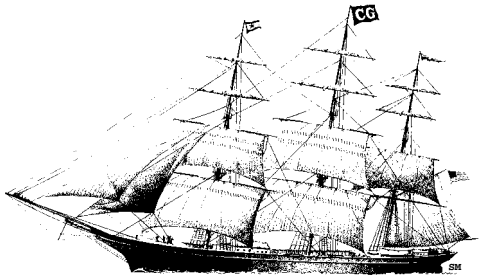
THE CLIPPER GROUP
Captain's Log™



Navigating Information Technology Horizons℠

Published Since 2001 Report #TCG2004028 March 30, 2004

The Beginning of I.T. Civilization – IBM's System/360 Mainframe

Analyst: Mike Kahn

Management Summary

The computer that most of us know as *the mainframe*, the IBM *System/360*, was born more than forty years ago.[1]T So, what's the big deal? Does anyone care? What the heck is a mainframe, anyway? My phone has been ringing for weeks, and the press wants to know whether IBM's fortieth anniversary celebration is a nostalgic trip down memory lane by over-the-hill baby boomers, or something really worthy of consideration. So bear with me on a somewhat nostalgic journey to answer these questions. If you want to know why I am nostalgic, see my confessions in the shaded box at the top of Page 3. *Read on, it's worth the ride*!

Part 1 – In the Beginning

This story really begins more than 40 years ago, in what must seem like the dark ages of computing. Yes, it was called *computing* back then, a gerund, or verb turned into a noun. Computing's origins were largely mathematically based, replacing scores of humans at bulky electromechanical calculators doing computations. *Data processing* was a phrase just coming into vogue. It employed computing to replace what were largely clerical tasks – bookkeeping, accounting, and report generation. (Bear with me!)

Computing and *data processing* were largely seen as separate and incompatible kinds of applications. Each was separately conceived and the procurement of a computer was justified based on that singular need. There were many computers of this singular type being sold as the 1960s unfolded. IBM had at least four, with additional ones for the military. There were different systems for large and small enterprises, and for North America and the rest of the world. It was the same with computers from many other manufacturers. None used the same architecture – there seemed to be no reason to do so at the time; these were single-purpose computers.

Computing was *specialized*. The idea of sharing components between computer lines seemed impractical or impossible, given the different missions of the responsible organizations. These were semi-custom computers, designed, configured, built, and tuned to the customers' needs. Any hope for portability or extensibility was at the FORTRAN or COBOL compiler level and, because of the differing word sizes (number of bits in a byte and bytes in a word) being used, programs often were not portable without

[1] Earlier computers, such as the IBM 701, were also called mainframes, but most industry watchers define the beginning of the mainframe era to coincide with the introduction of the IBM System/360 in 1964.

IN THIS ISSUE

recoding.[2]

While there was some off-purpose use of these computers, i.e., for other than the intended original application (even then, no cycle was to go unused), these programs were run serially (one at a time). The idea of, much less the need for, a generalized multipurpose computer didn't exist.

Because computers were specialized, so were many of the peripherals and interfaces. What we take for granted today – such as USB and PCI connectivity – just didn't exist. If you bought a new computer, you had to buy new consoles (mostly hard copy, but later CRTs), tape drives, printers, and even card readers. While there was some standardization in punch card formats and tape media, the coding schemes were often unique.

Worse yet, were the control programs (operating systems). Each was different, and closely tied to the underlying hardware architecture.[3] There was no reason for them to be the same across a range of computers and different geographies. Operators were specially trained for the computer that they would keep running. In many ways, they were the heroes of the day, using their intelligence to overcome all of the hardware and software shortcomings and programming mistakes. There were no standard rules of the road, so to speak. Best practices were developed on the job for each computer.

