EXPLOITING CAFS—ISP

Working party report
July 1984

ICL COMPUTER USERS ASSOCIATION (UK)

CAFS SIG

EXPLOITING CAFS—ISP

Working party report
July 1984

CONTENTS

PREFACE TO THE SECOND EDITION

1 MANAGEMENT SUMMARY
1.1 The CAFS working party objectives
1.2 An Overview of the CAFS objectives
1.3 The Report Structure
1.4 Conclusions

2 THE BUSINESS CASE FOR CAFS
2.1 The Need for CAFS
2.2 Identification of Good CAFS Applications
2.3 End-user Benefits
2.31 Searching Speed
2.32 Ease of Use
2.33 Iterative Enquiry
2.34 New Opportunities
2.4 DP Department Benefits
2.41 User Satisfaction
2.42 Simplified Systems Design
2.43 Faster Application Development
2.44 Changed Systems Requirements
2.45 Better Hardware/Software Utilisation
2.5 Future Office Automation Systems
2.6 Current and Future Applications
2.61 An ICL Classification of Systems
2.611 Text Systems
2.612 Matching Systems
2.613 Logging Systems
2.614 Change-driven Systems
2.615 Functional Modules and Techniques
2.62 Working Party Members' Systems
2.621 Financial Control Systems
2.622 Merchandising and Stock Control Systems
2.623 Credit Control Systems
2.624 Research Applications

3 THE CAFS SYSTEM
3.1 The System Definition
3.2 The Hardware
3.21 Mainframes and Disc Drives
3.22 The DCU/2
3.23 The DSC
3.24 The CAFS-ISP Engine
3.25 DM1 and ESTRIEL
3.3 The Software
3.31 SV201 and VME.8.01
3.32 DDS.650
3.33 IDMS.350
3.34 QUERYMASTER.210
3.35 PSAM: the pre-selective access mechanism
3.4 A Walk-through an End-user query
3.5 The Relational CAFS Interface RCI

4 MAKING USE OF CAFS
4.1 Introduction
4.2 Existing Systems and Data Files
4.21 The Decision to CAFS-search
4.22 Mandatory Changes
4.23 Recommended Considerations
4.231 ISAM-specific Points
4.232 IDMS-specific Points
4.2321 In-situ Enquiry?
4.2322 Duplexing
4.2323 Relocated and Fragmented Records
4.2324 AREA-to-channel mapping
4.2325 Database Procedures
4.2326 IDMS-Restructure
4.3 The Evolution of Existing Systems
4.4 Future Systems and Data Files
4.41 ISAM Files
4.411 Prime-key Optimisation
4.412 Variable-length Records
4.42 IDMS Databases
4.421 Analysis and Logical Design Implications
4.422 Physical-Database Design Implications
4.4221 Physical Clustering
4.4222 Data-clustering
4.4223 Data-duplication
4.4224 Derived Data
4.5 Using RCI

5 CAFS FIELD-TRIAL EXPERIENCE
5.1 The four ICL field-trial goals
5.11 Performance
5.12 Compatability and Co-existence
5.13 Ease of Design
5.14 New Opportunities
5.2 Field Trial Sites
5.21 British Telecom
5.22 North Thames Gas
5.23 Severn Trent Water Authority
5.24 A South African Site
5.25 Southern Water Authority
5.26 W H Smith
5.3 The RCI Field Trial

6 PERFORMANCE AND SIZING
6.1 Basic Parameters
6.11 The Disc Speed DS
6.12 The Blocking Factor BF
6.13 The Fragment Factor FF
6.14 The Packing Density PF
6.2 Resource Utilisation
6.21 Design-time Resources
6.22 Run-time Resources
6.221 Computer-system Resources
6.2211 The Mainframe OCP Resource
6.2212 The Disc-channel Resource
6.222 End-user Resources
6.3 Early Experience
6.31 North Thames Gas
6.32 Southern Water Authority
6.33 CAFS v Non-ICL Query Systems
6.4 Disc-Channel Subsystem Queuing Models
6.5 A Sizing Exercise
6.6 A PSAM Benchmark

7 RECOMMENDATIONS FOR THE FUTURE
7.1 CAFS/QUERYMASTER Facility Enhancement
7.2 Performance
7.3 Coexistence and Control
7.4 Ease of Design
7.5 Other CAFS interfaces
7.6 Future Environments
7.7 Other Hardware Engines

8 FEEDBACK
8.1 Feedback Request
8.2 Feedback Form

GLOSSARY OF ACRONYMS AND TERMS

BIBLIOGRAPHY

ICL'S Catalogue Sheet CS793 on CAFS-ISP

ICL .... "We can even find a noodle in a hat rack"
PUNCH .... "The Thrut Will Out"

ICL's Brochure P1531 on CAFS-ISP

PREFACE TO THE SECOND EDITION

The first edition of this report on CAFS Exploitation has been taken up within six weeks of its publication. We have therefore taken the opportunity in this second edition to include information on the latest developments in the CAFS system.

These can be summarised briefly as:

1) CME Release 4 now supports CAFS via VME8.01; CME/2, CME/3 & CME/G3
2) CME Release 4 will support VME8.11 in the same timeframe as SV211
3) ICL has announced RCI, the Relational CAFS Interface
4) A PSAM benchmark has been made available
5) SV211's IDMS.400 RESTRUCTURE can remove relocation/fragmentation

Items 1 and 2 will of course be good news for CME users. A COBOL/CAFS interface was nominated by the working party as the top priority; RCI is such an interface.

Four new sections have therefore been added in this second edition. These are highlighted in the CONTENTS list by a " | " in the right-hand margin and are:

3.5 The Relational CAFS Interface RCI
4.5 Using RCI
5.3 The RCI Field Trial
6.6 A PSAM benchmark

Other changes in the body of the report are largely a consequence of these additions and are also marked by a " | ".

The added sections have elsewhere been collected together in an addendum to the first edition. This addendum is available from the same sources as the report itself and has already been widely distributed.

8/8/84

1.1 The CAFS Working Party Objectives

In the summer of 1982, the ICLCUA CAFS Special Interest Group defined three subject areas for working party activity. These were:

1) Interfaces with compilers and databases
2) End-user Language facilities and display methods
3) Text-handling and office automation

The CAFS SIG convened one working party to address the first subject with the following terms of reference:

1) Review facilities and map requirements onto them
2) "Database or CAFS" or "Database on CAFS"
3) Training needs for users to bridge to new techniques
4) Prepare specifications to cover gaps in software

The working party interpreted the topic broadly as the Data Processing Professional's, rather than the End-user's, view of and relationship with CAFS. This report is the result of the working party's activities.

The report content for good reasons exceeds the terms of reference in their strictest sense. For example, we examine QUERYMASTER, which is deemed to be an end-user tool by ICL, from both the DP and end-user perspectives. First, this is the only interface to CAFS in the current SV201. Secondly, it is necessary for the DP department to understand the end-user's interface to CAFS. Thirdly, the other subjects have not yet been addressed by other active working parties.

We have, under conditions of confidentiality, had full access to ICL's plans for a CAFS/COBOL interface and have contributed to developments in this area. The introduction of such a product has been heralded in the VME Statement of Direction of 9th November '83 and at the CAFS SIG open meeting in December '83.

ICL has just announced this product as the Relational CAFS Interface RCI. We have therefore added a preliminary outline of its facilities, techniques and exploitation in this report.

The value of the report would not be enhanced by further delay in its publication and we trust that you find the following content useful.

1.2 An Overview of the CAFS Objectives

'CAFS' in the context of this report specifically means 'CAFS-ISP' and does not include the previous CAFS.800 product. It is only available on ICL 'S' series 2900 computers running under the VME operating system. It is currently marketed in two forms:

1) An enhancement kit to one or both of a DCU/2's UFEDs
2) A DSC which retails as a single free-standing unit comprising a 1/2 DCU/2 with CAFS on its UFED

The purpose of CAFS is to allow searching for data conforming to specific criteria off-line from the 2900 Order Code Processor by means of hardware. The hardware searching considerably enhances the response times for whole or partial file-searches whilst simultaneously freeing up mainframe processing capacity. Orders of magnitude improvements in speed and reduction of OCP-load are normal.

The introduction of CAFS into computer configurations has no effect whatsoever on existing applications software.

CAFS is transparent to the user interface. The majority of data in existing ISAM files and IDMS databases is automatically and invisibly accessible to CAFS via QUERYMASTER. This ensures that Users can gain immediate benefits by installing CAFS without a large prerequisite investment in application software and data restructuring.

CAFS is expected to ease systems design and can therefore be looked at as a productivity tool. System and Data Design may be done with the immediate updating requirements in mind without need to worry about infrequently-used or artificial access-paths which can be replaced by the CAFS hardware.

Programs featuring ad-hoc or occasional access to data may not even need to be written. Such programs take minutes to run and days or weeks to produce. Instead, it may well be possible for the User to perform an interactive query using QUERYMASTER and CAFS. It has been suggested that prototyping can be assisted by CAFS.

Existing systems may have their life considerably extended as small enhancements, taking advantage of CAFS hardware, may well be able to achieve much of what would be required by a total redesign using conventional methods.

CAFS gives the Data Processing Professional an opportunity to design systems which are more responsive to users' real needs. Facilities that might previously have been desirable but were rejected as being too slow and/or costly using conventional equipment, have now become feasible.

The sizing rules for computers have now been turned upside-down and need re-writing as CAFS has introduced a new dimension in computer hardware and software. Clearly CAFS is an opportunity to be grasped by the DP professional and passed on to the users to enable them in time to create new business opportunities.

1.3 The Report Structure

The report first looks at the business case for CAFS. In Chapter 2, it examines the need to searching, gives a management/audit review, considers the role of decision support and finally presents the Working Party's and others' view of likely uses.

Chapter 3 deals with the totality of what is required to produce a CAFS system.

Chapter 4 shows how use can be made of CAFS. It starts by looking at existing applications and then progresses through, extending those applications, to considering entirely new systems. Both conventional files and IDMS databases are discussed.

In Chapter 5, we describe the experiences of the field-trial sites.

Chapter 6 considers performance and sizing topics. It looks at the basic parameters which affect performance and considers the resources required at design time and run time. It concludes by looking at some results returned by early CAFS users.

In Chapter 7, we make recommendations for the future development of the CAFS system.

1.4 Conclusions

We were clear until very recently that the lack of a CAFS/COBOL interface was the major current omission in ICL's CAFS offering. ICL has now rectified this in RCI and the RCI field trial sites have actively and successfully used the product.

ICL marketing strategy for CAFS appears to be "CAFS with everything". Recent product announcements would indicate that the packaging and pricing of CAFS is aimed at selling all new 2900 computer configurations with CAFS as an integral part of the system. Clearly the pricing of CAFS enhancement modules makes it extremely attractive for current VME 'S' Series 2900 users to add CAFS to their configurations and obtain the performance benefits and new opportunities that CAFS presents.

The CAFS product is clearly only at the starting point of a long development path. This should not however deter users from dipping their toes in the water at this stage. In the context of the current prices for complete computer systems, the CAFS subsystem has much to offer at a low price even in its current first development state.

2.1 The Need for CAFS

The computer industry has invented a battery of techniques for enabling selective access to stored data. DP departments share the indexed-file with secretaries but are alone in their use of secondary-key, random-access and CODASYL-database access techniques.

However, it is often necessary to search the data in ways which are not supported by designed-in access techniques. For ad hoc, infrequent and unanticipated queries, the data or a large part of it must be searched in a serial fashion from end to end.

The business case for CAFS rests primarily in its ability to speed the search of data files by an order of magnitude. This allows ad hoc management information requests, using complex search criteria, to be addressed at a speed and cost which is both practical and acceptable to many current and future users of large data processing systems.

The justification for using CAFS in specific applications will obviously vary greatly. CAFS is clearly needed for selecting from large files but is also proving useful for small files. These selection requirements may be the primary purpose of the systems (e.g. for Police type systems) or to enable reports to be produced for traditional data processing systems.

2.2 Identification of Good CAFS Applications

Overall there must be a need for systems that involve complex data searching, in addition to any other functions that can be handled by simple key accessing.

Alternatively, there may be the possibility of using CAFS searching as a substitute for conventional keyed accessing, where this provides scope for cheaper, simpler, and faster systems development and implementation.

Particular application possibilities are indicated by:

1) an extensive demand for unpredictable online enquiries
2) a number of batch reports which can be replaced by CAFS-assisted online queries
3) a database structure which would without CAFS have to be more complicated to cope with relatively rare processes, either occasional batch jobs or unusual types of enquiry
4) a database structure which would without CAFS have to be more complicated for performance reasons
5) a simple ISAM file which with CAFS can satisfy the requirements that would otherwise have to be met by an IDMS database or a complex inverted file indexing system
6) an ISAM file which with CAFS can be used alongside an IDMS database, producing a hybrid system with the best characteristics of both organisations
7) enquiries on the textual elements stored within conventional records which will be significantly assisted by CAFS
8) a need for fuzzy enquiries, using incomplete or fuzzy criteria.

2.3 End-user Benefits

The facilities of CAFS and all computer systems are ultimately aimed at the end-user. Therefore, this section assesses CAFS specifically from the end-user's point of view.

2.31 Searching Speed

The raw search speed of CAFS allows end-users to achieve acceptable terminal response times for management information requests which have not been catered for in the system design.

This will enable a manager to pursue his line of enquiry more effectively. He will work online rather than submitting a sequence of batch jobs and his train of thought will not be interrupted by waiting for the computer.

2.32 Ease of Use

CAFS has been added to the VME system in a way which preserves existing interfaces and adds as little as possible to those interfaces.

QUERYMASTER is the only current interface to CAFS and is an easy-to-use end-user-level interface. QUERYMASTER rather than the end-user decides how and when to use CAFS, though the DP department can at its discretion provide the end-user with a residual ability to inhibit and enable CAFS searching.

CAFS is visible therefore only in the sense that it does not always perform 100% of the work needed to answer a query. Response-times will therefore vary from current times to 6-60 times faster.

The end-user therefore needs education as to **what** he can now ask for rather than **how** he does it.

2.33 Iterative Enquiry

Users commonly want to ask a sequence of questions for many reasons. Their next question is determined by the previous response. They ask the wrong questions, they want to browse, they want to refine the last query and they want to see the answers sorted and organised differently.

The speed of CAFS makes online ad hoc query genuinely possible, whereas traditional batch systems often give unacceptable response times.

2.34 New Opportunities

There are many systems, where because data structures do not easily support certain types of enquiry, the cost of running a job can be out of proportion to its perceived worth. Some of these jobs, with the aid of CAFS, could be run for a fraction of the cost. This could apply equally to batch and TP Systems, especially where large data volumes require to be searched in order to complete the processing task.

2.4 DP Department Benefits

The following benefits are seen to accrue to DP Management due to extensive use of CAFS.

2.41 User Satisfaction

Because more of their management information and decision support requirements are being computer supported, as well as their more routine batch systems, users should be more satisfied with computer departments. CAFS should give users the immediate access to their data which they enjoyed with smaller-scale files in the pre-computer age.

