IDC
Analyze the Future

WHITE PAPER

A Platform for Enterprise Data Services: The Proven Power and Flexibility of IMS from IBM

Sponsored by: IBM

Carl W. Olofson
October 2009

IDC OPINION

Enterprises are engaged in a wide range of efforts to integrate their data and make it available to a new class of applications, resulting in a heightened demand for constantly available data and a platform for managing that data that is scalable and continuously available. They have shown a willingness to investigate a range of new technologies, even moving outside the relational sphere, to accomplish these objectives.

Moving forward, as structured enterprise data is increasingly being integrated with other forms, such as various content formats and nonrelational data structures, relational databases are proving inadequate for some of the workloads involved. Service-oriented architecture (SOA) and Web-based applications demand high throughput and reliability as well as support for complex data structures. As enterprises consider exotic new approaches to data management, one well-established option is often overlooked: IBM Information Management System (IMS).

Many enterprises believe that mainframe technology in general and IMS in particular are "old technology" and nearly obsolete. But in fact, mainframe systems and IMS have been continuously evolving, incorporating the very newest in hardware and software technology in order to deliver data services that:

- Are continuously available on a round-the-clock basis
- Deliver high performance coupled with extreme reliability
- Ensure efficiency and cost-effectiveness at steady levels as data volume grows
- Offer high availability and scalability in a highly manageable environment
- Reduce the risks to the business that accompany potential downtime

METHODOLOGY

This white paper was prepared using data collected from IBM briefings and materials, interviews with a number of high-end IMS users, and background knowledge gleaned from many years of IDC research on the subject of large-scale database management.

Global Headquarters: 5 Speen Street Framingham, MA 01701 USA P.508.872.8200 F.508.935.4015 www.idc.com

IN THIS WHITE PAPER

This white paper considers the need for alternatives to the relational model in dealing with certain classes of data required in these applications. It outlines the limitations inherent in the relational data management approach in dealing with certain workloads and use cases — limitations that make the relational database management system (RDBMS) a less than ideal choice for some SOA and Web-based applications. The document looks at common prejudices regarding the mainframe computing environment in general and IMS in particular. It shows how well-established technology is not necessarily "old technology" but can be just what's needed to achieve operational and service-level agreement (SLA) requirements.

This paper reviews how, despite the rise of relational data management as the dominant paradigm for databases, IMS and its hierarchical approach remain vital for certain workloads. It shows how IMS is well-suited to the management of a variety of difficult data structures, including so-called "bill of materials" structures, and how it is ideal as a repository for XML data. The document also looks at the way IMS supports SOA applications. It also illustrates how, through the connectivity features of z/OS and IMS, this DBMS can provide services to Java or .NET applications that are designed to interoperate with it at a Web services level, requiring no explicit knowledge of IMS on the part of either the programmer or the user.

This paper shows how IMS, which operates in the centrally managed mainframe context, is straightforward to administer and how that task has been made even easier by advanced tools. It also incorporates the experiences and observations of a number of IMS users who have applied its technology to highly challenging business problems.

SITUATION OVERVIEW

Many of the world's largest and most critical database workloads are managed on the IBM mainframe-based hierarchical DBMS, IMS. Although these databases have run securely and reliably with good performance for decades, some people feel that IMS represents "old technology" and should be replaced. They note the emergence of scalability and high-availability technology in various distributed relational DBMS products and believe that those products are cheaper to manage and offer a better future direction.

It is certainly a prudent and intelligent practice to review one's technology portfolio from time to time and ask whether that portfolio is continuing to meet the needs of the enterprise now and whether it will continue to do so in the future. In examining a DBMS, one should answer the following key questions:

- Do the overall service levels of the DBMS (availability, scalability, reliability, security, and performance) continue to meet the needs of the enterprise?
- Does the DBMS have a strong future in the enterprise? Can it be staffed appropriately going forward, and does the vendor offer a compelling vision for the future?

- Are there other products available that meet the following criteria?

  - Are just as suitable for the workload and data organization requirements of the database

  - Offer the same or better service levels for that workload

  - Present as compelling a future vision

  - Are so much less expensive to operate and maintain in terms of total cost (hardware, software, and staff time) that they justify the significant cost and risk associated with database and application conversion to a completely different method of data management

It is important to answer each of these questions carefully and thoughtfully, based on the results of research and investigation.