In retrospect, **prior to System/360 was the *Primitive Age of Computing***. Computer development was a "tribal affair", and there was little interaction between tribal communities. To summarize, as President John F. Kennedy was sworn into office in 1961, computers were:

1. Special purpose, for either scientific or commercial computing,
2. Designed for a 4-5 year product sales life, maybe including a mid-life kicker,
3. With no concept of upgrading to a larger system within the "family",
4. With one-of-a-kind architectures and technologies,
5. With parts that were not interchangeable,
6. With proprietary peripherals and interfaces (even from the same vendor),
7. With unique operating systems,
8. Capable of running one program at a time, and
9. Requiring a lot of tending by specially-trained operators and vendor support personnel, who had to be quite knowledgeable,
10. Because there were no standards for computing or principles of operations, and
11. Any minor failure would bring the system to a halt.

This is the backdrop for the beginning of a new era of computing that began later that year at IBM and resulted in the System/360 in 1964. Read on for that story.

Part 2 – Conceiving the Need

Hindsight is wonderful. You get to look at what happened, what evolved, and, equally important, what didn't, and you try to analyze what wisdom you would have brought to that prior time. It's easy to jump to the "right" conclusion. So, I decided to take a more serious look at IBM's decision to build what became the System/360, capable of running a full spectrum of applications (and representing the 360 degrees of a compass). In late December 1961, a very confidential report from a 13-person IBM task force was finalized. Officially known as the *Final Report of the SPREAD[4] Task Group*[5], this report laid out the requirements for a new line of computer systems that would address all of the above limitations, and much more. For me, recently reading this report must have been like religious scholars reading the Dead Sea Scrolls for the first time. It allowed me to travel through time, like to the beginning of the universe, at least as I have professionally known it.

[2] In what was finally recognized as a source of the Y2K problem, many early programmers, especially those who had been assembly language programmers, crammed as much data into as few bytes as possible, yielding a hodge-podge of physically-defined data. If programmers stuck to the more universal data definitions within a FORTRAN or COBOL program, portability was increased dramatically. But these were times when efficiency predominated (think about the severe limits and very high costs of memory and external storage).

[3] For example, the IBM 704 FORTRAN Monitor System (FMS) could only compile and run FORTRAN programs. Later, on the IBM7094, IBSYS could run FORTRAN, COBOL, and assembler programs.

[4] Officially, SPREAD refers to the name of the group responsible: Systems Programming Research and Development.

[5] The report was reproduced in the **Annals of the History of Computing**, Vol. 5, No. 1, January 1983, by AFIPS Press, and is now available from the IEEE Computer Society. I had the advantage of getting a copy of the original report from the IBM archives, with its typewritten look and feel and its hand-drawn graphs on graph paper.

So much that we all have taken for granted is now clear. **In retrospect, System/360 was the beginning of the modern age of computing.** Previously, I always looked at System/360 as if it was the constant, something that I knew first hand, but today I see it as the beginning of "now" – anything earlier was "then" and unenlightened.

The landmark report was written in just 60 days, but boldly laid out principles for overcoming the limitations listed above, including the pros and cons of their technical decisions, the timetable needed to bring the resulting product to market in less than three years, the marketing challenges and customers' impacts that needed to be addressed, and the internal organizational (and global) implications and recommendations. I am sure that the team knew that what they were creating would be important to IBM. However, I suspect that only hindsight allows us to realize the historical significance to the computing industry.

Part 3 – Betting the Company

The next month, in January 1962, the report was distributed and the debate began, pitting this proposal against many other proposals for new processors and further development. IBM's senior executives knew that they were at a crossroads. They knew that they could not let all of its divisions independently develop continuing generations of new computers. They also knew that this was a mammoth undertaking, far beyond anything that had been done previously. They knew that the investment would be expensive, which, at about six billion dollars, was more than twice its annual revenue and about 24 times IBM's annual profit. Moreover, they knew that if this project failed, it would forever cripple the company, because improvements to existing products and investments in competing new processor ideas would have to be cut to fund its development. They knew that this processor family would have to be the only family for the rest of the decade. **IBM's executives decided to bet the future of the company to implement the SPREAD recommendation**s.

