FreezeFrame

MACRO 4
A Division of UNICOM Global

Application performance analysis

Eliminate performance hotspots from your mainframe application code

FreezeFrame is a comprehensive application performance analysis solution for z/OS

Designed specifically for z/OS systems, FreezeFrame is a performance analysis solution that helps you to isolate application problems through detailed observation and to correct badly performing code. FreezeFrame can be used with ExpeTune, which identifies problem areas for analysis by FreezeFrame. Through the use of FreezeFrame you can:

- Pinpoint performance problems in both batch and online applications, relating them to lines of code in the program listings
- Produce useful and informative reports that are easy to understand
- Obtain a clear and precise picture of where and how your applications use system resources
- View the program source associated with data that is provided through observation reports
- Measure and report on application performance to outline how system resources are used throughout your application landscape
- Assess the performance of Java Virtual Machines (JVMs)
- Exploit the power of z/OS through native and Eclipse interfaces

z/OS application performance analysis

FreezeFrame provides a user-friendly view of how system resources are used by applications in the z/OS Sysplex environment. FreezeFrame supports batch, CICS, DB2, IMS and ADABAS. Support is also provided for Java JVMs, IBM WebSphere Application Server and IBM MQ subsystems.

Measurements are taken during an observation session; activity in any target address space can be sampled at any frequency and duration. Multiple address spaces and enclaves are also supported.

Each observation consists of many execution snapshots. Each snapshot records what the target address space was doing at that instant in time; this includes details about the system object and its state. The recorded data is then aggregated and attributed to system objects.

System states are described as executing, waiting or queued and cover all components within the address space. System objects may be load modules, CICS transactions and so on.

By mapping observed system states to system objects, the performance analysis reports provide a meaningful picture of how resources are consumed and allow users to pinpoint slow or incorrectly executing programs.

FreezeFrame

Macro 4's integrated suite for cross-platform, z/OS and Db2 performance management:

SUPERMON® for Java
Cross-platform Java application performance management

ExpeTune
z/OS application performance management

ExpeTune DB
Db2 performance management

FreezeFrame
z/OS application performance analysis

www.macro4.com

Measuring performance through observation

Observations can be started manually, scheduled for a specific time, or started automatically when a specific event occurs. FreezeFrame then uses non-intrusive stealth sampling technology (SST) to collect information on resource usage.

The measurement engine never schedules I/O or any other task in the profiled address space, resulting in minimal system impact during sampling. The measurement engine will:

- Report on all OS supervisor calls, not just a subset
- Report on activity while the local lock is held
- Collect data for UNIX System Services (USS) and Hierachical File System (HFS)

The SST approach provides a precise view of an application's performance by sampling program execution with minimal system impact. During an observation session, FreezeFrame repeatedly takes snapshots of system activity in the target address space and writes the information to an observation file.

Threshold monitor

FreezeFrame can monitor jobs and trigger a measurement when a specified threshold is reached. The threshold criteria are: EXCP count, CPU time and elapsed time.

Monitoring Db2 and SQL activity

For applications that use Db2, the execution of SQL calls tends to use most of the CPU time, and is the major contributor to the overall elapsed time.

The reliance of businesses on web services and internet technologies means that mainframes must also process SQL requests that come in from remote systems via the network.

FreezeFrame will allow you to find defective SQL and tune it, thus reducing CPU usage and improving throughput:

- Intercept every Db2/SQL call made
- Simply perform an observation of the process, for example, batch job or CICS system, and focus on how Db2/SQL activity contributes to the CPU usage and elapsed time of the overall process
- Use the distributed data facility to monitor and report on the impact of remote SQL requests on your system resources
- Perform observations over complete enclaves via the workload management interface
- Retrieve and report on static Db2 EXPLAIN data

FreezeFrame can also obtain Db2 accounting trace data from SMF.

Java virtual machines (JVMs)

A number of Java-specific parameters can be reported upon, including heap usage and CPU usage by thread. Support is provided for a pre-loaded JVMTI agent as an alternative to using the Java Attach API.

Source program mapping

Many observation reports allow you to load and view the program source associated with a specific line entry in the report; this is available for CSECTS, Db2 SQL statements and CICS command object types.

Language environments

FreezeFrame has source program mapping support for COBOL, PL/I, Assembler, C, C++ and Java. For Natural, measures are performed by statement number.

Intuitive interfaces

The FreezeFrame application performance analysis functionality is available via native and Eclipse interfaces, both of which are intuitive and easy to use.

Please contact us for more information:

USA	Tel: +1 973 526 3900	Email: market.usa@macro4.com
Europe	Tel: +44 1293 872000	Email: market@macro4.com