The History and Growth of IBM's DB2

Donald J. Haderle
Haderle Consulting

Cynthia M. Saracco
IBM

IBM's Database 2 (DB2) relational database management system (RDBMS) shipped in the early 1980s and drove billions of dollars of revenue to IBM and other firms within its first decade. The product spawned a wealth of add-on tools and shaped the future of mainframe computing. Today, DB2 spans multiple operating systems and is widely deployed across a broad spectrum of industries.

Relational database management system (RDBMS) technology now represents a strong IBM business, but it was hardly a sure bet in the 1970s and early 1980s.[1] Indeed, the widespread deployment that DB2 enjoys today masks the many challenges of its formative years. Chief among these was building a viable product based on unproven technology while demonstrating tangible value to IBM's hardware-oriented business.

In this article, we explore the early years of mainframe DB2. The first two sections discuss database computing in the 1970s and the state of IBM and its customer base at that time, setting the context that led to IBM's subsequent decision to form a distinct DB2 development group. We then examine how DB2 evolved into a commercial offering that ultimately influenced enterprise computing around the globe. In doing so, we explore three stages in the evolution of mainframe DB2 during its first 15 years, when Don Haderle served as DB2's chief software architect and later the chief technology officer of IBM Information Management. In the final section, we briefly discuss DB2 beyond that time and the development of DB2 for open systems, although due to space limitations, we won't do those topics complete justice.

Database Management in the 1970s

Early DBMSs, such as IBM's Information Management System (IMS) and Cullinet's Integrated Database Management System (IDMS), supported bill of materials (BOM), material resource planning (MRP), and other applications critical to business processes in manufacturing, finance, retail, and other industries. These products featured hierarchical or network data models and provided both database and transaction management services. However, database schema changes required rewriting application programs, and programmers had to understand the complex principles of concurrency and consistency—advanced thoughts at the time. As a result, application upgrades were often complicated and time consuming. Furthermore, database management standards didn't exist in the 1970s, and this limited adoption of DBMS offerings. In 1969, E.F. Codd introduced the relational data model,[2] and various research prototypes (such as the University of California, Berkeley's Ingres,[3] IBM's System R,[4] and IBM's Peterlee Relational Test Vehicle[5]) explored support of this model. In particular, System R[6] introduced the Structured Query Language (SQL)[7] in 1976 and proved the potential of RDBMS technology for on-line transaction processing. Pratt & Whitney demonstrated its usage in 1977.

Midrange computers from Digital Equipment Corp. (DEC) and IBM supported small and medium businesses in the 1970s as well as departmental functions within large businesses. However, mainframes provided the power that large businesses needed for core operations. Midrange and mainframe platforms used different hardware technologies, yielding distinct cost attributes and processing capabilities.

 Published by the IEEE Computer Society 1058-6180/13/$31.00

IDMS and IMS DL/1 were only available on mainframes. In 1979, Oracle introduced an RDBMS on DEC based on information IBM published about System R. Similarly, IBM delivered an RDBMS for System/38 in 1980.[8] Both addressed the midrange market for query processing (decision support), not on-line transaction processing.

Batch processing was the most popular execution model on mainframes in the 1970s. Firms stored data in files, and concurrent read/write access from multiple applications wasn't supported. DBMSs evolved partly to address this, but programming was complex. In 1980, Frost & Sullivan reported that fewer than 25 percent of enterprises used a DBMS.[9]

The DB2 Decision for Mainframes

With that industry backdrop, let's turn to what was happening in IBM during this time. IBM derived most of its revenue and profit from mainframe hardware, including peripherals. DBMS customers used more storage and processing capacity than others, so IBM sought to drive greater DBMS adoption. However, IBM depended on independent software vendors to support the latest IBM hardware. These vendors often delayed doing so until the new hardware enjoyed a strong installation base. This slowed hardware sales. As a result, IBM's storage division funded the Eagle project in 1976 to develop advanced DBMS and transaction systems.

By then, IBM's IMS and Customer Information Control System (CICS) had a proven track record of contributing to IBM's mainframe sales and exploiting IBM hardware. However, both were first released on S/360 in 1969, and they needed enhancements to keep pace with evolving customer concerns.

The Eagle team focused on three fronts:

- merging the different transaction models offered by IMS and CICS,
- overhauling low-level infrastructure services for logging, memory management, transaction management, and the like, and
- building a higher-level software layer that could accommodate multiple data models, including flat file, relational, hierarchical, and network models.[10]

Unfortunately, technical leaders estimated that it would take a decade or so to upgrade IMS and CICS with Eagle technology; support for new data models would come even later. IBM business leaders found that timeline unacceptable.

After seeing System R, customers seized on the relative ease of creating new databases and applications, particularly for data analysis and report generation.

At the time, IBM had three distinct operating systems for the mainframe, each targeting different customer needs. VM (Virtual Machine) served decision support and experimentation in enterprises, DOS/VSE (Disk Operating System/Virtual Storage Extended) served small businesses, and MVS (Multiple Virtual Storage) served medium and large businesses needs. Bob Jolls, who managed the Advanced Database Department for the VM and DOS/VSE operating systems, had to find a way to deliver Eagle technology to customers earlier. Firms working with the System R prototype on VM lobbied for a fully supported product. In 1979, Jolls concluded that IBM could deliver the System R prototype on VM and VSE within two years, and this led to IBM's release of SQL/DS in 1981. Later that year, Jolls also decided to similarly provide System R's capability on MVS, after which the MVS offering would add DL/1 on top of this same storage infrastructure, enabling programmers to access the same database via SQL or DL/1. Eagle's objectives of merging the IMS and CICS transaction models and enhancing their infrastructures were left to the IMS and CICS development teams to resolve.

