DATAKINETICS
Z PERFORMANCE & OPTIMIZATION

45 YEARS
SERVING THE GLOBAL FORTUNE 500

PURE HIGH-PERFORMANCE IN-MEMORY TECHNOLOGY FOR THE MAINFRAME

Powering the World's Largest Businesses

A DATAKINETICS WHITE PAPER

Table of Contents

Maximizing mainframe performance 3
Background: The challenges facing legacy systems 3
Enterprise data: The 80/20 rule 3
What is Reference Data 4
Optimizing reference data access using DataKinetics in-memory technology 5
tableBASE in-memory table performance 5
Benefits of using tableBASE in-memory technology 5
tableBASE performance improvement techniques 6
I/O elimination 6
But what about buffering? 7
The code-path shortcut 7
tableBASE test results 8
Tests results on customer systems 8
Tests results third-party test systems 9
Tests results over time 9
Conclusions 10

Maximizing mainframe performance

Maximizing mainframe system performance is an ongoing process, and amounts to a full-time job both for IT organizations running mainframe systems and for IBM. The z/OS operating system and the venerable Db2 database form the backbone of the most important IT assets possessed by almost all of the biggest banks, financial institutions, insurers, government departments and retailers in the world. These systems are not going anywhere—they are not being replaced, and they are not being phased out. However, they are continually being improved upon. IBM periodically releases new versions of z/OS and Db2 that result in better performance. Most of its biggest customers also invest in best-in-class in-memory technology—tableBASE®—which augments mostof the improvements that IBM makes.

Background: The challenges facing legacy systems

Over several decades, businesses have evolved from having predictable and stable data access requirements typically from strictly internal sources, into perpetual-operation enterprises with data access requests coming from businesses, business partners, and customers 24-hours-a-day, 7-days-a-week, and 365-days-a-year.

Originally, data access was internal and well controlled, but over the decades, data started to be accessed first by business partners via modem access or dedicated data lines, and then by customers via the internet, and now by mobile devices. Data access increased dramatically; it was not well controlled, nor was it completely predictable. At the same time, data volume increased dramatically.

Despite the enormous advancement in IT technology and the increased I/O response times now possible, resulting in lightning-fast transaction times, IT systems today still have great difficulty keeping up with the demand to access data. IT organizations are struggling more than ever to deal with the ever-increasing volumes of data that they are required to process.

IT has made advances to help cope with ever-increasing access demand and volume, by implementing data caching, buffering, and more. And, while in many cases this is sufficient, high transaction-throughput environments still have difficulty keeping up. The struggle continues. Businesses must find and use the fastest way to access data, and to apply those solutions to the data that they access most often.

Enterprise data: The 80/20 rule

An organization's data is critical to the business of the organization, often it is the most valuable part of the organization, and it is used every day by the organization: Banks use their customer account data every second of every day; insurance companies use their policy data for every claim; and retailers use their pricing data every day, in every transaction. But not all data is the same—some data is accessed more often than other data. A general 80/20 rule can be applied to enterprise data: 20% of all data is used most of the time, while 80% of it is used rarely. This is true for most large IT organizations, but for some businesses—typical for banking—as little as 5% of the data results in 80% of all accesses to the organization's databases. This type of data is known as reference data.

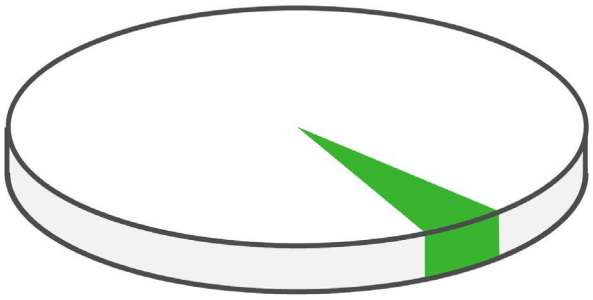

Figure 1: A small amount of data accounts for the most access to the enterprise database

What is Reference Data

What type of data are we talking about? To answer this, we borrow from Malcolm Chisholm's work (www.refdataportal.com), and describe enterprise data using the model in Figure 2.

