TREXIN INSIGHT PAPER

August 2017

MOVE GROUPS WITHIN AN APPLICATION MIGRATION PROJECT

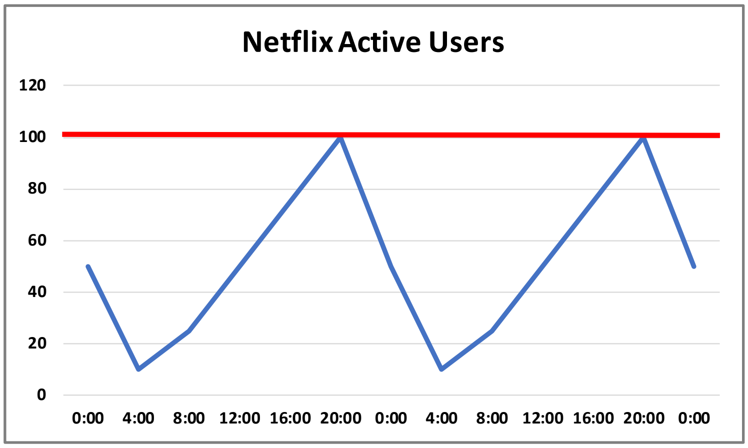
Offering insight into application dependencies and maintaining them during a migration to the cloud.

Before we jump into what a Move Group is within an Application Migration, let's first explain why this type of migration exists and the massive benefit it can have on an organization. If you're thinking this is going to be another Cloud article, then you're 100% correct. I'll be discussing the importance of Move Groups when migrating hundreds of applications from current state data warehouses to the cloud.

You may be asking yourself, what is the cloud and what makes it so special? Why are so many companies, which range across various disciplines, striving to integrate their entire IT infrastructure on it?

The cloud is a network of servers that collectively facilitate flexibility and scalability, whether you're running applications or storing pictures from your iPhone. Okay, so what does *this* mean? In simple terms, think of thousands of servers sitting in a warehouse (known as a datacenter) that millions of people in your area can use to store their information. Rather than buying an external hard drive or using a floppy disk (what's up, 1997?), someone can simply say, "Hey, Apple, I would like 5 more GB of storage to keep my pictures from the summer of 2017." They click a button and a server in that datacenter provides the space.

Now that we understand what the cloud is, you might be wondering why is it important and why should I care. You may have noticed above that I used the terms "flexibility" and "scalability." Apart from being flashy 'buzzwords,' these concepts define the purpose in which the cloud can serve in your everyday life. The cloud enables flexibility with infrastructure, whether it is utilized by a single individual or a 100,000-person organization.

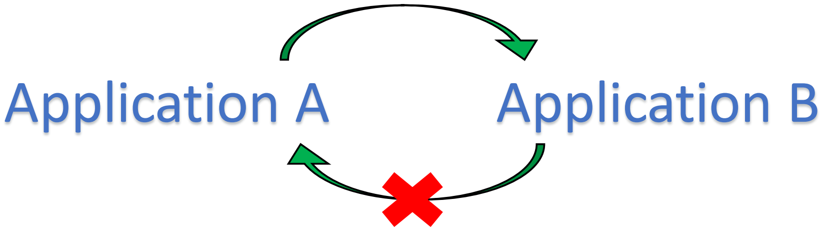


Think of Netflix as an instructive analogy. Most working-class adults tend to watch their shows or movies on weekends or during weeknights. Accordingly, there will not be as much activity during weekdays and during the middle of the night. To illustrate this phenomenon in (pretend) numbers, let's say Netflix has 100 active users during peak hours; and, at their lowest, they experience only 10 active users. If Netflix was running on the current state data warehouse, they would always have to pay for their peak utilization (100 active users) regardless if only 10 active users were on the application. To extrapolate this to a larger scale, think of an organization that has 1,500 applications and the potential savings the cloud can offer. Flexibility is key – why pay for what is not being used? Now imagine Netflix doubles in size and instead of 100 active users, 200 people are using the application during the weekend. Netflix can implement the same practice that we as individuals use when we need more space on our cell phone – they request and instantly receive more capacity to support the increase in active users. When a company is growing fast, the cloud enables scalability on demand.