2.42 Simplified System Design

Because less complex data structures need to be developed to support complex search criteria with the use of CAFS, simplified design should be achieved [see Ch 4].

2.43 Faster Application Development

Because systems can hopefully be simpler, faster application development, especially for management information systems, should be achievable.

2.44 Changed Systems Requirements

Requirements for routine batch reports and special ad-hoc batch reports should be reduced due to on-line CAFS Systems.

In addition the extensive use of QUERYMASTER facilities, with CAFS backing, should allow DP Departments to concentrate on getting the base systems developed correctly and hopefully give longer life to these base systems.

2.45 Better Hardware/software Utilisation

With the increasing cost of software development and the decreasing cost of computer hardware, CAFS allows system development costs to be reduced.

Hardware spend will to some extent be reduced, deferred and transferred from spend on mainframe upgrade to spend on CAFS. This is because the burden of data-searching will have been transferred from the central OCP to the available CAFS engines.

Because CAFS replaces software implementation by hardware and simplifies system design [2.42], there will be less spend on system and software development.

2.5 Future Office Automation Systems

In the future, CAFS is seen as a way of providing the necessary response times for office automation systems, where large volumes of text may need to be searched. This will cover such office systems as word processing, electronic mail, electronic filing and retrieval, and personal computing. Some of these requirements may be satisfied by QUERYMASTER but the real need is for full text processing capabilities, backed up by CAFS.

2.6 Current and Future Applications

The section 2.2 above lists some common indicators of likely CAFS applications. This section lists some concrete examples of systems which the working party and ICL envisage will be implemented in the immediate or near future.

2.61 An ICL Classification of Applications

These applications exhibit one or more of the characteristics examined above [2.2] and are listed without comment.

2.611 Text Systems

Bibliographic: publishers, researchers
Client notes: service industry - legal, medical, public administration
Dictionaries: phrases, quotations, words
Documentation: 'help' information, user-guides, reference manuals
Indexes: research - abstracts, book-indexes, case-law, patents, reviews
Market research: profiling, segmentation
Public enquiry: timetables, viewdata and PRESTEL services
Public record: court transcripts, public enquiries, 'HANSARDs'
Registries: births, marriages, deaths, land, housing, directories

2.612 Matching Systems

Many of these systems are designed to relate a likely 'buyer' with a likely 'seller' in the most general sense. Matching criteria are inspecific and fuzzy and are usually refined in an iterative process fed by the information received.

Catalogues: hypermarkets, mail-order, special offers
Classified advertisements: local agencies, specialists, 'yellow pages'
Credit control: debt-analysis
Image matching: codified feature recording
Personnel systems: profiling, recruiting, skill-assessment, training
Telephone selling: cars, property

2.613 Logging Systems

Incidents are recorded in the knowledge that the information is going to be useful but without a specific idea of its future use.

Accidents/incidents: rail/road/air/sea transport sector and insurance
Manufacturing: (planned) maintenance, quality control
Product Monitoring: customer service, durables - car, electronics, house
Sales: feedback to sales management and marketing

2.614 Change-driven Systems

The need to control change is a fundamental requirement of any system. Here we are highlighting systems in which system and change are intimately linked for example in the time dimension. This may be because the system is interrupt-driven or polls the recorded changes rapidly using CAFS.

Auditing: exception analysis, integrity review
Command and Control systems: military and industrial
Epidemic monitoring: case-recognition
Impact analysis: change control in design and production
Inference systems: free-association or rule-driven, interviewing
Knowledge engineering: 'data-driven' systems, rule-triggering
Patient care: intensive care and preventative medicine
Plant monitoring: complex industrial and public utility systems

2.615 Functional Modules and Techniques

These are not complete systems but are common system-components.

Batching-up enquiries: using the 16-channel parallelism of CAFS
Data cleaning/validation: GIGO! Applicable in transition to ADRAM'd data
Exception reporting: see also auditing
File stripping: from either public or corporate/departmental data
Journal analysis: activity analysis, problem-shooting
Name matching: users accessing with available data rather than 'DP' keys
Prototyping: simple data-design with CAFS substituting for key-access
Syndication: simultaneous use of multiple CAFS-engines on a search

2.62 Working Party Members' Systems

The following applications are those identified by four members of the working party as possible current and future applications of CAFS within their own installations.

Most of the uses of CAFS envisaged by working party members encompass giving users the ability to extract ad-hoc management information from large data files at VDU terminals using Querymaster and CAFS. In many cases a similar service could not be provided without CAFS.

Specific uses are seen as follows.

2.621 Financial Control Systems

Nominal Ledger systems are increasingly being asked to provide management information reports in various formats to help control businesses. CAFS is seen as a way of achieving this. Specific examples for retail outlets are branch performance and profitability reports and league performance of branches by turnover per square foot. In the public sector, similar examples would be performance indicators of responsibility centres and analyses of the cost of providing a specific service.

2.622 Merchandising and Stock Control Systems

For retail organisations with computerised stock replenishment systems for their branches, automatic allocation systems require to examine stock and sales both by line number within branch and by branch within line number. As only one can be supported efficiently by conventional methods CAFS is seen as a way of providing the required speed of processing. In addition complex routine and ad-hoc management information requirements will be able to be fulfilled.

2.623 Credit Control Systems

For organisations operating with large computerised volumes of data, CAFS is seen as a way of providing some operational and management information subsystems which otherwise could not be effectively supported. These include selective promotional mailing and debt-recovery systems involving complex criteria.

2.624 Research Applications

CAFS is seen as a way of providing a fast search mechanism through large data files used in various university research projects. Specific examples follow below.

1) **Cancer Studies:**
   Selecting subsets of records from large (100+ Mb) data sources on the basis of multiple search keys e.g. primary and secondary causes of death, sex, occupation etc.
2) **Forestry:**
   Currently a wood structure index and timbers database holding details of 250,000 trees has been built up. Because of the current limita tions of the hardware and the high costs involved it is rarely accessed.
3) **DNA Databases:**
   Interest has been expressed in using CAFS to search for matching strings in DNA databases. This is an example of the sort of analysis which is currently not feasible with existing hardware but would be possible using CAFS.

Many other potential research applications have been identified.

The purpose of this chapter is to describe the facilities of the CAFS system in broad terms. The word 'system' needs emphasising because the contribution which CAFS will make stems from the system integration of a number of components. The CAFS hardware is just one of these components. The components and their new 'CAFS specific' features are documented here.

It is also important to recognise that individual facilities of CAFS-system components are not necessarily included in the CAFS-system. Two examples might clarify this point:

1) QUERYMASTER searches serial files; CAFS/QM does not in SV201
2) CAFS-ISP hardware performs quorum searching; CAFS/QM does not

The interfacing of the CAFS-related components will undoubtedly be enriched in the future.

It is not necessary or possible in a few pages to describe the system in ultimate detail. The information here includes the main facts needed to understand the other chapters in the report.

3.1 The System Definition

The CAFS system combines hardware and software to enable data searching and information selection to be done at 'disc-drive' speeds. It replaces central mainframe software activity by peripheral disc-channel hardware activity.

There should be no confusion between ICL's previous CAFS.800 engine and the current CAFS-ISP system offering. CAFS.800 demonstrated the effectiveness of the hardware approach to information retrieval [13, 14, 15]. There, the similarity with CAFS-ISP ends.

The CAFS.800 engine is a stand-alone DME device working on specially re-engineered EDS60 drives with new software. The CAFS-ISP hardware can be slotted into existing VME systems with a quick field-upgrade.

3.2 The Hardware

3.21 Mainframes and Disc Drives

The CAFS system is available with VME on all 2966-family mainframes, that is the 2953, 2957, 2958, 2966 & 2988. Single OCP mainframes can accomodate up to 6 CAFS engines; dual and superdual configurations can accomodate up to 8 engines. Section 4.2324 describes how to syndicate a query to deploy all 6 or 8 CAFS units in parallel.

The CAFS system will search data-files stored on MDSS-family disc-drives which are connected to DCU/2 or DSC disc-channels. These are the EDS80 exchangeable and FDS160/FDS640 fixed drives. These drives have a peak data transfer rate of 1.2 Mb/sec; the effective CAFS-system data-transfer rate can be calculated [Ch6] from this.

3.22 The DCU/2

The DCU/2 disc control unit contains two DCM disc control modules. Each DCM controls one disc-channel with up to 16 disc-drives on it. Each DCM can have a CAFS-engine attached to it in a field-upgrade.

3.23 The DSC

The DSC disc control unit is initially configured with one DCM with built-in CAFS-engine on the same platter. The DSC extension module is another DCM plus CAFS platter. Each DSC DCM channel may connect to 32 disc-drives rather than 16 disc-drives.

The introduction of the DSC anticipates that CAFS-capability on a channel will become the rule rather than the exception.

3.24 The CAFS-ISP Engine

The CAFS engine sits in the disc-channel and is physically attached to the DCU/2 DCM in the field or to the DSC DCM in the factory. A block diagram of the CAFS engine and a second-level hardware description is given in the ICL Catalogue Sheet CS793 reprinted as pp57-8.

The CAFS-engine components perform a number of functions:

1) the LFU identifies database pages, records and fields
2) the 16 KC key-channels evaluate atomic selection criteria
3) the SEU search evaluation unit identifies 'hit' records
4) the RU retrieves required data-fields from hit records
5) the RP retrieval processor controls CAFS buffers, counts, totals etc.

The power of the CAFS-engine stems from its architecture. The processes above are carried out in parallel by a set of microprocessors working simultaneously. There is no reason in **principle** why the number of key channels should not be increased or why the micros should not increase in functionality.

3.25 DM1 and ESTRIEL

The recently-published ICL 'VME Statement of Direction' [9] announces ICL's intention to support existing workloads on future mainframes. This includes ICL's intention to provide CAFS system capability on those mainframes.

3.3 The Software

The CAFS SIG Working Party was given the task of looking at 'CAFS and DP'. Its original charter anticipated that there would be many CAFS-related changes which the DP department had to manage. Just as the hardware upgrade is straightforward, there are remarkably few changes on the software side. This arises from the ICL approach of introducing CAFS in as 'transparent' a way as possible, leaving the existing interfaces CAFS-independent.

New versions of the software components are required but these are the normal system upgrades.

3.31 SV201 and VME8.01

The CAFS system is available exclusively on VME and CME. SV201 and VME8.01 are the first versions of the VME system and the VME base respectively. CME Release 4 supports VME8.01 in its CME/2, CME/3 and CME/G3 regimes. VME8.11 will become available on CME Release 4 at the same time as it becomes available in SV211.

Note that CAFS is not available on DME, TME or on the DME side of CME. Specifically, CAFS cannot search DME files via ADRAM.

SV201 is not a force-fed release but is required by those sites requiring CAFS, superdual mainframe configurations or new releases of Data Management software such as AM.100, DDS.650, IDMS.350, PDS.110, QM.210 or TPMS.320.

3.32 DDS.650

DDS is ICL's Data Dictionary System. It is a database documenting an organisation's business and information systems. This data is used to control and expedite the use of ICL's Data Management products of which IDMS.350 and QM.210 are relevant here.

DDS stores its data in elements; each element has a number of properties. New elements have been defined and existing elements have been modified by the addition of new properties; existing properties have also been modified. These modifications extend the IDMS and QUERYMASTER interfaces and introduce TEXT as a new data item.

The AREA element has a new property, the ITEM and SUBSCHEMA elements have extensions to their properties and the SECTION element has been added.

The AREA element's new property defines the CAFS-searchability of an IDMS area:

```
*PAGE-FORMAT IS { SYSTEM DEFAULT }
                { SEARCHABLE     }
```

The ITEM element TYPE property has a new TEXT format.

The SUBSCHEMA element RECORDMAP property has a PRIVACY-LOCK for SEARCH; note that this is defaulted to NO and must be deliberately set to YES to allow CAFS searching.

The new SECTION element is an implementation level data element that represents an unordered sequence of text items. The element is used by QUERYMASTER.

TEXT data must be the last data in a field and is stored in a new 'signalled-text' format. This uses approximately 50% more space than the standard PIC(X) COBOL format for the same text.

See [1] for the full definitions of these new or changed DDS components.

3.33 IDMS.350

This is ICL's Integrated Database Management System, an implementation and enhancement of the CODASYL database standard. This is the first version of IDMS which allows CAFS searching on IDMS databases. The DP department decides on an AREA basis which AREAs should be CAFS searchable.

Chosen AREAs are dumped using IDMS-DUMP, redocumented in DDS and restored in a new format to enable CAFS searching. It is important to stress that this is a simple within-page reformat and not a change on the scale of IDMS-RESTRUCTURE or IDMS-REORGANISE.

3.34 QUERYMASTER.210

QUERYMASTER executes queries on serial and ISAM RECMAN files and IDMS(X) databases. It provides a relational read-only interface to the end-user.

ICL's Data Management architecture uniquely combines the complementary benefits of CODASYL data-access techniques and the RELATIONAL end-user interface. CAFS supports this software combination with hardware power.

This is the first version of QUERYMASTER which allows CAFS-searching or handles TEXT. It exploits CAFS at the discretion of its inbuilt optimiser. The strategies for retrieving necessary data and using CAFS are documented in [6, Ch7]. The necessary conditions for exploiting CAFS are documented in [6, Ch7]; see especially the Publication Notices.

3.35 PSAM

PSAM is the Pre-Selective Access Mechanism which ICL will provide as a separately-charged part of SV211's VME8.11.

PSAM enables batch programs accessing RECMAN serial and ISAM files to use CAFS without any change to their code. It is therefore an excellent interface to CAFS for programs which the DP department does not want to or cannot change. These programs may be for example antiques, undocumented, in RPG2, 'one-offs' or packages. Some programs generated by APPLICATION MASTER and REPORTMASTER will benefit from the use of PSAM.

Let us suppose that a program is only interested in records from a file which satisfy a particular condition C_1. Let C_2 be a possibly-weakened form of C_1 which the CAFS system can evaluate. All records satisfying C_1 will also satisfy C_2; C_2 therefore serves as a first filter of the records. Note that it is important that C_2 should not be stronger than and filter out more records than C_1. An 'overstrong' C_2 will effectively change the job done by the program.

PSAM allows the DP department, with an SCL command, to 'front-end' a program with CAFS. The program will continue to ask for records but will only receive records satisfying C_2.

The program will continue to evaluate given records to see if they satisfy condition C_1. The CAFS benefit is that far fewer records will be presented to the mainframe. A much higher proportion of records, indeed 100% if $C_1 = C_2$, will be hit records. The program will therefore run using less OCP mill and complete much earlier. Chapter 6 includes the results of a first benchmark on PSAM.

The PSAM user-interface is described in the VME User Guide [12, 7.3.3.1] from which we take the following example.