Evaluating IMS and the IBM Mainframe Environment

It is important in considering the true worth of a mainframe environment to understand the nature of that environment and how it differs from the leading reduced instruction set computer (RISC) or x86-based computer technology; the latter two being referred to as "distributed systems." Because of the simplicity of management and low cost of these distributed systems, they have become dominant in most facets of business computing. They are definitely not the preferred choice, however, for all business computing workloads. Mainframes are often the better choice for key foundation applications, especially those that demand very high throughput, reliability, and availability and high degrees of integrated operations among related software systems.

Mainframe Versus Distributed Systems

As was mentioned earlier, distributed systems tend to be cheap and relatively easy to deploy individually. For large, complex application systems that require large amounts of computer processor power, they are often networked together, sometimes into "clusters," which are computers connected through a high-speed link that use special features of the operating system and file system to run cooperatively as a single virtual system. When distributed systems are combined in this way, they quickly form highly complex computing environments that require teams of people to manage and are subject to many possible failures mainly due to human error. They are normally combined into such complex clustered systems to handle very large, very high throughput workloads that require extreme reliability.

Mainframe systems are capable of such workloads without being combined in a complicated manner because they are designed to work this way. When more extreme scalability is required, the systems can be combined in shared workload combinations through a feature known as Parallel Sysplex. Since mainframes are designed to interoperate in a Parallel Sysplex environment, setting up mainframes is relatively straightforward compared with setting up large clusters of distributed systems. Why is this?

A Mainframe Computer Is an Integrated Network of Specialized Computer Components

Mainframe computers are usually compared incorrectly with their distributed counterparts. A mainframe computer is not the equivalent of a distributed computer, even a very large one. This is because a mainframe computer is really a preengineered network of specialized processors that automatically distribute and coordinate compute, I/O, and network operations. Even its performance is measured using a different metric. Because distributed system processors perform relatively simple operations, they are measured using clock pulses or ticks in a metric called a "cycle." Mainframe operations are complex, and because they are designed for high degrees of multiprocessing and virtualized interrupt handling at the chip level, there is no such thing as a "cycle," so they are measured by the metric "millions of instructions per second" (MIPS).

A distributed system handles everything — compute instructions, memory fetches, I/O operations, etc. — through a single processor or set of processors, each of which may or may not have multiple cores. Mainframe computers have a set of processors for compute instructions and memory fetches, another set for I/O operations, and another set just for managing the state of the system. Distributed systems have only one kind of interrupt. It actually interrupts the running process. If the user hits a key on the keyboard, it interrupts the process. If an I/O device returns a value to the system, it interrupts the process, etc. Mainframe computers have such interrupts also, but they are called "immediate interrupts" and normally indicate an error state. All other interrupts are handled outside the main process, setting flags that the OS deals with in due course. These are called "deferred interrupts." Also, most of the processing involved in disk, terminal, network, or other I/O is not done by the central processor but by specialized processors designed to control terminals, disk systems, networks, and so on.

Because of these differences, mainframe computer processes flow more smoothly than those in their distributed counterparts, especially when many different workloads are running on the same system. Also because of these differences, it is not appropriate to think of a mainframe as a single computer compared with distributed systems; rather, a mainframe should be thought of more as a tightly integrated network of related specialized computer components clustered around a set of central processors.

Mainframes Use Virtualization to Deliver Consolidated Operations

Over the years, distributed systems hardware and software vendors have developed a number of technologies and techniques enabling combinations of distributed systems to emulate some of the capabilities of mainframes. These technologies and techniques include clusterware, clustered file systems, and virtual machine software. They have all included using virtualization to insulate applications and databases from the specifics of the hardware. This is necessary because in distributed systems, applications and databases normally refer to their physical resources directly by name or address.

