

## Kettner's Corner #09 - zTidBits@CPO (PR/SM CPU Dispatching)



John Kettner is a member of the Z CPO team and teaches classes on z/OS fundamentals to customers and IBMers. John will share his expertise sharing his answers on key questions and topics on Z.

[Check out the System z Project office](#)

[Website:](#)

<http://w3.ibm.com/sales/competition/compdlib.nsf/weball/9CF22F2F65E01639872571>  
[OpenDocument](#)

### **03/07/07 - #09 zTidBit@CPO (PR/SM CPU Dispatching)**

From its inception beginning with the z990, mainframe systems are designed to run always in LPAR mode. This decision makes it possible to exploit the fact that the LPAR hypervisor is always present and it is utilized to conceal changes to the underlying system infrastructure from user operating systems and applications. This was a natural progression, since the great majority of Z users already operated their systems in LPAR mode rather than as a single operating system in basic (non-LPAR) mode, even when the user was running only one instance of an operating system on a Z Platform. This decision allows the system to be a truly PR/SM-managed multiprocessor. This allows PR/SM to understand the underlying system topology of the mainframe system and make decisions in its allocation of CPU resources for logical partitions.

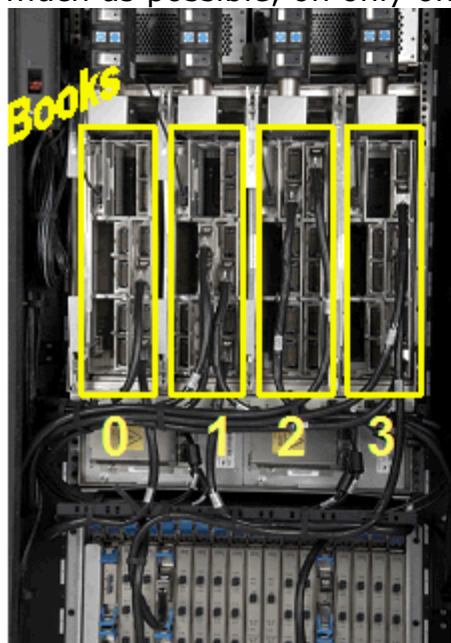
### ***Book topology of the mainframe***

At present, each book or processor housing of the z9 contains from twelve to sixteen processing units (CPUs) that are available for use, as well as some amount of installed memory. The books are numbered from 0 to 3, with the CPUs on each book using the book number as the first hexadecimal digit in their assigned CPU addresses. Each CPU contains two dedicated Level 1 (L1) caches, one for instructions and one for operands. All of the CPUs on a book share a 40-MB L2 cache. Attached to each book is up to 128GB of real memory, which is conceptually accessed through the local L2 cache for a maximum total of 512 GBs on z9. The L2 caches are connected with a bidirectional ring interconnect to allow any L2 cache to access any memory via L2-to-L2 communication.

### ***Resource Allocation to LPARs***

Since each book of a z9 system has its own L2 memory cache, processors are shared on that book. Access to local L2 is relatively rapid, however, as storage is shared by CPUs on different books, interbook cache communication is required in order to maintain coherency of the data, which takes significantly more time. Updating such storage can be particularly time-consuming, since more steps are required to maintain coherency. Optimal system performance can be achieved by minimizing the amount of data sharing by CPUs on different books. Also, access to memory is faster from a CPU on the book on which the memory resides than from a CPU of a different book. The LPAR hypervisor attempts to minimize multiplebook accesses of the same piece of memory as well as any off-book memory accesses. Since a given range of memory is

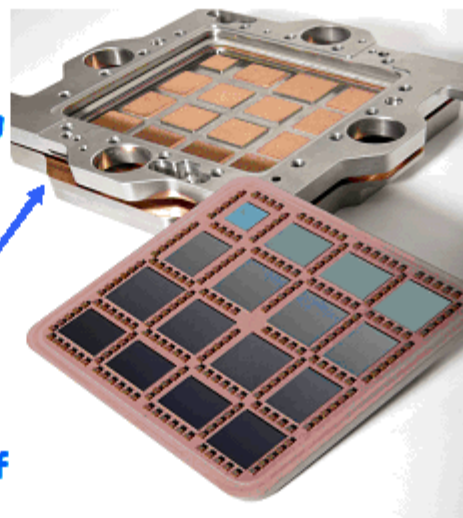
owned (accessible) by only one logical partition, multiple-book access can be minimized by having the logical CPUs of the owning logical partition dispatched, as much as possible, on only one book.



The past Z990 Central Processor Complex (CPC) introduced a packaging concept known as a Book.

Books contain processor housings containing the CPUs.

Books plug into cards, which plug into slots of the CPC cage board.