A program ISSUESUMMONS processes a file containing records of several different types. Each record-type starts with a four-character field TYPE. ISSUESUMMONS is only interested in field where:

TYPE = 4 and AMOUNT_OWING $\geq$ £300,000.00

The following SCL, with STCC being the short form of 'SET_CAFS_CRITERIA', exploits the PSAM option:

```
ASSIGN_FILE (MYMASTER, MAINFILE, LEVEL = C)
STCC (LNAME = MAINFILE,
      ITEMS = TYPE (1:4) & AMOUNT_OWING (9:9),
      CONDITION = "(TYPE EQ 4) AND (AMOUNT_OWING GE 300000)")
ISSUESUMMONS
```

Note that CAFS passes back only records of the right type. In contrast with the QUERYMASTER interface, records of more than one type can be returned via PSAM by one CAFS scan. Field positions and lengths are expressed in terms of character positions. Up to 14 selection criteria can be included in the condition. No inter-field comparisons are allowed.

A word of caution. Had the monetary unit of the filed-data been pence rather than pounds, CAFS would have passed back records where the AMOUNT OWING was £3000 or over. The program would have done further filtering but would have produced the required result. Had the mistake about the monetary unit or the condition been the other way round, the PSAM/program combination would not have produced the required answer.

3.4 A Walk-through an End-user Query

In response to an end-user query, QUERYMASTER will present a single table of information. This information will have been collected from a number of different record-types filed in an IDMS database and/or a number of flat files.

QUERYMASTER contains an optimiser which decides in what order it will access the necessary physical record-types. The DP department should beware of defining query-views which allow QUERYMASTER alternative paths to the data. Such query-views may provide an ambiguity which would cause the end-user to get a correct answer to a question he did not ask.

The end-user has a limited ability to advise the optimiser on the first record-type to be accessed and the data-relationships to be used. The end-user may also discover how QUERYMASTER approached the data by using the reporting category ACCESS-PATH. The optimisation strategy is defined in [6, Ch7]. Note that QUERYMASTER does not have access to volumetric information nor to information on the selectivity of the query's conditions.

QUERYMASTER will use an access-key first as a selector if possible. However, QUERYMASTER may subsequently need to search the full extent of a file in which records of a certain type are to be found. It will then use CAFS if possible; the necessary conditions for CAFS-searching are defined in [6, Section7.4].

Let us now concentrate on one CAFS search. This search will be broken into a number of CAFS-search-fragments. Each fragment is contained within one particular disc-cylinder and within one particular file-extent. Each fragment will be no larger than the system setting for a maximum search-fragment; this is defaulted to ten tracks.

In this way, the system prevents a CAFS search from monopolising a disc-channel for an open-ended amount of time. The question of CAFS searches interacting with disc-channel requirements is discussed in more detail in Chapter 6.

3.5 The Relational CAFS Interface RCI

RCI provides both a program interface to CAFS and a read-only relational interface to data. RCI can be used by both batch and TP programs. As RCI.100, it will be available to work with SV211-level software namely VME8.11, DDS.700, TPMS.400, IDMS.350/400 and COBOL.131.

The principle facilities and benefits of RCI are seen to be:

1) a transparent COBOL interface to CAFS
2) a read-only relational interface to IDMS/RECMAN data
3) value-based privacy to add to CODASYL's item-based privacy
4) additional program/data independence for simpler systems-maintenance
5) simplified programming
6) selection on and processing of text-fields

RCI comes from the same family as QUERYMASTER and PDS, thus extending the relational interface from personal to corporate databases, from programmers to end-users.

Just as QUERYMASTER presents a hit-table of logical-records to the end-user, RCI presents a hit-file of logical-records to the COBOL program. These logical-records may consist of data joined together from several physical records; they may also satisfy some condition which can be parameterised at run-time.

Just as the end-user accesses data via QUERYMASTER and a QUERY VIEW, the application program accesses data via RCI and an APPLICATION VIEW (AV). The AV contains definitions of one or more RELATIONAL VIEWS (RVs); an RV may be thought of as a logical-file of logical-records.

Like QUERYMASTER, RCI decides when and how to use CAFS. It contains the same optimiser to determine the appropriate data-navigation. It has very similar data-access and CAFS-exploitation facilities to QUERYMASTER. The range of data-types which can be 'viewed' simultaneously via RCI is greater than than for QM.210 and in line with the range for QM.250.

There is architectural intention in the QM/RCI similarity. ICL expects RCI users to prototype and test RVs with QUERYMASTER and this has been found beneficial in the RCI field-trial [qv Ch5].

It is intended that RCI will make programming, especially IDMS DML programming, more easily accessible to more staff. Low-level data-management will be done by RCI rather than by the program. The IDMS-specific concept of 'currency' will be less often encountered given the assistence of RCI.

4.1 Introduction

CAFS can be made use of in a data processing installation either with existing systems and data files or for new systems and data files.

In both cases the right environment has to be created in order to exploit the CAFS system.

This section outlines how existing systems can be amended to use CAFS and the analysis and design implications of CAFS on new developments.

In designing new systems to utilise CAFS, a central problem to be addressed is the degree of normalisation that should be introduced. CAFS is best suited to use large unnormalised records, where related data items are held in the same record. However traditional processing systems are most flexible when designed with fully normalised records. There is no simple answer to this design dichotomy, and each case must be resolved on its own merits.

4.2 Existing Systems and Data Files

CAFS can be used with existing systems where there is a demand for on-line enquiries or extensive ad-hoc reports from existing data files, which have not been designed to support the required access paths.

All the conditions and observations in this section apply also in the case of new systems [qv 4.3]. The content of this section, apart from the mandatory changes, is determined by a judgement of what is worthwhile. This involves an assessment of the value and difficulty of the listed tasks. The judgement of this report is of necessity qualitative; individual sites can make a more quantified decision.

Some changes to data are not independent of existing valued code or may materially affect the performance of existing systems. It is likely that these changes would be regarded as too strategic, ie too difficult, for implementation on existing data.

4.21 The Decision to CAFS-search

The CAFS facility offers a realistic choice between searching the operational data in situ or searching an alternative possibly subsetted copy. In some cases it is necessary to search the 'live' data for reasons of objectivity or performance; in other cases, the use of the data calls for a 'snapshot' of the live data. In the third category of cases, a decision has to be made between 'in situ' and 'offline' searching.

The current use of the data and access-requirement to it are the major factors affecting this decision.

Imagine a high-volume TP system supported by object-data and operating at a peak-period. The introduction of CAFS-searching on that data could overload disc-channels and elongate terminal-response times unacceptably. This extreme example indicates the issues involved.

If it is not possible to CAFS-search the data in situ, several alternatives are available.

4.22 Mandatory Changes

Once the decision has been made to CAFS-search a given file, that file must meet the current CAFS-system requirements as defined in Chapter 3.

All DME files must be converted to VME files. This can be achieved using an ADRAM conversion aid. An audit of the data on the VME side after conversion will be assisted by CAFS.

VME files may not be accessible by CAFS because they are on the wrong volume, disc-drive or channel. They can be addressed by CAFS by attaching disc-drives to a CAFS-capable disc-channel or by mounting EDS80 disc-volumes on a drive connected to a CAFS-capable channel. The files themselves may need to be copied to EDS80, FDS160 or FDS640 disc-volumes as these are the only types searched by CAFS.

Data which is not in ISAM or IDMS files cannot currently be searched by CAFS although serial-file searching is anticipated in a future VME release. Such data therefore needs to be filed in an ISAM or IDMS structure. IDMS areas which are to be searched by CAFS must be converted to a new page-format. This is a one-off job achieved by a dump/restore procedure.

The software environment must be that defined in Chapter 3, specifically VME8.01, DDS.650, QUERYMASTER.210 and IDMS.350 for IDMS users. QUERYMASTER is the only current interface to CAFS; this implies that target-files must be documented in DDS.

4.23 Recommended Considerations

These changes are not mandatory and are suggested in the cause of improving the performance [qv Ch6] or applicability of CAFS.

A file which is fragmented into a number of file-extents should be mapped into as few extents as possible. Disc-volumes with faulty tracks may increase the number of file-extents. Obviously, discs with no faulty tracks are preferred. If there are faulty tracks, these should be marked as used rather than replaced by alternative tracks.

Data not searched by CAFS should be allocated to non CAFS-capable disc-channels as far as the requirements to balance the system load across channels allow. System files, program libraries, TP-parameter and TP-slot files are in this category.

4.231 ISAM-specific Points

A CAFS-search on the main body of an ISAM file will be supplemented by a **separate** search on each required block of the overflow area. This is less efficient than a single CAFS-search on the same number of blocks as it increases the OCP resource and elapsed time required.

CAFS-searching is therefore a reason for reducing the percentage of records in the overflow area. This can be done by reorganising the file more frequently and/or lowering the packing density.

4.232 IDMS-specific Points

CAFS complements the access-methods defined by CODASYL. Physical-database design is geared to the high-volume use of the data. Therefore it is possible that the lower-volume uses of the data would be better implemented via CAFS-access than via IDMS-navigation.

IDMS records of a particular type are typically less densely packed in an area than records in an ISAM file. The CAFS data-delivery rate will therefore be lower. However, to measure the usefulness of CAFS in terms of its delivery-rate is incorrect. This should be measured in terms of the reduction in elapsed-time for a query and in terms of the reduction in resource-requirement on OCP and disc-channels.

These points cover the decision to CAFS-search the IDMS-database, duplexing, relocated/fragmented records, the AREA-to-channel mapping, DB-procedures and IDMS-restructure.

4.2321 In-situ Enquiry?

If CAFS is to be used to search an IDMS-area, that area must be converted page by page. This, as stated above, is a linear one-off process but it does bring a run-time overhead.

Database pages are converted between the old 'non-CAFS' and the new 'SEARCHABLE' formats every time they move between disc and store. If there are 'R' records on the page, the costs in machine instructions are:

40 + 30R PLI for reading the page; 40 + 40R PLI for writing

To put these figures into context, the 'average' IDMS DML statement uses some 1000 PLI in IDMS.350. IDMS.350 uses less PLI than its IDMS.200 and IDMS.320 predecessors.

The decision to allow CAFS-searching is made for each IDMS area.

A further consideration is that CAFS-searching is done with area-level locking or with no locks set. The former may be incompatible with other uses of the data and the latter may be unacceptable.

The alternatives to free in-situ enquiry include:

1) snapshotting to ISAM files
2) duplexing the database
3) keeping a restored IDMS-dump online for CAFS and fast-recovery
4) scheduling queries in quiescent periods

4.2322 <u>Duplexing</u>

The obvious way to handle an increased workload is to double the number of access channels. The needs of resilience and increased accessibility are simultaneously served by siting the AREA-plexes on different DCMs.

The usefulness of duplexing increases as the percentage of read-only activity increases; CAFS-searching is a read-only activity and uses only one plex. Both plexes are in the same page-format, either SEARCHABLE or not. IDMSX normally flip-flops accesses to the two plexes but if a CAFS-search is proceeding on one plex, IDMSX directs requests to the other plex.

The system does not prevent CAFS-searching proceeding on both plexes simultaneously. The question does arise therefore as to whether both plexes should be on CAFS-capable drives. If CAFS is only available on one plex, CAFS-requests are queued on that plex, leaving the other channel open for non-CAFS TP-access.

Duplexing is only available with IDMSX as are several other facilities useful in the CAFS/QUERYMASTER context. These include page-extend, secondary-indexes and multiple-areas for a single record-type.

4.2323 <u>Relocated and Fragmented Records</u>

These cannot be recognised as hit-records in a CAFS-search. QUERYMASTER will either comment when it is told of the offending record or, if a standard modification is made to the software, will comment the first time it sees such a record.

IDMSX-index records can be relocated at run-time but these records are not currently CAFS-searchable. Other relocated records can only be created by IDMS-RESTRUCTURE and are removed by IDMS-REORGANISE. IDMS.400 RESTRUCTURE available in SV211 will avoid creating and will remove relocated records.

Fragmented records can be either fixed-length or variable-length.

Fixed-length records can only be fragmented by STORE if there is a 'MINIMUM ROOT' or 'MINIMUM FRAGMENT' clause in the IDMS STORAGE-SCHEMA. The fragmentation is removed by IDMS-REORGANISE by omitting such clauses.

Variable-length records for which 'MINIMUM ROOT IS RECORD LENGTH' is defined in the STORAGE-SCHEMA will not be fragmented when first stored. They may be fragmented if lengthened. IDMS-REORGANISE, and at IDMS.400 IDMS-RESTRUCTURE, will re-STORE them and therefore remove the fragmentation. IDMS-databases should therefore be monitored for the presence of fragmentation.

4.2324 <u>AREA-to-Channel Mapping</u>

The distribution of IDMS-AREAs across disc-channels may be affected by the new traffic-pattern on the AREAs and by the distinction between CAFS-searchable AREAs and non CAFS-searchable AREAs.

Queries may usefully be **syndicated** to a number of CAFS engines. IDMSX enables records of one type to be stored in multiple AREAs which may be accessed by multiple CAFS units. Eight CAFS engines working in parallel give almost eight times the search-speed because the lightly-loaded OCP can take eight times the work without becoming a bottleneck.

Each AREA storing the target record-type should be on its own CAFS-capable disc-channel. IDMS-subschemas must be tailored to each AREA to prevent QUERYMASTER searching other AREAs in sequence. A QUERYMASTER AVM must be set up to search each AREA. The query must then be **syndicated** from a file using say the VME event-mechanism; answers from each component-search will be consolidated using the event-mechanism again.

This requires some work on the part of the systems-management and is not an intrinsic part of the current CAFS-system. It also requires that the user file his enquiry rather than operate online and so is perhaps restricted to enquiries of a 'production' nature.

Syndication is a prime example of distributed processing. The query is distributed to a number of CAFS-engines. Note that requests for AVERAGES do not distribute correctly as an AVERAGE of AVERAGEs is unlikely to be correct.

4.2325 <u>Database Procedures</u>

QUERYMASTER will invoke 'after-GET' DB-procedures in order to enforce the privacy requirement in a CAFS-independent way.

'After-GET' procedures are not compatible with CAFS-searching if they change the data-format. Compression and encryption of data are performed in this way. Procedures for example for privacy, security and audit which do not change the data-format are compatible with CAFS-searching.

4.2326 <u>IDMS-Restructure</u>

An IDMS-RESTRUCTURE is a less complicated task than an IDMS-REORGANISE. With the exception of recommending the removal of relocated/fragmented records, section 4.2 considers only changes brought about by IDMS-RESTRUCTURE.

Data and database-structure may be subtracted, added or changed.

The removal of features of an IDMS-database implies that existing affected programs should be changed or discarded. Superchief-records, access-infrastructure, especially record/set indexes, and derived-data are all candidates for removal. This will particularly be the case if the database design was comprehensively enthusiastic rather than basic and prototype.

A skeletal database-design serving only basic requirements can be enhanced to facilitate both QUERYMASTER and CAFS. QUERYMASTER is helped by the presence of record-indexes and by OWNER-pointers. CAFS is helped if the number of record-types required for a query can be reduced by duplicating the data. This last technique must be used with care and is discussed below [4.4223].

