

2012-03-19 Mon

Contents

1	What do we mean by thread?	2
1.1	Thread of execution in program	2
1.2	Operating system thread	2
2	Are the SPEC CPU2006 benchmarks multi-threaded?	2
2.1	Supporting documentation	2
2.2	What is the SPEC CPU2006 configuration?	3
2.3	SPEC configuration options for number of CPUs and cores?	3
3	CoreHopping	4
3.1	Citation details	4
3.2	Benchmarks used:	4
3.3	Test environment	4
3.4	Benchmark mix	4
4	Predict-and-act :: Temperature aware dynamic workload scheduling in multsocket CPU servers	5
4.1	Citation details	5
4.2	Test environment	5
4.3	Workloads:	5
4.3.1	Starting point	5
4.3.2	Results	6

1 What do we mean by thread?

1.1 Thread of execution in program

- Compiler optimization
- Requires the use of compilers that understand how to do this optimization and work with some sort of threading library like Intel's Thread Building Blocks (Intel compiler) or MPI.

1.2 Operating system thread

- Scheduling unit in operating system
- Used to be called "Lightweight process" in Unix
- Applications have to use some sort of threading directive such as POSIX pthreads (as used in PARSEC) to implement parallel algorithms.

2 Are the SPEC CPU2006 benchmarks multi-threaded?

CONCLUSION Individual SPEC CPU2006 benchmarks are single-threaded serial implementations.

IMPLICATION Prior work reporting "multi-threaded" results using these benchmarks are in fact reporting on workloads composed of:

- multiple copies of the same benchmark running at the same time
- multiple benchmarks ran at the same time (approach taken in our paper)

2.1 Supporting documentation

- From the CPU2006 web site (<http://www.spec.org/cpu2006/docs/config.html>) "Note: the SPEC CPU2006 benchmarks do not require multi-threading, and do not contain threading directives; but a compiler is allowed to try to automatically decompose threads, and there's nothing in the run rules to prohibit you from linking with your preferred version of a supported malloc library."

- The shell scripts provided by SPEC to execute these benchmarks provide what is called a “rate” metric. Again from SPEC CPU2006 web site (<http://www.spec.org/cpu2006/Docs/readme1st.html#Q15>) “For the SPECrate metrics, multiple copies of the benchmarks are run simultaneously. Typically, the number of copies is the same as the number of CPUs on the machine, but this is not a requirement. For example, it would be perfectly acceptable to run 63 copies of the benchmarks on a 64-CPU machine (thereby leaving one CPU free to handle system overhead).”
- Note the issues with the rate metric
 1. Multiple copies of the same benchmark
 2. Note that SPEC considers any runs generated by parallelizing compiler as a “speed” run, not a “rate” run.

2.2 What is the SPEC CPU2006 configuration?

- The SPEC benchmarks use the GCC compiler.
 1. It does not any sort of automatic thread decomposition. Therefore, any sort of multithreading done in this case will require to use of threading directives.
 2. The benchmarks are built using the “-o2” option which only does very simple optimizations such as instruction scheduling. In particular, it does not any form of loop-rolling

2.3 SPEC configuration options for number of CPUs and cores?

- Use of the “CPU” terminology is deceptive, it maps to what we are calling “logical cores” in our work.
- These configuration options appear to be used for two purposes
 1. Computing the total number of logical CPUs
 2. Reporting the hardware configuration in results.

3 CoreHopping

- Authors report similar results to ours on single benchmarks! “Figure 4b shows that the temperature differences are consistent over all the experimented benchmarks from CPU2006 suite. On average, the peak core temperature is 4 degC higher on core 1. Simultaneous multi-threading causes further increase in the on-core temperatures.”
- NOTE: use of term “simultaneous multi-threading” in this quote is NOT talking about application multi-threading but processor virtualization similar to Hyperthreading in the Intel architecture.

