Newsletter of the
Society of HPC Professionals

Issue 01 Number 01 Page 1
March 2012

Memory Access for MultiThreaded Codes

Bill Menger

So why is it that software never works the way you think it should? Perhaps your code is trying too hard to avoid doing mathematics by using lookup tables or by accessing pre-calculated results. Back in the old days, I recall one of our PhDs at Conoco writing a quick lookup-table version of the square-root function for our Cray. It was faster than the supplied Cray library because it could calculate the answer from combinations out of the lookup table. Well, more recently I've seen some software that ran slower — perhaps by 4X — by trying to use precalculated results from a lookup instead of doing the math all over again in a tight loop. We need to think about why something like this could occur. The answer is in the title — it is the expense of memory access! Memory channels are not just marketing gimmicks from the chip makers, but they provide the highways in and out of the CPU cores that allow our HPC codes to run.

As part of an effort at work, I ran a sequence of tests using OpenMP on a dual-socket 6-core HPC node. It was very interesting to me to see how the code would run when trying different numbers of threads and different methods of using OpenMP. I have to give Brad Nemanich of Texas Multicore Technology the nod for making me think about this and giving me the reasons to consider looking at OpenMP with a more critical eye. The test ran a semblance calculation on about 230 Mbytes of trace data. I used gfortran and the gnu OpenMP libraries. Subsequent tests with both Intel and PGI compilers run much faster, partly because their NUMA and OpenMP libraries are well written. In order to see how memory channels affected the codes, I tried three different affinity tests for the code and modified the number of cores being accessed from 1 to 12. (see table).

CALENDAR

SAVE THE DATE — The 2012 Fall Technical Meeting is scheduled for Wednesday, October 17th in Houston!

Monthly Lunch Meetings start April 11th and continue the second Wednesday of each month. The Lunch meetings will be held at UDI and each month will cover a new topic.

Be on the lookout for the next Webinar. An email announcement will be sent.

THE SOCIETY OF HPC PROFESSIONALS

THE SOCIETY
OF
HPC PROFESSIONALS

10690 Shadow Wood Drive
Suite 132
Houston, TX 77043-2840 USA

http://www.hpcsocity.org/
info@hpcsociety.org

GOLD SPONSORS

AMD The future is fusion · intel · QLOGIC · UDI · WesternGeco

SILVER SPONSORS

BROCADE · DataDirect NETWORKS · Microsoft · panasas

BRONZE SPONSORS

APPRO · ARISTA · BLUEARC · Bright Computing · IBM · NEXENTA · PCPCDirect

FORCE10

Newsletter of the
Society of HPC Professionals

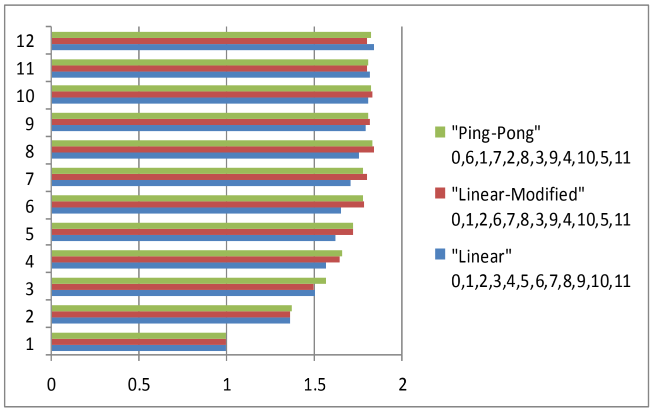
Issue 01 Number 01 Page 2
March 2012

Memory Access for MultiThreaded Codes Continued

Bill Menger

Run times in seconds shown in light blue		CHIP-0						CHIP-1					
AFFINITY	NCORES ==>	1	2	3	4	5	6	7	8	9	10	11	12
"Linear"	0,1,2,3,4,5,6,7,8,9	9.86	7.21	6.59	6.2	6.04	5.94	5.75	5.61	5.48	5.39	5.4	5.32
"Linear-Modified"	0,1,2,6,7,8,3,9,4,10,5,11	9.86	7.21	6.5	5.96	5.68	5.51	5.46	5.34	5.38	5.35	5.39	5.51
"Ping-Pong"	0,6,1,7,2,8,3,9,4,1	9.85	7.18	6.32	5.87	5.76	5.52	5.55	5.36	5.42	5.36	5.42	5.38
"Linear"	0,1,2,3,4,5,6,7,8,9	49.08	36.06	32.5	31.59	30.25	29.78	28.81	28.07	27.47	27.37	27.03	26.79
"Linear-Modified"	0,1,2,6,7,8,3,9,4,10,5,11	49.26	36.08	33.01	29.82	28.62	27.39	27.25	26.7	27.09	26.76	27.49	26.94
"Ping-Pong"	0,6,1,7,2,8,3,9,4,1	49.1	35.66	31.04	29.82	28.34	27.57	27.42	26.92	27.11	26.93	27.18	26.93

It seems that the memory chunk for this code was residing on the first of the memory channel sets closest to chip-0. In the table you can see the three different affinity models used. The Linear model lights up chip 0 first, successively using core 0, then core 1, etc, moving to chip-1 after using all 6 cores on chip-0. The Linear-modified does the same but only lights up the number of cores per chip equal to the number of memory channels (3 in this case), and the "ping-pong" model uses one core per chip successively until all the cores are full. The Graph shows relative speedup using the three methods. From 1 core to 2 cores gave the best speedup ratio, and after about 6 cores, the code speedup remains flat. In other words, I could have just left the other 6 cores idle or run on a smaller node. My initial assumption was that the ping-pong method would bring about better speedups using fewer cores, but the differences are slight between affinity models. This tells me that the traces were likely in proximity to chip-0 and so even using the second chip was a waste of effort. While not definitive, I hope this small study sparks some interest in opening dialogue about using threads, affinity, or even differences in compilers. Happy computing!



Additions to the Board

New board members Scott Denham and Steve Hebert were voted in at the December 2011 board meeting. We wish to thank Gary Crouse who acted as Executive Director until the end of last year. The board members and their profiles are available on the Society website.

New Officers for 2012

Elections for 2012 are complete and the new officers for the society are as follows:

President: Earl J. Dodd
Vice President: Susan Baldwin
Treasurer: Scott Denham
Secretary: Steve Hebert

We wish to thank our past officers for their volunteer efforts in establishing the society and helping it to function. The officers will be elected each year from a slate of nominees provided by the Board of Directors. To serve as an officer you must be a paying member of the society. Dues are $40 per year.

THE SOCIETY OF HPC PROFESSIONALS

THE SOCIETY
OF
HPC PROFESSIONALS

10690 Shadow Wood Drive
Suite 132
Houston, TX 77043-2840 USA

http://www.hpcsocity.org/
info@hpcsociety.org