By contrast, mainframe computers have always featured virtualization since the first System/360 rolled off the line in the early 1960s. Virtualization on the mainframe has evolved to support:

- References to files and devices by symbolic names
- References to memory locations by symbolic names
- Virtual memory addresses, using disk-based paging so that the amount of addressable memory could exceed the actual memory on the computer
- References to databases, applications, and all other resources by names that are easily changed by system programmers, applications managers, and database administrators without changing any source code, using standard system management features

As a result, the mainframe has tables, catalogues, and directories that show exactly where and what everything is and allow for easy management of applications, databases, their environments, their files, their security, and so forth, in a centralized way.

Because mainframes have evolved in this way, they typically host many applications and databases, while distributed systems are usually specialized and host just one application or database. In fact, virtual machine software is commonly used to "consolidate" applications and databases on distributed systems. Virtualization and workload consolidation are a natural state for the mainframe. Because of this, the mainframe environment ensures that mixed workloads are completely isolated and secure.

How IMS Is Different from Other DBMS Products

IMS is a hierarchical database management system designed and developed for use on IBM mainframe computers. It includes built-in multiprocessing, coordinated data access support, and a data access language called DL/I. It has evolved since the mid-1960s to handle a wide variety of data management workloads and remains commonly used for manufacturing, finance, retail, telecommunications, and highly complex records and accounts management applications, among others. Although its install base has grown very slowly compared with the install base of the leading alternative, which would be relational database management systems, its customer base has remained steady due to the critical nature of the data it manages and due to the fact that it does its job in a way that is hard, if not impossible, to duplicate.

Relational Database Management Systems

The dominant paradigm today for database management is relational, which represents all data in the form of tables having rows and columns, with some columns containing primary and secondary keys and other columns containing references (called foreign key references) to the keys of other tables. This approach is based on the work of IBM Fellow Dr. E. F. Codd, who used mathematical set theory to drive his work. It has been quite popular since the mid-1980s because of its simplicity and general applicability in managing business records. Relational database management has been found quite suitable for managing tabular data such as accounting data. Because it features a relatively standardized access language (SQL) and access interfaces (ODBC and JDBC), it enables a community of DBAs to emerge that can address the DBMS products of multiple vendors. This provides a ready staffing pool, makes it easy for programmers to develop in a nonvendor-specific way, and presents the illusion of portability, although porting applications from one vendor's relational DBMS to another's is nightmarishly difficult and risky.

The flat, tabular data model is not, however, appropriate for all types of data structure. The relational data model was developed by Dr. Codd as a means of providing easy access to a wide range of data without requiring special knowledge about the organization of the database. It is not aimed at operational efficiency or at handling all manner of possible data structures and relationships.

IMS and the Hierarchical Data Model

IMS is based on a hierarchical data organization technology that has its roots in the first managed database projects of the early 1960s, with a long series of innovations developed by a talented team, led by IBM's legendary Vern Watts. Unlike a relational database, IMS organizes its data in operational hierarchies (that is, hierarchies based on the normal aggregation of records for bulk processing). In this hierarchy, records (called "segments") are linked together, which supports very fast access for the mainline processing for which the hierarchy is designed. To move through a data structure in a relational database, a SQL request must contain table joins, each of which is resolved by the RDBMS by performing an index lookup, finding the matching rows, retrieving them, finding the foreign key values, performing another index lookup, finding the matching rows of the joined table, finding the foreign key values, etc. In IMS, the DBMS simply walks the chain from segment to segment, retrieving large chunks of related data in a single operation.

The hierarchical data model is very good for dealing with complex data structures having many levels, especially when many related records (or segments) must be retrieved together. It is especially popular for manufacturing applications, for instance, because of the ease with which it handles parts databases. These "bill of material"–like structures require functions that explode (find the parts of parts) or implode (find the parts of which a given part is a component) a part structure. Relational databases can't handle this as a smooth series of operations, instead requiring the execution of a SQL statement within a program loop that contains explosion or implosion application logic.

IMS should not be regarded as superior across the board for all data management problems; many workloads are more appropriately handled using relational technology. But it must be acknowledged that certain workloads, especially those involving complex data structures with many levels and those requiring very high transaction throughput, require a DBMS with the features and capabilities of IMS and are not well-served by relational DBMSs.

