, so that any necessary modifications to product prices are carefully documented and do not have to be successively adjusted manually at both POS. And network capability of the POS is essential. Orders for tables must be visible promptly and completely on both POS and it must be possible to print slips and receipts at both POS. To prevent an order slip being printed multiple times, both POS must exchange data promptly - ideally at transactional real time. Once the original order slip has been printed at one POS, the other POS is only able to print a copy of the slip with the clear indication “copy”.

The common product master data at the single location makes this solution relatively simple. The second POS could directly access the database of the first, as a network connection between the two POS is necessary anyway. For this reason, this approach is possible without any replication technique.

Unfortunately the main POS was severely damaged. Result: the second POS no longer has access to the main one and cannot be used, because the database is located on the main POS and unfortunately this is irreparably damaged. The entire product master data is lost. Now the second POS is to be used as the main one. Due to the special hardware either no backup has been made, or the

last backup is outdated, so that all data must be re-entered manually.

A simple solution would be to back up the data of the main POS regularly on the second POS, so that this, if necessary, can then continue to work with up-to-date data.

But what does regularly mean? Let’s assume that the damage occurs while the entire restaurant operation is in full swing. Both daily as well as hourly data backups would in such a case mean disaster: the service staff would have to ask customers what they have consumed in order to bill food and beverages, they would have little idea which tables have already been billed, which orders have already been delivered, etc. As no data access is possible at all, the experienced restaurateur could, as was customary decades ago, continue working: orders are written on coasters, and when the customer wishes to pay, it is all added together and paid in cash. The problem is temporarily solved in the short term, but not in the broader sense from the point of view of the owner and certainly not the local tax office.

Depending on the amount of data, a small to medium restaurant could certainly perform hourly backups. But a complete system failure could also damage an important module in the network, for example by over-voltage. In such a case you will run out of possible solutions very quickly.

Master-Slave Replication

We have developed a live replication known as master-slave for a customer, who uses Firebird-based software for his bar in Manhattan with hundreds of tables. A central database server, which stores the data for all POS systems, replicates all write operations in near real-time onto a second database server. During normal operation all work with the application software is performed solely on the master, in order to prevent data conflicts. The software was adapted so that it can immediately switch to the slave should the master fail, the slave becoming the new master without any further interaction. When the former master is up and running again, this can be prepared by an admin as the new slave or, following a brief downtime of both servers, become the new master again.

We implemented all functions for this using Firebird internal functions. Only a few external scripts are required, which start each real-time replication process based on the Firebird Event Alerter technology. Technically, the same solution can be used to prepare a master multi-slave environment by synchronising two or more slaves, on which read queries can be performed at any time without any restrictions. This is also an excellent solution for an increased volume in reporting, and is also suitable to make a database fully available for read operations, for example, on a laptop. This enables data to be made available at locations without any reliable network access, such as on a train or plane. Once the laptop can re-access the corporate network, the delta replicates everything since the last network connection, so that even very large databases with hundreds of gigabytes are easily mobile.

Back to the restaurant business. The outage caused by the damage is over, and the experience used to set up a master-slave replication for the data, so that all data is also stored on the second POS in case of failure of the main one. And the eventual failure of the second POS must not have any negative impact on the main POS.

Master-Master Replication

The opportunity arises for this increasingly successful enterprise to take over a competitor around the

corner. This company has however established itself with a different product range, which should continue to be maintained. A full copy of the existing POS data is of no use here, as they have other products and other prices for the same item numbers, which are already established in the menu. This therefore requires separate product master data. Unfortunately there is neither a wireless connection via WLAN nor direct network cabling between the locations. Growth of the first restaurant has in the meantime meant that it was necessary to add a PC to the network for the Manager, for the complex recording and maintenance of working hours and planning, as well as with the product master data.

The new restaurant suffers from an acute staff shortage, so that employees occasionally need to be employed at both sites. To ensure a scrupulous and legally compliant payroll, a prompt input of all working hours in each restaurant is vital. Each manager must be able to input and maintain this data for their own companies independently of the other restaurant. A master-slave solution is therefore no longer a suitable solution. The system integrator recommends a terminal server connection to the other restaurant via the Internet, but experience at the new site has shown that the Internet connection is not only very slow but also extremely unreliable.

An expensive dedicated line is not an option, and UMTS or LTE makes little sense due to the poor coverage. Furthermore the unreliable power supply for the pizza oven often causes the Internet connection to fail for several minutes at a time, so that it is not feasible for the manager to edit staff hours and planning data during the standard opening times.