4.3 The Evolution of Existing Systems

Existing data-systems are evolved by the addition of new data and new data-subject areas.

QUERYMASTER itself contributes to easier evolution by removing the distinction between IDMS- and ISAM-held data. Extensions to existing IDMS-based systems may therefore be prototyped using ISAM initially.

The most significant new data-type is TEXT and CAFS offers even greater performance-boost on TEXT than it does on conventional data.

It is probable that free TEXT will not live comfortably within already-designed data-systems and its separate storage in a new IDMS-AREA or ISAM-file seems more attractive. ISAM-files offer a crucial advantage for TEXT records which may be modified and lengthened: they cannot become fragmented in an ISAM-file and remain CAFS-searchable.

4.4 Future Systems and Data Files

Future systems and data files should be designed with CAFS useage in mind if CAFS is considered to be an effective method of implementing specific business requirements. Prior to evaluating the design implications of CAFS on future systems development it is useful to classify systems as follows in order to assess the likely usefulness of CAFS.

1) **Batch Systems:**
   require significant data searching for non key items.
2) **High Volume TP systems:**
   CAFS is unlikely to be useful as access is probably by pre-defined keys. The very high volume of operational transactions may well overwhelm a battery of CAFS engines if such keys are not implemented.
3) **Low Volume Online Systems:**
   CAFS is very likely to be of significant benefit for these types of systems.
4) **Office and Text Systems:**
   In the future CAFS is likely to be of use in these types of systems.

In addition two further systems implications of CAFS need to be considered when designing new systems. The first is that CAFS may move more systems from batch to TP environments. This is likely for some ad-hoc management information systems. The second is that CAFS will give the opportunity to develop some new systems that previously would not have been either cost effective or timely. The new CAFS facilities to handle text efficiently are relevant here.

Encrypted and 'packed-decimal' data should be avoided. The following sections outline the design implications specific to ISAM and IDMS.

4.41 ISAM Files

CAFS should prove especially useful for developing systems around large volume IS files. Although these files should support large volume predefined regular enquiries, identified during the analysis phase of a project, there will be many management information requests which would be suited to the use of CAFS processing. Obviously if search criteria can be satisfied via primary or secondary keys then CAFS should not be used.

ISAM-files as now implemented in SV201 offer two advantages over IDMS-databases. They do not fragment records and they can limit the CAFS-search using the primary-key. They also newly support alternative-keys.

As well as the conditions outlined for existing systems in section 4.2, the following design implications also need to be considered.

4.411 Prime-key Optimisation

As stated above, the prime-key serves two purposes. It traditionally provides fast access to specific records. Secondly, it is used if possible to limit the extent of a CAFS-search. This second purpose may weigh in the design of the primary-key format.

4.412 Variable-length Records

CAFS will always be able to perform searches on the fixed-length part of the data as these records are never fragmented in ISAM-files. However, several caveats still remain with respect to variable-length records.

Lengthening records may well have a severe effect on the amount of overflow in the file and slow-down the CAFS search. QUERYMASTER does not currently use the CAFS counting hardware when counting variable-length records. Finally, QUERYMASTER's ability to handle the variable-length part of a record is limited at the moment.

4.42 IDMS Databases

The IDMS-design process is today affected by the availability not just of CAFS, but of APPLICATION MASTER, QUERYMASTER, REPORTMASTER and the need to reorganise the database from time to time. As these products improve in their ability to handle IDMS databases, their effect on the design-process will diminish. Nevertheless, there will for some time presumably be some residual set of IDMS features whose exploitation can reduce the applicability of other products.

The topical goals of information-systems and the recent availability of new production-tools are changing the objectives of the design process. Simpler database designs are likely to serve many of these new objectives.

4.421 Analysis and Logical Design Implications

IDMS supports pre-defined regular enquiries, identified during the analysis phase of a project. These are then expedited by the data base design. One of the major limitations of this approach in the past has been the ability to predict enquiries of an unspecified and ad hoc nature. This aspect has traditionally either taken up much analysis time or has been ignored.

In anticipating the use of CAFS/QUERYMASTER, the analysis phase need only consider the main business processes in detail. The management information questions need not be determined in advance. However, in order to achieve this situation, there are several factors which must be considered during analysis and design.

In principle data analysis and data design should be simplified, since although the same data is involved fewer processes need to be considered. This should lead to the situation where the logical schema design will closely mirror the entity model produced during data analysis. This should result in fewer record and set-types.

In most database systems, the implemented schema contains record-types and sets that do not represent entities and relationships in the data-model or in the real world. Typical reasons for this are:

1) the need to access data by multiple keys
2) the need for data-subset totals often causing maintenance problems
3) the need to shorten search-paths through large unsorted sets
4) the need to hold superchief records as access-points

CAFS is likely to remove the need for some of these extra record-types as it effectively replaces IDMS-navigation by searching.

In a traditional database design, several set-types are often implemented purely to satisfy low-volume ad-hoc enquiries. Again, CAFS may eliminate the need for them by substituting for IDMS-navigation.

By simplifying IDMS data-structures, the read-only CAFS unit curiously contributes to performance on the update side.

4.422 Physical-Database Design Implications

The previous section treating existing IDMS-databases should be read in conjunction with this one.

4.4221 Physical Clustering

CAFS searches records of one record-type. These will be stored in one IDMS-area and possibly several IDMSX-areas. Within an area, the records will reside in a specific page-range if this is defined and in the secondary-overflow space.

CAFS-searches can be limited to less than a full AREA if page-ranges and overflow-areas are consciously defined rather than by default. More thought should be given to the segregation of different types of data which need not be clustered together.

An archive-policy separating 'historical' from 'live' data contributes to the reduction of AREA, page-range and CAFS-search sizes. Similarly, the splitting of one record-type into 'heavily-used' and 'lightly-used' data can reduce searching times.

4.4222 Data-clustering

This section defines a 'Data-Cluster type' as:

1) an 'entry' record-type such as a CALC or DIRECT record, plus
2) record-types stored in VIA-sets with this entry record-type as owner

The standard approach in placing record-types in page-ranges is as follows:

Define a one-to-many mapping from Data-Cluster types to page-ranges

Thus, Data-Cluster types should not share page-ranges. A single Data-Cluster type may usefully map to more than one page-range by clustering member-records in another AREA.

4.4223 Data-Duplication

Databases were adopted among other reasons to enable consistent, topical and trustworthy data to be held on file. The typical physical duplication of data in many 'conventional' files had previously contributed in large measure towards the removal of these three key data-properties.

It is therefore thought that the objective of database-storage is to eliminate the physical duplication of data. It is not. The objective was and is to guarantee or at least expedite the integrity of the data.

Third normal-form analysis of the data removes any unnecessarily duplicated data. Curiously though, the relational model relies on the duplication of key-data to relate data in one table with data in other.

Databases are not in practice implemented at the physical level in strict third normal-form. Performance considerations dictate otherwise and CAFS may suggest further departure from TNF.

CAFS performs selection on physical rather than virtual records. There is therefore some advantage to be gained by propagating identifying indexes, alternative-keys and other important data down IDMS-hierarchies.

This data is likely to be static or slow-moving but clearly the integrity of the data is threatened. Strict control must be exercised over programs which change physically-duplicated data as there is no built-in IDMS-mechanism for 'simultaneously' and automatically changing physical copies of a data-item in a manner transparent to the programmer.

Controls provided by DDS and the subschema-mechanism must be used fully.

4.4224 Derived Data

Derived data, such as counts, totals and averages, can be derived either in-line or on-demand. CAFS may well enable the latter method to replace the former.

CAFS can also be used effectively for integrity-checking and auditing the database, again removing the number of functions which need to be designed and programmed into the core-system.

The use of CAFS-hardware totalling in the future by QUERYMASTER on IDMS will help considerably here.

4.5 Using RCI

RCI gives program access to CAFS capabilities. It has already been exploited by COBOL programs, by APPLICATION MASTER and by REPORTMASTER. More than one site envisages the provision of Viewdata services using BULLETIN and CAFS. No doubt, RCI will be accessed via COBOL by programs written in other languages. RCI will enable search-tasks to be performed online which were previously thought infeasible.

RCI eases system implementation by providing additional data/program independence and by removing much of the drudgery of file-handling. This implies that existing staff will code more quickly and less-experienced staff will be able to write programs more effectively. Systems will be more resilient to future changes.

The implications of RCI for database design seem at first sight to be exactly the same as those of QUERYMASTER and CAFS described above.

There are some initial pointers to good RCI/COBOL program design. First, the program must be able to monitor and react appropriately to various status and response-codes returned by RCI. The program is to RCI what the end-user is to QUERYMASTER.

The programmer should also note that CAFS-searches are probably being done with no locks applied. It is therefore possible for a concurrent-process to alter or even erase records which have just been passed back from RCI to the program. The program should not take RCI-returned data as a baseline for update; instead it should READ or FIND the data again.

In the TP context, a design balance must be struck between buffer-sizes, search-limits and time-out times. An RCI search cannot currently be restarted after the search-limit has been reached.

In the multiphase TP context, a performance comparison can be made between two approaches to searching:

1) perform the search in one phase and store the results in a hit-file
2) spread the search across phases, linking with SAVE/RESTORE commands

The first technique will spend less time RCI-searching and is recommended if the hit-file overhead is not considerable.

5.1 The four ICL field-trial goals

ICL management asked for four areas to be particularly studied by the CAFS-system field trial sites. These were:

1) Performance
2) Compatibility and co-existence
3) Ease of design
4) New opportunities

It should be stated at the outset that the likely success in achieving the four ICL objectives in any depth would be dependent upon the successful delivery of the pre-release software products that would allow the trial to take place. An ambitious schedule was set during which virtually the whole of the ICL main-line software product range, from VME/B KERNEL at the bottom through to QUERYMASTER at the top, was to be issued and field-trialled in a 26-week period. Most of the software products were scheduled to go through at least two releases and many of the releases were integrated with other activities such as the APPLICATION MASTER field-trial. In short the field-trial sites were going to be subjected to a new issue of at least one major piece of software every two weeks. It must be said that there was a great deal of scepticism as to whether:

1) ICL could sustain the scheduled delivery of the software
2) the field trial sites would be able to sustain the rate of change if ICL succeeded on its part

However, most prospective users of CAFS were pleased to see that ICL seemed prepared at last to back CAFS after years of apparent lack of top management commitment and, sceptical or otherwise, they were prepared to put their backs into the field trial.

Unfortunately, however, much of the scepticism regarding software deliveries was justified. Serious delays in software occurred and trials were much curtailed relative to the plan.

5.11 Performance

This may be looked at in two different ways:

1) Reliability of hardware and software
2) Increase in throughput and reduction of load.

With regard to the reliability of hardware and software all sites contacted reported a very high level of performance with few or no problems experienced. This was well in excess of expectation for a new product with massive software changes. Regarding the effect of CAFS on the timing of specific jobs all the work done was by using QUERYMASTER and only one site had experimented on IDMS files. A performance boost of between 6 and 60 depending upon circumstances was typical for those cases where CAFS facilities were utilised. Response times appeared to increase linearly with the number of concurrent CAFS searches being undertaken on the same CAFS unit.

Most users found it disconcerting that the performance boost given to queries was so enormous that very uneven responses [qv 2.31] could be obtained by end users depending upon whether their query caused CAFS to be utilised or otherwise. It was felt that some warning should be given by QUERYMASTER if CAFS was not going to be utilised to satisfy the query so that end-users could choose to abort/rephrase their query.

A reduction of load on the OCP in the order of 90% to 99.94% had generally been experienced when CAFS was utilised. This had caused some sites who charge their users to seriously reconsider their charging structure and/or the level of ad hoc enquiry that could now be contemplated but which was previously not economically justifiable.

5.12 Compatibility and Co-existence

With regards to compatibility most field trial sites found that within a very short time of hand-over they were able to use CAFS on existing applications with little or no changes, particularly where the files were in ISAM format. ICL had produced a product and its matching software whose presence in the user's system did not require changes to existing applications and which stood a very good chance of enhancing many of those existing applications with either no changes or, alternatively, small changes to them.

Regarding co-existence, most field trial sites felt that they needed to do more work in this area to establish the effect of CAFS searching upon co-existing TP applications. As originally delivered, CAFS searches lock out other disk activity on that channel while up to ten tracks are searched. It is felt that this may well have an adverse effect on TP response and at best produce a more lumpy, less predictable, TP service. Since the original software release, the constant '10' has been parameter-ised and is defined at system set-up time. This gives the user's Support Group more control over evenness and speed of TP response at the expense of CAFS response. ICL is considering ways of making this parameter more dynamic in future releases of VME and the 'lock-out' effect may well be absent on DM1 disc-systems.

The consensus view of field trial sites is that a further sizing parameter has been introduced by CAFS which is as yet not clearly understood and in any event will ultimately depend upon individual sites' profiles of work. Control facilities are being provided by ICL but where these cannot provide the desired result the CAFS and TP environments can be kept entirely separate by copies, duplexing etc. In the end, with declining costs of disk store and an environment moving towards unattended operation, this may well not be as expensive as at first it seems.

5.13 Ease of Design

Very little had been done by the field trial sites in this area. This is mainly due to the fact that CAFS facilities in IDMS were delivered relatively late. Even if these obstacles had not been in place it is debatable whether any site would have done both a CAFS and non-CAFS design to produce a direct, measurable, comparison.

Most sites felt that there were cases where designs would prove easier but this was as yet still an unsubstantiated claim for CAFS.

5.14 New Opportunities

The field trial sites considered that the reduction of cost and increase in speed given to QUERYMASTER by CAFS represented a new opportunity to provide management support information that would previously have been ruled out on the basis of cost. This was further enhanced by the fact that there was a good chance that in providing the formerly unprovidable there was a good chance that systems and programming resources would not be required once the User Views had been set up.

The field trial did not expose any new application system that would have previously been shelved without CAFS though doubtless such applications do exist.

5.2 Field Trial Sites

The field trial sites included:

British Telecom
North Thames Gas Board
Severn Trent Water Authority
A South African Site
Southern Water Authority
W H Smith

5.21 British Telecom

This site was using CAFS on one of its sites as an experiment for the purposes of evaluation of that product. It was preparing a demonstration from a billing system comprising 200,000 customers and had extracted 1/12 of the data into 2 ISAM files of 6Mbytes. Searches comprising 1 or 2 joins were taking typically 5 seconds. All testing had been on a lightly loaded machine.

The previous policy of cost justifying one-offs would now require to be reviewed in the light of experience so far. This site felt that the limitations on QUERYMASTER were more significant than those on CAFS.

Reduction in the OCP utilisation meant that questions not askable before could now be contemplated.

The site was vague as to the quantifiable cost benefits of CAFS and felt that it needed to do work in this area, mainly on user access but also programmer productivity.

5.22 North Thames Gas

This user has made many presentations on CAFS to User Groups and is fairly well documented elsewhere in minutes etc. Most of the work it has done is in the area of performance and no work has been done on design implications. Its experiences have reinforced its belief in the product.

