

# Why is Optimization Important?

This section provides information regarding the importance of using optimization facilities as provided by Adabas Fastpath in today's data processing environments:

- Increasing Database Management System (DBMS) Workloads
  - Increasing Application Processing Overhead
  - Conventional Optimization Solutions
- 

## Increasing Database Management System (DBMS) Workloads

A DBMS adds significant value to a computer. Modern systems are constantly increasing the DBMS workload and the trend shows no sign of reversing itself.

The increased amount of data to be managed by the DBMS is accompanied by the need for

- more processing power; and
- improved system responsiveness.

A more efficient DBMS goes a long way toward controlling the costs associated with these increasing requirements.

## Increasing Application Processing Overhead

A DBMS makes it possible for systems designers to create highly complex computer systems more easily than in the past. Consequently, more and more data is stored in the DBMS.

But there is a cost.

The application system client and the DBMS server reside in separate processes within the operating system. To satisfy a DBMS query, the following must occur:

1. The client prepares the query
2. The client issues the query
3. The query is shipped to the server
4. The server processes the query and prepares the response
5. The response is shipped back to the client
6. The client receives the result

Shipping data from one process to another within an operating system is resource intensive. Experience shows that benefit is gained by minimizing this overhead.

## Conventional Optimization Solutions

As systems designers have recognized the rising costs associated with increased DBMS usage, some have chosen to make highly used (popular) data permanently available to the client process in the form of memory-based tables, thus avoiding some part of the DBMS activity.

However, this solution has some basic problems and associated costs:

- the application system must have special logic to use these tables; and
- the data is difficult to change (as it is no longer in the DBMS).

In an attempt to resolve the difficulties of tables in memory, other solutions preload data into memory from the DBMS. However, data copied into the client process must be synchronized with the DBMS copy to ensure that clients operate only from copies that are current. Otherwise, serious data integrity problems may be created.

To avoid the need for a synchronization mechanism, some systems preload static (unchanging) data only. Others simply ignore the possible consequences. These approaches provide limited benefit, at best.

Furthermore, the cost of adding complexity to application systems for such limited gain is often overlooked. It can cause a major degradation in the organization's responsiveness to commercial change that may only come to light at the most inopportune time.