MTB.481

To: Distribution

From: Matt Pierret, on behalf of Data Management Task Group

Date: 12/17/86

Subject: A framework of goals toward more complete data management capabilites on Multics.

INTRODUCTION

This document will set forth the goals of the data management effort in Multics. Goals will be presented in phases, somewhat logically dividing the work we will set out to do. In the following sections, we will discuss the services data management should provide and the aspects of data management.

SERVICES

The issues regarding services of data management will only be touched on briefly, and will be the subject of a subsequent MTB. However, these are the services as we see them, to give a general feel for the topic:

- Relational interface (MRDS)
- Natural language interface (INTELLECT)
- Command-level end-user interface (LINUS)
- MRDS tools
- Index management
- Forms facility
- Macro facility
- Core of DSA DS&R
- Data dictionary

Possible extensions to the planned DSA core are:

- Database machines
- DSA External Access Laver
- Conceptual schema

Some hoped for, time-permitting opportunities are:

- IDS-like interface
- Good report generation

ASPECTS

With each aspect will be presented related items which in some way describe that aspect as it would be ideally, as it is currently, and in terms of goals to attempt to achieve. The goals are phased, indicating those goals are to be achieved as a set, and in what order the phases of goals can be expected to be sought after. In some cases it is not realistic to phase goals, and so are not. Goals preceded by a '*' are considered to be more important than others.

Reliability

IDEALLY

- Data is never lost.

CURRENTLY

- Data is lost on process termination
- On user hitting quit during MRDS processing
- On ESD failure
- On media failure

TARGETED GOALS

Phase I

- * Data is not lost on process termination
- * On quit
- * On system crash with succesful ESD

Phase II

- Data is not lost on ESD failure
- On media failure

Recovery

IDEALLY

⊗ Recovery process yields a consistent, usable database in the state it was in at last commit,
⊗ Is automatic,
⊗ Is fast,
⊗ Does not grow in cost as the database grows,
⊗ Allows the use of undamaged data in the interim,
⊗ Supports checkpoint/rollback,
⊗ Supports networking

CURRENTLY

⊗ dumper
⊗ save
⊗ reload
⊗ quiesce

TARGETED GOALS

Phase I
⊗ * Recovery process leaves a consistent, usable database in same state as at the time of last commit.

Phase II
⊗ Is reasonably fast
⊗ Allows use of undamaged data
⊗ Does not grow in cost as the database grows
⊗ Supports checkpoint/rollback

Phase III
⊗ Supports networking

Salvaging

IDEALLY

⊗ Detect damage ASAP
⊗ Damaged data is not thrown away
⊗ Damaged data can be analyzed logically

CURRENTLY

⊗ Totally blind to existence of databases

TARGETED GOALS

Phase I

⊗ * Able to analyze bad data logically
⊗ Detect damage ASAP

Phase II

⊗ Localize damage
⊗ Invoke recovery mechansim
⊗ User exits in salvager

Performance

IDEALLY

⊗ Competitive with best systems
⊗ Use of parallelism wherever possible.

CURRENTLY

⊗ Not competitive

TARGETED GOALS

Phase I

⊗ * Competitive with general purpose systems, defined here as five (5) times as fast as MR6.5 MRDS in vcpu time and possible to achieve some simple update in less than 10 i/o's on database pages and control structures, in a balanced configuration.

Capacity

IDEALLY

⊗ No limits
⊗ No degradation of performance beyond theoritical lower limit as database grows
⊗ No catastrophic discontinuity as database grows

CURRENTLY

⊗ MRDS limits: 128 relations per database
10,000 tuples per relation
200 attributes per model
64 database openings

TARGETED GOALS

Phase I

⊗ MRDS limits: 1K relations per database
100,000,000 tuples per relation
10K attributes per model
1K database openings

Phase II

⊗ Satisfy requirements for National Library of Medicine and IRS's TAS system; the TAS database has about 100 files of various characteristics; it has a file with 188 million records, another with 12 million records of an average length of 1K (equivalent to 48 billion bytes or 48K MSF components or 280 disk packs (451 type))

Administrative

IDEALLY

⊗ Dynamic meters
⊗ Dynamic restructuring
⊗ Dynamic tuning
⊗ Dynamic relocation
⊗ Placement
⊗ Central administration NOT NEEDED

CURRENTLY

⊗ Non-dynamic restructuring
⊗ Relocation by hand
⊗ Placement by hand

TARGETED GOALS

Phase 1

⊗ * Documentation for administration, including tuning
⊗ * Builtin meters

Phase II

⊗ Dynamic restructuring

⊗ Dynamic relocation

⊗ Placement

Phase III

⊗ Incremental definition of databases as alternative to cmdb and rmdb.