The stage was set for DB2. IBM established the DB2 development team in San Jose, California, where local expertise in MVS access methods, programming languages, and database management technology (for example, IMS) was abundant and the Eagle team was located. A number of Eagle members joined the DB2 team, although some Eagle staff departed for competing firms before DB2's first release. During this time, many IBM researchers, such as Codd, continued to

Figure 1. Marilyn Bohl, Don Haderle, and Bob Jackson led DB2 Version 1 development in the early 1980s.

work separately on advanced projects. Their efforts are the subject of another article in this issue.

Stage 1: Development and Delivery of DB2 Version 1

Development of the first version of DB2 began in earnest in 1980 and continued through its limited availability in 1983 until it became generally available in 1985. During this time, Haderle and his cohorts met with many customers who clamored for solutions that would help them rapidly satisfy business needs.[11] Customers lamented the complexity of managing their existing data centers, the lack of skills, and the increasing backlog of business requirements. After seeing System R, they seized on the relative ease of creating new databases and applications, particularly for data analysis and report generation.

Staffing and Priorities

The DB2 team needed to provide strong integration with the hardware, operating system, application programming languages, and transaction systems of MVS to dramatically reduce the time to develop and deploy new databases and applications and to achieve respectable cost/performance. Leading this effort were Marilyn Bohl, the development manager; Don Haderle, the overall technical leader; and Bob Jackson, the technical lead for DB2's integration with MVS. Bohl had served as a team lead in CICS, a manager in IMS, and a second-level manager of IBM's sorts and assembler products. Haderle had a decade of experience developing applications, operating systems, file systems, compilers, and security systems. Jackson had spearheaded MVS I/O subsystem development in the early 1970s. Figure 1 depicts these three DB2 leaders in the early 1980s.

A host of experts in CICS, IMS, Cobol, Fortran, and other technologies joined DB2, which consisted of several hundred people. John Nauman managed query processing development, Josephine Cheng led query optimization, Dick Crus led the database manager,[12] Bob Gumaer and Jim Teng created the buffer manager,[13] Jay Yothers led application binding, Akira Shibamiya led performance, Roger Reinsch led the subsystem team, and Earl Jenner with Cliff Mellow developed the log-based recovery mechanisms. Roger Miller joined DB2 from the Fireman's Fund Insurance, bringing a critical customer perspective. (For three decades, he served as the DB2 customer ombudsman.) IBM Vice President J. (Jack) D. Kuehler served as DB2's executive sponsor.

Like the cobbler's children, the DB2 team was starved for compute time during the early days. Network wiring limitations at the site restricted the number of available terminals, and mainframes were expensive. The team couldn't afford a dedicated MVS machine. Yothers built an MVS emulator for VM so the team could test its software during the day. Native MVS testing occurred after hours or on weekends when others weren't using the system. DB2 developers shared 30 terminals, forcing the team to operate on all shifts.

System R's Relational Data System (RDS) served as the basis for DB2's query processing engine. However, DB2 developers rewrote the code into PL/S[14] and hosted it atop the Eagle storage engine, which handled storage, buffering, serialization, consistency, and durability on the foundation of an extended MVS operating system.[15]

Performance Challenges

Haderle and Shibamaya created a performance model for the initial DB2 design and projected that DB2's processor consumption would be three to four times that of DL/1, 10 times that of IMS FastPath, and more than 10 times that of native file managers. These differentials were significant. Processor and storage costs comprised the majority of mainframe expenses at the time, and customers monitored such costs closely.

With considerable effort, the DB2 team reasoned that it might be able to close the gap by a factor of two to three. Continual hardware improvements over time would diminish the importance of processor cost relative to the overall cost of computing.

Unfortunately, the forecasted cost performance gap proved true.[16] The first version of DB2 was more expensive to use and less robust than IMS DL/1 or CICS File Control for mission-critical applications. Customers begrudgingly accepted this added expense only for occasionally used applications or new applications that needed to be deployed quickly.

Key Strengths

Fortunately, in 1983 an Austrian customer validated that developing and deploying solutions with DB2 was more than 10 times faster than existing technology. This was DB2's key strength at the outset. Three factors enabled DB2 to dramatically improve application development, database administration, and system administration:

First, IMS and CICS required firms to shut down their transaction systems to define or alter a data model as well as to create or alter an application's database requests. In most cases, such work was done on weekends when the operational system was not used. With DB2, this work could be done online without impacting existing operations. This reduced development and test efforts by several factors.

Second, SQL syntax was consistent across programming languages and interactive use. Programming language preprocessors extracted the SQL statement and replaced it with a `CALL` to DB2 with proper binding to language structures, allowing programmers to perceive SQL as an extension of Cobol, Fortran, and PL/1. Programmers could test SQL interactively before embedding it in their application.

Third, a DB2 database could be accessed from any application environment within the MVS ecosystem, allowing data to be shared across applications in CICS, IMS, TSO, and batch. This was novel (other DBMSs had more limited connectivity) and made DB2 databases as available to applications as the file system.

Early Applications and Usage Patterns

The killer application for DB2 was QMF (Query Management Facility),[17] IBM's query and report writer. In the early to mid-1980s, business users rarely accessed data directly—the tools were too difficult for anyone but technical staff (IT professionals) to use. Instead, users submitted requirements to IT, and programmers manually coded the logic necessary to obtain the desired data.

