

Measuring CPU eligible for Mobile Workload Pricing

Ian J Mitchell



Contents

- 2 Introduction
- 3 What to Measure
- 4 Example 1: Routing to dedicated subsystem instances
- 4 Example 2: Distinction by tranid
- 5 Example 3: Using CICS Origin Data
- 6 Example 4: Using IMS LTERM Override
- 8 Summary

Introduction

This document provides guidance to help you measure CPU consumption which is eligible for IBM Mobile Workload Pricing for z/OS.

To recap from the announcement letter, Mobile Workload Pricing (MWP) can reduce the cost of growth for mobile transactions processed by programs such as CICS®, IMS™, and DB2® for z/OS.

In order to take advantage of MWP it is necessary to measure the CPU resource consumed by the eligible transactions and use the new Mobile Workload Reporting Tool (MWRT). Using these measurements, MWRT calculates the billable MSUs for sub-capacity eligible programs.

MWP has an objective to be applicable to existing systems running existing workloads with minimal disruption (for example, no requirement to define new LPARs).

No new measurement capabilities have been created to support MWP – it is expected that existing CPU measurement capabilities of z/OS and/or the *mobile defining* products will be sufficient to fulfil the quantification requirements of MWP.

Eligibility of workload for MWP and the form of measurement is agreed between the client and IBM in the form of a Contract Addendum.

Included in this document are a number of examples of measurement techniques – this is not an exhaustive list and the examples provide non-prescriptive illustrations.

The focus of this document is not to discuss the

precise eligibility criteria for a given workload. Rather the focus is on how to provide the data which quantifies the CPU resource consumption of that workload.

The measurement of eligible workload requires that the mobile work is distinguishable from the non-mobile work in the system.

This distinction is a crucial element of qualifying for MWP. Measurement of CPU eligible for MWP must be supported by evidence that the request which consumed the CPU resource originated with a mobile device or a mobile application. It is not acceptable to declare CPU consumption eligible without evidence of the distinction.

Two broad classes of distinction have been identified:

- **Explicit** – the data associated with the incoming request contains markers or tags which ONLY requests from mobile devices or mobile applications have.
- **Implicit** – the context of the request identifies that it is coming from a mobile device or application.

All acceptable distinctions originate outside of System z and z/OS – it is not possible to measure CPU eligible for MWP without using a distinction flowing into the system.

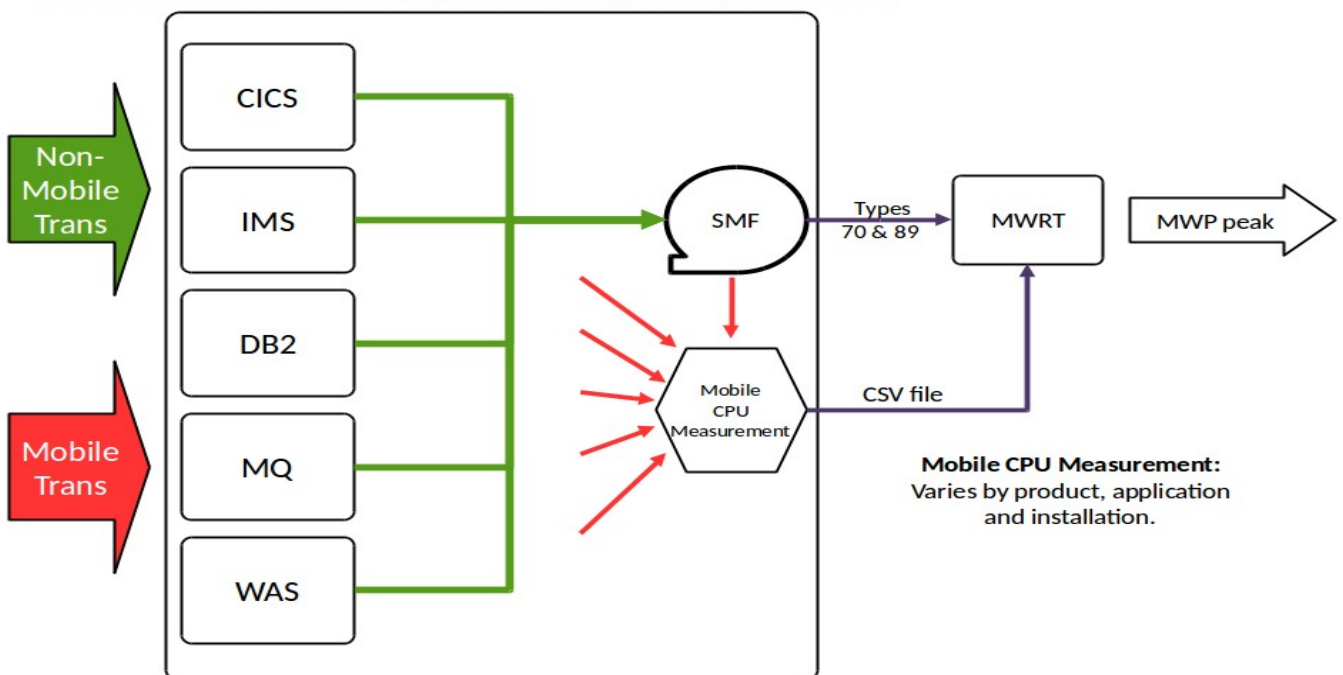
What to Measure

As explained above, MWRT requires CPU consumption measurements in order to produce the peak MSU for the LPAR which determines the charge for the MLC software.