This user also felt that QUERYMASTER limitations were more important than CAFS limitations and also felt that providing a simple diagram of the User View of IDMS files limited the end-users' use of QM/CAFS.

The approach to using IDMS had not fundamentally been changed by CAFS but databases were currently designed for efficient retrieval rather than update and this would probably reverse. It was felt that CAFS would go some way towards removing the invisible backlog that nobody knows about at the present time.

Detailed performance results from North Thames Gas are quoted in Chapter 6.

5.23 Severn Trent Water Authority

All performance work had been done on ISAM files and this had involved some conversion from serial files. To make the most of the product searching of serial files would be helpful. Considerable increases in performace had been noted by end-users who before were using QUERYMASTER alone.

Three designs for new systems had been considered. Of these only one had looked promising with regard to productivity increases. The remaining two systems were relatively simple with update paths and retrieval paths identical. However this user felt that there were possible saving in delaying conversion of conventional ISAM systems to IDMS.

No serious degradation of TP response had been noted given the co-existence of CAFS and TP.

5.24 A South African Site

Tests were performed on both an ISAM file and an IDMSX database. The ISAM file used was the 123Kb file set up by the QUERYMASTER installation test. This size of file is really too small for a valid test of CAFS performance and on average only marginal benefits were obtained using CAFS.

Comparison tests were performed on two IDMS areas, one of 32Mb and one of 2Mb. The tests either did not use CAFS, used CAFS or used CAFS in conjunction with a TPMS/AM service simulating six messages per second.

Tests of QUERYMASTER running concurrently with TPMS were also done with and without CAFS. The TPMS service consisted of a TP terminal simulator run of 2088 messages being input from 12 virtual terminals as quickly as possible using 4 AVMs. The QUERYMASTER session was accessing the same database that the TPMS service was updating. As was expected the test showed that concurrent CAFS access to the database used by TPMS had a severe effect on TPMS response times.

The tests performed showed that using CAFS, it was possible to search at around 800Kbytes/sec without using appreciable OCP power to do so. Ratios of search times with and without CAFS were found to be approximately 8:1. However, non-CAFS searches in these tests on a dedicated DCU2 achieved 45-50 IO's per second.

5.25 Southern Water Authority

This site had bought a single CAFS unit as part of a new dual 2966 replacement system with the objective of evaluating CAFS. There were considerable problems in implementing the 2966 due to problems experienced on the NPS on the new 2966. This, together with the late arrival of release 8.00 of VME/B, meant that approximately 12 weeks of the trial were lost. On the balancing side when release 8.00 was received CAFS was introduced into a live (ie **all** work) system with full target software for the first time.

This customer is pleased with the reliability and performance of the total system and in view of the increased use of QUERYMASTER on this site is looking at CAFS as a means for delaying an OCP update and as a means of extending existing systems.

It feels that there are very large potential savings to be made in two major systems which are likely to be started this year and has, as a result of a successful trial, ordered full CAFS connectivity for all FDS filestore occupying sixteen drives.

5.26 W H Smith

This site is currently also a user of CAFS.800. They felt that conversion from CAFS.800 to CAFS-ISP was not good but accepted that this was not a CAFS problem.

This site was also particularly hit by the lack of a COBOL interface to CAFS as plans had been made to use CAFS in a POS application with very large volumes and for which conventional processing would not permit acceptable responses (new opportunity area?).

It felt that the CAFS hardware had lived up to its promises but was particularly concerned by late delivery of software.

It did not, in the light of the field trial experience, regard CAFS as a productivity aid and was concerned that ICL had not yet advised on the effect of CAFS on TP response times and on how to lay out the physical filestore configuration.

5.3 The RCI Field Trial

The Relational CAFS interface has been used extensively on a number of sites. The field-trial has demonstrated that RCI meets its objectives and provides facilities as intended. The objectives were:

1) to provide COBOL access to CAFS-searching capability
2) to provide a relational read-only interface to data
3) to simplify the programming task
4) to provide additional program/data independence
5) to provide text-matching and processing facilities

Although no site-specific quotes are available as yet, there is some early quantification of RCI's capabilities.

RCI/CAFS searching was found to be over eight times faster than the previous COBOL-program search. It is perhaps too early to regard this as a 'typical' ratio but there was nothing in the trial to suggest that a factor of eight might be at one or the other extreme.

RCI certainly simplified programming. It was found that it eliminated much of the drudgery of file-handling and data-reformatting; a 30% reduction in coding might be expected. RCI in particular helped junior programming staff write IDMS programs quickly. It did this by eliminating the need for much IDMS DML and specifically by eliminating the use of 'IDMS-currencies' and the 'one-currency' bottleneck.

While most field-trial sites were primarily interested in program-exploitation of CAFS, the increased program/data independence and the data-joining relational interface were appreciated by the sites involved. QUERYMASTER was used for prototyping RCI usage as intended.

The field-trial sites also commented favourably on the short learning-curve needed for COBOL/RCI programming. RCI syntax fits with both QUERYMASTER and COBOL syntax; programmers at all levels treated RCI as another file-handler. One site wrote and successfully ran its first COBOL/RCI program in half-an-hour.

RCI has been extensively exercised in an ICL staff-training workshop which demonstrated RCI in various contexts. RCI modules were called from APPLICATION MASTER and REPORTMASTER programs and accessed IDMS and RECMAN files in TP and batch. Experiments were also carried out to compare the performance of single-phase and multi-phase RCI searching.

All the sites who actively field-trialled RCI have expressed a wish to take RCI on license when it becomes generally available.

CAFS-ISP has been integrated into VME to achieve a radical improvement in searching speed. 'Performance' is the chief goal of the CAFS system.

'Ease of design' and 'compatibility/coexistence' are supplementary goals ensuring that users can exploit CAFS quickly and easily. Examples of such use are given in Section 4.1.

CAFS-ISP creates new opportunities merely by increasing searching speed by an order of magnitude. Let us suppose that four CAFS engines are available on the mainframe and that an 'order of magnitude' equals 'ten'. Then in a balanced system, information can be retrieved ten times faster and the number of end-users that can be supported can grow by a factor of forty.

Searches that were not feasible or not cost-effective become possible in batch or even online.

In this chapter we assess the potential/actual performance of the CAFS system and examine the main parameters which affect it. We conclude with some benchmark results and a sizing exercise.

Performance-boost is governed by:

1) the CAFS data-delivery rate
2) the CAFS contribution in servicing the end-user's enquiry
3) the speed of the non-CAFS service

The approach taken in this chapter is to concentrate on simple examples. Only in this way can the effect of each factor be appreciated. We examine the performance of CAFS searches on a dedicated machine and in the context of multiprocess activity.

6.1 Basic Parameters

CAFS intercepts and filters data as it is channelled from the disc-drive to the disc control unit. The following parameters therefore always affect the data-delivery speed on a CAFS system:

1) DS - nominal delivery rate of the disc-volume involved
2) BF - blocking-factor; the proportion of the track used by blocks
3) FF - fragment-factor; governed by the division of the CAFS-search into fragments
4) PF - packing-factor; the proportion of the data-blocks occupied by the records relevant to the CAFS search

The basic upper limit on searching-speed is therefore:

File-search rate $\leq$ DS * BF * FF Mbytes/sec
Data-delivery rate $\leq$ DS * BF * FF * PF Mbytes/sec

Other influences which will subtract from the data-rate as perceived by the application program or the end-user include:

1) Head movement cylinder-to-cylinder
2) The fragmentation of the file into file-extents
3) The management of record buffers in CAFS and mainframe
4) The effect of the multiprogramming context
5) The front-end effect of a communication system
6) Overflow in ISAM files

In fact, the total system serving the end-user's cognitive process is:

[End-user] --> [Comms System] --> [VME] --> [Head Mover] --> [Channel]

It is not possible to model this total system here concisely or with clarity. The focus must be on the disc-channel subsystem 'server', the 'head mover' subsystem being separated as head movement is triggered in parallel with searching.

As with all sizing exercises, it is necessary to define the viewpoint of the sizer, stating clearly what system is being sized and what parameters are being considered.

6.11 The Disc Speed DS

For the MDSS family of drives, DS = 19616 * 60 bytes/sec = 1.17696 Mb/sec.
The current CAFS hardware will accomodate disc speeds up to 3.6 Mb/sec.

6.12 The Blocking Factor BF

This is the proportion of the disc-track which is used by blocks. It is determined by the block-size as follows.

Block Size (bytes)	Blocks/ track	Bytes/ track	Blocking -factor
2048	9	18432	0.939641
3072	6	18432	0.939641
4096	4	16384	0.835237
6144	3	18432	0.939641
9216	2	18432	0.939641
18432	1	18432	0.939641
2057	9	18513	0.943770
3155	6	18930	0.965029
4801	4	19204	0.978997
6447	3	19341	0.985981
9739	2	19478	0.992965
19616	1	19616	1.000000

A BF of 0.939641 is obviously likely and will be taken as 'typical'.
For this value of BF, DS * BF = 1.105920 Mb/sec.

6.13 The Fragment Factor FF

Each CAFS search is broken down into fragments. These will not cross cylinder boundaries, will not cross file-extent boundaries and will be no longer than 'N' tracks. 'N' is defined at system set-up time and is defaulted to 10.

'N' is chosen to strike a balance between out-and-out CAFS-search performance and TP-performance interference. The effect of the fragment-size on CAFS-search performance is shown below. If N=1, CAFS fragments are short tasks like current disc-channel tasks; the effect of adding a 'CAFS' load to the channel will be the same as adding a similar non-CAFS load. If N=10 say, the CAFS-searching effect on TP will be greater. The **additive** effect on TP response-times is proportional to N and the number of CAFS-searches running at the time.

The inter-fragment time is an integer number of disc-revolutions. It does however vary even within one CAFS search on an unmodified DCU/2. The inter-fragment time has been observed and averaged as:

A	1)	DCU/2 unmodified:	3.3 revolutions
B	2)	DCU/2 with standard modifications:	2.0 "
	3)	DSC as standard:	2.0 "

The fragment-factor therefore is determined by whether the disc-control unit is in state A or B and by the maximum fragment length. CAFS field-trial sites operated in condition A; B is now the normal condition.

Fragment Size	A	B
1	0.232558	0.333333
2	0.377358	0.5
3	0.476190	0.6
4	0.547945	0.666667
5	0.602410	0.714286
7	0.679612	0.777778
8	0.707965	0.8
10	0.751880	0.833333
20	0.858369	0.909091
40	0.923788	0.952381

Given DS = 1.17696 Mb/sec, BF = 0.939641 and a DSC or modified DCU/2, we can calculate some file-search speeds for 'typical' fragment-sizes:

Fragment Size	File-search Speed: Mb/s
5	0.789943
10	0.921600
20	1.005382
40	1.053257

6.14 The Packing Density PF

RECMAN file blocks and IDMS-pages will normally not be fully occupied by data even in the case of a single record-type file:

1) ISAM files are frequently multi-record type
2) IDMS DB page-zones/overflow are also frequently multi-record type
3) Each IDMS-page has header/footer information
4) IDMS areas have interspersed space-management pages
5) Packing density should by design be less than 100%:
    a) to lower overflow frequency in ISAM and IDMS
    b) to avoid space-management overhead in IDMS
    c) to anticipate changing volumes in dynamic data

It is worth noting that these parameters are already intrinsic to system-layout and system-design. They do not arise because of the introduction of CAFS and are not aggravated by CAFS.

Low packing densities do not subtract from the value of CAFS. This point was mentioned earlier [Ch 4]. Unused file-space and non-target data-types are the first examples of CAFS' effectiveness in filtering out disc-space which is irrelevant to the search and to any subsequent processing.

6.2 Resource Utilisation

We should consider here the resources used at design-time and at run-time. These include computer-system resource, DP-resource and end-user resource.

6.21 Design-time Resources

There is an intrinsic limit to the amount of design effort which should go into a system. Clearly, the designer cannot contribute to as-yet unidentified system requirements. However, he will often spend considerable effort on the more peripheral and volatile system facilities. CAFS removes this temptation to some extent by making serial searching a viable alternative to data-retrieval through some complex access-infrastructure.

End-users are playing a larger role in system design in partnership with DP staff. CAFS therefore saves both end-user and DP time at the design stage.

6.22 Run-time Resources

6.221 Computer-system Resources

6.2211 The Mainframe OCP Resource

CAFS will relieve the OCP of 0-100% of its work in searching and selecting data. It is not the case that CAFS either does nothing or everything.

A number of benchmarks have been run during the field-trial and these will help to profile the contribution that CAFS can be expected to make in each case. These are examined in Section 6.3 below. The working party is indebted to the field-trial sites for permission to quote these figures.

CAFS has been observed in these trials to save up to 99.94% of the OCP resource.

6.2212 The Disc-Channel Resource

It is not generally appreciated that a search uses less disc-channel resource with CAFS than without. In a recent typical study on a 9 block/track file, 'channel-occupancy' was halved by the use of CAFS. This reduction will be achieved because:

1) whole tracks rather than blocks are scanned at a time
2) multiprogramming has less effect on head-positioning

6.222 End-user Resources

Elapsed time costs. It costs the organisation in end-user's waiting time; an executive is probably costing his company £1 per minute.

If the CAFS system is supporting the organisation's interface with the customer, elapsed time subtracts from the service offered to the customer. This means lost business or lost customer-satisfaction which also means lost business.

Elapsed time costs in the loss of timeliness of the information. It costs by interrupting rather than supporting the end-user's cognitive process.

6.3 Early Experiences

All results were achieved on 2966 systems with DCU/2s. The DCU/2 modification mentioned in Section 6.13 was not available. In all cases, the CAFS-search fragment was set to be ten tracks. On systems with OCPs less powerful than the 2966, the benefit of CAFS would be even more marked.

6.31 North Thames Gas

North Thames Gas ran a number of benchmarks to confirm the basic search-power of CAFS and to run a number of CAFS/QUERYMASTER scripts in parallel.

The search-power tests were first run on a 3.5Mb ISAM 'Customer' file containing customer-type records each having a customer-name field. The queries used were:

A) LIST COUNT WHERE CUSTOMER-NAME STARTSWITH A_1 or A_2 or ..... or A_s
B) LIST COUNT WHERE CUSTOMER-NAME CONTAINS A_1 and A_2 and .. and A_s

'H' denotes the number of hit-records found in the search.
's' denotes the number of search-criteria and was varied here from 1 to 15.
'R"' denotes the response-time in seconds.

