Gartner

Research

Publication Date: 8 January 2010 ID Number: G00171622

Mainframe Modernization: When Migration Is the Answer

Dale Vecchio

Mainframe migrations can be executed in a number of ways. Which choice is best will depend on a number of factors, which we identify in this research.

Key Findings

- The size of mainframe organizations evaluating migrations of some or all their workloads, as measured in millions of instructions per second (MIPS), has grown dramatically in the past two years, to around 2,000 to 3,000 MIPS and higher.
- Rehosting has emerged as a relatively low-cost, low-risk approach to shifting workload quickly and saving on hardware and software costs.
- Code transformation technology has improved dramatically over the last several years and is often used to shift workload to newer platforms and languages, while preserving its inherent business value.

Recommendations

- Consider rehosting when the existing applications still represent valid commercial business processes or government agency missions.
- When shifts to modern platforms also imply greater interest in newer languages, evaluate code transformation solutions to evolve the application from legacy skills dependencies.
- Rewriting existing business applications offers the greatest flexibility, albeit at the greatest cost and risk.

ANALYSIS

Before migrating workloads from legacy mainframe environments to newer, more-modern architectures, organizations should consider the difficulty of extricating themselves from decades of architectural dependence. Mainframe migrations have a cultural aspect that must be considered. Although many technical differences affect such migrations, the impact of change on people and the processes they use to deliver and operate applications is significant. The differences between the mainframe platform and its related application and operational architectures, and Windows or Unix/Linux environments, need to be rationalized as part of the migration effort.

Companies have invested time, money and effort into these applications, and should not expect to extricate themselves from these systems in a few months. Whether organizations migrate in one step, or gradually shift workloads to new platforms, they should plan on a one- to three-year migration timeline. Larger mainframe environments can be aggressive, but more-conservative approaches will increase the time frame and elongate the return on the migration investment. In addition to the technical issues, business support must be nurtured. Trying to justify and implement such a project based solely on technical merit may increase risks, lengthen the time frame and reduce perceived value.

Application Portfolio Management Is the First Step Toward Migration

Application portfolio management (APM) is a critical first step for any IT modernization activity. It provides an objective framework for making modernization decisions, and can also provide sufficient information to help plan a migration effort. When using a top-down approach, APM helps organizations objectively identify modernization strategies for their legacy application portfolios. Top-down means starting at the high-level definition of business applications, determining their cost, risk and value, and working downward, collecting more-detailed levels of knowledge sufficient to make modernization decisions. Alternatively, APM can be executed from the bottom up, collecting more-detailed information about applications, their components and their interrelationships. This detailed information can be built up into a more general decision-making level, thus the bottom-up characterization.

It is also important to build mainframe migration plans. In addition to finding the pieces and parts of the existing portfolio and defining the relationships between them, this step can identify application candidates for retirement. Eliminating applications from the migration process is beneficial. For these applications, the planning is on retirement, not migration. For many companies, newly developed or purchased applications have already been added to the portfolio. Subsequently, integration with current mainframe applications may be a concern. These new applications on distributed platforms may also become the driving force behind modernization or migration alternatives, rather than APM. We often see a specific application used as a proof-of-concept to understand the impact of such a migration on the technology, the development and the operational staff, and the processes associated with building or operating these applications. Once understood, gradual migration of the remaining applications can occur.

Rehosting, Code Transformation or Packaged Migration

When a decision is made to move workload off the mainframe platform, the options available include rehosting the application, transforming the code base to a more modern language and runtime environment, or migrating to a packaged solution. Based on our interactions with clients, Gartner sees a growing desire to move to packages, if possible.

Packaged solutions are *not* acceptable alternatives for all applications, and we interpret this desire to reflect a broader wish to eliminate as much custom-made code as possible from application portfolios. When migrating workload from one set of languages and architectures (e.g., COBOL and procedural/mainframe) to another (e.g., Java and application servers), take advantage of the myriad solutions that run on Linux, Unix or Windows platforms. Replacing portions of the existing handwritten portfolio (for example, batch reports with commercial off-the-shelf solutions, such as Crystal Reports) is desired. There is much less need to develop code to produce a report when so many other options exist. As another alternative, utilizing a data warehouse or business intelligence (BI) query tool may eliminate the need for batch reports all together.

Clients need to think differently when migrating a workload from the mainframe to Linux, Unix or Windows. They should leverage the capabilities of these platforms, rather than simply trying to replicate the mainframe environment in its entirety. This may reduce the size of the handwritten legacy code that needs to be migrated. Rehosting and/or code transformation can then be considered for the portion of the portfolio that remains. Rehosting solutions (see "Mainframe Rehosting Products Offer Lower Risks") provide a viable approach to shifting workload more quickly and with minimal change. These solutions work best for COBOL, although some support other languages. Rehosting applications that use pre-relational databases, such as CA IDMS, CA Datacom, Software AG's Adabas, IBM's DL/1 or Cincom's SUPRA PDM, have an additional complexity, because they must be migrated to a relational database on Linux, Unix or Windows platforms. Depending on the approach taken, significant change is required to the rehosted source code, since, potentially, the entire database navigation logic could be affected. Sometimes, transparency solutions can simplify these database migration efforts. For example, Alchemy Solutions' NeoData product enables the migration of VSAM data to Microsoft SQL Server on a Microsoft Windows platform, without requiring any modification to the VSAM calls that currently exist in the COBOL programs. Tiburon provides a similar approach for CA Datacom conversions.