The only feasible solution here is a master-master replication. At each site a complete database provides all relevant data, which can be used without any problems, even with a poor Internet connection. The employee master data, which is available for all locations, ensures that work hours and break times can be reported across all locations at the end of the month to the accountant.

When replicating, a suitable data model must be selected so that a master-master replication can be implemented expediently. Should a complete breakdown of the Internet lines prevent an online connection, the database server should still be able to generate data using primary and foreign keys at each location in such a way that they can be sent to the other master upon restoration of the online connection, without an overlapping of any newly created records.

Although almost any database can be prepared for a master-master replication with certain modifications, it makes much more sense to adapt an existing data model with the help of an experienced consultant, such as IBExpert for example. The initial additional expense will significantly reduce future, unplanned additional expenditure, which will certainly occur without such adjustments.

By adapting the data model, the data can now optionally be fully replicated and displayed in full at both locations or by using appropriate techniques such as the updatable view in Firebird, displayed only to the extent as is expedient for each local site. An item x should only be visible and alterable for the site at which this product x is sold. Firebird views are an ideal means for this. Optionally, the employee base should display all employees and their planned work times, in order to schedule employees of the partner restaurant’s workforce. Trade agreements regarding bonuses for night shifts, etc., mean it is essential to evaluate all working hours across all locations at the end of each month.

Upgradability

The company's success prompts the management to outsource the administration, so that a further new location for the database is required. In addition, further expansion is already planned with another restaurant in the neighbouring town. This also has a different concept and therefore other items in the product master. To ensure the sustained success of the company in this growth phase, the management decides to centralise procurement. Each product in the product master must now be assigned to one or several purchase items. Since each concept uses different product numbers, which cause an overlap in the products used, and it is also necessary to assign different quantities to each product number, a data model needs to be developed, which can keep up with the growth of the business. Important aspects, such as regionally or seasonally available products should also be taken into consideration in the planning at this stage as well as any legal requirements, such as declarable allergens.

In order to give Procurement an insight into the revenue data, not only the total turnover of each POS needs to be replicated with immediate effect, but each item sold should be capable of evaluation by branch, table number, time and service personnel. The previously relatively modest amount of data is becoming considerably larger. Since all data should be sent following the Z report at the end of the night shift, the period of time is limited before Procurement can determine the success or failure of the previous day, and utilise the data for the purchase of seasonal products. Certain products such as fresh fish should only be purchased if the stock levels from the previous day require this and only then, if there was a demand for this product. Perhaps too much fresh fish was ordered, however nothing could be charged or orders were cancelled, because the quality was too poor. It would not be economical to only be able to perform this analysis at the end of the month.

Unfortunately an ADSL line at each site is the only financially feasible option, which means that although the data download is relatively fast, the data upload times for sending also need to be taken into consideration. The first restaurant has continued its success, and on sunny Sundays or public holidays tens of thousands of detailed data are exported from the POS. The execution of an SQL command with parameters via an ADSL upload is not the best option. Both for the SQL command itself and for each parameter value TCP/IP data packets are sent, for which at least the ping time needs to be scheduled. With a ping time of 25ms to the server and an insert command with 20 fields, you need to reckon with at least a half second per record. In reality, this is unfortunately a great deal more. Since the day has only 86,400 seconds, we soon quite clearly reach the limits with more than 100,000 records. The requirements for the replication need to be implemented in such a way that the data model and suitable choice of SQL commands minimize the overhead. By suitably processing the data before sending, using Firebird technology we can increase the upload and download capacity by several thousand percent, compared to the parameterized individual commands just described. Without the appropriate know-how it is easy to reach capacity limits. Along with other methods, we use SSH as the encryption and compression protocol. This way we can access each server as if it were in our own network using special configurations, even though neither port clearances nor static IP address or the like need to be set in the router. Operation via wireless connections is even possible with our technology.

The enterprise continues to expand and eventually central servers are installed in data centers, because upload times are beyond the headquarters’ capacity; there are constantly new sites being integrated into the enterprise, the employee base is growing significantly; the first restaurant abroad is opened; the server in the data center is supplemented by servers in another data center, in order to be prepared for eventual downtime of either data center; the employees should be able to view and manage their recorded work hours via web interfaces on their smartphones, guests should be able to reserve tables online, and so on.

This small, fictitious example can be applied to almost all industries, whether you produce a cup of coffee or a screw, clean windows, or you are a software vendor whose customers require such a

solution.

