Solving the limited connection Remote Warehouse problem with High Availability Decentralized Solutions

Table of Contents

Executive Overview

Today's US Government, manufacturers, distributors and even more 3PLs facilities are increasingly burdened with managing and supporting their own supply chain IT infrastructure. Contributing factors for that process could be US Military operations overseas, global economy, outsourced manufacturing, M&A, and global on-line market. All these factors usually diminish organization's ability to keep SCM in general and inventory/warehouse software systems online and accurate during critical operation times.

Inventory movement, receiving, shipping and transfers are core operations that have to continue even if connections to remote warehouses systems is not stable or are not available temporarily. The problem is further increased by the traditional methods of addressing it with a separate warehouse system for each remote location. Maintaining separate data collection systems in every location results in complicated interfaces and costly data synchronization and management projects. As a result, many IT managers find it impossible to respond to ever changing requirements of the modern business without proper tools.

A decentralized system vs. distributed system.

The primary difference is how/where the "decisions" are made and how the information is shared throughout the control nodes in the system.

Decentralized means that there is no single point where the decision is made. Every node makes a decision for its own behavior and the resulting system behavior is the aggregate response.

Distributed means that the processing is shared across multiple nodes, but the decisions may still be centralized and use complete system knowledge. [3]

CoreIMS solution for remote warehouses with limited connectivity provides full featured decentralize option utilizing SQL server toolset. This solution has been implemented and tested with US Government overseas.

3 What is the difference between decentralized and distributed systems https://medium.com/distributed-economy/what-is-the-difference-between-decentralized-and-distributed-systems-f4190a5c6462

Availability Through Replication

By employing a native feature of the SQL database called replication, many organizations are realizing, when used in the right way, their system architecture can become even simpler and provide flexible scheduled and unscheduled downtime for the entire organization. Replication is the process of sharing information to ensure consistency between redundant resources, such as software or hardware components, to improve reliability, fault-tolerance, or accessibility.[1]

In addition to replication, local inventory systems are updated to keep track of inventory records. The result is an accurate local system, which represents the results of the activities, performed by that local users combined with the data from the remote enterprise system. A robust and intelligent queuing system is required to complete this architecture.

By using the CoreIMS Platform™ and the techniques outlined in this document, many IT managers are finding they can provide a simple real-time solution to the very complex problem of availability.

1 "Replication (computer science)" Wikipedia: The Free Encyclopedia. Wikimedia Foundation, Inc. 28 January 2011

SQL Server Replication

"Replication is a set of technologies for copying and distributing data and database objects from one database to another and then synchronizing between databases to maintain consistency. Use replication to distribute data to different locations and to remote or mobile users over local and wide area networks, dial-up connections, wireless connections, and the Internet.
Transactional replication is typically used in server-to-server scenarios that require high throughput, including: improving scalability and availability; data warehousing and reporting; integrating data from multiple sites; integrating heterogeneous data; and offloading batch processing. Merge replication is primarily designed for mobile applications or distributed server applications that have possible data conflicts. Common scenarios include: exchanging data with mobile users; consumer point of sale (POS) applications; and integration of data from multiple sites. Snapshot replication is used to provide the initial data set for transactional and merge replication; it can also be used when complete refreshes of data are appropriate. With these three types of replication, SQL Server provides a powerful and flexible system for synchronizing data across your enterprise." [2]

2 Microsoft documents

https://docs.microsoft.com/en-us/sql/relational-databases/replication/sql-server-replication?view=sql-server-2017

Other Synchronization Options for Remote Warehouse

SQL server replication is a good option when there are no conflicts during the process expected. For scenarios like data aggregation, reporting etc. – replication is quite simple architecturally and does the job well. If your goal is remotes warehouse with limited data and warehouses do synchronize only a small subset of data, this is a great fit.

For truly decentralized extensive solutions, where sync rules are complex and conflicts are present or expected, it would be extremely hard to implement all business rules based on SQL Server Replication. The unfortunate fact is SQL Replication does not have good error messages either. If something goes wrong, it's very hard for the user to figure out what, where and why. There is no efficient debug mechanism and sometimes error messages are extremely misleading.

SQL is not a good language to implement complex business rules in decentralized environment. While it is possible theoretically, for complex logic unfortunately project complexity grows exponentially. This is why we had no choice but to develop our own SyncTool with sophisticated sync logic.

Since we have several existing options with CoreIMS Platform (SQL Replication for 4.0 and SyncTool for 3.6), we have to acknowledge that we have successful implementations with both scenarios.

The Remote Warehouse Dilemma – A Common Problem

The distribution and manufacturing process is no longer a one-warehouse option. The cost-savings of outsourcing is making it more and more attractive for distribution, manufacturing and USG to move some or all of their key processes overseas. While the savings to the organization justify these activities, the result is a much more complex environment for the IT organization. The need for continuous operations is more of a requirement than ever before and managers must cope with its complexity every day.

The traditional approach has changed in time from multiple warehouse systems in every location to single WMS accessible through the browser from any location to multiple warehouse systems again. Traditionally, IT managers design and build custom data synchronization routines to shuttle data back and forth from the local system to the enterprise system and back. This method addresses the short-term problem but leaves the organization with a system that has a huge number of points for failure and high vulnerability. Most organizations must employ full-time multiple highly qualified resources to make sure the synchronization routines are running properly and the data is flowing smoothly. Most importantly it is never fully reliable and not REAL-TIME.