		---NON-CAFS---		----CAFS-----		-Reductions--		
s	H	R"	KPLI	R"	KPLI	R"	PLI	
1	30	141	111,110	4.3	341	97%	99.7%	\
4	162	146	118,196	4.4	515	97%	99.6%	|
8	258	149	122,712	4.5	752	97%	99.4%	| **A**
12	366	153	126,864	4.6	997	97%	99.2%	|
15	438	154	129,516	19.7	18,852	87%	85.4%	/
1	660	328	453,984	15.4	11,362	95%	97.5%	\
4	96	443	674,513	13.9	11,066	97%	98.4%	|
8	0	468	714,451	14.2	11,753	97%	98.4%	| **B**
12	0	482	737,730	14.5	12,456	97%	98.3%	|
15	0	499	755,199	15.0	12,996	97%	98.3%	/

Note that with the 'STARTSWITH' query, CAFS-response time jumped when the number of selection-criteria increased from 12 (or 14 in fact) to 15. This is because the number of selection criteria 'OR'd together exceeded the number of CAFS-key-channels available.

Under these circumstances, no record could be excluded in a single CAFS pass. QUERYMASTER therefore does not attempt to evaluate selection-criteria using the CAFS hardware but continues to pick out the customer-records using CAFS.

A more intelligent approach which can be instituted 'manually' is to trigger two CAFS-searches using half of the selection criteria in each pass; beware however of generating duplicates by 'hitting' records twice. In 'production' queries, QUERYMASTER macros exploiting MAKE, EXTEND and SORT facilities can be used to advantage to preserve a simple end-user interface while taking advantage of this two-pass approach.

Note that CAFS does not evaluate a 'CONTAINS' condition. 'CONTAINS' should be expressed as 'STARTSWITH' wherever possible. Nevertheless, CAFS makes its performance-contribution by picking-out the customer-records which are here only some 1% of the total file content.

There were three sets of tests on IDMSX databases:

A) 64Mb file: LIST COUNT WHERE STREET STARTSWITH A_1 OR ... OR A_s
B) 64Mb file: LIST COUNT WHERE DATA = D_1 OR ... OR D_s
C) 315Mb file: LIST COUNT WHERE DATE = D_1

		--NON-CAFS---		----CAFS-----		-Reductions--		
s	H	R"	KPLI	R"	KPLI	R"	PLI	
1	2	237	73,527	81.0	373	66%	99.5%	\
4	18	229	73,390	73.0	220	68%	99.7%	| **A**
8	36	229	73,522	72.0	229	69%	99.7%	/
1	3	344	135,194	73.0	299	79%	99.8%	\
4	35	356	135,477	74.0	1,137	79%	99.2%	| **B**
8	96	357	135,681	75.0	1,767	79%	98.7%	/
1	29	2635	531,549	379.0	499	86%	99.9%	\
1	16	5429	1,587,702	366.0	2683	93%	99.8%	/ C

Here we notice lower factors in the R" reduction but OCP load-reduction similar to that in the previous ISAM benchmarks.

Some concurrency tests were also run by North Thames Gas. First, the multiple CAFS/QUERYMASTER scripts and secondly the TP & CAFS/QUERYMASTER combination:

1 QUERYMASTER non-CAFS script	8912 seconds	
1 CAFS/QUERYMASTER script	379 seconds	
3 CAFS/QUERYMASTER scripts	896 seconds	
5 CAFS/QUERYMASTER scripts	1551 seconds	
1 TP script	206 seconds	
1 TP script with 1 CAFS/QM script running	357 seconds	(+ 151")
1 TP script with 5 CAFS/QM scripts running	776 seconds	(+ 570")

Here we can see the overlapping of OCP and disc-channel activity. TP responses will vary by quantum amounts governed by the number of CAFS-searches going on. A multiple-I/O TP transaction will find its I/Os interleaved with the fragments of any present CAFS search. The CAFS-search-induced delay to the TP task is therefore proportional to the number of I/Os in the TP task and to the length of the CAFS-search fragment.

6.32 Southern Water Authority

The Authority ran a series of tests to confirm the search power of CAFS, investigate concurrency behaviour, vary the CAFS-fragment size and monitor performance as the CAFS system developed. Below are the results of searching a 1.7M-record ISAM file of 63-byte records:

		-----NON-CAFS----		----CAFS----		--Reductions--	
S	H	R"	KPLI	R"	KPLI	R"	PLI
0	0	12292	11,568,000	137	7,026	98.89%	99.94%
4	?	5844	6,617,000	139	3,719	97.62%	99.94%
4	?	7020	8,598,000	205	53,690	97.08%	99.38%

The file had a block size of 2048 bytes and was on an FDS640. Here are the results of varying the CAFS-fragment size for two enquiries:

A) LIST COUNT PREV-A
B) LIST TOTAL WHERE SUBJCODE = "***A" AND
    (SUBJCODE = "aaaa" to "bbbb" OR SUBJCODE = "cccc" to "dddd"
    OR SUBJCODE = "eeee" to "ffff")

Fragment Size	-----Enquiry A----- Time in Seconds	Records/ Second	-----Enquiry B----- Time in Seconds	Records/ Second
1	400	4351	468	3719
2	251	6934	317	5490
3	210	8288	275	6329
4	176	9889	244	7133
6	154	11301	223	7805
8	140	12432	211	8249
10	133	13086	-	--
15	127	13704	208	8368
20	117	14876	-	--
40	113	15402	200	8702

Enquiry A's performance correlates closely with Section 6.13, column A.

6.33 CAFS v non-ICL Query Systems

The figures above all compare CAFS/QUERYMASTER with QUERYMASTER alone. This section illustrates the power of CAFS by comparing it with non-ICL enquiry systems.

On the 25th January 1984, ICL demonstrated a direct comparison of CAFS and two IBM-architecture enquiry systems. These were:

ICL	2966	1.1Mips	**CAFS/QUERYMASTER**	Overall Cost £ 700,000
IBM1	3033-AP	8 Mips	Easytrieve (Pansophic)	Overall Cost £?,???,???
IBM2	3081-G	10 Mips	Dataquery (Datacom)	Overall Cost £3,500,000

Three terminals in the Putney Management Systems Centre were linked through a System 25 to the three mainframes in the trial. The System 25 managed all protocol matters using C03 to ICL and SNA to both IBM1 and IBM2.

The same file was established on all three machines. It was a multi-record type index-sequential file of 23Mb and contained some 130,000 records. ICL asked each bureau to set up the file in exactly the same way. In the event, staff at IBM2 separated out the different record-types to separate files giving the effect of a relational database. Thus, while ICL and IBM1 searched the whole file, IBM2 only searched 30,000 records.

The same enquiry was set up on all three terminals. It asked for a list of all products where stock was above a threshold level, where no orders had been received since a given date, and where various other conditions prevailed. This is a very typical simple enquiry.

The enquiry was submitted simultaneously on all three systems and a triple clock on an ICL PC showed the response times for each system. The table below shows the results together with results logged during the preparation of the demonstration. The response times in seconds are reasonably consistent. Mainframe seconds/enquiry (Msc/e) and instructions/enquiry (Ins/e) were rather more difficult to establish because the three systems measured this in slightly different ways. It seems fair however to quote the figures below. Again, the charging systems of the bureaux varied but the cost per enquiry has been established.

	Response times (secs)								**Msc/e**	**Ins/e**	**Cost/enquiry**
ICL	49	48	44	49	49	51	47	48:	0.8	0.9	£ 10
IBM1	204	209	220	219	227	219	218	239:	16	128	£???
IBM2	235	138	201	148	183	211	162	184:	33	330	£160

It is possible to measure the resource required for enquiries in terms of:

1) Initial system cost
2) Seconds of OCP time or MIPS
3) Seconds of lost executive-waiting time
4) Charge per enquiry

It does not matter what success criteria are chosen: the CAFS/QUERYMASTER combination scores dramatically.

The demonstration was independently audited as a fair comparison and on balance the conditions were thought to favour the two IBM systems. In the opinion of the auditors:

> "ICL's CAFS product provides significant benefits for an average user performing ad-hoc retrievals on a substantial database when compared to an IBM alternative."

6.4 Disc-Channel Subsystem Queuing Models

The question "How does CAFS searching affect TP" is often asked. We have already given in Section 6.13 above a simple answer to this question. The **additive** effect is proportional to the fragment size 'N' and the number of CAFS-searches being executed concurrently.

For example, if a TP-transaction requires 5 I/Os, N=10 and there is 1 CAFS-task in play, then we can expect that transaction's response-time to be 5 * 10 * 1 * 0.017 = 0.85 seconds longer than if the CAFS task was not there.

The end-user is likely to be affected more by the variability of the response than by the increased mean-response-time induced by CAFS searches. He will experience the **additive** rather than the **multiplicative** effect of CAFS on his response-times.

We therefore add only the fundamental 'pint-pot' queuing observation rather than the results of a queuing model or simulation of the disc-channel:

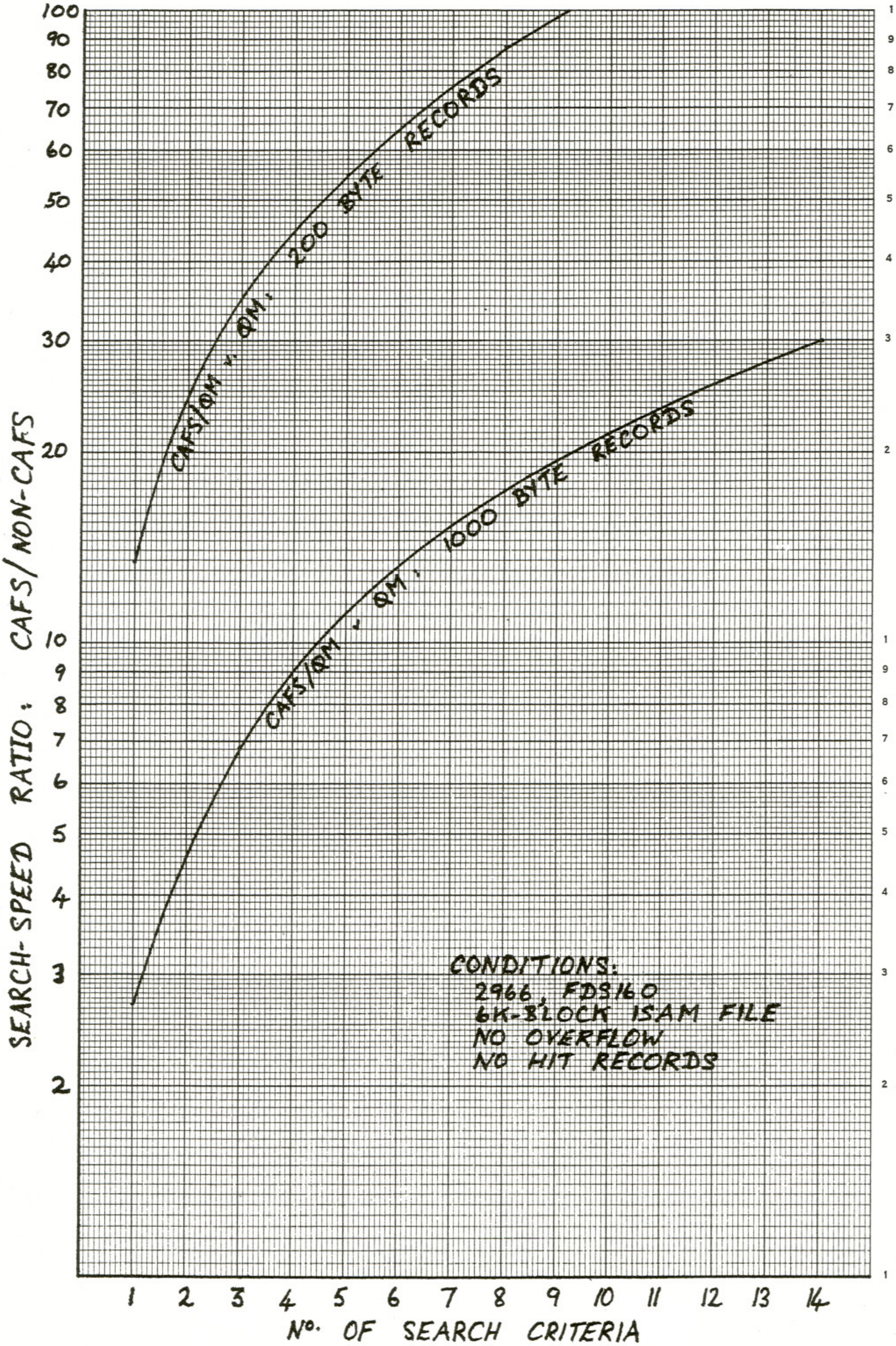
> In all queuing situations, as the load rises to capacity, mean and variation of response time rise hyperbolically (ie dramatically).

6.5 A Sizing Exercise

The next page shows ICL's prediction of the way CAFS would speed up QUERYMASTER searches. It models a 2966/CAFS combination searching a 6k-block FDS160 ISAM file with no overflow and encountering no hit records.

The increased performance available from CAFS will be greater if the mainframe is less powerful than a 2966.

The reader may be interested to compare this prediction with the actual results above and with his own experiences.



SEARCH-SPEED RATIO: CAFS/NON-CAFS
No. OF SEARCH CRITERIA
CAFS/QM v. QM; 200 BYTE RECORDS
CAFS/QM v. QM; 1000 BYTE RECORDS
CONDITIONS:
2966, FDS160
6K-BLOCK ISAM FILE
NO OVERFLOW
NO HIT RECORDS
100
90
80
70
60
50
40
30
20
10
9
8
7
6
5
4
3
2
1 2 3 4 5 6 7 8 9 10 11 12 13 14

6.6 A PSAM Benchmark

The following benchmark was run by ICL and demonstrates the way in which PSAM moves the labour of data-searching from the OCP to the disc-channel.