Such logic was typically written in RPG, Cobol, or a similar programming language, although fourth-generation programming language (4GL) tools (such as Ramis, Nomad, and Focus) were sometimes used. Generating new reports was a large part of the application demand facing IT leaders, and QMF responded to this.

QMF allowed IT professionals to construct queries and reports interactively. And it provided a scripting language for building reports from sets of queries. IT created and saved such scripts, making them available to business users for interactive or scheduled execution. IT typically scheduled scripts to run periodically (perhaps every night or at the end of every quarter) so that timely reports would be available for the business. This delighted business users, who began to demand more. Early DB2 systems had thousands of QMF scripts.[18]

During this period, DB2 applications and databases were simple. IT staff often ported existing batch applications, carrying over data structures with little change. This allowed application function to be brought online and data to be available for querying. Unfortunately, applications generally retained existing record-level access based on primary or secondary key values. Joining or filtering data was done within applications rather than by DB2. This led to horrible performance/because the processor cost for retrieving lots of unnecessary columns was enormous. Retraining programmers to use database joins and retrieve only the columns they needed was a significant effort. After all, applications previously retrieved the record with all the fields in a memory data structure and passed that structure to all of its subprograms, so programmers had no idea exactly which fields the subprograms needed.

Data was generally loaded in bulk through high-speed non-SQL programs. Batch programs took daily or weekly feeds from existing systems and updated DB2 so that the data could be queried and reports generated. At this time, the volume of data entering databases via online transactional applications was relatively small (just millions or billions of bytes, not the trillions used today).

Driving DB2 Adoption

A 1982 paper by John Zachman[19] helped set the stage for DB2 adoption, albeit indirectly (and perhaps unintentionally). At that time, basic business operations that needed to be treated as a logical unit (such as order processing, billing, and fulfillment) were supported by discrete applications that required their own copies of the same data, often in different formats. Forward-thinking IT organizations worried about the growing complexity of managing the data flows required to support such disparate applications, and this paper presented a compelling vision that DBMSs might help them cope with that complexity. Consequently, when IBM released DB2, some IT architects viewed DB2 as a way to improve data and application integration over the long term.

From the outset, IBM understood that standardizing SQL was critical. Firms wanted alternatives provided by standardization; furthermore, national governments, including the US, Germany, and England, demanded it in their requisitions. In 1982, the American National Standards Institute (ANSI) formed an X3H2 subgroup to standardize SQL,[20] and the first version was adopted in 1986. ISO adopted the initial SQL standard in 1987. This, coupled with embrace of the standard by IBM, Oracle, Teradata, Sybase, and others, led to its widespread use.

Limited and General Availability

DB2's limited availability in 1983 brought the software to 40 to 60 customers. Senior IBM executives would permit DB2 to be generally available only when the early customers were delighted, but customers found glaring errors, especially with recovery and serviceability. For example, DB2 recorded data for recovery on dual logs. Although IBM emphasized the importance of those logs to customers, one early customer overwrote the logs before the data was synchronized, creating a recovery nightmare. Consequently, the DB2 team quickly developed facilities to recover from a myriad of situations.

During DB2's limited availability, IBM System/370 processors and MVS upgraded to 31-bit addressable memory,[21] prompting significant DB2 enhancements and lengthy testing. As it turned out, DB2 desperately needed the additional memory, so the delay was a godsend. In 1985, IBM finally made DB2 generally available to MVS customers with a low-key product announcement,[22] positioning it as a decision-support offering. (IBM business leaders were concerned that DB2 would stretch thin the staff in the field who were supporting IMS.) IMS DL/1 remained the high-performance transactional offering and strove to keep its revenue stream intact.

A number of IBM customers quickly realized DB2's value. DB2 was not cost competitive for significant transaction processing vis-à-vis IMS DL/1, IDMS, Adabas, or Datacom/DB. But it leveraged firms' existing investments in IMS, CICS, and MVS and helped IT respond to new business imperatives. DB2's support for transactions, batch, and interactive models was critical.

Meanwhile, Teradata debuted in 1984 with a highly parallel architecture running on microprocessors capable of handling significantly larger data volumes than DB2. Companies in banking and retail loaded detail records into Teradata for analysis and used DB2 to work with the summarized data. At the outset, DB2 provided inter-query parallelism; intra-query parallelism (using multiple processors to satisfy a single query) would not be supported until 1993. There were more pressing problems.

Stage 2: "Ready for Prime Time"

From 1985 through 1988, DB2 development focused on making the product ready to handle mission-critical workloads. This required dramatic reductions in processor consumption to improve cost and throughput. Also mandatory were concurrency, serviceability, and manageability improvements as well as support for referential integrity.

From 1986 to 1987, IBM released two DB2 upgrades and a performance modeling tool that sales personnel could use to forecast hardware requirements.[23] However, the big news was version 2. Delivered in 1988, it launched a new era for the product. As the headline of one trade journal noted, DB2 was finally "ready for prime time." Getting there took some doing.

In 1980, IBM envisioned SQL and DL/1 sharing data and slated this capability for the version 1 release 2 of DB2. But the DB2 team quickly realized that its DL/1 performance would be unattractive to IMS customers. Furthermore, migration issues would be significant because not every DL/1 function call would behave compatibly. Resolving these problems would have taken years and prevented the DB2 team from responding to other customer demands. Fortunately, early customers such as Boeing Computer Services (BCS)[24] and Caterpillar helped IBM understand their needs and resolve the different roles that IMS and DB2 would play. As such, DB2 abandoned its strategy to support multiple data models and focused exclusively on relational technology.

Delivered in 1988, DB2 version 2 launched a new era for the product.

