Clareti VME

TP+

Transforming the way VME works for you

TP+ is a TP Monitor designed specifically to take advantage of the architecture and facilities of UNIX

Introduction

This document provides a technical overview of tp+, in terms of the facilities it provides and the user requirements it is designed to meet.

The Need for TP+

Online Transaction Processing has been well established in the mainframe market for more than two decades and OLTP systems lie at the heart of the operation of most organisations. As the capabilities of smaller open platforms increase, so there is a growing demand to implement critical OLTP systems within such environments. In turn this gives rise to the need for system software which can deliver the performance and the disciplines of data integrity, resilience and recovery which are essential to these high profile systems.

tp+ has been specifically developed to address this need by implementing the concepts and features of a mainframe TP monitor under the UNIX environment.

What is TP+

tp+ is a TP Monitor built to a well established model but is designed specifically to take advantage of the architecture and facilities of UNIX. It provides the features essential to Online Transaction Processing (OLTP) Systems which are not provided by UNIX itself.

As a "classic" TP Monitor tp+ is built to the same proven and successful model as IBM's CICS and TPS, Gresham's own TP monitor for ICL & IBM Mainframes. This model is designed to provide high performance and throughput with a minimum of resources. It also offers comprehensive levels of Resilience and Recovery without involvement of application code.

tp+ supports application code written in COBOL or C providing access to system functions through a simple command interface. Only pure application logic needs to be coded and many activities such as validation are simplified by the facilities offered by the integral Screen Painter. Comprehensive diagnostic facilities are provided to assist the application programmer.

This "Classic" TP Monitor approach offers a low cost and low risk route for porting mainframe OLTP systems to UNIX.

tp+ is "native" to UNIX in that it has been developed to utilise the strengths of UNIX and to adhere to standard interfaces. The Process Architecture is used to ensure the independence of system users whilst Shared Text (Code) and Shared memory (Data) features are used extensively to minimise memory occupancy. All UNIX System Calls made by the tp+ systems software conform to the System V Interface definition (SVID) and standard Inter-process Communications features are used to ensure Data Integrity as well as portability. tp+ has also been developed with a view to XA compliance in order to ensure the widest possible applicability.

Within the UNIX framework tp+ provides the features that are essential to any serious OLTP system. Firstly, full screen (template mode) terminal handling is supported. Secondly, Data Management facilities include support for flexible and efficient file structures; record locking mechanisms and "Commit" procedures to provide data integrity; and a sophisticated buffering system which ensures high performance together with a high level of resilience. Finally, comprehensive recovery procedures protect the OLTP system from transaction aborts, system failure (hardware or software) and media failure.

Structure of TP+

An OLTP system based on tp+ is a single linked program incorporating both the system software and the application code. At run time, a number of copies of this program run as separate processes.

A single Application Process is loaded for each active ("logged on") user and this process carries out the complete processing of all transactions entered by that user. In addition to the Application Processes there are two special versions of the tp+ program which remain loaded whenever the service is running. The "Control"

Process carries out the start-up procedures and a small number of other general functions including the maintenance of a diagnostic trace if required. The "Daemon" Process monitors the state of the buffering system and carries out checkpointing procedures as and when required.

The code is loaded as Shared Text so that one copy is shared between all processes. This helps to minimise store occupancy without jeopardising the independence of each system user. Main store usage is further reduced by the use of shared memory to hold system-wide information such as Screen Template definitions. Shared memory is also used to hold the buffer pool and to provide efficient data integrity mechanisms.

The tp+ structure provides efficiency, compactness, true independence and data integrity for multi user systems. The "single program" model of tp+ keeps costly inter-process communications to a minimum. The use of shared text, libraries and memory optimises main store occupancy. Safe multi-threading is assured because each user's transactions are carried out in a separate process. Data integrity is provided by locking and Commit procedures carried out through shared memory.

Features of TP+

"Visible" features of tp+ include terminal and data handling options whilst, behind the scenes, the software covers equally important areas such as Security (Privacy) provision, Data Integrity, Resilience & Recovery and Buffering.

Terminal Management

tp+ drives terminals in the full screen (template) mode used by most mainframe OLTP systems. Screens are divided into protected and unprotected areas. Users can move freely between unprotected areas entering data as required but the data is only submitted for processing once the "Send" (or equivalent) key is pressed. Through the Screen Painter, tp+ provides comprehensive validation of input fields and only "valid" messages are passed for processing by Application Code. Further Screen Painter facilities plus a high level programming interface simplify the construction of output displays and enable screens to be amended without affecting Application Code.

