Legac-E Education

Is BLKSIZE optimisation still relevant?

In August 2003, the following question was raised, "I was wondering if the information I saw about optimizing the block size is relevant given that 3390's are emulated on fixed block architecture drives and hopefully the dead space on tracks is ignored."

The text below is the response returned via E-mail, and it is posted here as it may be of more general interest.

Having often reflected on this very point the conclusion drawn was that "emulated" is the operative word. As the disk subsystems emulate 3390 or 3380 disks, there are three contexts in which a data sets BLKSIZE (or CISIZE in VSAM) needs be viewed:

- Physical Capacity
- Logical Capacity
- Performance.

Physical Capacity

Whilst modern disk systems, from which ever source; EMC, HDS, or IBM, are emulated as 3390 (or 3380) disks that is only a hardware perspective and not what z/OS sees. The actual disks in the device are similar to Winchester Drives, like the ones in PCs, but z/OS does not support these, as. z/OS does not intrinsically support Fixed Block Architecture (FBA) disks. z/OS only supports Count Key Data (CKD) disks. While z/OS will believe it is writing to one volume, the disk sub-system is probably striping the data across several volumes and in that context the Physical Capacity issue is more difficult to quantify than with real 3390 or 3380. and therefore might be considered irrelevant. **The other two contexts are still as relevant today as they ever were.**

Logical Capacity

This is the capacity of a given area of the disk allocated to a specific data set irrespective of the size of that area. z/OS will manage the allocated area based on a calculation of how many blocks of data will fit per track/cylinder. (See www.legac-e.co.uk/JCLdocs/CAP3390.pdf) If this calculation on SPACE and BLKSIZE is inaccurate then once any secondary requests for space are exhausted, an abnormal termination in the form of an **Sx37** will occur. This is one way in which BLKSIZE optimization is very relevant, and why using System Determined BLKSIZE (SDB) should be considered.

Legac-E Education

Is BLKSIZE optimisation still relevant?

Performance

This can be the crunch area when a poor BLKSIZE has been chosen. A simple 80 byte image data set with 800,000 records with require 800,000 entries to the I/O handling code if the data set is unblocked, which will result in a loss of control for each I/O and hence a degradation in performance observed as greater elapse times in batch jobs. Blocking 100 records per block reduces this to 8,000 entries to the I/O handling code, a 99% reduction in system overhead for the data set significantly improving throughput.

Virtual Storage Impact

The BLKSIZE (or CISIZE for VSAM) will affect virtual storage use, as typically it will influence the size of data buffers in memory. The number of buffers varies according to data set type, for example a sequential data set using the queued technique (QSAM) has 5 buffers by default. This means that the data set will require an area of virtual storage at least equal to 5 times the BLKSIZE. This in turn will affect the REGION size associated with the job step. **The trade-off is additional storage for better performance.**

The Human Element

The real issue these days is do people need to perform the calculation for optimization, and the answer to that is no. Whilst it is useful for people to understand what is happening and why, for the majority of data sets there is no need to perform the optimization, the system will do it if the correct parameters are coded such as:

- **BLKSIZE=0** - to indicate use System Determined Block-size.
- **SPACE=(record-size,(quantity**, ... i.e. use device independent space definition, which will require **AVGREC**
- **AVGREC=** - to indicate the size of the quantity units in the SPACE parameter
- Allocate VSAM data sets directly via Job Control rather than via the IDCAMS utility.