During version 2 development, staffing changes continued to occur, with some DB2 developers and IBM researchers departing for competing firms or establishing independent consultancies.[25] As a result, Josephine Cheng became the DB2 query processing manager, and Yun Wang became the query processing technical lead.

CPU Cost Correction

Reducing DB2's processor consumption meant inspecting the most performance-sensitive code paths and tightening the code, which required changes in the operating system, the transaction managers,[26] and DB2. Highly sensitive code paths were rewritten in Assembler Language to control the instruction sequence. The hardware team analyzed instruction streams and improved their pipeline process for DB2. DB2 needed to marshal columns and search arguments and the application itself, which was expensive. Saving artifacts allowed reuse for subsequent executions.

With all this work, DB2's processor consumption was brought within 25 percent of DL/1's by 1988. The gap between DB2 and other non-IBM DBMSs was even narrower. IMS Fastpath provided a more constrained yet useful data model that delivered orders of magnitude better performance for target applications, and DL/1's hierarchical model was still better suited for bill of materials and other applications.

With DB2 version 2, a 3090-600S mainframe with 256 Mbytes of main memory could process 270 transactions per second at 87.7 percent processor utilization, providing a 120 percent improvement in throughput over the prior release.[27] Together with other features (such as referential integrity and increased reliability), this made customers feel safe deploying DB2 for online transaction processing. From this point on, DB2 was used heavily for transaction processing.

Serviceability, Reliability, and Concurrency

Staff at BCS, particularly Walt Crush, helped the DB2 team understand the imperative for greater levels of availability and reliability. Business units funded BCS and established service-level agreements that specified financial penalties to BCS if they failed to meet availability targets. BCS database administrators were responsible for such service. They needed to be able to identify any problem violating the contract's terms and take appropriate action, such as terminating the offending application(s), shifting workloads, and so on. The mandate was to keep the system available for the masses, sacrificing a few. Service-level agreements were common among MVS customers and heavily influenced DB2's development efforts.

Servicing SQL problems proved problematic. System R compiled queries, producing an executable program with embedded calls to low-level database services. The compiler managed machine registers, limiting the addressability of the compiled code and data areas. If the limits were exceeded, the compiler would declare the statement to be "too complex" and the query would fail. With the complexity of SQL increasing, this had become a common event. Servicing bugs in the compiled code also proved troublesome, so the DB2 team redesigned the SQL runtime to generate interpretive instructions Although some had thought that the compiled query would run more efficiently, this wasn't the case.

Concurrency was also a major issue. DB2 locked data and index pages, which created hot spots and problems for small tables. An elegant solution would require years of effort, and the team needed stopgaps. In 1986, Irv Traiger, manager of the database department at IBM's Almaden Research Center, proposed that R&D pool resources to jointly work on key issues. Thus, IBM's Data Base Technology Institute (DBTI) was formed. DBTI stepped up to the challenge, providing advanced technology that drove DB2. Patricia Selinger served as its manager and Josephine Cheng led development. Concurrency challenges emerged as an early area of focus. C. Mohan spearheaded efforts in this area with Aries,[28] developing algorithms for fine grain concurrency and recovery. In 1988, DB2 version 2 subset data and index pages (i.e., organized a 4k page into smaller units) for locking and recovery as a stopgap measure. It took until 1995 to implement the full technology, aggravated by changes in the IBM System/390 architecture, which we'll discuss shortly.

Benchmarks also influenced DB2 development efforts during this period and beyond. The team created internal benchmarks from customer experiences and used competitive benchmarks (such as the Model 204 benchmark[29]) to investigate performance issues. Gordon Steindel and Craig Madison from the Great West Life (GWL) Insurance Company in Winnipeg, Ontario,[30] created a benchmark that proved important for DB2 throughout the 1980s. This benchmark employed a wonderful set of queries that mirrored what IBM saw from its leading customers. GWL's tests showed that DB2 excelled in transaction processing and Teradata excelled in query processing, so the DB2 team created many algorithms and techniques to improve its query processing, including skip sequential processing,[31] hybrid join,[32] and function pushdown. Haderle traveled frequently to the frozen tundra of Winnipeg in the late 1980s, where the GWL team graciously hosted him, benchmarked and critiqued DB2 technology upgrades, and introduced him to curling and other winter diversions.

Partnerships, Press, and Product Positioning

Administering any DBMS in the 1980s required considerable skill. An application's success depended on database administrators, who amassed considerable clout and influenced which DBMS would be used for an application. Administrators needed tools to manage DB2 and tune performance. IBM's Norris van den Berg convinced software vendors such as BMC, Platinum, and Candle to develop tools for the nascent DB2 product. IBM Share and Guide user groups created working sessions on DB2. In addition, independent DB2 user groups sprung up around the world, attracting administrators who swapped tips. In 1988, these groups formally united as the International DB2 User Group, a nonprofit organization that now hosts numerous conferences, workshops, and other activities.

IBM sponsored various efforts to educate its sales force and customer base and promote DB2 technology.[33] At this time, IBM lacked a direct software sales force, and regional software specialists[34] supported hardware sales teams.

Trade journals kept DBMS news highly visible.[35] Consulting agencies, such as Codd & Date and Database Associates, drove attention to RDBMS technology in general and occasionally DB2 in particular.

Application software vendors were important in 1988, but not nearly as important as they would be in another decade. Most large enterprises created their own applications. As such, the technology was focused on their needs. With DB2 making strides in mainframe shops, getting application vendors (such as SAP and Hogan) to port to DB2 was a bit easier.

