HiperDispatch helps provide increased scalability and performance of higher *n*-way and multibook System z10 systems by improving the way workload is dispatched across the server. HiperDispatch accomplishes this by recognizing the physical processor where the work was started and then dispatching subsequent work to the same physical engine.
-NOTE This intelligent dispatching helps reduce the movement of cache and data, and improves CPU time and performance.

HiperDispatch is available only with z10 PR/SM and z/OS

Short History Lesson - With the introduction of the z990, the zSeries LPAR hypervisor became aware of machine CP topology and began optimizing the allocation of logical partition CP resources to the physical resources accordingly. This support was enhanced in the z9 family of processors to dynamically re-optimize logical partition resource assignments as configuration changes were made. This support optimized physical CP selection for dispatching logical partition CPs in order to minimize disruption to the v various levels of internal caches on the machine.

For z10, IBM introduces a new level of synergy between the z10 LPAR hypervisor and z/OS software for managing logical CP resource allocations. HiperDispatch is the true start of having z OS software become aware of the topology of the machine, presented as a logical partition topology, to then provide dynamic affinity dispatching with regards to CP placement n the z10 book structure. One can think of the way z LPAR has traditionally managed shared logical partitions as being ***horizontally*** polarized. That is, the processing weight for the logical partition is equally divided between all the online logical CPs in the logical partition. In turn, the OS running in the logical partition is obligated to use all of those online logical CPs equally in order to be assured of receiving its fair share of the physical CP resources.

HiperDispatch introduces a new, optional form of polarization for managing the shared logical CPs of a logical partition called ***vertical*** polarization. z/OS running with HiperDispatch uses the logical partition topology information to decide how to group its logical CPs to set up its work queues and to exploit the vertical configuration of logical CPs to pack work into a smaller set of logical CPs, optimizing processor cache usage.

With HiperDispatch there is a common understanding in the z10 LPAR hypervisor and the z/OS software such that work can be concentrated on a smaller number of logical CPs within the logical partition that reflects the actual assigned weight of the LPAR.

With HiperDispatch enabled in a logical partition, z/OS will redispatch its tasks back to a smaller set of logical CPs, and LPAR in turn can dispatch those logical CPs back to the same physical CPs, optimizing use of both the L1 and L2 caches. Work can still expand and flow into more logical processors dynamically should workload conditions warrant it.

HiperDipatch is enabled within an LPAR via software controls.

To enable HiperDispatch under z/OS, specify a new keyword in the IEAOPTxx member of SYS1.PARMLIB. HiperDispatch=YES needs to be specified to turn on this mode of dispatching. The enablement is dynamic and can be changed without an IPL..

There are *no new* hardware controls or settings to enable use of HiperDispatch within a logical partition, however, the existing "control global performance data" security setting must be enabled for proper operation of HiperDispatch in a logical partition.

HIPERDISPATCH and zAAP -With today's book structure of z servers, CPU access to memory on a different book as implied previously takes considerably longer than access to memory and cache on the same book. Therefore, the HiperDispatch approach of dispatching units of work in z/OS takes the book structure of machines into account. When a unit of work is interrupted, for example,due to an I/O interrupt, the operating system will no longer dispatch the unit of work on any available CPU but preferably on CPU that shares cache and memory with the CPU where it had run previously.

CPUs that share a cache are grouped in **affinity nodes**. An affinity node is a set of CPUs that share hardware cache at a level with minimum storage access delay and optimum performance when running units of work.

Based on WLM-specified recommendations, the dispatcher will preferably dispatch units of work on the affinity node where they were processed previously. WLM will monitor the utilization levels of the affinity nodes and move work as needed to balance the utilization while maintaining specified performance goals.

If the total physical processor equivalent of the defined weight of a partition is less than 1.5 regular CPUs, a request to switch to vertical dispatching mode will still be honored by System Resource Manager (SRM) but a console message is issued as a warning that the system is not configured as recommended (**zAAPs** will not be considered for deciding when to switch to vertical mode). If during operation in vertical dispatching mode, the total physical processor equivalent of the defined weight of a partition drops below 1.5 regular CPUs, SRM will also issue a warning message the first time this happens.

Hiperdispatch is retrofitted back to zOS 1.7,