The Power of Open

Overview

It is a great irony that one of the earliest examples of open-source software was developed for the mainframe. In 1959 General Motors contributed its GM-NAA I/O operating system code to the SHARE operating system – a platform designed to facilitate the exchange of software. This approach paved the way for batch processing innovation, yet was discontinued just three years later.

So began a reign of over half a century for one of the most proprietary environments in enterprise computing. Many companies, within the vanguard of the then new software business, took advantage of the multi-million dollar opportunity this lock-in presented. Even today, some of the world’s largest companies still count license fees for proprietary software as their primary source of revenue.

However, the Internet, which changed a great many things, also changed forever the realities of software development. The ease with which programmers could collaborate led to an explosion in software sharing, which became the open-source movement we know today. This movement was described in academic terms in Harvard Business School’s The Principles of Distributed Innovation1, and also by Bill Joy, co-founder of Sun Microsystems, who described it succinctly:

“...no matter who you are, most of the smartest people work for someone else”2

Some of the world’s largest organizations are caught in the nexus of dependency on legacy applications and supporting tools, used under expensive licensing models, which provide no more functionality than open-source solutions. Consequently, these organizations suffer from:

- Stifled innovation compared with born-on-the-web competitors.
- A skills shortage that threatens to erode the very bedrock upon which organizational IT is built.

---

1 The Principles of Distributed Innovation, 15 October 2007, Karim R. Lakhani & Kill A. Panetta, Harvard Business School
For organizations still dependent on legacy applications for system-of-record operations, the benefits of open-source are significant. This document explores both the obvious and non-obvious benefits of rehosting mainframe applications into a predominantly open-source architecture.

What Does It Mean to Rehost Mainframe Applications onto an Open-Source Architecture?

A mainframe environment consists of customer applications, usually written in COBOL or PL/I, and a huge stack of proprietary software that exists to enable and keep these applications running smoothly. This stack includes things like operating systems, database and transaction management systems, backup and recovery utilities, performance management tools, scheduling and automation systems and so on. These licensable software artefacts are hosted on a unique hardware architecture available from just a single vendor.

The proprietary operating environments serve to tie customers into complex, costly contract cycles, but this is just the tip of the iceberg. Whether used for scheduling, compilation, monitoring and performance or capacity planning, each strand of the intricate web of (highly expensive) third-party software locks customers further into this environment.

No organization would choose such a model in the 21st century. By rehosting mainframe applications and data onto an open architecture, we mean replacing everything except the customer applications with open, ubiquitous alternatives – alternatives that are either open-source or based on more open architectures such as x86 hardware. This approach presents a significant opportunity for organizations willing to embrace it, as we’ll go on to explain.

However, moving customer mainframe applications or data to open systems can be a crunching gear change for organizations. Attempts are often made to seek solutions in the open-source world for enterprise computing challenges that are almost identical in implementation to their proprietary mainframe counterparts.

Such an approach misses one of the key points of open-source: open-source enjoys significantly more innovation than proprietary systems. Solutions in the open-source world are likely to be more advanced on a number of levels, but they will also be different. Witness the innovation around TCP/IP versus the proprietary peer to peer equivalent of LU6.2.

Furthermore, solutions in the open-source world enjoy high-levels of participation and available talent, and will continue to do so. Which do you think would attract a higher response rate, a job advertisement for cluster design expertise based on OpenStack, or on Sysplex?

The correct model for rehosting mainframe applications on open systems is to take maximum advantage of all that the open world has to offer. Bringing over the proprietary baggage of the mainframe defeats the object of a mainframe rehosting exercise. There are few things, in infrastructure terms, that can be done on a mainframe, that cannot be achieved in an open environment. Furthermore, open alternatives will be better, faster, cheaper, and achieve greater levels of innovation over time.

An Open-Source Strategy

In its recent paper What Every CIO Must Know About Open-Source Software, Gartner observes:

“Historically, open-source software solutions have been most competitive in markets where vendors struggle to create competitive differentiation between their respective product portfolios. This includes very mature ‘legacy’ markets dominated by megavendors that have collectively matured the market to a point where even the weakest competitor is ‘good enough’ for most customers.”