In 1988, if computing cost or transaction throughput was the customer imperative, IBM promoted IMS DL/1 or Fastpath.[36] If response to business initiatives was the customer imperative, IBM promoted DB2.

DB2's competition for transactional work came primarily from pre-RDBMSs (such as IDMS, Datacom/DB, and Adabas), which struggled during the late 1980s. For example, Datacom/DB was a Codasyl DBMS with an added SQL layer, which Orrin Stevens later acknowledged to be a troublesome implementation.[37] IDMS/R was also a Codasyl DBMS with an added SQL layer; its relational support was not very competitive and the product lacked the application productivity improvements that DB2 offered. Adabas was an inverted list DBMS with a SQL layer; it had a small presence in the United States with a larger one in Germany. Cincom offered the SUPRA RDBMS, which lacked SQL support until 1989. Oracle never attracted a large following on MVS.

In 1985 IBM announced the 3090 series with two-, four-, and six-way symmetric multiprocessors (SMPs), a significant departure from prior one-, two- and four-way processors. DB2's threading model scaled well as the number of processors increased, and all shared services (such as memory management and locking) used fine-grained serialization, which permitted any level of multiprogramming. By contrast, DB2's primary mainframe competitors—IDMS and Datacom/DB—had a threading model that topped out at two processors. As a consequence, DB2 could generate a higher transaction rate on IBM's latest processors. This and other improvements propelled DB2's adoption.

By 1989, half of all mainframe customers had DB2 installed. Furthermore, hardware sales driven by DB2, coupled with software sales of DB2 and related offerings, repaid IBM's costs to develop DB2.[38]

By 1989, half of all mainframe customers had DB2 installed.

In 1989, Tandem delivered Nonstop SQL, which would become a formidable DB2 competitor until the mid-1990s. Midrange systems lacked the throughput offered by mainframes until the mid-1990s, so Oracle, Sybase, Informix, and Ingres didn't compete directly with mainframe DB2 until then.

With the delivery of DB2 version 2 in 1988, DB2 was primed for widespread deployment of online transaction processing applications with acceptable performance and cost coupled with tools to manage it. And the product bug rate was now tolerable. Decision support was widely used with increasing complexity of analysis and increasing database sizes (100 Gbytes at this time).

Stage 3: Availability, Performance, and Distributed Databases

While reveling in the success of version 2, the DB2 team still faced many challenges. From 1988 through 1991 (version 3), DB2 continued to improve compute cost and throughput for transactions. It also dramatically improved performance for business intelligence workloads.

Bala Iyer led the design of algorithms for data compression and sorting, working with the processor teams to process these functions efficiently. Storage costs plunged with hardware assisted data compression,[39] and processing cost for sorts (integral in query processing) also improved significantly with processor assists.[40] Yun Wang led improvements to the DB2 query processor,[41] and Iyer led work with the hardware teams, creating many of the query processing algorithms that would distinguish DB2.

In 1989 system availability became the top customer requirement. In the early 1980s, online systems were available for one or two shifts (16 hours) Monday through Saturday. Batch processing occurred on the remaining shift. On weekends, the system would be down for 12 to 24 hours for maintenance. By 1989, leading firms operated without downtime and needed 24/7 availability.

Jim Teng led DB2's effort on high availability over the next decade, working with the IBM storage group on hardware-related technical requirements. The DB2 recovery log was replicated locally and remotely. Recovery scripts were created to restart DB2 on the takeover system.

Version 3 of DB2 also delivered significant support for distributing computing. Work in this area began in 1987 when Jackson returned from the UK (where he had supported DB2's launch in Europe) to lead DB2's distributed database team with Curt Cotner, who had extensive experience building distributed applications. Client-server computing was taking off, thanks to the growing popularity of PCs and other devices. IBM Research had been exploring distributed topologies through System R*,[42,43] an extension to System R. Screen scrapers were available in 1988, which allowed PCs and other devices to act as 3270 terminals and access QMF and other MVS applications. But one could not easily have an application residing on the external device access DB2 on MVS.

IBM needed to establish protocols so applications and databases on heterogeneous platforms could communicate. This required cooperation from various IBM product teams, including DB2 on MVS, SQL/DS on VM and VSE, and DB2 on AS/400. IBM convened a cross-divisional team to create the Distributed Relational Database Architecture (DRDA).[44] DB2 implemented this in 1989,[45] allowing applications on MVS to remotely access another DB2 database. By 1991, applications outside of MVS could remotely access DB2 databases—a full-service client-server topology. Although DRDA allowed remote applications to connect to databases with different SQL capability, the ANSI/ISO Remote Database Access (RDA) standard (which was ratified in 1993) emphasized application portability with a limited SQL capability. The majority of the applications developed for DB2 used DRDA to take advantage of advanced function in the target database.

During this time, IBM's overall software efforts were growing more complex. The firm was struggling to continue to deliver high quality, timely software products. Given IBM's significant investments in software, this was no trivial matter. As a result, IBM formed a distinct software organization (Programming Systems) and appointed Earl Wheeler as its executive. DB2 and other select software offerings reported to this organization. However, hardware still drove IBM revenues, and the software division had to prove that it could meet the needs of IBM's hardware business.

Stage 4: Sysplex and Open Systems

DB2's exploitation of MVS hardware and software represented a large part of its value, and a cross-product Architecture Review Board (ARB) helped IBM achieve this. Led by Gary Ferdinand, the board included representatives from the broad MVS ecosystem, including processors, storage, IMS, CICS, and DB2. The board established blueprints that shaped joint initiatives, synchronized development plans, and coordinated delivery schedules. After all, if the hardware improved but other components didn't support it, then it had no value.

