

DataKinetics  
Solutions

## tableBASE proof-of-concept: A quick and painless process

Proof of concept without pain

October 2009

# Contents

- tableBASE proof of concept.....3
- About tableBASE .....3
- Is tableBASE right for you?.....3
- What about the risk? .....4
  - A tableBASE proof-of-concept is NOT risky .....4
- A simple proof-of-concept procedure for tableBASE .....5
- Sample calculations .....6
  - Savings for your test application .....6
  - Savings projected across multiple applications .....7
  - Savings projected toward your MIPS usage and future needs .....8
- The next step .....9
- Conclusion ..... 10
- About DataKinetics..... 10

## tableBASE proof of concept

### About tableBASE

DataKinetics has proven itself a trusted vendor to hundreds of customers, including 20% of the Fortune 50 and some of the world's largest banks, insurance companies, credit-card processors, brokerage houses and retailers. DataKinetics' customers are typically dealing with billions of transactions daily through their critical information systems.

Optimizing the MIPS processed by a mainframe can save companies millions of dollars annually, a benefit that appeals directly to the executive levels of major corporations. DataKinetics reduces its customers' use of MIPS by at least 50% and often by 80% or more. For companies that use 2,500 MIPS/year, that means savings of at least \$1.8-million.

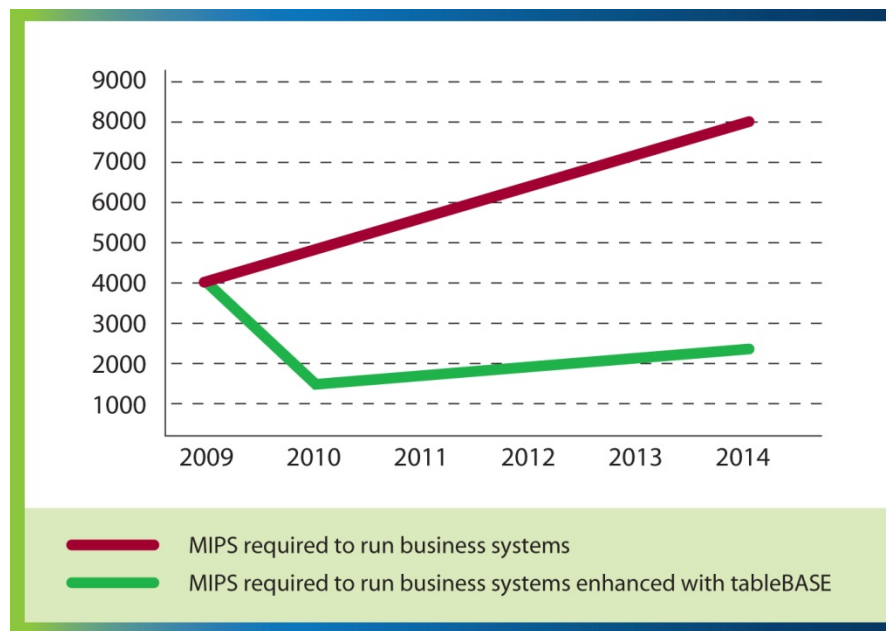


Fig 1: Potential for DataKinetics tableBASE improvement in your mainframe infrastructure

### Is tableBASE right for you?

If your IT organizations are running high-transaction applications, and you're looking to boost throughput, tableBASE is right for you. If you're stalling at either a large hardware upgrade commitment, or a large migration commitment, and you'd like to avoid the high cost, tableBASE is right for you.

## ***What about the risk?***

When an organization has mission-critical applications that represent tremendous value and competitive advantage, they tend to be risk-adverse when it comes to change. Concerns range from how long it would take to just to get through the trial stage, and whether or not it will be a positive outcome, to whether or not they have the resources and expertise to carry out and support modifications to their legacy code.

### **A tableBASE proof-of-concept is NOT risky**

The truth is, changing your applications is neither risky, nor is it difficult— you can run a very quick and easy demo within your testing environment. The benefits and potential of tableBASE to reduce application execution time and resource consumption will be obvious right away. Further, there will be absolutely no changes to the program logic within your application.

Making small changes for the proof-of-concept can take as little as a few hours. At the most, it will take a couple of days to a week of your time. One very small sample change is all that is needed to determine the potential impact of tableBASE.

A tableBASE proof-of-concept exercise is very straightforward and can be performed by your own people within your own testing environment. However, DataKinetics specialists are available to assist free of charge.