The current trend towards the Cloud is in our opinion a hype, which should be regarded with caution. Once the Internet connection becomes unreliable, your entire business can come to a standstill. Many an excavator has, at a single stroke, thrown whole business parks back in time. But no excavator can trigger what Telecom engineers can induce nationwide again and again by installing insecure software updates in the central control systems. If the outage only lasts a few hours, then you should be content. A few days ago one of our customers had 45 locations offline for about 15 hours, due to a large-scale failure in the corporate network of Telekom. When an Internet provider guarantees 98% availability, it must be assumed that the Internet is not available for 2% of the year: this does not necessarily mean 7 x 24 hours, but may also be 14 x 12 or 21 x 8 hours, i.e. 21 complete working days without Internet. We have deliberately not included weekends in the calculation.

Depending on the company size and the importance of the software and data for the business, using the Internet for data exchange can be a very good strategy. But only if you can continue working at any location without a connection, and when all data for each connection can be immediately forwarded once the connection has been reinstated.

However, when lorries are queuing on the forecourt and no one in logistics can print a delivery note, because no one can reach the database in the cloud, then depending on the industry sector, an outage of even a few hours can be unacceptable. Submitting to a total dependency on the cloud provider is not really desirable in my view. Every enterprise should seriously question a complete outsourcing including hardware and data. With the aid of a suitable partner such as IBExpert , you will be able to establish your own, distributed enterprise database cloud.

If the company, or in the case of a software vendor - your customer, needs to cover more than 100 locations, and you are looking for an established, replicable database, we would like to encourage you to first study the price lists in advance, and then calculate precisely the budget required per site for direct licensing costs for MSSQL, Oracle or even for MySQL, before you can even begin to establish a solution.

Partial replication or connections across slow lines are not envisaged for many of these products, the only solution here is to calculate and plan a further budget for rapid dedicated lines.

Interested? In our regular Bootcamps we show you the basics. You can also book a Bootcamp at any time worldwide at your own premises, to find out how our know-how can help you and your software to keep pace with your enterprise’s growth, or the growth of your clients.

About our customer project

The customer currently deploys database servers supplied by IBExpert at 115 locations. An additional 15 servers are used as backups or for special tasks.

All locations are connected via normal ADSL connections over the Internet to the data center. 3 million master data records (employee base, work hours, etc.) are replicated transaction-safe at almost real time with all other server boxes. Around 25,000 records are created, modified or deleted daily in these tables. Therefore every day approximately 25,000 x 130 = 3.25 million master data records are distributed in 24/7 operation. Typically, all master data is synchronized in less than 5 seconds at all sites.

The sales data, read from the POS systems each morning from 05:00 onwards, generates between 1,500-50,000 records daily from each site. This data is sent only to the headquarters and needs to be available very quickly (target: by no later than 08:00) on the administration database system. A total of 7 database servers form the backend for the web server and other functions, and are the target for the sales data. Therefore each night from 05:00 onwards about 5 million records in total are transmitted and distributed through the central cluster in less than 3 hours from all sites to 7 databases. A total of around 38 million records are distributed in 24 hours in this way.

IBExpert performs both the production and provision of hardware as well as operational responsibility for all locations. If a server fails, a replacement is dispatched immediately by IBExpert or collected directly by the customer's IT department. All master data is always synchronized immediately after startup. Our technology does not require any modifications to the router such as port shares or other location-specific settings. Any missing sales data is transmitted within a few hours back to the database server as soon as it is online.

The need for backups at the sites is eliminated completely, as all data can always be restored from the central database. All database servers are delivered mechanically sealed; there are only connections for power and for a network cable. The only accessible operating element is the power switch. Any potential misuse is prevented by lack of ports for keyboard, screen and other peripheries. In the local network, the end customer software allows unlimited access to the installed Firebird server and continues work without any restrictions should the Internet connection fail. Furthermore some parts are also used as a NAS drive and certain paths are also replicated.

Currently we are implementing all communication with full redundancy in a second data center, to allow for the eventuality of a failure at one of the data centers.

The software producer responsible for the client software can freely develop his Firebird-based software with just a few restrictions and rules. Metadata alterations are reported to us and distributed using IBExpert tools in appropriate time slots to all database servers.

From:
http://ibexpert.com/docu/ - **IBExpert**

Permanent link:
http://ibexpert.com/docu/doku.php?id=01-documentation:01-06-white-papers:firebird-replication

Last update: **2023/06/20 16:32**