Projects that involve migration of data and applications from IMS to a relational database usually run much longer and cost much more than originally estimated, and if they finish at all, the results are generally quite unsatisfactory. In researching this paper, this analyst learned of two such projects that ran well over budget, did not deliver the desired result, and were ultimately abandoned. Such stories are common, and past research has encountered a number of them.

IMS as an Application Platform

Unlike most DBMS products, IMS is also a platform upon which applications may be deployed. It has a fully secure multiprocessing environment capable of supporting a large number of applications serving thousands of concurrent users. This transaction platform is known as IMS Transaction Manager (TM). Despite the common perception that mainframe facilities such as IMS are executed only as isolated, backroom functions accessed through "green screen" terminals, TM makes IMS an integral part of an SOA-based, grid-based, or .NET-based distributed computing environment.

TM can act as a platform for services that interact with clients using standard Web services conventions. Such conventions include the Simple Object Access Protocol (SOAP) and the Web Services Description Language (WSDL). In addition to the common mainframe programming languages (Assembler, PL/I, COBOL), TM also supports service code written in Java, C, and C++. Such code not only can take advantage of the reliability and high throughput characteristics of the mainframe but also can be integrated into an environment that is managed through the same centralized management facilities that govern all mainframe applications code, making such things as resource and standardized security management quite simple. Services deployed in TM are also fully compatible with and functional in the .NET environment.

IMS Tools and the Question of Skilled Staff

Over the years, IBM has continued to develop and provide tools that help automate and ease the activities involved in IMS database development and management. This is important because as the existing workforce ages, many managers are wondering where the next generation of IMS DBAs and application developers will come from.

The IBM suite of IMS tools and associated tools for cross-database development and management of databases, database applications, and data have evolved to a point where a generally competent IT professional can understand IMS and how to manage it without having to acquire specialized training and years of experience. In addition, IBM has instituted a program called the Academic Initiative for System z, which has enrolled over 600 universities and technical schools around the world, producing a new generation of computer technicians with System z and IMS skills. Finally, IBM is in the process of launching over 20 regional IMS User Groups worldwide. These groups meet two or three times a year and provide IMS users free opportunities to update their technical skills and share experiences with other local IMS users. For all these reasons, worries that it will be difficult to staff IMS projects in the future are generally unfounded.

XML, SOA, Open Connectivity, and Cloud Computing

IMS should not be considered a platform only for "legacy" applications. It has a variety of capabilities, which, in combination with its native functionality, make it a suitable platform for data services and operations in the next generation of computing environments, including XML management, SOA-based data services, and cloud deployments.

Because of its hierarchical organization, IMS is a natural fit in support of XML, which is also organized hierarchically. A document structure can be represented in an IMS schema in a completely straightforward way, allowing efficient access and management of XML data.

SOA applications involve deploying the application as sets of components that provide published services over a network. The services are usually defined using WSDL, and the protocols are governed using SOAP. Data services in this environment are independent of the specifics of the database and often require quick responses to what are, in some cases, complex data requests. IBM has built into the z/OS environment support for linking its components, including IMS, to the network. IMS provides capabilities to access IMS transactions and data through Web services in an SOA environment. These and other TM features allow IMS to be used as a resource for the newest of applications, including those written in Java and in the .NET environment.

New with IMS 11 is the ability to directly support access across a network using standard facilities such as Java Connector Architecture 1.5 (Java EE), JDBC, SQL, and DRDA. This means that any application on any platform on the network can interact with IMS just as easily as it could with any standards-supporting relational DBMS. It also means that IMS can be integrated transparently into the same distributed applications and grid-based application deployment constructs as any other leading data source.

Providing services "in the cloud" requires dead certain continuous availability with high performance and throughput at all times. Cloud-based data services are often multitenant and multiapplication, requiring a data management platform that is capable of highly complex workloads and configurations and yet is easy to manage with built-in features designed to mitigate the complexity. IMS is such a platform. In fact, from a cost perspective, the more it's loaded, the cheaper it becomes by any unit of measure (total data under management, transactions, concurrent users or threads, etc.).

Considering the Alternatives

Despite the foregoing discussion, many people believe that migrating their applications to a relational database would be cheaper and less risky in the long run. A strong case can be made, however, that just the opposite is true.

