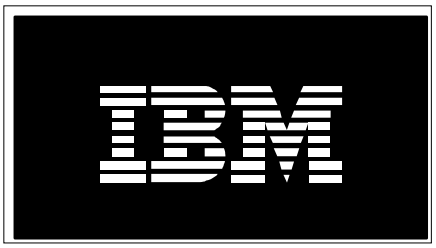

Under the Hood: What's This Multi-Core Computing Really?

January 28, 2010

Disclaimer – What's This Multi-Core Computing Really?

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This information may include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or programs(s) at any time without notice. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.
THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. IBM is not responsible for the performance or interoperability of any non-IBM products discussed herein.

The performance data contained herein was obtained in a controlled, isolated environment. Actual results that may be obtained in other operating environments may vary significantly. While IBM has reviewed each item for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere.
Statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

What's This Multi-Core Computing Really?

If you've been following the trade press over the last couple years, you've probably gathered an inkling that some sort of paradigm shift in computer design - called Multi-Core Computing - was underway. Many of the articles seem to suggest that considerable investment in new software would be required to take advantage of this change. Relax. The sun won't be suddenly setting in the east. You've been dealing with most of what's implied by Multi-Core Computing for some time now. There are some differences associated with "multi-core" and we'll be going over that here, but you already know that evolutionary change in computing paradigms is like a fact of life.

For starters, when was that last time that IBM provided a PowerPC-based computer which had exactly one processor in a system? Right. Indeed, the three pre-POWER7 processor designs have had two processor cores on each chip; POWER7 has eight. You've been dealing with multi-core SMPs (Symmetric Multi-Processing) for a long time now. Many of your systems have multiple of such chips, already making rather large SMPs. It also happens that in the two pre-POWER7 processor designs each of the processor cores have supported 2-way SMT (Simultaneous Multi-Threading), a capability where each core can concurrently execute the instructions stream of multiple tasks; POWER7 supports 4-way SMT. The point is that multiprocessing and all of the software concepts that go along with it have been around for quite a while and you have been using it. Certainly a lot of the programs produced are largely single-threaded in nature, but you've still been able to use all of the SMP cores and the SMT hardware threads you've paid for.

So why all the alarm calls related to this multi-core computing? Let's start with the basics. In pre-POWER7 PowerPC-based systems, an eight-core system required the purchase of a system with four processor chips. In POWER7-based systems, that same 8-way SMP requires only one chip. Whether you wanted - and as important, wanted to pay for - the compute capacity of one through eight cores, it's all there with this one chip; from a physical hardware point of view, the cost is the same. If you want more capacity than that, you get another chip with up to another 8 cores. But even though the hardware capacity is right there in your system, there is no requirement for you to use it. In fact, you aren't necessarily even spending money for the energy of this excess capacity; the system will effectively shut down the unused cores. More on all this later.

There is, though, one thing about computing performance that we've gotten used to over a lot of years and that is changing. What's the one statistic you've watched most when buying your home computer. Frequency. Higher has always been better, right? And IBM has agreed, providing processors with ever higher frequencies, most recently with POWER6's 5 GHz processors. But it is also true that the amount of heat that these processors generate - and must be cooled - increases faster than the frequency. It doesn't take long to realize that that relationship is not sustainable. Frequency will increase in the future, but simply not at the historical rates. If business plans have been based on the assumption that the previous rate of frequency increase will continue, some change to these assumptions is required.

But as important as frequency is to computer performance, it is not and has not been the end all, whether for system capacity or for single-threaded performance. IBM has rather proven that

with its POWER7 design. In a complex computer system, there is a lot of processing that goes on which executes at rates independent of core frequency and are also critical to performance. Even though the frequencies of POWER7 are lower than the POWER6 processors, the core-to-core capacity of corresponding POWER7 systems exceeds that of POWER6 by a considerable amount. Even many applications where there is a single task executing on a core find that POWER7 produces superior results over that of POWER6. But let's be up front here; frequency does matter. There are classes of single-threaded applications in isolation where the advantages of POWER7's design remain insufficient to make up for the difference in frequency; for some uses POWER7 is slower.

And, again, in the future frequency will not be improving at historical rates. But there does seem to remain an expectation that computer performance will improve over time. So computer designers have used in POWER7 - and will be using - all of the other knobs and new inventions also historically used to improve performance.

For example, even for true single-threaded performance, cache design and cache topology has mattered big time for a lot of years now. Most of us know that there is such a thing as cache, but that's about it, right? But without it, the high frequencies of the cores would be largely irrelevant. When a cache miss requiring an access of memory can take many hundreds approaching thousands of processor cycles, finding a way to save even one cache miss improves performance by as much as saving hundreds of instructions. Complex cache designs have proven to be as important, sometimes more so, as the rates at which instructions execute in the processor cores. Indeed, for a lot of cases, where applications need to be made to execute faster, optimizing data structures with knowledge of the cache can often buy more than focusing on just path length. Cache is that important. And, of course, POWER7's cache design makes some important advances.

Squeezing more cores - and their cache - onto the same chip as is done in POWER7 provides a further benefit relative to cache miss latencies. It is often the case that one thread - Thread A - executing on a chip's core - Core 1 - will have filled some needed data into Core 1's cache prior to the point where a Task B on a the same chip's Core 2 wants access to it. The needed data, being already in a cache local to the chip, can then be accessed more rapidly; this is relative to the access latency from some memory or from the cache of another chip. For what it is worth, IBM i's kernel and the hypervisor know of this effect and attempt to optimize to it.