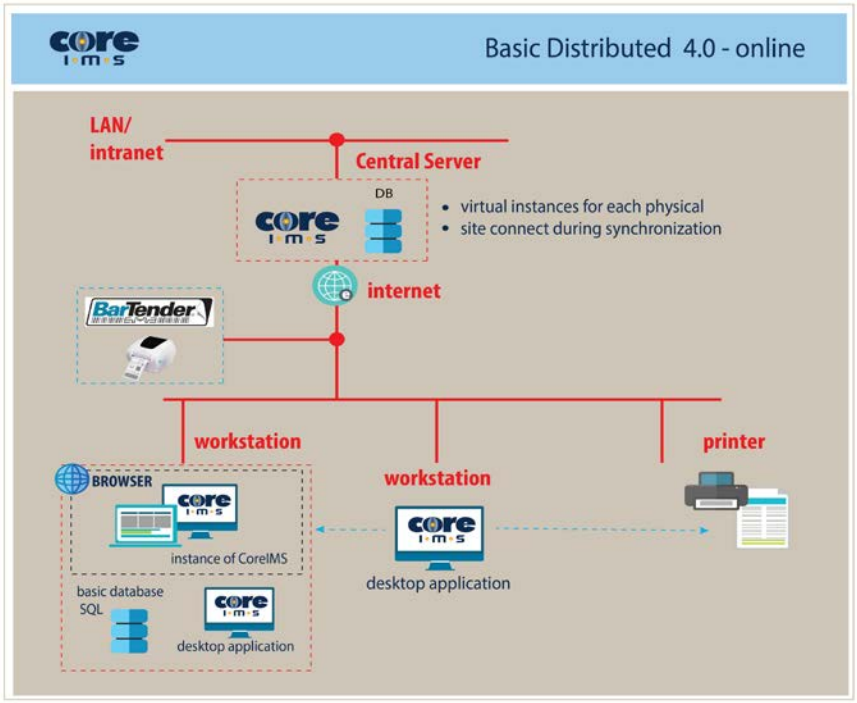


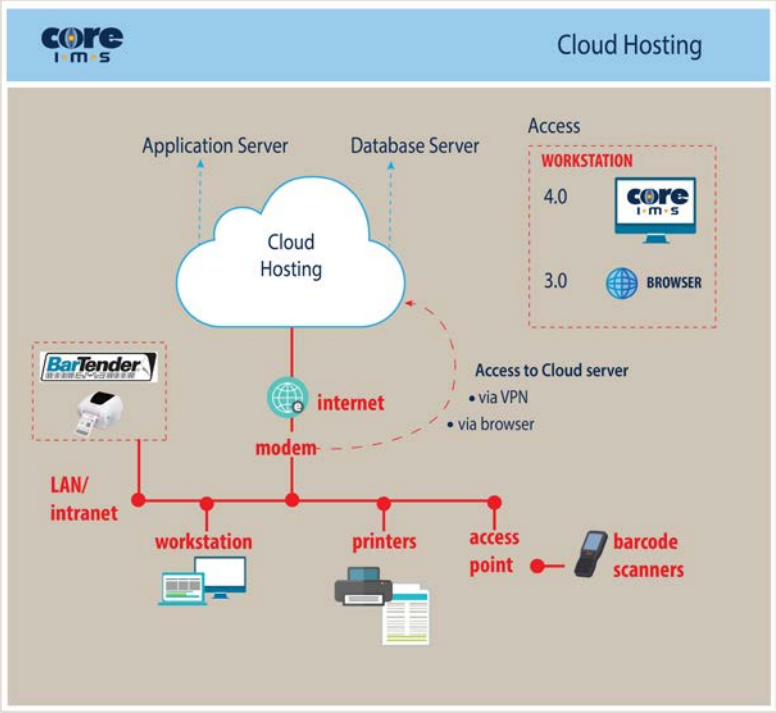
Figure 1: *Many manufacturers are moving to a distributed warehouse model*

Using Replication Techniques to Solve the Remote System Problem

Despite the fact that problems with remote warehouses are well documented, companies continually face this issue. This problem is further exasperated due to growing globalization and USG demands overseas.

The goal is to have a replicated view that is a combination of the current activity of the local warehouse combined with the activity of the enterprise at the same time. In this environment, the maintenance of multiple warehouses and costly unreliable interfaces are eliminated. Relevant activity of the local warehouse is added to the most recent enterprise data. The local warehouse user continues to operate even if the connection to the remote enterprise is down. When the connection is re-established or the enterprise comes back online, the queued transactions are executed and the local activity tables are cleared.

To achieve this goal, the system must have a lightly-coupled architecture to the enterprise to minimize the number of replicated tables required. Further, it must contain a robust queuing mechanism to keep track of the local activity performed and it must use replication and database views to bring all of the data together to provide an accurate view to the local warehouse user.



CPI Approach to The Remote Warehouse Problem

At the heart of the CoreIMS Platform™ is the CoreIMS Sync Tool with Replication option. This Sync Tool is a robust and intelligent queuing mechanism that records and queues the local activity. When the enterprise is available, it intelligently submits the transactions. Another important component is CPI Foundations for CoreIMS Suite of products including PBM and M3.

Conclusion

Warehouse data collection systems have provided value to organizations for years. However, the remote system problem has affected many IT organizations' ability to effectively manage downtime, data synchronization, and integrity. A system that employs advanced replication techniques; intelligent queuing and coupled architecture will help reduce costs while dramatically increasing system uptime.

CPI provides a robust and flexible approach that enables IT to create a simplified architecture based on these principles. The pre-tested, packaged solution of the CoreIMS Platform™, offers an unparalleled solution all based on a standard industry-leading platform. Intelligent replication techniques coupled with CPI technology can reduce complexity, improve change management, and improve the organization's ability to add additional remote warehouses in the future.