Almost all of an enterprise's data can be described as either Master Data or Transaction Data. Master Data is virtually all of the organization's data that is not directly related to transactions. It includes reference data and structure data. Transaction Data is strictly transaction-related data—data that is related to maintenance and updates to master data, or data about a transaction—it includes temporary data, transaction data and audit data.

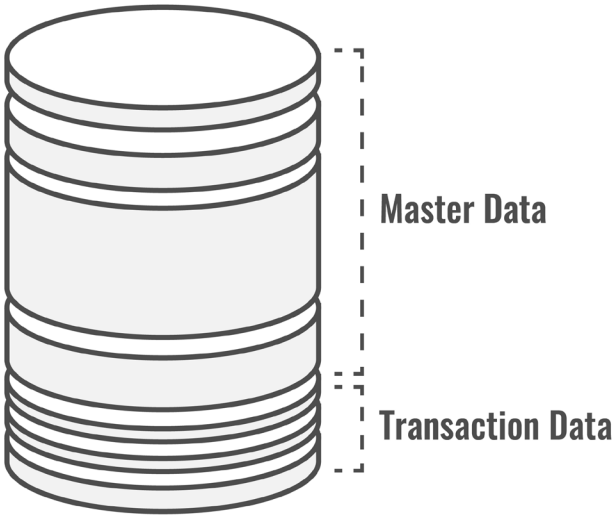

Figure 2: Master Data and Transaction Data

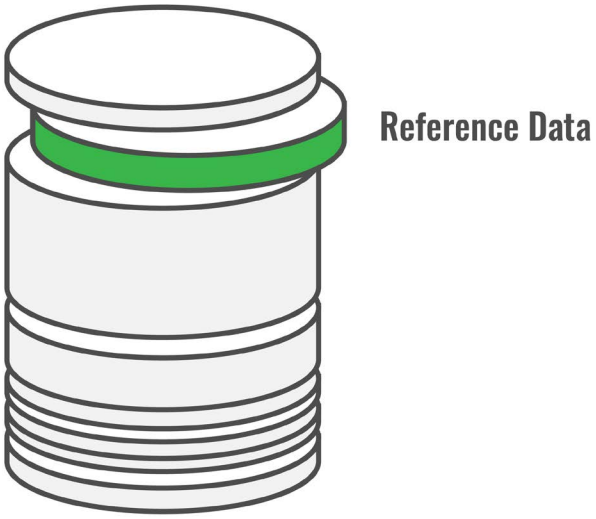

Figure 3: Reference Data

The part of the data that accounts for 80% of the accesses to the organization's database comprises a subset of the master data, Reference Data (see Figure 3). Reference Data rarely changes, and is accessed most often—examples are codes for account status and account types, customer status and customer types, and bad credit card lists.

For organizations with transaction-intensive mainframe environments, reference data is associated with the largest portion of resource usage. They are responsible for 4x more I/O and associated CPU cycles than all other types of data combined. Optimization of this type of data can make a dramatic impact on program run time, general application performance, and can even help reduce mainframe operating costs by reducing I/O and CPU usage.

Optimizing reference data access using DataKinetics in-memory technology

What is best-in-class data access optimization? Unquestionably, it is using tableBASE "pure" high-performance in-memory tables to augment your existing DBMSs and their buffering facilities. The reference data that is used most often by the applications—a very small amount of data—is copied from the DBMS into tableBASE "pure" high-performance in-memory tables, where it is accessed using a simple API. In this way, the reference data that is used most often by your applications is accessed much faster. All other data is accessed in the normal way from the DBMS and its in-memory buffers.

tableBASE in-memory table performance

In-memory tables provide the fastest access to reference data; thereby improving the performance of any application that needs to access large amounts of reference data. A product that provides the fastest access to reference data must have the following attributes: Data must be stored in memory, data must be in a table structure (data in rows, able to be searched using keys, indexes), access must be optimized for speed, and it must have a short code path.

Technique	Data in-memory	Table structure	Optimized for fast access	Shortest code path	Score
VSAM Buffers	YES	NO	NO	NO	1/4
Db2 Buffers	YES	YES	NO	NO	2/4
tableBASE	YES	YES	YES	YES	4/4

Table 1: tableBASE vs other in-memory techniques

Of the various techniques available that provide in-memory access, only tableBASE provides all four advantages.