The Relational Data Model

To begin with, one should consider the relational data model itself. This is a model, as was previously stated, that represents data in flat table structures with references to other tables. It does not explicitly support complex data structures, and when such structures are defined in a relational schema, the hierarchical nature of that structure is in the head of the data designer, in the data models, and in documentation, but not in the schema itself. This means that the database cannot protect the integrity of that structure or optimize access based on that structure. It simply does not exist in the operational parameters of the RDBMS. This fact alone has made a fat living for some DBAs and database consultants who specialize in using arcane tuning techniques to make a relational database do what a hierarchical database does naturally: optimize data management based on its logical structure.

Besides this, there are some data structures that IMS handles well but that relational databases deal with in a very awkward way. An example given earlier is that of the "bill of materials" structure. This analyst has spoken with a number of people who have found that after attempting conversion of databases containing such structures from IMS (and other hierarchical or network databases) to a relational database, the database operations are so cumbersome and inefficient that the resulting application simply cannot meet its required service levels, even with extreme tuning, very large buffers, and high-end symmetrical multiprocessor (SMP) systems.

The Cost and Risk of Conversion

In calculating the cost and evaluating the risk involved in converting from IMS to a relational database, one should first consider the required process. The following is a highly simplified representation of that process:

1. Acquire, load, configure, and deploy the system or systems intended to support the new database and its applications (assuming the migration is to a distributed platform). This means buying computers, network equipment, and storage arrays and hiring the administration staff to manage them all. It also means buying the relational DBMS, management tools, development tools, and perhaps some applications software and probably hiring a consultant to help the staff set it all up.

2. Analyze the existing IMS schema, and develop a relational schema that, as nearly as possible, supports the data operations and definitions that are supported by the IMS database. Because this involves translating hierarchical data organization into a relational form, this is not a simple activity.

3. Completely rewrite every application that uses IMS as an application using the new relational database. This is necessary because IMS applications are designed based on the structure of the IMS database, and their organization is not at all suitable for using a relational database. This is a highly error-prone process, especially where there is code that is 20 years old or older, because it has been maintained for so long that understanding and duplicating its functionality in detail is decidedly problematic.

4. Migrate the data. This generally involves custom code, although there are some data migration tools available. Even with commercial data migration tools, however, the transformation rules that get the data from its hierarchical source to its relational target will be as complex as any application program, and again, there is a strong risk of introducing data errors at this stage.

5. Parallel test. Before one pulls the plug on IMS, it is customary to run parallel tests of all application functions, including online and daily, weekly, monthly, quarterly, and annual batch operations. Online parallel testing should be done for at least three months, which would require duplicate operations for that period and the staff to support it. During this step, one must also adjust and tune the new system to attempt to emulate the service levels of the IMS system. Most users underestimate the difficulty of this task.

6. Pull the plug on the old system, rationalize the staff, dispose of the old systems, and yes, all that costs money, too.

Calculating the True Cost of Ownership

The costs involved in carrying out these activities include a substantial outlay for equipment, floor space and air conditioning during parallel testing (or lab time, if an external lab is used), software license and maintenance fees, hiring new staff, hiring consultants, and retraining or laying off old staff. An additional, usually overlooked cost is the sometimes significant disruption of business caused by switching to a new system, retraining business users, and dealing with the almost inevitable outages that occur in the early going of the transition. Beyond all of this, there is a considerable risk of failure, forcing the organization to abandon the project and eat all the costs incurred.

Several of the sources interviewed for this paper indicated that senior management, uncomfortable with the idea that their businesses were dependent on "old technology," had pressed their technical teams to investigate conversion to a relational distributed system database. Each determined that the costs and risks easily outweighed the benefits and that no other system could deliver the functionality of the IMS platform. In one case, a conversion was actually attempted, then abandoned, because the technical team could not bring even a subset of its application functionality up to an acceptable level of performance on a distributed system RDBMS, even with the vendor's own consultants involved.

IMS Customer Experiences

How Many IMS Administrators Does It Take to Drive over 70 Million Banking Transactions per Day for over 820 Banks?