Technology Requirements



Mobile Workload Pricing: Technology Requirements



CPU consumption measurements may be recorded at multiple levels. Two levels relevant for MWP are the address space and the transaction (or request).

If the distinction is used to route the requests to address spaces such that they exclusively handle mobile requests, then measuring the CPU at the address space level is appropriate.

If the distinction is not used to achieve such division of mobile from non-mobile workloads, then measurement of CPU consumption at the individual transaction level is required.

In all cases it will be necessary to aggregate the measurements to match the LPAR granularity of the underlying data used by MWRT. The use of tools such as CICS Performance Analyzer and IMS Performance Analyzer can assist in this task.

Example 1: Routing to dedicated subsystem instances

This is an example of implicit evidence being used to qualify measurements of CPU used for mobile transactions.

This technique relies on implementing a routing mechanism such that all mobile requests go to a distinct set of processing resources (CICS or IMS regions for example) and then simply measuring the CPU consumed by those regions.

This has the advantage that transaction-level monitoring is not required – it does not need CICS SMF 110 records. It does require that the distinction of mobile requests is used to route work to processing resources reserved for them exclusively – it would be unacceptable to allow any non-mobile requests to be served by the

'mobile regions'.

As well as the first region to which the request is routed (eg a dedicated CICS terminal-owning region) the other systems used to serve the request can be similarly dedicated to mobile workloads by defining the appropriate topology of connections. This would include dedicated CICS Application Owning Regions and DB2 instances.

There would be no direct evidence on a transaction by transaction basis that the workload was serving mobile transactions.

Within this scheme, it is possible to use shared processing resources (such as CICS File-owning regions) which are serving non-mobile requests, but it will not be possible to measure the CPU consumed in such regions and include it as eligible for MWP.

Example 2: Distinction by tranid

Transaction names (or tranids) are a very familiar mechanism used to distinguish workloads of different types. Although most commonly used as a functional distinction, it is common to use them for accounting and charge-back purposes. Also tranids are already commonly used to inform routing and workload management decisions. Obviously, they are also included in the transaction level monitoring records and many performance analysis tools have excellent support to split out work by tranid, so extending their use to distinguish workload for mobile CPU measurement might be appropriate.

However, this will mean defining new tranids in order to ensure that eligible and non-eligible

workload is robustly distinguished. Also, you might have application or system code which is sensitive to tranid values already, so overloading their use for MWP might not be appropriate, and you would want to consider one of the other options.

In CICS TS, tranids dedicated to mobile workload can be associated with the same application programs and run alongside tasks using the existing tranids for non-mobile requests.

Also, not all parts of the overall processing for a given request necessarily use the same tranid – for example, the tranid for mirror tasks used when a command is function-shipped is typically CSMI. It is possible to correlate all the tasks used to serve a request to CICS which begin in a distinct tranid but then use tasks running under the generic mirror tranid by using the CICS unit-of-work id, but this is more complex than the technique described below using origin data.

In IMS using this technique all mobile transactions would require a unique transaction code and transaction class so they could only run in selected dependent regions. If an individual transaction is needed by both mobile and non-mobile workloads, it would need to be cloned. This would require definitional changes, process changes and possibly application changes. It would also require process changes to create and support additional dependent regions on all the IMS systems.

The next examples rely on attributes of the request's context which can be used to qualify transaction-level CPU measurements.

Example 3: Using CICS Origin Data

CICS Origin Data was introduced in CICS Transaction Server version 4.2 as part of the capability to track transactions from point of origin throughout their processing no matter how many tasks are involved.

When a task enters the system, various attributes of the request's context are captured as origin data. This forms part of the association data which CICS flows to all tasks¹ involved in serving the request – for example, it is sent to a file-owning region when a file request is shipped. Origin data is included in the CICS SMF 110 record for each task along with the CPU consumed by the task.

These characteristics make the contents of origin data excellent candidates to use to filter mobile from non-mobile tasks in the SMF data.

Tasks which originate from any form of TCP/IP communication with CICS includes the port number to which the client sent the request.

CICS uses the TCPIPService resource definition to control which ports are used to receive work into CICS.

This makes distinct port numbers an excellent mechanism to use to externally distinguish mobile from non-mobile work. That is to say, ensuring mobile workload requests are sent to distinct port numbers separate from the ports used for non-mobile workload.

Defining a distinct TCPIPService is all that is required to enable a new port to receive mobile

¹ Origin data flows over MRO and IPIC connections, it does not flow over connections using LU6.2/APPC.

work – the same tranids and programs can be used, but the origin data recorded in the tasks' SMF 110 records will include the distinct port number.