Benefits of using tableBASE in-memory technology

tableBASE helps to solve a variety of common business IT challenges. Here is a list of some of the most widely used tableBASE solutions:

- Provide capacity and scale to service ever-increasing volume of transactions
- Complement the DBMS by off-loading the repeated access of small amounts of reference data to high-performance in-memory tables
- Reduce cost-per-transaction
- Increase capacity of batch window and meet contractual SLAs
- Reduce overall mainframe MSU/MIPS usage
- Reduce mainframe TCO and IT capital and operational expense

tableBASE performance improvement techniques

Using in-memory tables to access reference data allows for faster data access and improved application performance in several ways, chief among these are I/O elimination and code-path reduction.

I/O elimination

In-memory tables are accessed at memory speed, which is 1000 times faster than I/O speed, and up to 100 times faster than buffered I/O speed. Many current systems source data from databases as shown in Figure 4. Access for all data is at I/O speed, or buffered I/O speed.

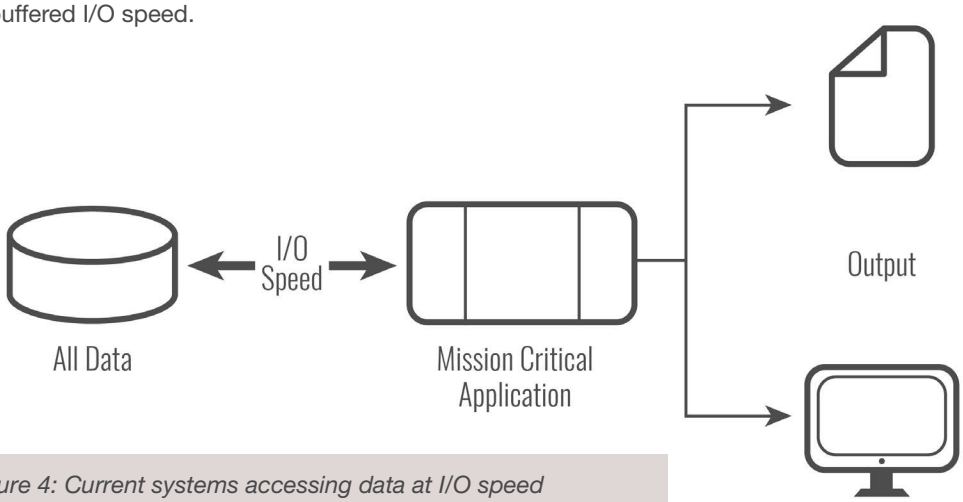

Figure 4: Current systems accessing data at I/O speed

Since only a small amount of an enterprise's data access needs to be optimized, the first step is to identify the data that is accessed most often. You can start small by selecting the data most often accessed by a small number of transaction-intensive applications (look for applications with ratio of reads to writes of at least 50:1 or more). The next step is to copy that data from the database into one or more tableBASE in-memory tables. The result is shown in Figure 5.