3.1 Citation details

@inproceedings{Kursun2008, Author = {Kursun, Eren}, Booktitle = {Proc. of the 2008 IEEE International Conference on Computer Design}, Date-Added = {2011-07-13 16:35:38 -0500}, Date-Modified = {2011-07-13 16:35:38 -0500}, Doi = {10.1109/ICCD.2008.4751874}, Isbn = {978-1-4244-2657-7}, Month = oct, Pages = {280-285}, Publisher = {IEEE}, Title = {Variation-aware thermal characterization and management of multi-core architectures}, Url = {http://dx.doi.org/10.1109/ICCD.2008.4751874}

3.2 Benchmarks used:

- Daxpy, Gromacs, Hmmer, Libquantum, Mcf, Milc, Namd, Perlbench

3.3 Test environment

- CPU: 1 Quad-core POWER5 processor
- OS: Linux 2.4 kernel

3.4 Benchmark mix

- No extensive detail provided about the exact mix
- Statement: “Figure 11 shows an example case illustrating the reduction in maximum chip temperature through variation-aware scheduling of hot threads from CPU2006 (for example, namd) and cold threads (for example, hmmer)”
- Statement: “The right bar (again referencing Figure 11 in paper) shows the proposed scheme can reduce peak temperatures by 4.5 degC”.

- This was reported on peak temperatures of 60-74 degC on testbed processor.

4 Predict-and-act :: Temperature aware dynamic workload scheduling in multisocket CPU servers

NOTE authors recently published a journal paper with results on a simulated version of the Intel Xeon.

- These results are more in line with what was observed on our testbed.

4.1 Citation details

@article{Ayoub2011, Author = {Ayoub, R. and Indukuri, K. and Rosing, T.S.}, Doi = {10.1109/TCAD.2011.2153852}, Issn = {0278-0070}, Journal = {IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems}, Keywords = {control theoretic approach;cooling costs;dynamic thermal management;dynamic workload scheduling;energy savings;fan subsystems;multisocket CPU servers;multitier approach;temperature frequency spectrum;thermal distribution;dynamic scheduling;fans;microprocessor chips;power aware computing;power consumption;processor scheduling;resource allocation;temperature distribution;}, Month = {Sept.}, Number = 9, Pages = {1359 -1372}, Title = {Temperature aware dynamic workload scheduling in multisocket CPU servers}, Volume = 30, Year = 2011, Bdsk-Url-1 = {http://dx.doi.org/10.1109/TCAD.2011.2153852}

4.2 Test environment

- CPU: 2 Quad Core Xeon E4440 processor (2 chips, 8 cores)
- Simulation environment HotSpot simulator using Xeon floorplan

4.3 Workloads:

4.3.1 Starting point

- Conference paper (quoted in current draft) used SPEC CPU2006, journal paper use SPEC CPU2000.
- Workload combinations used in their test

- Table notation: <number of copies><benchmark>
- For example, 3eon says 3 copies of eon benchmark

Workload	CPU Socket A	CPU Socket B
W1	3eon	eon + mcf + gcc
W2	2eon + mcf	eon + bzip2 + mcf
W3	2bzip2 + 2mcf	2bzip2 + 2mcf
W4	2perl + 2eon	2gcc + 2mcf
W5	2perl + 2bzip2	2gcc + 2mcf
W6	2perl + 2bzip2	2gcc + 2mcf
W7	2perl + bzip2	gcc+2mcf
W8	perl + 3gcc	perl + 2gcc
W9	perl + 3gcc	perl + 2gcc
W10	perl + 3gcc	perl + 2gcc
W11	perl + 3gzip	perl + gcc + gzip
W12	3eon	3eon
W13	2eon+mcf	2eon+mcf
W14	2bzip2	2bzip2

4.3.2 Results

- 3-4 degC savings on temperatures ranging from 36 degC to 42 degC
- Single digit saving in CPU energy savings, in many cases significantly less than 5% per figure where data is reported
- Note that work focused on reducing energy consumption due to cooling costs (particularly fans) so ways results are presented in this paper can be very deceptive as to their actual results.