The Screen Painter also allows multi-level Menus and comprehensive context sensitive help systems to be built into any tp+ system without the need for any application code.

Print Handling

The secure creation of UNIX print data files is supported under normal commit control. The programming interface also enables a UNIX spooler (such as "lp") to be activated to output the created documents.

A tp+ Print Option which enables applications which produce printed output to produce it in a controlled way which is co-ordinated with the completion (or abnormal termination) of on-line transactions is available. It is a separately licensed option used in conjunction with, and in addition to, the basic tp+ Transaction Processing product.

The Print Option can be used to assist the migration of systems running on ICL mainframes under the Gresham TPS Print Option to UNIX-based systems under tp+.

Data Handling and File Management

tp+ includes an ISAM file management system geared specifically to the requirements of transactional working.

Both serial and indexed file structures are supported. The indexed file organisation allows each data file to be supported by up to 99 indexes all of which are held in a single physical file. All except the Prime index for each file can support duplicate keys. Keys can be multi-part and these parts need not be contiguous within the data record.

Indexes are held and maintained in a balanced tree structure and a utility program is provided to create the indexes from the data file.

tp+ supports efficient and fully resilient Serial and ISAM file structures. Both forward and backward browsing are supported. An SQL Interface to Relational Databases is planned for future development.

Application Programming Interfaces

tp+ supports application code written in COBOL and "C"; other languages (e.g. 4GLs) can be used where there is a facility to call external "C" code.

Separate application code environments are provided:

- Interactive - for online template driven transactions.
- External - for batch programs or processes driven by non-tp+ user interfaces which need to access tp+ managed data in parallel with

Clareti VME

TP+

Transforming the way VME works for you

Interactive work with full commit control.

- Batch - for batch programs which are only to be run when the tp+ service is unloaded.

The X/Open ISAM API is supported from "C" or by using standard COBOL file handling verbs.

Distributed Access

The ISAM Resource Manager in tp+ may be replaced by the ISAM-XA option. This fully compliant product will allow files used by tp+ to be accessed simultaneously by a distributed TP system within the same network.

With the ISAM-XA option, tp+ can support global transactions managed by a Transaction Manager such as Novell's TUXEDO alongside its own local transactions. For these transactions, tp+ implements a full 2-phase commit through the X/Open XA interface.

Security (Privacy)

Access to a tp+ service is controlled by the standard UNIX security facilities but these are extended in ways particularly appropriate to OLTP systems.

Firstly, once the user has gained access to the service, tp+ uses the "Login Name" to determine which applications within the service the user can access. These permissions are defined online, and can be so amended, by the System Administrator. This mechanism also enables the system to present different welcome screens to different users.

A special facility allows users to change permissions without logging out simply by specifying an alternative login name and password.

Finally, users of a tp+ service can be denied access to the UNIX shell via the "profile" mechanism of UNIX. For such users, the act of logging in automatically causes an Application Process to be loaded and a welcome screen to be displayed. Logging out from tp+ automatically logs the user out of the shell.

Data Integrity

Data integrity within a tp+ service is assured by the provision of Record Locking and the 2-phase Commit procedure together with the use of UNIX's Synchronous Write option.

Record level locking is implemented to ensure data integrity in a multi-user environment. tp+ ensures that updates do not get "lost" due to conflicts between concurrent messages from different terminals.

The Commit procedures define precisely and securely the endpoint of each transaction and thus allow locks to be released safely. The Commit procedures adopted by tp+ conform to 2-phase commit requirements so that tp+ is capable of co-ordinating recovery between itself and external systems (e.g. Relational Database Management Systems or other OLTP systems in a distributed TP environment).

The Commit procedures depend crucially on the ability to guarantee that disc writes have actually taken place (i.e. they are not simply buffered in a Cache). To achieve this, tp+ uses the Synchronous Writes option provided by UNIX.

Resilience & Recovery

tp+ provides resilience against Transaction failure, System failure and Media failure. Protection is achieved by a combination of automatic Before and After Image logging, the Commit procedures and Checkpointing. In this respect, a Checkpoint represents a moment in time when all files are guaranteed to be in step with the online transactions processed.

