

Practical Capacity Planning for Memory – Part 1

Fabio Massimo Ottaviani – EPV Technologies

The cost of mainframe memory has been substantially reduced during recent years; however its current cost is still in the order of some thousands of Euro per GB.

On the other hand the number of GBs supported by the newest z/OS releases has greatly increased making it possible to almost eliminate paging activity and to fully exploit the many available DIM (Data In Memory) techniques.

While it's true that z/OS can sustain high paging activity, paging and especially page faults are very disruptive to online applications performance.

If a CICS region is getting 10 page faults per second this means that 10 transactions per second are stopped (or a transaction is stopped 10 times) and have to wait for the required pages to be loaded from page datasets. Furthermore you have to consider that CPU cycles are needed to perform paging. This is one of the sources of uncaptured CPU time.

Maintaining data in memory has two additional advantages: it improves performance by avoiding the I/O operations delay and it eliminates the CPU consumption still needed to perform I/O operations.

It's also important to remember that I/O operations can only be performed by standard CPUs and not by specialty processors (zAAP and zIIP) so the I/O reduction has positive effects on the software bill too.

All these factors mean that memory capacity planning is still a very important activity to perform.

The good news is that from a conceptual point of view the process is always the same:

- evaluate current capacity;
- estimate the utilization baseline;
- estimate the growth;
- forecast future resource needs.

The bad news is that dealing with memory, because of its nature, is more complicated than with other resources.

In this paper we'll discuss a practical methodology to do that.