## *A simple proof-of-concept procedure for tableBASE*

### Prerequisites:

- Access to application source code within a test environment
- tableBASE trial version installed
- technical personnel

### Procedure:

1. Install tableBASE into your test environment.
2. Get baseline data from your existing application.
  - Run the application in your test environment
  - Record program execution time - both CPU and Elapsed
  - Record I/Os (by DDname preferred)
3. Open the application source code using the TSO editor.
4. Locate a Read Only I/O operation.
  - Often the most frequently accessed table is the best candidate
  - Could be an SQL statement, a VSAM operation, an I/O routine:

```
MOVE L-RANDATA-VALW-CURRENT-
EXEC SQL
  SELECT * INTO :DCLTEST000
  FROM TEST000
  WHERE COL1 = :COL1
END-EXEC
```

- Estimate the total of all Read and Write I/O operations in the application
5. Replace the Read Only I/O with a Call to tableBASE:

```
*****
CALL 'TBLBASE' USING
      W-TB-PARM
      W-TB-COMMAND-AREA
      W-ITEM-AREA
*****
```

6. Copy the selected table into the test tableBASE library.
7. Compile, test and execute the improved application.
8. Compare the application's performance to the baseline established in Step 2.
9. Calculate the potential application improvements, and determine the financial impact of tableBASE.  
(See next page.)

## Sample calculations

These numbers are examples only—your applications will have different numbers—substitute your actual numbers to obtain your projected benefit from the use of tableBASE.

### Savings for your test application

An application (App A) accesses a DB2 database: five times, here are the details:

- Fetch 1 (random) is for 1,000,000 rows of 100-byte data from database table A, Read-Only. (This fetch will be replaced with a single call to tableBASE.)
- Fetch 2 is for 200,000 rows from database table B, Read/Write. (Being R/W, this is not a good candidate for optimization.)
- Fetch 3 (random) is for 100,000 rows from database table C, Read Only. (This fetch is a good candidate for optimization.)
- Fetch 4 (random) is for 400,000 rows from database table D, Read Only. (This fetch is a good candidate for optimization.)
- Fetch 5 (random) is for 800,000 rows from database table E, Read Only. (This fetch is a good candidate for optimization.)
- The application is set up for CPU-second measurement for all fetches and processing.

#### Data known before testing

Original application execution time (t1), measured:	= 262.5 CPU sec
Original 1,000,000-row fetch time (t2), measured:	= 95.5 CPU sec
Usage dedicated to non-I/O processing:	= 22.5 CPU sec

#### Measured test data and calculated data

New time (t3), for Fetch 1, measured, using tableBASE calls	= 4.1 CPU sec.
Savings: (t3/t2 *100)	= 96%
Potential fetch time with all candidate fetches replaced with tableBASE calls:	= 28.7 CPU sec.
Potential APP A execution time (t4):	= 51.2 CPU sec.
Savings: (t4/t1 *100)	= 80%

See Table 1 for more details.

**Table 1: APP A fetch times, before (measured) and after (calculated)**

	DB2 source	Type	Before: I/O time (CPU-sec)	After: Estimated Total (CPU-sec)
Fetch 1	tableA: 1,000,000 rows	R-O	95.50	4.10 (measured)
Fetch 2	tableB: 200,000 rows	R/W	19.50	19.50
Fetch 3	tableC: 100,000 rows	R-O	10.00	0.50 (calculated)
Fetch 4	tableD: 400,000 rows	R-O	40.00	1.60 (calculated)
Fetch 5	tableE: 800,000 rows	R-O	75.00	3.00 (calculated)
total fetch time	-	-	240.00	28.70 (calculated)
non-fetch processing	-	-	22.50	22.50
Total exn time APP A	-	-	262.50	51.20 (calculated)

### Savings projected across multiple applications

Taking the execution time reduction shown above, you can extrapolate what your real savings could be, per year, if you optimized all of your applicable mainframe applications with tableBASE.

If your cost-per-CPU-second is \$0.22,  
and you run the application 200 times daily:

using DB2 calls, the associated cost per day: = \$11,550.00.  
with tableBASE calls, the cost per day = \$ 2,252.80

Assuming you have other applications, and you can reduce execution time for them by an average of better than 75%, your savings over the course of a year could be dramatic, as show in the table below:

**Table 2: tableBASE provides dramatic savings in program execution time and I/O CPU savings**

	CPU-sec DB2	tableBASE	% saved	Times run per day	Savings/day (CPUsec)	Savings /day (\$)	Savings /year (CPUsec)	Savings /year (\$)
App A	262.5	51.2	80.50%	200	42260	\$ 9,297.20	15424900	\$ 3,393,478.00
App B	10	4.5	55.00%	1000	5500	\$ 1,210.00	2007500	\$ 441,650.00
App C	2	1.5	25.00%	725	362.5	\$ 79.75	132312.5	\$ 29,108.75
App D	30	8.5	71.67%	200	4300	\$ 946.00	1569500	\$ 345,290.00
App E	22	10	54.55%	100	1200	\$ 264.00	438000	\$ 96,360.00
App F	33	5	84.85%	50	1400	\$ 308.00	511000	\$ 112,420.00
Total	359.50	80.70	78%	2275.00	55022.50	\$ 12,104.95	20083212.50	\$ 4,418,306.75

## Savings projected toward your MIPS usage and future needs

If you are projecting a hardware upgrade to accommodate increased demand on your mainframe data, tableBASE can change the fiscal spending picture for you.

For example, if your current MIPS usage is 2000 MIPS, and you plan to upgrade to 4000 MIPS next year, optimizing your applications with tableBASE can permit you to postpone or even cancel your upgrade plans. The reason is that the MIPS that were previously allocated to I/O can be recovered, and reallocated to your increased processing requirements.

In our example:

- Your current mainframe costs are \$1,500,000 per year.
- However, you're running out of processing capacity, and you project that you will need to double it next year.
- Your planned upgrade for next year will increase your mainframe costs to \$3,000,000 per year.
- However, if you optimize your applications using tableBASE, you may be able to recover 90% of your MIPS, and push out the upgrade for, say, three years.
- This can save on IT spend, substantially.

The following table compares your planned spend against a post-tableBASE optimization spend:

**Table 3: tableBASE provides dramatic savings in your upgrade planning**

	Year 1 mainframe costs	Year 2 costs	Year 3 costs	Year 4 costs	Year 5 costs	Total 5 years	Savings
<b>Original upgrade plan</b>	\$1,500,000	\$3,000,000	\$3,000,000	\$3,000,000	\$3,000,000	\$ 13,500,000.00	n/a
<b>Upgrade plan with tableBASE</b>	\$1,500,000	\$1,500,000	\$1,500,000	\$1,500,000	\$3,000,000	\$ 9,000,000.00	\$ 4,500,000.00

**Note:** In a best case scenario, you might be able to stave off your planned upgrade indefinitely, which means that your savings over five years could be **\$6,000,000**.

## *The next step*

If your calculations show that DataKinetics tableBASE will benefit your business, now is the time to get buy-in from others within your organization. The most persuasive argument will be a comparison between your current cost projections and potential cost savings with tableBASE, using the calculations on the previous page.

Once the sales process, has started, DataKinetics tableBASE experts can help you complete the optimization of your mainframe applications:

- You can begin by replacing all of the Read-Only I/O operations within the sample application.
- After testing, the first application will be ready for deployment into your production environment.
- Then, DataKinetics can help you to optimize other candidate applications.
- Beyond optimization projects like this, DataKinetics can also help to optimize your applications in other ways, through table driven programming. Contact DataKinetics for further information on table driven programming.

## ***Conclusion***

Contrary to some opinions, altering your legacy mainframe application code is not a risky endeavor, nor is it expensive or time-consuming. In short order, and at little cost, you can complete a proof-of-concept exercise that will show you what DataKinetics tableBASE can do to benefit your business. Savings can be considerable, and have been for our client base, which includes 20% of the Fortune 50.

## ***About DataKinetics***

DataKinetics®, the leader in critical information systems optimization, enables data centers within IBM System z Mainframe, Linux and distributed environments to achieve superior performance gains, capacity, and scale using existing applications and hardware. DataKinetics delivers distinct competitive advantage to global Fortune 500 companies to keep them a step ahead of the competition. For 30 years DataKinetics products have been enhancing hundreds of thousands of mission-critical transactions per second every day for banking, insurance, credit card, brokerage house, and retail industries—without fail. Visit [www.dkl.com](http://www.dkl.com) for more information.

*DataKinetics*  
202-2460 Lancaster Rd.  
Ottawa, Ontario, Canada  
K1B 4S5

Phone: +1-800-267-0730  
Phone: +1-613-523-5500  
Fax: +1-613-523-5533  
Email: [info@dkl.com](mailto:info@dkl.com)  
URL: [www.dkl.com](http://www.dkl.com)

© DataKinetics Ltd., 2009. All rights reserved. No part of this publication may be reproduced without the express written permission of DataKinetics Ltd. DataKinetics and tableBASE are registered trademarks of DataKinetics Ltd. All other trademarks, registered trademarks, product names, and company names and/or logos cited herein, if any, are the property of their respective holders.