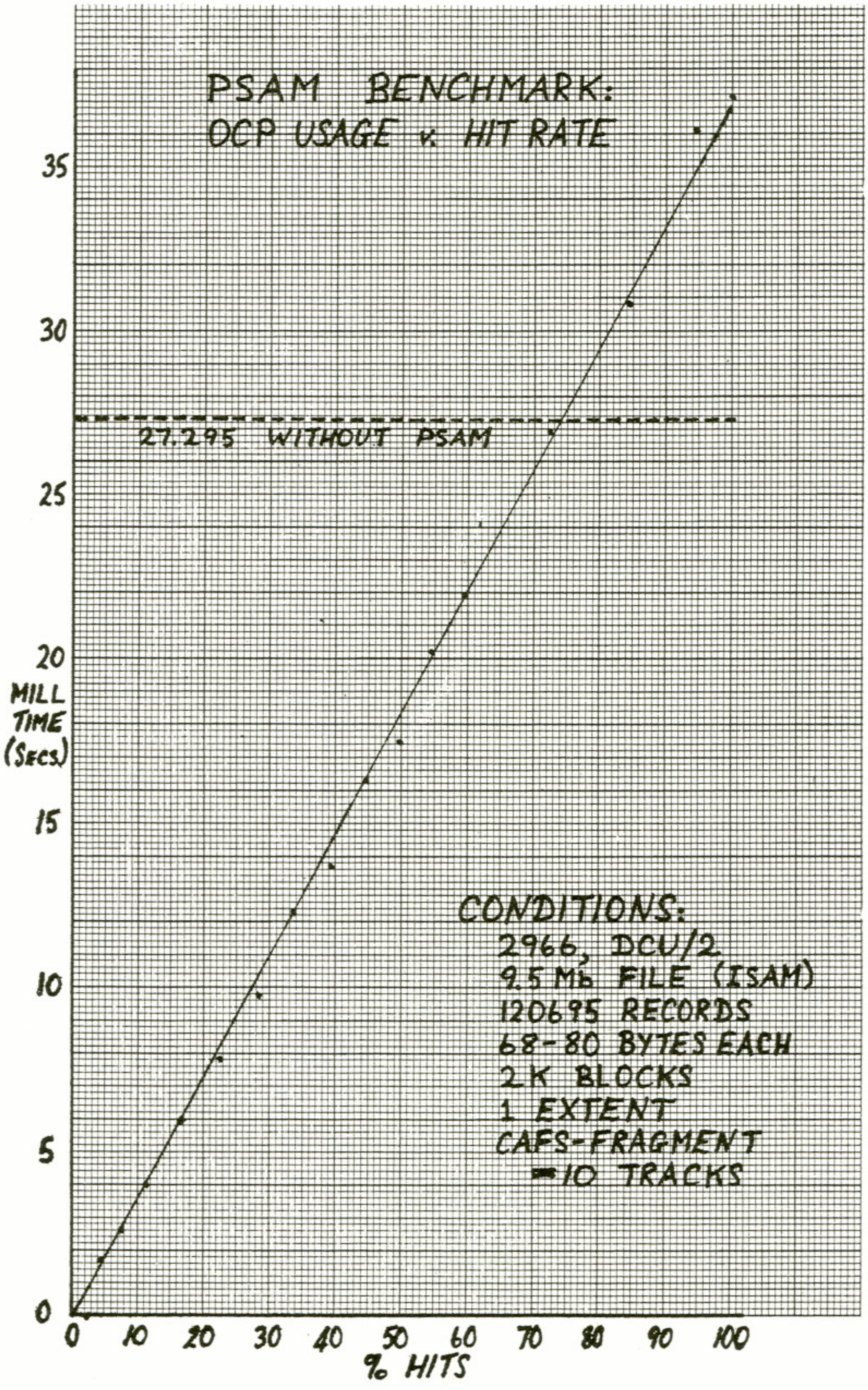
The parameter determining the residual work to be done by the OCP is clearly the hit-rate of the search. No other parameters are though to have a significant effect.

For completeness however, here are the full details of the benchmark. A 2966 ran a search-only program which scanned a 9.5Mb ISAM file. The file consisted of 120695 68/80-byte records in 2k blocks in a single extent. The records were packed at 90% density with no overflow. CAFS searching was done via a DCU/2 with a default CAFS-search fragment-length of 10 tracks.

Unassisted by CAFS, the program consumed 27.295 seconds of OCP-time regardless of the hit-rate. Via the PSAM interface, the program performed as follows:

Hit Records	Hit rate	Mill Time (Secs)
232	0.2%	0.222
5455	4.5%	1.735
8955	7.4%	2.622
13363	11.1%	4.017
19841	16.4%	5.970
27175	22.5%	7.810
34138	28.3%	9.767
40901	33.9%	12.273
47587	39.4%	13.692
53867	44.6%	16.315
59917	49.6%	17.426
65686	54.4%	20.152
71596	59.3%	21.932
86999	72.1%	26.927
101533	84.1%	31.619
113935	94.4%	36.116
119734	99.2%	37.031
120695	100.0%	37.110

As can be seen from the following graph, the results demonstrate a consistent linear relationship between the hit-rate and the residual OCP load. The OCP break-even point in this case comes at 73%.


PSAM BENCHMARK:
OCP USAGE v. HIT RATE
27.295 WITHOUT PSAM
CONDITIONS:
2966, DCU/2
9.5 Mb FILE (ISAM)
120695 RECORDS
68-80 BYTES EACH
2K BLOCKS
1 EXTENT
CAFS-FRAGMENT
=10 TRACKS
MILL TIME (Secs)
0
5
10
15
20
25
30
35
0
10
20
30
40
50
60
70
80
90
100
% HITS

These recommendations contribute to the initial goals of performance, ease of design, coexistence and opportunity; they also range further afield.

7.1 CAFS/QUERYMASTER Facility Enhancement

Existing computer users have made a large investment in the storage of data and in programs to access their data. In so doing they have made full use of all the facilities available in existing file handling and database software products. This operational data will form the basis for most users' decision support developments.

If the existing CAFS/QUERYMASTER system is to be an effective component of users' Decision Support Systems, several aspects require improvement in the short term. Most of the following recommendations are QUERYMASTER specific but some of the improvements come from other parts of the CAFS system.

CAFS/QUERYMASTER should support Hash-random and Serial RECMAN files. It could also provide a read-only interface to PDS. A serial-file interface to CAFS will also allow CAFS-searching on QUERYMASTER hit-files.

QUERYMASTER should provide the same user-interface to the variable-length part of a variable-length record as to the fixed-length part of the record.

QUERYMASTER should use CAFS-hardware facilities for counting and totalling wherever possible. These facilities are not currently used when searching variable-length ISAM records or IDMS databases. The hit-rate is the parameter which determines the added response time: for the auditor, the hit-rate is usually 100%.

CAFS/QUERYMASTER should not be affected by relocated and fragmented IDMS records. It should always provide correct answers at a speed largely unaffected by the presence of such records.

QUERYMASTER should use CAFS to search all data items which can be described by COBOL PICTURE clauses. At present it will not deal with floating point data and there are restrictions on some other numeric data items.

In IDMS data structures it should be possible to use CAFS to search record and set index blocks. It should also be possible to use these indexes to limit a CAFS-search to specific pages. This would be useful where further evaluation of the target-records is needed by CAFS.

As more textual data is stored in computer systems the potential for large record sizes will increase. In this context the current QUERYMASTER record-length restriction of 4k bytes should be increased to at least the current CAFS restriction of 32k.

QUERYMASTER should exploit CAFS when evaluating a 'CONTAINS' condition in the query. This will obviously be done in the case of small data-field by implementing the 'CONTAINS' using elastic and don't-care characters. Can a CAFS key-channel search in a future version for a string-match in more than one position in a data-field?

7.2 Performance

Many of the above recommendations bring a performance as well as a facility benefit.

CAFS performs search and selection on physical records at 'hardware' speeds; the end-user enquires on virtual records. The data must be stored in a number of physical records for reasons of performance and integrity. QUERYMASTER joins together data from several physical records to present the user with his virtual records. This joining processes is a highly time-consuming one.

The recommendation here is that ICL should follow-up CAFS with hardware to replace the software performing the join operation. There have already been references to such a machine in the ICL Technical Journal [13, 15].

The current CAFS data-retrieval speed is a function of the MDSS disc drives. The CAFS hardware itself will support disc speeds of up to 3.6Mb/sec. Disc-speeds above this will require a re-engineering of the CAFS engine but it is assumed that the relevant microchip technology is advancing at a faster rate than the disc technology.

The effect of such improvements in disc-performance will be to improve response times for CAFS applications and to reduce the impact of CAFS use on operational support applications. However, Parkinson's law applies. As increasing volumes of data are held on-line there will be a need for a significant improvement in performance just to maintain current levels of service. This requirement for much large volumes of data will lead to a requirement for more disc storage, more channels and faster access.

The large user investment in present generation discs, DCUs and CAFS raises a requirement for compatibility in these areas for all new developments. New discs must be capable of attachment to existing DCUs; new DCUs must support existing discs.

7.3 Coexistence and Control

It is essential that the user DP department be given the tools to control the effect of using CAFS on the high volume traditional DP work. It should be possible to prevent CAFS users overloading discs or channels used by a TP service (possibly by use of a priority mechanism).

It should be possible to insist that particular QUERYMASTER accesses can only be made when CAFS support is available.

When using duplex data it should be possible to constrain CAFS searches to one plex while TP reads use the other plex. TP updates should take priority on both plexes.

It should be possible to change the size of the maximum CAFS search-fragment more dynamically than at system set-up time. The value, currently defaulted to '10 tracks', should be set according to the workload of the moment. The pre-emption of long CAFS-tasks by shorter tasks requiring the same disc-channel would make this less critical.

End-user queries may take an unacceptably long time especially if CAFS makes little contribution. In such cases, the end-user should be warned in qualitative or quantitative terms. QUERYMASTER should have access, through the Data Dictionary for example, to volumetric data. The end-user should receive comforting signals that his query is being serviced and should be able to abort the query.

The CAFS end-user should be able to confirm easily that the CAFS-hardware is operational.

QUERYMASTER should warn of a possible mismatch of question and answer if it has a choice of access-paths to IDMS-held data.

7.4 Ease of Design

IDMS design involves many considerations, for example:

1) the choice of data-items, records, page-ranges, areas, files, disc-volumes and disc-channels
2) the mapping of data-items to records to page-ranges etc.

Data-definitions, together with data and transaction volumes and response-cost criteria are the input-data for physical-database optimisation. The lack of algorithms for optimising a full database-design may explain the lack of design-tools.

The design phase however is becoming an increasingly dominant phase of system-development because implementation is now more automated. CAFS merely introduces a few more design decisions.

Design-tools such as the DRAGON system [11] are now welcome and further applications of knowledge-engineering technology are anticipated. These must naturally interface to DDS.

7.5 Other CAFS-interfaces

QUERYMASTER is the first available interface to the CAFS hardware. The 'VME Statement of Direction' of November '83 [9] anticipates the provision of the RSI and PSAM interfaces in VME 8.11, part of SV211.

The general availability of the CAFS/COBOL interface RCI is eagerly awaited; interfaces to APPLICATION MASTER, DDS, PDS, REPORTMASTER, text and office products will also be vital in the future.

CAFS will be required wherever significant quantities of data are stored. In a distributed network this could be on mainframes, (2900 and, later, ESTRIEL) on distributed mainframes (DM1) and on file servers associated with local networks (DRS, PERQ).

7.6 Future Environments

New data structures will undoubtedly emerge as the storage of text and digitised voice/image is integrated with the storage of traditional computing data. It is essential that CAFS keeps in step with these developments. New document storage-format and international character-set standards will also have to be considered.

To achieve this ICL will need a co-ordinated approach to all developments, in both hardware and software. This approach will need to cover not only in-house products but must also be applied to collaborative ventures at a very early stage.

7.7 Other Hardware Engines

The use of hardware engines to provide new or improved facilities is undoubtedly a sound concept, as is the philosophy of transparent implementation supporting standard language features and existing interfaces. Generic processes first identified in software will be implemented in hardware; parallelism such as that found in CAFS will be a key feature. Other areas where this technique could be applied in the future are:

1) Digitisation and compression of voice/image data
2) Recognition and translation of alphanumeric characters embedded in digitised data
3) Processing of digitised data by comparison of content
4) Graphical display of data
5) Support for file transfers across a distributed network, including translation for alien data formats (e.g. from word processors) and provision of an integrity mechanism.

This is the first report produced by the ICLCUA CAFS SIG Working Party.

The working party intends to continue its leading-edge role with ICL's help and collaboration. It intends to produce a second report which will report on the field use and facilities of the CAFS system at the SV201, SV211 and SV212 levels.

8.1 Feedback Request

You, the reader, have an opportunity to direct and help the working party's efforts. Let us know, for example:

1) if you would like to join the working-party itself
2) how you are currently using CAFS and what your own impressions are
3) how you plan to use CAFS
4) what information you would like from the CAFS SIG or from ICL
5) your views of this report
6) how you would like to see the CAFS system developed
7) if you are a member of the CAFS SIG
8) if you would like to become (or nominate) a member of the CAFS SIG

8.2 Feedback Form

Please use the inserted form or copy the next page to input to the CAFS SIG working party. The form is not the basis for a CAFS questionnaire but the answers will help to give the right context for your key comments.

Please return this form to:
Guy Haworth, ICL (CAFS), Kings House, 33 Kings Rd., READING RG1 3PX

Name: Job Title:

Company: ICL Representative:

Address: Telephone Number:

Business:

Computer System(s) (OCPs, main/backing store, CAFS/non-CAFS channels):

Comments for the ICLCUA CAFS SIG Working Party:

Ed 2

GLOSSARY of ACRONYMS and TERMS

ADRAM	Alien Data Record Access Method
AM	Application Master
AV	Application View (RCI)
AVM	Application Virtual Machine
BCS	British Computer Society
BF	Blocking Factor
CAFS	Content-addressable Filestore
CME	Concurrent Machine Environment
DCM	Disc Control Module
DCU/2	Disc Control Unit
DDS	Data Dictionary System
DME	Direct Machine Environment
DML	Data Manipulation Language (CODASYL, IDMS)
DP	Data Processing
DS	Disc Speed
DSC	Decision Support Controller
EDS	Exchangeable-disc store
FDS	Fixed-disc store
FF	Fragment Factor
ICLCUA	ICL Computer Users' Association
IDMS	Integrated Data Management System
ISAM	Index Sequential Access Method
ISP	Information Search Processor
KC	Key Channel
LFU	Logical Format Unit
MIP	Misleading Index of Performance
OCP	Order Code Processor
PDS	Personal Database System
PF	Packing Factor
PLI	Programming Language Instruction
PSAM	Pre-selective Access Mechanism
QM	Querymaster
RAM	Record Access Method
RCI	Relational CAFS Interface
RECMAN	Record Manager
RM	Reportmaster
RP	Retrieval Processor
RSI	Restricted System Interface
RU	Retrieval Unit
RV	Relational View (RCI)
SCL	System Control Language
SEU	Search Evaluation Unit
SIG	Special Interest Group
STCC	SET-CAFS-CRITERIA (SCL for PSAM)
SV	System Version (software set)
TME	Transaction Machine Environment
TNF	Third Normal Form
TPMS	Transaction Processing Management System
UFED	Unified Fixed and Exchangeable Disc Controller (= DCU/2 or DSC DCM)
VME	Virtual Machine Environment

BIBLIOGRAPHY

DDS

[1] R00120/00 (second amended reprint) "VME 2900 Data Dictionary System (DDS 650)

IDMS

[2] R00154/00 (second amended reprint) "IDMS Part 2: Database Establishment (IDMS350/IDMSX350)"

[3] R00155/00 (second amended reprint) "IDMS Part 3: Using a Database (IDMS350/IDMSX350)"

[4] R00156/00 (second amended reprint) "IDMS Part 4: Database Programming (IDMS350/IDMSX350)"

[5] R00153/00 (second amended reprint) "IDMS Part 5: Database Design (IDMS350/IDMSX350)"

QUERYMASTER

[6] R00261/00 "Using DDS to Prepare a Query View (200 level)"

[7] R00260/00 "Using Querymaster (200 level)"

[8] R00246/00 "Running Querymaster (200 level) in VME"

ICL Statements of Direction

[9] VME (November '83)

[10] CAFS (August '84)

[11] "DRAGON: The Development of an Expert Sizing System", M J R Keen (1984). Database & Network Journal v14 no1.