In this case, the answer is 10. Fiducia, a major IT service provider in Germany with 45 years of experience in the field, provides IT services to over 820 banks representing a total of 10,217 branches; 101,500 workstations; 6,548 local servers; and 22,832 ATMs. It supports 63 million accounts, 5.5 million of which are directly managed online by customers. The underlying data engine is a configuration of eight IMS instances deployed on two Parallel Sysplex configurations of System z systems with total processing power of 55,000 MIPS. These systems drive an average of 2,750 banking transactions per second. In addition to the eight main systems, four others provide supporting operations. DB2 for z/OS and another relational DBMS are linked to IMS for query and reporting support. These systems are linked together and accessible by a variety of means, including IMS Connect for TCP/IP, DB2, and CICS.

In addition to the 12 IMS production systems, 12 IMS systems are used for development and testing. The production systems together have averaged 99.7% availability over the past few years. Most administrators of a distributed system supporting even a fraction of this workload would be thrilled with such availability. The entire management staff is 22 people, including 10 administrators supporting the IMS systems (including their development and production databases), eight DBAs supporting the DB2 databases, and four supporting the integration environment, which is deployed using WebSphere MQ. By contrast, most distributed environments running RDBMSs require a team of programmers for each database of this size and scope. The IMS staff manages 12 production and 12 test databases systems, 24 in all.

Fiducia is in the business of enabling banks to conduct business. Its mandate is to ensure the ability of its banks to do so with absolute reliability and dependability while delivering a level of performance that satisfies its client banks' customers. The ability to process banking transactions quickly and reliably is a critical success factor for banks. "Fiducia must provide the highest level of performance and availability possible, and IMS helps us to satisfy our service-level agreements," said one system programmer responsible for IMS.

Where Telephone Services Are Involved, Failure Is Not an Option

For Telcordia, a major provider of data services to telecommunications companies, providing a guaranteed response to events in the telephone network is a critical requirement. This service provider installs and supports the core data systems that drive tier 1 telephone service providers around the world. Its annual revenue is $700 million, and it has customers in the United States, Western Europe, India, and Brazil. Its system is based on IMS. These data services each execute millions of transactions per day with subsecond response times. They use Parallel Sysplex configurations to ensure high availability. Some of the larger systems average 5 million transactions per day, supporting an average of 250 gigabytes of active data. "Our data systems are designed to provide consistently high throughput all the time," said Telcordia's Paul Gandolfo, a senior technical specialist in application development and performance. "Our customers bet their businesses on the services we provide, and we bet our service performance and reliability on IMS."

But it's not the size or the throughput that matters so much here; rather, it's the reliability of the system. We all understand that availability is at the core of the service a telephone company provides. What should also be understood is that in order for telephone companies to deliver that fault-tolerant, continuously available telephone service while still providing the kinds of user services and custom account management that customers have come to expect, the underlying business transaction system must be just as dependable, and just as efficient, as the switching system. Telephone companies transact business in real time. Downtime is not an option, delays in processing are not an option, and errors are not an option. Telcordia relies on IMS to deliver consistent performance and rock solid reliability.

When Processing Debit Card Transactions, Seconds Count

One of North America's largest retail banking operations handles over 40 million retail banking transactions per day, including drive-through banking, ATMs, debit card purchases, online applications, and interactions with other institutions around the world, meeting service levels of a few seconds. The workload is handled by IMS deployed in a Parallel Sysplex configuration of 12 IMS systems, receiving card transactions through a front-end CICS system. These are the elements of the environment for two key banking applications: one handles the online transaction processing, and the other operates in the background maintaining and balancing the accounts. The former application uses IMS Fast Path, while the latter is a full-function High Availability Large Database (HALDB) implementation. The total amount of data managed is in the 40 terabyte range, with plenty of room for expansion. The system is continuously available, according to the head of IMS administration. "We plan for four outages per year per plex, but we don't usually need them all. Each lasts a few hours or less. Since we perform them in a rolling fashion, there is no actual outage from the user's perspective."