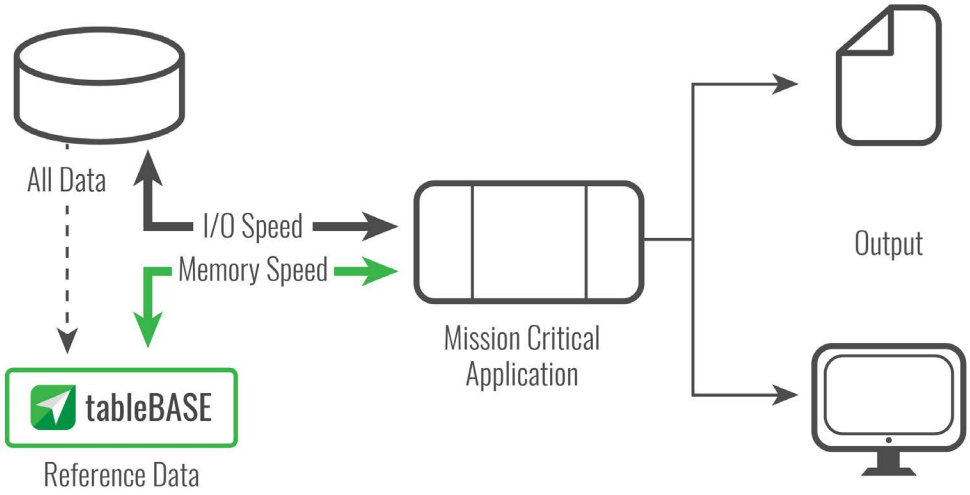

Figure 5: DataKinetics in-memory tables augmenting the database

Applications now access reference data that is used most often from "*pure*" in-memory tables, and continue to access the balance of the data from the database as before. Much of the I/O (and associated CPU cycles and MSU cost) has been eliminated by the use of in-memory tables.

But what about buffering?

While it is true that buffering can eliminate I/O as well, that is where the comparison stops. Buffering, whether Db2 buffering, VSAM buffering, or any other type of buffering, is not optimized for fast access. Buffering is simply the temporary placement of some data into memory buffers to eliminate I/O—data is still accessed from memory-based buffers as if the data were still on disk. This is actually the biggest advantage to using tableBASE—it is optimized for fast memory-speed access.

The code-path shortcut

In-memory tables are accessed using the shortest possible code path—as close as possible to the calling application. Buffered data uses the same code path that is used for accessing data from disk; tableBASE does not. Figure 6 shows an approximation of the Db2 code path taken for accessing data on disk; the code path taken to access data in memory buffers just eliminates the DASD path.

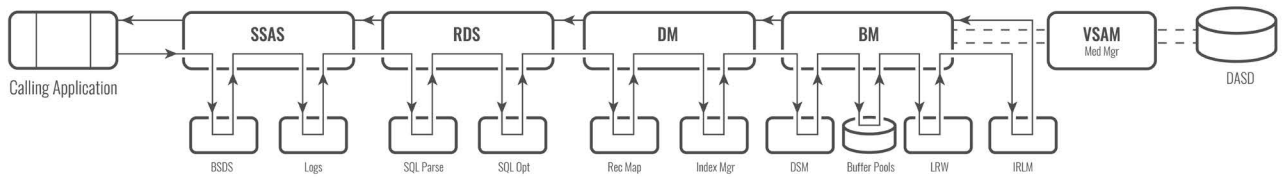

Figure 6: Disk-access code path

For comparison, Figure 7 shows the code path used for accessing data using tableBASE.

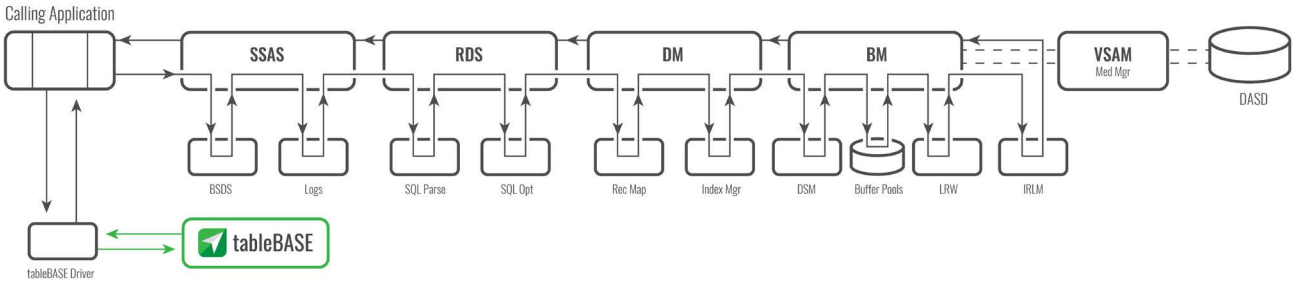

Figure 7: DataKinetics in-memory table-access code path

Note how short the code-path is when using tableBASE to access reference data. All of the Db2 subsystem overhead is avoided—it is not needed for Read-Only reference data access. This code path shortcut is the real difference maker for tableBASE, and is uniquely responsible for making tableBASE the fastest way to get at your reference data.

tableBASE test results

Db2 buffering is a standard technique used by virtually all Db2 shops to improve the performance of Db2 over standard DASD I/O speed. Like any buffering technique, Db2 buffering is sufficient for most applications, but fails to deliver in the most demanding transaction-intensive environments.