Networking

IDEALLY

⊗ DSA compatibility

⊗ Perform competitively

⊗ Works transparent to user

⊗ Accounting, security, administration functions in a network environment addressed

⊗ Redundant storage and communication links utilized

⊗ Not communication link dependent

⊗ Network optimization available to administrator

CURRENTLY

⊗ Nothing

TARGETED GOALS

Phase I

⊗ * Know full story (networking in general and DSA plans in specific)

⊗ Provide internal support for distributed database system

Phase II

⊗ Support DSA DS&R and networking functions in general

Data independence

For a detailed definition of the various forms of data independence which follow, see DSA Data Storage & Retrieval Vol. I & II.

IDEALLY

⊗ Provide format independence,
⊗ (mixed) media independence,
⊗ model independence,
⊗ machine independence,
⊗ unit of measure independence,
⊗ view independence

CURRENTLY

⊗ Have format independence,
⊗ Unit of measure independence

TARGETED GOALS

Phase I
⊗ Provide subset views (virtual relations)
⊗ synonyms

Phase II
⊗ (mixed) media independence

Atomicity

IDEALLY

⊗ Provide commit/rollback facility
⊗ User-defined atomicity units
⊗ Allow reads to be non-atomic
⊗ Provide checkpoint/restart
⊗ Provide some default atomicity

CURRENTLY

⊗ vfile_ supports atomicity on single operation

TARGETED GOALS

Phase I
⊗ * Provide checkpoint/restart, commit/rollback
⊗ * Support atomicity over any unit (set of units)
⊗ Support atomicity over multiple databases

Concurrency

IDEALLY

- Handles maximal number of simultaneous users
- Is competitive (performance and function-wise)
- Not process id dependent

CURRENTLY

- MRDS supports concurrency on a per relation basis
- vfile_ suports on per record, per operation basis

TARGETED GOALS

Phase I

- * Competitive
- * Use physical record or logical record level locking
- * Deadlock detection, handling

Phase II

- Per field level locking
- Subset level locking

Security

IDEALLY

- Security granularity to field instantiation
- Modes able to distinguish between all functions
- Control delegatable, revocable, unsubvertable, at many levels
- Security is a function of person and function
- encryption

CURRENTLY

- Granularity to field
- Rings, ACLs
- Is function of person
- Encode procs

TARGETED GOALS

Phase I

- Not yet sure of what direction to take on this subject

Transaction processing (TP)

IDEALLY

⊗ Data management works in transaction processing environment (and vice versa)
⊗ Not process id dependent
⊗ Works over networks

CURRENTLY

⊗ Uses current transaction mode in vfile_ and is subject to all of its problems

TARGETED GOALS

Phase I
⊗ Data management work in TP environment (and vice versa)
⊗ Not process id dependent
Phase II
⊗ Works over network

Documentation

IDEALLY

⊗ Sufficient documentation in all areas.

CURRENTLY

⊗ Incomplete set of documentation.

TARGETED GOALS

Phase I
⊗ * Sure to document limitations at minimum.
⊗ Include user guides, program logic manuals, reference manuals and administrative manuals.

Accounting

IDEALLY

- Granularity to field instantiation
- Consider user to be invoker
- Consider person to be billed not necessarily invoker

CURRENTLY

- Nothing

TARGETED GOALS

- Provide hooks to be used by general accounting system

Interaction with OS

IDEALLY

- Operating system uses data management services
- Data management system part of operating system
- Casual creation of databases possible
- All done under auspices of one single storage system

CURRENTLY

- Single storage system, but with much pain and misery

TARGETED GOALS

- Operating system should take into account special characteristics of databases

History

IDEALLY

- Timed versions

CURRENTLY

- No such facility

TARGETED GOALS

Phase 1
- Online, usable versions

Other aspects

Some other aspects were touched upon lightly. They are not presented as goals because they were were not discussed to that level of detail.

Logical integration of data will not be provided. We define logical integration as the ability to look at multiple databases of different types (e.g., IDS and MRDS) as if they were one.

Data dictionary and database machine support will be provided in some as yet undetermined way. A data dictionary for Multics (Information Resource Dictionary System) is being developed now at FSO.