When the bank acquired another large North American bank, it needed to expand to handle the extra workload. So, it went from a six- to an eight-LPAR configuration, which leaves plenty of room for growth. The bank has tended to be conservative, allocating more resources than needed to ensure that it meets its service-level agreements. A staff of 10 manages the databases and applications involved. The bank had considered a move toward distributed systems, but there was no configuration it could come up with that could handle the volume of processing with the reliability that IMS delivers and that could be managed altogether by just 10 people. Instead, if it acquires a bank with a distributed systems operation, it will migrate the distributed systems and merge them into its IMS-based system.

For this bank, the ability to expand operations while maintaining a high degree of dependability is a critical success factor now and moving forward. It has determined that the best platform for realizing those benefits is IMS.

FUTURE OUTLOOK

It is often argued that mainframe technology in general and IMS in particular are "old technology" and that they cannot move to meet the changing demands of the times. This argument has actually been made since the emergence of so-called "open systems" in the early 1990s, yet the mainframe in fact has continued to adapt to changing times, requirements, and circumstances, and so has IMS. Far from being headed for the trash heap of history, mainframe computers and IMS are actually well-poised to meet the next generation of computing demands, which will be based on concentration of functionality, an expectation of continuous availability, virtualized deployment, and scalability; these are especially relevant in the emerging area of cloud computing.

CHALLENGES/OPPORTUNITIES

IMS will continue to be challenged by competing technologies, including the current leading RDBMS products and new database management technologies that are still emerging. There can be no question that these technologies can address the needs of certain workloads better than IMS. As IMS continues to evolve, however, in all probability it will be able to maintain a strong claim on the highly complex data management and high throughput workloads that it has historically served so well.

CONCLUSION

IMS is not "old technology," but it is time-tested technology. Its platform, the mainframe, is far from obsolete and in many ways is out in front regarding virtualization and consolidation capabilities that new technologies are still catching up to. For the vast majority of IMS users, the emphasis should not be on "How can I get off IMS?" Rather, the emphasis should be on "How can I expand my use of IMS to take maximum advantage of its potential?"

CASE STUDY

GAD eG is a company that provides IT services for approximately 500 commercial, cooperative, and retail banks in German-speaking areas. It offers 24 x 7 banking services including datacenter and online services. Its principal service package is called bank21, which offers integrated business processes for all manner of banking transactions and data management. Founded in 1963, the company has 365 million € in annual revenue and 1,518 employees.

bank21 is a comprehensive set of data management and consultancy services. The underlying IT system is a distributed application with central and local components that communicate over an intranet; the central component is application server based, while the local applications run as client deployments running Java Web Start and browser-based applications.

The core of the central system is a set of mainframe computers that communicate with local systems and with online users, ATMs, etc., through Web services hosted by WebSphere 6.0 for z/OS. The data workhorse is a collection of 30 IMS control regions running in 16 LPARs on 8 computers under Parallel Sysplex. They process about 50 million transactions per day on an average day, but that number goes up to 60 million near the end of each quarter. About 60% of the transactions come from banks and banking machines, while online banking users account for the other 40%. Data services are provided through IMS Transaction Manager, and connectivity is provided through IMS Connect.

The IMS databases total about 20 terabytes. Using IBM DataPropagator, subsets of the data are streamed to DB2 databases for query and reporting. Four people manage the IMS databases, with an additional five DBAs performing database design work as part of the development team. The IMS systems are never taken down; because maintenance is done on a revolving basis with the Parallel Sysplex environment, it is never necessary to interrupt service. Last year, GAD reported a total of five unplanned outages, each lasting less than half an hour, and none of which impacted customer service.

As of 2007, this system managed 60,000 bank branches and 16,000 ATMs. The data represented 29.47 million bank accounts containing 2.06 billion bookkeeping items. Because of the highly centrally organized nature of the mainframe, these systems are easily managed with a small team. Features provided by IMS Transaction Manager and connectivity through WebSphere Application Server and application bridges enable them to interoperate in a transparent and fluid way with distributed computers, ATMs, and Web browsers.

GAD remains committed to IMS because IMS has enabled it to deliver the levels of performance and reliability that its client banks require. The company expects that by sticking with IMS as its platform, it will be able to do so in the future. The system appears to the user to be fully modern in every way because it is; it's IMS.

Copyright Notice