When development capability is available from around the world, rather than just inside a corporate IT department, or from a single software vendor, innovation and integration accelerates

Gartner, What Every CIO Must Know About Open-Source Software, 30 March 2017 (G00301608), Mark Driver
dramatically. This in turn attracts more talent, which creates further innovation. A truly virtuous circle. For this reason, open-source software has become the predominant model for technologies such as: webserver, big data, analytics, mobile development, cloud infrastructure and sophisticated web-based applications, which can have limited options for mainframe customers, or be expensive to implement due to the requirement for proprietary integration software.

Examples of Innovation and Modernization Enabled by An Open-Source Architecture for Legacy Applications

What follows is a brief outline of some of the benefits of rehosting mainframe workloads into an infrastructure based on open systems. It is by no means exhaustive, but illustrates a small proportion of the innovation available to companies that make this switch. While some will argue that none of the items listed are beyond the capacity of a mainframe, that argument misses the key point. To enable a mainframe for these kinds of workloads requires further licensing of proprietary software.

**Mitigating the Impending Skills Crisis**

One of the largest drivers behind mainframe rehosting projects is the impending skills crisis facing organizations whose legacy systems are serviced by a diminishing talent pool. As BMC stated in a 2017 survey, “44% of respondents indicate that staffing and skills are key challenges due to the changing workforce, and 36% of respondents indicate this is a priority in the coming year.” Organizations that do not make this move are in danger of rendering their core IT infrastructure inoperable by a modern workforce.

When customer applications are rehosted on open systems, the fundamental operating architecture becomes far more accessible to a modern workforce. It is true that original source code knowledge – PL/I, COBOL, Assembler and others – is required for maintenance, but educating a modern developer in these fields is trivial; the coding language isn’t the key challenge. Instead, issues lie within the myriad of arcane development procedures with which modern developers have little or no familiarity. “Simple” procedures such as compiling a program require interaction with a character-based development tool and batch processes that have no parallel in the modern world. By moving the application development environment to a modern alternative such as Eclipse or Visual Studio, within a modern DevOps Toolchain, the development and maintenance backlog that plagues many mainframe organizations evaporates almost overnight.

**Leveraging Development and Scaling Benefits of Containers**

Once legacy applications and data are liberated to open systems, organizations can begin to incorporate them into their datacenter transformation strategies. One example is in the use of containers.

Born-on-the-web companies make widespread use of open-source container technology to drive constant innovation. Containers provide the perfect balance of isolation and resource usage. This isolation provides well-documented benefits for development speed, horizontal scaling, security and resource optimization.

Container technology, while very widely implemented, is entirely absent on the traditional legacy mainframe operating system. Consequently, development and scaling pressures, which come from expanded digital internet services, cannot be satisfied on mainframes with anything like the efficiency of container-enabled platforms.

**Introducing Familiar DevOps Tools to Legacy Applications**

Programming languages come and go, but the desire to deliver software more rapidly is a constant theme. Much emphasis has been placed in recent years on agile development, DevOps, shift-left testing, source-code management; the list goes on. All this innovation has occurred off-mainframe. For today’s programmers, such tooling is table stakes and is more important than the programming language they are asked to work in. By moving mainframe applications and their development backlog into this modern development architecture, an organization can now tap into a pool of programmers that is several orders of magnitude larger, and can deliver updates faster and more regularly.

---

4 12th Annual Mainframe Survey of more than 1,200 executives and technical professionals, 2017, BMC Software
Turning Sluggish Applications and Data into Agile Tools Using Microservices

Although the notion of service-orientation has been around for a while, and indeed quite widely implemented on mainframes, the addition of containerization to the discussion has enabled the concept of Microservices. Unlike service-oriented mainframe applications, Microservices are independently deployable, scale horizontally and are released with automated processes. They are small and fine-grained from the perspective of the service consumer. The task of transforming monolithic mainframe applications into Microservices can be daunting unless they can be containerized, delivered and deployed using a modern DevOps Toolchain. Migration to a modern data architecture also enables some of the data barriers to Microservices to be overcome. The virtues of a Microservices-enabled architecture cannot be overstated. It has quickly become the go-to pattern for enterprise applications required to support digital channels for product and service delivery.