Part 4 – Delivering the Family

IBM's existing products resisted extinction for most of the two-plus years that System/360 was in development. IBM needed to keep the revenue flowing while the development was ongoing, and to stage carefully its announcement to minimize the impact to revenue during the transition. On April 7, 1964, the System/360 was announced, with much fanfare, and to the total astonishment of IBM's competitors, who were struggling to keep up with IBM's prior offerings. The first Model 40 was shipped in April of the following year.

The beauty of this family was not just in its ability to solve all of the shortcomings listed above, although the 50-times range of the family (from smallest processor to largest) was considered to be unimaginable by many, but in

Confessions of a Mainframe Bigot

Let's begin with a confession. I am an over-the-hill baby boomer, whose first contact with a computer was in 1966 when I was a high school senior. I needed a computer to do my meteorology science fair project. At that point in time, nobody had a computer. Few even knew what a computer was, and even fewer could ever imagine the need to use one, or that they ever would. *Oh, what we take for granted today.* But I knew that I could not forecast the weather without one. Only a few institutions had computers, and I *had* to get access to one. I didn't care which one, as long as it could "compute". So, I enrolled myself as an evening student at what was then Miami-Dade Junior College, and took a course in FORTRAN programming. The computer was an IBM System/360 mainframe, a Model 40, I believe. That fall I learned how to write FORTRAN, and snuck my science project into the card hopper whenever there was some idle time. Thus began my lifelong career in the computer industry.

I have never worked for IBM as an employee, but was involved with the mainframe at some of its pivotal points. I have been involved with many computers and architectures, from minicomputers to grid. I consider myself open-minded, but do put on my *mainframe bigot* hat whenever the opportunity arises, because the mainframe has been the proving ground for much of the innovation in the computing industry, especially for commercial systems, and it has been the standard for comparison for competing platforms. And, of course, for much of the mainframe's history – *I was there.*

Not the Whole Story — Why Mainframe Software is Really the Crown Jewel

It's too easy just to think of a mainframe as a box, and the current zSeries servers as the great-great-grandchild of the original mainframe box, the System/360. It is very hard for most "real folks" who exist outside of the data center, maybe somewhere in the application or end-users' world, to see the intelligence that resides in the box - in terms of the operating systems, middleware, scheduling and management software, and other utilities - that bring the hardware to life. It's easier to personify the box – and love it or hate it – as in *I really hate my PC*. It's probably not the PC that you hate, but rather the operating system (such as a version of *Windows*), or the middleware (such as a browser), or an application (such as a specialized search engine), or a utility (such as anti-virus protection), or a combination of these, that brings out your ire. When it comes to a server, especially a server capable of running many workloads simultaneously, the less you see of it, the better it is. And the more automated it is, in terms of self-management and optimization, the better it is.

So it is not surprising that most people think that they know what a mainframe is, but are unable to name a mainframe operating system or hypervisor, a controller for virtual machine images. That's too bad, because these behind-the-curtain software stacks (collections) really put the capabilities onto the hardware, and not vice versa. You might ask, then, *so why can't a mainframe operating system, with all of its goodness, run on a PC?* In fact, it can, and many of the other traditional mainframe vendors basically have chosen to emulate the original instruction set of their mainframe on a commodity-chip-based server. IBM, too has done this too in its past, but the performance was just too constrained for the on-demand workloads of the 21st century. It was easier, and more efficient, to put Linux or Java onto the mainframe hardware than to put the mainframe operating environment on commodity hardware. The fact that the oh-so-modern Java and Linux are maximized on a traditional IBM mainframe (i.e., many sessions running under VM as a hypervisor) points to the inherent value of this relatively unknown class of mainframe control software.

There have been a number of series of operating systems for IBM's mainframes – from the mainline original *OS1* and *OS2* to OS/*MFT* to OS/*MVT* to *TSS* to *MVS* to *OS/390* to *z/OS*. The lineage carries the rights to the crown as the jewels of IBM's Mainframe Empire. Almost of all of the black magic that other platforms' operating systems hope to deliver under the banner of "mainframe-like" descend from this lineage. But there are other family lines in IBM's royal family that have given us virtual machines and hypervisor control (*VM* and *z/VM*), scaled down functionality with greater operational simplicity (the original *DOS* to *VSE* to *z/VSE*), to large-scale transaction processing (*TPF* - primarily for airline reservation systems), and to several others developed outside of IBM (including *MTS* - *Michigan Terminal System* - from my college and early professional days).

