www.pipelinepub.com
Volume 17, Issue 11

Achieving Zero Downtime Data Migration

By: Bernd Pruessing

Data migration—the process of preparing, extracting, and transforming data and transferring it from one system platform to another—is difficult. Research shows that 40 percent of data migration projects are over time, over budget, or fail entirely. Data migrations have become even more complex in recent years because of new technologies, such as virtualized functions in the data center, and new use cases like automation. These make it even more challenging to extract data from the source, transform it, and load it into the new target system.

For service providers and telecom operators, management systems are especially complicated and susceptible to the common challenges of data migration. After all, their networks have complex data models that encompass many different technologies from various suppliers. However, if the right framework is followed, a zero-downtime system replacement is achievable.

Reasons for data migration

First, let’s discuss why it is necessary to migrate data in the first place.

Network inventory management systems are vital for keeping track of IT and network assets. They make it possible to know what devices are on the network, what their configurations are and how they are interconnected, when software licenses will expire, and whether any assets are reaching end of life. They also enable the planning of new equipment for new customers, planning of changes for network optimization, and the dismantling of equipment when customers leave.

Every telecommunications company will, at some time, need to consolidate inventory systems or replace a legacy inventory management solution. One common reason is to address new technology challenges such as 5G rollouts or hybrid, VNF-based networks. These technologies have requirements that many current management systems are not able to handle. Virtual network functions, 5G, FTTx, and flex-grid optical networks all require more advanced methods than most older systems can provide.

A strategic goal of many operators is to accommodate new, comprehensive planning and network automation use cases, which may not be possible with existing tools. Manual documentation, which includes Excel, AutoCAD, and cable plans, are still in use but not compatible with the increasing dynamics in modern networks and the final goal of network automation. Eliminating fragmented system landscapes that have accumulated over time is another frequently cited reason. Often such fragmented solutions work in silos, with some combination of proprietary databases and specific inventory managements for data center IP network and DWDM network employed. Having multiple parallel systems like this is a drag on efficiency.

Whatever the trigger, the data within the inventory management system is an asset that must be properly documented. This data is the single source of truth for the network and supports critical business functions such as asset management, capacity management, equipment and service planning and rollouts, impact analysis, alarm enrichment and many others. Important decisions are made based on this data, so the system's availability and its accuracy are of the utmost importance. A smooth, zero-downtime data migration and efficient go-live of the new system is essential to prevent operational damage and business impacts.

Adopting the delta migration methodology

Traditional data migration processes have inherent flaws that jeopardize migration success. These frameworks are based on sequential data dumps and multiple migration iterations and adaptation of migration rules. Each time a migration sequence is performed, the risk of content changes corrupting the final migration run is a real threat. This type of migration process also requires system downtime between shuttingdown the old system and cutting over to the new, and there is a high risk that this final migration will fail. Any system downtime should be avoided at all costs, as the longer the system downtime, the higher the impact on operational processes and ultimately the business.

The preferred process uses a continuous delta migration methodology. In this approach, the new system runs concurrently with the existing system and only the delta between the existing and new databases is migrated at each migration run. This eliminates the need for downtime as the two databases are synchronized. This is done by comparing the full data between the source and destination platforms but only creating and changing data that are new or different. As the read and compare operation works much faster than the create operation, smaller, faster data transfers and migration cycles can occur to enable the continual adaptation of migration rules.

Because both systems run in parallel and data is continuously being aligned, the ongoing delta migration process does not require any downtime. When the migration quality reaches the required level, the old system can simply be switched off and users can proceed with using the data in the new system. A zero-downtime migration will have been achieved.

Having an integration framework available is a precondition to efficiently implement a continuous delta migration project, which makes a zero-downtime system replacement possible. Implementing all the mechanisms from scratch is far too time-consuming and expensive an endeavor. The framework should support openness and configurability. Configurability is especially important because it eliminates the need for costly and time-consuming programming, a hallmark of the overall solution. The target system must allow the user to adapt the models as much as possible to the models of the solution to be replaced. This makes the move of end users to the new systems much smoother, as they easily can find the data in the new system according to their experiences with the old. It also significantly lowers the complexity for the data migration.