Data Sharing

Through this board, one topic surfaced that shaped much of DB2's development efforts in the early 1990s: supporting a massively parallel processing (MPP) mainframe. In 1986, IBM hardware experts led by Rick Baum projected that improved microprocessors and network interconnects would enable new platforms to compete against mainframes at a much lower cost during the next decade. Teradata, based on Intel's x86 microprocessor, was an example of this. IBM began rebasing its mainframes on microprocessors. To scale effectively, multiple SMPs were linked to create an MPP topology that allowed data sharing; that is, all data was accessible from any node in the MPP configuration. This configuration was dubbed the Sysplex, and it caused a major redesign of the DB2 kernel.

C. Mohan, Inderpal Narang, Jeff Josten,[46] and Jim Teng developed the necessary algorithms for DB2, working with other MVS experts. (For complete attribution, see the IBM Systems Journal dedicated to the Sysplex.[47]) Josten and Teng undertook the enormous effort to upgrade DB2—an effort that touched the very core of the product, including serialization, buffer management, and recovery. DB2 delivered Sysplex support in 1995.

At the same time, Yun Wang led the transformation of DB2 query processing to support intra-query parallelism,[48,49] which was critical for strong runtime performance of data-intensive queries. The approach was delivered in stages from 1993 to 1997, culminating with Sysplex query parallelism in version 5. DB2's improvements in this area enabled the product to support query processing of large databases (terabytes or more).

Portability and the IBM Software Business

During this time, IBM software management changed dramatically. Its Programming Systems Division had demonstrated that it could improve product quality and adequately serve IBM's hardware business. Lou Gerstner, then CEO, consolidated all software into a single Software Group run by Steve Mills, who convinced IBM to invest in software as a business. The new division continued to support IBM hardware but had its own revenue targets and maintained a dedicated software sales force. In time, this team would drive billions of dollars in IBM software revenue through RDBMS sales.

Independent software vendors such as Oracle, Informix, Ingres, and Sybase proved popular with customers on open systems platforms, competing and winning business away from IBM. However, once IBM perceived software as a distinct business opportunity, this opened the door for IBM to offer a relational DBMS on non-IBM platforms, including HP, Sun, and Windows.

To establish an IBM presence as quickly as possible, Janet Perna formed a dedicated "workstation" DBMS team in Toronto, Canada. This team partnered with IBM Research and the DB2 developers in California to deliver a respectable IBM alternative. (A separate organization in Rochester, Minnesota, built DB2 on AS/400. This offering, released in 1988, evolved from IBM's System/38, which also shipped with a RDBMS.)

IBM concluded that developing a common code base for DB2 across all platforms would be too expensive for existing products (DB2 on MVS, SQL/DS, and DB2 on AS/400) and hinder quick support for new platforms. Consequently, IBM developed DB2 for open systems on a new code base written largely in C. IBM ported this DB2 code base to a variety of non-mainframe platforms, including OS/2, Windows, Unix, and Linux, and integrated advanced technologies from research projects (such as Starburst[50]) into the engine. Developers and researchers continued their collaboration in subsequent releases, emphasizing scalability, performance, and other features.[51]

The DB2 product suite drives billions of dollars in revenue to IBM annually and supports mission-critical applications for most major corporations around the world.

IBM strove for functional consistency across its DB2 product suite, but customer priorities differ among mainframe and non-mainframe users. IBM formed a cross-product DBMS technical council to drive SQL consistency across its product line and minimize changes required for applications that needed run against different DB2 offerings. For the most part, customers found this approach acceptable. IBM added various SQL extensions to its DB2 offerings during this time, including table expressions, outer joins, recursion, stored procedures, triggers, and other features to enhance query capabilities, comply with standards, and address customer demands. In addition, support for double-byte characters allowed DB2 to host native text for the many alphabets that exceed 256 characters (such as Chinese, Japanese, and Korean).

DB2 for Unix/Windows faced considerable competition when it was first released. Competitors had a foothold in many accounts, and they had a healthy suite of business partners and value-added resellers. Increasingly, sales of DBMS, and other, software were being driven through this indirect channel, and IBM's Software Group formed a dedicated team to pursue alliances. IBM established partnerships with hardware vendors that competed with offerings produced by other IBM divisions. Customers' preferences for "open systems" mandated this shift.

More Growth Opportunities

Data warehousing also emerged in the early and mid-1990s as a new direction for DB2. Query language extensions, improved autonomics, indexing enhancements, improved database scalability, and new query optimization techniques (such as new join methods) were among the technologies needed to address the demands of data warehouse customers running online analytical processing, business intelligence, and data mining workloads. IBM ultimately released its InfoSphere Warehouse product suite, featuring DB2 as the underlying DBMS.

Evolving customer workloads prompted further changes to the DB2 product suite over time. For example, IBM invested in DBMS tools to simplify DB2 application development, database management, and analytics. Another effort involved supporting database federation technology, which presents users with a single-site image of disparate and physically distributed data sources.[52] Web-based computing drove support for Web service access to DB2 data and native support for XML data. The latter allows hierarchical structures to coexist with tabular structures in the same database and provides ''bilingual'' query services through new support for XQuery (XML Query).

Conclusion

Today, DB2 enjoys a wide installation base among enterprises throughout the world. The product suite, anchored by its flagship mainframe DBMS, drives billions of dollars in revenue to IBM annually and supports mission-critical applications for most major corporations around the world. However, its success wasn't a given during its early days, as DB2 faced considerable business and technical challenges during its formative years. Ultimately, combined efforts from key IBM business leaders, developers, and researchers enabled IBM to overcome these challenges and deliver a popular suite of software offerings based on RDBMS technology.