Now that we have a firm understanding of what the cloud is, let's dive deeper into my sweet spot within an Application Migration… the Application Backlog. My current role within an Application Migration is managing the complete list of applications that need to either migrate to the future state cloud environment or die on the vine. This list of applications is comprised of numerous attributes that identify and prioritize specific applications to migrate. Our main attribute is the application's "business value," or the value that migrating an application from its current state infrastructure will bring to the organization. There are two indicators of business value opportunity: the number of AIX servers (physical hardware) and WAS licenses (application server licenses). Other factors (which are just a subset of attributes) include the level remediation needed to migrate from the current infrastructure to its future state cloud environment, Disaster Recovery requirements, and finally if the application is a part of a Move Group. By leveraging the application attributes, I am able weigh each application's migration value and rank them in a prioritized order. This enables our planning team to put a lens over a large mass of data and easily select restriction-free applications to potentially enter remediation (and the rest of the migration process).

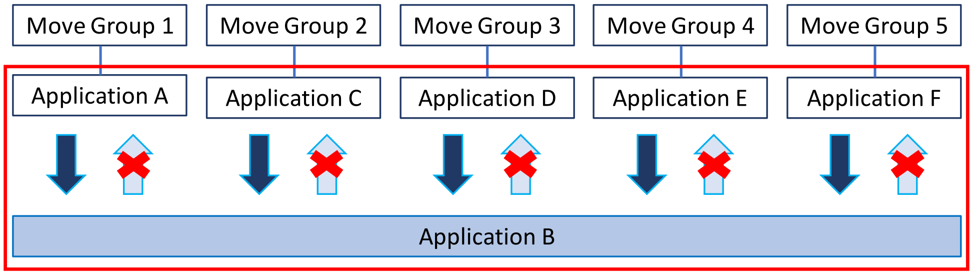
Now you might have thought to yourself "what is a Move Group?" when I mentioned it in the previous paragraph or maybe you just skipped over it because you've lost interest. Regardless, it's time to granularize this topic and explain its importance within an Application Migration.

A "Move Group" is a group of applications that contain a dependency and therefore must be migrated together. For example, you might have an application that is 100% a front-end application, such your Chase bank app on your phone. Call it "Application A." Well Application A might pull data strictly from a backend application that feeds a group of five different applications; call this "Application B." Hypothetically, Application A can't function without the data from Application B, therefore it is *fully dependent* on Application B and must be migrated together. There are cases, however, where Application A is dependent on Application B (as we see above), but Application B *is not dependent* on Application A. This means Application B could be migrated any time before Application A; they wouldn't necessarily have to go together.



Within our Application Migration program, we refer to Application B above as a "Child Application." Each Move Group will have a parent application that is dependent on every one of its child applications to migrate. In addition, one child application can also be a part of numerous Move Groups as stated in my example above. We can call my example above Move Group 1 ("MG1") and, as described, Application B feeds a total of five applications with its backend data. What this means essentially is that Application B is the child application in five different Move Groups. When it comes to ranking applications, it's extremely important to take this into consideration and understand that five parent applications are dependent on one child application. Why? Because if your Prioritized Backlog, or ranked list of applications has the child application tied to MG5 and that Move Group has the least value out of the five, it will be prioritized last. How I rank Move Groups is the child application receives a decimal value of its parent application (i.e. Application A = 10; Application B = 10.1). If Application B is tied to MG5, Application A would sit alone at 10 potentially leaving it unclear to the planning team that Application A is dependent on a child application, or Application B. Therefore, it's critical for the parent application with the highest priority (and highest rank) to have the child application of multiple Move Groups ranked with it in the Prioritized Backlog.

A Move Group does not have to be a two-way dependency; just because Application A is reliant on Application B to migrate does not mean the opposite is required. Relating to the Move Group example above, Application B is the child application for five Move Groups. We now understand this means each parent application from the five Move Groups is 100% dependent on Application B to migrate, however, this doesn't mean Application B is dependent on any of the five parent applications. In this hypothetical, Application B is an extremely valuable application. Application B's independence from any of the five parent applications (or Move Groups) allows it to migrate individually. Rather than giving it a decimal ranking, it is given an integer and can be migrated at any time before its parents.



Every application will eventually be migrated and it's more than likely the Planning Manager will discover that Application A is dependent on Application B from the IT Product Manager. However, the goal of the Backlog Manager is to produce clean, precise, and unequivocal data within the Prioritized Backlog. The cleaner the data, the more accurate expectations are with Product Managers and thus, the more value contributed to the planning team. In large organizations, Move Groups are as ubiquitous as they are dense in size. Breaking them down and comprehensively analyzing the dependencies takes a special application architect and an effective relationship with the Backlog Manager. The more synergistic the team, the smoother the process. The smoother the process, the faster applications enter remediation. The faster applications are remediated, the quicker an application migration program realizes value.

This TIP was written by Steven Bills from our Aligned Enterprise Capability and Products & Distribution Practice. Steven welcomes comments and discussion on this topic and can be reached at steven.bills@trexin.com.