Finally, code transformation solutions can be used to convert applications written in one language, (such as COBOL) to more-modern languages (such as C#, Java or C++). Code transformation solutions all share a common technological approach. Vendors have developed solutions that parse source code into an intermediate representation. This representation is some form of an abstract syntax tree (AST). ASTs represent the original program and logic, *independent* of its original source syntax. Vendors can then use language grammars, similar to those used by compilers, to generate new source code in the new desired language. This approach, while technically sound, may yield some problems. When migrating from one procedural language (such as IBM's PL/1 or Software AG's Natural), to another (such as COBOL), this approach is valid. However, when migrating from one procedural language (such as COBOL, PL/1 or Natural) to an object-oriented language (such as Java, C#, Visual Basic or C++), some additional steps are required. Most vendors transform the *syntax* of the language, and *then* re-engineer the program to match the form of true object-oriented programs.

Restructuring a procedural program with shared working storage, a trademark of procedural programs to object-oriented classes and methods, requires additional effort. Vendors such as Annubex, Ateras, EvolveWare, MigrationWare, MetaWare, The Software Revolution (TSRI) and Trinity Millennium Group (Trinity) provide code transformation solutions for many legacy languages. When evaluating code transformation vendors, consider those that transform the programs to the new application architecture, if this is desired. Trinity is showing significant capability in this regard, particularly when moving to Microsoft's C# or Java.

Creating a High Quality-of-Service Runtime Platform

While much of the focus of modernization decisions is on the applications, the operational infrastructure is of equal importance. For organizations that depend on the mainframe, infrastructure is key. When moving to any Linux, Unix or Windows platform, organizations must be prepared to create not only a hardware infrastructure environment, but a management discipline with this infrastructure that will create a highly reliable and available runtime platform. This does not come "out of the box." Many companies have already developed this capability, as they have been running mixed workload environments for some time. For those that have not, this is as important to the modernization decision as what language to choose, what package to use or what operating system is desired. It *is* possible to create a highly reliable environment, despite the perception that Linux, Unix or Windows platforms are *not* as reliable as a mainframe. The mainframe is clearly accepted as a highly reliable platform, and creating one in a Linux, Unix or Windows environment does require different approaches, but it is possible to get very close at a lower cost. The challenge is that no single vendor controls the whole stack, as is the case with the IBM mainframe.

Linux, Unix or Windows environments are a mix of products from several vendors, such as HP, Fujitsu, Oracle, Microsoft and IBM. Creating an operational management environment also requires a mix of different products from different vendors. Furthermore, the approaches to many of the traditional management disciplines of source/change control, print management, billing/job accounting, network management, system management and job scheduling are different.

When the Mainframe Is the Answer

IBM's Linux for System z is becoming somewhat of a growth opportunity for the company. Initially, much of the workloads for this environment came from distributed servers that handled utility workloads, such as file serving, print serving, Web serving or e-mail applications. More recently, we have seen an increase in the use of Linux for System z for Oracle, SAP applications and WebSphere. Typically, most compute-intensive Linux applications are not run on System z. The challenge for IBM is to grow Linux for System z workloads without impacting traditional z/OS software revenue. If customers are driven to modernize to newer application environments or languages, then IBM would certainly prefer that to be WebSphere and Java on Linux for System z, rather than someone else's hardware platform.

Gartner has seen IBM have greater success with Linux for System z in its largest mainframe customers. We have found less interest or commitment to Linux for System z in the lower end of the System z installed base (less than 1,500 MIPS; see "Mainframe Modernization: When the Platform Is the Solution").

Do Not Underestimate the Political and Cultural Implications of the Migration Decision

Despite the many surmountable technical challenges associated with a mainframe migration, organization should not underestimate what will be serious political and culture resistance to such a change. IBM will not stand by idly watching continued migrations to other platforms. It continues to make significant investments in this platform and will do everything it can to defend its continued validity in the marketplace. Expect IBM account teams to attempt to delay or derail migrations by calling into doubt the ability to match mainframe service levels, the performance of transformation and service partners, and by taking a variety of pricing actions. Business partners that challenge IBM's dominance will be discarded, losing any advantage the business partnership provided.

Baby boomers in your organization, often a significant portion of the mainframe workforce, will be threatened by migration decisions. They may become roadblocks to your efforts, demonstrating little cooperation or support. For them, this may be threatening to their continued employment and will be seen in a very negative fashion. Don't deny their feelings; yet don't allow them to persuade you from modernizing (see "How Today's Application Development Organizations Can Manage Retiring IT Skills").

REGIONAL HEADQUARTERS

Corporate Headquarters
56 Top Gallant Road
Stamford, CT 06902-7700
U.S.A.
+1 203 964 0096

European Headquarters
Tamesis
The Glanty
Egham
Surrey, TW20 9AW
UNITED KINGDOM
+44 1784 431611

Asia/Pacific Headquarters
Gartner Australasia Pty. Ltd.
Level 9, 141 Walker Street
North Sydney
New South Wales 2060
AUSTRALIA
+61 2 9459 4600

Japan Headquarters
Gartner Japan Ltd.
Aobadai Hills, 6F
7-7, Aobadai, 4-chome
Meguro-ku, Tokyo 153-0042
JAPAN
+81 3 3481 3670

Latin America Headquarters
Gartner do Brazil
Av. das Nações Unidas, 12551
9° andar—World Trade Center
04578-903—São Paulo SP
BRAZIL
+55 11 3443 1509