Acknowledgments

We thank those who provided materials or comments helpful to this article: in alphabetical order, Marilyn Bohl, Don Chamberlin, Burt Grad, Warren Lucas, Roger Miller, Norris van den Berg, and Mel Zimowski.

References and Notes

1. ''Relational Data Base Management System Announced for Large Enterprises,'' press release ISG047, IBM, June 1983.
2. E.F. Codd, ''A Relational Model of Data for Large Shared Data Banks,'' *Comm. ACM,* vol. 13, no. 6, 1970, pp. 377–387.
3. M. Stonebraker, ''Retrospection on a Database System,'' *ACM Trans. Database Systems* (TODS), vol. 5, no. 2, 1980, pp. 225–240.
4. D.D. Chamberlin et al., ''SEQUEL 2: A Unified Approach to Data Definition, Manipulation, and Control,'' *IBM J. Research and Development,* vol. 20, no. 6, 1976, pp. 560–575.
5. S.J.P. Todd, ''The Peterlee Relational Test Vehicle – A System Overview,'' *IBM Systems J.,* vol. 15, no. 4, pp. 285–308.
6. M.W. Blasgen et al., ''System R: An Architectural Overview,'' *IBM Systems J.,* vol. 20, no. 1, 1981, pp. 41–62.
7. Donald Chamberlin and Raymond Boyce invented SQ L. For his efforts, Chamberlin was named IBM Fellow in 2003, ACM Fellow in 1994, IEEE Fellow in 2007, and CHM Fellow in 2009.
8. In 1978, IBM announced S/38, which featured a relational DBM S. General availability occurred in 1980.
9. Frost and Sullivan, ''Data Base Management Services Software Market,'' report A747, 1979, pp. 208–213.
10. Chris Date worked on Eagle's development of the Unified Data Language (UDL)[i], a single data language that would encompass all data models. Franco Putzolu from IBM Research worked with Dick Crus and others to extend System R's data manager to support multiple data models. Don Haderle led the design for the overall subsystem. See C.J. Date ''An Introduction to the Unified Database Language (UDL)'' Proc. 6th Int'l Conf. Very Large Data Bases, 1980.
11. Even today, firms' appetites appear to be insatiable. Chris Murphy, ''IT Is Too Darn Slow,'' *InformationWeek,* 28 Feb. 2011.
12. R.A. Crus, ''Data Recovery in IBM Database 2,'' *IBM System J.,* vol. 23, no. 2, 1984, pp. 178–188.
13. J.A. Teng and R.A. Gumaer, ''Managing IBM Database 2 Buffers to Maximize Performance,'' *IBM Systems J.,* vol. 23, no. 2, 1984, pp. 211–218.
14. DB2 was written in PL/S, a 3GL similar to PL/1 but with language constructs that allow system privileged requests (e.g., changing protect keys). By contrast, the System R prototype was written in PL/1.
15. D.J. Haderle and R D. Jackson, ''IBM Database 2 Overview,'' *IBM Systems J.,* vol. 23, no. 2, 1984, pp. 112–125.
16. C. Loosley, ''IBM Database 2 Performance Measurements,'' *Info IMS,* vol. 5, no. 1, 1985.