VME

[12] R00074 "VME 2900 User Guide (SV211)"

ICL Technical Journal

[13] "The Content Addressable File Store, CAFS", V A J Maller: v1 no3 ('79) pp265-279

[14] "Personnel on CAFS: a case study", J W S Carmichael: v2 no3 ('81) pp244-252

[15] "Associative data management system", L E Crockford: v3 no1 ('82) pp82-96

[16] "Content Addressing - the key to Simple Flexible Design", A J Walker: pp84-96 of "Database Design Update" edited by G J Baker & S Holloway. Proceedings of the BCS Database Specialist Group Conference (April '84)

RCI

[17] R00241/00 "Relational Programming from COBOL (100 level)"

[18] R00251/00 "Using DDS to Prepare an Application View (100 level)"

[19] R00252/00 "Running RCI (100 level) in VME"

NB: these three manuals may be amalgamated

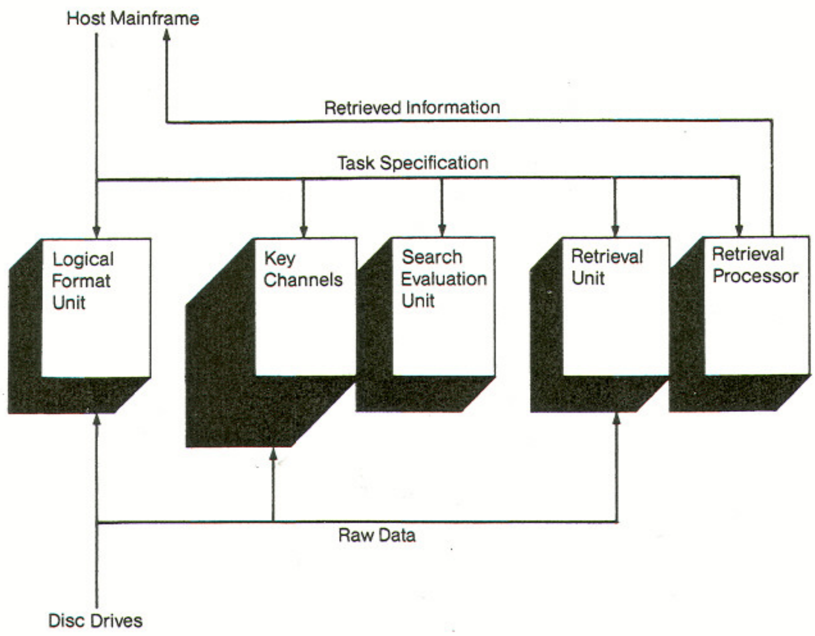
ICL

CAFS

Information Search Processing

CAFS-ISP is an intelligent hardware-based searching engine, available with ICL's 2966 family of computers and operating within the VME environment. It uses content-addressing techniques to perform fast searches of data or text stored on discs; all fields are equally accessible as search keys. Software in the mainframe generates a search task; the CAFS-ISP hardware performs the search, and returns the hit records to the mainframe. Because special hardware is used, the searching process is very much more efficient than searching performed by any software method.

- Content addressing
- Fast searching
- 16 key channels
- Precise or fuzzy matching
- Boolean logic—AND, OR, NOT and Quorum logic
- Counts and totals
- Finds maximum and/or minimum values of a stored data item
- Index sequential and serial files
- Searches IDMS and IDMSX databases
- Data and text



CAFS Information Search Processing

Disc Controllers

CAFS-ISP can be attached to the Disc Control Module (DCM) of either a DCU2 or a DSC disc controller. A DCU2 can have 1 or 2 DCMs, each of which may have an associated CAFS-ISP module; in this case the DCMs are mounted on one platter, with the CAFS-ISPs on an adjacent platter. A DSC consists of 1 DCM with 1 CAFS-ISP on the same platter; an extension module with another DCM and CAFS-ISP may occupy a second platter. (A platter occupies one half-cabinet.)

A single processor can support up to 6 CAFS-ISPs; a dual can have up to 8.

Discs

CAFS-ISP can be used with the ICL EDS80 exchangeable disc drives or the FDS160 and FDS640 fixed disc drives. These drives have a peak data transfer rate of 1.2 megabytes per second, and can be searched by CAFS-ISP at nearly 1 megabyte per second.

Data

CAFS-ISP can be used to search data of conventional format in ICL Serial and Index Sequential files, as well as in IDMS and IDMSX databases. To make text directly searchable by CAFS-ISP an additional data-type—'self-identifying format'—is introduced; text in this format can be included in any record.

Logical Format Unit (LFU)

This unit scans the incoming data as it arrives from the disc drives, and identifies critical points such as the start of a database page, the start of a record, the start of a relevant field. It also interprets the structure of text in self-identifying format. By passing control signals to the other units it instructs them where to find the elements appropriate for searching and retrieval.

Key Channels (KCs)

16 KCs are available. Their function is to match the user's selection criteria against the values in target records. A search task is assigned an appropriate number of KCs, each of which is preloaded with one search criterion. Under control of the LFU, all KCs scan the data stream independently and in parallel. As each relevant field is located the KC compares the user's target value with the value found in the record, storing the result (<, =, or >) for subsequent evaluation. Each KC can also be used to test whether its target field is present in a target record or not. Masks can be used to exclude irrelevant bits or characters from the comparison process, giving a very powerful 'fuzzy matching' capability.

Search Evaluation Unit (SEU)

The SEU decides whether or not a record is a 'hit', based on the stored findings of the KCs. The evaluation may use boolean logic, or quorum logic, or a combination of both. In boolean mode, individual KC findings are linked with the right mix of ANDs, ORs, and NOTs to give a definite decision on whether a record qualifies. In quorum mode a record can be adjudged a hit if it satisfies any m out of n conditions. Combining the modes allows records to be selected on a mixture of mandatory and optional conditions.

Retrieval Unit (RU)

The RU retrieves either complete hit records or specified fields out of such records. It extracts these from the data stream under the joint control of the LFU—which informs it where relevant fields are to be found—and the SEU—which tells it which records are hits.

Retrieval Processor (RP)

The RP controls the store in which extracted data is held pending transfer to the host mainframe. It can also perform some post-retrieval functions, such as:

- □ counting hit records;
- □ totalling specified fields;
- □ finding maximum or minimum values;
- □ discarding data which after totalling is not required to be returned to the host.

When any such functions have been performed, the RP returns the selected information to the host.

For further information contact

International Computers Limited
World Headquarters
ICL House, Putney
London SW15 1SW England

Produced and published by
ICL Group Communication Division, Putney.

CS793 Printed in England / KPL

We can even find a noodle in a hat rack.

Think of all the data accumulated by your company.

Think how all that data is growing every minute of every day. And how the more information you can store, the harder it is to retrieve.

That is why ICL have developed a system to search through data at breathtaking speed.

The system is called CAFS. The letters stand for Content Addressable File Store, but its full name is CAFS-ISP. That stands for Information Search Processor.

CAFS won't just find a needle in a haystack. Sometimes you might not know exactly what you're looking for, so you can ask CAFS to cast its net wider. It's called "Fuzzy Matching." Which is why CAFS can find a noodle in a hatrack, if you so desire.

This ability to sift through to the right information is already proving invaluable to CAFS users.

North Thames Gas, for example, have found that CAFS comes up with the right answer up to thirty times faster than their non-CAFS equipment.

CAFS is now in use by a wide range of local authorities and utilities, and a steadily increasing number of private sector companies in the retail, manufacturing, and service industries.

As you may have gathered by now, CAFS-ISP is unique. There is literally no faster way of retrieving data. Yet it is just one of the many recent technological innovations from ICL designed to meet the growing information needs of your company. The system makes smaller demands on your computing power, and is now available on most ICL mainframe computers.

CAFS has already made noticeable improvements to the productivity of many organisations, and is proving extremely profitable to them.

But be warned. The Inland Revenue are using CAFS too.

We should be talking to each other. ICL

FOR MORE INFORMATION ON CAFS-ISP OR ANY OTHER ICL SYSTEMS, CONTACT INFOPOINT ON FREEFONE ICL.

Michael BYWATER

The Thrut Will Out

WHEN the dog barks, when the cat howls, when the eyeballs ungum and see "We Can Even Find A Noodle In a Hatrack" gazing from the cheap pulpy matutinal newsprint, the natural assumption is that the brain has yet to surface. Wrong. What it means is that the bright sparks at ICL computers have come up with a means of getting the right answer out of your computer even if you ask the wrong question, spell the name backwards or turn blue and die at the keyboard. With a whirr and a bleep, primeval fears are once more unleashed.

What fears? Of men out there, writing it all down in greasy little books; men in grubby flannel trousers with binoculars pressed to their red-rimmed eyes who ring up the office from their Cortina, and say things like "It is Mister Fanshaw here, is that Mr Bywater, Mr *M.K.H.* Bywater...? As you have failed to . . ."

How nasty. How unutterably horrid. The instinct is to bellow down the telephone: "How dare you call yourself 'Mister'? You're not entitled to that honorific. Go to hell, scumbag! And anyway, this isn't Mr Bywater; he's *dead*, you and your sort killed him with your telephone calls and plain brown envelopes and stupid questions and lies about bills and blondes and agreements and overdrafts, he could have been one of the great ones, he could, given time, have grown a moustache and buggered off to the Gilbert Islands, but you wouldn't give him a chance . . ."

. . . but it wouldn't do any good; it's all down in the little book, ready to be telephoned over to some pimply draggle-haired frippet called Janine or Torvill, who will type it into the machine where it stays for ever or until it is retrieved by some brooding glob of malevolence, whichever is first.

One fights back, after one's fashion.

One has certain venial little misdemeanours at one's command. Nothing so outmoded as the false beard, blue spectacles and slouch hat here; new technology demands new weapons, the date of birth filled in, now American style, now European; the Previous Address If Less Than Three Years altered slightly but significantly from form to form; the postal code recalled in just the wrong order, the name spelt differently on each credit card, Thank you, Mr N. K. H. Bowater, that will do nicely.

Nothing, you understand, that wouldn't stand up in Court, my memory, your Honour, no, I really can't understand it, I suppose the girls at Head Office must have misread my handwriting, Nichael Tywater of *Pinch* magazine, how silly, haha, all a misunderstanding of course, no deception intended, the fact that our names are on the same registration card *of course* doesn't mean that we checked in at the same time, the room number must be wrong, oh dear, I shall certainly sue, defiling the memory of my dead brother like that. What? I claimed he was killed at Bloemfontein? No no no no no, died of sprue while selling Bakelite butt-buffers to the Flemish infantry at Floem Beinton, good market, generous commission, opportunity to proceed up the management ladder were it not for a tragic clerical error, yes he *did* leave a lot of debts and angry husbands but nothing to do with me, understandable confusion but I don't really see what I can do.

Not any more.

Used to be a time when computers were literal, GIGO we called it, Garbage In Garbage Out, a comforting little algorithm to murmur as one crouched, blotto, beneath window level, crawling from writ to summons past the bit where the Sherendale escritoire used to be before they took it away, did I say Sherendale, my mistake, I meant Hienz Baked Banes, it's all we eat now since the gas was cut off.

By mistake.

Because now they have Fuzzy Search, which sounds like something the Brixton Police might use and in fact is. All sounds jolly cosy (We can even find a noodle in a hatrack, ho ho, aren't we lovable rustic types, our gleaming red noses all ready to poke into other people's business *but only in fun of course*, nothing sinister here, no need to worry, just buy! buy! buy! persecute! muckrake! hound down!) but what it means in short is that, if you want to lay down a smokescreen, you're going to have to lie like mad.

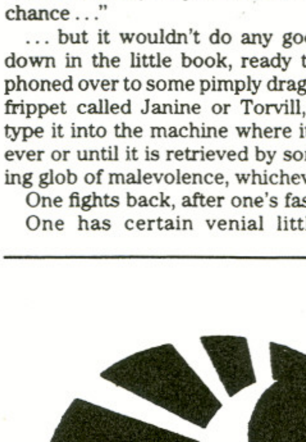
Don't pretend you haven't thought it all out while lying in the darkened drawing room with legal seals on your sleepless eyelids, ignoring the doorbell. Don't tell me that you haven't sat there, night after night, doing pathetic "accounts" which are nothing more than a list of your ghastly liabilities put in alphabetical order and bound in a special legalistic folder which has plunged you another £5.57 further into debt. I know the interior monologue which pertains on these occasions . . .

. . . *If I can fend off the Gas for six weeks then the cheque from the insurance company will arrive providing they don't notice that I've put two "L"s in Rolex which should fool the computer into not discovering that I've already claimed for it and anyway the false receipt will fool it, anyway, they always check the Post Code first, and then I can pay off the furniture people and then I can get an Access card because if I spell my name Brawn instead of Brown they won't find all those outstanding judgement debts, but they're going to write to the Bank aren't they they're going to write to the bloody bloody bloody Bank, they'll say they haven't got a customer called Brawn, but then the address is right, nearly . . . maybe the Bank'll just assume it's a spelling error, then I can have an Access card and take out cash on tick and pay the bills and then it'll all be all right because this time next year things should be okay if I can get rid of that bloody Sandra Oboe, she's costing me a fortune and I'm sure Mr Oboe's getting suspicious, I shouldn't have given my mother-in-law's address when we checked into the hotel, I'm sure that little bloke was following me, O God O God O God . . .*

And it won't work any more, any of it, because next time you spell yourself Smoth or Jines or Frad Bleggs the computer is going to click, whirring, into CAFS (Content Addressable File Store) and come up with all the alternative people you might be or have pretended to be, and then where will you be? Tell too big a lie and the cross-checks will break down. No way out. Only one thing for it.

(Lorches towerds bittle of Glinfeddoch and tikes hoge sweg, in vain hipe thit nabedy nutices, but it's too late. The machine has locked on. Outside in the thickening dusk men in raincoats prod their pocket computers and transmit valuable data down the RS232 ASCII TTY Modem, wasting not a second in filing that all-important information which in today's fast-paced world of sneaking, spying and skilldoggery could make all the difference between escape and detection. All-innocent of the tragedy unfurling within the precincts of 27 Acacia Mansions/21 Acaica Masnions/72 Accaico Mensions, Mrs Sandra Oboe trips lightly up the path, sparing not a glance for the three men crammed into the telephone kiosk fighting to plug in their data lines. She knocks on the door of number 27/21/72. No answer. Lets herself in.

Oh horror. Depending from the ingenious antlers on the wall are (a) a Homburg; (b) an old greasy trilby, peppered with Strand cigarette burns; (c) the hopeless greying corpse of her illicit beloved. "Oh you noodle!" she weeps fondly, "I never thought to find you in a hatrack!" which just goes to show that computers will never beat us humans when it comes to the real thing.)



Document Reference: ICLCUA.001

First Edition: July 1984
Second Edition: August 1984

Published by:
ICLCUA (UK)
PO Box 42
Bracknell, Berks
UK