In fact, POWER7 builds on this advantage still further. I'd need to get into some esoteric cache management concepts - in this case called "lateral cast-out"- to really explain, but suffice it to say that POWER7 cache design attempts to store older data into the cache of other cores on the same chip in order to keep the data in the chip's cache just a little bit longer. Again, these are cache-related advances producing better performance.

POWER7 also adds more performance capacity to each core through the support of 4-way SMT. Describing SMT (and SMT4) more fully is the subject of another article, but what the notion of SMT builds on is that there is a lot more performance capacity available for executing instructions in a core than is ordinarily used by a single thread's instruction stream. For POWER7 the design is such that there is often enough capacity available for the core to

concurrently execute the instruction streams of four threads. As a rule of thumb, depending upon the type of workload, SMT4 provides roughly 1.5-2X more throughput over that provided by a single thread executing alone on a core.

Still, while in SMT4, the four tasks are competing for the common resources of each core, one such resource being the L1 instruction and data caches. Partly due to this competition for the L1 cache(s), POWER7 adds a new level of core-local cache considerably larger than the L1 caches and accessible in only a very few cycles. It's worth noting that when there happens to be fewer - say one - tasks executing on that core, this new cache provides faster access to still more data as well.

Realizing that this resource contention is occurring, providing more throughput but also slowing the execution speed of each component thread, the cores can be individually and automatically reconfigured to provide all of a core's resource to as few as a single thread. The OS knows of SMT's performance characteristics and does arrange for threads to execute alone on cores when system utilization allows it.

All this and more allows POWER7 to meet the ever-increasing performance expectations, but in perhaps what some might consider nontraditional ways.

The basic premise of this article has been that there is nothing particularly revolutionary about "multi-core computing". These are all still PowerPC processors, capable of executing the programs you've run on preceding systems. You've also seen that there are some evolutionary differences, some perhaps quite novel. But there are some differences, some that might not be showing through immediately. We'll look at one such next.

There are with POWER7 up to four times as many cores on each chip as POWER6. It follows that there is considerably more capacity per chip than any preceding processor designs. Even your low end system is going to have at least one of these chips. As mentioned earlier, this number of cores is physically in your system whether used or not. Today you might not need that much and you certainly don't want to pay for what you don't need. So consider three concepts, again directly related to what you are used to.....

1. **Physical Core Count ...** This is merely the number of physical chips multiplied by the number of physical cores on that chip. Each additional chip provides some number of additional cores. They may or may not be used.
2. **Active Core Count** ... Here we need to describe the notion of CUoD (Capacity Upgrade on Demand). Some systems - at this writing the low end systems - do not support CUoD; instead all the physical cores in the system are also Active Cores. Other systems do support CUoD. Here, whether for one partition or many dedicated- or shared-processor partitions, you get to specify the amount of capacity in terms of cores that you really want. This automatically translates into the activation of some particular physical cores equal to the needed capacity. Active cores also have their L3 cache enabled. The remaining inactive cores and their cache are left in a very low power state. They are there as needed, but these - perhaps temporarily - inactive cores do not execute instructions.

3. **Licensed Processor Count** ... Each partition - each instance of an OS - specifies a number of virtual processors. These virtual processors execute only on active cores. For dedicated partitions, this corresponds exactly to a number of cores, cores which are intended to execute instructions. Similarly, for shared-processor partitions, the capacity of the shared-processor pool is represented in terms of cores. So the licensed processor count is the core capacity requirements of all dedicated-processor partitions plus the core capacity of the shared processor pool. This count can be less than or equal to the active core count. Because of CUoD, the two are likely to remain relative close. However, for systems not supporting CUoD (where all cores and cache are active), the number of cores where instructions are intended to execute may often be considerably fewer than the number of active cores. Wherever a chip has more active cores than licensed cores, the L3 cache of that chip remains available for use by the cores executing instructions. For example, any chip with eight active cores and four cores enabled to execute instructions, also has an extra 16 Mbytes of L3 cache also available for their use (over that of the four core's 4M L3 per licensed core).

You've paid for the hardware as some number of processor chips, each chip having direct access to some part of the system's memory DIMMs (and indirect access to all memory). You've also paid for core activation (with CUoD) and for the OS licenses based on core capacity. So it is quite possible that the number of cores that you intend to use will be considerably fewer than the number of cores physically in the system. So, yes, multi-core computing might be providing the potential for considerably more capacity, but perhaps today you have simply decided not to use it nor pay for what you are not using. The relatively new news is that very considerable additional compute capacity is there for your use if you want it. Of course, using it in the future implies that the workload you envision for it is capable of using all of the additional cores.

Building upon this notion - of having fewer cores licensed than active - is an additional capability provided outside of the low end systems. Suppose that you find yourself with an average of four or fewer OS licensed cores per 8-core chip. If these happened to really have been allocated as four or fewer OS-licensed cores per chip, it would be simpler to cool such a chip than if the chip had 8 cores executing instructions; such a chip is simply consuming less power. As a result, some classes of systems provide a notion called "TurboCore" where the frequency can be boosted upward simply because the chips can be cooled. Further, in these systems, all of the L3 cache of "TurboCore" also active. As a result, the fewer OS-licensed cores see both a larger amount of cache and a higher frequency.

As a conclusion, with POWER7's multi-core chips, more compute capacity is being provided via more processor cores, as well as advanced processor core and cache design. But that does not mean that the world of computing has changed today in some revolutionary way. Yes, POWER7 processor frequencies are less than that provided by POWER6 processors, but that it only one part of what provides for both capacity and single-threaded performance. Performance and the potential for very considerable improvements in system capacity is being provided by POWER7, but largely only if you really want it.

IBM Corporation
Systems and Technology Group
Route 100
Somers, New York 10589

Produced in the United States of America
February 2013