## HiperDispatch

PR/SM and z/OS work in tandem to use processor resources more efficiently. HiperDispatch is a function that combines the dispatcher actions and the knowledge that PR/SM has about the topology of the system.

Performance can be optimized by redispatching units of work to the same processor group, which keeps processes running near their cached instructions and data, and minimizes transfers of data ownership among processors and processor drawers.

The nested topology is returned to z/OS by the Store System Information (STSI) instruction. HiperDispatch uses the information to concentrate logical processors around shared caches (L3 at PU chip level, and L4 at drawer level), and dynamically optimizes the assignment of logical processors and units of work.

z/OS dispatcher manages multiple queues, called *affinity queues*, with a target number of eight processors per queue, which fits well onto a single PU chip. These queues are used to assign work to as few logical processors as are needed for an LPAR workload. Therefore, even if the LPAR is defined with many logical processors, HiperDispatch optimizes this number of processors to be near the required capacity. The optimal number of processors to be used is kept within a processor drawer boundary, when possible.

**Tip:** z/VM V6.4 and later also support HiperDispatch.

## Logical partitions

PR/SM enables z15 servers to be initialized for a logically partitioned operation, supporting up to 85 LPARs. Each LPAR can run its own operating system image in any image mode, independently from the other LPARs.

An LPAR can be added, removed, activated, or deactivated at any time. Changing the number of LPARs is not disruptive and does not require a POR. Certain facilities might not be available to all operating systems because the facilities might have software corequisites.

Each LPAR has the following resources that are the same as a real CPC:

- Processors

  Called *logical processors*, they can be defined as CPs, IFLs, ICFs, or zIIPs. They can be dedicated to an LPAR or shared among LPARs. When shared, a processor weight can be defined to provide the required level of processor resources to an LPAR. Also, the capping option can be turned on, which prevents an LPAR from acquiring more than its defined weight and limits its processor consumption.

  LPARs for z/OS can have CP and zIIP logical processors. The logical processor types can be defined as all dedicated or all shared. The zIIP support is available in z/OS.

  The weight and number of online logical processors of an LPAR can be dynamically managed by the LPAR CPU Management function of the Intelligent Resource Director (IRD). These functions can be used to achieve the defined goals of this specific partition and of the overall system. The provisioning architecture of z15 servers, as described in Chapter 8, “System upgrades” on page 331, adds a dimension to the dynamic management of LPARs.

  PR/SM is enhanced to support an option to limit the amount of physical processor capacity that is used by an individual LPAR when a PU is defined as a general-purpose processor (CP) or an IFL that is shared across a set of LPARs.