DATAKINETICS
Z PERFORMANCE & OPTIMIZATION

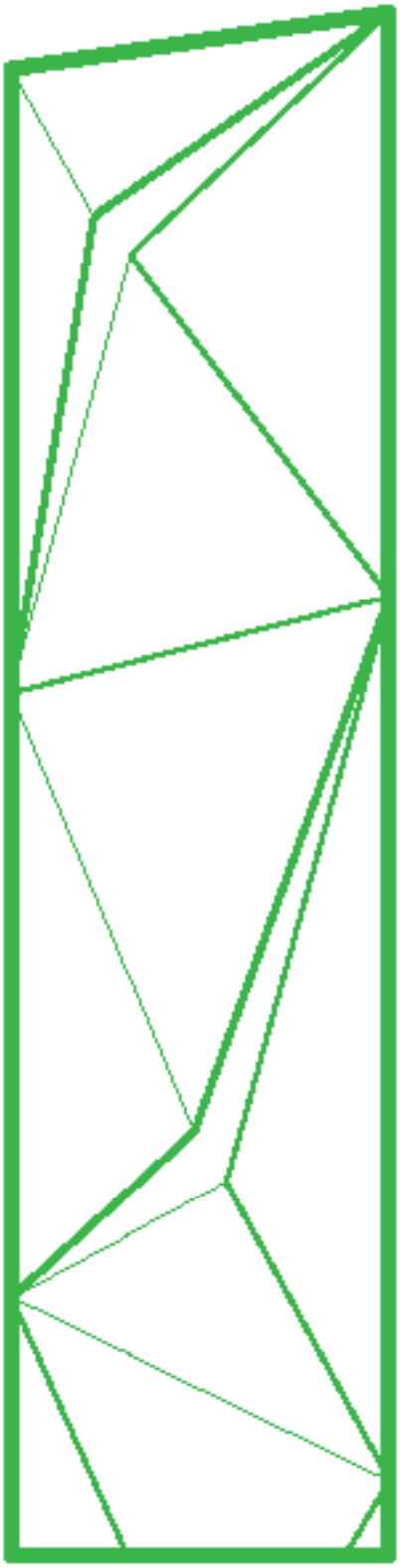
SERVING THE GLOBAL FORTUNE 500 Since 1977

# BATCH OPTIMIZATION FOR MAINFRAME DATACENTERS

A DATAKINETICS WHITEPAPER

# Table of Contents

# Batch Optimization for Mainframe Datacenters



## Mainframe Batch Processing Today

Batch processing is used for some of an organization's most critical business operations, including credit card settlement and reconciliation, daily consolidation of business transactions, processing files with large amounts of data from business partners, and more. It is also used across industries for database maintenance, bulk database updates, ETL for populating data warehouses, running analytics on data warehouses, creating backups, archiving of historical data, and so on. These all are very time sensitive and key processes in business operations.

In essence, batch jobs perform many read, write, and sort activities on sequential files. There is no manual intervention or input required unless, of course, a job does not end successfully. The job should run automatically once it begins and continue until it is completed. Batch window processing often takes place during off-hours or non-peak hours, and it is not unusual for OLTP to wait on the completion of the batch processing, as it requires files to be updated or tables to be current. Again, timing of execution and completion within a predefined time window (or windows) is critical.

## Pressures on the Batch Window

In today's connected world, demands for 24/7 OLTP is universal. Global business has removed any restrictions on time; a company that does business globally must be available around the clock to handle all time zones equally. Mobile and e-commerce have also had a major impact on business operations. They, too, require that businesses have operations available 24/7 to meet the demands of global customers. Banks must also be responsive at all hours for transaction processing, which has put tremendous pressure on batch window processing.

Other pressures on the batch window come from the need to handle larger volumes of data, and the need to incorporate additional functions. Naturally with these changes, the processing time required to complete the batch jobs increases, sometimes exceeding the available batch windows; often leading to extreme congestion in the batch window.

## Failure to complete batch processing on time

Batch pressure can adversely impact the company's ability to deliver value (timely results) and even cause changes in the company's business model. To complicate matters, there can be statutory limitations associated with completion of these activities, such as crediting interest to customers, producing paychecks for payees, and generating payments to business partners.