A logical partition is provided access to a contiguous range of absolute addressable memory. The allocation of physical memory to a logical partition and the establishment of physical CPU affinity for its logical CPUs should result in a configuration in which cross-book cache interrogation and multiple-book access to a given piece of memory are minimized. A given range of addressable memory increments is owned by only one logical partition, so performance is optimized by dispatching the logical CPUs of the owning logical partition on as few books as possible. Though this *could* be achieved by limiting dispatch of the logical CPUs of a shared logical partition to one "preferred" book, we would lose the ability to dispatch all of them simultaneously if the number of available physical CPUs on the book was less than the number of online logical CPUs.

### ***Dispatching decisions on the Mainframe***

Physical CPU selection by the LPAR hypervisor dispatcher is controlled by affinity masks. Each logical CPU has an affinity mask, which represents the candidate physical CPUs on which the logical CPU can run. This mask is called the *global affinity mask*, since it represents the complete set of physical CPUs on which the logical CPU can run. These masks take into account any asymmetrical features a processor might have, physical CPUs that have been dedicated to logical CPUs, and special types of CPUs (such as ICFs, zIIPs, or zAAPs).

For the mainframe each logical CPU is assigned a primary home book. In addition, a second, primary book affinity mask is established for each logical CPU with the candidate physical CPUs on the assigned home book. Both the global and the primary book affinity masks are created as a result of various resource allocation algorithms. For a *dedicated* logical CPU, the global and the primary book affinity masks are equal and contain only one physical CPU. For a *shared* logical CPU, the primary book affinity mask contains all of the physical CPUs from *as an example from* book 2 and the global affinity mask contains all of the physical CPUs from books 2 and 3. The dispatcher selection process is then modified to utilize this primary book information.

The **first** choice for assigning a logical CPU is to an idle physical CPU (i.e., a physical CPU in wait state). The available physical CPU must be a valid candidate for the logical processor. Prior to the advent of the z990 system, the hypervisor first sought to match the global affinity mask of the logical processor with the idle physical CPUs. If the physical CPU on which a logical CPU was last dispatched was currently available *and* no other logical CPU

had since been dispatched on that physical CPU, the logical CPU will be assigned to that same physical CPU.

If it is determined that the last physical CPU on which the logical CPU was dispatched is *not* available, a search is made of the remaining idle candidates for the least-recently dispatched physical CPU. The least-recently dispatched algorithm is used in an attempt to choose a physical CPU that is least likely to be a "good choice" for some other logical CPU that may soon request assignment.

The **second** change to the assignment algorithms for the z9 system is in this least-recently-dispatched search. In the event that a logical CPU has migrated away from its home book because of previous conditions and contention, where should it be dispatched next? Is the best approach to immediately try to bring home that logical CPU, or to attempt to make use of the residual L2 cache on the book to which the logical CPU migrated? The notion that the migration would have taken place because of contention on the home book may, in itself, indicate that the home book was overallocated for the actual assigned workload(s).

Initially, it appeared that the best course of action to obtain a performance benefit would be to attempt to find a physical CPU on the book to which the logical CPU was last dispatched, even if this is not the home book of the logical CPU. This would dynamically make use of the resources to which the logical CPU was last dispatched. However, the best result is to return to the home book as quickly as possible. Thus, the net change for zSystem is that the search for the least-recently dispatched idle physical CPU was modified to search one of the following three lists:

1. A list of idle candidate physical CPUs on the home book of the logical CPU.
2. A list of idle candidate physical CPUs on the same book to which this logical CPU was last dispatched.
3. A list of all idle candidate physical CPUs on the machine.

**Finally**, the dispatcher must deal with situations in which no idle candidate physical CPUs are available for selection. Basically, if no physical CPUs are idle, the lowest-priority dispatched unit of work (logical CPU) on a candidate physical CPU is sought. The displaced unit of work must be of a lower priority than the unit of work being dispatched. Although PR/SM makes the best choice possible when allocating resources to a logical partition, there are times when a less than optimal solution is all that can be achieved, primarily due to resource fragmentation.

If you have question for John, send him an email at: John H Kettner/New York/IBM.

For a direct link to the Z Project Office Website Click

Here: <http://w3.ibm.com/sales/competition/compdlib.nsf/weball/9CF22F2F65E016398725>;  
OpenDocument

---

**Document  
details**

**Date:** Mar. 7,  
2007

**Content  
owner:**  
[John H Kettner](#)

For IBMers  
only. Not  
intended for  
business  
partners or  
customers.

[Report  
handling](#)

[guidelines](#)

**Content  
provider:**

[IBM SWG  
Competitive  
Project Office](#)

[Feedback](#)



[Tag and  
save this](#)

page to your

[dogear favorites.](#)

[What is dogear ?](#)