17. J.J. Sordi, ''The Query Management Facility,'' *IBM Systems J.,* vol. 23, no. 2, 1984, pp. 126–150.
18. J. Boyle et al., ''QMF Usability: How it Really Happened,'' *Proc. IFIP INTERACT: Human-Computer Interaction,* IFIP, 1984, pp. 877–882.
19. J. Zachman, ''Business Systems Planning and Business Information Control Study: A Comparison,'' *IBM Systems J.,* vol. 21, no. 1, 1982, pp. 31–53.
20. Leading IBM's early participation in SQL standards efforts were Bob Engles (who wrote the first draft of the ANSI SQL standard submission) and Phil Shaw (who was IBM's initial representative on the ANSI SQL committee). In later years, IBM's key contributors to the SQL standards efforts included Hugh Darwen, Nelson Mattos, and Mel Zimowski.
21. In 1985, IBM's 3081 processor executed 5 to 10 MIPs (million instructions per second) with 32 Mbytes of main memory. In 2011, IBM constructed Watson to compete on the game show *Jeopardy* using 10 racks of IBM Power 750s with 2870 processor cores able to execute 80 TIPs (trillion instructions per second) with 15 Tbytes of main memory.
22. ''IBM Database 2 (DB2) Is Announced with Availability Planned for Third Quarter 1984,'' announcement letter 5740-XYR, IBM, 1984. (General availability actually occurred in 1985.)
23. Chris Loosely led this effort.
24. Jim Ruddy, who worked for the Boeing database team from 1978 to 1995, provided valuable insight. In 1995 he joined IBM DB2 development, where he helped to make life better for DB2 DBAs.
25. John Naumann and others departed to Tandem; Jerry Baker, John Mortenson, and David Beech joined Oracle; and Marilyn Bohl moved to Digital Research. In addition, E.F. Codd and Chris Date formed a consulting company, as did Colin White and Chris Loosely.
26. Bob Ojala and Fred Orosco streamlined the interfaces between IMS, CICS, and DB2.
27. D. Hauser and A. Shibamiya, ''Evolution of DB2 Performance,'' *InfoDB,* vol. 6, no. 4, 1992, pp. 2–13.
28. C. Mohan et al., ''ARIES: A Transaction Recovery Method Supporting Fine-Granularity Locking and Partial Rollbacks Using Write-Ahead Logging,'' *ACM Trans. Database Systems* (TODS), vol. 17, no. 1, 1992, pp. 94–162.
29. P.E. O'Neil, ''Model 204 Architecture and Performance,'' *High Performance Transaction Systems,* LNCS 359, Springer-Verlag, 1989, pp. 40–59.
30. Steindel and Madson published the benchmark in a paper in 1987. Neither they nor we have been able to resurrect a copy of that paper.
31. C. Mohan et al., ''Single Table Access Using Multiple Indexes: Optimization, Execution, and Concurrency Control Techniques,'' *EDBT,* 1990, pp. 29–43.
32. J.M. Cheng et al., ''An Efficient Hybrid Join Algorithm: A DB2 Prototype,'' *Proc. 7th Int'l Conf. Data Eng.* (ICDE), IEEE CS, 1991, pp. 171–180.
33. In particular, Charles Bontempo of IBM's Systems Research Institute taught database management courses to IBM's best and brightest.
34. Warren Lucas, one of IBM's first software sales specialists, was instrumental in driving DB2 into New York accounts and vital to DB2's success.
35. Controversy about compliance with E.F. Codd's 12 rules for relational technology arose in the trade press, including *Computerworld*'s Oct. 14 and 21, 1985 issues. However, this had little impact on DB2 sales because potential customers gave higher priority to serviceability, availability, performance, efficiency, and other areas.
36. Vern Watts, one of the founding fathers of IMS, was an ardent DB2 supporter who helped IMS customers distinguish use cases for DL/1 versus DB2. He helped DB2 establish a foothold in many IMS accounts.
37. O. Stevens Jr., ''The History of Datacom/DB,'' *IEEE Annals of the History of Computing,* vol. 31, no. 4, 2009, pp. 87–91.
38. As a humorous aside, the hardware divisions paid for DB2 development costs. Just as it was about to be cash positive, DB2 was moved into PRGS (software division), which then reaped the benefits. To this day, IBM hardware venture capitalists remind Haderle how the software division ''stole'' their investment. Ironically, sales from DB2 on z/OS have since been underwriting other IBM software development projects for years.
39. B.R. Iyer and D. Wilhite, ''Data Compression Support in Databases,'' *Proc. 20th Int'l Conf. Very Large Data Bases* (VLDB), Morgan Kaufman, 1994, pp. 695–704.
40. P.J. Varman, B.R. Iyer, and D.J. Haderle, ''Parallel Merging on Shared and Distributed Memory Computers,'' *PARBASE / Architectures,* 1990, pp. 231–249.
41. P. Gassner et al., ''Query Optimization in the IBM DB2 Family,'' *IEEE Data Eng. Bull.,* vol. 16, no. 4, 1993, pp. 4–18.
42. R. Williams et al., ''R*: An Overview of the Architecture,'' *Proc. Int'l Conf. Data and Knowledge Bases* (JCDKB), 1982, pp. 1–27.

43. B.G. Lindsay, ''A Retrospective of R*: A Distributed Database Management System,'' *Proc. IEEE,* vol. 75, no. 5, 1987, pp. 668–673.
44. R. Reinsch, ''Distributed Database for SAA,'' *IBM System J.,* vol. 27, no. 3, 1988, pp. 362–369.
45. George Zagelow managed this effort. The technical cross divisional lead was Roger Reinsch. DBTI contributed as well from IBM Research, particularly Pat Selinger and Bruce Lindsay.
46. J.W. Josten et al., ''DB2's Use of the Coupling Facility for Data Sharing,'' *IBM System J.,* vol. 36, no. 2, 1997, pp. 327–351.
47. ''S/390 Sysplex Cluster,'' *IBM Systems J.,* vol. 36, no. 2, 1997 contains nine articles providing an overview of the technology.
48. Y. Wang, ''DB2 Query Parallelism: Staging and Implementation,'' *Proc. 21th Int'l Conf. Very Large Data Bases* (VLDB), Morgan Kaufman, 1995, pp. 686–691.
49. C. Mohan et al., ''Parallelism in Relational Database Management Systems,'' *IBM System J.,* vol. 33, no. 2, 1994, pp. 349–371.
50. L.M. Haas et al., ''Extensible Query Processing in Starburst,'' *Proc. 1989 ACM SIGMOD Int'l Conf. Management of Data,* ACM, 1995, pp. 377–388.
51. C.K. Baru and G. Fecteau, ''An Overview of DB2 Parallel Edition,'' *Proc. 1995 ACM SIGMOD Int'l Conf. Management of Data,* ACM, 1995, pp. 460–462.
52. V. Josifovski et al., ''Garlic: A New Flavor of Federated Query Processing for DB2,'' *Proc. 2002 ACM SIGMOD Int'l Conf. Management of Data,* ACM, 2002, pp. 524–532.

Donald J. Haderle is a retired IBM Fellow and former chief technology officer of IBM's information management organization. After retiring from IBM, he founded Haderle Consulting and served on advisory boards at several software firms, including Vertica Systems, Aerospike, Boardwalktech, and Parstream. He is an ACM Fellow and was elected to the National Academy of Engineering in 2008. Today, he continues to consult with venture capitalists, start-up companies, and other businesses. Contact him at donhaderle@yahoo.com.

Cynthia M. Saracco is a senior software solution architect at IBM's Silicon Valley Laboratory. She specializes in emerging technologies and information management. For more than a decade, she worked for Don Haderle on information management strategy issues spanning a diverse range of technologies, including business intelligence, spatial data, temporal data, object-oriented data management, database federation, XML data management, and database application development for the Web. Contact her at saracco@us.ibm.com.