The same is true if you use the WebSphere Liberty Profile in CICS as a server to receive HTTP requests. Since CICS makes every application thread within Liberty a CICS task, there is origin data for those threads and this includes the port on which Liberty is listening for work. So again, it is a simple matter to define a distinct TCP/IP port (this time in server.xml) to allow mobile requests to be received and recorded via SMF 110 records in a way which can be used to measure the mobile CPU.

Here is an example of the origin data for a Liberty task – the same fields are populated for tasks entering the system via a TCPIPService.

Origin data		
Origin Adapter Data 1	ODAPTRDATA1	
Origin Adapter Data 2	ODAPTRDATA2	
Origin Adapter Data 3	ODAPTRDATA3	
Origin Adapter ID	ODAPTRID	
Origin Appl ID	ODAPPLID	IYK2Z32E
Origin Appl ID Net ID	ODNETWORKID	GBIBMUYA
Origin Client IP Address	ODCLNTIPADDR	9.78.79.4
Origin Client Port	ODCLNTPORT	49630
Origin Facility Name	ODFACILNAME	LIBERTY
Origin Facility Type	ODFACILTYPE	JVMSEVER
Origin IP Address Format	ODIPFAMILY	IPV4
Origin Net ID	ODNETID	
Origin Task	ODTASKID	0000054
Origin Task Start Date	ODSTARTTM	20140912095605.532286
Origin Task Start Time	ODSTARTTIME	2014-09-12T09:56:05.532286+00:00
Origin Transaction ID	ODTRANSID	LIBX
Origin User ID	ODUSERID	WAKELIN
Origin VTAM LU Name	ODLUNAME	
Previous hop data		

Note on DB2 or other resource manager CPU within a CICS workload

When a CICS task executes a request related to another resource manager (for example an SQL request in DB2) it uses one of the L8 TCBs managed by CICS. A large proportion of the CPU consumed by the execution of the request is actually on that L8 TCB rather than a TCB owned

by the resource manager (either in the CICS address space or that of the resource manager). So by using the SMF 110 CPU measurement for the CICS task, you are including the CPU consumed in executing the request which you would not consider as 'CICS' CPU.

This means that you need not, and should not, use the resource manager's own CPU measurement (for example, the DB2 101 record) since that would account for that CPU consumption twice.

Example 4: Using IMS LTERM Override

In the same way you need CICS SMF 110 records to capture transaction-level CPU measurements and workload distinction characteristics, IMS Transaction Level Statistics (TLS) is required to mix mobile and non-mobile workloads in the same IMS regions.

When TLS is enabled, IMS writes a x'56FA' log record for each transaction instance. That record includes the CPU time for the transaction instance, the LTERM name, the USERID along with other information. Within an application schedule, there could be multiple x'56FA' records where some of the transactions are from mobile and some are not. The x'07' log record would have the total CPU time for the application for the schedule. Adding up the CPU times in the x'56FA' records that are identified as mobile, provides the CPU for the mobile workload.

In the days of 3270 input, the logical terminal (LTERM) name that the user logged onto flowed in with the transaction. Some IMS applications are dependent on having an LTERM name and

make processing decisions based on it. As customers move to TCP/IP input there are still dependencies on the LTERM name even though there is no LTERM associated with TCP/IP input. IMS Connect allows an LTERM override in order for the user to pass a name into IMS to use as the LTERM name. This name is made available to the application and is written in IMS log records. If the application issues a program to program switch (inserts a message to another IMS application), this LTERM name is propagated to next application and will appear in its log records. IMS TM Resource Adapter, IMS Connect, MQ Bridge have methods to override the LTERM name that comes into OTMA.

This means the overridden LTERM name is a good candidate to distinction mobile requests in IMS. The IMS Connect client can pass in an LTERM Override name that uniquely identifies the transaction as coming from Mobile. This field is written in the x'56FA' and is propagated on program switch transactions. For example, all LTERM Override names from mobile could start with "MOB*", while other sources such as online banking start with other prefixes such as "ONL*".

Example 5: Explicit distinction by request application payload

In working with a number of clients, we have seen examples of standard practise to include data in application request payload which describes the type of request. This sometimes includes explicit distinction of requests from mobile consumers.

This data may already be used for workload

routing purposes and for accounting.

In the CICS environment, it is possible to add this data to the monitoring performed by CICS for the task using the EXEC CICS MONITOR command.

If your standards are already accommodating such explicit distinction of requests, then extension to use this for mobile workloads might be something you are already doing.

Such distinction would be a good candidate to measure mobile workload CPU consumption to provide data for MWRT.

Summary

This paper has described some techniques which can be used to measure CPU consumption for a distinct mobile workload using existing features of current releases of some of the mobile defining products with minimal disruption. This enables the generation of the required input to the Mobile Workload Reporting Tool.

The intention was to provide examples rather than instructions, as the least disruptive technique may vary according to many factors both technical and procedural.

We anticipate updating this paper to describe further examples which cover other of the products which can be used to define and measure mobile workload.

If you have questions or comments please email ianj_mitchell@uk.ibm.com.

Trademarks

IMS is a trademark of IBM Corporation in the United States, other countries, or both.

z/OS, System z, IBM, CICS, DB2, zEnterprise, WebSphere and z/VM are registered trademarks of IBM Corporation in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.