In the event of a transaction failure, tp+ will automatically abort the transaction by reversing any updates carried out by it. This is achieved by using Before Images saved automatically by the system software during the transaction processing. In most cases the transaction abort process reverses in-store updates only if the Commit point has not been reached.

Following a system failure, the next reload of the tp+ service will cause System Recovery to be invoked automatically. System recovery will restore files to reflect all committed transactions but not those which were current at the time of the crash.

This is achieved by applying Before Images to restore files to the last Checkpoint and then using After Images to "Roll Forward" all Committed transactions.

Following a media failure or other data corruption, tp+ is able to recover from restored archive files by rolling forward using After Image logs.

The Buffer System

The buffering system used by tp+ is central to its performance and resilience capabilities.

A single pool of buffers is shared at run time between all files and all

processes within the service. This helps to maximise the utilisation of the buffer space. The more space that can be allocated to this buffer pool, the greater the benefit that will be derived.

The buffering system reduces disc reads through the use of a retention system. Once a block has been read, any future access to that block can be satisfied from store unless and until that buffer needs to be freed for other use. Preferential retention factors allow priorities to be specified so that data can be held or discarded dependent upon the likelihood of re-use.

Disc writes are minimised by delaying updates as long as possible. At the Commit point of a message, the After Images are forced to disc but the data records (and Before Images) are not. Thus if several updates to a single block occur within a short time, it is only necessary to write that block to disc once.

The forced writing of After Images at the Commit point of a transaction is sufficient to ensure complete resilience but the retention and delayed update procedures help to optimise performance. Furthermore, the more store that can be made available for buffering, the greater the retention and thus the better the performance.

The buffering system is monitored and managed by the Daemon Process whose activities can be tuned as required.

The Daemon carries out three functions. Firstly it flushes the buffers (by forcing writes of updated blocks) to maintain a specified threshold percentage of free space for use by new transactions. Secondly, it implements Checkpoints as and when required. The frequency of checkpointing can be amended at any time by the system administrator who also has the facility to force a checkpoint if necessary. Smoothing the checkpoint procedure is the Daemon's third task. It does this by anticipating the checkpoint and forcing gradually more and more data to disc as the checkpoint approaches.

SUMMARY OF BENEFITS

Protects Investment

The development of critical OLTP applications represents a very high level of investment in code, data and skills achieved over many years. By providing a "mainframe like" OLTP environment, tp+ offers a transition path to UNIX which gives the opportunity to preserve this investment whilst gaining the benefits of the new open platforms.

Functionality

tp+ makes application development faster and simpler by providing processes for recovery, data integrity, navigation and screen handling.

An online menu-driven system allows system administrators to define, monitor and control the system, terminals, files, users and transactions and provides for the security of those systems.

A full screen painter facility provides developers with WYSIWYG capability of building consistent screens, as well as defining the validation logic, navigation logic etc.

Indexed sequential file, serial sequential file and VME data store capabilities are incorporated with tp+ via Gresham's DataServe.

High Throughput

UNIX provides an online environment for users to submit batch oriented jobs. tp+ upgrades this environment to provide a mechanism for many users to share the resources of the machine, providing high throughput with fast response for each user.

Standard COBOL Code

Mainframe COBOL TP programs leave some of the processing to the TP monitor, such as screen handling, validation, help text, error messages etc - tp+ performs the same logic, so conversion of the code is straightforward.

Consistent Screen Definitions

tp+ provides an external mechanism for defining layouts from the COBOL program; screen formats can be converted from mainframe systems and copied directly to the tp+ screen definition file.

UNIX Platforms

tp+ is available on the major UNIX platforms including Hewlett Packard 9000 series (HP/UX), IBM RS6000 (AIX), ICL DRS3000 and DRS6000 and Data General AViiON (DG/UX).

Transforming the way VME works for you

About Gresham

Gresham Computing plc (LSE:GHT) specialises in the provision of real-time financial solutions to banks and corporates, and has a well-deserved reputation for technical excellence, reliability and a strong service culture. Our storage division helps the largest data users to better manage the unrelenting growth of data.

Further information

For more information on how Clareti System Management can help your company visit www.gresham-computing.com or you can email us at info@gresham-computing.com

Alternatively you can contact our offices directly.

Europe, Middle East and Africa
T +44 (0)20 7653 0200

Americas
T +1 212 792 4125

Asia Pacific
T +61 2 9955 7660

Gresham
real-time financial certainty