Additionally, there can be financial penalties associated with Service Level Agreements (SLAs) not being met. Therefore, it is vital that batch window processing take place as efficiently and as quickly as possible for companies to maintain their operations and fulfill their commitments to their customers, employees, business partners, and to meet their legal obligations. Some estimates show that up to 50% of workloads are batch.

## Contemporary Solutions for Batch Woes

There are several options in the marketplace that have been used to solve batch window congestion problems. The best place to start is the *IBM Redbook* on *Batch Modernization on z/OS*. Most of these solutions work quite well, and mainframe shops running significant amounts of batch should be implementing many of them now. Implementing the best of these solutions, or even all of them, may or may not be enough to achieve your batch goals, whether they be performance or cost related. However, there are alternative contemporary options —

**Scheduling**: Scheduling solutions require scheduling software, ongoing monitoring efforts, and are of moderate complexity. They require constant vigilance and frequent intervention by the operators. Effectiveness decreases over time. This solution will be costly, and may or may not be effective in reaching your specific batch goals.

**Hardware Upgrades**: Adding additional hardware equates to more MIPS. This option can be expensive but low in complexity because no application code changes are needed, no changes to the database are needed, and no changes are needed to existing monitoring and optimization processes. This solution will be also costly, and also may or may not be effective in reaching your specific batch goals.

**Grid Workflow**: Grid workflows require a detailed understanding of the interdependencies of workflows among the batch processes, and the ability to parcel out these processes to the different resources that make up the grid. This alternative necessitates hardware, software, and code changes and still requires ongoing monitoring efforts. It can be of medium to high complexity to implement and changes as the business evolves.

**Application Re-architecture and optimization**: Application re-architecture is excessively time-consuming, requiring extensive code changes with the associated complexity and cost. Alternately, application optimization could have fewer code changes, but sometimes changing existing code can be more complex than rewriting it. Depending on how complete the knowledge is of the application and the other programs with which it interacts, this can be a high risk alternative.

**Db2 optimization**: This can take significant elapsed time to implement, particularly when it is difficult to identify what needs to be optimized. Ongoing optimization would still be required as the Db2 queries change over time and with the implementation of new requests for data. Often specialized software tools are required to pursue this option, and in the end, it may not make a difference in reaching your batch goals.

**Run Batch and OLTP Concurrently**: Running batch and OLTP concurrently removes the concern about a batch job not completing within a window of time, but this option typically reduces the performance of OLTP and negatively impacts client experiences. Transaction response time is often slower because batch processes are running at the same time. There are challenges in accessing the same data concurrently. Lockouts would keep data from being accessed by either of the two different processes at the same time or data may not be current if another process using it. The result is deterioration in the company's customer's experience.

**Data in Memory (DIM)**: One of IBM's recommendations from the *IBM Redbook* on *Batch Modernization on z/OS*, is to use DIM to reduce repeated I/O calls of the same data. According to the Redbook, "It is recommended to read such data only once from the database and cache it somewhere for further processing. This will prevent your system from running unnecessary round trips to the database." IBM's DIM solution requires spare memory and spare CPU capacity, additional software, and ongoing monitoring and optimization. The inherent code changes are typically extensive, so as a result, this solution is highly complex and expensive to implement. In IBM's own words, "Implementing data in memory techniques is a complex task."

Despite all of these contemporary solutions, there is only so much improvement possible, and often, when systems and applications change, these efforts have to be repeated. Fortunately, there are a handful of third-party batch optimization solutions that have been helping IT organizations reach their batch goals for years – and for the most part, they are "fire-and-forget" - their impact is long-term. And most of them improve performance and lower operational cost at the same time.

## Modern Mainframe Batch Performance and Cost Optimization Techniques

These batch performance and optimization solutions are proven techniques used by the Fortune Global 500 today, and are helping to power high-intensity transaction processing without the need for additional hardware, memory, or CPU. They do not require ongoing monitoring and optimization of the processes. They include:

- High-performance in-memory technology
- IT business intelligence
- Soft capping automation

### In-memory optimization of batch applications

High-performance mainframe in-memory technology can be used to accelerate your existing batch applications – particularly those in environments experiencing ultra-high transaction processing rates. It augments the database, as well as existing contemporary batch solutions, like data buffering.

This technology works by allowing select data to be accessed using a much shorter code path than most data. The typical Db2 code path requires 10,000 to 100,000 machine cycles – and that includes any type of buffered access. Figure 1 shows this code path (top).

