

# Set Processing vs. Record Processing Performance

## Dynamic Data Restructuring vs. Prestructured Data Storage

D L Childs

<iis@umich.edu>

### 1. INTRODUCTION

Record processing systems and set processing systems are very different. This paper attempts to clarify the major differences and demonstrate the performance advantages of set processing.

- Record processing accesses physically identifiable *records*. Data is pre-organized for *indexed-record* access.

- Set processing accesses mathematically identifiable *sets*. Data is dynamically re-organized for *relevant-set* access.

Record processing systems require applications to know how available data is physically stored.

Set processing systems only require applications to know how available data is mathematically identified.

### 2. DYNAMIC DATA RESTRUCTURING

A simple fact of data processing is that not all applications want to access the same data, in the same form, for the same purpose. This can present a design and development problem if the data of interest must be pre-organized in storage to provide access for multiple applications. What may be a near optimal organization for one, might be a near disaster for another.

#### 2.1 Data Loading

Before any application can access data, the data must be *loaded*. Record processing systems need to define and build *indexed-record* structures. Set processing systems need to define and implement *set processing* operations.

No matter which data processing system is chosen, significant planning and development effort are required.

Physically based indexed-record structures require new data organizations be built for every new collection of data.

Mathematically based set processing operations allow existing operations to be used for any new collection of data.

#### 2.2 RDS: Relevant Data Sets

After a data loading process the performance of a system depends on how quickly applications can access and process relevant data.

Record processing systems build generally relevant data access considerations into predefined data organizations,

Set processing systems only need provide mathematical recognition of available data. No requirements are imposed by set processing strategies for requiring specific knowledge of application needs for relevant data.

However, it is generally recognized that information used for choosing indexed-record structures can also be used to partition load data for initial application needs.

Though it is well known that data access strategies are challenging to develop<sup>1</sup>, it may not be intuitively obvious that set processing strategies are any less challenging nor even that they provide any improved performance.

### 2.3 Dynamically Derivable Data

Dynamically derivable relevant data sets are what make the performance difference.

As applications spin through execution, relevant data becomes increasingly sparse. Application execution can accelerate if only the remaining relevant data is presented for access. Prestructured data organizations can not adapt, but set processing operations do *automatically*.

The result of every set operation is a relevant data set that did not exist at the beginning of application's execution. Feeding this resultant set to subsequent operations automatically reduces the amount of non-relevant data that has to be processed. (see [1] appendix D for an SQL execution).

The only design decision that needs to be made is whether to preserve these new sets for future applications or to delete them at the termination of the execution.

### 3. CONCLUSION

Set processing systems have been in commercial use since 1972.<sup>[2]</sup> Comparisons with commercial systems confirms orders of magnitude performance improvement.<sup>[3]</sup> Though record processing has proven successful for over thirty years, it may be time to consider the set processing alternative<sup>2</sup>.

### 4. REFERENCES

- [1] Childs: *Set Processing at the I/O Level* <http://xsp.xegegis.org/Spio.pdf>
- [2] MICRO RDBMS (1972-1998) [http://en.wikipedia.org/wiki/Micro\\_DBMS](http://en.wikipedia.org/wiki/Micro_DBMS)
- [3] Stout: *Information Access Accelerator*, IBI 2005 [http://xsp.xegegis.org/Iaa\\_ibi.ppt](http://xsp.xegegis.org/Iaa_ibi.ppt)
- [4] Larsen: *The Business Value of Intelligent Data Access* [http://www.ca.com/files/whitepapers/data-access-white-paper-final-030909\\_201548.pdf](http://www.ca.com/files/whitepapers/data-access-white-paper-final-030909_201548.pdf)
- [5] Stonebraker: *The End of an Architectural Era* <http://www.ahzf.de/itstuff/papers/vldb07hstore.pdf>
- [6] **XSP Technology**: (webpage), <http://xsp.xegegis.org>

<sup>1</sup>For a wide range of reasons, designing for and maintaining optimal data access poses a genuine challenge to even the most sophisticated enterprises. [4]

<sup>2</sup>The DBMS vendors (and the research community) should start with a clean sheet of paper and design systems for tomorrow's requirements, not continue to push code lines and architectures designed for yesterday's needs. [5]