This framework also puts the user in control to prevent vendor lock-ins. When the new inventory data solution is standard off the shelf, open and configurable, the customer can decide how involved they want the solution provider to be. The customer can use their preferred integrator, do the integration work themselves, or take advantage of consulting services and professional development services of the solution provider for software extensions, if required. Execution flexibility is a desirable option to have available.

Components of a successful migration framework

A successful, zero-downtime system replacement can be realized with a framework that uploads and aligns data with the new application. The framework should encompass interfaces with data sources like NMS, EMS, Managers, BSS/OSS or any other database. This framework is a software concept that will govern the entire migration process, which encompasses the upload, transformation, and alignment of any kind of entities, attributes, or relations. The alignment process can be run based on a predefined schedule or on demand. The framework should log the results of the process and inform users of successful data uploads, data clashes or any other errors that may need to be handled by a planner or operator.

Beside alignment rules, which can be defined in a graphical ETL tool or written as JAVA code, the framework should provide configuration options—for example, mapping tables to map source data to the new system, or black- or whitelists to include and/or exclude entities from migration.

The most important feature of such a framework is the calculation of deltas between the source and the new target system. Such delta calculation would speed up the alignment process between the source systems and target system dramatically, as only missing entities must be created in the target system. This accelerated procedureallows the running of an alignment process several times a day. For each, run migration rules and mapping tables can be adjusted according to the latest results. The process can be repeated until the required migration quality has been achieved.

Other migration challenges can be addressed through the following processes:

Inventory grouping

In practice, migrations are more complex than the scenarios described above. Often, several source data bases will need to be migrated, which means data from the different sources must be merged into one system. Even if there is only one migration source, additional data will need to be reconciled from NMS/EMS in the future. The incorporation of this data must be considered.

Besides delta reports, the migration framework should have the ability to match data from different sources via identifiers or rules based on matches between different attributes such as locations, slots or port numbers.

Logging and reporting

Logging and reporting enable data statistics to be created and the progress of the migration project to be tracked. This includes not only the number of migrated objects, but also the classification of errors and statistics about open migration bugs. Project leaders and migration architects can then identify problem areas and fix migration rules and data quality issues faster, which in turn streamlines the entire data migration process and ensures the required migration quality is achieved.

Reconciling data from NMS/EMS

If the reconciliation of interfaces already exists in the legacy system, either a migration of these interfaces to the new system is required, or the operator must add additional reconciliation interfaces to other NMS and EMS. Reconciliation is an important function for keeping the data in the inventory system, which is the digital twin of the network infrastructure, up to date. Reconciling as much data as possible from the network is a major goal of any inventory solution.

Because a zero-downtime migration approach is similarly implemented as a daily reconciliation interface from NMS/EMS systems, the same framework is used to support migration of data from a legacy system as well as reconciliation of data from NMS/EMS or any other management systems. The functions previously explained are applicable for both.

Migrating interfaces to new OSS/BSS

Like NMS/EMS, some operational or business support systems may have interfaces with the legacy inventory system. These interfaces must be migrated as well. There should be multiple options for OSS/BSS northbound integration as use cases can be very different. A broad interface with access to all data is crucial, one that self-extends in case of model changes in the database and delivers event-driven updates of the northbound systems.

A standard REST interface should provide query, create, update, and delete functionalities for all of the different objects in the database. The interface should also be self-extending, so if the administrator adds new entities or attributes to the database, the REST interface will be automatically extended and provide query, create, update, delete functionalities for the new entity and corresponding attributes without programming. An event API interface should be REST-based as well and should be able to send information to configurable REST endpoints. The interface logic should also be able to catch events generated by user activities.

Overall, data migration and data reconciliation can be challenging tasks. Having support from an experienced service provider and the right frameworks available significantly lower the risk of the project, allow for zero downtime, and enable an efficient go-live of the new system. This is essential to prevent operational damages and business impacts.