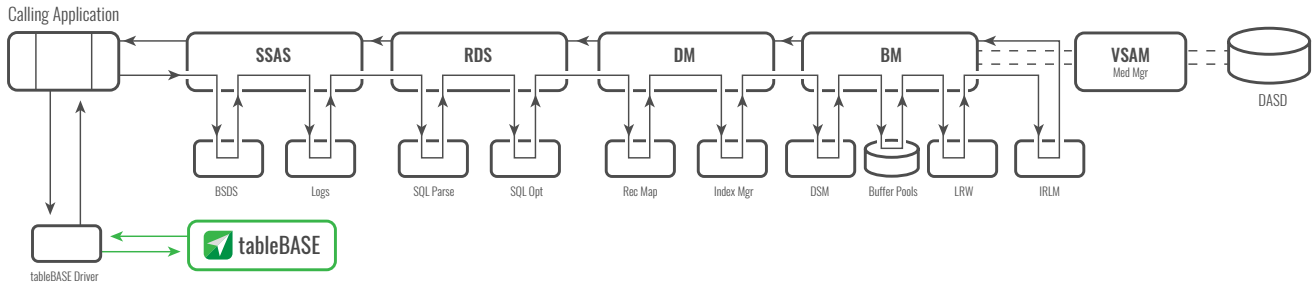



*Figure 1: Different code path lengths*

Data accessed using high-performance in-memory technology uses only 400 machine cycles – Figure 1 also shows this code path (bottom). Only a small portion of your read-only data – the data accessed most often – needs to be accessed in this way.

How does that work? Small amounts of data that gets accessed for most or all transactions – account numbers, interest rates, etc., are copied into high-performance in-memory tables. From there, it is accessed via a small, tight API. All other data access is unchanged. No changes are required to application logic or to the database.

Using this technique, it is possible to not only sharply reduce batch I/O, but more importantly, it can significantly reduce elapsed time – which can solve a batch window congestion problem, It can also reduce CPU usage – which translates directly to reduced MSU usage and therefore reduced operational costs associated with any affected application. (See Figure 2).

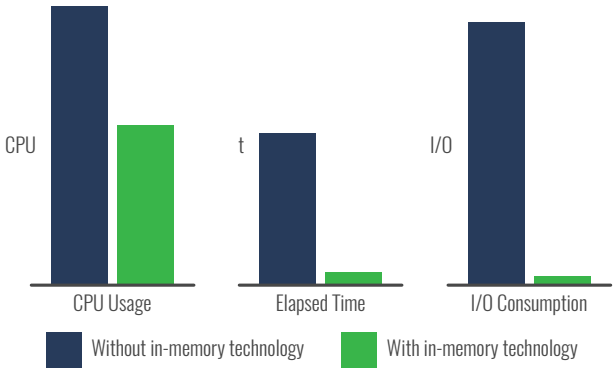



*Figure 2: Customer Results – Reduction in CPU Usage, elapsed time, I/O*

## IT business intelligence

IT organizations collect tremendous amounts of data about their own computing resources every day – both mainframe, midrange servers locally, or in third-party datacenters. So much data is collected, that you could call it their own "IT Big Data." And with the right toolsets, this IT data can be used to reduce the cost of batch running on your mainframe, and can help identify low-priority batch candidates to offload for running on other platforms.

IT business intelligence identifies lower-priority batch workloads that are potential candidates for reprioritization, re-platforming or even elimination. This can directly contribute to improved performance, especially during peak and mission-critical workloads (see Figure 3).