Cloud Exploitation

Once in a modern environment, another (perhaps even more attractive) benefit to the organization is that of flexible and scalable datacenter management through private or public cloud deployment. For organizations formerly locked into the contractual and cost complexity of the mainframe’s “on/off capacity on demand”, applications and associated data can now execute within containers on cloud infrastructure with any provider. AWS, Microsoft and Google, amongst others, have demonstrated the scaling, cost and agility benefits of utilising open-source solutions running on flexible cloud architectures. Companies that do the same can react to periods of heightened business pressure automatically, with a range of providers to choose from, significantly reducing cost and contract complexity and improving uptime and performance.

Real-Time Analytics

No one ever argues for slower access to corporate data for decision support. The faster a business can respond to trends the better chance it has to capitalize upon those trends. Consequently, real-time analytics has become a holy grail for management information. The mainframe sticks out like a sore thumb in pursuit of this strategy. Transactional updates that target impenetrable, proprietary database systems make it difficult for this information to feed analytics in anything close to real-time.

However, once these transactions are running in an open environment, using a database such as PostgreSQL to store updates, the organization can benefit from a plethora of tools which put this critical information into the hands of key decision makers at the same speed as other source data.

Security

The mainframe is a fine example of security by obscurity. People often mistake its relative network isolation, protected by extensive off-box perimeter security, as evidence of its inherent relative security strengths. In certification terms, both RedHat® Enterprise Linux and z/OS® enjoy comparable accreditation. Linux® security has been forged in the white-heat of Internet connectivity. Extensive hardening of the kernel and the user space is commonplace. Linux is routinely penetration-tested around the world on an internet scale, and many vulnerabilities are patched long before they can be exploited. A fine-grained privilege system ensures that access rights are more appropriately assigned. The Mainframe security environment enjoys none of the above as it has never needed to evolve such characteristics. And, as in all previous cases, the availability of staff trained to manage Linux security far exceeds those that understand the arcane world of mainframe security.

A Software Defined Mainframe Makes It All Possible

At LzLabs our mission is to create revolutionary software solutions, leveraging the creativity of open-source innovation and the power of cloud computing to reduce the risk of legacy application modernization. The LzLabs Software Defined Mainframe® (LzSDM), eliminates the need to modify and recompile mainframe application source code; and preserves mainframe data in its native encoding format when migrating mainframe applications to an open-source architecture. LzSDM® is “standing on the shoulders of giants” such as Linux, PostgreSQL, LDAP and, more generally, open-source software (OSS) in its implementation of APIs functionally-equivalent to their
legacy mainframe counterparts. As The Principles of Distributed Innovation\textsuperscript{1} illustrates, OSS has been developed and tested by tens of thousands of people. Additionally, it has been optimized and stabilized across hundreds of thousands (and in some cases millions) of installations around the world, handling workloads of all shapes and sizes.

LzLabs decided it would make no sense for LzSDM\textsuperscript{®} to perpetuate the proprietary infrastructure that extends the lock-in to a mainframe, while providing no competitive differentiation for the applications. The use of more common open-source technologies makes the LzSDM environment look familiar to new hires. It helps our customers attract talent to sustain operations; more rapidly develop applications and reduce dependence on proprietary solutions. Finally, the LzSDM provides a reliable, manageable platform for enterprise-scale system of record applications that will continue to evolve thanks to the power of a global open-source development network.

Find out more at www.lzlabs.com
About LzLabs
LzLabs GmbH is a software company that develops innovative solutions for enterprise computing customers including the LzLabs Software Defined Mainframe®. The LzLabs managed software container provides enterprises with a viable way to migrate applications from mainframes onto Red Hat Linux computers or into cloud environments such as Microsoft Azure. The company was founded in 2011 and is headquartered in Zürich, Switzerland. Learn more at LzLabs.com