When you add IBM' mainframe-originated database products (IMS and DB2) and transaction processing system (*CICS*), all still in heavy, mission-critical use today by the largest enterprises, to the rich history of mainframe operating systems, then you really can begin to understand why the last forty years are not just about the mainframe hardware. **It's the whole mainframe offering, now including open-systems middleware, like *WebSphere* and *MQseries*, plus a strong heritage of customer support and service, that has moved the mainframe to its royal status.**

the completeness of the solution delivered at the same time by IBM, from multiple development and manufacturing sites around the world. System/360 had standard peripheral interfaces, and the offering included a range of attachable tapes, disks, printers, communication devices, terminals, and more, to meet the variety of customers' needs. More importantly, it was a general-purpose machine (capable of both scientific and commercial computing), and capable of running several applications at the same time, with security between applications. It was delivered with a manual called *Principles of Operation*, which spelled out the architecture, capabilities, new characteristics, and guidelines for using this multipurpose asset. This became the

Bible for a whole generation of systems professionals[6].

What was done originally as the investment of the decade (the product definition indicated that it needed to last until at least until 1970) has turned into one of the two biggest technology investments of the twentieth century, the other being the investment to put a man on the moon, also done at the same time, and likely could not have been done without the System/360. The concepts have endured.[7] Its architecture has endured. Its principles of operation have endured. Of course, all have been refined, enhanced, and extended in the last four decades. Nevertheless, **the zSeries of today is clearly identifiable as the much improved, but genetically-identifiable descendent of the System/360.** This may be the second longest (so far) run of a common technology, next to the Volkswagen *Beetle*, and the IBM mainframe will surpass that before long. Read on to find out why this has been so enduring.

Part 5 – Pushing the Puck

System/360 was a big success because IBM saw what needed to be done, and set out to achieve that difficult goal. They had the wisdom to see where enterprises wanted to go with their computing and, for many, would really have to go to achieve enterprise objectives that were increasingly dependent on information processing. There's an old saying that a good hockey player skating up the ice with the puck doesn't look at where the puck is, but where he wants it to go, and then skates the puck to that point.

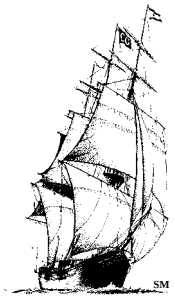
Over four decades, IBM has skated the System/360, and its successors, to where the requirements were going to be, enriching it many times over along the way, and serving the growing and changing needs of its customers. The thought-to-be-contemporary concepts of complex workload management, cluster management, virtual memory, storage, and machines, logical partitions, remote synchronization and recovery, and dozens too many to list here, may all seem like recent incarnations in the open systems arena, but these are decades old concepts pioneered on the mainframe. For the last decade, we have seen claims that new age systems are "mainframe like" or deliver "mainframe values". **There is no higher compliment to the IBM mainframe family**.

While investment in its mainframe, now the zSeries, may have waned (as a percentage of R&D investment) as IBM has gone on to offer a number of platforms (*xSeries*, *pSeries*, and *iSeries*), it still leads as IBM's premiere platform, with the most advanced capabilities first delivered on zSeries and then shared with the other eServer family members. In doing so, zSeries continues to redefine itself, most recently as an extensible Linux environment and as a Java application execution environment. Ten years ago, who could have anticipated these kinds of redefinition? **IBM's mainframe was boldly conceived forty-three years ago and continues to be boldly reconceived today.**

If you look between the lines of IBM's *On Demand* vision, you see the need for a utility-like server solution that offers security, flexibility, open standards, workload management to policy specifications, and resource optimization. If that sounds like a mainframe, don't be surprised. As the mainframe becomes increasingly freed of the burden of being labeled as proprietary and as customers begin to focus on delivering resources to applications and users according to policy-driven specifications, enterprises will realize that what they want is a mainframe, and not something that strives to be one. **The mainframe has the traction to continue to lead the way to IBM's new bold initiative – on demand – that, in many ways, is similar to that risky decision of 40 years ago.**

Part 6 – Place Your Bet

The mainframe is not dead and is not dying. Don't be so foolish as to ignore its capabilities and possibilities without close inspection; betting against the mainframe goes against forty years of success.