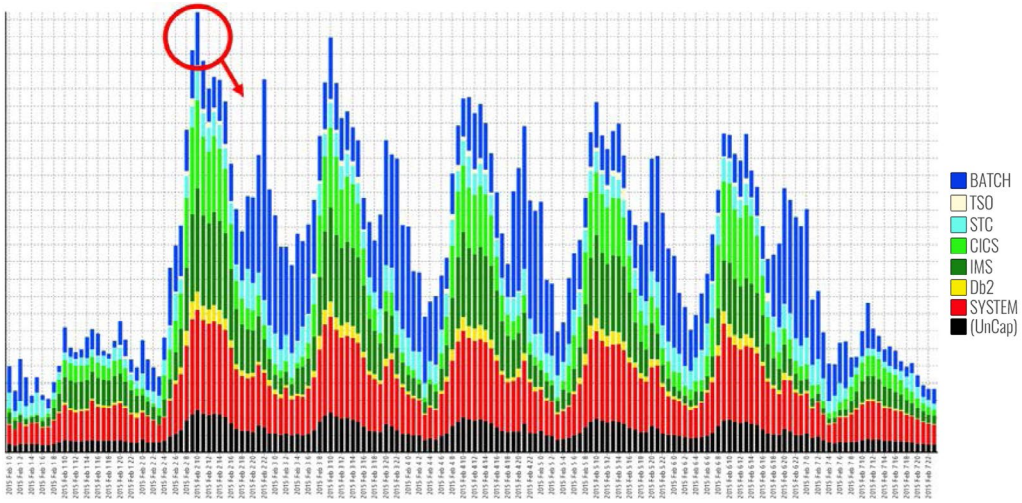



*Figure 3: A low-priority batch workload contributes to the peak workload of the week*

IT business intelligence can also show which departments are using mainframe resources, and how much that is costing. This information can further help to re-prioritize batch processing based on the new-found transparency of departmental spending patterns.

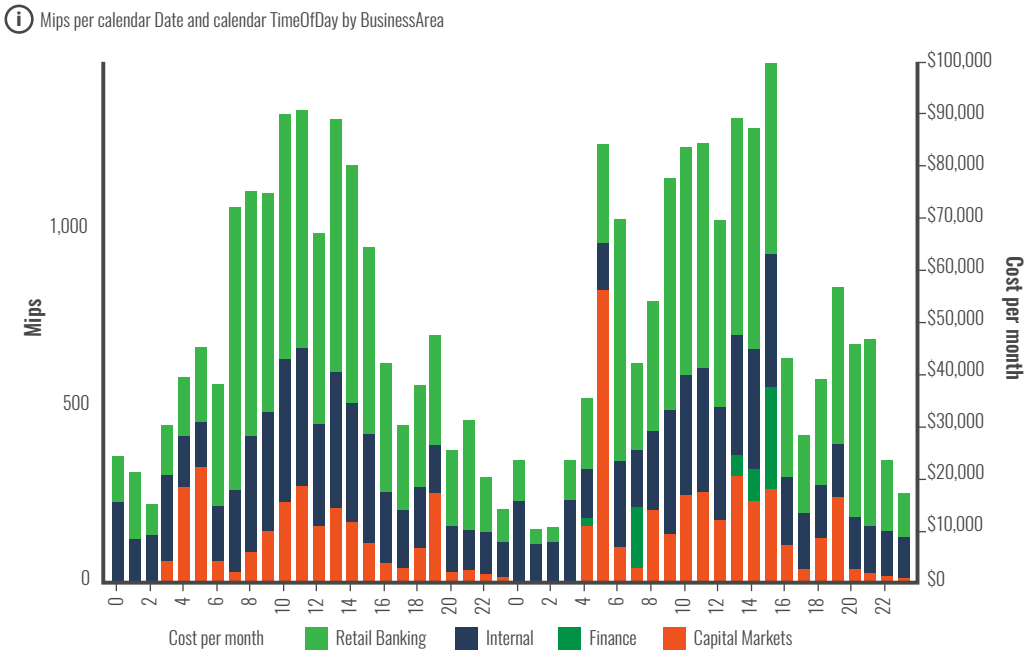



*Figure 4: Business information on mainframe resource usage per organizational unit*

# Conclusion

Third-party batch optimization solutions can help large IT organizations running mission-critical batch processing to reduce their execution times from two times to two orders of magnitude, depending on their business type and the specific characteristics of their batch applications. Each solution shown in this paper can make a significant difference by itself – together, they can make a large dent in ongoing batch processing costs, and most will help reduce batch run times.

## The next step

To see how much of an impact these unique batch optimization solutions would have on your business, a "proof of concept" trial can be arranged. The steps of the trial would include identifying potential problem areas for practical tests, applying well-targeted solution(s), providing test data to demonstrate solution impact, and then estimating an overall feasibility and potential production impact.

DataKinetics Professional Services staff will work with you to outline a high-level project plan and approach that will review existing application code, environments, and to help implement the proof of concept. This consists of migrating sample programs and data for the proof of concept, which will demonstrate the benefits. We will work with you to provide a proposal based on your current IT plan.