Tests and comparisons have been completed by both independent third-party testing organizations, and DataKinetics customer IT organizations.

In all cases, systems augmented using tableBASE allow data to be accessed by applications at a rate considerably superior to any other technique. Here are the highlights of the testing that has taken place.

Tests results on customer systems

Using test data from a customer site, you can see how much of an impact tableBASE can make. DataKinetics tableBASE Db2 optimization (using Db2, Db2 buffers augmented by tableBASE) out-performs just Db2 + Db2 buffer optimization by a wide margin: up to 3000% faster.

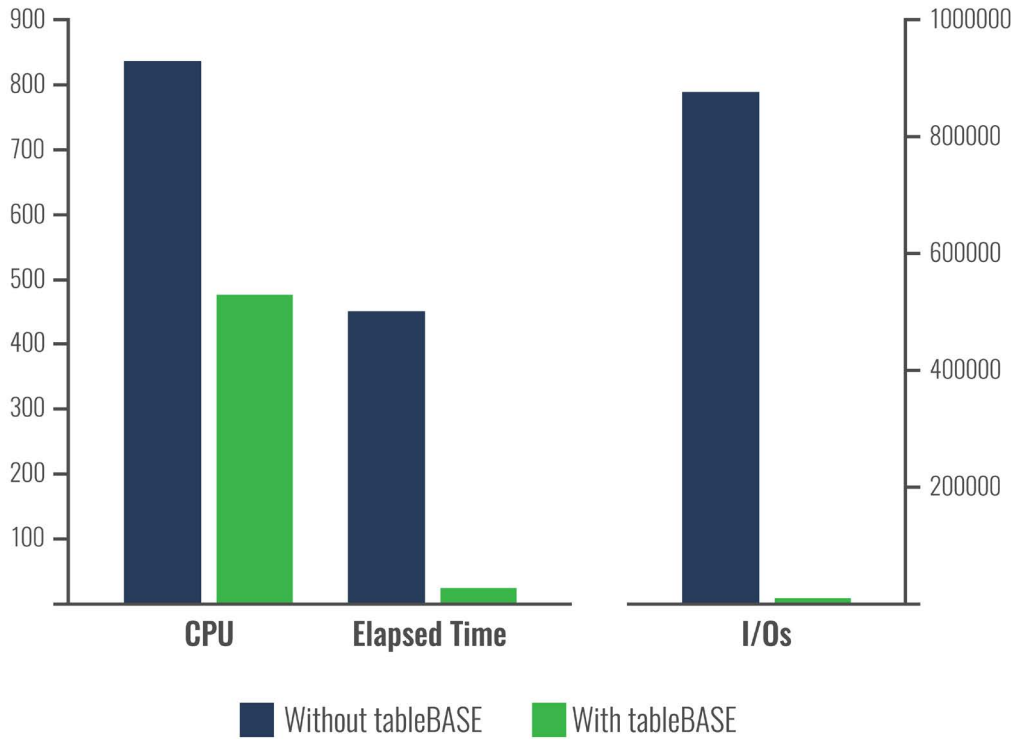

Figure 8: Buffered Db2 versus tableBASE: Customer experience

Tests results third-party test systems

DataKinetics testing of buffered Db2 data access as compared to tableBASE data access shows similar results. A benchmark test case was designed and run at IBM's Poughkeepsie, NY testing facility. The tests were performed by IBM personnel, and to obtain the best possible Db2 results, IBM personnel buffered the entire test Db2 table into memory. The results showed that data could be accessed by tableBASE 30 times faster (3000% faster) than it could be accessed by highly buffered Db2. This is not typical of real-world Db2 usage, but is typical of real-world tableBASE usage. Figure 9 shows the IBM testing—the results are very similar to real customer experiences, shown in Figure 8.