[6] Including me as a student, then at The University of Michigan. In fact, three decades ago I put my System/360 *Green Card*, a handy reference guide, in the nightstand by my bed, so it would be there when the late-night call might come. For sentimental reasons, it still is l there today.

[7] Including the need for a complex operating system that eliminated many of the needs for attending staff. Many may criticize IBM's mainframe operating systems for being complex but, in reality, they hid much of the complexity, through scripting and automation, while attending to complex requirements.

About The Clipper Group, Inc.

The Clipper Group, Inc., is an independent consulting firm specializing in acquisition decisions and strategic advice regarding complex, enterprise-class information technologies. Our team of industry professionals averages more than 25 years of real-world experience. A team of staff consultants augments our capabilities, with significant experience across a broad spectrum of applications and environments.

- ***The Clipper Group can be reached at 781-235-0085 and found on the web at www.clipper.com.***

About the Author

Mike Kahn **is Managing Director and a cofounder of The Clipper Group.** Mr. Kahn is a veteran of the computer industry, having spent more than three decades working on information technology, spending the last third at Clipper. For the vendor community, Mr. Kahn specializes on strategic marketing issues, especially for new and costly technologies and services, competitive analysis, and sales support. For the end-user community, he focuses on mission-critical information management decisions. Prior positions held by Mr. Kahn include: at International Data Corporation - Director of the Competitive Resource Center, Director of Consulting for the Software Research Group, and Director of the Systems Integration Program; President of Power Factor Corporation, a Boston-based electronics firm; at Honeywell Bull - Director of International Marketing and Support; at Honeywell Information Systems - Director of Marketing and Director of Strategy, Technology and Research; with Arthur D. Little, Inc. - a consultant specializing in database management systems and information resource management; and, for Intel Corporation, Mr. Kahn served in a variety of field and home office marketing management positions. Earlier, he founded and managed PRISM Associates of Ann Arbor, Michigan, a systems consulting firm specializing in data management products and applications. Mr. Kahn also managed a relational DBMS development group at The University of Michigan where he earned B.S.E. and M.S.E. degrees in industrial engineering.

- ***Reach Mike Kahn via e-mail at MikeKahn@clipper.com or via phone at (781) 235-0085 Ext. 21. (Please dial "1-21" when you hear the automated attendant.)***

Regarding Trademarks and Service Marks

The Clipper Group Navigator, **The Clipper Group Explorer**, **The Clipper Group Observer**, **The Clipper Group** *Captain's Log*, and *"clipper.com"* are trademarks of The Clipper Group, Inc., and the clipper ship drawings, *"Navigating Information Technology Horizons"*, and *"teraproductivity"* are service marks of The Clipper Group, Inc. The Clipper Group, Inc., reserves all rights regarding its trademarks and service marks. All other trademarks, etc., belong to their respective owners.

Disclosure

Officers and/or employees of The Clipper Group may own as individuals, directly or indirectly, shares in one or more companies discussed in this bulletin. Company policy prohibits any officer or employee from holding more than one percent of the outstanding shares of any company covered by The Clipper Group. The Clipper Group, Inc., has no such equity holdings.

Regarding the Information in this Issue

The Clipper Group believes the information included in this report to be accurate. Data has been received from a variety of sources, which we believe to be reliable, including manufacturers, distributors, or users of the products discussed herein. The Clipper Group, Inc., cannot be held responsible for any consequential damages resulting from the application of information or opinions contained in this report.