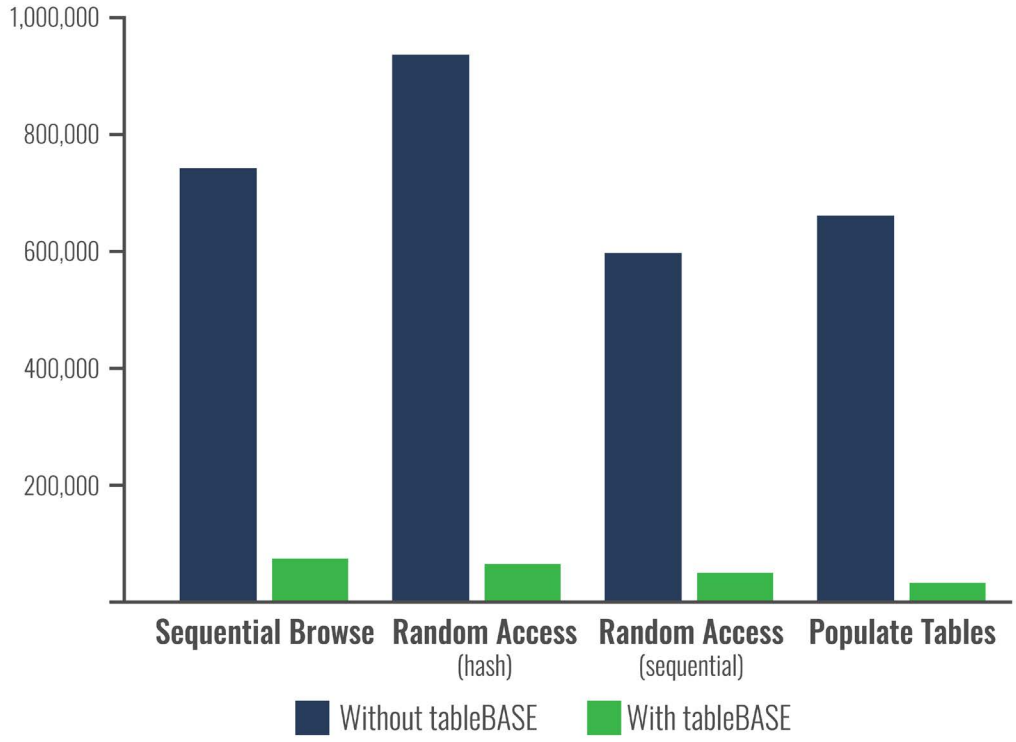

Figure 9: Buffered Db2 versus tableBASE, as tested

Tests results over time

It could be said that test results from a decade ago are no longer valid now because IBM has improved the performance of both their z/OS operating systems and their Db2 database. And that is true: every new version that IBM releases provides some measure of performance improvement.

However, since tableBASE is a z/OS application, any performance improvement brought about by an OS upgrade will also enhance the performance of tableBASE. And while Db2 performance increases are significant, the tableBASE code path length will always provide a means to improve Db2 applications (and database) performance.

> *"A modern data architecture is based on polyglot data persistence, using the right data storage technology for each use case. That can be a relational DBMS like Db2, or a NoSQL DBMS for big data analytics, or tableBASE for reference data requiring high-speed data access."*
>
> **— Craig S. Mullins, Industry Analyst, Mullins Consulting, Inc.**

Conclusions

Businesses have difficulty accessing their data fast enough to satisfy the growing demands on their mainframe systems. In-memory technology can be used to augment the applications and databases of these systems, improving system, application and database performance, while virtually increasing system capacity.

The fastest way to access data is from memory, and since only 5% to 20% of data accounts for most of an organization's data accesses, it makes sense to optimize access to the data that is used most often. DataKinetics tableBASE can be used to help applications access data 3000% faster than is possible using any other technique. It is the solution of choice for in-memory optimization in the mainframe world, and is being used today by 20% of the Fortune 50. tableBASE has never been removed from a system once installed.

The next step

To see how much of an impact these unique cost optimization solutions would have on your business, a "proof of concept" trial can be arranged. The steps of the trial would include identifying a specific problem area, addressing it, measuring the difference, and then assessing the overall cost impact.

Your organization and the Professional Services staff at DataKinetics will collaborate to outline a high-level project plan and approach that will review applicable environments, infrastructure, application code, etc., and to help implement the proof of concept. We will work with you to provide a proposal based on your current IT plan. Contact DataKinetics for more information.